A Single Visual-Servo Controller of Mobile Robots with Super-Twisting Control

H. M. Becerra[1], J. B. Hayet[1] and C. Sagüés[2]

[1] *Centro de Investigación en Matemáticas (CIMAT), C/ Jalisco S/N, Col. Valenciana. C.P. 36240, Guanajuato, Gto., Mexico. E-mail:* (hector.becerra, jbhayet)@cimat.mx

[2] *Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, C/ María de Luna 1, E-50018, Zaragoza, Spain. E-mail:* csagues@unizar.es

Abstract

This paper presents a novel approach for image-based visual servoing, extending the existing works that use the trifocal tensor (TT) as source for image measurements. In the proposed approach, singularities typically encountered in this kind of methods are avoided. A formulation of the TT-based control problem with a virtual target resulting from the vertical translation of the real target allows us to design a *single* controller, able to regulate the robot pose towards the desired configuration, without local minima. In this context, we introduce a super-twisting control scheme guaranteeing continuous control inputs, while exhibiting strong robustness properties. Our approach is valid for perspective cameras as well as catadioptric systems obeying the central camera model. All these contributions are supported by convincing numerical simulations and experiments under a popular dynamic robot simulator.

Keywords:
Visual servoing, mobile robots, super-twisting control, trifocal tensor

1. Introduction

The goal of an image-based visual servoing (IBVS) scheme is to minimize a task function that is specified directly from visual data in order to control the motion of a robot by using visual feedback. Classical approaches of IBVS use points as visual features [1], given that points are the simplest geometrical primitives that can be extracted from an image. In the last years, the research community on IBVS has dedicated important efforts in order to define better task functions from different geometrical primitives (lines, moments, etc.) [2].

Geometric constraints between images have been extensively exploited for the visual control of mobile robots [3, 4, 5, 6, 7, 8]. These schemes have improved robustness and have allowed to avoid local minima problems of the classical schemes where over-constrained solutions are obtained. The homography-based control has the practical limitation of relying on planar scenes only [3, 4]. Hence, more general constraints like the ones induced by epipolar and trifocal geometries have been preferred. However, the epipolar geometry is ill-conditioned with a short baseline, and moreover, the controllers based on the epipolar constraint are subject to singularities. This has been tackled by using control schemes that switch between two different controllers when degeneracies and singularities appear [5, 6].

The present work has been inspired by [9], where the authors have suggested the use of a virtual target in order to avoid some degeneracies of the essential matrix and singularities of an epipolar visual control. In that work, the generation of a virtual target relies on the transfer relations associated with the essential matrix. However, the transfer of points can fail for certain configurations, like collinear projection centers [10]. In this work, we propose the use of the TT, which is known to be more robust and better defined than the epipolar geometry. The transfer relations associated with the TT can be used to transfer the visual content (points and lines) of two source images into the third one (a virtual target image) without degenerate cases. In the literature, the TT has been exploited for IBVS but with some limitations [7, 8]. In the former work, an over-constrained controller that may suffer from local minima problems is proposed from the TT. This is overcome in [8], by defining a square control system and by using measurements from the radial TT in a control law that switches between controllers when a singularity occurs. It is worth emphasizing that the importance of the TT for vision-based control is that it is more robust than two-view geometry thanks to the additional information of a third view, and that the problem of short baseline with epipolar geometry is overcome.

The contribution of this work is two-fold. First, the use of a virtual target from the 2D TT provides additional information that avoids the need of switching to a different controller, in such a way that a single controller, free of singularities, achieves regulation of the robot pose (position and orientation), in contrast to our previous works [6, 8], where two discontinuous controllers are used in a switching scheme. The controller is designed from a square trajectory tracking control system, free of local minima with the use of adequate desired trajectories. Some preliminary results around this contribution have been presented in a conference paper [11]. Second, for the first time to the knowledge of the authors, a super-twisting control is proposed to be used in a IBVS scheme. This recent control technique is a particular case of higher order sliding modes control [12], which results in continuous control inputs, unlike classical sliding modes control [13]. The importance of the super-twisting control for visual servoing applications is its intrinsic robustness achieved from continuous signals against perturbations. In our particular case, the visual servoing problem is transformed to a trajectory tracking problem. However, the trajectory is generated on-line and its time-derivative is unknown, which causes the appearance of a bounded perturbation in addition to the effects of image noise, so that a robust controller must be used.

An additional benefit of the proposed IBVS scheme is that it is valid for different types of cameras, in particular, those obeying the unified projection model [14]. This is a first contribution of our proposal with respect to the work [7], which exploits the 2D TT using conventional cameras. However, the main contribution over [7] is that the controller presented in this work is designed from a square control system for trajectory tracking, free of local minima with the use of adequate desired trajectories.

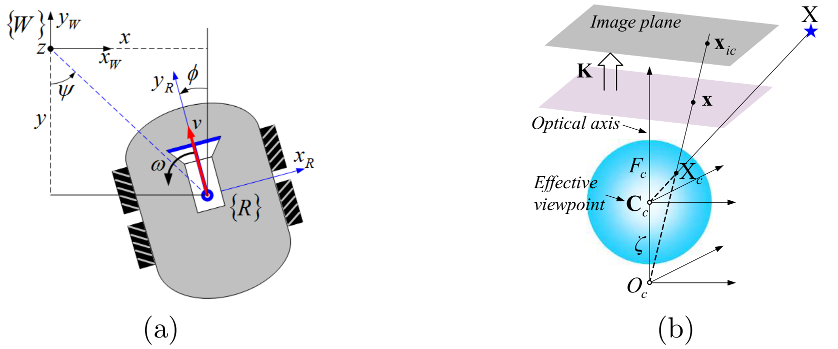


Figure 1: Representation of the robot model and the camera model. (a) Robot frame definition in the Cartesian plane, with the z-coordinate corresponding to the vertical position of the camera in the world frame. (b) Generic camera model of central cameras [14].

In [7], because of the over-constrained error function used, the computed robot velocities are a least squares solution (pseudoinverse) of a non-square error system with potential problems of local minima.

The paper is organized as follows. Section 2 describes the mathematical modeling of the robot, the camera and the visual measurements provided by the TT for coplanar camera locations. In Section 3, the generation of the virtual target relying on the TT is detailed. Section 4 describes the controller design from the TT and using the super-twisting control. In Section 5, we establish conditions for the stability of the closed loop system and the corresponding proof is presented. Section 6 shows the performance of the proposed approach through realistic simulations and experiments. Finally, Section 7 summarizes the conclusions.

2. Robot and Visual Information Models

2.1. Robot Kinematics

Let $\chi = (x, y, \phi)^T$ be the state vector of a differential drive robot shown in Fig. 1(a), where x and y are the robot position coordinates in the plane, and ϕ is its orientation. Assume that a central camera is fixed to the robot in such a way that the robot and camera reference frames coincide. The kinematic model of the camera-robot system expressed in the state space can be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\sin\phi & 0 \\ \cos\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix}, \tag{1}$$

being ν and ω the translational and angular velocities, respectively. In the sequel, the notation $s_\phi \overset{\text{def}}{=} \sin\phi$ and $c_\phi \overset{\text{def}}{=} \cos\phi$ is used. Although the robot motion is on the plane, its perception is carried out on the 3D world. Hence, hereafter, we will also make use of the full coordinates of 3D points. More precisely, we will denote the coordinates of a 3D point in the camera frame as a 3×1 matrix $\mathbf{X}$.

2.2. The Trifocal Tensor for Generic Cameras

A desirable feature for a visual control scheme is its applicability for different types of cameras, from conventional to omnidirectional ones. This way, an adequate type of camera can be chosen depending on the application. Mostly, a camera with wide field of view is preferred in order to avoid the loss of some visual features during the navigation. Furthermore, the recent literature has shown that the use of geometric constraints is a good strategy to achieve generic control schemes. This is supported by the fact that a unique representation exists to model properly all vision systems having approximately a single center of projection [14]. This unified projection model allows the computation of a geometric constraint, like the TT, in the same way for any central vision system.

According to the unified projection model, the coordinates of the point on the sphere, $\mathbf{X}_c$, corresponding to a 3D point $\mathbf{X}$, can be computed from point coordinates $\mathbf{x}$ on the normalized image plane (refer to Fig. 1(b)) and the sensor parameter ζ as follows [14]:

$$\begin{aligned} \mathbf{X}_c &= \left(\eta^{-1} + \zeta\right) \bar{\mathbf{x}}, \\ \bar{\mathbf{x}} &= \left[\begin{array}{cc} \mathbf{x}^T & \frac{1}{1+\zeta\eta} \end{array} \right]^T \end{aligned} \qquad (2)$$

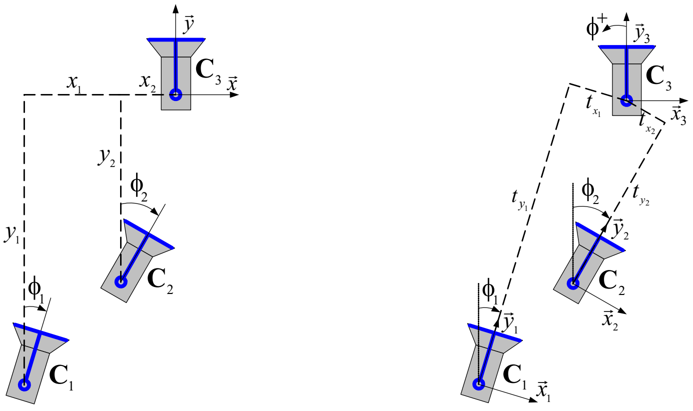
where $\eta = \frac{-\mu - \zeta \|\mathbf{x}\|^2}{\zeta^2 \|\mathbf{x}\|^2 - 1}$, $\mu = \sqrt{1 + \left(1 - \zeta^2\right) \|\mathbf{x}\|^2}$. In this work, we assume that the camera is calibrated if omnidirectional vision is used, which allows us to exploit the representation of the points on the unit sphere to estimate the TT. However, if a conventional camera is used, calibration is not needed and the TT can be computed from normalized points according to their distribution on the image [10].

The TT encapsulates all the geometric relations between three views, independently of the structure of the scene [10]. Thus, it has been used for motion estimation [15] and reconstruction [16]. The TT has 27 elements (18 of which being independent) and it can be expressed as $\mathbf{TT} = \{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$, with $\mathbf{T}_i \in \mathbb{R}^{3\times3}$ for $i = 1, 2, 3$. In this work, we focus on the use of interest points as image features. Consider three corresponding points projected on the unitary sphere $\mathbf{p}$, $\mathbf{p}'$ and $\mathbf{p}''$ and expressed in homogeneous coordinates, i.e. $\mathbf{p} = (p^1, p^2, p^3)^T$. The incidence relation between these points is given as

$$[\mathbf{p}']_\times \left(\sum_i p^i \mathbf{T}_i \right) [\mathbf{p}'']_\times = \mathbf{0}_{3\times3} \qquad (3)$$

where $[\mathbf{p}]_\times$ is the common skew symmetric matrix. This expression provides a set of nine equations, however, only four of them are linearly independent. Hence, seven triplets of point correspondences are needed to compute the 27 elements of the tensor.

Consider a framework where images are taken from three different coplanar locations, i.e., with a camera moving at a fixed distance from the ground plane. In this case, several elements of the tensor are zero and only 12 elements are in general non-null. Fig. 2 depicts the upper view of three cameras with global reference frame in the third view, in such a way that the corresponding camera locations are $\mathbf{C}_1 = (x_1, y_1, \phi_1)$, $\mathbf{C}_2 = (x_2, y_2, \phi_2)$ and $\mathbf{C}_3 = (0, 0, 0)$. The TT can be analytically deduced for this framework as done in [7], resulting in the following expressions for the non-null elements:



(a) (b)

Figure 2: Geometry between three camera locations in the plane. (a) Absolute locations with respect to a reference frame in $\mathbf{C}_3$. (b) Relative locations.

$$\begin{aligned} T^m_{111} &= -t_{x1}c_{\phi_2} + t_{x2}c_{\phi_1}, T^m_{112} = t_{x1}s_{\phi_2} + t_{y2}c_{\phi_1}, \\ T^m_{121} &= -t_{y1}c_{\phi_2} - t_{x2}s_{\phi_1}, T^m_{122} = t_{y1}s_{\phi_2} - t_{y2}s_{\phi_1}, \\ T^m_{211} &= -t_{x1}s_{\phi_2} + t_{x2}s_{\phi_1}, T^m_{212} = -t_{x1}c_{\phi_2} + t_{y2}s_{\phi_1}, \\ T^m_{221} &= -t_{y1}s_{\phi_2} + t_{x2}c_{\phi_1}, T^m_{222} = -t_{y1}c_{\phi_2} + t_{y2}c_{\phi_1}, \\ T^m_{313} &= -t_{x1}, T^m_{323} = -t_{y1}, T^m_{331} = t_{x2}, T^m_{332} = t_{y2} \end{aligned} \tag{4}$$

where $t_{x_i} = -x_i c_{\phi_i} - y_i s_{\phi_i}$, $t_{y_i} = x_i s_{\phi_i} - y_i c_{\phi_i}$ for $i = 1, 2$ and where the superscript m indicates that the tensor elements are given by metric information. In practice, the estimated tensor has an unknown scale factor and this factor changes as the robot moves. We can fix a common scale during the navigation by normalizing each element of the tensor as $T_{ijk} = T^e_{ijk}/T_N$, where T^e_{ijk} are the estimated TT elements obtained from point matches, T_{ijk} are the normalized elements, and T_N is a suitable normalizing factor. We can see from Eq. (4) that T_{313} and T_{323} are constant and non-null, assuming that the camera location $\mathbf{C}_1$ is different from $\mathbf{C}_3$. Therefore, any of these elements is a good candidate as normalizing factor.

3. A Virtual Target from the TT

In visual servoing, the set-point of the control loop is given by a target image previously acquired. In the sequel, as described in Fig. 2, $\mathbf{C}_1$, $\mathbf{C}_2(t)$, and $\mathbf{C}_3$ are respectively the initial, current (at time t) and target camera-robot location. Notice that $\mathbf{C}_1$ and $\mathbf{C}_3$ remain fixed during the robot evolution. The pose regulation problem consists in driving the robot autonomously to reach $\mathbf{C}_2(t) = \mathbf{C}_3$, where the current image observed by the camera (corresponding to $\mathbf{C}_2(t)$) is the same as the target image. On the one hand, it is numerically troublesome to estimate the TT if two images are the same and some elements must be discarded for control purposes in that case [7]. On the other hand, the use of the radial TT (first 8 expressions of Eq. (4)) has resulted in the need of a few controllers in order to accomplish the pose regulation task [8]. Inspired by [9], we have derived the equations of the TT

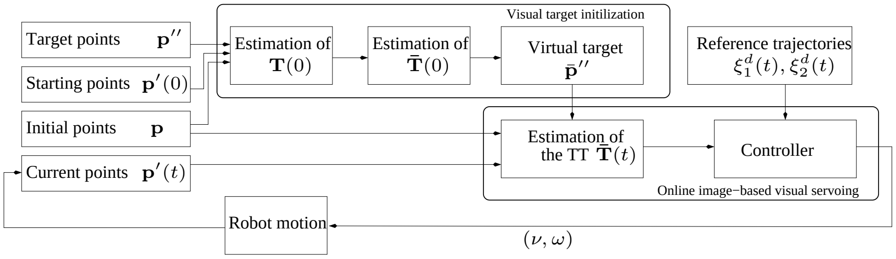

Visual target initilization
Target points $\mathbf{p}''$
Starting points $\mathbf{p}'(0)$
Initial points $\mathbf{p}$
Current points $\mathbf{p}'(t)$
Estimation of $\mathbf{T}(0)$
Estimation of $\bar{\mathbf{T}}(0)$
Virtual target $\bar{\mathbf{p}}''$
Reference trajectories $\xi_1^d(t), \xi_2^d(t)$
Estimation of the TT $\bar{\mathbf{T}}(t)$
Controller
Online image-based visual servoing
Robot motion
(ν, ω)

Figure 3: Overview of the TT-based visual servoing methodology with a virtual target.

considering a new, virtual target location $\bar{\mathbf{C}}_3$. This target location is the same as before but it is *shifted* by a vertical translation (t_z) with respect to $\mathbf{C}_1$ and $\mathbf{C}_2(t)$ as shown in Fig. 4. Thus, the *planar* geometry among $\mathbf{C}_1$, $\mathbf{C}_2(t)$ and $\bar{\mathbf{C}}_3$ stays the same as in Fig. 2 and the image at the end of the motion will be different to the virtual target image (associated to $\bar{\mathbf{C}}_3$). The benefits of using a virtual target are that any non-null element of the new tensor $\overline{\mathbf{TT}}(t)$ (corresponding to views at $\mathbf{C}_1$, $\mathbf{C}_2(t)$, and $\bar{\mathbf{C}}_3$) may be chosen for control purposes, as they are well-defined.

To implement this strategy, consider the overview depicted in Fig. 3. We define image points configurations $\mathbf{p}$ (initial) and $\mathbf{p}''$ (target) corresponding to locations $\mathbf{C}_1$ and $\mathbf{C}_3$, respectively. Then, the robot moves to a configuration $\mathbf{C}_2(0)$ from which the feedbacked control starts ($t = 0$). From this configuration, the points $\mathbf{p}$ have been tracked into points $\mathbf{p}'(0)$, and the $\mathbf{TT}(0)$ (at time 0) can be computed. We also deduce the $\overline{\mathbf{TT}}(0)$ associated to the virtual target, and, next, the virtual target position $\bar{\mathbf{p}}''$ can be estimated to be used in the control loop. Then, for $t > 0$, the control loop uses the entries of the TT computed from $\mathbf{p}, \bar{\mathbf{p}}''$ and $\mathbf{p}'(t)$ (current image) to drive the robot to the target location. Now, we detail the generation of a virtual target by exploiting the point-transfer property of the TT.

3.1. From Real TT to Virtual TT

By considering the non-planar case of camera locations, and introducing $\bar{\mathbf{C}}_3$ instead of $\mathbf{C}_3$, we have that at time 0, the $\overline{\mathbf{TT}}(0)$ (relating $\mathbf{p}, \mathbf{p}'(0), \bar{\mathbf{p}}''$) differs from $\mathbf{TT}(0)$ (as in Eq. 4, computed from $\mathbf{p}, \mathbf{p}'(0), \mathbf{p}''$) by the following elements:

$$\begin{aligned}
\bar{T}^m_{113}(0) &= t_{z_2} c_{\phi_1}, \bar{T}^m_{123}(0) = -t_{z_2} s_{\phi_1}, \bar{T}^m_{131}(0) = -t_{z_1} c_{\phi_2}(0), \bar{T}^m_{132}(0) = t_{z_1} s_{\phi_2}(0), \\
\bar{T}^m_{213}(0) &= t_{z_2} s_{\phi_1}, \bar{T}^m_{223}(0) = t_{z_2} c_{\phi_1}, \bar{T}^m_{231}(0) = -t_{z_1} s_{\phi_2}(0), \bar{T}^m_{232} = -t_{z_1} c_{\phi_2}(0)
\end{aligned} \tag{5}$$

where $t_{z_1} = t_{z_2} = t_z$, given that the global reference frame is now $\bar{\mathbf{C}}_3$. We recommend to fix the distance t_z as a portion of the normalizing factor, e.g., $t_z = 0.1 T_N$, so that, the scale factor of the real tensor is introduced in the new elements. The angles ϕ_1 and $\phi_2(0)$ (that will evolve as $\phi_2(t)$ in the control loop) can be estimated from the original tensor $\mathbf{TT}(0)$ (Eq. 4 up to scale) as follows:

$$\phi_1 = \arcsin\left(\tfrac{T_{323}T_{212} - T_{313}T_{121}}{T_{323}T_{332} + T_{313}T_{331}}\right), \;\; \phi_2(0) = \arccos\left(\tfrac{T_{332}T_{121} + T_{331}T_{212}}{T_{323}T_{332} + T_{313}T_{331}}\right). \tag{6}$$

Hence, by using both Eq. 5 and Eq. 6, we can deduce $\overline{\mathbf{TT}}(0)$.

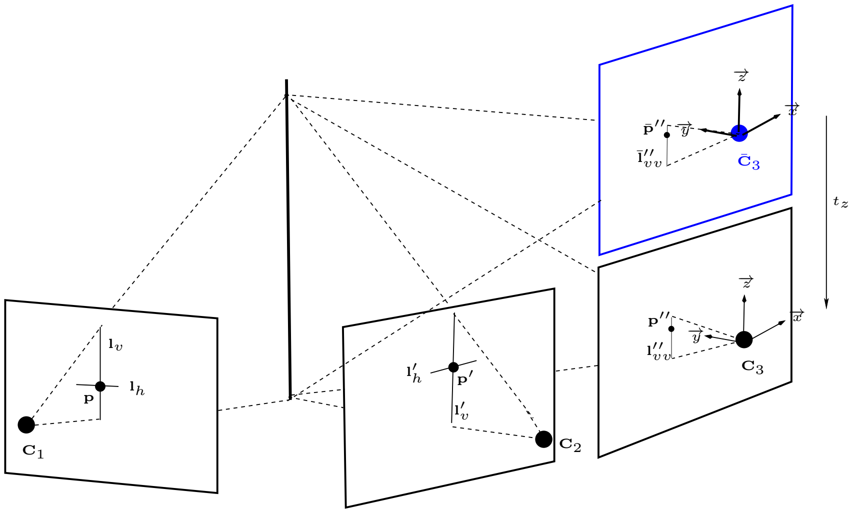


Figure 4: Setup for the generation of a virtual target image, above the real target. From the **TT** associated to $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ and an arbitrary vertical translation t_z, the $\overline{\mathbf{TT}}$ associated to $\mathbf{C}_1$, $\mathbf{C}_2$, $\bar{\mathbf{C}}_3$ can be estimated. Then, $\overline{\mathbf{TT}}$ is used for point transfer.

3.2. Generating the Virtual Target

Generating the virtual target implies transferring the projections of points seen at $\mathbf{C}_1$ and $\mathbf{C}_2(0)$ into the view corresponding to $\bar{\mathbf{C}}_3$. If $\mathbf{p}$ and $\mathbf{p}'(0)$ are two such points at $\mathbf{C}_1$ and $\mathbf{C}_2(0)$, detected by an interest point detector and matched, then observe that, in general, they do not satisfy the epipolar constraint induced by the first two views. Hence, the Eq. 3 has no solution.

A simple workaround uses the fact that the TT gives a unique transferred straight line for the third view, given any two straight lines equations in the first and second views. As illustrated by Fig. 4, consider a pair of horizontal/vertical lines $(\mathbf{l}_h, \mathbf{l}_v)$ passing through $\mathbf{p}$, and a second pair of lines $(\mathbf{l}'_h, \mathbf{l}'_v)$ passing through $\mathbf{p}'(0)$. Then, by construction, the transferred point $\bar{\mathbf{p}}''$ should belong in the third view to the straight line corresponding to any the pairs $(\mathbf{l}_h, \mathbf{l}'_h)$, $(\mathbf{l}_h, \mathbf{l}'_v)$, $(\mathbf{l}_v, \mathbf{l}'_h)$, $(\mathbf{l}_v, \mathbf{l}'_v)$, each of which being computed with $\overline{\mathbf{TT}}(0)$. Notice that we have chosen pairs of horizontal/vertical lines for simplicity but any pair of lines could be chosen, more generally, given that any pair of lines in the first and the second view define two planes and a 3D line corresponding to their intersection. In Fig. 4, the process is illustrated for the pair $(\mathbf{l}_v, \mathbf{l}'_v)$, which image is given by [10]:

$$\bar{\mathbf{l}}''_{vv} = (\mathbf{l}_v^T \bar{\mathbf{T}}_1(0)\mathbf{l}'_v, \mathbf{l}_v^T \bar{\mathbf{T}}_2(0)\mathbf{l}'_v, \mathbf{l}_v^T \bar{\mathbf{T}}_3(0)\mathbf{l}'_v)^T.$$

Each of the aforementioned 4 straight line pairs leads to a linear constraint on the homogeneous coordinates of $\bar{\mathbf{p}}''$. Hence, a linear system can be formed to determine the homogeneous coordinates of $\bar{\mathbf{p}}'' \in \mathbb{R}^3$ in a least square sense:

$$\bar{\mathbf{p}}'' = \min_{\|\bar{\mathbf{q}}''\|=1} \|\mathbf{L}\bar{\mathbf{q}}''\|^2$$

where $\mathbf{L}$ is a 4×3 matrix containing the four line equations. Classically, $\bar{\mathbf{p}}''$ is extracted as the singular vector of $\mathbf{L}$ associated to the smallest singular value.

4. A Single Controller from the TT

In the literature of visual control, switching control laws have been proposed to solve the pose regulation problem of mobile robots by exploiting a geometric constraint [5, 6, 8]. At least two different controllers with an appropriate switching policy are used in such approaches. This is done in order to deal with degeneracies of the visual measurements and singularities of the controllers. In this section, we present a single controller that is capable to drive the robot to the target pose (position and orientation). On the one hand, the controller takes advantage of the robust and well defined visual measurements provided by the trifocal tensor estimated from the virtual target, $\overline{\mathbf{TT}}$. On the other hand, an adequate pair of measurements has been chosen in such a way that no singularity appears in the computation of the robot velocities. Additionally, the robust super-twisting control is used in order to ensure a good trajectory tracking.

4.1. Input-Output Linealization

Given that the TT is an over-constrained measurement (more elements of the tensor than robot's degrees of freedom), there are several options to define two system's outputs in order to have a square control system. This is convenient to avoid the use of pseudoinversion in the computation of the robot velocities and consequently to avoid local minima. An option is to carry out the positioning task in two stages as in [8]. However, in the paper herein, we aim for controlling the three degrees of freedom of the mobile robot through a single controller, free of singularities and free of local minima. To achieve that, after an analysis of the information provided by the TT estimated from the virtual target, we have chosen the following pair of visual measurements as outputs of the camera-robot system:

$$\xi_1 = \bar{T}_{332}, \quad \xi_2 = \frac{\bar{T}_{132}}{\bar{T}_{131}}. \tag{7}$$

In the sequel, we will denote $x = x_2$, $y = y_2$ and $\phi = \phi_2$, which is the robot pose in the plane with respect to the reference frame attached to $\bar{\mathbf{C}}_3$. The chosen outputs are related to the camera-robot state as follows:

$$\xi_1 = \alpha \left(x s_\phi - y c_\phi \right), \qquad \xi_2 = -\tan \phi$$

where α is an unknown scale factor. An important consideration is that these outputs are valid in the range $|\phi| < \pi/2$, so that, we assume that the initial orientation of the robot accomplish such condition. Notice that both outputs are null if the robot has reached the target location. The invertibility of the dynamic system formed with the time derivative of these outputs has been the main reason of their choice. It can be seen that $\xi_1 = 0$ and $\xi_2 = 0$ imply that $\phi = 0$, $y = 0$ and x is a degree of freedom of the solution, which means that the orientation and longitudinal error are corrected while the lateral error may be different from zero (zero dynamics from the control theory point of view [17]). However, given that ξ_1 is related to the longitudinal position and ξ_2 depends directly on the orientation, the lateral deviation can be corrected with the tracking of an adequate trajectory for ξ_2 as the robot moves forward. It is desired to drive the outputs to zero in a fixed time horizon, which is a trajectory tracking control problem. Let us define the following error functions:

$$e_1 = \xi_1 - \xi_1^d, \qquad e_2 = \xi_2 - \xi_2^d$$

where ξ_1^d and ξ_2^d are smooth desired trajectories with null final value. The tracking problem can be faced by using the input-output linearization technique [17]. It needs the time derivatives of the error functions and consequently the time derivatives of the outputs, which are obtained as follows:

$$\begin{aligned} \dot{\xi}_1 &= \alpha\left(\dot{x}s_\phi + x\dot{\phi}c_\phi - \dot{y}c_\phi + y\dot{\phi}s_\phi\right), \\ \dot{\xi}_2 &= \frac{t_{z_1}\dot{\phi}c_\phi\bar{T}_{131} - \bar{T}_{132}t_{z_1}\dot{\phi}s_\phi}{\bar{T}_{131}^2}. \end{aligned}$$

Using the kinematic model of the robot (see Eq. 1), we have:

$$\begin{aligned} \dot{\xi}_1 &= \alpha\left(-\nu s_\phi^2 - \nu c_\phi^2 + (xc_\phi + ys_\phi)\,\omega\right), \\ \dot{\xi}_2 &= \frac{t_{z_1}c_\phi\bar{T}_{131} - \bar{T}_{132}t_{z_1}s_\phi}{\bar{T}_{131}^2}\omega. \end{aligned}$$

Knowing that $\alpha\left(xc_\phi + ys_\phi\right) = -\bar{T}_{331}$, $t_{z_1}c_\phi = -\bar{T}_{131}$ and $t_{z_1}s_\phi = \bar{T}_{132}$, the error system can be expressed as

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} -\alpha & -\bar{T}_{331} \\ 0 & -\frac{\bar{T}_{131}^2+\bar{T}_{132}^2}{\bar{T}_{131}^2} \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix} + \begin{bmatrix} \varrho_1(\dot{\xi}_1^d, \sigma) \\ \varrho_2(\dot{\xi}_2^d, \sigma) \end{bmatrix}. \tag{8}$$

This system can be written as $\dot{\mathbf{e}} = \mathbf{J}\mathbf{u} + \varrho$, where $\mathbf{J}$ is the interaction matrix that relates the robot velocities to the rate of change of the visual measurements of Eq. 7 and ϱ represents a perturbation term depending on the time derivatives of the desired trajectories and the image noise parameter σ, which is the standard deviation of a Gaussian noise. We will assume the terms $\dot{\xi}_1^d$, $\dot{\xi}_2^d$ and the image noise as unknown bounded perturbations, so that a robust controller, able to reject their effects, must be proposed. Indeed, as described later, the desired trajectory ξ_2 will be generated online as the robot moves. Consequently $\dot{\xi}_2^d$ is unknown and cannot be used as a feedforward term in the tracking controller. Notice that the perturbations fulfill the matching condition [17]; they belong to the range space of the input vector. Then, they can be rejected by using a robust controller. In the sequel, for simplicity we will denote $\varrho_1(\dot{\xi}_1^d, \sigma) = \varrho_1$ and $\varrho_2(\dot{\xi}_2^d, \sigma) = \varrho_2$.

In order to find out adequate robot velocities to track the desired trajectories, the error system must be inverted, which is possible given that:

$$\det(\mathbf{J}) = \alpha\frac{\bar{T}_{131}^2+\bar{T}_{132}^2}{\bar{T}_{131}^2} = \frac{\alpha}{\cos^2\phi} \neq 0.$$

Hence, the robot velocities are given by:

$$\left[\begin{array}{c} \nu \\ \omega \end{array}\right] = \left[\begin{array}{cc} -\frac{1}{\alpha} & \frac{\bar{T}_{331}\bar{T}_{131}^2}{\alpha\left(\bar{T}_{131}^2+\bar{T}_{132}^2\right)} \\ 0 & -\frac{\bar{T}_{131}^2}{\bar{T}_{131}^2+\bar{T}_{132}^2} \end{array}\right] \left[\begin{array}{c} \upsilon_1 \\ \upsilon_2 \end{array}\right] \tag{9}$$

where υ_1 and υ_2 are auxiliary control inputs that define the convergence of the error functions. Notice that α is a scale factor for the translational velocity and it is absorbed by the auxiliary controls with appropriate gains. The auxiliary controls are defined in the next section by using a robust control technique.

4.2. Super-Twisting Control

In previous works, we have proposed the use of classical Sliding Modes Control (SMC) for visual servoing of mobile robots [6, 8]. Such approaches deal efficiently with the occurrence of singularities due to the nature of the visual measurements used. Additionally, SMC provides the advantage of robustness against system uncertainties and perturbations, like measurement noise. Although classical SMC has shown good performance in an uncountable number of applications, like in VS, its major criticism has been the discontinuous behavior of the computed control inputs that may derive into the chattering problem [13]. In order to alleviate this drawback, the theory of Higher Order Sliding Modes (HOSM) has been recently developed [12]. Particularly, the second order sliding modes control called Super-Twisting Algorithm (STA) has been efficiently applied for mobile robots control [18, 19]. STA offers the robustness of classical SMC with continuous control inputs. Moreover, STA does not need the time derivative of the sliding surface to be measured, as demanded by other HOSM.

In the VS application of this paper, the reasons of the use of STA are two-fold: 1) In order to correct the lateral deviation from an adequate desired trajectory of the orientation, ξ_2^d must be defined as an unknown function, which introduces an unknown perturbation $\dot{\xi}_2^d$ encoded in ϱ_2. 2) The image noise generates non-deterministic perturbations (ϱ_1, ϱ_2) in the error system of Eq. 8 and they must be rejected. These issues can be overcome by defining the auxiliary inputs of Eq. 9 using the STA as follows:

$$\begin{array}{rcl} \upsilon_1 & = & -k_1 \left|e_1\right|^{\frac{1}{2}} \operatorname{sgn}(e_1) + u_1, \\ \dot{u}_1 & = & -k_2 \operatorname{sgn}(e_1), \\ \upsilon_2 & = & -k_3 \left|e_2\right|^{\frac{1}{2}} \operatorname{sgn}(e_2) + u_2, \\ \dot{u}_2 & = & -k_4 \operatorname{sgn}(e_2) \end{array} \tag{10}$$

where k_1, k_2, k_3, k_4 are control gains that must accomplish the conditions given in the stability analysis section. It is worth emphasizing that the auxiliary control inputs υ_1 and υ_2 are continuous as analyzed in [12]. In order to clarify this fact, υ_1 is plotted for e_1 in the interval [-1,1] in Fig. 5. We have used $u_1(-1) = 0$ for numerical integration, $k_1 = 0.1$ and k_2 takes different values, so that, u_1 has different effects on the behavior of υ_1. It can be seen that the auxiliary control υ_1 is continuous, similarly υ_2, and it does not depend on the control gains.

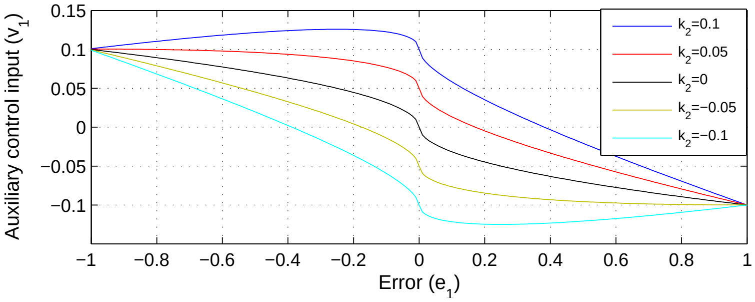


Figure 5: Plot of the auxiliary control input v_1 given by the super-twisting control for different control gains k_2. The resultant signal v_1 is continuous, similarly for v_2

4.3. Desired Trajectories

The desired trajectories must drive smoothly the outputs of the system from their initial values to zero in a fixed time horizon. Thus, the desired trajectory for ξ_1 is always defined as follows:

$$\begin{aligned} \xi_1^d &= \frac{\xi_1(0)}{2}\left(1+\cos\left(\frac{\pi t}{\tau}\right)\right), & 0 \le t \le \tau \\ \xi_1^d &= 0, & t > \tau \end{aligned} \tag{11}$$

where τ is a user-defined time horizon in which the robot reaches the target location. Given that the orientation control also must drive the robot to correct the lateral deviation, a desired trajectory ξ_2^d related to x is proposed. Let us define the following angle that can be seen in Fig. 1(a), which depends on the lateral deviation:

$$\psi = \arctan\left(-\frac{\bar{T}_{332}s_\phi - \bar{T}_{331}c_\phi}{\bar{T}_{332}c_\phi + \bar{T}_{331}s_\phi}\right)$$

where ϕ is given by Eq. 6. According to the initial value of the angle ψ in comparison with the value of the initial orientation ϕ, the desired trajectory for ξ_2 is defined in two different forms. If $|\phi(0)| > |\psi(0)|$, then:

$$\begin{aligned} \xi_2^d &= \xi_2(0)\frac{\psi(t)}{\psi(0)}, & 0 \le t \le \tau \\ \xi_2^d &= 0, & t > \tau. \end{aligned} \tag{12}$$

Notice that the condition $|\phi| > |\psi|$ means that the robot is oriented in such a way that lateral deviation, and consequently $\psi(t)$, tends to decrease as the robot moves forward. On the contrary, and excepting $\phi(0) = \psi(0) = 0$, if $|\phi(0)| \le |\psi(0)|$, the desired trajectory for ξ_2 is defined in two main segments as follows:

$$
\begin{aligned}
\xi_2^d &= \tfrac{\xi_2(0)+\xi_2^*(\tau_1)}{2} + \tfrac{\xi_2(0)-\xi_2^*(\tau_1)}{2}\cos\left(\frac{\pi t}{\tau_1}\right), 0 \le t \le \tau_1 \\
\xi_2^d &= \xi_2(\tau_1)\frac{\psi(t)}{\psi(\tau_1)}, \qquad \tau_1 < t \le \tau \\
\xi_2^d &= 0, \qquad\qquad t > \tau
\end{aligned}
\tag{13}
$$

where $\xi_2^*(\tau_1)$ is an intermediate goal to be reached in τ_1 seconds, and which can be defined by $\xi_2^*(\tau_1) = \kappa \tan(\psi(0))$. The factor κ is related to the curvature of the motion during $t \le \tau_1$. The first segment of ξ_2^d drives the robot to reach the condition $|\phi(\tau_1)| > |\psi(\tau_1)|$, and after that, the corresponding trajectory depending on $\psi(t)$ is used. We suggest to define $\tau_1 = \tau/2$. As it can be seen in the next section, the robot motion from these desired trajectories does not result in rotations in site. This is preferred because, mostly, rotations in site must be performed in a short time, which is dynamically demanding for the system.

5. Stability Analysis

In this section, we provide conditions for the selection of appropriate gains in the super-twisting controller and we prove the relationship between the control of the chosen visual measurements and the camera-robot's state.

Lemma 1. *Suppose that the perturbation terms of the system in Eq. 8 are globally bounded by:*

$$|\varrho_1| \le \gamma_1 |e_1|^{\frac{1}{2}}, \ |\varrho_2| \le \gamma_2 |e_2|^{\frac{1}{2}}$$

for some constants γ_1, $\gamma_2 > 0$. Then, the origin $(e_1, e_2) = (0, 0)$ is an equilibrium point that is globally asymptotically stable if the control gains satisfy:

$$
\begin{aligned}
k_1 &> 2\gamma_1, \ k_2 > \frac{k_1\gamma_1^2}{8(k_1 - 2\gamma_1)} \\
k_3 &> 2\gamma_2, \ k_4 > \frac{k_3\gamma_2^2}{8(k_3 - 2\gamma_2)}
\end{aligned}
$$

Proof. The closed loop behavior of the error system in Eq. 8 is given by:

$$
\begin{aligned}
\dot{e}_1 &= -k_1 |e_1|^{\frac{1}{2}} \operatorname{sgn}(e_1) + u_1 + \varrho_1, \\
\dot{u}_1 &= -k_2 \operatorname{sgn}(e_1), \\
\dot{e}_2 &= -k_3 |e_2|^{\frac{1}{2}} \operatorname{sgn}(e_2) + u_2 + \varrho_2, \\
\dot{u}_2 &= -k_4 \operatorname{sgn}(e_2)
\end{aligned}
\tag{14}
$$

Notice that the dynamics of e_1 and e_2 are decoupled one from another and the stability of e_1 and e_2 can be proven separately. Let us define the following candidate Lyapunov function for the first block as proposed in [20] for a super-twisting controller:

$$V(e_1, u_1) = 2k_2 \left|e_1\right| + \tfrac{1}{2}u_1^2 + \tfrac{1}{2}\left(k_1 \left|e_1\right|^{\frac{1}{2}} \operatorname{sgn}(e_1) - u_1\right)^2,$$

which can be expressed in the quadratic form

$$V(e_1, u_1) = \tfrac{1}{2}\Psi^T P \Psi, \tag{15}$$

where $\Psi = \begin{bmatrix} \left|e_1\right|^{\frac{1}{2}} \operatorname{sgn}(e_1) & u_1 \end{bmatrix}^T$ and $P = \begin{bmatrix} 4k_2 + k_1^2 & -k_1 \\ -k_1 & 2 \end{bmatrix}$. It can be seen that $V(e_1, u_1)$ is positive definite and radially bounded if $k_2 > 0$. By expanding the quadratic form, we find that the time-derivative of the candidate Lyapunov function is given by:

$$\begin{aligned}\dot{V}(e_1, u_1) = {}& \left(2k_2 + \tfrac{1}{2}k_1^2\right) \tfrac{d}{dt}\left(\left|e_1\right|\right) + 2u_1\dot{u}_1 \\ & -k_1\left[\dot{u}_1 \left|e_1\right|^{\frac{1}{2}} \operatorname{sgn}(e_1) + u_1\left(\tfrac{d}{dt}\left(\left|e_1\right|^{\frac{1}{2}}\right)\operatorname{sgn}(e_1) + \left|e_1\right|^{\frac{1}{2}} \tfrac{d}{dt}\left(\operatorname{sgn}(e_1)\right)\right)\right].\end{aligned}$$

We have that $\frac{d}{dt}\left(\left|e_1\right|\right) = \dot{e}_1\operatorname{sgn}(e_1)$, $\frac{d}{dt}\left(\left|e_1\right|^{\frac{1}{2}}\right) = \frac{1}{2}\left|e_1\right|^{-\frac{1}{2}} \frac{d}{dt}\left(\left|e_1\right|\right)$ and the last term of the previous expression $\left|e_1\right|^{\frac{1}{2}} \frac{d}{dt}\left(\operatorname{sgn}(e_1)\right) = 0$ as can be verified in [20]. Then, the time-derivative can be expressed as:

$$\dot{V}(e_1, u_1) = \left(2k_2 + \tfrac{1}{2}k_1^2\right)\dot{e}_1\operatorname{sgn}(e_1) + 2u_1\dot{u}_1 - k_1\operatorname{sgn}(e_1)\left(\dot{u}_1\left|e_1\right|^{\frac{1}{2}} + \tfrac{1}{2}u_1\left|e_1\right|^{-\frac{1}{2}}\dot{e}_1\operatorname{sgn}(e_1)\right).$$

Using the closed loop dynamics given by Eq. 14, and after some simple mathematical operations, the time-derivative can be written as follows:

$$\begin{aligned}\dot{V}(e_1, u_1) = {}& -\tfrac{1}{2}k_1\left|e_1\right|^{-\frac{1}{2}}\left(2k_2\left|e_1\right| + k_1^2\left|e_1\right| - 2k_1u_1\left|e_1\right|^{\frac{1}{2}}\operatorname{sgn}(e_1) + u_1^2\right) \\ & +\varrho_1\left|e_1\right|^{-\frac{1}{2}}\left(\left(2k_2 + \tfrac{1}{2}k_1^2\right)\left|e_1\right|^{\frac{1}{2}}\operatorname{sgn}(e_1) - \tfrac{1}{2}k_1u_1\right).\end{aligned}$$

Then, it can be expressed in the following form:

$$\dot{V}(e_1, u_1) = -\tfrac{k_1}{2}\left|e_1\right|^{-\frac{1}{2}}\Psi^T Q \Psi + \varrho_1\left|e_1\right|^{-\frac{1}{2}} q_1^T \Psi \tag{16}$$

where $Q = \begin{bmatrix} 2k_2 + k_1^2 & -k_1 \\ -k_1 & 1 \end{bmatrix}$ and $q_1^T = \begin{bmatrix} 2k_2 + \frac{1}{2}k_1^2 & -\frac{1}{2}k_1 \end{bmatrix}$. Notice that if the perturbation term is null, the stability condition is determined by the matrix Q, which must be positive definite in order to have $\dot{V}$ negative definite. This is accomplished if $k_1 > 0$ and $k_2 > 0$. Otherwise, in a real situation where there exists a bounded perturbation term, the following analysis holds. We know that $\left|\varrho_1\right| \leq \gamma_1\left|e_1\right|^{\frac{1}{2}}$, so that, in the extremes of the inequality $\varrho_1 = \gamma_1\left|e_1\right|^{\frac{1}{2}}\operatorname{sgn}(e_1)$. Then, the time-derivative of the candidate Lyapunov function accomplishes:

$$\begin{aligned}\dot{V}(e_1, u_1) \leq {}& -\tfrac{1}{2}k_1\left|e_1\right|^{-\frac{1}{2}}\left(2k_2\left|e_1\right| + k_1^2\left|e_1\right| - 2k_1u_1\left|e_1\right|^{\frac{1}{2}}\operatorname{sgn}(e_1) + u_1^2\right) \\ & +\gamma_1\left|e_1\right|^{\frac{1}{2}}\operatorname{sgn}(e_1)\left|e_1\right|^{-\frac{1}{2}}\left(\left(2k_2 + \tfrac{1}{2}k_1^2\right)\left|e_1\right|^{\frac{1}{2}}\operatorname{sgn}(e_1) - \tfrac{1}{2}k_1u_1\right),\end{aligned}$$

which can be simplified as:

$$\dot{V}(e_1,u_1) \leq -\tfrac{1}{2}k_1 \left|e_1\right|^{-\frac{1}{2}} \left[\left(2k_2 + k_1^2 - \left(\tfrac{4k_2}{k_1} + k_1\right)\gamma_1\right)\left|e_1\right| - \left(2k_1 - \gamma_1\right)u_1 \left|e_1\right|^{\frac{1}{2}} \operatorname{sg}(e_1) + u_1^2\right].$$

This expression can be written as follows:

$$\dot{V}(e_1,u_1) \leq -\tfrac{k_1}{2}\left|e_1\right|^{-\frac{1}{2}} \Psi^T \tilde{Q}\Psi, \text{ with } \tilde{Q} = \begin{bmatrix} 2k_2 + k_1^2 - \left(\tfrac{4k_2}{k_1} + k_1\right)\gamma_1 & -k_1 + \tfrac{1}{2}\gamma_1 \\ -k_1 + \tfrac{1}{2}\gamma_1 & 1 \end{bmatrix}$$

Thus, the time-derivative of the candidate Lyapunov $\dot{V}$ is negative definite if $\tilde{Q}$ is positive definite ($\tilde{Q} > 0$), which is satisfied if

$$2k_2 + k_1^2 - \left(\tfrac{4k_2}{k_1} + k_1\right)\gamma_1 > 0 \text{ and } \det\left(\tilde{Q}\right) > 0.$$

Operating the first condition, we have $2\left(k_1 - 2\gamma_1\right)k_2 + k_1^2\left(k_1 - \gamma_1\right) > 0$. This is accomplished for $k_1 > 2\gamma_1$ and $k_2 > 0$. However, a more conservative bound has to be found for k_2 from the determinant positivity, which is equivalent to:

$$2\left(k_1 - 2\gamma_1\right)k_2 - \tfrac{1}{4}k_1\gamma_1^2 > 0.$$

Both following conditions ensure that $\det(\tilde{Q})$ is positive and consequently that $\dot{V}$ is negative definite:

$$k_1 > 2\gamma_1, \;\; k_2 > \frac{k_1\gamma_1^2}{8\left(k_1 - 2\gamma_1\right)}$$

These conditions establish that if the control gains are larger than some terms depending on a bound of the perturbations, then asymptotic stability is achieved regardless of the value of the gains. Therefore, this is a global condition for stability, so that the error dynamics e_1 is globally asymptotically stable. Similar conditions are derived following the same procedure for the second block of Eq. 14.

The global asymptotic stability of the equilibrium point of the tracking error system is consistent with previous stability proofs of control systems using a super-twisting controller [12, 20].

Theorem 1. *The camera-robot system is in the target location if and only if the visual measurements, given by Eq. 7, both reaches zero in τ seconds. Therefore, the desired target location $(x, y, \phi) = (0, 0, 0)$ is the only equilibrium point of the system's state.*

Proof. On the one hand, it is trivial to see that if the camera-robot system is in the target location, then the visual measurements are both equal to zero (see Eq. 7). On the other hand, Lemma 1 ensures that the error system of Eq. 8 is globally asymptotically stable, i.e., the tracking error is ensured to be zero using adequate control gains independently of the perturbations ϱ_1, ϱ_2. Indeed, the initial errors are zero because the desired trajectories are defined to initiate from the values of the visual measurements at zero time (Eqs. 11-13). Given the asymptotic stability of the closed-loop system, the tracking error remains at zero, i.e. $(e_1, e_2) = (0, 0)$ and then $\xi_1 = \xi_1^d$, $\xi_2 = \xi_2^d$. Thus, we have:

$$\begin{aligned} \xi_1 &= \alpha t_y = \frac{\xi_1(0)}{2}\left(1 + \cos\left(\frac{\pi t}{\tau}\right)\right), \\ \xi_2 &= -\tan\phi = \frac{\xi_2(0)}{\psi(0)} \arctan\left(\frac{x}{y}\right). \end{aligned} \quad (17)$$

where t_y is the relative position between the current and target positions with respect to the current reference frame. Using the non-holonomic motion constraint of the robot expressed as

$$\frac{\dot{x}}{\dot{y}} = -\tan\phi,$$

the behavior of the visual measurement ξ_2 is defined by:

$$\xi_2 = \frac{\dot{x}}{\dot{y}} = \frac{\xi_2(0)}{\psi(0)} \arctan\left(\frac{x}{y}\right).$$

We can use that $\arctan\left(\frac{x}{y}\right) \approx \frac{x}{y}$ in the range $\frac{x}{y} \in [-1, 1]$. Thus, the solution of the approximated differential equation is given by:

$$x = C_i y^{\frac{\xi_2(0)}{\psi(0)}} \quad (18)$$

where C_i is the constant of integration in the solution of the differential equation. As we have said previously, ξ_1 is related to the position correction in the $y-$axis. The reduction of the longitudinal error y is defined by the sinusoidal function that reaches zero in τ seconds. So that, $\xi_1 \to 0 \Rightarrow y \to 0$ and, from Eq. 18, $y \to 0 \Rightarrow x \to 0$. Therefore, as given by Eq. 17, $x \to 0 \Rightarrow \xi_2 \to 0$, and finally, the angular error ϕ follows the prescribed behavior given the simultaneous convergence of x and y.

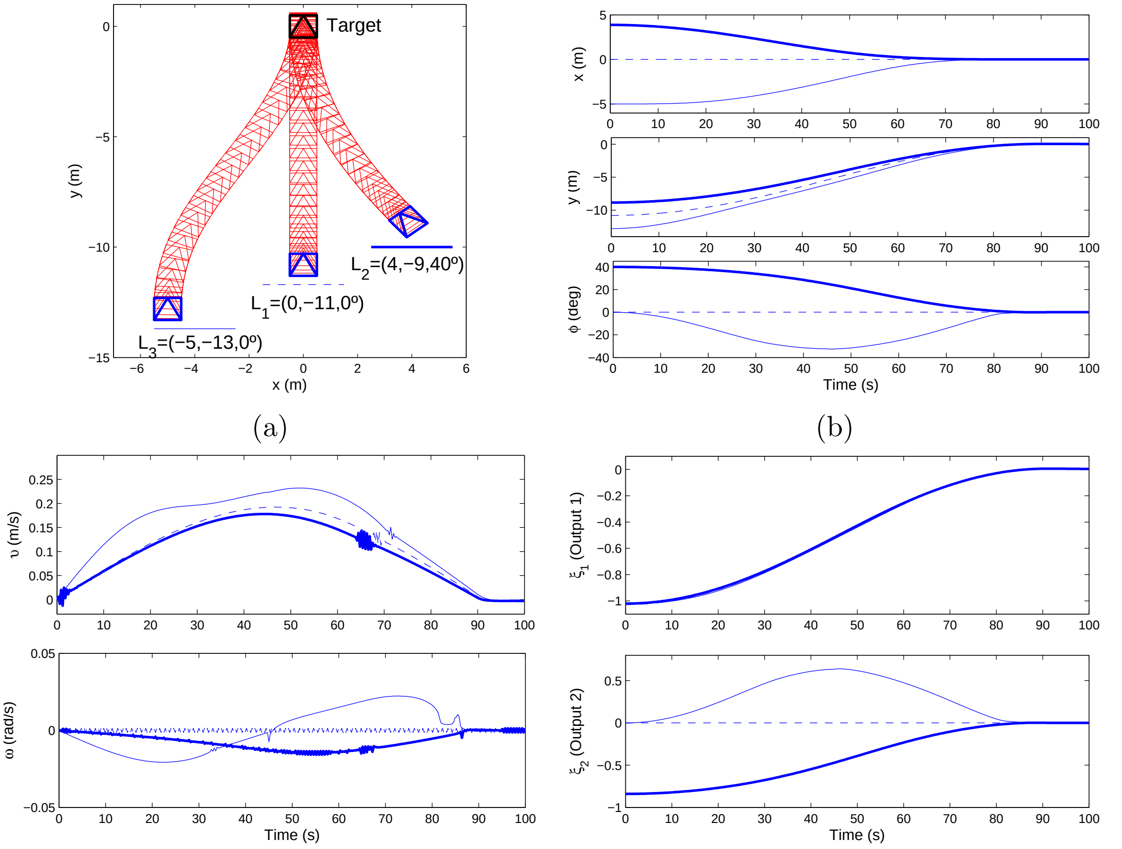
6. Controller Evaluation

In this section, the performance of the proposed scheme for pose regulation tasks is shown via simulations and experiments with a popular dynamic robot simulator.

6.1. Simulation Results

The simulation results have been obtained by using Matlab scripting with a closed loop time of 0.3s. The TT is estimated from synthetic images generated using the parameters presented in Table 1. Synthetic images are generated through the generic camera model [14]. The control gains have been fixed as $k_1 = 0.5$, $k_2 = 0.02$, $k_3 = 0.1$ and $k_4 = 0.01$. The time to complete the regulation task has been defined as $\tau = 90$s.

The resulting paths on the plane, from three different initial locations, can be seen in Fig. 6(a). The case of the initial location at $\mathbf{L}_1 = (0, -11, 0^o)$ is special, given that $\phi(0) = \psi(0) = 0$. In such a case, $\xi_2^d = 0$ during the navigation. This case is also



(a) (b)

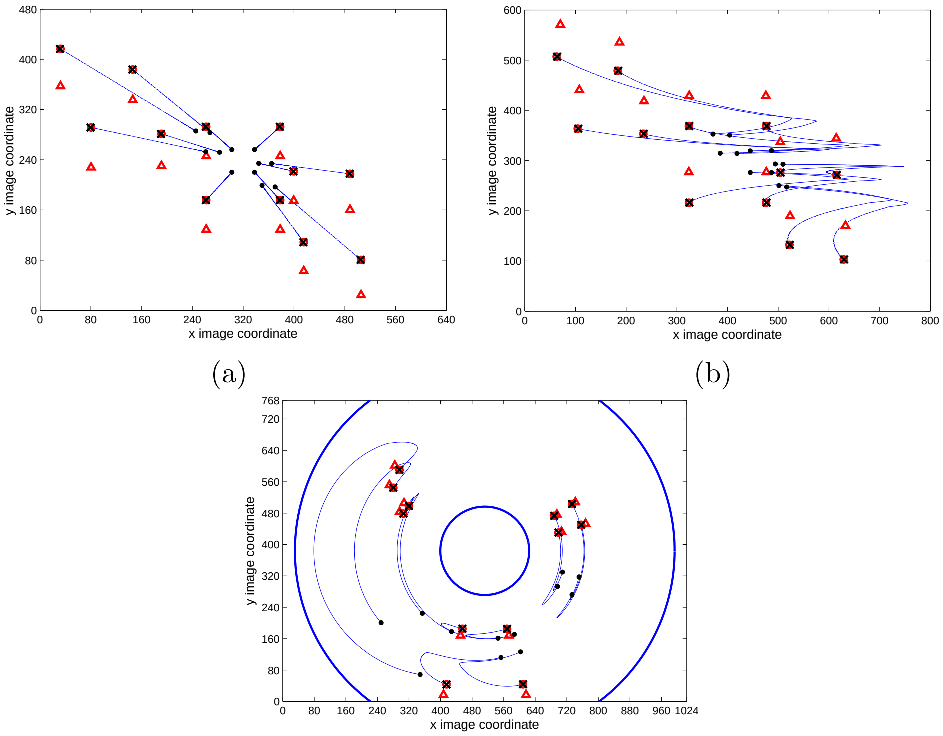
(c) (d)

Figure 6: Simulation results for three different initial locations showing the robot navigation and the behavior of important variables. (a) Paths on the $x-y$ plane. (b) Evolution of the robot pose. (c) Computed velocities. (d) Evolution of the visual measurements.

	Image size	ξ	f (mm)	α_x	α_y
Conventional	640×480	0	6	560	561
Fisheye	800×600	2.916	8	990	994
Paracatadioptric	1024×768	1	6	895	898
Hypercatadioptric	1024×768	0.9662	6	895	898

Table 1: Parameters for the different types of camera.

special because an epipolar control is not able to solve it. In the case of the initial location at $\mathbf{L}_2 = (4, -9, 40^o)$, ξ_2^d is defined by Eq. 12. For $\mathbf{L}_3 = (-5, -13, 0^o)$, ξ_2^d is defined by Eq. 13. In all three cases, the robot reaches the target with good precision and carries out a smooth motion, as shown in Fig. 6(b). This behavior is obtained with the robot velocities of Fig. 6(c), which are provided by the super-twisting control. Notice that the velocities are practically continuous. Just at some moments, when the trajectory tracking deviates lightly, a small component in the form of a triangular signal may appear in the control inputs. In Fig. 6(d), the evolution of the visual measurements given by Eq. 7 is shown. Due to the normalization of ξ_1, the plots for the three cases look similar.



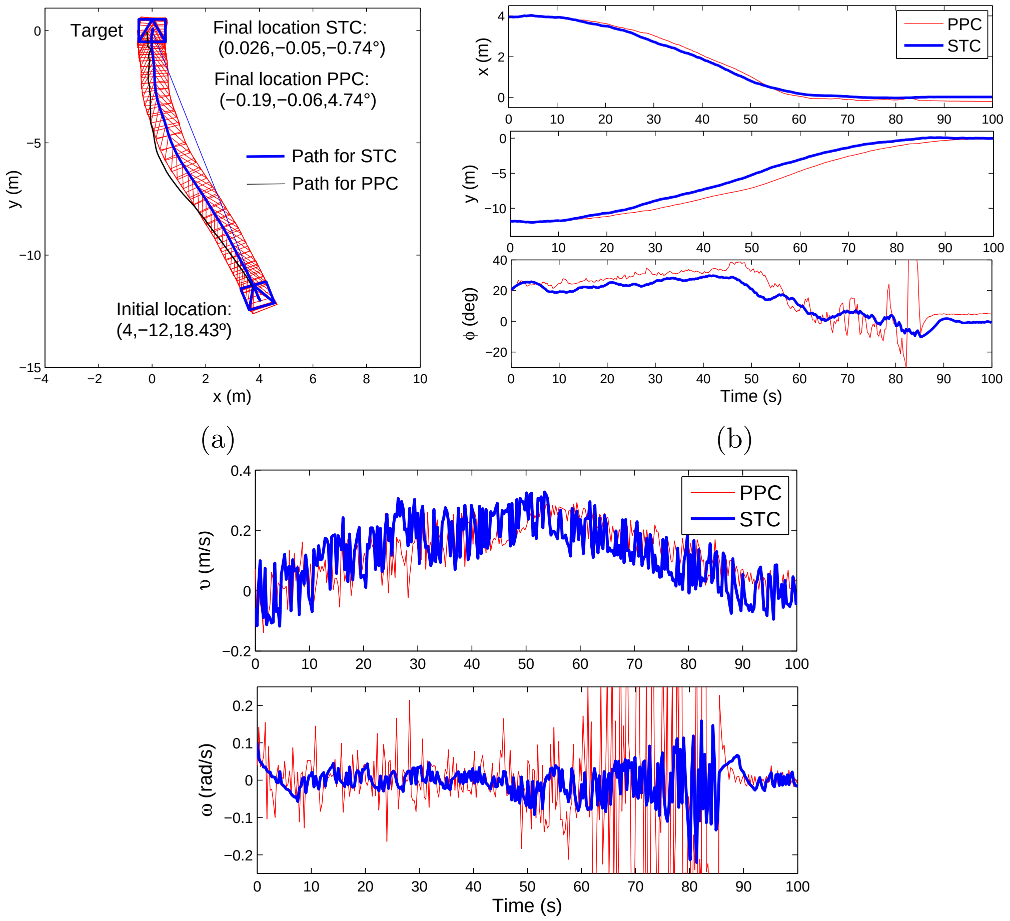
(a) (b)

(c)

Figure 7: Motion of the image points for the three initial locations using (a) a conventional camera for points motion for L_1. (b) a fisheye camera for points motion for L_2. and (c) a paracatadioptric system for points motion for L_3. The markers represent: "·"=initial image, "O"=real target image, "△"=virtual target image and "×"=image at the end of the motion.

In order to show the validity of the proposed VS scheme for different cameras, the motion of the point features is presented in Fig. 7(a) using a conventional camera from $\mathbf{L}_1$. A fisheye camera is simulated for the case with initial location $\mathbf{L}_2$ in Fig. 7(b). For the control from $\mathbf{L}_3$, a paracatadioptric vision system is used and the motion of the image points is shown in Fig. 7(c).

A performance comparison of the Super-Twisting Control (STC) with a classical Pole-Placement Control (PPC) is presented to show the benefits of the former. We evaluate both controllers under Gaussian image noise of 1 pixel of standard deviation and under a perturbation in rotation. This perturbation is simulated by adding a constant rotational velocity ω_{per} =0.1 rad/s to the control signal ω, in such a way that the robot is drifting to the left as it moves if the controller does not compensate adequately. The PPC has been used in several vision-based schemes, e.g., [7, 9]. For such a control, the auxiliary inputs of Eq. 9 are defined as $v_1 = -k_1 e_1 + \dot{\xi}_1^d$, $v_2 = -k_2 e_2 + \dot{\xi}_2^d$. Notice that $\dot{\xi}_2^d$ is unknown given the definition of the desired trajectory and we assume $\dot{\xi}_2^d = 0$, which may affect the behavior of the PPC. Fig. 8(a) shows that the regulation task is successfully accomplished in spite of the perturbed situation using the STC and with superior accuracy than the PPC. The initial location corresponds to $\phi(0) = \psi(0)$ as depicted by the line along the longitudinal axis of the robot at the initial location. That configuration generates a singularity for epipolar control, which do not occur in the proposed approach.

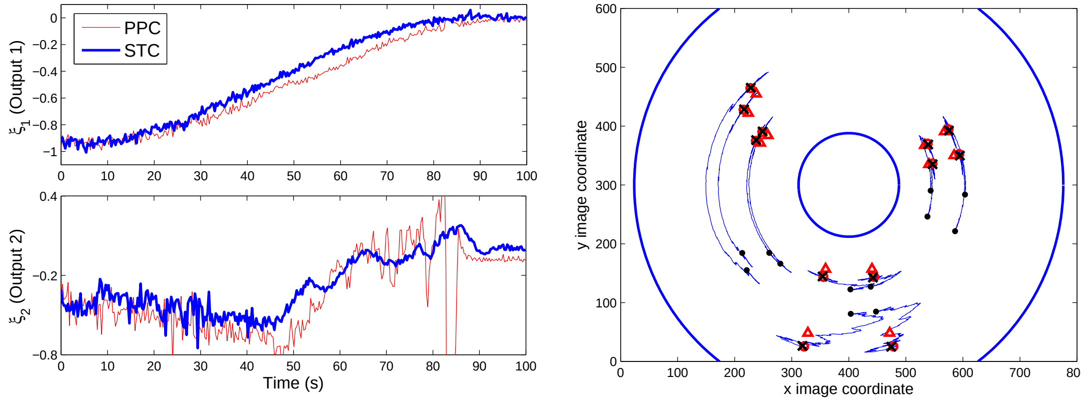


(a)

(b)

(c)

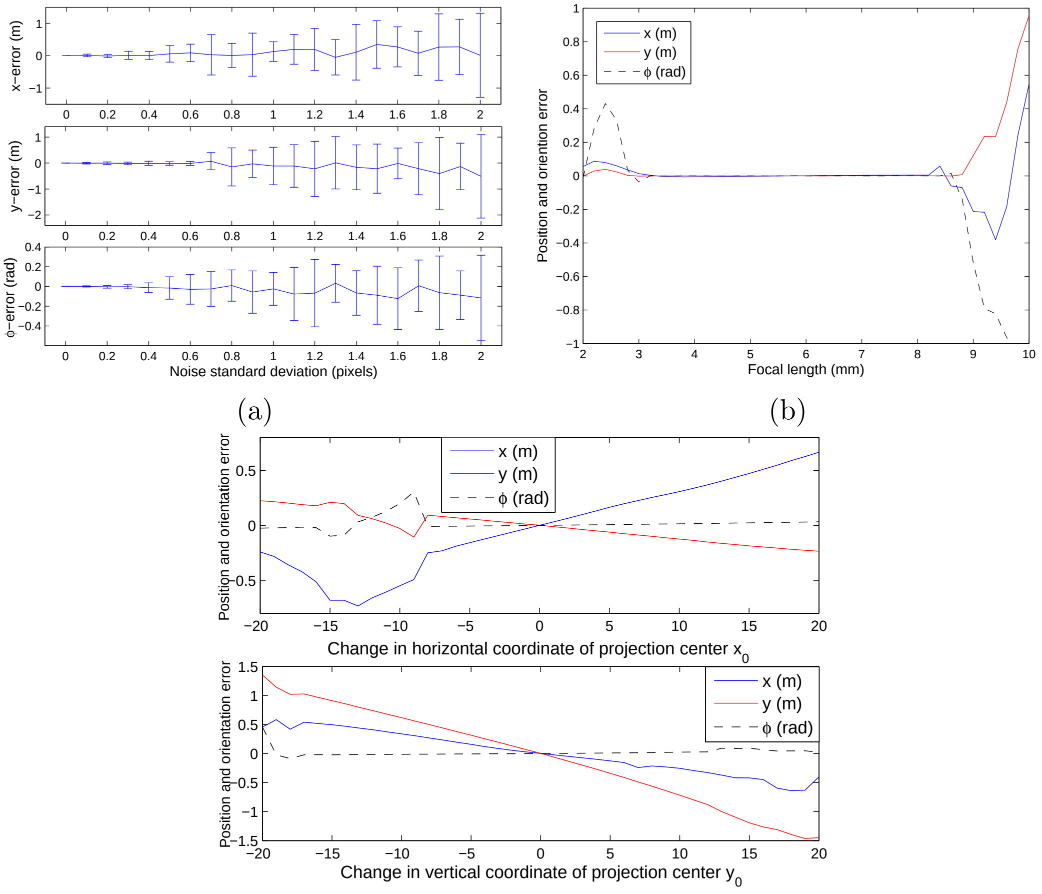
Figure 8: Performance comparison between the Super-Twisting Control (STC) and a Pole-Placement Control (PPC) including image noise and a perturbation in rotation. (a) Resulting paths. (b) Evolution of the robot state. (c) Computed velocities.



(a)

(b)

Figure 9: Behavior of the visual measurements including image noise and a perturbation in rotation for the Super-Twisting Control (STC) in comparison with a Pole-Placement Control (PPC). (a) Visual measurements. (b) Image points motion using the STC.



(a) (b)

(c)

Figure 10: Analysis of robustness against noise and parameters uncertainty. (a) Robustness against image noise. (b) Robustness to focal length variation. (c) Robustness to optical center variation.

As can be seen in Fig. 8(b-c), the effect of image noise is really harmful for PPC because the amplitude of the control inputs directly depends on the amplitude of the error signals. On the contrary, the STC is able to reduce significantly the effect of image noise and effectively reject the effect of the perturbation. The visual measurements are well behaved for the STC as shown in Fig. 9(a). The vision system in this case is hypercatadioptric and the motion of the image points for the STC is presented in Fig. 9(b).

An important aspect of any visual control scheme is the robustness against image noise and uncertainty in parameters. Fig. 10(a) shows the mean and standard deviation of the mean squared error for each state variable of the camera-robot over 50 Monte Carlo runs while varying the standard deviation of the image noise from 0 to 2 pixels. It can be seen that the effect is really small even for the largest image noise. As described in section 2.1, the image points matched in the three images have to be transformed to calibrated coordinates through the internal camera calibration parameters before computing the TT. In the simulations depicted in Fig. 10(b) the known focal length is fixed to 6mm for the TT estimation while its real value to generate the synthetic images is changed from 2mm to 10mm. It can be appreciated that a variation about ±2mm in the focal length is efficiently supported by the control scheme, with no practical effect on the final location. In Fig. 10(c) the coordinates of the projection center are considered in the center of the image for the control law, while their real value is changed on each direction. The final position is more sensitive to this variation than the orientation, so that, the position error increase as the variation of the projection center increases. However, the control scheme provides good robustness and keeps a good performance of the navigation task in spite of small calibration errors.

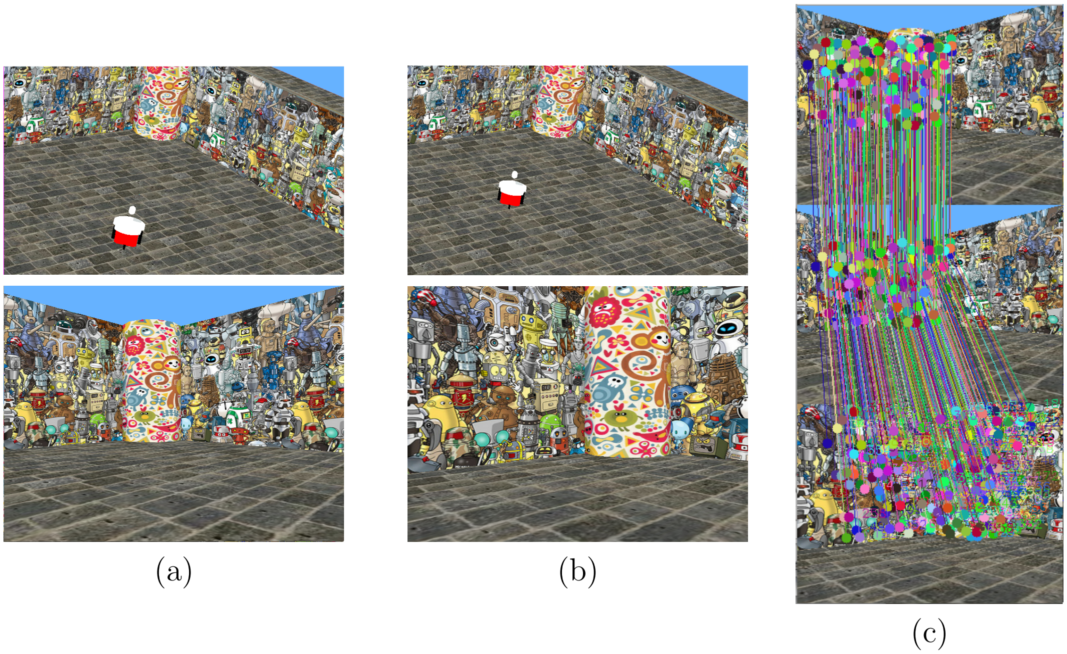


Figure 11: Visual configuration of the experiments. (a) Image at $\mathbf{C}_1$ (bottom) and corresponding pose (up). (b) Image at $\mathbf{C}_3$ (bottom) and corresponding pose (up). (c) Matches between SIFT features in $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$, used to evaluate $\mathbf{TT}(0)$ and the virtual target.

6.2. Experiments with Dynamic Robot Simulation

Experiments have been conducted with the widely used Webots simulator. The set-up is the following one: A Pioneer 2 robot with a single perspective camera is simulated, included the dynamics of the robot, not involved in the previous simulations. Dynamic simulation induces for example a small sway motion of the camera, that introduces errors in the model deduced until here from the planar assumption. The environment is made of textured walls on which the feature points used for the evaluation of the trifocal tensor are found. The images taken by the robot camera are 640 × 480. A reference image is chosen beforehand, at some position $\mathbf{C}_3$ inside the scene. This reference image is depicted in Fig. 11(b). Then, the robot is settled at an initial configuration $\mathbf{C}_1$, where the taken image is depicted at Fig. 11(a). After that, it performs an arbitrary forward motion up to a configuration $\mathbf{C}_2$.

At $\mathbf{C}_2$, and as described in Section 3, we proceed to the generation of the virtual target. For this, we first detect multiple scale interest points between views

$\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$. Given the viewpoint change at view $\mathbf{C}_3$, a multiple scale detector is necessary. Here, we used SIFT points for that task [21]. Then, point matches are determined among the three views by using SIFT descriptors around each detected point and forming pairwise matches (between $\mathbf{C}_1, \mathbf{C}_2$, first, and $\mathbf{C}_2, \mathbf{C}_3$, then). Three-fold matches $\mathbf{p}, \mathbf{p}'(0), \mathbf{p}''$ are formed by selecting pairs of pairwise matches sharing the same points in configuration $\mathbf{C}_2$. With these matches, a RANSAC strategy is used to estimate $\mathbf{TT}(0)$ (the tensor at time 0). From Eq. 5, we deduce the tensor $\overline{\mathbf{TT}}(0)$ associated to the virtual target.

For the estimation of the tensor for subsequent frames, we have chosen to avoid using the costly multiple-scale point detection at each time step. Instead, we used KLT [22] features $\mathbf{q}$, detected at $\mathbf{C}_1$, tracked into $\mathbf{C}_2$ as $\bar{\mathbf{q}}'(0)$ and transferred into $\overline{\mathbf{C}}_3$ as $\bar{\mathbf{q}}''$ with $\overline{\mathbf{TT}}(0)$ (see 3.2). At each subsequent frames, the tensor $\overline{\mathbf{TT}}(t)$ is evaluated by tracking the KLT points $\bar{\mathbf{q}}'(t)$.

Also, to make the task of the tensor estimation easier, we make use of the fact that ϕ_1 stays a priori constant over the sequence. From Eq. 5, this makes 3 degrees of freedom disappear from the 27 original ones. Given also that seven other elements are zero-valued, then the Direct Linear Transform (DLT) linear estimation can be reduced to 17 a priori non-null matrix elements. Since each three-fold match gives four constraints, we need 5 matches for each RANSAC iteration.

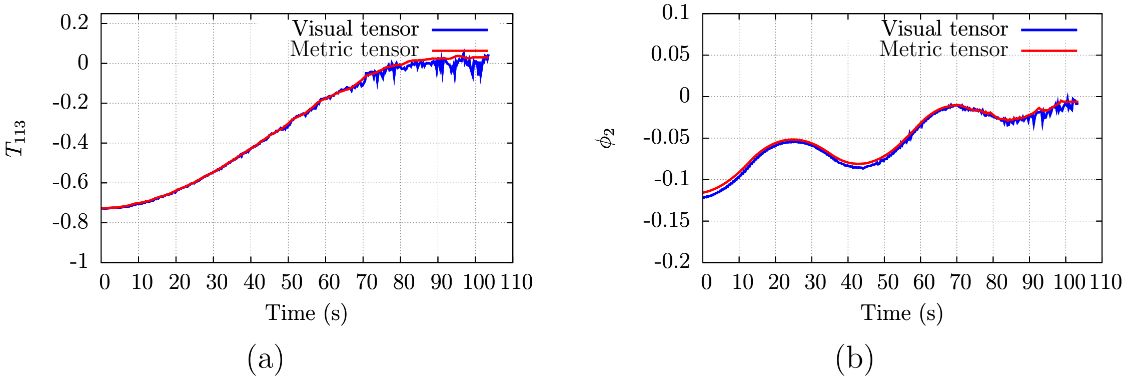


Figure 12: Evaluation of the quality of the visual tensor estimation along time in the Webots simulation, comparing our visual estimation with a ground truth value. (a) Evolution of the term $\overline{\mathbf{TT}}(t)_{113}$. (b) Evolution of $\phi_2(t)$, computed as in Eq. 6.

In Fig. 12, we give an illustration of the quality of the visual tensor estimation $\overline{\mathbf{TT}}(t)$, through the evolution of two estimated quantities derived from it vs. their ground truth values. The ground truth value is computed by using the theoretical tensor given the true planar pose of the robot. The two depicted elements are $\overline{\mathbf{TT}}(t)_{113}$ and $\phi_2(t)$, both derived from $\overline{\mathbf{TT}}(t)$. As it can be seen, the estimation is fair, although it becomes noisier when time goes on, mainly because of the loss of visible interest points. In Fig. 13, we depict the evolution of the visual measurements used in our control scheme, ξ_1 and ξ_2. In both cases, we compare the estimated value to the ground truth value that should be applied to the controller if we knew the real robot pose. In Fig. 14(a), we show in one particular run, the evolution of the robot pose. It can be verified that position and orientation all converge smoothly to the desired position, namely zero.

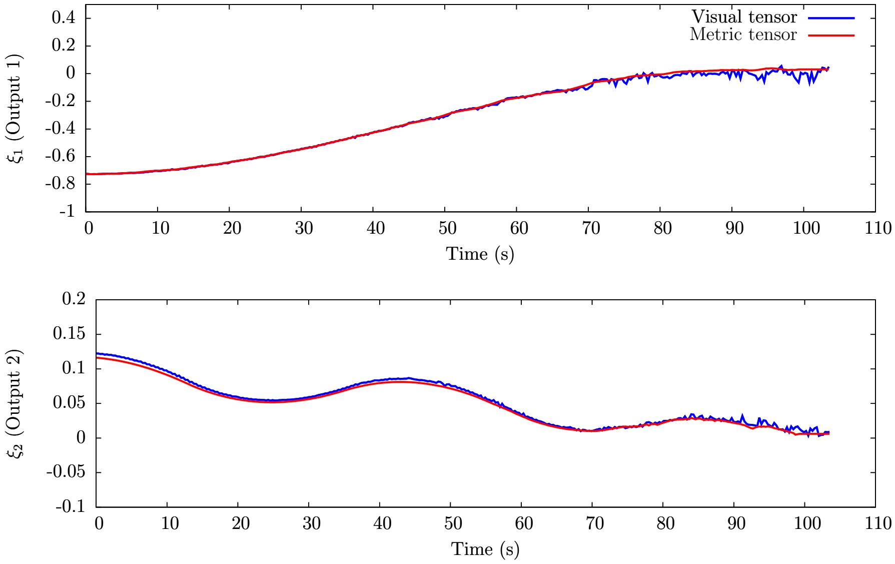


Figure 13: Behavior of the visual measurements along time in the Webots simulation. In blue, the measurements extracted from the estimated visual tensor. In red, the ground truth measurements estimated based on the current real robot pose.

7. Conclusions

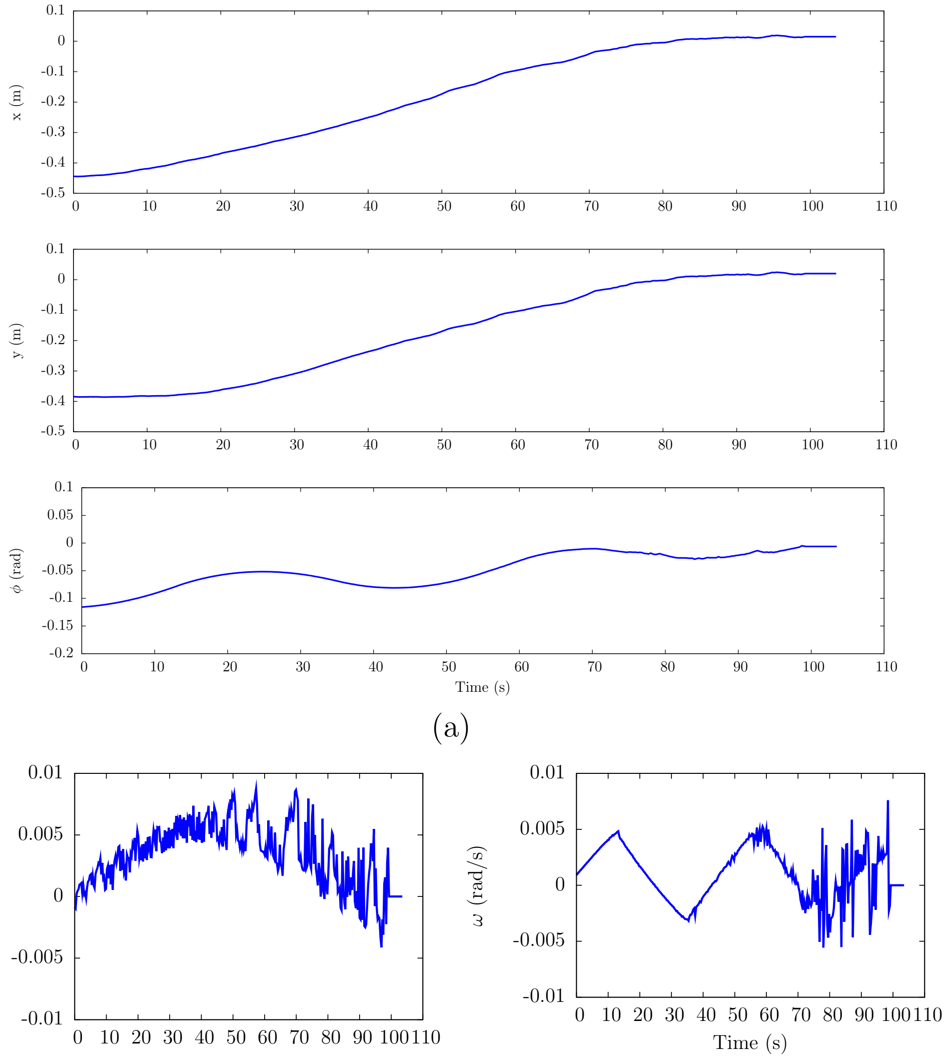
We have presented a novel approach for image-based visual servoing that performs robot pose regulation based on a few elements of the trifocal tensor relating the initial view from the robot, the current one and the target view. Contrary to other approaches using the trifocal tensor or the epipolar geometry, we manage to avoid singularities in the control scheme by defining a virtual target, easily deduced from the real one by a vertical translation. Thus, a single controller, tracking adequate trajectories, is able to solve the pose regulation problem without the need of switching to a different controller. Moreover, we have introduced the use of super-twisting control, developed elsewhere, in visual servoing. We have illustrated the pertinence of these two aspects with convincing results obtained in simulation, with comparisons against the classical pole-placement control, and with experiments using dynamic robot simulation.

References

[1] F. Chaumette, S. Hutchinson, Visual servo control, part I: Basic approaches, IEEE Robotics and Aut. Mag. 13 (14) (2006) 82–90.

[2] F. Chaumette, S. Hutchinson, Visual servo control, part II: Advanced approaches, IEEE Robotics and Aut. Mag. 14 (1) (2007) 109–118.

[3] Y. Fang, W. E. Dixon, D. M. Dawson, P. Chawda, Homography-based visual servo regulation of mobile robots, IEEE Transactions on Systems, Man, and Cybernetics, Part B 35 (5) (2005) 1041–1050.



(a)

(b)

Figure 14: Performance of the proposed controller in the Webots simulation. (a) Evolution of the robot position and orientation. The target pose is $(0, 0, 0)$. (b) Evolution of the controls (linear and angular velocities) during the experiment, as a function of time (in seconds).

[4] A. Usai, P. D. Giamberardino, Visual feedback for nonholonomic mobile robots: Homography based approach, in: G. Rigatos (Ed.), Intelligent Industrial Systems, IGI Global, 2010, pp. 152–181.

[5] G. L. Mariottini, G. Oriolo, D. Prattichizzo, Image-based visual servoing for nonholonomic mobile robots using epipolar geometry, IEEE Transactions on Robotics 23 (1) (2007) 87–100.

[6] H. M. Becerra, G. López-Nicolás, C. Sagüés, A sliding mode control law for mobile robots based on epipolar visual servoing from three views, IEEE Transactions on Robotics 27 (1) (2011) 175–183.

[7] G. López-Nicolás, J. Guerrero, C. Sagüés, Visual control through the trifocal

tensor for nonholonomic robots, Robotics and Autonomous Systems 58 (2) (2010) 216–226.

[8] H. M. Becerra, G. López-Nicolás, C. Sagues, Omnidirectional visual control of mobile robots based on the 1D trifocal tensor, Robotics and Autonomous Systems 58 (6) (2010) 796–808.

[9] G. López-Nicolás, C. Sagüés, J. Guerrero, Parking with the essential matrix without short baseline degeneracies, in: IEEE International Conference on Robotics and Automation, 2009, pp. 1098–1103.

[10] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, 2004.

[11] H. M. Becerra, J. B. Hayet, C. Sagüés, Virtual target formulation for singularity-free visual control using the trifocal tensor, in: Mexican Conference on Pattern Recognition, Lecture Notes in Computer Science, accepted, June, 2013.

[12] A. Levant, L. Fridman, Higher order sliding modes, in: W. Perruqueti, J. Barbot (Eds.), Sliding Mode Control in Enginnering, Marcel Dekker, N.Y., USA, 2002, pp. 53–101.

[13] V. Utkin, J. Guldner, J. Shi, Sliding Mode Control in Electromechanical Systems, CRC Press, Boca Raton, 1999.

[14] C. Geyer, K. Daniilidis, An unifying theory for central panoramic systems and practical implications, in: European Conference on Computer Vision, 2000, pp. 445–461.

[15] Y. Yu, K. Wong, M. Chang, S. Or, Recursive camera-motion estimation with the trifocal tensor, IEEE Trans. on Syst., Man, and Cyb. - Part B: Cybernetics 36 (5) (2006) 1081–1090.

[16] L. Quan, M. Lhuillier, Structure from motion from three affine views, in: International Conference on Pattern Recognition, 2002, pp. 1–6.

[17] J. Slotine, W. Li, Applied nonlinear control, Prentice Hall, 1991.

[18] M. Lopez-Franco, A. Salome-Baylón, A. Y. Alanis, N. Arana-Daniel, Discrete super twisting control algorithm for the nonholonomic mobile robots tracking problem, in: Int. Conf. on Electrical Engineering Computing Science and Automatic Control, 2011, pp. 1–5.

[19] E. S. Elyoussef, E. R. D. Pieri, M. Jungers, U. F. Moreno, Super-twisting sliding modes tracking control of a nonholonomic wheeled mobile robot, in: Int. IFAC Symp. on Robot Control, Sept., 2012, pp. 429–434.

[20] J. Moreno, M. Osorio, A Lyapunov approach to second-order sliding mode controllers and observers, in: IEEE Conference on Decision and Control, 2008, pp. 2856–2861.

[21] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. Journal of Computer Vision 60 (2004) 91–110.

[22] J. Bouguet, Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm, Tech. rep., Intel Corporation Microprocessor Research Labs (1999).