

A position-based visual servoing scheme for following paths with nonholonomic mobile robots

Andrea Cherubini, François Chaumette, Giuseppe Oriolo

Abstract—We present a visual servoing scheme enabling nonholonomic mobile robots with a fixed pinhole camera to reach and follow a continuous path on the ground. The controller utilizes only a small set of features extracted from the image plane, without using the complete geometric representation of the path. The scheme is position-based, and a Lyapunov-based stability analysis is carried out. The performance of our control design is experimentally validated on a car-like robot equipped with a pinhole camera.

I. INTRODUCTION

In many recent works, mobile robot navigation is done by processing information from the vision sensors [1]. In some cases, the vision system developed for navigation relies on the geometry of the environment and other metrical information, for driving the vision processes and performing self-localization. In this case, *position-based* visual servoing techniques can be used to control the robot. The feedback law is computed by reducing errors in estimated pose space. Alternative visual navigation systems use no explicit representation of the environment in which navigation takes place. In this case, *image-based* visual servoing techniques [2] can be used to control the robot: an error signal measured directly in the image is mapped to actuator commands, as in [3], [4], and [5]. Here, we present a position-based path following (PF) scheme enabling nonholonomic mobile robots with a fixed pinhole camera to reach and follow a continuous path on the ground, by processing a small set of features in the image plane, as in [6].

In the PF task, the controller must drive some suitable *path error function*, indicating the position of the robot with respect to the path [7], [8] to a desired value (usually, zero). Many articles have focused on the design of visual controllers for tracking a reference path, especially in the field of autonomous vehicle guidance [9], [10], [11]. Most of these works address the problem of zeroing the lateral displacement and orientation error of the vehicle at a particular *lookahead* distance. However, these studies require a complete geometric representation of the path. In [12], differential flatness properties are used to generate effective path following strategies. In [13], the PF problem is formulated by controlling the shape of the curve in the image plane. The practical implementation is, however, rather sophisticated, employing an extended Kalman filter to dynamically estimate the path curve derivatives up to order three.

A. Cherubini and F. Chaumette are with INRIA-IRISA, Campus de Beaulieu 35042, Rennes, France {acherubi, chaumette}@irisa.fr

G. Oriolo is with the Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Ariosto 25, 00185 Roma, Italy oriolodis@dis.uniroma1.it

In summary, most of the aforementioned approaches impose constraints on the path shape, curvature, and initial configuration. Moreover, they rely on a highly accurate online extraction of the path shape. The main contribution of this work is that the proposed visual servoing scheme requires only some visible path features, along with a coarse camera model, and that it guarantees convergence even when the initial error is large.

The paper is organized as follows. In Sect. II, the PF problem is defined along with all the relevant variables utilized in our method. In Sect. III, we propose and illustrate a position-based PF control scheme. In Sect. IV, a Lyapunov-based stability analysis of the control scheme, taking into account the robot kinematic constraint on maximum curvature, is carried out. Experiments are reported in Sect. V. In the conclusion, we summarize the results, and propose directions for future research.

II. PROBLEM DEFINITION

In this work, we focus on the path following task for nonholonomic mobile robots equipped with a fixed pinhole camera. The workspace where the robot moves is planar: $\mathcal{W} = \mathbb{R}^2$. The *path* p to be followed is represented by a continuous curve in \mathcal{W} . A *following direction* is associated to the path (see Fig. 1(a)). We name r the point on the robot sagittal plane that should track the path. With reference to Fig. 1, let us define the reference frames: world frame $\mathcal{F}_W(W, x', y', z')$, robot frame $\mathcal{F}_R(r, x, y, z)$ and image frame $\mathcal{F}_I(I, X, Y)$ (I is the image plane center). The robot state coordinates (i.e., the robot *generalized coordinates*) are $q(\varepsilon) = [x'(\varepsilon) \ y'(\varepsilon) \ \theta(\varepsilon)]^T$, where $\varepsilon \in \mathbb{R}$ is a parameter with infinite domain, $[x'(\varepsilon) \ y'(\varepsilon)]^T$ represent the Cartesian position of r in \mathcal{F}_W , and $\theta(\varepsilon) \in]-\pi, +\pi]$ is the orientation of the robot frame y axis with respect to the world frame x' axis (positive counterclockwise). The camera optical axis has a constant tilt offset $0 < \rho < \frac{\pi}{2}$ with respect to the y axis, and the optical center C is positioned in the robot sagittal plane at $[x \ y \ z]^T = [0 \ t_y \ t_z]^T$. We also define the camera frame $\mathcal{F}_C(C, x_c, y_c, z_c)$, shown in Fig. 1(c).

We choose $u = [v \ \omega]^T$ as the pair of control variables for our system; these represent respectively the linear and angular velocities (positive counterclockwise) of the robot. Point r is chosen as the projection on the ground of the wheel center in the case of a unicycle robot, and as the rear axis center in the case of a car-like robot. Then, in both cases, the state equation of the robot is:

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} u \quad (1)$$

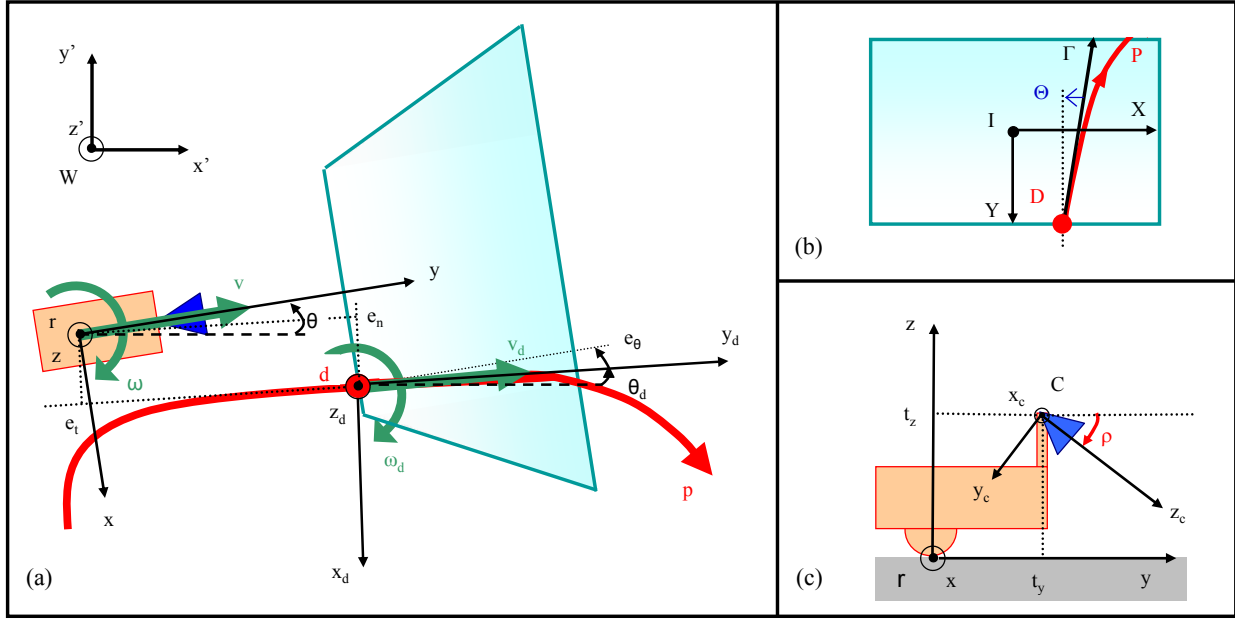


Fig. 1. Relevant variables utilized in this work. The task for the robot (represented in orange), equipped with a fixed pinhole camera (blue) is to follow the red path, noted p . The camera field of view and its projection on the ground are represented in cyan. (a) Top view: frames \mathcal{F}_W , \mathcal{F}_R and \mathcal{F}_P , robot configuration, desired reference configuration, path error $[e_t \ e_n \ e_\theta]^T$, applied (v, ω) and desired (v_d, ω_d) control variables. (b) Image plane view: frame \mathcal{F}_T and image relevant variables. (c) Side view: frames \mathcal{F}_C and \mathcal{F}_R , optical center position in \mathcal{F}_R and camera tilt offset ρ .

In some cases, the robot kinematic constraints can impose a bound c_M on the instantaneous applicable curvature:

$$\left| \frac{\omega}{v} \right| < c_M \quad (2)$$

In the case of a unicycle robot, there is no such bound. Instead, for a car-like robot, the curvature bound is imposed by the steering angle constraint.

Recalling [8], the objective of PF is to drive error $e(\varepsilon) = q(\varepsilon) - q_d(\varepsilon) = [e_{x'}(\varepsilon) \ e_{y'}(\varepsilon) \ e_\theta(\varepsilon)]^T$ to a desired value $\hat{e}(\varepsilon)$. Usually, $\hat{e}(\varepsilon)$ is zero. The vector $q_d(\varepsilon) = [x'_d(\varepsilon) \ y'_d(\varepsilon) \ \theta_d(\varepsilon)]^T$ defines a *desired reference configuration*, such that point $d(\varepsilon) = [x'_d(\varepsilon) \ y'_d(\varepsilon)]^T \in \mathcal{W}$ belongs to p , and $\theta_d(\varepsilon) \in]-\pi, +\pi]$ is the desired robot orientation (see Fig. 1(a)). We assume that in \mathcal{F}_W , the path curve p , can be expressed by a twice differentiable function. Then, $\theta_d(\varepsilon)$ is the orientation of the path tangent at d in \mathcal{F}_R ¹.

The PF task is often formalized by projecting the \mathcal{F}_W errors $[e_{x'}(\varepsilon) \ e_{y'}(\varepsilon) \ e_\theta(\varepsilon)]^T$ to the *path frame* $\mathcal{F}_P(d, x_d, y_d, z_d)$. Frame \mathcal{F}_P is linked to the path at d , with z_d parallel to z , y_d coincident with the path tangent at d in the following direction, and x_d completing the right-handed frame. The path error in \mathcal{F}_P consists of the tangent error e_t (i.e., the error projection on y_d), the normal error e_n (i.e., the error projection on x_d), and the orientation error e_θ , i.e.:

$$\begin{cases} e_t = e_{x'} \cos \theta_d + e_{y'} \sin \theta_d \\ e_n = e_{x'} \sin \theta_d - e_{y'} \cos \theta_d \\ e_\theta = \theta - \theta_d \end{cases} \quad (3)$$

With this formalism, the PF task consists of driving error $[e_t(\varepsilon) \ e_n(\varepsilon) \ e_\theta(\varepsilon)]^T$ to a desired error $[\hat{e}_t \ \hat{e}_n \ \hat{e}_\theta]^T$.

¹ θ_d is always defined, since we have assumed that the path curve can be expressed by a twice differentiable function in \mathcal{F}_W , and this property is preserved in \mathcal{F}_R .

For the nonholonomic model (1), the dynamics of the \mathcal{F}_P path errors e_t , e_n and e_θ are:

$$\begin{cases} \dot{e}_t = -v_d - \omega_d e_n + v \cos e_\theta \\ \dot{e}_n = \omega_d e_t - v \sin e_\theta \\ \dot{e}_\theta = \omega - \omega_d \end{cases} \quad (4)$$

where v_d and ω_d are the components of the tracking control u_d . These must be compliant with the path curvature at d in \mathcal{F}_R , noted c_d^2 :

$$\omega_d = c_d v_d \quad (5)$$

In opposition to *trajectory tracking*, where the desired trajectory evolution is determined by rigid law $\varepsilon = \varepsilon(t)$ (i.e., ε is associated to the time t), in PF we can choose the relationship that defines the desired reference configuration $q_d(\varepsilon)$ to be tracked by the robot. We call such relationship *path following constraint*. The path following constraint eliminates one of the 3 error coordinates. Moreover, in PF, the robot should move at all times independently from $q_d(\varepsilon)$ (clearly, a control law must concurrently ensure convergence to the path). Thus, a motion must be imposed to the robot to guarantee it progresses. This is the *motion exigency* condition as defined in [8]. In most works, the path following constraint is chosen as $e_t = \text{const} = 0$, and the motion exigency as $v = v_d = \text{const} > 0$. For this formulation of the PF problem, the system becomes:

$$\begin{cases} \dot{e}_n = -v_d \sin e_\theta \\ \dot{e}_\theta = \omega - \frac{v_d c_d \cos e_\theta}{1 + e_n c_d} \end{cases} \quad (6)$$

In [7], a nonlinear feedback controller on ω that asymptotically stabilizes this system to $[e_n \ e_\theta]^T = [0 \ 0]^T$ under some

² c_d is always defined, since we have assumed that the path curve can be expressed by a twice differentiable function in \mathcal{F}_W , and this property is preserved in \mathcal{F}_R .

conditions on the initial robot configuration is proposed:

$$\omega = -k_1 v_d e_n \frac{\sin e_\theta}{e_\theta} - k_2 |v_d| e_\theta + \frac{v_d c_d \cos e_\theta}{1 - e_n c_d} \quad (7)$$

with k_1 and k_2 appropriate gains. Path following constraint $e_t = \text{const} = 0$ does not guarantee path visibility. Instead, since here the robot camera is the only sensor available, we want to ensure path visibility. Hence, we use a path following constraint that keeps the reference point d in the camera field of view. The path following constraint chosen in this work will be detailed in the next section.

Similarly to [6], [7], and [14], we express the motion exigency as: $v = v_d = \text{const} > 0$, and we apply a nonlinear feedback on ω based on the features of a visible path point. Under the assumption that a portion of the path is always visible, we utilize the features of the *first* (considering the path direction) visible path point d , of coordinates $[x \ y \ 0]^T$ in \mathcal{F}_R , projected to $D = [X \ Y]^T$ on the image plane (see Fig. 1(b)). We note: P the projection of p on the image plane, Γ the oriented (according to the path direction) tangent of P at D , $\Theta \in]-\pi, \pi]$ the angular offset from Γ to the $-Y$ axis (positive counterclockwise)³.

III. CONTROL DESIGN

Although the feedback law is based on the 3D features (x , y , e_θ , c_d) of the path point, the PF task is defined by the path image features. In practice, the PF task is to drive D to the bottom pixel row of the image plane with $X = \Theta = 0$ (see Fig. 2(g)). Depending on the position of D in the image, the path follower switches between two primitive controllers: a *row controller*, and a *column controller*. In both primitive controllers, the task is to drive the path features to a desired configuration, while D is constrained to a line in the image: a row of pixels ($Y = \text{const}$) in the first case, and a column of pixels ($X = \text{const}$) in the second case. These conditions determine the path following constraint outlined in Sect. II. The path follower utilizes the 2 primitive controllers in general initial conditions, based on a switching mechanism. Consider for instance the initial configuration in Fig. 2(a), with D on the top pixel row of the image. Initially, the row controller must be used to drive D to a lateral pixel column (e.g., the left column, as in Fig. 2(b)). Afterwards, the column controller will be used to drive D along the left pixel column of the image to the bottom left corner (Figures 2(c), 2(d) and 2(e)). Finally, the row controller should be used to drive D along the bottom row of the image plane to the desired configuration (Fig. 2(g)).

In the remainder of this section, we will describe how the 3D path features are derived from the corresponding image path features, and we will illustrate the implementation of the two primitive controllers.

A. Deriving the path 3D features

The path 3D features in \mathcal{F}_R must be derived from the image features by considering a pinhole camera model;

³ Γ and Θ are always defined, since we have assumed that the path curve can be expressed by a twice differentiable function in \mathcal{F}_V and this property is preserved in \mathcal{F}_I .

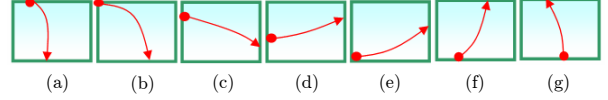


Fig. 2. Seven possible configurations of P in the image plane. Point D is represented by the red circle.

radial distortion and principal point error are neglected. Hence, the five camera parameters used for projecting are $\mathcal{P} = [f_X \ f_Y \ \rho \ t_y \ t_z]^T$, where the focal lengths in horizontal and vertical pixel size f_X and f_Y are the camera intrinsic parameters, and ρ , t_y and t_z are the extrinsic parameters shown in Fig. 1(c).

For simplicity, let us consider a camera model with $f_X = f_Y = 1$, known in the literature as *normalized perspective camera model*. The mapping between the \mathcal{F}_I and \mathcal{F}_C coordinates of a ground point gives⁴:

$$x_c = \frac{X t_z}{\sin \rho + Y \cos \rho} \quad (8)$$

$$y_c = \frac{Y t_z}{\sin \rho + Y \cos \rho} \quad (9)$$

$$z_c = \frac{t_z}{\sin \rho + Y \cos \rho} \quad (10)$$

The robot frame coordinates of the ground point can then be easily derived by using the homogenous transformation from \mathcal{F}_R to \mathcal{F}_C (i.e., the camera extrinsic parameters). For the orientation of the tangent at d , we obtain:

$$e_\theta = \text{ATAN2}(-\sin \Theta (\sin \rho + Y \cos \rho) - X \cos \Theta \cos \rho, \cos \Theta) \quad (11)$$

To derive c_d (i.e., the path curvature at d in \mathcal{F}_R), the image path points are initially projected to \mathcal{F}_R . Afterwards, the equation of the path osculating circle in d (thus, the value of c_d) is derived using least square interpolation.

1) *Row controller*: The task of the row controller is to drive (x, y, e_θ) to a desired set point $(\hat{x}, \hat{y}, \hat{e}_\theta)$ under constraint $Y = \text{const} = Y^*$ (i.e., D is constrained to a pixel row in the image plane). This is equivalent to constraining d to the projection of the pixel row on the ground (see Fig. 1(a)), i.e. to the line of equation:

$$y = \text{const} = y^* \quad (12)$$

The above equation, which defines the PF constraint, can be rewritten by introducing the position errors in \mathcal{F}_P :

$$\dot{y} = \frac{d}{dt} (e_n \sin e_\theta - e_t \cos e_\theta) = 0 \quad (13)$$

Using (4), simple calculations lead to:

$$v = v_d \cos e_\theta - \omega x \quad (14)$$

under the constraint that $|e_\theta| \neq \pm \frac{\pi}{2}$, which is plausible, assuming that Γ is never parallel to the X axis of frame \mathcal{F}_I ⁵. Replacing (14) and (4) in:

$$\dot{x} = \frac{d}{dt} (-e_n \cos e_\theta - e_t \sin e_\theta) \quad (15)$$

⁴Equations (8) - (10) do not present singularities, since by construction the image projection of any ground point has $Y > -\tan \rho$.

⁵This singularity can be avoided by temporarily switching to the column controller.

primitive controller	row	column
\mathcal{X}_1	x	\bar{x}
\mathcal{X}_2	e_θ	\bar{e}_θ
\mathcal{A}_1	$\tan e_\theta$	$\tan \bar{e}_\theta \cos \beta - \sin \beta$
\mathcal{A}_2	$-\frac{c_d}{\cos e_\theta}$	$-\frac{c_d \cos \beta}{\cos \bar{e}_\theta}$
\mathcal{B}_1	$y^* + x \tan e_\theta$	$\bar{y}^* + \bar{x} \tan \bar{e}_\theta$
\mathcal{B}_2	$1 - \frac{c_d x}{\cos e_\theta}$	$1 - \frac{c_d \bar{x}}{\cos \bar{e}_\theta}$
\mathcal{G}_1	λ_x	$\bar{\lambda}_x$
\mathcal{G}_2	λ_θ	$\bar{\lambda}_\theta$
\mathcal{E}_1	$x - \hat{x}$	$\bar{x} - \hat{\bar{x}}$
\mathcal{E}_2	$e_\theta - \hat{e}_\theta$	$\bar{e}_\theta - \hat{\bar{e}}_\theta$

TABLE I
COMPONENTS OF: \mathcal{X} , \mathcal{A} , \mathcal{B} , \mathcal{G} , AND \mathcal{E} FOR THE TWO PRIMITIVE CONTROLLERS

The components of \mathcal{X} , \mathcal{A} , \mathcal{B} , \mathcal{G} and \mathcal{E} for the two controllers are recalled in Table I. Hence, the following stability analysis is valid for both controllers. Let us consider the quadratic Lyapunov function candidate:

$$\mathcal{V} = \frac{|\mathcal{E}|^2}{2} \quad (34)$$

Taking the time derivative of this function along a solution of the closed-loop system gives:

$$\dot{\mathcal{V}} = \mathcal{E}^T \dot{\mathcal{X}} \quad (35)$$

Using (31) and (32) leads to:

$$\dot{\mathcal{V}} = \mathcal{E}^T (\mathcal{A}v_d - \mathcal{B}\mathcal{B}^+ (\mathcal{G}\mathcal{E} + \mathcal{A}v_d)) \quad (36)$$

If we set $\mathcal{G}_1 = \mathcal{G}_2 = \mathcal{G}^* > 0$, since $v_d > 0$, and $\mathcal{E}^T \mathcal{B}\mathcal{B}^+ \mathcal{E} = \frac{(\mathcal{B}^T \mathcal{E})^2}{\mathcal{B}^T \mathcal{B}} > 0$ for $\mathcal{B} \neq 0$, then $\dot{\mathcal{V}}$ is negative definite if and only if:

$$\frac{\mathcal{E}^T (\mathcal{A} - \mathcal{B}\mathcal{B}^+ \mathcal{A})}{\mathcal{E}^T \mathcal{B}\mathcal{B}^+ \mathcal{E}} < \frac{\mathcal{G}^*}{v_d} \quad (37)$$

To verify the Lyapunov sufficient condition (37), the robot kinematic constraint on the maximum applicable curvature c_M must be analyzed, since it imposes a constraint on the maximum applicable gain \mathcal{G}^* . Replacing (32) in (2):

$$-c_M + \mathcal{B}^+ \mathcal{A} < -\frac{\mathcal{G}^*}{v_d} \mathcal{B}^T \mathcal{E} < c_M + \mathcal{B}^+ \mathcal{A} \quad (38)$$

From (38), we can derive a sufficient condition for (37):

$$\left| \frac{\mathcal{E}^T (\mathcal{A} - \mathcal{B}\mathcal{B}^+ \mathcal{A})}{\mathcal{E}^T \mathcal{B}} + \mathcal{B}^+ \mathcal{A} \right| < c_M \quad (39)$$

In (39) we have expressed a sufficient condition for asymptotic stability as a condition on the maximum applicable curvature c_M , hence on the robot kinematic model. This condition is also determined by the path characteristics, which must be compliant with the robot nonholonomic constraint. Condition (39) will be verified numerically, depending on the robot parameters and on the desired states, as will be shown in the next section.

V. EXPERIMENTS

In this section, we report the experimental results obtained by applying the proposed path following control scheme. Two experiments are shown in the video clip attached to this paper. All experiments have been carried out with a CyCab. CyCabs are 4 wheel drive, 4 wheel steered intelligent vehicles designed to carry two passengers. In our CyCab, all computations except the low-level control have been executed on a laptop with a 2 GHz Centrino processor. A 70° field of view, forward looking, B&W Marlin F-131B camera is mounted on the robot. The robot is used in car-like mode (i.e., only the front wheels are used for steering), and the camera is used in auto shutter mode, with image resolution 320x240 pixels. The maximum curvature constraint (2) must be considered. In particular, for a car-like robot, it is:

$$c_M = \frac{\tan \phi_M}{L} \quad (40)$$

where ϕ_M is the robot maximum applicable steering angle, and L is the distance between the front and rear wheel axes. For CyCab, $\phi_M = 0.40$ rad and $L = 1.21$ m; thus, $c_M = 0.30 \text{ m}^{-1}$. The system has been coarsely calibrated, and we obtained: $f_X = f_Y = 240$ pixels, $\rho = 0.545$ rad, $t_y = 550$ mm and $t_z = 1625$ mm. The path used in the experiments is composed of two straight lines of length 6 m joined by a 60° arc of circle of radius 10 m (i.e., $c_d = \pm 0.1 \text{ m}^{-1}$, with the sign of c_d depending on the path direction to be followed by the robot). The path features are derived by tracking straight lines and arcs of parabola with the ViSP visual servoing software [15]. The applicable steering angle ϕ used to control CyCab is derived from the angular speed ω (calculated using either (19) or (30)):

$$\phi = \text{ATAN} \frac{L\omega}{v_d} \quad (41)$$

In all the experiments, we set $v_d = 0.2 \text{ ms}^{-1}$.

In order to verify the control robustness, the experiments are repeated by considering a calibration error on the camera parameters. For the calibrated camera experiments, we have numerically verified the sufficient asymptotic stability condition (39) as the system state variables evolve. The state loci that verify (39) are represented in Fig. 4. In our experimental setup, the value of c_d can be 0 (for the straight path portions) or $\pm 0.1 \text{ m}^{-1}$ (for the arc of circle). Hence, in Fig. 4, the state loci are represented for $c_d = 0$ (above) and $c_d = \pm 0.1 \text{ m}^{-1}$ (below). Besides, in the proposed experiments, three instances of the primitive controllers are used: bottom row controller, right column controller, top row controller. Hence, in Fig. 4, the state loci are represented for each of these three controllers: top row (left loci), right column (center), and bottom row (right). For the top row controller, the range of e_θ is discontinuous: $]-\pi, -\frac{\pi}{2}[\cup]\frac{\pi}{2}, \pi]$. Thus, the top row loci are represented by using the orientation of y_d with respect to $-y$ (noted e_θ^t). The desired state values are also indicated in the figure for each controller. Note that in all cases, the desired state values belong to the loci where the asymptotic stability condition is verified. The plots show that even when the norm of the tracking error \mathcal{E} is small (i.e.,

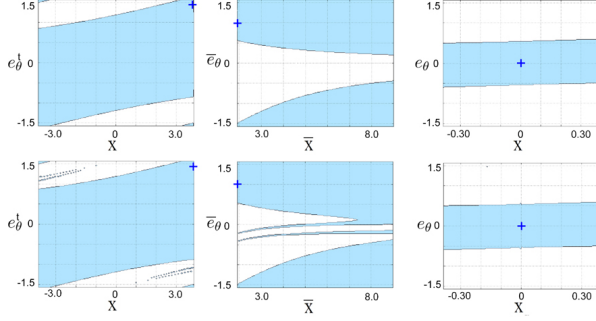


Fig. 4. Loci of the state variables (x, \bar{x} in m, $e_{\theta}^t, e_{\theta}, \bar{e}_{\theta}$ in rad) that verify the Lyapunov sufficient asymptotic stability condition (cyan) for $c_d = 0$ (above) and $c_d = \pm 0.1 \text{ m}^{-1}$ (below), and for: top row controller (left), right column controller (center), and bottom row controller (right). The desired states are indicated with the blue cross.

when (39) is ill-posed), the system is asymptotically stable. In the remainder of this section, the loci of Fig. 4 will be used to verify the asymptotic stability condition during the experiments.

In a first experiment, Cycab is initially positioned on the path with the correct orientation and small initial error: D is on the bottom pixel row of the image plane (see Fig. 5, top left). The row controller is used to drive the states to $\hat{X} = \hat{\Theta} = 0$, with $\mathcal{G}^* = 0.3$. The robot positions and processed images at consecutive time frames while Cycab follows the path are shown in Fig. 5. The evolution of the relevant variables during the experiment is shown in Fig. 6. The robot is able to successfully follow the path, and the tracking errors \mathcal{E}_1 and \mathcal{E}_2 (respectively red and blue curves) are low throughout the experiment. At the end of the experiment, both errors are below 0.10. Both errors increase when the robot reaches the discontinuity in the path curvature (frame 335). Correspondingly, ϕ increases in order to compensate for the error and enables CyCab to follow the curve. Using the right loci in Fig. 4, we verify that the state variables of the bottom row controller (which is the only primitive controller used here) guarantee the asymptotic stability condition throughout the experiment.

In a second experiment, CyCab is initially far from the path, with D on the top pixel row (see Fig. 7, top left). A switching strategy combining both position-based controllers (19) and (30), is used. Initially (phase 1), the row controller (19) is used, to drive point D to a lateral pixel column. Since initially $-\frac{\pi}{2} < e_{\theta} < -\pi$, the controller selects the right side column. We use $\mathcal{G}^* = 24$. Afterwards (phase 2), the column controller (30) is used to drive D along the right pixel column of the image to the bottom right corner. We use $\mathcal{G}^* = 0.4$. Finally (phase 3), the row controller (19) is used, with adaptive gain: $\mathcal{G}^* = 0.34 \exp^{-30||\mathcal{E}||} + 0.02$ ($||\mathcal{E}|| = \sqrt{e_x^2 + e_{\theta}^2}$ is the error norm), to drive D along the bottom row of the image plane to the desired states $\hat{X} = \hat{\Theta} = 0$. The evolution of the relevant variables during the experiment is shown in Fig. 8. The state errors are plotted in the top graphs, for phases 1 to 3 (left to right). The path curvature c_d (purple) and steering angle ϕ (green) are plotted in the bottom graph. Once again, the robot is able to successfully follow the path, and the tracking errors converge. The controller initially saturates the steering angle

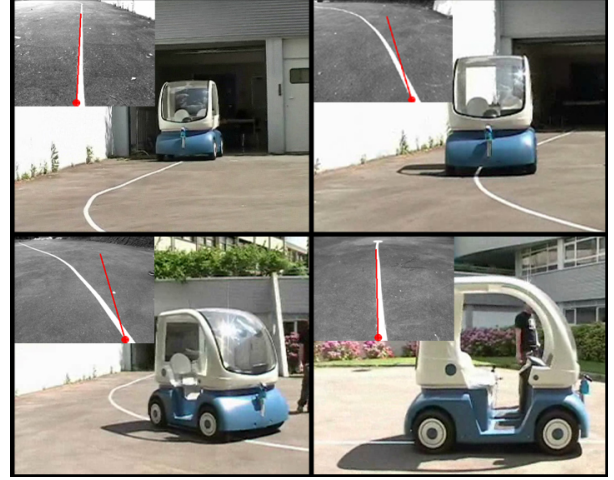


Fig. 5. First experiment: Cycab is initially positioned on the path with small initial error. The robot positions and corresponding processed images at consecutive time frames are shown during PF. The point D and tangent T derived by image processing are indicated respectively by a red circle and a red line.

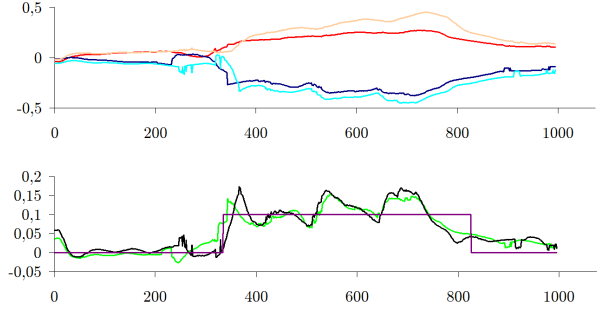


Fig. 6. Evolution of relevant variables during the first experiment. Top: errors \mathcal{E}_1 in m, and \mathcal{E}_2 , in rad (red and blue: correct camera calibration, pink and cyan: coarse calibration). Bottom: c_d in m^{-1} (purple) and ϕ in rad (green: correct camera calibration, black: coarse calibration).

ϕ to its maximum value $\phi_M = 0.40$ rad in order to enable the robot to reach the path. At the end of phase 3, both errors are below 0.10. Note that at the end of phases 1 and 2, the errors on the tangent orientation \mathcal{E}_2 have not reached 0. This occurs because the switching condition is imposed by the error on the position of d (i.e., by the values of \mathcal{E}_1). The iteration steps with state variables not verifying the asymptotic stability condition (i.e., values of \mathcal{X} outside the loci of Fig. 4) are highlighted in yellow in Fig. 8. The plots show that, during most of phase 2 and during the beginning of phase 3, condition (39) is not verified. Nevertheless, the system is able to converge, as outlined above.

The two experiments have been repeated by considering a random calibration error of either +10% or -10% on each of the five camera parameters in \mathcal{P} . The evolution of the relevant variables in the coarse calibration experiments is also shown in Fig. 6 and 8 (pink and cyan for the errors, black for ϕ), for comparison with the calibrated camera experiments. The robot is able to successfully follow the path in all three cases. However, the convergence rate is slightly lower and the steering angle oscillates a bit more than in the calibrated camera experiments.

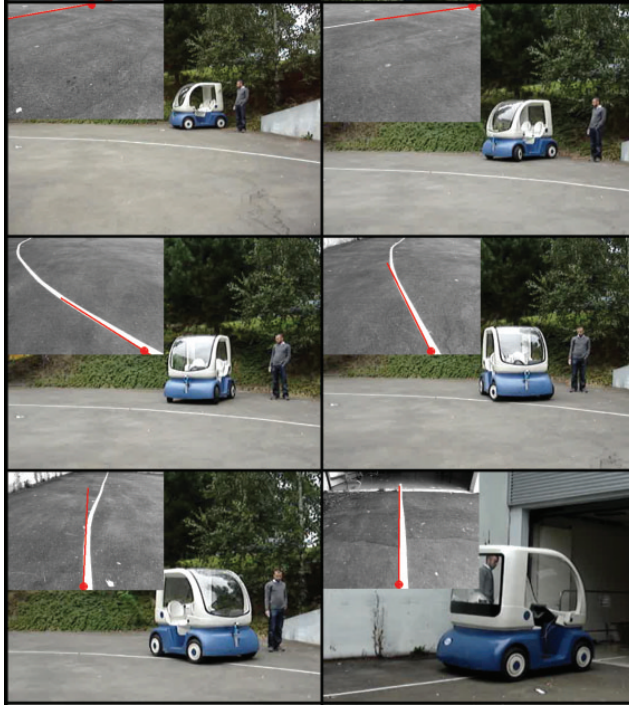


Fig. 7. Second experiment: the initial error is large (D is on the top pixel row of the image plane).

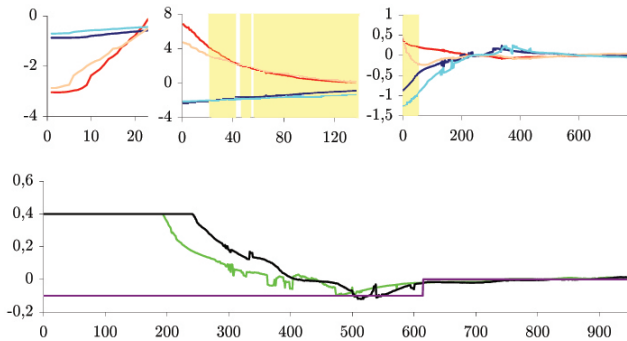


Fig. 8. Evolution of relevant variables during the second experiment. Top: errors during phases 1 to 3 (left to right). \mathcal{E}_1 (in m) and \mathcal{E}_2 (rad) are plotted (red and blue: correct camera calibration, pink and cyan: coarse calibration). The iteration steps with state variables not verifying the asymptotic stability condition are highlighted in yellow. Bottom: c_d in m^{-1} (purple) and ϕ in rad (green: correct camera calibration, black: coarse calibration).

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a position-based visual servoing scheme enabling nonholonomic mobile robots with a fixed pinhole camera to reach and follow a continuous path on the ground. The control scheme utilizes only a small set of path features extracted from the image plane, without using the complete geometric representation of the path. The features are: the position of a path point, and the path tangent orientation and curvature at that point. A major contribution of our work is that the control scheme can be used in general initial conditions, thanks to a switching strategy between two primitive controllers. A Lyapunov-based stability analysis has been carried out. The performance of our controller has been experimentally validated on a car-like robot from two different initial conditions. Robustness was also verified,

by adding camera model errors in the 3 experiments. The system converges in all the experiments, even when the state variables temporarily fail to verify the Lyapunov sufficient asymptotic stability condition. In our opinion, many areas for further work could be explored. Adapting the linear velocity to the path curvature might improve the system performance, without jeopardizing stability. Besides, the control scheme could be used to tackle other classical problems in the nonholonomic mobile robots literature, such as posture stabilization and navigation from an image memory.

VII. ACKNOWLEDGMENTS

The authors thank the members of Projet Lagadic INRIA-IRISA for their support in the CyCab experiments.

REFERENCES

- [1] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation: a Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, 2002.
- [2] F. Chaumette and S. Hutchinson, "Visual servo control tutorial, part I and II", *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, and vol. 14, no. 1, 2007.
- [3] A. Remazeilles, F. Chaumette and P. Gros, "Robot motion control from a visual memory", *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4695–4700, 2004.
- [4] H. H. Abdelkader, Y. Mezouar, N. Andreff, P. Martinet, "Omni-directional Visual Servoing From Polar Lines", *IEEE International Conference on Robotics and Automation*, 2006.
- [5] G. L. Mariottini, G. Oriolo, D. Prattichizzo, "Image-Based Visual Servoing for Nonholonomic Mobile Robots Using Epipolar Geometry", *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 87–100, 2007.
- [6] J. B. Coulaud, G. Campion, G. Bastin and M. De Wan, "Stability analysis of a vision-based control design for an autonomous mobile robot", *IEEE Transactions on Robotics*, 2006.
- [7] C. Canudas de Wit, B. Siciliano and G. Bastin Eds., *Theory of Robot Control, Communication and Control Engineering*, Springer-Verlag, London, 1996.
- [8] F. Diaz del Rio, G. Jimenez, J. L. Sevilano, S. Vicente and A. Civit Balcells, "A generalization of path following for mobile robots", *IEEE International Conference on Robotics and Automation*, 1999.
- [9] M. Bertozzi, A. Broggi and A. Fascioli, "Vision-based intelligent vehicles: state of the art and perspectives", *Robotics and Automation Systems*, vol. 32, pp. 1–16, 2000.
- [10] E. Royer, J. Bom, M. Dhome, B. Thuilot, M. Lhuillier and F. Marmoiton, "Outdoor autonomous navigation using monocular vision", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [11] H.-B. Zhang, K. Yuan S.-Q. Mei and Q.-R. Zhou, "Visual navigation of an automated guided vehicle based on path recognition", *Third International Conference on Machine Learning and Cybernetics*, 2004.
- [12] R. Rao, V. Kumar and C. Taylor, "Visual servoing of a UGV from a UAV using differential flatness", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 743–748, 2003.
- [13] Y. Ma, J. Kosecka and S. Sastry, "Vision-guided navigation for a nonholonomic mobile robot", *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 521–536, 1999.
- [14] S. Skaff, G. Kantor, D. Maiwand and A. Rizzi, "Inertial navigation and visual line following for a dynamical hexapod robot", *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 2, pp. 1808–1813, 2003.
- [15] 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 and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.