2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

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, Universita di Roma La Sapienza , Via Ariosto 25, 00185 Roma, Italy oriolo@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 $\mathcal{F}_{\mathcal{W}}(W, x', y', z')$, robot frame $\mathcal{F}_{\mathcal{R}}(r, x, y, z)$ and image frame $\mathcal{F}_{\mathcal{I}}(I, X, Y)$ (I is the image plane center). The robot state coordinates (i.e., the robot generalized coordinates) are $q(\varepsilon) = [\, x'(\varepsilon)\ y'(\varepsilon)\ \theta(\varepsilon)]^{\mathsf{T}}$, where $\varepsilon \in \mathbb{R}$ is a parameter with infinite domain, $[x'(\varepsilon)\ y'(\varepsilon)]^{\mathsf{T}}$ represent the Cartesian position of r in $\mathcal{F}_{\mathcal{W}}$, 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 $[x\ y\ z]^{\mathsf{T}} = [0\ t_y\ t_z]^{\mathsf{T}}$. We also define the camera frame $\mathcal{F}_C(C, x_c, y_c, z_c)$, shown in Fig. 1(c).

We choose $u = [\, v\ \omega]^{\mathsf{T}}$ 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:

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

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 $\mathcal{P}$. 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 errors $[e_t\ e_n\ e_\theta]^T$, applied (v, ω) and desired (v_d, ω_d) control variables. (b) Image plane view: frame $\mathcal{F}_I$ 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.

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

$$\frac{\omega}{v} < c_M \tag{2}$$

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 $e(\sigma) = q(\sigma) - q_d(\sigma) = [e_{x^0}(\sigma)\ e_{y^0}(\sigma)\ e_\theta(\sigma)]^T$ to a desired value $\hat{e}(\sigma)$. Usually, $\hat{e}(\sigma)$ is zero. The vector $q_d(\sigma) = [x_d^0(\sigma)\ y_d^0(\sigma)\ \theta_d(\sigma)]^T$ defines a desired reference configuration, such that point $p_d(\sigma) = [x_d^0(\sigma)\ y_d^0(\sigma)]^T \in \mathcal{W}$ belongs to $\mathcal{P}$, and $\theta_d(\sigma) \in]-\pi, +\pi]$ is the desired robot orientation (see Fig. 1(a)). We assume that in $\mathcal{F}_W$, the path curve $\mathcal{P}$, can be expressed by a twice differentiable function. Then, $\theta_d(\sigma)$ is the orientation of the path tangent at p_d in $\mathcal{F}_R$[1].

The PF task is often formalized by projecting the $\mathcal{F}_W$ errors $[e_{x^0}(\sigma)\ e_{y^0}(\sigma)\ e_\theta(\sigma)]^T$ to the path frame $\mathcal{F}_P(d; x_d, y_d, z_d)$. Frame $\mathcal{F}_P$ is linked to the path and 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.:

$$\begin{cases} e_t = e_{x^0}\cos\theta_d + e_{y^0}\sin\theta_d \\ e_n = e_{x^0}\sin\theta_d - e_{y^0}\cos\theta_d \\ e_\theta = \theta - \theta_d \end{cases} \tag{3}$$

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

For the nonholonomic model (1), the dynamics of the path errors e_t, e_n and e_θ are:

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

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

$$\omega_d = c_d v_d \tag{5}$$

In opposition to trajectory tracking, where the desired trajectory evolution is determined by rigid laws $\sigma = \sigma(t)$ (i.e., σ is associated to the time t), in PF we can choose the relationship that defines the desired reference configuration $q_d(\sigma)$ 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 $q_d(\sigma)$ (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 $\dot{e}_t = \text{const} = 0$, and the motion exigency as $v = v_d = \text{const} > 0$. For this formulation of the PF problem, the system becomes:

$$\begin{aligned} \dot{e}_n &= -v_d \sin e_\theta \\ \dot{e}_\theta &= \omega - \frac{v_d c_d \cos e_\theta}{1 + e_n c_d} \end{aligned} \tag{6}$$

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

[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 $\mathcal{F}_R$.

[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 $\mathcal{F}_R$.

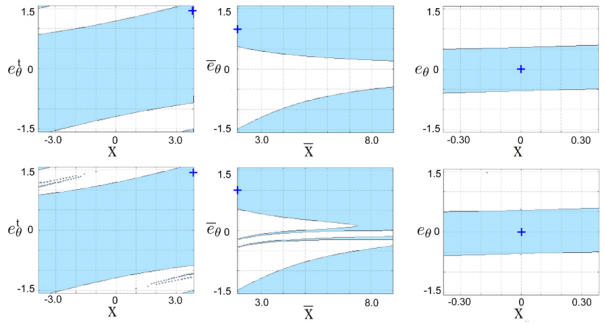

Fig. 4. Loci of the state variables (x, x in m, e^t_θ, e_θ, e_θ in rad) that verify the Lyapunov suf cient asymptotic stability condition (cyan) for c_d = 0 (above) and c_d = $\pm$0.1 m^{-1} (below), and for: top row controller (left), right column controller (center), and bottom row controller (right). 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 rst 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 $\hat{X}$ = ^= 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 $-\frac{}{2} < e < -\pi$, 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 ($||\mathcal{E}||$ = $\sqrt{e_x^2 + e^2}$ is the error norm), to drive D along the bottom row of the image plane to the desired states $\hat{X}$ = ^= 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

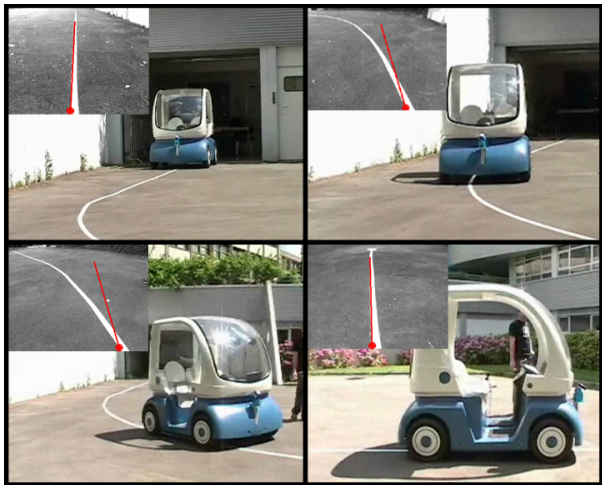

Fig. 5. First experiment: Cycab is initially positioned on the path with small initial error. The robot positions and corresponding processed images at consecutive time frames are shown during PF. The point D and tangent T derived by image processing are indicated respectively by a red circle and a red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

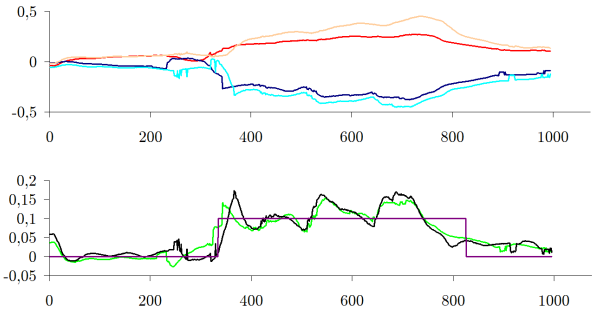

Fig. 6. Evolution of relevant variables during the rst experiment. Top: errors $\mathcal{E}_1$ in m, and $\mathcal{E}_2$, in rad (red and blue: correct camera calibration, pink and cyan: coarse calibration). Bottom: c_d in m^{-1} (purple) and ϕ in rad (green: correct camera calibration, black: coarse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

ϕ to its maximum value ϕ_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 veri ed. 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 ve 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.