Switching Visual Servoing Approach for Stable Corridor Navigation

J.M. Toibero, C.M. Soria, F. Roberti, R. Carelli and P. Fiorini

Abstract— **This paper presents a novel image-based controller for the visual servoing of mobile robots navigating in corridors. Two strategies based on perspective line detection are considered: an image-based approach and a position-based approach. Both are fully described including stability analysis and experimental results on real robots,discussing limitations and benefits of each approach supporting the experimental results with theoretical concepts. Next, it is proposed a stable switching approach in order to obtain a new controller with better performance when compared with both previous controllers acting alone. The stability at switching times is proved by considering Multiple Lyapunov Functions.**

I. Introduction

Wheeled mobile robots are mechanical devices capable of operating in an environment with a certain degree of autonomy. The environment can be classified as *structured* when it is perfectly known and motion can be planned in advance, or as *partially structured* when there are uncertainties that imply some on line planning of motions. Autonomous navigation refers to the capability of capturing environment information through external sensors, such as vision, distance or proximity sensors. Although distance sensors (e.g., ultrasound and laser types) are the most usual sensors to detect obstacles or to measure robot-to-obstacle distances, vision-based sensors are widely used. Vision-based sensors allow to supply a large amount of information from images [2]. Visual servo control allows the robot control through the use of image data. Classically, there have been two approaches to visual servo control: image-based servo control (IBVS) and position-based servo control (PBVS). The main difference yields in the place where control error are measured: in the image for IBVS approaches, whereas for PBVS some features are measured on the image, but used to estimate the current camera position [6]. [8] and [9] present a good survey about visual servoing. In several applications, robots navigate in environments presenting natural or artificial lines which could be used to guide the robot, e.g. the intersection lines between walls and floor at indoor navigation in civil and industrial buildings or the navigation in agricultural applications (e.g. between crop lines in grapevine cultures). As regards motion along corridors, several control algorithms based on artificial vision have been proposed. In [13], image processing is used to detect perspective lines and to guide the robot following the corridor centerline. This work considers a simple control law and does not provide stability analysis. In [3], ceiling perspective lines are employed for robot guidance, although the work lacks a rigorous proof of convergence. In [5], perspective lines are used to find the absolute orientation within the corridor. [4] develop a technique to detect and to approximate line markers by using a second-order polynomial in the image plane. No control algorithm is proposed in this paper. We followed [1] to obtain a stable algorithm that allows the robot navigate at the corridor center. This is the *Position-Based Control Approach* of Sect. III. In this paper, we present a novel *Image-Based Control Approach* for the stable navigation of Wheeled Mobile Robots in corridors using an on-board camera. We prove stability for this controller by considering a combination of the classical feedback linearization scheme along with Lyapunov stability theory for non linear systems in order to obtain stable control commands for the robot velocities. We discuss simulation and experimental results for the performances of both strategies under similar conditions. Next, we state some conclusions about benefits and drawbacks for each approach, and based on these conclusions we propose a switching controller in order to improve the control system performance. Finally, we analyze the stability of this switching controller by considering Multiple Lyapunov Functions, proving the control system stability.

The remainder of this paper is organized as follows: in Section II we introduce the basic concepts related to visual servoing for mobile robots, in particular our focus is on the perspective lines detection and control. Next, in Section III we describe a PBVS controller [1] which analysis give useful information in order to compare performance results, while in Section IV we propose a new strategy based only on the image information avoiding any transformation. Therefore, in Section VI we compare the performances of both control systems approaches using a Pioneer 3AT mobile robot navigating in a real corridor setting. Next it is presented a switching controller based on the vanishing point position on the screen and analyze its stability at switching times. Finally, in Section VII we state some conclusions and future work.

II. Visual Servoing based on Perspective Lines

The robotic system considered in this paper (see Fig. 1) is as in [1]. Composed by a mobile robot with an attached camera and two parallel lines on the floor. These lines may be a texture change on the floor or be the intersection between the floor and the walls in case of corridors inside buildings. Next, we set the notation: let us consider three coordinate systems i) one global $[W]$ attached to the floor, ii) other fixed to the robot $[R]$ and iii) the last fixed

This work was supported by CONICET (Argentina)

J.M. Toibero, C.M. Soria, F. Roberti and R. Carelli are with the Instituto de Automática, Universidad Nacional de San Juan, ARGENTINA mtoibero@inaut.unsj.edu.ar

P. Fiorini is with the Dipartamento di Informatica, Università degli Studi di Verona, ITALIA

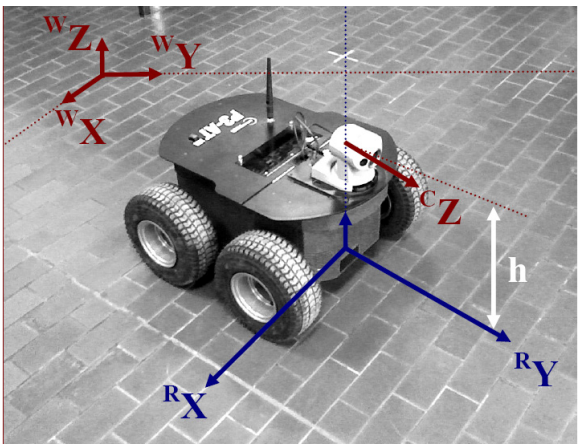


Fig. 1. Coordinate Systems

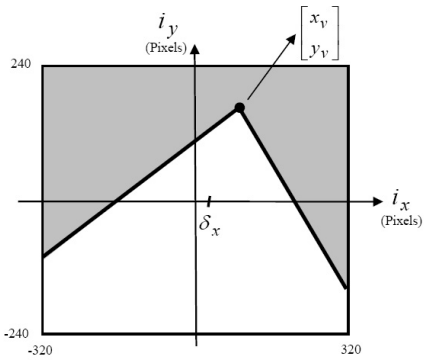


Fig. 2. Image features: vanishing point x_v and middle point δ_x

to the camera $[C]$. A point $\vec{p}$ is represented in coordinate frames $[W]$, $[R]$ and $[C]$ by ${}^W\vec{p}$, ${}^R\vec{p}$ and ${}^C\vec{p}$, respectively. The relative position of the origin of frames $[C]$ and $[R]$ in $[R]$ and $[W]$ respectively are denoted ${}^R\vec{p}_{oc}$ and ${}^W\vec{p}_{or}$. It is also assumed that the origins of coordinate system $[R]$ and $[C]$ lie on the same vertical line at a distance h, and axes ${}^CZ$, ${}^CY$, ${}^RZ$ and ${}^CY$ lie on the same plane. Also, the axes ${}^WX$, ${}^WY$, ${}^RX$ and ${}^RY$ are on the same plane, therefore ${}^W\vec{p}_{zor} = 0$. The tilt camera angle (γ) is assumed constant and $\gamma \neq 0$. This paper considers the mobile robot referred as the *unicycle robot* which has two independently driven wheels on the same axle and a castor wheel. The posture of the robot corresponds to that of the coordinate frame $[R]$ with respect to the world frame $[W]$. To simplify notation, let's define $x \equiv {}^W p_{xor}$ and $y \equiv {}^W p_{yor}$. The posture kinematic model is given by

$$\dot{x} = \nu \sin(\theta); \quad \dot{y} = \nu \cos(\theta); \quad \dot{\theta} = \omega \tag{1}$$

where ν is the linear velocity along the axis ${}^CY$ of the robot, and ω is the angular velocity along ${}^CZ$, which are considered as the robots inputs.

Any point ${}^W\vec{p}$ can be referred to the camera system $[C]$ through a well known convenient transformation [1]. Next, a *imaging model* gives the image feature vector $\vec{i} = [i_x \; i_y]^T$ of the corresponding point ${}^W\vec{p}$ referred to $[W]$. If we assume a pin-hole camera, this relation is the *perspective projection model*

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = -\frac{\lambda}{{}^C p_z} \begin{bmatrix} \alpha_x \, {}^C p_x \\ \alpha_y {}^C p_y \end{bmatrix}, \tag{2}$$

where λ is the focal length; α_x and α_y are the horizontal and vertical camera scale factors. This way, it is possible to obtain a relation between the perspective lines in the workspace and the corresponding projection line in the image space. So, the control objective is to command the mobile robot in such a way that the robot navigates in the middle between the straight parallel lines on the floor at a desired speed $\nu_d \neq 0$. *Remarks.* This goal must be achieved by sensory feedback using only visual information about the parallel lines on the image plane. Besides, the camera is assumed calibrated with respect to the robot frame.

From Fig. 2, the following two key image features can be extracted from the image: the vanishing point $[x_v \; y_v]^T$ and the middle point δ_x. The vanishing point corresponds to the intersection of the perspective lines on the image plane and the middle point is measured from the intersection of these lines and the horizontal axis i_x. Therefore, in terms of these image features (the image error vector), the control objective is achieved provided that

$$\lim_{t\to\infty} \vec{e}_i(t) = \lim_{t\to\infty} \begin{bmatrix} x_v(t) \\ \delta_x(t) \end{bmatrix} = \vec{0} \tag{3}$$

and $\lim_{t\to\infty} \nu(t) = \nu_d$. Without loss of generality, in this paper it will be considered a constant value for the linear velocity $\nu = \nu_d$.

III. Position-Based Controller (PBVS)

This control objective (3) can be formulated in the workspace. If d is the distance between the guidelines on the floor, the robot position error relative to the center of the corridor is

$$\tilde{x} = x - d/2, \tag{4}$$

hence, the control objective is equivalent to

$$\lim_{t\to\infty} \vec{e}_p(t) = \lim_{t\to\infty} \begin{bmatrix} \tilde{x}(t) \\ \theta(t) \end{bmatrix} = \vec{0}. \tag{5}$$

where $\vec{e}_p$ is the position error vector. Next, we must apply a transformation from image features to workspace variables. Considering two parallel guidelines at ${}^WX = 0$ and ${}^WX = d$, the vanishing point can be obtained from

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \begin{bmatrix} k_1 \, \tan\theta \\ -\alpha_y \, \lambda \, \tan\gamma \end{bmatrix} \tag{6}$$

where $k_1 = \alpha_x \, \gamma / \cos\gamma$; whereas the middle point between the intersection of the perspective lines and the horizontal axis i_x on the image plane is,

$$\delta_x = k_2 \frac{\tilde{x}}{\cos\theta} + k_3 \, \tan\theta \tag{7}$$

where $k_2 = -\alpha_x \lambda \sin(\gamma)/h$ and $k_3 = \alpha_x \lambda \cos(\gamma)$. This way, we can write

$$\vec{e}_i = \begin{bmatrix} x_v \\ \delta_x \end{bmatrix} = \begin{bmatrix} k_1 \, \tan\theta \\ \frac{k_2}{cos\theta} \, \tilde{x} + k_3 \, tan\theta \end{bmatrix} = \vec{e}_i(\vec{e}_p) \tag{8}$$

Conversely, the workspace variables θ and $\tilde{x}$ can be written as a function of the image features x_v and δ_x. Note that from (6) and from (7) it results

$$\vec{e_p} = \begin{bmatrix} \theta \\ \tilde{x} \end{bmatrix} = \begin{bmatrix} \frac{\tan\theta}{k_1} \\ \frac{\cos\theta}{k_2}\ (\delta_x - k_3\ \tan\theta) \end{bmatrix} = \vec{e_p}(\vec{e_i}) \tag{9}$$

The proposed control law [7] for the PBVS approach is

$$\omega_p = -k_{pos}\ \tanh\left(k_{pos2}\,\theta\right) - k_{\tilde{x}}\,\tilde{x}\,\nu\,\frac{\sin\left(\theta\right)}{\theta} \tag{10}$$

where $k_{pos} > 0$ is a design constant that sets a desired saturation value for the first term, $k_{pos2} > 0$ can be chosen to affect the range of influence of θ and $k_{\tilde{x}} = K_{\tilde{x}}/(1 + \|\tilde{x}\|) > 0$ with $K_{\tilde{x}} > 0$ also allows to handle saturation of the angular velocity. Considering the following candidate Lyapunov function

$$V_p = \int_0^{\tilde{x}} \eta\, k_{\tilde{x}}(\eta)\, d\eta + \frac{\theta^2}{2} \tag{11}$$

whose time derivative along systems' trajectories is

$$\dot{V}_p = k_{\tilde{x}}\,(\tilde{x})\,\dot{\tilde{x}} + \theta\,\dot{\theta} = -k_{pos}\,\theta\ \tanh\left(k_{pos2}\,\theta\right) \leq 0. \tag{12}$$

Finally, by invoking LaSalle's invariance principle (see e.g. [12]) it can be proved the asymptotic stability of this control system, i.e., condition (5) is fulfilled.

IV. Image-Based Controller (IBVS)

Considering only image information, the control objective is to find the robot commands in such a way that (3) is fulfilled. To this aim, we first consider the time derivatives of $\vec{e_i}$ (8)

$$\dot{\vec{e}}_i = \begin{bmatrix} k_1\ \left(1 + \tan^2\theta\right)\ \omega_i \\ k_2 \tan\theta\nu + \left(k_2\tilde{x}\sec\theta\tan\theta + k_3\sec^2\theta\right)\omega_i \end{bmatrix} \tag{13}$$

from (6) and (9) it follows that

$$\dot{\vec{e}}_i = \begin{bmatrix} \dot{x_v} \\ \dot{\delta_x} \end{bmatrix} = \begin{bmatrix} \frac{k_1^2 + x_v^2}{k_1}\,\omega_i \\ \frac{k_2\,x_v}{k_1}\,\nu + \left(\frac{\delta_x\,x_v}{k_1} + k_3\right)\,\omega_i \end{bmatrix} \tag{14}$$

Considering now that ν is constant, the only command is ω_i, and the two previous equations can be written as

$$\begin{bmatrix} \dot{x_v} \\ \dot{\delta_x} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{k_2\,x_v\,\nu}{k_1} \end{bmatrix} + \begin{bmatrix} \frac{k_1^2 + x_v^2}{k_1} \\ k_3 + \frac{\delta_x\,x_v}{k_1} \end{bmatrix}\,\omega_i \tag{15}$$

From this last equation it is possible to obtain an expression for ω_i which guarantees that $\delta_x \to 0$. To do so, we introduce k_{pI} as a positive design constant

$$\omega_i = \frac{k_1}{k_1\,k_3 + \delta_x\,x_v}\,\left(-\frac{k_2\,\nu.\,x_v}{k_1} - k_{pI}\,\delta_x\right) \tag{16}$$

This way, by replacing (16) in (15)

$$\dot{\delta_x} + k_{pI}\,\delta_x = 0 \tag{17}$$

proving that $\delta_x \to 0$ for $t \to 0$ provided that $k_{pI} > 0$. Additionally, to prove the asymptotic stability it could be considered also the following candidate Lyapunov function

$$V_{i\delta_x} = \frac{\delta_x^2}{2} \tag{18}$$

which derivative along trajectories is given by

$$\dot{V}_{i\delta_x} = \delta_x\,\dot{\delta_x} = \delta_x\,(-k_{pI}\,\delta_x) = -k_{pI}\,\delta_x^2 < 0 \tag{19}$$

Now, we want to prove the asymptotic stability for x_v. To do so, we replace the recently obtained control command: ω_i (16) into the first row of (14)

$$\dot{x_v} = \frac{k_1^2 + x_v^2}{k_1}\,\omega_p \tag{20}$$

$$\dot{x_v} = -\frac{k_1^2 k_2\nu\,x_v + k_2\nu\,x_v^3 + k_1 k_{pI}(k_1^2 + x_v^2)\,\delta_x}{k_1^2\,k_3 + (k_1 x_v)\,\delta_x} \tag{21}$$

Now, we propose the following Lyapunov function candidate [12]

$$V_i = \frac{x_v^2}{2}, \tag{22}$$

being its time-derivative along system trajectories:

$$\dot{V}_i = x_v\,\dot{x}_v = \frac{-k_1^2 k_2\nu\,x_v^2 - k_2\nu\,x_v^4 - K_{aux1}\,\delta_x}{k_1^2\,k_3 + (k_1 x_v)\,\delta_x} \tag{23}$$

where $K_{aux1} = k_1 x_v k_{pI}(k_1^2 + x_v^2)$ was introduced in order to clarify the notation. As we have proved that (16) causes the asymptotic convergence of δ_x to zero, we can eliminate δ_x in the previous equation, obtaining

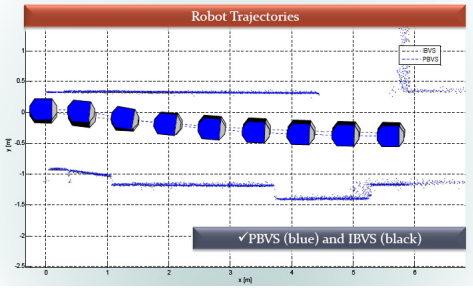
$$\dot{V}_i = -\frac{k_2\nu}{k_3}\,x_v^2 - \frac{k_2\,\nu}{k_1^2\,k_3}\,x_v^4 < 0 \tag{24}$$

So, we have proved that (16) not only guarantees the asymptotic convergence of δ_x to zero, but also that $x_v \to 0$ for $t \to \infty$. Therefore, by considering the proposed control action for the angular velocity we have proved the achievement of the control objective (3).

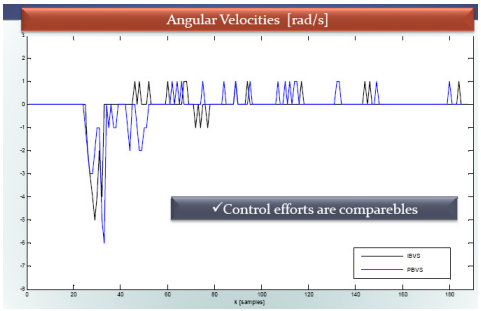
A. Experimental Results

We performed some experimental tests on real office corridors. We considered both strategies, and we report the results in the next figures. The linear velocity was set to 200[mm/s] and the controllers were adjusted in order to obtain comparable control efforts. Figure 3(a) show the robot trajectories and Fig. 3(b) the associated angular velocities.

The robot trajectories are very similar, but analyzing the control states we found that the IBVS strategy shows a better performance when controlling the states on the image: x_v (Fig. 4) and δ_x (Fig. 5); whereas the PBVS strategy seems to work better for the variables on the robot workspace: θ (Fig. 6) and $\tilde{x}$ (Fig. 7).



(a)Robots Trajectories



(b)Angular Velocity

Fig. 3. Robots' trajectories and Angular velocities

Fig. 4. Comparison: X_v

V. Switching Approach for Visual Servoing

Using PBVS or IBVS depends on the expected results. As reported in many papers: specially in [6] for robot manipulators, each strategy have its proper benefits and drawbacks. For the purposes of this paper, we focus the attention on the cases where the image feature is closer to the screen limits. In other words, as PBVS strategies compute the control actions in order to correct the robot position on the corridor regardless of the features position on the screen, one of the main drawbacks could appear when the features are closer to the screen limits. In many of these cases the corresponding PBVS control action could loose the features placing them out of the screen. In fact, the relation between $\dot{x}_v$ and the control action ω is $\sec^2(\theta)$, and consequently, x_v can suddenly change for relatively small variations of the robot heading θ. On the other hand, as IBVS strategies computes the control actions from the errors obtained directly from the screen, it could be expected a better performance in the same cases, i.e., preserving the features inside the screen, but showing a slow convergence

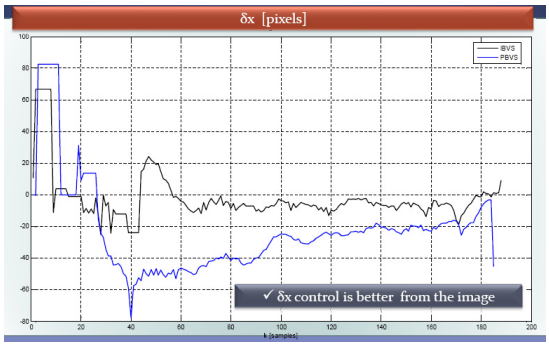


Fig. 5. Comparison: δ_x

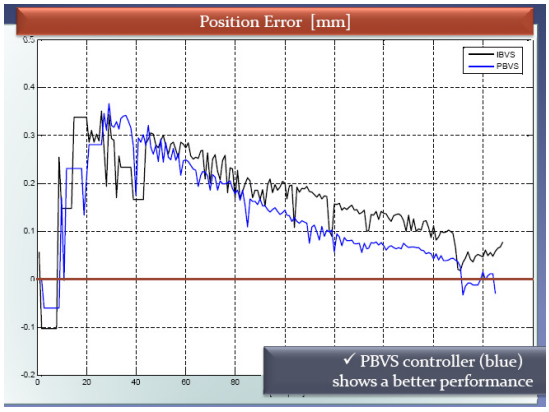


Fig. 6. Comparison: distance to the center of the corridor $\tilde{x}$

for the position states. In practice, it could be then advisable to select a low value for k_{pI} in order to avoid undesired oscillations on the robot and to increase the time to get the center of the corridor for the IBVS. Contrarily, the PBVS approach has a faster response for small $\tilde{x}$-values. Next, we propose a switching controller which is based on the position of the x_v-feature on the image. This way, the image plane is partitioned as appears in Fig.8, where for values of $\|x_v\| \leq x_{vsw}$ the PBVS approach is active and for values of $\|x_v\| > x_{vsw}$ the IBVS is active.

Therefore, the switching controller proposed in this paper is intended to overcome the limitations of each approach acting alone. Namely, the control system will be

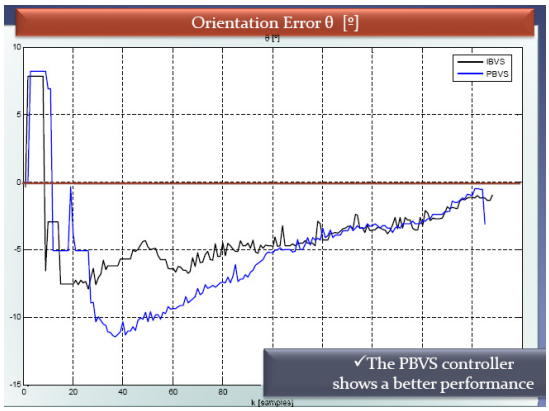


Fig. 7. Comparison: orientation error $\tilde{\theta}$

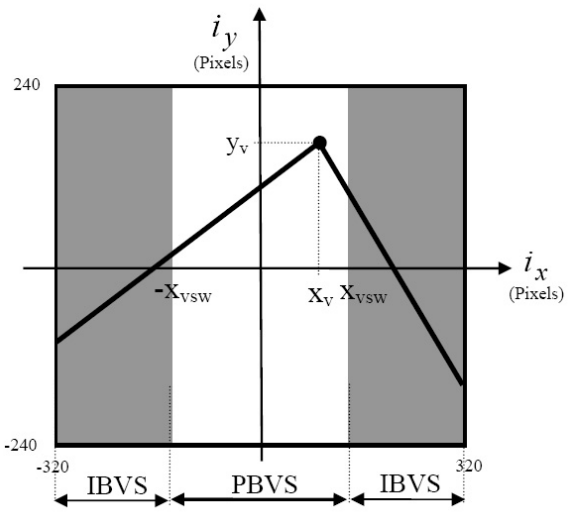


Fig. 8. Switching strategy

able to deal with adverse initial conditions that could be difficult to handle when considering only the PBVS and, on the other hand, it will be also able to handle disturbances that could cause a sudden change on the position for the x_v feature (provided these disturbances are not big enough to directly loss the image features). The selection of the x_v state instead of δ_x to define the switching surface is supported by the fact that x_v is directly related to the robot bearing movements, and this kind of movements are responsible for abrupt variations on the image features positions on the screen. Moreover, the proposal of our switching controller is based on the better (although slower) behavior of the IBVS approach when the features approaches the limits of the screen (since an image-to-position jacobian it is not required when the control is considered directly on the screen) whereas the PBVS's controller adjustments are more intuitive to perform since the errors have physical meaning, and shows a better performance for small values of θ.

A. Stability analysis at switching times

A switching system is the simplest expression of a hybrid system. We consider a switching system compound by two subsystem (both previously described controllers) as can be seen in Fig.9.

The supervisor generates the switching signal σ which

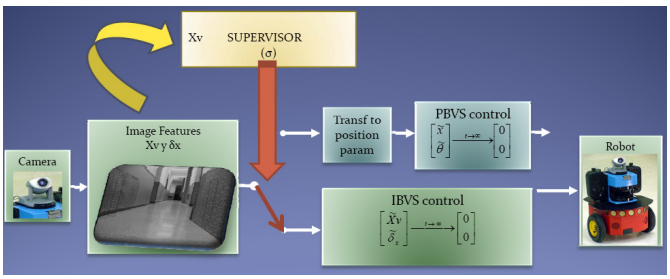


Fig. 9. Proposed Switching System

indicates the active subsystem at any time. This signal must be picked in such a way that there are finite switches in finite time, and is chosen by the controller depending on the X_v-value attempting to improve the control of abrupt bearings. In order to prove Lyapunov stability of an equilibrium point of a hybrid system we may consider the dynamics of its subsystems and the switching rules. Nevertheless, the idea behind theoretical conditions is to find a Lyapunov function that is non-increasing along continuous actions and also over each impulse, then the system will be stable [11].

In order to discuss the stability of our switching system we consider multiple Lyapunov functions [10]. It is a known fact that even if we have Lyapunov functions for each subsystem individually, we need to impose restrictions on switching to guarantee stability. Specifically, each candidate Lyapunov function with an equilibrium point at the origin must be positive definite and must be nonincreasing when its associated subsystem is active, and finally, the sequence of values of each candidate Lyapunov function at the times when became active must be a decreasing sequence. Next, we show that our switched system is stable.

Proposition:
The hybrid switched system compound by the controllers described in Sect.III and Sect.IV which switching rule is:
i) If the PBVS controller is active and $x_v > x_{VSW}$, switch to the IBVS controller at instants t_{2n} with $n = 0, 1, 2, ...$ (priority switch),
ii) If the IBVS controller is active and $x_v \leq x_{VSW}$, switch to the PBVS controller at instants t_{2n+1} only if $V_p(t_{2n+1}) \leq V_p(t_{2n-1})$ or $V_p(t_{2n}) \leq V_p min$. Being $V_p min$ a threshold value which indicates that the control objective was accomplished.
is stable in the sense of Lyapunov.

Proof:
It was established that both Lyapunov have an equilibrium point at the origin, furthermore from (8) and (9): $\vec{e}_i = 0 \Leftrightarrow \vec{e}_p = 0$, i.e., all pose error have an associated features position error and viceversa. Consequently, $V_i(\vec{e}_i) = V_p(\vec{e}_p) = 0 \Leftrightarrow \vec{e}_i = 0 = \vec{e}_p = 0$.
The positive definite property of V_i and V_p is proved by (22) and (11). Also, from the stability analysis of both controllers follows that $\dot{V}_i < 0$ and $\dot{V}_p < 0$, then, each Lyapunov function is non-increasing when the associated controller is active.

The switching logic gives priority to the IBVS controller. This switching is always allowed since it prevents to lose the image features out of the screen (as occur in the switching instants t_2, t_4 and t_6 in Fig.10). The deactivation of the IBVS controller is ruled by V_p, but since it occurs only when $x_v = x_{vsw}$, the discrete sequence associated to V_i at the deactivation instant will be non-increasing. On the other hand, the discrete sequence associated to the activation instants of V_p must be non-increasing. This request is fulfilled since the condition $V_p = V_{p(t_{2n-1})}$ is required to allow the IBVS to PBVS switching (as occurs in t_3) or when the

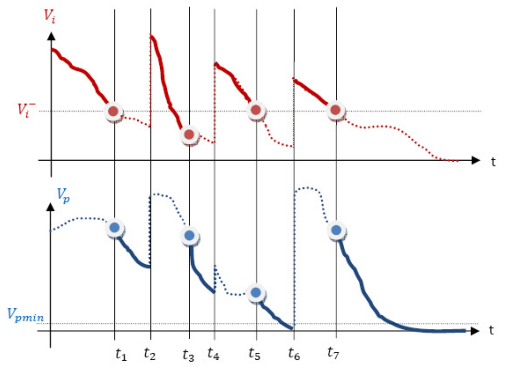


Fig. 10. Lyapunov functions at switching instants

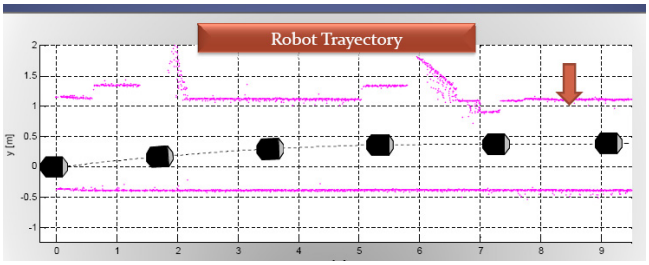


Fig. 11. Switching strategy: robot trajectory

control objective is supposed accomplished at $V_p \leq V_{pmin}$ (as occurs in t_7). This way, the discrete sequences for the deactivation instants of the IBVS and for the activation instants of the PBVS are both non-increasing, establishing a sufficient condition for the stability of this switched control system. In Fig.10 it is shown the behavior of both Lyapunov functions, if the corresponding controller is active the Lyapunov function is denoted with a filled line, otherwise (when is inactive) is denoted with a dotted line.

Note about chattering. As $x_v \rightarrow 0 \Leftrightarrow \theta \rightarrow 0$, chattering effects at the switching surface $x_{VSW} = k_1\, tan(\theta_{SW})$ are avoided since both controllers asymptotically guarantee that $x_v \rightarrow 0$ for the IBVS, or $\theta \rightarrow 0$ for the PBVS.

Note about perturbations. It is assumed that the perturbations which appear at t_2, t_4 causing the IBVS to PBVS switching are such that $x_v \leq x_{MAX}$, given the x_{MAX} a half of the resolution value in pixels.

VI. Experimental results for the Switching Approach

In this Section we show the performance for the switching strategy, under similar experimental conditions when compared with the experiments of the previous section, but aggregating a new perturbation on the corridor in order to force the switching. This perturbation is applied on the corridor wall, changing artificially one of the perspective lines, making the same corridor to appear more narrow after the image processing.

VII. Conclusions

In this paper it has been presented a novel image-based controller for the visual servoing of a mobile robot navi-

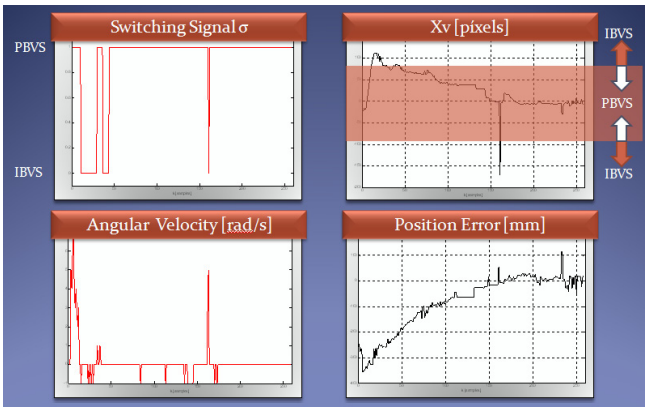


Fig. 12. Switching Controller: Experiment Report

gating in corridors. We focus on two strategies for visual servoing based on perspective line detection: an image-based approach and a position-based approach. Both were fully described including its stability analysis and we have been show experimental results on real robots. Besides, we have proposed a switching controller in order to allow the robot to cope with the loss of the feature problem that typically appears when the robot is commanded under position based strategies.

References

[1] R. Carelli, R. Kelly, O.H. Nasisi, C. Soria and V. Mut, Control based on perspective lines of a nonholonomic mobile robot with camera-on-board, *International Journal of Control*, vol.79, 2006, pp 362-371.

[2] S. Hutchinson, G.D. Hager and P. Corke, A tutorial on visual servoing, *IEEE Transactions on Robotics and Automation*, vol.12, 1996, pp. 651670.

[3] Z. Yang and W. Tsai, Viewing corridors as right parallelepipeds for visionbased vehicle localization, *IEEE Transactions on Industrial Electronics*, vol. 46, 1999, pp. 653661.

[4] H. Schneiderman and M. Nashman, A discriminating feature tracker for vision-based autonomous driving, *IEEE Transactions on Robotics and Automation*,vol.10, 1994, pp. 769-775.

[5] S. Servic and S. Ribaric, Determining the absolute orientation in a corridor using projective geometry and active vision,*IEEE Transactions on Industrial Electronics*,vol. 48, 2001.

[6] N.R. Gans and S.A. Hutchinson, Stable visual servoing through hybrid switched-system control,*IEEE Transactions on Robotics*,vol. 23, 2007.

[7] J.M. Toibero, R. Carelli and B. Kuchen, Stable Wall-following Control for Wheeled Mobile Robots,*ROBOTICA*, vol.27, 2008.

[8] F. Chaumette and S. Hutchinson, Visual Servo Control PART 1: Basic Approches,*IEEE Robotics and Automation Magazine*,vol.6,82-90, 2006.

[9] F. Chaumette and S. Hutchinson, Visual Servo Control PART 2: Advanced Approaches,*IEEE Robotics and Automation Magazine*,vol.7,109-118, 2007.

[10] M.S. Branicky, Multiple Lyapunov Functions and other analysis tools for switched and hybrid systems, *IEEE Trans. on Automatic Control*, vol.43, 1998, pp 475-482.

[11] D. Liberzon, *Switching in Systems and Control*, Birkhauser; 2003.

[12] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice Hall, NY; 1993.

[13] R. Frizera, H.J. Schneebeli and J. SantosVictor, "Visual navigation: combining visual servoing and appearance based methods", *in Proc. of SIRS98, Int. Symp. on Intelligent Robotic Systems*, Edinburgh, Scotland, 1998.