

Position based visual servoing : keeping the object in the field of vision

Benoit Thuilot, Philippe Martinet, Lionel Cordesses and Jean Gallice
LASMEA, Université Blaise Pascal, UMR 6602 du CNRS,
63177 Aubière cedex, France
E-mail: thuilot@lasmea.univ-bpclermont.fr

Abstract

Visual servoing requires an object in the field of view of the camera, in order to control the robot evolution. Otherwise, the virtual link is broken and the control loop cannot continue to be closed.

In this paper, a novel approach is presented in order to guarantee that the object remains in the field of view of the camera during the whole robot motion. It consists in tracking an iteratively computed trajectory. A position based modeling adapted to a moving target object is established, and is used to control the trajectory. A non-linear decoupling approach is then used to control the robot.

Experiments, demonstrating the capabilities of this approach, have been conducted on a cartesian robot connected to a real time vision system, with a CCD camera mounted on the end effector of the robot.

1 Introduction

Over the last decade, visual servoing applications have increased very quickly. The use of a camera in closed-loop control schemes seems to be efficient, and many control laws and modeling have been proposed [6]. However, some problems remain unsolved or not well addressed. Among them, there are the problem of local minima [1], and the problem which occurs when the object, used to control robot motion, does not remain in the field of view of the camera. In this paper, the second problem is addressed.

In classical image based visual servoing ($2D$ approach), this problem occurs when a high value in rotation angle along the optical axis is necessary to perform the servoing task. The regulation task is supposed to make an exponential decay of each visual information: some of them, may not remain in the field of view of the camera.

In position based approach ($3D$ approach), the main goal is to guide the robot with regard to a target object, since an estimation of the $3D$ pose of the robot is obtained from an embedded camera and a reconstruction algorithm. During this servoing task, the target object

has clearly to remain in the field of view of the camera. Nevertheless, as no constraint regarding the trajectory of visual features in image space is used in $3D$ approach, one cannot prevent from the lost of these visual features.

Recently, different approaches have been developed. In [9], the authors use a mixed $2D$ - $3D$ approach in order to take into account for the size of the shape in the image space. They define an ellipsis which includes all the features used in the reconstruction algorithm, and design a control law in order to keep it in the image plane. In [8] and [3], original approaches based on the use of potential fields in image space are presented. More precisely, in [8] some potential fields are defined on the image sides, and are integrated in path planning as a constraint to be satisfied in order to avoid the lost of the object in the field of view of the camera. In [3], potential fields are used to retreat the camera when the object features approach the boundary of the image plane. Such a motion is achievable since the control scheme ensures that the camera translation along the optical axis, and the camera rotation around it, are decoupled from the other camera motions. In [11], a new formalism based on the combination of both robust quadratic stabilization and saturation nonlinearities representations can take into account for any constraint ($2D$, $3D$, saturation). However, most of the time the solution is conservative, and the performances of the control law are weak. In [5], the authors propose to generate intermediate reference images by interpolation in order to take into account for the possible lost of the object or the presence of obstacles.

In this paper, we also propose to rely on a trajectory, but it will be computed on line. Therefore, our approach does not rely on any open-loop path planning technique, but belongs actually to closed-loop control schemes, exhibiting then all the advantages of such techniques. We first present kinematic modeling of the pose features, when the objective is to follow a mobile target. Next, a control law achieving this goal is designed. Then, this result is adapted to ensure that the object remains in the field of vision when achieving positioning tasks. Finally, experimental results on our robotic platform and parallel vision system show the validity of this approach.

2 Notations

In this paper, the following notations are used:

- italic characters, boldface characters and capital boldface characters denote respectively scalars, vectors and matrices.
- $\mathbf{a}_{|i}$ means that vector \mathbf{a} is expressed in frame \mathcal{F}_i .
- $\mathbf{R}_{i/j}$ denotes the rotation matrix that expresses vectors in frame \mathcal{F}_i into vectors in frame \mathcal{F}_j :

$$\mathbf{a}_{|j} = \mathbf{R}_{i/j} \mathbf{a}_{|i}$$

- $\mathbf{v}_{i/j|k}$ and $\boldsymbol{\omega}_{i/j|k}$ denote respectively the translation and rotation velocity vectors of frame \mathcal{F}_i with respect to frame \mathcal{F}_j , expressed in frame \mathcal{F}_k .
- $[\mathbf{a}]_{\times}$ denotes the skew symmetric matrix associated with vector $\mathbf{a} = [a_x, a_y, a_z]^T$:

$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

It is well known that:

$$[\boldsymbol{\omega}_{j/i|i}]_{\times} = \dot{\mathbf{R}}_{j/i} \mathbf{R}_{j/i}^T = -\mathbf{R}_{j/i} \dot{\mathbf{R}}_{j/i}^T \quad (1)$$

3 Modeling

Frames

In the sequel, three frames are considered (see figure 1):

- \mathcal{F}_C denotes the sensor frame (actual camera frame)
- \mathcal{F}_T denotes the target frame (desired camera frame).
- \mathcal{F}_A denotes an absolute frame.

The control objective is to drive the camera in order that \mathcal{F}_C converges to \mathcal{F}_T .

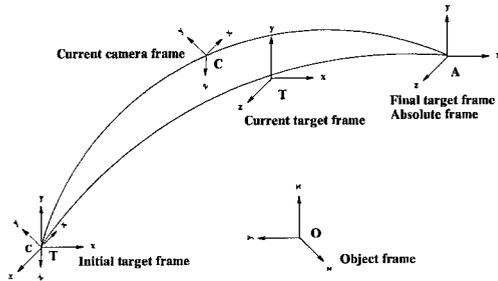


Figure 1: Frames used in the modeling.

System variables

The sensor configuration is hereafter described by the 6-dim. vector $\boldsymbol{\zeta} = [\mathbf{x}^T, \mathbf{y}^T]^T$:

- \mathbf{x} : 3-dim. vector describing the relative position between \mathcal{F}_C and \mathcal{F}_T , namely: $\mathbf{x} = \overline{TC}_{|T}$,
- \mathbf{y} : 3-dim. vector describing the relative orientation of \mathcal{F}_C with respect to \mathcal{F}_T . More precisely, $\mathbf{y} = \sin \theta \mathbf{u}$, where \mathbf{u} and θ are respectively the unitary 3-dim. axis vector and the positive scalar amplitude of the rotation existing between the 2 frames. Alternative orientation representations could also have been chosen, see for instance [12].

In this paper, we address the case where no constraint is placed on the camera motion. The system exhibits then 6 dof. In order to simplify the modeling, the 6-dim. system control vector $\boldsymbol{\tau}$ has been expressed with respect to \mathcal{F}_C :

$$\boldsymbol{\tau} = [\mathbf{v}^T, \boldsymbol{\omega}^T]^T = [\mathbf{v}_{C/A|C}^T, \boldsymbol{\omega}_{C/A|C}^T]^T \quad (2)$$

In the general case, the target frame \mathcal{F}_T is moving. Its twist is hereafter denoted by:

$$\boldsymbol{\tau}_{ref} = [\mathbf{v}_{ref}^T, \boldsymbol{\omega}_{ref}^T]^T = [\mathbf{v}_{T/A|T}^T, \boldsymbol{\omega}_{T/A|T}^T]^T \quad (3)$$

State space model of the system is now derived.

Position state space equation

By definition:

$$\mathbf{x} = \overline{TC}_{|T} = \mathbf{R}_{A/T} \mathbf{x}_{|A} \quad (4)$$

Derivating equation (4) leads to:

$$\dot{\mathbf{x}} = \dot{\mathbf{R}}_{A/T} \mathbf{x}_{|A} + \mathbf{R}_{A/T} \dot{\mathbf{x}}_{|A} \quad (5)$$

Reporting (1) into (5) provides us with:

$$\dot{\mathbf{x}} = [\boldsymbol{\omega}_{A/T|T}]_{\times} \mathbf{R}_{A/T} \mathbf{x}_{|A} + \mathbf{R}_{A/T} \dot{\mathbf{x}}_{|A} \quad (6)$$

Moreover:

$$\begin{aligned} \mathbf{R}_{A/T} \dot{\mathbf{x}}_{|A} &= \mathbf{R}_{A/T} \frac{d}{dt} \left(-\overline{AT}_{|A} + \overline{AC}_{|A} \right) \\ &= \mathbf{R}_{A/T} \left(-\mathbf{v}_{T/A|A} + \mathbf{v}_{C/A|A} \right) \\ &= -\mathbf{R}_{A/T} \mathbf{v}_{T/A|A} + \mathbf{R}_{A/T} \mathbf{R}_{C/A} \mathbf{v}_{C/A|C} \end{aligned} \quad (7)$$

For the sake of simplicity, let us introduce $\mathbf{R} = \mathbf{R}_{C/T}$. Then, by reporting (7) into (6), and by using (2) and (3), we finally obtain:

$$\dot{\mathbf{x}} = -[\boldsymbol{\omega}_{ref}]_{\times} \mathbf{x} - \mathbf{v}_{ref} + \mathbf{R} \mathbf{v} \quad (8)$$

Orientation state space equation

From Rodrigues formula, it can easily be shown that:

$$[\mathbf{y}]_{\times} = \frac{1}{2} (\mathbf{R} - \mathbf{R}^T) \quad (9)$$

Derivating (9) and using (1) leads to:

$$[\dot{\mathbf{y}}]_{\times} = \frac{1}{2} \left([\boldsymbol{\omega}_{C/T|T}]_{\times} \mathbf{R} + \mathbf{R}^T [\boldsymbol{\omega}_{C/T|T}]_{\times} \right) \quad (10)$$

Then, after basic but tedious computations (see [7]), it can be shown that:

$$\dot{\mathbf{y}} = \frac{1}{2} (\text{trace}(\mathbf{R}) \mathbf{I}_3 - \mathbf{R}) \boldsymbol{\omega}_{C/T|T} \quad (11)$$

Considering that:

$$\begin{aligned} \boldsymbol{\omega}_{C/T|T} &= \boldsymbol{\omega}_{C/A|T} + \boldsymbol{\omega}_{A/T|T} \\ &= \mathbf{R} \boldsymbol{\omega} - \boldsymbol{\omega}_{ref} \end{aligned} \quad (12)$$

the final expression is:

$$\dot{\mathbf{y}} = \frac{1}{2} (\text{trace}(\mathbf{R}) \mathbf{I}_3 - \mathbf{R}) (\mathbf{R} \boldsymbol{\omega} - \boldsymbol{\omega}_{ref}) \quad (13)$$

Global state space equations

By merging (8) and (12), state space representation of the system can be written as:

$$\dot{\boldsymbol{\zeta}} = \mathbf{A}_0 \boldsymbol{\zeta} + \mathbf{A}(\boldsymbol{\zeta}) (\boldsymbol{\tau} - \boldsymbol{\tau}_{ref|C}) \quad (14)$$

with:

$$\begin{aligned} \mathbf{A}_0 &= \begin{pmatrix} -[\boldsymbol{\omega}_{ref}]_{\times} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix} \\ \boldsymbol{\tau}_{ref|C} &= \begin{pmatrix} \mathbf{R}^T \mathbf{v}_{ref} \\ \mathbf{R}^T \boldsymbol{\omega}_{ref} \end{pmatrix} \\ \mathbf{A}(\boldsymbol{\zeta}) &= \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{B}(\mathbf{y}) \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R} \end{pmatrix} \\ \mathbf{B}(\mathbf{y}) &= \frac{1}{2} (\text{trace}(\mathbf{R}) \mathbf{I}_3 - \mathbf{R}) \end{aligned} \quad (15)$$

By using Rodrigues formula in (15), it can be established successively (see [7]) that:

$$\begin{aligned} \text{trace}(\mathbf{R}) &= 1 + 2 \cos \theta \\ \mathbf{B}(\mathbf{y}) &= \cos \theta \mathbf{I}_3 - \frac{\sin \theta}{2} [\mathbf{u}]_{\times} - \frac{1 - \cos \theta}{2} [\mathbf{u}]_{\times}^2 \end{aligned} \quad (16)$$

$\mathbf{B}(\mathbf{y})$ can be shown to be invertible provided that $\theta < \frac{\pi}{2}$. Moreover, it can be derived that:

$$\mathbf{B}^{-1}(\mathbf{y}) = \frac{1}{\cos \theta} \mathbf{I}_3 + \frac{\sin \theta}{1 + \cos \theta} [\mathbf{u}]_{\times} + \frac{1 - \cos \theta}{\cos \theta} [\mathbf{u}]_{\times}^2 \quad (17)$$

Particular case : fixed target

When the target frame is motionless (i.e. the robotic task is reduced to a positioning task), state space representation (14) is simplified into:

$$\dot{\boldsymbol{\zeta}} = \mathbf{A}(\boldsymbol{\zeta}) \boldsymbol{\tau} \quad (18)$$

4 Control design

The control objective, i.e. bringing \mathcal{F}_C on \mathcal{F}_T , can be written as bringing $\boldsymbol{\zeta}$ to 0. Nonlinear decoupling approach provides us with the following control law:

$$\boldsymbol{\tau} = \mathbf{A}^{-1}(\boldsymbol{\zeta}) (-\mathbf{K} \boldsymbol{\zeta} - \mathbf{A}_0 \boldsymbol{\zeta}) + \boldsymbol{\tau}_{ref|C} \quad (19)$$

where gain matrix \mathbf{K} is given by $\mathbf{K} = \begin{pmatrix} k_x \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & k_y \mathbf{I}_3 \end{pmatrix}$. Reporting (19) into (14) establishes clearly that $\boldsymbol{\zeta}$ exponentially decreases. Moreover, the convergence of \mathbf{x} and \mathbf{y} are decoupled, and their exponential decay can be adjusted independently by tuning scalar gains k_x and k_y .

When the target is motionless, we recover the classical control law (as previously established in [7]):

$$\boldsymbol{\tau} = -\mathbf{A}^{-1}(\boldsymbol{\zeta}) \mathbf{K} \boldsymbol{\zeta} \quad (20)$$

5 Keeping the object in the camera field of view

We now focus on positioning tasks, i.e. we want to bring \mathcal{F}_C to a special configuration.

Control law (20) requires the on-line measurement of the camera frame configuration $\boldsymbol{\zeta}$. To this end, an object whose geometrical model is known, is located in the scene. As long as it remains in the camera field of view, $\boldsymbol{\zeta}$ can be computed. However, control law (20) does not ensure that this condition is satisfied during the whole camera motion.

Therefore, we propose here to achieve positioning tasks by using general control law (19). The target frame trajectory can then be used to keep the object in the camera field of view. In order to demonstrate the feasibility of this approach, a very simple design of \mathcal{F}_T trajectory is presented herebelow, and has been investigated experimentally, as displayed in the following section. In future works, a more thorough design of \mathcal{F}_T trajectory should however be considered, in order to improve control law (19) features.

As abovementioned, control law (19) ensures that the \mathcal{F}_C translation and orientation behaviours (i.e. behaviours of \mathbf{x} and \mathbf{y}) are decoupled. This feature allows to simplify \mathcal{F}_T trajectory design as follow: we can let \mathbf{y} exponentially decrease without any reference trajectory, and then ensure that the object remains in the camera field of view by only specifying a target frame \mathcal{F}_T translation. In other words, we can consider that \mathcal{F}_T orientation is fixed (namely identical to the orientation of \mathcal{F}_C goal configuration), therefore $\boldsymbol{\omega}_{ref} = 0$, and we have then only to design \mathbf{v}_{ref} .

The most elementary way to keep the object in the camera field of view is to force its frame origin to describe

a straight line in the camera image plane, and to control the distance between the camera and the object such that, not only the object frame origin, but each object feature, belongs to the image plane.

Let us first impose the object frame origin motion. Let $(u, v)^T$ denote the projection of the object frame origin in the image plane, $(u_i, v_i)^T$ and $(u_f, v_f)^T$ denote its initial and final values (as evaluated from respectively \mathcal{F}_C initial and goal configurations), see figure 2. Our objective can then be written as:

$$\begin{cases} u(\mathbf{y}) &= u_i + (u_f - u_i) \alpha(\mathbf{y}) \\ v(\mathbf{y}) &= v_i + (v_f - v_i) \alpha(\mathbf{y}) \end{cases} \quad (21)$$

where $\alpha(\mathbf{y})$ is a smooth function monotonously varying from 0 when $\mathbf{y} = \mathbf{y}_i$ (with \mathbf{y}_i the initial value of \mathbf{y}) to 1 when $\mathbf{y} = \mathbf{0}$. $\alpha(\mathbf{y})$ allows to tune the convergence to the goal location with respect to those of the orientation. We have here chosen $\alpha(\mathbf{y}) = 1 - \frac{\|\mathbf{y}\|}{\|\mathbf{y}_i\|}$. For any values $\|\mathbf{y}_i\| \neq 0$, this relation ensures a suitable trajectory generation. When $\|\mathbf{y}_i\| \approx 0$, $\|\mathbf{y}_i\|$ can be shifted to any arbitrary non null value, and then the main results remain unchanged.

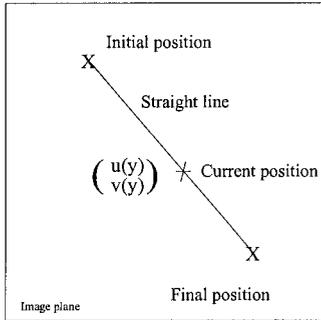


Figure 2: Trajectory in image plane.

In order to completely describe the target frame location, it remains now to specify the depth of the object frame origin in the camera frame. Provided that the projection of that point in the image plane at initial and/or final position is not located on the boundary of the image plane, and provided that the initial and/or final position of the camera is not glued to the object (these conditions are very largely satisfied in most practical applications), one can rely on the simplest choice:

$$z_d = z_i + (z_f - z_i) \alpha(\mathbf{y}) \quad (22)$$

z_i and z_f are respectively the depth of the object frame origin in initial and final camera frames, and z_d is the desired depth when \mathcal{F}_C orientation has reached the value \mathbf{y} . In some rare applications where the 2 abovementioned conditions are not met, relation (22) should then be replaced by another one, dedicated to the application that is considered. In all forthcoming experiments, relation (22) has successively been used, although object features at initial or final position were in a corner of the image plane.

From relations (21)-(22), we can now easily compute, as a function of \mathbf{y} , the location of the camera that would ensure that the object frame origin moves according to a straight line in the image plane. This defines target frame \mathcal{F}_T location. It is then only tedious computations to derive the expression of v_{ref} as a function of \mathbf{y} , see [2].

Control law (19) ensures then that \mathcal{F}_C location converges to \mathcal{F}_T location. Moreover since initial \mathcal{F}_C and \mathcal{F}_T locations are identical (by definition, see figure 2, orientation part does not interfere here), from a practical point of view, \mathcal{F}_C location will always remain very close to \mathcal{F}_T location. This ensures that the object remains always in the camera field of view, as desired.

6 Experimental results

This section presents experimental results obtained with our robotic platform. Our experimental cell is composed of a cartesian robot with 6 dof (see figure 3). A CCD camera is embedded on the end effector and is connected to the vision parallel architecture Windis. The management of the system is ensured under VxWorks Real Time Operating System [10].

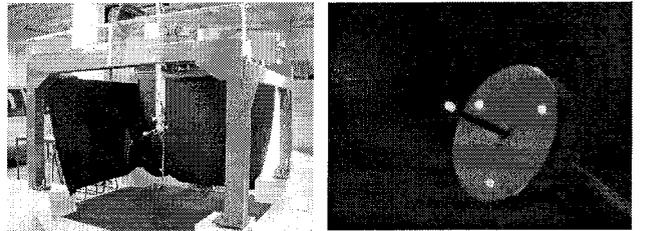


Figure 3: Robotic platform and reference object.

The vision process has been implemented on the Windis architecture. The reference object (see figure 3) is composed of four points (4 LED) which define a tetrahedron. On the low level board, the grey levels and a list of selected pixels corresponding to the highest gradient are extracted. The pose of the object is extracted from the images and an internal model with Dementhon's algorithm [4]. All of this implementation is made at twice video rate (80 ms).

We have defined 2 special camera configurations. In C_1 , the object is centered in image plane (the camera configuration with respect to the object frame is $([0, 0, 0.6], [0, 0, 0])^T$ in (meter, degree)). In C_2 , the object is in the right down corner of the image (the camera configuration is then $([0.86, 0.48, 0.67], [-20, -20, -68])^T$). The experiments consist in moving from one configuration to the other one.

We have first experimented the classical control law (20). All the gain values are fixed to 0.125. Figure 4-right

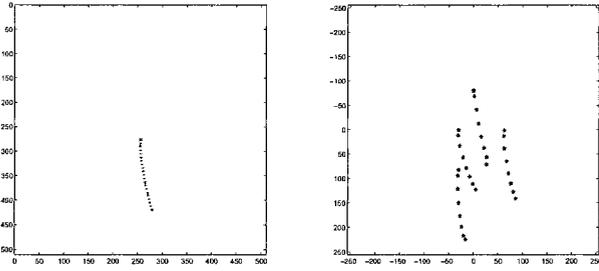


Figure 4: Trajectories in image plane.

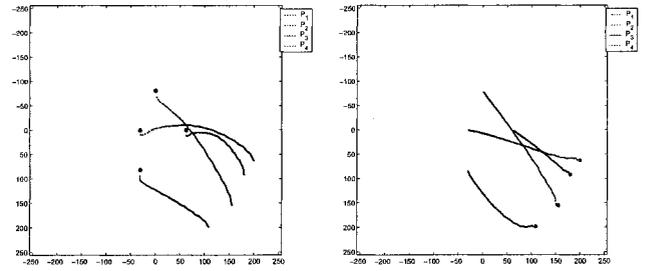


Figure 6: Trajectories of the points in image plane.

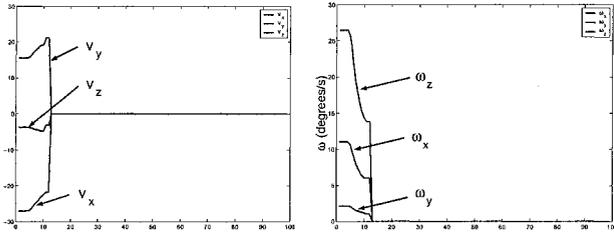


Figure 5: Evolution of the velocities

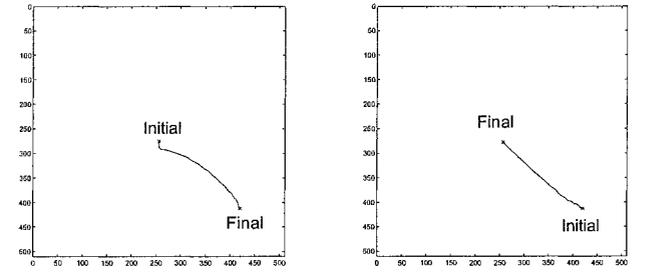


Figure 7: Trajectories of the object frame origin.

shows that LED P_4 leaves the camera field of view. Therefore, the pose of the object can no longer be computed (figure 4-left shows the trajectory of the object frame origin), ζ can no longer be evaluated, and therefore the robot must stop. Figure 5 shows the corresponding translation and rotation velocities.

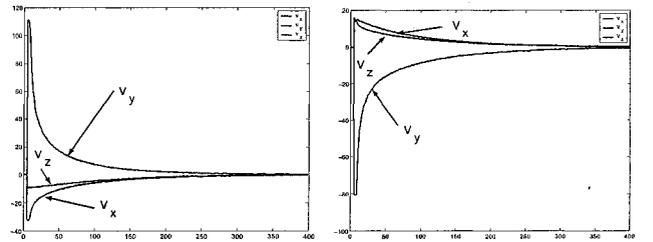


Figure 8: Translation velocities (mm/s)

Secondly, we have experimented the new control law (19). All the gain values are also fixed to 0.125. Left part of figures 6 to 11 deals with a motion from C_1 to C_2 , when the right part deals with the reverse motion. It can be checked (figure 6) that the 4 LED always remain in the camera field of view. Moreover, the trajectory of the object frame origin (figure 7) is close to straight line, as expected. Figures 8 and 9 present the corresponding translation and rotation velocities. Figures 10 and 11 show the evolution of state vector $\zeta = (x^T, y^T)^T$. During the tracking, except at the beginning, the tracking error is always less than 1cm. It persists a noise on the third component of x more important than the others. This is probably due to the depth estimator used in this experimentation.

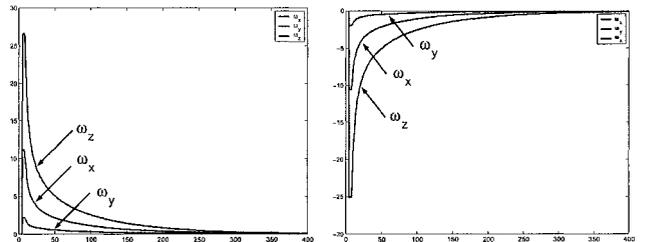


Figure 9: Rotation velocities (degrees/s)

In the last experimentation, we analyze the effect of an increase in the gain values from 0.125 to 0.5. Figure 12 shows that the object frame origin remains always in the field of view of the camera, but its trajectory is no longer a straight line. We think that this effect is due to approximations in kinematic modeling and also to calibration errors (extrinsic and intrinsic parameters of the camera).

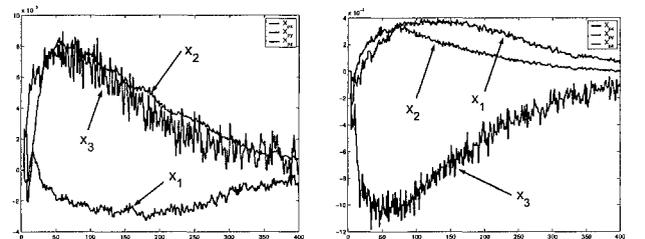


Figure 10: Evolution of the state vector x

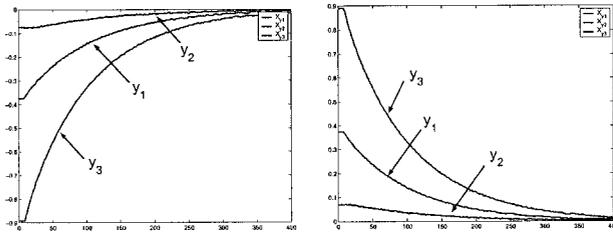


Figure 11: Evolution of the state vector y

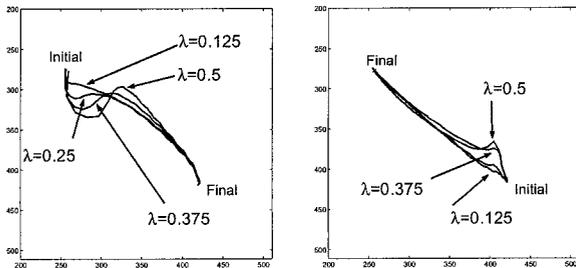


Figure 12: Trajectories of the object frame origin

7 Summary and Conclusions

In this paper, we have proposed a new control strategy to ensure that the object remains in the camera field of view when using a position based visual servoing approach. It consists in tracking a iteratively computed trajectory. The originality of this work is based on the choice of the generated trajectory and on its capabilities to take into account for the current state of the robot. Experimentations have been carried out on our robotic platform, and have demonstrated the validity of this approach.

Nevertheless, many improvements must still be done. In this paper, the design of the target frame trajectory ($\omega_{ref} = 0$, a straight line in image plane) is as simple as possible from a computational point of view. However, from an experimental point of view, there is no guarantee that robot motion is then optimum in some sense. We are currently working on target frame trajectory design, in order to introduce nonlinear curves in image plane which may ensure better performances from robot motion point of view. It could also allow us to deal with nonholonomic constraints on robot motion.

References

- [1] F. Chaumette. *Potential problems of stability and convergence in image-based and position-based visual servoing*. in The Confluence of Vision and Control, D. Kriegman, G. Hager, A.S. Morse (eds.), LNCIS Series, No 237, pp. 66-78, Springer-Verlag, 1998.
- [2] L. Cordesses. *Commande de robots: application à*

l'asservissement visuel 3D et au guidage d'engins agricoles par GPS. PhD Thesis, Clermont-Ferrand University, France, July 2001.

- [3] P.I. Corke and S.A. Hutchinson. *A new hybrid image-based visual servo control scheme*. in Proc. of the IEEE Int. Conf. on Decision and Control, vol. 3, pp. 2521-2526, Sydney, Australia, Dec. 2000.
- [4] D.F. Dementhon and L.S. Davis. *Model-Based Object Pose in 25 Lines of Code*. in the International Journal of Computer Vision, vol. 15, No 1/2, pp. 123-141, June 1995.
- [5] K. Hashimoto and T. Noritsugu. *Potential Problems and Switching Control for Visual Servoing*. In Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, vol. 1, pp. 423-428, Takamatsu, Japan, Nov. 2000.
- [6] S.A. Hutchinson, G.D. Hager and P.I. Corke. *A Tutorial on Visual Servo Control*. In IEEE Transactions on Robotics and Automation, vol.12(5), pp. 651-670, Oct. 1996.
- [7] P. Martinet and J. Gallice. *Position based visual servoing using a nonlinear approach*. In Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, vol. 1, pp. 531-536, Kyongju, Korea, Oct. 1999.
- [8] Y. Mezouar and F. Chaumette. *Design and Tracking of Desirable Image Trajectories under Mechanics and Visibility Constraints*. In Proc. of the IEEE Int. Conf. on Robotics and Automation, vol.1, pp. 731-736, Seoul, Korea, May 2001.
- [9] G. Morel, T. Leibzeit, J. Szewczyk, S. Boudet and J. Pot. *Explicit incorporation of 2D constraints in vision based control of robot manipulators*. In Int. Symp. on Experimental Robotics, P. Corke and J. Trevelyan (eds.), LNCIS Series, No 250, pp. 99-108, Springer Verlag, 2000.
- [10] P. Rives, J.L. Borrelly, J. Gallice and P. Martinet. *A Versatile Parallel Architecture for Visual Servoing Applications*. In Proc. of the Workshop on Computer Architecture for Machine Perception, pp. 400-409, News Orleans, USA, Dec 1993.
- [11] S. Tarbouriech and P. Soueres. *Advanced control strategy for the visual servoing scheme*. In Proc. of the IFAC Int. Symp. on Robot Control, vol. 2, pp. 457-462, Vienna, Austria, Sept. 2000.
- [12] C. Samson, Le Borgne M. and Espiau B. *Robot Control. The task function approach*. ISBN 0-19-8538057. Clarendon Press, Oxford, 1991.