

## Positioning a Camera Parallel to a Plane Using Dynamic Visual Servoing

Armel Crétual    François Chaumette  
IRISA / INRIA Rennes  
Campus de Beaulieu  
35042 Rennes cedex, FRANCE  
E-mail {acretual, chaumett}@irisa.fr

### Abstract

*Visual servoing is generally based upon geometrical features. Recent developments were made in the way of a generalization of this approach to dynamic features. The idea is that velocity in the image can be measured without the constraint of having an a priori knowledge of the scene. In this paper, a control law to position a camera mounted on the end effector of a robot, in such a way the image plane becomes parallel to a planar object, is presented. A dynamic visual servoing approach is used by defining a control loop upon the second order spatial derivatives of the optical flow. A fixating task, which guarantees the object stays in the camera field of view is joined to the first one. Then, results obtained on a 6 d.o.f. robot are laid out for the two studied tasks.*

### 1 Introduction

Vision-based control or visual servoing, as described in [1, 2, 3], proposes to establish a closed-loop control using visual features directly measured in the images acquired by the camera. By using geometric features, a linear relation between variation of these features and the kinematic screw of the camera can be built. Specifying a temporal evolution requested for this features, a simple inversion of the obtained relation gives the translational and rotational velocities of the camera that brings this evolution. This method has been applied for numerous tasks such as target tracking [4, 5], positioning with respect to an object [1, 6], reconstruction of 3D objects by active vision [7].

However, use of geometric features in visual servoing imposes the presence of visual marks on the observed scene, to get the position of their projection in the image. Therefore, a new approach in visual servoing is to extract no more geometric features, but dynamic ones which allows us to turn one's back on this strong a priori constraint. New tasks have been able to be defined, such as alignment of optical axis with an unknown translational camera motion [8], active fixation coupled with contour tracking [9], docking maneuver

[10] and camera self orientation with control of time to contact [11]. In these two last papers, the planar object orientation is estimated from the first order optical flow. Then, a sideways translation in the direction of increasing distance between the camera and the object combined with a fixating rotation ensures the docking with a partially open loop.

Work that we present in this paper consists in establishing a control law by visual servoing that brings the image plane parallelly to a motionless planar object. The chosen approach is to exploit features related to the camera motion in the image. First, in Section 2, we define precisely the task we wish to complete. In Section 3, we briefly recall general concept of visual servoing, in particular when geometric features are used. Then, we develop in Section 4, a control scheme based on the regulation to zero of the second order parameters of the projected motion. Two control law are then built, a first one, where only the camera orientation is controlled and a second one which consists in adding a fixation task to the first one, which constrains the camera position in such a way the same physical point always appears in the image center is then presented. The multi-resolution algorithm of motion parameters estimation used is briefly presented in Section 5. Finally, experimental results obtained on a 6 d.o.f. Cartesian robot are presented in Section 6.

### 2 Task to be performed

The object is considered to be planar and motionless. Vector  $\vec{n}$  represents the normal direction to this plane (see Figure 1). The optical axis of the camera is the axis  $\vec{Z}_c$  of the camera frame. The task consists in positioning the image plane parallelly to the object plane, which comes down to align the optical axis with the normal  $\vec{n}$ . Points  $P_i$  and  $P_c$  represent the physical points projected at the image center in the initial position and after convergence respectively.

To complete the alignment task between the optical axis and the normal to the object plane, which will henceforth be simply called alignment task, the

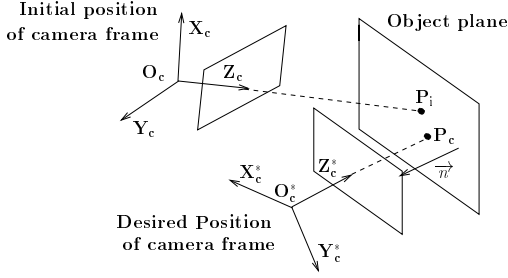


Figure 1: Context and expected result of the task kinematic screw  $V_c$  of the camera is controlled. We therefore wish to define a relation between this screw and the acquired measures.

We wish to solve the alignment problem by two ways. The first one consists in assuming that the plane can always be viewed by the camera. In this case, camera pan and tilt only have to be controlled. In the second one, we want to compensate the displacement of the object in the image due to the rotational motion. A fixation task which always maintains the same point as the intersection of the optical axis and the object, meaning in particular that  $P_c = P_i$ , is thus performed. The rotational d.o.f. being constrained by the alignment task, we complete this fixation using the translational d.o.f. along the orthogonal axes to the optical axis.

Having the image plane parallel to the object plane means that the perspective projection keeps the ratios between all distances and vice versa. On the contrary, if these two planes are not parallel, ratios are not kept. We could have use this property to build our control law, but it would have meant to know the ratio between the 3D lengths of segments. This a priori strong knowledge on the scene drives us to reject this method. Consequently, we have developed an approach based on dynamic information we present now.

### 3 Visual servoing

Let  $V_c = (T, \Omega)^T$  be the kinematic screw between the camera and the fixed frames where  $T = (T_x, T_y, T_z)$  and  $\Omega = (\Omega_x, \Omega_y, \Omega_z)$  represent its translational and rotational component respectively. If  $s$  is a vector of geometric features and the object is motionless, the derivative  $\dot{s}$  can be expressed as a linear function of the screw  $V_c$  in the form:

$$\dot{s} = L V_c$$

where  $L$  is called the interaction matrix related to  $s$  [1]. The vision-based task  $e$  (to be regulated to 0), corresponding to the regulation of  $s$  to a desired value  $s^*$ , is defined by:

$$e = C(s - s^*)$$

where  $C$  is chosen in order to ensure convergence (see condition (1)).

If the error decreasing is wished to be exponential, what means  $\dot{e} = -\lambda e$ , we get from [8]:

$$V_c = -\lambda \hat{L}^+ C^+ e$$

where  $\hat{L}^+$  is the pseudo-inverse of a model or an approximation  $\hat{L}$  of  $L$ .

The exponential decay will be ensured and will remain stable under the condition [8]:

$$C L \hat{L}^+ C^+ > 0 \quad (1)$$

The optimal choice would be to have  $C = L^+$ . However,  $L^+$  is often impossible, or at least, heavy to compute. Therefore, another way is to consider  $C$  as a constant matrix which can be chosen equal to the identity if the number of visual features considered in  $s$  is equal to the number of the camera d.o.f. controlled to perform the task (see [1] for more details).

## 4 Alignment

### 4.1 Associated task function

Let  $Z = Z_p + \gamma_1 X + \gamma_2 Y$  be the equation of the object plane in the camera frame. The condition of parallelism between the image and object planes is conveyed by the fact that this equation checks:

$$\begin{pmatrix} \left( \frac{\partial Z}{\partial X} \right) \\ \left( \frac{\partial Z}{\partial Y} \right) \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2)$$

Let  $(x, y)^T$  be the coordinates of a point in the image plane. Its velocity in the image is a quadratic function of  $x$  and  $y$ , whose parameters can be extracted from two successive images. We can write [12], [13]:

$$\begin{cases} \dot{x} &= a_1 + a_2 x + a_3 y + b_1 x^2 + b_2 xy \\ \dot{y} &= a_4 + a_5 x + a_6 y + b_3 y^2 + b_4 xy \end{cases} \quad (3)$$

with

$$\begin{cases} a_1 = -v_x - \Omega_y & a_4 = -v_y + \Omega_x \\ a_2 = \gamma_1 v_x + v_z & a_5 = \gamma_1 v_y - \Omega_z \\ a_3 = \gamma_2 v_x + \Omega_z & a_6 = \gamma_2 v_y + v_z \\ b_1 = -\gamma_1 v_z - \Omega_y & b_2 = -\gamma_2 v_z + \Omega_x \\ b_3 = b_2 & b_4 = b_1 \end{cases} \quad (4)$$

$$\text{where } (v_x, v_y, v_z) = \frac{1}{Z_p} T$$

If  $v_z \neq 0$ , condition (2) is equivalent to:

$$b_1 + \Omega_y = b_2 - \Omega_x = 0$$

As a consequence, we choose as vector of measures:

$$s = \begin{pmatrix} b_1 + \Omega_y \\ b_2 - \Omega_x \end{pmatrix} = - \begin{pmatrix} \gamma_1 v_z \\ \gamma_2 v_z \end{pmatrix} \quad \text{denoted} \quad \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

We get the following desired value  $s^*$  of vector  $s$  :

$$s^* = (0, 0)^T$$

And consequently, we choose  $e = s$  meaning  $C = \mathbb{I}_2$ .

$\gamma_1$  and  $\gamma_2$  also appear in the linear terms  $a_2, a_3, a_5$  and  $a_6$  of the image flow. In [10] and [11], these linear terms are used to realize alignment. However,  $\gamma_1$  and

$\gamma_2$  are multiplied by  $v_x$  or  $v_y$ , which means that the translational velocity parallel to the image plane,  $T_x$  and  $T_y$ , must not be equal to zero to obtain a measure of the plane orientation. Ensuring such condition does not seem to be realistic and is even contradictory with the realization of the fixation task. That is why we have chosen to use the quadratic terms  $b_1$  and  $b_2$ , combined with a camera motion in the direction of the optical axis.

We now want to determine the link between derivative  $\dot{s}$  and camera pan and tilt,  $\Omega_x$  and  $\Omega_y$ , which are the two camera degrees of freedom devoted to the realization of the alignment. From [12], we get:

$$\begin{aligned} \frac{\dot{Z}_p}{Z_p} &= \gamma_1(v_x + \Omega_y) + \gamma_2(v_y - \Omega_x) - v_z \\ \dot{\gamma}_1 &= \gamma_2(\Omega_z - \gamma_1\Omega_x) + (\gamma_1^2 + 1)\Omega_y \\ \dot{\gamma}_2 &= \gamma_1(\gamma_2\Omega_y - \Omega_z) - (\gamma_2^2 + 1)\Omega_x \end{aligned} \quad (5)$$

The derivative  $\dot{s}$  of the vector of measures  $s$  is:

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \end{pmatrix} = - \begin{pmatrix} \dot{\gamma}_1 v_z + \gamma_1 \frac{\dot{T}_z - \dot{Z}_p v_z}{Z_p} \\ \dot{\gamma}_2 v_z + \gamma_2 \frac{\dot{T}_z - \dot{Z}_p v_z}{Z_p} \end{pmatrix} \quad (6)$$

Substituting (5) in (6) and using a constant velocity  $T_z$  (which means  $\dot{T}_z = 0$ ), we get:

$$\dot{s} = v_z \begin{pmatrix} -\Omega_y + \gamma_1(\gamma_1 v_x + \gamma_2 v_y - v_z) - \gamma_2 \Omega_z \\ \Omega_x + \gamma_2(\gamma_1 v_x + \gamma_2 v_y - v_z) + \gamma_1 \Omega_z \end{pmatrix} \quad (7)$$

This can be expressed under matricial form by:

$$\dot{s} = L \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} + \delta_m + \delta_s \quad (8)$$

with:  $L = \begin{bmatrix} 0 & -v_z \\ v_z & 0 \end{bmatrix}$ ,  $\delta_m = \begin{pmatrix} -\gamma_2 v_z \Omega_z \\ \gamma_1 v_z \Omega_z \end{pmatrix} = \Omega_z \begin{pmatrix} s_2 \\ -s_1 \end{pmatrix}$

and:

$$\begin{aligned} \delta_s &= \begin{pmatrix} \gamma_1 v_z (\gamma_1 v_x + \gamma_2 v_y - v_z) \\ \gamma_2 v_z (\gamma_1 v_x + \gamma_2 v_y - v_z) \end{pmatrix} \\ &= (v_z - \gamma_1 v_x - \gamma_2 v_y) s \end{aligned}$$

## 4.2 Control law

An exponential decrease of the error is expected:

$$\dot{s} = -\lambda s = L \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} + \delta_s + \delta_m$$

Consequently, the control law is given by:

$$\begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} = -\widehat{L}^{-1} (\lambda s + \delta_s + \delta_m)$$

where the estimate  $\widehat{L}$  of  $L$  is:  $\widehat{L} = \begin{bmatrix} 0 & -\widehat{v}_z \\ \widehat{v}_z & 0 \end{bmatrix}$

using as approximation  $\widehat{v}_z$  of  $v_z$  (see (4)):

$$\widehat{v}_z = \frac{a_2 + a_6}{2} \quad (9)$$

In fact, we have  $a_2 + a_6 = 2v_z + \gamma_1 v_x + \gamma_2 v_y$ . However, the term  $\gamma_1 v_x + \gamma_2 v_y$  can be considered negligible in

front of  $v_z$ , and especially near convergence. Indeed we thus have:  $\gamma_1 = \gamma_2 = 0$  (and  $v_x = v_y = 0$  if fixation is also required). Using the same approximation, we obtain  $\delta_s = \widehat{v}_z s$ . The control law thus becomes:

$$\begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} = \frac{1}{\widehat{v}_z} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \left( (\lambda + \widehat{v}_z) \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \Omega_z \begin{pmatrix} s_2 \\ -s_1 \end{pmatrix} \right)$$

where  $s_1 = b_1 + \Omega_y$  and  $s_2 = b_2 - \Omega_x$  are measured using the algorithm presented in the next section (to obtain  $b_1$  and  $b_2$ ) and using the measure of the camera pan and tilt, and where  $\Omega_z$  is a measure of a possible rotation around its optical axis.

Finally, since  $L$  and  $\widehat{L}$  are always of full rank 2 and  $C = \mathbb{I}_2$ , the positivity condition (1) is here given by:

$$C L \widehat{L}^{-1} C^{-1} = \frac{v_z}{\widehat{v}_z} \mathbb{I}_2$$

Thus, convergence will be ensured as soon as  $\widehat{v}_z$  has the same sign as  $v_z$  ( $\neq 0$ ) which appears to be a very weak condition.

## 4.3 Control law for fixation

Fixation is obtained by maintaining to zero the 0<sup>th</sup> order terms  $a_1$  and  $a_4$  of the image velocity, which means we want to regulate  $(v_x + \Omega_y)$  and  $(v_y - \Omega_x)$ . This means:

$$\begin{pmatrix} a_1 \\ a_4 \end{pmatrix} = 0 \Leftrightarrow \begin{cases} v_x = -\Omega_y \\ v_y = \Omega_x \end{cases}$$

Since  $\Omega_x$  and  $\Omega_y$  are controlled by the alignment task, we use  $T_x$  and  $T_y$  for fixation and compensation of the rotational motion in the following form:

$$\begin{aligned} \begin{pmatrix} T_x \\ T_y \end{pmatrix} &= \widehat{Z}_p \begin{pmatrix} -\Omega_y \\ \Omega_x \end{pmatrix} \\ &= \frac{\widehat{Z}_p}{\widehat{v}_z} \left( (\lambda + \widehat{v}_z) \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \Omega_z \begin{pmatrix} s_2 \\ -s_1 \end{pmatrix} \right) \end{aligned}$$

where  $\widehat{Z}_p = \widehat{T}_z / \widehat{v}_z$  is the estimation of the depth  $Z_p$ , and  $\widehat{v}_z$  is now deduced from the following relations:

$$\begin{aligned} a_2 + a_6 &= 2v_z + \gamma_1 v_x + \gamma_2 v_y \\ &= 2v_z - s_3 \frac{\widehat{T}_x}{\widehat{T}_z} - s_4 \frac{\widehat{T}_y}{\widehat{T}_z} \end{aligned}$$

with  $\widehat{T}$  being the measured values of  $T$ . Therefore:

$$\widehat{v}_z = \frac{1}{2} \left( a_2 + a_6 + s_3 \frac{\widehat{T}_x}{\widehat{T}_z} + s_4 \frac{\widehat{T}_y}{\widehat{T}_z} \right)$$

## 5 Motion model estimation

The motion parameters are estimated using the robust multi-resolution estimator (RMRmod) presented in [14]. The image associated to a stage of the multi-resolution pyramid is obtained from the lower one by a classical Gaussian filtering. Let  $\Theta_t$  be the vector of the eight parameters of the quadratic motion model at instant  $t$ . Beginning from the upper one, estimation

in each level consists in a minimization with respect to  $\Theta_t$  of the criterion:

$$\begin{aligned} C(\Theta_t) &= \sum_p \rho(DFD(p, \Theta_t)) \\ &= \sum_p \rho(I_{t+1}(p + d_{\Theta_t}(p)) - I_t(p)) \end{aligned}$$

where points  $p$  are all the points of the image,  $I$  is the intensity function,  $d_{\Theta_t}$  is the displacement due to  $\Theta_t$  and  $\rho$  is Turkey's biweight function.

As  $DFD(p, \Theta_t)$  is not linear in regard to  $\Theta_t$ , an incremental strategy is built. The first value of estimate  $\widehat{\Theta}_t$  is set to null and successive refinements  $\Delta\Theta_t$  are computed using first order approximations  $r_p$  of  $DFD(p, \Theta_t)$ . At each iteration, increment  $\Delta\Theta_t$  is given by:

$$\Delta\Theta_t = \arg \min_{\Delta\Theta_t} \sum_p \rho(r_p(\Delta\Theta_t))$$

Increments are cumulated until a predefined convergence criterion is met. Estimation at next level is initialized by the value reached in the current one. This scheme exploits an Iterative Weighted Last Squares procedure, and in fact, only involves the computation of the spatio-temporal derivative of the intensity function.

## 6 Experimental results

The experimentations are made on a 6 d.o.f. Cartesian AFMA robot and with an SunVideo image processing board providing subsampled images to an Ultra Spark, where the motion parameters estimation is performed. The camera is mounted at the end of the effector to get an "eye-in-hand" system. A typical scene can be seen on Figure (2).

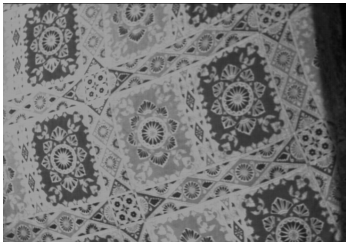


Figure 2: Typical scene

Due to the complexity of the RMRmod algorithm, especially to compute the quadratic terms of motion,  $256 \times 256$  images have to be used to provide sufficient information. Nevertheless, the quadratic parameters are more significative far from the center of the image, and the RMRmod algorithm allows to compute the motion model on a part of the image. Thus, to reduce computation duration, estimation is only performed on the area defined by the  $256 \times 256$  image where a disc of radius 75 pixels has been withdrawn at the center. Therefore, the rate of our control scheme is increased of about 20%, compared with the case where estimation is done over all the image. Nevertheless, this rate is only 1.2 Hz. Consequently, the gain of the control law must be kept very small near convergence

to prevent from oscillations. However, this gain can be higher far from convergence, to decrease time to convergence. Thus,  $\lambda$  has been implemented under the following form:

$$\lambda = \lambda_{min} + \lambda_{max}/n$$

with  $\lambda_{min}$  the value of  $\lambda$  near convergence,  $\lambda_{max}$  a higher value and  $n$  the number of iterations.

### 6.1 Alignment

We implemented the alignment task on the experimental cell with  $\lambda_{min} = 0.04$ ,  $\lambda_{max} = 0.1$  and  $T_z = 1$  cm/s. Initial angular errors were approximately 12 degrees on  $x$  axis and 15 degrees on  $y$  one. In this experiment, where fixation was not ensured, we have set  $T_x = T_y = 0$ . We have also constrained  $\Omega_z = 0$ .

Results are displayed in Figures 3 to 6, with respectively the task function components  $s_1$  and  $s_2$ , the rotation velocities  $\Omega_x$  and  $\Omega_y$  computed by the control law, the angular errors on each axis,  $\arctan(\gamma_1)$  and  $\arctan(\gamma_2)$  and the constant parameters of the motion model  $a_1$  and  $a_4$ . Angular errors are computed by angular difference between the optical axis and the known normal to the plane directions in the fixed frame. Of course, the normal direction is unused in the control law.

Convergence is well obtained and remains stable despite noisy measurements. Rotational parameters are zero at convergence, which guarantees stability. Angular errors decrease quite quickly despite the low gain value. Residual angular errors after convergence (inferior to 1 dg) are, in part, due to uncertainty on this direction of about 1 degree on each axis. Constant parameters are displayed here to be compared with the following alignment and fixation task.

### 6.2 Alignment and fixation

We also implemented the fixation task combined with the alignment one with  $\lambda_{min}$  and  $T_z$  unchanged. 2D motion estimation can not be done when displacements in the image are too great. Fixation task precisely ensures this displacements stay small. Therefore, gain can be higher than without fixation at the beginning of the task. Thus,  $\lambda_{max}$  has been set to 0.25. In the presented experiment, initial angular errors were approximately 30 degrees on  $x$  axis and 25 degrees on  $y$  one. Very important errors can now be taken into account since fixation will allow the camera to always observe the considered object. Finally,  $\Omega_z$  was also kept zero in this case.

Results are presented in Figure 7 to 11 with respectively the task function components, the rotation velocities and the translation ones given by the control law, the angular errors and the constant parameters.

Measurements stays noisy, but convergence is still obtained. Rotational and translational parameters also converge to zero. Convergence can also be noticed

with the decreasing angular errors. Values of constant parameters are 2 times smaller than without fixation at the beginning of control and reach a zero value earlier which ensures a correct realization of the fixation task. By way of comparison, the point projected at the center of the image at convergence have been “manually” retrieved on the initial image for the two cases. Initial position was the one considered in the previous experiment (12 and 15 degrees of angular errors). For the alignment only task, displacements were respectively 116 and 126 pixels upon  $x$  and  $y$  axes, and for the alignment and fixation task, -5 and -20 pixels.

## 7 Conclusion

We presented a vision-based control approach using dynamic features to complete the task of alignment between the optical axis of the camera and the normal direction to a planar object. This regulation using control of the rotation has been completed with a fixation task to ensure that the object is always kept in the image. Those theoretical results have been validated by tests on an experimental eye-in-hand system.

Our major problem is the important duration of each iteration which imposes a small gain with adequate tuning. A first amelioration that could be brought is to make the motion parameters estimation faster. This task can also be improved by positioning the image plane parallelly to a moving object whose projection does not take up a preponderant part of the image. This can be done for the first part by estimation of the object motion and for the other one by motion-based segmentation.

## References

[1] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robotics & Automation*, 8(3):313–326, June 1992.

[2] G. Hager, S. Hutchinson, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics & Automation*, 12(5):651–670, Oct. 1996.

[3] K. Hashimoto, editor. *Visual servoing. Real-time control of robot manipulators based on visual sensory feedback*. World scientific series in robotics and automated systems. World scientific, 1993.

[4] J.T. Feddema and C.S.G. Lee. Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(5):1172–1183, Oct. 1990.

[5] P.K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robotics & Automation*, 9(2):152–165, Apr. 1993.

[6] P. Martinet, F. Berry, and J. Gallice. Use of first derivative of geometric features in visual servoing. In *ICRA'96*, volume 4, pages 3413–3419, Minneapolis, MN, Apr. 1996. IEEE.

[7] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin. Structure from controlled motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):492–504, May 1996.

[8] V. Sundareswaran, P. Bouthemy, and F. Chaumette. Exploiting image motion for active vision in a visual servoing framework. *International Journal of Robotics Research*, 15(6):629–645, Dec. 1996.

[9] C. Colombo, B. Allotta, and P. Dario. Affine visual servoing : A framework for relative positioning with a robot. In *ICRA'95*, pages 464–471, Nagoya, Japan, May 1995.

[10] J. Santos-Victor and G. Sandini. Docking behavior via active perception. In *Int. Symposium on Intelligent Robotic Systems*, Pisa, July 1995.

[11] P. Questa, E. Grossmann, and G. Sandini. Camera self orientation and docking maneuver using normal flow. In *SPIE AeroSense'95*, Orlando, Florida, Apr. 1995.

[12] M. Subbarao and A. Waxman. Closed-form solutions to image equations for planar surface in motion. *Computer Vision, Graphics, and Image Processings*, 36(2):208–228, Nov. 1986.

[13] S. Negahdaripour and S. Lee. Motion recovery from image sequences using only first order optical flow information. *International Journal of Computer Vision*, 9(3):163–184, Dec. 1992.

[14] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, Dec. 1995.

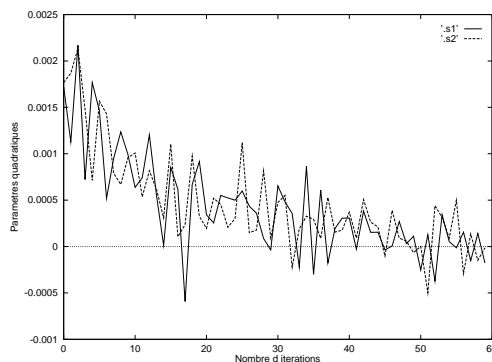


Figure 3: Alignment task: task function

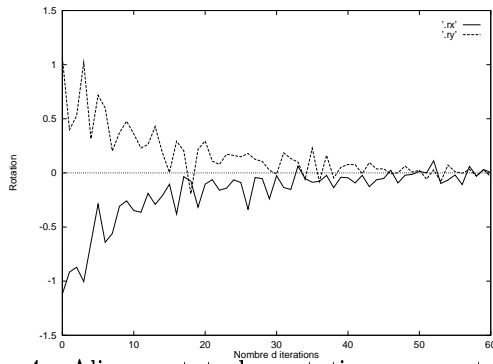


Figure 4: Alignment task: rotation parameters (in dg/s)

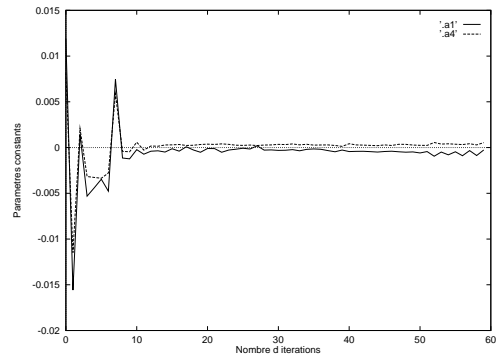


Figure 8: Alignment and fixation task: constant parameters

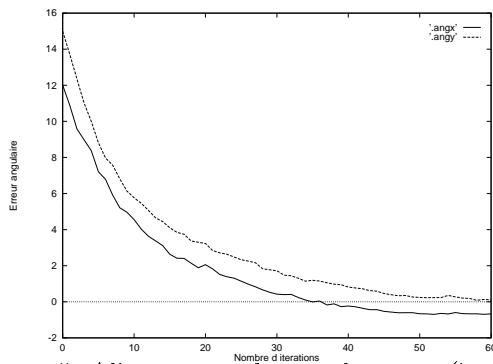


Figure 5: Alignment task: angular error (in dg)

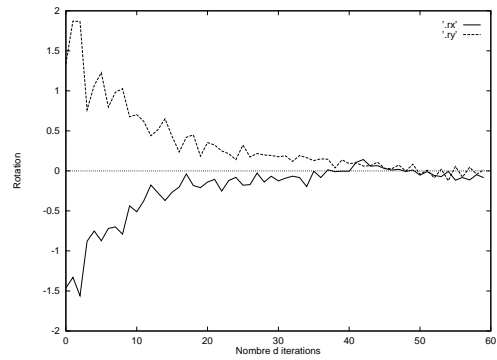


Figure 9: Alignment and fixation task: rotation parameters (in dg/s)

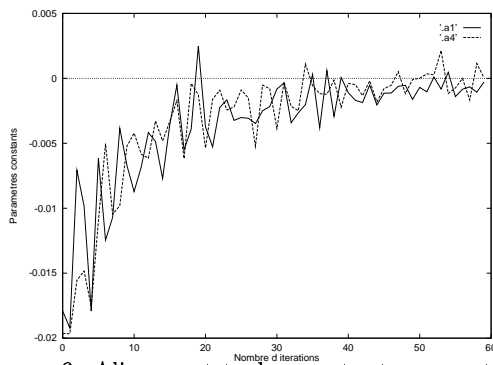


Figure 6: Alignment task: constant parameters

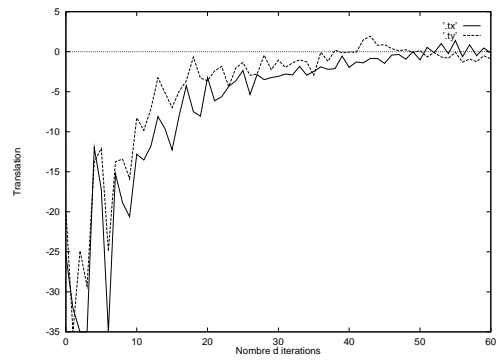


Figure 10: Alignment and fixation task: translation parameters (in mm/s)

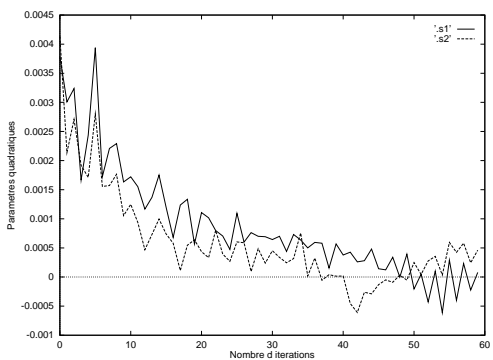


Figure 7: Alignment and fixation task: task function

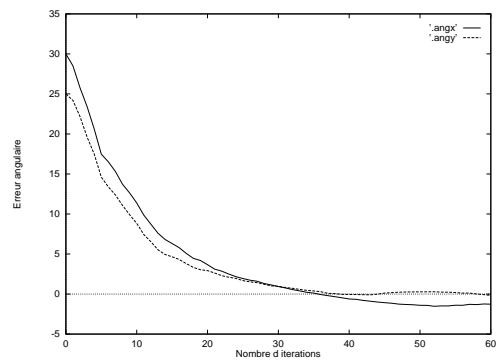


Figure 11: Alignment and fixation task: angular error (in dg)