Trajectory generation by visual servoing

F. Berry, P. Martinet, J. Gallice
Laboratoire des Sciences et Matériaux pour l'Electronique et dAutomatique
Université Blaise Pascal de Clermont Ferrand
UMR 6602 du C.N.R.S, F-63177 Aubière Cedex, France
E-Mail: berry, martinet, gallice@lasmea.univ-bpclermont.fr

Keywords: *Active vision, Visual servoing, Trajectory generation, Time varying reference feature.*

Abstract

This paper describes a new approach of the problem of trajectory generation in workspace by visual servoing. Visual servoing is based on an array of measurements taken from a set of images and used each time as an error function to compute a control vector. This is applied to the system (robot and camera) and enables it to move in order to reach a desired situation, at the end of the task, directly depicted in the image. The originality of this work is based on the concept of time varying reference feature. Classically, in visual servoing, the reference features are static and the task to be achieved is similar to a positioning task. We define a specific task function which allows us to take into account the time varying aspect and we synthesize a new control law in the sensor space. This control law ensure the trajectory control in the workspace. Considering that any trajectories in workspace can be depicted as a combination of rotation and translation, we have tested our new approach using these two elementary trajectories.

1 Introduction

The goal of this paper is to show the use of visual servoing in order to generate a trajectory in the robot workspace.
For the last couple of years, developments in the field of visual sensors permitted their use in a closed loop control. However, works dealing with the integration of visual information in a robot control loop show two possible approaches [19].

The "look and move" approach is a technic where the pose of the robot is estimated with the output data of the vision system. Then, from this estimation, it is possible to control the robot in the workspace. However, even if this sort of visual servoing was preferred for a long time because of its simplicity of adjustment, various kinds of errors cause interferences [18], [6]. First of all, there are errors in the visual extraction due to the visual sensor itself. Then, errors in the modeling of the robot which are due to the difficulty of considering all the physical and electrical characteristics of the robotic system. Finally, there are errors in robot pose estimation, arising from a lack of precision in the camera calibration, and some interpretation errors.

"Visual servoing" approach eliminates these errors by introducing the sensor information directly in the control loop [6], [11], [3]. For that, Samson and al. in [2] developed the formalism of the task function where the control is directly specified in terms of regulation in the sensor space. The robotic task to achieve is a positioning task in relation with a fixed or a moving target. Many works were done using a moving target object. The object motion can be estimated directly in image space [5], [7], [14] or in the robot workspace [9], [21]. In this case, the use of a predictive filter (i.e Kalman filter) is absolutely essential to ensure an accurate object tracking.
For motionless objects, in visual servoing approach, we find many references in [1], [12], [20], [3] and [15] too. The use of the task function concept defined in sensor space allows us to introduce the notion of hybrid task. This task is made of a combination of a primary task, which realizes the visual servoing, and a secondary task. The secondary task can be considered as a minimization of a cost function. The main applications are trajectory tracking [7], [13], singularities and joints limit avoidance [4].
To complete this kind of task, it is necessary to have n degrees of freedom which are let available by the primary task [6], [8]. However, when the primary task needs m degrees of freedom in workspace to be well performed (with m=6-n, for a 6 d.o.f robot), the use of a secondary task which needs n' (n'>n) d.o.f becomes impossible. In this paper, we propose an alternative way of solving the problem of trajectory generation [10] in robot workspace. Therefore we introduce the concept of time varying reference feature com-

puted from the desired sensor trajectory in workspace.

In the first part, we adapt the task function formalism to take into account the concept of time varying reference feature and we deduce the expression of a new control law. In the second part, we consider two elementary trajectories (Translation and Rotation). We explain how to obtain the expression of the time varying visual features. Finally, we present the first results.

2 Visual servoing

Let us consider a sensor ζ fixed on the 6 d.o.f robot end effector. The local environment of the sensor is made of a target χ. The sensor signal $\underline{s}$ provided by ζ represents the image of χ. We shall note the state of the robot in the joint space by $\underline{q}$ and its pose in the workspace by $\underline{r}$. For the following application, the workspace is the world cartesian coordinate space. We suppose the variations of $\underline{s}$ depend only on the relative motion of χ with respect to ζ and we may write the differential as:

$$\frac{d\underline{s}(\underline{q}(t),t)}{dt} = \frac{\partial \underline{s}}{\partial \underline{q}} \cdot \frac{d\underline{q}}{dt} + \frac{\partial \underline{s}}{\partial t} \tag{1}$$

The term $\frac{\partial \underline{s}}{\partial t}$ denotes the velocity of the target. It may be zero in many cases or often unknown in other applications, such as target tracking. In our case, we consider a motionless target ($i.e \frac{\partial \underline{s}}{\partial t} = 0$).

The first term may be written like:

$$\frac{\partial \underline{s}}{\partial \underline{q}} \cdot \frac{d\underline{q}}{dt} = \frac{\partial \underline{s}}{\partial \underline{r}} \cdot \frac{d\underline{r}}{d\underline{q}} \cdot \frac{d\underline{q}}{dt} \tag{2}$$

where $\frac{\partial \underline{r}}{\partial \underline{q}}$ represents the robot jacobian, which is only dependent on the robot itself. The following formalism will be developed only in robot workspace. So 1 may be rewritten as:

$$\frac{d\underline{s}(\underline{r}(t),t)}{dt} = \frac{\partial \underline{s}}{\partial \underline{r}} \cdot \frac{\partial \underline{r}}{dt} \tag{3}$$

where $\frac{\partial \underline{r}}{dt}$=T=$(\overrightarrow{V}, \overrightarrow{\Omega})$ is the kinematic screw.

It represents the relative velocity between the camera and its environment. The term $\frac{\partial \underline{s}}{\partial \underline{r}}$, called interaction screw, characterizes the interaction between the sensor and its environment. The concept of interaction screw is fundamental for modeling systems using exteroceptive sensor. It allows to take into account most information required to design and analyze sensor based control schemes. We note L^T the interaction matrix or image jacobian associated with the interaction screw. Generally the goal of a sensor based control may be expressed as a regulation to zero of an output function $\underline{e}(\underline{r},t)$ called task function [2]. In visual servoing, the task function has the following expression:

$$\underline{e}(\underline{r},t) = C[\underline{s}(\underline{r},t) - \underline{s}^*] \tag{4}$$

where:

- $\underline{s}^*$ is considered as a reference target image to be reached in the image frame.
- $\underline{s}(\underline{r},t)$ is the value of visual information currently observed by the camera. This information depends on the situation between the sensor and the scene (noted $\underline{r}$).
- C is a constant matrix, which allows to take into account more visual information than the number of robot degrees of freedom, with good conditions of stability and robustness.

Until now, all applications were limited to positioning or tracking task with which the vector $\underline{s}^*$ is constant. We introduce the concept of time varying reference feature which allows us to define a new task function:

$$\underline{e}(\underline{r},t) = C[\underline{s}(\underline{r},t) - \underline{s}^*(t)] \tag{5}$$

The trajectory of $\underline{s}^*(t)$, in sensor space, leads to the desired trajectory in the workspace. We consider an exponential decay of all components of the task function $\underline{e}(\underline{r},t)$, so:

$$\dot{\underline{e}}(\underline{r},t) = -\lambda \underline{e}(\underline{r},t) \tag{6}$$

(λ is a positive scalar constant) and we have:

$$\dot{\underline{e}}(\underline{r},t) = \frac{\partial \underline{e}}{\partial \underline{r}} \cdot \frac{\partial \underline{r}}{dt} + \frac{\partial \underline{e}}{\partial t}$$

From 6, we get:

$$\frac{d\underline{r}}{dt} = \left(\frac{\partial \underline{e}}{\partial \underline{r}}\right)^+ \cdot \left(-\lambda \underline{e} - \frac{\partial \underline{e}}{\partial t}\right) \tag{7}$$

where $\frac{d\underline{r}}{dt}$ is the velocity screw. Usually $\left(\frac{\partial \underline{e}}{\partial \underline{r}}\right)^+$ can be considered as the identity matrix, which means to choose $C = \widehat{L^T}^+$. We can rewrite 7 in the following form:

$$T = -\lambda \underline{e} - \frac{\partial \underline{e}}{\partial t} \tag{8}$$

From the derivation of equation 5, we obtain:

$$\left(\frac{\partial \underline{e}}{\partial t}\right) = C \cdot \left(\frac{\partial \underline{s}(\underline{r},t)}{\partial t} - \frac{d\underline{s}^*(t)}{dt}\right) \tag{9}$$

If we consider a motionless target $\left(\frac{\partial \underline{s}(\underline{r},t)}{\partial t} = 0\right)$, we obtain:

$$\left(\frac{\partial \underline{e}}{\partial t}\right) = -C\frac{d\underline{s}^*(t)}{dt}$$

Finally, we obtain the following control law:

$$T = -\lambda L^{T+} \cdot (\underline{s}(\underline{r},t) - \underline{s}^*(t)) + L^{T+}\frac{d\underline{s}^*(t)}{dt} \qquad (10)$$

which ensures the trajectory generation. In relation 10, T is composed of two terms. The first term ensures the visual servoing to maintain a rigid link between sensor and target. The second term expresses the influence of the trajectory generation. It allows to compensate the tracking error with a large efficiency. In effect, if we consider the monovariable case, the visual servoing scheme becomes like in figure 1, where $\ell = L^T$ is the image jacobian. In this case and without taking into account any noise, we get $S = S^*(t)$ without any tracking error. If we suppress the derivated term $\frac{dS^*(t)}{dt}$ in the visual servoing scheme, we have a tracking error proportional to $1/\lambda$.

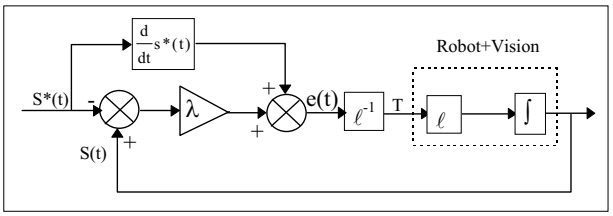

Figure 1: Servoing Scheme

3 Trajectory generation

In this part, we present the trajectory generation in workspace from the concept of time varying reference feature. Consider a point M links rigidly to a fixed object with coordinates (x, y, z) expressed in a moving reference frame $\Re$ (camera frame). The velocity screw $T = (\vec{V}, \vec{\Omega})$ is applied to $\Re$. The evolution of this point is given by the well known kinematic equation:

$$\frac{d}{dt}\overrightarrow{OM} = -V(O) - \Omega \wedge \overrightarrow{OM}$$

If we use the matrix notation (i.e $\underline{p} = (x, y, z)^T$, $V = (Vx, Vy, Vz)^T$, $\omega\tilde{r} \equiv \vec{r}\wedge$), we get the relation:

$$\frac{d}{dt}\underline{p} = -V - \omega\ \tilde{r}\ \underline{p} \qquad (11)$$

where ω is the amplitude of the rotation velocity, and $\tilde{r}$ is the unitary anti symmetric matrix:

$$\tilde{r} = \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{pmatrix}$$

As long as $\underline{V}$ and $\omega\tilde{r}$, expressed in the camera frame $\Re$ are constant, the general solution of equation 11 is given by:

$$\underline{p}(t) = \exp(-\omega\ \tilde{r}\ \Delta t)\underline{p}(t_o) - \exp(-\omega\ \tilde{r}\ \Delta t) \int_0^{\Delta t} \exp(\omega\ \tilde{r}\ \tau)d\tau\underline{V} \qquad (12)$$

with $\Delta t = t - t_o$.

Considering that $\tilde{r}^3 = -\tilde{r}$, it is possible to develop the exponential term as:

$$\exp(-\omega\ \tilde{r}\ \Delta t) = I_3 - \sin(\omega\Delta t)\ \tilde{r} + (1 - \cos(\omega\Delta t))\ \tilde{r}^2 \qquad (13)$$

Using 13 in 12, we express $\underline{p}(t)$ with:

$$\underline{p}(t) = \left(I_3 - \sin(\omega\Delta t)\ \tilde{r} + (1 - \cos(\omega\Delta t))\ \tilde{r}^2\right)\underline{p}(t_o) - \left(I_3 - \frac{1-\cos(\omega\Delta t)}{\omega\Delta t}\ \tilde{r} + \frac{\omega\Delta t - \sin(\omega\Delta t)}{\omega\Delta t}\ \tilde{r}^2\right)\underline{V}\Delta t \qquad (14)$$

Now, we used this relation to define a reference trajectory in order to achieve two specific tasks. We consider in the first section a simple translation along a fixed vector. In a second section, we present a rotation around a given axis linked to the object.

3.1 Translation

The generation of time varying reference feature in case of translation along a vector is relatively easy. Let us consider:

- $\vec{d}$: the displacement during Δt.
- $\vec{v}$: the translation velocity.

M is a fixed point such that $\overrightarrow{OM} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ at time t in $\Re$ and its image depicted by the sensor is given by $(X\ Y)^T$ with:

$$X = \frac{x}{z}.F_x \quad and \quad Y = \frac{y}{z}.F_y \qquad (15)$$

where F_x and F_y are the intrinsic parameters of the camera.

From 14, we get the new position at time $t + \Delta t$ (where $\omega = 0$),

$$\underline{p}(t + \Delta t) = \underline{p}(t) - \underline{V}\Delta t$$
$$\overrightarrow{OM}(t + \Delta t) = \overrightarrow{OM}(t) - \overrightarrow{V}\Delta t \qquad (16)$$

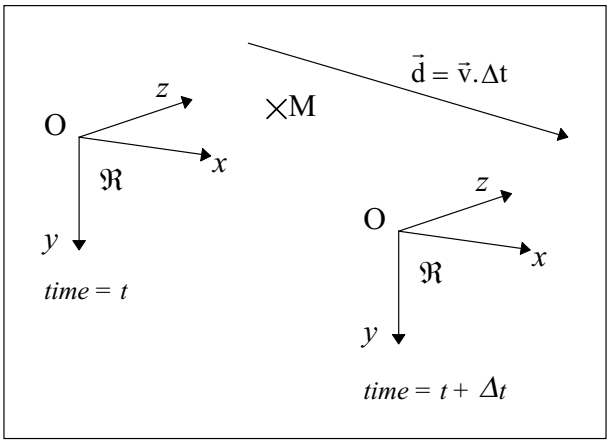

Figure 2: Translation trajectory

3.2 Rotation

Let us consider the case of the sensor rotation around any axis with the following parameters:

- $\vec{d}$: the oriented radius of the rotation.
- $\vec{r}$: the unitary vector defining the rotation axis.
- ω: the angular velocity around the axis.

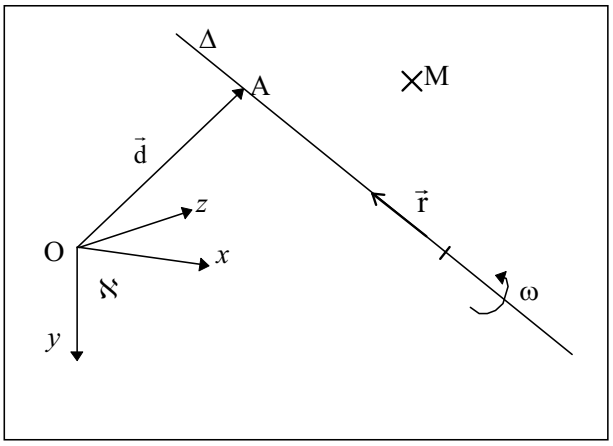

Figure 3: Rotation trajectory

In this case, we can write $\underline{V} = \omega\ \widetilde{r}\ \underline{d}$. If $\underline{d}$ is constant in the frame $\aleph$ (case of angular motion), then from 14, we get:

$$\underline{p}(t+\Delta t) = \underline{p}(t) + \Big(-\sin(\omega\Delta t)\ \widetilde{r} + (1-\cos(\omega\Delta t))\ \widetilde{r}^{\,2}\Big)\ (\underline{p}(t) - \underline{d})$$

Finally, we can write:

$$\overrightarrow{OM}(t+\Delta t) = \overrightarrow{OM}(t) + \Big(-\sin(\omega\Delta t)\ \widetilde{r} + (1-\cos(\omega\Delta t))\ \widetilde{r}^{\,2}\Big)\ \Big(\overrightarrow{OM}(t) - \underline{d}\Big) \qquad (17)$$

As in the case of translation, we can apply the perspective projection (see relation 15) to deduce the expression of X and Y in function of $\omega\Delta t$.

4 Results

In this part, we present the first results tested with our experimental robotic platform. This is a cartesian robot with 6 degrees of freedom (built by the firm AFMA Robot) and the parallel vision system WINDIS [16] [17]. This whole platform is controlled by a VME system, and can be programmed in C language under VxWorks real time operating system. A CCD camera is mounted on the end effector of the robot and is connected to the vision system WINDIS.

The target is composed of a cube (the length of the edge is 25cm) and 4 LED. Two LED locate each edge and we choose the object frame centered in the cube. In this frame, the 4 LED have the followings coordinates: $x = \pm 12.5cm, y = \pm 7.5cm, z = -12.5cm$. The application runs at video rate (40 ms), but the vision system introduces a dataflow latency of 80 ms. The vision process is based on the extraction of the center of gravity of the illuminated points in the image and also on sorting the features. At first, to illustrate the translation trajectory, we perform a sliding task along a cube side. In a second part, we experiment a rotational trajectory around a cube edge.

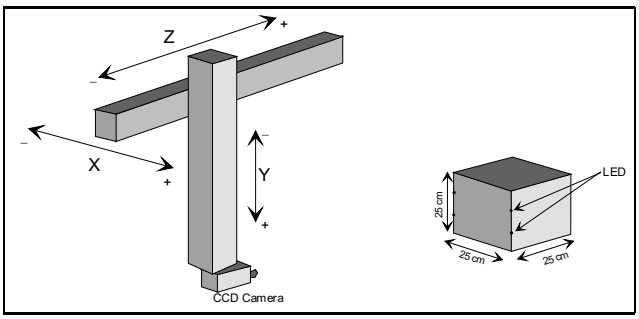

Figure 4: Overview of the scene

No specific calibrations have been done for all the experimentations and the estimation of the depth of the 4 LED used in visual servoing is not accurate.

4.1 Sliding along the cube side

In this experimentation, we perform a translation along a cube side using the sensor signal provided by the camera. The translation velocity V_x is fixed to: 4 $cm.s^{-1}$ and λ=0.1 . Using the relation (16), we have:

$$\begin{cases} x(t+\Delta t) = x(t) - V_x \Delta t \\ y(t+\Delta t) = y(t) - V_y \Delta t \\ z(t+\Delta t) = z(t) - V_z \Delta t \end{cases}$$

For a translation along the x axis the trajectory in image space is given by (14):

$$X^*(t) = \frac{x(t) - V_x \Delta t}{z(t)}.F_x$$

$$Y^*(t) = \frac{y(t)}{z(t)}.F_y$$

The initial position of the sensor frame in the object frame is (-0.174m, 0.044m, -0.077m) and the orientation one is (2^o, 4.2^o, -0.2^o).

In the following results, we compare many characteristics to illustrate the interest of this new control law.
In the figure 5, we compare the error E defines as $E = \sqrt{\sum_{i=1}^{i=4}(X_i - X_i^*)^2 + (Y_i - Y_i^*)^2}$. As we can see, the amplitude of the error with derivative term is lower and the convergence of our control law is faster.

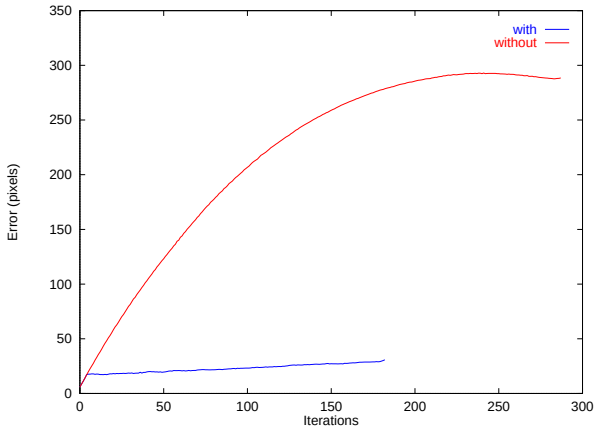

Figure 5: Comparison of error with and without derivative term

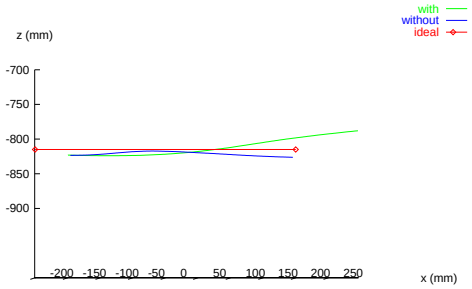

Figure 6: Top view of the trajectory

A top view of the corresponding trajectory is shown in figure 6. In both cases, we obtain a trajectory close to the desired trajectory. Some disturbances appear during the servoing task, due to the lack of accuracy in calibration.

In figure 7, we compare the same case with a oscillatory trajectory centered in 0. The amplitude of the signal is 0.15 m and λ=0.2. In this experimentation, the trajectory of the end effector starts at random position. The tracking of desired trajectory is well performed with the new control law. Figure 8 shows us the abscis behavior of the first point in image space, and we can note that the tracking is well done. In the other case, we have a static error during the servoing task. We have exactly the same results for all visual informations.

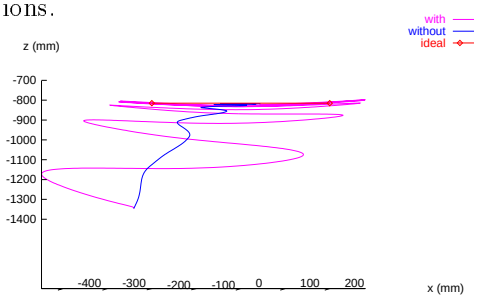

Figure 7: Comparison of an oscillatory trajectory with and without derivative term

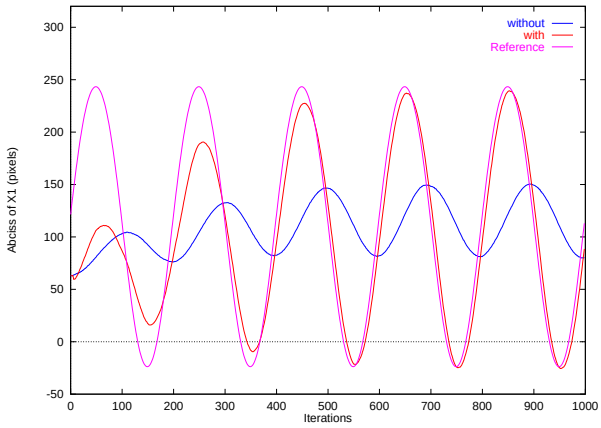

Figure 8: Comparison beetween the signal for X1 (abciss of the first point

4.2 Rotation around an cube edge

For this kind of task, we choose the rotation velocity $\omega = 0.04\ rad.s^{-1}$. We fix the distance between the sensor and the rotation axis to 0.8 meter. From the relation 14, we write the equations of the time varying reference feature. In this case, vector $\vec{r}$ defining the rotational axis is parallel to the y axis, so $\vec{r} = [0, r_y, 0]^T = [0, 1, 0]^T$ and we have:

$$\widetilde{r} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \widetilde{r}^2 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad \overrightarrow{d} = \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix}$$

From (14), we get:

$$\begin{pmatrix} x(t+\Delta t) \\ y(t+\Delta t) \\ z(t+\Delta t) \end{pmatrix} =$$

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} + \left[\begin{pmatrix} 0 & 0 & -\sin(\omega\Delta t) \\ 0 & 0 & 0 \\ \sin(\omega\Delta t) & 0 & 0 \end{pmatrix} + \begin{pmatrix} -1+\cos(\omega\Delta t) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1+\cos(\omega\Delta t) \end{pmatrix} \right] \cdot \begin{pmatrix} x(t) \\ y(t) \\ z(t)-d \end{pmatrix}$$

We can rewrite it like:

$$\begin{pmatrix} x(t+\Delta t) \\ y(t+\Delta t) \\ z(t+\Delta t) \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} + \begin{pmatrix} -1+\cos(\omega\Delta t) & 0 & -\sin(\omega\Delta t) \\ 0 & 0 & 0 \\ \sin(\omega\Delta t) & 0 & -1+\cos(\omega\Delta t) \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ y(t) \\ z(t)-d \end{pmatrix}$$

and develop like:

$$\begin{cases} x(t+\Delta t) = x(t)\cos(\omega\Delta t) - (z(t)-d)\sin(\omega\Delta t) \\ y(t+\Delta t) = y(t) \\ z(t+\Delta t) = x(t)\sin(\omega\Delta t) + (z(t)-d)\cos(\omega\Delta t) + d \end{cases}$$

Using relation 14, we obtain the expression of the time varying reference feature:

$$X^*(t) = \frac{x(t)\cos(\omega\Delta t) - (z(t)-d)\sin(\omega\Delta t)}{x(t)\sin(\omega\Delta t) + (z(t)-d)\cos(\omega\Delta t) + d}.F_x$$

$$Y^*(t) = \frac{y(t)}{x(t)\sin(\omega\Delta t) + (z(t)-d)\cos(\omega\Delta t) + d}.F_y$$

The initial position of the sensor frame in the object frame is (-0.1540m, 0.014m, -0.811m) and the orientation one is (-0.9^o, -3.8^o, 0^o). In figure 9, we compare the ideal trajectory with the trajectory obtained in real experimentation.

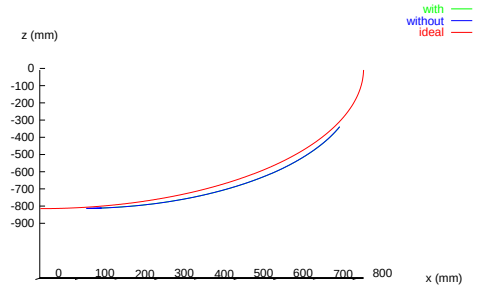

Figure 9: Top view of the trajectory

In both cases, the trajectory are similar. However, in the velocity graph (figure 10 and 11) we notice a faster behavior with the derivative term. In order to perform the rotational task, it is necessary to combine a translation along x axis and a rotation around y axis in the sensor frame. We can see this combination which occurred in figure 10 and 11.

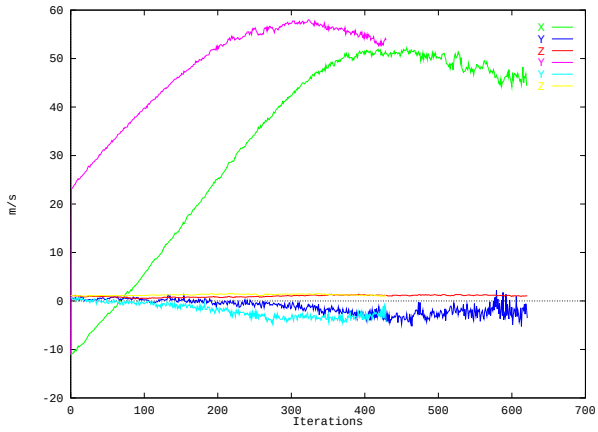

Figure 10: Translation velocity

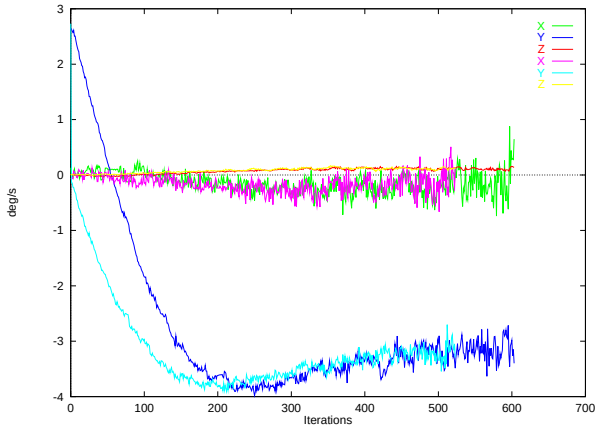

Figure 11: Rotation velocity

The new control law presented in this paper, does not allow to be used in all cases. Indeed, we have to respect the dynamic characteristics of the system in the sensor trajectory generation.

5 Conclusion

A new trajectory generation, based on visual servoing approach has been developed. The main goal of the present work is to show the validity of this new approach and to present a new aspect of trajectory generation. The first results obtained from experimentation, allows us to validate our approach on elementary trajectories. The generation of more complex trajectories, needs the management of the link between the elementary trajectories. That is why, we are now working on the problem of the linkage between two elementary trajectories and the estimation of a 3D target. The calibration of the camera and a better estimation of the depth should improve the accuracy of the trajectory.

References

[1] Espiau B., Chaumette F., and Rives P. A new approach to visual servoing in robotics. In *ICRA92*, volume 8, pages 313–326, Paris, june 1992. IEEE Trans. on Robotics and Automation.

[2] Samson C., Le borgne M., and Espiau B. *Robot control : the task function approach*. Oxford University Press, 1991.

[3] Khadraoui D., G. Motyl, P. Martinet, and J. Gallice. Visual servoing in robotics scheme using a camera/laser stripe sensor. *IEEE Trans. on Robotics and Automation*, 12(5):743– 749, October 1996.

[4] Marchand E., A. Rizzo, and F. Chaumette. Avoiding robot joint limits and kinematics singularities in visual servoing. volume A, pages 297– 301, Vienne, Autriche, August 1996. 13^th IARP/IEEE Int. Conf in Pattern Recognition, ICPR'96.

[5] Bensalah F. and F. Chaumette. Détection de rupture de modèle appliquée à l'asservissement visuel. Technical report, Rapport de Recherche INRIA, November 1994.

[6] Chaumette F. *La relation vision-commande : Théorie et application à des tâches robotiques*. PhD thesis, IRISA/INRIA Rennes, July 1990.

[7] Chaumette F. and A. Santos. Tracking a moving object by visual servoing. In *IFAC93*, volume 9, pages 409– 414. IFAC 12^th Triennal World Congress, july 1993.

[8] Motyl G., P. Martinet, and J. Gallice. Visual servoing with respect to a target sphere using a camera/laser stripe sensor. pages 591– 596. ICAR93, 1-2 November 1993.

[9] Wang J. and W.J. Wilson. 3d relative position and orientation estimation using kalman filter. IEEE Int. conf. on Robotics and Automation, May 1992.

[10] Latombe J.C. *Robot motion planning*. Kluwer Academic Publishers, USA, 1991.

[11] Feddema J.T and Mitchell O.R. Vision guided with feature-based trajectory generation. In *ICRA90*, volume 5, pages 691–700. IEEE Trans. on Robotics and Automation, october 1989.

[12] Zheng J.Y., Q. Chen, and S. Tsuji. Active camera guided manipulation. pages 632– 638. IEEE Int. Conf. on Robotics and Automation, April 1991.

[13] Papanikolopoulos N., P.K. Khosla, and T. Kanade. Vision and control techniques for robtic visual tracking. IEEE Int. Conf. on Robotics and Automation, April 1991.

[14] Papanikolopoulos N., P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14– 35, February 1993.

[15] Martinet P., F. Berry, and J. Gallice. Use of first derivative of geometric features in visual servoing. volume 4, pages 3413– 3419. IEEE Int. Conf on Robotics and Automation, April 1996.

[16] Martinet P., P. Rives, P. Fickinger, and J.J. Borelly. Parallel architecture for visual servoing applications. Workshop on Computer Architecture for Machine Perception, CAMP'91, December 1991.

[17] Rives P., J.J. Borelly, J. Gallice, and P. Martinet. Parallel architecture for visual servoing applications. Workshop on Computer Architecture for Machine Perception, CAMP'93, December 1993.

[18] Bajcsy R. Active perception. volume 76. Proceeding of the IEEE, August 1988.

[19] Hutchinson S., G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotic and Automation*, 12(5):651– 670, October 1996.

[20] Sundareswaran V., P. Bouthémy, and F. Chaumette. Active camera self-orientation using dynamic image parameters. ECCV, May 1994.

[21] W.J Wilson. Visual servo control of robots using kalman filter estimates of relative pose. IFAC 12^th Triennal World Congress, 1993.