



HAL
open science

Classification and realization of the different vision-based tasks

François Chaumette, Patrick Rives, Bernard Espiau

► **To cite this version:**

François Chaumette, Patrick Rives, Bernard Espiau. Classification and realization of the different vision-based tasks. *Visual Servoing*, 7, pp.199 - 228, 1993, World Scientific Series in Robotics and Intelligent Systems, 10.1142/9789814503709_0007. hal-01548352

HAL Id: hal-01548352

<https://inria.hal.science/hal-01548352>

Submitted on 27 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

“Visual Servoing - Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback”, World Scientific Series in Robotics and Automated Systems, Vol. 7, pp. 199-228, K. Hashimoto Editor, London, 1993.

CLASSIFICATION AND REALIZATION OF THE DIFFERENT VISION-BASED TASKS

François CHAUMETTE
*IRISA / INRIA Rennes, Campus de Beaulieu,
35042 Rennes-cedex, France*

Patrick RIVES
*INRIA Sophia Antipolis, 2004 Route des Lucioles,
06565 Valbonne, France*

and

Bernard ESPIAU
*INRIA Sophia Antipolis, 2004 Route des Lucioles,
06565 Valbonne, France*

ABSTRACT

This paper describes the different robotics tasks which can be realized using the visual servoing approach. A classification of these tasks, based on the virtual linkage formalism, is proposed and the different properties that the visual features and their related interaction matrix have to satisfy are presented. For the most classical virtual linkages between a camera and its environment, several combinations of visual features are selected. Finally, we recall the application of the task function approach to visual servoing by focusing on automatic control aspects.

1. Introduction

Several levels in the problem of task planning may usually be distinguished. In particular, at the high level, problems of path planning and off line programming have often been addressed. Sophisticated algorithms using Computational Geometry results allow us to compute efficiently free trajectories in the configuration space taking into account geometric and sometimes kinematic constraints [14], [2]. Unfortunately, in most cases, it is not obvious to deduce from the paths computed by these methods an efficient control scheme which would perform the task in an accurate way. This is due to the fact that, because of disturbances coming from the imperfect modeling of the robot and its environment and of measurement errors, further robustness properties

are really required. For completing correctly the task, the path provided by the high level should therefore be transformed in a specification at the control level. For doing that, we express a task at a low-level by a set of constraints on the motion of a frame linked to the robot with respect to a reference frame associated with the environment.

In this framework, this paper focuses on the utilization of the so-called visual servoing approach as an adequate way to perform non-contact sensor-based tasks in robotics. By *visual servoing*, we mean the use of vision sensors embedded in a closed-loop control scheme and providing with some measurements of the interaction between the robot and its local environment at a sufficiently high sampling rate for ensuring the stability of the robot's control loop. Non-contact sensing is useful in the achievement of many kinds of robotics tasks in various application domains. For example, relative positioning errors due to the inaccuracies in the modeling of robot kinematics, or elasticities in the case of large light structure for space application, can thus be compensated. Another important problem, which arises in the case of mobile robots, is on-line obstacle avoidance when the environment is not well known, imperfectly modeled or liable to unexpected changes. Sensor-based control is again an elegant way of addressing the problem.

In the following, we therefore propose a framework allowing to design and perform efficiently *Vision-based Tasks*. In order to be easily tractable by the user, the modeling aspect is based on the well-known theory of mechanisms. Besides, efficiency and robustness will be considered at the control level by using the task function approach introduced by Samson [20].

The paper is organized as follows. In section 2, we recall the concept of interaction, which links a mobile camera to its environment. Explicit results are given for the most usual geometrical primitives of a scene (points, segments, lines, circles, spheres and cylinders). In section 3, we present the formalism of *virtual linkage*, which allows us to classify the different vision-based tasks. For the most classical virtual linkages (rigid, revolute, prismatic, etc.), several combinations of visual features are obtained by using the properties of their interaction matrices. Robustness issues of the vision-based tasks are also evoked. Section 4 is devoted to control aspects and to the application of the task function approach to visual servoing. Finally, section 5 gives simulation results of the realization of several vision-based tasks by a six-jointed robot arm and a mobile robot with non-holonomic constraints.

2. Modeling Interactions with the Environment

2.1 General Framework

From a general point of view, let us consider a sensor (S), rigidly fixed to a robot, with associated frame (F_S). This sensor perceives a part of the environment which contains a target (T) with associated frame (F_T). Let the one-dimensional signal s_j be the useful part of the signal provided by (S). We shall further assume in the following

that given (S) and (T) , the signal s_j is only function of the relative location (position and attitude) of (T) with respect to (S) . In practice, this assumption is reasonable for a large class of sensors providing with geometric information, like range finders or force sensors. Let us remark that, in the peculiar case of vision sensors, the use of data based on geometric features like points, lines,..., is authorized, while data based on photometric properties, which may depend on the lightning conditions, cannot be considered. The relative location of (F_T) with respect to (F_S) can be parametrized by an element \bar{r} of the displacement space SE_3 . Moreover, let us consider that the location of (F_S) with respect to a fixed frame (F_0) is a function of the robot joint coordinates \underline{q} and let us use the variable t to parametrize the location of the target (when its motion is independent), we may thus write:

$$s_j(\bar{r}(q, t)) = s_j(F_S, F_T) \quad (1)$$

where s_j is a C^2 function with value in \mathbb{R} and domain in SE_3 . The signal s_j admits a derivative represented by an element of se_3 and we can write the differential:

$$\dot{s}_j = \frac{\partial s_j}{\partial \bar{r}} \bullet \frac{d\bar{r}}{dt} = H_{s_j} \bullet T_{ST} \quad (2)$$

where:

- $T_{ST} = (V, \omega)$ is the velocity screw of the frame F_T with respect to the frame F_S , where V and ω represent its translational and rotational components respectively;
- \bullet denotes the screw product;
- $H_{s_j} = \frac{\partial s_j}{\partial \bar{r}}$ is a screw which characterizes the variations of the signal s_j .

With an obvious breach of notation, Eq. 2 can also be written under matrix form:

$$\dot{s}_j = L_{s_j}^T (T_S - T_T) \quad (3)$$

where T_S and T_T are the sensor and target velocity screw respectively, expressed in the sensor frame F_S .

As an important remark, we can note that for a given situation (S, T) , H_{s_j} and $L_{s_j}^T$ only depend on the characteristics of the sensor itself and of the target. It should be emphasized that it is *independent* of the robot itself. Thereby, H_{s_j} and $L_{s_j}^T$ are only representative of the *interaction* between the sensor and the robot's environment. For this reason, they are called *interaction screw* and *interaction matrix* respectively, and contain most of the information required to design sensor-based control schemes.

This general framework is valid for a large class of sensors as long as they satisfy the assumption mentioned previously. This is in particular the case of visual servoing, as shown now.

2.2 Case of an Image Sensor

Let us model a camera by a perspective projection (see figure 1). In the following, all the used variables (point coordinates, screws,...) are expressed in the camera frame $F_S(O, \vec{x}, \vec{y}, \vec{z})$.

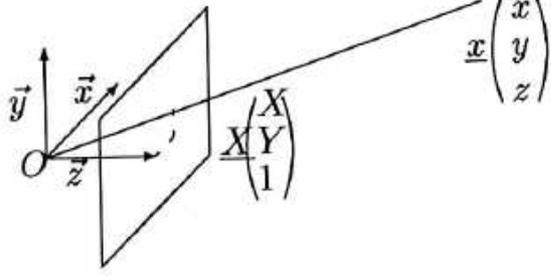


Figure 1: A simple camera model

Without loss of generality, the focal length is assumed to be equal to 1, so that any point with coordinates $\underline{x} = (x y z)^T$ is projected on the image plane as a point with coordinates $\underline{X} = (X Y 1)^T$ with:

$$\underline{X} = \frac{1}{z} \underline{x} \quad (4)$$

In [22] and [10], an experimental learning approach is proposed to compute the interaction matrix related to points. But it is also possible to derive it in an explicit way: see for example [7]. More generally, in [6], a systematic method for computing the interaction matrix of any set of visual features defined upon geometrical primitives is proposed for this camera model. As it will be shown in the next section, the choice of the visual features able to realize any vision-based task is essentially based on these interaction matrices. We now present the obtained results for the parameters describing the projection in the image of the most usual primitives (points, segments, lines, circles, spheres and cylinders). Let us note that all the computations leading to the following expressions are detailed in [3].

2.2.1. Points

Consider a point m with coordinates $\underline{x} = (x y z)^T$ in the scene and $\underline{X} = (X Y)^T$ in the image. The interaction matrices L_X^T and L_Y^T related to X and Y correspond to the well-known ‘optical flow’ equations and are written:

$$\begin{aligned} L_X^T &= \begin{bmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{bmatrix} \\ L_Y^T &= \begin{bmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{bmatrix} \end{aligned} \quad (5)$$

2.2.2. Segments

Let us consider a segment, ended by the points m_1 and m_2 . It may be represented either by the coordinates of its end points, M_1 and M_2 , (we then have $L_{X_1}^T, L_{Y_1}^T, L_{X_2}^T, L_{Y_2}^T$ given by Eq. 5), or by the following parameters: length l , orientation α and coordinates X_c, Y_c of its center. In that last case, the related interaction matrices may be derived as:

$$\begin{aligned}
L_l^T &= \begin{bmatrix} \lambda_1 \cos \alpha & \lambda_1 \sin \alpha \\ \lambda_2 l - \lambda_1 (X_c \cos \alpha + Y_c \sin \alpha) & l [X_c \cos \alpha \sin \alpha + Y_c (1 + \sin^2 \alpha)] \\ -l [X_c (1 + \cos^2 \alpha) + Y_c \cos \alpha \sin \alpha] & 0 \end{bmatrix} \\
L_\alpha^T &= \begin{bmatrix} -\lambda_1 \sin \alpha / l & \lambda_1 \cos \alpha / l \\ \lambda_1 (X_c \sin \alpha - Y_c \cos \alpha) / l & -X_c \sin^2 \alpha + Y_c \cos \alpha \sin \alpha \\ X_c \cos \alpha \sin \alpha - Y_c \cos^2 \alpha & -1 \end{bmatrix} \\
L_{X_c}^T &= \begin{bmatrix} -\lambda_2 & 0 \\ \lambda_2 X_c - \lambda_1 l \cos \alpha / 4 & X_c Y_c + l^2 \cos \alpha \sin \alpha / 4 \\ -(1 + X_c^2 + l^2 \cos^2 \alpha / 4) & Y_c \end{bmatrix} \\
L_{Y_c}^T &= \begin{bmatrix} 0 & -\lambda_2 \\ \lambda_2 Y_c - \lambda_1 l \sin \alpha / 4 & 1 + Y_c^2 + l^2 \sin^2 \alpha / 4 \\ -X_c Y_c - l^2 \cos \alpha \sin \alpha / 4 & -X_c \end{bmatrix} \quad (6)
\end{aligned}$$

with $\lambda_1 = (z_1 - z_2) / z_1 z_2$ and $\lambda_2 = (z_1 + z_2) / 2 z_1 z_2$.

2.2.3. Straight lines

A straight line is here represented as the intersection of two planes under the following form:

$$h(\underline{x}, \underline{p}) = \begin{cases} a_1 x + b_1 y + c_1 z = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \quad (7)$$

with $d_2 \neq 0$ in order to exclude degenerated cases. The equation of the projected line in the image plane is written:

$$g(\underline{X}, \underline{P}) = X \cos \theta + Y \sin \theta - \rho = 0 \text{ with } \begin{cases} \cos \theta = a_1 / \sqrt{a_1^2 + b_1^2} \\ \sin \theta = b_1 / \sqrt{a_1^2 + b_1^2} \\ \rho = -c_1 / \sqrt{a_1^2 + b_1^2} \end{cases} \quad (8)$$

and we can obtain:

$$\begin{aligned}
L_\theta^T &= \begin{bmatrix} \lambda_\theta \cos \theta & \lambda_\theta \sin \theta & -\lambda_\theta \rho & -\rho \cos \theta & -\rho \sin \theta & -1 \end{bmatrix} \\
L_\rho^T &= \begin{bmatrix} \lambda_\rho \cos \theta & \lambda_\rho \sin \theta & -\lambda_\rho \rho & (1 + \rho^2) \sin \theta & -(1 + \rho^2) \cos \theta & 0 \end{bmatrix} \quad (9)
\end{aligned}$$

with $\lambda_\theta = (a_2 \sin \theta - b_2 \cos \theta)/d_2$ and $\lambda_\rho = (a_2 \rho \cos \theta + b_2 \rho \sin \theta + c_2)/d_2$.

2.2.4. Circles

A circle may be represented as the intersection of a sphere and a plane:

$$h(\underline{x}, \underline{p}) = \begin{cases} (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \\ \alpha(x - x_0) + \beta(y - y_0) + \gamma(z - z_0) = 0 \end{cases} \quad (10)$$

Its projection on the image plane takes the form of an ellipse or a circle (except in degenerated cases where the projection is a segment), which may be represented as:

$$g(\underline{X}, \underline{P}) = \frac{(X - X_c + E(Y - Y_c))^2}{A^2(1 + E^2)} + \frac{(Y - Y_c - E(X - X_c))^2}{B^2(1 + E^2)} - 1 = 0 \quad (11)$$

$$\text{with } \begin{cases} X_c = (K_1 K_3 - K_2 K_4)/(K_2^2 - K_0 K_1) \\ Y_c = (K_0 K_4 - K_2 K_3)/(K_2^2 - K_0 K_1) \\ E = (K_1 - K_0 \pm \sqrt{(K_1 - K_0)^2 + 4K_2^2})/2K_2 \\ A^2 = 2(K_0 X_c^2 + 2K_2 X_c Y_c + K_1 Y_c^2 - K_5)/(K_0 + K_1 \pm \sqrt{(K_1 - K_0)^2 + 4K_2^2}) \\ B^2 = 2(K_0 X_c^2 + 2K_2 X_c Y_c + K_1 Y_c^2 - K_5)/(K_0 + K_1 \mp \sqrt{(K_1 - K_0)^2 + 4K_2^2}) \end{cases}$$

where the coefficients K_i 's are obtained from the polynomial equation of an ellipse:

$$g(\underline{X}, \underline{P}) = K_0 X^2 + K_1 Y^2 + 2K_2 XY + 2K_3 X + 2K_4 Y + K_5 = 0 \quad (12)$$

More precisely, we have:

$$\begin{cases} K_0 = a^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2ax_0 \\ K_1 = b^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2by_0 \\ K_2 = ab(x_0^2 + y_0^2 + z_0^2 - r^2) - bx_0 - ay_0 \\ K_3 = ac(x_0^2 + y_0^2 + z_0^2 - r^2) - cx_0 - az_0 \\ K_4 = bc(x_0^2 + y_0^2 + z_0^2 - r^2) - cy_0 - bz_0 \\ K_5 = c^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2cz_0 \end{cases} \quad (13)$$

with $a = \alpha/(\alpha x_0 + \beta y_0 + \gamma z_0)$, $b = \beta/(\alpha x_0 + \beta y_0 + \gamma z_0)$ and $c = \gamma/(\alpha x_0 + \beta y_0 + \gamma z_0)$.

The natural parametrization (X_c, Y_c, E, A, B) is always ambiguous. Furthermore, the orientation E is not defined when the ellipse is a circle ($A = B$). So we use a representation based on the normalized inertial moments $(X_c, Y_c, \mu_{20}, \mu_{11}, \mu_{02})$ with:

$$\begin{cases} \mu_{20} = (A^2 + B^2 E^2)/(1 + E^2) \\ \mu_{11} = E(A^2 - B^2)/(1 + E^2) \\ \mu_{02} = (A^2 E^2 + B^2)/(1 + E^2) \end{cases} \quad (14)$$

and we can obtain:

$$\begin{aligned}
L_{X_c}^T &= \begin{bmatrix} -1/z_c & 0 & X_c/z_c + a\mu_{20} + b\mu_{11} \\ X_c Y_c + \mu_{11} & -1 - X_c^2 - \mu_{20} & Y_c \end{bmatrix} \\
L_{Y_c}^T &= \begin{bmatrix} 0 & -1/z_c & Y_c/z_c + a\mu_{11} + b\mu_{02} \\ 1 + Y_c^2 + \mu_{02} & -X_c Y_c - \mu_{11} & -X_c \end{bmatrix} \\
L_{\mu_{20}}^T &= \begin{bmatrix} -2(a\mu_{20} + b\mu_{11}) & 0 & 2[(1/z_c + aX_c)\mu_{20} + bX_c\mu_{11}] \\ 2(Y_c\mu_{20} + X_c\mu_{11}) & -4\mu_{20}X_c & 2\mu_{11} \end{bmatrix} \\
L_{\mu_{11}}^T &= \begin{bmatrix} -a\mu_{11} - b\mu_{02} & -a\mu_{20} - b\mu_{11} & aY_c\mu_{20} + (3/z_c - c)\mu_{11} + bX_c\mu_{02} \\ 3Y_c\mu_{11} + X_c\mu_{02} & -Y_c\mu_{20} - 3X_c\mu_{11} & \mu_{02} - \mu_{20} \end{bmatrix} \\
L_{\mu_{02}}^T &= \begin{bmatrix} 0 & -2(a\mu_{11} + b\mu_{02}) & 2[(1/z_c + bY_c)\mu_{02} + aY_c\mu_{11}] \\ 4Y_c\mu_{02} & -2(Y_c\mu_{11} + X_c\mu_{02}) & -2\mu_{11} \end{bmatrix} \tag{15}
\end{aligned}$$

with $z_c = 1/(aX_c + bY_c + c)$.

2.2.5. Spheres

A sphere is represented by:

$$h(\underline{x}, \underline{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0 \tag{16}$$

The image of a sphere is an ellipse (a circle when $x_0 = y_0 = 0$) with equation:

$$g(\underline{X}, \underline{P}) = K_0 X^2 + K_1 Y^2 + 2K_2 XY + 2K_3 X + 2K_4 Y + K_5 = 0 \tag{17}$$

$$\text{with } \begin{cases} K_0 = r^2 - y_0^2 - z_0^2 \\ K_1 = r^2 - x_0^2 - z_0^2 \\ K_2 = x_0 y_0 \\ K_3 = x_0 z_0 \\ K_4 = y_0 z_0 \\ K_5 = r^2 - x_0^2 - y_0^2 \end{cases}$$

The interaction matrix related to an ellipse has already been given. Results are therefore given by Eq. 15 with the particular following values for a , b and c :

$$\begin{cases} a = x_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \\ b = y_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \\ c = z_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \end{cases} \tag{18}$$

that describe the plane in which belong the limb points of the sphere through:

$$1/z = aX + bY + c \tag{19}$$

2.2.6. Cylinders

A cylinder may be represented by the equation:

$$h(\underline{x}, \underline{p}) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - (ax + by + cz)^2 - r^2 = 0 \quad (20)$$

$$\text{with } \begin{cases} a^2 + b^2 + c^2 = 1 \\ ax_0 + by_0 + cz_0 = 0 \end{cases}$$

where r is the radius of the cylinder, a, b and c are the coordinates of its direction vector and x_0, y_0 and z_0 are the coordinates of the nearest point belonging to the cylinder axis from the projection center.

The projection of a cylinder on the image plane is (for non-degenerated cases) a set of two straight lines with equation:

$$\begin{cases} D_1 : X \cos \theta_1 + Y \sin \theta_1 - \rho_1 = 0 \\ D_2 : X \cos \theta_2 + Y \sin \theta_2 - \rho_2 = 0 \end{cases} \quad (21)$$

with:

$$\begin{cases} \cos \theta_1 = \frac{rx_0/A - \alpha}{\sqrt{(rx_0/A - \alpha)^2 + (ry_0/A - \beta)^2}}, \cos \theta_2 = \frac{rx_0/A + \alpha}{\sqrt{(rx_0/A + \alpha)^2 + (ry_0/A + \beta)^2}} \\ \sin \theta_1 = \frac{ry_0/A - \beta}{\sqrt{(rx_0/A - \alpha)^2 + (ry_0/A - \beta)^2}}, \sin \theta_2 = \frac{ry_0/A + \beta}{\sqrt{(rx_0/A + \alpha)^2 + (ry_0/A + \beta)^2}} \\ \rho_1 = \frac{rz_0/A - \gamma}{\sqrt{(rx_0/A - \alpha)^2 + (ry_0/A - \beta)^2}}, \rho_2 = \frac{rz_0/A + \gamma}{\sqrt{(rx_0/A + \alpha)^2 + (ry_0/A + \beta)^2}} \end{cases}$$

$$\text{where } \begin{cases} A = \sqrt{x_0^2 + y_0^2 + z_0^2 - r^2} \\ \alpha = cy_0 - bz_0 \\ \beta = az_0 - cx_0 \\ \gamma = bx_0 - ay_0 \end{cases}$$

and we finally obtain:

$$\begin{aligned} L_{\theta_1}^T &= [\lambda_{\theta_1} \cos \theta_1 \quad \lambda_{\theta_1} \sin \theta_1 \quad -\lambda_{\theta_1} \rho_1 \quad -\rho_1 \cos \theta_1 \quad -\rho_1 \sin \theta_1 \quad -1] \\ L_{\rho_1}^T &= [\lambda_{\rho_1} \cos \theta_1 \quad \lambda_{\rho_1} \sin \theta_1 \quad -\lambda_{\rho_1} \rho_1 \quad (1 + \rho_1^2) \sin \theta_1 \quad -(1 + \rho_1^2) \cos \theta_1 \quad 0] \\ L_{\theta_2}^T &= [\lambda_{\theta_2} \cos \theta_2 \quad \lambda_{\theta_2} \sin \theta_2 \quad -\lambda_{\theta_2} \rho_2 \quad -\rho_2 \cos \theta_2 \quad -\rho_2 \sin \theta_2 \quad -1] \\ L_{\rho_2}^T &= [\lambda_{\rho_2} \cos \theta_2 \quad \lambda_{\rho_2} \sin \theta_2 \quad -\lambda_{\rho_2} \rho_2 \quad (1 + \rho_2^2) \sin \theta_2 \quad -(1 + \rho_2^2) \cos \theta_2 \quad 0] \end{aligned} \quad (22)$$

$$\text{with } \begin{cases} \lambda_{\theta_1} = (y_0 \cos \theta_1 - x_0 \sin \theta_1)/A^2 \\ \lambda_{\rho_1} = -(x_0 \rho_1 \cos \theta_1 + y_0 \rho_1 \sin \theta_1 + z_0)/A^2 \\ \lambda_{\theta_2} = (y_0 \cos \theta_2 - x_0 \sin \theta_2)/A^2 \\ \lambda_{\rho_2} = -(x_0 \rho_2 \cos \theta_2 + y_0 \rho_2 \sin \theta_2 + z_0)/A^2 \end{cases}$$

3. Vision-based Tasks

In the previous section, we shown how sensor signals could be modeled. We now study how they can be used in the design of vision-based tasks at a task planning level. Indeed, it seems useful to provide the user with systematic methods allowing to specify vision-based tasks with a natural formalism. The one we chose rely on the well-known theory of mechanisms. More precisely, we shall show that the properties of the interaction screw can be analyzed in terms of constraints in the configuration space, and represented by a formalism very similar to the one which is currently used for kinematics of contacts between solids.

3.1 The Notion of Virtual Linkage

Eq. 2 computes the differential of the sensor signal s_j as a screw product. It may occur that, for a peculiar value s_{jd} of the signal associated to a given location \bar{r} , certain displacements T_{ST}^* of the sensor with respect to the target do not result in any change of the value s_{jd} . For example, when the camera motion is parallel to the projection line joining a point in the scene to the optical center of the camera, then the signals $s_{1d} = X$ and $s_{2d} = Y$ where (X, Y) are the coordinates in the image of the point, remain unchanged.

Given the interaction screw H_{s_j} , the velocity screws T_{ST}^* that leave s_j invariant are solutions of:

$$H_{s_j} \bullet T_{ST}^* = 0 \quad (23)$$

The definition of the screw product implies that T_{ST}^* is a screw *reciprocal* to H_{s_j} in the six-dimensional vector space of screws. Let us now consider a more general case, where we can extract from the image more than one visual signal, and let us denote as \underline{s} the vector regrouping all the visual signals $\underline{s} = (s_1, \dots, s_p)^T$. Thereby, the motions around the location \bar{r} which leave \underline{s} unchanged are characterized by the reciprocal subspace (T_{ST}^*) of the subspace spanned by the set of screws $(H_{s_1}, \dots, H_{s_p})$. Using the interaction matrix $L_{\underline{s}}^T = (L_{s_1}, \dots, L_{s_p})^T$, the subspace (T_{ST}^*) is characterized through:

$$(T_{ST}^*) = \text{Ker } L_{\underline{s}}^T \quad (24)$$

As well known, the formalism of screws is also a straightforward manner to describe contacts between objects. This suggests to make an analogy between true contacts and sensor-based tasks. The underlying idea is the following: imposing $\dot{\underline{s}} = 0$ is equivalent to introducing constraints in the configuration space SE_3 in order to achieve a *virtual contact* between the target and the sensor. Going on with this analogy, we can now introduce the notion of *virtual linkage*:

A set of k compatible constraints: $\underline{s}(\bar{r}) - \underline{s}_d = 0$ determines a virtual linkage between the sensor (S) and the target (T). At a location \bar{r} where the constraints are satisfied, the dimension N of the reciprocal subspace (T_{ST}^) in se_3 is called the **class** of the virtual linkage at \bar{r} .*

In this definition, the ‘class’ of the virtual linkage is the number of degrees of freedom of the sensor corresponding to the motions which leave \underline{s} unchanged. The concept of virtual linkage, which may include the physical linkage when contact sensors are used, will allow us to design the wished sensor-referenced robotics tasks in a simple way. This also establishes a connection with the approach known as ‘hybrid control’, which is traditionally used in control schemes involving contact force sensors.

In a more practical way, the design of a sensor-based task (here a vision-based task) should be done by the user through the following steps:

1. expressing the robot’s task in terms of virtual linkage;
2. from the a priori knowledge on the application, selecting the visual signals which realize the virtual linkage;
3. building from these signals the task function which ensures a good realization of the task (see section 4);
4. building the control scheme which allows to regulate the task function (see section 4).

Note that, in order to make easier the achievement of first and second steps, it may be useful to constitute a ‘library of canonical vision-based tasks’. This library will contain the most classical linkages which are used in practice and some examples of signals \underline{s} able to realize them.

3.2 Classification of the vision-based tasks

3.2.1. Rigid Linkage

The class number of this linkage is 0. It corresponds to a positioning task such that there locally exists a unique equilibrium location between the camera and the environment. In fact, we should have $(T_{ST}^*) = (0 \ 0 \ 0 \ 0 \ 0 \ 0)^T$, which means that no camera motion is allowed at the equilibrium location $\underline{s}(\bar{r}) = \underline{s}_d$. We therefore have to choose at least six visual features s_j such that the related interaction matrix be of rank 6. This offers wide possibilities (the coordinates of three points or the parameters describing three lines for example), but care should be taken of possible ambiguities and isolated singularities (see section 3.3).

3.2.2. Prismatic Linkage

In order to realize such a linkage, the subspace (T_{ST}^*) corresponding to the selected visual features should have the form:

$$(T_{ST}^*) = (a \ b \ c \ 0 \ 0 \ 0)^T \tag{25}$$

where the direction given by the vector $(a, b, c)^T$ is the one which is not constrained by the linkage. To realize this class-1 virtual contact, different solutions exist. For example, the use of (ρ_i, θ_i) parameters associated with the projection of at least three lines of direction $(a, b, c)^T$ easily solves the problem. Other features are proposed in [3].

3.2.3. Revolute Linkage

We only consider here the particular case of a linkage allowing the camera to rotate around its optical axis. We have therefore to find visual features such that the interaction matrix kernel be spanned by:

$$(T_{ST}^*) = (0 \ 0 \ 0 \ 0 \ 0 \ 1)^T \quad (26)$$

A first idea is to consider a centered circle parallel to the image plane. Unfortunately, the equilibrium position is then singular. This may be seen in representing ellipses by their moments, which leads to the following interaction matrix (see Eq. 15):

$$L_{s_d}^T = \begin{pmatrix} -1/z_d & 0 & 0 & 0 & -1 - R^2 & 0 \\ 0 & -1/z_d & 0 & 1 + R^2 & 0 & 0 \\ 0 & 0 & 2R^2/z_d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2R^2/z_d & 0 & 0 & 0 \end{pmatrix} \quad (27)$$

where $z - z_d = 0$ is the equation of the circle plane and R is the radius of the projected circle. In that case, we have:

$$(T_{ST}^*) = \begin{pmatrix} 0 & 1 + R^2 & 0 \\ 0 & 0 & -1 - R^2 \\ 0 & 0 & 0 \\ 0 & -1/z_d & 0 \\ 0 & 0 & -1/z_d \\ 1 & 0 & 0 \end{pmatrix} \quad (28)$$

We indeed find in (T_{ST}^*) the expected motion, but for exactly representing the desired linkage, supplementary visual features should be taken into account. We may for example choose (see figure 2):

- visual signals associated with a second centered circle, also parallel to the image plane (either in the same plane with different radius or in a different plane with the same radius),
- coordinates of the projection in the image of the point $(0, 0, z_d)^T$.

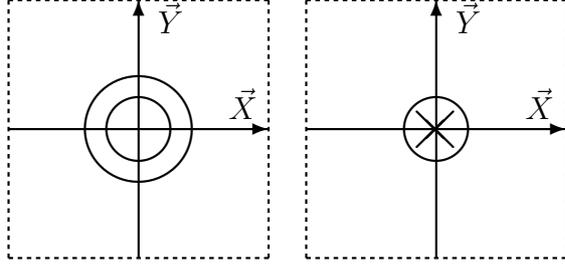


Figure 2: Revolute linkage using circles

3.2.4. Prismatic/Revolute Linkage

This class-2 linkage is characterized by:

$$(I_{ST}^*) = \begin{pmatrix} a & 0 \\ b & 0 \\ c & 0 \\ 0 & a \\ 0 & b \\ 0 & c \end{pmatrix} \quad (29)$$

where the vector $(a, b, c)^T$ is both the rotation axis and the translation direction of unconstrained camera motions. To realize it, let us firstly consider a cylinder with radius r , the generating line of which is parallel to the image plane and meets the point $(0, 0, z_d)^T$. Its equation is (cf. section 2.3.5):

$$h(\underline{x}, \underline{p}) = (bx - ay)^2 + (z - z_d)^2 - r^2 = 0 \quad (30)$$

where the vector $(a, b, 0)^T$ with $a^2 + b^2 = 1$ is the direction of the generating line. The cylinder is projected on the image plane as two parallel straight lines which are symmetric with respect to the center of the image (see figure 3). Their equations are:

$$\begin{cases} D_1 : & bX - aY + r/\sqrt{z_d^2 - r^2} = 0 \\ D_2 : & -bX + aY + r/\sqrt{z_d^2 - r^2} = 0 \end{cases} \quad (31)$$

By using the ρ, θ line representation, we get:

$$\begin{cases} \cos \theta_1 = -\cos \theta_2 = b \\ \sin \theta_1 = -\sin \theta_2 = -a \\ \rho_1 = \rho_2 = r/\sqrt{z_d^2 - r^2} \end{cases} \quad (32)$$

from which we may derive, by using Eq. 22, the expressions of associated interaction

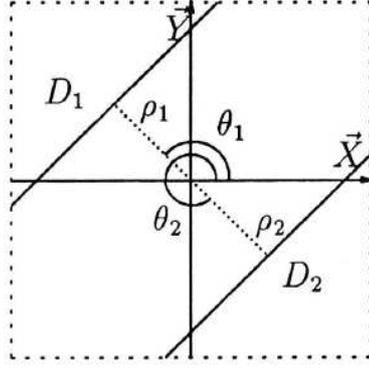


Figure 3: Prismatic/revolute linkage using a cylinder

matrices:

$$\begin{aligned}
 L_{\rho_1}^T &= \begin{bmatrix} \lambda_\rho b & -\lambda_\rho a & -\lambda_\rho \rho & -a(1+\rho^2) & -b(1+\rho^2) & 0 \end{bmatrix} \\
 L_{\theta_1}^T &= \begin{bmatrix} 0 & 0 & 0 & -b\rho & a\rho & -1 \end{bmatrix} \\
 L_{\rho_2}^T &= \begin{bmatrix} -\lambda_\rho b & \lambda_\rho a & -\lambda_\rho \rho & a(1+\rho^2) & b(1+\rho^2) & 0 \end{bmatrix} \\
 L_{\theta_2}^T &= \begin{bmatrix} 0 & 0 & 0 & b\rho & -a\rho & -1 \end{bmatrix}
 \end{aligned} \tag{33}$$

with $\lambda_\rho = -z_d/(z_d^2 - r^2)$.

We may easily compute the space of invariant screws, (T_{ST}^*) :

$$(T_{ST}^*) = \begin{pmatrix} a & b(1+\rho^2) \\ b & -a(1+\rho^2) \\ 0 & 0 \\ 0 & \lambda_\rho a \\ 0 & \lambda_\rho b \\ 0 & 0 \end{pmatrix} \tag{34}$$

Since, for any screw $T = (V, \omega)$ and any point m , we have $\omega(m) = \omega(O)$ and $V(m) = V(O) + \omega \times Om$, the set of screws (T_{ST}^*) may be expressed in a point of the linkage axis like $(0, 0, z_d)^T$. We then have:

$$(T_{ST}^*) = \begin{pmatrix} a & 0 \\ b & 0 \\ 0 & 0 \\ 0 & a \\ 0 & b \\ 0 & 0 \end{pmatrix} \tag{35}$$

Therefore, these visual features allows us to realize the desired linkage the axis of which is parallel to the image plane ($c = 0$). If the desired axis is \vec{z} , the linkage may

be derived from the case of the revolute one by choosing as parameters the coordinates $(X_{c_i}, Y_{c_i})^T$ of the considered circle centers.

3.2.5. Plane/plane contact

In order to realize this class-3 linkage, we have to find visual features such that the space of invariant screws be spanned by:

$$(T_{ST}^*) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (36)$$

Since the plane/plane contact only allows motions parallel to the image plane, the scene may be made for example of any regular (periodic) polygonal planar mesh such that there always exists in the camera field the required set of adequate image features. In order to preserve isotropy, we also impose that all mesh edges have same length (see figure 4).

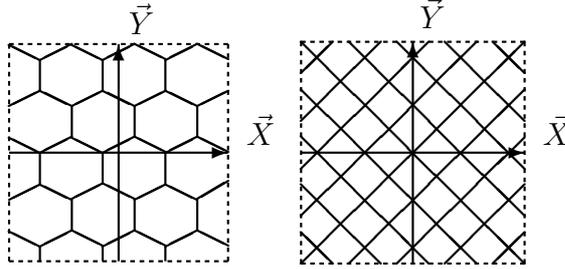


Figure 4: Plane/plane contact using mesh

We may then choose as visual signals the length of *at least* three projected mesh edges which do not belong to the same line. In practice we choose a connected set of edges as close as possible to the image center. We may verify that, for any edge represented by (X_c, Y_c, l, α) , we have (see Eq. 6):

$$L_l^T = \begin{bmatrix} 0 & 0 \\ l/z_d & l[X_c \cos \alpha \sin \alpha + Y_c(1 + \sin^2 \alpha)] \\ -l[X_c(1 + \cos^2 \alpha) + Y_c \cos \alpha \sin \alpha] & 0 \end{bmatrix} \quad (37)$$

By choosing at least three edges, (T_{ST}^*) has the required form. Let us emphasize that, in order to improve the resulting motion, the use of a larger set of edges, together with a process handling in a smooth way edge occurrence and removal, should be preferred.

3.2.6. Ball-and-socket linkage

Let us end this survey of the most classical linkages with this class-3 contact. The related feature is the set of the parameters representing the projection in the image of a sphere with center coordinates $(0, 0, z_d)^T$. This projection is a centered circle, and the interaction matrix associated with the representation X_c , Y_c , and $\mu = (\mu_{20} + \mu_{02})/2$ of the circle is:

$$\begin{aligned} L_{X_c}^T &= \begin{bmatrix} -1/z_d & 0 & 0 & 0 & -1 - \mu & 0 & 0 \end{bmatrix} \\ L_{Y_c}^T &= \begin{bmatrix} 0 & -1/z_d & 0 & 1 + \mu & 0 & 0 & 0 \end{bmatrix} \\ L_{\mu}^T &= \begin{bmatrix} 0 & 0 & 2\mu/z_d & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (38)$$

Expressed in the point of coordinates $(0, 0, z_d)^T$, we have:

$$(T_{ST}^*) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (39)$$

and we see that all rotations are free, which is the definition of the ball-and-socket linkage.

3.2.7. Other linkages

In this section, we have only described linkages corresponding to a few classical mechanical ones. The study may indeed be generalized according to the applications. In fact, specifically visual tasks are realizable as soon as it is possible to define a feature \underline{s}_d such that $\underline{s}(\bar{r}) - \underline{s}_d = 0$ effectively represents the desired objective.

Let us for example consider the case of target tracking. This task may be expressed as the positioning of the projection of an object in the image center. The required visual signals are the coordinates (X_c, Y_c) of the object mass center in the image. The associated virtual linkage is of class 4, since we have at a desired position $X_c = Y_c = 0$:

$$(T_{ST}^*) = \begin{pmatrix} 0 & 0 & z_c & 0 \\ 0 & 0 & 0 & z_c \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (40)$$

To conclude, let us emphasize that similar study may be done when the configuration space of the camera is of dimension less than 6. In the general case, this restriction

comes from the kinematics of the system which handles the camera. When the added constraint is holonomic, the motion is locally restricted to a submanifold of SE_3 . If the associated constraints equations may be explicitly expressed in SE_3 or in the tangent space, they may be added to the set of visual-based constraints or directly to the interaction matrix. In practice, it is even often possible to use the previously defined linkages and to exploit motion restrictions to simplify them. The case of non-holonomic constraints is evoked in section 5.3.

3.3 Performance and Robustness of the Vision Based Tasks

Let us suppose that the geometric features to be controlled in the image have been chosen (for example, \underline{s} is a set of coordinates of peculiar points in the image like corners). Let us also assume that in order to complete the task, the control of a minimum of m features is needed, and that it exists, in the image, p features such that $p > m$ which can be potentially used. The question which arises is: what criterion may be used to select the ‘best’ set of signals, in terms of performance and robustness? To be significant, this criterion has to take into account several different aspects, like the quality of the features extraction (computation cost, accuracy, robustness...), or more specific control aspects (stability, convergence rate, decoupling...). Finding such a criterion is a hard task and a few works has been done in this area [7].

Presently, only the canonical case of a positioning task (rigid linkage) using the coordinates of three points in the image as visual signals, has been really investigated. Obviously, we assume that the three points in the scene which project on the image, constitute a rigid triangle. Although this case is the simplest (for a positioning task based on points as features), it allows us to point out some general properties which are required to the mapping which is defined by $L_{\underline{s}}^T$, between the location space (SE_3) and the measurement space (\mathbb{R}^6 in this particular case). In a certain sense, this mapping can be viewed as similar to the inverse kinematic model (from SE_3 to the joint space) well known in robotics. As in the case of the inverse kinematic model, it is interesting to address the problem of the unicity of the positioning between the camera and the object. In other words, is the mapping a one to one mapping ? This question, often referred as the “perspective-3-point” problem (P3P) [9] has been largely discussed and it can be shown that there are, in general, four configurations \bar{r} of SE_3 physically admissible which bring to a same image of the three points. A consequence of these multiple solutions, is that, reaching a value \underline{s}_d will not ensure that the right desired configuration \bar{r}_d has been really reached since the convergence towards a given solution will be strongly dependent of the initial configuration. Moreover, as in the case of robots, this multiplicity leads up to infer the existence of singularities in the interaction matrix.

We will therefore concentrate on another issue, by adopting a control point of view. Let us assume that the user's objective can be expressed as the regulation to zero of a n -dimensional C^2 function, $\underline{e}(q, t)$, called *task function*, during a time interval $[0, t_m]$ [20]. When sensors are used, the vector $\underline{s}(q, t)$ obviously contributes to the design of $\underline{e}(q, t)$ (see section 4). We can write:

$$\dot{\underline{e}} = \frac{\partial \underline{e}}{\partial \underline{q}} \dot{\underline{q}} + \frac{\partial \underline{e}}{\partial t} = \frac{\partial \underline{e}}{\partial \underline{s}} \frac{\partial \underline{s}}{\partial \bar{r}} \frac{\partial \bar{r}}{\partial \underline{q}} \dot{\underline{q}} + \frac{\partial \underline{e}}{\partial t} \quad (41)$$

The characteristics of the Jacobian matrix $\frac{\partial \underline{e}}{\partial \underline{q}}$ and of its model, which will be later used in a feedback control scheme in the space of visual signals, have a strong influence on the robustness and convergence properties of the control law. It is important to notice that this matrix is the product of three terms which are very different in nature:

- $\frac{\partial \bar{r}}{\partial \underline{q}}$ is the classical robot Jacobian which appears in virtually all robotics problems.
- $\frac{\partial \underline{s}}{\partial \bar{r}}$ is what we called the *interaction matrix* or task Jacobian. It is canonically related to the vision problem without any reference to a particular robot kinematics, this last fully appearing in the robot Jacobian.
- $\frac{\partial \underline{e}}{\partial \underline{s}}$ depends on the way we take into account the set of visual signals in order to build the task function. Here arises the tricky problem of visual signals selection, vastly discussed in [8]. Without addressing this question in details, and by considering the robustness of image features extraction, it is easy to understand that the most reliable image features in the sense of image processing should be preferred.

Let us assume that the robot Jacobian matrix $\frac{\partial \bar{r}}{\partial \underline{q}}$ is nonsingular in our working area and let us now focus on the task Jacobian. The important question which arises is: does it exist any configuration where this Jacobian becomes of rank not full? If so, may we characterize such configurations?

Unfortunately, no result may apparently be obtained in the most general case, and only the P3P case could be completed [15]. After intensive use of symbolic calculus, it was shown that, for the P3P problem, Jacobian singularities are encountered if and only if:

1. The three points are aligned (degenerate triangle): for any position of the optical center and attitude of the camera frame, a singularity always exists.
2. The optical center lies on the cylinder which includes the three points and the axis of which is perpendicular to the plane containing these points: for any attitude of the camera frame, the configuration is singular (see figure 5).

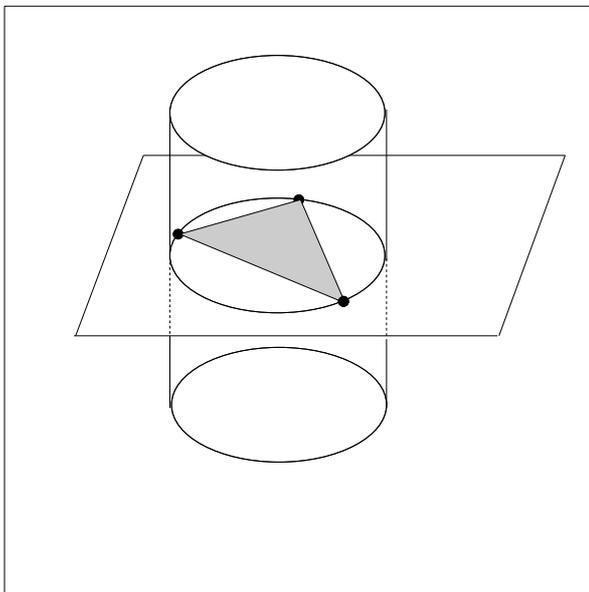


Figure 5: Cylinder of singularities

We may say in conclusion that the choice of well-suited visual signals is not easy and that work remains to be done to define a relevant criterion of signal selection. Even in the simple case of the three points, the computations are tedious and generalization to more sophisticated primitives is not straightforward. However, the P3P example is interesting since it points out an analogy between singularities in the robot's Jacobian and singularities coming from the task itself. A practical way to avoid this kind of singularities is clearly to introduce redundancy in the task by selecting more visual signals than necessary. For the rigid linkage case for example, four coplanar points are enough to avoid task singularities.

4. Expression of the Task Function and Control

Embedding visual servoing in the task function framework allows us to take advantage of general results helpful for analysis and synthesis of efficient closed loop control schemes taking explicitly into account redundancy in the measurements, for example in so-called hybrid control schemes [20]. We here only recall the obtained results, all the developments being fully described in [20] and, in the particular case of vision-based control, in [6].

We define a vision-based task, \underline{e}_1 , considered as the main one, with dimension $m \leq n$, $m = 6 - N$, N being the class of the virtual linkage associated to \underline{e}_1 . A secondary task, \underline{e}_2 , can thus be combined with \underline{e}_1 and is expressed as the minimization of a cost function h_s , with gradient $\underline{g}_s = (\frac{\partial h_s}{\partial \underline{r}})^T$. The task function, also called *hybrid task* when \underline{e}_2 consists in a trajectory tracking, achieves the goal of the minimization of h_s under the constraint $\underline{e}_1 = 0$ when it takes the form:

$$\underline{e} = W^+ \underline{e}_1 + (\mathbf{I}_n - W^+ W) \underline{g}_s^T \quad (42)$$

where the vision-based task \underline{e}_1 is written:

$$\underline{e}_1 = C(\underline{s}(\underline{q}, t) - \underline{s}_d) \quad (43)$$

The terms of the previous equations have the following meaning:

- \underline{s}_d is the desired value of the selected visual features;
- $\underline{s}(\underline{q}, t)$ is their current value, taken from the image at each time;
- W is a $m \times n$ full-rank matrix such that $\text{Ker } W = \text{Ker} \left(\frac{\partial \underline{e}_1}{\partial \underline{r}} \right)$;
- W^+ is the pseudo-inverse of W ;
- $(\mathbf{I}_n - W^+W)$ is an orthogonal projection operator on $\text{Ker } W$.
- $C = \frac{\partial \underline{e}_1}{\partial \underline{s}}$ is called combination matrix and can be defined as [6]:
 - $C = WL_{\underline{s}}^{T+}$ when the interaction matrix is known, or when its 3D terms (such as the range z of a point in Eq. 5) can be estimated on-line, using for example structure from known motion techniques [4]. In that case, W can be chosen such that $\text{Ker } W = \text{Ker } L_{\underline{s}}^T$.
 - $C = WL_{\underline{s}|\underline{s}=\underline{s}_d}^{T+}$ when isolated singularities may occur during the convergence of \underline{e} or when the value of the interaction matrix can not be updated at each time. Assumptions on the shape and on the geometry of the considered primitives in the scene have thus generally to be done and we set W such that $\text{Ker } W = \text{Ker } L_{\underline{s}|\underline{s}=\underline{s}_d}^T$.

A general control scheme aimed to regulate the task function \underline{e} is described in [20]. We here only present the simplified control scheme that we have used to perform the experimentations described in the next section. Similar control approaches and other ones using the optimal control theory can be also found in [11], [13] and [17]. Dynamics effects such as delay and mechanical vibrations are also studied in [5].

For making \underline{e} exponentially decrease and then behave like a first order decoupled system, we set:

$$\dot{\underline{e}} = -\lambda \underline{e} \quad \text{with } \lambda > 0 \quad (44)$$

We also have:

$$\dot{\underline{e}} = \frac{\partial \underline{e}}{\partial \underline{r}} T_S + \frac{\partial \underline{e}}{\partial t} \quad (45)$$

where the camera velocity screw T_S relies to $\dot{\underline{q}}$ through the inverse robot Jacobian matrix. We assume that the robot Jacobian is perfectly known and non singular, and that the setting variable tunable by the user is simply the desired joint velocity, $\dot{\underline{q}}$, as for most of industrial robots. Then, the desired camera velocity screw T_S can be considered as a pseudo-control vector. We thus obtain, by setting $\frac{\partial \underline{e}}{\partial \underline{r}}$ as the identity matrix \mathbf{I}_n , since C and W have been chosen in order to satisfy stability conditions (see [6] for more details):

$$T_S = -\lambda \underline{e} - \widehat{\frac{\partial \underline{e}}{\partial t}} \quad (46)$$

where:

- λ is the proportional coefficient involved in the exponential convergence of \underline{e} and has to be tuned in order to ensure the global stability of the system and to optimize the time to convergence [19];
- $\widehat{\frac{\partial \underline{e}}{\partial t}}$ can be written under the form:

$$\widehat{\frac{\partial \underline{e}}{\partial t}} = W^+ C \widehat{\frac{\partial \underline{s}}{\partial t}} + (\mathbf{I}_n - W^+ W) \frac{\partial g^T}{\partial t} \quad (47)$$

The secondary cost function generally allows to know $\frac{\partial g^T}{\partial t}$. On the other hand, vector $\widehat{\frac{\partial \underline{s}}{\partial t}}$ represents an estimation of the contribution of a possible autonomous target motion. If the target moves, this estimation has to be introduced in the control law in order to suppress tracking errors. It may be obtained using classical filtering techniques such as Kalman [12] [16] [19] or $\alpha - \beta - \gamma$ filters [1].

5. Results

We firstly present simulation results in the realization of two classical virtual linkages (prismatic/revolute and ball-and-socket ones) using a camera mounted on a six degrees of freedom robot arm. Then, other vision-based tasks using a camera mounted on a mobile robot with non-holonomic constraints will be considered.

5.1. Positioning with Respect to a Cylinder

Let us consider a motionless cylinder with radius r . We want to realize a prismatic/revolute linkage, with axis \vec{y} , between it and the camera (see section 3.2.4). At a right position, the cylinder equation is:

$$h(\underline{x}, \underline{p}) = x^2 + (z - z_d)^2 - r^2 = 0 \quad (48)$$

and its projection is characterized by two straight lines with equations:

$$\begin{cases} D_1 : X + r/\sqrt{z_d^2 - r^2} = 0 \\ D_2 : X - r/\sqrt{z_d^2 - r^2} = 0 \end{cases} \quad (49)$$

By choosing for \underline{s} the parameters ρ_1, θ_1, ρ_2 and θ_2 who represent these lines, we obtain $\underline{s}_d = (\rho_d, 0, \rho_d, \pi)$ with $\rho_d = -r/\sqrt{z_d^2 - r^2}$. The interaction matrix associated with \underline{s}_d is easily derived using Eq. 33:

$$L_{|\underline{s}=\underline{s}_d}^T = \begin{pmatrix} \lambda_\rho & 0 & -\lambda_\rho \rho_d & 0 & -(1 + \rho_d^2) & 0 \\ 0 & 0 & 0 & -\rho_d & 0 & -1 \\ -\lambda_\rho & 0 & -\lambda_\rho \rho_d & 0 & 1 + \rho_d^2 & 0 \\ 0 & 0 & 0 & \rho_d & 0 & -1 \end{pmatrix} \quad (50)$$

with $\lambda_\rho = -z_d/(z_d^2 - r^2)$. Since this matrix is of full rank 4, we may choose $W = L_{|\underline{s}=\underline{s}_d}^T$ and $C = WL_{|\underline{s}=\underline{s}_d}^{T+} = \mathbf{I}_4$. We therefore have:

$$\underline{e} = W^+(\underline{s} - \underline{s}_d) + (\mathbf{I}_6 - W^+W) \underline{g}_s^T \quad (51)$$

where:

$$W^+ = \begin{pmatrix} \lambda_\rho/2l & 0 & -\lambda_\rho/2l & 0 \\ 0 & 0 & 0 & 0 \\ -1/2\lambda_\rho\rho_d & 0 & -1/2\lambda_\rho\rho_d & 0 \\ 0 & -1/2\rho_d & 0 & 1/2\rho_d \\ -(1+\rho_d^2)/2l & 0 & (1+\rho_d^2)/2l & 0 \\ 0 & -1/2 & 0 & -1/2 \end{pmatrix}$$

$$(\mathbf{I}_6 - W^+W) = \begin{pmatrix} (1+\rho_d^2)^2/l & 0 & 0 & 0 & \lambda_\rho(1+\rho_d^2)/l & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_\rho(1+\rho_d^2)/l & 0 & 0 & 0 & \lambda_\rho^2/l & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{with } l = \lambda_\rho^2 + (1 + \rho_d^2)^2$$

Simulation results for this task are given in figure 6. Top windows represent the relative position of the camera (drawn as a pyramid) with respect to the cylinder. Middle windows show the camera image. Left and right windows correspond to initial and final positions respectively. Finally, in bottom windows, the time evolution of every component ($s_i - s_{d_i}$) (at left), of $\|\underline{s}(\vec{r}) - \underline{s}_d\|$ (in the middle) and of the six components of T_S are plotted.

In that simulation, we set $\lambda = 0.1$. The secondary task consists in moving with a constant velocity $V_x = 0.5$ cm/s along the \vec{x} camera axis and $V_y = -0.5$ cm/s along its \vec{y} axis. The secondary cost function is then:

$$h_s = \frac{1}{2}\beta_x(x - x_0 - V_x t)^2 + \frac{1}{2}\beta_y(y - y_0 - V_y t)^2 \quad (52)$$

where β_x and β_y are two positive scalar weights. Therefore:

$$\underline{g}_s^T = \begin{pmatrix} \beta_x(x - x_0 - V_x t) \\ \beta_y(y - y_0 - V_y t) \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial \underline{g}_s^T}{\partial t} = \begin{pmatrix} -\beta_x V_x \\ -\beta_y V_y \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (53)$$

and the velocity screw T_S to be used as an input in order to control \underline{e} is:

$$T_S = -\lambda \underline{e} + \begin{pmatrix} \beta_x V_x (1 + \rho_d^2)^2 / l \\ \beta_y V_y \\ 0 \\ 0 \\ \beta_x V_x (1 + \rho_d^2) \lambda_\rho / l \\ 0 \end{pmatrix} \quad (54)$$

Now, by choosing $\beta_x = l / (1 + \rho_d^2)^2$ and $\beta_y = 1$, the translational camera velocities along \vec{x} and \vec{y} axes have the desired values V_x and V_y when $\underline{e} = 0$.

5.2. Positioning with Respect to a Sphere

Figure 7 presents the realization of a virtual linkage of kind "ball-and-socket" between a motionless sphere and the camera (cf section 3.2.6). We have $\underline{s} = (X_c, Y_c, \mu)$, $\underline{s}_d = (0, 0, \mu_d)$ and:

$$L_{|\underline{s}=\underline{s}_d}^T = \begin{pmatrix} -1/z_d & 0 & 0 & 0 & -1 - \mu_d & 0 \\ 0 & -1/z_d & 0 & 1 + \mu_d & 0 & 0 \\ 0 & 0 & 2\mu_d/z_d & 0 & 0 & 0 \end{pmatrix} \quad (55)$$

We choose $W = L_{|\underline{s}=\underline{s}_d}^T$ and thus:

$$C = W L_{|\underline{s}=\underline{s}_d}^{T+} = \mathbf{I}_3 \quad (56)$$

For this task realizing the desired linkage, we therefore have $\underline{e}_1 = \underline{s} - \underline{s}_d$, and, since no secondary task was specified in this experiment, the desired camera velocity takes the form:

$$T_S = -\lambda W^+ \underline{e}_1 = -\lambda \begin{pmatrix} -z_d/2 & 0 & 0 \\ 0 & -z_d/2 & 0 \\ 0 & 0 & z_d/2\mu_d \\ 0 & 1/2(1 + \mu_d) & 0 \\ -1/2(1 + \mu_d) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} (\underline{s} - \underline{s}_d) \quad (57)$$

5.3. Vision Based Control Applied to Mobile Robot

Applying vision-based control theory to mobile robots provides with an interesting framework for implementing reflexive tasks. However it is necessary to pay a particular attention to non holonomic constraints, which imply that the dimension of the configuration space is larger than the number of degrees of freedom really controllable.

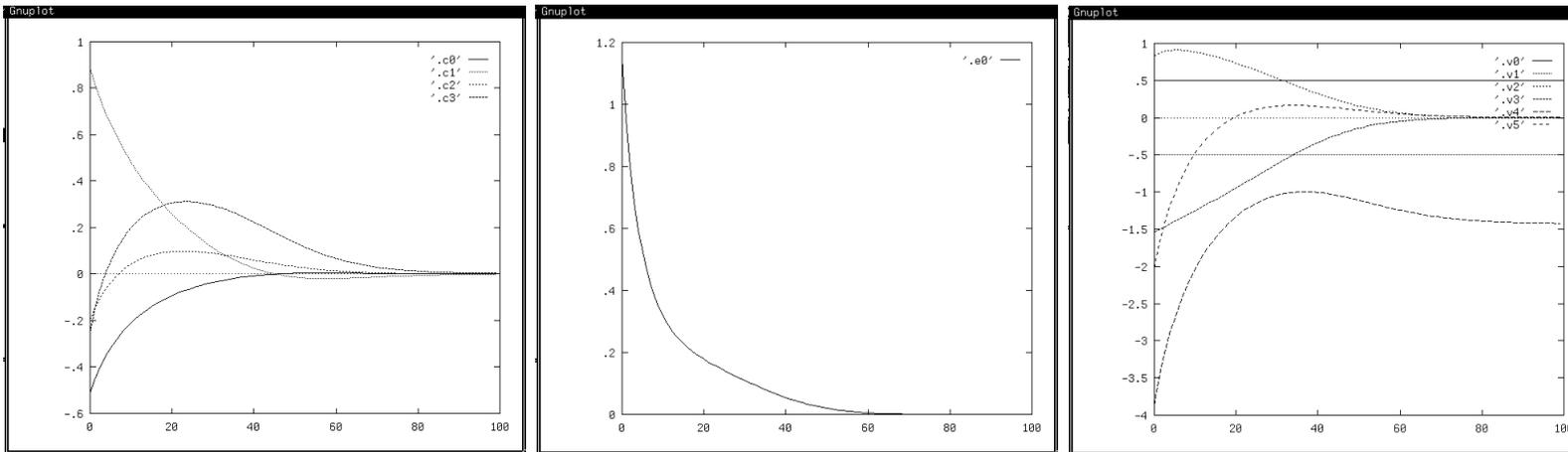
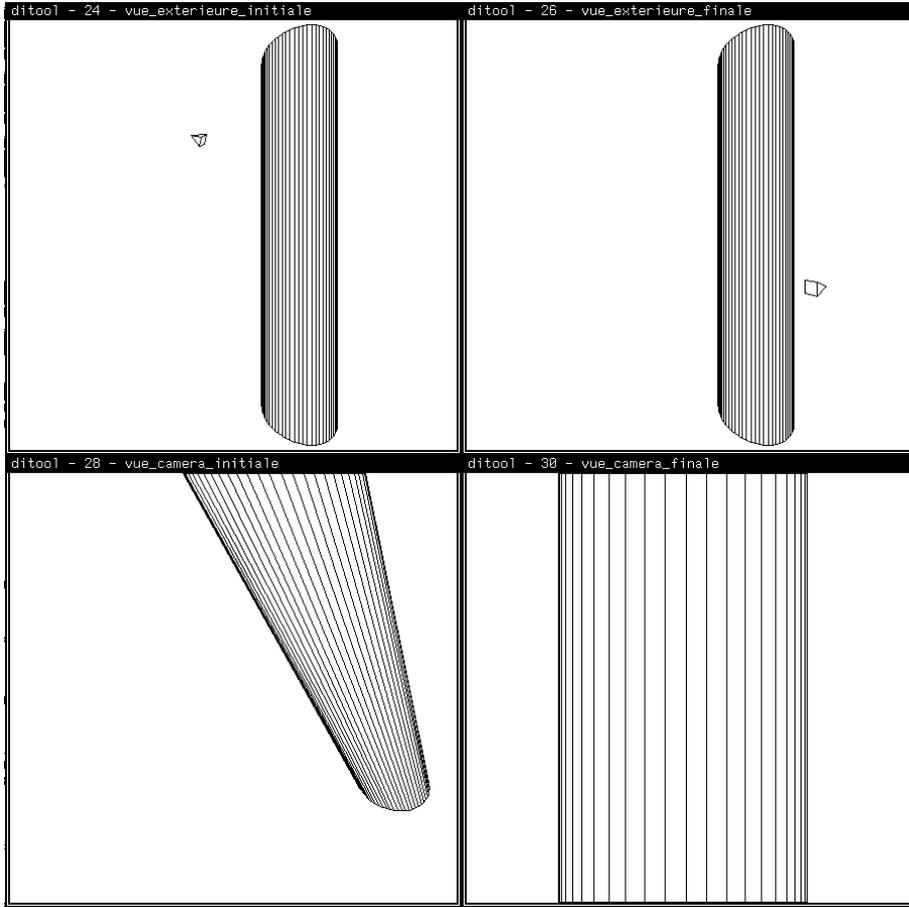


Figure 6: Positioning with respect to a cylinder

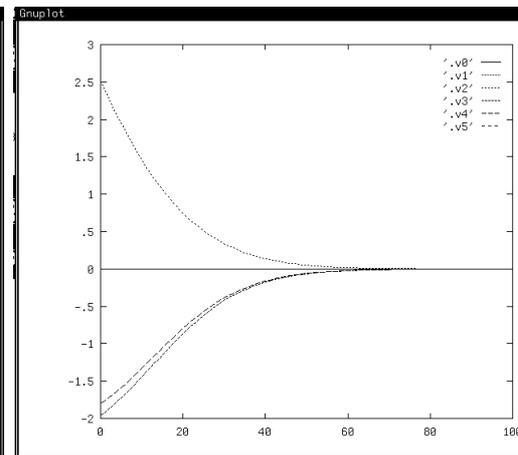
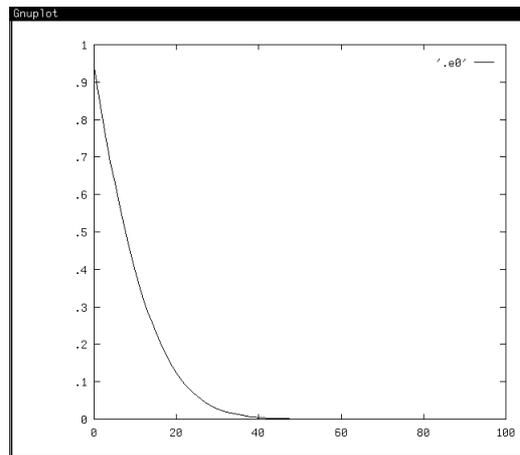
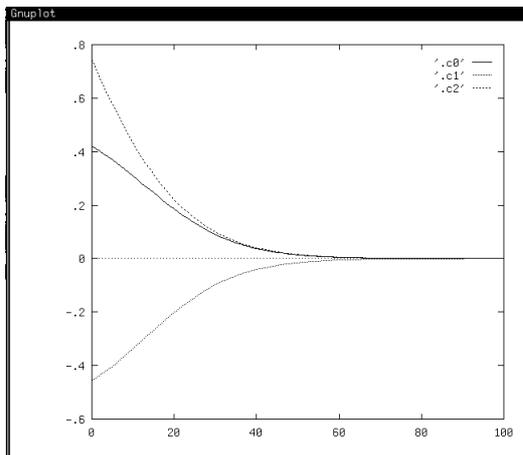
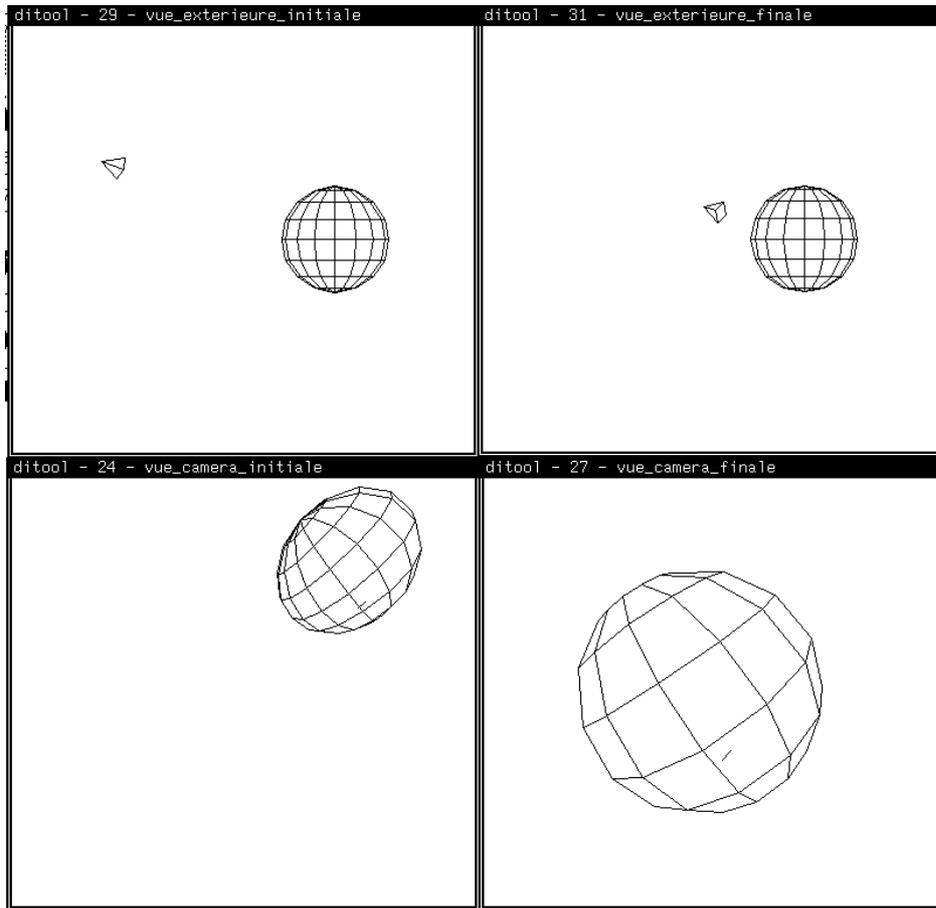


Figure 7: Positioning with respect to a sphere

Due to this fact, it is now known [21] that if we consider a two-wheeled non holonomic cart, despite the controllability of the system, the stabilization of the cart's configuration around a given terminal configuration is not possible using a pure stationary state feedback. This particularity has for consequence that, for a camera rigidly attached to the mobile base, we cannot ensure a convergence of the task error by using simple gradient-like methods in the image frame. One way to overcome this problem consists in adding some degrees of freedom to the camera by mounting it on the wrist of a manipulator. Then, if we consider the whole mechanical system constituted by the cart and its manipulator as a single kinematic chain, we can show that the mapping between the configuration space (d.o.f. of the cart + d.o.f. of the manipulator) and the location of the camera is nonsingular excepted in few isolated configurations. It becomes therefore possible to fully control the motion of the camera without being limited by the cart non holonomic constraints. By this way, we can execute visual servoing tasks directly defined in the camera frame in the same manner as previously done in the case of manipulators. Obviously, in such an approach, we are unable to control explicitly the trajectory of the cart but, by using some trivial geometric considerations, we can bound the volume which will be occupied by the cart during its displacement. This approach has been applied to a system constituted by a two-wheeled cart with a camera mounted on a three d.o.f. manipulator (see figure 8).

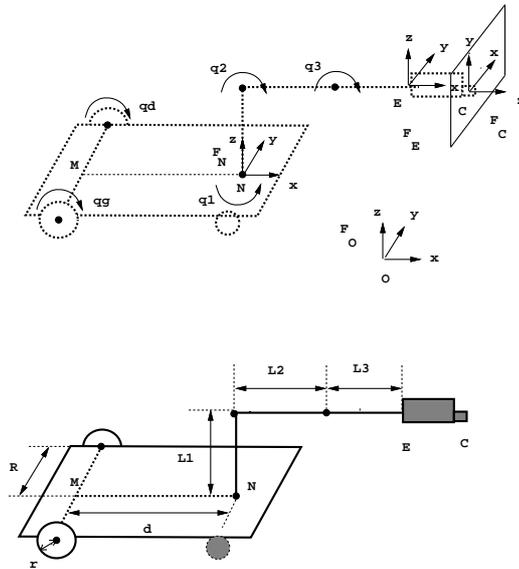


Figure 8: Robot Description: Frames and Notations

We present here only a few results which have been obtained in simulation (for more details concerning this application, see [18]). In order to validate visual servoing approach applied to mobile robot, we give a simple example where the robot task is defined by two successive visual servoing sub-tasks. We set the robot to an initial

position with respect to an environment constituted by several segments (see figure 9). Initial arm configuration is $q_1 = \pi, q_2 = -\frac{\pi}{4}, q_3 = 0$.

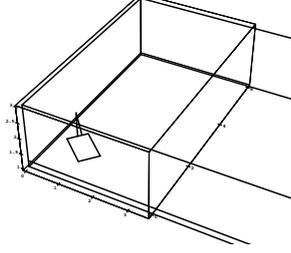


Figure 9: Initial Configuration of the Robot

The task consists of wall following. Knowing the size of the skirting board of the room, we can compute by Eq 9 the interaction matrix taking as features the two corresponding parallel lines. This gives:

$$L_{\underline{s}|\underline{s}=s_d}^T = \begin{pmatrix} -0.156323 & 0.236455 & -0.0509507 & 0.0991288 & -0.149942 & -1 \\ -0.240955 & 0.364468 & -0.0785346 & -0.861134 & -0.569307 & 0 \\ -0.158875 & 0.265158 & -0.0766261 & 0.127409 & -0.212643 & -1 \\ -0.243381 & 0.406197 & -0.117384 & -0.910519 & -0.545556 & 0 \end{pmatrix} \quad (58)$$

The matrix kernel, after reduction by taking into account that the rotation around the optical axis is fixed to zero, gives the camera trajectory which does not modify the feature in the image. This trajectory is defined by the camera screw:

$$T_{ST}^* = (1.118667, 1, 1, 0, 0, 0)^T \quad (59)$$

By applying this screw during 4 seconds in simulation (see figure 10), we illustrate that this “kernel trajectory” is a translation along the wall. We can see that the robot “manœuvres” in order to realize the aimed camera motion without using any heuristic method.

The second vision-based subtask consists in moving around the corner of the room. For doing that, we use the vertical line formed by the intersection of the walls. After a half-turn of the arm and moving q_2 on zero, we have now a vertical segment on the image. Taking this segment as a new feature for visual servoing and knowing the height of this segment, the simulator can compute the interaction matrix corresponding to this new task. Here again the reduced kernel is only of dimension one and we can deduce:

$$T_{ST}^* = (-1, 0.0000148184, 0.0145816, 0, 0.126032, 0)^T \quad (60)$$

In this second sub-task when this screw is applied to the camera motion, it rotates around the vertical segment (see figure 11).

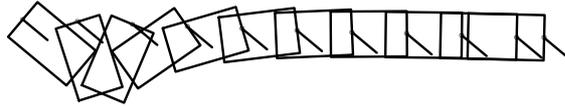


Figure 10: Wall Following

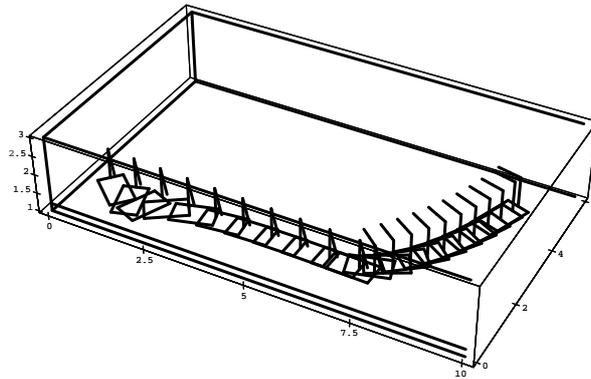


Figure 11: Rotation Around a Vertical Segment

Let us finally note that experimental results obtained with a six-jointed robot using the approach detailed in this paper are described in [6] and [3].

References

- [1] P.K. Allen, B. Yoshimi, A. Timcenko, “Real-Time Visual Servoing”, *IEEE Int. Conf. on Robotics and Automation*, Vol. 1, pp. 851-856, Sacramento, California, April 1991.
- [2] J. Barakan, J. C. Latombe, “On non Holonomic Mobile Robots and Optimal Maneuvering”, *Revue d’Intelligence Artificielle*, Vol. 3, n. 2, pp. 77-103, Ed. Hermes, Paris, 1989.
- [3] F. Chaumette, “La relation vision-commande : théorie et applications à des tâches robotiques”, *Ph-D Thesis*, Rennes I University, France, July 1990.
- [4] F. Chaumette, S. Boukir, “Structure from Motion using an Active Vision Paradigm”, *11th IAPR International Conference on Pattern Recognition*, Vol. 2, pp. 41-45, Den Haag, The Netherlands, August 1992.
- [5] P.I. Corke, M.C. Good, “Dynamics Effects in High Performance Visual Servoing”, *IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp. 1838-1843, Nice, France, May 1992.
- [6] B. Espiau, F. Chaumette, P. Rives “A New Approach to Visual Servoing in Robotics”, *IEEE Transactions on Robotics and Automation*, Vol. 8, n. 6, pp. 313-326, June 1992.
- [7] J. T. Feddema, C. S. G. Lee, O. R. Mitchell, “Automatic Selection of Image Features for Visual Servoing of a Robot Manipulator”, *IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp. 832-837, Scottsdale, Arizona, May 1989.
- [8] J. T. Feddema, C. S. G. Lee, O. R. Mitchell, “Weighted Selection of Image Features for Resolved Rate Visual Feedback Control”, *IEEE Transactions on Robotics and Automation*, Vol. 7, n. 1, pp. 31-47, February 1991.
- [9] M. A. Fischler, R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, *Communications ACM*, Vol. 44, pp. 381-395, 1981.
- [10] F. Harashima, H. Hashimoto, T. Kubota, “Sensor-based Robot Control Systems”, *IEEE Int. Workshop on Intelligent Motion Control*, Vol. 1, pp. 1-10, Istanbul, Turkey, August 1990.

- [11] K. Hashimoto, T. Kimoto, T. Ebine, H. Kimura, "Manipulator Control with Image-based Visual Servo", *IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 2267-2271, Sacramento, California, April 1991.
- [12] A.E. Hunt, A.C. Sanderson, "Vision-Based Predictive Robotic Tracking of a Moving Object", *Technical Report CMU-RI-TR-82-15*, Carnegie-Mellon University, January 1982.
- [13] W. Jang, Z. Bien, "Feature-based Visual Servoing of an Eye-In-Hand Robot with Improve Tracking Performance", *IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 2254-2260, Sacramento, California, April 1991.
- [14] J. P. Laumond, "Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints", *Int. Conf. on Intelligent Autonomous Systems*, pp. 346-354, Amsterdam, The Netherlands, 1986.
- [15] H. Michel, P. Rives, "Singularities in the Determination of the Situation of a Robot Effector from the Perspective View of Three Points", *To appear as INRIA Research Report*, Sophia-Antipolis, France, 1993.
- [16] N. Papanikolopoulos, P.K. Khosla, T. Kanade, "Vision and Control Techniques for Robotic Visual Tracking", *IEEE Int. Conf. on Robotics and Automation*, Vol. 1, pp. 857-864, Sacramento, California, April 1991.
- [17] N. Papanikolopoulos, B. Nelson, P.K. Khosla, "Full 3D Tracking using the Controlled Active Vision Paradigm", *7th IEEE Symposium on Intelligent Control*, Th. Henderson editor, pp. 267-274, Glasgow, Scotland, August 1992.
- [18] P. Rives, R. Pissard-Gibollet, "Reactive Mobile Robots Based on a Visual Servoing Approach", *13rd Annual Conference on Artificial Intelligence, Simulation and Planning*, Perth, Western Australia, July 1992.
- [19] A. Santos, F. Chaumette, "Target Tracking by Visual Servoing", *IRISA Research Report*, n. 683, Rennes, France, October 1992.
- [20] C. Samson, B. Espiau, M. Le Borgne, *Robot Control: the Task Function Approach*. Oxford University Press, 1991.
- [21] C. Samson, K. Ait-Abderrahim, "Feedback Control of a non Holonomic Wheeled Cart in Cartesian Space", *IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp. 1136-1141, Sacramento, California, April 1991.
- [22] L. E. Weiss: "Dynamic Visual Servo Control of Robots. An Adaptive Image based Approach", *Technical Report CMU-RI-TR-84-16*, Carnegie-Mellon University, April 1984.