SOME ISSUES IN THE CONTROL OF RIGID ROBOTS IN A SENSORY SPACE

Bernard Espiau*, François Chaumette**

(*) ISIA-ENSMP, Rue Claude Daunesse, 06565 Valbonne Cedex, France
(**) IRISA-INRIA Rennes, Campus de Beaulieu, 35042 Rennes, France

Abstract: The aim of this paper is to examine some problems related to robot control in an output sensory space, with focus on the particular case of visual sensors. A single sensor is here considered as a mapping from SE_3 to $\mathbb{R}$. This assumption allows us to set that the related jacobian is a particular set of screws, which authorizes some unification in the analysis. Then, we define the concept of virtual linkage as a way of expressing the constraints induced by the sensors. We then use the redundancy approach of C. Samson in order to design a correct task function, specifying in that way the right output working space. Some facts in the control are then recalled.

All the analysis, originally suited for proximity, range or force sensors, is applied to the case of visual sensors. The various involved mappings and the needed assumptions are precised; the design of adequate task functions which use special image features is presented, with some indications on the practical derivation of the jacobian models. Some results and comments are finally given.

1 Introduction

1.1 Notation

The following notation will be employed:

Let E be the three-dimensional affine euclidean space, the related vector space being $\mathbb{R}^3$. The configuration space of a rigid body, which is also the frame configuration space, is the Lie group of displacements, SE_3 (Special Euclidean Group), isomorphic to $\mathbb{R}^3 \times SO_3$ where SO_3 is the group of rotations. It is a six-dimensional differential manifold. An element of SE_3, called a 'position' (i.e. location and attitude owing to the previous isomorphism) is denoted as $\bar{r}$. The tangent space to SE_3 at identity is denoted as se_3, and its dual, or cotangent space, se_3^*. se_3 is a Lie algebra isomorphic to the Lie algebra of equiprojective fields of E in $\mathbb{R}^3$, which means that any element (field) of se_3 is no more

than the classical velocity screw. A screw H is also defined by its vector u and the value of its field in a point P of E. We may therefore write: $H = (H(P), u)$.

Frames are denoted as F, with origin O. A given screw expressed in F is then also a vector in $\mathbb{R}^6$. Finally, the velocity of a frame F_i with respect to a frame F_j is denoted as V_{ij}.

The screw product is the bilinear mapping associated with $\begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix}$. It may be written, for any considered point P:

$$H_1 \bullet H_2 = <u_1, H_2(P)> + <u_2, H_1(P)> \tag{1}$$

where $<,>$ is the usual scalar product between two vectors of $\mathbb{R}^3$. Let S be a screw space. The screw product induces an isomorphism between S and its dual S^*, which is itself a screw space.

We will denote the skew-symmetric matrix associated with a vector of $\mathbb{R}^3$ as $As(.)$; a matrix A $(n \times n)$ will be said positive if $x^T A x > 0$, $\forall x \neq 0 \in \mathbb{R}^n$.

We will also consider a rigid robot, the state equation of which is given by

$$\Gamma = M(q)\,\ddot{q} + N(q, \dot{q}, t) \ , \ \dim(q) = \dim(\Gamma) = n \tag{2}$$

where Γ is the vector of applied external forces (actuator torques), M is the kinetics energy matrix, N gathers gravity, centrifugal, Coriolis and friction forces, and $(q, \dot{q})$, the joint position and velocity, is the state vector of the system.

It is assumed that an actuator is associated to every degree of freedom of the robot. We will also assume here for simplicity that $n = 6$. Let F_6 be a frame linked to the 'last' body, and F_0 a reference frame. The robot jacobian, J, is the jacobian associated to the mapping from $q \in \mathbb{R}^6$ to the position of F_6 with respect to F_0. We do not consider here the case where $J(q)$ falls singular. Some techniques to cope with this problem are presented in [11].

1.2 A Flavour of the Problem

Decoupling and linearizing (2) in the joint space is trivial as soon as the dynamics is known and computed. However, control in joint space is generally of little interest for the user: it is at least wished to control the position of F_6. The ideal decoupling and linearizing control takes then the form:

$$\Gamma = M(q)\,J^{-1}(q)\,u + N(q, \dot{q}) - M(q)\,J^{-1}(q) \begin{bmatrix} \vdots \\ \dot{q}^T W_i(q, t) \dot{q} \\ \vdots \end{bmatrix} \tag{3}$$

where $W_i(q,t)(i = 1, \cdots, n)$ is the partial derivative of the i-th row of $J^T(q)$ with respect to q, and u the new control vector. The need for nonsingularity of $J(q)$ appears here. Nevertheless, this kind of control, in SE_3, is not suitable in more complex (and interesting) applications, especially when exteroceptive sensors are used. Another working space is then required. This situation is a particular case of the more general 'control in task space', developed in [8] and [11], which will be briefly stated in section 4.

We will therefore try in the present paper to show how a control in sensory space, extending in some way the scheme (3), may be designed, and we will apply this approach to the case of visual sensors (other cases are examined in [11] and [4]). It should be emphasized that, in robotics, this area, known as 'visual servoing', is not as largely investigated as classical robot vision. Some relevant references are [2], [5], [6], [7], [13], [14], [12]. The related works will not be discussed here, since done in [3], to which we refer the reader.

In fact, modelling aspects and design of the adequate task function (i.e. of the output space associated to (2)) are the most delicate points, and we shall focus the development on these aspects. Section 2 will be therefore devoted to general considerations on sensor-environment interactions; the concept of hybrid task (i.e combining tasks expressed in SE_3 and in a sensory space) is presented in section 3, while section 4 examines the specific case of visual features. Finally experimental results are presented in section 5, followed by few remarks on the need for on-line estimation schemes.

2   Modelling of Sensor-environment Interaction

2.1   The Interaction Screw

We restrict our study to the case where, formally:

- *A sensor is completely defined by a C^2 mapping from SE_3 to $\mathbb{R}^k$.*

This assumption implies in particular that, for a given sensor, relative environmental modifications of the geometrical kind are the only ones allowed to make the sensor output varying. This is true for many kinds of proximity, range force and visual sensors. Let us now link a frame F_T to the part of environment observed by the sensor, and another, F_S, to the sensor itself. The reference frame may be F_S or F_T, or, even, when the environment is time-invariant, any associated frame F_o. The sensor output s may then be written: $s(F_S, F_T)$. Furthermore, let us assume that the sensor mobility is got through a generalized coordinate system, q, which constitutes a local chart of SE_3. Then, when the observed objects are autonomously mobile themselves, s may be also written $s = s(q,t)$, the independent time variable t representing this contribution of the objects motion. The

six variables q_i are for example the joint angular positions of a rigid manipulator which handles a camera.

Let us now examine a one-dimensional component s_j of s. Owing to the above preliminaries and to the definition of s, we know that its differential at $\bar{r}$, $ds_{j|_{\bar{r}}}$, is a linear mapping from $se_{3|_{\bar{r}}}$ to $\mathbb{R}$. It is also known that the differential of any analytic function from a manifold M to $\mathbb{R}$ may be identified with an element of the cotangent space. In our case, this implies that the differential of s_j at $\bar{r}$ is simply an element of se_3^*, that is to say a screw. Recalling that an element of se_3 is the velocity screw V, we may finally write at $\bar{r}$ in SE_3 the basic screw product:

$$\dot{s}_j = H_j \bullet V_{ST} \tag{4}$$

where V_{ST} is the velocity of the frame F_T with respect to the frame F_S, $\bullet$ is the screw product defined above, and H_j is a screw, the expression of which depends both on the environment characteristics and on the sensor itself. It therefore fully characterizes the interaction between a sensor and its environment, and we thus call it *Interaction Screw*.

2.2 The Concept of Virtual Linkage

A set of *compatible* and *independent* constraints, $s(\bar{r}) - s^* = 0$, where s^* is stationary, constitutes a **virtual linkage** between the sensor (S) and the objects of the environment (T). Let thus V^* be a virtual motion at $\bar{r}$ keeping constant the sensor output component s_j, i.e. preserving the satisfaction of the jth constraint. V^* is solution of the equation:

$$H_j \bullet V^* = 0 \tag{5}$$

and is therefore a screw *reciprocal* to H_j. Let us now return to the full sensor output vector, s, with dimension k. The set of the motions V^* leaving s invariant is S^*, the subspace reciprocal to the screw subspace S spanned by the set $\{H_1 \cdots H_j \cdots H_k\}$ in se_3.

In a position where these constraints are satisfied, the dimension, N, of S^* may be called the **class** of the virtual linkage in $\bar{r}$.

- **Remark:** With an obvious breach of notation, equation (4) may also be written:

$$\dot{s}_j = L_j^T V_{ST} \quad \text{where} \quad L_j^T = H_j \begin{bmatrix} 0 & I_3 \\ I_3 & 0 \end{bmatrix} \tag{6}$$

L_j^T is the matrix-form of the interaction screw H_j, in a given frame F and in a chosen point O. In the same way, the matrix form of the set $\{H_1 \cdots H_k\}$ is called *Interaction Matrix*, and is denoted as L^T. With a similar breach of notation, we may write $S^* = \text{Ker } L^T$.

The interest of this approach lies both in its generality and in the unification it realizes. Let us simply emphasize two aspects of these advantages:

- **Computation:** Knowing that the robot direct kinematics is a mapping from $\mathbb{R}^n$ to SE_3, it is easy to see that every column of the robot jacobian matrix, J, is the matrix representation of a screw. In most of the computations needed in practice, the three useful transformations are therefore simply:

  - the change of basis (i to j), with matrix R_{ij} (rotation matrix);

  - the change of frame (F_i to F_j), with homogeneous matrix $\bar{R}_{ij} = \begin{bmatrix} R_{ij} & [O_iO_j]_{F_i} \\ 0 & 1 \end{bmatrix}$

  - the change of reference frame for screws, with matrix $Ad_{ij} = \begin{bmatrix} R_{ij} & R_{ij}As(O_iO_j)_{F_j} \\ 0 & R_{ij} \end{bmatrix}$
  which is required for transforming the robot jacobian expression.

  Since the assumption made in sensor modelling leads also to consider sets of screws, it appears that the three transformations above are also the only ones to be used. This finally allows to obtain some unity in the computational issues.

- **Virtual linkage:** This concept may be related to the basic kinematics of contacts, as classically used in the theory of mechanisms. The idea 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 will also establish a connection with the approach known in the litterature as 'hybrid control', which is traditionnaly used in control schemes involving contact force sensors. This finally shows that many types of sensors may be used within a single framework: the one of *hybrid tasks* which realize *virtual linkages*.

3   Tasks and Control Design

3.1   The concept of task function

The dynamic behaviour of a rigid manipulator is described by equation (2). The task to be performed may then be specified as an *output function* associated to (2). More precisely, it may be shown ([11]) that the user's objective may in general be expressed as the regulation to zero of some n-dimensional C^2 function, $e(q,t)$, called *task function*, during a time interval $[0,T]$. An immediate example of task function is

$$e(q,t) = x(q) - x_d(t) \tag{7}$$

where $x_d(t)$ is for example a parametrization of the desired position of a robot wrist in SE_3. Many other cases are presented in [11]. When sensors are used, it appears that the sensor vector $s(q, t)$ has to contribute to the design of the task function, in a way explained later.

As detailed in [8] and [11], the problem of regulating e is well-posed if e has some specific properties. One of them is the existence and the unicity of a C^2 *ideal trajectory*, $q_r(t)$, such that $e(q_r(t), t) = 0$, $t \in [0, T]$ and $q_r(0) = q_0$, where q_0 is a given initial condition. Another one, very important, is the non-singularity of the task-jacobian matrix $\frac{\partial e}{\partial q}(q, t)$, around $q_r(t)$. When all the required conditions are satisfied, which will be implicitly assumed in the following, the task function is said to be 'admissible'. Efficient control laws may then be designed.

3.2 Control and stability

We only give here an intuitive idea of the used approach and of the obtained results. All the related developments may be found in [8] and [11]. Let us consider the exact decoupling and feedback linearization in the task space: in a way similar to (3), it is easy to see that an adequate control is:

$$\Gamma = M \left(\frac{\partial e}{\partial q} \right)^{-1} u' + N - M \left(\frac{\partial e}{\partial q} \right)^{-1} f \tag{8}$$

with:

$$f(q, \dot{q}, t) = \left[\begin{array}{c} \vdots \\ \dot{q}^T W_i(q, t) \dot{q} \\ \vdots \end{array} \right] + 2 \frac{\partial^2 e}{\partial q \partial t}(q, t) \, \dot{q} + \frac{\partial^2 e}{\partial^2 t}(q, t) \tag{9}$$

where $W_i(q, t)(i = 1, \cdots, n)$ is the partial derivative of the i-th row of $\left(\frac{\partial e}{\partial q} \right)^T (q, t)$ with respect to q. We may choose a PD feedback of the form:

$$u' = -\lambda G \left(\mu D e + \dot{e} \right) \tag{10}$$

G and D being positive matrices, λ and μ being positive scalars, all to be tuned by the user.

The ideal control scheme (8) (10) requires a perfect knowledge of all its components, which is neither possible, nor even wished. A more realistic approach consists in generalizing the previous control as:

$$\Gamma = -\lambda \hat{M} \left(\frac{\widehat{\partial e}}{\partial q} \right)^{-1} G \left(\mu D e + \frac{\widehat{\partial e}}{\partial q} \dot{q} + \frac{\widehat{\partial e}}{\partial t} \right) + \hat{N} - \hat{M} \left(\frac{\widehat{\partial e}}{\partial q} \right)^{-1} \hat{f} \tag{11}$$

where the carets point out that models (approximations, estimates) are used instead of the true terms. In this general expression, all the terms but μ, D and G are allowed to be functions of q and t, even of $\dot{q}$ for $\lambda, \hat{f}$ and $\hat{N}$.

A stability analysis of the system (2) with control (11) was done by Samson ([11]) in a nonlinear framework. Two main classes of sufficient stability conditions (in the sense of the boundedness of $\|e(t)\|$) were then exhibited: **gain** conditions (these tuning parameters leave more or less possibilities to the user) and **modelling** conditions. Among them, the most critical concerns the task itself, and has the form:

$$\frac{\partial e}{\partial q}\left(\widehat{\frac{\partial e}{\partial q}}\right)^{-1} > 0 \tag{12}$$

This essential condition allows to characterize the robustness of the task itself with regard to uncertainties and approximations.

It may already be noticed that, when we are interested in the motion of the end effector, we may write $\frac{\partial e}{\partial q} = \frac{\partial e}{\partial \bar{r}} \frac{\partial \bar{r}}{\partial q}$, where $\frac{\partial \bar{r}}{\partial q}$ is the robot jacobian matrix, J. When it is known and nonsingular, as we shall assume afterwards, the choice $\widehat{\frac{\partial e}{\partial q}} = \widehat{\frac{\partial e}{\partial \bar{r}}} J$ allows condition (12) to be reduced to:

$$\frac{\partial e}{\partial \bar{r}}\left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} > 0 \tag{13}$$

3.3 Hybrid Tasks

Regulating sensor signals is generally not the unique user's objective; very often, this task has to be combined with another such that a trajectory tracking.

Generally, the problem specification leads in a first step to defining a sensor-based task vector, $e_1(q, t)$, with $m \leq n$ independent components, the regulation of which constitutes the part of the global task which requires the use of exteroceptive sensors. How to derive such a vector when using visual sensors will be described later. A second objective, for example a desired sensor motion, might me represented in a first glance by a second vector $e_2(q, t)$. However, e_1 and e_2 would be gathered in a single task vector $e(q, t)$ admissible, such that the two tasks are compatible and independent.

It may indeed be shown that a more efficient way of setting the problem consists in embedding it in the framework of *task redundancy*. In this approach, e_1 is considered as prioritary, and e_2 is defined as the representation of the constrained minimization of a secondary cost function.

3.3.1 The Redundancy Framework

Let us assume that $J = \frac{\partial r}{\partial q}$ is known and nonsingular everywhere needed. Let e_1 be a m-dimensional main task, with jacobian matrix J_1^r $(= \frac{\partial e_1}{\partial r})$ in SE_3, and let h_s, with gradient $g_s^r = \frac{\partial h_s}{\partial r}$, be a secondary cost function to be minimized (the choice of h_s is discussed in [11]). Minimizing h_s under the constraint $e_1 = 0$ requires the subspace of motions left free by this constraint to be determined. This comes back to knowing the null space of J_1^r, Ker J_1^r (or the range of $J_1^{r^T}$, $R(J_1^{r^T})$) along the ideal trajectory. In other words, it has to be found any $m \times n$ full rank matrix W, such that:

$$R(W^T) = R\left(J_1^{r^T}\right) \tag{14}$$

along the robot's ideal trajectory, $q_r(t)$.

Once this matrix is determined, it may rather easily be shown ([9], [11]) that a task function minimizing h_s under the constraint $e_1 = 0$ is:

$$e = W^+ e_1 + \alpha \left(I_n - W^+ W\right) g_s^{r^T} \tag{15}$$

where α is a positive scalar, W^+ is the pseudo-inverse of W, and $(I_n - W^+ W)$ is an orthogonal projection operator on the null space of W, i.e. on that of J_1^r.

- **Remark:** When e_1 is made from sensor signals and h_s expresses a trajectory tracking task in SE_3, the task represented by (15) is then called 'hybrid task'.

It clearly appears that the computation of the jacobian matrix related to (15), possibly required in the control scheme, may be complex. The positivity condition (13) may then be of some interest. It may indeed be shown that if, in addition to (14), W satisfies the property:

$$J_1^r W^T > 0 \tag{16}$$

along $q_r(t)$, then, under 'normal circumstances' (see [11]) the jacobian matrix of e in SE_3 is such that

$$\frac{\partial e}{\partial \bar{r}}\left(I_n + \gamma\left(I_n - W^+ W\right)\right) > 0 \tag{17}$$

along $q_r(t)$, and $\forall \gamma \geq \gamma_m(\alpha) \geq 0$. The condition (13) is therefore satisfied by taking:

$$\left(\frac{\widehat{\partial e}}{\partial \bar{r}}\right)^{-1} = \left(I_n + \gamma\left(I_n - W^+ W\right)\right) \tag{18}$$

More, when α is 'small enough', then $\gamma_m = 0$, $\frac{\partial e}{\partial \bar{r}}$ is positive, and we may choose:

$$\frac{\widehat{\partial e}}{\partial \bar{r}} = I_n \tag{19}$$

3.3.2 The Specific Case of Sensor Signals

Let us know apply this approach to the use of sensor signals as defined in section 2.1. Let us recall that the vector s is of dimension k. Recall that the jacobian of s in SE_3 corresponds to the interaction matrix L^T. The dimension of L is $6 \times k$ and its rank is m, $N = 6 - m$ being the class of the associated virtual linkage.

We are interested in regulating s around a desired value or trajectory $s^*(t)$. Let $C(t)$ be a 'combination matrix', with dimension $m \times k$, such that the matrix CL^T is of full rank m along $q_r(t)$. The main task may then be written ([10]):

$$e_1 = C(t) \left(s(\bar{r}, t) - s^*(t) \right) \tag{20}$$

One of the advantages of the existence of C is the possibility of taking into account more sensors (k) than the actual dimension of the constraints they specify (m).

The jacobian matrix of e_1 in SE_3 is then $J_1^{\bar{r}} = \frac{\partial e_1}{\partial \bar{r}} = CL^T$ and we may easily show that $R(J_1^{\bar{r}T}) = R(L)$. Owing to (15), the task to be regulated may finally be written:

$$e = W^+ C \left(s(\bar{r}, t) - s^*(t) \right) + \alpha \left(\mathbf{I}_6 - W^+ W \right) g_s^{\bar{r}T} \tag{21}$$

W must ideally satisfy property (14), which then becomes $R\left(W^T\right) = R(L)$; this also means that the rows of W are made from basis vectors of S. Finally, property (16) becomes

$$CL^T W^T > 0 \tag{22}$$

which prevents C and W from being chosen independently. For example, (22) may be satisfied by selecting: $C = WL$ or $C = WL^{Tg}$ where L^{Tg} is the generalized inverse of L^T. In many cases, C and W may be chosen constant, as assumed in the following.

3.3.3 Model of $\frac{\partial e}{\partial t}$

In the control equation (11), an expression of the term $\frac{\widehat{\partial e}}{\partial t}$ is needed. Considering again the task function given by (21), we then have:

$$\frac{\partial e}{\partial t} = W^+ \frac{\partial e_1}{\partial t} + \alpha \left(\mathbf{I}_n - W^+ W \right) \dot{g}_s^{\bar{r}T} \tag{23}$$

Vector $\frac{\partial e_1}{\partial t}$ represents, when C is constant, the contribution of a possible autonomous target motion and is in general unknown. The choice made in many cases is $\frac{\widehat{\partial e_1}}{\partial t} = 0$. If the target moves, this choice may lead to a tracking error, the size of which decreases with λ. On the other hand, if, as in trajectory tracking, the used secondary cost function allows to know $\dot{g}_s^{\bar{r}}$, we may choose:

$$\frac{\widehat{\partial e}}{\partial t} = \alpha \left(\mathbf{I}_n - W^+ W \right) \dot{g}_s^{\bar{r}T} \tag{24}$$

4   Case of a Visual Sensor

4.1   Framework

Let us reduce a camera to a perspective projection model (Figure 1). All the used variables, for example the camera velocity screw $(v(O), \omega)$, denoted in the following as V_c, the point coordinates, the interaction screws, will be assumed to be expressed in the frame $(O, \vec{x}, \vec{y}, \vec{z})$ linked to the camera.
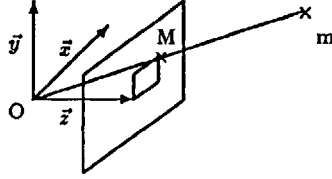

Figure 1: A simple camera model

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

$$\underline{X} = \frac{1}{z}\,\underline{x} \tag{25}$$

- **Remark:** It would seem to be necessary to complete this geometrical model with a photometric model. Some arguments given in [3] justify the absence of such a model in the present development.

Let us consider a single rigid solid in E, to which tridimensional primitives (points, lines, vertices...) may be associated. A set of such primitives is called a 'scene feature'. A configuration of the scene feature is an element p of the set $\mathcal{P}_s$ of all possible configurations. When a scene feature fully characterizes the position (location and attitude) of the associated rigid body, the dimension n' of $\mathcal{P}_s$ is 6. Otherwise, it is smaller.

Let us denote as f the perspective projection mapping, with $f(p) = P \in \mathcal{P}_i$; dim $\mathcal{P}_i = m < n'$. P is called an 'image feature'. It is assumed that p belongs to an open subset $U \subseteq \mathcal{P}_s$ such that $P = f(p)$ is not a degenerated element of $\mathcal{P}_i$ (case where, for example, a line projects onto the image plane as a point, a circle as a segment,...). This restriction implies that it exists a complete parametrization $\underline{P}$ of P on the open set $V = f(U) \subseteq \mathcal{P}_i$. Moreover, we assume that this parametrization is differentiable and minimal (the dimension of $\underline{P}$ is m).

Let $\underline{p}$ a complete and unique parametrization of p in U (in the sense that a single parametrization is necessary and sufficient to represent any configuration of the scene

feature p in U. The dimension of $\underline{p}$ is $n \geq n'$. If we write $\underline{p} = \phi(p)$ and $\underline{P} = \psi(P)$, we have (see Figure 2):

$$\underline{P} = \psi \circ f \circ \phi^{-1}(\underline{p}) \tag{26}$$

$$
\begin{array}{ccccc}
W \subseteq SE3 & \longrightarrow & U \subseteq \mathcal{P}_s & \longrightarrow & V \subseteq \mathcal{P}_i \\
(\bar{r}) & \delta & (p) & f & (P) \\
 & & \Big\downarrow \phi & & \Big\downarrow \psi \\
\mathbb{R}^n & \longrightarrow & \mathbb{R}^m & \longrightarrow & \mathbb{R} \\
(\underline{p}) & \psi \circ f \circ \phi^{-1} & (\underline{P}) & \sigma & (s)
\end{array}
$$

Figure 2: From $\bar{r}$ to s

Finally the group of displacements acts on $\mathcal{P}_s$ through the mapping δ and we assume that the displacements are restricted to an open set W of SE_3 such that $\delta(W) \subseteq U$ in order to prohibit any degenerated case. We then have:

$$\underline{p} = \phi \circ \delta(\bar{r}) \tag{27}$$

whence:

$$\underline{P} = \psi \circ f \circ \delta(\bar{r}) \tag{28}$$

The components s_j of the 'signal sensor' will then be chosen as a function $s_j = \sigma(\underline{P})$ such that σ is differentiable, the most frequent situation being to choose for s_j a component of $\underline{P}$.

The derivation of the interaction matrix reduces therefore to the computation of the expression $\frac{\partial s}{\partial \underline{P}} \frac{\partial P}{\partial P} \frac{\partial P}{\partial p} \frac{\partial p}{\partial \bar{r}}$, and, often practically to the computation of $\frac{\partial s}{\partial \underline{P}} \frac{\partial P}{\partial p} \frac{\partial p}{\partial \bar{r}}$. This form is not always the most adequate for an analytic computation of L^T, and another method may sometimes be preferred which allows to directly obtain $\frac{\partial \underline{P}}{\partial \bar{r}}$.

Indeed, the ith primitive of a scene feature may generally be described by an equation of type:

$$h_i(\underline{x}, \underline{p}_i) = 0 \tag{29}$$

where h_i defines the kind of the primitive and the value of $\underline{p}_i$ corresponds to one of its configurations.

In practice, the interaction matrix associated to the parametrization $\underline{P}_i$ of each primitive is computed, and the global interaction matrix associated to $\underline{P}$ is obtain by concatenation of all the elementary interaction matrices. In the sequel, h_i, p_i, $\underline{p}_i$, P_i and $\underline{P}_i$ related to the ith primitive will be respectively denoted h, p, $\underline{p}$, P and $\underline{P}$. By using (25),

equation (29) becomes:

$$h(z\underline{X}, \underline{p}) = 0 \tag{30}$$

i.e.

$$h'(\underline{X}, z, \underline{p}) = 0 \tag{31}$$

Under the condition $\frac{\partial h'}{\partial z} \neq 0$ which is ensured in all the non-degenerated cases, the implicit function theorem ensures the existence of a unique function μ around a solution $\underline{x}_0$ of (31), such that:

$$z = \mu(\underline{X}, \underline{p}) \tag{32}$$

Let us give two simple examples of basic primitives. For each of them, we give the function μ and the projection in the image of $h(\underline{x}, \underline{p}) = 0$, which will be written under the form:

$$g(\underline{X}, \underline{P}) = 0 \tag{33}$$

- *Case of a point*: Let m_i be a point of E with coordinates $\underline{x}_i = (x_i \, y_i \, z_i)^T$. We have:

$$h(\underline{x}, \underline{p}) = \begin{cases} h_1 = x - x_i = 0 \\ h_2 = y - y_i = 0 \\ h_3 = z - z_i = 0 \end{cases} \tag{34}$$

In this simple case, μ is obtained from h_3: $z = z_i$ and g by $g(\underline{X}, \underline{P}) = \begin{cases} X - X_i = 0 \\ Y - Y_i = 0 \end{cases}$ where X_i and Y_i are the coordinates of the projection of m_i in the image.

- *Cases of a straight line and of plane primitives*: h is then two-dimensional ($h = (h_1 \ h_2)^T$). We may for example choose the equation of the plane in which the primitive lies as a function h_2 (in the line case, h_2 is not unique). μ is then obtained from $h_2'(\underline{X}, z, \underline{p}) = 0$. With (32), $h_1'(\underline{X}, z, \underline{p}) = 0$ gives $\tilde{h}(\underline{X}, \underline{p}) = 0$, with dim $\tilde{h} = 1$, which, after change of parametrization, may be written $g(\underline{X}, \underline{P}) = 0$.

The case of tri-dimensional primitives is treated in [3], [1].

Knowing that the rigidity assumption implies $\dot{g} = 0$, $\forall \underline{X} \in P$, we may now compute the interaction matrix L^T associated to $\underline{P}$. Differentiation of (33) gives:

$$\frac{\partial g}{\partial \underline{P}}(\underline{X}, \underline{P}) \, \dot{\underline{P}} = -\frac{\partial g}{\partial \underline{X}}(\underline{X}, \underline{P}) \, \dot{\underline{X}} \ , \ \forall \underline{X} \in P \tag{35}$$

Differentiating (25) leads to the wellknown optic flow equations, which may be written:

$$\dot{\underline{X}} = L_{of}^T(\underline{X}, z) \, V_c \tag{36}$$

where V_c is the velocity of the camera with respect to the scene, and with:

$$L_{of}^T = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \tag{37}$$

Using (32) in (37), gives:

$$L_{of}^T(\underline{X}, z) = L_{of}'^T(\underline{X}, \underline{p}) \tag{38}$$

Finally, (35), (36) and (38) lead to:

$$\frac{\partial g}{\partial \underline{P}}(\underline{X}, \underline{P})\, \dot{\underline{P}} = -\frac{\partial g}{\partial \underline{X}}(\underline{X}, \underline{P})\, L_{of}'^T(\underline{X}, \underline{p})\, T \ , \ \forall \underline{X} \in P \tag{39}$$

This equation may be solved, either by explicitly using (33) in (39) and identifying, or by choosing m points $\bar{X}_i$ of Q and solving (see [1]).

4.2 Two examples of usual features

4.2.1 Points

Consider a point m_i of E with coordinates $\underline{x}_i$; then, $\underline{p} = (x_i\, y_i\, z_i)$ and $\underline{P} = (X_i\, Y_i)$. The matrix forms L_{X_i} and L_{Y_i} of the two interaction screws H_{X_i} and H_{Y_i} , expressed in F and in O, are given by (37) and may be written:

$$\begin{array}{l} L_{X_i}^T = [\quad -1/z_i \quad 0 \quad X_i/z_i \quad X_iY_i \quad -(1+X_i^2) \quad Y_i \quad] \\ L_{Y_i}^T = [\quad 0 \quad -1/z_i \quad Y_i/z_i \quad 1+Y_i^2 \quad -X_iY_i \quad -X_i \quad] \end{array} \tag{40}$$

Various sensor signals may be generated from image points, as described in [3].

4.2.2 Straight lines

A straight line in E may be represented as the cross-section of two planes:

$$h(\underline{x}, \underline{p}) = \begin{cases} a_1 x + b_1 y + c_1 z + d_1 = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \tag{41}$$

If we exclude the degenerated case where the projection centre belongs to the straight line ($d_1 = d_2 = 0$), the equation of the projected line in the image plane is:

$$AX + BY + C = 0 \text{ with } \begin{cases} A = a_1 d_2 - a_2 d_1 \\ B = b_1 d_2 - b_2 d_1 \\ C = c_1 d_2 - c_2 d_1 \end{cases} \tag{42}$$

Since the parametrization (A, B, C) of 2D straight lines is not minimal, another one should be preferred. The most used, $\underline{P} = (a, b)$, is unadequate because two charts non compatible ($Y = aX + b$, $X = aY + b$) have to be used. We therefore choose $\underline{P} = (\rho, \theta)$ and the equation of a straight line $\mathcal{D}$ is then:

$$g(\underline{X}, \underline{P}) = X\cos\theta + Y\sin\theta - \rho = 0 \tag{43}$$

- **Remark:** The ambiguity on the representation (ρ, θ) may be easily overcome.

It may then be shown that the related interaction screws are:

$$L_\theta^T = [\quad \lambda_\theta \cos\theta \quad \lambda_\theta \sin\theta \quad -\lambda_\theta\rho \quad -\rho\cos\theta \quad -\rho\sin\theta \quad -1 \quad]$$
$$L_\rho^T = [\quad \lambda_\rho \cos\theta \quad \lambda_\rho \sin\theta \quad -\lambda_\rho\rho \quad (1+\rho^2)\sin\theta \quad -(1+\rho^2)\cos\theta \quad 0 \quad] \tag{44}$$

$$\text{with} \quad \lambda_\theta = (a_2b_1 - a_1b_2)/\sqrt{A^2 + B^2}$$
$$\text{and} \quad \lambda_\rho = [(c_2a_1 - c_1a_2)\cos\theta + (c_2b_1 - c_1b_2)\sin\theta]/\sqrt{A^2 + B^2}$$

Other more complex cases (circles, spheres...) are examined in [1].

4.3 Vision-based Task Design

4.3.1 On the Model of the Interaction Matrix

It has been seen that, when data provided by a mobile camera are used as sensor signals, the associated interaction matrices L^T take the form $L^T(\underline{p}, \underline{P})$ where $\underline{P} = \underline{P}(\bar{r}, t)$ may be measured in the image, and where $\underline{p} = \underline{p}(\bar{r}, t)$ represents the 3D information coming from the considered primitives. Since this last information is a-priori unknown, it is necessary to choose a model $\hat{L}$ of L. The task vector (21) has therefore to be derived by choosing $C = W\hat{L}$ or $C = W\hat{L}^{T^\circ}$. Properties (14) and (16) will thus be satisfied if we may ensure that, respectively, $R\left(\hat{L}\right) = R(L)$ and $CL^TW^T > 0$. Recall that the interest of satisfying these properties lies in the possibility of simply choosing the identity matrix as a model of $\frac{\partial e}{\partial \bar{r}}$. Several possibilities exist:

- $\hat{L} = L(\hat{\underline{p}}, \underline{P})$ when $\underline{p}$ may be concurrently estimated.

- $\hat{L} = L(\underline{p}^*, \underline{P})$ where $\underline{p}^*$ is the value of $\underline{p}$ at $s = s^*$ which realizes $e_1 = 0$. Some further assumptions are then needed. We will come back to this point later.

- $\hat{L} = L(\hat{\underline{p}}^*, \underline{P})$ where $\hat{\underline{p}}^*$ is an estimate of $\underline{p}^*$ when no 3D information is available.

The above choices need the matrix C to be updated at the same rate as the control loop. This may be difficult, for example when C is chosen equal to $W\hat{L}^{T^\circ}$, because of the computing time required by the computation of generalized inverses. It is also sometimes necessary to anticipate the possible crossing of an isolated singularity. A simpler solution consists in using a *constant* model $\hat{L}$, determined within the task design step. It may then be chosen $\hat{L} = L(\underline{p}^*, \underline{P}^*)$, also denoted as $L_{|s=s^*}$, which is the value of the interaction matrix at the position corresponding to the selected feature, $s = s^*$.

The positivity condition (22) is then often only satisfied in the neighbourhood of the desired position $s = s^*$, whatever the choice of C. Fortunately, it should be emphasized

that this condition is only *sufficient*, and, in practical experiments, the convergence of the control law was always obtained even from initial conditions far away from the goal position. This choice of $\hat{L}$ requires the knowledge of $\underline{p}^*$, which is equivalent to making assumptions about the geometry of the 3D scene. Such assumptions may often be done when the task is being defined, and seem then not too strong: for example, if the task consists in positioning the camera in front of a door, it may be assumed that there is a door in the scene, and that characteristic signals of this door (for example its 4 corners) may be extracted. In addition, if it is desired to place the camera at a given range of the door, its dimensions should be known in order to determine the goal feature in the image.

However, in some cases, the form assumption is the only condition required for the image feature determination (in the previous case this means that the obtained range would be indifferent to the user). Then, if $L^T(\underline{p}^*, \underline{P}^*)$, where the value of $\underline{p}^*$ is unknown, may be written as $L^T(\underline{p}^*, \underline{P}^*) = B^T(\underline{P}^*) \, D(\underline{p}^*, \underline{P}^*)$ where D is a $n \times n$ positive matrix, and where B and L are of same rank, we may choose $\hat{L} = B(\underline{P}^*)$ when the positivity condition is satisfied around s^*, for example when D is diagonal or when B is of full rank.

Finally, when the previous conditions are not fulfilled, or if no scene knowledge is available (for example when it is wished to track an unknown object with a goal image feature extracted from an initial image of the object), it may be chosen $\hat{L} = L(\hat{\underline{p}}^*, \underline{P}^*)$ where $\hat{\underline{p}}^*$ is an estimate of $\underline{p}^*$, not necessarily very accurate, but ensuring the asymptotic stability of the controlled system in some neighbourhood of $s = s^*$. Ensuring the positivity condition to be satisfied is then difficult, even when $s = s^*$, since the value $L_{|s=s^*}$ remains unknown.

In the experiments we have conducted, form and dimensional assumptions were done. Therefore, $\hat{L} = L_{|s=s^*}$. Furthermore, the matrix C was always chosen equal to $W\hat{L}^{T^s}$, where W is such that $R\left(W^T\right) = R(\hat{L})$, because of better obtained decoupling properties than with the choice $C = W\hat{L}$.

5 Results and Concluding Remarks

Several examples, obtained in simulation or with an experimental testbed, are reported in [3] and [1]. We only give here a simple illustration of the proposed approach.

Let us consider a task aimed to position a camera with respect to a 'road', which is symbolized by three parallel straight lines in a plane (lateral and central white bands). The goal position is such that the camera lies at a height y^* at the middle of the right lane and that the camera axis $\vec{z}$ coincides with its direction and its axis $\vec{y}$ is vertical.

By using equations (41) and (43), fonctions $h(\underline{x}, \underline{p})$ and $g(\underline{X}, \underline{P})$ associated to the three

lines are immediatly obtained:

$$h_1(\underline{x}, \underline{p}^*) : \begin{cases} y + y^* = 0 \\ x + l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_1^* = \arctan(-l/4y^*) \\ \rho_1^* = 0 \end{cases} \tag{45}$$

$$h_2(\underline{x}, \underline{p}^*) : \begin{cases} y + y^* = 0 \\ x - l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_2^* = \arctan(l/4y^*) \\ \rho_2^* = 0 \end{cases} \tag{46}$$

$$h_3(\underline{x}, \underline{p}^*) : \begin{cases} y + y^* = 0 \\ x - 3l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_3^* = \arctan(3l/4y^*) \\ \rho_3^* = 0 \end{cases} \tag{47}$$

The sensor signals to be selected for describing this task are the parameters which represent the three lines: $s = (\theta_1, \rho_1, \theta_2, \rho_2, \theta_3, \rho_3)$. Therefore: $s^* = (\theta_1^*, 0, \theta_2^*, 0, \theta_3^*, 0)$. Furthermore, the interaction matrix associated with s^* may be easily derived from (44):

$$L^T_{|s=s^*} = \begin{pmatrix} -\cos^2\theta_1^*/y^* & -\cos\theta_1^*\sin\theta_1^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \sin\theta_1^* & -\cos\theta_1^* & 0 \\ -\cos^2\theta_2^*/y^* & -\cos\theta_2^*\sin\theta_2^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \sin\theta_2^* & -\cos\theta_2^* & 0 \\ -\cos^2\theta_3^*/y^* & -\cos\theta_3^*\sin\theta_3^*/y^* & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \sin\theta_3^* & -\cos\theta_3^* & 0 \end{pmatrix} \tag{48}$$

$L^T_{|s=s^*}$ is always of rank 5, and Ker $L^T_{|s=s^*} = (0\ 0\ 1\ 0\ 0\ 0)^T$.

Let us now apply the previous approach to the derivation of e. The following 5×6 matrix may be chosen as a matrix W:

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{49}$$

The combination matrix C is chosen equal to $W L^{Ts}_{|s=s^*}$ and, by using (21), the following task vector e is obtained:

$$e = W^+ W L^{Ts}_{|s=s^*}(s(\bar{r}, t) - s^*) + \alpha \left(\mathbf{I}_6 - W^+ W\right) g_s^{rT} \tag{50}$$

The secondary task may consist in specifying a time trajectory along $\vec{z}$, for example a constant velocity v. The associated secondary cost to be minimized is $h_s = \frac{1}{2}(z(t) - z_0 - vt)^2$ with $z(0) = z_0$. Therefore $g_s^r = (0\ \ 0\ \ (z(t) - z_0 - vt)\ \ 0\ \ 0\ \ 0)$. Note that tasks e_1 and $e_2 = z(t) - z_0 - vt$ are then compatible and independent since:

$$e = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} L^{Ts}_{|s=s^*} (s(\bar{r}, t) - s^*) + \alpha \begin{pmatrix} 0 \\ 0 \\ z(t) - z_0 - vt \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{51}$$

Figure 3 gives an example of the obtained behaviour. Left and right top windows show respectively initial and final positions of the camera (symbolized by a pyramide) with respect to the target. Middle windows represent the associated images. On bottom windows, the time variation of $\|s(\bar{r}, t) - s^*\|$ and of the components of T_c are respectively plotted. Finally, Figure 4 presents a sequence of real images and the obtained plots corresponding to the positioning of a camera handled by a six jointed robot with respect to a 4-point target.
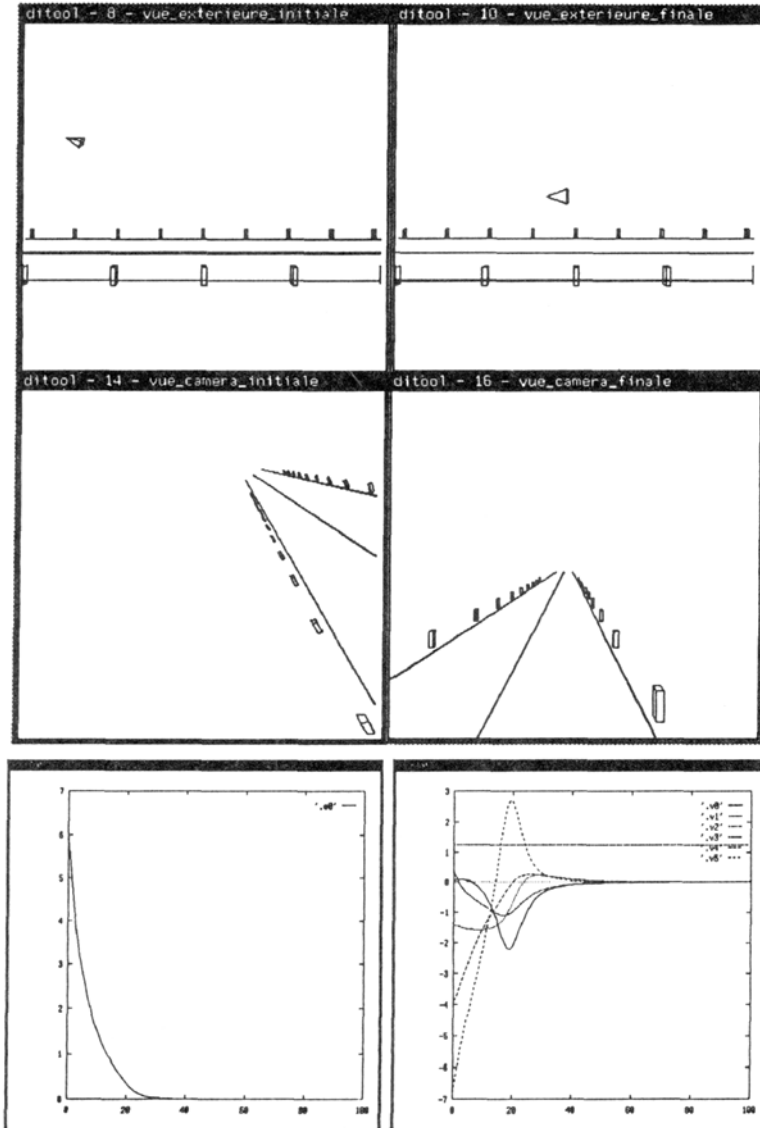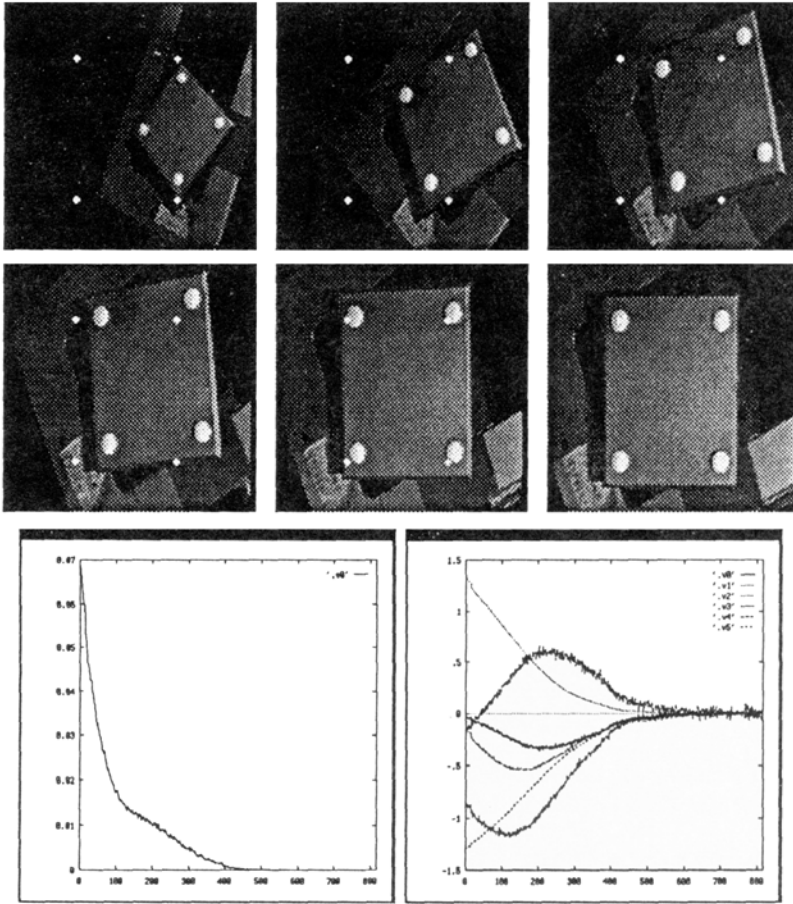

Figure 3: Road following

Figure 4: Positioning task

One possible development of this work lies in the use of an adaptive approach of the control scheme. Indeed, we may consider that intrinsic system parameters (inertia, kinematics, camera parameters...) which are not liable to large variations, may be computed or estimated off-line. On the contrary, uncertainties on the environment, which have a strong influence on the control behaviour, have to be considered carefully. In the present case, an example of uncertainty is the own object velocity, which contributes to the term $\frac{\widehat{\partial e}}{\partial t}$. Setting it equal to zero, as done here, may lead to tracking errors all the more large that the gains are small. Since the system is very sensitive to this parameter, an on-line estimation of the task vector velocity or of the object velocity within an indirect adaptive control approach would be useful.

Indeed, let us consider the vision-based-task function (with a constant combination matrix):

$$e(q,t) = C\left(s(q,t) - s^*\right) \tag{52}$$

Let $\mathcal{T}$ the object velocity, we then have:

$$\dot{e} = CL^T(q,t)\left(J(q)\,\dot{q} - \mathcal{T}\right) \tag{53}$$

and

$$\ddot{e} = CL^T(q,t)\,J(q)\,\ddot{q} + f(q,\dot{q},\mathcal{T},\dot{\mathcal{T}},t) \tag{54}$$

with

$$f(q,\dot{q},\mathcal{T},\dot{\mathcal{T}},t) = CL^T(q,t)\left(\begin{bmatrix} \vdots \\ \dot{q}^T \frac{\partial J_i^T}{\partial q}\,\dot{q} \\ \vdots \end{bmatrix} - \dot{\mathcal{T}}\right) + C\begin{bmatrix} \vdots \\ (J(q)\dot{q} - \mathcal{T})^T \frac{\partial L_i}{\partial \bar{r}}\,(J(q)\dot{q} - \mathcal{T}) \\ \vdots \end{bmatrix}$$

where J_i and $L_i(i = 1,\cdots,n)$ are the i-th row of $J(q)$ and $L(q,t)$.

We can use the control scheme (11) with a model $\hat{L}$ of the interaction matrix (see section 4.3.1) and with:

$$\widehat{\frac{\partial e}{\partial q}} = C\hat{L}^T J(q) \quad , \quad \widehat{\frac{\partial e}{\partial t}} = -C\hat{L}^T \hat{\mathcal{T}} \tag{55}$$

and

$$\hat{f} = C\hat{L}^T\left(\begin{bmatrix} \vdots \\ \dot{q}^T \frac{\partial J_i^T}{\partial q}\,\dot{q} \\ \vdots \end{bmatrix} - \hat{\dot{\mathcal{T}}}\right) + C\begin{bmatrix} \vdots \\ (J(q)\dot{q} - \hat{\mathcal{T}})^T \frac{\partial \hat{L}_i}{\partial \bar{r}}\,(J(q)\dot{q} - \hat{\mathcal{T}}) \\ \vdots \end{bmatrix} \tag{56}$$

where $\hat{\mathcal{T}}$ and $\hat{\dot{\mathcal{T}}}$ which appears in $\hat{f}$ may be obtain by estimation algorithms. A simple case, where $\mathcal{T}$ is assumed constant, is treated in [1].

References

[1] F. Chaumette: *La relation vision-commande: théorie et application à des tâches robotiques*, Thèse de l'Université de Rennes I, Juillet 1990.

[2] P. I. Corke, R. P. Paul: *Video-Rate Visual Servoing for Robots*, First International Symposium on Experimental Robotics, Montreal, Canada, June 1989.

[3] B. Espiau, F. Chaumette, P.Rives: *Une nouvelle approche de la relation vision-commande en robotique*, INRIA Research report no 1172, March 1990; also submitted to IEEE Trans. on Robotics and Automation.

[4] B. Espiau, J.P. Merlet, C. Samson: *Force Feedback Control and Non-contact Sensing: a Unified Approach*, 8th CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators, 2-6 july 1990, Cracow-Poland.

[5] J. T. Feddema, C. S. G. Lee and O. R. Mitchell: *Automatic selection of image features for visual servoing of a robot manipulator*, Conf. IEEE Robotics and Automation, Scottsdale, Arizona, USA, May 14-19, 1989.

[6] J. T. Feddema, O. R. Mitchell: *Vision-Guided Servoing with Feature-Based Trajectory Generation*, IEEE Transaction on Robotics and Automation, Vol. 5, n.5, October 1989.

[7] M. Kabuka, E. McVey, P. Shironoshita: *An Adaptive Approach to Video Tracking*, IEEE Journal of Robotics and Automation, 4(2):228-236, April 1988.

[8] C. Samson: *Une approche pour la synthèse et l'analyse de la commande des robots manipulateurs*, Rapport de recherche INRIA, n.669, Mai 1987.

[9] C. Samson, B. Espiau, M. Le Borgne: *Robot Redundancy: an Automatic Control Approach*, NATO Advanced Research Workshop on Robots with Redundancy, Salo, Italia, Juin 1988.

[10] C. Samson, B. Espiau: *Application of the Task Function Approach to Sensor-Based-Control of Robot Manipulators*, IFAC, Tallin, USSR, July 1990.

[11] C. Samson, B. Espiau, M. Le Borgne: *Robot Control: the Task Function Approach*, Oxford University Press, 1990.

[12] A. C. Sanderson, L. E. Weiss: *Adaptive Visual Servo Control of Robots*, Reprinted in *Robot Vision*, A. Pugh, Ed. Bedford, UK:IFS Pub. Ltd., 1983.

[13] L. E. Weiss: *Dynamic Visual Servo Control of Robots. An Adaptive Image based Approach*, Technical Report, CMU-RI-TR-84-16; Carnegie Mellon, 1984.

[14] L. E. Weiss, A. C. Sanderson: *Dynamic Sensor-Based Control of Robots with Visual Feedback*, IEEE Journal of Robotics and Automation, Vol. RA-3, n. 5, Oct. 1987.