A New Approach to Visual Servoing in Robotics

Bernard Espiau, *Member, IEEE*, François Chaumette, and Patrick Rives

Abstract—**This paper describes a new approach to vision-based control in robotics. The basic idea consists of considering a vision system as a specific sensor dedicated to a task and included in a control servo loop. Once the necessary modeling stage is performed, the framework becomes one of automatic control, and naturally stability and robustness questions arise. The paper is organized as follows: in the introduction, state-of-the-art visual servoing is reviewed. Then the basic concepts for modeling the concerned interactions are given. The *interaction screw* is thus defined in a general way, and the application to images follows. Starting from the concept of task function, the general framework of the control is then described, and stability results are recalled. The concept of the hybrid task is also presented and then applied to visual sensors. The paper ends with the presentation of several simulation and experimental results, and some guidelines for future work are drawn in the conclusion.**

I. Introduction

HOW to use vision has always been a major research area in robotics. Early studies in this field, in the 1970's, were mainly motivated by problems of pattern recognition. The obtained results together with the huge improvement in available computing power lead us to consider that the recognition, the localization, or the inspection of a motionless part are now in today's state of the art. In parallel, and partly owing to the attention paid to mobile robots, researches have investigated more complex questions: stereo vision, outdoor scene analysis, etc. In complement, major work was done in the domain of the analysis of image sequences (dynamic vision) [4]. The original motivations came from telecommunication (motion-compensated coding) or military (target tracking, recognition of moving objects) applications. Associated algorithms were issued from the signal processing area and rapidly permitted considering the possibility of "real-time" implementation, i.e., of working at the video rate.

If we now return to the domain of robotics, a recent trend is to use exteroceptive noncontact sensors inside the control servo loops themselves and not only as sources of data used in higher decision levels. Such an approach may prove to be very useful if it is necessary, for example, to compensate for small positioning errors, to grasp objects moving on a conveyor belt, to track a seam in arc welding, or, more generally, to be adaptable to environmental uncertainties. Applications of this kind were realized in manipulation robotics as well as in teleoperation or mobile robotics. However, the sensors were mainly optical proximeters [3], [8], [10], [12], [28], or acoustic range finders [2], [16].

In fact, it is also possible to consider a mobile vision sensor as a device able to provide useful information for realizing closed-loop control schemes with respect to the environment. Taking into account this particular goal requires the following:

- The ability to extract from an image the information sufficient and pertinent for completion of a task (in general, a positioning task). Note that this may require the design of dedicated targets.
- The implementation of control algorithms simple enough to work at a rate compatible with the desired bandwith of the closed-loop system, but, further, with an acceptable robustness with regard to unavoidable uncertainties existing about the sensor and the environment.

This approach, known as *visual servoing*, is the central issue of this paper.

It should be emphasized that this approach differs from the ones referred to as *dynamic vision* approaches, which exploit, without controlling it, the motion of the sensor or of the objects in the scene. In the *visual servoing* approach, the first step consists in defining in the image a particular set of characteristic features that constitutes the goal to be reached. The second step will be to design a control that will ensure convergence toward the configuration corresponding to the goal image by starting from a different initial condition.

Early studies in this domain, in the 1970's, were mainly based on heuristics [1], [17]. More formalized approaches arose around 1982, setting out two kinds of problems:

- the choice and the extraction of the visual feature elements to be used in the control;
- the analysis and the synthesis of the control schemes with the point of view of automatic control theory.

For the first point, it should be noted that the characteristic parameters allowing definition of the goal image were often selected in relation to the existence of algorithms allowing their extraction within a reasonably short time interval. This is why it is not surprising that most of the work performed used low-level primitives like *contours* (points, segments [19], [11], [15]) or *regions* (surface, barycenter, inertia axis [17], [29]). The original approach of [25] should also be cited: here, the image of a polyhedron is described by a graph, the nodes of which represent the surfaces of the faces and the vertices the lengths of the edges.

For the second point, the most relevant results in the literature come from Carnegie–Mellon University [26], [29], [30]. Two kinds of studies have been conducted: the analysis of the mapping between the screw space of the camera motion and the space of velocity fields in the image (which

Manuscript received March 30, 1990; revised October 17, 1991. Portions of this work were presented at the IEEE International Conference on Robotics and Automation, Sacramento, CA, May 1991.

B. Espiau is with ISIA-ENSMP, Sophia Antipolis, 06565 Valbonne, France.
F. Chaumette is with IRISA / INRIA Rennes, campus de Beaulieu, 35042 Rennes-Cedex, France.
P. Rives is with INRIA Sophia Antipolis, 06565 Valbonne, France.
IEEE Log Number 9107018.

later will be called "interaction screw") and the design of control schemes with the study of their stability. Concerning the first point, results about sensitivity with respect to the choice of image features were obtained [13], [18], [30]. Except for [13], the results concerned simplified cases: pure translational camera motion and the assumption of regularity of the mapping. It should also be noticed that no general methodology for deriving an analytical form of this mapping has been proposed (in [18], for example, the Jacobian matrix of the mapping is estimated using a perturbation method; in [13], only points are studied). In [30], the analysis of control laws is investigated with three points of view: decoupling, adaptive control, and model-based control. A stability analysis leading to the synthesis of a controller can also be found in [9] and [29].

It appears that these referenced works do not, in general, investigate in depth the choice of the visual features to be used in connection with the applications. Furthermore, a related analysis of control schemes is not inserted in an overall approach of the control problem, taking into account specific aspects of robotics, such as the effects of modeling errors or approximations.

In this paper, the problem of "visual servoing" is studied with a more general point of view under two aspects. Our first point will consist of proposing a general methodology for the design of tasks that use a visual sensor inside the control loop. This first modeling stage will be followed by an analysis of control aspects, focusing on the robustness properties of a control scheme. This analysis, partially reported in [20], is based on the original approach of the robot control problem of Samson *et al.* [24]. The objective of the control is, in the present case, the robust positioning of a mobile camera with regard to the environment, with a task directly expressed as an error with respect to a goal image.

The paper is organized as follows. Section II is devoted to modeling questions: after having derived a general model of the interaction between a sensor and a rigid environment, the special case of images is treated, and four relevant examples are examined: points, lines, circles, and spheres. In Section III, control aspects are considered. In Section III-A, the basics are recalled: task functions, redundancy, hybrid tasks, control schemes, stability issues. Section III-B presents an application of this general approach for using a camera. Finally, Section IV gives several experimental and simulation results.

II. Modeling

A. Generalities

1) Notation: The following notation will be employed:

Let us consider the tridimensional 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). It is a six-dimensional differential manifold. An element of SE_3 called a "location" (i.e., position 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 in $\mathbb{R}^3$, and an element (field) of se_3 is also known as a classical velocity screw. A screw H is also defined by its vector u and the value of its field in a point O such that for any P

$$H(P) = H(O) + u \times \vec{OP}. \tag{1}$$

We will therefore simply write $H = (H(O), u)$.

For any considered point O, the screw product is defined by

$$H_1 \bullet H_2 = < u_1, H_2(O) > + < u_2, H_1(O) > \tag{2}$$

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^*. So, S^* is itself a screw space.

2) The Interaction Screw: Let us now examine the mathematical concepts that may be associated with a sensor. We will indeed restrict our study to the class of sensors such that, formally:

Property 1: A sensor is completely defined by a differentiable mapping from SE_3 to $\mathbb{R}^p$.

This property 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 vary. Let us link a frame F_E to the part of environment observed by the sensor, and another, F_S, to the sensor itself. The sensor output s can then be written $s(F_S, F_E)$. Let us now consider that the sensor motion is obtained by using a 6-DOF physical device. Its configuration is represented by a generalized coordinate system q, which is assumed to be a local chart of SE_3 (a submanifold of SE_3 might also be considered). Then, when the observed objects are autonomously mobile themselves, s may be also written $s = s(q, t)$, the independent time variable t representing the contribution of the objects motion. The six variables q_i are, for example, the joint angular positions of a rigid manipulator that handles a camera.

Let us now examine component s_j of s. Owing to the above preliminaries and to Property 1, 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 to $\mathbb{R}$ can 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, we can therefore write at $\bar{r}$ in SE_3 the basic relation

$$\dot{s}_j = H_j \bullet T_{SE} \tag{3}$$

where T_{SE} is the velocity screw of the frame F_E 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. We can then set the following definition:

Definition 1: Under Property 1, the screw H_j defined by the expression $\dot{s}_j = H_j \bullet T_{SE}$ is called the *interaction screw.*

With an obvious breach of notation, (3) may also be written

$$\dot{s}_j = L_j^T \, T_{SE}, \quad \text{where } L_j^T = H_j \begin{bmatrix} 0 & \mathbb{I}_3 \\ \mathbb{I}_3 & 0 \end{bmatrix}. \tag{4}$$

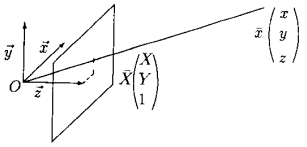


Fig. 1. A simple camera model.

L_j^T is the matrix-form of the interaction screw H_j, in a given frame F and in a chosen point O. Therefore, we have the following definition:

Definition 2: The matrix form of the set $\{H_1 \cdots H_p\}$ is called *Interaction Matrix*, and is denoted as L^T.

It will appear later that the interaction matrix plays an important role in modeling aspects as well as in the control itself.

3) The Concept of Virtual Linkage: Let T^* be a virtual motion at $\bar{r}$ keeping constant the sensor output s_j. T^* is solution of the equation

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

and is therefore a screw *reciprocal* to H_j. The set of motions T^* leaving the p components of s invariant is S^*, the subspace reciprocal to the screw subspace S spanned by the set $\{H_1 \cdots H_p\}$ (using the interaction matrix, we have $S^* = \text{Ker } L^T$).

More precisely, we have:

Definition 3: A set of *compatible* and *independent* constraints $s(\bar{r}) - s_d(t) = 0$ constitutes a *virtual linkage* between the sensor and the objects of the environment. Let m be the dimension of S. In a location where these constraints are satisfied, the dimension $N = 6 - m$ of S^* is called the *class* of the virtual linkage in $\bar{r}$ and at time t.

This concept is an immediate extension of the basic kinematics of contacts as classically used in the theory of mechanisms. The concept of virtual linkage will allow us later to design the 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.

Remark: When $m = p$, the dimension of the signal vector s is adequate, in the intuitive sense that it is indeed the number of degrees of freedom to be controlled from s. However, the case $p > m$ often offers some practical advantages, for example, because of the filtering effects it may induce. We will therefore explicitly include this case in the control design.

B. Case of an Image Sensor

1) Framework: Let us reduce a camera to a perspective projection model (Fig. 1).

The velocity screw of the camera frame $F_S(O, \vec{x}, \vec{y}, \vec{z})$, given Fig. 1, with respect to the scene is denoted as $T = (v(O), \omega)$ where $v(O)$ and ω are, respectively, the translational and rotational velocities. In the following, all variables (point coordinates, screws, etc.) will be assumed to be expressed in that frame.

$$\begin{array}{ccccccc} R \subseteq SE3 & \longrightarrow & U \subseteq \mathcal{P}_S & \longrightarrow & V \subseteq \mathcal{P}_I & & \\ (\bar{r}) & \delta & (P_S) & \pi & (P_I) & & \\ & & \downarrow \phi & & \downarrow \psi & & \\ & & \mathbb{R}^{n_S} & \longrightarrow & \mathbb{R}^{n_I} & \longrightarrow & \mathbb{R}^p \\ & & (p_S) & \psi \circ \pi \circ \phi^{-1} & (p_I) & \sigma & (s) \end{array}$$

Fig. 2. From relative location to image features.

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

$$\bar{X} = \frac{1}{z}\,\bar{x}. \tag{6}$$

Remark: It appears that this geometrical model should be accompanied with a photometric model. However, in this paper, we consider that measurements involving photometric parameters would be inadequate to constitute sensor signals as defined previously. Indeed, Property 1 is generally violated with such measurements because of their large sensitivity to the lighting parameters that may vary independently of the camera or of the objects motions. Moreover, the derivation of the interaction screw from a photometric model appears to be hazardous because of drastic uncertainties: the usual simplifying assumptions (Lambertian surfaces, diffuse lighting) are in general unrealistic, and the induced errors would then make the computation of the interaction screw irrelevant.

2) Basic Modeling Issues: We examine in this subsection the different components that form the visual features s. In particular, we set the conditions under which Property 1 is satisfied. Let us first state (see Fig. 2) the following definition:

Definition 4: A *scene feature* is a set of tridimensional geometrical primitives (points, lines, vertices, etc.) rigidly linked to a single body. A *configuration* of the scene feature is an element P_S of the n_S-dimensional set $\mathcal{P}_s$ of all possible configurations ($n_S \leq 6$).

Let π be the perspective projection mapping. Then we have the following definition:

Definition 5: An *image feature* is a set of primitives in the image plane which corresponds to the projection of a scene feature. A *configuration* of the image feature is an element P_I, with $P_I = \pi(P_S)$, of the n_I-dimensional set $\mathcal{P}_I$ of all possible configurations.

Furthermore, the group of displacements acts on $\mathcal{P}_S$ through the mapping δ such that

$$P_S = \delta(\bar{r}). \tag{7}$$

Let us now make the following assumption:

Assumption 1: Displacements are restricted to an open set R of SE_3 such that, $\forall \bar{r} \in R$, none degenerated case (where, for example, a line projects onto the image plane as a point or a circle as a segment) occurs.

Let U be an open set of $\mathcal{P}_S$ including P_S such that $U = \delta(R)$ and let V be an open set of $\mathcal{P}_I$ such that $V = \pi(U)$. We are now ready to state Assumption 2:

Assumption 2: U and V are differential manifolds.

This allows us to define two local charts [27]:

- around P_S: (U, ϕ): $p_S = \phi(P_S)$, where $p_S \in \mathbb{R}^{n_S}$ is a parametrization of P_S (for example, if P_S is a point, p_S may be chosen as its three coordinates);
- around P_I: (V, ψ): $p_I = \psi(P_I)$, where $p_I \in \mathbb{R}^{n_I}$ is a parametrization of P_I (with the same example, p_I may be chosen as the two coordinates in the image plane of the projection of the point).

Therefore

$$p_I(p_S) = \psi \circ \pi \circ \phi^{-1}(p_S) \tag{8}$$

and

$$p_I(\bar{r}) = \psi \circ \pi \circ \delta(\bar{r}). \tag{9}$$

Now, s has to be chosen such that p_I is able to generate an useful sensor signal. The associated mapping is

$$s = \sigma(p_I) \tag{10}$$

which is supposed to be such that:

Assumption 3: σ is a differentiable mapping from $\psi(V)$ to an open set Ω of $\mathbb{R}^p$.

Now, an immediate consequence of all the above issues is:

Property 2: Under Assumptions 1–3, the sensor signal s defined in (10) belongs to the class specified by property P_1 on the open set Ω of $\mathbb{R}^p$. Furthermore, we have formally

$$\frac{\partial s}{\partial \bar{r}} = \frac{\partial \sigma}{\partial p_I} \frac{\partial \psi}{\partial P_I} \frac{\partial \pi}{\partial P_S} \frac{\partial \delta}{\partial \bar{r}}. \tag{11}$$

This form is not the most adequate to an analytic computation of the interaction matrix, and the method described in the next subsection is often preferred.

3) Derivation of the Interaction Matrix: Let us specify the configuration $\mathcal{P}_{S_j}$ of the jth primitive of a scene feature by an equation of the type

$$h(\bar{x}, \bar{p}) = 0, \quad \forall\, \bar{x} \in \mathcal{P}_{S_j} \tag{12}$$

where h defines the kind of the primitive and the value of $\bar{p}$ corresponds to its configuration, and where $\forall \bar{x} \in \mathcal{P}_{S_j}$ means "for any point of the tridimensional space, with coordinates $\bar{x}$, belonging to the geometrical primitive."

Remark: If the scene feature is constituted by only one primitive and if $\bar{p}$ is chosen as a chart-representation (which is not here required), we have $\bar{p} = p_S$.

By using (6), (12) becomes

$$h(z\bar{X}, \bar{p}) = 0. \tag{13}$$

That is

$$h'(\bar{X}, z, \bar{p}) = 0. \tag{14}$$

Under the trivial condition $\partial h'/\partial z \neq 0$, which is satisfied in all the nondegenerated cases, the implicit function theorem ensures the existence of a unique function μ around a solution $\bar{x}_0$ of (12) such that

$$z = \mu(\bar{X}, \bar{p}). \tag{15}$$

Using (15) in (14) allows us to write the configuration $\mathcal{P}_{I_j}$ corresponding to the jth primitive of the image feature under the form

$$g(\bar{X}, \bar{P}) = 0, \quad \forall\, \bar{X} \in \mathcal{P}_{I_j} \tag{16}$$

where the set of parameters $\bar{P}$, obtained from $\bar{p}$, is of dimension l and where $\forall\, \bar{X} \in \mathcal{P}_{I_j}$ means "for any point of the image plane, with coordinates $\bar{X}$, belonging to the geometrical image primitive."

In practice, the sensor s is chosen as a differentiable function of $\bar{P}$: $s = f(\bar{P})$. Therefore

$$\frac{\partial s}{\partial \bar{r}} = \frac{\partial s}{\partial \bar{P}} \frac{\partial \bar{P}}{\partial \bar{p}} \frac{\partial \bar{p}}{\partial \bar{r}}. \tag{17}$$

The computation of $\partial s/\partial \bar{P}$ and $\partial \bar{P}/\partial \bar{p}$ is generally trivial. On the other hand, the term $\partial \bar{p}/\partial \bar{r}$ is not always computable. In that case, we have to use another method to directly compute $\partial \bar{P}/\partial \bar{r}$.

Remark: When s depends on the configuration of k differents primitives (as, for example, the distance between two points or the orientation between two lines), we have $s = f(\bar{P}_1, \cdots, \bar{P}_k)$ and $\partial s/\partial \bar{r}$ can be obtained in a similar way:

$$\frac{\partial s}{\partial \bar{r}} = \sum_{i=1}^{k} \frac{\partial s}{\partial \bar{P}_i} \frac{\partial \bar{P}_i}{\partial \bar{r}}. \tag{18}$$

Knowing that the rigidity assumption implies

$$\dot{g} = 0, \quad \forall\, \bar{X} \in \mathcal{P}_{I_j} \tag{19}$$

we can now compute the interaction matrix L^T associated with $\bar{P}$. Differentiation of (16) gives

$$\frac{\partial g}{\partial \bar{P}}(\bar{X}, \bar{P})\, \dot{\bar{P}} = -\frac{\partial g}{\partial \bar{X}}(\bar{X}, \bar{P})\, \dot{\bar{X}}, \quad \forall\, \bar{X} \in \mathcal{P}_{I_j}. \tag{20}$$

Differentiating (6) leads to the well known optical flow equations, which can be written

$$\dot{\bar{X}} = L_{of}^T(\bar{X}, z)\, T \tag{21}$$

where T is, as previously set, the velocity of the camera with respect to the scene, and with

$$L_{of}^T = \begin{bmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{bmatrix}. \tag{22}$$

Using (15) in (22) gives

$$L_{of}^T(\bar{X}, z) = L_{of}'^T(\bar{X}, \bar{p}). \tag{23}$$

Finally, (20), (21), and (23) lead to

$$\frac{\partial g}{\partial \bar{P}}(\bar{X}, \bar{P})\, \dot{\bar{P}} = -\frac{\partial g}{\partial \bar{X}}(\bar{X}, \bar{P})\, L_{of}'^T(\bar{X}, \bar{p})\, T\ , \ \forall \bar{X} \in \mathcal{P}_{I_j}. \tag{24}$$

This equation can be solved either by explicitly using (16) in (24) and identifying or in the following way:

The matrix L^T is of dimension $l \times 6$. By choosing l points with coordinates $\bar{X}_i$ belonging to the image primitive, we get

$$\frac{\partial g}{\partial \bar{P}}(\bar{X}_i, \bar{P}) = \alpha_i^T(\bar{P})$$
$$-\frac{\partial g}{\partial \bar{X}}(\bar{X}_i, \bar{P}) L_{of}'^T(\bar{X}_i, \bar{p}) = \beta_i^T(\bar{p}, \bar{P}). \quad (25)$$

The interaction matrix is then

$$L^T(\bar{p}, \bar{P}) = \begin{bmatrix} \alpha_1^T \\ \vdots \\ \alpha_l^T \end{bmatrix}^{-1} (\bar{P}) \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_l^T \end{bmatrix} (\bar{p}, \bar{P}). \quad (26)$$

Remark: It should be emphasized that the computation of L^T using (26) requires the nonsingularity of the matrix $(\alpha_1 \cdots \alpha_l)$. This is ensured when the dimension l of the parametrization $\bar{P}$ is minimal (i.e., the l chosen points belonging to the image primitive are independent).

4) Examples of Usual Features: Let us now examine a few basic primitives:

a) Points: Consider a point m_i with coordinates $\bar{x}_i$. Then $\bar{p} = \bar{x}_i = (x_i\, y_i\, z_i)^T$ and $\bar{P} = \bar{X}_i = (X_i\, Y_i)^T$ where X_i and Y_i are the coordinates of the projection of m_i in the image. The interaction matrices L_{X_i} and L_{Y_i} related to X_i and Y_i are given by (22) and are written

$$L_{X_i}^T = [\, -1/z_i \quad 0 \quad X_i/z_i \quad X_iY_i \quad -(1+X_i^2) \quad Y_i \,]$$
$$L_{Y_i}^T = [\, 0 \quad -1/z_i \quad Y_i/z_i \quad 1+Y_i^2 \quad -X_iY_i \quad -X_i \,]. \quad (27)$$

Various sensor signals can be generated from image points [6]: for example, length and orientation of a segment, surface and mass center of a polygon, etc.

b) Straight lines: A straight line can be represented as the intersection of two planes:

$$h(\bar{x}, \bar{p}) = \begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0. \end{cases} \quad (28)$$

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

$$AX + BY + C = 0 \quad (29)$$

with

$$A = a_1d_2 - a_2d_1$$
$$B = b_1d_2 - b_2d_1$$
$$C = c_1d_2 - c_2d_1.$$

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

$$g(\bar{X}, \bar{P}) = X\cos\theta + Y\sin\theta - \rho = 0. \quad (30)$$

Remark: The ambiguity of this representation (the same line may be parametrized by $(\rho, \theta+2k\pi)$, and $(-\rho, \theta+(2k+1)\pi)$ is partly overcome by defining a direction for the lines, i.e., fixing the sign of ρ. It will be seen that the ambiguity on the multiple possible choices of θ has no consequences on the values of the interaction screws. It can also be easily overcome as shown in Section IV-B-2.

Let us now derive the interaction screws related to this parametrization. Equation (20) can then be written

$$\dot{\rho} + (X\sin\theta - Y\cos\theta)\dot{\theta} = \cos\theta\dot{X} + \sin\theta\dot{Y} \quad \forall\, (X, Y) \in \mathcal{D}. \quad (31)$$

Furthermore, the function μ of (15) is obtained from h_1 or h_2:

$$1/z = -(a_iX + b_iY + c_i)/d_i \quad (32)$$

with $i = 1$ if $d_1 \neq 0$ or $i = 2$ if $d_2 \neq 0$.

From (30), X is expressed as a function of Y when $\cos\theta \neq 0$ (or Y as a function of X in the other case). By using (22) and (32), (31) is written

$$(-\dot{\theta}/\cos\theta)\, Y + (\dot{\rho} + \rho\tan\theta\,\dot{\theta}) = Y\; K_1T + K_2T, \quad \forall\, Y \in \mathbb{R}. \quad (33)$$

K_1 and K_2 are given at the bottom of this page.

This leads to

$$\dot{\theta} = -K_1\, \cos\theta\, T$$
$$\dot{\rho} = (K_2 + \rho\sin\theta K_1)\, T. \quad (34)$$

Therefore, we have L_θ^T and L_ρ^T, given by (35) at the bottom of this page.

c) Plane primitives: h is then two-dimensional ($h = (h_1\ h_2)^T$). Let us, for example, choose the equation of the plane in which the primitive lies for h_2. The function μ of (15) is then obtained from $h_2'(\bar{X}, z, \bar{p}) = 0$. With (15), $h_1'(\bar{X}, z, \bar{p}) = 0$ gives

$$\tilde{h}(\bar{X}, \bar{p}) = 0 \quad \text{with } \dim \tilde{h} = 1 \quad (36)$$

$$K_1 = [\, \lambda_1\cos\theta \quad \lambda_1\sin\theta \quad -\lambda_1\rho \quad \rho \quad \rho\tan\theta \quad 1/\cos\theta \,]$$
$$K_2 = [\, \lambda_2\cos\theta \quad \lambda_2\sin\theta \quad -\lambda_2\rho \quad \sin\theta \quad -\cos\theta - \rho^2/\cos\theta \quad -\rho\tan\theta \,].$$

with $\lambda_1 = (-a_i\tan\theta + b_i)/d_i$ and $\lambda_2 = (a_i\rho/\cos\theta + c_i)/d_i$.

$$L_\theta^T = [\, \lambda_\theta\cos\theta \quad \lambda_\theta\sin\theta \quad -\lambda_\theta\rho \quad -\rho\cos\theta \quad -\rho\sin\theta \quad -1 \,]$$
$$L_\rho^T = [\, \lambda_\rho\cos\theta \quad \lambda_\rho\sin\theta \quad -\lambda_\rho\rho \quad (1+\rho^2)\sin\theta \quad -(1+\rho^2)\cos\theta \quad 0 \,] \quad (35)$$

with $\lambda_\theta = (a_i\sin\theta - b_i\cos\theta)/d_i$ and $\lambda_\rho = (a_i\rho\cos\theta + b_i\rho\sin\theta + c_i)/d_i$.

which, after change of parametrization, is written $g(\bar{X}, \bar{P}) = 0$. Let us now consider an example of a plane primitive: the circle.

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

$$h(\bar{x}, \bar{p}) = \begin{cases} (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 - r^2 = 0 \\ \alpha(x-x_0) + \beta(y-y_0) + \gamma(z-z_0) = 0. \end{cases} \tag{37}$$

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

$$g(\bar{X}, \bar{P}) : X^2 + A_1 Y^2 + 2A_2 XY + 2A_3 X + 2A_4 Y + A_5 = 0 \tag{38}$$

with

$$\begin{aligned} A_1 &= \left[b^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2by_0\right]/A_0 \\ A_2 &= \left[ab(x_0^2 + y_0^2 + z_0^2 - r^2) - bx_0 - ay_0\right]/A_0 \\ A_3 &= \left[ac(x_0^2 + y_0^2 + z_0^2 - r^2) - cx_0 - az_0\right]/A_0 \\ A_4 &= \left[bc(x_0^2 + y_0^2 + z_0^2 - r^2) - cy_0 - bz_0\right]/A_0 \\ A_5 &= \left[c^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2cz_0\right]/A_0 \end{aligned}$$

where

$$A_0 = a^2(x_0^2 + y_0^2 + z_0^2 - r^2) + 1 - 2ax_0 \quad (\neq 0)$$

and

$$\begin{aligned} a &= \alpha/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ b &= \beta/(\alpha x_0 + \beta y_0 + \gamma z_0) \\ c &= \gamma/(\alpha x_0 + \beta y_0 + \gamma z_0). \end{aligned}$$

The function $\mu(\bar{X}, \bar{p})$, required for the interaction screw derivation, is easily obtained from h_2

$$1/z = aX + bY + c. \tag{39}$$

Then we can obtain (40), given at the bottom of this page (see [6] for more details)

Remark: The interaction matrix L^T related to an ellipse is always of rank 5 except when the projection of a circle is a *centered* circle in the image plane. In that case, we have

$$A_1 = 1, A_2 = A_3 = A_4 = 0, A_5 = -r^2/z_0^2 \Leftrightarrow a = b = x_0 = y_0 = 0. \tag{41}$$

L^T is here of rank 3. This singularity (loss of rank) is isolated since the rank of the matrix L^T remains 5, even in the other cases when the projection is a circle:

$$A_1 = 1, A_2 = 0 \Leftrightarrow \begin{cases} a = b = 0 \\ \text{or} \\ a = 2x_0/(x_0^2 + y_0^2 + z_0^2 - r^2) \\ b = 2y_0/(x_0^2 + y_0^2 + z_0^2 - r^2). \end{cases} \tag{42}$$

e) Tridimensional primitives: Let us now consider spheres, cylinders, torus, etc. (or even the intersection of some tridimensional primitives). In that case, it is often possible to exhibit the functions μ and g given by (15) and (16). We then consider the contour of the primitive projected on the image. The function $g(\bar{X}, \bar{P})$ is thus the limb equation, and the matching between 3-D points and contour points may be expressed as $h_0'(\bar{X}, z, \bar{p}) = 0$, with dim $h_0' = 1$, which provides function $z = \mu(\bar{X}, \bar{p})$.

f) Spheres: A sphere is an example of a tridimensional primitive with the equation

$$h(\bar{x}, \bar{p}) = (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 - r^2 = 0. \tag{43}$$

The functions $g(\bar{X}, \bar{P})$ and $\mu(\bar{X}, \bar{p})$ have to be determined as a first step. By using (6) in (43), we obtain a polynomial having $1/z$ as variable

$$\frac{1}{z^2}(x_0^2 + y_0^2 + z_0^2 - R^2) - \frac{2}{z}(x_0 X + y_0 Y + z_0) + X^2 + Y^2 + 1 = 0. \tag{44}$$

Points belonging to the image contour of a sphere are such that the intersection between their viewline and the sphere is unique. In the present case, it is equivalent to say that the discriminant Δ of (44) vanishes. Therefore, we have

$$(x_0 X + y_0 Y + z_0)^2 - (x_0^2 + y_0^2 + z_0^2 - R^2)(X^2 + Y^2 + 1) = 0 \tag{45}$$

which can be written, after a change of parametrization, as

$$g(\bar{X}, \bar{P}) : X^2 + A_1 Y^2 + 2A_2 XY + 2A_3 X + 2A_4 Y + A_5 = 0. \tag{46}$$

$$\begin{aligned} L_{A_1}^T &= \begin{bmatrix} 2bA_2 - 2aA_1 & 2A_1(b - aA_2) & 2bA_4 - 2aA_1A_3 \\ 2A_4 & 2A_1A_3 & -2A_2(A_1 + 1) \end{bmatrix} \\ L_{A_2}^T &= \begin{bmatrix} b - aA_2 & bA_2 - a(2A_2^2 - A_1) & a(A_4 - 2A_2A_3) + bA_3 \\ A_3 & 2A_2A_3 - A_4 & A_1 - 2A_2^2 - 1 \end{bmatrix} \\ L_{A_3}^T &= \begin{bmatrix} c - aA_3 & a(A_4 - 2A_2A_3) + cA_2 & cA_3 - a(2A_3^2 - A_5) \\ -A_2 & 1 + 2A_3^2 - A_5 & A_4 - 2A_2A_3 \end{bmatrix} \\ L_{A_4}^T &= \begin{bmatrix} A_3b + A_2c - 2aA_4 & A_4b + A_1c - 2aA_2A_4 & bA_5 + cA_4 - 2aA_3A_4 \\ A_5 - A_1 & 2A_3A_4 + A_2 & -2A_2A_4 - A_3 \end{bmatrix} \\ L_{A_5}^T &= \begin{bmatrix} 2cA_3 - 2aA_5 & 2cA_4 - 2aA_2A_5 & 2cA_5 - 2aA_3A_5 \\ -2A_4 & 2A_3A_5 + 2A_3 & -2A_2A_5 \end{bmatrix}. \end{aligned} \tag{40}$$

where

$$A_1 = (R^2 - x_0^2 - z_0^2)/(R^2 - y_0^2 - z_0^2)$$
$$A_2 = x_0 y_0/(R^2 - y_0^2 - z_0^2)$$
$$A_3 = x_0 z_0/(R^2 - y_0^2 - z_0^2)$$
$$A_4 = y_0 z_0/(R^2 - y_0^2 - z_0^2)$$
$$A_5 = (R^2 - x_0^2 - y_0^2)/(R^2 - y_0^2 - z_0^2).$$

The image of a sphere is thus an ellipse (a circle when $x_0 = y_0 = 0$).

The μ function is easily obtained from the double root of (44) corresponding to $\Delta = 0$ and we have

$$1/z = aX + bY + c \tag{47}$$

with

$$a = x_0/(x_0^2 + y_0^2 + z_0^2 - R^2)$$
$$b = y_0/(x_0^2 + y_0^2 + z_0^2 - R^2)$$
$$c = z_0/(x_0^2 + y_0^2 + z_0^2 - R^2).$$

The derivation of interaction screws related to an ellipse represented by $\bar{P} = (A_1, \ldots, A_5)^T$ was already done for the circle. Results are therefore given by (40) with the particular values given above for a, b, and c. The related matrix L^T is always of rank 3 for any sphere configuration. Since three parameters are sufficient to characterize the image of a sphere, we may choose the set (A_3, A_4, A_5) because the associated interaction matrix is then always of full rank, contrary to the use of A_1 or A_2.

Similar results may be obtain for other 3-D primitives. For example, cylinders are also studied in [6].

III. Control

A. General Framework

This work is embedded in the larger domain of sensor-based control. A general approach to the problem is presented in [21] and [24], and since all developments may be found in these references, we will only recall here the main characteristics, without any proof.

1) The Concept of Task Function: It is known that the dynamic behavior of a rigid manipulator is described by the model

$$\Gamma = M(q)\ddot{q} + N(q, \dot{q}, t), \quad \dim(q) = \dim(M) = n \tag{48}$$

where Γ is the vector of applied external forces (actuator torques), M is the kinetics energy matrix, and N gathers gravity, centrifugal, Coriolis, and friction forces.

Equation (48) is the *state* equation of the system, with the natural associated state $(q, \dot{q})$ (it is assumed that an actuator is associated to every degree of freedom of the robot). The task to be performed may then be specified as an *output function* associated with (48). The problem is indeed well conditioned if the passage between the "control space" and the "output space" is regular in some sense.

More precisely, we can show [24] 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 a *task function*, during a time interval $[0, t_m]$. Several cases of task functions are presented in [24]. When sensors are used, it appears that the sensor vector $s(q, t)$ has to contribute to the design of $e(q, t)$ in a way explained later.

The problem of regulating e is well conditioned if e presents some specific properties. One of them is the existence and the uniqueness of a C^2 *ideal trajectory*, $q_r(t)$, such that $e(q_r(t), t) = 0$, $\forall t \in [0, t_m]$ and $q_r(0) = q_0$, where q_0 is a given initial condition. Another very important property is the nonsingularity of the task-Jacobian matrix $(\partial e/\partial q)(q, t)$ around $q_r(t)$. When all the required conditions are satisfied, the task function is said to be "admissible," which then allows the realization of efficient control laws more or less robust according to the task admissibility.

2) Control and Stability: We only give here an intuitive idea of the approach used and of the results obtained. Let us consider the problem of exact decoupling and feedback linearization in the task space. By differentiating twice $e(q, t)$, we obtain

$$\ddot{e}(q, t) = \frac{\partial e}{\partial q}(q, t)\, \ddot{q} + l(q, \dot{q}, t) \tag{49}$$

with

$$l(q, \dot{q}, t) = \begin{bmatrix} \vdots \\ \dot{q}^T \frac{\partial E_i^T}{\partial q}(q, t)\, \dot{q} \\ \vdots \end{bmatrix} + 2\frac{\partial^2 e}{\partial q \partial t}(q, t)\, \dot{q} + \frac{\partial^2 e}{\partial^2 t}(q, t) \tag{50}$$

where $E_i (i = 1, \cdots, n)$ is the ith row of $\partial e/\partial q$.

Since e is assumed to be admissible, (48) and (49) may be combined, which leads to

$$\Gamma = M\left(\frac{\partial e}{\partial q}\right)^{-1} \ddot{e} + N - M\left(\frac{\partial e}{\partial q}\right)^{-1} l. \tag{51}$$

A control that decouples and linearizes in the task space is

$$\Gamma = M\left(\frac{\partial e}{\partial q}\right)^{-1} u + N - M\left(\frac{\partial e}{\partial q}\right)^{-1} l \tag{52}$$

where u is, for example, a proportional-derivative feedback of the form

$$u = -k\, G\, (\mu\, D\, e + \dot{e}) \tag{53}$$

where G and D are positive matrices, k and μ are positive scalars, all to be tuned by the user.

The ideal control scheme (52) and (53) 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

$$\Gamma = -\, k\, \hat{M}\left(\widehat{\frac{\partial e}{\partial q}}\right)^{-1} G\left(\mu\, D\, e + \widehat{\frac{\partial e}{\partial q}}\, \dot{q} + \widehat{\frac{\partial e}{\partial t}}\right) + \hat{N} - \hat{M}\left(\widehat{\frac{\partial e}{\partial q}}\right)^{-1} \hat{l} \tag{54}$$

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 $k, \hat{l}$, and $\hat{N}$. The control (54) includes most existing schemes: computed torque, resolved motion rate or acceleration control, indirect adaptive control, etc.

An original theorem concerning the *global* stability of the system (48) with control (54) was established in [24] in a nonlinear framework. Two main classes of sufficient stability conditions (in the sense of the boundedness of $||e(t)||$) were then exhibited:

1) *gain* conditions. These tuning parameters leave more or fewer possibilities to the user. We do not consider them here.
2) *modeling* conditions. Among them, those related to robot dynamics are not too strong in practice, owing to the symmetric-positive definiteness of the kinetics energy matrix. Another one, much more critical and less known, is related to the task, and may be expressed as

$$\frac{\partial e}{\partial q}\left(\widehat{\frac{\partial e}{\partial q}}\right)^{-1} > 0 \tag{55}$$

along the ideal trajectory (recall that a matrix $A\ (n \times n)$ is positive if $x^T Ax > 0\ ,\ \forall x \neq 0 \in \mathbb{R}^n$). We will return later to the application of this essential condition, which allows characterization of the robustness of the task itself with regard to uncertainties and approximations.

It can already be noticed that, when we are interested in the motion of the end effector, we can write $\partial e/\partial q = (\partial e/\partial \bar{r}) \cdot (\partial \bar{r}/\partial q)$, where $\partial \bar{r}/\partial q$ is the classical robot Jacobian matrix. When it is known and nonsingular, as we shall assume afterward, the choice $\widehat{\partial e/\partial q} = (\widehat{\partial e/\partial \bar{r}}) \cdot (\partial \bar{r}/\partial q)$ allows condition (55) to be reduced to

$$\frac{\partial e}{\partial \bar{r}}\left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} > 0. \tag{56}$$

3) Hybrid Tasks

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

Generally, the problem specification leads first 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 that 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 be represented at the first glance by a second $(n-m)$-dimensional vector $e_2(q,t)$.

However, e_1 and e_2 would be gathered in a single task vector $e^T = (e_1^T\ e_2^T)$ having the required properties for being admissible, in particular, the nonsingularity of $\partial e/\partial q$ along $q_r(t)$. This implies that e_1 and e_2 would be compatible and independent, which intuitively means, in terms of virtual linkage, that the secondary goal $(e_2 = 0)$ could be reached using all and only all the realizable motions left available by the virtual linkage associated to the sensor-based task.

It can 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 a priority, and e_2 is defined as the representation of the constrained minimization of a secondary cost function. Let us now recall some basic results in this domain, taken from [22] and [24].

b) The redundancy framework: Let us choose SE_3 (with generic element $\bar{r}$) as configuration space. We assume that $\partial \bar{r}/\partial q$ is nonsingular everywhere needed. Let e_1 be an m-dimensional main task, with Jacobian matrix $J_1 = \partial e_1/\partial \bar{r}$ and let h_s be a secondary cost function to be minimized (the choice of h_s is discussed in [24]).

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, Ker J_1 (or the range of J_1^T, $R(J_1^T)$) along the ideal trajectory. In other words, it has to be found any $m \times n$ full-rank matrix W such that

$$R(W^T) = R\left(J_1{}^T\right) \tag{57}$$

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

Once this matrix is determined, it is rather easy to show [22], [24] that a task function minimizing h_s under the constraint $e_1 = 0$ is

$$e = W^+ e_1 + \beta\left(\mathbb{I}_6 - W^+ W\right)\frac{\partial h_s}{\partial \bar{r}}^T \tag{58}$$

where β is a positive scalar, W^+ is the pseudoinverse of W, and $(\mathbb{I}_6 - W^+W)$ is an orthogonal projection operator on the null space of W, i.e., on that of J_1.

It clearly appears that the computation of the Jacobian matrix related to (58), possibly required in the control scheme, may be complex. The positivity condition (56) is then of some interest. It can indeed be shown that if, in addition to (57), W satisfies the property

$$J_1 W^T > 0 \tag{59}$$

along $q_r(t)$, then, under "normal circumstances" (see [24]) the Jacobian matrix of e in SE_3 is such that

$$\frac{\partial e}{\partial \bar{r}}\left(\mathbb{I}_6 + \gamma(\mathbb{I}_6 - W^+W)\right) > 0 \tag{60}$$

along $q_r(t)$, and $\forall \gamma \geq \gamma_m(\beta) \geq 0$.

The condition (56) is therefore satisfied by taking

$$\left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} = \left(\mathbb{I}_6 + \gamma(\mathbb{I}_6 - W^+W)\right). \tag{61}$$

Moreover, when β is "small enough," then $\gamma_m = 0, \partial e/\partial \bar{r}$ is positive, and we may choose

$$\widehat{\frac{\partial e}{\partial \bar{r}}} = \mathbb{I}_6. \tag{62}$$

Remark: When e_1 is made from sensor signals and h_s expresses a trajectory tracking task in SE_3, the task represented by (58) is then called a "hybrid task." The positivity property, which allows the choice (62) to be made, explains why the classical scheme known as "hybrid control" may work even if an explicit expression of the sensor signal variation is not used in the control equation.

c) The specific case of sensor signals: Let us apply this approach to the use of sensor signals. We are interested in regulating s around a desired value or trajectory $s_d(t)$. Let us recall that the vector s is of dimension p, and the Jacobian of s in SE_3 corresponds to the interaction matrix L^T. The dimension of L is $6 \times p$ and its rank m for $s = s_d(t)$, $N = 6 - m$ being the class of the associated virtual linkage (cf. Section II-A-3).

Let $C(t)$ be a combination matrix with dimension $m \times p$ such that the matrix CL^T is of full rank m along $q_r(t)$. The main task can then be written [23]

$$e_1 = C(t)\left(s(\bar{r}, t) - s_d(t)\right). \tag{63}$$

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

The Jacobian matrix of e_1 in SE_3 is

$$J_1 = \frac{\partial e_1}{\partial \bar{r}} = CL^T \tag{64}$$

and we can easily show that $R(J_1^T) = R(L)$ along $q_r(t)$.

Owing to (58), the task to be regulated is finally written

$$e = W^+C\left(s(\bar{r}, t) - s_d(t)\right) + \beta\left(\mathbb{I}_6 - W^+W\right)\frac{\partial h_s}{\partial \bar{r}}^T. \tag{65}$$

W must satisfy the property of (66), which then becomes $R(W^T) = R(L)$. This also means that the rows of W are made from basis vectors of S (evaluated in the camera frame). Interesting is the case where the subspace S, defined in Section II-A-3, is invariant in all the camera locations for which the virtual linkage constraint $s(\bar{r}, t) - s_d(t) = 0$ is satisfied. It occurs when $s_d(t)$ is constant, $\forall t$, or, more generally, when S^* (the subspace reciprocal to S which is the set of motions keeping constant $s(\bar{r}, t)$) is invariant. In that case, the knowledge of S allows us to find W constant, which satisfies (57). In other cases, for example, when a task consists in a sequence of several virtual linkages, W has only to be evaluated at each change of S.

Finally, (59) becomes

$$CL^TW^T > 0 \tag{66}$$

which allows us to choose C knowing L and W. It can be shown that (66) is, for example, satisfied by selecting $C = WL$ or $C = WL^{T^+}$. Let us note that this last choice ensures the best behavior of the control law. Indeed, if we simply consider the case $m = 6$, we obtain $\partial e/\partial \bar{r} = \mathbb{I}_6$, instead of LL^T. The choice of (62) is thus perfectly justified by setting $C = WL^{T^+}$.

B. Application to Visual Sensors

It has been seen that, when data provided by a mobile camera are used as sensor signals, the associated interaction matrices L^T depend both on parameters measured in the image, $\bar{P}$, and on 3-D information coming from the considered primitives, $\bar{p}$. Since this last information is *a priori* unknown, it is necessary to choose a model $\hat{L}$ of L. Properties (57) and (59) will thus be satisfied if we may ensure that, respectively, $R(\hat{L}) = R(L)$ 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 $\partial e/\partial \bar{r}$.

Several possibilities exist:

- $\hat{L} = L(\hat{\bar{p}}, \bar{P})$ when $\bar{p}$ is concurrently estimated.
- $\hat{L} = L(\bar{p}_d, \bar{P})$ where $\bar{p}_d$ is the value of $\bar{p}$ at $s = s_d(t)$ which realizes $e_1 = 0$. Some further assumptions are then needed. We will come back to this point later.
- $\hat{L} = L(\hat{\bar{p}}_d, \bar{P})$ where $\hat{\bar{p}}_d$ is an estimate of $\bar{p}_d$ when no 3-D information is available.

With the above choices, the matrix C needs to be updated at the same rate as the control loop. This may be not efficient, for example, when C is chosen equal to $W\hat{L}^{T^+}$, because of the computing time required by the computation of pseudoinverses. It is also sometimes necessary to anticipate the possible crossing of an isolated singularity (case of the circle in Section II-B-4). Such a crossing would lead to a loss of rank for L^T and thus to a reduction of the dimension of e_1. Taking into account these unexpected singularities might be complex, and a simpler solution consists in using a model $\hat{L}$, determined within the task design step. We may then choose $\hat{L} = L(\bar{p}_d, \bar{P}_d)$, also denoted as $L_{|s=s_d}$, which is the value of the interaction matrix at a location corresponding to the selected feature $s = s_d(t)$.

The positivity condition (66) is then often only satisfied in the neighborhood of $s = s_d(t)$, whatever the choice of C. Fortunately, it should be emphasized that this condition is only *sufficient*, and we shall see in the examples of Section IV that the convergence of the control law may be obtained even from initial conditions far away from the goal configuration.

This choice of $\hat{L}$ requires the knowledge of $\bar{p}_d$, which is equivalent to making assumptions about the *shape* and the *geometry* of the 3-D scene. Such assumptions may often be done within the task design step and then seem 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 four 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.

Finally, when 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), we may then choose $\hat{L} = L(\hat{\bar{p}}_d, \bar{P}_d)$ where $\hat{\bar{p}}_d$ is an estimate of $\bar{p}_d$, not necessarily very accurate, but ensuring the asymptotic stability of the controlled system in some neighborhood of $s = s_d(t)$. Ensuring the positivity condition to be satisfied is then difficult, even when $s = s_d(t)$, since the value $L_{|s=s_d}$ remains unknown.

IV. Experimental Results

In the cases presented, the camera is assumed to have six degrees of freedom. The goal consists in positioning the camera with respect to a target. Let us note that shape and geometrical assumptions on the different targets have been made. Therefore, $\hat{L} = L_{|s=s_d}$. Furthermore, we have only

considered the case where the vision-based task consists in regulating $s(\bar{r}, t)$ around a desired value s_d and not a trajectory $s_d(t)$. This limitation allows us to choose constant matrices W and C in (65).

Before showing the results obtained, we describe the simplified version of the control scheme used in the following. It will be pointed out that condition (56) remains essential even in this simpler case.

A. Motion Rate Control: Simplified Version of the General Control Scheme

The basic idea of control simplification consists of trying to determine that the task error approximately behaves like a first-order decoupled system, i.e.,

$$\dot{e} = -\lambda\, e. \tag{67}$$

We also have

$$\dot{e} = \frac{\partial e}{\partial \bar{r}}\, T + \frac{\partial e}{\partial t} \tag{68}$$

where the velocity screw T relies to $\dot{q}$ through

$$\dot{q} = \left(\frac{\partial \bar{r}}{\partial q}\right)^{-1} T \tag{69}$$

$\partial \bar{r}/\partial q$ being the robot Jacobian matrix. We assume that the setting variable tunable by the user is simply the desired joint velocity $\dot{q}_c$, as for most industrial robots. Then if it is possible to ensure that $\dot{q}_c \simeq \dot{q}$ (for example, by using a large gain control, or when accelerations and disturbances are not too large), the velocity setting variable in se_3, that is to say, the desired velocity screw T_c, may be considered as a pseudocontrol vector. Its general form is

$$T_c = \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} (-\lambda\, e - \widehat{\frac{\partial e}{\partial t}}). \tag{70}$$

Using (70) in (68) then gives

$$\dot{e} = -\lambda\, \frac{\partial e}{\partial \bar{r}} \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} e - \frac{\partial e}{\partial \bar{r}} \left(\widehat{\frac{\partial e}{\partial \bar{r}}}\right)^{-1} \widehat{\frac{\partial e}{\partial t}} + \frac{\partial e}{\partial t} \tag{71}$$

which would give the evolution (67) if the models were perfect. Assuming in (71) that $\partial e/\partial t = \widehat{\partial e}/\partial t = 0$ (which is not strictly necessary for the analysis [24]), we can show that condition (56) suffices to ensure that $\|e\|$ decreases.

Let us now come back to the task function given by (65). Since s_d, W, and C are chosen constant, we have

$$\frac{\partial e}{\partial t} = W^{+} C \frac{\partial s}{\partial t} + \beta\, (\mathbb{I}_6 - W^{+} W) \frac{\partial}{\partial t} \left(\frac{\partial h_s}{\partial \bar{r}}\right)^{T}. \tag{72}$$

The vector $\partial s/\partial t$ represents the contribution of a possible autonomous target motion and is often unknown. The choice made in the following is $\widehat{\partial s}/\partial t = 0$. If the target moves, this choice leads to a tracking error, the size of which decreases with λ (In [7], a simple estimation scheme of $\partial s/\partial t$ is proposed such that this tracking error progressively vanishes for a constant velocity of the target). However, if, as in trajectory tracking, $(\partial/\partial t)(\partial h_s/\partial \bar{r})$ is known, we may choose

$$\widehat{\frac{\partial e}{\partial t}} = \beta\, (\mathbb{I}_6 - W^{+} W) \frac{\partial}{\partial t} \left(\frac{\partial h_s}{\partial \bar{r}}\right)^{T}. \tag{73}$$

Finally, it is assumed that the selected interaction models allow us to satisfy condition (59) around the goal configuration and that β is small enough in the hybrid tasks of form (65) that we used. The choice $\widehat{\partial e}/\partial \bar{r} = \mathbb{I}_6$ may therefore be done. Consequently, the simple form of the set velocity screw (i.e., used as the input of a low-level large gain control) is

$$T_c = -\lambda\, e - \beta\, (\mathbb{I}_6 - W^{+} W) \frac{\partial}{\partial t} \left(\frac{\partial h_s}{\partial \bar{r}}\right)^{T}. \tag{74}$$

B. Simulation Results

In this section, simulation results related to the realization of two tasks performed from points (positioning) and lines (road following) are presented. Several other cases are presented in [6].

1) Positioning: Let us suppose that we want to set the camera with respect to a plane object that may be characterized by four points defining a square. Adequate sensor signals for this positioning task are the image coordinates of the four points: $s = (X_1, \ldots X_4, Y_1, \ldots, Y_4)$. If the goal location is such that the image plane is parallel to the object plane with the four image points forming a centered square, the image feature is $s_d = (-a, a, a, -a, a, a, -a, -a)$ where $a = l/2z_d$, l being the vertex length and z_d the final desired range. The associated interaction matrix is easily obtained through (27) and is given by (75) at the bottom of this page.

It is also possible to compute $L^{T^{+}}_{|s=s_d}$, which is given by (76) at the bottom of the following page.

The matrix $L^{T}_{|s=s_d}$ is of full rank, which allows us to choose the identity matrix $\mathbb{I}_6$ as W . The combination matrix is then

$$L^{T}_{|s=s_d} = \begin{bmatrix} -1/z_d & 0 & -a/z_d & -a^2 & -1-a^2 & a \\ -1/z_d & 0 & a/z_d & a^2 & -1-a^2 & a \\ -1/z_d & 0 & a/z_d & -a^2 & -1-a^2 & -a \\ -1/z_d & 0 & -a/z_d & a^2 & -1-a^2 & -a \\ 0 & -1/z_d & a/z_d & 1+a^2 & a^2 & a \\ 0 & -1/z_d & a/z_d & 1+a^2 & -a^2 & -a \\ 0 & -1/z_d & -a/z_d & 1+a^2 & a^2 & -a \\ 0 & -1/z_d & -a/z_d & 1+a^2 & -a^2 & a \end{bmatrix}. \tag{75}$$

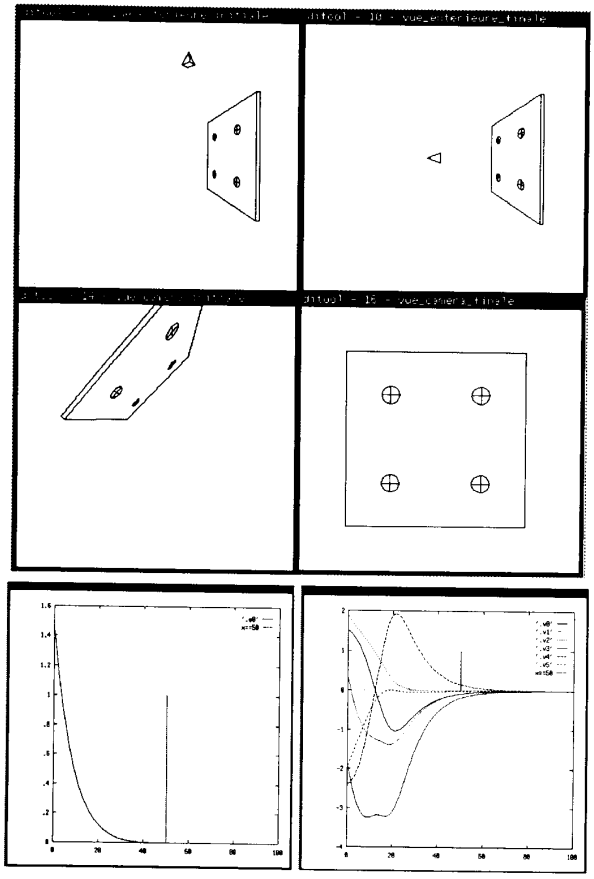


Fig. 3. Positioning with respect to a square target.

$C = L^{T^+}_{|s=s_d}$, which leads, owing to (65), to the following expression for the task function e to be regulated according to $T_c = -\lambda e$:

$$e = e_1 = L^{T^+}\big(s(\bar{r}, t) - s_d\big). \tag{77}$$

Fig. 3 gathers some simulation results related to this example. The left and right top windows show, respectively, the initial and final locations of the camera (symbolized by a pyramid) with respect to the target. The left and right middle windows represent the associated images. On the left and right bottom windows, respectively, the time variation of $\|s(\bar{r}, t) - s_d\|$ and of the components of T_c are plotted. In this example, the value of λ is set to 0.1.

The exponential decreasing of $\|s - s_d\|$ and the convergence of the control law are always ensured even for an initial location of the camera far away from the desired one. Furthermore, the vertical line plotted on the curves at iteration 50 shows the time from which the positivity condition (56) is satisfied. These results emphasize the sufficiency of the condition (the convergence is ensured although the condition is not initially respected) and the good behavior of the control law by choosing $\hat{L} = L_{|s=s_d}$ and $C = W\hat{L}^{T^+}$.

2) "Road" Following: Let us now consider a task whose aim is 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 configuration is such that:

- the camera lies at a height y_d at the middle of the right lane, and
- the camera axis $\vec{z}$ coincides with its direction and its axis $\vec{y}$ is vertical.

By using (28) and (30), the desired functions $h(\bar{x}, \bar{p}_d)$ and $g(\bar{X}, \bar{P}_d)$ associated to the three lines are immediately obtained:

$$h_1(\bar{x}, \bar{p}_d) : \begin{cases} y + y_d = 0 \\ x + l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_{1_d} = \arctan(-l/4y_d) \\ \rho_{1_d} = 0 \end{cases} \tag{78}$$

$$h_2(\bar{x}, \bar{p}_d) : \begin{cases} y + y_d = 0 \\ x - l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_{2_d} = \arctan(l/4y_d) \\ \rho_{2_d} = 0 \end{cases} \tag{79}$$

$$h_3(\bar{x}, \bar{p}_d) : \begin{cases} y + y_d = 0 \\ x - 3l/4 = 0 \end{cases} \Rightarrow \begin{cases} \theta_{3_d} = \arctan(3l/4y_d) \\ \rho_{3_d} = 0 \end{cases} \tag{80}$$

where l is the width of the road. The sensor signals to be selected for describing this task are the parameters representing the three lines: $s = (\theta_1, \rho_1, \theta_2, \rho_2, \theta_3, \rho_3)$. Therefore $s_d = (\theta_{1_d}, 0, \theta_{2_d}, 0, \theta_{3_d}, 0)$. Furthermore, the interaction matrix associated with s_d may be easily derived from (35) as shown by (81) at the bottom of the next page.

$L^T_{|s=s_d}$ is of rank 5, and S^* = Ker $L^T_{|s=s_d} = (0\ 0\ 1\ 0\ 0\ 0)^T$. As expected, the considered image feature is invariant with respect to a translational motion along $\vec{z}$ and constitutes a virtual linkage of class 1.

Remarks:

- If a mobile robot with translations along $\vec{x}$, $\vec{z}$ and rotation around $\vec{y}$ as degrees of freedom is used, the same task may be realized with a single straight line: $s = (\theta_1, \rho_1)$.
- Ambiguity of the (ρ, θ) representation, evoked in Section II-B-4, which concerns the possible choices for θ ($\theta +$

$$L^{T^+}_{|s=s_d} = \begin{bmatrix} z_d\,c_1 & z_d\,c_1 & z_d\,c_1 & z_d\,c_1 & -z_d\,c_2 & z_d\,c_2 & -z_d\,c_2 & z_d\,c_2 \\ -z_d\,c_2 & z_d\,c_2 & -z_d\,c_2 & z_d\,c_2 & z_d\,c_1 & z_d\,c_1 & z_d\,c_1 & z_d\,c_1 \\ -z_d\,c_3 & z_d\,c_3 & z_d\,c_3 & -z_d\,c_3 & z_d\,c_3 & z_d\,c_3 & -z_d\,c_3 & -z_d\,c_3 \\ -c_4 & c_4 & -c_4 & c_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_4 & -c_4 & c_4 & -c_4 \\ c_3 & c_3 & -c_3 & -c_3 & c_3 & -c_3 & -c_3 & c_3 \end{bmatrix} \tag{76}$$

where $c_1 = -1/4$, $c_2 = (1 + a^2)/4a^2$, $c_3 = 1/8a$, and $c_4 = 1/4a^2$.

$2k\pi, \forall k \in \mathbb{N}$), is overcome by modulating to 2π the error $(\theta - \theta_d)$.

Let us now apply the approach of Section III-A-3 for the derivation of e. The following 5×6 matrix may be chosen as a matrix W:

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{82}$$

The combination matrix C is chosen equal to $WL^{T^+}_{|s=s_d}$ and, by using (65), the following task vector e is obtained:

$$e = W^+ W L^{T^+}_{|s=s_d}\left(s(\bar{r},t) - s_d\right) + \beta\left(\mathbb{I}_6 - W^+W\right)\frac{\partial h_s}{\partial \bar{r}}^T. \tag{83}$$

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 $h_s = \frac{1}{2}\left(z(t) - z_0 - Vt\right)^2$ with $z(0) = z_0$. Therefore:

$$\frac{\partial h_s}{\partial \bar{r}} = \begin{pmatrix} 0 & 0 & \left(z(t) - z_0 - Vt\right) & 0 & 0 & 0 \end{pmatrix} \tag{84}$$

Note that tasks e_1 and $e_2 = z(t) - z_0 - Vt$ are then compatible and independent since

$$e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot L^{T^+}_{|s=s_d}\left(s(\bar{r},t) - s_d\right) + \beta \begin{bmatrix} 0 \\ 0 \\ z(t) - z_0 - Vt \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{85}$$

T_c is obtained from (74), in which we now have $(\partial/\partial t)(\partial h_s/\partial \bar{r}) = (0\ 0\ -V\ 0\ 0\ 0)$.

Simulation results of Fig. 4 are structured as in the previous case. Parameters are tuned with the following values: $\lambda = 0.1$, $\beta = 1$, and $V = 1.25$ cm/s.

C. Experiment with a Six-Jointed Robot

The positioning task with respect to a square target was implemented on an experimental testbed including a CCD camera mounted in the effector of a six-jointed robot.

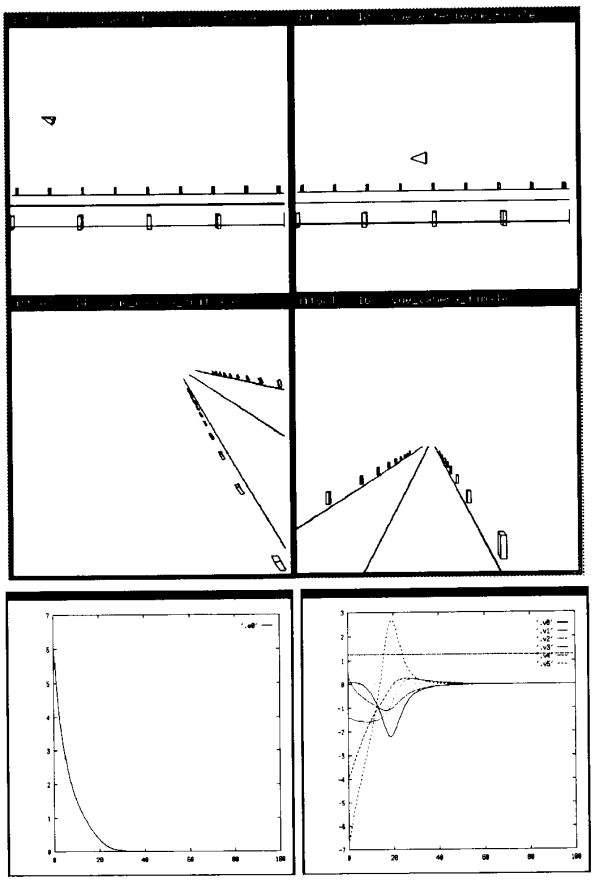


Fig. 4. Road following.

A camera calibration step allowing identification of the parameters of the model shown in Fig. 1 was done, and the transformation matrix from the camera frame to a frame linked to the end-effector was identified using methods given in [5]. The computation of the inverse robot Jacobian matrix is realized on a 68020-based dedicated board with a sampling rate of 5 ms. The useful part of the scene is a set of four white disks on a dark background. Owing to this simplicity, mass centers of image disks and velocity screw T_c are computed in less than 20 ms, which ensures that the video rate is respected. Thus, visual features of all odd frames are taken into account in the control. For more complex scenes, requiring a vision process time higher than 20 ms, an asynchronous feature data extraction could be used as illustrated in [14].

$$L^T_{|s=s_d} = \begin{bmatrix} -\cos^2\theta_{1_d}/y_d & -\cos\theta_{1_d}\sin\theta_{1_d}/y_d & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \sin\theta_{1_d} & -\cos\theta_{1_d} & 0 \\ -\cos^2\theta_{2_d}/y_d & -\cos\theta_{2_d}\sin\theta_{2_d}/y_d & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \sin\theta_{2_d} & -\cos\theta_{2_d} & 0 \\ -\cos^2\theta_{3_d}/y_d & -\cos\theta_{3_d}\sin\theta_{3_d}/y_d & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \sin\theta_{3_d} & -\cos\theta_{3_d} & 0 \end{bmatrix}. \tag{81}$$

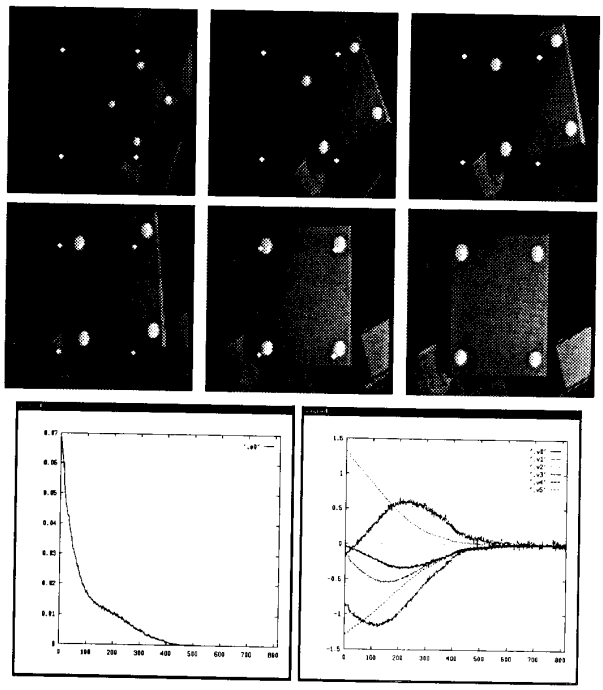


Fig. 5. An experiment of positioning in front of a square.

Fig. 5 shows a sequence of six images taken during a positioning task. The used value of λ is 0.1. The plottings of $\|s - s_d\|$ and T_c are close to the one obtained in simulation.

Let us finally point out that, when the object motion is slow enough to ensure that it keeps lying inside the camera field from one iteration to the next, a tracking error decreasing with λ is observed (cf. Section IV-A). The stability of the system, however, is not affected. This error may be reduced by increasing the gain λ (which must not be set too high in order to preserve the stability of the system) or by introducing in T_c an estimation of the object velocity.

V. Conclusion

The objective of this paper was to show that technical progress now permits control of the interaction between a robot and its environment by using data provided by a visual sensor directly inside a closed-loop control scheme. When exploiting such an approach in robotics, two questions need to be answered:

- What kinds of robotics tasks are really concerned? Furthermore, is it possible to define *generic classes of tasks* and how are they to be specified?
- For a given class of tasks, how can one design an efficient control with good properties of stability and robustness?

Concerning the first point, it should be recalled that the proposed approach contains an underlying fundamental assumption: only geometrical variations of the environment are liable to make the sensor output vary. From this, it appears that the tasks to be considered are those requiring the control of geometrical interaction between a robot and its environment. Typically, it is desirable to control the location and the attitude of a frame linked to a sensor with regard to a frame linked to the environment. A mechanics-based formalism may then be used, as in assembly applications. Tasks may be specified as basic *linkages* (point on plane, line on plane, etc.) that represent in a simple way the desired constraints and the available degrees of freedom. This formalism extended to noncontact sensor-based tasks allows definition of elementary task classes (positioning, line following, etc.) to be combined in more complex applications. In the *visual servoing* approach, we have established for several simple scene primitives (point, line, circle, sphere, etc.) the structure and properties of interaction screws, thereby allowing us to associate a linkage under the above formalism with a scene feature. Because of the nature of a visual sensor, the main difficulty encountered comes from the infinite number of possibilities that exist for relating a given type of linkage to an image feature. Preferring one feature over another depends, therefore, on other criteria related both to image processing aspects (robustness or implementation ease of feature extraction algorithms) and to control requirements (sensitivity, decoupling, etc.). Much work remains to be done in this area.

The second point, which concerns the implementation of robust control schemes, has already been deeply investigated in the general framework of robot manipulator control. In the case of visual servoing, it is still necessary to take into account the possibility of tracking moving objects with unknown trajectories. Since the use of large gains is limited in practice because of sampling rates, it seems to be necessary to include in the control an algorithm estimating on-line the relevant parameters of the motion of a target.

In conclusion, let us emphasize that the implementation of visual-based servo control loops requires some coherence between the sampling frequency at the image level and the bandwith of the system to be controlled. Although the features used in the proposed approach are low level and no semantical interpretation is needed, it still seems necessary to use dedicated real-time architectures for the primitive extraction step. Expected technical progress, however, will surely enable the application fields of this approach to be extended in the next years.

References

[1] G. J. Agin, "Real time control of a robot with a mobile camera," SRI International, Tech. Note 179, Feb. 1979.

[2] G. André and R. Fournier, "Generalized end effector control in a computer aided teleoperation system with application to motion coordination of a manipulator arm on a oscillating carrier," in *Proc. Int. Conf. Advanced Robotics* (Tokyo), Sept. 1985, pp. 337–344.

[3] D. J. Balek and R. B. Kelley, "Using gripper mounted infrared proximity sensors for robot feedback control," in *Proc. IEEE Int. Conf. Robotics Automat.* (Saint Louis, MO), Mar. 1985, pp. 282–287.

[4] P. Bouthemy, "A maximum likelihood framework for determining moving edges," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 5, pp. 499–511, May 1989.

[5] F. Chaumette and P. Rives, "Réalisation et calibration d'un système expérimental de vision composé d'une caméra embarquée sur un robot manipulateur," INRIA Res. Rep., no. 994, Mar. 1989.

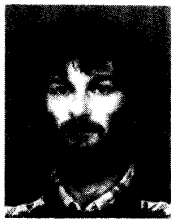
[6] F. Chaumette, "La relation vision-commande: Théorie et applications à des tâches robotiques," Ph.D. dissertation, Rennes I University, Rennes, France, July 1990.

[7] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *Proc. IEEE Int. Conf. Robotics Automat.* (Sacramento, CA, Apr. 1991), vol. 3, pp. 2248–2253.

[8] E. Cheung and V. Lumelsky, "Proximity sensing in robot manipulator motion planning: System and implementation issues," *IEEE Trans. Robotics Automat.*, vol. 5, no. 6, pp. 740–751, Dec. 1989.
[9] P. I. Corke and R. P. Paul, "Video-rate visual servoing for robots", in *Proc. 1st Int. Symp. Experimental Robotics* (Montreal, Canada), June 1989.
[10] B. Espiau and R. Boulic, "Collision avoidance for redundant robots with proximity sensors," in *Proc. 3rd Int. Symp. Robotics Res.* (Gouvieux, France, Oct. 1985). Cambridge, MA: MIT Press, 1985, pp. 243–251.
[11] B. Espiau and P. Rives, "Closed-loop recursive estimation of 3D features for a mobile vision system," in *Proc. IEEE Int. Conf. on Robotics Automat.* (Raleigh, NC), vol. 3, Apr. 1987, pp. 1436–1443.
[12] B. Espiau, "Sensory-based control: Robustness issues and modeling techniques; Application to proximity sensing," *NATO Advanced Res. Workshop Kinematic and Dynamic Issues in Sensor-Based Control* (Il Ciocco, Italy), Oct. 1987, pp. 3–44.
[13] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell, "Automatic selection of image features for visual servoing of a robot manipulator," in *Proc. IEEE Int. Conf. Robotics Automat.* (Scottsdale, AZ), May 1989, vol. 2, pp. 832–837.
[14] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. Robotics Automat.*, vol. 5, no. 5, pp. 691–700, Oct. 1989.
[15] C. L. Fennema and W. B. Thomson, "Velocity determination in scenes containing several moving objects," *Computer Graphics Image Process.*, vol. 9, pp. 301–315, 1979.
[16] A. Flynn, "Combining sonar and infrared sensors for mobile robot navigation," *Int. J. Robotics Res.*, vol. 7, no. 6, pp. 5–14, Dec. 1988.
[17] A. L. Gilbert *et al.*, "A real-time video tracking system," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 1, pp. 47–56, Jan. 1980.
[18] M. Kabuka, E. McVey, and P. Shironoshita, "An adaptive approach to video tracking," *IEEE J. Robotics Automat.*, vol. 4, no. 2, pp. 228–236, Apr. 1988.
[19] P. Rives, "Dynamic vision: Theoretical capabilities and practical problems," in *Proc. NATO Workshop Kinematic and Dynamic Issues in Sensor Based Control* (Italy), Oct. 1987, pp. 251–280.
[20] P. Rives, F. Chaumette, and B. Espiau, "Visual servoing based on a task function approach," in *Proc. 1st Int. Symp. Experimental Robotics* (Montréal, Canada), June 1989.
[21] C. Samson, "Une approche pour la synthèse et l'analyse de la commande des robots manipulateurs," INRIA Res. Rep. 669, May 1987.
[22] C. Samson, B. Espiau, and M. Le Borgne, "Robot redundancy: An automatic control approach," presented at NATO Advanced Research Workshop on Robots with Redundancy, Salo, Italia, June 1988.
[23] C. Samson and B. Espiau, "Application of the task function approach to sensor-based-control of robot manipulators," in *Proc. 11th Int. Federat. Automat. Contr. World Congr. (Tallin, Estonia, U.S.S.R., Aug. 1990), vol. 9, pp. 286–291.*
[24] C. Samson, B. Espiau, and M. Le Borgne, *Robot Control: the Task Function Approach.* Oxford, England: Oxford University Press, 1991.
[25] A. C. Sanderson and L. E. Weiss, "Image based visual servo control using relational graph error signal," in *Proc. Int. Conf. Cybern. and Soc.* (Cambridge, MA), Oct. 1980, pp 1074–1077.
[26] ——, "Adaptive visual servo control of robots," in *Robot Vision*, A. Pugh, Ed. Bedford, UK: IFS Pub. Ltd., 1983, pp. 107–116.
[27] M. Spivak, *A Comprehensive Introduction to Differential Geometry*, Boston: Publish or Perish, 1970.
[28] C. Wampler, "Multiprocessor control of a telemanipulator with optical proximity sensors," *Int. J. Robotics Res.*, vol. 3, no. 1, pp. 40–50, 1984.
[29] L. E. Weiss, "Dynamic visual servo control of robots. An adaptive image based approach," Carnegie-Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-84-16, 1984.
[30] L. E. Weiss and A. C. Sanderson, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robotics Automat.*, vol. RA-3, no. 5, pp. 404–417, Oct. 1987.

Bernard Espiau (M'85) graduated from the Ecole Nationale Supérieure de Mécanique de Nantes in 1972. He received the Docteur-Ingénieur in Automatic Control and the Docteur d'Etat in Applied Mathematics degrees in 1975 and 1981, respectively.

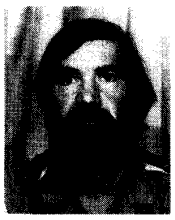
As a Research Director, he led a robotics project in the Rennes Laboratory of INRIA up to 1988. He is presently Director of the Institut Supérieur d'Informatique et Automatique, Sophia-Antipolis, Valbonne, France, and Associated Professor at the Ecole des Mines de Paris. His research interests are robot control, sensor-based control, and real-time programming issues. With C. Samson and M. Le Borgne, he is a coauthor of the book *Robot Control: The Task-function Approach* (Oxford, England: Clarendon Press, 1991).



François Chaumette graduated from the Ecole Nationale Supérieure de Mécanique de Nantes in 1987 and received a Doctorate degree in computer science from Rennes University in 1990.

He conducted his dissertation research at the laboratories of IRISA / INRIA Rennes, Rennes, France, where he is now employed as a researcher. His research interests include robotics, computer vision, vision-based control, and active dynamic vision.

Dr. Chaumette received in 1990 the AFCET / CNRS Prize of the best French dissertation in automatic control.



Patrick Rives received the Doctorat de 3ième cycle degree in robotics from the Université des Sciences et Techniques du Languedoc, Montpellier, France, in 1981 and the "Habilitation à diriger les recherches" degree in 1991.

After spending one year as a Research Fellow at the INRS Laboratory in Montreal, Canada, he joined, in 1982, the Image Processing Group at the INRIA Laboratory in Rennes. He is presently in the Robotics Team at the INRIA Laboratory of Sophia Antipolis, Valbonne, France. His research interests include image processing, sensor-based control, and reactive approaches applied to mobile robots.