A Redundancy-Based Iterative Approach for Avoiding Joint Limits: Application to Visual Servoing

François Chaumette and Éric Marchand

***Abstract*—We propose in this paper new redundancy-based solutions to avoid robot joint limits of a manipulator. We use a control scheme based on the task function approach. We first recall the classical gradient projection approach and we then present a far more efficient method that relies on the iterative computation of motion that does not affect the task achievement and ensures the avoidance problem. We apply this new method in a visual servoing application. We demonstrate the validity of the approach on various real experiments as well as on the control of a virtual humanoid.**

***Index Terms*—Gradient projection approaches, iterative approach, joint limits avoidance, visual servoing.**

I. Introduction

WITHIN a reactive context, planning a robot trajectory is not always possible. If the control law computes a motion that exceeds the robot joint limits, the specified task will not be achieved. Control laws taking into account the region of space located in the vicinity of these joint limits have thus to be considered.

In order to avoid joint limits, Chang and Dubey [2] have proposed a method based on a weighted least norm solution for a redundant robot. This method does not try to maximize the distance of the joints from their limits but it dampens any motion in their direction. Thus, it avoids unnecessary self-motion and oscillations. Another approach has been used by Nelson and Khosla [3] and applied to visual servoing. It consists of minimizing an objective function which realizes a compromise between the main task and the avoidance of joint limits. During the execution of the task, the manipulator moves away from its joint limits and singularities. However, such motions can produce important perturbations in the visual servoing since they are generally not compatible with the specified task. Another approach, known as Gradient Projection Method (e.g., [4], [5]), uses robot redundancy and has been widely used to solve joint limits problems. It relies on the evaluation of a cost function seen as a performance criterion function of the joints position. The gradient of this function, projected onto the null space of the main task Jacobian, is used to produce the motion necessary to minimize the specified cost function as far as possible. The main advantage of this method wrt. [2], [3] is that, thanks to the choice of adequate projection operator, the joint limits avoidance process has **no** effect on the main task: avoidance is performed under the constraint that the main task is realized.

In this paper, we first recall how the gradient projection approach can be used to avoid joint limits [5], [6]. Unfortunately, it appears that the success of this method relies on a parameter (the amplitude of the secondary task wrt. the main task) that has to be precisely tuned in order to ensure the joint avoidance process. We show that, if badly chosen, the task may fail. We therefore propose an original and far more efficient solution to the joint limits avoidance problem. It consists in generating automatically robot motions compatible with the main task by iteratively solving a system of linear equations. The advantage of this method is that it ensures to stop any motion that moves the robot in the neighborhood of its joint limits.

To validate our approach, we apply the proposed method to a visual servoing problem. Visual servoing [7]–[9] is a closed loop reacting to image data. As in the general case, if the control law computes a motion that exceeds a joint limit, visual servoing fails. This specific problem has been already considered in the literature [3], [6]. In a previous paper [6], we considered an extension of the Gradient Projection Method. In this paper, we apply the proposed framework to vision-based positioning and tracking tasks.

Section II of this paper recalls the approach proposed in [6] to avoid joint limits. In Section III we present the original iterative method. In Section IV we quickly present the visual servoing framework and we give, in Section V, experimental results obtained using both an eye-in-hand system composed of a camera mounted on the end-effector of a six d.o.f. robot, and using a virtual humanoid.

II. Avoiding Joint Limits Using Task Function Approach

In this section we present the classical gradient projection approach usually considered to constrain the robot motion and, in particular, its application to the joint limits avoidance.

A. Gradient Projection Approaches

A robotic task can be seen as the regulation to zero of a task function [5] defined by

$$\mathbf{e} = \mathbf{J}_1^+ \mathbf{e_1} + \alpha \mathbf{J}_\mathbf{1}^\perp \mathbf{e_2} \tag{1}$$

Manuscript received February 2001. This paper was recommended for publication by Associate Editor C. Melchiorri and Editor S. Hutchinson upon recommendation of the reviewers' comments. This paper was presented in part at the IEEE International Conferference on Robotics and Automation, ICRA'2000.

The authors are with IRISA/INRIA Rennes, Campus de Beaulieu, 35042 Rennes-Cedex, France (e-mail: chaumett@irisa.fr; marchand@irisa.fr).

Publisher Item Identifier S 1042-296X(01)09911-6.

where

- $\mathbf{e_1}$ is the main task to be achieved that induces m independent constraints on the n robot joints (with $m < n$).
- $\mathbf{e_2}$ is a secondary task.
- $\mathbf{J_1^+}$ and $\mathbf{J_1^\perp}$ are two projection operators which guarantee that the robot motion due to the secondary task is compatible with the constraints involved by $\mathbf{e_1}$. More precisely, $\mathbf{J_1} = \partial \mathbf{e_1}/\partial \mathbf{q}$ is the $m \times n$ full rank Jacobian matrix of task $\mathbf{e_1}$. $\mathbf{J_1^+}$ is the pseudo-inverse of $\mathbf{J_1}$ and $\mathbf{J_1^\perp}$ is defined by $\mathbf{J_1^\perp} = \mathbf{I_n} - \mathbf{J_1^+ J_1}$. Each column of $\mathbf{J_1^\perp}$ belongs to $\mathrm{Ker}\ \mathbf{J_1}$, which means that the realization of the secondary task will have no effect on the main task $(\mathbf{J_1 J_1^\perp e_2} = 0, \forall \mathbf{e_2})$. However, if modeling errors are introduced in $\mathbf{J_1}$ and $\widehat{\mathbf{J}}_\mathbf{1}$ is used in the definition of the task function, $\widehat{\mathbf{J_1^\perp}}\left(= \mathbf{I} - \widehat{\mathbf{J}}_\mathbf{1}^+ \widehat{\mathbf{J}}_\mathbf{1}\right)$ no more exactly belongs to $\mathrm{Ker}\ \mathbf{J_1}$, which may induce perturbations on $\mathbf{e_1}$ due to the secondary task. In practice, experimental results show that this is not an important problem. Let us finally note that, if $\mathbf{e_1}$ constrains all the n degrees of freedom of the manipulator (i.e., $m = n$), we have $\mathbf{J_1^\perp} = 0$. It is thus impossible in that case to consider any secondary task.
- α is a scalar which sets the amplitude of the control law due to the secondary task. Tuning this scalar has proved to be a non trivial issue. We will see latter on how to consider this problem efficiently.

To make $\mathbf{e}$ decrease exponentially and then behave like a first order decoupled system, we get:

$$\dot{\mathbf{q}} = -\lambda \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{2}$$

where

$\dot{\mathbf{q}}$	joint velocity given as input to the robot controller;
λ	proportional coefficient involved in the exponential decrease of $\mathbf{e}$;
$\widehat{\frac{\partial \mathbf{e}}{\partial t}}$	approximation of $\partial \mathbf{e}/\partial t$ involved to minimize potential tracking errors.

B. Joint Limits Avoidance

The most classical way to solve the joint limits avoidance problem is to define the secondary task as the gradient of a cost function h_s $\left(\mathbf{e_2} = (\partial h_s/\partial \mathbf{q})^{\mathrm{T}}\right)$. This cost function must reach its maximal value near the joint limits and its gradient must be equal to zero when the cost function reaches its minimal value [5]. Several cost functions h_s which reflect this desired behavior have been presented in [2], [5] and [6]. We briefly recall the most efficient of the cost functions proposed in [6].

Let us denote $\bar{\mathbf{q}}_{i_{\min}}$ and $\bar{\mathbf{q}}_{i_{\max}}$ the lower and upper limits that are not to be crossed. Activation thresholds on axis i are defined by $\tilde{\mathbf{q}}_{i_{\min}}$ and $\tilde{\mathbf{q}}_{i_{\max}}$ such that

$$\begin{aligned} \tilde{\mathbf{q}}_{i_{\min}} &= \bar{\mathbf{q}}_{i_{\min}} + \rho \Delta \bar{\mathbf{q}} \\ \tilde{\mathbf{q}}_{i_{\max}} &= \bar{\mathbf{q}}_{i_{\max}} - \rho \Delta \bar{\mathbf{q}} \end{aligned} \tag{3}$$

where $\Delta \bar{\mathbf{q}} = \bar{\mathbf{q}}_{i_{\max}} - \bar{\mathbf{q}}_{i_{\min}}$ and $0 < \rho < 1/2$ (typically, $\rho = 0.1$).

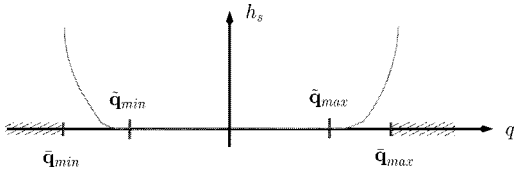


Fig. 1. Evolution of the cost function wrt. joint position.

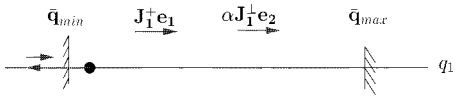


Fig. 2. Influence of the gain α on the efficiency of the gradient projection approach: if α is too small $\mathbf{e_2}$ is inefficient.

The cost function is thus given by (see Fig. 1)

$$h_s = \frac{1}{2} \sum_{i=1}^{n} \frac{s_i^2}{\Delta \bar{\mathbf{q}}} \tag{4}$$

where

$$s_i = \begin{cases} \mathbf{q}_i - \tilde{\mathbf{q}}_{i_{\max}}, & \text{if } \mathbf{q}_i > \tilde{\mathbf{q}}_{i_{\max}} \\ \mathbf{q}_i - \tilde{\mathbf{q}}_{i_{\min}}, & \text{if } \mathbf{q}_i < \tilde{\mathbf{q}}_{i_{\min}} \\ 0, & \text{else .} \end{cases} \tag{5}$$

Components of $\mathbf{e_2}$ and $\partial \mathbf{e_2}/\partial t$ take the form

$$\mathbf{e_{2}}_i = \begin{cases} \dfrac{\mathbf{q}_i - \tilde{\mathbf{q}}_{i_{\max}}}{\Delta \bar{\mathbf{q}}}, & \text{if } \mathbf{q}_i > \tilde{\mathbf{q}}_{i_{\max}} \\ \dfrac{\mathbf{q}_i - \tilde{\mathbf{q}}_{i_{\min}}}{\Delta \bar{\mathbf{q}}}, & \text{if } \mathbf{q}_i < \tilde{\mathbf{q}}_{i_{\min}} \\ 0, & \text{else.} \end{cases} \quad \frac{\partial \mathbf{e_{2}}_i}{\partial t} = 0 \tag{6}$$

This cost function is similar to the Tsai's manipulability measure used in [3]. It is, however, simpler since it directly sets the activation thresholds with ρ. Let us finally note that, in all cases, $\mathbf{e_2}$ and $\partial \mathbf{e_2}/\partial t$ are continuous, which will ensure a continuous control law.

The parameter α that sets the amplitude of the control law due to the secondary task is very important [see (1)]. Indeed, as pointed out in [2], if α is too small, $\mathbf{e_2}$ may be insufficient to avoid a joint limit (see Fig. 2). Furthermore, if α is too large, it will result in some overshoot in the effector velocity. Therefore, α is usually set based on trial and errors. We now propose a simple new solution to this important problem.

C. Tuning the Influence of the Secondary Task

The simplest solution to this problem is to select the most critical axis and to compute automatically the minimum value of α to stop any motion on this axis (see Fig. 3). More precisely we define a critical axis as an axis whose joint position is between its joint limits and its activation threshold and that moves toward its joint limits because of the effect of $\mathbf{e_1}$. We first determine the effect of the primary task $\mathbf{e_1}$. This can be done by performing a prediction step. Assuming that the robot is located in $\mathbf{q}(t)$, if we do not consider a secondary task, predicted position $\hat{\mathbf{q}}(t+1)$ is given by

$$\hat{\mathbf{q}}(t+1) = \mathbf{q}(t) + \dot{\mathbf{q}} \Delta t \tag{7}$$

$$= \mathbf{q}(t) - \lambda \mathbf{J_1^+ e_1} \Delta t. \tag{8}$$

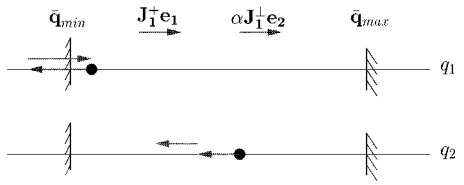


Fig. 3. Tuning the influence of the secondary task: select the most critical axis $\mathbf{q}_k$ and compute α such as $\dot{\mathbf{q}}_k = 0$.

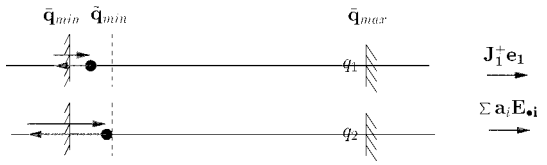


Fig. 4. New algorithm: stopping several axes in critical area remains to estimate a gain vector $\mathbf{a}$.

For all axes in the critical area that moves toward the joint limit (i.e., $\widehat{q}_i(t+1)$ is nearer from its joint limits than $q_i(t)$), we select the axis k for which $\widehat{\mathbf{q}}_k(t+1)$ is the closest from its joint limits. Then we compute α in order to stop any motion on this axis (i.e., $\widehat{\mathbf{q}}_k(t+1) - \mathbf{q}_k(t) = 0$). Using (2), the constraint $\Delta\mathbf{q}_k = 0$ is equivalent to $\mathbf{e}_k = 0$ and using (1) leads to compute α as

$$\alpha = -\frac{(\mathbf{J}_1^+\mathbf{e}_1)_k}{(\mathbf{J}_1^+\mathbf{e}_2)_k}. \tag{9}$$

The considered joint is stopped but it does not move away its joint limit. However, this method does not ensure that another axis does not move toward its joint limits (see Fig. 3). We, therefore, propose in the Section III a new redundancy-based approach to cope with these problems.

III. A New Approach: Iterative Computation of Adequate Motions

A. Requirements and Overview

As seen in Section II, a good solution to achieve the avoidance task is to cut any motion on axes that are in critical situation (i.e., between $\tilde{\mathbf{q}}$ and $\bar{\mathbf{q}}$ and getting closer $\bar{\mathbf{q}}$). Considering that $\mathbf{q}_k$ is one of these axes, we have to compute a velocity $\dot{\mathbf{q}}_k = 0$. In the previous paragraph, we considered such a condition but the result was to compute the minimum value of α (for all the axes) that ensures this task. If possible, it is more interesting to compute such a gain on each axis. As described below, the proposed approach to achieve this goal relies on the resolution of a linear system. Another drawback of the previous approach is that, because of the new computed control law, other axes may enter in the critical area. In the new framework, this can be handled by applying the same algorithm iteratively. Finally, it is always possible to consider a secondary task that moves the axes away from the critical area, as will be shown in Section III-E.

B. Basic Algorithm

A general task function that uses redundancy can be defined by[1]

$$\mathbf{e} = \mathbf{J}_1^+\mathbf{e}_1 + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i\mathbf{E}_{\bullet i} \tag{10}$$

where

- $n_\mathbf{a} = \dim \mathrm{Ker}\ \mathbf{J}_1 = n - m$.
- $\sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i\mathbf{E}_{\bullet i}$ defines motions that try to ensure that the robot will never encounter its joints limits. Within this term:
  - — $\mathbf{E}$ is a basis of $\mathrm{Ker}\ \mathbf{J}_1$ of dimension $n \times n_\mathbf{a}$. In this way, the computed motions will have no effect on the main task.
  - — $\mathbf{a}$ is a vector of gains that will be automatically computed.

Consider that several axes are in critical situation, we determine vector $\mathbf{a}$ in order to stop the motion on these axes (see Fig. 4). From (10), for each axis k in critical situation we obtain:

$$\dot{\mathbf{q}}_k = 0 \Longleftrightarrow \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i\mathbf{E}_{ki} = -\left(\mathbf{J}_1^+\mathbf{e}_1\right)_k. \tag{11}$$

If $p_\mathbf{a}$ axes are in critical situation, we can define from (11) a linear system $\mathbf{F}\mathbf{a} = \mathbf{s}$ where $\mathbf{F}$ is of dimension $p_\mathbf{a} \times n_\mathbf{a}$ while $\mathbf{s}$ and $\mathbf{a}$ are of dimension $n_\mathbf{a}$. More precisely, we have

$$\underbrace{\begin{pmatrix} \vdots \\ \mathbf{E}_{k\bullet} \\ \vdots \end{pmatrix}}_{\mathbf{F}} \mathbf{a} = \underbrace{\begin{pmatrix} \vdots \\ -(\mathbf{J}_1\mathbf{e}_1)_k \\ \vdots \end{pmatrix}}_{\mathbf{s}}$$

where k represents each axis in critical situation. We have three possible cases:

- when $p_\mathbf{a} > n_\mathbf{a}$, we have more axes in critical situation than redundant axes. Of course in that case, the total efficiency of the method cannot be ensured;
- when $p_\mathbf{a} = n_\mathbf{a}$, there is only one solution but the problem can be solved;
- when $p_\mathbf{a} < n_\mathbf{a}$, the system features multiple solutions.

In any case, a solution is given by $\mathbf{a}_0^* = \mathbf{F}^+\mathbf{s}$. The resulting control law is given by

$$\dot{\mathbf{q}} = -\lambda\left(\mathbf{J}_1^+\mathbf{e}_1 + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_{0i}^*\mathbf{E}_{\bullet i}\right). \tag{12}$$

C. Iterative Solution

Let us consider more deeply the last configuration ($p < n_\mathbf{a}$). Using $\mathbf{a}_0^*$ any motion on the $p_\mathbf{a}$ axes in critical situation are stopped. However, with the resulting control law (12), other axes may enter in the critical area (see Fig. 5). This undesired situation can be handled. Indeed when $p_\mathbf{a} < n_\mathbf{a}$, the linear system features multiple solutions and any generalized inverse of $\mathbf{F}$

[1]If $\mathbf{M}$ is a matrix, we note $\mathbf{M}_{\bullet i}$ its ith column and $\mathbf{M}_{i\bullet}$ its ith row.

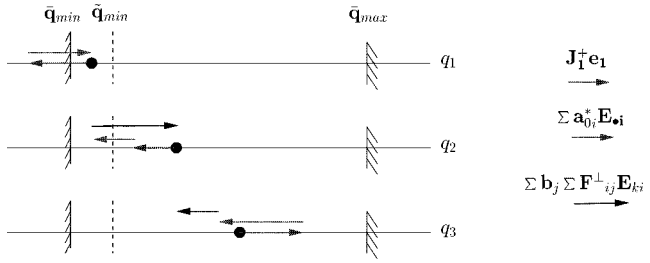


Fig. 5. Iterative algorithm: with the new motion generated by both $\mathbf{e_1}$ and $\sum \mathbf{a}^*_{0_i} \mathbf{E}_{\bullet i}$, new axes may enter the critical area. The iterative version of our algorithm allows to handle this problem.

may be used (and not only the unique pseudo-inverse $\mathbf{F}^+$ of $\mathbf{F}$). $\mathbf{a}^*$ can in fact be chosen as

$$\mathbf{a}^* = \mathbf{a}^*_0 + \sum_{j=1}^{n_\mathbf{b}} \mathbf{b}_j \mathbf{F}^{\perp}_{\bullet j} \tag{13}$$

where $\mathbf{F}^{\perp} = \mathbf{I}_{n_\mathbf{a}} - \mathbf{F}^+\mathbf{F}$ is a basis of $\mathrm{Ker}\ \mathbf{F}$ and $n_\mathbf{b} = \dim \mathrm{Ker}\ \mathbf{F}$. The new motions involved by $\sum_j \mathbf{b}_j \mathbf{F}^{\perp}_{\bullet j}$ are built in the kernel of the constraint (i.e., projected onto the null space of $\mathbf{F}$). The resulting control law has therefore still no effect on the main task and ensures that the $p_\mathbf{a}$ axes initially in critical situation are stopped. Replacing $\mathbf{a}^*$ by its value defined in (13), we get

$$\dot{\mathbf{q}} = -\lambda \left[\mathbf{J_1^+ e_1} + \sum_{i=1}^{n_\mathbf{a}} \left(\mathbf{a}^*_0 + \sum_{j=1}^{n_\mathbf{b}} \mathbf{b}_j \mathbf{F}^{\perp}_{\bullet j} \right)_i \mathbf{E}_{\bullet i} \right]. \tag{14}$$

To determine the vector $\mathbf{b}$, like in the previous case, we build a linear system considering that $\dot{\mathbf{q}}_k = 0$ for all the $p_\mathbf{b}$ axes $\mathbf{q}_k$ that will enter in critical situation area according to the prediction (computed using (8) where $\dot{\mathbf{q}}$ is given by (12)). After some rewriting, each line of the system is given by

$$\sum_{j=1}^{n_\mathbf{b}} \mathbf{b}_j \sum_{i=1}^{n_\mathbf{a}} \mathbf{F}^{\perp}_{ij} \mathbf{E}_{ki} = -\left(\mathbf{J_1^+ e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}^*_{0_i} \mathbf{E}_{\bullet i} \right)_k. \tag{15}$$

As in the previous case, there are three possible cases regarding the dimension of $\mathbf{b}$. Here again, from the obtained $\mathbf{b}^*$ and the corresponding control law $p_\mathbf{c}$, new axes may enter in the critical area. Therefore, the determination and resolution of linear systems is repeated iteratively (and can be repeated as long as $p_\mathbf{a} + p_\mathbf{b} + p_\mathbf{c} + \cdots < n_\mathbf{a}$).

D. Continuity Issue

Let us note that the control law presented in Section III-B is not always continuous. Indeed, as soon as the number of equations of the linear systems involved in our method changes, a discontinuity in the computed gains ($\mathbf{a}$, $\mathbf{b}$, etc.) will occur. For example, when all axes are far away from their joint limits, we have of course $\mathbf{a} = \mathbf{b} = 0$. If an axis becomes critical, at least one component of $\mathbf{a}$ will be different from zero. The norm of $\mathbf{a}$ will mainly depend on the norm of $\mathbf{J_1^+ e_1}$, which induces a discontinuity in $\mathbf{a}$ and thus in $\dot{\mathbf{q}}$. However, since $\mathbf{a}$ and $\mathbf{b}$ are

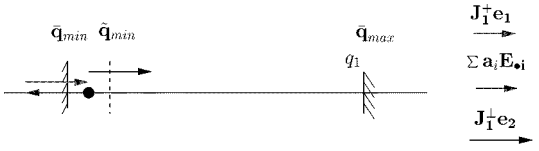


Fig. 6. Considering our algorithm and a secondary task $\sum \mathbf{a}_i \mathbf{E}_{\bullet i}$ stops the motion toward the joint limits while $\mathbf{J_1^\perp e_2}$ generates a motion toward the noncritical area.

Fig. 7. Camera mounted on the end-effector of an Afma cartesian robot.

primarily computed from the pseudo-inverse of the matrix involved in the linear system, their value is by definition the one whose norm is minimal. The obtained discontinuity is thus the minimal possible one.

Furthermore, to decrease the effect of discontinuity, a basic idea is to slightly modify the formulation of the linear systems presented above. Indeed, to produce a smooth decay of the axis velocity, we modify the linear systems (11) and (15) in order to weight, with a coefficient γ_k, the participation of an axis to the avoiding process function of its distance to the joint limit. Between the joint limit $\bar{\mathbf{q}}$ and the threshold $\tilde{\mathbf{q}}$, the velocity of the corresponding axis should decrease and must stop when it reaches $\bar{\mathbf{q}}$. The linear system (11) is then replaced by

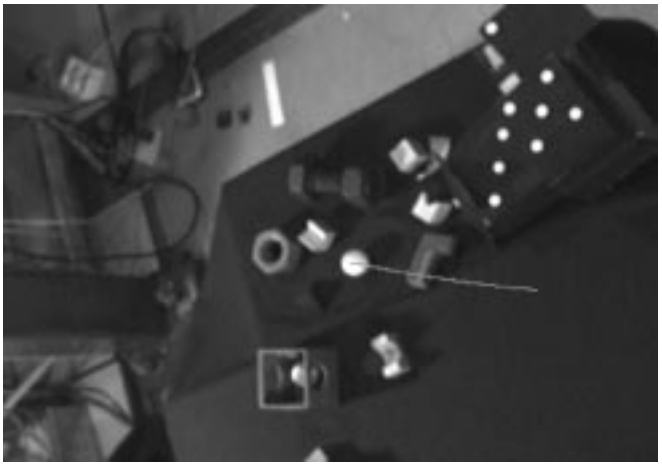
$$\sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{ki} = -\gamma_k \left(\mathbf{J_1^+ e_1} \right)_k \tag{16}$$

where

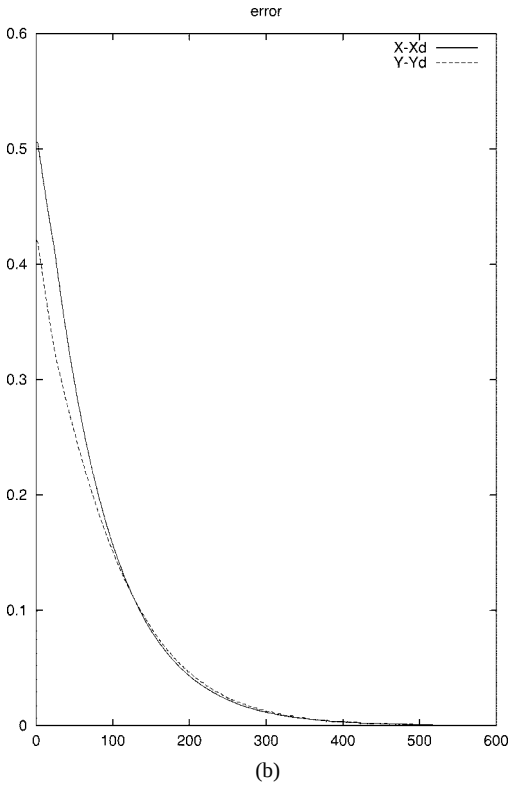
$$\gamma_k = \begin{cases} \dfrac{\hat{\mathbf{q}}_k(t+1) - \tilde{\mathbf{q}}_{k_{\min}}}{\bar{\mathbf{q}}_{k_{\min}} - \tilde{\mathbf{q}}_{k_{\min}}}, & \text{if } \bar{\mathbf{q}}_{k_{\min}} \le \hat{\mathbf{q}}_k(t+1) \le \tilde{\mathbf{q}}_{k_{\min}} \\ 0, & \text{if } \tilde{\mathbf{q}}_{k_{\min}} \le \hat{\mathbf{q}}_k(t+1) < \tilde{\mathbf{q}}_{k_{\max}} \\ \dfrac{\hat{\mathbf{q}}_k(t+1) - \tilde{\mathbf{q}}_{k_{\max}}}{\bar{\mathbf{q}}_{k_{\max}} - \tilde{\mathbf{q}}_{k_{\max}}}, & \text{if } \tilde{\mathbf{q}}_{k_{\max}} \le \hat{\mathbf{q}}_k(t+1) \le \tilde{\mathbf{q}}_{k_{\max}} \end{cases} \tag{17}$$

where $\hat{\mathbf{q}}_k(t+1)$ is still given by (8). Using this prediction in the computation of γ_k (and not the current value $\mathbf{q}_k(t)$) is important to be sure that an axis will not reach its joint limits whatever the value of $\mathbf{J_1^+ e_1}$. In the same way, (15) is now defined by

$$\sum_{j=1}^{n_\mathbf{b}} \mathbf{b}_j \sum_{i=1}^{n_\mathbf{a}} \mathbf{F}^{\perp}_{ij} \mathbf{E}_{ki} = -\gamma_k \left(\mathbf{J_1^+ e_1}_k + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}^*_{0_i} \mathbf{E}_{\bullet i} \right)_k \tag{18}$$



(a)



(b)

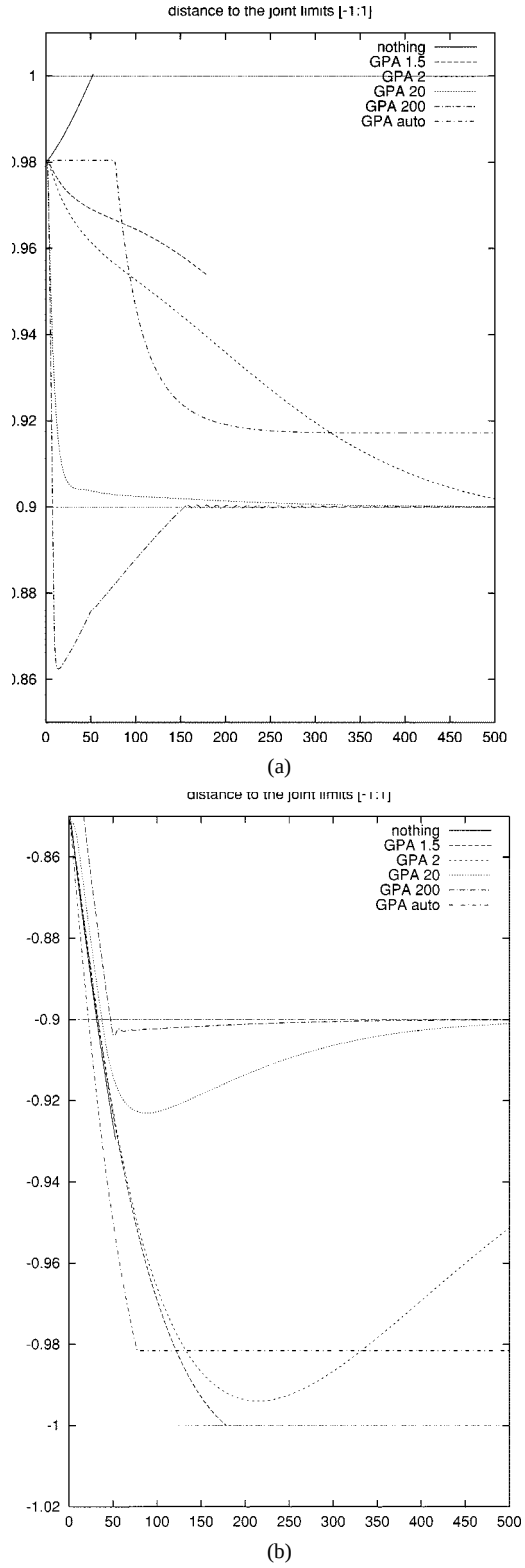
Fig. 8. Visual servoing experiments, servo on a point: (a) final image and target trajectory, (b) error in the image $\mathbf{P} - \mathbf{P_d}$ versus the number of iterations.

where the prediction $\widehat{\mathbf{q}}_k(t+1)$ involved in the computation of γ_k is now given using control law (12) with value $\mathbf{a}_0^*$ obtained by solving (16).

As will be shown on the experimental results presented in Section IV, the discontinuity obtained in practice after using these gains γ is really small. Furthermore, the dynamic of the robot will efficiently smooth the remaining discontinuities.

E. Moving Away From the Joint Limits

The presented framework provides a complete solution to ensure that, if a solution exists, the joints in critical situation will not encounter their limits. It could also be interesting to generate a motion that moves the joints away from their limits (see Fig. 6).



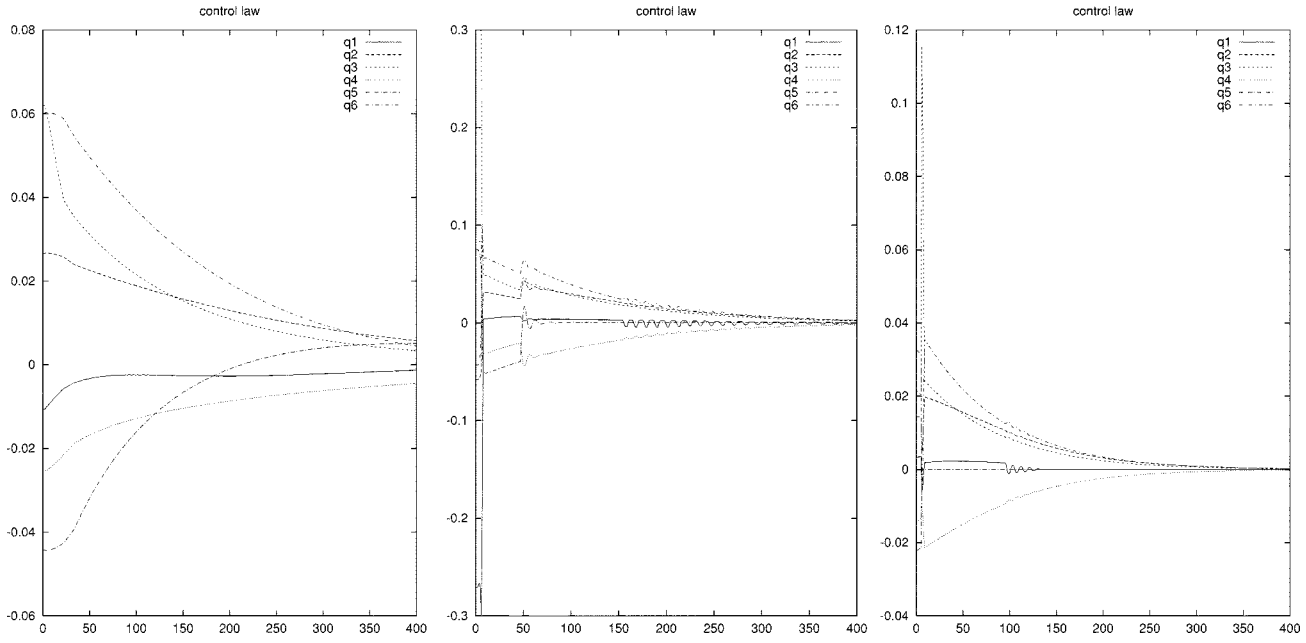
(a)

(b)

Fig. 9. Gradient projection approach: behavior of axes 1 and 5.

This can be simply achieved by introducing a cost function (such as the one proposed in (4)) in the task function $\mathbf{e}$.

$$\mathbf{e} = \mathbf{J_1^+ e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{\bullet i} + \alpha \mathbf{J_1^\perp e_2}. \tag{19}$$



(a) (b) (c)

Fig. 10. Gradient projection approach: control law for $\alpha = 2$, $\alpha = 200$ and for the automatic tuning of α.

Now, tuning gain α is no more critical since it is not involved to avoid joint limits. Finally, the initial linear system to be solved is still the one given by (16). However, the linear system (18) has to be changed. Indeed the control law obtained after the computation of $\mathbf{a}_0^*$ is now

$$\dot{\mathbf{q}} = -\lambda \left(\mathbf{J}_\mathbf{1}^+ \mathbf{e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_{0i}^* \mathbf{E}_{\bullet i} + \alpha \mathbf{J}_\mathbf{1}^\perp \mathbf{e_2} \right) \tag{20}$$

and we obtain

$$\sum_{j=1}^{n_\mathbf{b}} \mathbf{b}_j \sum_{i=1}^{n_\mathbf{a}} \mathbf{F}_{ij}^\perp \mathbf{E}_{ki} = \gamma_k \left(-\mathbf{J}_\mathbf{1}^+ \mathbf{e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_{0i}^* \mathbf{E}_{\bullet i} + \alpha \mathbf{J}_\mathbf{1}^\perp \mathbf{e_2} \right)_k . \tag{21}$$

IV. Application to Visual Servoing

We applied the proposed method to image-based visual servoing. Let us denote $\mathbf{P}$ the set of selected visual features used in the visual servoing task. To ensure the convergence of $\mathbf{P}$ to its desired value $\mathbf{P_d}$, we need to know the interaction matrix (or image Jacobian) $\mathbf{L}_\mathbf{P}^\mathrm{T}$ defined by the classical equation [9]:

$$\dot{\mathbf{P}} = \mathbf{L}_\mathbf{P}^\mathrm{T} \mathbf{T_c} \tag{22}$$

where $\dot{\mathbf{P}}$ is the time variation of $\mathbf{P}$ due to the camera motion $\mathbf{T_c}$.

Control laws in visual servoing are generally expressed in the operational space (i.e., in the camera frame), and then computed in the articular space using the robot inverse Jacobian. However, in order to combine a visual servoing with the avoidance of joint limits, we have to directly express the control law in the articular space. Indeed, manipulator joint limits are defined in this space.

This leads to the definition of a new interaction matrix such that

$$\dot{\mathbf{P}} = \mathbf{H_P} \dot{\mathbf{q}}. \tag{23}$$

Since we have $\mathbf{T_c} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{J}(\mathbf{q})$ is the robot Jacobian, we simply obtain:

$$\mathbf{H_P} = \mathbf{L}_\mathbf{P}^\mathrm{T} \mathbf{J}(\mathbf{q}). \tag{24}$$

If l visual features are selected, the dimension of $\mathbf{H_P}$ is $l \times n$. If the visual features are independent, the rank m of $\mathbf{H_P}$ is equal to l, otherwise $l > m$. The vision-based task $\mathbf{e}_1$ is then defined by

$$\mathbf{e_1} = \mathbf{C}(\mathbf{P} - \mathbf{P_d}) \tag{25}$$

where $\mathbf{C}$, called combination matrix, has to be chosen such that $\mathbf{J_1} = \mathbf{C}\mathbf{H_P}$ is full rank. It can be defined as $\mathbf{C} = \mathbf{W}\mathbf{H}^+_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}}$, where $\mathbf{W}$ is a full rank $m \times n$ matrix such that $\mathrm{Ker}\ \mathbf{W} = \mathrm{Ker}\ \mathbf{H}_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}}$ (see [5], [9] for more details). If $\mathbf{H_P}$ is full rank m, we can set $\mathbf{W} = \mathbf{H}_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}}$, then $\mathbf{C} = \mathbf{I}$ and $\mathbf{J_1} = \mathbf{H_P}$ is a full rank $m \times n$ matrix. If rank m of $\mathbf{H_P}$ is less than l, we have $\mathbf{J_1} = \mathbf{W}\mathbf{H}^+_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}}\mathbf{H}_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}}$ which is also a full rank $m \times n$ matrix.

We can then use the framework presented in Sections I–III.

V. Experimental Results

A. Controlling a 6 d.o.f Robot

All the joint limits avoidance approaches presented in this section have been implemented on an experimental testbed composed of a CCD camera mounted on the end effector of a six degrees of freedom robot (see Fig. 7). The implementation of the control law as well as the image processing has been done on a 400 MHz PC running the Linux Operating System. All matrices

$\mathbf{J}^+$, $\mathbf{J}^\perp$ are easily computed from the singular value decomposition of $\mathbf{J}$. Each iteration is achieved in 80 ms.

1) Positioning Task wrt. a Point: The specified visual task consists in a gazing task. If $\mathbf{P} = (X, Y)$ describes the position in the image of the projection of the center of gravity of an object, the goal is to observe this object at the center of the image: $\mathbf{P}_\mathbf{d} = (0, 0)$. The interaction matrix related to this task is given by the classical equation:

$$\mathbf{L}_\mathbf{P}^\mathrm{T} = \begin{pmatrix} -\frac{1}{z} & 0 & \frac{X}{z} & XY & -(1+X^2) & Y \\ 0 & -\frac{1}{z} & \frac{Y}{z} & 1+Y^2 & -XY & -X \end{pmatrix} \tag{26}$$

where z is the depth of the point.

Fig. 8 shows the results of a successful experiments. Fig. 8(a) shows the final image acquired by the camera. The superimposed line depicts the obtained 2D target trajectory. Fig. 8(b) shows the exponential decay of the error $\mathbf{P} - \mathbf{P}_\mathbf{d}$. Though minor variations may arise, this 2D behavior is similar for all the successful experiments reported in this section.

In the presented experiments, the initial robot position is located in the vicinity of two joint limits $(\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_3)$ while $\mathbf{q}_5$ is located near the threshold $\tilde{\mathbf{q}}_{5_\mathrm{min}}$. If no particular strategy is considered to avoid joint limits, the visual task fails. On all the plots dealing with the joint positions, positions $\mathbf{q}$ are normalized between $[-1; 1]$ where -1 and 1 represent the joint limits $\bar{\mathbf{q}}_\mathrm{min}$ and $\bar{\mathbf{q}}_\mathrm{max}$. On these plots, the thresholds $\tilde{\mathbf{q}}$ are located in ± 0.9 [corresponding to $\rho = 0.1$ in (3)].

2) Gradient Projection Approach: We performed a set of experiments using the cost function defined in Section II-A with various values of the α coefficient. Fig. 9 depicts the joint position of two axes of the robot (Fig. 9(a) for axis 1 and Fig. 9(b) for axis 5) during the experiments for these different values. If α is too small, the motion generated by the main task in the direction of the joint limits is not compensated enough by the secondary task. The robot encounters the limit of axis 1 at iteration 54 for $\alpha = 0$ (i.e., no secondary task) and the limit of axis 5 after iteration 180 for $\alpha = 1.5$. In each case, the specified task is not achieved. The task is achieved with $\alpha \geq 2$. However, if α is too high, it may result in too large velocities as shown in Fig. 10 (with $\alpha = 200$, oscillations are observed on axis $\mathbf{q_1}$, see Fig. 9(a), and translation velocities of nearly 1 m/s are produced!). As pointed out in [2], tuning α is therefore performed based on trial and error. This solution is not acceptable.

The last plot (GPA auto) on Fig. 9 depicts the results obtained using the approach proposed in Section II-C where α is automatically computed. In a first time, motion on axis $\mathbf{q}_1$ is stopped since it is the closest from its joint limit. Then, after iteration 80, axis 5 becomes closer from its joint limits than axis 1, it is then stopped. We can see in Fig. 10(c) that the resulting control law is unstable at the beginning of the process since high α values are computed to avoid the joint limits. Furthermore, using this approach, only a single axis can be stopped and configurations can be exhibited where the system is completely unstable.

3) Results With the New Iterative Approach: The following results deal with experiments considering our new redundancy-based approach. As for the results dealing with the classical GPA, Fig. 11 shows the behavior of the axes $\mathbf{q_1}$ and $\mathbf{q_5}$. Various experiments have been carried out:

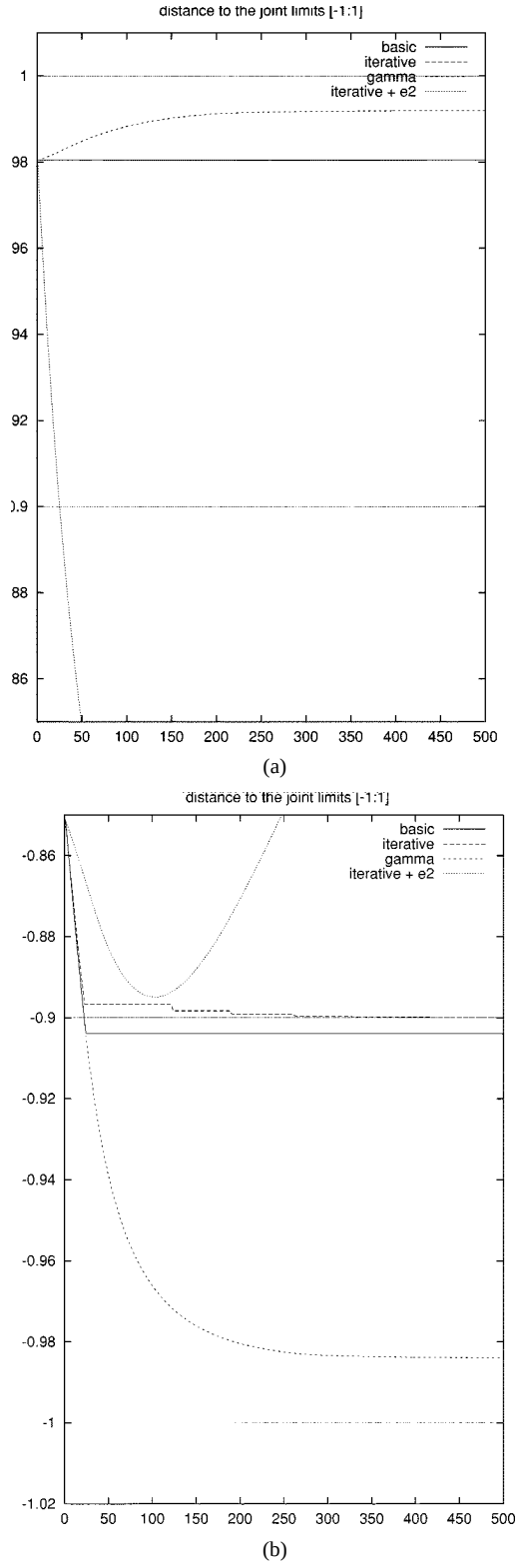


Fig. 11. New approach: behavior of axes 1 and 5.

- a basic version of our algorithm as described in Section III-B. Any axis k that is or enters in the critical area is stopped. Since $\mathbf{q_1}$ is already inside the critical area, $\mathbf{a_0^*}$ is computed to produce a motion $\dot{\mathbf{q}}_\mathbf{1} = 0$. Then, as soon as $\mathbf{q_5}$ enters the critical area (iteration 30) a new vector $\mathbf{a_0^*}$ is computed in order to simultaneously stop the motion on the axes $\mathbf{q_1}$ and $\mathbf{q_5}$. It remains that threshold

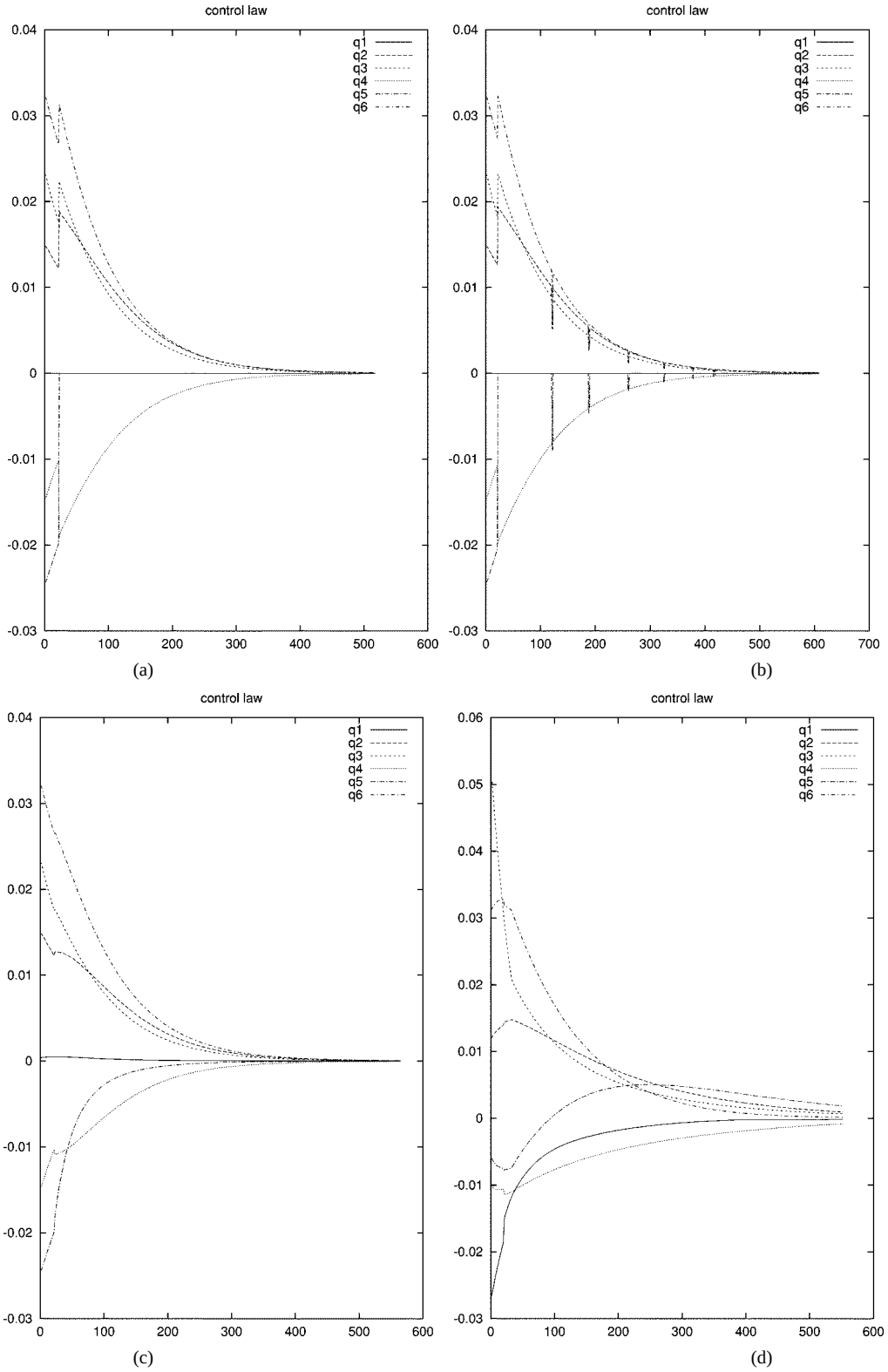


Fig. 12. Control law computed with the new approach: (a) basic algorithm. (b) iterative algorithm $\mathbf{e_2}$. (c) smoothing discontinuities (introducing gain γ_k). (d) iterative algorithm with a cost function $\mathbf{e_2}$.

$\tilde{\mathbf{q}}_5$ is crossed. The resulting control law is presented in Fig. 12(a).

- the iterative version of our algorithm has been built to ensure that an axis that is not already in the critical area will not cross the threshold $\tilde{\mathbf{q}}$. In fact, when the prediction considering $\mathbf{b} = 0$ shows that an axis will enter in the critical area, another solution with $\mathbf{b} \neq 0$ is computed. As can be seen, since $\mathbf{q_1}$ is initially in the critical area, we have $\dot{\mathbf{q}}_1 = 0$, and the computed motion for $\mathbf{q}_5$ allows the robot not to enter the critical area (green plot on Fig. 11). The "stairs" effects of the plot is due to the fact that the motion due to the visual task (i.e., $\mathbf{J}_1^+\mathbf{e_1}$) decreases over

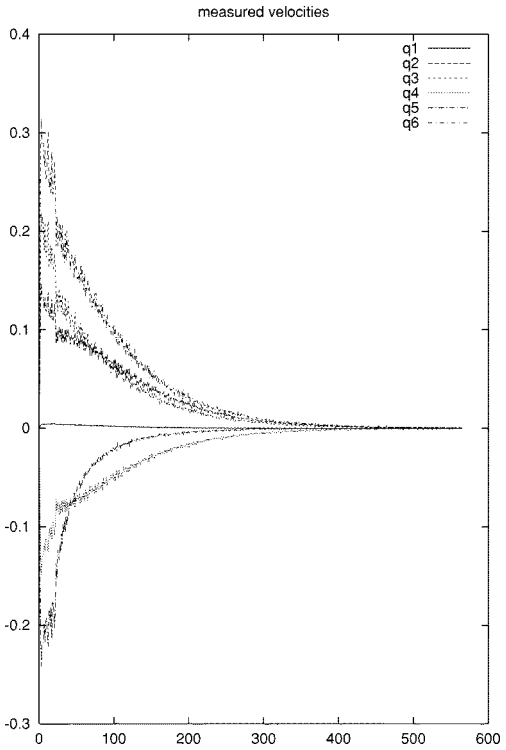
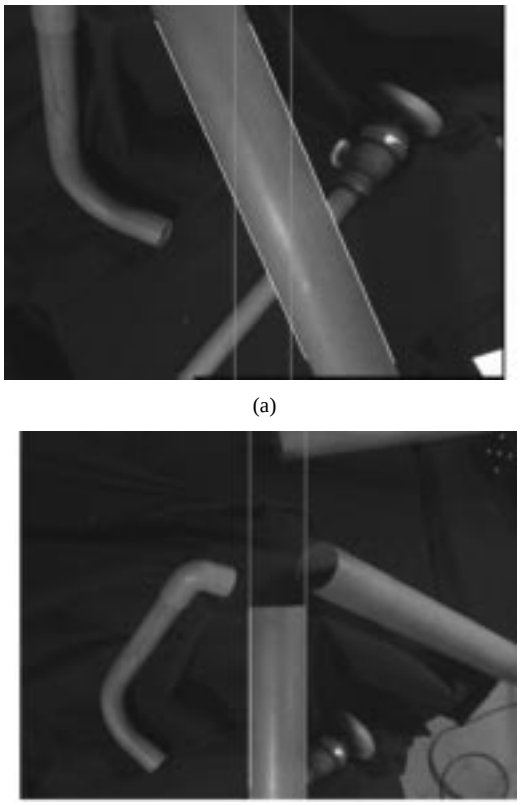


Fig. 13. Actual velocities measured using the robot odometry. The control law is computed in order to smooth discontinuities. If discontinuities remain due to the introduction (or suppression) of equations in the linear system, they are smoothed by the dynamic of the robot (to be compared to Fig. 12(c)).



(a)

(b)

Fig. 14. Visual servoing experiment, servo on a cylinder: initial and final image.

time. Therefore, the predicted position $\widehat{\mathbf{q}}_k(t+1)$ may be no longer in the critical area leading to a small motion on the corresponding axis. Therefore, the joint position tends to $\tilde{\mathbf{q}}$ but never reach this value. The computed control law on Fig. 12(b) reflects this behavior.

- To smooth the discontinuity of the control law, we introduced in Section III-D the gain γ_k that is a function of the distance to joint limits. Introducing this gain has many advantages. First it allows to consider a wider area. Indeed motion is no longer stopped when the critical axis reaches $\tilde{\mathbf{q}}$ but the computed velocity is smoothly decreased such that the motion is stopped only when the axis reaches the real joint limits that is $\bar{\mathbf{q}}$. Let us note that on the experiments reported here, considering this wider area allows a faster convergence of the task. Furthermore the joint limits $\bar{\mathbf{q}}$ are never reached. The second advantage of considering the gain γ_k is that it allows to smooth the discontinuities since the motions are no longer stopped abruptly. Let us however note that discontinuities always exist in the control law presented in Fig. 12(c). More precisely discontinuities remain when a new axis enters or leaves the critical area. Indeed, as already stated, the number of equations of the linear system to be solved is modified and the proposed solution (i.e., the new computed control law) may change. Since the solution with lowest norm is computed, the observed discontinuity is minimum. Furthermore, this discontinuity is smoothed by the dynamics of the robot and fully absorbed by the noise of the system as can be seen in Fig. 13.
- Finally, we considered the introduction of a secondary task $\mathbf{e_2}$ within our approach (see Section III-E). This last step allows to take advantages of both GPA and the new approach. Indeed it ensures that the joint limits will never be reached and the introduction of a cost function allows to move away from the critical area if it is considered to be interesting (see plot "iterative+e2").

4) Positioning wrt. a Cylinder: In this paragraph, we consider a positioning task wrt. a cylinder. This cylinder is expected to be vertical and centered in the image (see Fig. 14). This task constrains four degrees of freedom of the camera. The cylinder parameterization and the related interaction matrix can be found in [10]. If nothing is done, joint limits are encountered after a few iterations.

The iterative solution is considered on Fig. 15(a). As can be seen, axis 1 is initially in the critical area while axis 3 moves toward it (see after iteration 100). As expected the motion is reported on axes 2 and 6 to ensure the realization of the task. The gains γ_k are then introduced in the experiments depicted on Fig. 15(b) to smooth the discontinuities. Though the amplitude of the discontinuities is far less important, it is of course still present. As explained in Section III-E , it would be possible to move axis 1 away from its joint limit by adding a classical secondary task $\mathbf{e_2}$

B. Vision-Based Humanoid Torso Control

We also applied the new proposed approach to the vision-based control of a humanoid torso. A new application

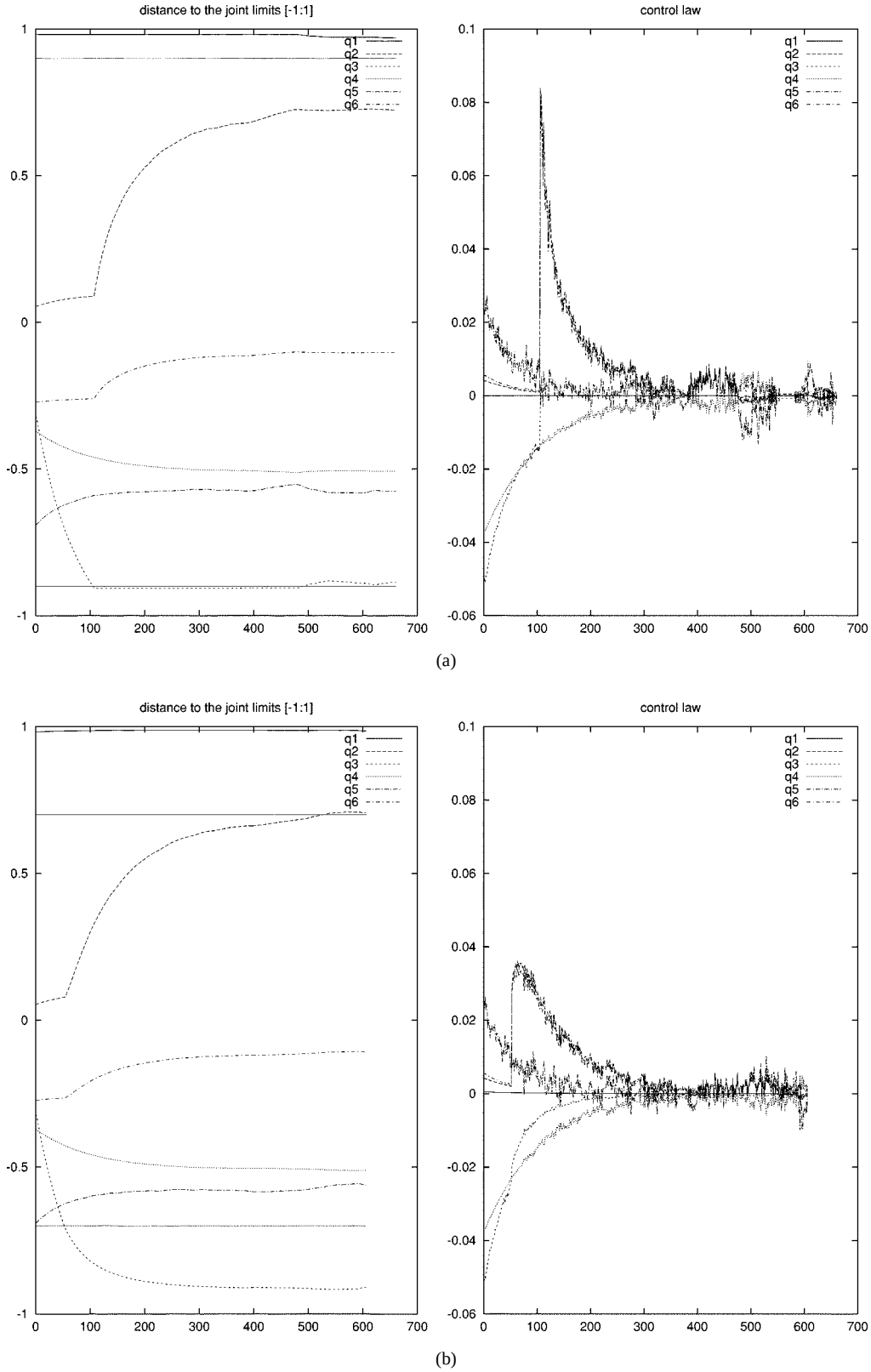


Fig. 15. Visual servoing on a cylinder: distance to the joint limits (left) and camera velocities (right). (a) Iterative algorithm. (b) Smoothing the discontinuities (introducing γ_k).

for visual servoing is computer animation [11] and within this wide domain the control of digital actors (also called virtual humanoids or avatars). Our goal in this paper is not to focus on the computer animation application. Here we just consider the humanoid as a specific robot. The eyes can be considered as a camera mounted on the end-effector of a highly redundant

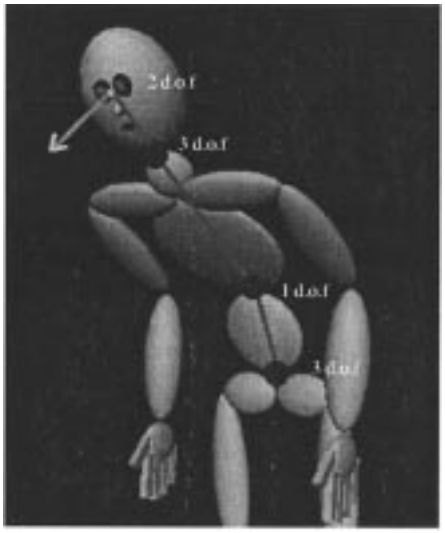
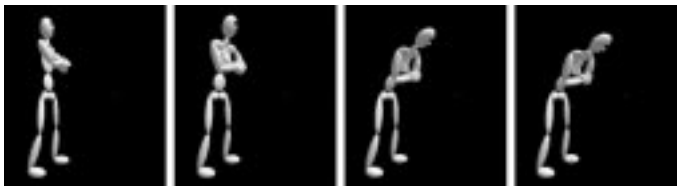
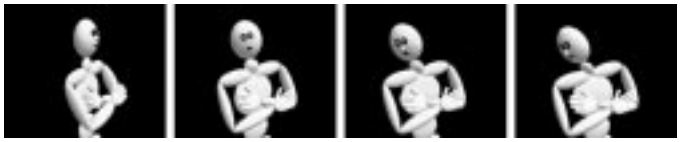


Fig. 16. Humanoid model and degrees of freedom.



(a) (b) (c) (d)

Fig. 17. Humanoid control: positioning wrt. a sphere.



(a) (b) (c) (d)

Fig. 18. Humanoid control: tracking a point.

robot. For the digital actor we consider the animation of the torso of an humanoid with 9 d.o.f. (see Fig. 16): pelvis (3 d.o.f), spine (1 d.o.f), neck (3 d.o.f), eyes (2 d.o.f). The modeling of this humanoid robot has been done using the Denavit–Hertenberg parameterization.

1) Positioning wrt. a Sphere: The first experiment (see Fig. 17) deals with a positioning task wrt. a sphere that has to be seen centered in the image and at a distance z_d in the camera frame. In that case the desired position in the image is given by $\mathbf{P_d} = (0, 0, \pi R^2)$ where πR^2 is the desired surface in the image. Three constraints are induced by this task which means that six d.o.f are then free to deal with the joint limits. The interaction matrix related to the visual task is given by [9]:

$$\mathbf{L}^{\mathrm{T}}_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}} = \begin{pmatrix} -\frac{1}{z_d} & 0 & 0 & 0 & -1-R^2 & 0 \\ 0 & -\frac{1}{z_d} & 0 & 1+R^2 & 0 & 0 \\ 0 & 0 & \frac{\pi R^2}{z_d} & 0 & 0 & 0 \end{pmatrix}. \tag{27}$$

2) Tracking a Point: The second experiment deals with a tracking task. It allows to demonstrate the capabilities of the joint limits avoidance algorithm. A point-object is crossing the scene. On Fig. 18, we can see that, at the beginning, mainly the eyes are moving. When they move near their joint limits the motion is automatically transferred to the neck, then to the pelvis.

VI. Conclusion

We have proposed an original method to avoid the joint limits of a manipulator. It consists in generating automatically camera motions compatible with the main task by iteratively solving a system of linear equations. This new approach is far more efficient than the classical gradient projection method. It avoids unnecessary motions, and unlike gradient projection methods, it guarantees the joint limits avoidance. We have demonstrated on real experiments within a visual servoing context the validity of our approach. Let us finally note that this new approach may be used for other problems where gradient projection approach are classically used, such as obstacle avoidance [5], [12].

Acknowledgment

The authors wish to thank V. Cadenat from LAAS (Toulouse) for her careful reading and remarks on the paper and N. Courty for providing the humanoid model.

References

[1] F. Chaumette and E. Marchand, "A new redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, San Francisco, CA, Apr. 2000, pp. 1720–1725.

[2] T.-F. Chang and R.-V. Dubey, "A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 286–292, Apr. 1995.

[3] B. Nelson and P. K. Khosla, "Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limits avoidance," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 255–269, June 1995.

[4] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 245–250, Mar. 1977.

[5] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control: The Task Function Approach*. Oxford, U.K.: Clarendon, 1991.

[6] E. Marchand, F. Chaumette, and A. Rizzo, "Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96*, vol. 3, Osaka, Japan, Nov. 1996, pp. 1083–1090.

[7] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.

[8] K. Hashimoto, "Visual servoing: Real time control of robot manipulators based on visual sensory feedback," in *World Scientific Series in Robotics and Automated Systems*, Singapore: World Scientific , 1993, vol. 7.

[9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, June 1992.

[10] F. Chaumette, "Visual servoing using image features defined upon geometrical primitives," in *IEEE Int. Conf. on Decision and Control*, vol. 4, Orlando, FL, Dec. 1994, pp. 3782–3787.

[11] N. Courty and E. Marchand, "Computer animation: A new application for image-based visual servoing," in *IEEE Int. Conf. on Robotics and Automation*, vol. 1, Seoul, Korea, May 2001, pp. 223–228.

[12] V. Cadenat, R. Swain, P. Soueres, and M. Devy, "A controller to perform a visually guided tracking task in a cluttered environment," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'99*, Kyongju, Korea, Oct. 1999, pp. 775–780.

François Chaumette was born in Nantes, France, in 1963 and graduated from École Nationale Supérieure de Mécanique, Nantes, in 1987. He received the Ph.D. degree and "Habilitation à Diriger des Recherches" in Computer Science from the University of Rennes in 1990 and 1998 respectively. Since 1990, he has been with IRISA/INRIA, Rennes, France, where he is now "Directeur de Recherche". His research interests include robotics, computer vision, and especially the coupling of these two research domains (vision-based control, active vision and purposive vision).

Dr. Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991.

Éric Marchand received the Ph.D. in Computer Science at INRIA Rennes and defended in 1996 at the University of Rennes. He spent one year as a Postdoctoral Associate in the AI lab of the Department of Computer Science at Yale University. Since 1997, he is an INRIA research scientist ("Chargé de recherche") at IRISA–INRIA Rennes in the Vista project. His research interests include robotics, perception strategies, and especially the cooperation between perception and action and visual servoing. He is also interested in the software engineering aspects of robot programming. More recently, he studies new application fields for visual servoing such as underwater robotics, augmented reality and computer animation.