HAL
open science

A new redundancy formalism for avoidance in visual servoing

Nicolas Mansard, François Chaumette

To cite this version:

Nicolas Mansard, François Chaumette. A new redundancy formalism for avoidance in visual servoing. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05, 2005, Edmonton, Canada, France. pp.1694-1700. inria-00351896

HAL Id: inria-00351896

https://inria.hal.science/inria-00351896v1

Submitted on 12 Jan 2009

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

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

2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

A new redundancy formalism for avoidance in visual servoing

Nicolas Mansard, François Chaumette
IRISA - ENS Cachan and INRIA Rennes
Campus de Beaulieu, 35043 Rennes, France
E-Mail : Firstname.Lastname@irisa.fr

***Abstract*— The paper presents a new approach to construct a control law that realizes a main task and simultaneously takes supplementary constraints into account. Classicaly, this is done by using the redundancy formalism. If the main task does not constrain all the motions of the robot, a secondary task can be achieved by using only the remaining degrees of freedom (DOF). We propose a new general method that frees up some of the DOF constrained by the main task in addition of the remaining DOF. The general idea is to enable the motions produced by the secondary control law that help the main task to be completed faster. The main advantage is to enhance the performance of the secondary task by enlarging the number of available DOF. In a formal framework, a projection operator is built which ensures that the secondary control law does not disturb the main task. A control law can be then easily computed from the two tasks considered. Experiments that implement and validate this approach are proposed. The visual servoing framework is used to position a 6-DOF robot while simultaneously avoiding occlusions and joint limits.**

***Index Terms*— Redundancy, avoidance, gradient projection approaches, visual servoing, joint-limit and occlusions avoidance**

I. Introduction

Classical control laws are based on the minimization of a task function which corresponds to the realization of a given objective. Usually, this main task only concerns the position of the robot with respect to a target and does not take into account the environment in which the robot evolves. However, to integrate the servoing into a complex real robotic system, the control law should also make sure that it avoids undesirable configurations, such as joint limits, kinematic singularities, or occlusions if visual servoing is used.

Two very different approaches have been proposed in the litterature. A first solution is to plan the trajectory, for example by using the potential field method [5], [12]. The idea is to choose in advance an optimal trajectory. This provides a complete solution, which ensures the obstacles avoidance when moving to complete the main task. However, the path planning requires a lot of information about the obstacles to avoid. This solution is thus less reactive to changes of the goal, of the environment or of the constraints.

Rather than deciding in advance which path should be used to reach the goal, another approach conciders the obstacles through an objective function to be minimized. This provides very reactive solutions, since it is very easy to modify the objective function during the servo. A first solution to take the obstacle objective function into account is to realize a trade off between the main task and the obstacle avoidance [13]. In this approach, the control law generates motions that try to make the main task function decrease and simultaneously take the robot away from the kinematic singularities and the joint limits. On the opposite, a second solution is to dampen any motion that brings the robot into the obstacles. This solution has been applied using the weighted least norm solution to avoid joint limits [2]. The control law does not induce any motion to take the robot away from the obstacles, but it forbids any motion in their direction. Thus, it avoids oscillations and unnecessary motions.

However, the use of one of these two methods can strongly disturb the execution of the main task. A second specification is thus generally added: the avoidance scheme should not disturb the main task. The Gradient Projection Method (GPM) has been initially introduced for non-linear optimization [14] and applied then in robotics [10], [15], [6]. The constraints imposed by the environment are embeded into a cost function. Using the potential field method [9], the gradient of this cost function is computed as a second task that moves the robot aside the obstacles. This gradient is then projected onto the set of motions that keep the main task invariant and added to the first part of the control law that performs the main task. The main advantage of this method wrt. [13] and [2] is that, thanks to the choice of the adequate projection operator, the avoidance task has absolutely no effect on the main task. The GPM has been used in numerous works, for example occlusion and joint-limit avoidance [11], animation of virtual humanoids [1], or human-machine cooperation using vision control [7]. However, since the avoidance is performed under the constraint that the main task is realized, the avoidance contribution can be so disturbed that it becomes unefficient. In fact, only the degrees of freedom (DOF) not controlled by the main task can be exploited to perform the avoidance. The more complicate the main task is, the more DOF it uses, and the more disturbed the avoidance will be. Of course, if the main task uses all the DOF of the robot, the avoidance cannot be applied at all.

Nevertheless, even if a DOF is controled by the primary control law, the avoidance law should be taken into account if it "goes in the same way" than the main task. Imposing the avoidance law to let the main task invariant can be a too strong condition. We rather propose in this article a more general solution that only imposes to the gradient not to increase the error of the main task. By this way, the free space on which the gradient is projected is larger. More DOF are thus available for avoidance, and the avoidance law is less disturbed.

To validate this approach, we apply the proposed method to

a visual servoing problem. Visual servoing consists in a closed loop reacting to image data [8], [6]. It is a typical problem where the constraints of the workspace are not considered into the main task. In our experiments, the GPM is used for robot joint limits and visual occlusions avoidance. We show that, in certain cases, using avoidance with GPM fails while our method succeeds.

The paper is organized as follow. Section II recalls the classical GPM formalism. Section III presents our original projection method. The visual servoing framework is quickly presented in Section IV. Section V describes several experiments that show the advantages of the proposed method.

II. Classical redundancy formalism

In this section, we recall the classical redundancy formalism used to compute a control law that takes into account the main task having priority and a secondary term that can be used for example for avoidance. This formalism arises from the general task function approach [15]

A. Control law without avoidance

Let $\mathbf{q}$ be the articular position of the robot. The main task function is $\mathbf{e}$. The robot is controlled using the articular velocities $\dot{\mathbf{q}}$. The jacobian of the main task $\mathbf{e}$ is $\mathbf{J}$ defined by:

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} \tag{1}$$

Let n be the number of DOF of the robot ($n = \dim \mathbf{q}$) and k be the number of DOF used to realize the main task ($k = \dim \mathbf{e}$). The task function is said to be full rank if its jacobian is full rank, i.e. if $\text{rank}(\mathbf{J}) = k$. If the main task is not full rank, an equivalent full-rank task function can be found, by multiplying the task function by an adequate combination matrix $\mathbf{C}$ (e.g. $\mathbf{C} = \mathbf{J}^+$, where $\mathbf{J}^+$ is the pseudo-inverse, or least-squares inverse of $\mathbf{J}$ [15]). In the following, the main task is assumed to be full rank.

A classical control law is obtained by setting a proportional relation between $\mathbf{e}$ and $\dot{\mathbf{e}}$:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{2}$$

where λ is a positive parameter that tunes the convergence speed. This differential equation sets an exponential decrease of the error. By using (1) and (2), the articular motion $\dot{\mathbf{q}}$ that realises this motion $\dot{\mathbf{e}}$ in the image is given by the least-square inverse:

$$\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{e}} = -\lambda \mathbf{J}^+\mathbf{e} \tag{3}$$

In the following, we need to be able to compute the effect of such a control law on the task function. This is done by introducing the differentials $\mathbf{de}$ and $\mathbf{dq}$ of $\mathbf{e}$ and $\mathbf{q}$. Since $\mathbf{dq} = \dot{\mathbf{q}}dt$ and $\mathbf{de} = \dot{\mathbf{e}}dt$, the control vector $\mathbf{dq}$ can be written from (3):

$$\mathbf{dq} = \mathbf{J}^+\mathbf{de} \tag{4}$$

By applying $\mathbf{dq}$, the robot reaches the position $\mathbf{q} + \mathbf{dq}$. The value of the task function at this position is computed using the first order approximation (1):

$$\mathbf{e}(\mathbf{q} + \mathbf{dq}) = \mathbf{e}(\mathbf{q}) + \mathbf{J}\mathbf{dq} \tag{5}$$

In the remaining, we note $\mathbf{e} = \mathbf{e}(\mathbf{q})$ to simplify the notations. It is easy to check that the evolution of the task function when applying $\mathbf{dq}$ is $\mathbf{de}$ as specify, by introducing the control vector $\mathbf{dq}$ computed in (4) into (5):

$$\mathbf{e}(\mathbf{q} + \mathbf{dq}) = \mathbf{e} + \mathbf{de} \tag{6}$$

In fact, specifying a motion $\mathbf{de}$ to be accomplished in the task space, the smallest articular motion that performs $\mathbf{de}$ is $\mathbf{dq}$ given by (4):

$$\mathbf{dq} = \min\{\mathbf{z} \setminus \ \mathbf{e}(\mathbf{q} + \mathbf{z}) = \mathbf{e} + \mathbf{de}\} \tag{7}$$

B. Redundancy

The solution $\mathbf{dq}$ computed above is only one particular solution of (6). The redundancy formalism [15] uses a more general solution which enables to consider a secondary criterion. The robot motion is given by:

$$\mathbf{dq} = \mathbf{J}^+\mathbf{de} + \mathbf{P}\mathbf{dq_2} \tag{8}$$

where $\mathbf{P}$ is the projection operator onto the null space of the matrix $\mathbf{J}$ (i.e. $\mathbf{P} = \mathbf{I_n} - \mathbf{J}^+\mathbf{J}$), and $\mathbf{dq_2}$ is an arbitrary vector, used to apply a secondary command, which will be performed without disturbing the main task $\mathbf{e}$ having priority. The articular motion $\mathbf{dq}$ given by (8) produces the specified motion $\mathbf{de}$ in the task function space:

$$\mathbf{e}(\mathbf{q} + \mathbf{dq}) = \mathbf{e}(\mathbf{q}) + \mathbf{J}\mathbf{J}^+\mathbf{de} + \mathbf{J}\mathbf{P}\mathbf{dq_2} = \mathbf{e} + \mathbf{de} + \mathbf{0} \tag{9}$$

since $\mathbf{J}\mathbf{J}^+ = \mathbf{I_k}$ ($\mathbf{J}$ is full rank) and $\mathbf{J}\mathbf{P} = \mathbf{J}(\mathbf{I_n} - \mathbf{J}^+\mathbf{J}) = \mathbf{0}$. $\mathbf{dq}$ is chosen to realize exactly the motion $\mathbf{de}$ in the task function space, and to perform at best a secondary task whose corresponding control law is $\mathbf{dq_2}$.

In this framework, the avoidance is performed by this second part of the control law $\mathbf{dq_2}$. The number of DOF available for the avoidance is thus $\dim(\mathbf{q}) - \text{rank}(\mathbf{J})$. Experiments presented in Section V show that, when the rank of $\mathbf{J}$ is high, it is very uneasy to perform the avoidance. We thus propose to enlarge the size of the space of motions available for avoidance, by computing the projection operator $\mathbf{P}$ differently.

III. Enlargement of the main-task free space

In the classical redundancy formalism, the control law $\mathbf{dq}$ is built to respect (6). However, it is not necessary to satisfy this egality. The control law $\mathbf{dq}$ has just to ensure that the task-function norm decreases. The problem can therefore be reformulated: let us search $\mathbf{dq}$ such that

$$||\mathbf{e}(\mathbf{q} + \mathbf{dq})|| \leq ||\mathbf{e} + \mathbf{de}|| \tag{10}$$

When using such a $\mathbf{dq}$, the convergence of the error will be at least as fast as the one obtained using (8). The control law does not provide anymore the exponential decoupled decrease of the main task function, but ensures that, at each step of the servo, the behaviour of each of its components will be at least as good as an exponential decrease.

We now build the general control law that respects the condition (10). First this condition is reformulated by using the singular values decomposition (SVD) (Section III-A). We

then build the free space of the main task wrt. (10) (Section III-B), and express the projection operator onto this space (Section III-C). The control law derived from this formalism is finally provided in section III-D.

A. Formulation of the condition

The control law has the following form:

$$\mathbf{dq} = \mathbf{J}^{+}\mathbf{de} + \mathbf{dq_2} \tag{11}$$

We search the most general conditions about $\mathbf{dq_2}$ such that $\mathbf{dq}$ respects the condition (10).

To simplify the formulation, adequate basis of both articular and task function spaces are chosen. Let $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$ be the result of the SVD of $\mathbf{J}$:

$$\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top} \tag{12}$$

with $\mathbf{V}$ a basis of the articular space, $\mathbf{U}$ a basis of the task function space, $\mathbf{\Sigma} = [\ \Delta_\sigma \ \ \mathbf{0}\ ]$, and Δ_σ is the diagonal matrix whose coefficients are the singular values of $\mathbf{J}$, noted σ_i ($\sigma_i > 0$).

Computing $\mathbf{e}(\mathbf{q}+\mathbf{dq})$ by using the basis $\mathbf{U}$ and $\mathbf{V}$ gives

$$\mathbf{e}(\mathbf{q}+\mathbf{dq}) = \mathbf{e} + \mathbf{J}\mathbf{dq} = \mathbf{e} + \mathbf{de} + \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}\mathbf{dq_2} \tag{13}$$

Since $\mathbf{U}$ is orthonormal ($\mathbf{U}^{\top}\mathbf{U} = \mathbf{I}$), the previous equation is equivalent to:

$$\mathbf{U}^{\top}\mathbf{e} + \mathbf{\Sigma}\mathbf{V}^{\top}\mathbf{dq} = \mathbf{U}^{\top}\mathbf{e} + \mathbf{U}^{\top}\mathbf{de} + \mathbf{\Sigma}\mathbf{V}^{\top}\mathbf{dq_2} \tag{14}$$

By writting the task function $\mathbf{e}$ with basis $\mathbf{U}$ and the articular vector with basis $\mathbf{V}$, the framework is reduced to the case where the jacobian $\mathbf{J}$ is diagonal. The task function vector $\mathbf{e}$ expressed using basis $\mathbf{U}$ is noted $\widetilde{\mathbf{e}}$. The articular vector $\mathbf{q}$ computed in basis $\mathbf{V}$ is noted $\widetilde{\mathbf{q}}$. Eq. (13) can be rewritten with a diagonal jacobian:

$$\widetilde{\mathbf{e}} + \mathbf{\Sigma}\widetilde{\mathbf{dq}} = \widetilde{\mathbf{e}} + \widetilde{\mathbf{de}} + \mathbf{\Sigma}\widetilde{\mathbf{dq_2}}, \tag{15}$$

and Condition (10) can be simply written as:

$$||\widetilde{\mathbf{e}} + \widetilde{\mathbf{de}} + \mathbf{\Sigma}\widetilde{\mathbf{dq_2}}|| \leq ||\widetilde{\mathbf{e}} + \widetilde{\mathbf{de}}|| \tag{16}$$

Since $\mathbf{\Sigma}$ is diagonal with positive terms, we now just have to check component by component if the control law $\widetilde{\mathbf{dq_2}}$ makes the absolute value of the error $\widetilde{\mathbf{e}} + \widetilde{\mathbf{de}}$ decrease. Each component of $\widetilde{\mathbf{dq_2}}$ that does not respect this rule should be reduced or nullified to ensure that the condition (16) is respected.

B. Construction of the free space

For some vector $\mathbf{a}$, let $\mathbf{F_a}$ be the set

$$\mathbf{F_a} = \{\mathbf{x} \setminus \ ||\mathbf{x}+\mathbf{a}|| \leq ||\mathbf{a}||\} \tag{17}$$

$\mathbf{F_a}$ is the ball of radius $||\mathbf{a}||$ and center $(-\mathbf{a})$. It is represented on Fig. 1 in the case of a 2D vector space. It corresponds to the free space of the task function in the task function space, i.e. the set of motions $\mathbf{\Sigma}\widetilde{\mathbf{dq}}$ that makes the main task function decrease. We have thus characterized the set of all the possible secondary motions $\widetilde{\mathbf{dq_2}}$ such that $\mathbf{dq}$ respects the initial condition (10). This condition can thus be written as:

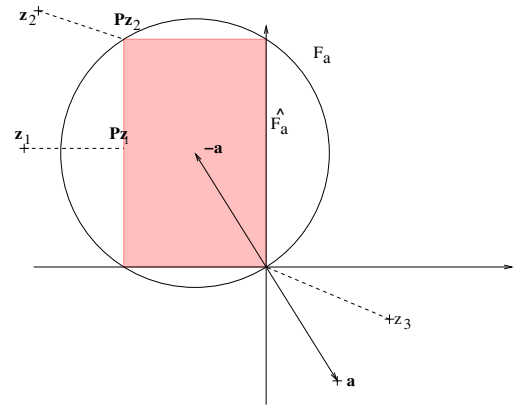

Fig. 1. The two sets $\mathbf{F_a}$ (the black circle) and $\widehat{\mathbf{F_a}}$ (the pink rectangle) in dimension 2. Three points $\mathbf{z_1}$, $\mathbf{z_2}$ and $\mathbf{z_3}$ are projected into $\widehat{\mathbf{F_a}}$ as a matter of example. Their projection are respectively $\mathbf{Pz_1}$, $\mathbf{Pz_2}$ and $\mathbf{0}$. The projection is realized by applying the projection operator computed in Section III.C

$$\mathbf{\Sigma}\widetilde{\mathbf{dq_2}} \in \mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}} \tag{18}$$

Given an arbitrary secondary command $\mathbf{z}$, we now want to modify this vector to obtain a second term $\mathbf{dq_2}$ that does not disturb the main task. If $\mathbf{z}$ belongs to the free space, it can be directly added to the primary control law ($\mathbf{dq_2} = \mathbf{z}$). Otherwise, it should be projected into the free space. The projection operator is computed using the analytical parametrization of $\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}$. By developing the square of the norms in (16), we obtain after some simple calculations:

$$2\sum_{i=1}^{k}(\widetilde{e_i} + \widetilde{de_i})\sigma_i\widetilde{dq_{2_i}} + \sum_{i=1}^{k}\sigma_i^2\widetilde{dq_{2_i}}^2 \leq 0 \tag{19}$$

In order to reduce the complexity of the equation to compute the projection operator into $\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}$, the set $\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}$ is reduced to its cartesian subset. A sufficient condition is thus:

$$\forall i \in [1..k], \quad 2(\widetilde{e_i} + \widetilde{de_i})\sigma_i\widetilde{dq_{2_i}} + \sigma_i^2\widetilde{dq_{2_i}}^2 \leq 0 \tag{20}$$

The set defined by (20) is noted $\widehat{\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}}$. It is represented with the corresponding $\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}$ set on Fig. 1. $\widehat{\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}}$ is in fact the ball defined by the norm $||.||_\infty$.

C. Construction of the projection operator

The projection operator into the free space $\widehat{\mathbf{F}_{\widetilde{\mathbf{e}}+\widetilde{\mathbf{de}}}}$ is noted $\mathbf{P}$. It is a vectorial operator that transforms any vector $\mathbf{z}$ into a secondary control law $\mathbf{dq_2} = \mathbf{P}(\mathbf{z})$ that does not disturb the main task, and such that $||\mathbf{z} - \mathbf{dq_2}||$ is minimal. Using the analytical characterization of the free space given by (20), this projection operator can be computed component by component within basis $\mathbf{V}$.

Condition (20) can be developed by dividing by $\sigma_i\widetilde{dq_{2_i}}$ (if

not zero). $\widetilde{\mathbf{dq_2}}$ belongs to the free space of the main task *iff*

$$\forall i \in [1..k], \quad \begin{cases} & \sigma_i = 0 \\ \text{or} & \widetilde{dq_{2_i}} = 0 \\ \text{or} & 0 < \sigma_i \widetilde{dq_{2_i}} \leq -2(\widetilde{e_i} + \widetilde{de_i}) \\ \text{or} & -\frac{2}{\sigma_i}(\widetilde{e_i} + \widetilde{de_i}) \leq \sigma_i \widetilde{dq_{2_i}} < 0 \end{cases} \tag{21}$$

For each component z_i of $\mathbf{z}$, the closest $\widetilde{dq_{2_i}}$ that respects (21) can be computed. By analysing each case separately, the general expression of $\widetilde{dq_{2_i}}$ is deduced:

$$\widetilde{dq_{2_i}} = \begin{cases} z_i & \text{if } i > k \\ z_i & \text{if } \sigma_i z_i \text{ respects } (21) \\ 0 & \text{if } \sigma_i z_i > 0 \text{ and } \widetilde{e_i} + \widetilde{de_i} < 0 \\ 0 & \text{if } \sigma_i z_i < 0 \text{ and } \widetilde{e_i} + \widetilde{de_i} > 0 \\ 2(\widetilde{e_i} + \widetilde{de_i}) & \text{otherwise} \end{cases} \tag{22}$$

This equation can be written under a matricial form:

$$\widetilde{\mathbf{dq_2}} = \mathbf{P}(\mathbf{z}) = \widetilde{\mathbf{P_z}}\mathbf{z} = \begin{pmatrix} p_1(\mathbf{z}) & & 0 \\ & \ddots & \\ 0 & & p_n(\mathbf{z}) \end{pmatrix} \mathbf{z} \tag{23}$$

where

$$p_i(\mathbf{z}) = \begin{cases} 1 & \text{if } \sigma_i z_i \text{ respects } (21) \\ 0 & \text{if } \sigma_i z_i > 0 \text{ and } \widetilde{e_i} + \widetilde{de_i} < 0 \\ 0 & \text{if } \sigma_i z_i < 0 \text{ and } \widetilde{e_i} + \widetilde{de_i} > 0 \\ -\frac{2(\widetilde{e_i} + \widetilde{de_i})}{\sigma_i z_i} & \text{otherwise} \end{cases}$$

It has to be noticed that $\mathbf{P}$ is not linear: the associated matrix $\widetilde{\mathbf{P}}(\mathbf{z})$ is computed from $\mathbf{z}$. The term $\widetilde{\mathbf{P_z}}\mathbf{z}$ is thus not linear in $\mathbf{z}$. Moreover, the matrix $\widetilde{\mathbf{P_z}}$ is not a projection matrix (its diagonal should be composed only of 0 and 1). It is only the matricial form of the non linear projection operator $\mathbf{P}$.

D. Control law

The projection operator $\widetilde{\mathbf{P_z}}$ is computed into the SVD basis $\mathbf{U}$ and $\mathbf{V}$. We note $\mathbf{P_z}$ this operator into the cannonical basis of the articular space:

$$\mathbf{P_z} = \mathbf{V}\widetilde{\mathbf{P_z}}\mathbf{V}^\top \tag{24}$$

We finally obtain the control law, which is very close to the classical redundancy formalisation (8):

$$\mathbf{dq} = \mathbf{J}^+\mathbf{de} + \mathbf{P_{dq_2}}\mathbf{dq_2} \tag{25}$$

where $\mathbf{dq_2}$ is an arbitrary vector, used to perform a secondary command without disturbing the decreasing speed of the main task error.

E. Comparisons and conclusion

As in the classical formalism, the projection operator is used to transform any secondary vector into a secondary control law that does not disturb the main task. Within the same basis $\mathbf{V}$, the projection operator of the classical redundancy is also a diagonal matrix, but whose coefficients are

$$p_i = \begin{cases} 1 & \text{if } i > k \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

In other terms, the projection operator that we have defined has more non zero coefficients. When a component of the main task function error is not zero, a DOF is freed up. Furthermore, the proposed formalism accelerates the decreasing of the error and takes the secondary part of the command in the same way.

The formalism has been proposed for any task function $\mathbf{e}$ and any secondary vector $\mathbf{z}$. It will now be applied in a visual-servoing framework in practical robotic experiments.

IV. Visual servoing

The method presented above has been applied in visual servoing. We quickly recall here the framework used to position a eye-in-hand robot wrt. a target, and simultaneously to take into account the joint limits and possible occlusions.

A. Main task function using visual servoing

In the experiments presented below, an eye-in-hand robot has to move with respect to a visual target (a rectangle composed of four points easily detectable). By choosing a very simple target, the experiments have focused on the control part of the work.

The task function $\mathbf{e}$ used in the following is the difference between the visual features $\mathbf{s}$ computed at the current time and the visual features $\mathbf{s}^*$ extracted from the desired image [6]:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{27}$$

The interaction matrix $\mathbf{L_s}$ related to $\mathbf{s}$ is defined such that $\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}$, where $\mathbf{v}$ is the kinematic camera screw. From (27), it is clear that the interaction matrix $\mathbf{L_s}$ and the task jacobian $\mathbf{J}$ are linked by the relation:

$$\mathbf{J} = \mathbf{L_s}\mathbf{M}\mathbf{J_q} \tag{28}$$

where the matrix $\mathbf{J_q}$ denotes the robot jacobian ($\dot{\mathbf{r}} = \mathbf{J_q}\dot{\mathbf{q}}$) and $\mathbf{M}$ is the matrix that relates the variation of the camera velocity $\mathbf{v}$ to the variation of the chosen camera position parametrization $\mathbf{r}$ ($\mathbf{v} = \mathbf{M}\dot{\mathbf{r}}$).

In the experiments presented below, the target was composed by a set of points (by choosing a very simple target, the experiments have focused on the control part of the work). In order to have a better and easier control over the robot trajectory, approximatively decoupled features are chosen as proposed in [3], [16]. Six visual features derived from the moments of the set of points have been chosen to control the six DOF of the robot. The two first features are the position x_g and y_g of the center of gravity. Since the considered object is discrete, the third feature a is based on the centered moments of order 2 to control the range between the robot and the target, as proposed in [16]. The fourth feature α mainly rotates the camera around the optical axis. It uses a combination of the three moments of second order to realize a decoupled motion [3]. The two last features s_x and s_y use third order moments to decouple the translational velocities υ_x and υ_y from their corresponding rotational velocities ω_y and ω_x. The reader is invited to refer to [3] for more details.

B. Potential Field Method

Potential fields are classically used to compute avoidance control laws [9]. In this approach, the robot moves according to a repulsive potential V pushing it away from the obstacles. At each iteration, an artificial force $\mathbf{F}(\mathbf{q})$ is induced by the potential function at the current position. The classical solution is to move the robot according to the gradient of the potential function, computed in the articular space. However, the potential function is generally expressed in the space of the obstacles to avoid (e.g. for occlusion, the potential function is generally expressed in the image space). It is thus difficult to compute the gradient of such a function directly in the control space.

We rather compute the gradient directly in the space of the configuration to avoid. Let $\mathbf{\Phi}$ be a parametrisation of this space. The potential field is now $V_{\mathbf{\Phi}} = V(\mathbf{\Phi}(\mathbf{q}))$. It can be shown that the artificial force associated to this potential field is [12]:

$$\mathbf{F}_{\mathbf{\Phi}}(\mathbf{q}) = -\left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{q}}\right)^{+} \vec{\nabla}_{\mathbf{\Phi}}^{\top} V_{\mathbf{\Phi}} \tag{29}$$

C. Occlusion and joint-limit avoidance laws

For each kind of obstacle, an avoidance control law can now be computed by simply defining an associate cost function. In this section, we present two cost functions, the first for the joint-limit avoidance, the second for the occlusion avoidance. The obtained control laws are also explicited.

1) Joint-limit avoidance: The cost function $V_{\mathbf{q}}^{joint}$ is defined directly in the articular space (the jacobian $\frac{\partial \mathbf{\Phi}}{\partial \mathbf{q}}$ defined in (29) is thus the identity matrix). It reaches its maximal value near its joint limits, and the gradient is nearly zero far from the limits.

The robot lower and upper joint limits for each axis i are denoted $\bar{q}_i^{\min}$ and $\bar{q}_i^{\max}$. The robot configuration $\mathbf{q}$ is acceptable if for each i, $q_i \in [\bar{q}_{\ell i}^{\min}, \bar{q}_{\ell i}^{\max}]$, where

$$\begin{cases} \bar{q}_{\ell i}^{\min} = \bar{q}_i^{\min} + \rho d\bar{q}_i \\ \bar{q}_{\ell i}^{\max} = \bar{q}_i^{\max} - \rho d\bar{q}_i \end{cases} \tag{30}$$

where $d\bar{q}_i = \bar{q}_i^{\max} - \bar{q}_i^{\min}$ is the length of the domain of the articulation i, and ρ is a tuning parameter, in $[0, 1/2]$ (typically, $\rho = 0.1$). $\bar{q}_{\ell i}^{\min}$ and $\bar{q}_{\ell i}^{\max}$ are activation thresholds. In the acceptable interval, the avoidance force should be zero. The cost fuction is (see Fig. 2) [4]:

$$V_{\mathbf{q}}^{joint}(\mathbf{q}) = \frac{1}{2} \sum_{i=1}^{n} \frac{{d_i}^2}{d\bar{q}_i} \tag{31}$$

where

$$d_i = \begin{cases} q_i - \bar{q}_{\ell i}^{\min}, & \text{if } q_i < \bar{q}_{\ell i}^{\min} \\ q_i - \bar{q}_{\ell i}^{\max}, & \text{if } q_i > \bar{q}_{\ell i}^{\max} \\ 0, & \text{otherwise} \end{cases}$$

2) Occlusion avoidance: Occlusion avoidance depends on data extracted from the image. An image processing step detects the occluding object (if any). The avoidance law should maximize the distance d between the occluding object and the visual target that is used for the main task. Let d_x and d_y be the x and y coordinates of the distance between the target and the occluding object ($d = \sqrt{d_x^2 + d_y^2}$) and $\mathbf{p_a}$ be the point of the occluding object that is the closest to the target.

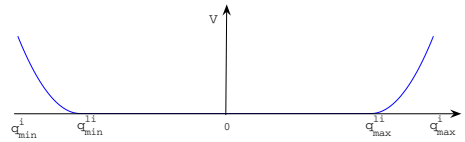

Fig. 2. Potential field of the joint-limit avoidance for one articulation

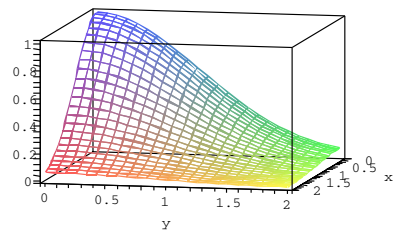

Fig. 3. Potential field of the occlusion avoidance in the image space

The cost function $V_{\mathbf{s}}^{occ}$ is defined directly in the image space. It is maximal when d is 0, and nearly 0 when d is high (see Fig. 3). Like in [11], we simply choose:

$$V_{\mathbf{s}}^{occ}(d) = e^{-\beta d^2} \tag{32}$$

where $\mathbf{s} = (x, y)$ denotes the image parameters. The parameter β is arbitrary and can be used to tune the effect of the avoidance control law. The gradient in the image space is obtain by a simple calculation:

$$\vec{\nabla}_{\mathbf{s}}^{\top} V_{\mathbf{s}}^{view} = \begin{pmatrix} -2\beta d_x e^{-\beta d^2} \\ -2\beta d_y e^{-\beta d^2} \end{pmatrix} \tag{33}$$

The artificial force that avoids the occlusions can be now computed using (29). The transformation from the image space to the articular space is given by [12]:

$$\mathbf{F}_{\mathbf{s}} = -\left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{q}}\right)^{+} \vec{\nabla}_{\mathbf{s}}^{\top} V_{\mathbf{s}}^{view} = -(\mathbf{L}_{\mathbf{p_a}} \mathbf{M} \mathbf{J}_{\mathbf{q}})^{+} \vec{\nabla}_{\mathbf{s}}^{\top} V_{\mathbf{s}}^{view} \tag{34}$$

where $\mathbf{M}$ and $\mathbf{J}_{\mathbf{q}}$ are the transformation matrices defined in (28), and $\mathbf{L}_{\mathbf{p_a}}$ is the well-known interaction matrix related to the image point $\mathbf{p_a} = (x_a, y_a)$.

V. Experiments and Results

Two different sets of experiments have been realized to outline the advantages of our method. The first set of experiments is a full-constraining positioning task with joint-limit avoidance. In the second set of experiments, the main task constrains only four DOF. Two DOF are available for an occlusion avoidance. In both experiments, the proposed approach frees up additionnal DOF that are used to perform a better avoidance and to complete the main task.

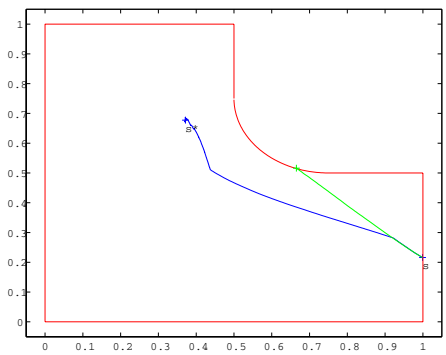

Fig. 4. Articular trajectories of the two first components of the articular vector. It corresponds mainly to the camera position in the plan XY. The joint limits are represented in red. The trajectory with the classical redundancy formalism is represented in green. It ends in the joint limits. The trajectory with the proposed method is in blue. The positioning task succeeds.

A. First experiment (six DOF constrained)

In the first experiment, the robot has to reach an unique position wrt. the visual target. The main task uses all the DOF of the robot. The projection operator computed using the classical redundancy formalism is null. Thanks to the choice of the adequate visual features, the trajectory of the camera is almost a straight line. As shown on Fig. 4, the articular domain of the robot is not convex. Without any avoidance law, the robot reaches its joint limits during the servo. Since there is no DOF left using the classical redundancy formalism, the main task fails when the robot reaches its joint limits as shown in Fig. 4 .

Using the method proposed above, the projection operator is not null as long as the error of the main task is not zero. Fig. 5 gives the rank of the projection matrix $\mathbf{P_z}$ during the execution. When the robot is near the joint limits, the projection operator is not null. The projected gradient is not null. This secondary control law is important enough to modify the trajectory imposed by the main task and to avoid the joint limits. Fig. 6 presents the evolution of three visual features whose value are modified by the secondary control law. The projection operator mainly accelerates the decreasing speed of the feature that controls the motion along the optical axis. Using the framework presented above, it is thus possible to free up some additional DOF that are not available within the classical redundancy formalism. The main task is correctly completed, and the servo is not slowed by the secondary control law.

B. Second experiment (four DOF constrained)

In the first set of experiments, no avoidance law can be taken into account by the classical redundancy formalism. It was thus easy to see that the performance of our framework is better. The next experiment will point out that, even when DOF are available for avoidance, a better behaviour of the robot can be obtained by considering a larger free space as done above.

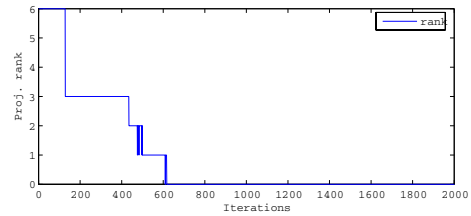

Fig. 5. Rank of the projection operator computed using the proposed approach during the servo, i.e. number of DOF left for the avoidance.

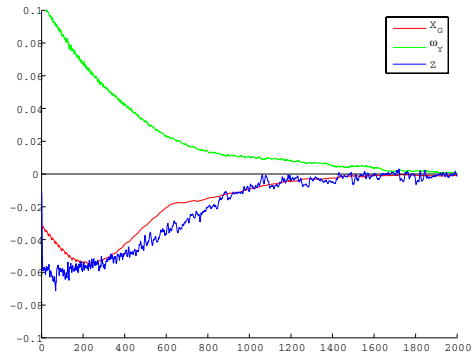

Fig. 6. Evolution of the visual features when applying the proposed method. The two features x_g and Z (plotted in red and blue) are modified by the avoidance law. On the opposite, feature s_y (plotted in green) is not modified

The main task is composed of four visual features. The robot has to move in order to center the object in the image, to rotate it properly around the optical axis, and to bring the camera at a distance of 1.5m of the target. Two DOF are thus available, that correspond mainly to motions on a sphere whose center is the target. During the servo, a second robot moves between the camera and the visual target, leading to an occlusion. The two available DOF are used to avoid this occlusion, as explained in Section IV-C.2.

Without any avoidance law, the visual target is quickly occluded, which makes the servo fail (Fig. 7-a). Using the classical redundancy formalism, the gradient is projected into a 2-dimensional space. Its norm is thus reduced, and the secondary control law is not fast enough to avoid the occlusion. Mainly, the projection forbids the motion along the optical axis, which is controled by one of the features of the main task. This motion is available using our approach (Fig. 8). The robot velocity is thus fast enough to avoid the occlusion (Fig. 7-c). The decreasing speed of some visual features has been accelerated to enlarge the free space of the first task (Fig 9). When the occlusion ends, the features decrease is no longer modified.

VI. Conclusion

In this paper, we have proposed a new general method to integrate a secondary term to a first task having priority. Our framework is based on a generalization of the classical redundancy formalism. We have shown that it is possible to

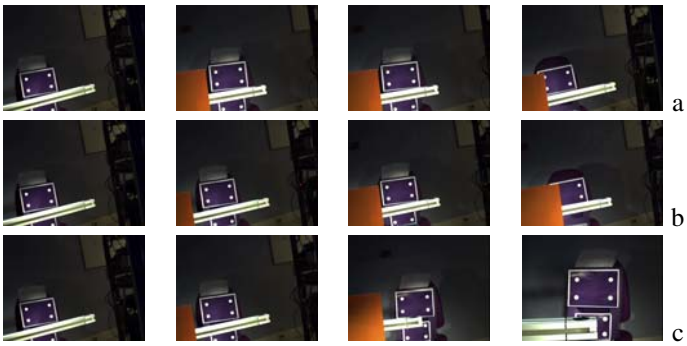

Fig. 7. Main phases of the servo without avoidance (a), with the avoidance law projected by the classical redundancy formalism (b) and with the control law projected by the proposed method (c). The pictures are taken by the embeded camera. The visual target is the four-white-points rectangle. The occluding object cared by a second robot is the orange shape that appears in the left of the image.

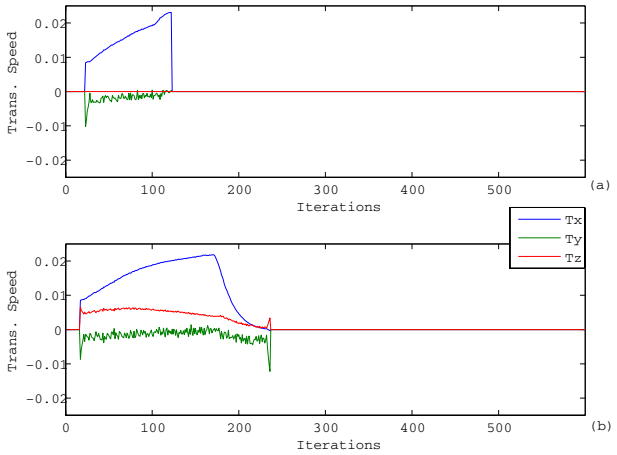

Fig. 8. Translational velocities of the avoidance control law projected using the classical redundancy formalism (a) and projected using the proposed approach (b). The motions along the camera axis (red) are not null using the proposed control law.

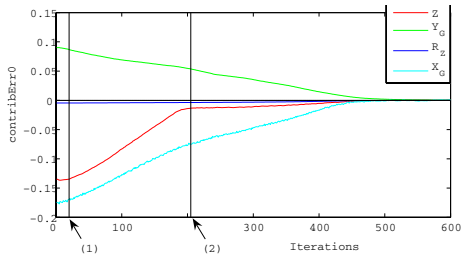

Fig. 9. Decreasing error of the visual features. The occlusion avoidance begins at Event 1. The decrease of the feature plotted in red is accelerated. The occlusion is completely avoided after Event 2. The decrease goes back to a normal exponential convergence.

enlarge the number of the available DOF, and thus to improve the performance of the avoidance control law. This control scheme has been validated on a 6-DOF eye-in-hand robotic platform. The robot has to position wrt. a visual target, and to avoid joint limits and occlusions.

We have shown that it is possible to find DOF during the accomplishment of a full-constraining task, and to enhance the avoidance process even when enough DOF are available.

Our current works aim at realizing a reactive servo, able to perform a full constraining task and simultaneously take into account the perturbations due to a real robotic system. The general idea is to free up as DOF as possible to perform the avoidance of any obstacle encountered during the servo. Using the method proposed in this article, additionnal DOF are collected at the very bottom level, directly in the control law. An avoidance can be realized even when the adequate DOF are already used by the main task. However, the number of DOF can be insufficient for example when the obstacles are numerous. We now focus on the choice of the main task, to obtain additionnal DOF by modifying the main task from a higher level.

References

[1] P. Baerlocher and R. Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 6(20), 2004.

[2] T.-F. Chang and R.-V. Dubey. A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE Trans. on Robotics and Automation*, 11(2):286–292, April 1995.

[3] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723, August 2004.

[4] F. Chaumette and E. Marchand. A redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing. *IEEE Trans. on Robotics and Automation*, 17(5):719–730, October 2001.

[5] N.J. Cowan, J.D. Weingarten, and D.E. Koditschek. Visual servoing via navigation functions. *IEEE Trans. on Robotics and Automation*, 18(4):521– 533, August 2002.

[6] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[7] G. Hager. Human-machine cooperative manipulation with vision-based motion constraints. *Workshop on visual servoing, iros02*, October 2002.

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

[9] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–98, 1986.

[10] A. Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. on Systems, Man and Cybernetics*, 7(12):868–871, December 1977.

[11] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, volume 3, pages 1988–1993, Leuven, Belgium, May 1998.

[12] Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549, August 2002.

[13] 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. Journal of Robotics Research*, 14(3):255–269, June 1995.

[14] J.B. Rosen. The gradient projection method for nonlinear programmimg, part i, linear constraints. *SIAM Journal of Applied Mathematics*, 8:181–217, 1960.

[15] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[16] O. Tahri and F. Chaumette. Image moment: generic descriptors for decoupled image-based visual servo. *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, April 2004.