Continuity of Varying-Feature-Set Control Laws

Nicolas Mansard, Anthony Remazeilles, François Chaumette

***Abstract*—Classical sensor-based control laws are based on the regulation of a set of features to a desired reference value. In this paper, we focus on the study of control laws whose feature set varies during the servo. In that case, we first show that the classical control laws that use an iterative least-square minimization are discontinuous. We then show that these discontinuities are due to the pseudo-inverse operator, which is not continuous at matrix rank change. To solve this problem, we propose a new inversion operator. This operator is equal to the classical pseudo-inverse operator in the continuous cases, and ensures the continuity everywhere. This operator is then used to build a new control law. This general control scheme is applied to visual servoing, in order to ensure the continuity of the control law when some visual features leave the camera field of view. The experiments prove the interest and the validity of our approach.**

***Index Terms*—Sensor-based control - velocity control - control continuity - linear algebra - least-square inverse - visual servoing - visibility constraint**

I. Introduction

A generic task function may be defined by a set of features computed from the sensor output that should be regulated to a desired value. Various control laws have been proposed to regulate such tasks to zero. For example, a generic approach to build stable control laws is proposed in [25]. Usually, the number and the type of features in the set are constant. The continuity of the control law and the stability of the system are then generally obtained outside some singularities that have to be avoided [21].

In this paper, we focus on task-based control schemes whose input set is not constant. Some works have already been proposed that consider servo schemes based on such a varying feature set. In [12], the features are removed from the set when they can not be computed anymore due to sensor-visibility lost. Similarly, the features detected as outliers are removed from the set in [8]. On the opposite, a feature can be removed when it is close enough from its desired value, in order to give more freedom to the robotic system [6], [24]. In [3], the system is controlled from learning-based joint control, and half-plane constraints are temporarily added to enforce situation-dependent constraints. Some specific features can also be added to enforce locally the system constraints, such as joint limits [23], occlusion [17] or obstacle avoidance [22].

In these papers, the properties of the obtained control scheme are studied on a case-by-case basis, in particular by using some hypotheses specific to the studied system. The results are thus difficult to generalize. In the following we thus propose to study such control laws as a generic control scheme called varying-feature-set control scheme. In particular, we will prove that the control laws computed directly from the varying feature set are not continuous in the general case. From this observation, a solution that ensures the continuity will be proposed.

Some classical control laws are firstly recalled and unified in Section II. Based on this unification, the varying-feature-set control scheme is defined in Section III. We then prove in Section IV that the classical control laws coming from this definition are unable to ensure the continuity everywhere. To remove these discontinuities, a new inverse operator is proposed in Section V. Some visual-servoing experiments finally presented in Section VI, as an experimental comparison of the classical control laws and the proposed solution.

II. State of the Art

In this section, we quickly recall some classical control laws whose feature set is varying during the servo. An effort is made here to homogenize the different notations used by each author.

A. Classical control law

Let us consider an error function $\mathbf{e}$ such that:

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

where $\mathbf{q}$ is the system configuration and $\mathbf{J} = \partial\mathbf{e}/\partial\mathbf{q}$ is the Jacobian of $\mathbf{e}$. The Jacobian $\mathbf{J}$ is a $k\times n$-matrix, where n is the number of degrees of freedom (DOF) of the system ($n = \dim\mathbf{q}$) and k is the size of the error function ($k = \dim\mathbf{e}$). The rank of $\mathbf{J}$ is denoted m. In (1), it is implicitly supposed that $\mathbf{e}(\mathbf{q})$ is derivable everywhere and does not depend on the time variable but through the configuration $\mathbf{q}$. A classical controller is then:

$$\dot{\mathbf{q}} = -\lambda\widehat{\mathbf{J}^{+}}\mathbf{e} \tag{2}$$

where λ is a positive parameter used as a gain to tune the convergence velocity, $\mathbf{A}^{+}$ stands for the pseudo inverse (or least-square inverse) of matrix $\mathbf{A}$ [2], and $\widehat{\mathbf{A}}$ is an approximation of $\mathbf{A}$.

Manuscript submitted October 15, 2007; revised Jully, 15, 2008 and accepted January, 29, 2009, under reference number 07-401.

At the time of writting, the authors were all with the INRIA Rennes, Campus de Beaulieu - 35 042 Rennes cedex, France (*nicolas.mansard@laas.fr, francois.chaumette@irisa.fr,anthony.remazeilles@gmail.com*)

Control law (2) ensures an exponential decrease of each component of $\mathbf{e}$ until regulation $\mathbf{e} = \mathbf{0}$ if $\mathbf{J}\widehat{\mathbf{J}^+} = \mathbf{I}_m$ (since in that case $\dot{\mathbf{e}} = -\lambda\mathbf{J}\widehat{\mathbf{J}^+}\mathbf{e} = -\lambda\mathbf{e}$). The global asymptotic stability can be obtained as soon as $\mathbf{J}$ and $\widehat{\mathbf{J}}$ are full-rank (*i.e.* $k = m$) and $\mathbf{J}\widehat{\mathbf{J}^+} > \mathbf{0}$ [25]. In all other cases (*i.e.* $k > m$), only the local asymptotic stability can generally be demonstrated. This classical control scheme has been widely used for sensor-based control [11], [26], [13], [19], [16].

In the following, all the computations will be realized using the hypothesis that $\mathbf{J}$ is perfectly known. We will show in the experiments that the obtained control scheme is robust to this hypothesis.

B. Evidence of discontinuities

We now underline the discontinuities that can occur when modifying the feature set. Let us consider a task $\mathbf{e_1}$, controlling m_1 of the n DOF of the system. At time t, we increase the task by adding a term $\mathbf{e_2}$, controlling m_2 DOF. The task is now $\begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \end{bmatrix}$. If the feature set is abruptly modified, is it possible to keep the control law continuity? If $m_1 < n$, then, obviously, the DOF not controlled by $\mathbf{e_1}$ and that are now controlled by $\mathbf{e_2}$ are subject to a discontinuity (passing from a zero control input to a non-zero control input). If $m_1 = n$, the problem is the same, whatever the value of m_2. Before time t, all the DOF of the system are controlled only by $\mathbf{e_1}$. After time t, a trade-off is realized by the pseudo inverse to fulfill at the same time $\mathbf{e_1}$ and $\mathbf{e_2}$, which produces a discontinuity in the general case.

For example, when realizing a visual servoing based on a multi-point target, the control law is different when considering a four-point target or a five-point target. Therefore, passing from four points to five points causes a discontinuity (an example of such a discontinuity is given in Section VI).

In the following, we will present several works where the classical control scheme (2) has been modified to take into account the varying dimension of the task $\mathbf{e}$ [8], [12], [24], [6]. We will emphasize the solutions that have been proposed in these articles to prevent this discontinuity to happen, and unify these works in an unique mathematical formulation. The common idea is to smooth the discontinuity that happens at feature activation and inactivation by introducing at the activation border a buffer area where the feature is partially active. This buffer area is defined by introducing a smooth activation function $\mathbf{H}$, as explained in the following paragraphs.

C. Some varying-feature-set control laws

1) Robust visual servoing: In [8], the problem of outliers in the input set is addressed by associating to each feature a weight computed from the confidence that this feature is not an outlier. The proposed error function is $\mathbf{e_q} = \mathbf{He}$, where $\mathbf{e} = (\mathbf{s} - \mathbf{s}^*)$ is the error between current and desired feature values, and $\mathbf{H} = \mathbf{Diag}(h_1, ..., h_k)$ is a weighting matrix used to remove the outliers from the feature set. The weights h_i are computed from a robust estimation algorithm. They vary from 1 when the robust estimation gives full confidence to 0 when the feature is doubtlessly an outlier. The derivative of the task $\mathbf{e_q}$ is $\dot{\mathbf{e}}_\mathbf{q} = \mathbf{H}\dot{\mathbf{e}} + \dot{\mathbf{H}}\mathbf{e}$. In [8], it is assumed that $\mathbf{H}$ is varying slowly. The second term of the derivative is thus neglected[1] and the control law is computed by analogy with (2):

$$\dot{\mathbf{q}} = -\lambda(\mathbf{HJ})^+\mathbf{He} \tag{3}$$

When the confidence in a feature decreases, it is smoothly removed from the control law by decreasing the value of the corresponding vector component $h_i e_i$. Simultaneously, it is also smoothly removed from the feature set by nullifying the corresponding line of the matrix $\mathbf{HJ}$. The weighting matrix $\mathbf{H}$ is thus used to smoothly remove or add a feature to the task. It will be proved in Section IV-C3 that this is enough to ensure the control law continuity as long as $(\mathbf{HJ})^+\mathbf{H}$ is full rank.

2) Continuous visual servoing despite changes of visibility: In [12], the authors directly address the problem of the control-law continuity when the number of features varies, in the particular case where this variation is due to visibility loss. When a feature leaves the camera field of view (fov), it has to be removed from the feature set, which thereof causes the control law to be discontinuous. As previously, the use of a weighting matrix enables to take care of these features leaving the fov. The error vector is written $\mathbf{e_q} = \mathbf{We}$ where $\mathbf{W} = \mathbf{Diag}(w_1, ..., w_k)$ is the weighting matrix used to smoothly remove a feature that is going to be non visible. The weight w_i is null when the feature is out of the fov, and is equal to 1 when the feature is at the center of the image frame.

It is possible to show [18] that the control law used in [12] is equivalent to:

$$\dot{\mathbf{q}} = -\lambda(\mathbf{HJ})^+\mathbf{He} \tag{4}$$

where $\mathbf{H} = \sqrt{\mathbf{W}}$ is a diagonal matrix whose coefficients vary continuously between 0 and 1. This control law is identical to the one obtained in [8].

3) Region reaching control: In [6] the main purpose is to bring the end-effector of the robot to a region instead of a point. The goal region is defined as the intersection of a set of k simple regions, each one being analytically defined by an inequality:

$$\forall i = 1..k, \quad e_i(\mathbf{X}) \leq 0 \tag{5}$$

where $\mathbf{X}$ is the position of the end effector in the Cartesian space. The control aims at reducing the value of the left part of each inequality until they are all negative. When an inequality is respected, the corresponding part of the control is stopped.

If the control law was developed to ensure $\mathbf{e} = \mathbf{0}$, the robot would be controlled toward the intersection of all the contours of the regions (5). In order to bring the robotic system inside this region, the task vector is defined to be $\mathbf{e_q} = \mathbf{H}(\frac{1}{2}{e_1}^2, ..., \frac{1}{2}{e_k}^2)$, where $\mathbf{H}$ is a diagonal matrix whose components are null if the corresponding regions have been reached, and equal to 1 otherwise. Using these notations, it

[1] Neglecting $\dot{\mathbf{H}}$ is a classical approximation that is also found in [12], [6], [24]. In practice, when $\mathbf{H}$ and $\mathbf{e}$ decrease in the same direction (*i.e.* when the gradient of both $\mathbf{e}$ and $\mathbf{diag}(\mathbf{H})$ are in the same half-space, which is typically the case when the minimum of $\mathbf{e}$ is inside the area defined by $\mathbf{H} = 0$, as done in [6], Section II-C3 or in [24], Section II-C4) this approximation does not disturb the stability of the control.

can be shown [18] that the control law proposed in [6] is equivalent to:

$$\dot{\mathbf{q}} = -\lambda(\mathbf{HJ})^{\top}\mathbf{He} \tag{6}$$

The form is similar to the two previous control laws, but with the transpose $(\mathbf{HJ})^{\top}$ instead of the pseudo inverse $(\mathbf{HJ})^{+}$.

4) Qualitative servoing: The main objective of the qualitative servo control proposed in [24] is to enlarge the convergence area, by explicitly requiring that the error $\mathbf{e}$ converges toward a confident interval, instead of a particular desired value as it is usually performed. In this sense, the system is thus required to perform a *qualitative convergence*.

The error function $\mathbf{e_q}$ is defined by $\mathbf{e_q} = \mathbf{H}(\mathbf{e}-\bar{\mathbf{e}})$, where $\bar{\mathbf{e}}$ is the limit of the convergence area: when a component of $\mathbf{e}$ is below its corresponding threshold $\bar{\mathbf{e}}$, the associated part of the control is inactivated. The activation matrix $\mathbf{H}$ is a diagonal matrix whose coefficients vary continuously between 0 to 1 as the system enters or leaves the convergence area.

The control law that regulates the error $\mathbf{e}$ into the confidence interval proposed in [24] is:

$$\dot{\mathbf{q}} = -\lambda(\mathbf{HJ})^{+}\mathbf{H}(\mathbf{e}-\bar{\mathbf{e}}) \tag{7}$$

Once more, we recognize the same form of the previous control laws.

D. Synthesis

Several control laws that deal with varying feature set have been presented in the previous subsections. All have been written using an equivalent framework, with similar notations, as a matter of comparison. A common control law equation can be noticed:

$$\dot{\mathbf{q}} = -\lambda(\mathbf{HJ})^{\boxplus}\mathbf{He} \tag{8}$$

where $^{\boxplus}$ is a matrix operator (the pseudo inverse or the transpose).

The control law is a composition of three parts: the matrix $\mathbf{HJ}$, the matrix operator $^{\boxplus}$, and the vector $\mathbf{He}$. The diagonal activation matrix $\mathbf{H}$ is used to smoothly remove or add features of $\mathbf{e}$ and also to nullify the corresponding lines of the Jacobian matrix. When $^{\boxplus}$ is the pseudo inverse, this second point is fundamental. Indeed, if the Jacobian line is not nullified (*i.e.* if $\dot{\mathbf{q}} = -\lambda\mathbf{J}^{+}\mathbf{He}$ is used), then the feature is taken into account into the least-square minimization, and the control law tries to minimize the motion of the inactivated feature, by imposing the velocity $\dot{e}_i = 0$ (which is a control in itself, and the result is very different to not constrain $\dot{e}_i$ at all).

To ensure the continuity of the control law, the simplest solution is to ensure the continuity of each of the components. In particular, it is not sufficient to ensure the continuity of $\mathbf{HJ}$ and $\mathbf{He}$ if the matrix operator $^{\boxplus}$ is not continuous. Thanks to a correct definition of $\mathbf{H}$, both the task vector and the Jacobian matrix are continuous at feature activation or inactivation. To ensure the control law continuity, it is then enough to ensure the continuity of the matrix operator $^{\boxplus}$. The pseudo inverse operator is continuous when the rank of the matrix is constant. It is very important to notice that this is the basic hypothesis in the three control laws presented above. This is a sufficient condition to ensure the continuity of the control law.

However, the pseudo inverse operator is not continuous at rank change. This means that these control laws are not continuous if the number of features decrease below a certain level (which is not considered in [8], [12], [24]). In [6], this case can happen. The continuity of the control law is then obtained by using the transpose operator instead of the pseudo-inverse operator (the transpose is always continuous, even at matrix-rank change). However, using the transpose can lead to a very non-optimal control, and the pseudo inverse is often a much more efficient solution [11].

In the following sections, the generic control law (8) will be proved to be continuous as long as the number of active features is sufficient (in a sense that will be defined precisely in the following). It will also be shown that strong discontinuities can appear when the number of active features is not sufficient. Based on this observation, we will build a new matrix operator that is continuous in all cases, and acts like the pseudo inverse outside of its discontinuities. From this new operator, a control law with a similar form will be proposed and proved to be continuous in all cases.

III. Definitions

In this section, we define all the notions that are required for the following study. We firstly propose a global definition to refer to the tasks whose form is similar to those presented in the previous section. Then we propose formal definitions to characterize some classical notions of the redundancy of a system with respect to a given task.

A. Varying-feature-set task

Definition 3.1 (Varying-feature-set task): Let $\mathbf{e}$ be any feature vector which is called task in the following. Its Jacobian is supposed to be of constant rank. The task $\mathbf{e_q}$ is a *varying-feature-set task* based on $\mathbf{e}$ if it respects:

$$\mathbf{e_q} = \mathbf{He} \tag{9}$$

where $\mathbf{H}$ is a diagonal matrix whose coefficients continuously vary within the interval $[0,1]$.

Remark 3.1: The four control schemes (3), (4), (6) and (7) recalled in the previous section are based on a varying-feature-set task.

B. Input redundancy and decoupling

Definition 3.2 (Full-rank matrix): The matrix $\mathbf{A}$ is full row rank (FRR) *iff* the number of its rows is equal to its rank. It is full-column rank (FCR) *iff* the number of its columns is equal to its rank.

Definition 3.3 (Non-redundant input): Let $\mathbf{e}$ be any task. The task $\mathbf{e}$ is said to be non redundant in input (or to have a non-redundant input) if its Jacobian matrix is FRR.

Definition 3.4 (Redundant input): On the opposite, a task has a redundant input if its Jacobian is not FRR.

Remark 3.2: If the matrix $\mathbf{J}$ is not FRR, it is possible to separate its lines into a generator set $\mathbf{J_0}$ and a redundant set $\mathbf{J_1}$ that can be defined as a linear combination of $\mathbf{J_0}$: $\mathbf{J_1} = \chi\mathbf{J_0}$. The couple $(\mathbf{J_0}, \chi)$ is called *factorization*. Its formal definition is the following.

Definition 3.5 (Matrix factorization): Let $\mathbf{J}$ be a non-FRR matrix. Let $\mathbf{P}$ be a permutation matrix, $\mathbf{J_0}$ a FRR matrix and χ a matrix such that:

$$\mathbf{J} = \mathbf{P}\begin{bmatrix} \mathbf{J_0} \\ \chi\mathbf{J_0} \end{bmatrix}, \tag{10}$$

then the set $\big(\mathbf{P}, \mathbf{J_0}, \chi\big)$ is called *factorization* of the matrix $\mathbf{J}$ by $\mathbf{J_0}$. $\mathbf{J_0}$ is the *generator* matrix of $\mathbf{J}$, and χ is the *multiplier* of the factorization. In order to simplify notations, the permutation $\mathbf{P}$ will be often omitted. Thereby, a factorization of $\mathbf{J}$ is denoted $\big(\mathbf{J_0}, \chi\big)$ ($\mathbf{P}$ is easily deduced from $\mathbf{J}, \mathbf{J_0}$ and χ).

Remark 3.3: If some columns of the multiplier χ are null, the factorization can be developed:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J_A} \\ \mathbf{J_B} \\ \chi_\mathbf{B}\mathbf{J_B} \end{bmatrix} \tag{11}$$

with $\mathbf{J_0} = \begin{bmatrix} \mathbf{J_A} \\ \mathbf{J_B} \end{bmatrix}$ and $\chi = [\mathbf{0}\ \chi_\mathbf{B}]$. Features corresponding to $\mathbf{J_B}$ and $\chi_\mathbf{B}\mathbf{J_B}$ are the redundant part of the input vector. $\mathbf{J_A}$ corresponds to the non-redundant part, because none of the features can be defined as a linear combination of the features associated to $\mathbf{J_A}$. On the opposite, if the multiplier χ does not have any null column, then the factorization is said to be fully redundant.

Definition 3.6 (Full-redundant input): The task $\mathbf{e}$ has a full-redundant input if all partitions of its Jacobian $\mathbf{J}$ of the form (11) result in $\mathbf{J_A} = \mathbf{0}$.

Corollary 3.1 (Characterization of a full-redundant input): The task $\mathbf{e}$ has a full-redundant input *iff* its Jacobian $\mathbf{J}$ can be written $\mathbf{J} = \mathbf{P}\begin{bmatrix} \mathbf{J_0} \\ \chi\mathbf{J_0} \end{bmatrix}$, where $\mathbf{P}$ is a permutation matrix, $\mathbf{J_0}$ is FRR and none of the columns of the multiplier χ is null.

Proof: The proof is given is [18]. ■

Definition 3.7 (Decoupled input feature): Let $\mathbf{e}$ be a feature set. The feature e_2 is distinguished[2] from the other features denoted $\mathbf{e_1}$. Let $\mathbf{J_1}$ and $\mathbf{J_2}$ be the Jacobians of $\mathbf{e_1}$ and e_2 respectively. The feature e_2 is said decoupled from the other features $\mathbf{e_1}$ of $\mathbf{e}$ if:

$$R(\mathbf{J_1^+}) \perp R(\mathbf{J_2^+}) \tag{12}$$

where $R(\mathbf{A})$ is the range of matrix $\mathbf{A}$.

Corollary 3.2: Two feature sets $\mathbf{e_1}$ and $\mathbf{e_2}$ are decoupled *iff*:

$$\mathbf{J_2}\mathbf{J_1^+} = \mathbf{0} \tag{13a}$$

and

$$\mathbf{J_1}\mathbf{J_2^+} = \mathbf{0} \tag{13b}$$

Proof: A well-known result concerning the kernel and the range of a matrix $\mathbf{A}$ is:

$$R(\mathbf{A}^\top) = N(\mathbf{A})^\perp \tag{14}$$

where $N(\mathbf{A})$ is the kernel of $\mathbf{A}$, and $\mathbf{E}^\perp$ is the orthogonal complementary of subspace $\mathbf{E}$. Using (12), we obtain $R(\mathbf{J_1}^\top) \subset N(\mathbf{J_2})$. Since $R(\mathbf{J_1}^\top) = R(\mathbf{J_1}^+)$, this proves (13a).

Reciprocally, if (13a) is true, then $R(\mathbf{J_1}^\top) = R(\mathbf{J_1}^+) \subset N(\mathbf{J_2})$. Since $R(\mathbf{J_2}^+) = R(\mathbf{J_2}^\top) = N(\mathbf{J_2})^\perp$, then $R(\mathbf{J_1^+})$ is orthogonal to $R(\mathbf{J_2^+})$. The dual equation is obtained by the same way. ■

Remark 3.4: If all the features of $\mathbf{e}$ are decoupled, then it is easy to show that $\mathbf{e}$ has a non-redundant input. Moreover, if one feature is decoupled, $\mathbf{e}$ can not have a full-redundant input.

Intuitively, the behavior of the control law when progressively inactivating a feature will differ if a redundant feature, a non-redundant or a decoupled one is considered. The four characterizations of the features that have been defined upper (redundant, non redundant, full redundant and decoupled) can be enlarged to the varying-feature-set tasks.

Definition 3.8 (Charasteristics of a varying-feature-set task): Let $\mathbf{e_q}$ be a varying feature set. The corresponding active task $\mathbf{e_A}$ is constructed by considering only the input features whose weight in $\mathbf{H}$ is not null. Furthermore:

- The varying-feature-set task $\mathbf{e_q}$ has a non-redundant active input if the associate active task $\mathbf{e_A}$ has non-redundant input.
- The varying-feature-set task $\mathbf{e_q}$ has a redundant active input if the associate active task $\mathbf{e_A}$ has a redundant input.
- The varying-feature-set task $\mathbf{e_q}$ has a full-redundant active input if the associate active task $\mathbf{e_A}$ has a full-redundant input.
- An active feature e_2 is decoupled from the other active features if e_2 is decoupled from the other features belonging to the associate active task $\mathbf{e_A}$.

Using these definitions, we will now study the control laws based on varying feature set (such as those recalled in Section II) in the general case.

IV. Varying-feature-set control scheme

This section considers the continuity of different control laws derived from Definition 3.1, and presented in Section IV-A. It will be shown in Sections IV-B to IV-D that when the Jacobian of the task is not fully redundant, none of these control laws is continuous.

A. Control laws based on a varying-feature-set task

Let $\mathbf{e_q}$ be a task characterized by a varying feature set such that $\mathbf{e_q} = \mathbf{He}$. Its derivative is:

$$\dot{\mathbf{e}}_\mathbf{q} = \mathbf{H}\dot{\mathbf{e}} + \dot{\mathbf{H}}\mathbf{e} \tag{15}$$

As seen above, it is usually considered that $\mathbf{H}$ varies slowly. With the approximation $\dot{\mathbf{H}} = \mathbf{0}$, we obtain the simple expression $\mathbf{J_{e_q}} = \mathbf{HJ}$ where $\mathbf{J}$ and $\mathbf{J_{e_q}}$ are respectively the Jacobians of $\mathbf{e}$ and $\mathbf{e_q}$.

Knowing $\mathbf{J_{e_q}}$, it is possible to apply directly the classical control scheme (2), obtaining thus:

$$\dot{\mathbf{q}} = -\lambda\widehat{\mathbf{J}^+_{\mathbf{e_q}}}\mathbf{e_q} = -\lambda\widehat{\big(\mathbf{HJ}\big)^+}\mathbf{He} \tag{16}$$

[2] Denoted with bold font are the vectors, *e.g.* $\mathbf{e_1}$ and matrices, *e.g.* $\mathbf{J}$, and with non-bold font the scalar variables, *e.g.* e_2.

Several choices can be considered for $\widehat{\mathbf{J}^+_{\mathbf{e_q}}}$. Apart from the classical approximation of $\widehat{\mathbf{J}}$ (see [15] for a review), we will focus on five possible choices concerning $\mathbf{H}$, in order to try to get the global continuity of the control law:

$$\dot{\mathbf{q}} = -\lambda\big(\widehat{\mathbf{H}}\mathbf{J}\big)^+\widehat{\mathbf{H}}\mathbf{e} \tag{17}$$

$$\dot{\mathbf{q}} = -\lambda\big(\mathbf{H}\mathbf{J}\big)^+\mathbf{H}\mathbf{e} \tag{18}$$

$$\dot{\mathbf{q}} = -\lambda\big(\mathbf{H}\mathbf{J}\big)^\dagger\mathbf{H}\mathbf{e} \tag{19}$$

$$\dot{\mathbf{q}} = -\lambda\big(\widehat{\mathbf{H}}\mathbf{J}\big)^+\mathbf{H}\mathbf{e} \tag{20}$$

$$\dot{\mathbf{q}} = -\lambda\big(\mathbf{H}\mathbf{J}\big)^+\widehat{\mathbf{H}}\mathbf{e} \tag{21}$$

$\widehat{\mathbf{H}}$ is an approximation of $\mathbf{H}$ defined as: $\widehat{\mathbf{H}} = \mathbf{Diag}\Big(\begin{cases} 1 & \text{if } h_i \neq 0 \\ 0 & \text{otherwise}\end{cases}\Big)$, and $\mathbf{A}^\dagger$ is the damped least square inverse of $\mathbf{A}$ [20], [9] (the interest of this inverse operator will be given in Section IV-D). It is trivial to obtain (18), (19) and (20) from the general relation (16). Since $\mathbf{H}^+\mathbf{H} = \widehat{\mathbf{H}}$, Eq. (17) and (21) are respectively obtained by approximating $\widehat{\mathbf{J}^+_{\mathbf{e_q}}}$ by $\big(\widehat{\mathbf{H}}\mathbf{J}\big)^+\mathbf{H}^+$ and $\big(\mathbf{H}\mathbf{J}\big)^+\mathbf{H}^+$. Even if these two derivations do not seem to be intuitive, their final formulations correspond to easily understandable situations, e.g. a full approximation and a partial one. In particular, the use of both $\mathbf{H}$ and $\widehat{\mathbf{H}}$ in (20) and (21) is explained in Section IV-E.

The following sections study the behavior of these different control laws, and especially their continuity at Jacobian-rank change.

B. Full approximation (17)

This first control law corresponds to the naive way to consider a task with a varying feature set: a component getting inside the activation area is directly considered within the minimization scheme, without any progressive activation. Of course, this kind of control law is not continuous [12], as explained in Section II-B. The importance of the discontinuity depends on the value e_2 and also on the way the addition of the line $\mathbf{J_2}$ modifies the singular values of the Jacobian.

C. No approximation (18)

To solve the discontinuity of (17), a logical solution is to use an activation matrix $\mathbf{H}$. When a feature gets inside the activation area, it is thus progressively (or smoothly) added within the control scheme, until full activation. The control law (18) can be found in [12], [8], [24], as presented in Section II.

This section shows that this control law is continuous as long as enough features are activated so that the input is fully redundant. It is also shown that the smoothness brought by $\mathbf{H}$ is not effective when the number of active features is not sufficient. More precisely, it will be shown that control laws (18) and (17) are surprisingly equivalent when the task is non redundant. Three cases are separately studied, whether the task is not redundant, redundant or fully redundant.

1) Non-redundant input signal:

Theorem 4.1: Let $\mathbf{e_q}$ be a varying-feature-set task whose active input is non redundant. The two control laws (17) and (18) are equal.

Proof: Let us first introduce another inverse of matrix $\mathbf{A}$, the generalized inverse [2]. It has been introduced in linear-feedback control in [27], and widely used since [4], [1]. A very good analysis of such an inverse can be found in [10]. Let $\mathbf{W}$ be a full-rank square matrix. The weighted generalized inverse matrix of $\mathbf{A}$ weighted on the left by the weights $\mathbf{W}$ is defined to be [10]:

$$\mathbf{A}^{\mathbf{W}\#} = (\mathbf{W}\mathbf{A})^+\mathbf{W} \tag{22}$$

The (full-rank) weight matrix $\mathbf{H}_f$ is defined from $\mathbf{H}$ by:

$$\mathbf{H}_f = \mathbf{Diag}\Big(\begin{cases} h_i & \text{if } h_i \neq 0 \\ 1 & \text{otherwise}\end{cases}\Big) \tag{23}$$

Using this definition, we can write:

$$(\mathbf{H}\mathbf{J})^+\mathbf{H} = (\mathbf{H}_f\widehat{\mathbf{H}}\mathbf{J})^+\mathbf{H}_f\widehat{\mathbf{H}} = (\widehat{\mathbf{H}}\mathbf{J})^{\mathbf{H}_f\#}\widehat{\mathbf{H}} \tag{24}$$

By a simple feature reordering, we suppose that the Jacobian can be written:

$$\mathbf{H}\mathbf{J} = \begin{bmatrix} \mathbf{H}_1\mathbf{J_1} \\ \mathbf{0} \end{bmatrix} \tag{25}$$

where $\mathbf{J_1}$ is FRR since the active input of $\mathbf{e_q}$ is non redundant. One of the major results of [10] is to prove that the weighted inverse (22) is invariant to the choice of $\mathbf{W}$ if $\mathbf{A}$ is FRR. Since $\begin{bmatrix} \mathbf{A} \\ \mathbf{0} \end{bmatrix}^+ = [\mathbf{A}^+\mathbf{0}]$, this result can easily be generalized to the case (25). Thus, since $\mathbf{H}_1\mathbf{J_1}$ is FRR ($\mathbf{J_1}$ is FRR and $\mathbf{H}_1$ is invertible), it is possible to write using (24):

$$(\mathbf{H}\mathbf{J})^+\mathbf{H} = (\widehat{\mathbf{H}}\mathbf{J})^{\mathbf{H}_f\#}\widehat{\mathbf{H}} = (\widehat{\mathbf{H}}\mathbf{J})^+\widehat{\mathbf{H}} \tag{26}$$

This last result proves that if the active input features are non redundant, the two control laws (17) and (18) are equal. If the active input is not redundant, the weights of $\mathbf{H}$ are not taken into account. The discontinuities are thus the same than when using the simple matrix $\widehat{\mathbf{H}}$. ■

2) Redundant input signal: The previous result can be easily extended to the case of a redundant input as long as the input is not fully redundant.

Theorem 4.2 (Weighted least square invariance): Let $\mathbf{J}$ be any matrix, and $\mathbf{W}$ be a diagonal and invertible weight matrix. Let $\mathbf{J}$ be a factorization such that:

$$\mathbf{J} = \mathbf{P}\begin{bmatrix} \mathbf{J_0} \\ \mathbf{J_1} \\ \chi_\mathbf{1}\mathbf{J_1} \end{bmatrix} \tag{27}$$

where $(\mathbf{J_0}, \mathbf{J_1})$ is FRR. The same factorization can also be applied to the weighting matrix $\mathbf{W} = \mathbf{P}\ \mathbf{Diag}\big(\mathbf{W_0}, \mathbf{W_1}, \mathbf{W_2}\big)$.

Then the non-redundant part $\mathbf{J_0}$ of $\mathbf{J}$ is invariant to the weights $\mathbf{W}$ when computing the weighted general inverse $\mathbf{J}^{\mathbf{W}\#}$:

$$\mathbf{J}^{\mathbf{W}\#} = \mathbf{J}^{\widetilde{\mathbf{W}}\#} \tag{28}$$

where $\widetilde{\mathbf{W}} = \mathbf{P}\ \mathbf{Diag}(\mathbf{I}, \mathbf{W_1}, \mathbf{W_2})$.

This result can be proved by using a full-rank decomposition of the Jacobian $\mathbf{J}$. The demonstration is technical and fastidious, and is thus not given here. The interested reader is referred to [18] for additional details.

Using the Theorem 4.2, the following corollary is immediate.

Corollary 4.1: Let $\mathbf{e_q}$ be a varying-feature-set task whose active input is redundant but not fully redundant. Then the non-zero weights h_i corresponding to the non-redundant features are not taken into account in the control law (18), that is to say a *partially* active feature $h \neq 0$ is taken into account as a fully active feature $h = 1$.

Let us consider a task $\mathbf{e_q}$ whose first feature is non redundant. The activation matrix is $\mathbf{H} = \begin{bmatrix} h & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix}$. Using Corollary 4.1, we can write if h is not null:

$$\begin{aligned} (\mathbf{HJ})^+\mathbf{H} &= (\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix}\mathbf{J})^+ \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix} \\ &\underset{h\to 0}{\longrightarrow} (\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix}\mathbf{J})^+ \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix} \end{aligned} \tag{29}$$

However, if h is null:

$$(\mathbf{HJ})^+\mathbf{H} = (\begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix}\mathbf{J})^+ \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix} \tag{30}$$

The matrix $(\mathbf{HJ})^+\mathbf{H}$ is not continuous when $h \to 0$. The control law is thus not continuous when some features that are non redundant are inactivated.

3) Full-redundant input signal: On the opposite, it is easy to show that (18) is continuous when the activated input is fully redundant. In this case, the activation or inactivation of any feature will not modify the rank of $\mathbf{HJ}$. The pseudo-inverse operator is continuous when the rank of the matrix is constant [2]. Since $\mathbf{HJ}$ is continuous, this proves that the control law is continuous.

In conclusion, the behavior of the system controlled by (18) is continuous when enough features are activated so that the input is fully redundant, and discontinuous when the input is only redundant or non redundant. The discontinuity of the classical solution (18) is due to to the discontinuity of the pseudo-inverse operator when the rank of the Jacobian matrix changes [2].

D. Using the damped least square inverse (19)

To compensate the discontinuities of the pseudo-inverse operator, it has been proposed to use the damped-least-square inverse [20], [14], [9]. This operator is defined by:

$$\mathbf{A}^{\eta\dagger} = \mathbf{V}\boldsymbol{\Sigma}^{\eta\dagger}\mathbf{U}^\top \tag{31}$$

where $(\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V})$ is the singular-value decomposition (SVD) of $\mathbf{A}$ and $\boldsymbol{\Sigma}^{\eta\dagger}$ is a diagonal matrix whose coefficients $\sigma_i^{\eta\dagger}$ are computed from the singular values σ_i by:

$$\sigma_i^{\eta\dagger} = \frac{\sigma_i}{\sigma_i^2 + \eta^2} \tag{32}$$

Theoretically, the introduction of η does not bring any new parameter to tune, since it only replaces a threshold used when computing the pseudo inverse to bound the singular-value inversion. We thus denote $\mathbf{A}^{\eta\dagger} = \mathbf{A}^\dagger$ in the following.

This operator can be used instead of the pseudo inversion in the classical control laws (2). As proposed in [7], the damped least square inverse reduces the effect of the control singularities. On the opposite, it also reduces the precision of the control. In particular, when the damped least squares are used with (2), the effect is not the nice decoupled decrease $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ since $\mathbf{J}\mathbf{J}^\dagger$ is not the identity. The higher the value of η, the better the smoothing, but the more the reference behaviour will be disturbed.

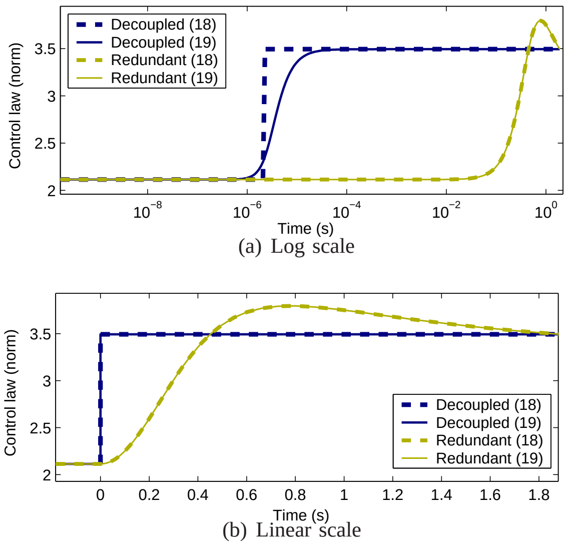
In the control law (19), the damping factor acts as a smoothing of the discontinuity. The smoothing is effective at rank changes (when a singular value passes from non zero to zero), that is to say at activation or inactivation of a non-redundant feature. In theory, the introduction of the damping factor thus solves the problem of the discontinuity encountered in (18). However this is not the case in practice. Practically, the damping is only effective around the threshold η.

Let us consider the activation of a non-redundant feature from $\mathbf{H}_{h=0} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$ to $\mathbf{H}_{h=1} = \mathbf{I}$. The matrix $(\mathbf{HJ})^\dagger\mathbf{H}$ is numerically equal to $(\mathbf{H}_0\mathbf{J})^\dagger\mathbf{H}_0$ when the corresponding singular value is very small compared to η, typically in $[0, 1e^{-3}\eta]$. It is numerically equal to $(\mathbf{H}_1\mathbf{J})^\dagger\mathbf{H}_1 = \mathbf{J}^+$ when the singular value is very large compared to η, typically in $[1e^{+3}\eta, 1]$. In practice, the matrix $(\mathbf{HJ})^\dagger\mathbf{H}$ passes from $(\mathbf{H}_0\mathbf{J})^\dagger\mathbf{H}_0$ to $\mathbf{J}^+$ in the very small interval $[1e^{-3}\eta,\ 1e^{3}\eta]$. For example, if $\eta = 1e^{-6}$ as classically done, the smoothing is effective into the interval $[1e^{-9}, 1e^{-3}]$, that is to say on an interval whose length is only 10^{-3}. On the opposite, when removing a redundant feature, the variation of $(\mathbf{HJ})^\dagger\mathbf{H}$ is smooth into $[0, 1]$, whose length is 1. The smoothing is effective on a large interval. The comparison between the smoothness is shown on Fig. 1. As shown by Fig. 1-(a), the control law (19) is smooth in theory. However, as shown by Fig. 1-(b), both control laws (18) and (19) are equivalent in practice, and discontinuous.

The direct consequence is that the damping parameter η has to be set very high which in turn reduces significally the performances of the control. The damped least squares are thus not a relevant solution to smooth the irregularities of the varying feature set.

E. Partial approximations (20) and (21)

We have shown in the previous sections that the classical control laws (17) and (18) are unable to ensure the continuity of the control law when the input is not fully redundant. These discontinuities can be understood naively by writing that $(\mathbf{HJ})^+\mathbf{H} = \mathbf{J}^+\mathbf{H}^+\mathbf{H} = \mathbf{J}^+$ (if $\mathbf{H}$ is invertible and $\mathbf{J}$ is FRR). The use of $\mathbf{H}$ both inside and outside the pseudo-inverse operator induces intuitively a simplification that cancels the smoothing. To prevent this simplification, a logical proposition is to use the true activation matrix $\mathbf{H}$ only once and its approximation $\widehat{\mathbf{H}}$ elsewhere. Two possibilities arise then: we can use the approximation $\widehat{\mathbf{H}}$ inside the pseudo inverse and use the exact matrix $\mathbf{H}$ outside of the inverse, in factor of the task $\mathbf{e}$ (this solution corresponds to (20)) ; or we can use $\mathbf{H}$ inside the inverse, and $\widehat{\mathbf{H}}$ outside (this gives then (21)).



(a) Log scale

(b) Linear scale

Fig. 1. Comparison between the smoothness of $(\mathbf{HJ})^+\mathbf{H}$ (control law (18)) and $(\mathbf{HJ})^\dagger\mathbf{H}$ (control law (19)) when adding a non-redundant feature, and when adding a redundant feature. The graph is shown twice, with log X-scale (a), and with linear X-scale (b). Parameter η is set to $1e^{-6}$.

Control law (20) smoothly nullifies the feature values that are getting close to the activation frontier, but abruptly removes the corresponding part of the Jacobian matrix. The Jacobian matrix is thus not continuous, but it is hoped to obtain a continuous control law by correcting the discontinuities with smoothed input features. The following theorem proves that the continuity is obtained only in the very particular case of a perfect decoupling.

Theorem 4.3: Let $\mathbf{e_q}$ be a varying-feature-set task based on $\mathbf{e}$. Control law (20) is continuous at feature activation or inactivation *iff* the activated features are decoupled from the other ones.

Proof: Let us first consider a task $\mathbf{e}$ where all the features but one are fully activated. This last feature is denoted e_2. Let $\mathbf{J_2}$ be its Jacobian. The Jacobian of the task $\mathbf{e}$ can then be written $\mathbf{J} = \begin{bmatrix} \mathbf{J_1} \\ \mathbf{J_2} \end{bmatrix}$, where $\mathbf{J_2}$ is single line. Since all the features corresponding to $\mathbf{J_1}$ are fully activated, the activation matrix can be written:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & h \end{bmatrix} \tag{33}$$

The pseudo inverse of $\mathbf{J}$ can be decomposed using a *divide and conquer* approach:

$$\mathbf{J}^+ = \begin{bmatrix} \mathbf{J_1} \\ \mathbf{J_2} \end{bmatrix}^+ = \left[\mathbf{J_1^+}\ \mathbf{J_2^+}\right] + \mathbf{X_{12}} \tag{34}$$

where $[\mathbf{J_1^+ J_2^+}]$ is the least-square minimization of each Jacobian $\mathbf{J_1}$ and $\mathbf{J_2}$ taken separately, and $\mathbf{X_{12}}$ is the part of the least squares corresponding to the coupling between the two Jacobians. To simplify the notations in the following, $\mathbf{X_{12}}$ is written $\mathbf{X_{12}} = \left[\mathbf{X_{12}^1}\ \mathbf{X_{12}^2}\right]$, where $\mathbf{X_{12}^2}$ is simple column.

Using this decomposition and (33), the inverse used in (20) can be written:

$$(\widehat{\mathbf{H}}\mathbf{J})^+\mathbf{H} = \begin{cases} \left[\mathbf{J_1^+}\ h\mathbf{J_2^+}\right] + \left[\mathbf{X_{12}^1}\ h\mathbf{X_{12}^2}\right] & \text{if } h \neq 0 \\ \left[\mathbf{J_1^+}\ \mathbf{0}\right] & \text{if } h = 0 \end{cases} \tag{35}$$

The discontinuity clearly appears in this formulation since:

$$\lim_{h\to 0}\left((\widehat{\mathbf{H}}\mathbf{J})^+\mathbf{H}\right) = \left[\mathbf{J_1^+}\ \mathbf{0}\right] + \left[\mathbf{X_{12}^1}\ \mathbf{0}\right] \tag{36}$$

which is different from $\left[\mathbf{J_1^+}\ \mathbf{0}\right]$ when $h = 0$ and if $\mathbf{X_{12}}$ is not null. To prove the equivalence between decoupling and continuity, we thus just have to prove that $\mathbf{X_{12}}$ is null *iff* e_2 is decoupled. Let us suppose that $\mathbf{X_{12}} = \mathbf{0}$, that is to say $\mathbf{J}^+ = [\mathbf{J_1^+ J_2^+}]$. The definition of the pseudo inverse gives $\mathbf{J}^+\mathbf{J}\mathbf{J}^+ = \mathbf{J}^+$. By developing the value of $\mathbf{J}$ in this equality, we obtain $\mathbf{J_2^+ J_2 J_1^+} = \mathbf{0}$ and finally $\mathbf{J_2 J_2^+ J_2 J_1^+} = \mathbf{J_2 J_1^+} = \mathbf{0}$. We obtain similarly $\mathbf{J_1 J_2^+} = \mathbf{0}$. Using Corollary 3.2, this proves that $\mathbf{e_1}$ and e_2 are decoupled. Reciprocally, if e_2 is decoupled, then $\mathbf{J}^+ = (\mathbf{J_1^+ J_1} + \mathbf{J_2^+ J_2})[\mathbf{J_1^+ J_2^+}]$ [2]. Corollary 3.2 finally proves that $\mathbf{J}^+ = [\mathbf{J_1^+ J_2^+}]$ and thus $\mathbf{X_{12}} = \mathbf{0}$. ■

In fact, (20) manages to smooth the part of the control corresponding to the minimization of e_2. However, it is unable to smooth the part $\mathbf{X_{12}^1}$ of the control that corresponds to the coupling between e_2 and the other features. When h is not null, the space corresponding to the coupling matrix $\mathbf{X_{12}^1}$ is used as a trade-off to realize both $\mathbf{e_1}$ and e_2. As soon as h really becomes null, $\mathbf{X_{12}^1}$ becomes instantaneously available for $\mathbf{e_1}$ alone, which results in a strong discontinuity in the control law.

Finally, control law (21) may not be realizable due to ill conditioning. Indeed, when some components of $\mathbf{H}$ are small, the matrix $\mathbf{HJ}$ can be very ill conditioned. Its pseudo inverse $(\mathbf{HJ})^+$ is thus very large. In (18), the large coefficients of $(\mathbf{HJ})^+$ are diminished when multiplied with $\mathbf{H}$. If the approximation $\widehat{\mathbf{H}}$ is used instead, no correction is brought since the small coefficients of $\mathbf{H}$ are approximated by 1. The matrix coefficients of $(\mathbf{HJ})^+\widehat{\mathbf{H}}$ can thus be very large, and the result on the control is unpredictable.

F. Conclusion

In this section, several control laws have been proposed, based on the classical methods in the state of the art. The continuity of these control laws has been investigated when the number of visual features varies, and the general characteristics observed are summarized on Table I. None of these control laws is continuous when the task is not fully redundant, *i.e.* when at least one feature can not be expressed as a combination of the others. Indeed, when this feature is activated or inactivated, the system earns or looses a degree of freedom. In this situation, the nice properties of continuity or stability demonstrated in [25] can not be obtained directly anymore.

V. Building a new inverse operator

Since the classical control schemes based on the pseudo-inverse operator are not able to ensure the continuity, a

	CONTROL LAW	**NON REDUNDANT CASE**		**REDUNDANT CASE**	
		Non redundant	Decoupled	Redundant	Fully redundant
(17)	$\left(\widehat{\mathbf{H}}\mathbf{J}\right)^+\widehat{\mathbf{H}}$	discontinuous	discontinuous	discontinuous	discontinuous
(18)	$\left(\mathbf{H}\mathbf{J}\right)^+\mathbf{H}$	discontinuous	discontinuous	discontinuous	continuous
(19)	$\left(\mathbf{H}\mathbf{J}\right)^\dagger\mathbf{H}$	discontinuous	discontinuous	discontinuous	continuous
(20)	$\left(\widehat{\mathbf{H}}\mathbf{J}\right)^+\mathbf{H}$	discontinuous	continuous	discontinuous	discontinuous
(21)	$\left(\mathbf{H}\mathbf{J}\right)^+\widehat{\mathbf{H}}$	ill conditioned			

TABLE I
SUMMARY OF THE CONTROL LAW BEHAVIORS

control law based on a new inversion operator is proposed in this section. We first propose a formal specification of the continuity properties that this operator should fulfill to ensure the control-law continuity. We propose then an implementation that respects these specifications, and we use it to build a new control law that is proved to be continuous. We finally prove that the obtained control law has similar properties of local stability than the classical pseudo-inverse-based control laws.

A. Formal definition

In a first time, let us properly define the properties of the operator we are looking for. This operator should be equivalent to the classical pseudo-inverse operator when all the features are fully active or inactive (*i.e.* when $\forall i = 1..k, \quad h_i \in \{0, 1\}$). The operator should also maintain the continuity when h goes smoothly from 0 to 1. This is formalized through the following definition:

Definition 5.1 (Continuous inverse): Let $\mathbf{A}$ be a matrix of size $(k \times n)$ and $\mathbf{H}$ be a diagonal activation matrix of size $(k \times k)$, whose components belong to $[0, 1]$. The continuous inverse $\mathbf{A}^{\ddagger\mathbf{H}}$ of a matrix $\mathbf{A}$ subject to an activation $\mathbf{H}$ respects the two following properties:

- if $\forall i = 1..k, \quad h_i \in \{0, 1\}$, then:
$$\mathbf{A}^{\ddagger\mathbf{H}} = (\mathbf{H}\mathbf{A})^+ = (\mathbf{H}\mathbf{A})^+\mathbf{H} \tag{37}$$
- The function $(\mathbf{A}, \mathbf{H}) \rightarrow \mathbf{A}^{\ddagger\mathbf{H}}$ is continuous with respect to $\mathbf{H}$.

B. Construction of a continuous inverse

We now propose an implementation of this definition, based on the study of (20) and particularly on the discontinuity observed in (36). The goal is to build an inverse of the following form:

$$\mathbf{J}^{\oplus\mathbf{H}} = \left[h_1\mathbf{J_1}^+ \ h_2\mathbf{J_2}^+\right] + h_1h_2\mathbf{X_{12}} \tag{38}$$

The generalization of such relation requires a more general definition of the coupling matrices.

Definition 5.2 (Coupling matrices of a matrix $\mathbf{J}$): The coupling matrices of a $k \times n$-matrix $\mathbf{J}$ are indexed by the subspaces $\mathcal{P}$ of the k first integers. They are defined recursively:

$$\begin{array}{rll} & \text{if} \quad \mathcal{P} = \varnothing, & \mathbf{X}_\varnothing = \mathbf{0}_{n\times k} \\ \text{otherwise} & \forall \mathcal{P} \in \mathfrak{P}(k), & \mathbf{X}_p = \mathbf{J}_\mathcal{P}^+ - \sum_{\mathcal{Q}\subsetneq\mathcal{P}} \mathbf{X}_\mathcal{Q} \end{array} \tag{39}$$

where $\mathfrak{P}(k) = \mathfrak{P}(1..k) = \Big\{\mathcal{P}| \quad \mathcal{P} \subset 1..k\Big\}$ are all the subsets composed of the k first integers, and $\mathbf{J}_\mathcal{P} = \mathbf{H}\mathbf{J}$ where h_i is equal to 1 if $i \in \mathcal{P}$, and to 0 otherwise (*i.e.* $\mathbf{J}_\mathcal{P}$ is the Jacobian matrix $\mathbf{J}$ whose only activated lines are those of $\mathcal{P}$).

Remark 5.1: If $\mathbf{J}$ has two lines ($\mathbf{J} = \begin{bmatrix}\mathbf{J_1} \\ \mathbf{J_2}\end{bmatrix}$), then the definition of the coupling matrix $\mathbf{X_{12}}$ corresponds to the notation given in (34).

Using this definition, it is now easy to build an inverse $\mathbf{J}^{\oplus\mathbf{H}}$ that respects the specification given in Definition 5.1.

Definition 5.3 (Continuous inverse $\mathbf{J}^{\oplus\mathbf{H}}$): Let $\mathbf{J}$ be a matrix of size $(k \times n)$ and $\mathbf{H}$ the corresponding activation matrix whose components $\big(h_i\big)_{i\in 1..k}$ belong to the interval $[0, 1]$. The continuous inverse of $\mathbf{J}$ activated by $\mathbf{H}$ is defined by:

$$\mathbf{J}^{\oplus\mathbf{H}} = \sum_{\mathcal{P}\in\mathfrak{P}(k)} \Big(\prod_{i\in\mathcal{P}} h_i\Big)\, \mathbf{X}_\mathcal{P} \tag{40}$$

Remark 5.2: The continuous inverse of a double-line Jacobian $\mathbf{J} = \begin{bmatrix}\mathbf{J_1} \\ \mathbf{J_2}\end{bmatrix}$, activated by $\mathbf{H}$ is:

$$\begin{aligned} \mathbf{J}^{\oplus\mathbf{H}} &= h_1h_2\mathbf{X}_{\{\mathbf{1},\mathbf{2}\}} + h_1\mathbf{X}_{\{\mathbf{1}\}} + h_2\mathbf{X}_{\{\mathbf{2}\}} \\ &= \Big[h_1\mathbf{J_1}^+ \ h_2\mathbf{J_2}^+\Big] + h_1h_2\mathbf{X_{12}} \end{aligned} \tag{41}$$

This last equation matches exactly the preliminary goal written in (38).

Definition 5.3 proposes a new operator to inverse a matrix $\mathbf{J}$ activated by a diagonal activation matrix $\mathbf{H}$. This operator will now be proved to respect the specification given in Definition 5.1.

Theorem 5.1 (Continuity of $\mathbf{J}^{\oplus\mathbf{H}}$): The inverse $\mathbf{J}^{\oplus\mathbf{H}}$ of $\mathbf{J}$ activated by $\mathbf{H}$ fulfills the specifications of a continuous inverse given in Definition 5.1.

Proof: Two points have to be proved. First of all, we prove that $\mathbf{J}^{\oplus\mathbf{H}}$ is equal to the classical pseudo inverse $(\mathbf{H}\mathbf{J})^+\mathbf{H}$ when the components of $\mathbf{H}$ are binary (*i.e.* no feature is within the transition region). Let $\mathcal{P}$ be the set of non-zero components of $\mathbf{H}$. Using the notations of Definition 5.2, we have thus to prove that $\mathbf{J}^{\oplus\mathbf{H}} = \mathbf{J}_\mathcal{P}^+$. Using (40), it is possible to write:

$$\mathbf{J}^{\oplus\mathbf{H}} = \mathbf{X}_\mathcal{P} + \sum_{\mathcal{Q}\subsetneq\mathcal{P}} \mathbf{X}_\mathcal{Q} \tag{42}$$

From (39), it is known that $\mathbf{X}_\mathcal{P} = \mathbf{J}_\mathcal{P}^+ - \sum_{\mathcal{Q}\subsetneq\mathcal{P}} \mathbf{X}_\mathcal{Q}$. Introducing (39) in (42), it is finally possible to obtain $\mathbf{J}^{\oplus\mathbf{H}} = \mathbf{J}_\mathcal{P}^+$.

The second point to prove is the continuity of the inverse with respect to the variation of $\mathbf{H}$. All the coupling matrix $\mathbf{X}_p$

Algorithm 1 Calculate $\mathbf{J}^{\oplus \mathbf{H}}$

```
Parameters: Jacobian $\mathbf{J}$, activation $\mathbf{H}$, coupling matrices
    $\mathbf{X}_{\varnothing} \dots \mathbf{X}_{\mathbf{1..k}}$
Ensure: $\mathbf{R} = \mathbf{J}^{\oplus \mathbf{H}}$
 1: $(k \times n) \Leftarrow \dim\mathbf{J}$
 2: $\mathbf{R} \Leftarrow \mathbf{0}_{\mathbf{n}\times\mathbf{k}}$
 3: for all $\mathcal{P} \subset 1..k$ do
 4:     $Prod \Leftarrow 1$
 5:     for all $i$ in $\mathcal{P}$ do
 6:         $Prod \Leftarrow Prod \times \mathbf{H}(i,i)$
 7:     end for
 8:     $\mathbf{R} \Leftarrow \mathbf{R} + Prod \times \mathbf{X}_{\mathcal{P}}$
 9: end for
10: return $\mathbf{R}$
```

are independent to $\mathbf{H}$ (by Definition 5.2). Thus the inverse (40) is simply a polynomial form of the h_i. Since a polynomial is always continuous, the continuity of the inverse with respect to the variations of $\mathbf{H}$ is demonstrated. ■

C. Computation of the continuous inverse

As given by (42), the continuous inverse operator is based on a sum of the coupling matrices. Algorithm 1 details the computation of the continuous inverse based on the values of all the coupling matrices. The computation of the coupling matrices is then detailed in Algorithm 2.

To perform the scan of all the members of $\mathfrak{P}(k)$ (as expressed in Line 3 of the algorithm), two solutions are possible. The first one is to exhaustively describe off-line all the subparts of $1..k$, and to store the result in a list, which can then be scanned during the control. The exhaustive description is costly, but has to be done only once. The second solution is to use the binary representation of the integers as a unique representation of each subpart of $1..k$ by defining the following bijective association: $x = \sum_{i=0}^{k} b_i 2^i \rightarrow \{i, \text{ so that } b_i = 1\}$, where $(b_0 ... b_k)$ is the binary representation of x. Using this bijection, the set of all subparts can be scanned directly by covering all the integers fro 0 to 2^k. On the opposite, the scan of $\mathcal{P}$ expressed by Line 5 of the algorithm is simply realized by expressing $\mathcal{P}$ as a list.

As already said, each specific coupling matrix $\mathbf{X}_{\mathcal{P}}$ can be computed as detailed in the recursive Algorithm 2. As previously, the loop Line 7 is performed by using the equivalent integer representation of the subset. The computation of each matrix $\mathbf{X}_{\mathcal{P}}$ has to be performed only once, by storing the already-computed value in an appropriate structure.

D. Continuous control law

Based on this new inversion operator, it is possible to propose the following control law:

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}^{\oplus \mathbf{H}} \mathbf{e} \tag{43}$$

Thanks to the nice properties of the continuous inverse, the control law (43) is continuous everywhere. Moreover, when all the features are fully active or fully inactive (*i.e.* $\forall i = 1..k, h_i \in \{0, 1\}$), the control law (43) is equivalent to the classical control law (2) (*i.e.* $\dot{\mathbf{q}} = -\lambda \mathbf{J}_{\mathbf{A}}^{+} \mathbf{e}_{\mathbf{A}}$, where $\mathbf{e}_{\mathbf{A}}$ is the active part of $\mathbf{e}$). As shown in the following paragraph, (43) has thus the same stability property of the equivalent task. In particular, if all the features are active at the desired position, the control law (43) is locally asymptotically stable at task completion.

Algorithm 2 Calculate the coupling matrix $\mathbf{X}_{\mathcal{P}}$

```
Parameters: Jacobian $\mathbf{J}$, subset $\mathcal{P}$
Ensure: $\mathbf{X}_{\mathcal{P}}$ is computed
 1: $(k \times n) \Leftarrow \dim\mathbf{J}$
 2: $\mathbf{H} \Leftarrow \mathbf{0}_{\mathbf{k}\times\mathbf{k}}$
 3: for all $i$ in $\mathcal{P}$ do
 4:     $\mathbf{H}(i,i) \Leftarrow 1$
 5: end for
 6: $\mathbf{X}_{\mathcal{P}} \Leftarrow \big(\mathbf{H}\mathbf{J}\big)^{+}$
 7: for all $\mathcal{Q} \subsetneq \mathcal{P}$ do
 8:     $\mathbf{X}_{\mathcal{P}} \Leftarrow \mathbf{X}_{\mathcal{P}} - \mathbf{X}_{\mathcal{Q}}$
 9: end for
```

E. Stability of the control law

Theorem 5.2 (Asymptotical statibility at binary activation): Let $\mathbf{e}$ be a task whose activation is denoted by the diagonal matrix $\mathbf{H}$ and whose Jacobian $\mathbf{J} = \partial\mathbf{e}/\partial\mathbf{q}$ is full rank. Let $\mathbf{e}_{\mathbf{q}}$ denote the equivalent active task ($\mathbf{e}_{\mathbf{q}} = \mathbf{H}\mathbf{e}$) whose Jacobian is denoted $\mathbf{J}_{\mathbf{q}}$. Then, when all the components of $\mathbf{H}$ are binary (*i.e.* $\forall i = 1..k,\ h_i = 0$ or $h_i = 1$), Control Law (43) has the same local properties of stability than the equivalent active task function $\mathbf{e}_{\mathbf{q}}$:

- it is stable in the sense of Lyapunov.
- it is asymptotically stable in the sense of Lyapunov if $\mathbf{J}_{\mathbf{q}}$ is full rank.

Proof: The control law associated to the task $\mathbf{e}_{\mathbf{q}}$ is

$$\dot{\mathbf{q}} = -\lambda \big(\mathbf{H}\mathbf{J}\big)^{+} \mathbf{H}\mathbf{e} \tag{44}$$

From the first point of Definition 5.1, control law (43) is equal to (44) when the components of $\mathbf{H}$ are 0 or 1. Then locally it has the same properties of stability as given by [25]. ■

This last result is particularly interesting when the activation is binary around the desired position, as formalized by the following corollary.

Corollary 5.1: (Local asymptotical stability around the desired position): Consider the task $\big(\mathbf{e}, \mathbf{H}, \mathbf{J}\big)$, whose desired position is a submanifold of the activation area, *i.e.* for all configurations $\mathbf{q}$, if $\mathbf{e}(\mathbf{q}) = 0$, then $\mathbf{H}(\mathbf{q}) = \mathbf{I}$. Then the control law (43) is asymptotically stable in a neighborhood around the desired position if the matrix $\mathbf{H}\mathbf{J}$ is full rank at the desired position.

Proof: Let $\mathcal{D}^* = \big\{\mathbf{q} :\ \mathbf{e}(\mathbf{q}) = \mathbf{0}\big\}$ denote the goal position, and $V = \frac{1}{2}\mathbf{e}^{\top}\mathbf{H}\mathbf{e}$. V is a positive continuous function of the configuration $\mathbf{q}$. In the manifold $\mathcal{D}^*$, we have $\mathbf{H} = \mathbf{H}^2$ and $\dot{\mathbf{H}} = 0$: thus the derivative $\dot{V}$ is equal to $\dot{V} = -\lambda\mathbf{e}^{\top}\mathbf{H}\mathbf{J}\big(\mathbf{H}\mathbf{J}\big)^{+}\mathbf{H}\mathbf{e} = -{\mathbf{e}_{\mathbf{q}}}^{\top}\mathbf{J}_{\mathbf{q}}{\mathbf{J}_{\mathbf{q}}}^{+}\mathbf{e}_{\mathbf{q}} < 0$ (with $\mathbf{e}_{\mathbf{q}}$ the equivalent active task function whose Jacobian is $\mathbf{J}_{\mathbf{q}}$). Since V is continuous with respect to the configuration in the closed set $\mathcal{D}^*$, there exists an open neighborhood $\mathcal{N}$ of $\mathcal{D}^*$ such that $\dot{V} < 0$ inside $\mathcal{N}$. In $\mathcal{N}$, the control law (43) is asymptotically stable in the sense of Lyapunov. ■

Remark 5.3: The case of Corollary 5.1 is the one considered in the experiments presented in Section VI: as explained in the following, at the desired position, all the visual features are in the fov of the camera and are thus active in the control law. On the opposite, it has not been possible yet to demonstrate the stability of the system when $0 < h < 1$. In particular, when the desired position is inside the activation border, we have only be able to verify experimentally that the behavior of the system seems appropriate.

Remark 5.4: *(Bounded-input bounded-output Stability)* If the Jacobian $\mathbf{J}$ is considered constant, the continuous inverse $\mathbf{J}^{\oplus\mathbf{H}}$ is simply a polynomial form:

$$\mathbf{J}^{\oplus\mathbf{H}} = Polyn(h_1, \dots, h_n) = \sum_{\mathcal{P}\in\mathfrak{P}(k)} monom_{\mathcal{P}}[h_i]\ \mathbf{X}_{\mathcal{P}} \tag{45}$$

where $monom_{\mathcal{P}}[h_i]$ is a product of the subpart of the h_i. Since the h_i are bounded by $[0, 1]$, the polynomial form is k-Lipschitz continuous, with a constant k that depends on the coupling matrices $\mathbf{X}_{\mathcal{P}}$. It is therefore possible to prove the bounded-input bounded-output (BIBO) stability. However, the better bound that we have found for the constant k is 2^k, where k is the number of features, which is not relevant in practice.

VI. Experimental results

We present in this section several experiments that study the behavior of the system running the new control law presented in the previous sections. The experiments have been realized in simulation, using the classical visual servoing scheme based on feature points [13], which is first recalled. Three typical experiments are then presented in details. During the execution, some points may leave the camera fov. They are then removed from the feature set. As done in [12], an activation buffer is defined at the image border to smoothly inactivate the feature that are leaving the fov.

A. Visual servoing implementation

The work presented above is general and could be applied to any robotic task defined by a derivable error $\mathbf{e}$. In the following, the error function is computed from visual features [13]:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{46}$$

where $\mathbf{s}$ is the current value of the visual features for task $\mathbf{e}$ and $\mathbf{s}^*$ their desired value. In the experiments, the visual features are the 2D positions $\mathbf{p_i} = (x_i, y_i)$ of a set of points, whose 3D positions within the camera frame are denoted $\mathbf{P_i} = (X_i, Y_i, Z_i)$. $\dot{\mathbf{s}}_\mathbf{i} = \mathbf{L}_{\mathbf{s_i}}\mathbf{v}$, where $\mathbf{v}$ is the instantaneous camera velocity. For one point $\mathbf{p_i}$, $\mathbf{L}_{\mathbf{p_i}}$ is the well-known matrix given in [11]. The interaction matrix of the task $\mathbf{e}$ is finally $\mathbf{L} = (\mathbf{L}_{\mathbf{p_1}}, \dots, \mathbf{L}_{\mathbf{p_{k/2}}})$. From (46), it is clear that the interaction matrix $\mathbf{L}$ and the task Jacobian $\mathbf{J}$ are linked by the relation:

$$\mathbf{J} = \mathbf{L}\mathbf{M}\mathbf{J}_\mathbf{q} \tag{47}$$

where $\mathbf{J}_\mathbf{q}$ is the robot Jacobian ($\dot{\mathbf{r}} = \mathbf{J}_\mathbf{q}\dot{\mathbf{q}}$) and $\mathbf{M}$ is the matrix that relates the variation of the camera velocity $\mathbf{v}$ to the variation of the camera pose parametrization ($\mathbf{v} = \mathbf{M}\dot{\mathbf{r}}$).

As already said in II-A, an approximation $\widehat{\mathbf{J}}$ has to be used in practice. In particular, different choices are possible for $\widehat{\mathbf{L}}$ [15]. We choose $\widehat{\mathbf{J}} = \mathbf{L}^*\mathbf{M}\mathbf{J}_\mathbf{q}$, where $\mathbf{L}^*$ is the interaction matrix computed at the desired position. This choice is frequently done since it reduces the risk of falling in a local minimum. However, due to this approximation, some points may leave the camera fov during the servo [5]. These points are then removed from the feature set as proposed in [12]. The error task is a varying-feature-set task based on (46), defined by $\mathbf{e_q} = \mathbf{He}$. $\mathbf{H}$ is the activation matrix whose coefficients h_i are defined by $h_{2i} = h_{2i+1} = \min(h_x(x_i), h_y(y_i))$ The horizontal activation function h_x is defined by:

$$h_x(x) = \begin{cases} 1 & \text{if } \bar{x}^- + \epsilon_x \leq x \leq \bar{x}^+ - \epsilon_x \\ 0 & \text{if } x \geq \bar{x}^+ \text{ or } x \leq \bar{x}^- \\ f_{\epsilon_x}\Big(x - (\bar{x}^+ - \epsilon_x)\Big) & \text{if } \bar{x}^+ - \epsilon_x \leq x \leq \bar{x}^+ \\ f_{\epsilon_x}\Big((\bar{x}^- + \epsilon_x) - x\Big) & \text{if } \bar{x}^- \leq x \leq \bar{x}^- + \epsilon_x \end{cases} \tag{48}$$

where $[\bar{x}^-,\ \bar{x}^+]$ is the horizontal range of the image, ϵ_x tunes the length of the transient interval and the transient function f_ϵ is defined by $f_\epsilon(x) = \frac{1}{2}\Big(1 + tanh(\frac{\epsilon}{x} - \frac{1}{1-x/\epsilon})\Big)$. The vertical activation function h_y is defined similarly. $\mathbf{H}$ defines around the image an *activation buffer* where the feature that is leaving the fov is progressively and smoothly inactivated.

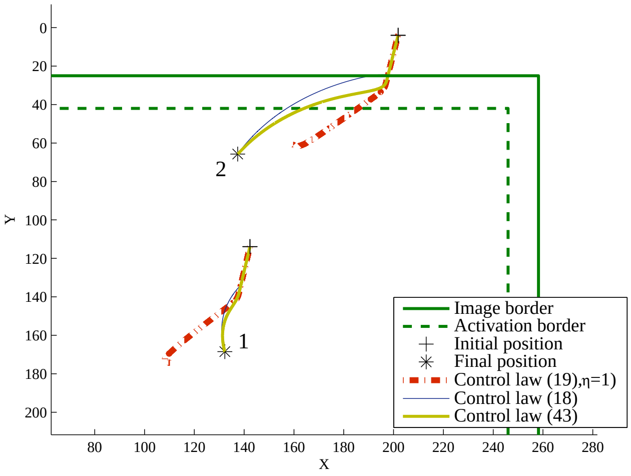


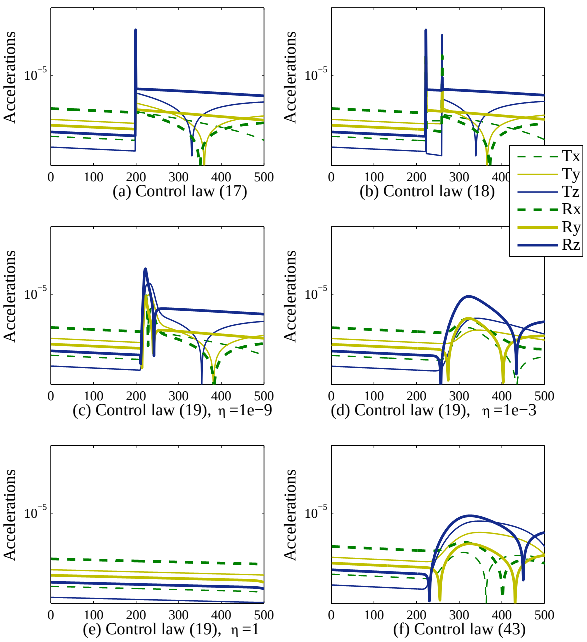
Fig. 2. Experiment 1: points trajectories in the image. At the beginning of the servo, Point 2 is out of the image. The two desired positions are within the full-activation area. Control law (19) using a too large value of η, even if the velocity is continuous, is unable to converge and thus to achieve the task. On the opposite, control laws (18) and (43) manage to reach the desired position. Control law (18) provides an image trajectory which is quite abrupt when the points enter the activation area. On the opposite, the trajectories using control law (43) are smooth.

B. First experiment: non-redundant task

The first experiment has been realized using a two-point target. The dimension and the rank of the task are thus both equal to 4 at full activation: the task is always non redundant. At the initial position, one point is out of the fov. We mainly consider the continuity of the control law when the point enters the fov. The experiment is summed up in Figures 2 and 3.

The continuity of control laws (17), (18), (19) and (43) are compared in Fig. 3. As shown in Section IV-C, (17) and (18) are equivalent in this case. An acceleration peak appears at rank change (see Fig 3-(a) and (b)). An inflection point (corresponding to a discontinuity of the derivative) also appears in the point trajectories (see Fig. 2).

Control law (19) is able to ensure the continuity if η is properly set. However, Fig 3-(c) and (e) point out the



(a) Control law (17)

(b) Control law (18)

(c) Control law (19), η =1e−9

(d) Control law (19), η =1e−3

(e) Control law (19), η =1

(f) Control law (43)

Fig. 3. Experiment 1: Comparison of the acceleration peaks when using control laws (17) on (a), (18) on (b), (19) with several tuning on graphs (c) to (e) ($\eta = 1e^{-9}$, $1e^{-3}$ and 1) and (43) on (f). The acceleration is measured at the robot end effector, *i.e.* at the camera focal point. Since the main relevant situation occurs when the second point enters the camera fov at iteration 200, the plots have been limited to the interval $[0,\ 500]$. Control laws (17) and (18) are unable to ensure the continuity. Concerning (19), the greater the tuning parameter η, the smaller the acceleration peaks are. A good compromise is obtained for this experiment with $\eta = 1e^{-3}$ (with $\eta = 1$, the behavior seems to be correct while looking the continuity, but Fig. 2 shows that it is not when considering the overall task). Control law (43) also provides a smooth control, with no peak of acceleration.

importance of the threshold η of the damped-least-square operator. When η is too small (Fig. 3-(c)), the same peak of acceleration as with (18) is obtained. The more η increases, the more the acceleration peak decreases (until disappearing, see Fig. 3-(d)). However, if η is too large, the convergence becomes very slow (like if using the transpose operator). In some particular cases, the robot is unable to converge (see Fig. 2). A good trade-off has to be carefully selected (it is obtained here with $\eta = 1e^{-3}$). This experiment emphasizes that it is very difficult to find a correct value for any conditions.

Finally, the original control law (43) provides a good behavior. The control law is continuous (see Fig. 3-(f)), and the acceleration is similar to the one obtained with (19) for the best tuning of η. The good properties of (43) are also visible on Fig. 2: the point trajectories are smooth and continuous.

C. Second experiment: oscillations at rank change

This experiment illustrates the problems that may occur if the control law is discontinuous. In the case presented below, the classical control laws oscillate when a point leaves the camera fov, which could even prevent the convergence. This problem is solved when using a continuous control law. The experiment is summed up in Figures 4, 5, and 6.

The target used in this experiment is the same as for Experiment 1. Two points are considered, which gives a task of dimension and rank 4. The required displacement is mainly a large rotation around the optical axis. During the rotation, $\mathbf{P}_2$ gets close to the activation border (see Fig. 4). At this point, if the two points are considered in the feature set, the computed control makes $\mathbf{P}_2$ leave the camera fov. On the opposite, if considering only the point $\mathbf{P}_1$ remaining in the fov, the control becomes mainly a pan-tilt motion that tends $\mathbf{P}_2$ to enter again the activation zone. The oscillation observed is thus due to this dilemma: if $\mathbf{P}_2$ is inactivated, it enters the image which activates it, which makes it leaves the image, etc.

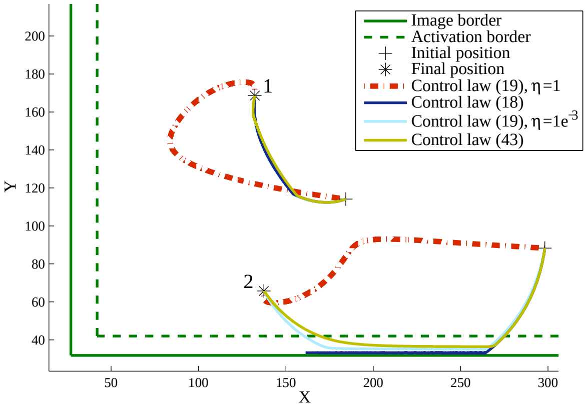


Fig. 4. Experiment 2: image point trajectories. The main required motion is a rotation around the optical axis. With the classical control law (18), Point $\mathbf{P}_2$ leaves the image. Due to discontinuities in the control law, oscillations appear then in the control, and $\mathbf{P}_2$ oscillates around the image border. The point is stuck up at the border and the servo is even unable to converge. With an appropriate tuning of control law (19) or with the continuous control law (43), the servo converges.

The oscillation occurs due to the control law discontinuity. Control laws (17), (18) and (19) with η too small oscillate, (see Fig. 5). Moreover, (18) is unable to leave the oscillation area: the servo does not converge. On the opposite, using (43), the control is smooth, and no oscillation appears (see Fig. 6).

D. Third experiment: redundant task

This last experiment points out the discontinuities that can occur with a redundant task when a non-redundant feature is inactivated. The target is composed of eight points. The global task is thus fully redundant. The desired motion is mainly a rotation around the optical axis. While performing this motion, a large amount of points initially close to the image borders leaves the camera fov. Due to the point inactivations, the task becomes redundant, then not redundant (between iterations 150 and 500, only two points remain within the camera fov). Finally, when the robot achieves the required motion at the end of the execution, all the points enter the image frame again and the global task converges. The experiment is summed up on Figures 7, 8 and 9. The behavior is equivalent when using (17), (18), (20) or (19) with a low gain η. Therefore, we only give the graphics for (19) with different values for η.

The point trajectories obtained with control law (19) are given in Fig. 7-(a). They are very abrupt, showing large inflection points when a point enters the image. When $\mathbf{P}_3$ enters back in the active area, it is temporarily stuck on the

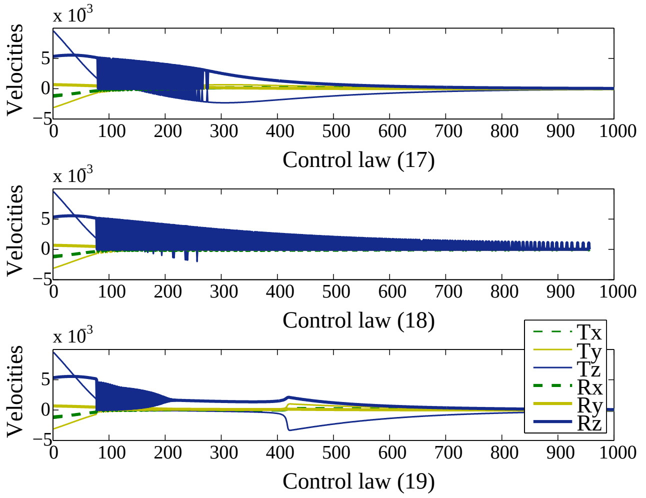


Fig. 5. Experiment 2: The six components of the camera velocities using control laws (17) without approximation, (18) with a partial approximation and (19) with the damped least square inverse operator. Oscillations appear when $\mathbf{P}_2$ reaches the image border (at Iteration 100).

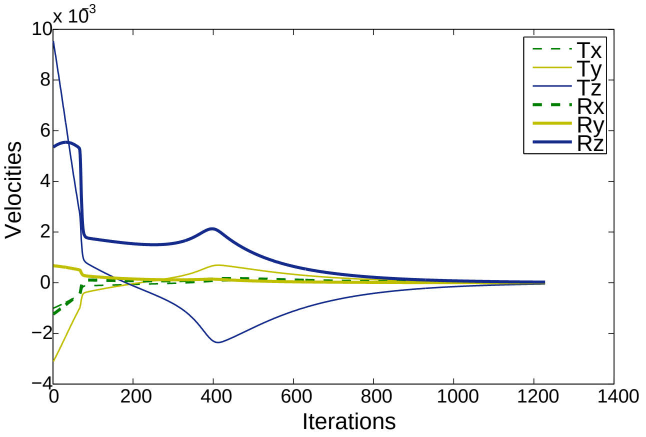
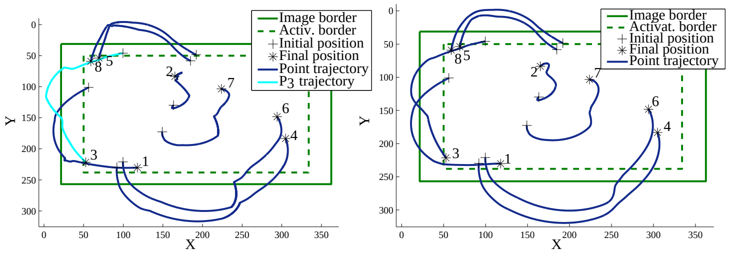


Fig. 6. Experiment 2: velocities of the camera using control law (43). The control law is continuous.

activation border, causing oscillation in the control law. This is due to a similar phenomena as in the previous experiment. The point is finally detached from the border when $\mathbf{P_1}$ enters the activation area (the task becomes then fully redundant). On the opposite, the trajectories using (43) are smooth, without any inflection point nor oscillation.

The required accelerations are shown on Fig. 8. When η is small, (19) produces strong peaks of acceleration (see Fig. 8-(a)). We can notice the peak at iteration 250. At this time, three points are active. The task is fully redundant. However, it is still ill conditioned [5], which finally also produces a discontinuity. While increasing the damping parameter η, the accelerations peaks are reduced (Fig. 8-(a) and (b)). For this experiment, the *good* value of η is $1e^{-2}$. This emphasizes the difficult choice of η, that has to be tuned differently for each experiment. On the opposite, Fig. 8-(d) shows that the control law (43) is continuous. No acceleration peak appears, whatever the rank of the conditioning of the task is.

Finally, the computation cost of the control law (43) is given in Fig. 9. The computations have been realized without special optimization on a classical desktop computer (Pentium 2.3GHz monocore). Moreover, all the computations are done online: no special pre-process is perform beforehand. The cost increases with the number of features inside the activation buffer. The total cost is far below the milisecond, which makes it sufficient for application on a real robot.



(a) Control law (18)

(b) Control law (43)

Fig. 7. Experiment 3: point trajectories (a) using (19) (b) using control law (43). Using (18), $\mathbf{P}_3$ then oscillates around the activation border. Using (43), the trajectories are smooth. The general form of the trajectories is similar to (a) but smoother, and $\mathbf{P}_3$ does not oscillate when it is activated again.

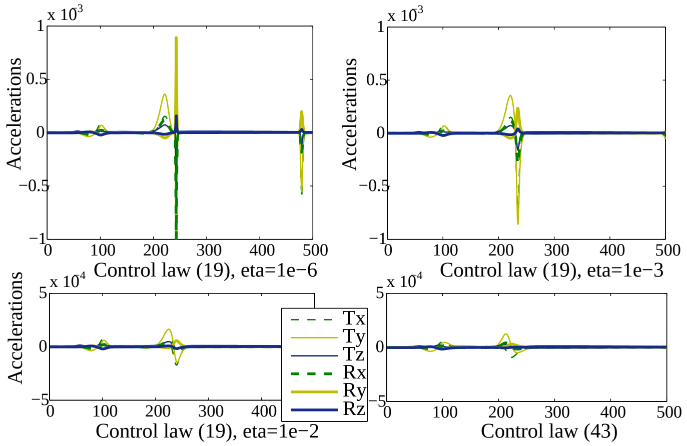


Fig. 8. Experiment 3: comparison of the accelerations observed when using control law (19) with different values of η and using control law (43).

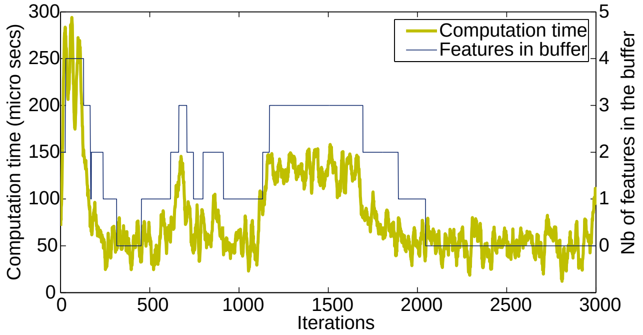


Fig. 9. Experiment 3: Cost of the control law computation. The cost directly depends on the number of features inside the activation buffer (*i.e.* whose activation parameter h is such that $0 < h < 1$.

VII. Conclusion

This paper has considered tasks defined by a set of features whose dimension is varying along the time. Such tasks have been addressed in the literature for several applications. Classically, the variation of the size of the feature set is performed by using an activation matrix whose value varies between 0 and 1. Within this context, a natural interrogation concerns the behavior of the resulting control law at the critical point of activation or inactivation of a feature. We have studied this situation, through several combinations and approximations of the Jacobian associated to such varying feature set.

The main point of this theoretical study is that unsing an activation matrix with a classical pseudo inverse is inefficient to ensure the continuity when the input signal is not fully redundant. It has been proved that the pseudo inverse is always discontinuous at the activation of a non redundant feature except in the particular case of a perfect decoupling.

To deal with this problem, an original inverse operator has been introduced. It insures the continuity even when the rank of the Jacobian changes. This new inversion operator has then been used to define a new control law, continuous everywhere. The good properties of this control law have finally been verified through several visual-servoing experiments.

References

[1] P. Baerlocher and R. Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 6(20):402–417, Aug. 2004.

[2] A. Ben-Israel and T. Greville. *Generalized inverses: theory and applications*. Wiley-Intersciencex, New York, 1974.

[3] R. Boulic and D. Thalmann. Combined direct and inverse kinematic control for articulated figure motion editing. *Computer Graphics Forum*, 11(4):189–202, Aug. 1992.

[4] T. Chang and R. Dubey. A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE T. Robot Autom*, 11(2):286–292, April 1995.

[5] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer, 1998.

[6] C. Cheah, D. Wang, and Y. Sun. Region-reaching control of robots. *IEEE Trans. Robot.*, 23(6):1260–1264, Dec. 2007.

[7] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE T. Robot Autom*, 13(3):398–410, June 1997.

[8] A. Comport, E. Marchand, and F. Chaumette. Statistically robust 2d visual servoing. *IEEE Trans. on Robotics*, 22(2):415–420, Apr. 2006.

[9] A. Deo and I. Walker. Robot subtask performance with singularity robustness using optimal damped least squares. In *IEEE Int. Conf. Robot. Autom. (ICRA'92)*, pages 434–441, Nice, France, May 1992.

[10] K. Doty, C. Melchiorri, and C. Bonivento. A theory of generalized inverses applied to robotics. *IJ. Robotics Research*, 12:1–19, Dec. 1993.

[11] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE T. Robot Autom*, 8(3):313–326, June 1992.

[12] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal. Continuous visual servoing despite the changes of visibility in image features. *IEEE Trans. Robot.*, 21(6):415–421, Apr. 2005.

[13] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE T. Robot Autom*, 12(5):651–670, Oct. 1996.

[14] L. Kelmar and P. Khosla. Automatic generation of forward and inverse kinematics of a reconfigurable modular manipulator system. *Journal of Robotic Systems*, 7(4):599–620, Aug. 1990.

[15] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. Robot. Autom. (ICRA'04)*, pages 1843–1848, New Orleans, USA, Apr. 2004.

[16] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 D visual servoing. *IEEE T. Robot Autom*, 15(2):238–250, Apr. 1999.

[17] N. Mansard and F. Chaumette. Task sequencing for sensor-based control. *IEEE Trans. on Robotics*, 23(1):60–72, Feb. 2007.

[18] N. Mansard, A. Remazeilles, and F. Chaumette. Continuity of varying-feature-set control laws. Technical Report 1864, conditionnaly accepted in IEEE TAC, IRISA, Rennes, France, Sep. 2007, available at ftp://ftp.irisa.fr/techreports/2007/PI-1864.pdf.

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

[20] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. *Trans. ASME Dynamic Sys. Measures and Control*, 108:163–171, Sep. 1986.

[21] B. Nelson and P. 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.

[22] M. Peinado, D. Meziat, D. Maupu, D. Raunhardt, D. Thalmann, and R. Boulic. Accurate on-line avatar control with collision anticipation. In *ACM symposium on Virtual reality software and technology*, pages 89 – 97, Newport Beach, USA, Nov. 2007.

[23] D. Raunhardt and R. Boulic. Progressive clamping. In *IEEE Int. Conf. Robot. Autom. (ICRA'07)*, pages 4414–4419, Roma, Italy, Apr. 2007.

[24] A. Remazeilles, N. Mansard, and F. Chaumette. Qualitative visual servoing: application to the visibility constraint. In *IEEE/RSJ Int. Conf. Int. Rob. Sys. (IROS'06)*, pages 4297–4303, Beijing, China, Oct. 2006.

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

[26] B. Siciliano and J-J. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE Int. Conf. on Advanced Robotics (ICAR'91)*, pages 1211–1216, Pisa, Italy, June 1991.

[27] D. Whitney. The mathematics of coordinated control of prosthetic arms and manipulators. *Trans. ASME Journal of Dynamic System, Measures and Control*, 94:303–309, 1972.

Nicolas Mansard graduated from École Nationale Supérieure d'Informatique et de Mathématiques Appliquées, Grenoble, France, in 2003, and received the M.S. (DEA) the same year in Robotics and Image Processing from the University Joseph Fourier, Grenoble. After three years in the Lagadic research group, IRISA, INRIA - Bretagne, he received the Ph.D. degrees in computer science from the University of Rennes, Rennes, France, in 2006. He then spent one year at Stanford University, with O. Khatib and one year in JRL-Japan, AIST, Tsukuba, Japan, with A. Kheddar and K. Yokoi. He is currently with LAAS/CNRS, Toulouse, France, in the GEPETTO group. His research interests are concerned with sensor-based robot animation, and more specifically the integration of reactive control schemes into real robot and humanoid applications. Dr. Mansard's thesis has been awarded in 2006 by the French research group GdR-MACS, and in 2007 by the French ASTI (Society for Telecomunication and Computer Science) and by the price of the best thesis of the Région Bretagne.

Anthony Remazeilles graduated in 2001 with an engineering degree (Computer Science) from INSA, Rennes, France and with a M.S. of artificial intelligence and computer vision from University of Rennes I. From 2001 to 2006, he was with the INRIA Rennes, under supervision of Francois Chaumette (Lagadic group), and Patrick Gros (TeXMeX group). He received the PhD. degree in Decembre 2004, and was then teaching assistant at the INSA Rennes (Computer Science Department). In 2006 and 2007, he works as a post-doc at LIST/CEA, Fontenay aux Roses, France, about the robotic assitance for injured people. Since 2008, he is with Fatronics, Spain.

François Chaumette (M98) received the Graduate degree from École Nationale Supérieure de Mécanique, Nantes, France, in 1987, and the Ph.D. degree in computer science from the University of Rennes, Rennes, France, in 1990. Since 1990, he has been with IRISA/INRIA Rennes Bretagne Atlantique, where he is "Directeur de recherche" and Head of the Lagadic Group. His research interests include robotics and computer vision, especially visual servoing and active perception. He is the coauthor of more than 150 papers published in international journals on the topics of robotics and computer vision. Dr. Chaumette has served over the last 5 years on the program committees for the main conferences related to robotics and computer vision. He has been the Associate Editor of the IEEE Trans. on Robotics from 2001 to 2005 and is now in the Editorial Board of the Int. Journ. of Robotics Research. He was the recipient of several awards including the AFCET/CNRS Prize for the Best French Thesis on Automatic Control in 1991, King-Sun Fu Memorial Best IEEE Trans. on Robotics and Automation Paper Award with Ezio Malis in 2002.