Continuous Control Law from Unilateral Constraints

Application to Reactive Obstacle Avoidance in Operational Space

Nicolas Mansard, Oussama Khatib

Abstract—The control approaches based on tasks, and particularly based on a hierarchy of tasks, enable to build complex behaviors with some nice properties of robustness and portability. However it is difficult to consider directly unilateral constraints in such a framework. Unilateral constraints present some strong irregularities (in particular at the level of their derivative) that prevents the insertion of unilateral-based tasks at the high-priority level of a hierarchy. In this paper, we present an original method to generalize the hierarchy-based control schemes to take unilateral constraint into account at the top-priority level. We develop our method first at the kinematic level then directly at the dynamic level using the operational space. The method is then validated on a various set of robots by realizing a visual servoing under the constraint of joint limits.

I. INTRODUCTION

Traditionally, the classical approach to control a robot has been to define the objective directly in the joint space. The control approaches based on the definition of a task objective, such as defined in the task function approach [20] or in the operational space formulation [11], have been introduced to simplify the control problem. Working directly in a properly-chosen task space produces a more intuitive manner to define the robot objective. It also enables to work directly in the sensor space, which closes more tightly the control loop and improves the control law robustness and accuracy [6]. Finally, since a same task space is valid for a large set of robot, a control scheme based on a task is very portable and could be easily adapted from one robot structure to another.

Moreover, these methods produce directly the kinematic or dynamic model to decouple the motions due to the task to the free remaining motions (i.e. the motions that let the task invariant) [12], [8]. A secondary task can then be applied in the space of free motions, and, recursively, a hierarchic set of tasks can be considered [16], [23]. Hierarchy of tasks are more and more popular to build complex behavior on very redundant robot such as humanoid robots [1], [17], [22], [15], [3].

A task is generally defined by an equality of reference such as \( e = 0 \) where \( e = s - s^*(t) \) is an error to be regulated to 0. The task thus represent a bilateral constraint. On the opposite, unilateral constraint are typically represented by an inequality \( e_i \leq 0 \) that has to be always respected during the robot motion. Unilateral constraints present a strong irregularity at the activation point \( e_i = 0 \). Due to this irregularity, it is impossible to consider an unilateral constraint as a classical (bilateral) task. Unilateral constraints such as avoidance constraints have of course been considered in a large number of works in the past, in particular via the gradient projection method [12]: the unilateral constraint is embedded in a cost function [10] whose gradient is projected in the space of free motion as a lowest-priority task. This method has the big advantage to smooth the irregularity of the unilateral function. However at the derivative level, the irregularity still exist and prevents the insertion of the unilateral constraint at any level of the task hierarchy but the lowest-priority one. Indeed, the unilateral constraints have always been considered as a secondary objective to be optimized only when enough degree of freedom (DOF) are available.

We propose here to study a solution to introduce the unilateral constraints at the top-priority level of the hierarchy. Some work have already been proposed to go in this direction. This generally requires the use of an external control that modified the hierarchy when the unilateral constraint is not respected. An equivalent bilateral constraint is then introduced as a top-priority task to force temporarily the execution of the unilateral constraint [22], [13]. On the opposite, we propose in this paper to use a specific inverse operator introduced in [14] to smooth the irregularity of the unilateral constraint while computing the control law. Using such an operator, we can generalize the notion of hierarchy of task for both bilateral and unilateral constraints at the kinematic and at the dynamic levels.

In the following section, we remind the basic scheme to compute the control law from a hierarchic set of tasks, and we explicitly point out the discontinuities coming from the unilateral constraints. We then extend this control law to take the unilateral constrains into account, first at the kinematic level in Section III, then at the torque-control level in the operational space in Section IV. The tasks used in the experiments are then given in Section V. A set of experiments that validate our approach is finally presented in Section VI.

II. INVERSE KINEMATIC CONTROL

We first remind the classical task-based control formalism. This formulation is only valid for tasks corresponding to bilateral constraints. We then show the evidence of discontinuities that arise when considering an unilateral constraint in this formulation. This reminder will be then used to build a continuous control law in Section III.

A. Considering only one task

Let \( q \) be the joint position of the robot. The main task function is \( e \), generally computed from the sensor output.
Let us consider that the robot is controlled using the joint velocities $\dot{q}$ (to improve the readability, we will generalize to torque control only in a second time, in Section IV). The Jacobian of the task $e$ is defined by:
\[
\dot{e} = \frac{\partial e}{\partial q} \dot{q} = J \dot{q}
\]
Let $n$ be the number of DOF of the robot ($n = \dim q$) and $m$ be the size of the main task ($m = \dim e$).

The controller has to regulate $e$ to 0 according to a reference decreasing behavior $\dot{e}^*$ chosen when designing the control law. By inverting (1), the joint motion $\dot{q}$ that realizes the reference behavior $\dot{e}^*$ is given by the least-square inverse:
\[
\dot{q} = J^+ \dot{e}^*
\]
where the notation $A^+$ refers to the Moore-Penrose inverse of the matrix $A$ [2]. By (2), we consider that the Jacobian matrix is perfectly known. If it is not the case, due to inaccuracy in the calibration process or uncertainties in the robot-scene model, an approximation $\tilde{J}$ has to be used instead of $J^+$. It is possible to prove that (2) is stable if $JJ^+ \geq 0$ and asymptotically stable iff $J \tilde{J}^+ > 0$ [20]. In the following, we make the assumption that $\tilde{J}$ is perfectly known, from which we deduce $JJ^+ = I_m$ if $m \leq n$, $JJ^+ = I_{m,n}$ otherwise: the control law is thus always stable, and asymptotically stable if $m \leq n$.

In the state of the art, an implicit condition of such control schemes is always the constant rankness of the Jacobian matrix $J$. Indeed, the pseudo-inverse $J^+$ is continuous with respect to $J$ only when the rank is constant. When the rank increases or decreases, the continuity is not ensured, which can result in an awkward or even dangerous behavior. This is typically what happens when the robot reaches a singularity.

### B. Evidence of discontinuity

In the previous paragraph, we have considered a task representing a bilateral constraint: typically, $e=0$. On the opposite, an unilateral constraint can be written $e < 0$. In that case, the reference behavior $\dot{e}^*$ is typically set to:
\[
\dot{e}^* = \begin{cases} -\lambda e & \text{if } e > 0 \\ 0 & \text{otherwise} \end{cases}
\]
Some control laws based on (2) have been proposed [7], [4], [5]. They can be put under the common following form [14]:
\[
\dot{q} = (HJ)^+ H \dot{e}^*
\]
where $H = \text{diag}(h_1, \ldots, h_m)$, and $h_i = \begin{cases} 1 & \text{if } e_i > 0 \\ 0 & \text{otherwise} \end{cases}$ [14]. In the papers cited above, the continuity of the control law is obtained by ensuring that the number of activated feature (i.e. the number of non-zero $h_i$) is large enough to ensure that $HJ$ has a constant rank. However, such a hypothesis is not admissible in the general case. We can easily check [14] that a large discontinuity of the control law arises each time the rank of $HJ$ changes (corresponding typically to what happens when the robot comes to a singularity - see Fig. 3 in the experiments).

In this article, we will focus on a general solution to solve this discontinuity and ensure a proper behavior of the robot whatever the variation of the matrix $H$.

### C. Extension to $k$ tasks

The solution (2) computed above is only one particular solution of (1): it is the solution of least norm that realizes the reference behavior $\dot{e}^*$. If the rank of $J$ is smaller than $n$, a second criterion can be taken into account using the redundancy formalism [20]. The robot motion is given by:
\[
\dot{q} = J^+ \dot{e}^* + P \dot{q}_2
\]
where $P$ is the projection operator onto the null space of the matrix $J$ (i.e. $P = I_n - J^+ J$), and $\dot{q}_2$ is an arbitrary vector, used to apply a secondary control law. Thanks to $P$, this secondary motion $\dot{q}_2$ is performed without disturbing the main task $e$ having priority.

Let us now consider two tasks $e_1$ and $e_2$. The control law performing exactly $\dot{e}_1^*$ and if possible $\dot{e}_2^*$ is [23]:
\[
\dot{q} = J_1 \dot{e}_1^* + (J_2 P_1^+) (e_2^* - J_2 J_1 \dot{e}_1^*)
\]
This equation can be generalized to perform $k$ tasks, while ensuring a proper hierarchy between them [23]:
\[
\begin{cases}
\dot{q}_0 = 0 \\
\dot{q}_i = \dot{q}_{i-1} + (J_i P_{i-1}^+) (\dot{e}_i^* - J_i \dot{q}_{i-1}) , \quad i = 1..k
\end{cases}
\]
where $P_i^k$ is the projector onto the null-space of the augmented Jacobian $J_i^k = (J_1, \ldots, J_i)$. The robot joint velocity realizing all the tasks is $\dot{q} = \dot{q}_k$. A similar recursion can be established to compute $P_i^k$ at low cost [1].

Once more, some strong discontinuities can occur if the rank of a jacobian matrix $J_i$ is not constant. In particular, while considering an unilateral-constraint task (the inverse $(J_i P_{i-1}^+)^+$ is then replaced by $(HJ_i P_{i-1}^+)^+ H$), a change of rank results in a discontinuity in $\dot{q}_i$ but also in a change of rank of the associate projection operator $P_i^k$. This discontinuity is then also reported to the next level $i+1$, propagating the discontinuity at all the following levels and amplifying the discontinuity of the final control law $\dot{q}_k$.

In the following section, we will consider an unilateral-constraint task $e_i$, and compute a control law that ensure the continuity both at level $i$ and at all the upper levels $i+1 \ldots n$.

### III. KEEPING THE CONTINUITY AT THE KINEMATIC LEVEL

In this section, we will build a new control law whose form is similar to (7) and ensures the continuity even when considering unilateral constraints. In a first time (Section III-A), we remind the control law proposed in [14] to take one task into account. We then generalized this control law for a hierarchy of $k$ tasks.

#### A. Considering only one task

Let us consider a task defined by its task function $e$ (vector dimension $m$), its Jacobian $J$ (matrix, dimension $n \times m$ and constant rank $r$) and its activation matrix $H$ (diagonal matrix of dimension $m \times n$ whose diagonal components...
\[(h_i)_{i=1...m} \text{ are in the interval } [0, 1]\). The continuous inverse of \( J \) activated by \( H \) is defined by \([14]\):

\[ J^{H\oplus} = \sum_{P \in \mathcal{P}(m)} \left( \prod_{i \in P} h_i \right) X_P \quad (8) \]

where \( \mathcal{P}(m) = \mathcal{P}([1..m]) = \{ P \mid P \subseteq [1..m] \} \) are all the subsets composed of the \( m \) first integers, and the \( X_P \) are the coupling matrices of \( J \) defined by:

\[
\begin{align*}
\text{if } P &= \emptyset, \quad X_\emptyset = 0_{n \times m} \\
\text{otherwise } \forall P \in \mathcal{P}(m), \quad X_P &= J_P^{-1} - \sum_{Q \subseteq P} X_Q
\end{align*}
\]

\[ J_P = H_0 J \] with \( H_0 \) a diagonal matrix whose diagonal components \( h_i \) are equal to 1 if \( i \in P \), and to 0 otherwise.

This inverse is proved to have two nice properties. It is continuous with respect to the variation of the activation matrix \( H \) and it is equal to the classical inverse \((HJ)^+ H\) when the component of \( H \) are binaries \((h_i = 0 \text{ or } h_i = 1)\). This means that the resulting control law is continuous and keeps the same behavior and all the properties of local convergence of the corresponding classical control law.

Finally, using this inverse, the control law that applies the task \( e \) upon activation \( H \) is the following \([14]\):

\[ \dot{q} = J^{H\oplus} \hat{e}^* \quad (10) \]

Using (10) we can verify that each component \( e_i \) of \( \hat{e}^* \) is:

- perfectly realized if the corresponding \( h_i \) is equal to 1.
- not taken into account if \( h_i \) is zero.
- partially realized otherwise.

B. Extension to two tasks

As in the classical redundancy case, (10) is not the unique solution of least norm. The general solution can be written:

\[ \dot{q} = \dot{q}_1 + P_\oplus \dot{q}_2 \quad (11) \]

where \( \dot{q}_1 = J^{H\oplus} \hat{e}^* \) and \( P_\oplus = I - J^{H\oplus} J \). Let us now consider two tasks \((e_1, H)\) and \( e_2 \). Introducing (11) in (1), we obtain:

\[ \dot{e}_2 = \dot{q}_1 + J_2 P_\oplus \dot{q}_2 \quad (12) \]

We now search the optimal \( \dot{q}_2 \) that performs \( e_2 \). Inverting directly this last equation, by analogy to (6) would give us the following control law:

\[ \dot{q} = \dot{q}_1 + P_\oplus (J_2 P_\oplus)^+ (\dot{e}_2^* - J_2 \dot{q}_1) \quad (13) \]

However, the matrix \( P_\oplus \) does not have a constant rank. Therefore, this control law will lead to the same discontinuities than the classical control laws, each time the rank of \( P_\oplus \) changes. Instead, we recognize a form similar to (4), with \( P_\oplus \) at the place of the activation matrix. We will apply once more the continuous inverse operator to solve the discontinuities. However, it is not possible to apply directly this operator since \( P_\oplus \) is not diagonal. We will thus first generalize the inverse operator for such a matrix.

C. Continuous inverse activated by a non diagonal matrix

Let \( Q \) be any matrix of size \( m \times n, n \leq m \) and \( W \) be a positive symmetric matrix of size \( m \) whose singular value are all between 0 and 1. We note \( U, \sigma \) the eigenvalue decomposition of \( W \), \( \sigma \) being the vector of eigen values, and \( S = \text{diag}(\sigma) \) the corresponding diagonal matrix \((W = USU^T)\). Using this decomposition, we can write:

\[ (WQ)^+ W = (SQ_u)^+ SU \quad (14) \]

where \( Q_u = U^T Q \).

We now recognize the previous form (4), with \( Q \) the matrix to be inverted, and \( S \) the activation matrix. This time, the matrix \( S \) is a diagonal matrix with components in \([0, 1]\). The continuity to the change of rank of \( S \) is obtained by applying the continuous inverse of \( Q_u \) activated by \( S \). We thus defined the continuous inverse of a matrix \( Q \) activated by any (non-diagonal) positive symmetric matrix whose eigenvalues are in \([0, 1]\) by the following:

\[ Q^{\oplus}\!W = Q_u^{\oplus}\!S U = \sum_{P \in \mathcal{P}(n)} \left( \prod_{i \in P} \sigma_i \right) X_P Q_u^{\oplus} U \quad (15) \]

where \( X_P Q_u \) are the coupling matrix of \( Q_u \).

Until now, we have only defined the continuous inverse activated on the left (that is to say, the activation matrix \( H \) acts on the rows of the matrix by multiplying on the left \( HQ \)). We can easily generalize this notion to the inverse activated on the right (i.e. where the activation matrix acts on the columns), noted \( Q^{\oplus\!\!\top} \):

\[ Q^{\oplus\!\!\top} = ((Q^\top)^{\oplus H})^\top = \sum_{P \in \mathcal{P}(n)} \left( \prod_{i \in P} h_i \right) X_P^\top \quad (16) \]

where the \( X_P \) are the coupling matrices of \( Q^\top \). Finally, the continuous inverse of \( Q \) activated on the right by a symmetric matrix \( W \) is defined similarly \( Q^{W\oplus} = ((Q^\top)^{\oplus W})^\top \).

D. Extension to \( k \) tasks

Considering the two tasks \((e_1, H)\) and \( e_2 \) of Section III-B, the control law (12) can now be replaced using this generalization. The continuous inverse is valid since the eigenvalues of \( P_\oplus \) are in \([0, 1]\) (this is obtained by proving that \( \forall x, x^\top P_\oplus x \leq 1 \), by developing the sum (16)). The control law that performs \( e_1 \) under activation \( H \) and \( e_2 \) if possible is then:

\[ \dot{q} = J_1^{\oplus H} e_1^* + J_2 P_\oplus (e_2^* - J_2 \dot{q}_1) \quad (17) \]

using the notations \( \dot{q}_1 \) and \( P_\oplus \) defined in (11). The matrix \( P_2^{\oplus\!\!\top} \) corresponding to the projection operator is obtained classically by:

\[ P_2^{\oplus\!\!\top} = P_\oplus - J_2 P_\oplus \quad (18) \]

The control law (17) can then be extended to \( k \) tasks \((e_1, H), e_2 \ldots e_k \) using the following recursion:

\[
\begin{cases}
\dot{q}_0 = 0 \\
\dot{q}_i = \dot{q}_{i-1} + (J_i)^{P_i^{\oplus\!\!\top}} (e_i^* - J_i \dot{q}_{i-1}), & i = 1 \ldots k
\end{cases}
\]

(19)
A similar recursion is used to compute the operators $P^i_\oplus$:

$$P^i_\oplus = P^{i-1}_\oplus - J_i P^{i-1}_\oplus J_2, \quad i = 1..k$$ (20)

and $P^0_\oplus = P_\oplus$.

### E. Conclusion

In this section, we have built a new control law to take into account a unilateral constraint at the first level of a hierarchy of tasks. This control law ensure the respect of the constraint and ensure also the continuity of the control law when the unilateral constraint is activated or unactivated.

All the control law have been written supposing that a set of local controllers provides a control of the robot in velocity $\dot{q}$. In the following section, we extend this result to the operational space control.

### IV. Extension to the Operational Space Control

Operational Space Control [11] has been proposed to compute a control law using directly the torque of the motor as a control input. Contrary to the inverse-kinematic-based control schemes, operational space control unify at the global level in a same formalism the forces applied by the robot and its displacement in the free space. Using this formalism, it is thus possible to set up any behavior in an unique control law.

In this section, we first recall the generic form of the control law in the operational space for a set of tasks. We is thus possible to set up any behavior in an unique control law.

All the control law have been written supposing that a set of local controllers provides a control of the robot in velocity $\dot{q}$. In the following section, we extend this result to the operational space control.

#### A. Classical control law

1) One task: The acceleration of the robot joints $\ddot{q}$ in free space is defined by the following well-known equation:

$$\mathbf{A}\ddot{q} + \mathbf{g} + \mu = \tau$$ (21)

where $\mathbf{A}$ is the inertia matrix, $\mathbf{g}$ is the gravity force, $\mu$ are the Coriolis forces and $\tau$ are the torques applied by the motors on the joints, used as the control input.

Given a task $e$ with Jacobian $\mathbf{J}$, it is possible to write:

$$\ddot{e} + b = J\Lambda^{-1}\tau$$ (22)

with $b = J\Lambda^{-1}(g + \mu) - J\dot{q}$. We define $\Omega = J\Lambda^{-1}\mathbf{J}^\top$ and $\Lambda = \Omega^{-1}$. By multiplying (22) by $\Lambda$, we obtain:

$$\Lambda\ddot{e} + \Lambda b = \Omega^{-1}\tau$$ (23)

where $\Omega^{-1}\tau$ is the generalized inverse of $\Omega$ weighted by $\Lambda^{-1}$. We can note that $\Omega^{-1}$ is the general-
ized inverse of $\Omega$ weighted by $\Lambda^{-1}$: $\Omega^{-1} = \Omega^{\Lambda^{-1}}$. We thus have $\Omega^{-1}\tau = I$. By multiplying (23) by $\mathbf{J}^\top$, we finally obtain the control law that provides the reference acceleration $\ddot{e}$:

$$\tau = \mathbf{J}^\top\Lambda(\ddot{e} + b)$$ (24)

2) Two task: As in the previous section, this control law is only a specific solution: this is the solution of the least acceleration energy [18]. The general solution is [11]:

$$\tau = \mathbf{J}^\top\Lambda(\ddot{e} + b) + \mathbf{N}^{\top}\tau_2$$ (25)

where $\tau_2$ is an arbitrary vector and $\mathbf{N}^\top = \mathbf{I} - \mathbf{J}^\top\Lambda\mathbf{J}^{\Lambda^{-1}} = \mathbf{I} - \mathbf{J}^\top\mathbf{J}_1$ is the projection operator that ensures the realization of $\ddot{e}$ despite $\tau_2$.

3) $k$ task: The secondary control entry $\tau_2$ can be used to perform a secondary task under the condition that the main task $\ddot{e}_1$ is perfectly realized. The control scheme can then be extended by recurrence to a set of $k$ tasks. Given a set of $k$ tasks $e_1 ... e_k$, the control law that performs the tasks while preserving the hierarchy is [21] ($i = 1..k$):

$$\tau_i = \tau_{i-1} + J_i\Lambda_{ii-1}(\ddot{e}_i + b - J_i\Lambda^{-1}\tau_{i-1})$$ (26)

with $\tau_0 = 0$, $\Omega_{ii-1} = J_i\Lambda^{-1}N_{i-1}^\top J_i$ and $\Lambda_{ii-1} = \Omega_{ii-1}^{-1}$. The control law to be applied on the robot is finally $\tau = \tau_k$. The projection operator is computed by:

$$\mathbf{N}_i^\top = N_{i-1}^\top - J_i^\top\Lambda_{ii-1}\mathbf{J}_i = N_{i-1} - J_i^\top\mathbf{J}_1$$ (27)

where $\mathbf{J}_1 = \mathbf{A}^{-1}\mathbf{J}\Lambda_{ii-1}$.

#### B. Unilateral constraint in the Operational Space

Let us now consider a task $(e, \mathbf{H})$ with $e$ the task function and $\mathbf{H}$ the activation matrix. We want to determine the torque entry to perform $e$ under activation $\mathbf{H}$. Applying directly the classical formalism with jacobian $\mathbf{HJ}$ leads to:

$$\tau = \mathbf{J}^\top\mathbf{H}(\mathbf{HJ}\mathbf{A}^{-1}\mathbf{J}^\top\mathbf{H})^{-1}\mathbf{H}(\ddot{e} + b)$$ (28)

As in the kinematic space, this control law produces some strong discontinuities and improper behaviors when the rank of $\mathbf{HJ}$ changes, in particular when $\mathbf{HJ}$ is ill-conditioned due to some small activation values in $\mathbf{H}$. In this last equation, we recognize the form $\mathbf{J}^\top\mathbf{H}(\mathbf{HJ}\mathbf{A}^{-1}\mathbf{J}^\top\mathbf{H})^{-1}$

$= (\mathbf{HJ}\mathbf{A}^{-1})^{\#\mathbf{A}}$, the inverse of $\mathbf{HJ}\mathbf{A}^{-1}$ weighted by the matrix $\mathbf{A}$. The matrix used in the control law is thus $(\mathbf{HJ}\mathbf{A}^{-1})^{\#\mathbf{A}}\mathbf{H}$, whose form is similar to (4). To prevent the discontinuity, we use as previously the continuous inverse proposed in [14]. The control law is then:

$$\tau = (\mathbf{J}^\top\mathbf{A}^{-1})^{\#\mathbf{H}}^{\#\mathbf{A}}(\ddot{e} + b)$$ (29)

where $(\mathbf{Q})^{\#\mathbf{H}}^{\#\mathbf{A}}$ denotes the generalized continuous inverse of $\mathbf{Q}$ activated by $\mathbf{H}$ and weighted by $\mathbf{A}$, defined by replacing all the classical pseudo-inverse in (8) and (9) by the corresponding generalized inverse weighted by $\mathbf{A}$. By developing the sum (8) in (29) and factorizing by $\mathbf{J}^\top$, we finally obtain the control law on the following form:

$$\tau = \mathbf{J}^\top\Lambda_\oplus(\ddot{e} + b)$$ (30)

where $\Lambda_\oplus = \Omega^{\#\mathbf{H}}$. A secondary term can be taken into account using the projection $\mathbf{N}^\top_\oplus = \mathbf{I} - \mathbf{J}^\top\Lambda_\oplus\mathbf{J}^{\Lambda^{-1}} = \mathbf{I} - \mathbf{J}^\top\mathbf{J}_1$, with $\mathbf{J}_1 = \mathbf{A}^{-1}\mathbf{J}\Lambda_\oplus$. The control law is then:

$$\tau = \mathbf{J}^\top\Lambda_\oplus(\ddot{e} + b) + \mathbf{N}^\top_\oplus\tau_2$$ (31)

$\tau_2$ being any arbitrary torque.
C. Extension to 2 tasks

This secondary torque \( \tau_2 \) can be used to perform a secondary task. Let \((e_1, H)\) and \(e_2\) be two tasks. The control law performing \( e_1 \) is \( (31) \). By introducing this control law in \((21)\), we obtain the equation of motion of the robot constraint by the main task:

\[
A\ddot{q} + g + \mu = J^\top A_{\oplus}(\dot{e}^* - b) + N_{\oplus}^\top \tau_2
\]  

(32)

By multiplying this last equation by \( J_2 A^{-1} \), we obtain the motion of the robot constrained by \( e_1 \) and expressed in the space of the task \( e_2 \):

\[
e_2^* + b_2 = J_2 A^{-1} \tau_1 + J_2 A^{-1} N_{\oplus}^\top \tau_2 \quad \text{(33)}
\]

where \( b_2 = J_2 A^{-1}(g + \mu) \) and \( \tau_1 \) is the control law defined in \((30)\) applied to \( e_1 \). It is very tempting to directly inverse \((33)\) to obtain the optimal control \( \tau_2 \) performing \( e_2 \), as it has been done in \((26)\), obtaining the following control law:

\[
\tau = \tau_1 + N_{\oplus} (J_2 A^{-1} N_{\oplus})^{-A}(e_2^* - b_2 - J_2 A^{-1} \tau_1)
\]  

(34)

However, the operator \( N_{\oplus} \) does not have a constant rank. It is thus necessary to proceed like in II-C, by activated the inverse by the projection operator \( N_{\oplus} \). The generalization of the continuous inverse is only valid for symmetrical matrices whose singular values are between 0 and 1. This is not the case of \( N_{\oplus} \), since this matrix is not an orthogonal projection operator. However, we can rewrite the inverse weighted by \( A \) under the following form:

\[
N_{\oplus} (J_2 A^{-1} N_{\oplus})^{-A} = N_{\oplus} \sqrt{A} (J A^{-1/2} (A^{-1/2} N_{\oplus} \sqrt{A}) + 1
\]  

(35)

It is therefore possible to normalize \( N_{\oplus} \) by setting:

\[
N_{\oplus} = A^{-1/2} N_{\oplus}^\top \sqrt{A} = I - (J_1 A^{-1/2}) H_{\oplus} (J_1 A^{-1/2})
\]  

(36)

Using the second part of this equation, it is easy to demonstrate the \( N_{\oplus} \) is normalized that is to say symmetrical and with proper singular values. The inverse \((35)\) is then finally

\[
\sqrt{A} N_{\oplus} (J_2 A^{-1/2} N_{\oplus})^{-A} \quad \text{and}
\]

\[
N_{\oplus} (J_2 A^{-1/2} N_{\oplus})^{-A} = (J_1 A^{-1/2})^\top (J_1 A^{-1/2})
\]

D. Extension to \( k \) tasks

This control law can easily be extended to a set of task \((e_1, H), \ldots, e_k\) by analogy to the developments done in \([21]\):

\[
\tau_i = \tau_{i-1} + \sqrt{A} (J_i A^{-1/2})^\top (e_i^* - b_i + J_i A^{-1} \tau_{i-1})
\]  

(38)

for any \( i = 1 \ldots k \). The recurrence is initialized with \( \tau_0 = 0 \). The control law to be applied on the robot is finally \( \tau = \tau_k \). A second recurrence is used to compute the projection operator:

\[
N_{\oplus} = N_{\oplus} - (J_1 A^{-1})^\top \sqrt{A} J_i A^{-1}
\]  

(39)

E. Conclusion

The control law \((38)\) generalizes the use of an unilateral constraint for an arbitrary set of tasks in the operational space. The final form is very close to the classical control law \((26)\). Moreover, very simple computations show that the control law \((38)\) is equal to \((26)\) when the singular values of \( N_{\oplus} \) are all 0 or 1.

Until this point, all the computations have been realized for any set of tasks. In the following section, we will introduce the specific set of tasks that we have used in the experiments.

V. APPLICATION TO VISUAL SERVOING

The framework presented above has been experimentally validated by realizing a visual-servoing task while respecting the constraints imposed by the joint limits. In this section, we present the tasks and the constraints that have been used in the experiments.

A. Visual servoing task

Visual servoing control law techniques \([9], [6]\) provide very efficient solutions to control robot motions. It supplies high positioning accuracy, good robustness to sensor noise and calibration uncertainties, and reactivity to environment changes. A visual servoing task is based on an error \( e_1 = s_i - s_i^* \) where \( s_i \) is the current value of the visual features \( e_1 \) and \( s_i^* \) their desired value \([6]\). The interaction matrix \( L_{s_i} \) related to \( s_i \) is defined so that \( s_i = L_{s_i} v \), where \( v \) is the instantaneous camera velocity. From the definition of \( e_1 \), it is clear that the interaction matrix \( L_{s_i} \) and the task Jacobian \( J_1 \) are linked by the relation:

\[
J_1 = L_{s_i} M J_q
\]  

(40)

where the matrix \( J_q \) denotes the robot Jacobian at the camera focal point, \( (\dot{r} = J_q q) \) and \( M \) is the matrix that relates the variation of the camera velocity \( v \) to the variation of the chosen camera pose parametrization \( (v = M \tilde{F}) \).

We have used visual features derived from the image moments. We consider a target with \( n_p \) clearly marked points. At each iteration, let \( P_i = (x_{i,j}, y_{i}) \) be the position of the points in the image. The moment \( m_{i,j} \) of the image is defined by \( m_{i,j} = \sum_{k=1}^{n_p} x_k y_k^j \).

A task of dimension 3 has been defined to control the position of the target and its dimension in the camera field of view. The two first feature are based on the position of the center of gravity: \((x_g, y_g) = (\frac{x_{min}}{m_{00}}, \frac{y_{min}}{m_{00}})\). The third feature \( a_Z \) uses the centered moments of order 2 to control the range between the robot and the target. The most intuitive solution is to consider the quadrilateral area, i.e. the first moment of the continuous object. Since the considered object is discrete, discrete centered moments of second order are used, as proposed in \([24]\). The reader is invited to refer to \([24]\) for more details.

B. Joint limits avoidance

The robot lower and upper joint limits for each axis \( i \) are denoted \( q_i^{\min} \) and \( q_i^{\max} \). The robot configuration \( q \) is
acceptable if, for all \( i \), \( q_i \in [q_i^{\text{min}}, q_i^{\text{max}}] \). Given \( n \) axes, the joint limits can be expressed by a set of \( n \) unilateral constraints:

\[
\forall i = 1..n, \quad q_i > q_i^{\text{min}} \quad \text{and} \quad q_i < q_i^{\text{max}}
\]  

The corresponding unilateral task can be written \( \tau_{0} = \mathbf{e}_{0}^{T} \mathbf{H}_{0} \mathbf{x} \), where \( \mathbf{e}_{0} = \mathbf{q}_{0}^{N} = (q_{0i}^{N})_{i=1..n} \) is the normalized joint state \( q_{0i}^{N} = 2 \frac{q_{0i} - q_{0i}^{\text{min}}}{q_{0i}^{\text{max}} - q_{0i}^{\text{min}}} - 1 \in [-1, 1] \). The activation matrix \( \mathbf{H}_{0} \) is defined by its diagonal components:

\[
h_i = \begin{cases} 1 & \text{if } e_i < -1 \text{ or } e_i > 1 \\ f_{\beta}(\beta - 1 - e_i) & \text{if } e_i \in [-1, -1 + \beta] \\ f_{\beta}(\beta - 1 + e_i) & \text{if } e_i \in [1 - \beta, 1] \\ 0 & \text{otherwise} \end{cases}
\]  

where \( \beta \) is the size of the transition interval, used as a gain to tune the robot behavior, and \( f_{\beta} \) is the transition function, defined to provide a \( C^\infty \) transition from 0 to 1:

\[
\forall x \in [0, \beta], \quad f_{\beta}(x) = \frac{1}{2} \left( 1 + \text{tanh} \left( \frac{1}{x/\beta - \beta} \right) \right)
\]  

This transition function has the very nice property to be \( C^\infty \) everywhere, including at the joining points 0 and \( \beta \). In the following, we will call activation buffer the area defined by \( f_{\beta} \) where the activation function passes from 0 to 1. The activation function \( h_i \) is plotted on Fig. 1.

VI. EXPERIMENTS AND RESULTS

Three experiments are presented here: the two first ones have been realized in simulation. We can thus test control laws that should not be tested on a real robot (due to joint limit violation or dangerous oscillations) and compare them to our control scheme. The third experiment has been realized on a Puma robot.

A. Experiments in simulation

We consider a 7-DOF PA-10 robot in simulation. The task to be executed is a 3-DOF positioning of an embedded camera with respect to a visual target. The joint limits avoidance is ensured by adding the corresponding task at the higher-priority position. We finally add at the lowest-priority level a posture task (to ensure the stability of the robot, all the DOF have to be constrained).

1) Experiment 1: at first, the target is positioned so that the task is feasible while respecting the joint limits. However, the shortest path to the desired position overcome the limits (see Fig. 2). We compare the execution using simply the pseudo-inverse like in (4) and using the continuous inverse proposed in (38). It is possible to ensure that the joint limits are avoided by using directly the pseudo-inverse as shown by Fig. 3. However, the control law is jerky. As explained in the first section, the behavior of the robot changes dramatically as soon as it enters in the activation buffer. The robot thus leaves immediately the area where the joint limit is taken into account. This kind of "entering-leaving" oscillation can be clearly observed on the axes 6 and 7. This oscillations are also clearly identifiable on the control law on axes 6 and 7 but also on axes 1 and 2. A last drawback is evident on the joint trajectories (Fig. 3-top): the robot cannot practically enter inside the activation buffer. In fact, it is stuck at the entrance of the buffer, where the oscillations appear. The robot is thus limited in practice to a smaller part of its articular domain. On the opposite, when using the control law (38), the control law is smooth (see Fig. 4). The joint limits are avoided while performing the task. No oscillation appears during any part of the displacement. Moreover, all the joint domain is used (axes 3 and 6 can freely move inside the activation buffer).

All the previous motion have been realized in free space. Finally, we apply a force on Axis 4 of the robot to disturb the control law. Since four DOF are still available, the robot can move freely. As shown by Fig. 5, Axis 4 moves freely. Some other axes follow the motion to maintain the task
completed. At \( t=11.4 \), Axis 7 reaches its joint limits. The robot thus loses one DOF, which consequently locks Axis 4. The robot is not compliant in the direction of the external force anymore. Axis 7 remains in the buffer while the force is applied on Axis 3. When the force is relaxed, the robot posture changes to take Axis 7 out of the activation buffer.

2) Experiment 2: the target is now positioned so that the required motion is not feasible inside the joint limits. When the robot reaches an equilibrium, we then move the target inside the joint-limit boundary to check that the control law is able to reach the reference position when this one is reachable. We first try to put the positioning task at the top-priority level. The joint limits avoidance are not prioritary anymore, and we cannot ensure that they will be respected. We have experimentally verified that in such a situation, the gain of the avoidance task as to be strongly increased to reach a level comparable to the gain of the positioning task. In our case, this correspond to multiply the gains by 4. As expected, we can see on Fig. 6 that the limits are violated. The robot reaches the desired position despite the joint limits by violating them. We now put the avoidance task back at the highest priority rank. We compare the execution of the continuous control law (38) on Fig. 7 with the results obtain when using directly the pseudo inverse on Fig. 8. Control law (38) produces a smooth behavior. The robot stops at the limit of its articular domain, as close as possible to the desired position. When the target is moved back inside the reachable domain, the robot moves freely away from its limits to reach the desired position. On the opposite, the control law using the pseudo inverse is jerky and could not be applied on a real robot.

Fig. 4. Positioning the joint limits while ensuring the joint limit avoidance using the continuous control law (38): the joint limits are avoided and the control law is smooth.

Fig. 5. Motion due to an external force applied on Axis 4, while applying the 3-DOF positioning task and the joint limits. The force started at time \( t=10.2 \) and ends at \( t=12.4 \)

Fig. 6. Avoidance task with low priority. At time \( t=1 \), the robot is required to reach a position out of its joint limits. Since the positioning task has priority over the avoidance task, the joint limits are overcome during the displacement, which corresponds to a failure of the execution.

Fig. 7. Joint limits prevent the execution of an non-feasible task. Like in Fig. 6 the task required at time \( t=1 \) is not feasible. The robot reaches at time \( t=6 \) an equilibrium position close to the desired position but inside the joint limits. At time \( t=8 \), the reference position is moved back inside the articular domain. The robot reaches the reference position without any problem.

Fig. 8. Oscillation while using the pseudo-inverse. Like in Fig. 7, the desired position between \( t=1 \) and \( t=8 \) is not reachable. At time \( t=8 \), the reference position is put back inside the articular domain. The robot is stuck at the border of the activation buffer. Moreover, the control law is completely jerky, and definitely not acceptable for a real robot.
scheme while insuring the respect of the joint limits. On different types of robot with a common visual servoing could be applied for a various set of tasks, constraints the actions of the tasks. The proposed scheme is generic that the unilateral constraints will be respected whatever unilateral constraints and bilateral tasks. This scheme ensures compute a generic control law from a hierarchic set of both

B. Experiment on the Puma robot

We have then validated our control scheme on a real robot. As in simulation, a camera is attached to the end effector. The task to be accomplished is the same than in simulation. As for the experiment described in 5, an external force is applied on the third joint of the robot to disturb it during the execution of the task. The robot first move freely in the direction of the applied force (Axis 3 moves freely). The wrist (Axis 6) moves to compensate the motion and execute the task. The robot start to resist when Axis 6 reaches its limit. Axis 4 can then be used instead of 6. When both Axes 3 and 4 reaches the activation buffer, the robot resists to the external force.

VII. CONCLUSION

In this paper, we have proposed an original scheme to compute a generic control law from a hierarchic set of both unilateral constraints and bilateral tasks. This scheme ensures that the unilateral constraints will be respected whatever the actions of the tasks. The proposed scheme is generic and could be applied for a various set of tasks, constraints and robots. We have demonstrate the validity by applying it on different types of robot with a common visual servoing scheme while insuring the respect of the joint limits.

ACKNOWLEDGMENT

This work is partially supported by grants from the ROBOT@CWE EU CEC project, Contract No. 34002 under the 6th Research program www.ROBOT-AT-CWE.EU. We would like also to thank particularly Jaeheung Park (park56@stanford.edu) and Luis Sentis (sentis@cs.stanford.edu) for valuable comments and help for both the theoretical part and the experimental part of this article.

REFERENCES