The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Constraints Compliant Control: constraints compatibility and the displaced configuration approach

Sébastien Rubrecht, Vincent Padois, Philippe Bidaud and Michel de Broissia

Abstract— **Most of the literature reactive control laws have much difficulties to handle properly constraints such as joint limits, obstacles and saturations, including them as equalities in the Inverse Velocity Kinematics (IVK) problem. Actually, it seems relevant to handle them through inequalities, as the constraints are more numerous than the number of DOFs. However, the intuitive constraints expression can lead to incompatibilities between constraints, which necessarily induces constraints violations. This paper brings two distinct contributions. First, the usual constraints inequalities are modified to make them compatible permanently. Second, an intuitive and efficient constraint compliant control law is proposed.**

I. Introduction

Practical applications in Robotics require a strict compliance to constraints while achieving the operational task. This is especially true in cases where constraints violation is highly critical (nuclear plants decommissioning, surgery,...). For example, this work is motivated by a project involving teleoperated maintenance in a tunnel boring machine. To ensure this strict compliance and a minimum level of operational performance, constraints and tasks can be handled offline through planning. However this approach is not viable when operational tasks or constraints are not known a priori. This is typically the case in applications involving teleoperation, co-manipulation or autonomous behaviors, i.e. a reactive behavior. To the best of our knowledge, there is no control approach in the literature allowing to ensure this strict compliance to constraints in general cases and the work in this paper is a contribution in that direction.

The constraints considered in this paper are as follows:

- No collision with the environment (static);
- No collision with joint (position) limits;
- No saturation of joints velocities and accelerations.

These constraints can be incompatible, i.e. the respect of one of them may induce the violation of another one. For example, the proximity of an obstacle or of a joint limit may require a too high deceleration value for one of the joints. Thus, it is necessary to formalize set of constraints equations accounting for this dependence. This **constraints compatibility** problem is not very often tackled in the literature. Once constraints compatibility is achieved, any control law ensuring the **compliance to constraints while leading to the best possible achievement of the operational tasks** can be chosen.

In the case of redundant manipulators, the control problem is classically formulated at the velocity kinematics level as the inversion of the following relation

$$\dot{X} = \frac{\partial X}{\partial q}\dot{q} = J(q)\dot{q} \qquad (1)$$

where $J(q)$, $\dot{X}$ and $\dot{q}$ are respectively the operational task Jacobian matrix, the operational velocity vector of size m and the joint velocity vector of size n. This problem has received a lot of attention (see for example [1], [2] and [3] for an overview). However the retained approaches are rarely constraints compliant since either operational tasks are given the highest priority while redundancy is used to deal with the constraints or constraints are given the highest priority but expressed as equalities. Since the n DOFs of a robot can only instantaneously comply with n equality conditions, the latter approach is doomed to fail given, for example, the $2n$ constraints associated to joint limits.

To deal with this strong limitation, the most general approaches consist in formulating the control problem as a convex optimization one where constraints are naturally expressed both through equalities and inequalities ([4], [5] and [6]). This type of approach induces, in the case of complex systems with multiple hierarchical levels, computation times which may not be suitable for a real-time implementation (even though some very recent work described in [7] exhibits a rather low computational complexity). Iterative approaches have also been proposed among which the one from Baerlocher et al. in [8] that introduces the idea of joint clamping and passive avoidance for joint limits constraints only. Passive avoidance is a principle according to which avoiding a constraint should not need a motion but rather an absence of motion in the concerned direction. Based on the coupling between active and passive avoidance, the work in [9] extends Baerlocher's approach to obstacle avoidance constraints. In this work, joints velocities constraints are treated using a scaling approach.

Joint acceleration constraints are more problematic since they are the one which induce possible incompatibilities between constraints. In [6], a framework is introduced to include any type of inequality and equality constraints. The presented results involve an incompatibility between

S. Rubrecht, V. Padois and P. Bidaud are with:
Université Pierre et Marie Curie
Institut des Systèmes Intelligents et de Robotique - CNRS UMR 7222
Pyramide Tour 55 - Boite Courrier 173
4 Place Jussieu, 75005 Paris, France
S. Rubrecht and M. de Broissia are with:
Bouygues Travaux Publics
78280 Guyancourt, France
Contact: rubrecht@isir.upmc.fr

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

acceleration and position constraints which is locally treated. However, the problem remains an open one.

Our first contribution is to formalize constraints so that compatibility is implicitly preserved (section II). While this formalization is general enough to be used by any constraint compliant control law such as the one proposed in [6], our second contribution is to propose an intuitive and more direct constraint compliant control law including joint accelerations limits (section III). The obtained performances are exposed and discussed in section IV. Section V is dedicated to conclusions and future works.

II. Constraints compatibility

This section aims at establishing constraints expressions that are compatible, *i.e.* inequalities that, beyond describing the constraints, also satisfy

$$\forall k, \exists \dot{q}(k), q(k) + \dot{q}(k)\delta t \in E_C(k) \tag{2}$$

where $q(k)$ is the robot configuration at iteration k, $\dot{q}(k)$ is the joint velocities at iteration k (solution of the control law problem) and $E_C(k)$ is the interval of *admissibility*. $E_C(k)$ can be seen as the interval in which lies the potentially feasible configurations $q(k+1)$ at the next iteration.

As one can one notice, a discrete formulation is retained to describe the instantaneous behavior of the robot at the joint level, i.e. $q(k+1)$ is approximated as $q(k) + \dot{q}(k)\delta t$. This is a widely acknowledged assumption in the literature.

$E_C(k)$ is the intersection of the admissibility interval of each constraint

$$E_C(k) = \bigcap_p (E_{C,p}(k)) \tag{3}$$

where $E_{C,p}(k)$ is the admissibility interval of constraint p. If (2) is not satisfied, then there is no possible motion complying with the constraints, and thus at least one constraint will be violated.

As there is no assumption on $\dot{q}(k)$, (2) is equivalent to

$$\forall k, E_C(k) \neq \varnothing. \tag{4}$$

The following part gives intuitive constraints expressions that are not compatible. Then, these expressions are modified to integrate their compatibility.

A. Intuitive constraints expression

The considered constraints are usually expressed by inequalities obtained based on (1) (*e.g.* in [5]):

- **Collision with the environment**

$$J_A \dot{q}(k) < \frac{d_{A,B}(k)}{\delta t} \tag{5}$$

for all pairs of point (A,B), where A belongs to the robot and B to the environment; $d_{A,B}$ is the distance between A and B; J_A is the jacobian (single line) of point A in direction $A \rightarrow B$. For a practical aspect, this infinite set of constraints is reduced to one constraint per segment of the robot (shortest distance).

- **Joint position limit (for the ith joint)**

$$\begin{aligned} J_{c_i^+}\dot{q}(k) &< \frac{q_{M,i} - q_i(k)}{\delta t} \\ \text{and } J_{c_i^-}\dot{q}(k) &< \frac{q_i(k) - q_{m,i}}{\delta t} \end{aligned} \tag{6}$$

where $J_{c_i^+} = [0, \ldots, 0, 1, 0, \ldots, 0]$ (the i^{th} term being 1) and $J_{c_i^-} = [0, \ldots, 0, -1, 0, \ldots, 0]$. $q_{m,i}$ and $q_{M,i}$ are respectively the minimum and the maximum joint position limits. The admissibility interval w.r.t. position limits is represented on Fig. 1.

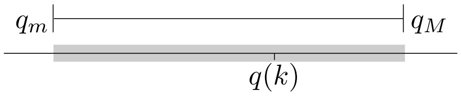

Fig. 1. Joint position limits admissibility interval in the joint space.

- **Joint velocity limit (for the ith joint)**

$$\begin{aligned} J_{c_i^+}\dot{q}(k) &< \dot{q}_{M,i} \\ \text{and } J_{c_i^-}\dot{q}(k) &< -\dot{q}_{m,i} \end{aligned} \tag{7}$$

where $\dot{q}_{m,i}$ and $\dot{q}_{M,i}$ are respectively the minimum and the maximum velocities of joint i. The admissibility interval w.r.t. velocity limits is represented on Fig. 2.

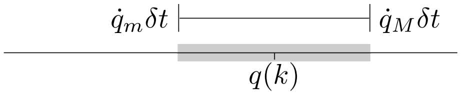

Fig. 2. Joint velocity limits admissibility interval in the joint space. The values written beside the interval limits are the expression of the boundaries w.r.t. the neutral constraint configuration.

- **Joint acceleration limit (for the ith joint):** it is obtained from the discrete time derivative of the joint velocity expression

$$\begin{aligned} J_{c_i^+}\dot{q}(k) &< \ddot{q}_{M,i}\delta t + \dot{q}_i(k-1) \\ \text{and } J_{c_i^-}\dot{q}(k) &< -\ddot{q}_{m,i}\delta t - \dot{q}_i(k-1) \end{aligned} \tag{8}$$

where $\ddot{q}_{m,i}$ and $\ddot{q}_{M,i}$ are respectively the minimum and the maximum accelerations of joint i. The admissibility interval of acceleration limits is represented in Fig. 3.

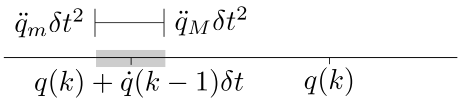

Fig. 3. Joint acceleration limits admissibility interval in the joint space. $q(k)$ does not necessarely belongs to it. The values written beside the interval limits are the expression of the boundaries w.r.t. the neutral constraint position.

These constraints, when satisfied, ensure the admissibility of the robot motion at the current iteration. However, if the admissibility intervals associated to these constraints become disjoint (incompatibility), at least one constraint will be violated at the next iteration. As an example, it is the case when a joint gets close to one of its position limits with a high velocity: the acceleration limits does not enable a sudden stop and the constraints become incompatible (cf. Fig. 4).

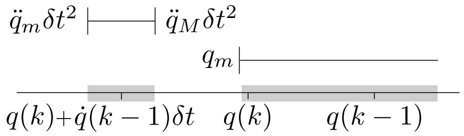

Fig. 4. A constraint incompatibility. The admissibility intervals of the joint position (right) and the joint acceleration (left) limits are disjoint: $q(k+1)$ will necessarily violate a constraint. The values written beside the intervals are the expression of the boundaries w.r.t. the neutral constraint position.

To avoid these cases, the compatibility between constraints is considered as a constraint in itself. However, there is no simple way to express the compatibility conservation; we propose to study the mutual impacts of the constraints to modify adequately their expressions in order to make them always compatible.

B. Methodology to study the constraints compatibility

When not considering the accelerations limits, each constraint is an interval (or a semi-interval for obstacles) that contains the current configuration $q(k)$. Thus, the space $E_C(k)$ is a convex space at least reduced to a point; so, these constraints are always compatible. The introduction of the joint acceleration limits endangers the compatibility (cf. Fig. 4). To proceed, we propose to study the addition of the joint acceleration limits in the different cases of constraints sets containing joint position limits, joint velocity limits and obstacles. This decomposes the problem into 7 cases.

In this framework, a reliable estimation of the deceleration capabilities on a time horizon is needed. The joint deceleration limits can be known (at least minorated) permanently, but it is in general not possible to ensure a minimal value for an operational deceleration (cf. II-D.1). So, the study is divided in two parts: first, the 3 cases involving only joint related constraints (position, velocity and acceleration limits) are treated. Then, the 4 cases involving obstacles are discussed.

C. Compatibility of sets without obstacle constraints

1) Joint acceleration and velocity limits: As in (8), the admissibility interval of the acceleration limits has its neutral value on $q(k)+\dot{q}(k-1)\delta t$ (cf. Fig 5). These constraints does not require any modification to be always compatible: for a given joint i, the size of $E_C(k)$ is at least $\ddot{q}_{M,i}(k)\delta t^2$ (joint at minimum velocity) or $\ddot{q}_{m,i}(k)\delta t^2$ (joint at maximum velocity).

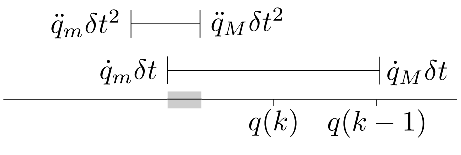

Fig. 5. Compatibility between joint velocity and acceleration limits. Illustration of the minimal intersection width at minimum (negative) velocity.

2) Joint acceleration and position limits: These constraints may become incompatible if a joint gets close to one of its position limits too fast (cf. Fig. 4): the admissible deceleration may not be sufficient to avoid the contact with the joint position limit. The only way to preserve compatibility is to slow down when getting close to a position limit. As shown in Annex A, ensuring

$$\begin{aligned}\dot{q}(k) &< \frac{(q_{max}-q(k))-\frac{1}{2}(s_1^2-s_1)\ddot{q}_m\delta t^2}{(s_1+1)\delta t}\\ \text{and } \dot{q}(k) &< \frac{(q_{min}-q(k))-\frac{1}{2}(s_2^2-s_2)\ddot{q}_M\delta t^2}{(s_2+1)\delta t}\end{aligned} \tag{9}$$

with

$$\begin{aligned}s_1 &= -\frac{\sqrt{-2\ddot{q}_m(q_{max}-q(k))}}{\ddot{q}_m\delta t}\\ s_2 &= \frac{\sqrt{-2\ddot{q}_M(q_{min}-q(k))}}{\ddot{q}_M\delta t}\end{aligned} \tag{10}$$

enables to decelerate the joint so that maximal deceleration and joint position limits are satisfied. In particular, it is ensured that the point of maximal deceleration $q_{\text{decM}}(k)$ always belongs to $E_C(k)$. Its expression is

$$q_{\text{decM}}(k) = q(k)+\dot{q}_{\text{decM}}(k)\delta t \tag{11}$$

where the maximal deceleration velocity $\dot{q}_{\text{decM}}(k)$ components are:

if $\dot{q}_i(k-1)>0$

$$\dot{q}_{\text{decM},i}(k) = \max(\dot{q}_i(k-1)+\ddot{q}_{m,i}\delta t, 0) \tag{12}$$

else

$$\dot{q}_{\text{decM},i}(k) = \min(\dot{q}_i(k-1)+\ddot{q}_{M,i}\delta t, 0). \tag{13}$$

3) Joint acceleration, velocity, and position limits: The inclusion of joint velocity limits in the previous case does not affect the existence of $E_C(k)$ as the modifications carried out for acceleration and position limits maintain the existence of $E_C(k)$ in the deceleration part (velocity decreasing) of the acceleration admissibility intervals. As the velocity limits only have an impact on the acceleration part (velocity increasing) of this interval, there is no particular modification to bring.

D. Compatibility of sets involving obstacles

1) Joint acceleration limits and obstacles: As in part II-C.2, these constraints become incompatible if the robots gets close to an obstacle too fast. So, the equations of Annex A may be applied. However, the acceleration term is not constant (obtained from (5))

$$\ddot{X}_{A,B} = J_A(q)\ddot{q}+\dot{J}_A(q)\dot{q} \tag{14}$$

($\ddot{X}_{A,B}$ being the acceleration of point A in the direction $A\rightarrow B$) as J_A and $\dot{J}_A$ are functions of q. The variations of $\ddot{X}_{A,B}$ imply a possible decrease of the decelerations capabilities. As the future operational inputs are not known (reactive framework), any prediction based strategy may turn out time consuming. Nonetheless, any decrease w.r.t. estimated accelerations capabilities is not allowed, as the modifications to ensure the compatibility (33) need these accelerations to be maintained. So, another strategy is adopted.

Given a robot point A close to an obstacle, a stop distance is computed

$$d_{stop,A}(k) = G_A(q_{stop}(k)) - G_A(q(k)) \tag{15}$$

where $G_A()$ is the forward kinematics model of the robot point A projected in the direction of the closest obstacle and $q_{stop}(k)$ is the configuration obtained when decelerating from $\dot{q}(k)$ to 0 at full joint deceleration. $d_{stop,A}(k)$ is a rough approximation of the distance that may be needed to decelerate if the joints are submitted to multiple constraints (other obstacles for example). In such context, a cautious principle is to suppose that each articulation may be submitted to a maximal deceleration at any time. Unfortunately, $d_{stop,A}(k)$ is not an upper bound of the maximum distance traveled in the obstacle directions during the deceleration motion. Actually, antagonists velocities (joints having separately opposite effects on the operational motion direction) may bring the robot briefly further from $q(k)$ than $q_{stop}(k)$. However, this is complex to determine since it would require to compute the volume described by the robot during the deceleration and then to compute the minimum distances between the obtained volume and the environment. The problem is simplified by resorting to an envelope e on the environment to increase safety.

To make the acceleration limits and the obstacles constraints compatible, the following test is carried out:
for each obstacle, if

$$d_{obstacle}(k) - d_{stop,A}(k) < e \tag{16}$$

then the following inequality is imposed

$$J_{ob}\dot{q} \leq \dot{q}_{\mathrm{decM}}(k) \tag{17}$$

where J_{ob} is the diagonal matrix containing ±1 on the joints number which associated element is not null in the jacobian line of the obstacle constraint. The term is 1 if $\dot{q}_i(k-1) > 0$, -1 otherwise.

This expression imposes a strict deceleration on all the joints having a motion contribution in the obstacle direction. (17) directly acts at the joint level (no limit is imposed on the operational velocity in the constraint direction) as the redundancy of the task associated to an operational limit does not enable to ensure the desired deceleration, again because of antagonist joints that may decelerate at any time. We mention that the test described by (16) and (17) is not exactly adequate as it deals with distances without taking into account the effect (17) on velocities. So, oscillations may be encountered.

2) Obstacles and joint acceleration and velocity limits: For the same reason than for part II-C.3, there is no further modification to bring to (17).

3) Acceleration limits, joint limits and obstacles: The presence of joint position limits with obstacles and acceleration limits does not require any modification, as they can be considered as obstacles of jacobians $J_{c_i^+}$ or $J_{c_i^-}$. We can mention that (9) and (17) ensure that $E_C(k)$ contains the configuration of maximum deceleration $q_{\mathrm{decM}}(k)$. So, these constraints are already compatible without any further modification.

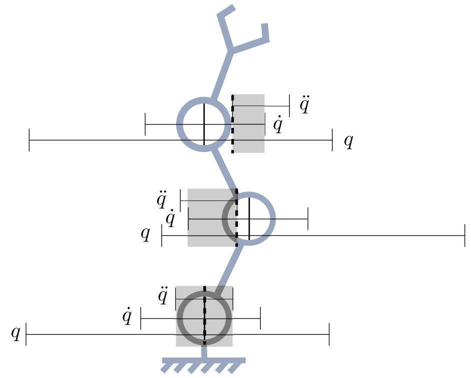

Fig. 6. 3R manipulator with, for each articulation, its admissibility intervals of joint position, velocity and acceleration limits. The vertical line at the center of each articulation is the current joint position. $E_C(k)$ is the grey zone, and the maximum deceleration configuration $q_{\mathrm{decM}}(k)$ is represented by dashes.

4) Obstacles and joint acceleration, velocities and position limits: As mentioned previously, the velocity limits impact the acceleration part (velocity increasing) of the acceleration limits, whereas the measures taken to respect acceleration limits, joint limits and obstacles maintain an admissible interval in the deceleration part of this constraint.

E. Summary

The constraints expression have been modified to preserve their compatibility:

- For sets involving joint position, velocity and accelerations limits, the compatibility impacts the expression of the joint position limits. The adopted approach guarantees a permanent compatibility between constraints. Any control method ensuring the compliance with these constraints ((7), (8), (9)) yields a safe behavior.
- For sets involving obstacles, joint position, velocity and accelerations limits, the difficulty to estimate the decelerations capabilities leads to a cautious approach that involves an action at the joint level (17) activated by a test at the operational level (16).

When considering these modifications, the compatibility is now preserved. In particular, $E_C(k)$ contains at least the point of maximal deceleration $q_{\mathrm{decM}}(k)$.

An example of the first case is represented on Fig. 6. These equations can now be included in any constraint compliant control law process.

III. Control Law

In this section, a constraint compliant control law that fully takes advantage of the constraints formulation is proposed. It resorts to a virtual configuration called *displaced configuration* to adapt the usual operational control scheme so that it can satisfy all the constraints enunciated previously.

A. Constraints compliance

The scheme of a usual reactive control law algorithm is described on algorithm 1. In general, the way to comply with the constraints in step 3 is to scale the solution obtained at step 2 by the maximum velocity exceedance (as in [10]). Actually, this method can be applied in the cases of constraints set without acceleration. In these cases, as $E_C(k)$ contains the current configuration $q(k)$, whatever the solution $\dot{q}(k)$ found in step 2

Algorithm 1 : General operational space control law scheme

```
1. Computation of $\dot{X}_{des}$ from $X_{des} - X$
2. Jacobian Inversion: $\dot{q} = IVK(J, \dot{X}_{des})$
3. Operations to comply with constr. not included in 2.
4. Send $\dot{q}$ to low level controller
```

$$\exists p \in [0,1], q(k) + \dot{q}_{(p)}(k)\delta t = q(k) + p\dot{q}(k)\delta t \in E_C(k) \quad (18)$$

and the solution $\dot{q}_{(p)}$ is thus admissible. It is important to have a common p for all joints, so that the operational directions are maintained (coherent scaling).

For joint acceleration limits, the fundamental difference is that the current configuration $q(k)$ does not always belong to the admissibility interval. As a consequence, for any solution $\dot{q}(k)$, (18) is not always satisfied. Thus, the simple scaling method cannot be used when considering acceleration limits. In order to use such a method, the use of a displaced configuration is addressed.

B. Use of a displaced configuration

Scaling can be used when the current configuration lies in the admissibility interval $E_C(k)$. As the expressions of the boundaries of $E_C(k) = [q_{b,m}, q_{b,M}]$ are known at each iteration (by calculating the intersection of the constraints admissibility intervals), it is possible to work from a displaced configuration $q_b(k)$. Adapted joint velocities can be computed and then scaled to satisfy the constraints. Even in the worst case of infinite scaling ($p = 0$ in (18)), the solution

$$\dot{q}(k) = \frac{q_b(k) - q(k)}{\delta t} + p\dot{q}_b(k) \quad (19)$$

is admissible. The use of this configuration in the control law modifies the usual algorithm according to algorithm 2.

Algorithm 2 : Control iteration using the displaced configuration

```
1. Choice of $q_b \in [q_{b,m}; q_{b,M}]$; $X_b \leftarrow g(q_b)$
2. Computation of $\dot{X}_{des,b}$ from $X_{des} - X_b$
3. Jacobian Inversion: $\dot{q}_b = IVK(J, \dot{X}_{des,b})$
4. Operations to comply with the constraints
5. Send $\dot{q}$ to low level controller
```

C. Choice of the displaced configuration

There are different possibilities for the choice of $q_b(k)$ in $E_C(k)$. Without any information on the operational inputs at the next iteration, a good compromise is to take $q_b(k)$ at the center of $E_C(k)$, that is for each articulation i

$$q_{b,i}(k) = \frac{q_{(b,m),i}(k) + q_{(b,M),i}(k)}{2}. \quad (20)$$

This choice leaves an equal space of admissible motion on both side of $q_{b,i}(k)$. It is important to keep some admissible space around $q_b(k)$ as the motion is scaled by the maximum exceedance of motion beginning at $q_b(k)$ w.r.t. the constraints limits (cf. (24)). This has been implemented in the first simulation of part IV.

Another possibility is to place q_b at the maximal deceleration configuration as it belongs to $E_C(k)$ (cf. part II-C.2)

$$q_b(k) = q_{\mathrm{decM}}(k). \quad (21)$$

On the one hand, it favors joint accelerations in the direction of the previous velocity which constitutes an *a priori* on the operational task. On the other hand, it is less efficient for operational motions with sudden direction changes. However, this effect can be minimized thanks to iterations, as explained in part III-D.3. This has been implemented in the second simulation of part IV.

D. Modified control algorithm

1) Step 1: Computation of $q_b(k)$ and $X_b(k)$: The choice of $q_b(k)$ is exposed in part III-C. The computation of $X_b(k)$ is obtained by directly applying the forward kinematics model to $q_b(k)$.

2) Step 2: Computation of $\dot{X}_{des,b}(k)$: $\dot{X}_{des,b}(k)$ is computed in the same way than $\dot{X}_{des}(k)$ in the initial algorithm, but taking $X_b(k)$ instead of $X(k)$. We propose

$$\dot{X}_{des,b}(k) = \max(h, ||X_{b\to des}(k)||) \frac{X_{b\to des}(k)}{||X_{b\to des}(k)||\delta t} \quad (22)$$

where $X_{b\to des}(k) = X_{des}(k) - X_b(k)$ and h is an operational scale factor to avoid large operational inputs so that the assumption of small displacements implied by linearization is valid.

3) Step 3: Inverse Kinematics: This step aims at computing the joint velocities $\dot{q}_b(k)$ that minimizes the error $||\dot{X}_{des,b}(k) - J\dot{q}_b(k)||$. The jacobian is assumed to be constant between $q(k)$ and $q_b(k)$, so no jacobian computation is done at configuration $q_b(k)$.

In order to take into account the motion capabilities of each articulation at the current iteration, the inverse velocity kinematics is weighted in the joint space ([11]) by the width of each articulation admissibility interval. This technique enables to balance the joints contributions to the operational motion w.r.t. their capabilities.

Two control laws are used in the simulations (part IV). In the first simulation (case of 2 hierarchized sets of tasks $(J_1, \dot{X}_1)$ and $(J_2, \dot{X}_2)$), the velocity kinematics inversion is done by

$$\dot{q}_b(k) = J_1^{\#}\dot{X}_{1,b}(k) + (J_2 P_{J_1})^{\#}(\dot{X}_{2,b}(k) - J_1^{\#}\dot{X}_{1,b}(k)) \quad (23)$$

where P_{J_1} is the projector on the kernel of J_1 computed thanks to a Singular Value Decomposition.

In the second simulation, the control law used is the one introduced in [9]. Briefly, this control law uses an extra level at the top of the hierarchy to include constraints to be avoided passively (no motion in these constraints direction). Iterations on the constraints combinations are carried out to find which constraints need to be avoided passively to comply with all constraints while minimizing the trajectory tracking error. In our case, the constraints chosen to be avoided passively are the one which exceedance is too high in the scaling.

E. Step 4: Compliance with the constraints

This step aims at making the solution calculated at step 3 $\dot{q}_b(k)$ compliant with the constraints. This operation is done by scaling, and as mentioned previously, it is important to perform it on the basis of $q_b(k)$ as it belongs to $E_C(k)$ (cf. part III-A). The general constraints expression is

$$J_C\dot{q}(k) \leq L \tag{24}$$

where J_C is the concatenation of all the jacobian constraints and L the concatenation of all associated limits ((7), (8), (9), (17)). The introduction of $\dot{q}_b(k)$ yields

$$J_C\dot{q}_b(k) \leq L - J_C\frac{q_b(k)-q(k)}{\delta t} \triangleq L_b. \tag{25}$$

Then, if (25) is not verified, the scaling is done by

$$\dot{q}_b(k) \leftarrow \dot{q}_b(k) \min_{\substack{0\leq i\leq j \\ (J_C\dot{q}(k))_i>0}} \left(\frac{L_i}{(J_C\dot{q}(k))_i}\right) \tag{26}$$

where j is the number of constraints.

IV. Simulations and Results

The simulations presented here illustrate the behavior of control laws using the displaced configuration. The first simulation is a simple example of a 3R planar manipulator dealing with constraints of joint position, velocities and acceleration limits. The second simulation involves a 7R manipulator in 3D that handles the same constraints than the first simulation and obstacles.

A. 3R planar manipulator

1) Simulation presentation: The objective of this simulation is to check on a simple but redundant manipulator that the considered constraints are always satisfied and that the manipulator performs properly the tasks.

The considered manipulator is a 3R planar manipulator. Its bodies lengths are (0.4*m*, 0.3*m*, 0.2*m*). Its joint limits are $\pm\frac{\pi}{2}$ *rad* for all DOFs; its velocity limits are ± 1 $rad.s^{-1}$ for the first and second DOFs and ± 1.5 $rad.s^{-1}$ for the third DOF; its acceleration limits are ± 5 $rad.s^{-2}$ for all DOFs. The manipulator is represented on Fig. 7. The tracked trajectory is a 2D point (no orientation) moving toward the basis of the manipulator. The mean distance between two points of the trajectory is around 3 *mm*. As the robot has 3 DOFs, it is redundant w.r.t to this trajectory tracking task. However, a second objective is arbitrarily applied on the third joint (blue arrow on Fig. 7) between $t = 0.6$ s and $t = 1$ s to check that a sudden jacobian rank change does not lead to a velocity discontinuity ([3]). Then, the robot reaches its position limits, making the constraints and the task incompatible. Finally, at $t = 3s$, the trajectory is left for a new operational objective (green point on Fig. 7) to check that the robot is able to move away from its constraints.

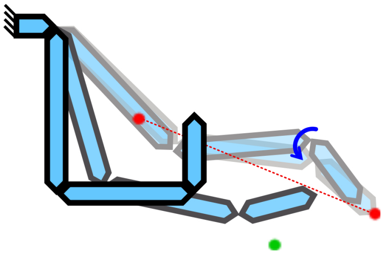

Fig. 7. The 3R robot motion sequence.

The displaced configuration is at the center of $E_C(k)$ (cf. (20)). The operational scaling coefficient h (cf. (22)) is 10 $m.s^{-1}$. The control law is (23) (the trajectory tracking being first priority objective). When the second objective is active (between $t = 0.6$ s and $t = 1$ s), $J_2 = [0 \quad 0 \quad 1]$ and $\dot{X}_2 = \alpha(q_{des} - q_3)$ where $\alpha = 30$ and $q_{des} = 0.5$ rad. The iteration increment is δt = 0.01 s.

2) Results: The simulation sequence is represented on Fig. 7. The graphs of positions, velocities and accelerations for each joint are represented on Fig. 8. Every constraint is always satisfied (grey horizontal lines). As expected, the joint positions and velocities are smooth, and the velocity slopes are bounded. Even if the manipulator reaches its limits, the control problem is solved in a unique iteration.

The displaced configuration is fundamental to continue the motion while some joints are at their position limit. Actually, without it, as the velocity kinematics inversion involves every joints, the motion exceedances on joints 1 and 2 would be infinite and the motion would be scaled to 0. The use of the displaced configuration at the center of $E_C(k)$ moves q_b away from the position limits, so the motion is not scaled to 0. However, it can happen that this method turns out to be insufficient to carry out properly the task. In these cases, any approach iterating on the passive avoidance of constraints (as in [8] and [9]) can be used.

B. 7DOFs manipulator

1) Simulation presentation: This simulation is an extension of the previous one as it involves the 3D manipulator with 7 DOFs represented on Fig. 9. The obstacles avoidance is added to the set of constraints. The aim of this simulation is to illustrate the behavior of the control law with a displaced configuration at the maximum deceleration point (cf. (21)) in a complex case.

The joint positions of the robot are limited to $\pm\frac{\pi}{2}$ *rad* , the velocities ± 1.5 $rad.s^{-1}$ and the accelerations ± 10 $rad.s^{-2}$.

As in IV-A, the tracked trajectory is a 3D point (no orientation) that by-passes the wall (Fig. 9). The mean distance between two points of the trajectory is 8 *mm*. As in

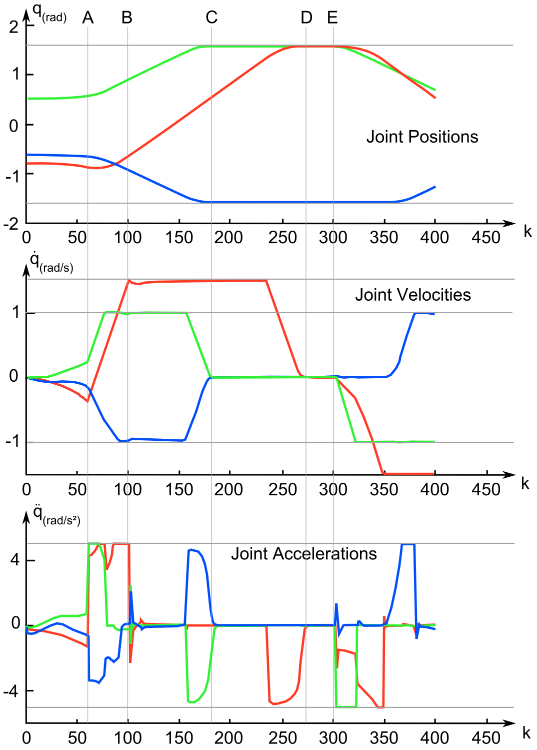

Fig. 8. Joints position, velocity and acceleration w.r.t iterations. Colors: DOF1: red; DOF2: green; DOF3: blue.

the previous simulation, the trajectory is not feasible, *i.e.* the manipulator is geometrically not able to track it perfectly.

The displaced configuration is the maximal deceleration configuration (cf. (21)) to better carry out smooth operational tasks. The environment envelope e is 1 cm. The operational scaling coefficient h (cf. (22)) is 10 $m.s^{-1}$. The control law used is the one introduced in [9] with the trajectory tracking as unique objective. The maximum scaling exceedance above which constraints are avoided passively is $ex_{max} = 10^3$. The iteration increment is $\delta t = 0.01\ s$.

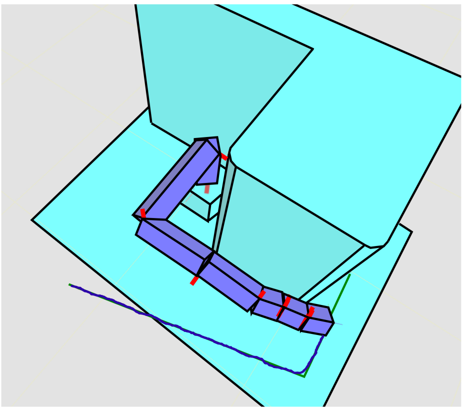

Fig. 9. 7 DOFs manipulator in the last configuration of the simulation. The representation of the joint axis are in red.

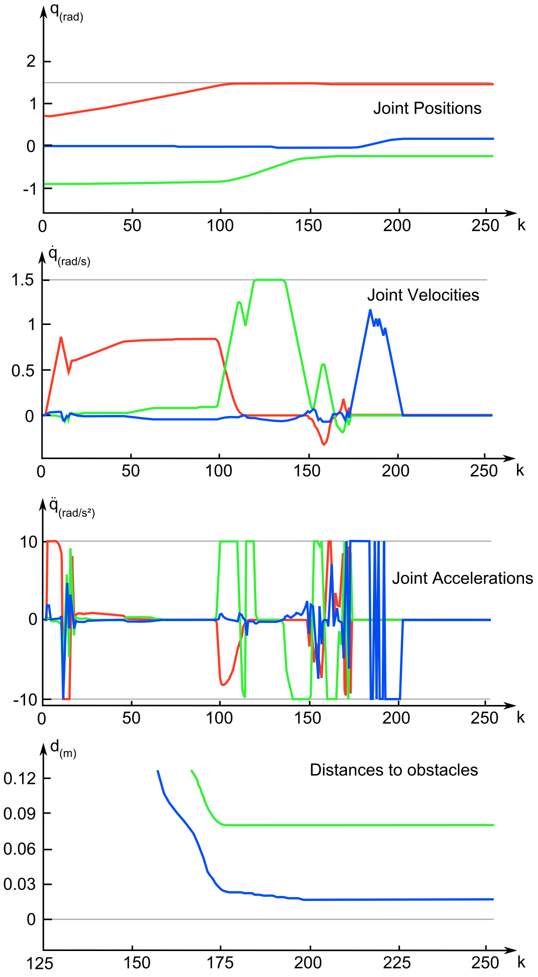

Fig. 10. Joints positions, velocities and accelerations w.r.t iterations. Distances to obstacles. Colors: DOF1: red; DOF4: green; DOF7: blue.

2) Results: The graphs of joints positions, velocities and accelerations of DOFs 1, 4 and 7 as well as the distances to the nearest point of the environment are represented on Fig. 10. As in the previous experiment the positions and the velocities are smooth, and every constraint is always satisfied. At iteration 95, the first DOF reaches its position limits. At iteration 135, the robot begins its deceleration close to the obstacle. Finally, all the DOFs but the last one are stuck by the obstacle (iteration 175) but the last DOF continues to minimize the tracking error until it reaches the envelope of the environment. As expected (part II-D.1), the last DOF is submitted to small velocity oscillations. Their impact can be visualized on the graph of Fig. 10 representing the distance to obstacles.

V. Conclusions and Future Work

In this paper we address the problems of constraints compatibilities and constraints compliant control.

Expressions of constraints that preserve the compatibility are established. As a consequence, any constraint compliant control law can integrate this formulation in order to ensure a permanent satisfaction of the contraints.

Then, an original and efficient control approach is developped. It takes a displaced configuration as the working configuration for inverting the velocity kinematics problem. The displaced configuration is chosen in the interval admissible w.r.t the constraints, so that the scaling method can be applied. As a consequence, this controller can handle constraints sets including joint acceleration limits.

The efficiency of the proposed approach is illustrated by two simulation cases.

Future work will focus on the estimation of the operational deceleration capabilities of the robot. This is needed to provide a more appropriate constraints compatibility expression and could be achieved by prediction of the robot expected motions during finite time horizon.

VI. Acknowledgements

This work is a part of the TELEMACH project and is supported by the PSIROB program of the French National Research Agency (ANR).

Appendix

A. Discrete calculus

Let $(q(k))_k$, $(\dot{q}(k))_k$, and $(\ddot{q}(k))_k$ be the sequences of joint positions, velocities and accelerations. In discrete time, the motion is modelized by

$$\begin{aligned} q(k+1) &= q(k)+\dot{q}(k)\delta t \\ \dot{q}(k+1) &= \dot{q}(k)+\ddot{q}(k)\delta t. \end{aligned} \tag{27}$$

As shown in [6], in case of constant acceleration $\ddot{q}_m$, the position evolution in s iteration is

$$q(k+s) = q(k)+s\dot{q}(k)\delta t+\frac{1}{2}(s^2-s)\ddot{q}_m\delta t^2. \tag{28}$$

If we suppose $\dot{q}(k)>0$ and $\ddot{q}_m<0$ (deceleration example), the condition $q(k+s)<q_{max}$ for all integer s leads to

$$\dot{q}(k) < \frac{q_{max}-q(k)}{s\delta t}-\frac{1}{2}(s-1)\ddot{q}_m\delta t. \tag{29}$$

The minimization of the right member can be obtained by relaxing the integer optimization problem ($s \rightarrow s_{\mathbb{R}}$) and differentiation this expression w.r.t $s_{\mathbb{R}}$

$$\left.\begin{array}{rl} s_{\mathbb{R}} & \geq 0 \\ -\frac{q_{max}-q(k)}{s_{\mathbb{R}}^2\delta t}-\frac{1}{2}\ddot{q}_m\delta t & =0 \end{array}\right\} \Leftrightarrow s_{\mathbb{R}} = -\frac{\sqrt{-2\ddot{q}_m(q_{max}-q(k))}}{\ddot{q}_m\delta t} \tag{30}$$

which then can be solved by finding the integer value $s_{\mathbb{N}}$ that minimizes the maximum velocity in (29). However, if $s_{\mathbb{R}}<1$, the method retained in [6] ($\dot{q}(k)\leq 0$) may violate the acceleration limit constraint. Moreover, this method is tight and may fail in case of any measure error.

The expression of the deceleration distance obtained from the relaxed expression is

$$d_{\mathbb{R},dec}(k) = s_{\mathbb{R},dec}\dot{q}(k)\delta t+\frac{1}{2}(s_{\mathbb{R},dec}^2-s_{\mathbb{R},dec})\ddot{q}_m\delta t^2 \tag{31}$$

with $s_{\mathbb{R},dec}$ given by equation (30). It is obvious that $d_{\mathbb{R},dec}(k)>d_{\mathbb{N},dec}(k)$ where $d_{\mathbb{N},dec}(k)$ would be the exact discrete deceleration distance at iteration k. To satisfy the constraints, a cautious condition is

$$d(k+1) > d_{\mathbb{R},dec}(k) \tag{32}$$

where $d(k+1)$ is the distance $q_{max}-q(k+1)$. It leads to

$$\dot{q}(k) < \frac{(q_{max}-q(k))-\frac{1}{2}(s_{\mathbb{R},dec}^2-s_{\mathbb{R},dec})\ddot{q}_m\delta t^2}{(s_{\mathbb{R},dec}+1)\delta t}. \tag{33}$$

The comparative deceleration of a joint with the 2 methods can be observed on Fig. 11.

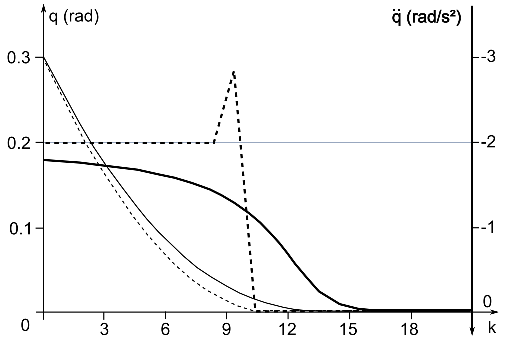

Fig. 11. Distances to boundary (thin) and accelerations (bold) for the velocity limits proposed in [6] (dashed) and in this work (plain). Numerical values: $\ddot{q}_m=-2\ rad.s^{-2}$ and $\delta t=0.05\ s$.

References

[1] A. Liegeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7(12), 1977, pp. 868–871.

[2] L. Sentis and O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, *The International Journal of Humanoid Robotics*, vol. 2(4), 2005, pp. 505-518.

[3] N. Mansard, O. Khatib and A. Kheddar, A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks, *IEEE Transactions on Robotics*, vol. 25(3), 2009.

[4] B. Faverjon, P. Tournassoud, A local based approach for path planning of manipulators with a high number of degrees of freedom, *In proceddings of the IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 1152–1159.

[5] F. T. Cheng, R. J. Sheu, and T.-H. Chen, The Improved Compact QP Method For Resolving Manipulator Redundancy, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25(11), 1995, pp. 1521–1530.

[6] W. Decré, R. Smits, H. Bruyninckx and J. De Schutter, Extending iTaSC to support inequality constraints and non-instantaneous task specification, *In proceedings of the IEEE International Conference on Robotics and Automation*, 2009, pp. 964–971.

[7] A. Escande, N. Mansard and P-B. Wieber, Fast Resolution of Hierarchized Inverse Kinematics with Inequality Constraints, *In proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[8] P. Baerlocher, and R. Boulic, An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels, *The Visual Computer: International Journal of Computer Graphics*, vol. 20(6), 2004, pp. 402-417.

[9] S. Rubrecht, V. Padois, P. Bidaud and M. de Broissia, Constraint Compliant Control for a Redundant Manipulator in a Cluttered Environment, *In proceedings of the 12th conference on Advances in Robot Kinematics*, 2010, pp. 367-376.

[10] V. Padois, J-Y. Fourquet and P. Chiron, Kinematic and dynamic model-based control of wheeled mobile manipulators: a unified framework for reactive approaches, *Robotica*, vol. 25, 2007, pp. 157–173.

[11] A. Ben Israel and T. N. E. Greville, Generalized Inverses: Theory and Applications.*Springer (2nd edition)*, 2003.