Motion safety and constraints compatibility for multibody robots — Source link ↗

Sébastien Rubrecht, Vincent Padois, Philippe Bidaud, Michel De Broissia ...+1 more authors

Institutions: Bouygues, Pierre-and-Marie-Curie University, Commissariat à l'énergie atomique et aux énergies alternatives

Related papers:

- Design and motion constraints of part-mating planning in the presence of uncertainties

- A computationally efficient safety assessment for collaborative robotics applications

- A Control Barrier Function Approach to Human-multi-robot Safe Interaction

- Algorithmic safety measures for intelligent industrial co-robots

- Prioritized motion-force control of multi-constraints for industrial manipulators

Motion safety and constraints compatibility for multibody robots

Sébastien Rubrecht, Vincent Padois, Philippe Bidaud, Michel de Broissia, Max da Silva Simoes

Motion safety and constraints compatibility for multibody robots

Sébastien Rubrecht · Vincent Padois · Philippe Bidaud · Michel de Broissia · Max Da Silva Simoes

Abstract In this paper we propose a methodology to ensure safe behaviors of multibody robots in reactive control frameworks. The permanent satisfaction of constraints being insufficient to ensure safety, this approach focuses on the constraints expression: the compatibility between these constraints is studied, and safe alternatives are ensured when compatibility cannot be established. A complete case study involving obstacles, joint position, velocity and acceleration limits illustrates the approach. A particular method is developed to take full advantage of a smooth state of the art avoidance techniques (Faverjon *et al.* 1987) while maintaining safety. Experiments involving a 6-DOF manipulator operating in a cluttered environment illustrate the reliability of the approach and validate the expected performances.

S. Rubrecht, M. de Broissia
Bouygues Travaux Publics
St Quentin en Yvelines, France
E-mail: s.rubrecht@bouygues-construction.com,
m.debroissia@bouygues-construction.com

V. Padois, P. Bidaud
Université Pierre et Marie Curie
Institut des Systèmes Intelligents et de Robotique
Paris, France
E-mail: vincent.padois@isir.upmc.fr,
philippe.bidaud@isir.upmc.fr

M. Da Silva Simoes
Commissariat à l'Energie Atomique et aux Energies Alternatives, CEA, LIST, Interactive Robotics Laboratory
Fontenay aux Roses F-92265, France
E-mail: max.simoes@cea.fr

1 Introduction

The starting point of the present work deals with the use of a robotic arm in charge of maintenance tasks in the inner part of the excavation room of a tunnel boring machine (Fig. 1). The robot is teleoperated to perform inspections and cleaning tasks in a fix and cluttered environment. The control of such a robot is reactive (teleoperation is not compatible with offline planning) and subject to real time constraints (force feedback requires control loop frequencies higher than 500 Hz). Although the features of this problem are common in the literature, safety issues remain. For example, most of the collisions avoidance methods do not take the system dynamics into account, which may cause collisions in tight environments and high speed motions. As the robot is subject to kinematic and dynamic constraints that cannot be ignored, it is important to take the physical properties of the system and its environment into account to ensure motion safety.

1.1 Safety criterions for control

The notion of *safety* for a system is a principle applied at various levels. At the design level, safety is often integrated directly in the system (Ikuta *et al.* 2003, Zinn *et al.* 2004, Haddadin *et al.* 2010). At the control level, the work related to offline optimal trajectory planning is closely linked to joint constraints management (Brady 1982, Biagiotti *et al.* 2008); in spite of the context differences[1], their recent adaptations to online frameworks (Kröger 2010) exhibit some similarities with reactive

---

[1] most of these approaches are exclusively concerned with joint physical limits as operational constraints are managed by path planning
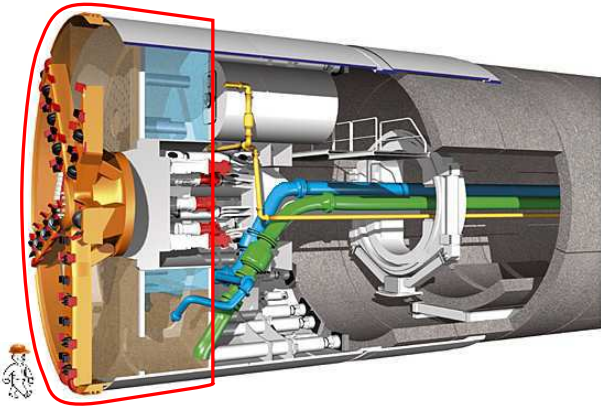
Fig. 1 Tunnel Boring Machine. The red line encircles the cutter head and the excavation room, which is the manipulator's working area

control techniques. In a strictly reactive context, safety has been neglected for a long time. Recently, Fraichard (2007) proposed 3 criteria to ensure safety:

1. to decide its future motion, a robotic system should consider its own dynamics;
2. to decide its future motion, a robotic system should consider the environment objects future behavior;
3. to decide its future motion, a robotic system should reason over an infinite time-horizon.

In case of a static environment (not known *a priori*), only the first criterion stands: the future behavior of the objects in the environment remains identical to the current one. The third criterion can thus be integrated in the first one if the consideration of the system's own dynamics is done over an infinite time-horizon (referred later as the *extended criterion 1*).

1.2 Safety of common approaches for collisions avoidance with multibody robots

The number of constraints being potentially higher than the number of DOF, the usual *active avoidance techniques* (approaches for which the avoidance requires a motion) involved in multi-objectives frameworks (issued from Khatib *et al.* (1986) and Maciejewski *et al.* (1985)) cannot lead to safety. Moreover, they do not involve dynamics in the avoidance magnitude computation.

Faverjon and Tournassoud (1987) proposed an avoidance technique included in a Quadratic Programming (QP) control law structure. This method limits the velocities toward obstacles by inequalities (*passive* avoidance), which is more likely to avoid the collisions whatever the number of obstacles. QP are now widely used in manipulators or humanoids control (Decré *et al.* 2009,

Escande *et al.* 2010), but the avoidance methods still do not include dynamics on an infinite time-horizon. It results that, to our knowledge, no control law for multibody robot passes the extended criterion 1.

In fact, most of the research work related to safety at the control level is led in the field of mobile robotics, *i.e.* single body mobile robots avoiding collisions: models are simpler, and the operational capabilities predictions are easier (operational deceleration limits do not depend on the robot configuration for example). As an example, the Dynamic Window Approach (DWA) (Fox *et al.* 1997) involves the acceleration limits of a mobile robot and ensures its safety in a fix environment (extended criterion 1). More recent developments in this domain are part of the framework based on the notion of Inevitable Collision State proposed by Fraichard *et al.* (2004): *e.g.* Martinez-Gomez *et al.* 2009, Althoff *et al.* 2010, Bautin *et al.* 2010.

To the best of our knowledge, although this framework could be used to assess the safety of a wider scope of applications, 1/ it has never been applied to multibody robots; 2/ it is limited to collisions avoidance with respect to dynamics, which can be formulated as the compatibility between the constraint of geometric collisions avoidance and acceleration limits. However, these are just two constraints among the many constraints that have to be faced in robotics: joint position, velocity, acceleration and torque limits (joint space), collisions with obstacles and forbidden regions (Cartesian space), contacts conservation constraints (Park *et al.* 2008), comanipulation and cooperation (Khatib *et al.* 2001), actuators temperature limits (Guilbert *et al.* 2008), etc. We can conclude that there is still a lack regarding robots safety (in particular for multibody robots) when considering a large variety of constraints.

1.3 Constraints compatibility

All these constraints can be considered at the velocity kinematics level for example, where the model is traditionally formulated as

$$\dot{\boldsymbol{X}}_{des}(t) = \frac{\partial \boldsymbol{X}_{des}(t)}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}}(t) = J_T(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t) \qquad (1)$$

where $J_T(\boldsymbol{q})$, $\dot{\boldsymbol{X}}_{des}$, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are respectively the operational task Jacobian matrix (size (m,n)), the operational desired velocity vector (size m), the robot configuration (size n) and the joint space velocity vector (size n). The QP formulation of the control problem has the advantage to explicit the constraints that are supposed
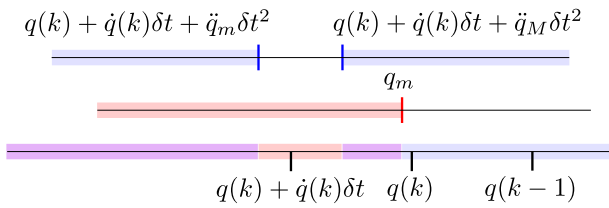
Fig. 2 Incompatibility between constraints represented in the joint space. The terms $q(k) + \dot{q}(k)\delta t + \ddot{q}_m(k)\delta t^2$ and $q(k) + \dot{q}(k)\delta t + \ddot{q}_M(k)\delta t^2$ are respectively the configurations induced by a full acceleration and full deceleration (approximation based on finite differences). The subscripts $_m$ and $_M$ denote respectively the minimum (negative value) and the maximum (positive value) limit of the considered variable. The control admissibility space of joint acceleration limits (up) and minimum joint position limit (middle) cannot be satisfied simultaneously: $q(k+1) \approx q(k) + \dot{q}(k)\delta t$ will inevitably violate one of the constraints

to be satisfied by the robot

$$\min_{\dot{q}(k+1)\in\mathbb{R}^n} ||\dot{X}_{des}(k+1) - J_T(q(k))\dot{q}(k+1)|| \quad (2)$$

$$\text{subject to} \qquad J_C\dot{q}(k+1) - b(k) \leq 0 \quad (3)$$

where k is the current time step, $\dot{q}(k+1)$ is the velocity vector chosen for the next time step (control input vector), J_C is the Jacobian of constraints (size (p, n)) and $b(k)$ is the constraints limit vector (size p). The feasibility of the problem (*i.e.* the existence of $\dot{q}(k+1)$ such that $J_C\dot{q}(k+1)-b(k) \leq 0$) is usually taken for granted; however, if the constraints expressions have not been carefully set-up, incompatibilities may occur. The typical case is the joint position limit violation because of limited accelerations illustrated by Fig. 2. If a joint gets close to one of its position limits with a high velocity, its deceleration capabilities may not be sufficient to avoid the collision with the boundary (joint position limit). As an example, a maximum deceleration $2\ rad/s^2$ imposed on a joint moving at $1.0\ rad/s$ requires $0.5\ s$ to actually stop; then, distance travelled is $0.25\ rad$. This example illustrates the fact that satisfying at each time the joint position and the joint acceleration limits does not prevent from a constraint violation due to incompatibility. Usually, virtual envelopes are set up around the physical limits to absorb such violations. These envelopes do not guarantee safety and often artificially limit the performances of the robot. Relying on a safe approach taking dynamics into account would enable to reduce significantly those envelopes.

The contribution of this paper is to propose a methodology to ensure safety at the control level. The control problem resolution it out of the scope of this paper: it is assumed that once the control problem is feasible, a control law algorithm such as the one proposed in Rubrecht et al. (2010a,b), solves it appropriately (Con-

straints Compliant Control law). The present work focuses on the formulation of the control problem.

The proposed methodology is applied to the case of a multibody robot in a static environment (extended criterion 1). Sect. 2 exposes the description retained for the robotic system and its constraints and proposes a definition of safety at the control level. Sect. 3 is dedicated to the methodology description, whereas Sect. 4 details case studies dealing with static obstacles, joint positions, velocities and accelerations constraints. Finally, a set of experiments illustrates the approach in Sect. 5.

2 Description for safety

In this section, an appropriate description of the robot and its constraints is introduced and a resulting definition of safety is proposed. In this work the control problem is formulated at the velocity kinematic level. The assumptions of this study are an exact perception of the system and the environment, an exact knowledge of the model and the system real capabilities and an exact execution when the desired joint input satisfies the constraints: $u(k) = \dot{q}_{des}(k+1)$ for time step k is exactly carried out at the next time step ($\dot{q}(k+1) = \dot{q}_{des}(k+1)$).

2.1 E-state

First, it appears that the description of the behavior of a robotic system Σ and its constraints through its state s as defined in the State Representation formalism is insufficient. As a matter of fact, an extended state vector (e-state) is defined and denoted σ; it gathers all the variables which allow to describe Σ and its constraints. The e-state is defined over continuous time ($t \in \mathbb{R}_+$) since it contains variables used to describe the physical system. For example, a n-DOF manipulator controlled at the velocity kinematic level and constrained by collisions avoidance and joint position, velocity and acceleration limits has the following e-state

$$\sigma = \begin{bmatrix} q^T & \dot{q}^T & \ddot{q}^T & d^T \end{bmatrix}^T \quad (4)$$

where d^T is a vector of distances to obstacles. In the same example, the state of Σ would be $s = q$. Conversely to σ, the control vector $u(k) = \dot{q}(k+1)$ belongs to $\mathbb{R}^n$ and it is defined over the discrete time ($k \in \mathbb{N}$). The e-state space is denoted $\mathscr{S}$ and the control space ($\mathbb{R}^n$) is denoted $\mathscr{C}$.

2.2 E-state constraints

The notion of constraint usually refers to both a test on the system ("Is the joint boundary exceeded?"- denoted by *e-state constraint*) and a prerequisite to motion ("The control input sent to the actuator should not lead to exceed the joint boundary."- denoted by *control constraint*). The e-state constraints describe if Σ satisfies safety at the current time, *i.e.* when not considering any time horizon. They can be expressed through Boolean functions such as

$$f : \begin{array}{c} \mathscr{S} \to \mathscr{B} \\ \boldsymbol{\sigma} \mapsto 1 \ \text{if the constraint is satisfied} \\ 0 \ \text{else,} \end{array} \qquad (5)$$

where $\mathscr{S}$ is the e-state space and $\mathscr{B}$ the Boolean space.

As an example of e-state constraint, $f_{PM,3}$ (P for Position limit, M for Maximum) describes the superior position limit of the 3rd joint

$$f_{PM,3} : \ \boldsymbol{\sigma}(t) \ \mapsto q_3(t) - q_{M,3} \leq 0 \qquad (6)$$

where $q_3(t)$ is the joint position of joint 3 at time t and $q_{M,3}$ is the maximum joint boundary value.

Any e-state satisfying the p e-state constraints imposed to Σ satisfies the property $\bigwedge_{i=1}^{p} (f_i(\boldsymbol{\sigma})) = 1$, where $\bigwedge$ is the logical conjunction operator (AND). This means that all e-state constraints are simultaneously true for the e-state $\boldsymbol{\sigma}$. In this case, $\boldsymbol{\sigma}$ is called an *instant-safe e-state*.

2.3 Subspaces of the e-state space and definition of safety

The e-state space $\mathscr{S}$ is composed of subspaces that can be identified. The subspace of $\mathscr{S}$ gathering all the instant-safe e-states is denoted $\mathscr{S}_A$. Conversely, the complementary subspace gathers the e-states violating an e-state constraint; it is denoted $\mathscr{S}_{VS}$ (VS for Violation e-State). As illustrated by Fig. 2, maintaining at each time step $\boldsymbol{\sigma}$ in $\mathscr{S}_A$ for the next time step is not sufficient to prevent an inevitable e-state constraint violation in the future. As a consequence, a part of $\mathscr{S}_A$ must never be reached to guarantee safety.

An e-state leading inevitably to an e-state constraint violation is called an Inevitable Violation e-State (IVS). It is an extension of the notion of Inevitable Collision State (ICS) defined by Fraichard *et al.* (2004) which denotes a state from which, whatever the sequence of control inputs sent, a collision finally occurs. Once an IVS is reached, the system can be considered as not safe anymore as an e-state constraint violation is going to happen. The space of IVS is a subspace of $\mathscr{S}_A$ denoted
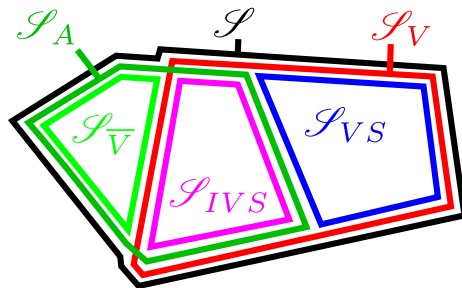

Fig. 3 Partitioning of the e-state space. To be safe, a system should not be able to reach $\mathscr{S}_V$

$\mathscr{S}_{IVS}$. The union of $\mathscr{S}_{IVS}$ and $\mathscr{S}_{VS}$ is denoted $\mathscr{S}_V$ and gathers all the e-states that should be avoided to ensure safety. The complement of $\mathscr{S}_V$ in $\mathscr{S}$ is denoted $\mathscr{S}_{\overline{V}}$. These subspaces are illustrated on Fig. 3. A definition of safety is then

Definition 1 : *The **safety** of a robotic system Σ is ensured at the control level if its e-state $\boldsymbol{\sigma}$ cannot reach $\mathscr{S}_V$.*

This definition enlightens the role of the constraints expression to limit the evolution of the system toward dangerous areas.

The last subspace to define in this section regards the space that can be reached by a system. Given an initial e-state $\boldsymbol{\sigma}_0$, $\mathscr{R}(\boldsymbol{\sigma}_0)$ denotes the space of all the reachable e-states on an infinite time horizon through all the possible constraint compliant control.

3 Methodology to ensure safety

This section exposes the methodology to ensure safety. The proposed methodology must be carried out offline, upstream from the robotic mission. The equivalent of e-state constraints should be formulated at the control level. Once their *validity* is ensured, they should either be proved *compatible*, or the permanent availability of an *alternative safe behavior* must be ensured on an infinite time horizon.

3.1 Step 1: Control constraints definition

The controller cannot act directly on the e-state $\boldsymbol{\sigma}$; it modifies it indirectly through the control vector $\boldsymbol{u}$. Inversely, at each time step, by imposing conditions on the e-state, each e-state constraint forbids an area of the control vector space $\mathscr{C}$. Hence, to each e-state constraint f is associated a control constraint F which can be defined as the function returning the space of *admissible* control vectors $\mathscr{C}_A(\boldsymbol{\sigma})$, *i.e.* the control vectors leading to an instant-safe e-state at the next time

step. A control constraint can be expressed as a Boolean function returning whether a given control vector belongs to $\mathscr{C}_A(\boldsymbol{\sigma})$ or not

$$F : \begin{array}{l} (\mathscr{S}, \mathscr{C}) \mapsto \mathscr{B} \\ (\boldsymbol{\sigma}, \boldsymbol{u}) \mapsto 1 \ \text{if } \boldsymbol{u} \in \mathscr{C}_A(\boldsymbol{\sigma}) \\ \quad\quad 0 \ \text{else.} \end{array} \tag{7}$$

The control input being discrete, control constraints are defined over discrete time ($k \in \mathbb{N}$). As an extension of the notation $\boldsymbol{\sigma}(t)$ ($t \in \mathbb{R}_+$), $\boldsymbol{\sigma}(k)$ ($k \in \mathbb{N}$) denotes the e-state at time step k.

In the example of the 3rd joint superior position limit, if the control is done at the velocity kinematic level, a possible control constraint is

$$F_{PM,3} : \ (\boldsymbol{\sigma}(k), \dot{\boldsymbol{q}}(k+1)) \ \mapsto \tag{8}$$
$$\dot{q}_3(k+1) - \frac{q_{M,3} - q_3(k)}{\delta t} \le 0.$$

It can be mentioned that from a practical point of view, the inequalities imposed on the system at each time step in the QP control law structure are an example of control constraints. At a given time step k, these terms are gathered in

$$J_C(\boldsymbol{q}(k))\dot{\boldsymbol{q}}(k+1) - \boldsymbol{b}(k) \le 0. \tag{9}$$

3.2 Step 2: Validity

In order to ensure safety, the first stage is to check that control constraints are *valid*.

Definition 2 *: Validity. Let $\boldsymbol{\sigma}$ be an instant-safe e-state at time step k, a control constraint F is said valid if its satisfaction implies the satisfaction of its associated e-state constraint f at next time step $k+1$ and for all time between k and $k+1$.*

$$\boldsymbol{\sigma} \in \mathscr{S}_A, k \in \mathbb{N}, \forall t \in [k\delta t; (k+1)\delta t] :$$
$$F(\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) = 1 \Rightarrow f(\boldsymbol{\sigma}(t)) = 1.$$

The validity of constraints is most of the time an assumption rather than a formally proved property. For example, constraints at various physical levels (position, velocity, acceleration, etc.) must be converted to the control physical level, which is often done thanks to first order approximations (finite differences). The control being in discrete time, the approximations induced by finite differences generate errors between the discrete ideal behavior and the real one. However, it is assumed that the sampling period is appropriately chosen to ensure that these errors remain acceptable with respect to the various usual sources of errors (model

approximations, sensors precision, etc.). As a remark, it is always possible to find valid control constraints by reducing their space of admissible control vector $\mathscr{C}_A$.

3.3 Step 3: Compatibility

A second stage to ensure safety is to check that the set of control constraints is *compatible*.

Definition 3 *: Compatibility. Given an initial e-state $\boldsymbol{\sigma}_0$ in $\mathscr{S}_{\overline{V}}$, a set of p control constraints is compatible if for all e-state $\boldsymbol{\sigma}$ in $\mathscr{R}(\boldsymbol{\sigma}_0)$, there exists $\boldsymbol{u}$ in $\mathscr{C}$ such that $\bigwedge_{i=1}^p (F(\boldsymbol{\sigma}, \boldsymbol{u})) = 1$.*

The following proposition establishes that validity and compatibility ensure safety.

Proposition 1 *: Let $\boldsymbol{\sigma}_0$ in $\mathscr{S}_{\overline{V}}$ be the e-state of Σ, a robotic system constrained by p e-state constraints. If the p control constraints of Σ are valid and compatible, then safety is ensured.*

Proof : Let Σ be a robotic system in an initial (time step 0) e-state $\boldsymbol{\sigma}_0$ belonging to $\mathscr{S}_{\overline{V}}$. As the control constraints are compatible, there exists $\boldsymbol{u}$ in $\mathscr{C}$ such that $\bigwedge_{i=1}^p (F(\boldsymbol{\sigma_0}, \boldsymbol{u})) = 1$. Thus the control problem is feasible and as all the constraints are valid, $\boldsymbol{\sigma}$ is an instant-safe e-state at time step 1 and for all time between time steps 0 and 1. This reasoning can be extended by recursion for all time steps. As a consequence, $\boldsymbol{\sigma}$ is maintained in $\mathscr{S}_A$ on an infinite time-horizon, which means that it is maintained in $\mathscr{S}_{\overline{V}}$; as a consequence, it cannot reach $\mathscr{S}_V$ and safety is ensured.

3.4 Step 4: Design of Alternative Safe Behaviors

The study of compatibility between control constraints is complex: an exhaustive method would consist to, given an initial e-state $\boldsymbol{\sigma}_0$, evaluate all the control constraints for all the e-states $\boldsymbol{\sigma}$ reachable from $\boldsymbol{\sigma}_0$ to detect empty intersections between control admissibility spaces $\mathscr{C}_A(\boldsymbol{\sigma})$. Given the diversity of constraints, it seems vain to look for generic methods to detect incompatibilities and modify control constraints appropriately to eradicate them. Moreover, sometimes incompatibilities cannot be resolved: when variables cannot be measured accurately, or when there is no model available, another method should be used to ensure safety.

A second way to guarantee safety is to ensure the permanent availability of a sequence of control solutions leading to instant-safe e-states on an infinite time horizon. At each time step, it is ensured that the controller will be able at next time step to switch to an infinite

sequence of controls leading to exclusively instant-safe
e-states. Similarly to the proof of proposition 1, $\boldsymbol{\sigma}$ is
maintained in $\mathscr{S}_A$ on an infinite time-horizon, which
means that it is maintained in $\mathscr{S}_{\overline{V}}$; as a consequence,
it cannot reach $\mathscr{S}_V$ and safety is ensured. This control
sequence is called an Alternative Safe Behavior (ASB -
referred as evasive manoeuvres by Parthasarathi *et al.*
(2007)). Dedicated ASBs are exposed in Sect. 4.5 ac-
cording to the specifications of the proposed case stud-
ies.

3.5 Summary and methodology

To describe the physical system Σ and its constraints in
continuous time, the e-state $\boldsymbol{\sigma}$ is proposed, and the sta-
tus of the system with respect to its constraints is given
by the e-state constraints. Based on this description, a
definition of safety at the control level is proposed: Σ
is safe if its e-state $\boldsymbol{\sigma}$ is not able to reach the forbidden
e-states $\mathscr{S}_V$. In order to prevent this, the control con-
straints F are defined, and the validity property keeps
a link between control constraints in discrete time and
e-state constraints in continuous time. To ensure safety,
either the compatibility property must be proved for the
set of control constraints, or the availability of an Al-
ternative Safe Behavior must be ensured on an infinite
time horizon.

The algorithm presented on Fig. 4 illustrates the fol-
lowing methodology:

1. Based on e-state constraints, formulate associated
   control constraints;
2. Prove validity;
3. Prove compatibility;
4. If step 3 is not possible, either go back to 1 and
   modify the control constraints expression or define
   permanently available ASBs. These ASB will then
   be computed at each time step to ensure that the
   controller is able to switch at the following time step
   to an infinite sequence of controls leading to exclu-
   sively instant-safe e-states.

4 Case studies

To illustrate the approach described previously, three
case studies are proposed, based on combinations of the
following constraints: {joint position limits - joint veloc-
ity limits - joint acceleration limits - collisions avoid-
ance} applied to a n-DOFs serial manipulator Σ con-
trolled at the velocity kinematic level. The first case
study considers compatible constraints with intuitive
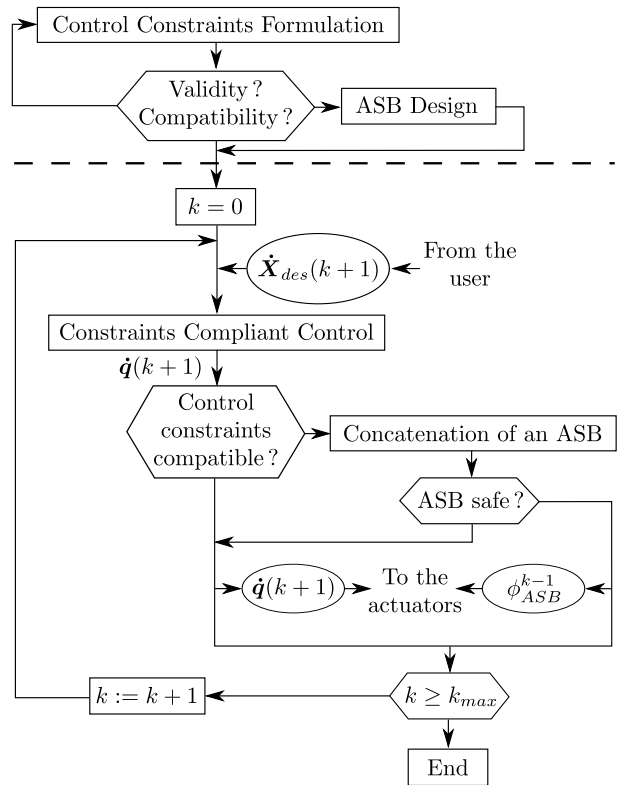expressions. The second case study involves constraints

Fig. 4 Safe controller algorithm. The part of the algo-
rithm above the dashed line is offline and is concerned with
the control problem formulation; the part of the algorithm
under the dashed line is online and is concerned with the
control problem resolution. From each identified e-state con-
straint (physical limit or induced by the mission), a control
constraint is formulated offline. If the validity of the con-
straints cannot be proved, a new formulation of the control
constraints must be expressed. It is always possible to find
valid control constraints by reducing their space of admis-
sible control vector $\mathscr{C}_A$. If the compatibility of the control
constraints cannot be proved, a new formulation can be ex-
pressed and evaluated, or an ASB must be established. Once
this is done, the reactive control loop is launched. At each
time step, the controller is fed with operational inputs and
solves the control problem thanks to any Constraints Com-
pliant Control algorithm. In particular, it can include usual
constraints avoidance techniques (*e.g.* the one of Maciejewski
et al (1985)). If the compatibility of the control constraints
defined offline could not be proved, an ASB sequence is con-
catenated to the desired joint motion: if the resulting behav-
ior is not safe, then the first control input of the ASB is sent
(represented by ϕ_{ASB}^{k-1}), which safety has been proved at a
previous time step; else, the control solution is sent to the
actuators

that must be modified to be proved compatible. The
third case study deals with constraints that cannot be
proved to be always compatible; as a result, the perma-
nent availability of an ASB is required.

4.1 E-state constraints expression

This first section exposes the e-state constraints expressions. These expressions describe if, for a given time t in $\mathbb{R}_+$, the e-state $\boldsymbol{\sigma}$ of Σ is instant-safe. The subscripts $_m$ and $_M$ denote respectively the minimum (negative value) and the maximum (positive value) limit of the considered variable.

Joint position limit

$$f_{PM} : \ \boldsymbol{\sigma}(t) \mapsto \boldsymbol{q}(t) \leq \boldsymbol{q_M} \qquad (10)$$
$$f_{Pm} : \ \boldsymbol{\sigma}(t) \mapsto \boldsymbol{q_m} \leq \boldsymbol{q}(t) \qquad (11)$$

Joint velocity limit

$$f_{VM} : \ \boldsymbol{\sigma}(t) \mapsto \dot{\boldsymbol{q}}(t) \leq \dot{\boldsymbol{q}}_M \qquad (12)$$
$$f_{Vm} : \ \boldsymbol{\sigma}(t) \mapsto \dot{\boldsymbol{q}}_m \leq \dot{\boldsymbol{q}}(t) \qquad (13)$$

Joint acceleration limit

$$f_{AM} : \ \boldsymbol{\sigma}(t) \mapsto \ddot{\boldsymbol{q}}(t) \leq \ddot{\boldsymbol{q}}_M \qquad (14)$$
$$f_{Am} : \ \boldsymbol{\sigma}(t) \mapsto \ddot{\boldsymbol{q}}_m \leq \ddot{\boldsymbol{q}}(t) \qquad (15)$$

Collisions avoidance
A collision is characterized by

$$\Sigma \bigcap \Omega \neq \emptyset \qquad (16)$$

where Σ is the system (meant here as the set of all the system points) and Ω is the set of all the obstacles points. The e-state constraint expression is then

$$f_O : \ \boldsymbol{\sigma}(t) \ \mapsto \forall A \in \Sigma, G_A(\boldsymbol{q}(t)) \notin \Omega \qquad (17)$$

where $G_A(\boldsymbol{q})$ is the geometric model of point A belonging to the robot.

4.2 Case study 1: Joint position limits, Joint velocity limits, Collisions avoidance

This case study involves three constraints: joint position limits, joint velocity limits and collisions avoidance. The following control constraints (assumed to be valid, see Sect. 3.2) are derived from (10) - (15) thanks to finite differences. They are given for a time step k in $\mathbb{N}$.

Joint position limit ($\mathbf{i}^{th}$ joint) F_P

$$F_{PM,i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (18)$$
$$J_{c_i^+} \dot{\boldsymbol{q}}(k+1) \leq \frac{q_{M,i} - q_i(k)}{\delta t}$$
$$F_{Pm,i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (19)$$
$$J_{c_i^-} \dot{\boldsymbol{q}}(k+1) \leq \frac{q_i(k) - q_{m,i}}{\delta t}$$

where $J_{c_i^+} = [0, \ldots, 0, 1, 0, \ldots, 0]$ (the ith term being 1) and $J_{c_i^-} = [0, \ldots, 0, -1, 0, \ldots, 0]$.

Joint velocity limit ($\mathbf{i}^{th}$ joint) F_V

$$F_{VM,i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto J_{c_i^+} \dot{\boldsymbol{q}}(k+1) \leq \dot{q}_{M,i} \qquad (20)$$
$$F_{Vm,i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto J_{c_i^-} \dot{\boldsymbol{q}}(k+1) \leq -\dot{q}_{m,i}. \qquad (21)$$

Collisions avoidance F_O

$$F_O : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (22)$$
$$J_{A,B}(\boldsymbol{q}(k)) \dot{\boldsymbol{q}}(k+1) \leq \frac{d_{A,B}(k)}{\delta t}$$

for all pairs of point (A, B), where A belongs to the robot and B to the obstacles; $d_{A,B}$ is the distance between A and B; $J_{A,B}(\boldsymbol{q}(k))$ is the (line) Jacobian of point A along the direction $A \to B$. For practical reasons, this infinite set of constraints is reduced to one constraint per segment of the robot (shortest distance). This assumption is frequently made despite its limits in some cases (as shown by Kanehiro *et al.* (2008)). It is considered sufficient in the present study.

The space of admissible control vectors for control constraints F_P, F_V and F_O are respectively denoted $\mathscr{C}_A^P$, $\mathscr{C}_A^V$ and $\mathscr{C}_A^O$.

Validity being assumed (cf. 3.2), the compatibility is checked.

Proposition 2 *: The set $\{F_P, F_V, F_O\}$ is compatible*

Proof : Let $\dot{\boldsymbol{q}}_0$ be the null control vector ($\dot{\boldsymbol{q}}_0 = \boldsymbol{0}$) and let $\boldsymbol{\sigma}_0$ be in $\mathscr{S}_{\overline{V}}$. For any $\boldsymbol{\sigma} \in \mathscr{R}(\boldsymbol{\sigma}_0)$, $\dot{\boldsymbol{q}}(k+1) = \dot{\boldsymbol{q}}_0$ belongs to $\mathscr{C}_A^P$, $\mathscr{C}_A^V$ and $\mathscr{C}_A^O$. As a result, for all $\boldsymbol{\sigma}(k)$ in $\mathscr{R}(\boldsymbol{\sigma}_0)$, $\dot{\boldsymbol{q}}_0$ is solution of the control problem and thus $F_P(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_0) \bigwedge F_V(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_0) \bigwedge F_O(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_0) = 1$. $\square$

As all the control constraints are proved to be always compatible, safety is ensured without modification or ASB required.

4.3 Case study 2: Joint position limits, Joint velocity limits, Joint acceleration limits

As in the previous case, this case study involves three constraints but collisions avoidance is replaced by joint acceleration limits. The control constraints for joint position and velocity limits are taken from case study 1 (18) - (21); the control constraint of joint acceleration limit is derived from (14) and (15) thanks to finite differences (i^{th} joint)

$$F_{AM,i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (23)$$
$$J_{c_i^+} \dot{\boldsymbol{q}}(k+1) \leq \ddot{q}_{M,i} \delta t + \dot{q}_i(k)$$
$$F_{Am,i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (24)$$
$$J_{c_i^-} \dot{\boldsymbol{q}}(k+1) \leq -\ddot{q}_{m,i} \delta t - \dot{q}_i(k).$$

The space of admissible control vectors for control constraint F_A is denoted $\mathscr{C}_A^A$.

Proposition 3 *: The sets of control constraints generated by* $\{F_P, F_A\}$ *are incompatible.*

Proof : Let $\boldsymbol{\sigma}_0$ *be in* $\mathscr{S}_{\overline{V}}$. *From* $\boldsymbol{\sigma}_0$, *any* $\boldsymbol{\sigma}_V \in \mathscr{R}(\boldsymbol{\sigma}_0)$ *for which a given joint satisfies*

$$\dot{q}(k) > \frac{q_M - q(k)}{\delta t} - \ddot{q}_m \delta t \qquad (25)$$

is such that $F_{PM,i}(\boldsymbol{\sigma}_V)$ *and* $F_{Am,i}(\boldsymbol{\sigma}_V)$ *are not compatible, which traduces that* $f_{PM,i}$ *and* $f_{Am,i}$ *cannot be satisfied simultaneously. As there is no assumption or constraint preventing from reaching* $\boldsymbol{\sigma}_V$, *then* F_P *and* F_A *are incompatible.* □

This incompatibility is illustrated on Fig. 2. It has been locally treated by Decré *et al.* (2009), but as shown in Rubrecht *et al.* (2010b), the proposed method is tight and can be smoothened by imposing that the joint distance to the joint position limit at next time step should remain superior to the current joint distance needed to decelerate. As a result, a modified expression of F_P is proposed (i^{th} joint)

$$F_{PM',i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (26)$$
$$J_{c_i^+} \dot{q}(k+1) \leq \frac{(q_M - q(k)) - \frac{1}{2}(s_1^2 - s_1)\ddot{q}_m \delta t^2}{(s_1 + 1)\delta t}$$
$$F_{Pm',i} : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \qquad (27)$$
$$J_{c_i^-} \dot{q}(k+1) \leq \frac{(q_m - q(k)) - \frac{1}{2}(s_2^2 - s_2)\ddot{q}_M \delta t^2}{(s_2 + 1)\delta t}$$

with

$$s_1 = -\frac{\sqrt{-2\ddot{q}_m(q_M - q(k))}}{\ddot{q}_m \delta t}, \qquad (28)$$

$$s_2 = \frac{\sqrt{-2\ddot{q}_M(q_m - q(k))}}{\ddot{q}_M \delta t}. \qquad (29)$$

The choice of (26) and (27) as the joint position limits constraints provoke a small reduction of the reachable positions. Actually, the resolution of $\dot{q}(k+1) = 0$ in (26) induces

$$q_M - q(k) = \frac{-\ddot{q}_m \delta t^2}{8} \qquad (30)$$

which means that the asymptotic value of the joint position according to this constraint is no longer q_M but $q_{M'} = q_M - \frac{-\ddot{q}_m \delta t^2}{8}$. The order of magnitude of this reduction is $\sim \delta t^2$, which can be considered negligible. However, it is a reduction of the space of reachable e-state, and all the compatibility studies involving this control constraint must be checked over the joint position space $\mathscr{S}_A^{P'} = [\boldsymbol{q_{m'}}; \boldsymbol{q_{M'}}]$ where $\boldsymbol{q_{m'}}$ and $\boldsymbol{q_{M'}}$ are the vectors of general term respectively $q_m + \frac{\ddot{q}_M \delta t^2}{8}$ and $q_M - \frac{-\ddot{q}_m \delta t^2}{8}$.

Proposition 4 *The set* $\{F_{P'}, F_V, F_A\}$ *is compatible.*

Proof Let $\boldsymbol{\sigma}_0$ be in $\mathscr{S}_{\overline{V}}$, the current time step k be in $\mathbb{N}$ and the current e-state $\boldsymbol{\sigma}(k)$ be in $\mathscr{R}(\boldsymbol{\sigma}_0)$. The design of (26) and (27) is based on the condition

$$\boldsymbol{\Delta q}(k+1) > \boldsymbol{d}_{\mathbb{R},dec}(k) \qquad (31)$$

where $\boldsymbol{\Delta q}(k+1)$ is the joint distance to the position limit at the next time step and $\boldsymbol{d}_{\mathbb{R},dec}(k)$ is a vector of upper bounds of the joint distances needed to stop at current time step (cf. Rubrecht et al. (2010b)). This condition implies that the vector of maximum deceleration velocity $\dot{\boldsymbol{q}}_{dec}(k+1)$ which general term is

$$\dot{q}_{dec}(k+1) = \begin{cases} \dot{q}(k) + \ddot{q}_m \delta t & if \ \dot{q}(k) \geq -\ddot{q}_m \delta t \\ \dot{q}(k) + \ddot{q}_M \delta t & if \ \dot{q}(k) \leq -\ddot{q}_M \delta t \ (32) \\ 0 & else \end{cases}$$

belongs to $\mathscr{C}_A^{P'}(\boldsymbol{\sigma})$. Then, by definition, it belongs to $\mathscr{C}_A^A(\boldsymbol{\sigma})$. Finally, as it reduces the velocity magnitude, it belongs to $\mathscr{C}_A^V(\boldsymbol{\sigma})$. As a result, for all $\boldsymbol{\sigma}(k)$ in $\mathscr{R}(\boldsymbol{\sigma}_0)$, $\dot{\boldsymbol{q}}_{dec}$ is solution of the control problem and thus $F_{P'}(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_0) \bigwedge F_V(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_0) \bigwedge F_A(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_0) = 1$. □

This incompatibility between control constraints being resolved, the control constraints are ensured to be always compatible, which ensures safety.

4.4 Case study 3: Joint position limits, Joint velocity limits, Joint acceleration limits, Collisions avoidance

This case study involves four constraints, gathering the two previous case studies: joint position limits, joint velocity limits, joint acceleration limits and collisions avoidance. The considered control constraints are (20) - (24), (26) and (27).

Proposition 5 *: The sets generated by* $\{F_O, F_A\}$ *are incompatible.*

Proof : Let $\boldsymbol{\sigma}_0$ *be in* $\mathscr{S}_{\overline{V}}$. *From* $\boldsymbol{\sigma}_0$, *any* $\boldsymbol{\sigma}_V \in \mathscr{R}(\boldsymbol{\sigma}_0)$ *for which*

$$J_{A,B}(\boldsymbol{q}(k))\dot{\boldsymbol{q}}(k) > \frac{d_{A,B}(k)}{\delta t} - J_{A,B}(\boldsymbol{q}(k))\ddot{\boldsymbol{q}}_m \delta t \ (33)$$

shows that $F_O(\boldsymbol{\sigma})$ *and* $F_{Am}(\boldsymbol{\sigma})$ *are not compatible, which shows that* $f_O(\boldsymbol{\sigma})$ *and* $f_{Am}(\boldsymbol{\sigma})$ *cannot be satisfied simultaneously. As there is no assumption or constraint preventing from reaching* $\boldsymbol{\sigma}_V$, *then* F_O *and* F_A *are incompatible.* □

As mentioned in Rubrecht *et al.* (2010b), the incompatibility induced by the simultaneous presence of F_O and F_A is complex. Actually, the operational acceleration depends on the robot configuration (derived from (22)).

$$\ddot{X}_{A,B} = J_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad (34)$$

which does not enable to rely on any value for the operational acceleration capabilities along a trajectory. In the worst cases, these capabilities may fall down to zero, which prevents to take a lower bound on which to rely for the deceleration capabilities estimation. As a result, ensuring compatibility between joint acceleration limits and collisions avoidance seems impossible without an exploration in the neighborhood of the current system e-state, which may turn time-consuming and thus not acceptable in real-time reactive control. In this case, the permanent availability of an Alternative Safe Behavior is required.

4.5 Alternative Safe Behavior

When the control compatibility cannot be proved, the permanent availability of an Alternative Safe Behavior is required, to be triggered in case of critical situation. As mentioned in Sect. 3.4, an ASB is a sequence of control solutions $\boldsymbol{u}$ leading to instant-safe e-states on an infinite time horizon. It must be computed at each time step and the safety of the resulting e-states must be checked, thus it should be fast to compute. To clarify the following descriptions, let ϕ denotes an infinite constraint compliant control input, *i.e.* an infinite sequence of controls $\boldsymbol{u}$ satisfying the control constraints at each time step.

4.5.1 Algorithm based on maximum joint deceleration ASB

As a preliminary observation, as the environment is assumed to be static, once an instant-safe e-state is static (no variation with respect to time), it remains safe until the end of time. Consequently, the first ASB proposed ϕ_{ASB1} is a full deceleration at the joint level. This deceleration is the most efficient way to stop the robot: it is fast as no Jacobian has to be recomputed at each time step, and the number of time steps necessary to obtain a static robot is minimized. As the control constraints of joint position, velocity and acceleration limits are compatible, the only remaining constraint to check on all e-states resulting from ϕ_{ASB1} is F_O, that is an intersection between the robot bodies and the environment. The method is detailed on algorithm 1. It is assumed that for $k = 0$, the initial e-state $\boldsymbol{\sigma}_0$ belongs to $\mathscr{S}_{\overline{V}}$. The algorithm is illustrated on Fig. 5.

4.5.2 Algorithm based on mixable joint deceleration ASB

As the robot may oscillate between two behaviors (between $\dot{\boldsymbol{q}}(k + 1)$ issued from the task, and ϕ_{ASB1}^{k-1} is-

---

Algorithm 1 : Maximum joint deceleration ASB

> **for all** $k \in \mathbb{N}$ **do**
>> (1) Compute an admissible solution $\dot{\boldsymbol{q}}(k + 1)$
>> (2) Compute deceleration trajectory ϕ_{ASB1}^k
>> **if** (3) for all $\boldsymbol{\sigma}$ resulting from ϕ_{ASB1}^k, $\boldsymbol{\sigma}$ is an instant-safe e-state **then**
>>> (4.1) Send $\dot{\boldsymbol{q}}(k + 1)$
>> **else**
>>> (4.2) Send the first element of ϕ_{ASB1}^{k-1}
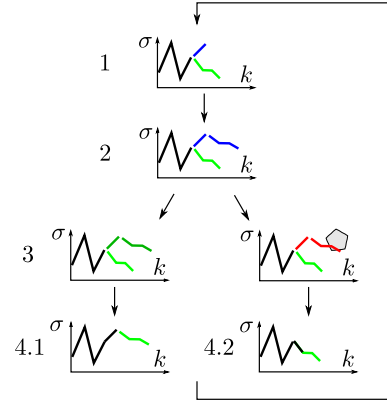>> **end if**
> **end for**

Fig. 5 Algorithm of maximum deceleration based ASB. *Blue* is for non validated motion, *red* is for non admissible motion and *green* for safe motion. 1/ Control solution computation $\dot{\boldsymbol{q}}(k+1)$; 2/ ASB1 profile computation ϕ_{ASB1}^k; 3/ Admissibility check; 4.1 (up) and 4.2 (down)/ Send appropriate output

sued from the ASB) a safe but rough behavior is expected from algorithm 1 when moving near obstacles. The problem lies in the *maximal* deceleration toward the static e-state; when ϕ_{ASB1} is chosen at one time step, it is likely to be retained until the robot stops. As shown on a simple example in Fig. 6, in most cases there is no available space during ϕ_{ASB1} for another motion than full deceleration. To have a small margin in the intersection on the control admissibility space, it is proposed to add a prediction ϕ_{ASB2} with reduced accelerations capabilities. At each time step, both predictions are tested (ϕ_{ASB1} based on $\ddot{\boldsymbol{q}}_{min/max}$; ϕ_{ASB2} based on $\alpha\ddot{\boldsymbol{q}}_{min/max}$, with $\alpha \lesssim 1$). Once one of these behavior leads the robot in intersection with the environment, the robot adopts a ϕ_{ASB1}^{k-1} (maximal deceleration). For both behaviors, in case of violation, $\ddot{\boldsymbol{q}}_{min/max}$ is applied. If ϕ_{ASB2} is violated (in most cases), the control will then have a margin in the control admissibility space during its deceleration, where another behavior can be inserted.

The method proposed by Faverjon and Tournassoud (1987) enables to illustrate this approach. This method is referred as the Smooth Avoidance Technique (SAT). Briefly, this method limits the operational velocity of each point of the robot bodies that gets close to an
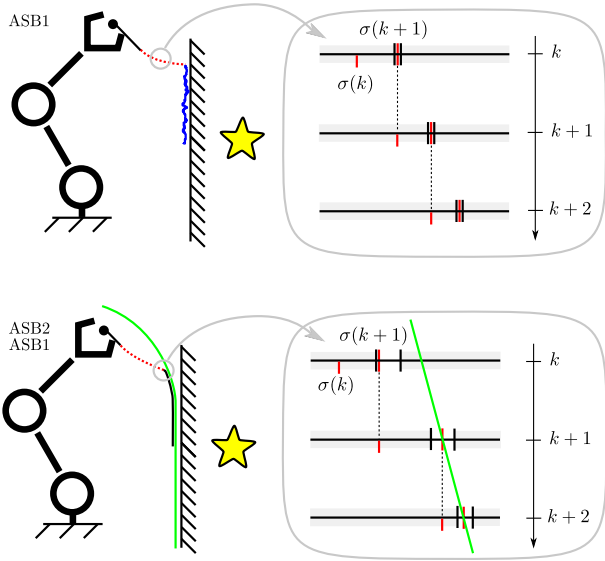
Fig. 6 Comparative behaviors of robots trying to reach a keypoint (star) behind a wall. On the right, the schemes are representations of the e-states projected on the joint space of the 2nd DOF of the system during 3 time steps. **Top**: maximum joint deceleration (ASB1). The motion of the robot is decomposed in three parts. *Black path*: the motion is computed through the control law, and at each time step the controller concatenates the control vector to be sent with a full deceleration, to check if a collision occurs and decide if the control vector should be sent or not. *Red dashed path*: a collision with the predicted full deceleration being detected, it is applied before sending the control law computed input; during the ASB deceleration, only the full deceleration control solution is admissible (top right). However, when the robot stops, it is close to the obstacle. *Blue path*: once near the obstacles, the controller oscillates between the control law solutions and ASB. **Bottom**: mixable joint deceleration (ASB1 & ASB2). As in the scheme at the top, the motion of the robot is decomposed in three parts. *Black path*: control law based motion; it is shorter than the upper one, because deceleration predictions are based on under-estimated capabilities. *Red dashed path*: the ASB is done with maximal deceleration capabilities, but as it has been triggered before, the control law solutions can be chosen in a small (but not reduced to a point) interval (bottom right). *Black path on the green curve*: as a result, the robot progression toward the wall can be damped by a smooth path constraint. *Green curve*: representation of the smooth path trajectory

obstacle. The velocity limitation is done through an inequality constraint in the QP framework ((2) and (3)). This limitation involves 2 parameters

$$\dot{d} = -a \frac{d - d_s}{d_i - d_s} \quad \text{for } d \leq d_i \tag{35}$$

where $\dot{d}$ is the temporal derivative of the distance d between the robot point and the obstacle, a is a positive coefficient for adjusting convergence speed, d_s is the security distance and d_i is the distance influence, *i.e.* the distance under which the constraint is activated.

The control constraint associated to (35) is

$$F_C : (\boldsymbol{\sigma}(k), \boldsymbol{u}(k)) \mapsto \tag{36}$$
$$J_A \dot{\boldsymbol{q}}(k) \leq -a \frac{d_{A,B}(k) - d_s}{d_i - d_s} \quad \text{for } d_{A,B}(k) \leq d_i.$$

Including the expression of this control constraint at a given time step does not necessarily yield a feasible control problem. However, it is acceptable to violate it as it does not involve security but rather a desired behavior. Checking the compatibility at each time step is not trivial: knowing if a set of linear constraint is compatible may require the resolution of the associated linear system. An approximate answer is given by checking whether the configuration of maximum deceleration is admissible. It is not a requirement for compatibility (there may be cases for which this configuration is not admissible whereas the constraints are compatible) but it is a sufficient condition. As a result, at each time step the compatibility between the SAT control constraint and the other constraints is checked: if the SAT is not compatible, it is not considered.

The final method is detailed on algorithm 2. As for algorithm 1, it is assumed that for $k = 0$, the initial e-state $\boldsymbol{\sigma}_0$ belongs to $\mathscr{S}_{\overline{V}}$.

Algorithm 2 : Mixable joint deceleration ASB

for all $k \in \mathbb{N}$ **do**
    Control constraints: $\{F_P, F_V, F_A, F_O, F_C\}$
    if $\bigwedge_{i=1}^{p} (F(\boldsymbol{\sigma}, \dot{\boldsymbol{q}}_{dec}(k+1))) \neq 1$ **then**
        Control constraints: $\{F_P, F_V, F_A, F_O\}$
    end if
    (1) Compute an admissible solution $\dot{\boldsymbol{q}}(k+1)$
    (2) Compute deceleration trajectories $\phi_{ASB1}^k, \phi_{ASB2}^k$
    if (3) for all $\boldsymbol{\sigma}_1$ resulting from ϕ_{ASB1}^k, $\boldsymbol{\sigma}_1$ is a safe e-state AND for all $\boldsymbol{\sigma}_2$ resulting from ϕ_{ASB2}^k, $\boldsymbol{\sigma}_2$ is a safe e-state **then**
        (4.1) Send $\dot{\boldsymbol{q}}(k+1)$
    else
        (4.2) Send the first element of ϕ_{ASB1}^{k-1}
    end if
end for

The algorithm is illustrated on Fig. 7.

5 Results

The following part details the results obtained with a 6-DOF manipulator. The results are composed of 3 experiments showing:

- the safe behavior obtained thanks to the resolution of the joint constraints compatibility;
- the safe behavior obtained thanks to the resolution of the joint constraints compatibility and the maximum joint deceleration ASB;
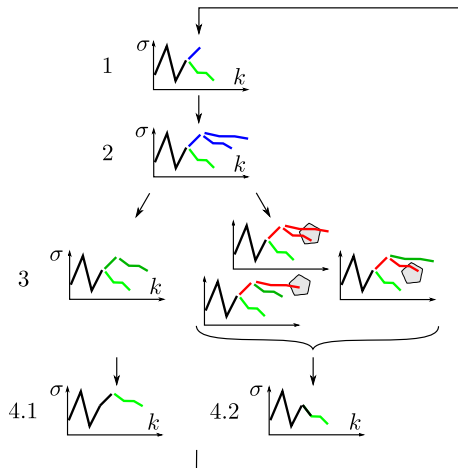
Fig. 7 Algorithm of mixable deceleration based ASB. *Blue* is for non validated motion, *red* is for non admissible motion and *green* for safe motion. 1/ Control solution computation $\dot{q}(k+1)$; 2/ ASB1 (ϕ_{ASB1}^k) and ASB2 (ϕ_{ASB2}^k) profile computation; 3/ Admissibility check; 4.1 (up) and 4.2 (down)/ Send appropriate output

- the safe behavior obtained thanks to the resolution of the joint constraints compatibility and the mixable joint deceleration ASB with the SAT.

5.1 Experiments presentation

The experiments were performed in a facility of the French Alternative Energies and Atomic Energy Commission (CEA), a government-funded technological research organization. The 6-DOF arm used in these experiments is a 100daN advanced remote hydraulic manipulator with force feedback capabilities, the Maestro (David *et al.* 2007), designed by CEA and transferred to Cybernetix[2]. It is usually used in various applications where remote handling with high strength and dexterity are needed, *e.g.* in nuclear or offshore hostile environments.

5.1.1 Experimental equipment

The robot's controller uses a generic hard real-time application, TAO2000 (Gicquel *et al.* 2001), developed by CEA for Computer Aided Teleoperation Systems (teleoperators) and coming from its experience for objects remote manipulation in hazardous environment. It can address both masters and slaves robots, whatever their kinematics and actuation technologies, providing them a whole generic set of useful features with nearly no specific development. This application provides, via a

---

Fig. 8 Teleoperated Maestro operating in front of the tunnel boring machine cutting wheel mockup

standard Ethernet link, a high level communication interface to control the robot and a low level real-time tuning and spying interface.

The Maestro works in front of a tunnel boring machine cutting wheel mock-up (Fig. 8). At each time step, the operational input sent is a desired velocity issued from a 3-DOF desired point (position only, no orientation). It induces a Degree Of Redundancy (DOR) of 3.

5.1.2 Initial assumptions versus experimental conditions

Despite the work carried out on safety, the assumptions enounced in Sect. 2 induce approximations which may provoke minor incompatibilities. These incompatibilities are localized and do not have a big impact on the robot behavior: as shown on the following results, the envelope needed to absorb them could be small with respect to what would be needed without the compatibility study. However, at the control level, an incompatibility provokes the impossibility to solve the problem. For practical reasons, the occurring incompatibilities are *denied* at the control level: for example, if the current position of the joint parameter $q_3(k)$ is inferior to the artificial minimum joint position $q_{m,3}^{art}$, then the inferior joint position limit is taken as the minimum between the current joint position and theoretical minimum joint position: $q_{m,3}$ gets $min(q_3(k), q_{m,3}^{art})$.

From a practical point of view, the envelopes around the joint position limits e_j and around the environment e_c are unknown from the controller and considered as an origin offset: for example, the controller considers that a collision occurs if the distance between the robot and the environment is lower than e_c.

Fig. 9 Views of the robot in initial position (extended) and at $t = 5.0s$ (fold up). The white arrow is the constant operational desired velocity from $t = 0.0\ s$ to $t = 5.0\ s$

5.2 Safe behavior with compatible constraints

This first experiment[3] illustrates the behavior of a multibody robot subject to control constraints modified to become compatible.

Task presentation. The robot is subject to a brutal fold up from a configuration of extended robot to a configuration in which the robot has reached its joint position limits (Fig. 9). During a first period (5.0 s), the desired operational velocities are maintained constant and maximum toward a point at the left infinite; then, the desired operational velocity brings back the robot toward the initial Cartesian point at lower velocity (the aim is to check that the deceleration toward the joint position limit is safe). The considered e-state constraints are joint position, velocity and acceleration limits ((10) - (15)). For the sake of clarity, the e-state constraints limits are the same for each joint: respectively $\pm 1.0\ rad$, $\pm 1.5\ rad/s$, $\pm 2.0\ rad/s^2$. The acceleration limits have been taken voluntarily low (lower than the robot actual capabilities) in order to better illustrate the results. For this particular experiment, the trajectory is considered in a (z,y)-plane (2 DOF desired velocity) and only 3 DOFs are used (see Fig. 9), which brings the DOR to 1.

Control law. The control constraints used to enforce the considered e-state constraints are F_V, F_A and $F_{P'}$ (respectively (20), (21), (23), (24), (26) and (27)). The control problem is expressed as a QP, and the solver is an efficient open source algorithm[4].

Given the limits of the robot, the joint position overshoot could reach 0.56 rad without the proposed methodology (by taking (18) and (19) as the joint position

control constraints for example). As a benefit of our approach, the envelope retained on joint position limits for this experiment is $e_j = 0.1\ rad$.

Results and analysis. The results are presented on Fig. 10. The $\dot{\boldsymbol{q}}_{des}$ values are the joint velocity sent to the actuators ($\dot{\boldsymbol{q}}(k + 1)$) and $\dot{\boldsymbol{q}}_{real}$ are the velocities actually carried out by the actuators. Only the 3 DOFs concerned by the planar trajectory are represented (the other are excluded from the model, so they remain fixed). During the first second, each joint contributes to the operational motion at its best: accelerations are maximal for each joint. Joint 5 is the first to undergo a deceleration (before reaching its maximum velocity) due to the initial proximity to its position limit. Joint 3 reaches its velocity limit for a short time. Joint 2 does not perform high accelerations due to the fact that the operational velocity is sufficiently high thanks to the other joints. At $t = 4.0\ s$, the small motion of joint 2 is induced by a disturbance which, given the system configuration, leads to the tracking of the desired Cartesian velocity. At $t = 5.0\ s$, the operational desired velocity is inverted (the robot goes back to its initial operational position), and the robot gets away from its boundaries without any difficulty. At the end of the experiment, ($t = 7.5\ s$), the deceleration is provoked by a reduction of the operational desired velocity; it is not provoked by any constraint. The envelope violation of joint 3 occurring at the beginning of the experiment ($t = 1.5\ s$) is attributed to the approximations discussed in Sect. 5.1.2 (especially the exact execution of the desired joint input). However, the envelope is slightly violated, which tends to show that it can reliably be reduced (maximum overshoot is $3.03E^{-2}\ rad$).

5.3 Safe behavior with ASB

The second experiment[5] illustrates the behavior of a multibody robot subject to incompatible control constraints; at each time step, the computed control input is sent if a consecutive deceleration toward a static e-state is admissible (see Sect. 4.5).

Task presentation. The robot is subject to various motions in the cluttered environment of the cutting wheel mock-up (Fig. 8). The desired operational velocity is issued from a 3D trajectory involving unreachable points. The considered e-state constraints are joint position, velocity and acceleration limits and collisions avoidance ((10) - (15) and (17)). The trajectory involves motions

---

[3] http://www.isir.upmc.fr/UserFiles/File/...
VpadoiS/Medias/JointPosLim.avi

[4] QuadProg++: http://sourceforge.net/projects/quadprog/

[5] http://www.isir.upmc.fr/UserFiles/File/...
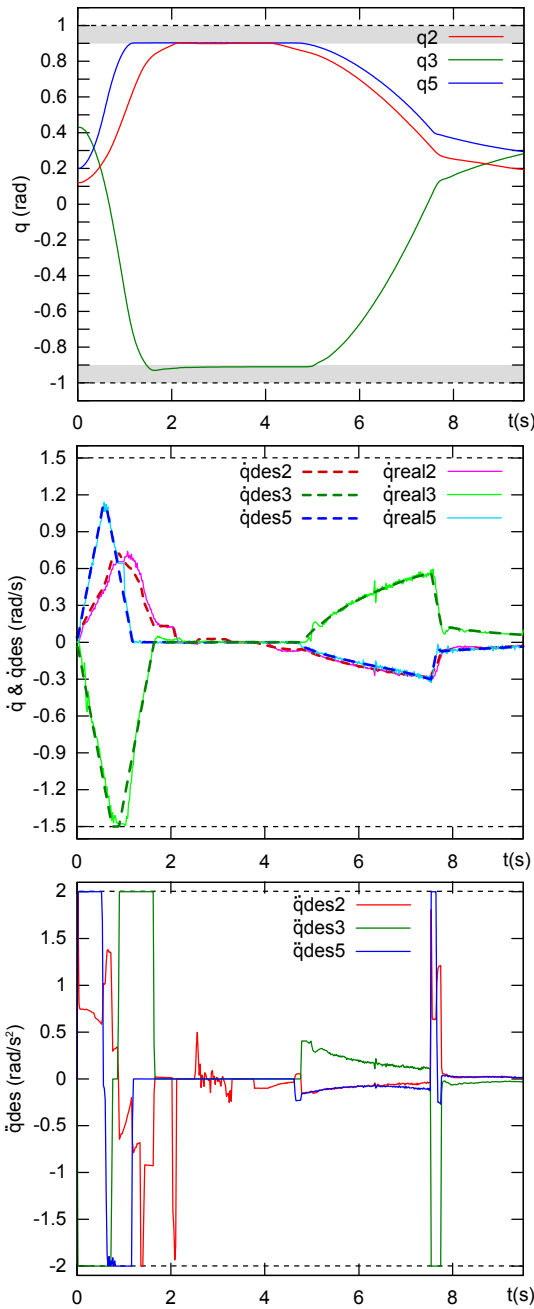VpadoiS/Medias/ObstAvoidASB1.avi

Fig. 10 Position, velocity and acceleration of the joint 2, 3, and 5 during experiment 1. The position is directly measured on the robot, the velocity $\dot{q}_{des}$ is the input sent to actuators and $\dot{q}_{real}$ is the measured one. The acceleration is computed from $\dot{q}_{des}$. All the variables remain between their limits. The control constraints modification imposes appropriate decelerations to satisfy the joint position limits

close to joint position limits. The joint position limits are $\pm 1.0\ rad$. The joint velocity limits are not reached during this experiment; the joint accelerations limits are set to $1.0\ rad/s^2$. The distances are computed in real-time using a CAD model of the environment (Fig. 8).

Control law. The control law is similar to the previous experiment. To deal with incompatible constraints (joint acceleration limits and collisions avoidance), the control uses algorithm 1. To differentiate accelerations due to the trajectory tracking and accelerations issued from ASB1, the acceleration value for prediction and alternative behavior is lower than the one retained for the control constraint: $0.9\ rad/s^2$. This modification has no major impact on the results but makes them clearer.

Given the limits of the robot, the joint position overshoot could reach $0.5\ rad$ without the proposed methodology. In the same conditions, given the dimensions of the robot, the potential collision without ASB would have required an envelope $\sim 1\ m$ to be avoided. As a benefit of our approach, the envelope retained on joint position limits is $e_j = 0.1\ rad$ and the envelope around the environment is $e_c = 0.1\ m$.

Results and analysis. The results are presented on Fig. 11. The 4 motions getting close to obstacles, easily identifiable at $t = 2.0\ s$, $t = 5.0\ s$, $t = 8.0\ s$ and $t = 17.0\ s$ on the graph of distance to environment, end-up in the envelope e_c. The 3 first motions ($t = 2.0\ s$, $t = 5.0\ s$ and $t = 8.0\ s$) gets close to obstacles with a reasonable velocity, but as there is no compatibility between collisions avoidance and acceleration limits, resort to the alternative behavior is needed (red squares). The fourth motion toward obstacles is done at higher speed; the deceleration begins nearly $1.0\ s$ before the impact (blue square). Finally a motion in the neighborhood of the obstacle generates high frequency oscillations on the acceleration ($t = 20.0\ s$). Actually, as the robot remains close to the obstacles, a deceleration at the joint level tends to maintain the robot close to the environment. As a result, oscillations between the trajectory tracking and the alternative behavior occur. Thanks to the envelope e_c taken, safety is preserved.

5.4 Integration of Smooth Avoidance Technique into the mixable joint deceleration ASB

This third experiment[6] illustrates the possibility to introduce usual collisions avoidance methods into a safe framework for a multibody robot. The resulting behavior remains safe and takes full advantage of the avoidance method.

Task presentation. The robot is in charge of reaching a setpoint from which it is separated by an infinite horizontal plane (Fig. 12). As in the first experiment, the
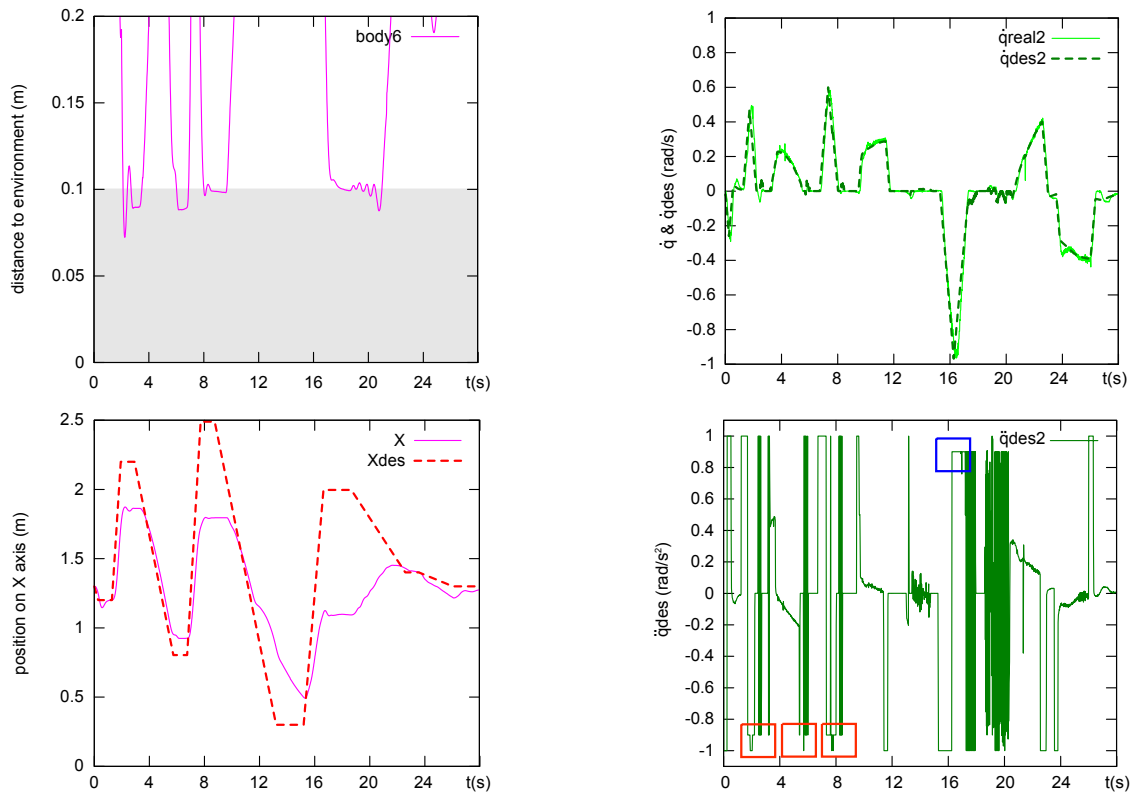
---

[6] http://www.isir.upmc.fr/UserFiles/File/...
VpadoiS/Medias/ObstAvoidASB2SAT.avi

Fig. 11 Results of experiment 2. *Left column*: shortest distance between body 6 and the environment, desired and real position along Cartesian axis X; *right column*: desired and real velocity of joint 2, accelerations of joint 2
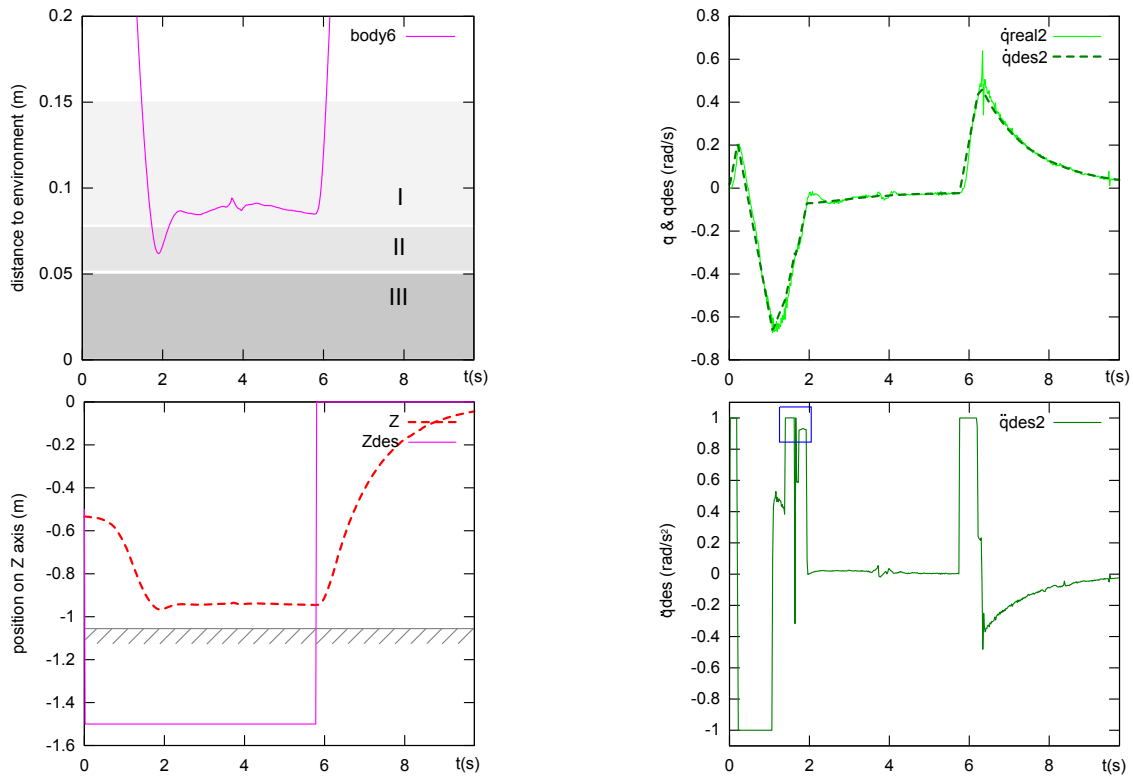

Fig. 13 Results of experiment 3. *Left column*: shortest distance between body 6 and the environment, desired and real position along Cartesian axis Z; *right column*: desired and real velocity of joint 2, accelerations of joint 2
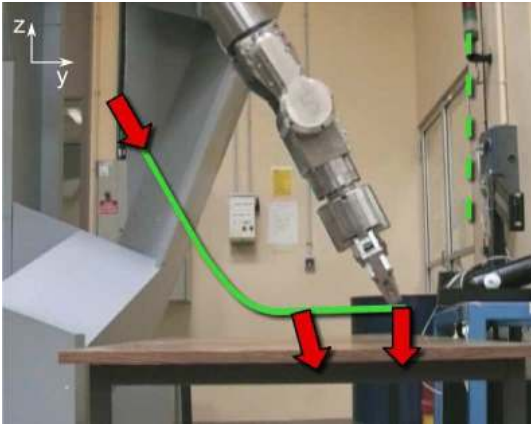
Fig. 12 Views of the robot during the trajectory. The red arrows shows the desired operational velocity input along the robot trajectory (green line)

trajectory is considered in a plane (2 DOF desired velocity) and only 3 DOFs are used (the same as in Fig. 9), which gets the DOR to 1. The magnitude of the desired velocity is maintained constant toward the desired point. At the end of the experiment, a second setpoint is given to get far from the obstacle. The considered e-state constraints are joint position, velocity and acceleration limits and collisions avoidance ((10) - (15) and (17)). The trajectory does not involve motion close to joint position limits. The joint velocity limits are not reached along this motion. The joint accelerations limits are set to $1.0\ rad/s^2$.

Control law. The control law is based on the one of experiment 2. The approach used to preserve safety in the previous experiment has the severe drawback to generate oscillations on the accelerations when the robot is moving along obstacles. Actually, the control alternates between the trajectory tracking and the alternative behavior at nearly each time step. As detailed in Sect. 4.5, the control constraint induced by the SAT (36) is added to the set of considered constraints. The following values have been used: $d_i = 0.15\ m$ (area I), $d_s = 0.07\ m$ (area II) and the envelope around obstacles is $e_c = 0.05\ m$ (area III). Taking different values for d_s and e_c eases the interpretation of the results.

Results and analysis. The results are presented on Fig. 13. As in experiment 2, the arrival on the obstacle causes the maximum overshoot in the area II. The robot never enters the security envelope (area III) as it is managed by the SAT. The distance to obstacle stabilizes during the sliding motion (see Fig. 12) until $t = 6.0\ s$ when another objective is given to the effector. The transition time can be detected on the acceleration

(blue square), when it switches from $1.0\ rad/s^2$ (deceleration coming from the alternative behavior) to approximately $0.93\ rad/s^2$. At that time ($t = 1.62\ s$), the distance to the obstacle is $10.9\ cm$, and the avoidance method begins to limit the robot motion along direction z. The acceleration is then smooth, the collision management being ensured by the SAT. During the motion along the obstacle (between $t = 2.0\ s$ and $t = 6.0\ s$), the velocity of joint 2 contributes to the motion, but the velocity is small as the setpoint is far under the table, increasing the angle between the desired velocity vector and the infinite plane toward orthogonality.

6 Conclusion

The work presented in this paper exposes a methodology to ensure safety of multibody robots behaviors. Satisfying the constraints at each time step may turn out insufficient because of constraints incompatibilities; as a consequence, to obtain a safe behavior, the control problem can be considered as a problem of constraints expression. The proposed approach enables to study the compatibility of constraints and establishes the link between constraints compatibility and safety. It also proposes alternatives if constraints compatibility cannot be established.

Complete case studies illustrate the approach. The constraints expression is modified to ensure compatibility when possible; if not, the permanent possible resort to a safe behavior is ensured. A particular method is developed to take full advantage of the usual avoidance techniques while maintaining safety. These works have been applied on a 6-DOF manipulator operating in a cluttered environment. The results obtained confirm the reliability of the approach and validates the expected performances.

Future works will address new applications and extensions of the proposed methodology. This approach can be applied to other control levels (*e.g.* torque control) and include other type of constraints related to physical limits (torque limits, jerk limits, power limits, etc.) or user specifications (contact persistence, co-manipulation, etc.). The extensions of the presented methodology include the adaptations of the work carried out on ICS to IVS. The concept of ICS has generated a significant amount of works in the field of mobile robotics: ICS-checker in the 2D case (Martinez-Gomez *et al.* 2008), solutions to approximate the ICS set (Parthasarathi *et al.* 2007), probabilistic approaches (Althoff *et al.* 2010, Bautin *et al.* 2010), etc. These works offer many perspectives to increase the use of safe multibody robots.

References

Althoff, D., Althoff, M., Wollherr, D., and Buss, M. (2010). Probabilistic collision state checker for crowded environments. In *Proceedings of the 2010 IEEE International Conference On Robotics and Automation*, pages 1492–1498.

Bautin, A., Martinez-Gomez, L., and Fraichard, T. (2010). Inevitable Collision States: A probabilistic perspective. In *Proceedings of the 2010 IEEE International Conference On Robotics and Automation*, pages 4022–4027.

Biagiotti, L. and Melchiorri, C. (2008). *Trajectory planning for automatic machines and robots*. Springer Verlag.

Brady, M. (1982). *Robot motion: Planning and control*. The MIT Press.

David, O., Measson, Y., Bidard, C., Rotinat-Libersa, C., and Russotto, F.-X. (2007). Maestro: a hydraulic manipulator for maintenance and decommissioning application. In *Transaction of the European Nuclear Conference*.

Decre, W., Smits, R., Bruyninckx, H., and De Schutter, J. (2009). Extending iTaSC to support inequality constraints and non-instantaneous task specification. In *Proceedings of the 2009 IEEE International Conference On Robotics and Automation*, pages 964–971.

Escande, A., Mansard, N., and Wieber, P.-B. (2010). Fast Resolution of Hierarchized Inverse Kinematics with Inequality Constraints. In *Proceedings of the 2010 IEEE International Conference On Robotics and Automation*, pages 3733–3738.

Faverjon, B. and Tournassoud, P. (1987). A Local Based Approach for Path Planning of Manipulators With a High Number of Degrees of Freedom. In *Proceedings of the 1987 IEEE International Conference On Robotics and Automation*, pages 1152–1159.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33.

Fraichard, T. (2007). A short paper about motion safety. In *Proceedings of the 2007 IEEE International Conference On Robotics and Automation*, pages 1140–1145.

Fraichard, T. and Asama, H. (2004). Inevitable collision states. A step towards safer robots? *Advanced Robotics*, 18(10):1001–1024.

Gicquel, P., Andriot, C., Lauture, F., Measson, Y., and Desbats, P. (2001). TAO2000: a generic control architecture for advanced computer aided teleoperation systems. In *Proceedings of the 9th ANS Topical Meeting on Robotics and Remote Systems*.

Guilbert, M., Joly, L., and Wieber, P.-B. (2008). Optimization of Complex Robot Applications under Real Physical Limitations. *The International Journal of Robotics Research*, 27(5):629–644.

Haddadin, S., Albu-Schffer, A., Eiberger, O., and Hirzinger, G. (2010). New Insights Concerning Intrinsic Joint Elasticity for Safety. In *Proceedings of the 2010 IEEE-RSJ International Conference On Intelligent Robots and Systems*, pages 2181–2187.

Ikuta, K., Ishii, H., and Nokata, M. (2003). Safety evaluation method of design and control for human-care robots. *The International Journal of Robotics Research*, 22(5):281.

Kanehiro, F., Lamiraux, F., Kanoun, O., Yoshida, E., and Laumond, J.-P. (2008). A local collision avoidance method for non-strictly convex polyhedra. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland.

Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98.

Khatib, O., Yokoi, K., Brock, O., Chang, K., and Casal, A. (2001). Robots in Human Environments. *Archives of Control Sciences, Special Issue on Recent Developments in Robotics*, I.11(3-4):123–138.

Kröger, T. (2010). *On-Line Trajectory Generation in Robotic Systems*, volume 58 of *Springer Tracts in Advanced Robotics*. Springer, Berlin, Heidelberg, Germany.

Maciejewski, A. and Klein, C. (1985). Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The International Journal of Robotics Research*, 4(3):109–117.

Martinez-Gomez, L. and Fraichard, T. (2008). An efficient and Generic 2D Inevitable Collision State-Checker. In *Proceedings of the 2008 IEEE-RSJ International Conference On Intelligent Robots and Systems*, pages 234–241.

Martinez-Gomez, L. and Fraichard, T. (2009). Collision avoidance in dynamic environments: an ics-based solution and its comparative evaluation. In *Proceedings of the 2009 IEEE International Conference On Robotics and Automation*, pages 100–105.

Park, J. and Khatib, O. (2008). Robot multiple contact control. *Robotica*, 26(5):667–677.

Parthasarathi, R. and Fraichard, T. (2007). An Inevitable Collision State-Checker for a Car-Like Vehicle. In *Proceedings of the 2007 IEEE Inter-*

national Conference On Robotics and Automation, pages 3068–3073.

Rubrecht, S., Padois, V., Bidaud, P., and Broissia, M. (2010a). Constraint compliant control for a redundant manipulator in a cluttered environment. *Advances in Robot Kinematics: Motion in Man and Machine*, pages 367–376.

Rubrecht, S., Padois, V., Bidaud, P., and de Broissia, M. (2010b). Constraints Compliant Control: constraints compatibility and the displaced configuration approach. In *Proceedings of the 2010 IEEE-RSJ International Conference On Intelligent Robots and Systems*, pages 677–684.

Zinn, M., Khatib, O., Roth, B., and Salisbury, J. (2004). Playing it safe. *IEEE Robotics & Automation Magazine*, 11(2):12–21.