Minimization of the rate of change in torques during motion and force control under discontinuous constraints

Yang Tan, Darwin Lau, Mingxing Liu, Philippe Bidaud, Vincent Padois

Minimization of the Rate of Change in Torques during Motion and Force Control Under Discontinuous Constraints

Yang Tan[1], Darwin Lau[1], Mingxing Liu[1], Philippe Bidaud[1,2] and Vincent Padois[1]

***Abstract*— Large and sudden changes in the torques of the motors of a robot are highly undesirable and should be avoided during robot control as they may result in unpredictable behaviours. One cause of large changes in torques is the presence of discontinuities in the constraints that the robot must satisfy, such as the avoidance of an obstacle or the breaking of contacts with the environment. In this paper, a model predictive control (MPC) approach to approximate constraints that can be predicted over a finite horizon is proposed to minimize the derivative of torques during robot control. The proposed method does not directly modify the desired task trajectory but the constraints to ensure that the worst case of changes in torques is well-managed. From the simulation results on the control of a Kuka LWR robot, it is shown that our approach significantly decreases the maximum derivative of joint torques for both force and acceleration task control examples.**

I. INTRODUCTION

The redundancy of robots makes it possible to simultaneously execute complex tasks while satisfying constraints. These constraints can either be related to some intrinsic limitations of the system, such as joint limits and actuation capacities, or to the surrounding environment, for example obstacles to avoid and contacts to maintain. These constraints have to be accounted for when solving the control problem and thus must be monitored. More specifically constraints can be: 1) added to the control problem when needed, e.g. when a contact is established; 2) updated within the control problem, e.g. when the distance to an obstacle is updated based on some sensor information; 3) removed from the control problem, e.g. when a contact is broken.

The addition, removal or abrupt change of a constraint is a source of discontinuity for the control law and may result in large instantaneous changes in the actuation torque and unpredictable behaviours of robots. Rapid changes in the motor torques are undesired for many reasons: damage to the actuators, bad control performance or even unstable robot motion. In fact, the problem of large rate of change in the actuation torque has not yet been fully resolved.

The problem of large changes in motor torques has been related to different types of causes and studied in the literature. First, sudden changes of task target in operational space can lead to great discontinuities in joint torques. In this case trajectory planning is a proper way to solve this problem. In [1] a minimum jerk path generation is proposed to produce a continuous Cartesian trajectory. The work [2] introduces a time-optimal, jerk-limited online trajectory planning, complying with the maximal velocity, acceleration and jerk of joints. In [3] a jerk-bounded trajectory planner is used to generate continuous trajectories online. Second, changes in hierarchical priorities amongst a set of tasks can also lead to discontinuities in joint torques. A large number of works [4]–[6] have aimed to create a continuous null space projector that allows switches among task priorities without causing large changes in motor control torques. A generalized continuous projector is used to deal with the task transitions in [7] with an optimization approach. Moreover, constraints can be handled by the task formulations. The method of classical artificial potential fields [8] is widely used to achieve the avoidance of obstacles. However, when the obstacle avoidance constraint switches between inactive and active, a discontinuous control law is produced. In [9] a reactive continuous null space projection is extended to prevent large changes in torques while activating/deactivating the constraints. The intermediate desired value approach [10] is applied to avoid large changes in torques by smoothly activating the obstacle avoidance constraints.

In addition to the previously identified causes of large rate of change in torques, discontinuities in constraints are also able to induce sudden changes in torques and have not been previously studied. When considering the discontinuity of constraints, two different situations may occur. In some situations, sudden changes in constraints cannot be known in advance and hence the controller cannot avoid the induced large change in motor torques. However, if the appearance, disappearance or sudden change of constraints can be known in advance, then the controller can handle this in a predictive manner. Changes in constraints can be known in advance in several ways, for example:

- constraint from the control scenario for the system can be anticipated, such as when a robot is about to sit on a chair or to put a foot on the ground;
- constraint is sensed ahead of time, such as the detection of an obstacle using vision before the obstacle avoidance actually becomes a constraint.

Optimization is one common approach to determine the motor torques required to produce a desired motion subject to constraints. This type of formulation is starting to prevail in Robotics as it allows to properly account for constraints which are expressed as inequalities [11]. If the objective

[1]Yang Tan, Darwin Lau, Mingxing Liu, Philippe Bidaud and Vincent Padois are with:
-Sorbonne Université, UPMC Univ Paris 06, UMR 7222, Institut des Systèmes Intelligents et de Robotique, F-75005, Paris, France
-CNRS Centre National de la Recherche Scientifique, UMR 7222, Institut des Systèmes Intelligents et de Robotique, F-75005, Paris, France
Email:{tan,lau,liu,bidaud,padois}@isir.upmc.fr
[2]Philippe Bidaud is with the ONERA, 91123 Palaiseau, France
Email:philippe.bidaud@onera.fr

function is quadratic and the system is subject to linear constraints, Quadratic Program (QP) solvers can be efficiently employed [12]–[15]. In this scheme, the computed torques may exhibit sudden changes due to three different reasons: 1) if the objective function is discontinuous; 2) if the solution of the QP is on the boundary of the constraint and the constraint is discontinuous; and 3) if the solution of the QP transitions between the interior and the boundary of the feasible set.

In this paper, a predictive control scheme to minimise the rate of change in torques over time in the presence of discontinuities in constraints is proposed. Assuming that information about the evolution of constraints within a finite horizon is known, a Model Predictive Control (MPC) scheme is employed. The MPC is used to design a new constraint that results in minimised changes in joint torques compared with the original discontinuous constraints. The new constraint, which is shown to be more conservative, replaces the original constraint, and the new constraints are used within a reactive controller scheme. The effectiveness of the proposed approach is demonstrated through simulations on a Kuka LWR robot with kinematic and force constraint discontinuities. The results show a significantly lower instantaneous change in joint torques compared with the controller output if the original discontinuous constraints are used.

One key feature of this approach is that the constraints of the reactive controller are modified rather than the objective function. As such, the proposed method does not directly modify the task trajectory but ensure that the worst case torque derivative is minimised. It is a very generic approach which can be applied independently from the way the control law is formulated. Furthermore, the proposed MPC formulation provides the potential to adjust the balance between the worst case change in joint torques and the conservativeness of the new constraint.

II. Control Framework

The equation of motion of a fixed-base robot with n degrees of freedom (DoF) can be derived from the Euler-Lagrange formalism and expressed as

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} + J_c(\boldsymbol{q})^T \boldsymbol{F}_c \ , \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are respectively the generalised coordinates, velocities and accelerations, $M(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the generalized inertia matrix, $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ is the vector of Coriolis, centrifugal and gravity induced joint torques, and $\boldsymbol{\tau}$ is the vector of joint torques. $\boldsymbol{F}_c$ is the external contact force, $J_c(\boldsymbol{q})^T$ is the Jacobian at the contact point. Defining the *action variable* as $\boldsymbol{\chi} = \begin{bmatrix} \ddot{\boldsymbol{q}}^T & \boldsymbol{\tau}^T & \boldsymbol{F}_c^T \end{bmatrix}^T$ allows the equation of motion (1) to be represented linearly as

$$A\boldsymbol{\chi} = \boldsymbol{b} \ . \tag{2}$$

A. Tasks

A motion task of a frame attached to the robot here is defined as the tracking of a desired trajectory in operational space. The kinematic relationship between the operational and joint spaces can be expressed as $\dot{\boldsymbol{x}} = J(\boldsymbol{q})\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{x}} = J(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{J}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$, where $J(\boldsymbol{q})$ and $\dot{J}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the task Jacobian matrix and its derivative, respectively. The tracking error $\boldsymbol{T}(\boldsymbol{\chi})$ relative to a 3D trajectory $\boldsymbol{x}^{ref}$ can be expressed linearly as $\boldsymbol{T}(\boldsymbol{\chi}) = J(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{J}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \ddot{\boldsymbol{x}}^{cmd}$, where the task command $\ddot{\boldsymbol{x}}^{cmd}$ is computed using a proportional-derivative (PD) controller with a feedforward term $\ddot{\boldsymbol{x}}^{ref}$: $\ddot{\boldsymbol{x}}^{cmd} = \ddot{\boldsymbol{x}}^{ref} + K_p(\boldsymbol{x} - \boldsymbol{x}^{ref}) + K_d(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^{ref})$, and K_p and K_d are the proportional and derivative gains, respectively.

When a body of the robot is expected to apply a desired force in operational space, the force error is: $\boldsymbol{T}(\boldsymbol{\chi}) = \boldsymbol{F}_c - \boldsymbol{F}_c^{ref}$, where $\boldsymbol{F}_c^{ref}$ is the desired value of the force.

B. Constraints

While performing the task, the robot must respect constraints, such as joint limits and collision avoidance with obstacles. It is shown in this section that different types of constraints can be expressed linearly with respect to the action variable $\boldsymbol{\chi}$ in the form

$$G\boldsymbol{\chi} \leq \boldsymbol{h} \ . \tag{3}$$

1) Joint limits: Bounds on the joint positions and velocities can be locally expressed with respect to the joint accelerations based on a discrete linear approximation with a time step of δt:

$$\dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}}_k + \ddot{\boldsymbol{q}}_k \delta t \leq \dot{\boldsymbol{q}}_{max} \ , \tag{4}$$

$$\boldsymbol{q}_{min} \leq \boldsymbol{q}_k + \dot{\boldsymbol{q}}_k \delta t + \ddot{\boldsymbol{q}}_k \frac{\delta t^2}{2} \leq \boldsymbol{q}_{max} \ , \tag{5}$$

where $\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k$ and $\ddot{\boldsymbol{q}}_k$ are the joint positions, velocities and accelerations, respectively. Bounds on joint torques can be directly expressed:

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau}_k \leq \boldsymbol{\tau}_{max} \ . \tag{6}$$

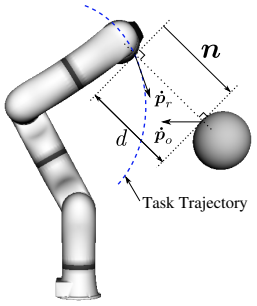


Fig. 1: Description of the obstacle avoidance constraint.

2) Obstacle avoidance constraint: As shown in Figure 1, the obstacle avoidance constraint can be defined as requiring the minimum distance $d = \|\boldsymbol{p}_{r,k} - \boldsymbol{p}_{o,k}\|$ between the robot body and the obstacle to be strictly non-zero, where $\boldsymbol{p}_{\bullet,k} = [x_{\bullet,k} \ y_{\bullet,k} \ z_{\bullet,k}]^T$ represents the a position in Cartesian space. Using a discrete linear approximation with a time step of δt, the minimum distance d_{k+1} can be expressed as

$$d_{k+1} = d_k + \boldsymbol{n}^T(\dot{\boldsymbol{p}}_{r,k} - \dot{\boldsymbol{p}}_{o,k})\delta t + \frac{\delta t^2}{2}\boldsymbol{n}^T(\ddot{\boldsymbol{p}}_{r,k} - \ddot{\boldsymbol{p}}_{o,k}) \ , \tag{7}$$

where $\boldsymbol{n}$ is the vector associated to the shortest distance from the robot to the object, which is assumed to evolve continuously in this paper. $\boldsymbol{p}_{r,k}$ and $\dot{\boldsymbol{p}}_{r,k}$ are the current position and velocity of the closest point of the robot with respect to the obstacle, $\ddot{\boldsymbol{p}}_{r,k}$ is the acceleration resulting from the next control action. They evolves continuously according to the continuous motion task or force task. It is worth noting that constraint (7) relies on the knowledge of $\boldsymbol{p}_{o,k}, \dot{\boldsymbol{p}}_{o,k}$ and $\ddot{\boldsymbol{p}}_{o,k}$ which are the current position, velocity and acceleration of the obstacle. If the velocity and acceleration of the obstacle are unknown, *i.e.* neither measured nor estimated, (7) assumes that the obstacle is quasi-static over a control period which can be a valid working assumption. Using the kinematic relationship between the operational space velocity/acceleration and the generalized coordinates in (7), the obstacle avoidance constraint can be expressed linearly with respect to $\ddot{\boldsymbol{q}}_k$ as:

$$\begin{aligned} & d_k + \boldsymbol{n}^T(J(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k - \dot{\boldsymbol{p}}_{o,k})\delta t \\ & + \frac{\delta t^2}{2}\boldsymbol{n}^T(J(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_k + \dot{J}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k - \ddot{\boldsymbol{p}}_{o,k}) \geq 0 \ . \end{aligned} \tag{8}$$

3) Contact constraint: Contact forces can be constrained according to different scenarios. For example, when there is no contact, contact force must be 0. For safety reasons, it can also be necessary to limit the maximum value of the contact force. In this case, the contact constraint can be written:

$$0 \leq f_n = \boldsymbol{n}_f^T \boldsymbol{F}_c \leq f_{max} \ , \tag{9}$$

where $f_n = \boldsymbol{n}_f^T \boldsymbol{F}_c$ is the normal contact force along the normal direction to the contact surface $\boldsymbol{n}_f$, and F_{max} is the maximum allowable force related to the scenarios.

C. Control Framework

The optimal action variable $\boldsymbol{\chi}^*$ can be determined through QP by minimising a weighted sum of the task error $\|\boldsymbol{T}(\boldsymbol{\chi})\|^2$ and the action effort $\|\boldsymbol{\chi}\|^2$, while satisfying constraints:

$$\begin{aligned} \boldsymbol{\chi}^* = \arg\min_{\boldsymbol{\chi}} \quad & \|\boldsymbol{T}(\boldsymbol{\chi})\|_Q^2 + \|\boldsymbol{\chi}\|_R^2 \\ \text{subject to} \quad & A\boldsymbol{\chi} = \boldsymbol{b},\ G\boldsymbol{\chi} \leq \boldsymbol{h} \end{aligned} \ . \tag{10}$$

The notation $\|\boldsymbol{a}\|_Q^2$ is the shorthand of the form $\boldsymbol{a}^T Q \boldsymbol{a}$. The weighting matrices Q and R in (10) allow to modulate the importance between the task objective and the regularization term. As a reactive controller, this QP is solved at each time step. Observing that (10) is a strictly convex QP problem, a unique globally optimal solution for the redundant robot can be determined. For this type of problem with constraints (4)-(8), G and A are matrices related to the rigid body model of the system and are always continuous. So time discontinuous solutions to (10) can occur under following situations:

- The original task reference is discontinuous. This task discontinuity could be handled by trajectory planning and the task reference is assumed to be continuous throughout this work;

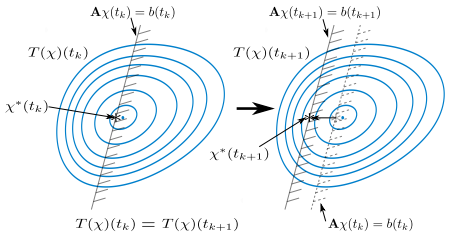


Fig. 2: The change of b_i in equality constraints can cause change in joint torques. The solution always lies on the constraint $A\boldsymbol{\chi} = \boldsymbol{b}$.

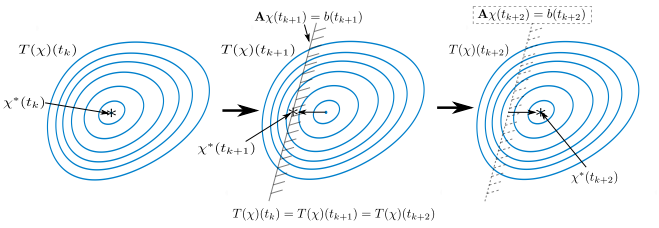


Fig. 3: The change of the dimension of $\boldsymbol{b}$ can cause change in joint torques. A new equality constraint is added at t_{k+1} and then is removed at t_{k+2}. The solution $\boldsymbol{\chi}$ may undergo a large change.

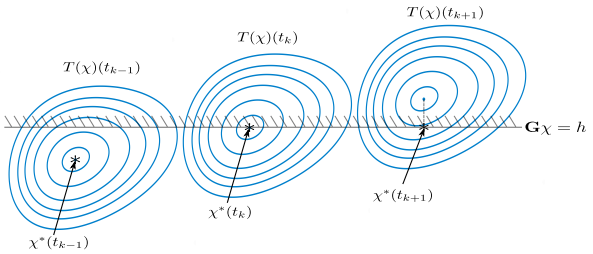


Fig. 4: Due to the evolution of task reference, the inequality constraint switches from inactive to active, which can cause large instantaneous changes in joint torques.

- Equality constraints (2) are active at all feasible solutions[1]. Assuming that the task reference is constant and A is continuous, there are two possible cases that large instantaneous changes in joint torques may occur. 1) Any b_i in $\boldsymbol{b} = [b_1 \ \dots b_m\ ] \in \mathbb{R}^m$ has large instantaneous changes. For example, in Figure 2, the large change of b_i can cause large change in joint torques. 2) The dimension of $\boldsymbol{b}$ increases or decreases with additions and removals of equality constraints, which is shown in Figure 3.
- Assuming G is continuous, the solution changes from being on the interior of the feasible set of (3) to the boundary, or vice versa (see in Figure 4). In other words, any of the constraints $G_i\boldsymbol{\chi} \leq h_i$ switches between inactive and active.

[1]If $\boldsymbol{\chi}$ is feasible and $G_i\boldsymbol{\chi} = h_i$, we say the ith inequality constraint $G_i\boldsymbol{\chi} \leq h_i$ is active at $\boldsymbol{\chi}$. If $G_i\boldsymbol{\chi} < h_i$, we say the constraint $G_i\boldsymbol{\chi} \leq h_i$ is inactive [16, p. 128].

III. MPC Continuous Constraint Generation

The discontinuous evolution of any active constraint causes inevitably discontinuous control torques as stated in Section II, which is not desirable. Therefore, smoothing the discontinuous evolution of inequality constraints is a prerequisite to minimize the changes in joint torques. As shown in Figure 5, given the information from sensors and/or environment contexts, we can know discontinuous variations in a look ahead window of N steps with T being the time step. A continuous constraint h_i^* can thus be generated. Then it can replace h_i in the constraint (3):

$$G\boldsymbol{\chi} \leq \boldsymbol{h}^* \ . \tag{11}$$

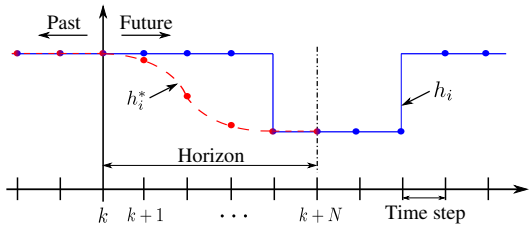


Fig. 5: Smoothing the discontinuous evolution of inequality constraints using a receding horizon technique.

Model Predictive Control (MPC) [17] is a good candidate to generate a smooth evolution of $\boldsymbol{h}^*$ with respect to $\boldsymbol{h}$. MPC, also known as receding horizon control [18], is based on iterative, finite horizon optimization of a dynamic model. MPC uses the current dynamic states of the system, the dynamic model, the variable targets and limits to calculate a set of future inputs. These inputs can hold variables close to their targets as well as respect their constraints in the future. At time step k, the current states are measured and a cost minimizing the tracking errors is computed for a time horizon $[k, k+N]$ to find a sequence of the system inputs until step $k+N$. Only the first step of the system inputs is fed to update the states of the system, then the calculation shifts to the next time step, yielding repeatedly a new set of system future inputs. With these properties, MPC is well qualified to generate smooth constraints evolution with respect to discontinuous constraints.

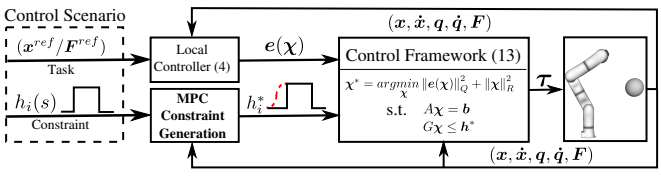


Fig. 6: Block diagram of the control framework.

The overall control framework is shown in Figure 6. MPC is used to deal with discontinuous constraints by generating a new constraint evolution that can minimise the rate of change in joint torques in the QP control framework. According to the types of constraints, two main continuous constraint generation based on MPC are presented in this section.

A. Continuous Force Constraint Generation

For the force constraint (9), the goal of MPC is to generate smooth maximum allowable force $\overline{f}$ with respect to f_{max}. Then, it can replace f_{max} in constraint (9):

$$0 \leq f_n \leq \overline{f} \ . \tag{12}$$

A discrete-time model of the force can be expressed as:

$$\overline{f}_{k+1} = \overline{f}_k + \dot{\overline{f}}_k T \ , \tag{13}$$

where $\overline{f}_k$ is the maximum allowable forces at time step k. $\dot{\overline{f}}_k$ is the first-order time derivative of $\overline{f}_k$. $\overline{f}_k$ should be optimized according to the original profile of f_{max}. Within the time horizon NT, the MPC for generating a optimized $\overline{f}$ can be formulated as:

$$\dot{\overline{f}}_k^* = \underset{\dot{\overline{f}}_k, \ldots, \dot{\overline{f}}_{k+N}}{\arg\min} \sum_{j=k}^{k+N} \|\overline{f}_j - f_{max,j}\|^2 + \gamma \|\dot{\overline{f}}_j\|^2 \tag{14a}$$

$$\text{subject to} \quad \overline{f}_{j+1} = \overline{f}_j + \dot{\overline{f}}_j T \tag{14b}$$

$$0 \leq \overline{f}_j \leq f_{max,j} \tag{14c}$$

where (14c) ensures that at each time step j every solution $\boldsymbol{\chi}^*$ that satisfies the constraint (12) also satisfies the constraint (9). The coefficient γ is the weight, governing the importance between the changes of $\overline{f}$ and the deviation from the original maximum allowable force. In (14), $\dot{\overline{f}}_k^*$ can be computed by a Quadratic Program solver. The generated maximum allowable force is $\overline{f}_{k+1} = \overline{f}_k + \dot{\overline{f}}_k^* T$.

Note that an alternative method to address this problem is to smooth the profile of f_{max} by applying a smooth polynomial spline [19,20]. However, the advantage of using MPC here is to be able to handle multiple constraints in an automatic and generic way, *e.g.* to ensure that the optimized $\overline{f}$ is compatible with other constraints. For example, by using MPC, two contact forces can easily satisfy the following three constraints simultaneously: 1) $0 \leq \overline{f}_1 \leq f_{1,max}$; 2) $0 \leq \overline{f}_1 \leq f_{2,max}$, and 3) $\overline{f}_1 + \overline{f}_2 \leq f_{3,max}$. However, it is difficult for the polynomial to deal with this problem. Moreover, the use of a polynomial approach requires manual choose of the start and end points of the segment to be smoothed and manual tuning of polynomial parameters and may lead to oscillation problems [21].

B. Continuous Motion Constraint Generation

According to the obstacle avoidance constraint (8), h_i can be expressed as:

$$h_i = d_k(\boldsymbol{p}_r, \boldsymbol{p}_o) + \boldsymbol{n}^T(J(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{p}}_o)\delta t + \frac{\delta t^2}{2}\boldsymbol{n}^T(\dot{J}(\boldsymbol{q})\dot{\boldsymbol{q}} - \ddot{\boldsymbol{p}}_o) \ . \tag{15}$$

h_i is a function of the position, velocity and acceleration of the obstacle. Considering the fact that the movement of the obstacle along any axis in operational space is independent, without loss of generality the position, velocity and acceleration along the $x-$axis are formulated in the following part of

this paper. In this case, a linear discrete-time dynamic model of the obstacle can be created in state space form:

$$\underbrace{\begin{bmatrix} x_{o,k+1} \\ \dot{x}_{o,k+1} \\ \ddot{x}_{o,k+1} \end{bmatrix}}_{\boldsymbol{X}_{o,k+1}} = \underbrace{\begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_{o,k} \\ \dot{x}_{o,k} \\ \ddot{x}_{o,k} \end{bmatrix}}_{\boldsymbol{X}_{o,k}} + \underbrace{\begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix}}_{B} \dddot{x}_{o,k} \ , \tag{16}$$

where $\boldsymbol{X}_{o,k}$ and $\dddot{x}_{o,k}$ are the state vector of the obstacle and the control action at time k, respectively. Matrices A and B implicitly describe the linearity of the system.

Using the dynamic model (16) recursively, at time k, the relationships between the control action vector and state vectors over a finite time horizon NT is given by:

$$\hat{\boldsymbol{X}}_o = \hat{A}\boldsymbol{X}_{o,k} + \hat{B}\boldsymbol{U}_o \ , \tag{17}$$

where,

$$\hat{\boldsymbol{X}}_o = \begin{bmatrix} \boldsymbol{X}_{o,k+1|k} \\ \boldsymbol{X}_{o,k+2|k} \\ \vdots \\ \boldsymbol{X}_{o,k+N|k} \end{bmatrix}, \quad \boldsymbol{U}_o = \begin{bmatrix} \dddot{x}_{o,k|k} \\ \dddot{x}_{o,k+1|k} \\ \vdots \\ \dddot{x}_{o,k+N-1|k} \end{bmatrix},$$

$$\hat{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.$$

Over the time horizon, only the position of the obstacle is measured by scenarios in advance, which is $\boldsymbol{x}_o^m = \begin{bmatrix} x_{o,k+1}^m & \cdots & x_{o,k+N}^m \end{bmatrix}^T$. In order to take advantage of $\boldsymbol{x}_o^m$, the model predicted position information can be extracted from $\hat{\boldsymbol{X}}_o$: $\hat{\boldsymbol{X}}_{o,x} = \begin{bmatrix} x_{o,k+1|k} & \cdots & x_{o,k+N|k} \end{bmatrix}^T$. Consequently, $\hat{A}_x$ and $\hat{B}_x$ can be extracted from $\hat{A}$ and $\hat{B}$, respectively. Therefore, the predicted position of the obstacle over the horizon can be formulated as:

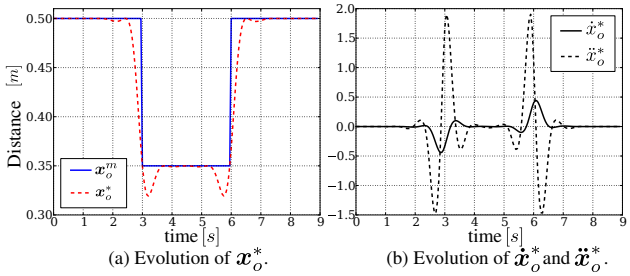
$$\hat{\boldsymbol{X}}_{o,x} = \hat{A}_x\boldsymbol{X}_{o,k} + \hat{B}_x\boldsymbol{U}_o \ . \tag{18}$$

Thus similarly to (14), the general form of the MPC problem to be solved is formulated as

$$\begin{aligned} \dddot{x}^*_{o,k|k} = \min_{\boldsymbol{U}_o} \quad & \alpha \left\| \hat{\boldsymbol{X}}_{o,x} - \boldsymbol{x}_o^m \right\|^2 + \beta \|\boldsymbol{U}_o\|^2 \\ \text{subject to} \quad & \hat{\boldsymbol{X}}_{o,x} = \hat{A}_x\boldsymbol{X}_{o,k} + \hat{B}_x\boldsymbol{U}_o \\ & \hat{\boldsymbol{X}}_{o,x} \leq \boldsymbol{x}_o^m \end{aligned} \ . \tag{19}$$

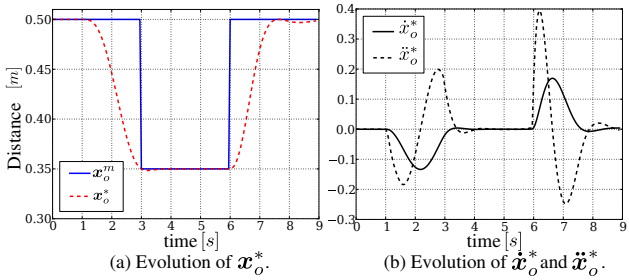
Then, at time k, $\dddot{x}^*_{o,k|k}$ can be calculated by a QP solver and it is used to update $\boldsymbol{X}^*_{o,k+1} = A\boldsymbol{X}^*_{o,k} + B\dddot{x}^*_{o,k|k}$. Therefore, continuous evolution of $\boldsymbol{X}_o^* = [\boldsymbol{x}_o^* \ \dot{\boldsymbol{x}}_o^* \ \ddot{\boldsymbol{x}}_o^*]^T$ can be generated (similarly for $\boldsymbol{Y}_o^*$ and $\boldsymbol{Z}_o^*$).

To illustrate this concept, a simple example is provided to handle the discontinuities in $\boldsymbol{x}_o^m$. MPC (19) is used with time step $T = 0.01s$, constant ratio $\alpha/\beta = 10^5$ and time horizon $NT = 2.0s$. In Figure 7(a), the generated $\boldsymbol{x}_o^*$ is continuous compared with the original position $\boldsymbol{x}_o^m$. Moreover MPC also generates continuous $\dot{\boldsymbol{x}}_o^*$ and $\ddot{\boldsymbol{x}}_o^*$ in Figure 7(b). With the constant ratio of α/β, MPC minimizes the error between the



(a) Evolution of $\boldsymbol{x}_o^*$. (b) Evolution of $\dot{\boldsymbol{x}}_o^*$ and $\ddot{\boldsymbol{x}}_o^*$.

Fig. 7: The optimized constraint with $T = 0.01s$, $NT = 2.0s$ and $\alpha/\beta = 10^5$.



(a) Evolution of $\boldsymbol{x}_o^*$. (b) Evolution of $\dot{\boldsymbol{x}}_o^*$ and $\ddot{\boldsymbol{x}}_o^*$.

Fig. 8: The resulting smooth constraint with $T = 0.01s$, $NT = 2.0s$, $\alpha/\beta = 10^5$ and a dynamic weighting matrix Q.

solution of $\boldsymbol{x}_o^*$ and the original position as well as reduces the instantaneous variation of $\ddot{\boldsymbol{x}}_o$.

It is shown in Figure 7 that there exist overshoots at $t = 3.0s$ and $t = 6.0s$. These overshoots are mainly due to a relatively large weight α associated to $\|\hat{\boldsymbol{X}}_0 - \boldsymbol{x}_o^{ref}\|^2$, which minimises the distance to the original reference, with respect to the weight β of the regulation term $\|\boldsymbol{U}_o\|^2$, which reduces the instantaneous variation of $\ddot{x}_o$. Thus a dynamic weighting matrix Q is added into (19):

$$\begin{aligned} \dddot{x}^*_{o,k|k} = \min_{\boldsymbol{U}_o} \quad & \alpha \|\hat{\boldsymbol{X}}_{o,x} - \boldsymbol{x}_o^m\|_Q^2 + \beta \|\boldsymbol{U}_o\|^2 \\ \text{subject to} \quad & \hat{\boldsymbol{X}}_{o,x} = \hat{A}_x\boldsymbol{X}_{o,k} + \hat{B}_x\boldsymbol{U}_o \\ & \hat{\boldsymbol{X}}_{o,x} \leq \boldsymbol{x}_o^m \end{aligned} \ , \tag{20}$$

where $Q = diag(a_j, a_{j+1}, \ldots, a_{j+N})$ is a diagonal matrix, and $a_j \in (0, 1], \forall j \in [k, k+N]$ determines the weight associated to each sample in the time horizon. Each a_j is computed with respect to the variation of $\boldsymbol{x}_o^m$:

$$a_j = \frac{max(\boldsymbol{x}_o^m) - x_{o,j}^m + \lambda}{max(\boldsymbol{x}_o^m) - min(\boldsymbol{x}_o^m) + \lambda} \ , \tag{21}$$

where λ is a regulation term to avoid a division by zero. a_j is dynamically changing in the time horizon $[k, k+N]$:

1) If $max(\boldsymbol{x}_o^m) = min(\boldsymbol{x}_o^m)$, then $a_j = 1.0$. The relation between α and β is set to $\alpha \geq \beta$, which means the approximation to the reference is always more important than the minimization of $\boldsymbol{U}_o$
2) If $max(\boldsymbol{x}_o^m) \neq min(\boldsymbol{x}_o^m)$, discontinuities might occur in the time horizon, then a_j tends to 1.0 for large variations of the reference value and a_j tends to 0 for small ones. In this case the approximation to the original reference is sacrificed to reduce variations of $\boldsymbol{U}_o$ as soon as the discontinuity is previewed in the time horizon.

The results using the dynamic weighting matrix Q are shown in Figure 8, which shows that there is no overshoot on $\boldsymbol{x}_o^*$ and MPC begins to react to future discontinuity once the change is detected in the horizon. In Figure 8(b) the max values of $\dot{\boldsymbol{x}}_o^*$ and $\ddot{\boldsymbol{x}}_o^*$ are much less than those in Figure 7(b).

Here, MPC works as a preprocessing of constraints before they are incorporated into the control framework. Its advantages of previewing future events and taking actions consequently make it possible to reduce large constraint variations. This can handle torque discontinuities due to discontinuous evolutions of active constraints. However, it cannot solve the problem caused by activation of the constraint (as shown in Figure 4).

C. Continuous Activation of Constraints

The discrete linear approximation over the discrete time step δt described in Section II leads to a truncation error. Indeed, with the decrease of δt, the error of h_i increases dramatically. In [22] a constraint compliant control law is proposed to solve this problem, but it is only applied in the case of inverse kinematics. The approach in [23] increases the discretization time step δt in the constraint (not in the control time step) properly to diminish the error for dynamic control problem. In this paper, the same method is adopted to solve discontinuities in constraint activations.

IV. SIMULATION RESULTS & DISCUSSION

The proposed methods are applied to the control of a 7-DOF Kuka LWR robot in simulation. The experiments are carried out in the robotic simulation software XDE [24]. The Kuka robot is actuated by joint torques to perform tasks in operational space under discontinuous force constraints and obstacle avoidance constraints. The constraint generation approach presented in Section III is applied here to reduce the instantaneous variations of $\boldsymbol{\tau}$.

A. Handling of Force Constraint

In this experiment, the end-effector force task is defined to push against a fixed object with a constant force value. Meanwhile, the force constraint on the maximum allowed contact force evolves discontinuously. The results using the proposed approach is compared with the baseline approach, which is based on the QP control framework (10) with the min jerk term $\|\dddot{\boldsymbol{q}}\|_W^2$ in the objective function to minimize the changes of joint torques. W is the weight of the min jerk term. In this experiment, Q is identity matrix, $R = 10^{-6}$ and $W = 10^{-3}$.

In Figure 9, at the beginning the impact force results in a big peak on the force, which is not considered in this work. At $t = 5.0s$, the force constraint becomes suddenly active, causing torque discontinuities. At $t = 6.0s$ the force constraint changes discontinuously, and discontinuous torques and big torque derivative are observed. At $t = 9.0s$, the constraint decreases to zero suddenly, which means the contact is removed, leading to discontinuities of torques and big torque derivatives. Although the term $\|\dddot{\boldsymbol{q}}\|_W^2$ can limit changes of joint torques as a penalty, it cannot preview sudden large changes of constraint in advance. Once the sudden large changes of constraints occur, joint torques have to change largely to respect the constraints in a reactive way.

In Figure 10(a), the new generated continuous force constraint is used instead of the original one. In Figure 10(b), it is shown that the new continuous force constraint is fully respected. With this continuous force constraint, the joint torques evolve smoothly and significant decreases of torque derivatives can be observed in 10(d).

B. Handling of Obstacle Avoidance Constraint

In this experiment, the end-effector is tracking a sinusoid trajectory as shown in Figure 11 and the robot has to avoid collision with a moving object. While the robot is performing the tracking task, the obstacle moves suddenly towards the robot and thus the tracking task has to be stopped to avoid the obstacle. After a while, the object moves away from the robot. During the whole process, the robot should respect the discontinuous obstacle avoidance constraint and handle the discontinuities due to its activation and deactivation.

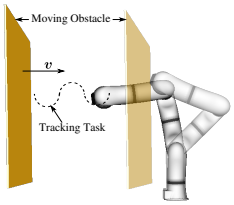


Fig. 11: The end-effector avoids a moving obstacle while performing the task.

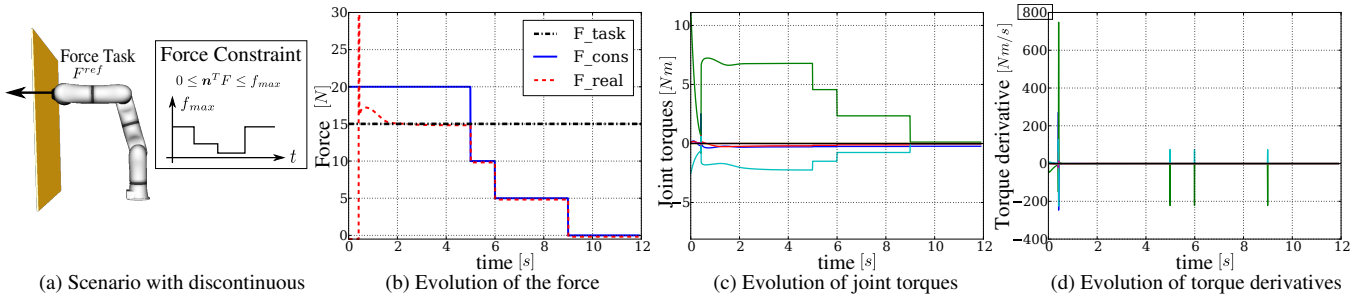
TABLE I: Maximum of joint torques and torque derivatives

Experiment	(i)	(ii)	(iii)	(iv)
Maximum of τ $[Nm]$	308.6	19.4	16.3	10.2
Maximum of $\dot{\tau}$ $[Nm/s]$	32358	1244	531	339

In order to demonstrate the effectiveness of the proposed methods, different experiments are carried out where:

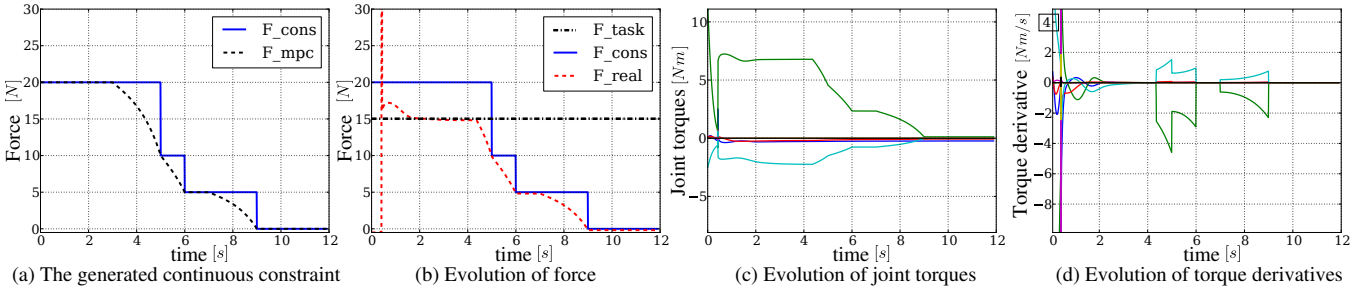
(i) only the QP control framework (10) with the original discontinuous obstacle avoidance constraint is used;
(ii) the new continuous obstacle avoidance constraint is generated by the proposed approach (19) with time horizon $NT = 1.5s$ and $\alpha/\beta = 10^5$ before being incorporated into the QP control framework (10);
(iii) the new continuous constraint is calculated based on the proposed approach (20) with the same time horizon and ratio as the experiment (ii);
(iv) the control framework is the same as the experiment (iii) but with a different ratio $\alpha/\beta = 10^3$.

Figure 12 shows the resulting trajectory of the robot, the position of the obstacle, evolution of joint torques and evolution of torque derivatives. In Figure 12(a), at $t = 3.8s$ and $t = 5.8s$ the obstacle moves suddenly and big peaks of joint torques and torque derivatives are clearly observed. The chattering of joint torques can be clearly observed



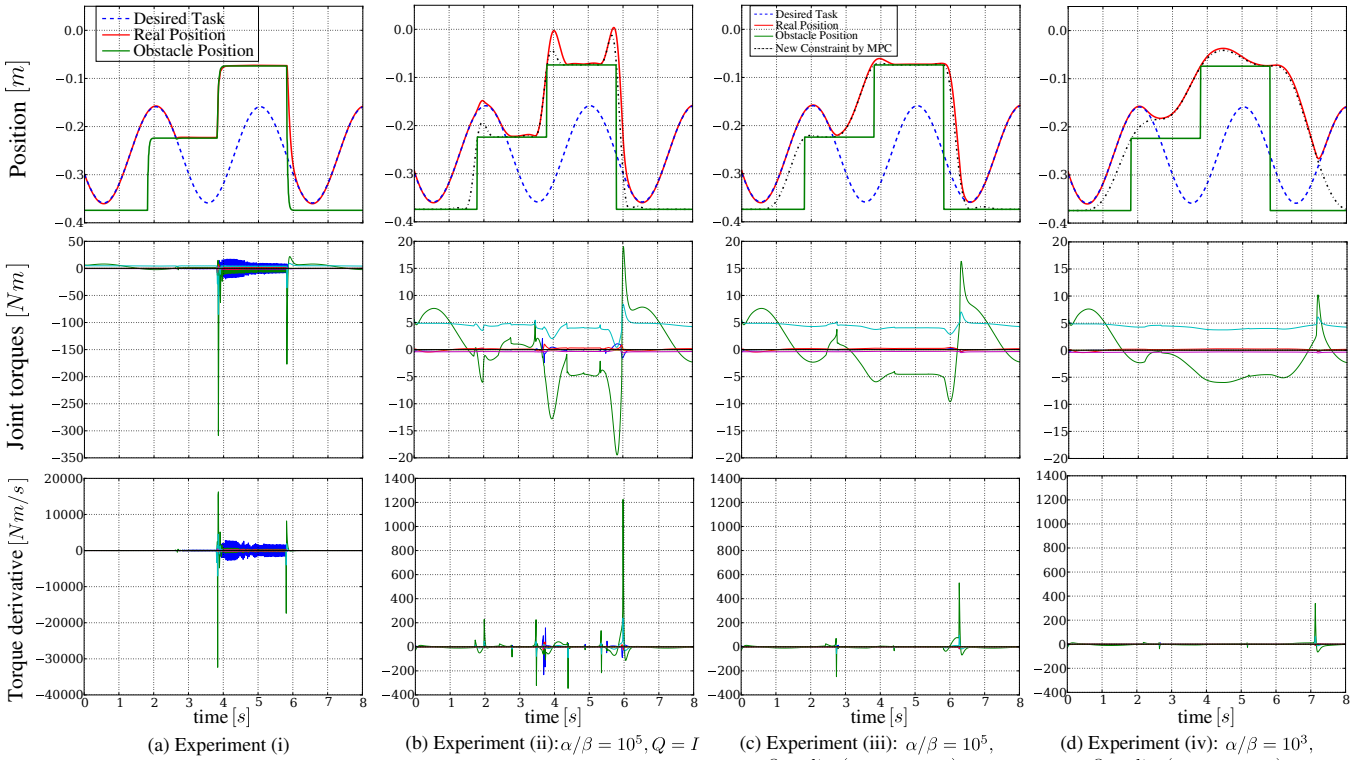
(a) Scenario with discontinuous force constrant (b) Evolution of the force (c) Evolution of joint torques (d) Evolution of torque derivatives

Fig. 9: Simulation results of the discontinuous force constraint with the minimum jerk term. The large torque derivatives are clearly observed in (d).



(a) The generated continuous constraint (b) Evolution of force (c) Evolution of joint torques (d) Evolution of torque derivatives

Fig. 10: Simulation results of the discontinuous force constraint with MPC smoothing. (d) The maximum of torque derivatives are significantly reduced.



(a) Experiment (i) (b) Experiment (ii): $\alpha/\beta = 10^5, Q = I$ (c) Experiment (iii): $\alpha/\beta = 10^5$, $Q = diag(a_j, \ldots, a_{j+N})$ (d) Experiment (iv): $\alpha/\beta = 10^3$, $Q = diag(a_j, \ldots, a_{j+N})$

Fig. 12: The resulting trajectory, joint torques and torque derivatives: the torque derivatives is greatly reduced in (c) compared with in (a).

when the constraint is active. This chattering of torques is caused by the truncation error. The results in Figure 12(b)-(d) demonstrate that the proposed approach can smooth the obstacle avoidance constraint and efficiently reduce torque

derivatives. The maximum observed joint torques and torque derivatives in Table I illustrate quantitatively the effectiveness of the proposed approach.

The weighting matrix Q is used to reduce the variation of the constraint itself in order to lower the rate of change on joint torques. Its effectiveness is demonstrated by the evolution of torque derivatives shown in Figure 12(c) compared with that in 12(b).

The ratio α/β determines the relative importance of minimization of the error between the new constraint and the original constraint over the minimization of instantaneous variation of the new constraint. A relative large α/β means that the new constraint is closer to the original one with relative large variations. Conversely, the distance to the original constraint is sacrificed to reduce variations with a relative small α/β. The influences of the ratio α/β are clearly shown in Figure 12(c) and 12(d): A larger α/β means the new continuous constraint is much closer to the original constraint with a much larger variation. This consequently leads to much bigger torque derivatives by comparing the maximal torques and torque derivatives in Table I with experiment (i) and (ii).

V. CONCLUSION

In this paper, a MPC-based constraint smoothing approach is proposed, which is applied to design a new constraint trajectory that results in minimised changes in joint torques compared with the original discontinuous constraints. Experiments involving a force constraint and an obstacle avoidance constraint show that the generated continuous constraints can successfully minimise instantaneous variations of joint torques caused by discontinuous constraints.

One key feature of this approach is that the constraints of the reactive controller are modified rather than the objective function. Therefore, the proposed method does not directly modify the task trajectory but ensure that the worst case torque derivative is minimised. As such, it is a very generic approach which can be applied independently from the way the control law and objective function is formulated.

Future work will focus on the application of this very general approach to the case of humanoids making and breaking contacts with their environment in simple activities such as walking, sitting, standing or leaning over an object.

ACKNOWLEDGMENT

This work was partially supported by the China Scholarship Council, and by the RTE company through the RTE/UPMC chair Robotics Systems for field intervention in constrained environments held by Vincent Padois.

REFERENCES

[1] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1988, pp. 364–369.

[2] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3248–3253.

[3] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 42–52, 2003.

[4] T. Petrič and L. Žlajpah, "Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem," *Robotics and Autonomous Systems*, vol. 61, no. 9, pp. 948–959, 2013.

[5] F. Keith, N. Wieber, P.-B.and Mansard, and A. Kheddar, "Analysis of the discontinuities in prioritized tasks-space control under discreet task scheduling operations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3887–3892.

[6] A. Dietrich, A. Albu-Schaffer, and G. Hirzinger, "On continuous null space projections for torque-based, hierarchical, multi-objective manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 2978–2985.

[7] M. Liu, S. Hak, and V. Padois, "Generalized projector for task priority transitions during hierarchical control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2015.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[9] A. Dietrich, T. Wimbock, A. Albu-Schaffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.

[10] H. Han, J. Lee, and J. Park, "A continuous task transition algorithm for operational space control framework," in *9th International Conference on Ubiquitous Robots and Ambient Intelligence*, 2012, pp. 148–152.

[11] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots," in *IEEE International Conference on Robotics and Automation*, no. 2, 2009, pp. 2939–2944.

[12] F.-T. Cheng, T.-H. Chen, Y.-S. Wang, and Y.-Y. Sun, "Obstacle avoidance for redundant manipulators using the compact qp method," in *IEEE International Conference on Robotics and Automation*, 1993, pp. 262–269.

[13] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 5, pp. 2126–2132, 2004.

[14] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1283–1290.

[15] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[17] J. Bellingham, Y. Kuwata, and J. How, "Stable receding horizon trajectory control for complex environments," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[18] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[19] C.-S. Lin, P.-R. Chang, and J. Y. S. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *Automatic Control, IEEE Transactions on*, vol. 28, no. 12, pp. 1066–1074, 1983.

[20] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for high performance motion systems," *Feedback*, vol. 14, p. 15, 2004.

[21] M. Schatzman and M. Schatzman, *Numerical analysis: a mathematical introduction*. Oxford University Press, 2002.

[22] S. Rubrecht, V. Padois, P. Bidaud, and M. De Broissia, "Constraints compliant control: constraints compatibility and the displaced configuration approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 677–684.

[23] J. Salini, "Dynamic control for the task/posture coordination of humanoids: toward synthesis of complex activities," These de doctorat, Université Pierre et Marie Curie, Paris, France, 2012.

[24] X. Merlhiot, J. Garrec, G. Saupin, and C. Andriot, "The xde mechanical kernel: efficient and robust simulation of multibody dynamics with intermittent nonsmooth contacts," in *International Conference on Multibody System Dynamics*, Stuttgart, Germany, 2012.