HAL open science

Adaptive-Gains Enforcing Constraints in Closed-Loop QP Control

Mohamed Djeha, Arnaud Tanguy, Abderrahmane Kheddar

To cite this version:

Mohamed Djeha, Arnaud Tanguy, Abderrahmane Kheddar. Adaptive-Gains Enforcing Constraints in Closed-Loop QP Control. 2020. hal-02490195v2

HAL Id: hal-02490195

https://hal.science/hal-02490195v2

Preprint submitted on 8 Jun 2020 (v2), last revised 27 Jul 2020 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive-Gains Enforcing Constraints in Closed-Loop QP Control

Mohamed Djeha[1], Arnaud Tanguy[2], and Abderrahmane Kheddar[1,2]

***Abstract*— In this letter, we revisit an open problem of constraints formulation in the context of task-space control frameworks formulated as quadratic programs. In most inverse dynamics implementations, the decision variables are: robot joints acceleration, interaction forces (mostly physical contacts), and robot torques. Nevertheless, many constraints, like distance and velocity bounds, are not written originally in terms of one of these decision variables. Previous work proposed solutions to formulate and enforce joint limits constraints. Yet, none of them worked properly in closed-loop, specifically when bounds are reached or when they are time-varying. First, we show that constraints like collision avoidance, bounds of center of mass, constraints on field-of-view, Cartesian and velocity bounds on a given link... are written in a generic class. Then, we formulate such a class of constraints with gain-parameterized ordinary differential inequality. An adaptive-gain method enforces systematically such class of constraints, and results on a stable behavior when their bounds (even when they vary with time) are reached in closed-loop. Experimental results performed on a humanoid robot validate our solution on a large panel of constraints.**

I. Introduction

Task-space multi-objective and multi-sensory control by means of soft or strict hierarchy quadratic programming (or *QP control* in short) is demonstrated in state-of-the-art humanoid robots realizing complex behaviors [1]–[6].

In almost all existing inverse dynamics QP controllers, the decision variables, are the contact forces, the torques, and the robot state acceleration. These are the output of the task-space controller sent as desired input to the low-level (actuators) controller[1]. The motivation of this paper came first some experimental failures. In the context of aircraft manufacturing robotics [6], we shall demonstrate the capability of humanoid robots in accessing narrow, cumbersome or confined spaces. In one specific use-case, our QP controller [3] repeatedly failed in the course of the execution at an extreme configuration. The log data revealed that similar failures happened when the joints reached their bounds, or in cases where joint velocities were too high. We also discovered later on that the QP either failed or generated oscillatory and discontinuous motions in closed-loop for other bounds (e.g. collision avoidance). Similar experimental failures encountered in closed-loop control have been reported in [7].

Formulating constraints, originally written in position or velocity spaces, in terms of their derivative to be expressed in decision variables is not new. Whether the robot is position or torque-controlled, it is necessary that any bound is reached with feasible velocity and acceleration drops. The early work tackling this topic is presented in [8]. A velocity linear damper is formulated to prevent collisions of robotic manipulator links with the environment obstacles. This formulation has been applied at the joints level in [9] and implemented in terms of acceleration in [10].

In order to avoid unfeasible decelerations near the distance bounds, a velocity constraint is formulated at the task-space level assuming constant deceleration in [11]; which was improved and applied at the joint level in [12]. This approach is extended to the joint acceleration level in [13], where an algorithm is proposed to account for joint jerk constraint. In [14]–[17], Taylor expansion with a step-size taken as a multiple of the control time-step, is used to formulate joints position and velocity constraints in the acceleration space.

In [18], joints position bounds are treated outside the QP control constraints-set through an explicit deceleration to push back a joint if it gets closer to its bounds. In [19], a parametrization of the feasible joint-space prevents the joints trajectories from reaching the bound limits.

In [20], a position-dependent velocity constraint is proposed to avoid QP failure when the humanoid robot knees are fully stretched while walking. The same formulation is given in [21] for redundant robotic manipulators. This constraint is also known as *viability constraint* in [22], where a discrete-time implementation in terms of joints acceleration is provided. All these approaches have the following shortcomings (explicitly acknowledged by their authors):

- A precise knowledge of the robot state is necessary and hence assumed;
- Undesirable discontinuous motions (chattering) near the constraints bounds in closed-loop;
- Mostly applied for joints bounds and few in collision avoidance;
- They do not handle time-varying bounds.

We propose a general formulation for distance and velocity limit constraints expressed in the acceleration space, that apply for a wider class of constraints. Our method is based on Ordinary Differential Inequality (ODI) solutions [23]. It does not require precise knowledge of the robot state (i.e. it accounts for uncertainties and noise). Our approach generates closed-loop smooth motion *near* and *at* the constraints bounds. It also allows having time-changing bounds. In

[1]M. Djeha and A. Kheddar are with the CNRS-University of Montpellier LIRMM Interactive Digital Humans group, Montpellier, France. mohamed.djeha@lirmm.fr

[2]A. Kheddar and A. Tanguy are with the CNRS-AIST Joint Robotics Laboratory, IRL, Tsukuba, Japan.

[1]The control community wouldn't term these QP computations as controllers since they generate the desired robot state. We claim that having the QP updated with the state feedback from the robot makes it a controller. Moreover, one can also enhance the model-based QP with actuator models such that it outputs the control signals.

Section II, we give an overview of the QP multi-task control formulation. In Section III, we investigate and explicit the drawbacks of two state-of-the-art methods [10] and [22]. Then, we provide the details of the proposed formulation, based on an adaptive-gains method. Finally, Section IV assesses our approach with humanoid HRP-4 experiments.

The following notations are adopted throughout the paper:

- $t \in \mathbb{R}^+$ denotes time, in seconds;
- dt denotes the control time-step, in seconds;
- $\epsilon(idt) \triangleq \epsilon_i$, $i \in \mathbb{N}$;
- superscript * denotes a *desired* entity;
- ${}^g x^T = \begin{bmatrix} e^T & \dot{e}^T \end{bmatrix}$, ${}^c x^T = \begin{bmatrix} h^T & \dot{h}^T \end{bmatrix}$, ${}^r x^T = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}$ denote the task-state, constraint-state and robot-state vectors, respectively.

II. Quadratic-Programming Control Formulation

A. Equation of Motion

The Equation of Motion (EoM) for a robot with a floating-base is given by:

$$\begin{bmatrix} M(\hat{q}) & -S & -J_{\text{tr}}^T \end{bmatrix} \chi + N(\hat{q}, v) = 0 \tag{1}$$

where $\chi^T = \begin{bmatrix} \dot{v}^T & \tau^T & f^T \end{bmatrix}$, $(\hat{q}, v)$ is the full state of the system including the floating-base. $\hat{q}^T = \begin{bmatrix} q^T & \xi^T \end{bmatrix}$ stacks the actuated joints vector q and the root $SE(3)$ of the robot, and $v^T = [\dot{q}^T \ \nu^T]$ where $\dot{q} = dq/dt$ and ν is the robot base frame linear and angular velocities. $M(\hat{q})$ is the inertia matrix. $N(\hat{q}, v)$ gathers Coriolis-centrifugal and gravitational terms. S is a diagonal selection matrix for actuated joints. f is the external contact forces and J_{tr} the translational Jacobians at contact points. Each non-sliding contact force f is constrained to remain strictly within the linearized friction cone $\mathcal{K}$ mapped by a finite set of generator vectors $\{\rho_j\}_{j=1,\cdots,\eta}$ [3]:

$$f \leq \sum_{j=1}^{\eta} \beta_j \rho_j, \ \beta_j \geq 0 \tag{2}$$

This robot motion shall be restricted by a set of limitations:

- torque range bounds:

$$\tau_{\min} \leq \tau \leq \tau_{\max} \tag{3}$$

- a set of constraints of the form:

$$H(q, \dot{q}) \leq 0 \tag{4}$$

where $H(q, \dot{q})$ can express different constraints, such as:

- joint position bounds:

$$q_{\min} \leq q \leq q_{\max} \tag{5}$$

- joint velocity bounds:

$$\dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max} \tag{6}$$

- center-of-mass $\text{CoM}(q)$ equilibrium [24] or any other constraint (eventually implying CoM velocity):

$$\text{CoM}(q) \in \mathcal{P} \ \text{ or } \ \bigcup \text{CoM}(q) \leq \Delta \tag{7}$$

  where $\mathcal{P}$ is a given polyhedron that can be rewritten into a set of inequalities; Δ being the given lower distance-bound vector.
- distance constraint $\delta(q)$ for none-desired collisions [10]:

$$\delta(q) \geq \delta_{\min} \tag{8}$$

- field-of-view $\text{FoV}(q)$ inclusion constraints in visual servoing or vision-based object tracking. One needs to keep the object of interest $\mathcal{O}$ within the field-of-view of the robot embedded camera, and out of any known possible occlusions. These are linked to the robot configuration:

$$\text{FoV}(q) \ni \mathcal{O} \tag{9}$$

  the inclusion is also written equivalently into a set of inequalities, see e.g. [25].

There are many other constraints that can be expressed in the general form of (4). Notice that, except constraints (1), (2) and (3), the QP decision-variables vector χ does not appear explicitly in constraints like (4). Hence, they cannot be integrated directly into the QP controller. When possible, one needs to rewrite such constraints in the form:

$$A(q, \dot{q})\chi + b(q, \dot{q}) \leq 0 \tag{10}$$

where A is a matrix and b a vector of appropriate dimensions[2]. In this work, we will show how to formulate constraints like (4) to be written as (10) resulting in a stable and smooth motion in the context of closed-loop QP control.

B. Task-Space QP Formulation

Let g and g^{ref} be the current and reference tasks respectively. Let us define $e = g - g^{\text{ref}}$, $\dot{e} = \dot{g} - \dot{g}^{\text{ref}}$ and $\ddot{e} = \ddot{g} - \ddot{g}^{\text{ref}}$ where $\dot{e} = J_g \dot{q}$, $\ddot{e} = J_g \ddot{q} + \dot{J}_g \dot{q}$, and J_g is the task Jacobian. The continuous task state-space representation is given by the following canonical form:

$${}^g\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} {}^g x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{11}$$

A classical way to stabilize the system (11) is to choose a task-state linear feedback control-law:

$$u = -\begin{bmatrix} P & D \end{bmatrix} \ {}^g x \tag{12}$$

where P and D are diagonal matrices of appropriate dimensions denoting the proportional and derivative feedback gains. Hence, the system (11) takes the closed-loop form:

$${}^g\dot{x} = \Gamma \, {}^g x, \ \Gamma = \begin{bmatrix} 0 & I \\ -P & -D \end{bmatrix} \tag{13}$$

where P and D are chosen so that Γ is Hurwitz [26], [27]. The goal is to have $\ddot{e} = u$ under explicit constraints (1), (10). This is handled by means of a QP minimization:

$$\begin{aligned} &\min_{\chi} \frac{1}{2} ||\ddot{e} - u||^2 \\ &\text{subject to: (1) (10)} \end{aligned} \tag{14}$$

[2]Constraints (2) and (3) can also be directly represented by formulation (10) since force and torque are part of the decision-variables vector χ.

If there is more than one task to achieve, the tasks need to be ordered either by means of strict [28], or soft [3], [14] hierarchy prioritization.

At each control step, the closed-loop QP (14) is built online, where the cost-function and the constraints are updated based on the current measured/estimated robot state ${}^{r}x$.

III. Constraints Formulation in Closed Loop Based on Feedback Adaptive Gains

For the sake of comparison with previous works (that didn't address a large panel of constraints), and without loss of generality, the constraints (4) are in practice either as a distance constraint:

$$h(q) \geq 0 \tag{15}$$

or as a velocity constraint:

$$\dot{h}(q, \dot{q}) \geq 0 \tag{16}$$

For the sake of clarity, constraints (15) and (16) are noted h and $\dot{h}$ respectively. To transform (15), (16) into the form (10), most of the existing approaches assume exact:

$$h_{i+1} = h_i + \dot{h}_i dt + \frac{1}{2}\ddot{h}_i dt^2 \tag{17a}$$

$$\dot{h}_{i+1} = \dot{h}_i + \ddot{h}_i dt \tag{17b}$$

where $\dot{h} = J_h \dot{q}$, $\ddot{h} = J_h \ddot{q} + \dot{J}_h \dot{q}$, J_h being the constraint Jacobian. To ensure that (15) will be fullfiled, the velocity $\dot{h}$ should be enforced to decrease until stop when $h = 0$.

In what follows, we review two state-of-the-art methods: velocity linear damper [10] and viability formulation [22]. There are two reasons for this choice: besides the fact that [22] is the most up-to-date work in this topic, both of [10] and [22] resolved the task-space control in the acceleration space (in contrast to the kinematics task-space control in [11], [12]).

A. Related Works

1) Velocity Linear Damper: enforces $\dot{h}$ to decrease linearly w.r.t h given the following formula [8]–[10]:

$$\dot{h}_{i+1} + \alpha h_i \geq 0 \tag{18}$$

where $\alpha > 0$ is a tuning parameter. Based on (17b), the second order form of (18) is given as (see e.g. [10]):

$$\ddot{h}_i + K_v \dot{h}_i + K_p h_i \geq 0 \tag{19}$$

resulting in a 2nd order linear differential inequality, where the stiffness K_p and damping K_v gains are given by: $K_p = \frac{\alpha}{dt}$, and $K_v = \frac{1}{dt}$. The gains tuning of this implementation is limited to α since the damping gain is constant.

2) Viability Formulation: this formulation is based on the assumption of a constant deceleration $\ddot{h}_{\text{const}}$, such that:

$$\dot{h}_{i+1} + \sqrt{\alpha h_{i+1}} \geq 0 \tag{20}$$

where $\alpha = 2\ddot{h}_{\text{const}}$. Considering (17a) and (17b), the implementation proposed in [22] is based on a switching logic that guarantees a constant deceleration until stop at $h = 0$. The resulting second-order differential inequality is nonlinear and complex, but it can be further simplified to:

$$\text{switching logic} \rightarrow \begin{cases} \ddot{h}_i + \frac{1}{dt}\dot{h}_i \geq 0 \\ \text{or} \\ \ddot{h}_i + \frac{1}{dt}\dot{h}_i + \frac{1}{dt}\sqrt{\alpha h_i} \geq 0 \end{cases} \tag{21}$$

If $\ddot{h}_{\text{const}}$ can be well estimated for the joint constraints [22], it is difficult and non-trivial to estimate if one attempts to extend this formulation to other constraints such as collision avoidance, CoM constraints, etc.

Yet, neither velocity linear damper nor viability formulation methods have been applied in closed-loop, or considered with varying bounds. In fact, in closed-loop control, ${}^{c}x$ is computed from ${}^{r}x$ obtained from the robot real state, and is subject to uncertainties and sensor noises. Hence, (17a) and (17b) become invalid because noises and estimation uncertainties are not taken into account. Moreover, the previous methods assume constant bounds on distance and velocity, which is explicitly stated in the conclusion section in [22].

Hereafter, we propose a second order ODI formulation of (15) and (16) that overcomes those shortcomings.

B. Second Order ODI Formulation of Constraints

Although (15) is a distance constraint, it induces implicitly a coupling with the velocity since it requires that the latter must drop to zero when the bound is reached, i.e. $h = 0$. An alternative to handle this coupling in the acceleration space is by means of a second order ODI.

Let us formulate (15) and (16) as follows:

$$(15) \xrightarrow[\text{as } 2^{\text{nd}} \text{ order ODI}]{\text{formulated}} \ddot{h}_i + K_v \dot{h}_i + K_p h_i \geq 0 \tag{22}$$

$$(16) \xrightarrow[\text{as } 2^{\text{nd}} \text{ order ODI}]{\text{formulated}} \ddot{h}_i + \gamma \dot{h}_i \geq 0 \tag{23}$$

where γ, K_v, and K_p are in $\mathbb{R}^+$. γ is used to tune the first order dynamics in terms of velocity $\dot{h}$ (23). However, the challenge here is how to tune the stiffness and damping gains K_p and K_v, respectively, such that (i) $h(t)$, the solution of (22), enforces (15), and (ii) $\dot{h}(t)$ drops to zero if $h = 0$. First, we show how ODI (22) is solved. Then, since ad-hoc gains-tuning is inappropriate for a generic approach, we present a systematic method to compute the gains. All the formulas are considered at time $t = idt$, and thereby the subscript $_i$ is dismissed.

1) Mathematical Formulation: let us consider the following second-order Ordinary Differential Equation (ODE):

$$\ddot{y} + K_v \dot{y} + K_p y = 0 \tag{24}$$

which is the saturated ODI (22) (i.e. $\geq$ is replaced by $=$). ODE (24) can be written as a system of two first-order ODE as follows[3]:

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = K \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \; K \in \mathbb{R}^{2\times 2} \tag{25}$$

[3]To explain the proposed approach, the distance constraint is considered as a scalar $h, y \in \mathbb{R}$. The generality is not lost since the constraints, that are often posed as vectors (e.g. joints constraints), are a stack of multiple scalar distance-constraints.

where K is the constraint-state feedback matrix. Thanks to Petrovitch theorem [23], it is shown that given (22) and (24), under initial conditions $h(t_0)=y(t_0)$, $\dot{h}(t_0)=\dot{y}(t_0)$ for $t=t_0$, we get: $h(t)\geq y(t),\ \forall t\geq t_0$. This theorem states that the ODE solution $y(t)$ is the lower bound for all the ODI solutions $h(t)$. The idea is to enforce $y(t)$ to satisfy $y(t)\geq 0$, and $\dot{y}(t)=0$ if $y(t)=0$. Hence, if $y(t)\geq 0 \Rightarrow h(t)\geq 0$. Thus, we will focus on the solution $y(t)$.

Given $K_v=2\xi\sqrt{K_p}$ where ξ is the damping coefficient to be set, the eigenvalues $\lambda_{1,2}$ of K are computed such that:

$$(K-\lambda I)\mu=0 \Rightarrow \lambda_{1,2}=-\frac{1}{2}K_v(1\mp\sqrt{1-\xi^{-2}}),\ \xi\geq 1 \quad (26)$$

where I is identity matrix, $\mu\in\mathbb{R}^2$ are the eigenvectors corresponding to the eigenvalues $\lambda\in\mathbb{R}$ such that $\lambda_{1,2}<0,\ \lambda_1\geq\lambda_2$ so that K is Hurwitz, and $\mu_{1,2}^T=\begin{bmatrix}1 & \lambda_{1,2}\end{bmatrix}$.

For computation efficiency, (22) is inserted into the constraints set of (14) only if $h\leq h_d$, where h_d is a predefined safety margin. Also, we denote by *initial state* $(y_{t_0},\dot{y}_{t_0})=(h_{t_0},\dot{h}_{t_0})$ at $t=t_0$, the corresponding constraint state.

Given an initial state, the phase trajectory $(y,\dot{y})$ converges asymptotically to the origin $(0,0)$ if (26) holds (K Hurwitz) [26]. This convergence property, due to the constraint-state feedback, allows to deal with time-varying bounds. However, the phase trajectory convergence does not depend solely on the eigenvalues. It depends also on the initial state: we can have different phase trajectories shapes for different initial states; yet for the same eigenvalues (see Fig. 1). In the worst case, the phase trajectory overshoots $y=0$, losing thereby any guarantee that (15) holds. This case occurs if the initial state is in the blue area in Fig. 1. Let us call this area the *unviable region* in contrast to the *viable region*, denoted in green, where $y(t)\geq 0,\ \forall t\geq t_0$. Hence, it is not trivial to find the suitable gains, since the constraint (15) can become active in many initial states. In what follows, we propose a systematic method to overcome this issue.

C. Gains Adaptation Method

In order to avoid overshooting, our solution is to adapt the gains to change the shape of the viable region so that the initial state is included. Hence, the phase trajectory is enforced to remain within the viable region, so that the convergence is achieved such that $\lim_{t\to+\infty} y(t)=0^+$. In fact, if $\lim_{t\to+\infty} y(t)=0^-$, it means that the phase trajectory overshoots. Given an initial state and (26) for $\xi>1$, (24) solution is:

$$y(t)=y_{t_0}\left(\frac{\lambda_2-\frac{\dot{y}_{t_0}}{y_{t_0}}}{\lambda_2-\lambda_1}\exp^{\lambda_1(t-t_0)}+\frac{\frac{\dot{y}_{t_0}}{y_{t_0}}-\lambda_1}{\lambda_2-\lambda_1}\exp^{\lambda_2(t-t_0)}\right)$$

The limit $\lim_{t\to+\infty} y(t)$ follows the first term dynamics since λ_1 is the dominant pole (the second term vanishes faster to 0):

$$\lim_{t\to+\infty} y(t)=\lim_{t\to+\infty} y_{t_0}\frac{\lambda_2-\frac{\dot{y}_{t_0}}{y_{t_0}}}{\lambda_2-\lambda_1}\exp^{\lambda_1(t-t_0)}=0^+$$
$$\Rightarrow\frac{\dot{y}_{t_0}}{y_{t_0}}=\frac{\dot{h}_{t_0}}{h_{t_0}}\geq\lambda_2,\ (h_{t_0}>0,\ \dot{h}_{t_0}\leq 0)$$

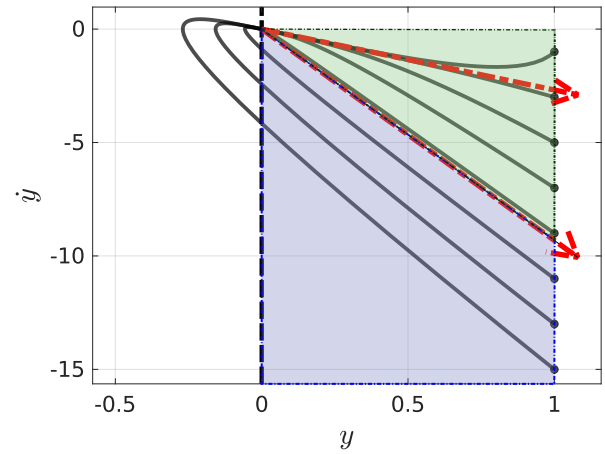

Fig. 1. Phase trajectories (in black) denoting (24)'s solutions depending on the initial states (dots at $y=1$), for the same gains. The corresponding eigenvectors are in red dashed arrows. The green zone represents the viable region whereas the blue zone denotes the unviable region. Note that these regions depend on the gains K_p and K_v.

The same result is obtained for $\xi=1$ ($\lambda_1=\lambda_2$):

$$y(t)=y_{t_0}\left(1+t(\frac{\dot{y}_{t_0}}{y_{t_0}}-\lambda_1)\right)\exp^{\lambda_1(t-t_0)}$$
$$\lim_{t\to+\infty} y(t)=\lim_{t\to+\infty} y_{t_0}\left(1+t(\frac{\dot{y}_{t_0}}{y_{t_0}}-\lambda_1)\right)\exp^{\lambda_1(t-t_0)}=0^+$$
$$\Rightarrow\frac{\dot{y}_{t_0}}{y_{t_0}}=\frac{\dot{h}_{t_0}}{h_{t_0}}\geq\lambda_1,\ (h_{t_0}>0,\ \dot{h}_{t_0}\leq 0)$$

In both case, the following formulas should be satisfied:

$$\frac{\dot{h}_{t_0}}{h_{t_0}}=\lambda_2\Rightarrow\begin{cases}K_v=-\frac{2(\dot{h}_{t_0}/h_{t_0})}{1+\sqrt{1-\xi^{-2}}}\geq 0\\ K_p=(\frac{K_v}{2\xi})^2\end{cases} \quad (27)$$

The gains are computed once, at each time the constraint is activated. The graphical interpretation of this result is to bring μ_2, being the borderline between the viable and unviable regions (see Fig. 1), to pass by the initial state. This is achieved by making μ_2 slope λ_2 equal to $\dot{h}_{t_0}/h_{t_0}$. The main advantage of this method is to be adaptive to any initial state (see Fig. 2). In fact, the latter depends on the dynamics resulting from the tasks and the current active constraints. One may choose high K_p and K_v gains which will result in a larger viable region. However, it is not recommended to use high gains in closed-loop, as this results in vibrations and chattering. In order to improve the damping, we propose nonlinear-gains $K_p=f_p(h)$ and $K_v=f_v(h)$ as follows.

D. Nonlinear-Gains Based Feedback

Consider a deformable mechanism having a set of parallel layers of identical springs and dampers, Fig. 3. These layers are shifted from each other by an infinitesimal distance $d\theta$. This model is equivalent to a nonlinear spring-damper system for which the equivalent stiffness and damping gains are K_p^{eq} and K_v^{eq}, respectively. Let $K_p^{\text{eq}}(h)=K_p+K_p^{\text{disp}}(h)$ where K_p is obtained from (27), and K_p^{disp} is the varying term due

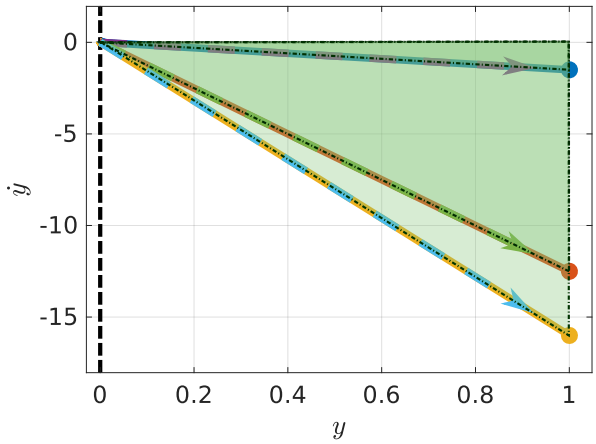

Fig. 2. Phase trajectories (in solid) are (24) solutions with our adaptive gains method. The viable region changes shape according to the initial state (dots at $y = 1$). The eigenvectors (dashed arrows) are delimiting the viable regions. The unviable region is not shown.

to the displacement. We get: $K_p^{\text{disp}}(h) = -\int_{h_d}^{h} K_p \varphi_p(\theta) d\theta$, where $\varphi_p(\theta)$ is the springs density per displacement $d\theta$. If $\varphi_p(\theta) = \varphi_p$ constant, we get: $K_p^{\text{disp}}(h) = K_p \varphi_p (h_d - h)$, and finally:

$$K_p^{\text{eq}}(h) = K_p[1 + \varphi_p(h_d - h)] \tag{28}$$

Similarly, we have: $K_v^{\text{eq}}(h) = K_v + K_v^{\text{disp}}(h)$. Following the same steps to obtain (28), we get:

$$K_v^{\text{eq}}(h) = K_v[1 + \varphi_v(h_d - h)] \tag{29}$$

where φ_v is dampers constant density per displacement $d\theta$. From (28) and (29), ξ^{eq} is obtained as:

$$\xi^{\text{eq}}(h) = \frac{K_v^{\text{eq}}(h)}{2\sqrt{K_p^{\text{eq}}(h)}} = \xi \frac{1 + \varphi_v(h_d - h)}{\sqrt{1 + \varphi_p(h_d - h)}} \tag{30}$$

$\varphi_v = \varphi_p = \varphi$ yields $\xi^{\text{eq}}(h) = \xi\sqrt{1 + \varphi(h_d - h)}$. The resulting gains (28) and (29) increase linearly while the phase trajectory is converging to the origin. Besides, if $\xi = 1$ the constraint dynamics evolves from critically damped ($\xi_{\text{eq}}(h = h_d) = \xi = 1$) to overdampings ($\xi_{\text{eq}}(h < h_d) > 1$): the damping is enforced and the velocity $\dot{h}$ is effectively reduced.

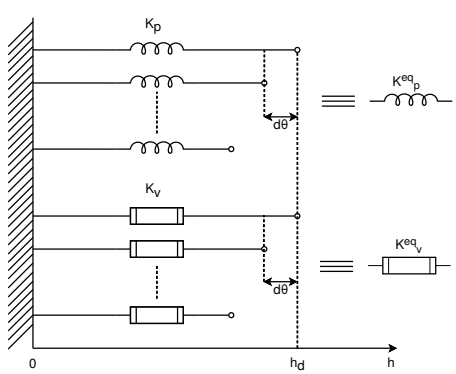

Fig. 3. Nonlinear spring-damper model.

IV. Experimental Results

We validate our approach using position-controlled humanoid robot HRP-4. The `mc_rtc`[4] framework is used to control the robot. The control loop runs at a frequency of 200 Hz. At each control cycle, the joints positions are provided by the encoders and the joints velocities are estimated numerically. In return, `mc_rtc` provides the desired joints positions q^* after double integrating $\ddot{q}^*$ solution of (14). All experimental scenarios are performed in closed-loop with set-point tasks assigned such that different constraints reach their bounds. First, we compare the state-of-the-art [10], [22] to our approach for collision avoidance constraints. Then, we extend and test our approach for other classes of constraints like joint bounds, CoM and velocity constraints. Excerpts video of the experiments is attached as multimedia material.

In these experiments, contacts between the robot feet and the ground are defined as equality constraints, and formulated also as a second order ODE:

$$\dot{X}_{\text{foot}^k/\text{ground}} = 0 \xrightarrow[\text{as } 2^{\text{nd}} \text{ order ODE}]{\text{formulated}} \ddot{X}_{\text{foot}^k/\text{ground}} + \gamma \dot{X}_{\text{foot}^k/\text{ground}} = 0$$

where $\dot{X}_{\text{foot}^k/\text{ground}}$ is the relative velocity between the foot-k ($k = \{1, 2\}$) and the ground, and $\gamma = 10$.

A. Collision Avoidance

To compare our method to [10], [22] we extend the viability formulation constraint –originally formulated solely for joints bounds in [22], to collision avoidance constraint: $h = \delta - \delta_{\min} \geq 0$, where δ is the distance between a pair of bodies. $\ddot{h}_{\text{const}}$ is computed such that $\ddot{h}_{\text{const}} = \frac{\dot{h}_{t_0}^2}{2h_d}$. The collision avoidance constraint is defined between the robot waist, and the right and left elbows, respectively (see Fig. 4). We fix the minimal allowed distance between each pair of bodies to $\delta_{\min} = 6\ cm$ with a safety margin $h_d = 4\ cm$. The damping coefficient and the springs and dampers density are fixed to the best ones: $\xi = 1.2$ and $\varphi_p = \varphi_v = \frac{0.5}{h_d}$. Starting by an initial posture where the shoulders are fully stretched, we assign joints position targets for both shoulders so that the elbows get to the waist. The results are shown in Fig. 5. The methods in [10], [22] lack robustness against noises and un-modeled nonlinearities encountered in closed-loop: jerky motions and chattering especially near $\delta_{\min}$ are noticed. In contrast, our approach exhibits a smooth motion, and the velocity decreases until stop when $\delta = \delta_{\min}$.

Figure 6 shows how our approach deals with variable distance bounds that are assumed to be constant in [22]. At $t = 0$, the constraint is violated $h < 0$. The QP solver does not fail, and δ converges to $\delta_{\min}$ if $\delta_{\min} \geq \delta_{\text{target}}$.

B. Joints Bounds and CoM Constraints

To validate our approach on the CoM constraint, we define a conservative polygon of support for the CoM [24]. Next, we define end-effector unreachable set-point target for the right hand so as during the reaching motion, the CoM reaches the polygon bounds. Different targets are defined, see Fig. 7.

[4] `https://jrl-umi3218.github.io/mc_rtc/index.html`

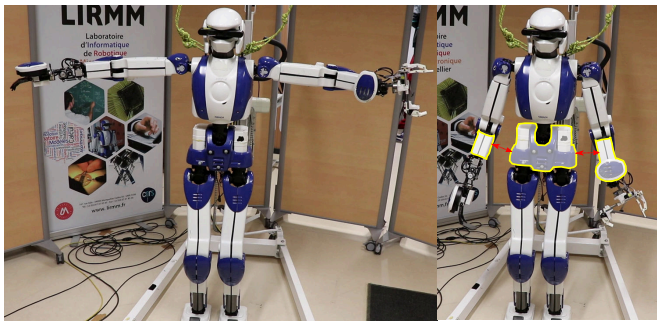

Fig. 4. Initial posture (left) and the pairs of bodies (right) concerned by the collision avoidance constraint and highlighted with yellow contour.

Then, we fix the right hand target, and move the left arm to reach the left shoulder roll and pitch joints limits; all constraints together. The robot floating-base is estimated using a Kinematic Inertial Observer based on the IMU measurements and the the robot kinematics chains [29]. Figure 8 shows the temporal and spatial evolutions of the estimated CoM in addition to the different time-frame where either the right or left arm is moving. We notice that the CoM is constrained inside the polygon except in the yellow-highlighted spots representing overshoots. The latter have a magnitude of $2\ mm$ at maximum and occur very briefly when the end effector task is assigned. This is due to the un-modeled flexibilities at the ankles which act as a perturbation against the constraint. However, these perturbations are damped and the constraint is stabilized at the boundary. Moreover, despite the CoM noisy-estimation, no jerky motions or chattering have been observed near or at the bounds.

Figure 9 shows how the robot adjusts its whole-body posture when the left arm is fully stretched in different directions to maintain the CoM at the polygon boundary (highlighted with the light green area in Fig. 8). The left-shoulder roll and pitch joints-positions bounds are reached smoothly with zero velocity as it is shown in Fig. 10.

C. Velocity Constraint

To validate our approach for velocity constraint formulation, we bound the relative velocity between the pair of bodies shown in Fig. 4. As in Section IV-A, variable velocity bounds are considered. The gain γ is fixed to $\gamma = 10$. Figure 11 shows how the velocity converges to the bound when the latter is variable. Again, despite the velocity noisy-estimation, no chattering or jerky motions have been observed during the experiment. In [10], [22], based on (17b), γ is taken as $\gamma = \frac{1}{dt} = 200$ which is a relatively high gain that could be critical in closed-loop. Indeed, with such gain, the QP solver failed to find a solution when the velocity approached the bound of $0.05\ m/s$.

V. Conclusion

In this letter, we propose a solution to deal efficiently with a class of constraints commonly found in task-space QP controllers based on acceleration, force and torque as decision variables. More particularly, we address distance and velocity constraints that originally are not written in terms of the decision control variables. First, we show that [10], [22] approaches have shortcomings when the constraints reach their bounds. The main limitations acknowledged by their authors are: (i) the need of precise models (no noise and estimation uncertainties); (ii) they do not operate well in closed-loop scheme, i.e. when the robot state is directly fed back to the QP controller (chattering), which is the essence of anything called control; and (iii) they do not handle variable bounds. We first show that the reason of these limitations is the 'gains' parametrization and subsequent tuning. Then, we propose a formulation for such a class of constraints at large based on ODI, with an automated adaptive gain tuning to overcome all the previously mentioned problems. Experiments conducted on the HRP-4 humanoid robot confirmed performance and efficiency of our approach, which is then integrated as part of our `mc_rtc` controller library.

Nevertheless, there remain some limitations that need further investigations. First, we assume no constraints on the feasible accelerations $\ddot{h}$. That is to say, if the constraint-set includes implicit or explicit constraint such that $\ddot{h} \leq \ddot{h}_{\max}$, it may result in *constraints incompatibility* as in [11], [13], [22], [30]; namely when the necessary amount of acceleration to enforce (15) is greater than $\ddot{h}_{\max}$. This may occur either if $\dot{h}_{t_0}$ is high or h_d is low. Currently, we are working on how to handle constraints compatibility within the proposed approach. At the same time (because this is somehow related), we aim at solving few shortcomings due to the feedback on the entire controller part, i.e. considering the cost part of the tasks with more stability, robustness and performances, together with a systematic setting of the tasks' gains.

References

[1] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[3] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, pp. 64–77, Feb 2019.

[4] J. Englsberger, G. Mesesan, A. Werner, and C. Ott, "Torque-based dynamic walking-a long way from simulation to experiment," in *IEEE International Conference on Robotics and Automation*, pp. 440–447, 2018.

[5] Y. Wang and A. Kheddar, "Impact-friendly robust control design with task-space quadratic optimization," in *Robotics: Science and Systems*, June 2019.

[6] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Englsberger, M. A. Roa, P. Wieber, F. Chaumette, F. Spindler, G. Oriolo, L. Lanari, A. Escande, K. Chappellet, F. Kanehiro, and P. Rabaté, "Humanoid robots in aircraft manufacturing: The airbus use cases," *IEEE Robotics Automation Magazine*, vol. 26, pp. 30–45, Dec 2019.

[7] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt, "Team ihmc's lessons learned from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.

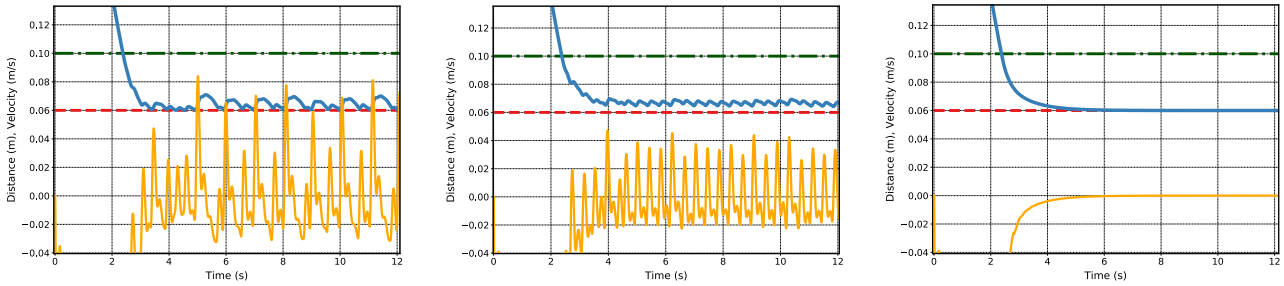

Fig. 5. Three methods performed in closed-loop: viability formulation [22] (left), velocity linear damper [10] (center) and our approach (right). $\delta_{\min}$ (in dashed red) is the lower bound of the distance between a pair of bodies (blue). The chattering is visible in the velocity (orange) for [22] and [10] methods, whereas it converges to zero smoothly in our approach. The safety margin is the space between the dash-dotted green and dashed red.

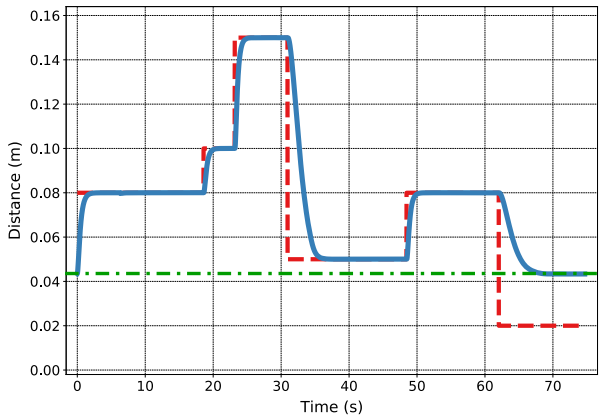

Fig. 6. Collision avoidance constraint with variable $\delta_{\min}$ (red dashed). The distance δ (blue) converges to $\delta_{\min}$ if $\delta_{\min} \geq \delta_{\text{target}}$ (green dash-dotted).

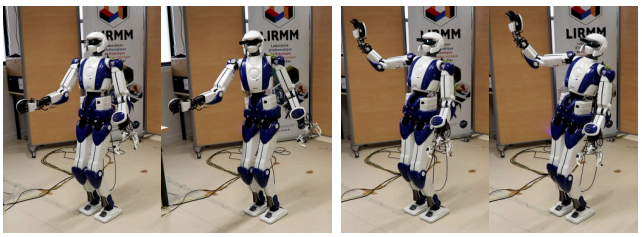

Fig. 7. HRP-4 trying to reach a far right hand set-point target to make the CoM hit the equilibrium polygon boundaries during motion.

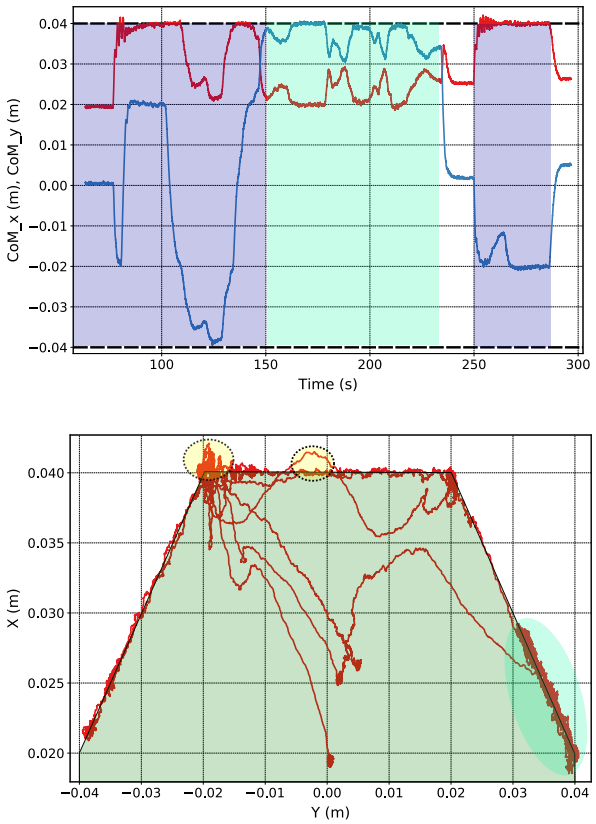

Fig. 8. CoM evolutions in time (top) and space (bottom). The dark green area denotes the CoM equilibrium polygon. The blue and light green zones highlight the periods in time as well as in space where the CoM moves due to the right hand end-effector task or the left arm movements, respectively.

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

[9] F. Kanehiro, M. Morisawa, W. Suleiman, K. Kaneko, and E. Yoshida, "Integrating geometric constraints into reactive leg motion generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4069–4076, 2010.

[10] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, *et al.*, "Multi-contact vertical ladder climbing with an hrp-2 humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 561–580, 2016.

[11] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, "Extending itasc to support inequality constraints and non-instantaneous task specification," in *IEEE International Conference on Robotics and Automation*, pp. 964–971, 2009.

[12] 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*, pp. 677–684, 2010.

[13] A. Meguenani, *Safe control of robotic manipulators in dynamic contexts*. Phd thesis, Université Pierre et Marie Curie - Paris VI, Oct. 2017.

[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*, pp. 1283–1290, 2011.

[15] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid con-

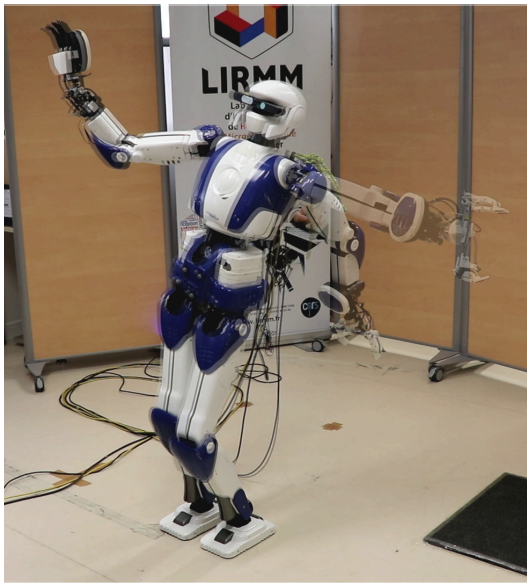

Fig. 9. Superposition of two pictures when the left arm is stretched in two different directions. The blur of the whole body shows how the robot adjusts its posture to keep the CoM at the boundary of the equilibrium polygon (light green area at the bottom in Fig. 8).

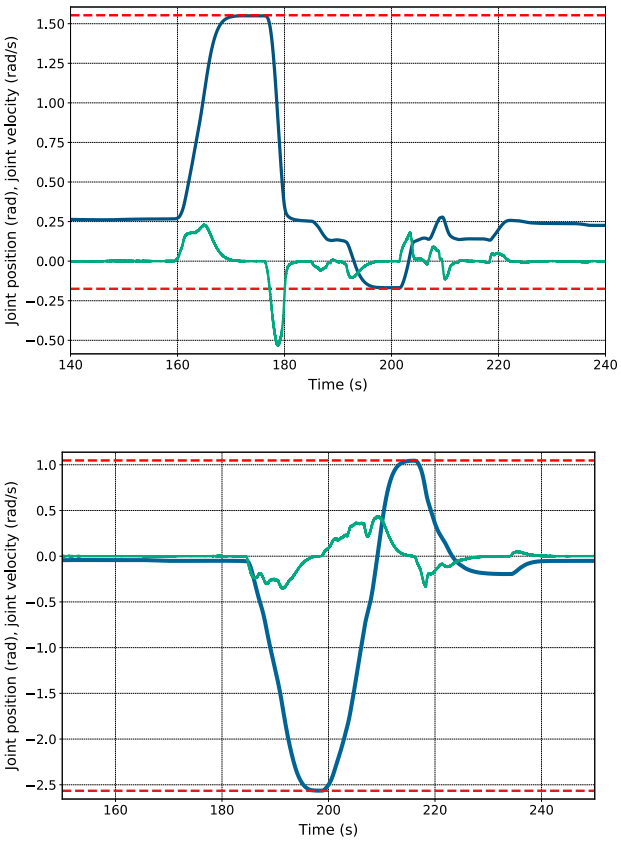

Fig. 10. Left shoulder roll (top) and pitch (bottom) joints positions (blue) and velocities (green). The upper and lower joints bounds are in red dashed.

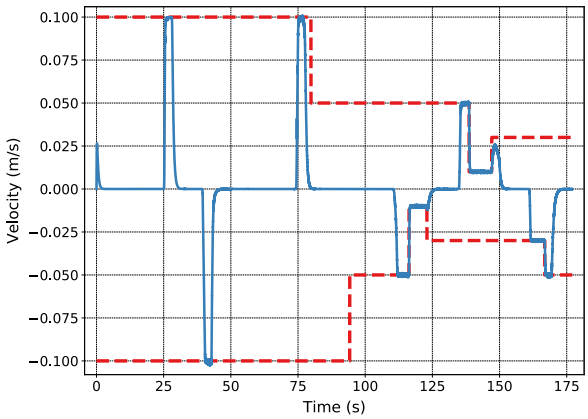

Fig. 11. Velocity (blue) evolution with variable bounds (red dashed).

tacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[16] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," in *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 444–450, Nov 2014.

[17] R. Lober, *Task compatibility and feasibility maximization for whole-body control*. Phd thesis, Université Pierre et Marie Curie - Paris VI, Nov. 2017.

[18] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson, "3d walking based on online optimization," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 21–27, 2013.

[19] M. Charbonneau, F. Nori, and D. Pucci, "On-line joint limit avoidance for torque controlled robots by joint space parametrization," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 899–904, Nov 2016.

[20] K. Hu, C. Ott, and D. Lee, "Online human walking imitation in task and joint space based on quadratic programming," in *IEEE International Conference on Robotics and Automation*, pp. 3458–3464, 2014.

[21] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.

[22] A. del Prete, "Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators," *IEEE Robotics and Automation Letters*, vol. 3, pp. 281–288, Jan 2018.

[23] A. F. Dragoslav S. Mitrinovic, J. Pecaric, *Inequalities Involving Functions and Their Integrals and Derivatives*. Mathematics and Its Applications 53, Springer Netherlands, 1 ed., 1991.

[24] H. Audren and A. Kheddar, "3-d robust stability polyhedron in multicontact," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 388–403, 2018.

[25] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, "Visual servoing in an optimization framework for the whole-body control of humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, pp. 608–615, April 2017.

[26] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[27] K. Bouyarmane and A. Kheddar, "On weight-prioritized multitask control of humanoid robots," *IEEE Transactions on Automatic Control*, vol. 63, pp. 1632–1647, June 2018.

[28] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[29] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in *IEEE International Conference on Robotics and Automation*, pp. 277–283, May 2019.

[30] S. Rubrecht, V. Padois, P. Bidaud, M. Broissia, and M. Da Silva Simoes, "Motion safety and constraints compatibility for multibody robots," *Auton. Robots*, vol. 32, p. 333–349, Apr. 2012.