



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-02490195v1

HAL Id: hal-02490195

<https://hal.science/hal-02490195v1>

Preprint submitted on 26 Apr 2020 (v1), 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 Djeha¹, Arnaud Tanguy², and Abderrahmane Kheddar³

Abstract—We revisit an open problem of a class of constraints obtained through derivation in task-space control frameworks formulated as quadratic programs. It is common that weighted or strict-hierarchy task-space formulation result in a set of constraints to fulfill strictly and others to meet at best. In most implementation implying dynamics, the decision variables are: robot’s joints acceleration, interaction forces (mostly physical contacts), and robot torques (that can be eliminated if torque bounds are known). However many constraints, e.g. joint constraints, do not write straightforwardly in terms of one of these decision variables. Previous work proposed solutions to write and enforce joint constraints. Yet, none of them worked properly in a closed-loop formulation of the QP when bounds are reached or when bounds change. In this letter, we show that joint constraints are part of a more general class of constraints such as collision avoidance, bounds of center of mass, constraints on field-of-view, Cartesian bounds on a given link, etc. We propose a general solution not only to enforce such a class of constraints at their bounds but also eliminating the chattering observed in all the existing methods; and more importantly, a systematic way to set the gains that allow stable behavior when bounds are reached in closed-loop.

I. INTRODUCTION

Task-space multi-objective and multi-sensory control by means of soft or strict hierarchy quadratic programming (QP) is demonstrated in state-of-the-art humanoid robots handling complex task objectives and behaviors. To cite a few examples: momentum control of Sarcos humanoid robot lower body [1]; stabilization of the planned footsteps trajectories for Atlas humanoid robot [2]; multi-robot control [3]; wrench distribution for walking stabilization [4], [5]; impact tasks awareness at contacts [6], [7]; and an integration in humanoid aircraft manufacturing [8].

In almost all existing QP formulated controllers, the decision variables, when dynamics is involved, 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. The motivation of this paper came first from an experimental observation. In the frame of aircraft robot manufacturing applications [8], we shall demonstrate the capability of humanoid robots (in our case, HRP-4) in accessing narrow and confined spaces. In one specific use-case experiments, our closed-loop QP controller [3] unexpectedly failed in the course of

the execution in some extreme configurations. The log data revealed that these 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 issues encountered in closed-loop control have been reported in [9].

How to write constraints, originally written in terms of position (or velocity), in terms of their derivative decision variables, is not new and several authors have proposed different formulations. It is important to reach any bound with feasible velocity and acceleration drops. The earlier work tackling this topic is presented in [10] where a velocity linear damper was formulated to prevent collision of a robotic manipulator links with the environment obstacles. This formulation has been applied at the joints level in [11], and implemented in term of acceleration in [12], [13]. In order to avoid unfeasible decelerations near the distance bounds, a velocity constraint is formulated at the task-space assuming a constant deceleration in [14]. This method has been improved and applied at the joint level in [15]. In [16] and [17], Taylor expansion is assumed to make use of the acceleration from joint position and velocity constraints where the expansion time step is taken to be a multiple of the control time step. In [18], joint position is not considered into the QP constraints. Instead, a deceleration is explicitly applied to push back a joint if it gets closer to its limit. In [19], a parametrization of the feasible joint-space has been proposed which should prevent the joints trajectories from collision with the bounds. However, this method suffers from singularities which makes it difficult to perform on real platforms. In [20], a position-dependent velocity constraint has been 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 term of joint acceleration is provided. All these approaches have the following shortcomings (explicitly acknowledged by the authors):

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

¹M. Djeha is with the CNRS-University of Montpellier LIRMM Interactive Digital Humans group, 161 rue Ada, 34095 Montpellier France mohamed.djeha@lirmm.fr

³A. Kheddar is also with the CNRS-AIST JRL UMI3218/RL, 1-1-1 Umezono, Tsukuba 305-8568, Japan kheddar@lirmm.fr

²A. Tanguy is with the AIST Intelligent Systems Research Institute, Humanoid Research Group, 1-1-1 Umezono, Tsukuba 305-8568, Japan. arnaud.tanguy@lirmm.fr

We propose a general formulation for distance and velocity constraints in term of acceleration that expand for a much larger (almost all) type of constraints. Our method is based on the Ordinary Differential Inequality (ODI) solutions [23]. It does not rely on any assumption about the robot state, and generates a smooth motion near the constraints limits. In Section II, we give an overview of the the QP multi-task control formulation. In Section III, we investigate and explicit the drawbacks of two state-of-the-art methods [12] and [22]. Then, we provide the details of the proposed formulation, based on an adaptive-gains method. Finally, Section IV assesses several constraints bound reaching experiments with the HRP-4 humanoid robot in closed-loop control.

The following notations are adopted throughout the paper:

- $t \in \mathbb{R}^+$ denotes time, in seconds;
- dt denotes the control time-step, in seconds;
- $\varepsilon(idt) \triangleq \varepsilon_i, i \in \mathbb{N}$;
- superscript $*$ denotes a *desired* entity;
- ${}^s x^T = [e^T \quad \dot{e}^T]$, ${}^c x^T = [h^T \quad \dot{h}^T]$, ${}^r x^T = [q^T \quad \dot{q}^T]$ 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) that describes a robot with a floating base is given by:

$$[M(\hat{q}) \quad -S \quad -J_{tr}^T] \chi + N(\hat{q}, v) = 0 \quad (1)$$

where: $\chi^T = [v^T \quad \tau^T \quad f^T]$, (\hat{q}, v) is the full state of the system including the floating base. $\hat{q}^T = \{q^T \quad \xi^T\}$ stacks the actuated joints vector q and the root $SE(3)$ of the robot, and $v^T = [\dot{q}^T \quad v^T]$ where $\dot{q} = dq/dt$ and v denotes linear and angular velocities of the robot base frame. $M(\hat{q})$ is the inertia matrix. $N(\hat{q}, v)$ encompasses the non linear terms (Coriolis-centrifugal and gravitational terms). S is a diagonal selection matrix for actuated joints. f denotes 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 K mapped by a finite set of generator vectors $\{\rho_j\}_{j=1, \dots, \mu}$ [24]:

$$f \leq \sum_{j=1}^{\mu} \beta_j \rho_j, \quad \beta_j \geq 0 \quad (2)$$

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

- torque range limits:

$$\tau_{\min} \leq \tau \leq \tau_{\max} \quad (3)$$

- a set of constraints of the form:

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

where $H(q, \dot{q})$ encompasses different types of constraints, for example:

- joint position constraint:

$$q_{\min} \leq q \leq q_{\max} \quad (5)$$

- joint velocity constraint:

$$\dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max} \quad (6)$$

- center-of-mass $\text{CoM}(q)$ equilibrium [25] or any other constraint (eventually implying CoM velocity):

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

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 $\delta(q)$ to avoid none-desired collisions [12]:

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

- field-of-view $\text{FoV}(q)$ inclusion constraints in visual servoing or vision-based object tracking. One need 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. There are linked to the robot configuration:

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

the inclusion also writes equivalently into a set of inequalities, see e.g. [26].

There are many other constraints that write in the form of (4) and that we do not mention here. Notice that the QP decision variables do not appear explicitly in such constraints. This is why their integration into the QP controller formulation cannot be made under this form. When possible, one needs to rewrite such constraints in the form:

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

where A is a matrix and b a vector of appropriate dimensions, and χ is the decision variable vector. In this work, we will show how to stabilize constraints like (10) that are originally written as (4) 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 task state-space representation is given by the following canonical form:

$${}^s \dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} {}^s x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (11)$$

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

$$u = -[P \quad D] {}^s x \quad (12)$$

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:

$${}^s \dot{x} = \Gamma {}^s x, \quad \Gamma = \begin{bmatrix} 0 & I \\ -P & -D \end{bmatrix} \quad (13)$$

where P and D are chosen so that Γ is Hurwitz [27], [28]. The goal is to have at best: $\ddot{e} = u$, while considering explicitly the constraints (10). This is handled by means of a QP minimization:

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

If there are more than one task to achieve, they need to be ordered either by means of strict [29], or soft [3], [16] hierarchy formulation.

At each control step, closed-loop QP (14) is built online, where the cost-function and the constraints are updated based on the current measured/estimated robot state ${}^r x$. Taylor expansion is widely used –e.g. [3], [13], [17], [22], [30], [31]; to describe ${}^r x_{i+1}^*$ given \ddot{q}_i^* , solution of (14):

$$\begin{aligned} q_{i+1}^* &= q_i^* + \dot{q}_i^* dt + \frac{1}{2} \ddot{q}_i^* dt^2 \\ \dot{q}_{i+1}^* &= \dot{q}_i^* + \ddot{q}_i^* dt \end{aligned} \quad (15)$$

where \ddot{q}_{i+1}^* is the input for the joint-level controllers in the case of position-controlled robots like the HRP humanoids series and most industrial robots at large.

III. CONSTRAINT IMPLEMENTATION 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:

$$h(q) \geq 0 \quad (16)$$

or as a velocity constraint:

$$\dot{h}(q, \dot{q}) \geq 0 \quad (17)$$

Most of the existing approaches are based on the following approximations:

$$h_{i+1} = h_i + \dot{h}_i dt + \frac{1}{2} \ddot{h}_i dt^2 \quad (18a)$$

$$\dot{h}_{i+1} = \dot{h}_i + \ddot{h}_i dt \quad (18b)$$

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 transform constraints (16), (17) into the form (10). To ensure that (16) will be satisfied, the velocity $\dot{h}(q, \dot{q})$ should be enforced to decrease until stop when $h(q) = 0$.

In what follows, we review two state-of-the-art methods: velocity linear damper [12] and viability formulation [22]. There are two reasons for this choice: beside the fact that [22] is the most up-to-date work in this topic, both of [12] and [22] considered the acceleration as a decision variable (in contrast to [14], [15] where the velocity is considered as a decision variable, kinematics only).

A. Related Works

1) *Velocity Linear Damper*: enforces \dot{h} to decrease linearly w.r.t h given the following formula [10]–[12]:

$$\dot{h}_{i+1} + \alpha h_i \geq 0 \quad (19)$$

where $\alpha > 0$ is a tuning parameter. Considering (18b), the second order form of (19) is given as, see e.g. [12]:

$$\ddot{h}_i + \frac{1}{dt} \dot{h}_i + \frac{\alpha}{dt} h_i \geq 0 \quad (20)$$

Resulting in a second order linear form 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. Indeed, the damping is constant and only the stiffness is variable.

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

$$\dot{h}_{i+1} + \sqrt{\alpha h_{i+1}} \geq 0 \quad (21)$$

where $\alpha = 2\ddot{h}_{\text{const}}$. Considering (18a) and (18b), the implementation proposed in [22] is based on a switching logic that guarantees the assumption of a constant deceleration and stop at $h = 0$. The resulting second order form is nonlinear and complex, but can be further simplified to:

$$\text{switching logic} \rightarrow \begin{cases} \ddot{h}_i + \frac{1}{dt} \dot{h}_i \geq 0 \\ \text{or} \\ \ddot{h}_i + \frac{1}{dt} \dot{h}_i + \frac{1}{dt} \sqrt{\alpha h_i} \geq 0 \end{cases} \quad (22)$$

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

However, the main and common drawback of both velocity linear damper and viability formulation methods is that they have never been used in closed-loop or varying bounds. This issue has even been explicitly stated in the conclusion section in [22]: “*an exact knowledge of the state and the model of the robot was assumed, hence the computed bounds are not robust to uncertainties*”. In fact, the exact knowledge of the constraint state ${}^c x$ is assumed in (18a) and (18b). Hereafter, we propose a method to formulate (16) and (17) in terms of acceleration (10) without any assumption on the knowledge of the robot state.

B. Second Order ODI Formulation of Constraints

Let us formulate the constraints (16) and (17) as a second order ODI:

$$(16) \rightarrow \ddot{h}_i + K_v \dot{h}_i + K_p h_i \geq 0 \quad (23)$$

$$(17) \rightarrow \ddot{h}_i + \gamma \dot{h}_i \geq 0 \quad (24)$$

γ is used to tune the dynamics of (24) which is a first order dynamics in term of velocity \dot{h} . However, the challenge here is tune the stiffness and damping gains K_p and K_v , respectively, to have a velocity damping while ensuring that (16) is satisfied. Namely, the constraint phase trajectory should be kept in the right half plane of the phase plot (h, \dot{h}) without overshooting $h = 0$ axis. Ad-hoc gains-tuning is inappropriate for a generic approach. Hence, we present a

systematic method to compute these 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):

$$\ddot{y} + K_v \dot{y} + K_p y = 0 \quad (25)$$

which is the saturated ODI (23). ODE (25) can be written as a system of two first-order ODE as follows:

$$\begin{bmatrix} \dot{y} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = K \begin{bmatrix} y \\ \dot{y} \end{bmatrix} \quad (26)$$

Thanks to Petrovitch theorem [23], it is shown that: given (23) and (25), under initial conditions $h(t_0) = y(t_0)$, $\dot{h}(t_0) = \dot{y}(t_0)$ for $t = t_0$, we get:

$$h(t) \geq y(t), \quad \forall t \geq t_0$$

This theorem means that the ODE solution $y(t)$ is the lower bound for all the ODI solutions $h(t)$. Hence, if $y(t) \geq 0 \Rightarrow h(t) \geq 0$. Hereafter, we will focus on the solution $y(t)$.

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

$$\lambda_{1,2} = K_v \left(-\frac{1}{2} \pm \frac{1}{2} \sqrt{1 - \xi^{-2}} \right), \quad \xi \geq 1 \quad (27)$$

where $\lambda_1 \geq \lambda_2$, $\Re(\lambda_{1,2}) < 0$ so that K is Hurwitz. The corresponding eigenvectors $\mu_{1,2}$ are:

$$\mu_{1,2}^T = [1 \quad \lambda_{1,2}] \quad (28)$$

For computation-time efficiency, (23) is inserted into the constraints set of (14) only if $h \leq h_d$, where h_d is the safety margin, and we say that the constraint (16) is active. Given an initial state $(y_0, \dot{y}_0) = (h_0, \dot{h}_0)$, the phase trajectory converges asymptotically to the origin $(0,0)$ if (27) holds. However, the phase trajectory shape does not depend solely on the eigenvalues. It depends also on the initial state. As it is shown in Fig.1, we can have different phase trajectories shapes for different initial states, but yet for the same eigenvalues. In the worst case, $y(t)$ can overshoot the axis $y = 0$, losing thereby any guarantee that (16) holds. This case happens if the initial condition is in the blue area in Fig. 1. We propose to 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 (16) can become active with many initial states. In what follows, we propose a systematic method to overcome this issue.

C. Gains Adaptation Method

In order to avoid overshooting, the initial state must not be in the unviable region. One solution is to adapt the gains to change the shape of the viable region so that the initial state is included. Since the eigenvector μ_2 is the border limit between the viable and unviable regions, the idea is to look for the suitable gains that bring μ_2 to pass by the initial state (h_0, \dot{h}_0) . This enforces the phase trajectory to remain within the viable region.

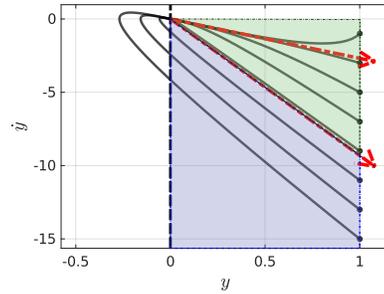


Fig. 1. Second order ODE (25) solutions (in black solid) depending on the initial states (nodes at $y = 1$), for the same eigenvalues where the corresponding eigenvectors are shown but red dashed arrows. The green zone represents the viable region where all the solutions satisfy the constraint (16). The blue zone denotes the unviable region where all the solutions overshoot the axis $y = 0$.

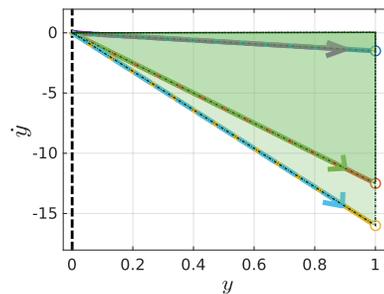


Fig. 2. Second order ODE (25) solutions (solid) with the adaptive gains method. The viable region changes its shape according the initial state (nodes at $y = 1$). The eigenvectors (dashed arrows) are delimiting the viable regions. The unviable regions are not shown for clarity of the illustration.

Given that μ_2 slope is equal to λ_2 , the following formulas should be satisfied:

$$\frac{\dot{h}_0}{h_0} = \lambda_2 \Rightarrow \begin{cases} K_v = \frac{2(h_0/\dot{h}_0)}{-1 - \sqrt{1 - \xi^{-2}}} \\ K_p = \left(\frac{K_v}{2\xi}\right)^2 \end{cases} \quad (29)$$

where ξ is to be chosen according to the desired dynamics: critically damped or overdamped. The gains are computed once, at each time the constraint is active (see Fig. 2)

The main advantage of this method is to be adaptive to any initial state. In fact, the latter depends on the dynamics resulting from the tasks and the current active constraints. One may choose a high K_p and K_v gains which will result on a wide viable region. However, it is not recommended to use high gains in closed-loop: it will result on strong vibrations and chattering.

However, because of noises and non modeled nonlinearities like flexibility and friction, the phase trajectory may get off slightly from the viable region. As a consequence, it may lead to a slight overshoot before converging to zero. For this reason, we proposed to improve this method by considering a nonlinear-gains feedback: $K_p = f_p(h)$, $K_v = f_v(h)$.

D. Nonlinear-Gains based Feedback

Formulation (25) can be seen a normalized spring-damper model equation. Let us assume a nonlinear model, which

means that K_p and K_v are not constant and depend on h . Let us consider the scheme in Fig. 3. It shows a virtual deformable object, constituted by a set of parallel layers of identical springs and dampers. These layers are shifted from each other by an infinitesimal distance $d\theta$. Equivalently, this model can be seen as nonlinear spring and damper, where the goal is to compute the equivalent stiffness and damping gains K_p^{eq} and K_v^{eq} , respectively, that correspond to the nonlinear spring and damper coefficients.

Let us have:

$$K_p^{\text{eq}}(h) = K_p + K_p^{\text{disp}}(h)$$

where K_p is obtained by (29), and K_p^{disp} is the varying part which is added due to the displacement. Hence,

$$K_p^{\text{disp}}(h) = - \int_{h_d}^h K_p \varphi_p(\theta) d\theta$$

where $\varphi_p(\theta)$ denotes 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:

$$K_p^{\text{eq}}(h) = K_p [1 + \varphi_p (h_d - h)] \quad (30)$$

Similarly, we have:

$$K_v^{\text{eq}}(h) = K_v + K_v^{\text{disp}}(h)$$

Following the same approach to obtain (30), we get:

$$K_v^{\text{eq}}(h) = K_v [1 + \varphi_v (h_d - h)] \quad (31)$$

where φ_v is the dampers density per displacement $d\theta$, and which is chosen to be constant. From (30) and (31), ξ^{eq} is obtained as:

$$\xi^{\text{eq}}(h) = \frac{K_v^{\text{eq}}(h)}{2\sqrt{K_p^{\text{eq}}(h)}} = \xi \frac{1 + \varphi_v (h_d - h)}{\sqrt{1 + \varphi_p (h_d - h)}} \quad (32)$$

If $\varphi_v = \varphi_p = \varphi$, this yields to:

$$\xi^{\text{eq}}(h) = \xi \sqrt{1 + \varphi (h_d - h)} \quad (33)$$

The resulting feedback gains (30) and (31) increase linearly while the phase trajectory is converging to zero. In addition, if $\xi = 1$, the constraint dynamics is changing from critically damped to more and more overdamped. From a physical standpoint, this has the effect of absorbing the kinetic energy as the deceleration time increases proportionally to ξ^{eq} .

IV. EXPERIMENTAL RESULTS

To validate our approach, we use a position-controlled humanoid robot HRP-4. The `mc_rtc`¹ 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 velocity is estimated using finite difference. In return, `mc_rtc` provides the

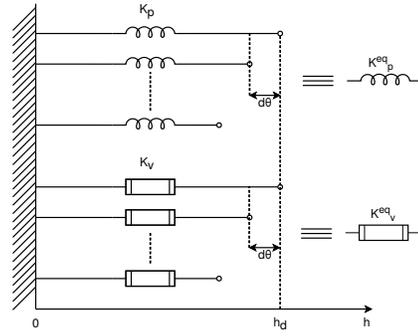


Fig. 3. Nonlinear spring-damper model.

desired joints positions q^* . Different experimental scenarios are performed in closed-loop. First, we compare between the state-of-the-art methods and our approach applied for collision avoidance constraints. Then, we extend and test our approach for other class of constraints like joint limits, CoM and velocity constraints.

A. Collision Avoidance

To compare our method to the state-of-the-art ones previously described, we extend the viability formulation constraint –originally implemented for the joints limits [22], to collision avoidance constraint: $h = \delta - \delta_{\min} \geq 0$. \dot{h}_{const} is computed such that $\dot{h}_{\text{const}} = \frac{\dot{h}_0^2}{2h_d}$. Collision 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 two bodies as $\delta_{\min} = 6$ cm with a safety margin $h_d = 4$ cm. The damping coefficient and the springs and dampers density are fixed to best: $\xi = 1.2$ and $\varphi_p = \varphi_v = \frac{0.5}{h_d}$. Starting by an initial posture where the shoulder are fully stretched, we give a joint position targets for the shoulders so that the elbows gets close to the waist. The experiment is performed using ‘velocity linear damper’, ‘viability formulation’ and our approach. The results are shown in Fig. 5. The state-of-the-art methods lack of robustness against the noises and un-modeled nonlinearities encountered in closed-loop: oscillations and chattering especially near δ_{\min} . In contract, our approach exhibits smooth behaviors and the velocity decreases to reach zero when the distance reaches its defined bounds.



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

¹https://jrl-umi3218.github.io/mc_rtc/index.html

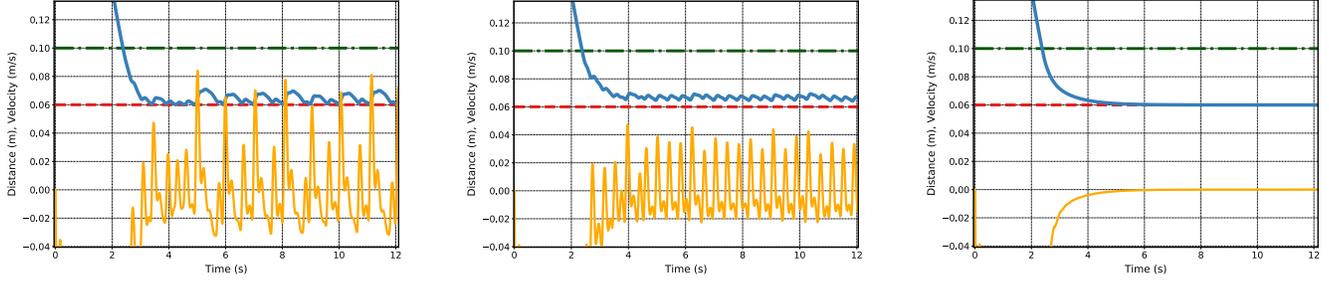


Fig. 5. Three different methods performed in closed loop: viability formulation [22] (left), velocity linear damper [13] (center) and our approach (right). δ_{\min} (in dashed red) is the lower limit of the distance between a pair of bodies (in blue solid). The chattering is visible in the velocity (in orange solid) for the other methods, whereas it converges to zero smoothly in our approach. The safety margin is the space between the dash dotted green and dashed red.

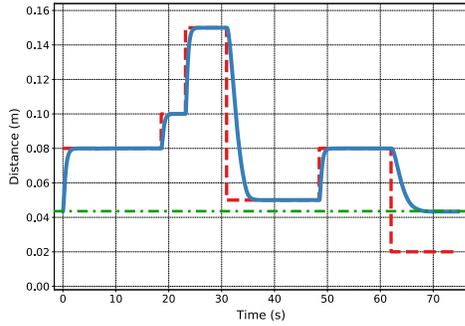


Fig. 6. Collision avoidance constraint with variable δ_{\min} (red dashed). The distance δ (blue solid) converges to δ_{\min} if $\delta_{\min} \geq \delta_{\text{target}}$ (green dash dotted).

An other property of the proposed approach is that it allows to deal with variable distance bounds that are usually assumed constant [22]. Figure 6 shows that at $t = 0$, the constraint is violated $h < 0$. However, δ converges to δ_{\min} in any case the latter is superior than the target δ_{target} and QP does not fail even though the constraint is briefly violated.

B. Joints limits and CoM Constraints

To validate our approach on the CoM constraint, we define an arbitrary polygon equilibrium region for the CoM [25]. Next, we define an end-effector task for the right hand for which the target is unreachable. The aim is to push robot's CoM to reach the polygon bounds. Different targets are defined to push to CoM to the different polygon boundaries (see Fig. 7). 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's floating base is estimated using a Kinematic Inertial Observer based on the IMU measurement and the kinematics chains of the robot [5]. Figure 8 shows the temporal and spatial evolution of the estimated CoM in addition to the different time-frame where either the right or left arm is moving. We can notice that the CoM is constrained to remain inside the polygon except in the yellow-highlighted places representing overshoots. The latter have a magnitude of 2 mm at maximum and occur very briefly when the end effector task is assigned. The

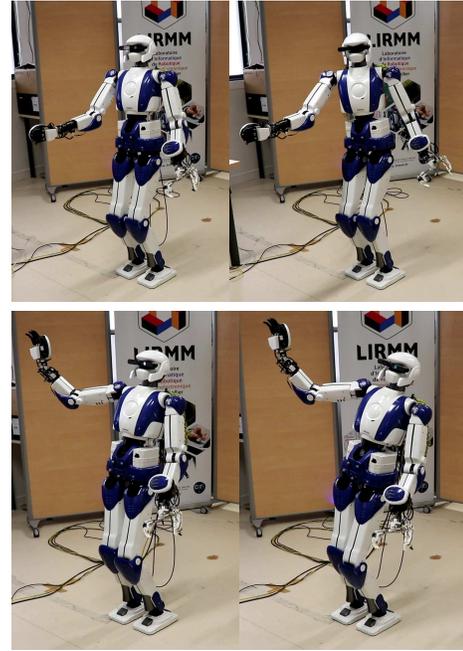


Fig. 7. HRP-4 trying to reach a far right hand target in order to push the CoM to the equilibrium polygon boundaries.

reason behind this is due to the un-modeled flexibilities at the ankles which act as a perturbation to the constraint at the boundary. However, these perturbations are damped and the constraint is stabilized at the boundary. Furthermore, it is important to notice that despite the noisy estimation of CoM, no discontinuities or chattering have been noticed during the experiment. Figure 9 shows how the robot adjust its whole-body posture when the left shoulder joints are fully stretched to maintain the CoM at the polygon boundary highlighted with the light green area in Fig. 8. The joints positions and velocities evolution of the left shoulder roll and pitch joints are shown in Fig. 10, where the joints limits are reached smoothly with a zero velocity.

C. Velocity Constraint

To validate our approach for velocity constraint formulation, we choose to bound the relative velocity between the bodies highlighted in Fig. 4. As in Section IV-A, variable

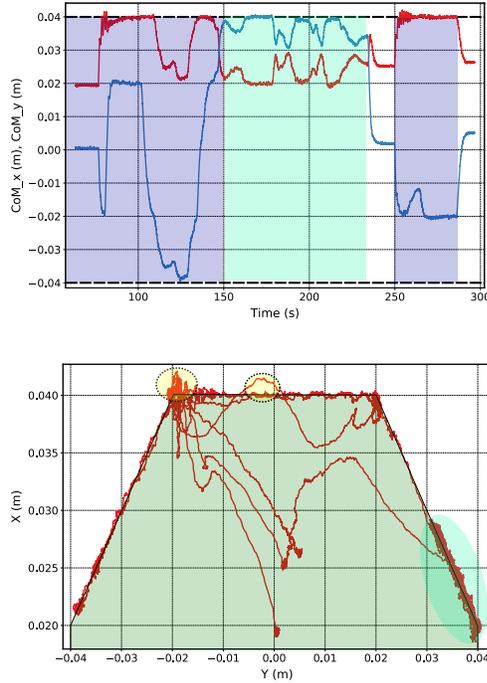


Fig. 8. CoM evolution in time (top) and in space (bottom). The dark green area denote the equilibrium region for the CoM where it should remain. The blue and light green zones highlight the periods in time as well as in space where CoM moves because of the right hand end effector task or because of the left arm movement, respectively.

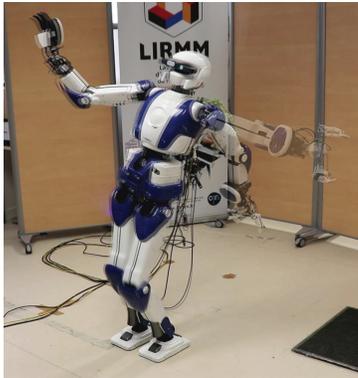


Fig. 9. Superposition of two pictures when two different joints limits are reached. The blur of the whole body shows how the robot adjusts its posture to keep the CoM at the boundary of the equilibrium polygon (green area in the bottom Fig. 8).

velocity limits are assumed. The gain γ is fixed to $\gamma = 10$. Figure 11 shows how the velocity evolution converges to the limits when the latter is variable. Again, despite the noisy estimation of the velocity, no chattering or discontinuities have been noticed during the experiment. In [12], [22], based on (18b), γ is taken such that: $\gamma = \frac{1}{dt} = 200$ which is a high gain and not recommended in closed-loop. In fact, QP failed to find a solution when the velocity approached the limit of 0.05 m/s while attempting to perform velocity constraint during experiment with $\gamma = 200$.

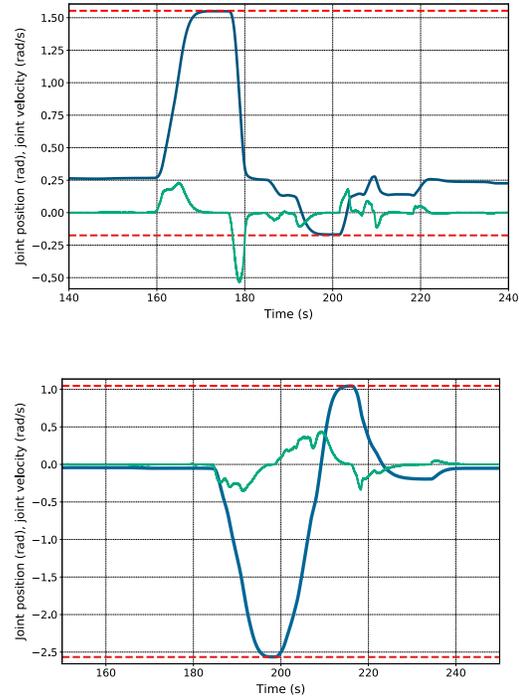


Fig. 10. Left shoulder roll (top) and pitch (bottom) joints positions (blue) and velocities (green). The upper and lower joints limits are shown in red dashed.

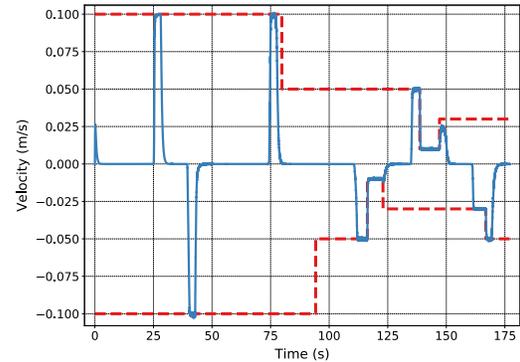


Fig. 11. Velocity (blue solid) evolution with variable bounds (red dashed).

V. CONCLUSION

In this letter, we propose a solution to deal efficiently with a class of constraints commonly found in task-space QP controllers that use acceleration, force and torque as decision variables. More particularly, we address constraints that originally do not write in the decision control variables only after successive numerical derivations. Example of such constraints are joint position limits, minimum allowable distance for collision avoidance, center-of-mass subscription, field-of-view and obstructions constraints in visual servoing, etc. First, we show that existing approaches, mainly velocity dampers and viability formulation have shortcomings when they reach their bounds. The main limitations acknowledged

by their authors are: (i) the need of precise models (no noise); (ii) they do not operate well in closed-loop scheme, i.e. when the state of the robot 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 investigated the reasons that highlighted limitations in the ‘gains’ parametrization and their subsequent tuning. We propose another formulation for such constraints at large, with an automated adaptive gain tuning to overcome all the previously mentioned problems. Experiments conducted on the HRP-4 humanoid robot confirmed our approach, which was then integrated as part of our `mrtec` controller library.

As future work, 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 robustness and performances.

REFERENCES

- [1] A. Herzog, N. Rotella, S. Mason, F. Grimmering, 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. Engelsberger, 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 (ICRA)*, pp. 440–447, IEEE, 2018.
- [5] 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 (ICRA)*, pp. 277–283, May 2019.
- [6] Y. Wang and A. Kheddar, “Impact-friendly robust control design with task-space quadratic optimization,” in *Robotics: Science and Systems (RSS) (A. Bicchi, H. Kress-Gazit, and S. Hutchinson, eds.)*, (Freiburg im Breisgau, Germany), June 2019.
- [7] Y. Wang, A. Tanguy, P. Gergondet, and A. Kheddar, “Impact-aware humanoid robot motion generation with a quadratic optimization controller,” in *IEEE International Conference on Humanoid Robots (Humanoids)*, (Toronto, Canada), Oct. 2019.
- [8] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Engelsberger, 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.
- [9] 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 ihmcs lessons learned from the darpa robotics challenge trials,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.
- [10] 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.
- [11] 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, IEEE, 2010.
- [12] 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.
- [13] J. Vaillant, K. Bouyarmane, and A. Kheddar, “Multi-character physical and behavioral interactions controller,” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 6, pp. 1650–1662, 2016.
- [14] W. Decré, R. Smits, H. Bruyninckx, and J. De Schutter, “Extending itac to support inequality constraints and non-instantaneous task specification,” in *IEEE International Conference on Robotics and Automation*, pp. 964–971, IEEE, 2009.
- [15] 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.
- [16] 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, IEEE, 2011.
- [17] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J.-Y. 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.
- [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 (Humanoids)*, pp. 21–27, IEEE, 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 16th International Conference on Humanoid Robots (Humanoids)*, 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 (ICRA)*, pp. 3458–3464, IEEE, 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] M. Michel Petrovitch, “Sur une manière d’étendre le théorème de la moyenne aux équations différentielles du premier ordre,” *Mathematische Annalen*, vol. 54, pp. 417–436, Sep 1901.
- [24] K. Bouyarmane and A. Kheddar, “Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4414–4419, Sep. 2011.
- [25] H. Audren, A. Kheddar, and P. Gergondet, “Stability polygons reshaping and morphing for smooth multi-contact transitions and force control of humanoid robots,” in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 1037–1044, Nov 2016.
- [26] 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.
- [27] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.
- [28] 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.
- [29] 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.
- [30] I. Kim and J.-H. Oh, “Inverse kinematic control of humanoids under joint constraints,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 74, 2013.
- [31] M. Faroni, M. Beschi, N. Pedrocchi, and A. Visioli, “Viability and feasibility of constrained kinematic control of manipulators,” *Robotics*, vol. 7, no. 3, p. 41, 2018.