# Impact-Aware Task-Space Quadratic-Programming Control

Yuquan Wang, Niels Dehio, Arnaud Tanguy, Abderrahmane Kheddar

# Impact-Aware Task-Space Quadratic-Programming Control

Journal Title
XX(X):1–19

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

**Yuquan Wang, Niels Dehio, Arnaud Tanguy, and Abderrahmane Kheddar**

**Abstract**

Generating on-purpose impacts with rigid robots is challenging as they may lead to severe hardware failures due to abrupt changes in the velocities and torques. Without dedicated hardware and controllers, robots typically operate at a near-zero velocity in the vicinity of contacts. We assume knowing how much of impact the hardware can absorb and focus solely on the controller aspects. Hybrid controllers with reset maps provided elegant solutions for given impact tasks. The novelty of our approach is twofold: (i) it uses the task-space inverse dynamics formalism that we extend by seamlessly integrating impact tasks; (ii) it does not require separate models with switches or a reset map to operate the robot undergoing impact tasks. Our main idea lies in integrating post-impact states prediction and impact-aware inequality constraints as part of our existing general-purpose whole-body controller. To achieve such prediction, we formulate task-space impacts and its spreading along the kinematic and potentially arborescent, structure of a floating-base robot with subsequent joint velocity and torque jumps. As a result, the feasible solution set accounts for various constraints due to expected impacts. In a multi-contact situation of under-actuated legged robots subject to multiple impacts, we also enforce the dynamic equilibrium margins. By design, our controller does not require precise knowledge of impact location and timing. We assessed our formalism with the humanoid robot HRP-4, generating maximum contact velocities, neither breaking established contacts nor damaging the hardware.

**Keywords**

Task-space control, quadratic optimization control, humanoids, impact tasks.

## 1 Introduction

In robotics, instantaneous impacts occur when robots make contact – intentionally or accidentally – at relatively high velocities (cf. Fig. 1). Impacts may cause high impulsive forces and velocity jumps at the contacting surfaces. Impact energy and jumps propagate through the robot's links in the joints, which might cause severe mechanical damage. Abrupt changes in velocity or torque mean discontinuities that may destabilize closed-loop controllers. Floating-base robots can also witness sudden breaks of pre-existing unilateral contacts resulting in a sudden loss of balance.

When possible, a simple solution is to plan and control new contacts carefully with near-zero contact admissible velocities. Otherwise, new contacts are made with relatively high velocities, e.g., walking or jumping humanoids. In this case, the robot continuous dynamic equations of motions are not a good match for the induced physics, and another set of equations must be considered together with a transition policy (called reset maps), see a review in Sec. 2. Such a transition policy requires knowledge of additional parameters that depend on the environment (and robot) stiffness, the impact model, the impact localization on the robot (and the environment), the contact normal, the exact impact time, and even more. Acquiring *in-situ*, instantaneously, and reliably these parameters are not always possible in practice.

In fact, a large part of handling properly robotic impacts must be tackled first from a hardware design perspective, see the introductory part of Sec. 2. It is the hardware design

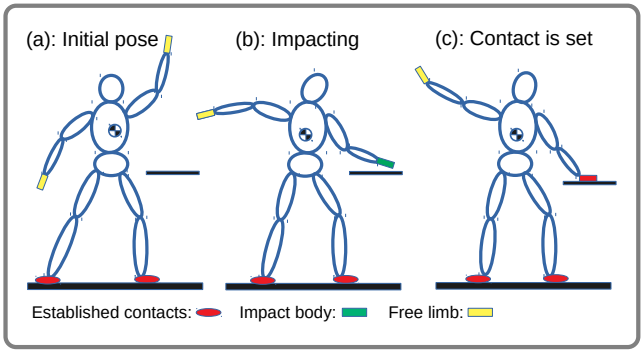



**Figure 1.** In this schematic sketch, a humanoid robot is impacting a wooden piece as we can commonly find in karate. In stance (a), there are $m_1 = 2$ established contacts (red), $m_2 = 0$ impacting end-effectors (green), and $m_3 = 2$ free limbs (yellow). Stance (b) is represented by $m_1 = 2$, $m_2 = 1$, $m_3 = 1$ and stance (c) by $m_1 = 3$, $m_2 = 0$, $m_3 = 1$.

Y. Wang, N. Dehio and A. Kheddar are with the CNRS-University of Montpellier, LIRMM, Interactive Digital Humans group, Montpellier, France.
A. Tanguy and A. Kheddar are with the CNRS-AIST Joint Robotics Laboratory, IRL, Tsukuba, Japan.
Y. Wang and A. Kheddar are in part with the Beijing Institute of Technology, Beijing, China.

**Corresponding author:**
Yuquan Wang
Email: wyqsnddd@gmail.com

that reveals how much impact the robot can achieve without failures. Once we know the hardware capabilities in terms of impact resilience, we can consider impact-aware motion planning and control. For classical robot controllers, the impact duration is too small to react within the impact events ($1-5$ ms), see details in Sec. 2. Yet, we can apply a low-level controller design to regulate the pre-impact velocity safely.

In this paper, we address general-purpose impact-aware tasks, including floating-base robots. Instead of explicitly designing a dedicated controller to handle task-aware impacts, we integrate impact and continuous dynamic models as well as constraints consistently as part of our whole-body multi-objective controller that relies on *quadratic program* (QP) solving (Bouyarmane et al. 2019).

The core idea is to perform a one-step-ahead prediction in every control cycle based on the impact model. That is to say, nearby intended or expected impact, we assume that it will happen in the next iteration. As a result, the controller becomes *aware* of impact-induced jumps and restricts the robot motion to meet impact with (possibly maximum) velocities that can handle the upcoming impact safely, i.e., within the pre-defined hardware resilience and task-dictated acceptable bounds. By our approach, the resulting robot motion is robust to uncertainties in impact time and location. This feature is the main novelty and the most appealing aspect of our approach w.r.t. State-of-the-art.

In more details, our main contributions are:

**C.1** We formulate the problem of impact propagation as an auxiliary *Impulse Distribution Quadratic Program* (IDQP) to estimate the impulses associated with kinematic branches and their effects on links of interest that are free or that are in unilateral-contacts (Sec. 4.4).

**C.2** Building upon the predicted impact effect, we gather all the *state jumps* (Sec. 4.5) that floating-base multi-limb robots may experience concerning the velocity space, force and torque space, and the centroidal space states.

**C.3** Approximating post-impact states using Euler-forward method, we propose a *generic template constraint* to bound the post-impact state of a generic quantity (Sec. 5.1).

**C.4** Our framework can also deal with multiple impacts simultaneously. Integrating impact constraints into the QP controller enables *multi-impact-aware multi-contact whole-body motion generation* (Sec. 5.5), which is validated through experiments on our humanoid robot HRP-4 (Sec. 6).

To sum-up, our novel approach ensures the feasibility of the underlying task-space quadratic programming controller, accounting for hardware safety and feedback state abrupt jumps. It allows a built-in seamless interplay between continuous and impact dynamics without branching and reset-maps. It is an alternative (and not necessarily a replacement or a competitor) to existing well-established hybrid and or reset map approaches.

## 2 Background

Analyses in Tsujita et al. (2008); Pashah et al. (2008) revealed that impact duration is typical of the range of milliseconds order or less, even for low-velocities. Suppose that the impact effects are given, it is non-trivial to devise a controller that prevents hardware from being somehow hindered within such a short period. Thus various hardware solutions have been proposed to mitigate the impulse. As it is not our primary concern, we report only a few in Sec. 2.1.

In specific scenarios, impacts have to be accounted for in the control design. Nevertheless, it is rather more a part of the task to achieve than something that we would like to control directly (Sec. 2.2). For example, biped walking induces impacts because of the nature of the walking task, and it is essential to control the gaits with dynamic balance. Instead, kicking a ball is rather a task for which we would like to control the impacts of the feet with the ball.

On the other hand, our proposed approach exploits on-purpose impacts to increase robot manipulation efficiency (e.g., kicking the ball) and generates a broad spectrum of impulsive forces directly related to task performance. We summarize controllers of this kind in Sec. 2.3.

### 2.1 Mitigating impact through hardware improvements

Hardware resilience to impact shall inform on how much impact a robot can perform without breaking links, joint mechanisms and embedded electronics. For example, employing *variable stiffness actuator* (VSA) technologies lowers damage risks at impacts. However, it may need more than 10 ms to generate the joint torque that can counterbalance the torque jump (Haddadin et al. 2009). Despite numerous VSA paradigms that have been proposed – refer to the excellent review by Vanderborght et al. (2013) – their force control capabilities suffer from limited bandwidth as pointed by Wensing et al. (2017).

Aiming to reduce the initial and post-impact forces, the *proprioceptive actuator* devised for the MIT Cheetah robot's leg provides a mechanical approach to mitigate impacts without added compliance (Wensing et al. 2017). More recently, Singh and Featherstone (2020) proposed a novel quadruped robot leg design that removes the shock propagation from the floating-base. Another strategy absorbs the impacts at foot-strike through passive springs in the ankle (Reher et al. 2016). Designing robust hardware specifically for impact tasks is appealing, and notable progress has been made.

However, the question remains on how to deal with impacts with existing robots. An outer softcover approach was described in Battaglia et al. (2009), where the thickness of the cover is computed as a function of latency and impact velocity. Generally, shock-absorbing mechanisms or soft soles are added to the ankle/feet of humanoids. There is certainly more to do and investigate in order to increase the resilience of robots to impacts. Nevertheless, it should be noted that for almost all the existing robots, gathering from their manufacturer parameters concerning tolerable impact bounds is already informative of the current facts. To our requests, we always had the same pair of answers: *Why do you need this?* And, after our explanations... *You are not*

*supposed to do that with our robots!* The exception is the Franka Emika Panda robot, for which such data are available upon request.

## 2.2 Handling task-induced impacts

Impacts are sometimes rather seen as perturbations. For example, Pagilla and Yu (2001) consider impact as a transient behavior and uses it for stability analysis rather than for explicit control design. Yet, Stewart (2000) reviewed that locomotion tasks have treated the impulse effects for two decades. The *complementarity dynamical system* (CDS) proposed by Hurmuzlu et al. (2004) offers a general conceptual tool to describe the dynamics of a walking robot, e.g., a biped with one or two contacts. Interested readers find a summary of the CDS controllability and stabilizability for mechanical applications in Brogliato (2003). Impact-induced state jumps, which are embedded in a reset map, are applied to the robot states depending on the complementarity condition status. However, Grizzle et al. (2014) pointed out that there is no effective control design for CDS.

Although hybrid control is a powerful modeling and control tool for non-smooth mechanical systems in general, e.g., for robots undergoing intermittent impacts (Johnson et al. 2016), it scales poorly concerning the number of potential contacts, refer to improvements by Posa et al. (2015). For instance, in the case of biped robots, there are two single support modes, double supports mode, and the transition phases. If the model is not complete enough, *Zeno phenomena* might occur and lead to infinite accumulations of impacts in a finite time (Or and Ames 2010). Nevertheless, hybrid approaches achieved successful applications in practice, e.g., efficient bipedal walking by Sreenath et al. (2011), walking over uneven terrain Manchester et al. (2011), dynamic bipedal vertical climbing in simulation by Aghasadeghi et al. (2012), and dynamic bipedal walking by Reher et al. (2016).

Rather than dealing with periodic impacts or cyclic behaviors, in this paper, we are interested in the local feasibility properties before and right after a single impact event. Hence, modeling using the *Poincaré map* (Grizzle et al. 2014), and the controllers developed on top of it do not apply straightforwardly to our aim. Comparing our approach with the impact dynamics model applied for walking (Hurmuzlu et al. 2004; Grizzle et al. 2014) or hybrid control approaches by Rijnen et al. (2017) is pointless. This is because we target general-purpose tasks implying any limb of the robot. Contrarily to walking only, we do not assume that the holonomic constraints, e.g., the position and orientation of a contacting limb, remain constant before and after the impact. Furthermore, we estimate the end-effector velocity jump for free limbs and for established unilateral contacts (**C.2**).

Another difference comes from the number of impacts and contacts. Switching from a single support phase to a double support phase merely induce a single impact concerning the other support (established contact). Grizzle et al. (2014) pointed out that conservation of generalized momentum (in the joint space) for a single kinematic branch is used in one form or another for most of the walking-related works. Instead of solving joint velocity jumps on a limb-by-limb basis, we solve the state jumps that are compatible with each limb due to the whole body momentum conservation and balance (**C.1** and **C.2**).

Alternatively, some control strategies simply avoid impacts when contacts are about to be created. Given the impact location, Pagilla and Yu (2001) modified the reference trajectories such that the reference velocity along the surface normal is zero. The control design based on *Zero-tilting Moment Point* (ZMP) ignores the impact dynamics and establishes contacts with close to zero velocity – checkout the example by Kajita et al. (2010). Grizzle et al. (2014) summarized that the impact-less reference trajectories are challenging to generate and inefficient to execute: we totally agree. Adding impact-awareness to the QP control framework (**C.4**), our proposed approach autonomously executes the reference motion with a possibly reduced, if it violates the hardware limits or other user-defined constraints (**C.3**), yet maximum safe contact velocity.

## 2.3 Handling on-purpose impact tasks

Commonly *flexible* models with regularization are used e.g. the mass-spring-damper (Hu et al. 2007; Stanisic and Fernández 2012; Heck et al. 2016) to enable well-established control tools for continuous-time dynamics. Recently, for dealing with general-purpose impact behaviors, Rijnen et al. (2017) proposed to switch once from pre- to post-impact reference based on the detection of the impact event, (referred to as reference spreading). Konno et al. (2011) solved a three-phase non-linear optimization problem to generate reference trajectories as well as the posture at the impact moment to maximize the force jump. These methods require off-line reference trajectory planning and switching, which compromises reactiveness. It further conflicts with the concept of a desirable general-purpose multi-objective controller that is based on real-time sensory feedback.

The reference switching is performed upon the impact detection. Note, that the observation is always delayed and may be inaccurate. Thus, in general, due to the unknown exact impact timing (Pagilla and Yu 2001), switching controllers would apply setpoints or references, that are defined for the pre-impact mode, during the post-impact mode for a specific time interval, no matter how small it is. The resulting undesired misbehavior imposes the risk of exceeding hardware limitations.

Our approach is rather conservative in that, nearby impact, the controller starts assuming the impact to happen in the next time step. Our proposed impact-aware QP controller (**C.4**) provides the mapping between the impact-induced state jumps and the pre-impact velocities in each iteration (**C.2**). The mapping enables the QP controller to regulate the robot motion in real-time such that the impact-aware constraints are respected (**C.3**). We can find a similar mapping provided by Partridge and Spong (2000) to control the trajectory of a three-link planar robot. The proposed QP controller is free of off-line trajectory computation and, therefore, robust to impact timing and location. The prediction of post-impact states that we propose (**C.2**) is based on the impact dynamics model along with the contact normal direction by Zheng and Hemami (1985) without considering tangential impulse and impact-induced friction.

A similar impact dynamics model is introduced by Hurmuzlu et al. (2004) to address the bipedal walking as summarized in Grizzle et al. (2014); Wieber et al. (2016). In the field of space robotics, Nenchev (2013) applied joint space momentum conservation to study the post-impact robot motion behavior. In this paper, we do not have the zero gravitational force assumption, and we utilize the centroidal momentum conservation (Orin et al. 2013).

Nava et al. (2016) pointed out that the state-of-the-art momentum controller may lead to unstable zero dynamics. Nenchev (2018) recently proposed a decoupled controller based on the relative momentum. Instead of applying the – not yet – fully exploited momentum-based control strategies, we prefer to limit the impact-induced centroidal momentum jumps (**C.3**).

Even though force jumps are typical for legged robots (Bergés and Bowling 2005), it is not straightforward to include an analytical solution of the tangential impulses. Recently, Jia et al. (2019) developed a flying object batting example, where a closed-form 2D impact dynamics model is applied. It generates the desired impulse by enumerating five possibilities. In the 3D cases, the closed-form solution is only available if we can control the initial sliding direction to an invariant subset, for more details, refer to (Jia and Wang 2017). Note that according to the energy conservation principle, it is not possible to treat impact dynamics along with the two tangential directions and normal direction separately. One caveat of this approach is the mathematical and computational complexity of the model.

Control formulations based on a quadratic program (QP) with linear constraints have become the predominant scheme for operating legged robots, (Kuindersma et al. 2016; Liu et al. 2016; Bouyarmane and Kheddar 2018) to name just a few. The approach allows pursuing multiple prioritized objectives simultaneously while satisfying strict bounds. It has been applied both to position- and torque-controlled robots, usually employing $1-5$ ms update loops. We adopt and further build on our initial concept proposed in simulation for fixed-base robots in Wang and Kheddar (2019) and the preliminary extension to the floating-base robot in Wang et al. (2019). Concerning these and previously cited contributions, this article provides (i) a detailed discussion on multi-impact cases and different least-squares solutions that were not covered before; (ii) additional constraints (**C.2**), e.g., angular momentum, framed as a generic impact-aware template (**C.3**); (iii) improving the impact effect estimation (**C.1**) by adding the centroidal *momentum conservation* constraint and minimizing a weighted minimum norm solution.

The Zero-tilting Moment Point (ZMP) is widely used as a balance criterion for biped walking on the planar ground, for example, refer to (Hildebrandt et al. 2019). Recently, the concept has been extended to the multi-contact case through excellent analysis in Caron et al. (2017). For tasks that require large impulsive forces, e.g., a nailing task discussed in Tsujita et al. (2008) and a wooden piece breaking task performed in Konno et al. (2011), ZMP is used to analyze the dynamic equilibrium of each robot configuration instance. Introducing impact-aware constraints (**C.3**) to bound the impact-induced state jumps (**C.2**) including ZMP, Center-of-Mass (COM) velocity, *Divergent Component of Motion* (abbreviated as DCM by Koolen et al. (2012)) and the centroidal momentum, our proposed method allows more reliable and robust motion generation.

To our best knowledge, intentionally generating high impacts with humanoids is studied only in very few works and for specific scenarios, e.g., Konno et al. (2011). Despite impressive results achieved with non-linear optimization for planning, their controller neither accounts for uncertainties in most impact parameters nor explicitly accounts for constraints in the closed-loop motion. We aim to extend state-of-the-art task-space multi-objective and multi-sensory whole-body control framework formulated as a quadratic program (QP) to encompass impact tasks (**C.4**).

## 3 Impact-unaware Constraint-based QP-Control

The robot task-space QP control – that we extend to impact-awareness in this paper – is detailed in Bouyarmane et al. (2019), including floating-base robots. In this section, we recall the formulation of the main constraints used in continuous dynamics with our notations. All variables are referring to the current time step, during which the QP decision (i.e., control) variables are computed. These variables are the robot state (including the floating-base) acceleration, contact forces, and motor torques. The latter can eventually be eliminated from QP using their bounds.

We introduce the kinematics and dynamics in Sec. 3.1; the joint space, contact space, and centroidal space constraints in Secs. 3.2, 3.3, and 3.4, respectively. We summarize the QP controller without considering the impacts in Sec. 3.5, and we point out its impact-related problems in Sec. 3.6.

### 3.1 Robot model

Model-based whole-body control requires to express multiple prioritized task objectives and constraints in the operational or the joint space (Dehio 2018; Lober et al. 2019; Dietrich and Ott 2020). The approach uses robot kinematic and dynamic models, see examples in Kuindersma et al. (2016); Nenchev (2013); Liu et al. (2016).

Consider an arborescent kinematic structure with $m = m_1 + m_2 + m_3$ independent end-effectors that we separate into three categories:

1. $m_1$ end-effectors with established contacts;
2. $m_2$ end-effectors for which we expect an upcoming impact (or dynamic contact transition), and
3. $m_3$ free limbs whose momentum jump can mitigate an external impulse propagating from other end-effectors.

We denote these three sets of end-effectors with $\sigma_{m_1}$, $\sigma_{m_2}$, and $\sigma_{m_3}$ respectively. Fig. 1 illustrates an exemplary impact task with varying $m_1$, $m_2$, $m_3$ over time.

For legged robots, we can represent the floating-base, which holds a configuration in $SE(3)$, by six virtual joints. Thus the generalized joint position writes $\boldsymbol{q} \in \mathbb{R}^{(n+6)}$. We choose the base link velocity expressed in the inertial frame $\boldsymbol{V}_b \in \mathbb{R}^6$ to construct the generalized velocities as:

$$\dot{\boldsymbol{q}} := [\boldsymbol{V}_b^\top, \dot{\boldsymbol{\theta}}^\top]^\top.$$

A floating-base robot with $m_1$ established contacts is governed with the following equation of motion:

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = B\boldsymbol{\tau} + \sum_{i \in \sigma_{m_1}} J_i^\top \boldsymbol{f}_i \tag{1}$$

where $B \in \mathbb{R}^{(n+6)\times n}$ selects actuated joints, $\mathbf{M}(\boldsymbol{q}) \in \mathbb{R}^{(n+6)\times(n+6)}$ denotes the inertia matrix, the vector $\mathbf{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{(n+6)}$ gathers both the Coriolis $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{(n+6)\times(n+6)}$ and the gravitational forces, vector $\boldsymbol{\tau} \in \mathbb{R}^n$ corresponds to the actuated joint torques. As can be noted, the $m_2$ end-efforts are not accounted at this stage.

## 3.2 Joint space constraints

Many constraints, such as joint limits, velocity limits, field-of-view, collision avoidance... to name just a few, do not express directly in the robot state acceleration. Such constraints need to be derived numerically prior to being added as a QP task objective or constraints. By approximating the derivative using Euler backward method, we predict (to some precision) the joint positions $\mathbf{q} \in \mathbb{R}^n$ at time step $k$ using the decision variable $\ddot{\mathbf{q}}_k$ and the measured joint positions and velocities at time step $k-1$:

$$\mathbf{q}_k = \ddot{\mathbf{q}}_k \Delta t^2 + \dot{\mathbf{q}}_{k-1}\Delta t + \mathbf{q}_{k-1} \tag{2}$$

where $\Delta t$ denotes the sampling period. Similarly, we obtain the joint velocities as:

$$\dot{\mathbf{q}}_k = \Delta t \ddot{\mathbf{q}}_k + \dot{\mathbf{q}}_{k-1}. \tag{3}$$

Given the joint limits $\underline{\mathbf{q}} \leq \mathbf{q}_k \leq \bar{\mathbf{q}}$ and the kinematic relationship (2) we obtain

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta t^2 \ddot{\mathbf{q}}_k \leq \begin{bmatrix} \bar{\mathbf{q}} \\ -\underline{\mathbf{q}} \end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix} \left(\mathbf{q}_{k-1} + \dot{\mathbf{q}}_{k-1}\Delta t\right). \tag{4}$$

For velocity bounds $\underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}}_k \leq \bar{\dot{\mathbf{q}}}$ and (3) yields

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta t \ddot{\mathbf{q}}_k \leq \begin{bmatrix} \bar{\dot{\mathbf{q}}} \\ -\underline{\dot{\mathbf{q}}} \end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix} \dot{\mathbf{q}}_{k-1}. \tag{5}$$

A closed-loop stable implementation of such a class of constraints is solved in Djeha et al. (2020).

Given the torque limits $\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}_k \leq \bar{\boldsymbol{\tau}}$ and the joint space dynamics (1) results in

$$\begin{bmatrix} I \\ -I \end{bmatrix} \mathbf{M}\ddot{\mathbf{q}}_k \leq \begin{bmatrix} B\bar{\boldsymbol{\tau}} \\ -B\underline{\boldsymbol{\tau}} \end{bmatrix} + \begin{bmatrix} I \\ -I \end{bmatrix} \left(J^\top \boldsymbol{f}_{k-1} - \mathbf{N}_{k-1}\right). \tag{6}$$

## 3.3 Contact space constraints

Assuming rigid contacts, we constrain zero contact acceleration $J_i\ddot{\mathbf{q}}_k + \dot{J}_i\dot{\mathbf{q}}_k = \mathbf{0}$. Another possible writing is by substituting (3)

$$J_i\ddot{\mathbf{q}}_k = -\dot{J}_i\dot{\mathbf{q}}_k = -\dot{J}_i\Delta t\ddot{\mathbf{q}}_k - \dot{J}_i\dot{\mathbf{q}}_{k-1},$$

which is rearranged as

$$(J_i + \dot{J}_i\Delta t)\ddot{\mathbf{q}}_k = -\dot{J}_i\dot{\mathbf{q}}_{k-1}. \tag{7}$$

We require the $i$-th contact wrench $\boldsymbol{W}_i = [\boldsymbol{f}^\top, \boldsymbol{\tau}^\top]^\top$ at the origin of its local contact frame to fulfill non-sliding constraints:

$$||\boldsymbol{f}_t|| < \mu \boldsymbol{f}_n \quad \text{and} \quad \boldsymbol{f}_n > 0$$

and, if needed, the center of pressure (CoP) within each contact area

$$||\boldsymbol{\tau}_t|| \in \mathrm{C}_{\text{area}} \quad \text{and} \quad \boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}$$

where $\mu$ denotes the friction coefficient, $\mathrm{C}_{\text{area}}$ the planar contact area. The half-plane representation of the above inequalities write:

$$\mathcal{C}_i \boldsymbol{W}_i \leq 0. \tag{8}$$

As long as (7) and (8) hold, the contact is sustained without slipping or taking off. The details of $\mathcal{C}_i$ are given in Caron et al. (2015) or similarly in (Grizzle et al. 2014, Sec. 3.3).

## 3.4 Centroidal space constraints

In the multi-contact setting, we define the Zero-tilting Moment Point (ZMP) as $\boldsymbol{z} \in \mathbb{R}^2$:

$$\boldsymbol{z} = \frac{\boldsymbol{n} \times \boldsymbol{\tau}_O}{\boldsymbol{n} \cdot \boldsymbol{f}_O} = \frac{\widehat{\boldsymbol{n}}\boldsymbol{\tau}_O}{\boldsymbol{n}^\top \boldsymbol{f}_O}. \tag{9}$$

The wrench $\boldsymbol{W}_O = [\boldsymbol{f}_O^\top, \boldsymbol{\tau}_O^\top]^\top$ denotes the net contact wrench measured in the inertial frame $\mathcal{F}_O$ whose origin is typically the center between the feet contacts (Caron et al. 2017). The ZMP is restricted by its feasible area $\boldsymbol{z} \in \mathcal{S}_s$ or $\boldsymbol{z} \in \mathcal{S}_{\boldsymbol{z}}$. In the simple case with coplanar contacts (for example double support stance on flat ground) the bounding polygon is defined by the support polygon $\mathcal{S}_s$, i.e., the convex hull of the ground contact points. Caron et al. (2017) derived a feasible *multi-contact ZMP area* $\mathcal{S}_{\boldsymbol{z}}$ for the more general case. Introducing $A_{\boldsymbol{z}}$ and $\mathbf{a}_{\boldsymbol{z}}$, the convex half-plane representation of $\mathcal{S}_{\boldsymbol{z}}$ becomes:

$$A_{\boldsymbol{z}}\boldsymbol{z}_k \leq \mathbf{a}_{\boldsymbol{z}}.$$

As pointed by Sugihara (2009), the ZMP is a measure of the force. Thus given the ZMP definition (9), we rearrange the above constraint with respect to the external wrench:

$$\mathcal{G}_{\boldsymbol{z}}\boldsymbol{W}_O \leq \mathbf{0}. \tag{10}$$

Using the unit vector $[0, 0, 1]^\top$ (the reverse direction of the gravity) as the surface normal $\boldsymbol{n} \in \mathbb{R}^3$, $\mathcal{G}_{\boldsymbol{z}}$ writes as: $\mathcal{G}_{\boldsymbol{z}} = \begin{bmatrix} -\mathbf{a}_{\boldsymbol{z}}\boldsymbol{n}^\top & A_{\boldsymbol{z}}\widehat{\boldsymbol{n}} \end{bmatrix}$.

When the robot is subject to external forces, Sugihara (2009) showed that the horizontal COM velocity $\dot{\boldsymbol{c}}_{x,y}$ has to remain inside a convex 2D polygon $\mathcal{S}_{\dot{\boldsymbol{c}}}$ in order to ensure the dynamic equilibrium. This area $\mathcal{S}_{\dot{\boldsymbol{c}}}$ relates to both the feasible ZMP area $\mathcal{S}_{\boldsymbol{z}}$ and the COM position $\boldsymbol{c}_{x,y}$. The constraint $\dot{\boldsymbol{c}}_{x,y} \in \mathcal{S}_{\dot{\boldsymbol{c}}}$ is transformed into the half-plane representation, introducing $\mathcal{G}_{\dot{\boldsymbol{c}}_{x,y}}$ and $\boldsymbol{h}_{\dot{\boldsymbol{c}}_{x,y}}$:

$$\mathcal{G}_{\dot{\boldsymbol{c}}_{x,y}}\dot{\boldsymbol{c}}_{x,y} \leq \boldsymbol{h}_{\dot{\boldsymbol{c}}_{x,y}}. \tag{11}$$

Due to the kinematic and actuation limits, the robot controller should minimize the angular momentum $\mathcal{L}_{\boldsymbol{c}} \in \mathbb{R}^3$ (Lee and Goswami 2012; Wiedebach et al. 2016). Suppose the angular momentum is bounded by $\mathcal{L}_{\boldsymbol{c}} \leq \bar{\mathcal{L}}_{\boldsymbol{c}}$, we take the angular part $A_{\omega\boldsymbol{c}}(\mathbf{q}_k)$ from the centroidal momentum matrix $A_{\boldsymbol{c}}(\mathbf{q}_k)$ proposed by Orin et al. (2013) to formulate:

$$A_{\omega\boldsymbol{c}}(\mathbf{q}_k)\dot{\mathbf{q}} \leq \bar{\mathcal{L}}_{\boldsymbol{c}}. \tag{12}$$

The Divergent Component of Motion (DCM) is widely used for gait generation, e.g. refer to the example by Englsberger et al. (2018). DCM is defined as a two-dimensional point $\boldsymbol{\xi} \in \mathbb{R}^2$ by

$$\boldsymbol{\xi} = \boldsymbol{c}_{x,y} + \frac{\dot{\boldsymbol{c}}_{x,y}}{\omega}, \tag{13}$$

where the scalar $\omega$ is defined by gravity and the COM position along the gravity direction: $\omega = \frac{g}{c_z}$. The feasible DCM support area $\mathcal{S}_{\boldsymbol{\xi}} = \mathcal{S}_{\boldsymbol{z}} \bigcap \mathcal{S}_{\boldsymbol{c}}$ equals the intersection of the *static-equilibrium COM area* $\mathcal{S}_{\boldsymbol{c}}$, and the multi-contact ZMP support area $\mathcal{S}_{\boldsymbol{z}}$. We calculate $\mathcal{S}_{\boldsymbol{\xi}}$, $\mathcal{S}_{\boldsymbol{c}}$ and $\mathcal{S}_{\boldsymbol{z}}$ following the derivation given by Caron et al. (2017). Fig. 11 shows an example captured from a real-time computation. Again, we represent the constraint $\boldsymbol{\xi} \in \mathcal{S}_{\boldsymbol{\xi}}$ using half-planes where $\mathcal{G}_{\boldsymbol{\xi}}$ and $\boldsymbol{h}_{\boldsymbol{\xi}}$ are constructed based on $\mathcal{S}_{\boldsymbol{\xi}}$:

$$\mathcal{G}_{\boldsymbol{\xi}} \boldsymbol{\xi} \leq \boldsymbol{h}_{\boldsymbol{\xi}}. \tag{14}$$

To our best knowledge, a real-time capable implementation of (11-14) has not yet been provided for QP controllers.

### 3.5 Impact-unaware whole-body QP controller

The whole-body controller is formulated as a quadratic program (QP) with desired task objectives and strict constraints (Bouyarmane et al. 2019). Prioritized tasks shall be met at best according to the associated cost function. Constraints shall be met strictly.

$$\begin{aligned}
\min_{\boldsymbol{\mu}} \quad & \sum_{i \in \mathcal{I}_o} w_i \|\boldsymbol{e}_i(\boldsymbol{\mu})\|^2 \\
\text{s.t.} \quad & \text{Joint Space constraints:} \\
& \text{Joint position: (4),} \\
& \text{Joint velocity: (5),} \\
& \text{Joint torque: (6),} \\
& \text{Contact Space constraints:} \\
& \text{Holding Contact position: (7),} \\
& \text{Fulfilling Contact wrench cone: (8),} \\
& \text{Centroidal space constraints:} \\
& \text{ZMP constraint: (10),} \\
& \text{COM velocity constraint: (11),} \\
& \text{Angular momentum constraint: (12),} \\
& \text{DCM constraint: (14),} \\
& \text{Other constraints, e.g., collision avoidance, etc.,}
\end{aligned} \tag{15}$$

where $\sum_{i \in \mathcal{I}_o} w_i \|\boldsymbol{e}_i(\boldsymbol{\mu})\|^2$ scalarizes multiple task objectives included in a set $\mathcal{I}_o$, $w_i$ weights the $i$-th task and $\boldsymbol{e}_i(\boldsymbol{\mu})$ is the task error, which is linear in terms of the QP decision variables $\boldsymbol{\mu}$, see the details in Bouyarmane et al. (2019). Both the generalized joint accelerations $\ddot{\boldsymbol{q}}_k$ and the generating vectors of the contact wrench cone $\boldsymbol{f}_\lambda(k)$ are optimized, hence $\boldsymbol{\mu}_k := \{\ddot{\mathbf{q}}_k, \boldsymbol{f}_\lambda(k)\}$.

### 3.6 Problem description

Impacts result in instantaneous jumps in the contacting end-effector's velocity, which affects many constraints expressed in terms of joint velocity such as robot state or kinematic bounds, ZMP, angular momentum, COM velocity, DCM, etc. As a result, the QP feasible search space, defined by the constraints, may suddenly shrink to an empty set. In this case, the QP is infeasible for the next control iteration. An illustrative toy example is given later in Sec. 5.6. Moreover, the controller must also be extended by constraints related to hardware-acceptable impacts obtained from mechanical stress limitations.

In order to deal with impact-induced state jumps, the main idea is to adjust the pre-impact contact velocity. Here, adjusting means that we shall find a compromise between task dictated impact-induced velocities (ideally, the maximum possible impact velocity if requiring the largest possible force jump achievable by the robot) and the QP controller feasibility set. In other words, we propose additional constraints that allow the controller to drive the robot to generate impacts that ensure a feasible closed-loop QP (i.e., non-empty search space including hardware impact limits) right after the impact (post-impact).

In the following, we explicitly show how to predict impact-induced state jumps (based on the assumption of rigid contact surfaces and the coefficient of restitution) in Sec. 4 and then explain how to formulate impact-aware constraints in Sec. 5.

## 4 Predicting impact-induced state jumps

We formulate an auxiliary quadratic optimization problem in Sec. (4.1-4.4) that predicts the impact effect. In order to apply task-purpose impact, we specify important impact-induced state jumps as functions of the QP decision variables in Sec. 4.5.

### 4.1 Contact velocity impact modeling

Let us consider a robot end-effector undergoing the desired impact task. We choose the following impact model: The scalar coefficient of restitution $c_\mathrm{r} > 0$ represents the relationship between the pre-impact end-effector velocity $\dot{\boldsymbol{x}}^- \in \mathbb{R}^3$ and the post-impact velocity $\dot{\boldsymbol{x}}^+ \in \mathbb{R}^3$ assuming no sliding

$$\dot{\boldsymbol{x}}^+_{k+1} = -c_\mathrm{r} P_{\boldsymbol{n}} \dot{\boldsymbol{x}}^-_{k+1} + (I - P_{\boldsymbol{n}}) \dot{\boldsymbol{x}}^-_{k+1}, \tag{16}$$

where $P_{\boldsymbol{n}} = \boldsymbol{n}\boldsymbol{n}^\top \in \mathbb{R}^{3\times 3}$ is the projection operator onto the contact normal $\boldsymbol{n} \in \mathbb{R}^3$. It is important to note that angular velocity terms are neglected. In this model, the pre-impact velocity is not necessarily aligned with the contact normal. The resulting three-dimensional end-effector velocity jump becomes*

$$\Delta \dot{\boldsymbol{x}}_{k+1} = \dot{\boldsymbol{x}}^+_{k+1} - \dot{\boldsymbol{x}}^-_{k+1} = \underbrace{-(1 + c_\mathrm{r}) P_{\boldsymbol{n}}}_{P_\Delta} \dot{\boldsymbol{x}}^-_{k+1}. \tag{17}$$

*The chosen impact model (16) can be replaced by experimental models. Replacing the resulting end-effector velocity jump in (17), our approach applies without further adaptation.

We approximate the pre-impact velocity (in the next control cycle) as:

$$\begin{aligned}\dot{\boldsymbol{x}}_{k+1}^{-} &= \dot{\boldsymbol{x}}_{k+1} = J_{k+1}\dot{\mathbf{q}}_{k+1} \\ &= (J_k + \dot{J}_k\Delta t + \ddot{J}_k\Delta t^2 + \ldots)(\dot{\mathbf{q}}_k + \ddot{\mathbf{q}}_k\Delta t) \\ &= \underbrace{J_k\dot{\mathbf{q}}_k}_{\dot{\boldsymbol{x}}_k} + \underbrace{J_k\ddot{\mathbf{q}}_k\Delta t + \dot{J}_k\dot{\mathbf{q}}_k\Delta t}_{\ddot{\boldsymbol{x}}_k\Delta t} + \underbrace{\dot{J}_k\ddot{\mathbf{q}}_k\Delta t^2 + \ldots}_{\approx\mathbf{0}}\end{aligned}$$

Neglecting all terms in $\Delta t^n$ where $n > 1$ since $\Delta t \leq 5$ ms, we expand (17) to express the jump $\Delta\dot{\boldsymbol{x}}_{k+1}$ with respect to the QP decision variable $\ddot{\mathbf{q}}_k$:

$$\Delta\dot{\boldsymbol{x}}_{k+1} = P_\Delta(J_k\dot{\mathbf{q}}_k + J_k\ddot{\mathbf{q}}_k\Delta t + \dot{J}_k\dot{\mathbf{q}}_k\Delta t), \tag{18}$$

where $\dot{\mathbf{q}}_k$ is obtained from the current robot state, and $\Delta t$ denotes the sampling period.

## 4.2 Contact force impact

Impacts result in impulses $\boldsymbol{\iota}$ characterized by contact force jumps $\Delta\boldsymbol{f}$ in end-effectors with established contacts, and velocity jumps $\Delta\dot{\boldsymbol{x}}$ in contact-free end-effectors.

Many complexities of rigid body contact dynamics can be avoided by the integral of contact forces $\boldsymbol{f}$ over the impact duration $\delta t$. Thus we define the impulse:

$$\boldsymbol{\iota} = \int_0^{\delta t} \boldsymbol{f}\, dt$$

whose unit is (N·s). Equivalently, on a discretized form, we can define the associated force jump

$$\Delta\boldsymbol{f} = \frac{1}{\delta t}\boldsymbol{\iota}. \tag{19}$$

There is no distinction between impulses $\boldsymbol{\iota}$ and finite force jump $\Delta\boldsymbol{f}$ over $\delta t$, see Chapter 48 by Wieber et al. (2016).

The stacked Jacobian $\boldsymbol{J} = [J_1^\top, \ldots, J_m^\top]^\top \in \mathbb{R}^{3m\times(n+6)}$ collects all the end-effector Jacobians that possibly experience an impulse. Considering the robot's joint space equations of motion (1) with the generalized force $B\boldsymbol{\tau} = \boldsymbol{J}^\top\boldsymbol{f}$, we obtain

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}\boldsymbol{J}^\top\boldsymbol{f} + \boldsymbol{\alpha}, \tag{20}$$

where $\boldsymbol{\alpha}$ denotes the *bias force* (i.e., free acceleration) that we do not consider for impact dynamics. When integrating (20) over the impact duration $\delta t$ the vector $\boldsymbol{\alpha}$ vanishes:

$$\Delta\dot{\mathbf{q}} = \mathbf{M}^{-1}\boldsymbol{J}^\top\boldsymbol{\iota}. \tag{21}$$

Lets us now map this relationship onto the space of all end-effectors by left-multiplying (20) by $\boldsymbol{J}$:

$$\Delta\dot{\boldsymbol{x}} = \Upsilon\boldsymbol{\iota}, \tag{22}$$

where $\Upsilon = \boldsymbol{J}\mathbf{M}^{-1}\boldsymbol{J}^\top$. Note that (22) characterizes the impulse propagation among the $m$ end-effectors. Let us expand (22), into three blocks:

$$\begin{bmatrix}\Delta\dot{\boldsymbol{x}}_{\sigma_{m_1}}\\ \Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}}\\ \Delta\dot{\boldsymbol{x}}_{\sigma_{m_3}}\end{bmatrix} = \begin{bmatrix}\Upsilon_{\sigma_{m_1}\sigma_{m_1}}, & \Upsilon_{\sigma_{m_1}\sigma_{m_2}}, & \Upsilon_{\sigma_{m_1}\sigma_{m_3}}\\ \Upsilon_{\sigma_{m_2}\sigma_{m_1}}, & \Upsilon_{\sigma_{m_2}\sigma_{m_2}}, & \Upsilon_{\sigma_{m_2}\sigma_{m_3}}\\ \Upsilon_{\sigma_{m_3}\sigma_{m_1}}, & \Upsilon_{\sigma_{m_3}\sigma_{m_2}}, & \Upsilon_{\sigma_{m_3}\sigma_{m_3}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\iota}_{\sigma_{m_1}}\\ \boldsymbol{\iota}_{\sigma_{m_2}}\\ \mathbf{0}\end{bmatrix}$$

where a block $\Upsilon_{\sigma_i\sigma_j} \in \mathbb{R}^{3m_i\times 3m_j}$ gathers matrices $\boldsymbol{J}_k\mathbf{M}^{-1}\boldsymbol{J}_l$ of the articulated bodies inertia matrices (when $k = l$) and the cross-coupling inertia when $k \neq l$; $\Delta\dot{\boldsymbol{x}}_{\sigma_{m_1}} \in \mathbb{R}^{3m_1}$ gathers the provisional velocity jumps of the $m_1$ already existing contacts which will have reaction impulse forces $\boldsymbol{\iota}_{\sigma_{m_1}} \in \mathbb{R}^{3m_1}$; $\Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}} \in \mathbb{R}^{3m_2}$ gathers the velocity jumps of the $m_2$ end-effectors that are supposed to contact with impulse force vector $\boldsymbol{\iota}_{\sigma_{m_2}} \in \mathbb{R}^{3m_2}$ at the next iteration; $\Delta\dot{\boldsymbol{x}}_{\sigma_{m_3}} \in \mathbb{R}^{3m_3}$ denotes the free end-effectors (i.e., no external force or impulse is acting on them), for which we aim to monitor the velocity jumps. Recall that $m = m_1 + m_2 + m_3$.

**Remark 4.1.** *The inverse of the operational space inertia used in (22) is constructed at the current time step $k$. We can compute a first-order approximation of predicted $\Upsilon(\mathbf{q}_{k+1})$ as follows:*

$$(J_k + \Delta t\dot{J}_k)(\mathbf{M}_k + \Delta t\dot{\mathbf{M}}_k)^{-1}(J_k^T + \Delta t\dot{J}_k^T)$$

*with the well-known decomposition $\dot{\mathbf{M}}_k = \mathbf{C}_k + \mathbf{C}_k^T$. Yet, since at the moment of contact our impact model assumes no change in the robot configuration, we consider that:*

$$\Upsilon(\mathbf{q}_{k+1}) \simeq \Upsilon(\mathbf{q}_k).$$

**Remark 4.2.** *The matrix $\Upsilon$ is further reduced to $\Omega \in \mathbb{R}^{m\times(m_1+m_2)}$*

$$\Omega = \begin{bmatrix}\Upsilon_{\sigma_{m_1}\sigma_{m_1}}, & \Upsilon_{\sigma_{m_1}\sigma_{m_2}}\\ \Upsilon_{\sigma_{m_2}\sigma_{m_1}}, & \Upsilon_{\sigma_{m_2}\sigma_{m_2}}\\ \Upsilon_{\sigma_{m_3}\sigma_{m_1}}, & \Upsilon_{\sigma_{m_3}\sigma_{m_2}}\end{bmatrix}.$$

*since all the blocks in $\Upsilon$ related to the free end-effectors will be nilled by the zero impulse force:*

$$\begin{bmatrix}\Delta\dot{\boldsymbol{x}}_{\sigma_{m_1}}\\ \Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}}\\ \Delta\dot{\boldsymbol{x}}_{\sigma_{m_3}}\end{bmatrix} = \Omega\begin{bmatrix}\boldsymbol{\iota}_{\sigma_{m_1}}\\ \boldsymbol{\iota}_{\sigma_{m_2}}\end{bmatrix}. \tag{23}$$

## 4.3 Centroidal momentum conservation

It is important to note that with redundant robots there exist multiple candidates of joint velocity jumps $\Delta\dot{\mathbf{q}}$ that can fulfill $\Delta\dot{\boldsymbol{x}} = \boldsymbol{J}\Delta\dot{\mathbf{q}}$ in (22). However, each $\dot{\mathbf{q}}$ determines a unique centroidal momenta $\boldsymbol{h_c} \in \mathbb{R}^6$. Thus we introduce the centroidal space impact dynamics separately to specify the centroidal momenta jump $\Delta\boldsymbol{h_c}$ uniquely.

The aggregation of external forces determines the derivative of the centroidal momentum $\boldsymbol{h_c}$:

$$\dot{\boldsymbol{h}}_{\boldsymbol{c}} = \begin{bmatrix}\dot{P}_{\boldsymbol{c}}\\ \dot{\mathcal{L}}_{\boldsymbol{c}}\end{bmatrix} = \begin{bmatrix}M\boldsymbol{g}\\ 0\end{bmatrix} + \sum_{i=1}^{m} J_{\boldsymbol{cp}_i}^\top\boldsymbol{f}_i, \tag{24}$$

where the scalar $M > 0$ denotes the total mass of the robot, $\boldsymbol{c} \in \mathbb{R}^3$ denotes the center of mass (CoM) and $\boldsymbol{p}_i \in \mathbb{R}^3$ denotes the $i$-th contact location (the CoP position).

The centroidal frame external force $\boldsymbol{f}_i = R_{\boldsymbol{cp}_i}\boldsymbol{f}_i^m \in \mathbb{R}^3$ is calculated from the measured force $\boldsymbol{f}_i^m$. The Jacobian $J_{\boldsymbol{cp}_i}^\top \in \mathbb{R}^{6\times 3}$ that calculates the induced wrench in the centroidal frame $\mathcal{F}_{\boldsymbol{c}}$ is given by: $J_{\boldsymbol{cp}_i}^\top := [I^\top, \overrightarrow{\boldsymbol{cp}_i}\times^\top]^\top$.

Integrating (24) over the impact duration $\delta t$ and assuming $m_1$ established contacts and $m_2$ impacts, we obtain the centroidal momentum jump as the sum of the external

impulses:

$$\begin{bmatrix}\Delta P_{\boldsymbol{c}}\\ \Delta \mathcal{L}_{\boldsymbol{c}}\end{bmatrix} = \sum_{i\in\sigma_{m_1}} J_{\boldsymbol{cp}_i}^{\top}\boldsymbol{\iota}_i + \sum_{i\in\sigma_{m_2}} J_{\boldsymbol{cp}_i}^{\top}\boldsymbol{\iota}_i, \tag{25}$$

where $[\Delta P_{\boldsymbol{c}}^{\top}, \Delta \mathcal{L}_{\boldsymbol{c}}^{\top}]^{\top} = A_{\boldsymbol{c}}\Delta\dot{\mathbf{q}} \in \mathbb{R}^6$ employing the centroidal momentum matrix (CMM) $A_{\boldsymbol{c}} \in \mathbb{R}^{6\times(n+6)}$. The constant term $M\boldsymbol{g}\delta t$ being negligible w.r.t other impulses. The compact form yields

$$A_{\boldsymbol{c}}\Delta\dot{\mathbf{q}} = S\begin{bmatrix}\boldsymbol{\iota}_{\sigma_{m_1}}\\ \boldsymbol{\iota}_{\sigma_{m_2}}\end{bmatrix}, \tag{26}$$

where $S \in \mathbb{R}^{6\times(m_1+m_2)}$ gathers the associated Jacobian matrices.

### 4.4 Impulse distribution quadratic program

We define the optimization variable as:

$$\boldsymbol{u} = [\Delta\dot{\mathbf{q}}^{\top}, \boldsymbol{\iota}_{\sigma_{m_1}}^{\top}, \boldsymbol{\iota}_{\sigma_{m_2}}^{\top}]^{\top} = [\Delta\dot{\mathbf{q}}^{\top}, \boldsymbol{\iota}_{\sigma_e}^{\top}]^{\top},$$

where $\sigma_e = \sigma_{m_1} \cup \sigma_{m_2}$, thus $\boldsymbol{\iota}_{\sigma_e} \in \mathbb{R}^{(m_1+m_2)}$, and all the impulses are defined in the inertial frame†. Assuming that the robot is applying one or multiple impacts along with given directions in the next control iteration, we formulate the Impulse Distribution Quadratic Program (IDQP) that returns the optimal $\boldsymbol{u}^*$:

$$\begin{aligned}
\min_{\boldsymbol{u}} \quad & \frac{1}{2}(\Delta\dot{\mathbf{q}}^{\top}\mathbf{M}^{\top}\gamma_a\mathbf{M}\Delta\dot{\mathbf{q}} + \boldsymbol{\iota}_{\sigma_e}^{\top}\gamma_b\boldsymbol{\iota}_{\sigma_e})\\
\text{s.t.} \quad & \text{Operational space dynamics:}\\
& J\Delta\dot{\mathbf{q}} = \Omega\boldsymbol{\iota}_{\sigma_e},\\
& \text{Centroidal momentum conservation:}\\
& A_{\boldsymbol{c}}\Delta\dot{\mathbf{q}} = S\boldsymbol{\iota}_{\sigma_e},\\
& \text{Controlled impacts:}\\
& J_i\Delta\dot{\mathbf{q}} = \Delta\dot{\boldsymbol{x}}_i \text{ for } i \in \sigma_{m_1}.
\end{aligned} \tag{27}$$

where scalar weights $\gamma_a, \gamma_b > 0$ impose the relative importance between the two quadratic costs $\|\mathbf{M}\Delta\dot{\mathbf{q}}\|^2$ and $\|\boldsymbol{\iota}_{\sigma_e}\|^2$. In general, these two costs cannot be both wholly minimized, i.e., they conflict with each other. We chose the inertia matrix in the cost function, to prioritize among the task-space impulse $\boldsymbol{\iota}$ and the joint space impulse equivalent $\mathbf{M}\Delta\dot{\mathbf{q}}$. We reformulate the IDQP in matrix form

$$\begin{aligned}
\min_{\boldsymbol{u}} \quad & \frac{1}{2}\boldsymbol{u}^{\top}H\boldsymbol{u}\\
\text{s.t.} \quad & B\boldsymbol{u} = \mathbf{b}
\end{aligned} \tag{28}$$

with block matrices

$$B = \begin{bmatrix}\boldsymbol{J}, & -\Omega\\ A_{\boldsymbol{c}}, & -S\\ \boldsymbol{J}_{\sigma_{m_2}}, & 0\end{bmatrix} \text{ and } \quad \boldsymbol{b} = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}}\end{bmatrix},$$

where we collect the Jacobians of the $m_2$ end-effectors that are applying impacts as:

$$\boldsymbol{J}_{\sigma_{m_2}} = [J_1^{\top}, \ldots, J_{m_2}^{\top}]^{\top} \in \mathbb{R}^{3m_2\times(n+6)}.$$

Given the impact dynamics model (18), the end-effector velocity jumps $\Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}} \in \mathbb{R}^{3m_2}$ are represented as the decision variable of (15), i.e., $\ddot{\mathbf{q}}$. The weighting matrix $H \in \mathbb{R}^{(n+3(m_1+m_2))\times(n+3(m_1+m_2))}$ is

$$H = \begin{bmatrix}\gamma_a\mathbf{M}^{\top}\mathbf{M} & 0\\ 0 & \gamma_b I\end{bmatrix}.$$

The solution to the IDQP (28), i.e., the joint velocity jump $\Delta\dot{\mathbf{q}}^*$ and the impulse $\boldsymbol{\iota}_{\sigma_e}^*$ at the time step $t_{k+1}$, can be obtained analytically:

$$\boldsymbol{u}^* = \begin{bmatrix}\Delta\dot{\mathbf{q}}^*\\ \boldsymbol{\iota}_{\sigma_e}^*\end{bmatrix} = B^{H,\dagger}\,\mathbf{b} = \begin{bmatrix}K_{\Delta\dot{\mathbf{q}}}\\ K_{\boldsymbol{\iota}}\end{bmatrix}\Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}} \tag{29}$$

where $B^{H,\dagger}$ is the $H$-weighted pseudo-inverse of $B$:

$$B^{H,\dagger} = H^{-1}B^{\top}(BH^{-1}B^{\top})^{-1} = \begin{bmatrix}\cdots, & K_{\Delta\dot{\mathbf{q}}}\\ \cdots, & K_{\boldsymbol{\iota}}\end{bmatrix}.$$

Only the right $3m_2$ columns (related to the $3m_2$ lower, nonzero elements of $\mathbf{b}$) of $B^{H,\dagger}$ are of interest. Note that $K_{\Delta\dot{\mathbf{q}}}$, $K_{\boldsymbol{\iota}}$ are decomposed into individual end-effector contributions:

$$K_{\Delta\dot{\mathbf{q}}} = [K_{\Delta\dot{\mathbf{q}}_1}, K_{\Delta\dot{\mathbf{q}}_2}, \cdots, K_{\Delta\dot{\mathbf{q}}_{m_2}}] \in \mathbb{R}^{(n+6)\times 3m_2}, \tag{30}$$

$$K_{\boldsymbol{\iota}} = \begin{bmatrix}K_{1,1}, & \cdots, & K_{1,m_2}\\ \vdots & & \vdots\\ K_{(m_1+m_2),1}, & \cdots, & K_{(m_1+m_2),m_2}\end{bmatrix}, \tag{31}$$

where $K_{\boldsymbol{\iota}} \in \mathbb{R}^{3(m_1+m_2)\times 3m_2}$.

**Example 1.** IDQP (28) solution visualization. *Assuming the robot applies an impact with the right palm, Fig. 2 visualizes the predicted force jumps (36) (blue arrows) at the right palm and the established contacts, i.e., the two feet.*

*The left arm is a free limb without contacts or impact. We can predict the end-effector velocity jump (37) (yellow arrow), which mitigates impulses on the other end-effectors.*

**Remark 4.3.** *The IDQP (28) is an estimator that does not restrict any contact to a fixed position. Thus we can observe $\Delta\dot{\boldsymbol{x}}_i \neq \mathbf{0}$ for $i = 1, \ldots, m_1$ regardless of whether an end-effector has established contact or not.*

*To sustain established contacts, we construct the contact wrench cone constraint (48) (which is the impact-aware version of (8) to be introduced in Sec. 5.3) using the predicted $\boldsymbol{\iota}_{\sigma_{m_1}}^*$. Depending on the feasibility of (48), the QP controller can reduce the contact velocity and hence also the propagated impulse $\boldsymbol{\iota}_{\sigma_{m_1}}^*$.*

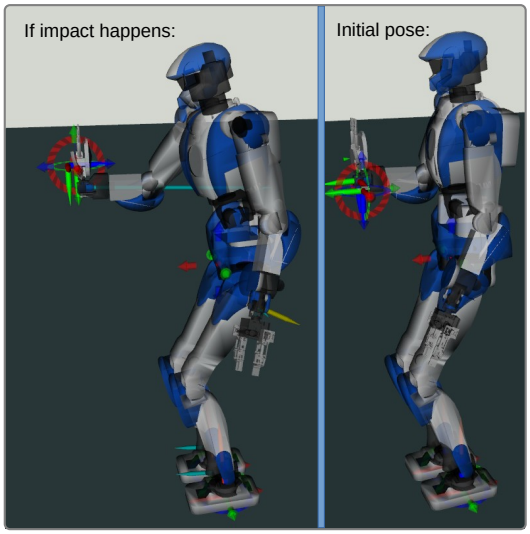
**Remark 4.4.** *It is easy to see that*

$$[B, 0]\begin{bmatrix}\boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}\\ \Upsilon^{-1}\boldsymbol{J}\boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}\end{bmatrix}\Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}} = \underbrace{[B, 0]\begin{bmatrix}\cdots, & \boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}\\ \cdots, & \Upsilon^{-1}\boldsymbol{J}\boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}\end{bmatrix}}_{[B,0]^{H,\dagger}}\mathbf{b}$$

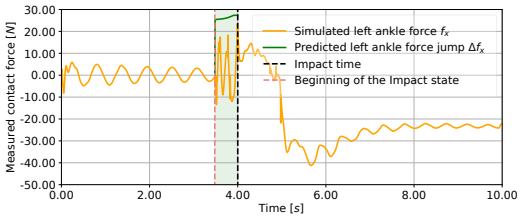
*where $\boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}$ is a $W$-weighted pseudo-inverse of $\boldsymbol{J}_{\sigma_{m_2}}$. Hence, a special solution for (28) is given by*

$$\begin{bmatrix}\boldsymbol{u}^*\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\Delta\dot{\mathbf{q}}^*\\ \boldsymbol{\iota}_{\sigma_e}^*\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}\\ \Upsilon^{-1}\boldsymbol{J}\boldsymbol{J}_{\sigma_{m_2}}^{W,\dagger}\end{bmatrix}\Delta\dot{\boldsymbol{x}}_{\sigma_{m_2}}, \tag{32}$$

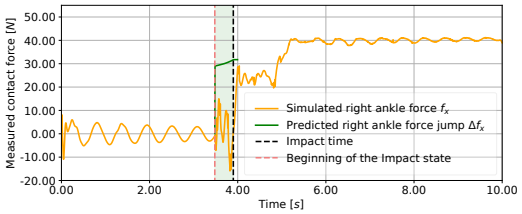
† The centroidal frame keeps the same orientation of the inertial frame. Thus we can use $\boldsymbol{\iota}$ for both (22) and (26).




(a) Illustration of Example. 1.




(b) Left ankle $x-$axis force jump of Example. 1




(c) Right ankle $x-$axis force jump of Example. 1

**Figure 2.** Fig. 2(a) visualizes the force jumps (light-blue) (36) and end-effector velocity jump (light-yellow) (37) obtained from the IDQP (28). The simulated force plots in Fig. 2(b) and Fig. 2(c) verify the predicted force jumps on the feet. We selected $\gamma_a = 1$ and $\gamma_b = 1$ for the optimization objective.

*where the zero impulse for the free end-effectors are explicit in order to invert the square $\Upsilon$ instead of pseudo-inverting the non-square $\Omega$. These predictions $\Delta\dot{\mathbf{q}}^*$ and $\boldsymbol{\iota}^*_{\sigma_e}$ correspond to (29) when choosing the extreme relative importance $\gamma_a = \infty$, $\gamma_b = 1$ and the relationship*

$$H = \begin{bmatrix} \infty W & 0 \\ 0 & I \end{bmatrix}.$$

*Setting a weight to infinity in a multi-objective optimization with soft weights is equivalent to imposing a strict nullspace-based hierarchy. The special solution (32) strictly minimizes the cost $\Delta\dot{\mathbf{q}}^\top W \Delta\dot{\mathbf{q}}$. The secondary cost $\boldsymbol{\iota}^\top_{\sigma_e}\boldsymbol{\iota}_{\sigma_e}$ is optimized without compromising the primary objective.*

*Note that $\boldsymbol{J}\boldsymbol{J}^{W,\dagger}_{\sigma_{m_2}} \neq I$ in the general case*[‡]*. Further note that*

$$\Upsilon^{-1}\boldsymbol{J}\boldsymbol{J}^{W,\dagger}_{\sigma_{m_2}} = \underbrace{(\boldsymbol{J}\mathbf{M}^{-1}\boldsymbol{J}^\top)^{-1}}\boldsymbol{J}W\boldsymbol{J}^\top_{\sigma_{m_2}}\underbrace{(\boldsymbol{J}_{\sigma_{m_2}}W\boldsymbol{J}^\top_{\sigma_{m_2}})^{-1}}$$

*in (32) involves two inverse-operations, however, each of them is much smaller than the inversion of $BH^{-1}B^T$ in (29).*

**Remark 4.5.** *Consider the special case $m_2 > 0$, $m_1 = 0$ and $m_3 = 0$, for example a fixed-base manipulator without existing contacts. Accordingly, $\boldsymbol{J}\boldsymbol{J}^{W,\dagger}_{\sigma_{m_2}} = I$ holds. The solutions obtained by (29) and (32) are equivalent given the relationship*

$$H = \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix}.$$

*The relative importance between costs $\gamma_a$, $\gamma_b$ is irrelevant. In other words, the two costs do not conflict with each other and are minimized simultaneously.*

## 4.5 Whole-body impact effects

Here, we summarize all the essential state jumps. In a generic form, the quantity jump $\Delta\boldsymbol{\lambda}_{k+1}$ relates to joint accelerations $\ddot{\mathbf{q}}_k$ and is linearly decomposed as

$$\Delta\boldsymbol{\lambda}_{k+1} = \mathcal{J}_{\Delta\boldsymbol{\lambda}}\ddot{\mathbf{q}}_k\Delta t + \mathcal{C}_{\Delta\boldsymbol{\lambda}}\dot{\mathbf{q}}_k, \tag{33}$$

where the matrix $\mathcal{J}_{\Delta\boldsymbol{\lambda}}$ and $\mathcal{C}_{\Delta\boldsymbol{\lambda}}$ incorporates the impact model and the impulse propagation through the kinematic tree. The coefficient of restitution contributes to $\mathcal{J}_{\Delta\boldsymbol{\lambda}}$ and $\mathcal{C}_{\Delta\boldsymbol{\lambda}}$ as a scalar multiplier by $(1 + c_\mathrm{r})$. Hence, higher coefficients lead to higher jumps. We specify the matrix $\mathcal{J}_{\Delta\boldsymbol{\lambda}}$ and $\mathcal{C}_{\Delta\boldsymbol{\lambda}}$ on a case-by-case basis in the rest of this section. These expressions are used in Sec. 5 to formulate impact-aware constraints.

### 4.5.1 Joint space state jumps

*Joint position jumps* According to Zheng and Hemami (1985); Konno et al. (2011); Wang and Kheddar (2019), there are no, or negligible joint position jumps during the impact transition $\Delta\mathbf{q}_{k+1} \simeq \mathbf{0}$.

*Joint velocity jumps* The joint velocity jump given in (29) is reformulated in terms of joint accelerations by substituting (18)

$$\begin{aligned}\Delta\dot{\mathbf{q}}_{k+1} = &\underbrace{\sum_{i=1}^{m_2} K_{\Delta\dot{\mathbf{q}}_i}P_{\Delta_i}J_i(k)}_{\mathcal{J}_{\Delta\dot{\mathbf{q}}}}\ddot{\mathbf{q}}_k\Delta t \\ &+ \underbrace{\sum_{i=1}^{m_2} K_{\Delta\dot{\mathbf{q}}_i}P_{\Delta_i}\left(J_i(k) + \dot{J}_i(k)\Delta t\right)}_{\mathcal{C}_{\Delta\dot{\mathbf{q}}}}\dot{\mathbf{q}}_k.\end{aligned} \tag{34}$$

*Joint torque jumps* The whole-body joint torque jump relates to the end-effector force jumps $\Delta\boldsymbol{\tau} = J^\top\Delta\boldsymbol{f}$. Hence aggregating the end-effector force jump $\Delta\boldsymbol{f}_{r,k+1}$ defined by

[‡]If and only if there is no pre-existing contacts and no free limbs, then $\boldsymbol{J}\boldsymbol{J}^{W,\dagger}_{\sigma_{m_2}} = I$, because $\boldsymbol{J}_{\sigma_{m_2}}\boldsymbol{J}^{W,\dagger}_{\sigma_{m_2}} = I$ always holds.

(36), we predict the joint torque jumps as:

$$\Delta\boldsymbol{\tau}_{k+1} = \underbrace{\frac{1}{\delta t}\sum_{r=1}^{m} J_r^\top \sum_{i=1}^{m_2} K_{r,i} P_{\Delta_i} J_i \, \ddot{\mathbf{q}}_k \Delta t}_{\mathcal{J}_{\Delta\boldsymbol{\tau}}} + \underbrace{\sum_{r=1}^{m} J_r^\top \sum_{i=1}^{m_2} K_{r,i} P_{\Delta_i} \left(J_i(k) + \dot{J}_i(k)\Delta t\right) \dot{\mathbf{q}}_k}_{\mathcal{C}_{\Delta\boldsymbol{\tau}}} \tag{35}$$

#### 4.5.2 Contact space state jumps

*End-effector force jumps* Substituting (18) in the predicted impulse (29), we formulate the force jump, i.e., defined by (19), of the $r$-th end-effector, in terms of joint accelerations:

$$\Delta\boldsymbol{f}_{r,k+1} = \underbrace{\frac{1}{\delta t}\sum_{i=1}^{m_2} K_{r,i} P_{\Delta_i} J_i \, \ddot{\mathbf{q}}_k \Delta t}_{\mathcal{J}_{\Delta\boldsymbol{f}_r}} + \underbrace{\frac{1}{\delta t}\sum_{i=1}^{m_2} K_{r,i} P_{\Delta_i} \left(J_i(k) + \dot{J}_i(k)\Delta t\right) \dot{\mathbf{q}}_k}_{\mathcal{C}_{\Delta\boldsymbol{f}_r}}. \tag{36}$$

*End-effector velocity Jumps* Given the joint velocity Jump (34), the $r$-th end-effector velocity jump is readily available:

$$\begin{aligned}\Delta\dot{\boldsymbol{x}}_{r,k+1} &= J_r \Delta\dot{\mathbf{q}}_{k+1} \\ &= \underbrace{J_r \mathcal{J}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{J}_{\Delta\dot{\boldsymbol{x}}}} \ddot{\mathbf{q}}_k \Delta t + \underbrace{J_r \mathcal{C}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{C}_{\Delta\dot{\boldsymbol{x}}}} \dot{\mathbf{q}}_k.\end{aligned} \tag{37}$$

#### 4.5.3 Centroidal space state jumps

*Angular momentum jump* Given the joint velocity jump $\Delta\dot{\mathbf{q}}$ by (34) and the angular part $A_{\omega\boldsymbol{c}}(\mathbf{q}_k)$ of the centroidal momentum matrix $A_{\boldsymbol{c}}(\mathbf{q}_k)$, we obtain $\Delta\mathcal{L}_{\boldsymbol{c}}(\mathbf{q}_{k+1})$ as:

$$\begin{aligned}\Delta\mathcal{L}_{\boldsymbol{c}}(\mathbf{q}_{k+1}) &= A_{\omega\boldsymbol{c}}(\mathbf{q}_k)\Delta\dot{\mathbf{q}}_{k+1}, \\ &= \underbrace{A_{\omega\boldsymbol{c}}(\mathbf{q}_k)\mathcal{J}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{J}_{\Delta\mathcal{L}_{\boldsymbol{c}}}} \ddot{\mathbf{q}}_k \Delta t + \underbrace{A_{\omega\boldsymbol{c}}(\mathbf{q}_k)\mathcal{C}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{C}_{\Delta\mathcal{L}_{\boldsymbol{c}}}} \dot{\mathbf{q}}_k,\end{aligned} \tag{38}$$

and we approximated $A_{\omega\boldsymbol{c}}(\mathbf{q}_{k+1}) \simeq A_{\omega\boldsymbol{c}}(\mathbf{q}_k)$.

*Planar COM Velocity Jump* If we are only interested in the planar COM velocity jump $\Delta\dot{\boldsymbol{c}}_{x,y} \in \mathbb{R}^2$, i.e., the $x$ and $y$ directions, we can obtain

$$\begin{aligned}\Delta\dot{\boldsymbol{c}}_{x,y}(k+1) &= \frac{1}{M} A_{\boldsymbol{vc}}(\mathbf{q}_k)\Delta\dot{\mathbf{q}}_{k+1}, \\ &= \underbrace{\frac{1}{M} A_{\boldsymbol{vc}}(\mathbf{q}_k)\mathcal{J}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{J}_{\Delta\dot{\boldsymbol{c}}_{x,y}}} \ddot{\mathbf{q}}_k \Delta t + \underbrace{\frac{1}{M} A_{\boldsymbol{vc}}(\mathbf{q}_k)\mathcal{C}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{C}_{\Delta\dot{\boldsymbol{c}}_{x,y}}} \dot{\mathbf{q}}_k,\end{aligned} \tag{39}$$

where $A_{\boldsymbol{vc}}$ takes the corresponding rows from $A_{\boldsymbol{c}}$, and we use the same approximation as (38).

*ZMP Jump* We use the predicted force jumps (36) to calculate the resultant wrench:

$$\begin{aligned}\Delta\boldsymbol{W}_O(k+1) &= \sum_{i=1}^{m_1} J_{O\boldsymbol{p}_i}^\top \boldsymbol{f}_i + \sum_{i=1}^{m_2} J_{O\boldsymbol{p}_i}^\top \boldsymbol{f}_i, \\ &= \underbrace{\sum_{i=1}^{m_1+m_2} J_{O\boldsymbol{p}_i}^\top \mathcal{J}_{\Delta\boldsymbol{f}_i}}_{\mathcal{J}_{\Delta\boldsymbol{W}_O}} \ddot{\mathbf{q}}_k \Delta t + \underbrace{\sum_{i=1}^{m_1+m_2} J_{O\boldsymbol{p}_i}^\top \mathcal{C}_{\Delta\boldsymbol{f}_i}}_{\mathcal{C}_{\Delta\boldsymbol{W}_O}} \dot{\mathbf{q}}_k,\end{aligned} \tag{40}$$

where the Jacobian $J_{O\boldsymbol{p}_i}^\top$ is defined as: $J_{O\boldsymbol{p}_i}^\top = [I^\top, \overrightarrow{O\boldsymbol{p}_i}\times^\top]^\top$. Injecting the wrench jump $\Delta\boldsymbol{W}_O = [\Delta\boldsymbol{f}_O^\top, \Delta\boldsymbol{\tau}_O^\top]^\top$ into the ZMP definition (9), we obtain the ZMP jump:

$$\Delta\boldsymbol{z}_{k+1} = \frac{\hat{\boldsymbol{n}}\Delta\boldsymbol{\tau}_O}{\boldsymbol{n}^\top \boldsymbol{f}_O + \Delta\boldsymbol{f}_O}. \tag{41}$$

*DCM Jump* Using the DCM definition (13) and the predicted $\Delta\dot{\boldsymbol{c}}_{x,y_{k+1}}$ (39), we can predict the impact-induced DCM jump $\Delta\boldsymbol{\xi}$ as:

$$\begin{aligned}\Delta\boldsymbol{\xi}_{k+1} &= \Delta\boldsymbol{c}_{x,y}(k+1) + \frac{\Delta\dot{\boldsymbol{c}}_{x,y}(k+1)}{\omega} \\ &= \frac{1}{\omega}\mathcal{J}_{\Delta\dot{\boldsymbol{c}}_{x,y}}\Delta\dot{\mathbf{q}}_{k+1} \\ &= \underbrace{\frac{1}{\omega}\mathcal{J}_{\Delta\dot{\boldsymbol{c}}_{x,y}}\mathcal{J}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{J}_{\Delta\boldsymbol{\xi}}} \ddot{\mathbf{q}}_k \Delta t + \underbrace{\frac{1}{\omega}\mathcal{J}_{\Delta\dot{\boldsymbol{c}}_{x,y}}\mathcal{C}_{\Delta\dot{\mathbf{q}}}}_{\mathcal{C}_{\Delta\boldsymbol{\xi}}} \dot{\mathbf{q}}_k.\end{aligned} \tag{42}$$

Note that the first summand $\Delta\boldsymbol{c}_{x,y}(k+1)$ can be neglected, but the second summand $\frac{\Delta\dot{\boldsymbol{c}}_{x,y}(k+1)}{\omega}$ instead is of importance for impact-awareness.

## 5 Impact-aware QP control based on Predicted State Jumps

Based on the predicted impact-induced state jumps, we systematically propose novel impact-aware constraints that guarantee the feasibility of the QP controller after the impact event. These new constraints extend the constraints formulation detailed in Sec. 3.

### 5.1 Constraining generic quantity jumps

We here propose to constrain the post-impact state of the generic quantity $\boldsymbol{\lambda}$

$$D\boldsymbol{\lambda}_{k+1}^{+} \leq \bar{\boldsymbol{\lambda}} \tag{43}$$

with the matrix $D$ representing half-planes and the offset vector $\bar{\boldsymbol{\lambda}}$ representing the upper and lower bounds. Substituting the *Euler forward method* $\boldsymbol{\lambda}_{k+1}^{-} = \boldsymbol{\lambda}_k + \dot{\boldsymbol{d}}_k\Delta t$ and the impact model $\boldsymbol{\lambda}_{k+1}^{+} = \boldsymbol{\lambda}_{k+1}^{-} + \Delta\boldsymbol{\lambda}_{k+1}$ we obtain

$$D\left(\boldsymbol{\lambda}_k + \dot{\boldsymbol{d}}_k\Delta t + \Delta\boldsymbol{\lambda}_{k+1}\right) \leq \bar{\boldsymbol{\lambda}}. \tag{44}$$

We reformulate the constraint regarding the QP decision variable $\ddot{\mathbf{q}}_k$ by using the generic expression for the impact-induced jump $\Delta\boldsymbol{\lambda}_{k+1}$ (33)

$$D\mathcal{J}_{\Delta\boldsymbol{\lambda}}\ddot{\mathbf{q}}_k\Delta t \leq \bar{\boldsymbol{\lambda}} - D\left(\boldsymbol{\lambda}_k + \dot{\boldsymbol{d}}_k\Delta t + \mathcal{C}_{\Delta\boldsymbol{\lambda}}\dot{\mathbf{q}}_k\right).$$

Note that this constraint is easily adapted such that only actuated joints $\boldsymbol{\theta}$ are constrained. Further note that it is typically challenging to obtain an accurate measurement of $\dot{\boldsymbol{d}}_k$. Therefore, we decided to approximate[§] the pre-impact state as $\boldsymbol{\lambda}_{k+1}^{-} \approx \boldsymbol{\lambda}_k$, and consequently, ignore the

[§]In the case of the impact-aware joint velocity constraint (with $\boldsymbol{\lambda} := \dot{\mathbf{q}}$) we choose $\dot{\boldsymbol{d}}_k$ as the QP decision variable $\ddot{\mathbf{q}}_k$ instead of using a direct measurement of joint accelerations.

term $\dot{\boldsymbol{d}}_k \Delta t \approx \mathbf{0}$:

$$D\mathcal{J}_{\Delta\boldsymbol{\lambda}}\ddot{\mathbf{q}}_k\Delta t \leq \bar{\boldsymbol{\lambda}} - D\left(\boldsymbol{\lambda}_k + \mathcal{C}_{\Delta\boldsymbol{\lambda}}\dot{\mathbf{q}}_k\right). \tag{45}$$

In the following subsections, we will consider a multitude of quantities: joint velocity, joint torque, fulfilling contact wrench cone, centroidal angular momentum, COM velocity, ZMP, and DCM. All these derivations will follow the same generic concept presented here. In the future, we can constrain more quantities easily in an impact-aware manner following the same approach.

## 5.2 Constraining joint space jumps

*5.2.1 Joint velocity* Choosing joint velocities as quantity $\boldsymbol{\lambda} := \dot{\mathbf{q}}$ in (45) with $D := [I, -I]^T$ and $\bar{\boldsymbol{\lambda}} := \begin{bmatrix}\bar{\dot{\mathbf{q}}} \\ -\underline{\dot{\mathbf{q}}}\end{bmatrix}$ yields:

$$\begin{bmatrix} I \\ -I \end{bmatrix}\left(I + \mathcal{J}_{\Delta\dot{\mathbf{q}}}\right)\ddot{\mathbf{q}}_k\Delta t \leq \begin{bmatrix}\bar{\dot{\mathbf{q}}} \\ -\underline{\dot{\mathbf{q}}}\end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix}\left(I + \mathcal{C}_{\Delta\dot{\mathbf{q}}}\right)\dot{\mathbf{q}}_k \tag{46}$$

*5.2.2 Joint torque* Injecting joint torques $\boldsymbol{\lambda} := \boldsymbol{\tau}$ in (45) with $D := [I, -I]^T$ and $\bar{\boldsymbol{\lambda}} := \begin{bmatrix}\bar{\boldsymbol{\tau}} \\ -\underline{\boldsymbol{\tau}}\end{bmatrix}$ holds:

$$\begin{bmatrix} I \\ -I \end{bmatrix}\mathcal{J}_{\Delta\boldsymbol{\tau}}\ddot{\mathbf{q}}_k\Delta t \leq \begin{bmatrix}\bar{\boldsymbol{\tau}} \\ -\underline{\boldsymbol{\tau}}\end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix}\left(\boldsymbol{\tau}_k + \mathcal{C}_{\Delta\boldsymbol{\tau}}\dot{\mathbf{q}}_k\right). \tag{47}$$

**Remark 5.1.** *In the experiments with the position-controlled robot HRP4, the torque measurement $\boldsymbol{\tau}_k$ is too noisy to be used in a constraint. Thus in practice, we applied:*

$$\begin{bmatrix} I \\ -I \end{bmatrix}\mathcal{J}_{\Delta\boldsymbol{\tau}}\ddot{\mathbf{q}}_k\Delta t \leq \begin{bmatrix}\Delta\bar{\boldsymbol{\tau}} \\ -\Delta\underline{\boldsymbol{\tau}}\end{bmatrix} - \begin{bmatrix} I \\ -I \end{bmatrix}\mathcal{C}_{\Delta\boldsymbol{\tau}}\dot{\mathbf{q}}_k$$

*which physically means that we are constraining $\Delta\underline{\boldsymbol{\tau}} \leq \Delta\boldsymbol{\tau}_{k+1} \leq \Delta\bar{\boldsymbol{\tau}}$. In our experiments, we selected $\Delta\bar{\boldsymbol{\tau}} = 0.4\bar{\boldsymbol{\tau}}$ and $\Delta\underline{\boldsymbol{\tau}} = 0.4\underline{\boldsymbol{\tau}}$.*

## 5.3 Fulfilling contact wrench cone

Considering in (45) all the $m_1$ established contacts using $\boldsymbol{\lambda} := \boldsymbol{W}$, $\bar{\boldsymbol{\lambda}} := \mathbf{0}$ and $D := \mathcal{C}_{\boldsymbol{f}}$ yields:

$$\mathcal{C}_{\boldsymbol{f}}\mathcal{J}_{\Delta\boldsymbol{f}}\ddot{\mathbf{q}}_k\Delta t \leq -\mathcal{C}_{\boldsymbol{f}}\left(\boldsymbol{W}_k + \mathcal{C}_{\Delta\boldsymbol{f}}\dot{\mathbf{q}}_k\right), \tag{48}$$

where the current $\boldsymbol{W}_k$ is measurable. $\mathcal{C}_{\boldsymbol{f}}$ collects the corresponding force columns from $\mathcal{C}$, see (8).

## 5.4 Constraining cnetroidal space state jumps

*5.4.1 Angular momentum* Suppose the angular momentum is bounded by $\mathcal{L}_{\boldsymbol{c}} \leq \bar{\mathcal{L}}_{\boldsymbol{c}}$, the angular momentum constraint fits (45) with $D := I$ and $\bar{\boldsymbol{\lambda}} := \bar{\mathcal{L}}_{\boldsymbol{c}}$:

$$\mathcal{J}_{\Delta\mathcal{L}_{\boldsymbol{c}}}\ddot{\mathbf{q}}_k\Delta t \leq \bar{\mathcal{L}}_{\boldsymbol{c}} - \left(\mathcal{L}_{\boldsymbol{c}k} + \mathcal{C}_{\Delta\mathcal{L}_{\boldsymbol{c}}}\dot{\mathbf{q}}_k\right). \tag{49}$$

*5.4.2 COM velocity* The horizontal COM velocity injected into (45) with $\boldsymbol{\lambda} := \dot{\boldsymbol{c}}_{x,y}$, $D := \mathcal{G}_{\dot{\boldsymbol{c}}_{x,y}}$ and $\bar{\boldsymbol{\lambda}} := \boldsymbol{h}_{\dot{\boldsymbol{c}}_{x,y}}$ holds:

$$\mathcal{G}_{\dot{\boldsymbol{c}}_{x,y}}\mathcal{J}_{\Delta\dot{\boldsymbol{c}}_{x,y}}\ddot{\mathbf{q}}_k\Delta t \leq \boldsymbol{h}_{\dot{\boldsymbol{c}}_{x,y}} - \mathcal{G}_{\dot{\boldsymbol{c}}_{x,y}}\left(\dot{\boldsymbol{c}}_{x,y}(k) + \mathcal{C}_{\Delta\dot{\boldsymbol{c}}_{x,y}}\dot{\mathbf{q}}_k\right) \tag{50}$$

*5.4.3 ZMP* Considering the ZMP $\boldsymbol{\lambda} := \boldsymbol{z}$ in (45) with $D := \mathcal{G}_{\boldsymbol{z}}$ and $\bar{\boldsymbol{\lambda}} := \mathbf{0}$ results in:

$$\mathcal{G}_{\boldsymbol{z}}\mathcal{J}_{\Delta\boldsymbol{W}_O}\ddot{\mathbf{q}}_k\Delta t \leq -\mathcal{G}_{\boldsymbol{z}}\left(\boldsymbol{W}_O(k) + \mathcal{C}_{\Delta\boldsymbol{W}_O}\dot{\mathbf{q}}_k\right), \tag{51}$$

where $\boldsymbol{W}_O(k)$ is computed based on force measurements.

*5.4.4 DCM* Last but not least, inject the DCM as quantity $\boldsymbol{\lambda} := \boldsymbol{\xi}$ in (45) with $D := \mathcal{G}_{\boldsymbol{\xi}}$ and $\bar{\boldsymbol{\lambda}} := \boldsymbol{h}_{\boldsymbol{\xi}}$:

$$\mathcal{G}_{\boldsymbol{\xi}}\mathcal{J}_{\Delta\boldsymbol{\xi}}\ddot{\mathbf{q}}_k\Delta t \leq \boldsymbol{h}_{\boldsymbol{\xi}} - \mathcal{G}_{\boldsymbol{\xi}}\left(\boldsymbol{\xi}_k + \mathcal{C}_{\Delta\boldsymbol{\xi}}\dot{\mathbf{q}}_k\right). \tag{52}$$

## 5.5 Impact-aware whole-body QP controller

Given the derived impact-aware constraints (46-52), we extend the baseline QP controller (15):

$$\begin{aligned}
\min_{\boldsymbol{x}:(\ddot{\mathbf{q}},\boldsymbol{f}_\lambda)} \quad & \ddot{\mathbf{q}}^\top Q\ddot{\mathbf{q}} + \sum_{i\in\mathcal{I}_o} w_i\|\boldsymbol{e}_i(\boldsymbol{x})\|^2 \\
\text{s.t.} \quad & \text{Joint Space constraints:} \\
& \text{Post-impact joint velocity: (46)}, \\
& \text{Post-impact joint torque: (47)}, \\
& \text{Joint position: (4)}, \\
& \text{Joint velocity: (5)}, \\
& \text{Joint torque : (6)}, \\
& \text{Contact Space constraints:} \\
& \text{Post-impact contact wrench cone: (48)}, \\
& \text{Fulfilling Contact wrench cone: (8)}, \\
& \text{Centroidal space constraints:} \\
& \text{Post-impact angular momentum: (49)}, \\
& \text{Post-impact COM velocity: (50)}, \\
& \text{Post-impact ZMP: (51)}, \\
& \text{Post-impact DCM: (52)}, \\
& \text{Further constraints, e.g., collision avoidance: (44).}
\end{aligned} \tag{53}$$

Given reference contact velocities that are too high to fulfill the constraints after the impact event, the impact-aware constraints would autonomously determine the maximum contact velocity that keeps (53) feasible. We will show with experiments in Sec. 6, the impact-aware QP controller (53) allows for considerably high contact velocities.

Implementation-wise, we require that the high contact velocity tasks are added no earlier than the impact-aware constraints; otherwise, the initial robot configuration might be infeasible for (46 -52).

The high contact velocity tasks are completed upon the detection of the impact. We can stop using the impact-aware constraints by switching the QP controller from (53) to (15). Since the constraints associated with (53) are more conservative than (15), the switch from (53) to (15) does not lead to infeasible solutions.

**Remark 5.2.** *We recommend selecting a conservative (high) coefficient of restitution. Note that higher coefficients lead to the prediction of more significant jumps, and consequently, to more restrictive impact-aware constraints. Accordingly, the resulting contact velocity will be slower and safe.*

*In case the selected coefficient of restitution is lower than the real one, the resulting impact-induced jump will be higher than expected. Consequently, the pre-impact velocity may be too fast and violate constraints.*

### 5.6 Illustrative example

Consider a planar manipulator with two rotational joints to illustrate the joint velocity constraint. The link length is $l_1 = l_2 = 0.5$ m. The joint velocity limits are $\dot{\mathbf{q}} = [0.9, 0.6]^T$ rad/s and $\underline{\dot{\mathbf{q}}} = -\dot{\mathbf{q}}$. We chose the joint configuration $\mathbf{q}_k = [0.0, 0.2\pi]^T$ rad for demonstration. The current end-effector velocity is $\dot{\boldsymbol{x}}_k = [0, 0.3]^T$ m/s. Note that the robot is non-redundant, and there exists only one solution in joint space that performs a desired end-effector motion: $\dot{\mathbf{q}} = J^{-1}\dot{\boldsymbol{x}}$ and its derivative $\ddot{\mathbf{q}} = J^{-1}\ddot{\boldsymbol{x}} - J^{-1}\dot{J}\dot{\mathbf{q}}$. The control loop is supposed to run with $\Delta t = 5$ ms.

In this example, the aim is to move the end-effector in the positive y-direction. In the next control cycle, we aim to achieve an artificially high end-effector acceleration $\ddot{\boldsymbol{x}}_{\text{des}} = [0.0, 120.0]^T$ m/s$^2$ as exclusive control task (and equal weighting for x- and y-direction). In other words, we can optimally track the end-effector task when minimizing the Euclidean norm $\left\| J\ddot{\mathbf{q}}_k + \dot{J}\dot{\mathbf{q}}_k - \ddot{\boldsymbol{x}}_{\text{des}} \right\|^2$. Contour lines and the grey-to-white fading in Fig 3 represent this objective function. We obtain the QP

$$\begin{aligned} \ddot{\mathbf{q}}_k = \underset{\ddot{\mathbf{q}}_k}{\operatorname{argmin}}\ & \ddot{\mathbf{q}}_k^\top J^\top J\ddot{\mathbf{q}}_k + 2\ddot{\mathbf{q}}_k^\top J^\top(\dot{J}\dot{\mathbf{q}}_k - \ddot{\boldsymbol{x}}_{\text{des}}) \\ \text{s.t. } & (5) \text{ [ and (46) for impact-awareness ]} \end{aligned} . \tag{54}$$

because the scalar term $(\dot{\mathbf{q}}_k^\top \dot{J}^\top - \ddot{\boldsymbol{x}}_{\text{des}}^\top)(\dot{J}\dot{\mathbf{q}}_k - \ddot{\boldsymbol{x}}_{\text{des}})$ is constant and does not affect the optimization process.

Dotted lines in Fig. 3 depict the classical joint velocity constraint without impact-awareness (5). The resulting feasible polytope in terms of joint accelerations constitutes a rectangle with an offset from the origin.

Next, let us expect an impact at the end-effector in the next iteration. The contact normal corresponds to the y-axis, and we select the coefficient of restitution $c_{\text{r}} = 0.02$. The convex bounds of the impact-aware joint velocity constraint (46) are indicated by dashed lines. Note that this constraint corresponds to a polyhedron with parallel bounds. In practice, we need to superimpose both constraints to obtain a shrunken, convex polytope, shown in blue solid lines. Extreme vertices are indicated by star symbols[¶].

The optimal impact-aware solutions found by (54) and send as references to the robot are marked in Fig. 3 by colored circles: employing (5) only as the impact-unaware solution (red) and the impact-aware solution (green) with (5) and (46). We cannot fully achieve the desired end-effector acceleration in both cases. The impact-aware solution is even more conservative because of the additional constraint.

The post-impact joint velocity is given by

$$\dot{\mathbf{q}}_{k+1}^+ = (I + \mathcal{J}_{\Delta\dot{\mathbf{q}}})\,(\Delta t\ddot{\mathbf{q}}_k + \dot{\mathbf{q}}_k)\,. \tag{55}$$

Without impact-awareness $\dot{\mathbf{q}}_{k+1}^+ = [-0.618, 1.345]^T$ rad/s violates the second joint's velocity limit. Instead, with impact-awareness $\dot{\mathbf{q}}_{k+1}^+ = [-0.281, 0.600]^T$ rad/s. The deceleration ensures the satisfaction of the joint velocity constraint in the next iteration.

## 6 Experiments

We validate our impact-aware QP controller (53) on the full-size humanoid robot HRP-4 with 34 actuated joints. Two sets of experiments are performed; each has different highlights and fulfills the constraints imposed by the impact event. We encourage interested readers to check the experiment videos[‖].

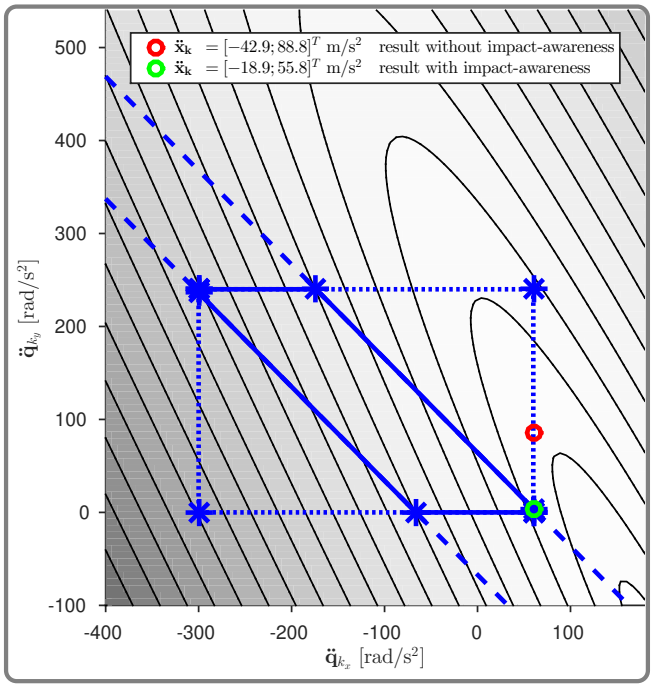



**Figure 3.** Feasible polytopes related to the joint velocity constraint for a 2-DoF planar manipulator. Circles indicate solutions of the QP (54).

**Experiment 1** Pushing a concrete wall with the maximum contact velocity –see the illustration in Fig. 4 and experiment setup in Sec. 6.1, with the following highlights:

**(H-1.1)** Impact-aware control updating the contact velocity in real-time instead of the desired, infeasible 0.8 m/s. At the impact time, the contact velocity is measured to be 0.35 m/s.

**(H-1.2)** High contact velocity jump and subsequent high contact force jump (137 N).

**(H-1.3)** Robustness to the uncertain location of the wall along the pre-defined contact normal.

**(H-1.4)** All the impact-aware constraints (46-52) are fulfilled despite the relatively high contact velocity and contact force jumps.

**Experiment 2** Grabbing a cardboard box, see the illustration in Fig. 6 and experiment setup in Sec. 6.2, with the following highlights:

**(H-3.1)** Swift box-grabbing motion without stopping for establishing contacts.

**(H-3.2)** Two simultaneous impacts.

For **Experiment 1** we first present the detailed experiment setup and confirm the highlights **(H-1.1)** and **(H-1.2)**. Then

[¶] The figure was generated using Matlab scripts to convert between halfspace- and vertex-representation.

[‖] Online: https://youtu.be/v1Jfy8-jiwE

we check the feasibility **(H-1.4)** in joint space (46-47) and and centroidal space (50-52) in different subsections.

In the internal visualization related to experiment video of **Option C**, the wall location does not match the reality. However, the controller is not affected as it is independent of the exact impact location **(H-1.3)**.

For **Experiment 2**, the highlights **(H-3.1)** and **(H-3.2)** are demonstrated by the contact velocity profile in Fig. (10(a)-10(b)) and the contact force profile Fig. (10(c)-10(d)).

## 6.1 Pushing a wall

We designed **Experiment 1** in order to show that the impact-aware constraints enable the QP to autonomously determine the maximum contact velocity online, regardless of an infeasible high reference contact velocity.

In Sec. 6.1.4, we compare three ZMP constraint formulations: each restricts the ZMP differently:

**Option A** : the feet's support polygon: $\boldsymbol{z} \in \mathcal{S}_s$.

**Option B** : the multi-contact ZMP area: $\boldsymbol{z} \in \mathcal{S}_{\boldsymbol{z}}$.

**Option C** : the static COM equilibrium area: $\boldsymbol{z} \in \mathcal{S}_c$.

We conclude that **Option C** leads to the highest contact velocity mentioned in **(H-1.1)**.

*6.1.1 Experimental setup* We present the technical details structured in the following three aspects.

*Robot configuration* The humanoid is initialized with a standing posture in double coplanar supports and commanded its right gripper's palm to hit a wall in front of it with an infeasible high reference contact velocity (0.8 m/s), see Fig. 7(e). The hand was prepared by mounting a 3D printed rigid plastic palm of 3 cm thickness.

*Parameter configuration* The ATI-45 force-torque sensors (mounted at the ankles and the wrists) are read at 200 Hz, thus, we select the impact duration $\delta t$ as 5 ms. The QP controller runs at the same frequency with sampling period $\Delta t = 5$ ms. The friction coefficient of the established contacts is $0.7$**. The coefficient of restitution is $c_{\mathrm{r}} = 0.02$. We select $H = I$ in the IDQP (28), i.e. $\gamma_a = 1$, $\gamma_b = 1$ and not using the term $\mathbf{M}^T\mathbf{M}$.

**Remark 6.1.** *Uncertain coefficient of restitution $\widetilde{c}_r$ leads to an uncertain prediction of the end-effector velocity jump $\Delta\widetilde{\dot{\boldsymbol{x}}}_{k+1}$ due to (17), and accordingly, to uncertain predictions $\widetilde{\boldsymbol{u}}^* = [\Delta\widetilde{\dot{\mathbf{q}}}^{*\top}, \widetilde{\boldsymbol{\iota}}^{*\top}]^\top$ due to (29). Given $\widetilde{c}_r$ is a scalar, the errors*

$$\boldsymbol{e}_{\boldsymbol{u}^*} = \begin{bmatrix} \boldsymbol{e}_{\Delta\dot{\mathbf{q}}^*} \\ \boldsymbol{e}_{\boldsymbol{\iota}^*} \end{bmatrix} = \begin{bmatrix} \Delta\widetilde{\dot{\mathbf{q}}}^* - \Delta\dot{\mathbf{q}}^* \\ \widetilde{\boldsymbol{\iota}}^* - \boldsymbol{\iota}^* \end{bmatrix}$$

*are proportional to the error*

$$\boldsymbol{e}_{c_r} = (1 + \widetilde{c}_r) - (1 + c_r) = \widetilde{c}_r - c_r.$$

*Through the analysis from several trial-runs, we choose the coefficient of restitution $c_r = 0.02$, which leads to a reasonable prediction of contact force jump $\Delta\boldsymbol{f}$, see Fig. 7(f). We can observe $\Delta\boldsymbol{f}$ from the ATI sensors more precisely than observing the joint velocity jumps $\Delta\dot{\mathbf{q}}$ from encoders.*

*Task description* Fig. 5(a) reports the finite state machine (FSM) that is utilized to modulate and change the behavior of the robot. The *Start* state initializes the right palm to be parallel to the wall. The impact-aware QP (53) is activated in the *Impact* state, where in each time step the impact-aware QP (53) autonomously identifies the current maximum contact velocity given the infeasible reference contact velocity 0.8 m/s, see Fig. 7(e). The transition from *Impact* state to *Admittance* state depends upon the *impact detection criterion*, i.e., 20 N measured by the force sensor. During the *Admittance* state, the robot regulates the contact force to 15 N for 14 seconds, thereby stabilizing the contact. Then the *Detach* state releases the contact and the *Reset* state returns to the initial configuration.

It is a good practice to regulate the contact wrenches to stabilize the under-actuated robot center of mass dynamics, e.g. Caron et al. (2019). Yet, in order to observe impact-induced state jumps without interference, especially the contact force jumps and the ZMP jumps, we decided not to apply admittance behavior for the established contacts (feet) in the experiments. Instead, we apply task-space position control for the feet.

*6.1.2 Contact velocity regulation* Given the infeasible high reference contact velocity 0.8 m/s displayed by purple dashed line in Fig. 7(e), the QP controller (53) updates the feasible contact velocity (displayed in orange line) concerning the impact-awareness in each time step. Thus we do not rely on a pre-specified contact location **(H-1.3)**.

The measured contact force jump is close to its prediction in Fig. 7(f), which indicates a reasonable coefficient of restitution $c_{\mathrm{r}}$. see Remark. 6.1.

*6.1.3 Joint space constraints* We present plots from two out of all the joints, namely the right arm shoulder roll joint and the elbow joint. In Fig. 7(a) and Fig. 7(b) the measured joint velocities fulfill the limits (44). To improve the visibility while scaling the figures properly, we ignored the well fulfilled joint velocity bounds $\{-2.052, 2.052\}$ rad/s in Fig. 7(a) and $\{-2.649, 2.649\}$ rad/s in Fig. 7(b).

The HRP-4 is not equipped with joint torque sensors. Thus we obtain the torque jumps using the well-known relation: $\boldsymbol{\tau} = J^\top \boldsymbol{f}$. We selected $\Delta\bar{\boldsymbol{\tau}} = 0.4\bar{\boldsymbol{\tau}}$ and $\Delta\underline{\boldsymbol{\tau}} = 0.4\underline{\boldsymbol{\tau}}$ for constraint (47). In Fig. 7(c) and Fig. 7(d), the joint torques are also well-bounded in line with (44).

*6.1.4 ZMP constraint* The feasible ZMP area is recently extended to general multi-contact settings, yet considering static contacts. Despite the well-established push recovery strategies for planar bipedal maneuvers by Stephens (2007); Sugihara (2009), in the case of an impact event resulting in a discontinuous change of the multi-contact situation, it is unclear which bounds apply to the ZMP. Running comparative experiment trials, we investigated three formulations of the ZMP bounding polygon illustrated in Fig. 11. These are injected into the impact-aware ZMP constraint (51).

**Option A**: The ZMP is restricted to stay in the support polygon $\mathcal{S}_s$ defined by the feet contacts: $\boldsymbol{z} \in \mathcal{S}_s$. In other

** Note that the coefficient of friction does not affect the solution obtained by the IDQP.

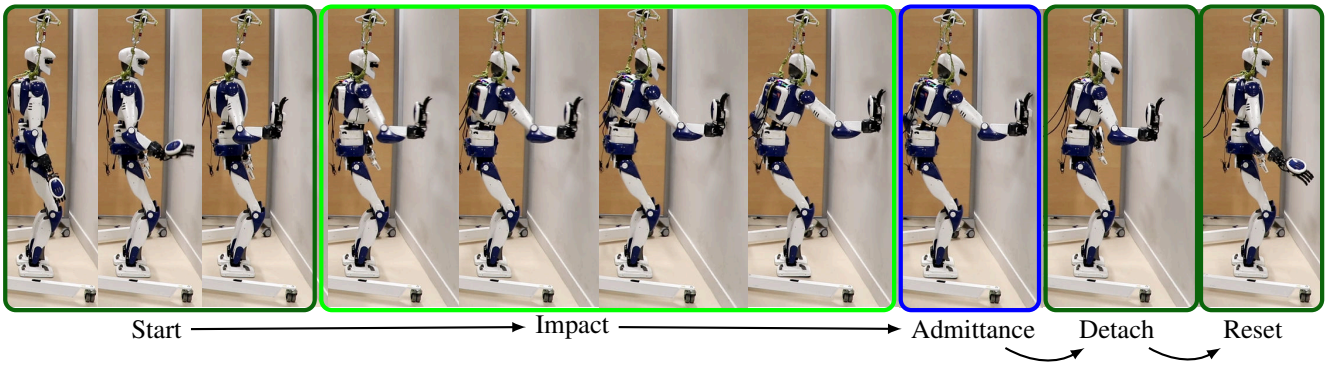
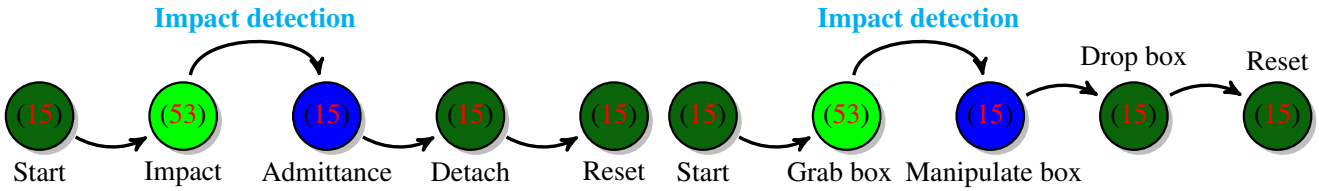



**Figure 4.** Snapshot of **Experiment 1** with **Option C**: the HRP-4 robot hit the wall with contact velocity $0.35$ m/s.




(a) **Experiment 1** (b) **Experiment 2**

**Figure 5.** Finite state machines of **Experiment 1** and **Experiment 2** with the corresponding snapshots in Fig. 4 and 6 respectively. The impact-aware QP (53) is applied in the green-highlighted *Impact* states. The contact (or contacts for box-grabbing) are set, and admittance tasks are activated in the blue-highlighted states *Admittance* and *Manipulate box*.

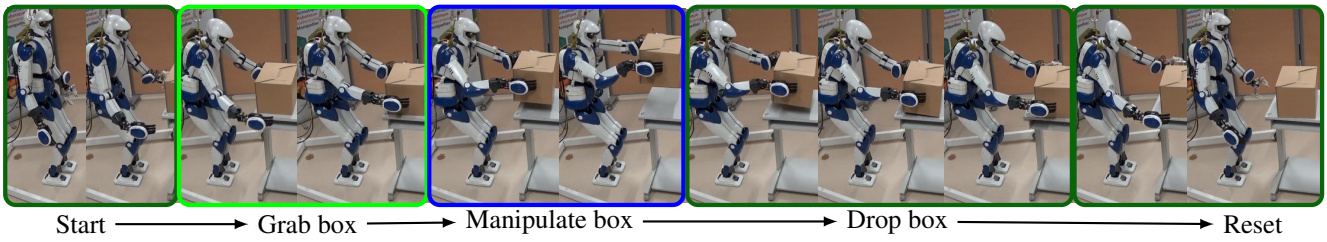



**Figure 6.** Snapshot of **Experiment 2**: the HRP-4 robot grabbing a box with contact velocities $0.15$ m/s at both hands.

words, the classical ZMP area for coplanar contacts is applied (Vukobratović and Borovac 2004), and the contact to be established by the hand is ignored. Fig. 9 plots the results. This choice leads to a conservative contact velocity $0.11$ m/s as shown in Fig. 9(a) and a small contact force jump 37 N displayed by Fig. 9(b).

**Option B**: We restrict the ZMP within the *multi-contact ZMP area* $\mathcal{S}_z$ defined by the feet contacts and the hand contact as formulated in Caron et al. (2017): $z \in \mathcal{S}_z$. Notice that we apply an optimized version that meets real-time requirements. Compared to **Option A**, we observe a slightly higher contact velocity $0.17$ m/s. Due to the similarity, we do not plot figures for this trial run.

**Option C**: Finally, given the contacts of feet and the hand, we restrict the ZMP within the *static COM equilibrium area* $\mathcal{S}_c$: $z \in \mathcal{S}_c$, for more details on $\mathcal{S}_c$, refer to Bretl and Lall (2008). We employ an optimized algorithm to compute $\mathcal{S}_c$. Fig. 12 illustrates the evolution over time of the real ZMP (9) and the predicted ZMP $z + \Delta z$ (41). Fig. 8 reports further results. The robot applies the contact velocity $0.35$ m/s, which is the highest among all trials **(H-1.1)**, observing the associated contact force jump 137 N **(H-1.2)**. We note that all constraints are fulfilled, the hardware does not break and the robot maintains balance **(H-1.4)**.

#### 6.1.5 *COM velocity constraints*

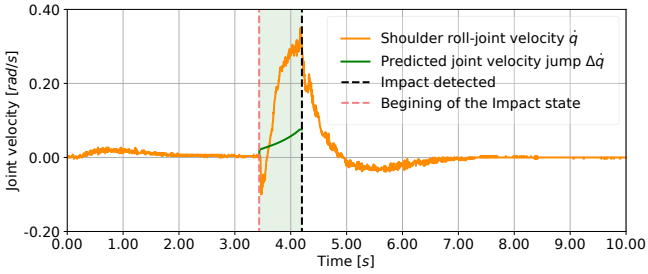
As we do not have a well defined bound for the COM velocity in Fig. (8(c)-8(d)) we only plot the actual value against the predicted impact-induced state jumps. The important observation is that the actual jump is close to the prediction.

For plots in Sec. 6.1.4 and Sec. 6.1.5, there exists a general observation that along with the contact normal direction (the $x$-axis plots), predictions of the floating-base state jumps can bound the actual jumps, and the discrepancies are relatively small. Since our impact model (17) considers only one-dimensional/directional impact, predictions of lateral direction ($y$-axis plots) are not as good. We consider this acceptable for this study since the magnitude of the lateral direction ($y$-axis plots) jumps are much smaller.

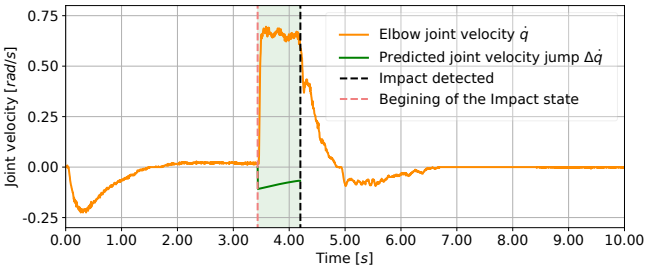
### 6.2 *Box-grabbing with a swift motion*

In **Experiment 2**, the HRP-4 robot lifts a cardboard box using swift motion without reducing its hands' speed to establish contacts.

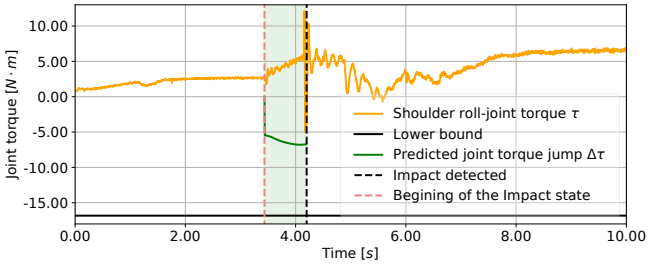
#### 6.2.1 *Experimental setup*



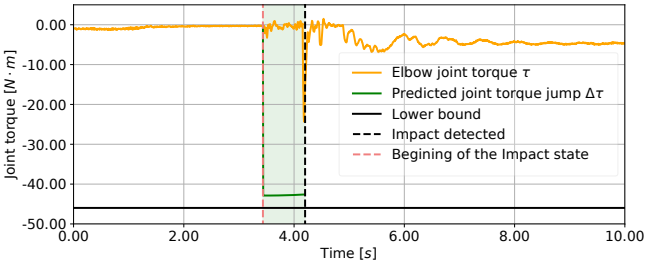

(a) **Experiment 1** Right arm shoulder-roll-joint velocity



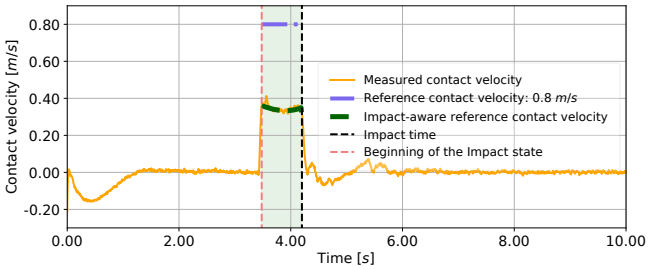

(b) **Experiment 1** Right arm elbow joint Velocity



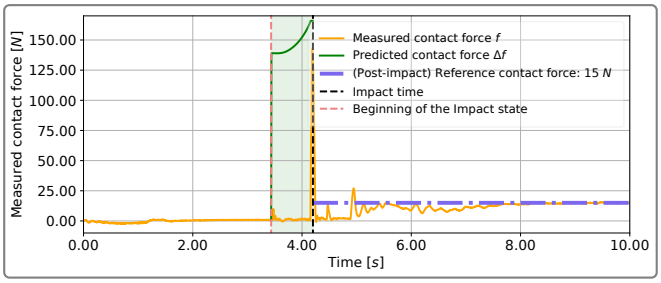

(c) **Experiment 1** Right arm shoulder-roll-joint torque




(d) **Experiment 1** Right arm elbow joint torque

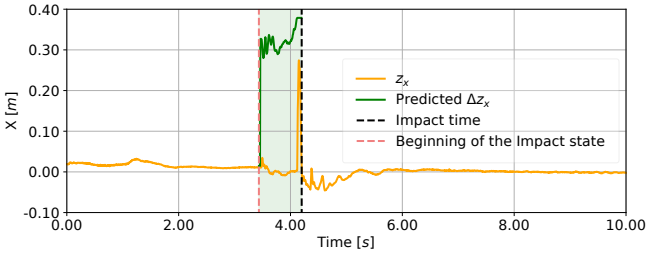



(e) **Experiment 1** contact velocity jump

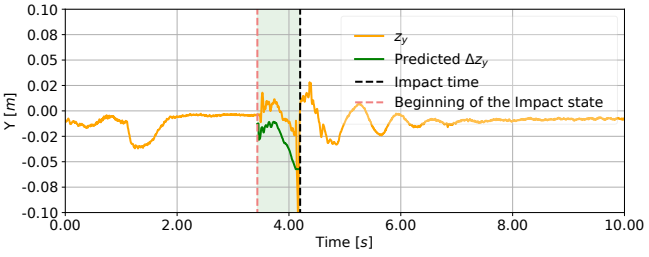



(f) **Experiment 1** contact force jump

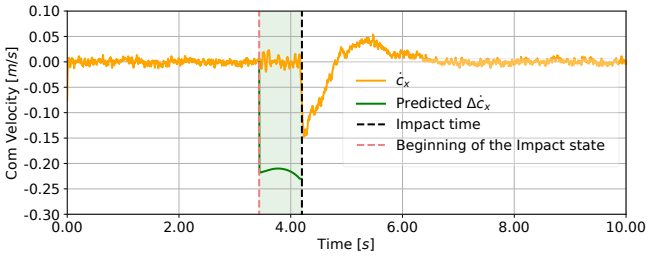
**Figure 7.** Impact-aware joint space and contact space constraints of **Experiment 1** with **Option C**.



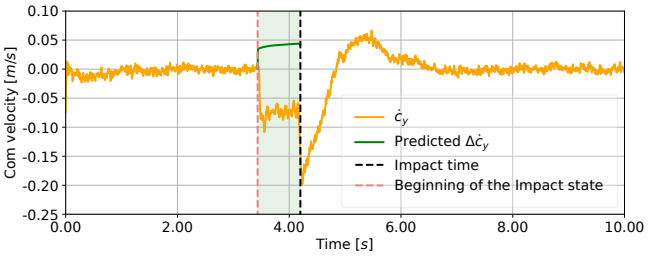

(a) **Experiment 1** $x-$axis ZMP $\boldsymbol{z}_x$




(b) **Experiment 1** $y-$axis ZMP $\boldsymbol{z}_y$

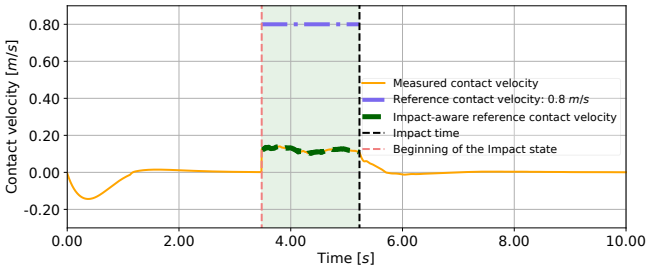



(c) **Experiment 1** $x-$axis COM velocity $\dot{\boldsymbol{c}}_x$

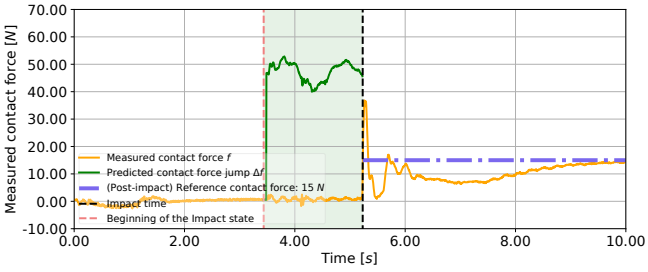



(d) **Experiment 1** Y-axis COM velocity $\dot{\boldsymbol{c}}_y$

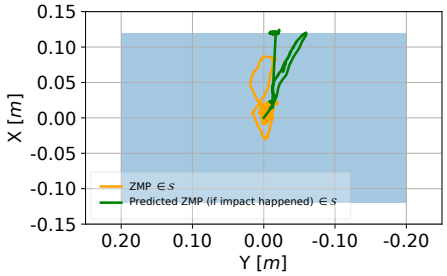
**Figure 8.** Impact-aware centroidal space constraints of **Experiment 1** with **Option C**.



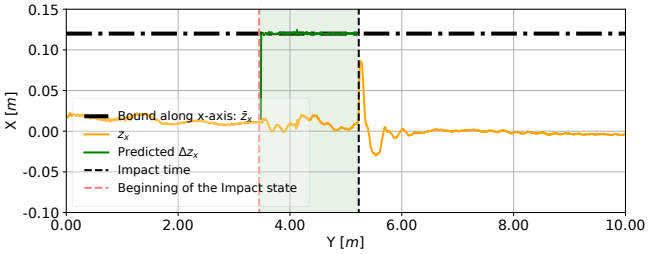

(a) **Experiment 1** with **Option A**: contact Velocity




(b) **Experiment 1** with **Option A**: contact force

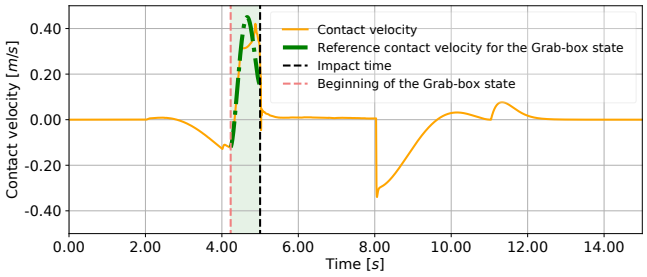



(c) **Experiment 1** with **Option A**: the ZMP is restricted in the support polygon

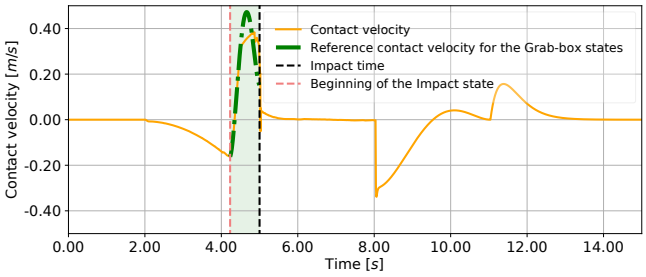



(d) **Experiment 1** with **Option A**: $x-$axis ZMP

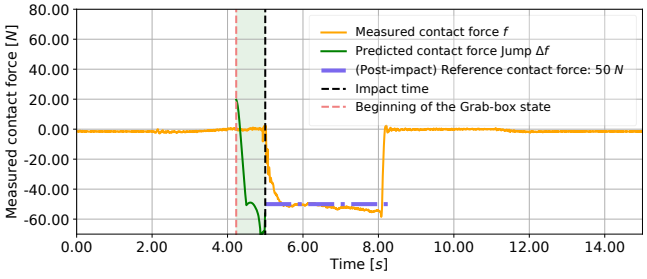
**Figure 9.** Impact-aware constraints of **Experiment 1** with **Option A**.



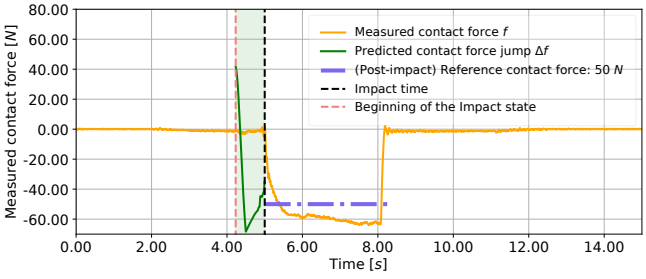

(a) **Experiment 2** Left arm contact velocity




(b) **Experiment 2** Right arm contact velocity




(c) **Experiment 2** Left arm contact force jump




(d) **Experiment 2** Right arm contact force jump

**Figure 10.** Contact velocities and contact force jumps of **Experiment 2**.

*Robot configuration* The initial robot configuration is the same as in **Experiment 1**. Whereas instead of a concrete wall, there is a table and a cardboard box in front of the robot. We mounted a felt on the plastic palm to increase the friction with the cardboard.

*Task description* Fig. 5(b) shows that **Experiment 2** follows a similar FSM as **Experiment 1**.

The *Start* state raises the two palms; then the *Grab box* state moves the two palms following pre-defined trajectories considering an approximate location and size of the box. The two hands contacted the box with velocities at 0.15 m/s, and all the constraints are respected. Upon the same *impact detection criterion*, the *Manipulate box* state activates two admittance tasks that regulate the contact forces to 50 N. The *Drop box* state moves the two palms following another set of pre-defined trajectories to release the contacts and then resume the initial robot configuration in the *Reset* state.

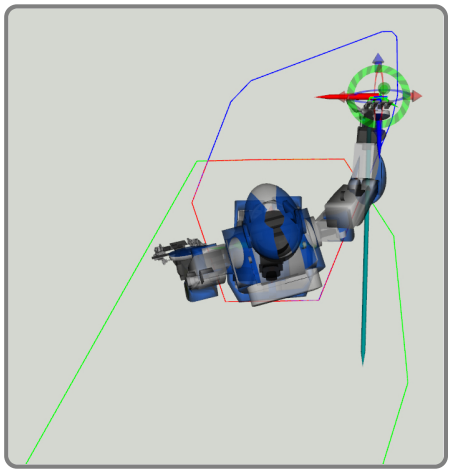


**Figure 11.** Illustration of the multi-contact ZMP area $\mathcal{S}_z$, that defines **Option B** (light green polygon), and the multi-contact static COM equilibrium area $\mathcal{S}_c$, that defines **Option C** (blue polygon). The light blue arrow on the right hand indicates the contact force jump. The red area indicates the DCM area: $\mathcal{S}_\xi = \mathcal{S}_z \bigcap \mathcal{S}_c$.

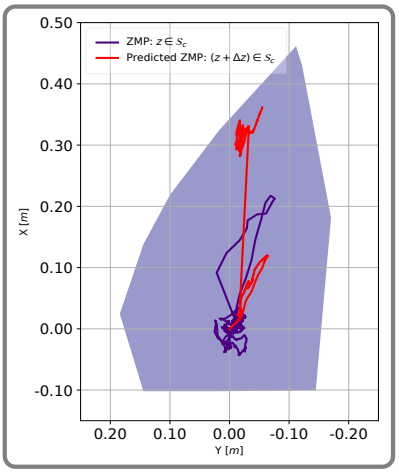



**Figure 12.** In **Experiment 1**, the ZMP is restricted in the multi-contact COM area $\mathcal{S}_c$ which is updated in every control cycle according to the current posture. This polygon deviates from Fig. 11 as it shows a slightly different stance.

*6.2.2 Results* We plotted the contact velocities for the left and right arms in Fig. 10(a) and Fig. 10(b), respectively. In both cases, the contact velocities are close to the reference 0.15 m/s. Thus we say the contacts are established without either slowing down or following pre-defined deceleration trajectories **(H-3.1)**. Comparing the impact detection time, we can find that the two impacts are simultaneous **(H-3.2)**.

The impact-induced contact force jumps of left and right arms are presented in Fig. 10(c) and Fig. 10(d), which are smaller compared to **Experiment 1** due to the slower reference contact velocities.

Like **Experiment 1**, the contact force jumps are well predicted (with a bit conservative margin). On the other hand different from **Experiment 1**, the reference contact velocities are well tracked (instead of reducing from 0.8 m/s to 0.35 m/s) as the impact-induced state jumps will not break the impact-aware constraints (48-52).

It is important to note that we here present only safe experiments. Demonstrating a QP failure due to a pre-impact velocity that was too large would result in hardware failure or a severe fall, which we can not afford with our robot.

## 7 Conclusion and future work

The primary and original contribution of this work is in enabling inverse dynamic tasked-based QP control to achieve robot motions with impacts safely. Our approach is capable of robustly executing physical contact established at non-zero speed by optimizing pre-impact velocities. In the vicinity of impact, the controller encompasses in a one-iteration ahead, the jumps that would be induced by a desired impact (shall it occur) and enforce all the constraints to be aware of (i.e., handle) subsequent jumps in the state velocity and contact forces constraint. As part of the prediction process, impact propagation along the constrained arborescent kinematic tree is considered. Our current framework ensures hardware feasibility and the maintenance of pre-existing contacts by autonomously determining the maximum yet *safe* contact velocity reference. It applies to floating-base robots in general multi-contact situations.

Our approach does not rely on planning contact at a specific location or time. Instead, it requires only the contact surface normal to be known. We achieved high contact velocities, and large force jumps in the experiments conducted with an HRP-4 humanoid. To our best knowledge, this is the first approach that performs impact-aware whole-body control based on constrained quadratic optimization.

Now that we confirmed the main concept and encapsulated impact-aware tasks in our general multi-objective and multi-sensory task-space QP control, we plan to investigate the following shortcomings: (i) investigate more advanced impact models that include friction in 3D; (ii) develop more refined impact propagation models, namely those based on energy spreading and assess them through ground-truth instrumentations; (iii) a more in-depth investigation of dynamic equilibrium under impacts; (iv) off-line and on-line identification of impact pertinent parameters to refine the controller performance; (v) sliding and tangential impacts so as, for example, to approach grabbed objects that are moving.

## Acknowledgment

This work was funded by the H2020 EU project I-AM. Yuquan Wang is partially supported by the National Natural Science Foundation of China (U1613216) and Shenzhen Fundamental Research Grant (JCYJ20180508162406177) through the Chinese University of Hong-Kong, Shenzhen.

The authors wish to thank Pierre Gergondet for his continuous support in setting up the `mc_rtc` controller and Stéphane Caron for many fruitful discussions regarding the implementation of multi-contact ZMP and DCM.

### References

Aghasadeghi N, Long A and Bretl T (2012) Inverse optimal control for a hybrid dynamical system with impacts. In: *IEEE International Conference on Robotics and Automation*. pp. 4962–4967.

Battaglia M, Blanchet L, Kheddar A, Kajita S and Yokoi K (2009) Combining haptic sensing with safe interaction. In: *IEEE/RSJ*

*International Conference on Intelligent Robots and Systems*. pp. 231–236.

Bergés P and Bowling A (2005) Impact forces in legged robot locomotion. In: *IEEE International Conference on Robotics and Automation*. pp. 3745–3751.

Bouyarmane K, Chappellet K, Vaillant J and Kheddar A (2019) Quadratic programming for multirobot and task-space force control. *IEEE Transactions on Robotics* 35(1): 64–77.

Bouyarmane K and Kheddar A (2018) On weight-prioritized multitask control of humanoid robots. *IEEE Transactions on Automatic Control* 63(6): 1632–1647.

Bretl T and Lall S (2008) Testing static equilibrium for legged robots. *IEEE Transactions on Robotics* 24(4): 794–807.

Brogliato B (2003) Some perspectives on the analysis and control of complementarity systems. *IEEE Transactions on Automatic Control* 48(6): 918–935.

Caron S, Kheddar A and Tempier O (2019) 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.

Caron S, Pham QC and Nakamura Y (2015) Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas. In: *IEEE International Conference on Robotics and Automation*. pp. 5107–5112.

Caron S, Pham QC and Nakamura Y (2017) Zmp support areas for multi-contact mobility under frictional constraints. *IEEE Transactions on Robotics* 33(1): 67–80.

Dehio N (2018) *Prioritized Multi-Objective Robot Control*. PhD dissertation, Technical University Braunschweig, Germany. URL https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00066108.

Dietrich A and Ott C (2020) Hierarchical impedance-based tracking control of kinematically redundant robots. *IEEE Transactions on Robotics* 36(1): 204–221.

Djeha M, Tanguy A and Kheddar A (2020) Adaptive-gains enforcing constraints in closed-loop qp control URL https://hal.archives-ouvertes.fr/hal-02490195/file/RAL_IROS_DJEHA.pdf.

Englsberger J, Mesesan G, Ott C and Albu-Schäffer A (2018) Dcm-based gait generation for walking on moving support surfaces. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 1–8.

Grizzle JW, Chevallereau C, Sinnet RW and Ames AD (2014) Models, feedback control, and open problems of 3d bipedal robotic walking. *Automatica* 50(8): 1955–1988.

Haddadin S, Albu-Schäffer A and Hirzinger G (2009) Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research* 28(11-12): 1507–1527.

Heck D, Saccon A, Van de Wouw N and Nijmeijer H (2016) Guaranteeing stable tracking of hybrid position-force trajectories for a robot manipulator interacting with a stiff environment. *Automatica* 63: 235–247.

Hildebrandt AC, Wittmann R, Sygulla F, Wahrmann D, Rixen D and Buschmann T (2019) Versatile and robust bipedal walking in unknown environments: real-time collision avoidance and disturbance rejection. *Autonomous Robots* 43(8): 1957–1976.

Hu G, Makkar C and Dixon WE (2007) Energy-based nonlinear control of underactuated euler-lagrange systems subject to impacts. *IEEE Transactions on Automatic Control* 52(9): 1742–1748.

Hurmuzlu Y, Génot F and Brogliato B (2004) Modeling, stability and control of biped robots–a general framework. *Automatica* 40(10): 1647 – 1664.

Jia YB, Gardner M and Mu X (2019) Batting an in-flight object to the target. *The International Journal of Robotics Research* 38(4): 451–485.

Jia YB and Wang F (2017) Analysis and computation of two body impact in three dimensions. *Journal of Computational and Nonlinear Dynamics* 12(4): 041012.

Johnson AM, Burden SA and Koditschek DE (2016) A hybrid systems model for simple manipulation and self-manipulation systems. *The International Journal of Robotics Research* 35(11): 1354–1392.

Kajita S, Morisawa M, Miura K, Nakaoka S, Harada K, Kaneko K, Kanehiro F and Yokoi K (2010) Biped walking stabilization based on linear inverted pendulum tracking. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4489–4496.

Konno A, Myojin T, Matsumoto T, Tsujita T and Uchiyama M (2011) An impact dynamics model and sequential optimization to generate impact motions for a humanoid robot. *The International Journal of Robotics Research* 30(13): 1596–1608.

Koolen T, De Boer T, Rebula J, Goswami A and Pratt J (2012) Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *The international journal of robotics research* 31(9): 1094–1113.

Kuindersma S, Deits R, Fallon M, Valenzuela A, Dai H, Permenter F, Koolen T, Marion P and Tedrake R (2016) Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots* 40(3): 429–455.

Lee SH and Goswami A (2012) A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots* 33(4): 399–414.

Liu M, Lober R and Padois V (2016) Whole-body hierarchical motion and force control for humanoid robots. *Autonomous Robots* 40(3): 493–504.

Lober R, Sigaud O and Padois V (2019) Task Feasibility Maximization using Model-Free Policy Search and Model-Based Whole-Body Control. *Frontiers in Robotics and AI* Preprint.

Manchester IR, Mettin U, Iida F and Tedrake R (2011) Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research* 30(3): 265–279.

Nava G, Romano F, Nori F and Pucci D (2016) Stability analysis and design of momentum-based controllers for humanoid robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 680–687.

Nenchev DN (2013) Reaction null space of a multibody system with applications in robotics. *Mechanical Sciences* 4(1): 97–112.

Nenchev DN (2018) The momentum equilibrium principle: Foot contact stabilization with relative angular momentum/velocity. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 17–24.

Or Y and Ames AD (2010) Stability and completion of zeno equilibria in lagrangian hybrid systems. *IEEE Transactions on*

*Automatic Control* 56(6): 1322–1336.

Orin DE, Goswami A and Lee SH (2013) Centroidal dynamics of a humanoid robot. *Autonomous robots* 35(2-3): 161–176.

Pagilla PR and Yu B (2001) A stable transition controller for constrained robots. *IEEE/ASME transactions on mechatronics* 6(1): 65–74.

Partridge CB and Spong MW (2000) Control of planar rigid body sliding with impacts and friction. *The International Journal of Robotics Research* 19(4): 336–348.

Pashah S, Massenzio M and Jacquelin E (2008) Prediction of structural response for low velocity impact. *International Journal of Impact Engineering* 35(2): 119–132.

Posa M, Tobenkin M and Tedrake R (2015) Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control* 61(6): 1423–1437.

Reher J, Cousineau EA, Hereid A, Hubicki CM and Ames AD (2016) Realizing dynamic and efficient bipedal locomotion on the humanoid robot durus. In: *IEEE International Conference on Robotics and Automation*. pp. 1794–1801.

Rijnen M, de Mooij E, Traversaro S, Nori F, van de Wouw N, Saccon A and Nijmeijer H (2017) Control of humanoid robot motions with impacts: Numerical experiments with reference spreading control. In: *IEEE International Conference on Robotics and Automation*. pp. 4102–4107.

Singh BRP and Featherstone R (2020) Mechanical shock propagation reduction in robot legs. *IEEE Robotics and Automation Letters* 5(2): 1183–1190.

Sreenath K, Park HW, Poulakakis I and Grizzle JW (2011) A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *The International Journal of Robotics Research* 30(9): 1170–1193.

Stanisic RZ and Fernández ÁV (2012) Adjusting the parameters of the mechanical impedance for velocity, impact and force control. *Robotica* 30(4): 583–597.

Stephens B (2007) Humanoid push recovery. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 589–595.

Stewart DE (2000) Rigid-body dynamics with friction and impact. *SIAM review* 42(1): 3–39.

Sugihara T (2009) Standing stabilizability and stepping maneuver in planar bipedalism based on the best com-zmp regulator. In: *International Conference on Robotics and Automation*. pp. 1966–1971.

Tsujita T, Konno A, Komizunai S, Nomura Y, Owa T, Myojin T, Ayaz Y and Uchiyama M (2008) Analysis of nailing task motion for a humanoid robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nice, France, pp. 1570–1575.

Vanderborght B, Albu-Schäffer A, Bicchi A, Burdet E, Caldwell DG, Carloni R, Catalano M, Eiberger O, Friedl W, Ganesh G et al. (2013) Variable impedance actuators: A review. *Robotics and autonomous systems* 61(12): 1601–1614.

Vukobratović M and Borovac B (2004) Zero-moment pointthirty five years of its life. *International journal of humanoid robotics* 1(01): 157–173.

Wang Y and Kheddar A (2019) Impact-friendly robust control design with task-space quadratic optimization. In: *Proceedings of Robotics: Science and Systems*, volume 15. Freiburg, Germany, p. 32.

Wang Y, Tanguy A, Gergondet P and Kheddar A (2019) Impact-aware multi-contact motion generation with a quadratic optimization controller. In: *IEEE-RAS International Conference on Humanoid Robots*. Toronto, Canada.

Wensing PM, Wang A, Seok S, Otten D, Lang J and Kim S (2017) Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics* 33(3): 509–522.

Wieber PB, Tedrake R and Kuindersma S (2016) Modeling and control of legged robots. In: *Springer handbook of robotics*. Springer, pp. 1203–1234.

Wiedebach G, Bertrand S, Wu T, Fiorio L, McCrory S, Griffin R, Nori F and Pratt J (2016) Walking on partial footholds including line contacts with the humanoid robot atlas. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 1312–1319.

Zheng YF and Hemami H (1985) Mathematical modeling of a robot collision with its environment. *Journal of Field Robotics* 2(3): 289–307.