



**HAL**  
open science

# Robot-Safe Impacts with Soft Contacts Based on Learned Deformations

Niels Dehio, Abderrahmane Kheddar

► **To cite this version:**

Niels Dehio, Abderrahmane Kheddar. Robot-Safe Impacts with Soft Contacts Based on Learned Deformations. ICRA, May 2021, Xi'an, China. hal-02973947v2

**HAL Id: hal-02973947**

**<https://hal.science/hal-02973947v2>**

Submitted on 15 Mar 2021 (v2), last revised 24 Mar 2021 (v3)

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

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

# Robot-Safe Impacts with Soft Contacts Based on Learned Deformations

Niels Dehio and Abderrahmane Kheddar

**Abstract**—Safely generating impacts with robots is challenging due to subsequent discontinuous velocity and high impact forces. We aim at increasing the impact velocity – the robot’s relative speed prior to contact – such that impact-tasks like grabbing and boxing are made with the highest allowable speed performance when needed. Previous works addressed this problem for rigid bodies’ impacts. This letter proposes a control paradigm for generating intentional impacts with deformable contacts that incorporates hardware and task constraints. Based on data-driven learning of the shock-absorbing soft dynamics and a novel mapping of joint-space limits to contact-space, we devise a constrained model-predictive control to maximize the intentional impact within a feasible, robot-safe level. Our approach is assessed with real-robot experiments on the redundant Panda manipulator, demonstrating high pre-impact velocities (up to 0.9 m/s) of a rigid end-effector on soft objects and an end-effector soft suction-pump on rigid or deformable objects.

## I. INTRODUCTION

Our work’s long-term context is to enable efficient robotic fast grabbing, tossing, and boxing objects in automated industrial sorting chains. In such applications, robots shall reach, pick and toss or place objects of different sizes, shapes, and materials from one location to another as fast as possible. The proposed frameworks in [1], [2] and many other related works slow down drastically robot motion (up to zero relative velocity) when establishing contact with the environment. We instead aim to generate powerful impacts intentionally and therefore we are interested in the maximum allowed end-effector velocity that does not damage the robot.

There are roughly two categories of objects or environments to contact: rigid and soft. We have already addressed the rigid case in [3], [4]; see also other works dealing with rigid robotic impact in their references section. In this paper, we focus on maximizing the impact velocity considering *deformable* contacts. The softness will consume large parts of the kinetic energy and allows higher pre-impact velocities w.r.t rigid bodies. Our main contribution is a model predictive control scheme that optimizes future deformations to be initiated with the highest possible end-effector velocity based on learned, constrained deformation dynamics. For the first time, our formulation maps joint-space hardware limits of a redundant kinematic chain onto the contact-space. Including these inequality constraints into the deformation dynamics, ensures safe operation over a sufficiently long preview-horizon. Compared to related works, our approach’s novelty is the explicit computation of the maximum robot-safe impact velocity. It applies to three distinct scenarios:



Fig. 1: Panda manipulators equipped with a rigid flat end-effector (left) and with a pump plus flexible suction cup (right). In the experiments, we identify the deformation dynamics for the soft dice (left) and the sucker (right) in order to initiate contacts with the maximum allowed impact.

- 1) rigid end-effector impacting a soft object,
- 2) end-effector equipped with soft material (e.g., suction cup) impacting a rigid object, and a
- 3) combination of 1 and 2 (soft object and end-effector).

We assessed our approach in various real-robot experiments with Panda manipulators, (c.f Fig 1 and shown in the video <https://youtu.be/juynq6x9JJ8>), and indeed observed high impact velocities without violating hardware limits and remaining in the elastic deformation domain.

## II. RELATED WORKS

Soft material undergoing contact forces implies local or global shape changes (deformation) and deformation dynamics [5]. Related works (i) propose deformation models and (ii) devise control schemes relying on these models.

*Deformation Models:* Due to its importance, extensive literature on contact dynamics is available to the community; among them, [6] provides an excellent early overview and [7] a recent survey. The finite element method (FEM) is a widely known numerical method that simulates material deformations by dividing a non-rigid body into many small parts called finite elements assembled into a mesh. This technique emerged in mechanical engineering and has been adopted in the field of robotics, see e.g., [8], [9], [10].

In recent work, [11] proposed a method for soft contact model identification. The approach in [12] combines estimation and control of the material’s contact normal stiffness

The authors are with the CNRS-University of Montpellier, LIRMM, Interactive Digital Humans group, France: niels.dehio@lirmm.fr

A. Kheddar is also with the CNRS-AIST Joint Robotics Laboratory, IRL, Japan.

in a single framework. While these works focus on contact point forces and point deformations only, [13] models the entire surface's deformation probabilistically. However, this model does not incorporate resulting contact forces.

*Control Schemes:* The control of contact transitions with soft material was studied extensively over the last decades, mainly to avoid bouncing, e.g., [14] for soft and rigid contacts. The approaches in [15] and [16] rely on a mass-spring interaction model and derive a continuous control law to stabilize the transition between non-contact and contact phase, without requiring force measurements. The analysis and experiments are, however, limited to a planar 2 DOF robot. More recently, the approaches described in [17], [18] and [19] propose parameter optimization for the more complex mass-spring-damper interaction model (Kelvin-Voigt model [20]). Non-linear tangential forces arising from soft interaction have been controlled with a simple toy robot in simulation [21]. Approaches with floating-base legged robots typically try to impose robust control that compensates for non-modeled soft contact properties, for example, with a quadruped in [22] and with a humanoid in [23], [24]. As an alternative to robust control schemes, [25] proposed an impact-aware planning method.

To the authors' knowledge, related works on impact or task-space model-predictive control did not consider the robot's structural hardware limits defined in joint-space, which implicitly represent bounds in the contact-space. A first step in this direction has been proposed in [26] for a reduced model, and later in [27], [28] for a non-redundant three degrees of freedom robot leg without considering impacts. Yet, none of these related works is explicitly aware of (or tries to optimize) the maximum possible impact velocity.

Our novel approach overcomes these two shortcomings: First, we map hardware limits onto the contact-space for redundant robots. Second, we maximize the pre-impact end-effector velocity subject to constrained deformation dynamics considering a reasonable long preview horizon for predicting accurate behaviors. Exploiting the shock-absorbing soft material, our contribution enables us to generate maximum but robot-safe impacts intentionally.

### III. DYNAMIC MODEL

Let the manipulator dynamics in joint-space governed by  $\tau + \mathbf{J}^T \mathbf{f} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}$ , and limited by  $\underline{\mathbf{q}} \leq \mathbf{q} \leq \bar{\mathbf{q}}$ ,  $\underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}}$ ,  $\underline{\ddot{\mathbf{q}}} \leq \ddot{\mathbf{q}} \leq \bar{\ddot{\mathbf{q}}}$ ,  $\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \bar{\boldsymbol{\tau}}$ ,  $\mathbf{0} \leq \mathbf{f} \leq \infty$ , where  $\mathbf{M}$  denotes the joint space inertia,  $\ddot{\mathbf{q}}$  are the joint accelerations,  $\mathbf{h}$  comprises gravity and Coriolis forces,  $\mathbf{J}$  constitutes the end-effector Jacobian which we assume to be full-rank in this letter, and  $\mathbf{f}$  is an unilateral contact force at the end-effector. Symbols  $\bar{\star}$ ,  $\underline{\star}$  denote upper and lower limits of a quantity  $\star$ , respectively. The structural hardware limits must be fulfilled to generate a feasible and robot-safe motion.

When gathered, the previous equations form a convex, high-dimensional polytope that represents the set of feasible

solutions in terms of  $[\dot{\mathbf{q}}^T, \ddot{\mathbf{q}}^T, \mathbf{f}^T]^T$ . It is given by<sup>1</sup>

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \boldsymbol{\tau} - \mathbf{h} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \mathbf{I}, & \mathbf{0}, & \mathbf{0} \\ \mathbf{0}, & \mathbf{I}, & \mathbf{0} \\ \mathbf{0}, & \mathbf{M}, & -\mathbf{J}^T \\ \mathbf{0}, & \mathbf{0}, & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \mathbf{f} \end{bmatrix} \leq \begin{bmatrix} \bar{\dot{\mathbf{q}}} \\ \bar{\ddot{\mathbf{q}}} \\ \bar{\boldsymbol{\tau}} - \mathbf{h} \\ \infty \end{bmatrix} \quad (1)$$

#### A. Deformation Dynamics

Soft material deforms under externally applied forces (stress). As a consequence, contact forces are functions of surface deformations [5]. In the following, we limit our analysis to soft materials and associated models with non-varying contact dynamics over time, i.e., obeying Hook's law of deformation. Furthermore, we assume that contact friction constraints are always satisfied during the penetration phase; i.e., there is no slipping (as this is what should be planned for the operations we target). We deal with one-dimensional deformation in the contact normal direction because the planned impacts are also along the contact normal.

Consider a reference position  $\mathbf{x}$  of a robotic end-effector. With respect to this reference, relative positions and velocities are denoted as  $\mathbf{z}, \dot{\mathbf{z}}$ , respectively. This notation is advantageous when considering deformations caused by the robot and choosing as reference  $\mathbf{x}$  the end-effector position that coincides with the soft material's initial contact position. In that case,  $\mathbf{z}$  describes the deformation and  $\dot{\mathbf{z}}$  the deformation rate of change. No deformation corresponds to  $\mathbf{z} = \mathbf{0}$ .

The material's constant, finite stiffness and damping are given by parameters  $\boldsymbol{\alpha}, \boldsymbol{\beta}$ , which will be identified with the approach described in Sec. V. We employ the well-known Kelvin-Voigt model [20] (a linear mass-spring-damper system) defining the resulting force  $\mathbf{f} = \mathbf{m}\ddot{\mathbf{z}}$  as a linear combination of contact deformation  $\mathbf{z}$ , and its derivative  $\dot{\mathbf{z}}$

$$\mathbf{f} = \boldsymbol{\alpha}\mathbf{z} + \boldsymbol{\beta}\dot{\mathbf{z}}, \text{ or equivalently formulated as } \mathbf{f} = \mathbf{E}\mathbf{s} \quad (2)$$

with the deformation-dependent state  $\mathbf{s} = [\mathbf{z}^T, \dot{\mathbf{z}}^T]^T$  and a constant matrix  $\mathbf{E} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$  representing the material's stiffness and damping properties. When applying the control input  $\mathbf{u} = \mathbf{J}\mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{h})$  as a function of joint torques  $\boldsymbol{\tau}$ , the second-order differential equation writes

$$\ddot{\mathbf{z}} = \mathbf{m}^{-1}\boldsymbol{\alpha}\mathbf{z} + \mathbf{m}^{-1}\boldsymbol{\beta}\dot{\mathbf{z}} + \mathbf{u} \quad (3)$$

In this paper,  $\mathbf{m} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$  is the effective mass associated with the penetrating end-effector [29]. The continuous-time state-space representation becomes  $\dot{\mathbf{s}} = \mathbf{A}_c\mathbf{s} + \mathbf{B}_c\mathbf{u}$  with

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{m}^{-1}\boldsymbol{\alpha} & \mathbf{m}^{-1}\boldsymbol{\beta} \end{bmatrix} \text{ and } \mathbf{B}_c = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \quad (4)$$

Considering a negligible change in the operational apparent mass (see later its variation curve in Fig. 4), we assume constant mass  $\mathbf{m}$  for small changes in the robot configuration due to relatively small indentations, and hence, constant  $\mathbf{A}_c$ .

Through discretization, we obtain the discrete-time state-space representation

$$\mathbf{s}_{i+1} = \mathbf{A}_d\mathbf{s}_i + \mathbf{B}_d\mathbf{u}_i \quad (5)$$

<sup>1</sup>In this paper we ignore  $\underline{\mathbf{q}} \leq \mathbf{q} \leq \bar{\mathbf{q}}$  without loss of generality.

where subscript  $i$  denotes steps at time  $t = i\Delta t$  with sample time  $\Delta t$ . Note that the resulting discretized acceleration  $\ddot{\mathbf{z}}_i$  is linear in terms of state  $\mathbf{s}_i$  and control input  $\mathbf{u}_i$ . Refer to the appendix for the derivation of matrices  $\mathbf{C}$  and  $\mathbf{D}$

$$\ddot{\mathbf{z}}_i = \mathbf{C}\mathbf{s}_i + \mathbf{D}\mathbf{u}_i \quad (6)$$

This deformation dynamics model described so far is well-known. In the next subsection, we propose a novel extension by deriving bounds on the system behavior, related to the particular penetrating manipulator.

### B. Constrained Deformation Dynamics

The above deformation dynamics are constrained when considering a penetration by the robot's end-effector, whose motion is limited by the hardware features. Task-space (or contact-space) bounds are typically configuration-dependent and result from mapping hardware limits, which are provided in joint-space.

In the following, motion objectives to be controlled in the end-effector's nullspace are incorporated via joint velocity and acceleration references  $\dot{\mathbf{q}}_0, \ddot{\mathbf{q}}_0$ . Such selfmotion is enforced by applying the nullspace projector  $\mathbf{N}$  associated with the one-dimensional contact-space Jacobian  $\mathbf{J}$

$$\mathbf{N}\dot{\mathbf{q}} = \dot{\mathbf{q}}_0 \text{ and } \mathbf{N}\ddot{\mathbf{q}} + \dot{\mathbf{N}}\dot{\mathbf{q}} = \ddot{\mathbf{q}}_0 \quad (7)$$

Let us assume  $\dot{\mathbf{N}}\dot{\mathbf{q}} \simeq \mathbf{0}$  for simplicity. Reformulating (7) subject to the decision variables in (1) yields

$$\begin{bmatrix} \mathbf{N} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \ddot{\mathbf{q}}_0 \end{bmatrix} \quad (8)$$

We are interested in the configuration-dependent task-space bounds with respect to end-effector (= deformation)-velocity, acceleration, and force in contact normal direction:

- The lower bound on the task-space velocity  $\dot{\mathbf{z}}$  becomes

$$\begin{aligned} \dot{\mathbf{z}} &= \min_{\dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}} \mathbf{J}\dot{\mathbf{q}} \\ \text{s. t. } &(1) \text{ and } (8) \end{aligned} \quad (9)$$

which can be simplified because the joint velocities are decoupled from both joint accelerations and force (refer to the block-diagonal matrices in (1) and (8))

$$\begin{aligned} \dot{\mathbf{z}} &= \min_{\dot{\mathbf{q}}} \mathbf{J}\dot{\mathbf{q}} \\ \text{s. t. } &\mathbf{N}\dot{\mathbf{q}} = \dot{\mathbf{q}}_0 \\ &\dot{\mathbf{q}} \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}} \end{aligned} \quad (10)$$

The upper velocity bound  $\bar{\dot{\mathbf{z}}}$  is obtained by replacing the min with max operator. The final deformation velocity constraint yields

$$\dot{\mathbf{z}} \leq \dot{\mathbf{z}} \leq \bar{\dot{\mathbf{z}}} \quad (11)$$

- The contact-space acceleration and force bounds are coupled and cannot be treated separately. Also note that the set given by (1) and (8) does not contain  $\ddot{\mathbf{z}}$  explicitly. Accordingly, we add the contact-space acceleration as a decision variable and introduce the relationship  $\ddot{\mathbf{z}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}$ . By choosing a vector  $\omega$  (treated as a ray that points in

a certain direction), we obtain a feasible tuple (or vertex) consisting of an extreme  $\ddot{\mathbf{z}}$  and  $\mathbf{f}$  through the linear program

$$\begin{aligned} \arg \min_{\ddot{\mathbf{q}}, \mathbf{f}, \ddot{\mathbf{z}}} & [\mathbf{f}, \ddot{\mathbf{z}}] \omega \\ \text{s. t. } & \begin{bmatrix} \mathbf{N} & \mathbf{0} & \mathbf{0} \\ \mathbf{J} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{f} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{q}}_0 \\ -\mathbf{J}\dot{\mathbf{q}}_{\text{cur}} \end{bmatrix} \\ & \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\tau} - \mathbf{h} \\ \mathbf{0} \end{bmatrix} \leq \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{M} & -\mathbf{J}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{f} \\ \ddot{\mathbf{z}} \end{bmatrix} \leq \begin{bmatrix} \ddot{\mathbf{q}} \\ \bar{\boldsymbol{\tau}} - \mathbf{h} \\ \infty \end{bmatrix} \end{aligned} \quad (12)$$

where  $\dot{\mathbf{q}}_{\text{cur}}$  are the current joint velocities. Solving (12) multiple times for different rays  $\omega$ , we collect a list of extreme vertices and halfplanes, representing a convex polytope. Refer to [30] for an efficient algorithm to select useful search directions (rays). The resulting halfspace representation is given by matrix  $\mathbf{P}$  (halfplanes) and vector  $\mathbf{p}$  (offset)

$$\mathbf{P} \begin{bmatrix} \ddot{\mathbf{z}} \\ \mathbf{f} \end{bmatrix} \leq \mathbf{p} \quad (13)$$

- The contact deformation  $\mathbf{z}$  satisfies scalar lower and upper bounds which do not relate to robot's hardware limits

$$\mathbf{z} \leq \mathbf{z} \leq \bar{\mathbf{z}} \quad (14)$$

In order not to break the contact and not to damage the material, these bounds are specified by  $\mathbf{0}$ , and the maximum allowed penetration depending on the material properties.

Due to these contact-space constraints, not all possible deformation trajectories (i.e., acceleration profiles) are physically achievable with particular robot hardware, which is an often-ignored fact. However, knowledge of the constrained deformation dynamics is of utmost importance for the model-preview proposed in the next section.

## IV. SOFT IMPACT-AWARE PREVIEW CONTROL

In this section, we formulate a problem for planning an optimized penetration-trajectory assuming the contact event will happen in the next iteration. In order to maximize the pre-impact end-effector velocity, we consider the initial velocity state as part of the optimization variables. Tracking this velocity upper bound with a velocity controller enables us to intentionally generate feasible and safe impacts with maximum pre-impact velocity.

### A. Penetration-Trajectory Planning

Let us formulate an optimization problem that plans a feasible and safe penetration-trajectory over a preview horizon of  $h$  steps, starting from the exact moment of the contact transition  $\mathbf{z}_1 = \mathbf{0}$  with the pre-impact velocity  $\dot{\mathbf{z}}_1$ , which is to be maximized. The optimization variables consist of the initial velocity state  $\dot{\mathbf{z}}_1$  and the subsequent control inputs  $\mathbf{u}_1, \dots, \mathbf{u}_h$ , concatenated into a vector  $\mathbf{U} = [\dot{\mathbf{z}}_1, \mathbf{u}_1, \dots, \mathbf{u}_h]^T$ . Based on these decision variables, the following constrained problem solver outputs also – indirectly through feedforward simulation – the evolution of deformation states  $\mathbf{s}_1, \dots, \mathbf{s}_h$  via (5), as well as predicted

accelerations  $\ddot{\mathbf{z}}_1, \dots, \ddot{\mathbf{z}}_h$  via (6), and contact forces  $\mathbf{f}_1, \dots, \mathbf{f}_h$  via (2).

$$\begin{aligned} & \arg \min_{\mathbf{U}} c(\mathbf{U}) \\ & \text{subject to} \quad \mathbf{s}_1 = \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{z}}_1 \end{bmatrix}, \quad \mathbf{s}_i = \begin{bmatrix} \mathbf{z}_i \\ \dot{\mathbf{z}}_i \end{bmatrix} \\ & \quad \mathbf{s}_{i+1} = \mathbf{A}_d \mathbf{s}_i + \mathbf{B}_d \mathbf{u}_i \\ & \quad \mathbf{z} \leq \mathbf{z}_i = [\mathbf{I}, \mathbf{0}] \mathbf{s}_i \leq \bar{\mathbf{z}} \\ & \quad \dot{\mathbf{z}} \leq \dot{\mathbf{z}}_i = [\mathbf{0}, \mathbf{I}] \mathbf{s}_i \leq \dot{\bar{\mathbf{z}}} \\ & \quad \mathbf{P} \begin{bmatrix} \dot{\mathbf{z}}_i \\ \mathbf{f}_i \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{C} \mathbf{s}_i + \mathbf{D} \mathbf{u}_i \\ \mathbf{E} \mathbf{s}_i \end{bmatrix} \leq \mathbf{p} \\ & \quad \dot{\mathbf{z}}_h = \mathbf{0} \end{aligned} \quad (15)$$

The optimization problem consists of multiple constraints:

- The state is composed of contact deformation and its rate of change:  $\mathbf{s}_i = [\mathbf{z}_i^T, \dot{\mathbf{z}}_i^T]^T$ .
- Deformation dynamics are satisfied:  $\mathbf{s}_{i+1} = \mathbf{A}_d \mathbf{s}_i + \mathbf{B}_d \mathbf{u}_i$ .
- Predicted positions  $\mathbf{z}_i$ , velocities  $\dot{\mathbf{z}}_i$ , accelerations  $\ddot{\mathbf{z}}_i$ , and forces  $\mathbf{f}_i$  are restricted to a feasible region, defined by the task-space bounds derived in the previous section. There is no need to constrain the control inputs  $\mathbf{u}_i$ .
- The trajectory terminates safely by enforcing a zero velocity in the terminal state  $\dot{\mathbf{z}}_h = \mathbf{0}$ .

In order to maximize the pre-impact velocity  $\dot{\mathbf{z}}_1$ , one may treat (15) as a linear program with the cost  $c_1(\mathbf{U}) = [-1, 0_1, \dots, 0_h] \mathbf{U}$ , which is fast to compute. However, it is important to note: given the maximum feasible pre-impact velocity, there may exist infinite solutions in terms of subsequent control inputs  $\mathbf{u}_1, \dots, \mathbf{u}_h$  that all result in a feasible, safe penetration behavior. Imposing the linear cost  $c_1(\mathbf{U})$  does not yield a unique solution. This is not problematic as long as we are only interested in the optimal pre-impact velocity. In other respects, this kind of redundancy is resolved by imposing a secondary objective  $c_2(\mathbf{U})$  in the nullspace of the primary objective  $c_1(\mathbf{U})$ . For example, a reasonable secondary objective is to minimize the overall control effort via the quadratic cost  $c_2(\mathbf{U}) = \sum_{i=1}^h \|\mathbf{u}_i\|^2$ . In the experiments later-on, the planned trajectory targets to reach a desired contact force  $\mathbf{f}_{\text{des}}$  by minimizing the force error in every iteration  $c_2(\mathbf{U}) = \sum_{i=1}^h \|\mathbf{f}_{\text{des}} - \mathbf{E} \mathbf{s}_i\|^2$ .

The COPRA library<sup>2</sup> allows us to formulate linear MPC problems conveniently. However, so far it restricts the user to optimization of control inputs  $\mathbf{u}_1, \dots, \mathbf{u}_h$  only. We extended the COPRA library to include also the optimization of the initial system state, i.e. the initial  $\dot{\mathbf{z}}_1$ . More technical details are provided via the website.

### B. Final Control Scheme

Without knowledge of contact location or contact time, our approach is to expect the impact event to happen in the next time step, in the direction of end-effector motion. Consequently, we solve (15) in each control cycle to obtain the maximum safe pre-impact velocity. The robot hardware and the task achievement are secured as long as the end-effector

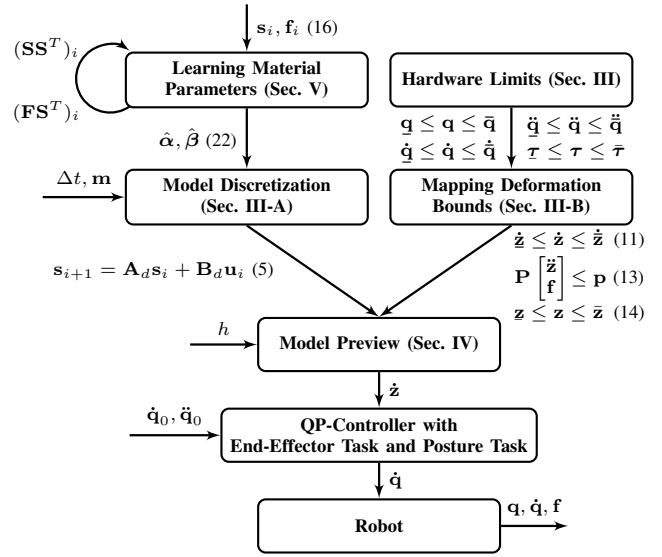


Fig. 2: Illustration of the proposed control scheme (before making contact) for intentional impact maximization.

motion respects this pre-impact velocity bound. Accordingly, we generate the maximum safe intentional impact by tracking the optimized  $\dot{\mathbf{z}}_1$  with the end-effector.

Note that the mapping between task-space and joint space is assumed locally constant during the preview. This implies that the Jacobian, joint-space inertia, gravity compensation term, and Coriolis forces are assumed with small variation during the penetration process, which is a feasible assumption for small deformations and a short horizon.

The final control scheme (c.f Fig 2 and Alg 1) requires the material's stiffness and damping to be known. The next section explains how to identify these parameters autonomously. Incremental learning allows us to update the material model during the penetration phase online.

## V. LEARNING MATERIAL DEFORMATION PROPERTIES

The soft material properties need to be learned (and updated) in a data-driven way unless precise knowledge is a priori available [11]. In order to collect the required data, the robot has to penetrate the deformable contact surfaces. Such exploratory motion can be generated without knowledge of the actual deformation model by tracking a pre-planned deformation trajectory that excites the soft contact properties.

### A. Offline-Learning

Given the robot's joint configuration, in each  $i$ -th control cycle, the contact state  $\mathbf{s}_i$  is recorded via forward kinematics, and the contact force  $\mathbf{f}_i$  is obtained through a wrist-mounted

---

#### Algorithm 1 pseudo code of proposed control scheme

---

- 1: **procedure** CONTROLLER
  - 2:   update model (22)
  - 3:   discretize deformation dynamics (5)
  - 4:   obtain task-space bounds (11) via (10), (13) via (12), (14)
  - 5:   plan penetration-trajectory (15)
  - 6:   solve multi-task QP-controller
  - 7:   **return** control command
- 

<sup>2</sup><https://github.com/jrl-umi3218/copra>

force-torque sensor or via external joint torque measurements  $\mathbf{f}_i = (\mathbf{J}_i^T)^\dagger \boldsymbol{\tau}_i$ , with the Moore-Penrose pseudoinverse  $(\cdot)^\dagger$ . All  $K$  recorded data points are concatenated into matrix form

$$\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K] \quad \text{and} \quad \mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_K] \quad (16)$$

In view of (2), employing linear least-square regression with scalar model error

$$\mathbf{e} = \frac{1}{K} \left\| \mathbf{F}^T - \mathbf{S}^T [\boldsymbol{\alpha}, \boldsymbol{\beta}]^T \right\|^2 \quad (17)$$

allows us to estimate the material's stiffness and damping  $\hat{\mathbf{E}}$

$$\hat{\mathbf{E}} = [\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}] = \mathbf{F}\mathbf{S}^\dagger = \mathbf{F}\mathbf{S}^T (\mathbf{S}\mathbf{S}^T)^{-1} \quad (18)$$

where  $(\cdot)^\dagger$  minimizes the Euclidean norm (the model error  $\mathbf{e}$ ). Adding a regularization term ensures numerical stability.

For new states  $\mathbf{S}$ , the corresponding contact forces  $\hat{\mathbf{F}}$  are predicted through simple multiplication with the learned  $\hat{\mathbf{E}}$

$$\hat{\mathbf{F}} = [\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \dots, \hat{\mathbf{f}}_K] = \hat{\mathbf{E}}\mathbf{S} \quad (19)$$

We can expect  $\mathbf{f} = \hat{\mathbf{f}}$ , and hence  $\mathbf{F} = \hat{\mathbf{F}}$ , when assuming zero sensor noise and the correct choice of  $\hat{\mathbf{E}}$  (and, of course, assuming that the linear model represents the reality).

### B. Online-Learning via Recursive Formulation

Given a large amount of training data, (18) can only be applied for off-line learning because of the time-consuming matrix inversion. We here extend [11] by employing the recursive least-squares algorithm [31, page 196] that is suitable for model updates in real-time. Given a new data point consisting of  $\mathbf{s}_i$  and  $\mathbf{f}_i$ , we update in each control cycle  $i > 0$  the intermediate terms (denoted in subscripts)

$$(\mathbf{S}\mathbf{S}^T)_i = (\mathbf{S}\mathbf{S}^T)_{i-1} + \mathbf{s}_i\mathbf{s}_i^T \quad (20)$$

$$(\mathbf{F}\mathbf{S}^T)_i = (\mathbf{F}\mathbf{S}^T)_{i-1} + \mathbf{f}_i\mathbf{s}_i^T \quad (21)$$

with initialization  $(\mathbf{S}\mathbf{S}^T)_0 = \mathbf{0}$  and  $(\mathbf{F}\mathbf{S}^T)_0 = \mathbf{0}$ . Note that the dimensionality of these intermediate terms is low and stays constant. The material model is improved within the real-time control loop by

$$\left(\hat{\mathbf{E}}\right)_i = (\mathbf{F}\mathbf{S}^T)_i (\mathbf{S}\mathbf{S}^T)_i^{-1} \quad (22)$$

The incremental update is computationally not demanding and its result is identical with (18).

## VI. REAL-ROBOT EXPERIMENTS

*Setup:* The experimental platform is a 7 degrees-of-freedom Panda manipulator from FrankaEmika, controlled at 1 ms update rate based on our existing QP-control framework [32]. We attached a 3D-printed end-effector with a flat circular contact surface (diameter 0.13 m). The robot's hardware limits are specified on the manufacturer's website<sup>3</sup>. For reproducibility, we ordered a similar dice as chosen by [11] and also selected the vertical axis as the direction of motion, typical for bin-picking scenarios. The soft dice

<sup>3</sup>[https://frankaemika.github.io/docs/control\\_parameters.html#constants](https://frankaemika.github.io/docs/control_parameters.html#constants)

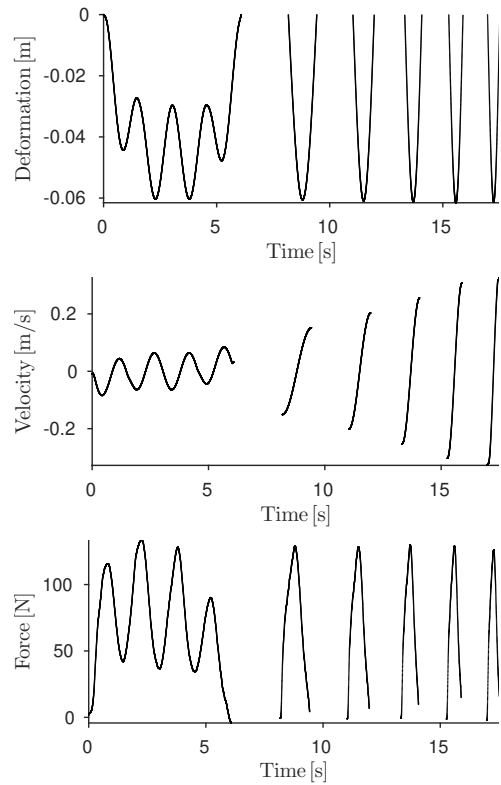


Fig. 3: Recorded data points for *dice* model identification.

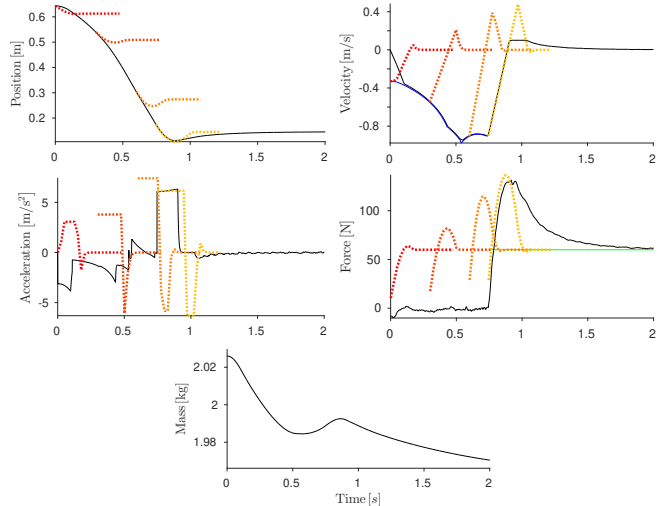


Fig. 4: Intentionally generating the maximum impact for the soft dice employing the proposed impact-aware controller (solid black lines). All values are represented in task-space. The dark-blue curve shows the maximum feasible pre-impact velocity at a given time step that is tracked (before making contact) and green the desired contact force of 60 N (after making contact). Planned trajectories are plotted at 0, 0.3, and 0.6 s and when detecting the contact at 0.75 s with color fading from red into orange (dotted lines). During the whole experiment, the effective mass varies by 0.05 kg. For comparison, the baseline approach with contact event at 1.38 s is shown with dashed cyan lines. The abscissa axis denotes time in all sub-figures.

weights 0.1 kg and has 0.16 m edge length. In additional experiments, we attached to the end-effector a Schmalz pump with two different flexible suction cups (length 0.03 m and 0.05 m) and use various other rigid and soft objects, c.f Fig 1. Therefore, our new interface implementation operates Panda and pump simultaneously at different control frequencies<sup>4</sup>.

*Learning Procedure:* Material properties are learned by tracking a predefined (sinusoidal) penetration-trajectory using position-control and high gains followed-up by five predefined impacts with varying pre-impact velocities (ranging from 0.15 m/s to 0.35 m/s). The final database for the dice contains more than ten seconds of penetration data. Fig 3 plots the deformation states  $\mathbf{z}_i$ ,  $\dot{\mathbf{z}}_i$  observed through forward kinematics and associated contact forces  $\mathbf{f}_i$  obtained via Panda’s built-in force sensor signal. Control cycles without contact are excluded from the dataset. We also identify the soft material parameters for two different flexible suction cups attached to the pump and the soft objects (please refer also to the video <https://youtu.be/juynq6x9JJ8>).

*Generating Impact:* We compute six vertices/halfplanes for (13) in each iteration employing the algorithm described in [30], which is sufficient to obtain an accurate estimate of the force-acceleration dependency. The optimization problem (15) is parameterized with  $h = 20$  steps and the discretization sample time  $\Delta t = 25$  ms, resulting in the preview horizon  $T = 0.5$  s and 204 inequality constraints. These parameters were selected based on preliminary experiments and constitute a trade-off between short time-steps (required for precise / accurate trajectory planning), a long preview horizon (required for robot safety), and fast computation (required for real-time application). We solve it in every control cycle at 1 kHz. These different frequencies are legitimate, as the optimization result constitutes an upper bound on the pre-impact velocity and should be as up-to-date as possible. A primary end-effector velocity task tracks the maximum possible pre-impact velocity. Besides, a secondary task enforces  $\dot{\mathbf{q}}_0 = \ddot{\mathbf{q}}_0 = \mathbf{0}$  to account for the redundancy, thereby avoiding elbow self-motion, which could affect the end-effector’s effective mass (Fig 2). We select 2 N as a threshold for contact detection. Afterward, we switch to an admittance controller [32] that tracks the reference force 60 N. We decided not to track the planned trajectory, as the admittance controller also incorporates the force sensor signal and thereby accounts for material model inaccuracies. More research is needed to incorporate the force measurements into the state space representation (5).

We conduct multiple maximum impact-experiments with various soft objects, starting in different configurations, with varying desired final contact forces. We here discuss only one experiment with the soft dice due to space limitations – please refer to the video for the other experiments. The robot end-effector starts approximately 0.46 m above the contact. Analyzing the logged data, hardware limits are satisfied in all trials. After 0.75 s, we detect the contact with a pre-impact end-effector velocity of 0.898 m/s. Steady-state is

reached after approximately 0.8 s of deformation, thereby confirming the chosen preview duration. The final estimated force is close to the desired force. Fig 4 also shows the planned trajectories at a few exemplary iterations (every 0.3 s and at the contact event). As discussed above, these are not necessarily unique: the admittance controller realizes a (slightly) different motion that takes longer to converge.

## VII. CONCLUSION AND FUTURE WORK

In order to speed up industrial automation processes that require contact such as grabbing, contact transitions should be accelerated, and hence, contact formations shall be made with impact under robot and task integrity. When evaluating the effect of the pre-impact end-effector velocity for soft contacts, constrained deformation dynamics must be considered, describing the system behavior after making high-velocity contact. This is a significant difference compared to impacts with a rigid material, where we have shown in our previous work that, although conservative, a single-step ahead prediction is sufficient for safe impact-aware control [4].

In this paper, we learn a contact force model for deformable contacts based on exploratory penetration data with impacts that is recursively updated during penetration phases. Next, we propose to map the robot’s hardware limits onto the contact-space to obtain constrained deformation dynamics. This allows us to plan constrained task-space trajectories for powerful intentional impacts: In the experiments, we track the maximum allowed pre-impact end-effector velocity without compromising the robot’s limitations during the deformation phase. Our novel control scheme is independent of whether the soft material is attached to the end-effector (e.g., a deformable suction cup) or part of the environment (e.g., a deformable sponge) and operates at a 1 kHz update rate on the redundant Panda manipulator.

We are currently extending our multi-arm object manipulation approach [33] to cope with multiple soft impacts. Future work will also tackle the planning of pre-impact nullspace postures, influencing the end-effector’s effective mass [29].

## APPENDIX

### A. Task-space acceleration $\ddot{\mathbf{z}}_i$ depends linearly on $\mathbf{s}_i$ and $\mathbf{u}_i$

Let us denote the matrix elements of matrices  $\mathbf{A}_d, \mathbf{B}_d$  as

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{a}_{1,1} & \mathbf{a}_{1,2} \\ \mathbf{a}_{2,1} & \mathbf{a}_{2,2} \end{bmatrix} \text{ and } \mathbf{B}_d = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

Hence,  $\dot{\mathbf{z}}_{i+1} = \mathbf{a}_{2,1}\mathbf{z}_i + \mathbf{a}_{2,2}\dot{\mathbf{z}}_i + \mathbf{b}_2\mathbf{u}_i$ . The acceleration  $\ddot{\mathbf{z}}_i$  is linear in terms of deformation state  $\mathbf{s}_i$  and control input  $\mathbf{u}_i$

$$\begin{aligned} \ddot{\mathbf{z}}_i &= \frac{1}{\Delta t} (\dot{\mathbf{z}}_{i+1} - \dot{\mathbf{z}}_i) = \frac{1}{\Delta t} [\mathbf{a}_{2,1}\mathbf{z}_i + (\mathbf{a}_{2,2} - 1)\dot{\mathbf{z}}_i + \mathbf{b}_2\mathbf{u}_i] \\ &= \begin{bmatrix} \mathbf{a}_{2,1} & \mathbf{a}_{2,2}-1 \end{bmatrix} \frac{1}{\Delta t} \begin{bmatrix} \mathbf{z}_i \\ \dot{\mathbf{z}}_i \end{bmatrix} + \begin{bmatrix} \mathbf{b}_2 \end{bmatrix} \mathbf{u}_i = \mathbf{C}\mathbf{s}_i + \mathbf{D}\mathbf{u}_i \end{aligned}$$

## ACKNOWLEDGMENTS

The authors thank Dr. Pierre Gergondet for his support and Julien Roux for providing a C++ implementation of [30].

This work was supported by the Research Project I.A.M. through the European Union H2020 program (GA 871899).

<sup>4</sup>[https://github.com/jrl-umi3218/mc\\_franka](https://github.com/jrl-umi3218/mc_franka)

## REFERENCES

- [1] H. Pham and Q.-C. Pham, "Critically fast pick-and-place with suction cups," in *IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 3045–3051.
- [2] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossing-bot: Learning to throw arbitrary objects with residual physics," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1307–1319, 2020.
- [3] Y. Wang and A. Kheddar, "Impact-friendly robust control design with task-space quadratic optimization," in *Proceedings of Robotics: Science and Systems*, 2019.
- [4] Y. Wang, N. Dehio, A. Tanguy, and A. Kheddar, "Impact-aware task-space quadratic-programming control," *arxiv*, 2020. [Online]. Available: <https://arxiv.org/abs/2006.01987>
- [5] N. Xydias and I. Kao, "Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results," *Int. Journal of Robotics Research*, vol. 18, no. 9, pp. 941–950, 1999.
- [6] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and Machine Theory*, vol. 37, no. 10, pp. 1213–1239, 2002.
- [7] F. Nadon, A. J. Valencia, and P. Payeur, "Multi-modal sensing and robotic manipulation of non-rigid objects: A survey," *Robotics*, vol. 7, no. 4, 2018.
- [8] K. Bouyarmane and A. Kheddar, "FEM-based static posture planning for a humanoid robot on deformable contact support," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2011, pp. 487–492.
- [9] G. De Magistris, A. Pajon, S. Miossec, and A. Kheddar, "Humanoid walking with compliant soles using a deformation estimator," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1757–1762.
- [10] E. Coevoet, A. Escande, and C. Duriez, "Soft robots locomotion and manipulation control using fem simulation and quadratic programming," in *Int. Conf. on Soft Robotics*, 2019, pp. 739–745.
- [11] M. Azad, V. Ortenzi, H.-C. Lin, E. Rueckert, and M. Mistry, "Model estimation and control of compliant contact normal force," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2016, pp. 442–447.
- [12] G. Ganesh, N. Jarrass, S. Haddadin, A. Albu-Schäffer, and E. Burdet, "A versatile biomimetic controller for contact tooling and haptic exploration," in *IEEE Int. Conf. on Robotics and Automation*, 2012, pp. 3329–3334.
- [13] S. Caccamo, P. Gler, H. Kjellström, and D. Kragic, "Active perception and modeling of deformable surfaces using gaussian processes and position-based dynamics," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2016, pp. 530–537.
- [14] W. L. Xu, J. D. Han, and S. K. Tso, "Experimental study of contact transition control incorporating joint acceleration feedback," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 3, pp. 292–301, 2000.
- [15] G. Hu, C. Makkar, and W. E. Dixon, "Energy-based nonlinear control of underactuated euler-lagrange systems subject to impacts," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1742–1748, 2007.
- [16] C.-H. Liang, S. Bhasin, K. Dupree, and W. E. Dixon, "A force limiting adaptive controller for a robotic system undergoing a noncontact-to-contact transition," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 6, pp. 1330–1341, 2009.
- [17] R. Zotovic Stanisic and Á. Valera Fernández, "Adjusting the parameters of the mechanical impedance for velocity, impact and force control," *Robotica*, vol. 30, no. 4, pp. 583–597, 2012.
- [18] D. Heck, A. Saccon, N. van de Wouw, and H. Nijmeijer, "Guaranteeing stable tracking of hybrid positionforce trajectories for a robot manipulator interacting with a stiff environment," *Automatica*, vol. 63, pp. 235 – 247, 2016.
- [19] V. Samy, K. Bouyarmane, and A. Kheddar, "Analysis of a simple model for post-impact dynamics active compliance in humanoids falls with nonlinear optimization," in *IEEE Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, 2018, pp. 62–67.
- [20] W. Flügge, "Viscoelasticity," *Waltham, Mass.:Blaisdell Publishing Company*, 1967.
- [21] M. Azad and M. Mistry, "Balance control strategy for legged robots with compliant contacts," in *IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 4391–4396.
- [22] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, "Stance: Locomotion adaptation over soft terrain," *arxiv*, 2019. [Online]. Available: <https://arxiv.org/abs/1904.12306>
- [23] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *Int. Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.
- [24] G. Mesesan, J. Engelsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control," in *IEEE/RAS Int. Conf. on Humanoid Robots*, 2019.
- [25] T. Stouraitis, L. Yan, J. Moura, M. Gienger, and S. Vijayakumar, "Multi-modal trajectory optimization for impact-aware manipulation," *arxiv*, 2020. [Online]. Available: <https://arxiv.org/abs/2006.13374>
- [26] V. Samy, S. Caron, K. Bouyarmane, and A. Kheddar, "Post-impact adaptive compliance for humanoid falls using predictive control of a reduced model," in *IEEE-RAS International Conference on Humanoid Robotics*, 2017, pp. 655–660.
- [27] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, "Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3363–3370, 2018.
- [28] R. Orsolino, M. Focchi, S. Caron, G. Raiola, V. Barasuol, D. G. Caldwell, and C. Semini, "Feasible region: An actuation-aware extension of the support region," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1239–1255, 2020.
- [29] N. Mansfeld, B. Djellab, J. R. Veuthey, F. Beck, C. Ott, and S. Haddadin, "Improving the performance of biomechanically safe velocity control for redundant robots through reflected mass minimization," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5390–5397.
- [30] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 794–807, 2008.
- [31] E. K. P. Chong and S. H. Zak, *An introduction to optimization, 2nd Edition*. John Wiley & Sons, 2001.
- [32] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, 2019.
- [33] N. Dehio, J. Smith, D. L. Wigand, G. Xin, H.-C. Lin, J. J. Steil, and M. Mistry, "Modeling and Control of Multi-Arm and Multi-Leg Robots: Compensating for Object Dynamics during Grasping," in *IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 294–301.