Frictional Contact-Implicit Inverse Dynamics

Etienne Ménager[1], Pierre Fabre[1], Antoine Bambade[2], Wilson Jallet[1], Alberto De Marchi[3] and Justin Carpentier[1]

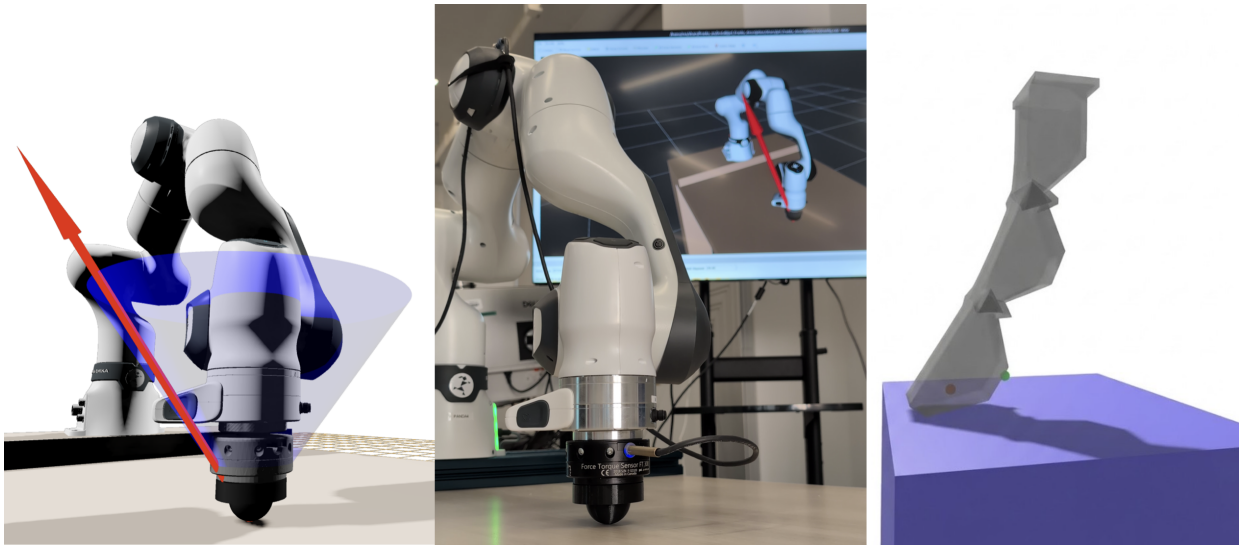


Fig. 1: **Applications of the proposed frictional contact-implicit inverse dynamics. Franka Panda arm (left/center).** The manipulator tracks both a given end-effector position and desired contact force in the z-direction in simulation (**left**) and on real-world hardware (**center**). **Soft finger (right).** A soft finger interacts with a box by establishing, maintaining, and breaking contact to reach a translational goal, illustrating frictional, contact-aware control on a deformable system.

***Abstract*—Task-space inverse dynamics is a versatile paradigm for real-time robot control, capable of handling under-actuation and contact interactions across diverse platforms from humanoids to deformable robots. However, in practice, contact-aware controllers typically assume predefined contact sequences, inherently limiting their ability to automatically select among breaking, sliding, or sticking contacts. In this work, we extend the classical quadratic programming formulation of inverse dynamics to a quadratic program with complementarity constraints (QPCC). This formulation natively accounts for actuator limits and frictional contacts, modeled as nonlinear complementarity constraints. To solve these QPCC problems, we propose an iterative optimization approach based on alternating minimization and augmented Lagrangian methods. Our solver alternates between minimizing a smooth, convex function that captures task objectives and system dynamics, and projecting onto sets that enforce actuator and nonconvex frictional contact constraints. By handling complementarity constraints through projection, our approach implicitly and automatically reasons about optimal contact modes. We demonstrate the efficiency and versatility of our approach across various scenarios, ranging from dynamic jumping and contact-rich motions with rigid robots to complex environmental interactions with soft robots.**

***Index Terms*—Numerical optimization, Robot control, Contact dynamics, Soft robots**

[1]Inria and Département d'Informatique de l'École Normale Supérieure, PSL Research University in Paris, 75013 Paris, France.

[2]work done while at EDF Lab, 7 boulevard Gaspard Monge, 91120 Palaiseau, France.

[3]University of the Bundeswehr Munich, Department of Aerospace Engineering, Institute of Applied Mathematics and Scientific Computing, 85577 Neubiberg, Germany.

Corresponding author: etienne.menager@inria.fr

I. Introduction

Contact modeling plays a crucial role for simulation and control of robotic systems that create and break contacts with the environment in order to move or manipulate objects. Whether in the context of rigid-body robotics or soft robotics, previous works have emphasized the importance of accurately capturing contact interactions, particularly in scenarios involving friction and sliding [1], [2]. These interactions are crucial not only for realistic simulations but also for enabling robust control in tasks such as manipulation and locomotion.

Yet, achieving effective control in contact-rich environments remains a challenge, due to the inherently nonlinear and non-convex nature of realistic contact models. Both rigid-body models and deformable-body models—such as those based on the finite element method (FEM)—are subject to this complexity, often necessitating advanced numerical methods to ensure stability and convergence.

Interestingly, despite their structural and physical differences, rigid and soft robotic systems share an underlying structure in the mathematical expression of contact constraints. In both domains, contact dynamics can be formulated as a linear complementarity problem (LCP) or a nonlinear complementarity problem (NCP), in which the Delassus matrix couples contact point velocities to contact forces. This observation enables the development of a unified mathematical framework for both domains.

Leveraging this insight, we show in this work that inverse dynamics problems in both rigid- and soft-body robotics can be formulated as quadratic programs with complementarity

constraints (QPCC), which belong to the general class of mathematical programs with complementarity constraints (MPCC) of the form

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{minimize}} \quad & f(\boldsymbol{z}), \\ \text{subject to} \quad & g(\boldsymbol{z}) = \mathbf{0}, \quad h(\boldsymbol{z}) \geq \mathbf{0}, \\ & \mathbf{0} \leq G(\boldsymbol{z}) \perp H(\boldsymbol{z}) \geq \mathbf{0}, \end{aligned} \tag{1}$$

with decision variable $\boldsymbol{z} \in \mathbb{R}^n$. The complementarity condition $\mathbf{0} \leq G(\boldsymbol{z}) \perp H(\boldsymbol{z}) \geq \mathbf{0}$ is a more involved constraint than the standard nonlinear inequality $h(\boldsymbol{z}) \geq \mathbf{0}$. It means that both $G(\boldsymbol{z})$ and $H(\boldsymbol{z})$ should be nonnegative (component-wise), but they cannot be nonzero at the same time, meaning that $G(\boldsymbol{z})^\top H(\boldsymbol{z}) = 0$ every time, which is, by essence, a non-convex constraint.

Such complementarity constraint can be further generalized to cone complementarity constraints of the form $\mathcal{K} \ni G(\boldsymbol{z}) \perp H(\boldsymbol{z}) \in \mathcal{K}^*$, where the zero complementarity should be understood in the sense of the duality relationship between the quantities $G(\boldsymbol{z})$ and $H(\boldsymbol{z})$ that respectively belong to the (more) generic cone $\mathcal{K}$ and its dual $\mathcal{K}^*$. This formulation naturally accommodates frictional and sliding contact phenomena [3] within a unified optimization framework applicable to both rigid [2] and deformable [4] robotic systems. In the context of frictional contacts, $\mathcal{K}$ can represent the so-called Coulomb friction cone.

From a purely mathematical optimization point of view, MPCCs are particularly difficult to solve due to the non-convexity of the feasible set, which fails standard constraint qualifications at every feasible point [5]. A widespread strategy is to relax the complementarity constraint and solve a sequence of nonlinear programs (NLPs) that better approximate the original MPCC. Such homotopy (or continuation) can take the form of a penalty term in the objective [6], [7] or an enlarged feasible set [8], [9]. Even special instances of MPCC such as linear and (convex) quadratic programs with complementarity constraints (LPCC and QPCC) pose numerical challenges. Although mixed-integer linear and quadratic solvers can address these problems and even find global solutions, they do not scale well with the number of integer variables. Forgoing the guarantee of global optimality, the penalty homotopy of [10] follows a sequential convex programming approach and exploits the structure of QPCC, reusing matrix factorizations. More details and references on theory and algorithms for MPCC can be found in the survey [11]. Our methodology for addressing (1) is inspired by the augmented Lagrangian (AL) framework, which bears strong theoretical guarantees [12], [13], and exploits the additional problem structure of QPCCs, leading to improved practical performance.

Contributions. In this paper, we introduce a unified contact-implicit operational-space control formulation that applies to both rigid and soft robots by explicitly handling frictional contact constraints for operational-space control. More precisely, we show that this formulation can be cast as a QPCC problem, where the objective function is a quadratic cost encapsulating the task, while frictional contacts are expressed as complementarity constraints. We focus on one-step, instantaneous inverse dynamics and provide a contact-aware low-level controller that can be combined with higher-level planning. Particularly, we introduce a tailored augmented Lagrangian-based algorithm to efficiently and accurately solve this problem, and we demonstrate that our method supports accurate control in the presence of complex contact interactions. We also discuss the convergence properties and theoretical guarantees of the proposed algorithm as well as the computational performance of the proposed approach.

Paper organization. As the paper tends to unify task-space inverse dynamics with frictional contacts for both rigid and soft robotics, we begin by introducing the problem separately in each domain. Section II-C and Section II-D describe the inverse dynamics formulation in rigid and soft robotics, respectively. These sections reflect the conventions and modeling approaches specific to each community, and can be read independently depending on the reader's background. In Section III, we propose a unified modeling formulation for contact-implicit inverse-dynamics formulated as QPCC instance with harmonized notations and introduce the optimization procedure for solving the underlying QPCC problems. The performance and versatility of the proposed approach are assessed in Section IV, followed by a discussion of its scope and limitations in Section V.

List of Symbols

Throughout this article, we use various symbols to represent physical and optimization quantities which are summarized hereafter. We use bold symbols to represent vectors (e.g., $\boldsymbol{x}$) and capital for matrices (e.g., X).

General symbols

$[\boldsymbol{x}_T \ x_N]$	Tangential and normal components of the vector $\boldsymbol{x}$.
μ	friction coefficient.
$\mathcal{K}_\mu$	Second-order Coulomb friction cone $\|\boldsymbol{x}_T\|_2 \leq \mu x_N$.
$\mathcal{K}_\mu^*$	Dual of the Coulomb friction cone.
$\Gamma_\mu(\boldsymbol{x}) \coloneqq [\mathbf{0} \ \mu\|\boldsymbol{x}_T\|_2]$	De Saxcé correction.
h	Integration time step.
$\boldsymbol{\lambda}_c, \boldsymbol{\lambda}_a$	Contact and actuation/torque impulses.
$\boldsymbol{f}_c, \boldsymbol{f}_a$	Contact and actuation/torque forces.
$\boldsymbol{\sigma}_c$	Contact point velocity.

Rigid-body dynamics

$\boldsymbol{q}$	Robot configuration.
$\boldsymbol{v}$	Robot generalized velocities.
$\mathcal{Q}$	Robot configuration space.
$\boldsymbol{\tau}$	Motor torques.
$M(\boldsymbol{q})$	Joint-space inertia matrix (JSIM).
$\dot{\boldsymbol{v}}$	Robot generalized accelerations.
$J_a(\boldsymbol{q}), J_c(\boldsymbol{q})$	Actuation and contact Jacobians.
$\boldsymbol{b}(\boldsymbol{q}, \boldsymbol{v})$	Nonlinear terms (gravity, Coriolis, centrifugal).
$\boldsymbol{v}_{\text{fr}}$	Contact- and actuation-free generalized velocities.

Soft-body dynamics

A	Impedance matrix defined as $A \coloneqq M - hD - h^2K$, combining the mass matrix M, the damping matrix D, and the stiffness matrix K.
H_i	$(i \in [e, a, c])$ Jacobian matrix of the effectors/actuators/contacts.

$W_{ij} := H_i A^{-1} H_j^T$ Admittance matrix in the effector/actuator/contact space.
$\boldsymbol{\delta}_e^{\text{free}}$ Free position of the effectors.
$\boldsymbol{X}_{e,\text{goal}}$ Target positions for the effectors.
$\boldsymbol{v}_{\text{free}}$ Free velocity of the contact points.
$[\boldsymbol{\lambda}_{a,\min}, \boldsymbol{\lambda}_{a,\max}]$ Actuation bounds.

II. Background and Problem Statement

Contact problems are typically formulated as linear complementarity problems or nonlinear complementarity problems, depending on the nature of the contact model. Such formulations are central to both rigid-body and soft-body simulations and have led to a wide variety of solution methods [2], [14]. However, the inverse dynamics paradigm introduces additional constraints and modeling variations, resulting in more complex optimization problems. In this section, we first review how contact is typically handled in simulation before detailing its classical treatment in rigid and soft robotics inverse dynamics.

A. Frictional contact modeling

Frictional contact is usually modeled through three core principles:

(i) The *Signorini condition* provides a complementarity constraint $0 \leq \lambda_N \perp \sigma_N \geq 0$, ensuring the normal force is repulsive, bodies do not interpenetrate, and no simultaneous separation motion and contact force exist. Signorini condition alone can be handled as an LCP;
(ii) The *Coulomb friction law*, characterized by the so-called friction coefficient μ, which states that the tangential force component is bounded by the normal force value $\|\boldsymbol{\lambda}_T\|_2 \leq \mu\lambda_N$;
(iii) The *maximum dissipation principle* states that the tangential force maximizes the power dissipated by the contact, namely $\boldsymbol{\lambda}_T \in \operatorname{argmax}_{\|\boldsymbol{y}\|_2 \leq \mu\lambda_N} -\boldsymbol{y}^\top \boldsymbol{\sigma}_T$.

These three principles are equivalent to the following NCP formula:

$$\mathcal{K}_\mu \ni \boldsymbol{\lambda} \perp \boldsymbol{\sigma} + \Gamma_\mu(\boldsymbol{\sigma}) \in \mathcal{K}_\mu^*, \quad \boldsymbol{\sigma} = G\boldsymbol{\lambda} + \boldsymbol{g}, \tag{2}$$

where G is the so-called Delassus matrix [15] that gives the system inertia projected on the contacts, linking contact forces to contact point velocities, $\mathcal{K}_\mu$ is the friction cone and $\mathcal{K}_\mu^*$ its dual cone, and $\Gamma_\mu(\cdot)$ the De Saxcé correction [16]. Physically speaking, the De Saxcé correction enforces the Signorini condition.

Without the De Saxcé term, the complementarity condition becomes a *conic* complementarity problem (CCP), which has the physical effect of enabling normal motions in the presence of sliding contact points, which is physically incorrect, prohibited by the Signorini condition. Using a pyramidal approximation of the cones turns this NCP into an LCP; both are special cases of the MPCC stated in (1). A comprehensive review of frictional contact modeling is available in [2], [3], [14].

The optimality conditions that arise from the contact complementarity problem (2) naturally partition the solution space into three mutually exclusive contact regimes. Each regime, illustrated in Fig. 2, represents an "optimal" way

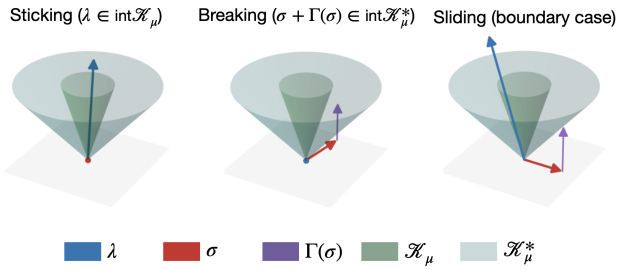


Fig. 2: **Geometric interpretation of the three frictional contact modes.** Each plot shows the friction cone $\mathcal{K}_\mu$ (green), its dual $\mathcal{K}_\mu^*$ (cyan), and the contact plane (grey). (a) **Sticking:** the relative velocity is zero ($\boldsymbol{\sigma} = \mathbf{0}$) and the force $\boldsymbol{\lambda}$ remains strictly inside $\mathcal{K}_\mu$. (b) **Breaking:** the contact force vanishes ($\boldsymbol{\lambda} = \mathbf{0}$) while the corrected relative velocity $\boldsymbol{\sigma} + \Gamma_\mu(\boldsymbol{\sigma})$ lies strictly inside $\mathcal{K}_\mu^*$. (c) **Sliding:** the normal velocity is zero ($\sigma_N = 0$); the tangential velocity $\boldsymbol{\sigma}_T \neq \mathbf{0}$ selects, through the maximum–dissipation principle, a boundary force $\boldsymbol{\lambda}_T = -\mu\lambda_N \frac{\boldsymbol{\sigma}_T}{\|\boldsymbol{\sigma}_T\|_2}$. The De Saxcé correction $\Gamma_\mu(\boldsymbol{\sigma})$ (purple) guarantees complementarity between the cones in all cases.

for the system to satisfy Signorini, Coulomb friction and maximum dissipation:

- **Sticking:** $\boldsymbol{\sigma} = \mathbf{0}$ and $\boldsymbol{\lambda} \in \text{int}\,\mathcal{K}_\mu$. Here, the contact point remains locked, preventing even the slightest microscopic slip. The relative velocity vanishes while the reaction force stays strictly inside the Coulomb cone.
- **Breaking:** $\boldsymbol{\lambda} = \mathbf{0}$ and $\boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma}) \in \text{int}\,\mathcal{K}_\mu^*$, which is equivalent to $\sigma_N > 0$. In this regime, the two bodies lose contact entirely. Physically, there is no normal reaction and therefore no tangential friction can develop.
- **Sliding:** $\sigma_N = 0$ and either $(\boldsymbol{\lambda}, \boldsymbol{\sigma}) = (\mathbf{0}, \mathbf{0})$ *or* $\boldsymbol{\lambda}_T = -\mu\lambda_N \frac{\boldsymbol{\sigma}_T}{\|\boldsymbol{\sigma}_T\|_2}$. In this scenario, the contact still resists penetration, but the tangential friction reaches its limit. The tangential reaction is opposite to the direction of motion.

This work employs the rigid-contact and Coulomb-friction model typical in robotics, in contrast to the more detailed compliant contact models commonly used in mechanical engineering. Rather than refining the contact law itself, we aim to harness its nonsmooth NCP formulation for inverse-dynamics computation. Adding compliance to the contact model as in [17] is straightforward, and does not require any extension of the proposed approach.

B. Simulation of frictional contacts

Contact is a key component of physics-based simulation engines for both rigid and soft robotics. Contact problems are commonly formulated either as LCPs or NCPs. These problems are solved using different numerical methods.

The projected Gauß-Seidel (PGS) algorithm is widely used in contact simulation for both LCP and NCP formulations. PGS solves complementarity conditions via successive projections. This strategy is adopted in several physics engines, such as SOFA [18], [19], Bullet [20], XDE [21], and PhysX

[22], where PGS serves as a general-purpose, computationally efficient method. However, PGS suffers from several known limitations. Notably, it may produce non-physical internal forces, and its convergence is highly sensitive to the conditioning of the problem.

More advanced strategies, such as the Incremental Potential Contact (IPC) model [23], [24], have been developed to enhance robustness and prevent interpenetration, especially in soft-body simulations. While this approach improves stability, it relies on a simplified contact model using smooth potentials and regularized friction, which may not fully capture the exact physical behavior in complex interactions.

Other numerical techniques have been proposed, including nonsmooth Newton methods [25], interior-point methods (IPM) [26], and alternating direction method of multipliers (ADMM) [4], [17]. The Drake framework [27] and MuJoCo exploit a dedicated Newton solver for handling multibody contact, with friction modeled as convex complementarity constraints, whereas Dojo [26] solves the frictional contact problem using interior-point techniques.

These methods form the backbone of forward simulation for contact-rich scenarios. However, when shifting to inverse dynamics, new modeling and algorithmic challenges arise. In the remainder of this section, we review these differences in the context of rigid and soft robotics.

As a preamble to the following two subsections that formalize the problem of frictional inverse dynamics for rigid and soft systems, it is essential to note that the rigid and soft robotics communities do not share a common language, problem formulation, or notation for describing similar control paradigms. To facilitate the reading and dissemination of this work in both communities, we have endeavored to adhere to the conventions of each community as closely as possible.

C. Contact-aware inverse dynamics for poly-articulated rigid robots

In the rigid-body dynamics case, the robot's state is described by the pair $(\boldsymbol{q}, \boldsymbol{v})$ where $\boldsymbol{q} \in \mathcal{Q}$ is the robot configuration (and $\mathcal{Q}$ the robot configuration space[1]) and $\boldsymbol{v} \in \mathfrak{q} \coloneqq \mathcal{T}_e\mathcal{Q} \cong \mathbb{R}^{n_v}$ are[2] the generalized velocities. The robot control inputs are the motor torques, denoted $\boldsymbol{\tau} \in \mathbb{R}^{n_u}$.

Prior work on simulation [17], [28] has investigated the inverse dynamics problem (in joint space) under frictional contact via impulse formulations, with efficient algorithmic derivations to solve the resulting feasibility problem (finding the joint torque $\boldsymbol{\tau}$ and frictional impulses $\boldsymbol{\lambda}_c$ which produce given feasible contact point velocities).

Here, we aim to extend this approach to task-space inverse dynamics, also known as operational-space control, in the presence of frictional contacts represented as complementarity constraints. This accounts for many problems in the robotics literature, such as driving the robot system's end-effector to an arbitrary task-space velocity or performing position-level tracking, while subject to dynamical constraints, including friction and underactuation. While state-of-the-art contact-aware inverse dynamics frameworks for instantaneous control assume predefined contact sequences (e.g., those given by a contact planner) [29]–[31], the approach developed in this work provides a contact-implicit formulation, enabling automatic discovery of the contact mode. Recent work on model-predictive control (MPC) has also focused on this aspect: contact-implicit formulations (coincidentally, task-space inverse dynamics can be seen as MPC with a horizon of just one step). These works handle the problem with either gradient-based or splitting methods.

↪ *Gradient-based methods.* Prior work by Todorov [32] defined a notion of *goal-directed* dynamics, which can be understood as a "forward" version of task-space inverse dynamics, and that the author applies to trajectory optimization (TO). The idea is to compute generalized accelerations and joint torques by minimizing a cost function, with torques given by MuJoCo's smooth, differentiable approximation of inverse dynamics $\boldsymbol{\tau} = \mathrm{ID}(\boldsymbol{q}, \boldsymbol{v}, \dot{\boldsymbol{v}})$ [28]. In a similar vein, [33], [34], [35] also rely on differentiating through smooth approximations of dynamics[3]. Although these methods are fully contact-implicit and require neither pre-planned contact sequences nor prior knowledge of contact modes, they do so by relying on smoothed or relaxed variants of rigid contact with Coulomb friction. Our approach also operates within this rigid-contact/Coulomb-friction setting, but it adheres to the nonsmooth NCP formulation with De Saxcé correction.

↪ *Splitting methods.* Our algorithm (detailed in the next section) is part of splitting methods (albeit for instantaneous control), a class of methods that decomposes complex optimization problems into simpler subproblems. In the MPC literature, a closely related method is consensus complementarity control (C3) [36], which applies to MPC problems with LCP dynamical constraints and linearized friction cones. The splitting in C3 includes a QP step to account for LCP constraints (which resembles the problem we solve in this paper), and the authors propose multiple *approximate* resolution techniques (one of which is akin to relaxing complementarity). In contrast, our method solves the problem with NCP constraints, accounting for the full frictional dynamics without any approximation.

Continuous-time multicontact dynamics. The positions of contact points will be denoted $\boldsymbol{c}(\boldsymbol{q})$ where $\boldsymbol{c}\colon \mathcal{Q} \to \mathbb{R}^{3n_c}$, with n_c the number of contact points.

In the context of systems composed of multiple rigid bodies coupled by joints, Lagrangian dynamics is the standard approach for describing their equations of motion, expressed in their generalized coordinates. The Lagrangian dynamics reads

$$M(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{v}) = J_a(\boldsymbol{q})^\top \boldsymbol{\tau} + J_c(\boldsymbol{q})^\top \boldsymbol{f}_c, \tag{3}$$

where the $\boldsymbol{f}_c$ are instantaneous contact forces, $M(\boldsymbol{q})$ is the joint-space inertia matrix (JSIM), $J_c = \partial \boldsymbol{c}/\partial \boldsymbol{q}$ is the contact Jacobian, and $J_a \in \mathbb{R}^{n_u \times n_v}$ is the actuation Jacobian. Hereafter, the explicit dependence on $\boldsymbol{q}$ and $\boldsymbol{v}$ is omitted for notational simplicity.

[1]For a floating-base system, (e.g., a legged robot), the configuration space is typically of the form $\mathcal{Q} = \mathrm{SE}(3) \times \mathbb{R}^{n_j}$ with n_j the number of joints.

[2]Here e denotes the neutral element of the group $\mathcal{Q}$.

[3]The former uses a linearized friction constraint in contrast to the conic (CCP) formulations of Todorov.

Accounting for unilateral contact, the *Signorini* condition is

$$0 \leq (\boldsymbol{f}_c)_N \perp \boldsymbol{c}_N(\boldsymbol{q}) \geq 0,$$

which leads to an LCP condition in the dynamics.

Impulse-based dynamics. Let $h > 0$ be a discrete time step. Using a semi-implicit Euler discretization, the velocity time-stepping scheme is given by

$$\boldsymbol{v}^+ = \boldsymbol{v} + hM^{-1}(J_a^\top \boldsymbol{\tau} + J_c^\top \boldsymbol{f}_c - \boldsymbol{b}).$$

The robot configuration is then updated using $\boldsymbol{v}^+$. Introduce the "free" (as in contact- and actuation-free) generalized velocity

$$\boldsymbol{v}_{\text{fr}} = \boldsymbol{v} - hM^{-1}\boldsymbol{b},$$

following [17], the discretized system dynamics can be written using an *impulse* formulation [37], replacing joint acceleration with joint velocities, contact forces $\boldsymbol{f}_c$ with contact impulses $\boldsymbol{\lambda}_c = h\boldsymbol{f}_c$ and torques $\boldsymbol{\tau}$ with torque impulses $\boldsymbol{\lambda}_a = h\boldsymbol{\tau}$ (e.g., their sum over the integration time step). Then, these dynamics can be rewritten as the affine map

$$\boldsymbol{v}^+ = \mathcal{F}(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c) \coloneqq \boldsymbol{v}_{\text{fr}} + M^{-1}(J_a^\top \boldsymbol{\lambda}_a + J_c^\top \boldsymbol{\lambda}_c). \quad (4)$$

Under the sole unilateral contact hypothesis, the dynamics are also subject to an LCP condition in the normal direction

$$0 \leq (\boldsymbol{\lambda}_c)_N \perp (\boldsymbol{\sigma}_c)_N \geq 0,$$

where $\boldsymbol{\sigma}_c = J_c\boldsymbol{v}^+ + \boldsymbol{\gamma}_c$ is the contact-point velocity, and $\boldsymbol{\gamma}_c = (0, 0, c_N(\boldsymbol{q})/h)$ comes from the Taylor expansion of $\boldsymbol{c}(\boldsymbol{q}^+)$.

Accounting for frictional effects, the dynamics instead follow an NCP condition

$$\mathcal{K}_\mu \ni \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c + \Gamma_\mu(\boldsymbol{\sigma}_c) \in \mathcal{K}_\mu^*,$$

where $\Gamma_\mu(\cdot)$ is the De Saxcé corrective term, as introduced in Sec. II-A.

Problem statement. Let $p_{\text{ee}}(\boldsymbol{q})$ be the positions or poses (e.g., a Cartesian product of $\mathbb{R}^3$ or $\mathrm{SE}(3)$) of the end-effectors at configuration $\boldsymbol{q}$, and J_{ee} the Jacobian of p_{ee}.

For convenience (to incorporate contact forces), we propose an impulse formulation for task-space inverse dynamics, leading to the following MPCC:

$$\begin{aligned}
\underset{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{v}^+}{\text{minimize}} \quad & \frac{1}{2}\|J_{\text{ee}}\boldsymbol{v}^+ + \boldsymbol{b}_{\text{ee}}\|_2^2, && \text{(5a)}\\
\text{subject to} \quad & \boldsymbol{v}^+ = \mathcal{F}(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c), && \text{(5b)}\\
& \boldsymbol{\sigma}_c = J_c\boldsymbol{v}^+ + \boldsymbol{\gamma}_c, && \text{(5c)}\\
& \mathcal{K}_\mu \ni \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c + \Gamma(\boldsymbol{\sigma}_c) \in \mathcal{K}_\mu^*, && \text{(5d)}\\
& \boldsymbol{\lambda}_a \in \mathbb{B}, && \text{(5e)}
\end{aligned}$$

with motor torque impulses limits $\mathbb{B} = [h\boldsymbol{\tau}_{\min}, h\boldsymbol{\tau}_{\max}]$, and drift term $\boldsymbol{b}_{\text{ee}}$. Notably, in the classic task-space inverse dynamics framework [30], [31], [38], contact modes are assumed to be known in advance and (5d) reduces to the two following simpler constraints: $\boldsymbol{\lambda}_c \in \mathcal{K}_\mu$ and $\boldsymbol{\sigma}_c = \boldsymbol{0}$. In other words, the contact forces should lie in the Coulomb friction cone, while contact points should not move.

In the following paragraph, we show that problem (5) is analogous to the generic problem (9) and falls within the general QPCC framework introduced in this work.

Standard problem form. We can express the problem in terms of the contact impulses $\boldsymbol{\lambda}_c$ and torque impulses $\boldsymbol{\lambda}_a$ only. Following (4), $\boldsymbol{v}^+$ is an affine function of $(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c)$, thus the contact-point velocity is

$$\boldsymbol{\sigma}_c = F\boldsymbol{\lambda}_a + G\boldsymbol{\lambda}_c + \boldsymbol{g},$$

where $G = J_cM^{-1}J_c^\top$ is the Delassus or *operational-space inertia matrix* (OSIM), $F = J_cM^{-1}J_a^\top$ is the Jacobian of the contact-point velocity with respect to control torque impulse input, and

$$\boldsymbol{g} = J_c\boldsymbol{v}_{\text{fr}} + \boldsymbol{\gamma}_c$$

is the contact-point "free" velocity. The task-space velocity is similarly rewritten

$$J_{\text{ee}}\boldsymbol{v}^+ = F_{\text{ee}}\boldsymbol{\lambda}_a + G_{\text{ee}}\boldsymbol{\lambda}_c + \boldsymbol{g}_{\text{ee}},$$

where the matrices have the same expressions as above, with the leftmost J_c matrix replaced by J_{ee}

$$F_{\text{ee}} = J_{\text{ee}}M^{-1}J_a^\top, \quad G_{\text{ee}} = J_{\text{ee}}M^{-1}J_c^\top,$$

and $\boldsymbol{g}_{\text{ee}} = J_{\text{ee}}\boldsymbol{v}_{\text{fr}}$. Additional cost terms solely in $\boldsymbol{\lambda}_a$ or $\boldsymbol{\lambda}_c$ can be accounted for by stacking the matrices onto F_{ee} or G_{ee} respectively (e.g. if one wants an additional term $\|\boldsymbol{\lambda}_a - \boldsymbol{\lambda}_{a,\text{ref}}\|_W^2$, replace F_{ee} by $[F_{\text{ee}}^\top \; W^{1/2}]^\top$, $\boldsymbol{b}_{\text{ee}}$ by $(\boldsymbol{b}_{\text{ee}}, -W^{1/2}\boldsymbol{\lambda}_{a,\text{ref}})$, and pad G_{ee} with rows of zeroes).

The difference between the expressions used in our approach and the ones from [17] is that we make the dependence on the input torque impulse $\boldsymbol{\lambda}_a$ explicit, since $\boldsymbol{\lambda}_a$ is a decision variable of the task-space inverse dynamics problem (whereas it is not a decision variable in the context of direct simulation). In this case, "free" velocities refer to the contact-*and* actuation-free velocities.

The resulting MPCC has a form analogous to (9) and reads

$$\begin{aligned}
\underset{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c}{\text{minimize}} \quad & \frac{1}{2}\|F_{\text{ee}}\boldsymbol{\lambda}_a + G_{\text{ee}}\boldsymbol{\lambda}_c + \boldsymbol{g}_{\text{ee}} + \boldsymbol{b}_{\text{ee}}\|_2^2, && \text{(6a)}\\
\text{subject to} \quad & \mathcal{K}_\mu \ni \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c + \Gamma(\boldsymbol{\sigma}_c) \in \mathcal{K}_\mu^*, && \text{(6b)}\\
& \boldsymbol{\sigma}_c = F\boldsymbol{\lambda}_a + G\boldsymbol{\lambda}_c + \boldsymbol{g}, && \text{(6c)}\\
& \boldsymbol{\lambda}_a \in \mathbb{B}, && \text{(6d)}
\end{aligned}$$

which fits the general form of optimization problems introduced and solved in this work.

D. Contact-aware inverse dynamics for soft-body systems

In soft robotics, while frictional contact models have been extensively studied in direct simulation, relatively few works incorporate accurate contact models directly into inverse modelling. In [39], [40], the authors employ penalty-based contact models. Other simplified models, such as variations of Piecewise Constant Curvature (PCC) [41], enable the control of the dynamics of soft robots interacting with their environment under strong assumptions regarding geometry and contact modeling.

Unlike prior work on devising optimization-based inverse dynamics for soft robots, as proposed, for instance, in [42], which only enforces the Signorini constraint while ignoring frictional effects, we aim to formulate an inverse-dynamics problem that also natively accounts for Coulomb friction and maximum dissipation effects. In the remainder of this section, we begin by revisiting the standard FEM dynamics, then review how inverse dynamics is traditionally treated in the soft-robot FEM literature, and finally extend this framework by incorporating friction cones into the complementarity system.

FEM modelling of soft robots. The behavior of soft robots can be described using the mechanical equations of continuous media, which can be discretized using the finite element method (FEM) [18], [43], [44] or the material point method (MPM) [45], [46]. In this article, we focus on FEM-based approaches where the system state is given by the pair $[(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n_q} \times \mathbb{R}^{n_q}, n_q = 3n_n]$, in which $\boldsymbol{q}$ stacks the Cartesian positions of the n_n nodes composing the mesh, and $\dot{\boldsymbol{q}}$ their velocities. FEM simulation software is used to obtain an approximate solution of these equations. In general, these equations can be written as

$$M\ddot{\boldsymbol{q}}_t = \boldsymbol{f}_{\text{int}}(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t) + \boldsymbol{f}_{\text{ext}} + H_a^\top \boldsymbol{f}_a + H_c^\top \boldsymbol{f}_c,$$

where M is the mass matrix, $\boldsymbol{f}_{\text{int}}$ the internal forces, $\boldsymbol{f}_{\text{ext}}$ the external forces, $\ddot{\boldsymbol{q}}_t$, $\dot{\boldsymbol{q}}_t$, $\boldsymbol{q}_t$ are respectively the acceleration, the velocity and position of the nodes of the FEM mesh, H_a can be seen as the Jacobian matrix of the actuation and depends on the type of the actuator, H_c is the Jacobian matrix of the contacts, $\boldsymbol{f}_a$ and $\boldsymbol{f}_c$ are the Lagrange multipliers which represent the actuation and contact forces.

The dynamics equation is discretized in time using an integration scheme. In soft robotics, an implicit Euler integration scheme is generally used because of its unconditional stability and its capacity to adapt to the time-stepping used in non-regular mechanics to deal with speed jumps on contact. In addition, internal forces are linearized using a first-order approximation. Noting h the time step, $\boldsymbol{q}_k = \boldsymbol{q}_{kh}$ and $\dot{\boldsymbol{q}}_k = \dot{\boldsymbol{q}}_{kh}$ the discretized position and velocity, and $d\dot{\boldsymbol{q}} \coloneqq h\ddot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k$, the discretized dynamics are then written

$$\boldsymbol{b}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) = h\boldsymbol{f}_{\text{int}}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) + h^2 K\dot{\boldsymbol{q}}_k + h\boldsymbol{f}_{\text{ext}}, \tag{7a}$$

$$Ad\dot{\boldsymbol{q}} = \boldsymbol{b}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) + hH_a^T \boldsymbol{f}_a + hH_c^T \boldsymbol{f}_c, \tag{7b}$$

where $A \coloneqq M - hD - h^2K$, K and D are respectively the impedance, stiffness and damping matrices.

To simplify the handling of actuation, contact, and task-space objectives, it is common in the FEM literature to project the full space dynamics (7) onto lower-dimensional constraint subspaces [47]. To each actuation (a), contact (c), or end-effector task (e), we can associate a Jacobian H_j. This projection leads to a set of linear relations between the displacements $\boldsymbol{\delta}_j$ and the impulses $\boldsymbol{\lambda}_i = h\boldsymbol{f}_i$ of the form

$$\boldsymbol{\delta}_j = \sum_{i \in \{a,e,c\}} hW_{ji}\boldsymbol{\lambda}_i + \boldsymbol{\delta}_j^{\text{free}},$$

where $\boldsymbol{\delta}_j^{\text{free}}$ denotes the displacement that would occur in the absence of applied forces, and $W_{ji} \coloneqq H_j A^{-1} H_i^\top$ is a coupling matrix that captures how forces on k influence motion on j. These matrices naturally arise in the FEM formulation and will serve as the fundamental building blocks for the inverse-dynamics problem developed in the next section.

Inverse dynamics with linear complementarity constraints. The inverse dynamics problem in FEM modeling is formulated as a quadratic optimization problem with (linear) complementarity constraints [42]. When controlling the position of a robot's end-effector (i.e, minimizing the distance between an effector and a goal), the problem can be written as

$$\underset{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c}{\text{minimize}} \quad \frac{1}{2}\left\| hW_{ea}\boldsymbol{\lambda}_a + hW_{ec}\boldsymbol{\lambda}_c + \boldsymbol{\delta}_e^{\text{free}} - \boldsymbol{X}_{e,\text{goal}} \right\|_2^2 \tag{8a}$$

$$\text{subject to} \quad 0 \le \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c \ge 0, \tag{8b}$$

$$\boldsymbol{\sigma}_c = W_{ca}\boldsymbol{\lambda}_a + W_{cc}\boldsymbol{\lambda}_c + \boldsymbol{v}_{\text{fr}}, \tag{8c}$$

$$\boldsymbol{\lambda}_a \in \mathbb{B}, \tag{8d}$$

with $\boldsymbol{X}_{e,\text{goal}}$ the goal position, $\boldsymbol{\sigma}_c$ the velocity of the contact points, $\boldsymbol{v}_{\text{fr}}$ the free velocity of the contact points and $\mathbb{B} = [h\boldsymbol{f}_{a,\min}, h\boldsymbol{f}_{a,\max}]$ the actuation bounds. In this problem, (8b) accounts for the Signorini law. This complementarity and positivity constraint forces either $\boldsymbol{\lambda}_{c,i}$ or $\boldsymbol{\sigma}_{c,i}$, the i-th component of the corresponding vectors, to be zero.

In [42], problem (8) is solved using an active-set strategy. More precisely, let the sets

$$I \coloneqq \{i : \lambda_{c,i} = 0\} \quad \text{and} \quad \bar{I} \coloneqq \{i : \lambda_{c,i} \neq 0\}$$

identify the active and inactive constraints, respectively, and let S_I and $S_{\bar{I}}$ be the corresponding selection matrices. The complementarity constraint can then be rewritten as

$$S_I\boldsymbol{\lambda}_c = \boldsymbol{0}, \quad S_{\bar{I}}\boldsymbol{\sigma}_c = \boldsymbol{0}, \quad S_{\bar{I}}\boldsymbol{\lambda}_c \ge \boldsymbol{0}, \quad S_I\boldsymbol{\sigma}_c \ge \boldsymbol{0},$$

thus transforming the LCP problem into a QP problem, as leveraged in former work for unilateral contact simulation [48]. The active set is then updated whenever one of the inequality constraints reaches its bound. However, this heuristic does not guarantee convergence in the non-convex case, as cycling between active sets is possible, and contact modes are discovered through active set modifications that occur externally to the optimization process. While more elaborate strategies can, in principle, be designed to avoid cycling between active sets, their computational cost can become prohibitively high for the large-scale frictional contact problems we target (due to the combinatorial blowup), so we consider them outside of this paper's scope. Coevoet et al. [49] extend the active-set strategy to account for Coulomb friction, but only in the sticking regime (excluding sliding). As in previous work, friction limitations are handled in the outer loop and not considered in the internal QP problems. Consequently, frictional contact modes (sliding, breaking, sticking) remain difficult to discover within the algorithm iterations automatically.

Inverse dynamics with nonlinear complementarity constraints. In what follows, we propose a formulation that unifies linear and nonlinear complementarity conditions within a single optimization-based inverse dynamics framework.

Rather than relying on heuristic mode detection or active-set strategies, contact interactions are directly embedded into the optimization via nonlinear complementarity constraints. In particular, we express Signorini law, Coulomb friction, and the maximum dissipation principle using an NCP formulation, as in (2). This leads to a general formulation of the form

$$\underset{\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_c,\boldsymbol{\sigma}_c}{\text{minimize}} \quad \frac{1}{2}\left\|hW_{ea}\boldsymbol{\lambda}_a + hW_{ec}\boldsymbol{\lambda}_c + \boldsymbol{\delta}_e^{\text{free}} - \boldsymbol{X}_{e,\text{goal}}\right\|_2^2 \tag{9a}$$

$$\text{subject to} \quad \mathcal{K}_\mu \ni \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c + \Gamma(\boldsymbol{\sigma}_c) \in \mathcal{K}_\mu^*, \tag{9b}$$

$$\boldsymbol{\sigma}_c = W_{ca}\boldsymbol{\lambda}_a + W_{cc}\boldsymbol{\lambda}_c + \boldsymbol{v}_{\text{fr}}, \tag{9c}$$

$$\boldsymbol{\lambda}_a \in \mathbb{B} = [\boldsymbol{\lambda}_{a,\min}, \boldsymbol{\lambda}_{a,\max}], \tag{9d}$$

where $\mathcal{K}_\mu$ is the friction cone with friction parameter μ, $\mathcal{K}_\mu^*$ is the dual cone, and $\Gamma(\boldsymbol{\sigma}_c)$ is the De Saxcé correction term. The resulting QPCC (9) has a form analogous to (6) for rigid robotics, and fits the general form of the nonconvex optimization problems discussed in this paper.

III. Contact-Implicit Task-Space Inverse Dynamics

This section presents the core contribution of this work. We first formalize the generic problem for contact-implicit task-space inverse dynamics, which accounts for frictional contact models encapsulated as NCP constraints. We then introduce a numerical scheme based on the augmented Lagrangian method (ALM) [12] to solve the proposed formulation. In particular, we derive the projection operator to account for the LCP/NCP constraints within the ALM formulation, and establish the underlying stationary conditions of the QPCC, which are used as stopping criteria. The final part summarizes the overall algorithm and analyzes its convergence properties, including the required pre-processing stage.

A. Unified problem formulation

In the context of task-space inverse dynamics control, the task objective function can be generically written as

$$f(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c) \coloneqq \frac{1}{2}\left\|A\boldsymbol{\lambda}_a + B\boldsymbol{\lambda}_c + \boldsymbol{b}\right\|_2^2,$$

where $\boldsymbol{\lambda}_a \in \mathbb{R}^{n_a}$, $\boldsymbol{\lambda}_c \in \mathbb{R}^{n_c}$, $A \in \mathbb{R}^{n_e \times n_a}$, $B \in \mathbb{R}^{n_e \times n_c}$, $\boldsymbol{b} \in \mathbb{R}^{n_e}$, where n_a, n_c, and n_e denote respectively the number of actuators, contact and end-effector dimensions. The A and B matrices respectively relate the input actuation $\boldsymbol{\lambda}_a$ and contact $\boldsymbol{\lambda}_c$ to the task residual, given by

$$r(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c) \coloneqq A\boldsymbol{\lambda}_a + B\boldsymbol{\lambda}_c + \boldsymbol{b}.$$

Let F be the objective f augmented with the equality constraint $\boldsymbol{\sigma}_c = C\boldsymbol{\lambda}_a + D\boldsymbol{\lambda}_c + \boldsymbol{g}$, i.e.,

$$F(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c) \coloneqq f(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c) + \mathcal{I}_{\{\boldsymbol{\sigma}_c = C\boldsymbol{\lambda}_a + D\boldsymbol{\lambda}_c + \boldsymbol{g}\}}(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c),$$

where $C \in \mathbb{R}^{n_c \times n_a}$, $D \in \mathbb{R}^{n_c \times n_c}$, $\boldsymbol{g} \in \mathbb{R}^{n_c}$ and $\mathcal{I}_{\{\boldsymbol{\sigma}_c = C\boldsymbol{\lambda}_a + D\boldsymbol{\lambda}_c + \boldsymbol{g}\}}$ denotes the indicator function[4] for the corresponding equality constraint.

[4]For a nonempty set $S \in \mathbb{R}^n$, the associated indicator $\mathcal{I}_S \colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is defined by $\mathcal{I}_S(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in S$, and $\mathcal{I}_S(\boldsymbol{x}) = \infty$ otherwise.

TABLE I: Definition of coefficient matrices in the soft- and rigid-body settings.

Variable/Parameter	Soft-body	Rigid-body
$\boldsymbol{\lambda}_a$	$\boldsymbol{\lambda}_a$	$\boldsymbol{\lambda}_a$
$\boldsymbol{\lambda}_c$	$\boldsymbol{\lambda}_c$	$\boldsymbol{\lambda}_c$
A	$\mathrm{h}W_{ea}$	F_{ee}
B	$\mathrm{h}W_{ec}$	G_{ee}
$\boldsymbol{b}$	$\boldsymbol{\delta}_e^{\text{free}} - \boldsymbol{X}_{e,\text{goal}}$	$\boldsymbol{g}_{\text{ee}} + \boldsymbol{b}_{\text{ee}}$
C	W_{ca}	$F = J_c M^{-1} J_a^\top$
D	W_{cc}	$G = J_c M^{-1} J_c^\top$
$\boldsymbol{g}$	$\boldsymbol{v}_{\text{fr}}$	$\boldsymbol{g} = J_c \boldsymbol{v}_{\text{fr}} + \boldsymbol{\gamma}_c$

The optimization problem for inverse dynamics—(6), (8) or (9)—can then be more generally reformulated as

$$\underset{\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_c,\boldsymbol{\sigma}_c}{\text{minimize}} \quad F(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c) + \mathcal{I}_{\mathbb{S}}(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c), \tag{10}$$

where $\mathbb{S} \coloneqq \mathbb{B} \times \mathbb{S}_c$, with $\mathbb{B} = [\boldsymbol{\lambda}_{a,\min}, \boldsymbol{\lambda}_{a,\max}]$ the actuation limits and $\mathbb{S}_c$ some nonconvex constraint set. For example, when considering the Signorini-only case expressed as an LCP, the corresponding nonconvex constraint set is

$$\mathbb{S}_c \coloneqq \{(\boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c) \,|\, 0 \le \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c \ge 0\},$$

whereas for an NCP it reads

$$\mathbb{S}_c \coloneqq \left\{(\boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c) \,\middle|\, \mathcal{K}_\mu \ni \boldsymbol{\lambda}_c \perp \boldsymbol{\sigma}_c + \Gamma(\boldsymbol{\sigma}_c) \in \mathcal{K}_\mu^*\right\}.$$

Interestingly, the standard problem (10) encapsulates task-space inverse dynamics problems with frictional contacts for both rigid and deformable systems. Table I summarizes the notations for both soft- and rigid-body problems using the ones exposed in Section II.

Finally, the optimization problem (10) can be rewritten, in splitting form, as

$$\underset{\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_c,\boldsymbol{\sigma}_c,\widehat{\boldsymbol{\lambda}}_a,\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c}{\text{minimize}} \quad F(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c) + \mathcal{I}_{\mathbb{S}}(\widehat{\boldsymbol{\lambda}}_a, \widehat{\boldsymbol{\lambda}}_c, \widehat{\boldsymbol{\sigma}}_c) \tag{11}$$

$$\text{subject to} \quad \boldsymbol{\lambda}_a = \widehat{\boldsymbol{\lambda}}_a, \quad \boldsymbol{\lambda}_c = \widehat{\boldsymbol{\lambda}}_c, \quad \boldsymbol{\sigma}_c = \widehat{\boldsymbol{\sigma}}_c.$$

The (equivalent) splitting reformulation (11) offers the possibility of leveraging simple computational oracles (namely well-structured linear systems and inexpensive projections) at the core of our approach, as detailed next.

B. Reminder on ALM and alternating minimization

To solve the generic problem (11), we leverage an augmented Lagrangian method (ALM), a framework known for its modularity and versatility [13]. ALM is particularly suited for problems with separable structure, as it decouples objective minimization from constraint enforcement through iterative updates. ALM can address problems of the form

$$\underset{\boldsymbol{x},\widehat{\boldsymbol{x}}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(\widehat{\boldsymbol{x}}) \quad \text{subject to} \quad \boldsymbol{x} = \widehat{\boldsymbol{x}}, \tag{12}$$

where f and g are (possibly nonsmooth nonconvex) functions. By naming $\boldsymbol{y}$ the dual variable associated with the equality constraint $\boldsymbol{x} = \widehat{\boldsymbol{x}}$, the augmented Lagrangian function of the optimization problem (12) is defined by

$$\mathcal{L}_\varrho(\boldsymbol{x}, \widehat{\boldsymbol{x}}, \boldsymbol{y}) \coloneqq f(\boldsymbol{x}) + g(\widehat{\boldsymbol{x}}) + \boldsymbol{y}^\top(\boldsymbol{x} - \widehat{\boldsymbol{x}}) + \frac{\varrho}{2}\left\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\right\|_2^2,$$

where $\varrho > 0$ is the AL penalty parameter. Let k denote the current iteration index. ALM loops over three main steps until some optimality convergence criteria are met:

(i) $\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_k \approx \operatorname{argmin}_{\boldsymbol{x},\widehat{\boldsymbol{x}}} \mathcal{L}_{\varrho_k}(\boldsymbol{x}, \widehat{\boldsymbol{x}}, \boldsymbol{y}_k)$: minimizing with respect to $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$, up to some tolerance ε_k,
(ii) $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \varrho_k(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k)$: updating the Lagrange multiplier estimate to drive $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}}$ closer,
(iii) selecting ε_{k+1} and ϱ_{k+1} to promote convergence to feasible stationary points.

Some well-established update rules for executing (iii) can be found in [12], [50], with associated convergence analysis. The minimization step (i) can be coarse and up to stationarity (as opposed to exact global/local optimality) without impairing the theoretical guarantees—this often leads to faster computation. Moreover, the Bound-Constrained Lagrangian (BCL) scheme of [12] often requires only few (around ten) ALM iterations in practice [51].

To exploit the specific structure of (11) during step (i), we adopt an alternating minimization (AltMin) scheme (also known as block coordinate descent or Gauss-Seidel). In practice, we seek a minimizer of $\mathcal{L}_{\varrho_k}(\cdot, \cdot, \boldsymbol{y}_k)$ by alternatively optimizing over either $\boldsymbol{x}$ or $\widehat{\boldsymbol{x}}$. More precisely, starting from the initial guess $\widehat{\boldsymbol{x}}_{k,0} \coloneqq \widehat{\boldsymbol{x}}_{k-1}$, the two substeps

(i.i) $\boldsymbol{x}_{k,j} = \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}_{\varrho_k}(\boldsymbol{x}, \widehat{\boldsymbol{x}}_{k,j-1}, \boldsymbol{y}_k)$,
(i.ii) $\widehat{\boldsymbol{x}}_{k,j} = \operatorname{argmin}_{\boldsymbol{y}} \mathcal{L}_{\varrho_k}(\boldsymbol{x}_{k,j}, \widehat{\boldsymbol{x}}, \boldsymbol{y}_k)$,

are repeated, for $j = 1, 2, 3 \ldots$, until $(\boldsymbol{x}_{k,j}, \widehat{\boldsymbol{x}}_{k,j})$ terminates in step (i), satisfying ε_k-stationarity for $\mathcal{L}_{\varrho_k}(\cdot, \cdot, \boldsymbol{y}_k)$.

C. ALM with complementarity constraints

In the particular case of contact-implicit task-space inverse dynamics, we propose to choose f to encode the task objective and linear constraints (such as dynamics or consistency relations), and g to capture both actuator limits and contact complementarity constraints. In what follows, we show how to apply ALM with AltMin to optimization problems with complementarity constraints, and more specifically for the formulation in (11).

Let $\boldsymbol{y} = (\boldsymbol{y}_a, \boldsymbol{y}_c, \boldsymbol{y}_\sigma)$ be the three dual variables respectively associated with the consensus constraint $\boldsymbol{\lambda}_a = \widehat{\boldsymbol{\lambda}}_a$, $\boldsymbol{\lambda}_c = \widehat{\boldsymbol{\lambda}}_c$ and $\boldsymbol{\sigma}_c = \widehat{\boldsymbol{\sigma}}_c$ in (11). Then, the augmented Lagrangian function for (11) reads

$$\begin{aligned}
&\mathcal{L}_\theta(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c, \widehat{\boldsymbol{\lambda}}_a, \widehat{\boldsymbol{\lambda}}_c, \widehat{\boldsymbol{\sigma}}_c, \boldsymbol{y}) \\
&\coloneqq F(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c) + \mathcal{I}_{\mathbb{S}}(\widehat{\boldsymbol{\lambda}}_a, \widehat{\boldsymbol{\lambda}}_c, \widehat{\boldsymbol{\sigma}}_c) + \begin{bmatrix} \boldsymbol{y}_a \\ \boldsymbol{y}_c \\ \boldsymbol{y}_\sigma \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{\lambda}_a - \widehat{\boldsymbol{\lambda}}_a \\ \boldsymbol{\lambda}_c - \widehat{\boldsymbol{\lambda}}_c \\ \boldsymbol{\sigma}_c - \widehat{\boldsymbol{\sigma}}_c \end{bmatrix} \\
&\quad + \frac{\xi}{2}\|\boldsymbol{\lambda}_a - \widehat{\boldsymbol{\lambda}}_a\|_2^2 + \frac{\eta}{2}\|\boldsymbol{\lambda}_c - \widehat{\boldsymbol{\lambda}}_c\|_2^2 + \frac{\rho}{2}\|\boldsymbol{\sigma}_c - \widehat{\boldsymbol{\sigma}}_c\|_2^2, \quad (13)
\end{aligned}$$

where $\theta \coloneqq (\xi, \eta, \rho)$ collects the strictly positive penalty parameters associated with each splitting constraint. Importantly, as depicted in the experimental section, choosing distinct values for these penalty parameters yields better performance than using a single common penalty parameter.

Applying ALM with AltMin, (i.i)-(ii) to (11) results in the following steps:

$$\begin{bmatrix} \boldsymbol{\lambda}_a^{k,j+1} \\ \boldsymbol{\lambda}_c^{k,j+1} \\ \boldsymbol{\sigma}_c^{k,j+1} \end{bmatrix} \leftarrow \operatorname{argmin} \mathcal{L}_{\theta_k}(\cdot, \cdot, \cdot, \widehat{\boldsymbol{\lambda}}_a^{k,j}, \widehat{\boldsymbol{\lambda}}_c^{k,j}, \widehat{\boldsymbol{\sigma}}_c^{k,j}, \boldsymbol{y}^k) \quad (14)$$

$$\begin{bmatrix} \widehat{\boldsymbol{\lambda}}_a^{k,j+1} \\ \widehat{\boldsymbol{\lambda}}_c^{k,j+1} \\ \widehat{\boldsymbol{\sigma}}_c^{k,j+1} \end{bmatrix} \leftarrow \operatorname{argmin} \mathcal{L}_{\theta_k}(\boldsymbol{\lambda}_a^{k,j+1}, \boldsymbol{\lambda}_c^{k,j+1}, \boldsymbol{\sigma}_c^{k,j+1}, \cdot, \cdot, \cdot, \boldsymbol{y}^k) \quad (15)$$

$$\begin{bmatrix} \boldsymbol{y}_a^{k+1} \\ \boldsymbol{y}_c^{k+1} \\ \boldsymbol{y}_\sigma^{k+1} \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{y}_a^{k} \\ \boldsymbol{y}_c^{k} \\ \boldsymbol{y}_\sigma^{k} \end{bmatrix} + \mathbf{diag}(\theta_k) \begin{bmatrix} \boldsymbol{\lambda}_a^{k+1} - \widehat{\boldsymbol{\lambda}}_a^{k+1} \\ \boldsymbol{\lambda}_c^{k+1} - \widehat{\boldsymbol{\lambda}}_c^{k+1} \\ \boldsymbol{\sigma}_c^{k+1} - \widehat{\boldsymbol{\sigma}}_c^{k+1} \end{bmatrix}. \quad (16)$$

In the following sections, we provide the explicit forms of the two minimization steps: the minimization of the smooth term, in Section III-D, and the projection onto the complementarity constraints, in Section III-E.

D. Step (i.i): minimizing the smooth term

More explicitly, the optimization step (14) is equivalent to

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{\lambda}_a^{k,j+1} \\ \boldsymbol{\lambda}_c^{k,j+1} \\ \boldsymbol{\sigma}_c^{k,j+1} \end{bmatrix} \in \operatorname*{argmin}_{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c} & f(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c) + \frac{\xi_k}{2}\left\|\boldsymbol{\lambda}_a - \widehat{\boldsymbol{\lambda}}_a^{k,j} + \frac{\boldsymbol{y}_a^k}{\xi_k}\right\|_2^2 \\
+ \frac{\eta_k}{2}\left\|\boldsymbol{\lambda}_c - \widehat{\boldsymbol{\lambda}}_c^{k,j} + \frac{\boldsymbol{y}_c^k}{\eta_k}\right\|_2^2 &+ \frac{\rho_k}{2}\left\|\boldsymbol{\sigma}_c - \widehat{\boldsymbol{\sigma}}_c^{k,j} + \frac{\boldsymbol{y}_\sigma^k}{\rho_k}\right\|_2^2 \\
\text{subject to} \quad & \boldsymbol{\sigma}_c = C\boldsymbol{\lambda}_a + D\boldsymbol{\lambda}_c + \boldsymbol{g}, \quad (17)
\end{aligned}$$

which is a strictly convex QP in $(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c, \boldsymbol{\sigma}_c)$. In particular, it always admits a unique solution. The optimality conditions for (17) result in the linear system

$$\begin{aligned}
&\begin{bmatrix} A^\top A + \xi_k I & A^\top B & C^\top \\ B^\top A & B^\top B + \eta_k I & D^\top \\ C & D & -\frac{1}{\rho_k} I \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_a^{k,j+1} \\ \boldsymbol{\lambda}_c^{k,j+1} \\ \boldsymbol{y}_{\text{eq}}^\star \end{bmatrix} \\
&\qquad = \begin{bmatrix} -A^\top \boldsymbol{b} + \xi_k \widehat{\boldsymbol{\lambda}}_a^{k,j} - \boldsymbol{y}_a^k \\ -B^\top \boldsymbol{b} + \eta_k \widehat{\boldsymbol{\lambda}}_c^{k,j} - \boldsymbol{y}_c^k \\ -\boldsymbol{g} + \widehat{\boldsymbol{\sigma}}_c^{k,j} - \frac{1}{\rho_k}\boldsymbol{y}_\sigma^k \end{bmatrix}, \quad (18)
\end{aligned}$$

and the optimal $\boldsymbol{\sigma}_c^{k,j+1}$ can be recovered as

$$\boldsymbol{\sigma}_c^{k,j+1} = C\boldsymbol{\lambda}_a^{k,j+1} + D\boldsymbol{\lambda}_c^{k,j+1} + \boldsymbol{g}. \quad (19)$$

Note that the linear system involves a symmetric quasi-definite matrix, which can be factorized efficiently [52, Thm 2].

E. Step (i.ii): projection onto complementarity constraints

The second step (15) separates into two projection steps:

$$\widehat{\boldsymbol{\lambda}}_a^{k,j+1} = \operatorname*{argmin}_{\widehat{\boldsymbol{\lambda}}_a \in \mathbb{B}} \frac{\xi_k}{2}\left\|\boldsymbol{\lambda}_a^{k,j+1} - \widehat{\boldsymbol{\lambda}}_a + \frac{\boldsymbol{y}_a^k}{\xi_k}\right\|_2^2 \quad (20a)$$

$$\begin{aligned}
\begin{bmatrix} \widehat{\boldsymbol{\lambda}}_c^{k,j+1} \\ \widehat{\boldsymbol{\sigma}}_c^{k,j+1} \end{bmatrix} \in \operatorname*{argmin}_{(\widehat{\boldsymbol{\lambda}}_c, \widehat{\boldsymbol{\sigma}}_c) \in \mathbb{S}_c} & \frac{\eta_k}{2}\left\|\boldsymbol{\lambda}_c^{k,j+1} - \widehat{\boldsymbol{\lambda}}_c + \frac{\boldsymbol{y}_c^k}{\eta_k}\right\|_2^2 \quad (20b) \\
&+ \frac{\rho_k}{2}\left\|\boldsymbol{\sigma}_c^{k,j+1} - \widehat{\boldsymbol{\sigma}}_c + \frac{\boldsymbol{y}_\sigma^k}{\rho_k}\right\|_2^2
\end{aligned}$$

We denote the arguments of the projection onto the complementarity set $\mathbb{S}_c$ by

$$\begin{bmatrix} \widetilde{\boldsymbol{\lambda}}_c^{k,j+1} \\ \widetilde{\boldsymbol{\sigma}}_c^{k,j+1} \end{bmatrix} \coloneqq \begin{bmatrix} \boldsymbol{\lambda}_c^{k,j+1} \\ \boldsymbol{\sigma}_c^{k,j+1} \end{bmatrix} + \begin{bmatrix} \frac{\boldsymbol{y}_c^k}{\eta_k} \\ \frac{\boldsymbol{y}_\sigma^k}{\rho_k} \end{bmatrix}. \tag{21}$$

Complementarity constraints are particularly challenging, as they define nonconvex nonsmooth feasible sets with no direct analytical treatment. However, a key strength of our approach lies in handling such constraints through projections. Although in general nontrivial, with formulation (11) the second step of the algorithm only requires projecting onto the constraint set, which can be derived and efficiently computed for this complementarity set. In our case, this step is central: we embed the contact complementarity conditions directly into the algorithm via projection operators, enabling implicit contact constraint handling.

Orthogonal projection onto (non-)convex sets. Given a *convex* set $S \subseteq \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^n$, the orthogonal projection $\boldsymbol{x}^*$ of $\boldsymbol{x}$ onto S is defined as the minimizer of the Euclidean distance from $\boldsymbol{x}$ to the set. This minimizer is *unique* by virtue of S being convex (and thus the distance function being strictly convex). When S is *non*-convex, the minimizer may be nonunique, resulting in a set-valued projection, equal to the set of all distance minimizers.

Orthogonal projection onto LCP constraints. In the LCP setting, the complementarity set $\mathbb{S}_c$ (see Sec. III) admits a simple projection. For a pair $(\lambda, \sigma) \in \mathbb{R}^2$, the projection is:

$$(\lambda^\star, \sigma^\star) \in \begin{cases} \{(\lambda, 0)\} & \text{if } \lambda \geq 0,\ \sigma < \lambda, \\ \{(0, \sigma)\} & \text{if } \sigma \geq 0,\ \lambda < \sigma, \\ \{(\lambda, 0), (0, \sigma)\} & \text{if } \lambda = \sigma > 0, \\ \{(0, 0)\} & \text{otherwise.} \end{cases}$$

This is illustrated in Fig. 3, where the feasible set corresponds to the union of the two positive coordinate axes. When $\lambda = \sigma > 0$, the projection becomes set-valued. In practice, we resolve this ambiguity by selecting the element in the set that preserves the contact mode (active or inactive) from the previous iteration, ensuring consistency across updates.

Orthogonal projection onto frictional NCP constraints. In the NCP setting, accounting for frictional contacts, we generalize the same approach for the LCP case, by projecting onto the nonlinear second-order cones. These projections are still efficient and fully compatible with the proposed ALM loop.

Following (20b), we compute the slanted projection

$$\underset{(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\sigma}}) \in \mathbb{S}_c}{\operatorname{argmin}} \underbrace{\frac{\eta_k}{2} \|\widehat{\boldsymbol{\lambda}} - \widetilde{\boldsymbol{\lambda}}_c^{k,j+1}\|_2^2 + \frac{\rho_k}{2} \|\widehat{\boldsymbol{\sigma}} - \widetilde{\boldsymbol{\sigma}}_c^{k,j+1}\|_2^2}_{=:\mathcal{P}(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\sigma}})}. \tag{22}$$

This is found by selecting the lowest cost $\mathcal{P}$:

- **Sticking:** the minimal cost is given by $\mathcal{P}\left(\operatorname{proj}_{\mathcal{K}_\mu}(\widetilde{\boldsymbol{\lambda}}_c^{k,j+1}), \mathbf{0}\right)$.
- **Breaking:** the minimal cost is given by $\mathcal{P}\left(\mathbf{0}, \operatorname{proj}_{\mathbb{R}^2 \times \mathbb{R}_+}(\widetilde{\boldsymbol{\sigma}}_c^{k,j+1})\right)$

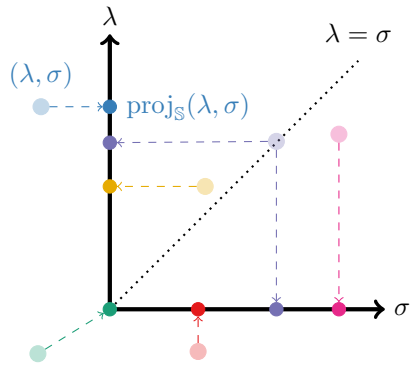


Fig. 3: **Orthogonal projection onto the complementarity set** $\mathbb{S} \coloneqq \{(\lambda, \sigma) \in \mathbb{R}^2 \,|\, 0 \leq \lambda \perp \sigma \geq 0\}$. Different colors indicate the projection outcome depending on the initial (λ, σ). Transparent points represent the current values, thick dots the projected values, and dashed arrows the projection paths.

- **Sliding:** $\sigma_N = 0$ and $\boldsymbol{\lambda}_T = -\mu\lambda_N \frac{\boldsymbol{\sigma}_T}{\|\boldsymbol{\sigma}_T\|_2}$, excluding the case where $\boldsymbol{\sigma}_T = \mathbf{0}$ (covered by the two previous modes). The minimal cost is given by

$$\min_{\boldsymbol{\sigma}_T, \lambda_N} \quad \mathcal{P}\left(\left[-\mu \frac{\boldsymbol{\sigma}_T}{\|\boldsymbol{\sigma}_T\|_2} \lambda_N \ \lambda_N\right], [\boldsymbol{\sigma}_T \ 0]\right).$$

  A local minimum is found using a dedicated alternating solver. At each iteration, we first optimize the intensities of the contact force and velocity with a fixed direction, then update the direction by solving a least-squares problem and projecting onto the unit sphere. Although the subproblem is nonconvex, the method is fast in practice and usually converges to a meaningful local minimum.

This approach, unlike that in [17] for computing the forward frictional dynamics, does not use a fixed-point iteration to handle the De Saxcé correction as a fixed term. Directly computing this projection, instead of altering the outer ALM loop, allows us to recover the provable convergence properties of ALM. See Appendix A for details.

F. Inner termination

The AL minimization step (i) requires reaching a prescribed optimality tolerance ε_k, which means monitoring convergence of the AltMin's iterates and stopping as soon as they are sufficiently close to an exact solution. To quantify stationarity, we define the inner residual

$$r_{\text{in}}^{k,j+1} \coloneqq \left\| \begin{bmatrix} \xi_k(\widehat{\boldsymbol{\lambda}}_a^{k,j+1} - \widehat{\boldsymbol{\lambda}}_a^{k,j}) \\ \eta_k(\widehat{\boldsymbol{\lambda}}_c^{k,j+1} - \widehat{\boldsymbol{\lambda}}_c^{k,j}) \\ \rho_k(\widehat{\boldsymbol{\sigma}}_c^{k,j+1} - \widehat{\boldsymbol{\sigma}}_c^{k,j}) \end{bmatrix} \right\|. \tag{23}$$

Owing to the recursive execution of steps (i.i)–(i.ii) by AltMin, the quantity $r_{\text{in}}^{k,j+1}$ gives a computable stationarity measure of $(\boldsymbol{\lambda}_a^{k,j+1}, \boldsymbol{\lambda}_c^{k,j+1}, \boldsymbol{\sigma}_c^{k,j+1}, \widehat{\boldsymbol{\lambda}}_a^{k,j+1}, \widehat{\boldsymbol{\lambda}}_c^{k,j+1}, \widehat{\boldsymbol{\sigma}}_c^{k,j+1})$ for $\mathcal{L}_\theta(\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \boldsymbol{y}_a^k, \boldsymbol{y}_c^k, \boldsymbol{y}_\sigma^k)$. A detailed derivation is provided in Appendix B. Thus, Alg. 1 stops repeating steps (i.i)–(i.ii) when $r_{\text{in}}^{k,j+1} \leq \varepsilon_k$, namely as soon as ε_k-stationarity is detected.

G. Stationary conditions

In the frame of mathematical programs with complementarity conditions, several optimality criteria might be used to assess convergence toward an optimal solution. In this work, we adopt the Lagrangian formalism to derive the stationarity conditions of our conic complementarity constrained problem (9) in the compact form (10). We follow the general approach of [13, §2.2]; results specialized to the LCP case can be found in [11, §2], [10, §2.1].

The NCP case. Introducing the multipliers $\boldsymbol{y}_a$ for the box constraints, $\boldsymbol{y}_c$ and $\boldsymbol{y}_\sigma$ for the complementarity conditions, and $\boldsymbol{y}_{\text{eq}}$ for the equality constraints, we now present a set first-order optimality conditions for (10), whose derivation is based on (11) and detailed in Appendix C. A primal-dual candidate $(\boldsymbol{\lambda}_a^\star, \boldsymbol{\lambda}_c^\star, \boldsymbol{\sigma}_c^\star, \boldsymbol{y}^\star)$ satisfies the (strong) stationarity conditions if:

- **Primal feasibility**:

$$\boldsymbol{\lambda}_a^\star \in \mathbb{B}, \qquad (\boldsymbol{\lambda}_c^\star, \boldsymbol{\sigma}_c^\star) \in \mathbb{S}_c, \tag{24a}$$

$$\boldsymbol{\sigma}_c^\star = C\boldsymbol{\lambda}_a^\star + D\boldsymbol{\lambda}_c^\star + \boldsymbol{g}. \tag{24b}$$

- **Dual feasibility**:

$$A^\top r(\boldsymbol{\lambda}_a^\star, \boldsymbol{\lambda}_c^\star) + \boldsymbol{y}_a^\star + C^\top \boldsymbol{y}_\sigma^\star = \boldsymbol{0}, \tag{25a}$$

$$B^\top r(\boldsymbol{\lambda}_a^\star, \boldsymbol{\lambda}_c^\star) + \boldsymbol{y}_c^\star + D^\top \boldsymbol{y}_\sigma^\star = \boldsymbol{0}. \tag{25b}$$

- **Complementarity**: These conditions depend on the location of $\boldsymbol{\lambda}_c^\star$ and $\boldsymbol{\sigma}_c^\star$ with respect to the cone $\mathcal{K}_\mu$ and its dual $\mathcal{K}_\mu^*$. Let $\widetilde{\boldsymbol{x}} = (-\boldsymbol{x}_T, x_N)$ be the reflection used in the second-order cone (SOC) formulation. For each frictional contact constraint:
  - if $\boldsymbol{\sigma}_c^\star + \Gamma(\boldsymbol{\sigma}_c^\star) \in \operatorname{int}(\mathcal{K}_\mu^*)$, then $\boldsymbol{\lambda}_c^\star = \boldsymbol{0}$ and $\boldsymbol{y}_\sigma^\star = \boldsymbol{0}$;
  - if $\boldsymbol{\lambda}_c^\star \in \operatorname{int}(\mathcal{K}_\mu)$, then $\boldsymbol{\sigma}_c^\star = \boldsymbol{0}$ and $\boldsymbol{y}_c^\star = \boldsymbol{0}$;
  - if both are on the boundary and $\boldsymbol{\lambda}_{c,T}^\star \propto \boldsymbol{\sigma}_{c,T}^\star$, then $\boldsymbol{y}_c^\star \in \mathbb{R}_+\widetilde{\boldsymbol{\lambda}}_c^\star$, $\boldsymbol{y}_\sigma^\star \in \mathbb{R}_+(\widetilde{\boldsymbol{\sigma}}_c^\star + \Gamma(\widetilde{\boldsymbol{\sigma}}_c^\star))$;
  - if $\boldsymbol{\lambda}_c^\star = \boldsymbol{0}$ and $\boldsymbol{\sigma}_c^\star + \Gamma(\boldsymbol{\sigma}_c^\star) \in \partial\mathcal{K}_\mu^* \setminus \{\boldsymbol{0}\}$, then $\boldsymbol{y}_c^\star \in \mathbb{R}_-^\star\widetilde{\boldsymbol{\sigma}}_c^\star$ and $\boldsymbol{y}_\sigma^\star \in \mathbb{R}_+(\widetilde{\boldsymbol{\sigma}}_c^\star + \Gamma(\widetilde{\boldsymbol{\sigma}}_c^\star))$;
  - if $\boldsymbol{\sigma}_c^\star = \boldsymbol{0}$ and $\boldsymbol{\lambda}_c^\star \in \partial\mathcal{K}_\mu \setminus \{\boldsymbol{0}\}$, then $\boldsymbol{y}_c^\star \in \mathbb{R}_+\widetilde{\boldsymbol{\lambda}}_c^\star$ and $\boldsymbol{y}_\sigma^\star \in \mathbb{R}_-^\star\widetilde{\boldsymbol{\lambda}}_c^\star$;
  - if $\boldsymbol{\lambda}_c^\star = \boldsymbol{\sigma}_c^\star = \boldsymbol{0}$, then $\boldsymbol{y}_c^\star \in \mathcal{K}_\mu^*$ and $\boldsymbol{y}_\sigma^\star \in \mathcal{K}_\mu$.

  For the box constraints on $\boldsymbol{\lambda}_a$, using sign-based slack splitting:

$$[\boldsymbol{y}_a^\star]_-^\top(\boldsymbol{\lambda}_a^\star - \boldsymbol{\lambda}_{a,\min}) = 0, \quad [\boldsymbol{y}_a^\star]_+^\top(\boldsymbol{\lambda}_a^\star - \boldsymbol{\lambda}_{a,\max}) = 0.$$

Primal feasibility, dual feasibility and complementary conditions ensure that all active inequalities (contact and box) are enforced, with their associated multipliers satisfying the standard complementarity and dual feasibility properties.

The LCP case. We consider the case where only the Signorini law is enforced, which leads to an LCP problem. In this specific setting, the friction cone $\mathcal{K}_\mu$ and its dual $\mathcal{K}_\mu^*$ reduce to the non-negative orthant $\mathbb{R}_+^m$. Consequently, the conic constraints simplify to the element-wise complementarity conditions: $0 \le \lambda_{c,i} \perp \sigma_{c,i} \ge 0$ for all i. In this case, the dual variables $\boldsymbol{y}_c$ and $\boldsymbol{y}_\sigma$ are also constrained component-wise.

Algorithm 1: Alternating minimization ALM for contact-implicit inverse dynamics.

1 **Initialize:** $\widehat{\boldsymbol{\lambda}}_c^0$, $\widehat{\boldsymbol{\lambda}}_a^0$, $\widehat{\boldsymbol{\sigma}}_c^0$, $\boldsymbol{y}_a^0$, $\boldsymbol{y}_c^0$, $\boldsymbol{y}_\sigma^0$, tolerance ϵ, maximum iteration limits $K, K_{\text{in}} > 0$
2 Optional pre-processing: Ruiz equilibration
3 **for** $k = 0, 1, \ldots, K$ **do**
4 Choose tolerance ε_k and penalty parameters θ_k
5 $(\widehat{\boldsymbol{\lambda}}_a^{k,0}, \widehat{\boldsymbol{\lambda}}_c^{k,0}, \widehat{\boldsymbol{\sigma}}_c^{k,0}) \leftarrow (\widehat{\boldsymbol{\lambda}}_a^k, \widehat{\boldsymbol{\lambda}}_c^k, \widehat{\boldsymbol{\sigma}}_c^k)$
6 **for** $j = 0, 1, \ldots, K_{\text{in}}$ **do**
7 $(\boldsymbol{\lambda}_a^{k,j+1}, \boldsymbol{\lambda}_c^{k,j+1}, \cdot) \leftarrow$ solve linear system (18)
8 $\boldsymbol{\sigma}_c^{k,j+1} \leftarrow$ (19)
9 $\widetilde{\boldsymbol{\lambda}}_a^{k,j+1}, \widetilde{\boldsymbol{\lambda}}_c^{k,j+1}, \widetilde{\boldsymbol{\sigma}}_c^{k,j+1} \leftarrow$ (21)
10 $\widehat{\boldsymbol{\lambda}}_a^{k,j+1} \leftarrow \operatorname{proj}_{\mathbb{B}}(\widetilde{\boldsymbol{\lambda}}_a^{k,j+1})$
11 $(\widehat{\boldsymbol{\lambda}}_c^{k,j+1}, \widehat{\boldsymbol{\sigma}}_c^{k,j+1}) \leftarrow$ (22)
12 $r_{\text{in}}^{k,j+1} \leftarrow$ (23)
13 **if** $r_{\text{in}}^{k,j+1} \le \varepsilon_k$ **then break**
14 **end**
15 $(\boldsymbol{\lambda}_a^{k+1}, \boldsymbol{\lambda}_c^{k+1}, \boldsymbol{\sigma}_c^{k+1}) \leftarrow (\boldsymbol{\lambda}_a^{k,j+1}, \boldsymbol{\lambda}_c^{k,j+1}, \boldsymbol{\sigma}_c^{k,j+1})$
16 $(\widehat{\boldsymbol{\lambda}}_a^{k+1}, \widehat{\boldsymbol{\lambda}}_c^{k+1}, \widehat{\boldsymbol{\sigma}}_c^{k+1}) \leftarrow (\widehat{\boldsymbol{\lambda}}_a^{k,j+1}, \widehat{\boldsymbol{\lambda}}_c^{k,j+1}, \widehat{\boldsymbol{\sigma}}_c^{k,j+1})$
17 Update dual variables via (16) according to BCL
18 $r_{\text{dual}}^{k+1} \leftarrow r_{\text{in}}^{k,j+1}$ and $r_{\text{prim}}^{k+1} \leftarrow$ (27)
19 **if** $\max\{r_{prim}^{k+1}, r_{dual}^{k+1}\} \le \epsilon$ **then break**
20 **end**
21 **Return:** $(\widehat{\boldsymbol{\lambda}}_a^{k+1}, \widehat{\boldsymbol{\lambda}}_c^{k+1}, \widehat{\boldsymbol{\sigma}}_c^{k+1})$

H. Pseudo-code summarizing the approach

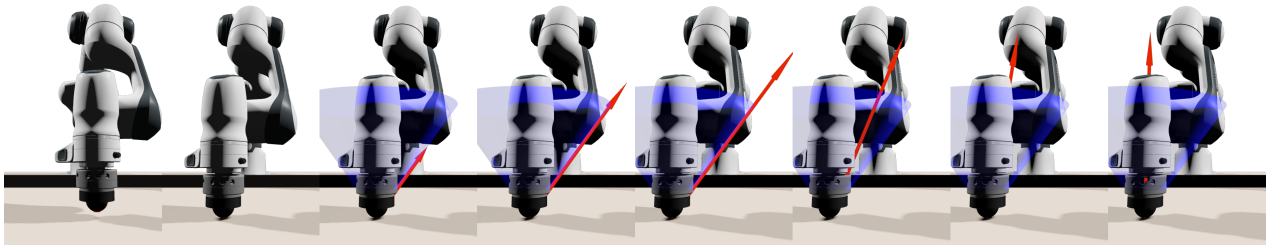
Algorithm 1 provides the pseudo-code associated with our solver. Our ALM-based solver applies equally to both the LCP and NCP formulations presented in Section III, with the appropriate choice of projection operator $\operatorname{proj}_{\mathbb{S}}$. The termination condition in Algorithm 1 is based on the stationarity conditions detailed in Section III-G. This provides a physically meaningful stopping criterion and remains valid for both LCP and NCP problems.

The convergence criterion in Algorithm 1 exploits the relationship between the (inner) stationarity for each subproblem and the (outer) measure of dual feasibility, owing to the update at Line 17. Following [12, Alg. 1] and [13, §3.3], dual feasibility and complementarity conditions can be easily monitored by means of the inner residual r_{in} defined in (23), see r_{dual} at Line 18. Hence, it suffices to compute the primal residual for (11), which is given by

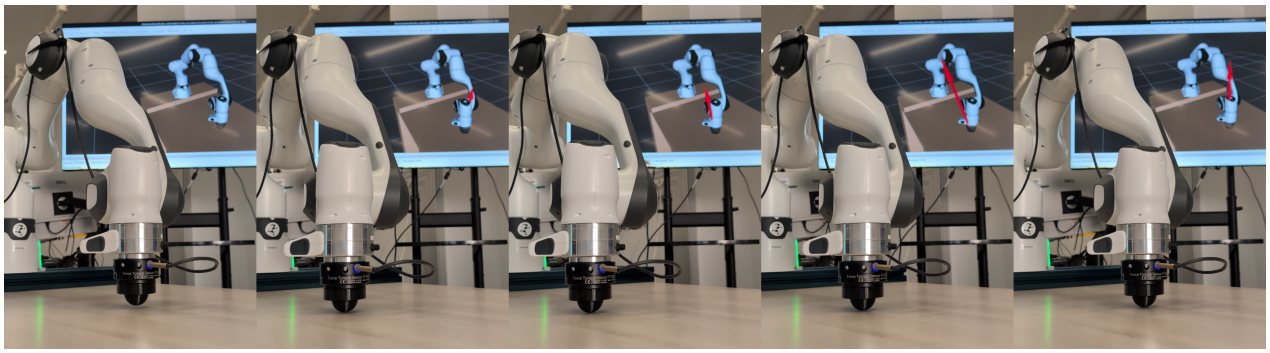
$$r_{\text{prim}}^k := \left\| \begin{bmatrix} \boldsymbol{\lambda}_a^k - \widehat{\boldsymbol{\lambda}}_a^k \\ \boldsymbol{\lambda}_c^k - \widehat{\boldsymbol{\lambda}}_c^k \\ \boldsymbol{\sigma}_c^k - \widehat{\boldsymbol{\sigma}}_c^k \end{bmatrix} \right\|_\infty. \tag{27}$$

The algorithm stops as soon as both these residuals fall below a user-defined tolerance $\epsilon > 0$.

Numerical equilibration. To improve numerical performance, as a pre-processing phase, we apply diagonal Ruiz equilibration [53] to the system solved during the proximal step. This ensures a better conditioning of the linear system and stability across scales. The procedure is further detailed in Appendix D.



(a) **Simulation.** The contact cone is in grey-blue, and the ground reaction force is the red arrow.



(b) **Real hardware.** RViz displays the robot model and ground reaction force (red arrow) in the background.

Fig. 4: **Franka sanding task.** The robot first makes contact with the ground, then starts sliding towards the left, which makes the reaction force lie on the boundary of the friction cone $\partial\mathcal{K}_\mu$. Then the robot comes to rest on the surface, the contact force vector lies strictly within the friction cone and transitions smoothly to its center.

I. Convergence guarantees

The augmented Lagrangian framework has been widely investigated, establishing convergence guarantees under weak assumptions [12], [13]. Among other properties, ALM is known to generate accumulation points that, if feasible, are asymptotically KKT optimal, namely candidates for local optima [13, Thm 3.3]. If feasibility remains unattained (due to the nonconvex constraints), accumulation points carry relevant information nevertheless: they are least-infeasible, in the sense that they minimize the constraint violation [13, Prop. 3.6].

Since our method is an instance of ALM (with a tailored subsolver that exploits the subproblems' structure), the convergence properties just described are directly inherited.

IV. Experimental validations

To assess the versatility and performance of our ALM-based contact-implicit inverse dynamics solver, we conduct several experiments on a diverse set of robotic systems, both rigid and soft. We begin our evaluation with rigid-body scenarios, including 3D and 6D pose tracking, angular velocity control, and manipulation tasks involving frictional contacts. We then extend our evaluation to soft robotics, comparing our method against active-set and generic solvers on LCP and NCP benchmarks, and demonstrate contact-rich control in soft systems, such as sliding and object positioning through frictional interactions.

All the examples have the same underlying formulation, highlighting the generality of our approach across domains. This collection of experiments illustrates the ability of our approach to handle a wide variety of actuation modes, contact conditions, and task objectives within a unified and modular framework. Our solver is written in C++ using the Eigen [54] linear algebra library, and it will be open-sourced after the review process.

A. Experimental validation on poly-articulated rigid robots

In this section, we investigate several experimental setups to demonstrate the contact-implicit inverse dynamics framework in the context of poly-articulated rigid robots, following the formulation detailed in Section II-C. The robot dynamics and tasks are modeled using the Pinocchio [55], [56] rigid-body dynamics library, which provides implementations for all the required quantities (e.g., the inverse of the joint-space inertia matrix, free velocity, frame Jacobians). We leverage the Coal [57] collision detection to compute the contact placement and associated normal of the contact point candidates. Our controller is running in closed-loop with an integration time-step $h = 10^{-3}$ seconds inside the Simple simulator [17] which is itself running with a 100 times smaller integration time-step to ensure accurate simulation of contact interactions.

Task formulation. In our experiments, we use classical task functions from the literature [58], but rewritten at the velocity level to fit our impulse formulation from Section II-C.

Experiment 1 – Franka sanding. The task consists in tracking a 6D task-space reference trajectory in which the robot first lowers its end-effector to reach the table, then slides along the y-direction while maintaining a constant orientation and exerting a prescribed normal force $\lambda_{z,\text{ref}}$ on the surface.

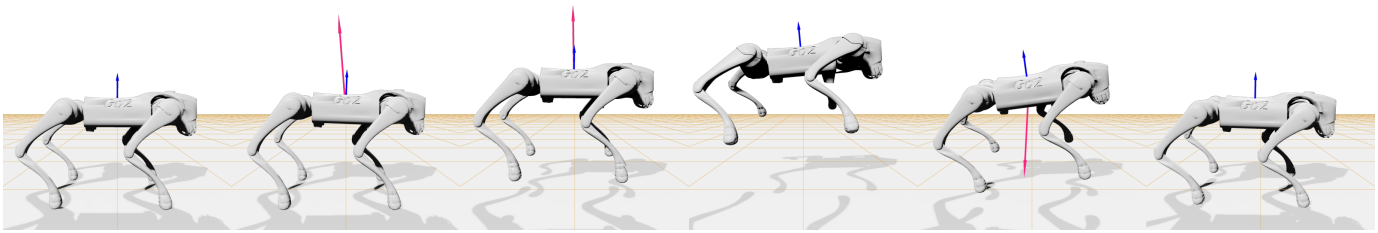


Fig. 5: **Go2 jumping task.** Timelapse of the contact-implicit inverse dynamics controller running in simulation, tracking a desired altitude for the base. The controller automatically determines that a break in contact is required and initiates a jump.

A visualization of the robot motion, including the end-effector friction cone and ground reaction force, is shown in Figure 4a. The quality of the controller in simulation is assessed in Figure 6, plotting both the optimizer progress measures at each control cycle, how well the target force is tracked, and the wall time of the ALM solver for each control cycle.

Note that recent reviews of contact models in robotics, such as [2], emphasize that NCP formulations capture sliding contact more faithfully than LCP formulations. Our Franka arm experiments are consistent with this observation: the NCP-based controller successfully regulates a circular polishing motion that relies on controlled sliding, as illustrated in the accompanying video.

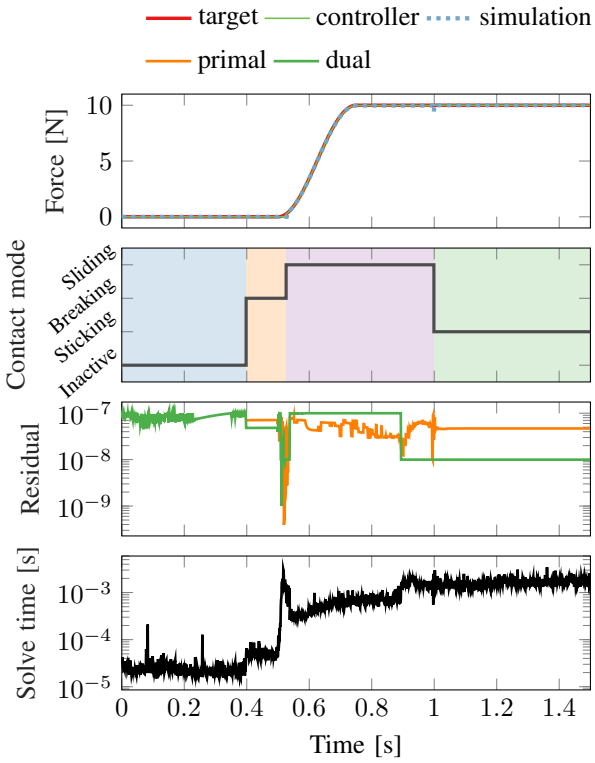


Fig. 6: **Franka sanding task (simulation). Top row.** z-direction ground reaction force (target, controller output, simulation value). **Second row.** Controller's inferred contact mode. **Third row.** Primal and dual feasibility measures. **Bottom row.** ALM solver wall times.

The Franka sanding task was also run on real hardware. The robot is equipped with a force sensor mounted on the wrist and a 3D-printed end-effector. We empirically identified the friction coefficient between the table and the 3D-printed end-effector. We then ran our controller in simulation with a setup (URDF, meshes, and friction coefficient) matching the real one and recorded the computed torques. Finally, we applied the torques in open loop at 1 kHz to the real Franka arm, with a PD controller that accounted for unmodeled joint friction and a black-box low-level controller running at 4 kHz. The behavior on the robot matched closely what we had in simulation in terms of contact modes and end-effector pose, as can be seen on Figure 4b, but the normal force tracking performance was degraded by the unavoidable PD correction. This may point toward the need to equip our controller with force-feedback capabilities in the future [59].

Experiment 2 – Go2 jumping. This task exhibits acrobatic movement on a quadruped robot. The objective is to track a desired height of the quadruped robot's base and determine whether the QPCC-based controller can break contact. Figure 5 displays a slideshow of the controller running in simulation.

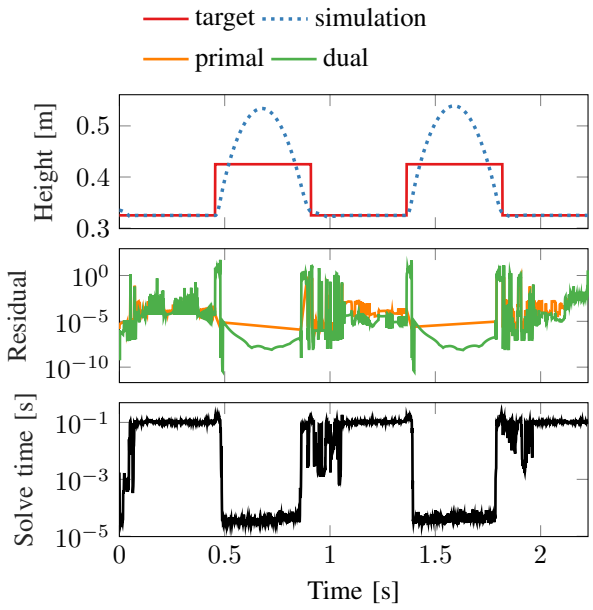


Fig. 7: **Go2 jumping task. Top:** Task target and simulator response. **Middle:** Primal and dual feasibility measures. **Bottom:** ALM solver wall time.

Figure 7 shows as before the solver progress measure at each control cycle, the ALM solver wall times, and how well the controller tracks the desired height.

B. Experimental results on soft robots

In this section, we demonstrate the versatility and efficiency of the proposed ALM-based approach for solving contact-implicit inverse dynamics problems in soft robotics with frictional contacts. The results are structured in four parts: simulation setup and robot models, evaluation in the LCP setting, extension to the NCP formulation, and control with frictional and sliding contacts. In both the LCP and NCP cases, we compare our approach to established baselines: MIQP with the Gurobi solver [60] and active-set method [42] are used as a reference for the LCP formulation, while two Ipopt-based methods [61] are considered in the context of control problems with NCP constraints. Throughout all experiments, we monitor three problem-level quantities—primal, dual, and complementarity—defined from the stationarity conditions in Sec. III-G. This choice aligns with what is typically reported in related work and provides a solver-agnostic view of solution quality. In the following, we will simply refer to these quantities as primal feasibility, dual feasibility, and complementarity, and report their corresponding scalar values as numerical measure.

Five different soft robot systems are considered:

- *Tower robot.* A pneumatically actuated structure with four internal cavities, introduced in [42]. Its behavior is highly nonlinear due to the presence of self-contact.
- *Trunk robot.* A silicone robot actuated by eight cables, as presented in [42], interacting with a rigid cylinder. It exhibits rich contact dynamics.
- *Quadrupod robot.* A deformable parallel mechanism actuated by servo motors, developed as part of this work to manipulate a ball through contact.
- *Sliding cube.* A deformable cube undergoing planar motion with controlled sliding, used to validate frictional control strategies.
- *Finger robot.* A silicone robot, whose design is based on [62], mounted on a base actuated along a single translational axis.

These systems are selected to cover diverse actuation types (pneumatic, cable-driven, motorized), geometries (parallel, elongated), and behaviors (sliding, self-contact). We consider both single-step control (trajectory tracking at each time step) and multi-step goal tracking using an exponentially decaying error signal to reduce overshooting. The units are millimeters for Trunk, Tower, Quadrupod and Finger, and centimeters for the sliding cube. We use absolute tolerance $\epsilon = 1$ millimeter in all cases. The systems are represented in Fig. 8.

Remark. Note that, in soft robotics, strict satisfaction of all constraints is often neither achievable nor necessary, given uncertainties in sensing, actuation, and modeling. Rather than enforcing exact optimality, we target solutions with low residuals—sufficiently accurate to ensure physically consistent behavior, yet fast enough for control. This compromise reflects the realities of soft robot deployment, where control within a few millimeters is typically adequate. Our approach embraces this trade-off while preserving key physical properties, including realistic transitions between sticking, sliding, and breaking in contact-rich interactions.

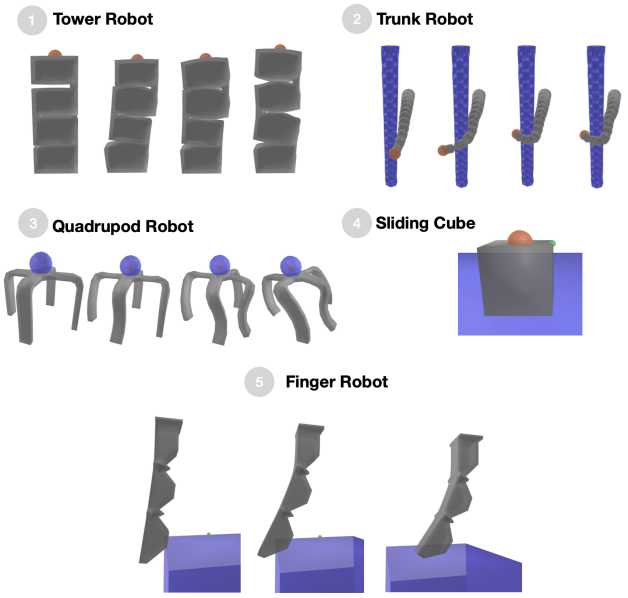


Fig. 8: Illustration of **soft robotic systems** used in this study: (1) Tower Robot, (2) Trunk Robot, (3) Quadrupod Robot, (4) Sliding Cube, and (5) Finger Robot. For the four robotic platforms, we show representative examples of contact configurations.

Simulation framework and soft-robotics models. We build our experiments on a differentiable version [4] of the SOFA framework [18], which allows accurate FEM-based simulation of soft robots, including nonlinear frictional contact as introduced in [17]. The simulation time step is task-dependent but remains within standard ranges for soft-robotics simulations: for the Tower, Trunk and Quadrupod robots we use a time step $h = 10$ ms, while for the Sliding Cube and Finger robot, we use a smaller step $h = 1$ ms.

LCP case and comparison against active set methods. We begin by evaluating our solver in the LCP setting, where the contact model is defined as a linear complementarity problem with Signorini constraints (i.e., non-penetration without friction).

TABLE II: Contact-point statistics (min, first quartile, median, third quartile, and max) across all configurations for each robot. Values correspond to the number of contact points.

Robot	min	Q1	median	Q3	max
Trunk	49	55	57	59	65
Tower	6	67	101	236	305
Quadrupod	66	97	119	133	181

We benchmark our ALM-based solver on three systems: Trunk, Tower, and Quadrupod. For each robot, we construct a dataset of 177, 166, and 179 different contact configurations, respectively. These configurations are sampled along actuation

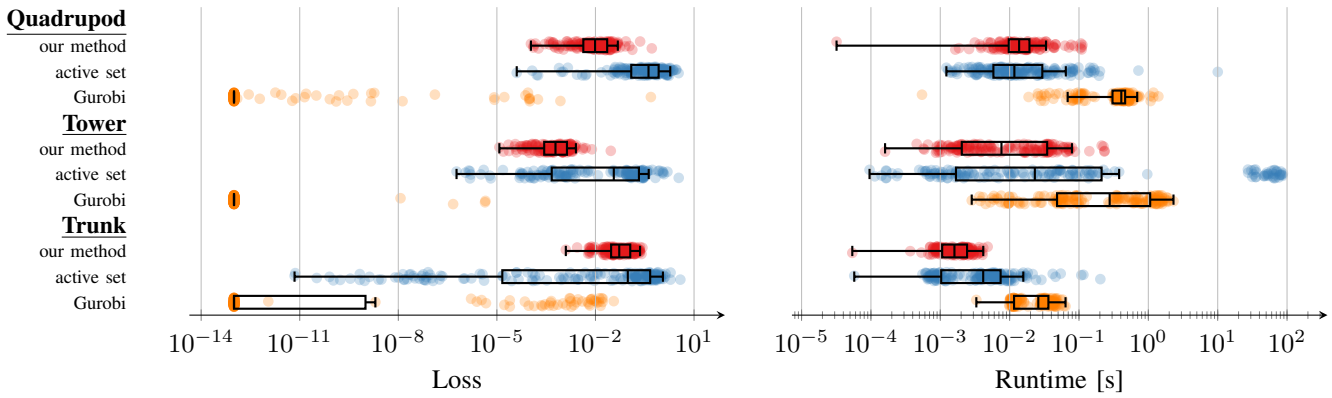


Fig. 9: **Comparison on LCP problems** across robots, considering reference method (active set), a mixed-integer quadratic programming solver (Gurobi), and our method (ALM-based approach). **Left:** loss. **Right:** computation time. While active set and our method satisfy the constraints, Gurobi violates them; see Table III.

trajectories that explore the robot's contact space through extremal actuation sequences. Visual examples of representative contact configurations are shown in Fig. 8. The distribution of the contact dimensions for the three robots is given in Table II. A grid search on the solver parameters is then performed over all the examples to identify the best parameter values across all robots and examples.

Quantitatively, we compare the methods in terms of computation time, loss value f, and solution quality (primal feasibility, dual feasibility, complementarity). The ALM-based approach performs as well or better as the active-set technique in terms of loss (always within millimeter or sub-millimeter range) and computational efficiency, as shown in Fig. 9 and Table III.

TABLE III: Stationarity measures for each robot in the LCP setting. Comparison between the active set method (reference), Gurobi solver and the proposed ALM-based solver. Mean values over all contact configurations problems. Tolerance $\epsilon = 1$ [mm].

Trunk Robot	Our method	Reference	Gurobi
Primal feasibility	$7.37 \cdot 10^{-1}$	$1.52 \cdot 10^{-5}$	$4.05 \cdot 10^{2}$
Dual feasibility	$2.59 \cdot 10^{-1}$	$3.32 \cdot 10^{-1}$	$1.96 \cdot 10^{4}$
Complementarity	$5.54 \cdot 10^{-2}$	$8.44 \cdot 10^{-2}$	/
Tower Robot	Our method	Reference	Gurobi
Primal feasibility	$9.47 \cdot 10^{-1}$	$2.64 \cdot 10^{-5}$	$2.14 \cdot 10^{3}$
Dual feasibility	$1.33 \cdot 10^{-1}$	1.44	$1.23 \cdot 10^{5}$
Complementarity	$5.99 \cdot 10^{-3}$	$1.89 \cdot 10^{-3}$	/
Quadrupod Robot	Our method	Reference	Gurobi
Primal feasibility	$8.17 \cdot 10^{-1}$	$1.26 \cdot 10^{-4}$	$4.32 \cdot 10^{5}$
Dual feasibility	$3.23 \cdot 10^{-1}$	1.80	$3.82 \cdot 10^{2}$
Complementarity	$8.62 \cdot 10^{-2}$	3.26	/

A generic QP solver such as Gurobi can achieve very low objective losses on the benchmark, but often fails to fully satisfy the contact constraints: non-negativity is enforced, yet the complementarity conditions between $\boldsymbol{\lambda}_c$ and $\boldsymbol{\sigma}_c$ are not, which results in large primal feasibility value. The active-set method yields lower primal feasibility, since it explicitly satisfies the primal constraints by construction. However, our method exhibits better dual feasibility and complementarity, leading to more balanced solutions.

This level of error is acceptable for our target applications, where position or force accuracy within a few millimeters is sufficient for effective control. In this context, tighter dual or complementarity tolerances further would provide limited practical benefit and is not strictly required. Nonetheless, in applications that demand higher accuracy (on primal and complementarity), the solver tolerances can be tightened, at the cost of more iterations and longer solve times. We illustrate this trade-off on the Trunk robot in Table IV, using the same dataset.

TABLE IV: Effect of tightening the solver tolerance ϵ on stationarity measures and runtime for the Trunk robot (LCP setting). Mean values over 177 problems.

Trunk Robot	$\epsilon = 1$ [mm]	$\epsilon = 10^{-1}$ [mm]	$\epsilon = 10^{-2}$ [mm]
Primal feasibility	$7.37 \cdot 10^{-1}$	$7.80 \cdot 10^{-2}$	$8.91 \cdot 10^{-3}$
Dual feasibility	$2.59 \cdot 10^{-1}$	$8.31 \cdot 10^{-2}$	$3.42 \cdot 10^{-2}$
Complementarity	$5.54 \cdot 10^{-2}$	$3.93 \cdot 10^{-2}$	$7.35 \cdot 10^{-3}$
Runtime [s]	$1.82 \cdot 10^{-3}$	$5.85 \cdot 10^{-3}$	$9.61 \cdot 10^{-2}$

NCP case and comparison against IP methods. We now extend our evaluation to the full nonlinear frictional contact formulation using the NCP model. The same trajectories as in the LCP case are reused, with contact dimensions now tripled due to 3D contact force components. As for the LCP case, a grid search is performed to find the best set of parameters across all robots and contact configurations. In addition, we compare our approach to two Ipopt-based baselines. In both cases, the conic constraints are represented as nonlinear inequalities, and the complementarity conditions are enforced via equality constraints of the form $\boldsymbol{\lambda}_{c,i}^{\top}(\boldsymbol{\sigma}_{c,i} + \Gamma(\boldsymbol{\sigma}_{c,i})) = 0$ for each contact point i. Analytical gradients are provided, while we use BFGS to approximate the Lagrangian Hessian.

Despite the increased complexity, our method maintains sub-millimeter precision, with stationarity measures comparable to the LCP case. Computation times increase slightly due to the size of the systems, with average runtimes around

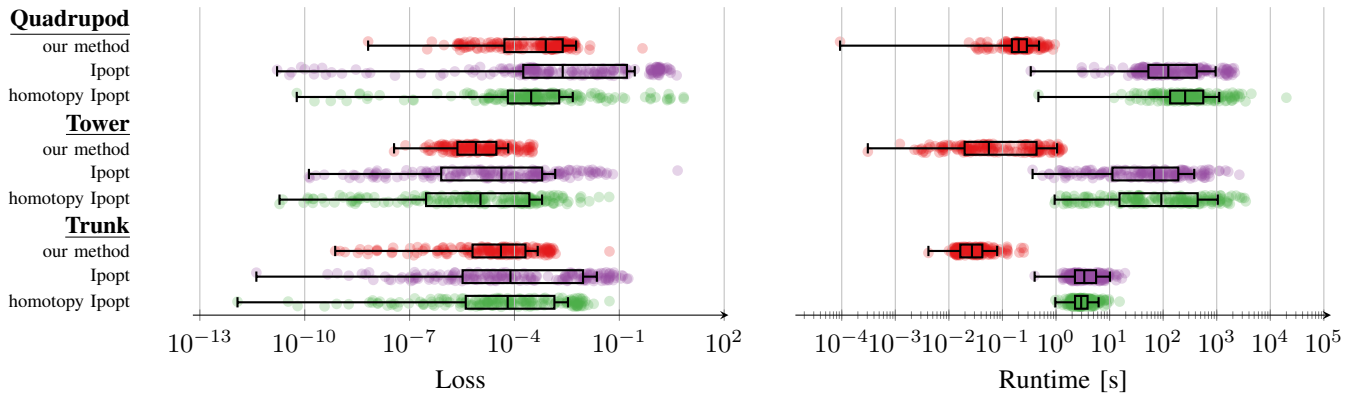


Fig. 10: **Comparison on NCP problems** across robots, considering a nonlinear programming method (Ipopt), a regularized scheme with homotopy, and our method (ALM-based approach). **Left:** loss. **Right:** computation time.

0.1-0.2 seconds per solve. The quality of the solution remains stable, with minimal loss degradation, as shown in Fig. 10. The average values of all stationarity measures remain strictly below one across the benchmark. While Ipopt can achieve low objective losses and fast local convergence (it is a second-order method), we observe that it occasionally returns solutions with poor stationarity measures. Moreover, the Python overhead in our implementation is negligible (less than 5% of total time), hence most of the runtime is spent inside the solver. Our method offers more consistent stationarity measures and faster, more predictable solve times, making it better suited to soft robotics settings where reliability is crucial.

Contact detection and filtering play a critical role here: to limit the size of the contact problem, we have to filter candidate contact points tetrahedron-wise, keeping only the most relevant ones for each interaction. Further improvements in runtime and conditioning could be obtained by leveraging more sophisticated contact culling techniques.

Contact-rich control tasks. We now focus on contact-rich control tasks, where at each time step the inverse-dynamics problem is solved to compute actuation inputs that track desired goals under sticking and sliding frictional contacts.

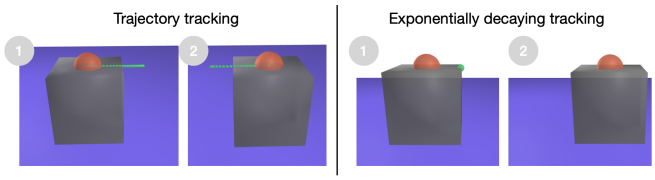


Fig. 11: Illustration of the **two tracking scenarios** considered in the cube example. **Left:** trajectory tracking. **Right:** target reaching with exponential decay. In both cases, (1) initial configuration and (2) final configuration. The end-effector is shown in orange, and the goal(s) are shown in green.

The first task involves pushing a deformable cube across a planar surface. Two control schemes are tested: (i) trajectory tracking, where each goal is reachable in one time step, corresponding to (11) with the matrices and vectors as outlined in Table I; and (ii) exponentially decaying tracking, where a distant goal is approached progressively over time. These two cases are represented in Fig. 11. Results are shown in Fig. 12.

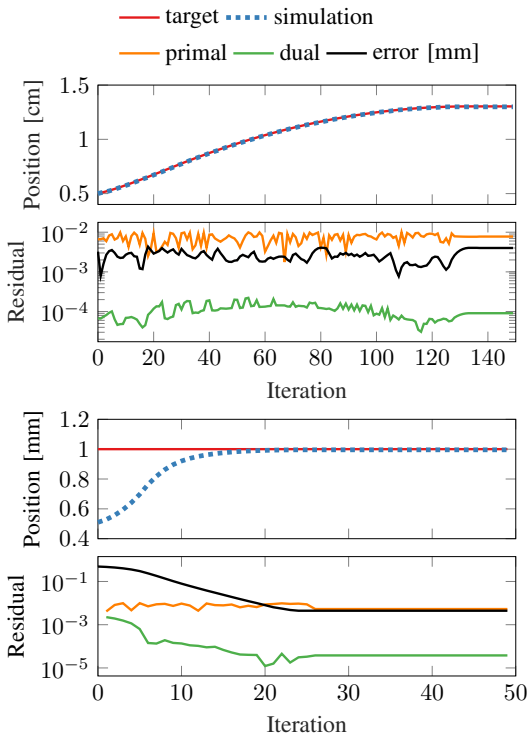


Fig. 12: Results obtained for the **control of a sliding cube on a surface**, in the two scenarios considered in this study: position of the end-effector along the y-axis (blue), target position (red), positioning error relative to the target (black), primal feasibility (orange), and dual feasibility (green). **Top:** results for trajectory tracking. **Bottom:** results for exponential error decay control.

For the exponential decay, let suppose that we want to minimize the norm of the error $\boldsymbol{e}(t) = W_{ea}\boldsymbol{\lambda}_a + W_{ec}\boldsymbol{\lambda}_c + \boldsymbol{\delta}_e^{\text{free}} - \boldsymbol{X}_{e,\text{goal}}$ with an exponentially decaying tracking. The

cost function can be written:

$$f(\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c) = \frac{1}{2}\|\dot{\boldsymbol{e}}(t) + \alpha \boldsymbol{e}(t)\|_2^2 \approx \frac{1}{2}\|(1+\alpha h)\boldsymbol{e}(t) - \boldsymbol{e}(t-h)\|_2^2,$$

with an explicit scheme to approximate a derivative, α is a parameter to control the decaying velocity (and $h > 0$ is once again an integration time step). This new loss function can be integrated in (11) by considering:

$$A = (1+\alpha h)W_{ea}, \qquad B = (1+\alpha h)W_{ec},$$
$$b = (1+\alpha h)(\boldsymbol{\delta}_e^{\text{free}} - \boldsymbol{X}_{e,\text{goal}}) - \boldsymbol{e}(t-h).$$

In both cases, at each time step, we solve the inverse problem to compute actuation forces, which are then applied in a physics-based simulation using the ADMM contact solver from [17]. This means that the computed control is validated in simulation, where contact dynamics come from an external solver.

Results show sub-millimeter tracking error for both control schemes, with stationarity measures consistently below the unit. The exponentially decaying approach avoids overshooting and results in trajectories that closely follow an exponential decay in error. These results demonstrate that the solver produces physically meaningful and stable control strategies even in the presence of sliding contacts.

Similar scenarios are considered for the Quadrupod robot. In these control tasks, the Quadrupod robot must reposition a ball through contact by coordinating its legs. As illustrated in Fig. 13 and Fig. 14, the same conclusions can be drawn as in the sliding cube example. The solver successfully computes actuation inputs that account for contact, friction, and compliance, yielding stable and precise trajectories.

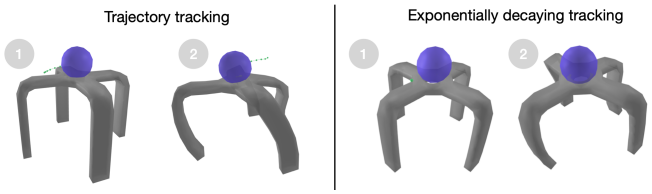


Fig. 13: Illustration of the **two tracking scenarios considered in the Quadrupod example**: initial configuration (1), final configuration (2), end-effector (orange), and target (green). **Left:** trajectory tracking. **Right:** target reaching with exponential decay.

The last experiment is based on the Finger robot. The goal is to control the fingertip so that it slides along a surface, using an exponentially decaying tracking scheme. As illustrated in Fig. 15, three phases can be identified: a free-space phase with no contact, a phase in which contacts block the translational motion of the fingertip, and a final phase where the finger slides along the surface. During the intermediate phase, the contact points remain in a sticking mode and act as a pivot, reorienting the fingertip to facilitate sliding. These contact configurations are shown from left to right in Fig. 8(5). Once again, the solver is able to control the robot while accounting for the different contact modes.

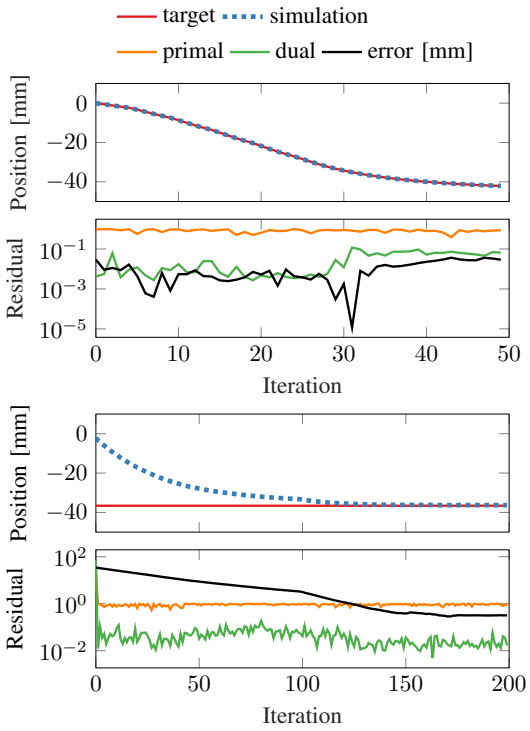


Fig. 14: Results obtained for the **control of the Quadrupod robot**, in the two scenarios considered in this study: position of the end-effector along the y-axis (blue), target position (red), positioning error relative to the target (black), primal feasibility (orange), and dual feasibility (green). **Top:** results for trajectory tracking. **Bottom:** results for exponential error decay control.

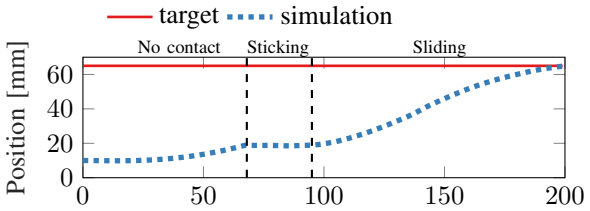


Fig. 15: Results obtained for the **control of the Finger robot**: position of the end-effector along the y-axis (blue), target position (red).

V. Conclusion

In this work, we presented a unified task-space contact-implicit inverse dynamics framework applicable to both rigid and soft robots. The approach extends standard instantaneous inverse dynamics with implicit reasoning about frictional contacts by formulating the problem as a QPCC. An ALM-based modular solver with dedicated projection operators enables efficient handling of frictional complementarity and box constraints, facilitating integration into existing simulation and control pipelines.

We demonstrated the method on a variety of contact-rich scenarios, including rigid and deformable systems such as a jumping quadruped, several soft robots (Tower, Trunk,

Quadrupod, and Finger), and a Franka arm performing simultaneous pose and force tracking. Across these tasks, the solver produced consistent, physically meaningful contact modes and successfully achieved tracking objectives, establishing frictional contact-implicit inverse dynamics as a viable low-level controller that can interface with higher-level planning, including reinforcement learning-based approaches.

Future work will focus on extending the formulation and approach to trajectory optimization or MPC with contact complementarity constraints, and on improving real-time performance by exploiting the problem sparsity structure (mesh topology for soft robots, kinematic tree for rigid robots). Further steps include deploying the approach on hardware, addressing model uncertainties in friction and contact geometry, and investigating state estimation under frictional contact and force feedback for robust closed-loop control.

REFERENCES

[1] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, and P. Volino, "Collision detection for deformable objects," *Computer Graphics Forum*, vol. 24, no. 1, pp. 61–81, 2005.

[2] Q. Le Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: A comparative analysis," *IEEE Transactions on Robotics*, vol. 40, pp. 3716–3733, 2024.

[3] V. Acary, M. Brémond, and O. Huber, "On solving contact problems with Coulomb friction: formulations and numerical comparisons," Research Report RR-9118, INRIA, 2017.

[4] E. Ménager, L. Montaut, Q. Le Lidec, and J. Carpentier, "Differentiable simulation of soft robots with frictional contacts," in *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*, pp. 1–8, IEEE, 2025.

[5] J. J. Ye, D. L. Zhu, and Q. J. Zhu, "Exact penalization and necessary optimality conditions for generalized bilevel programming problems," *SIAM Journal on Optimization*, vol. 7, no. 2, pp. 481–507, 1997.

[6] M. Anitescu, "On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints," *SIAM Journal on Optimization*, vol. 15, no. 4, pp. 1203–1236, 2005.

[7] K. Lin and T. Ohtsuka, "A gap penalty method for optimal control of linear complementarity systems," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 880–887, 2024.

[8] C. Kanzow and A. Schwartz, "A new regularization method for mathematical programs with complementarity constraints with strong convergence properties," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 770–798, 2013.

[9] K. Lin and T. Ohtsuka, "A non-interior-point continuation method for the optimal control problem with equilibrium constraints," *Automatica*, vol. 171, p. 111940, 2025.

[10] J. Hall, A. Nurkanović, F. Messerer, and M. Diehl, "LCQPow: a solver for linear complementarity quadratic programs," *Mathematical Programming Computation*, vol. 17, no. 1, pp. 81–109, 2025.

[11] Y. Kim, S. Leyffer, and T. Munson, "MPEC methods for bilevel optimization problems," in *Bilevel Optimization: Advances and Next Challenges* (S. Dempe and A. Zemkoho, eds.), pp. 335–360, Springer, 2020.

[12] A. R. Conn, N. I. M. Gould, and P. L. Toint, "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Numerical Analysis*, vol. 28, no. 2, pp. 545–572, 1991.

[13] A. De Marchi, X. Jia, C. Kanzow, and P. Mehlitz, "Constrained composite optimization and augmented Lagrangian methods," *Mathematical Programming*, vol. 201, no. 1, pp. 863–896, 2023.

[14] P. C. Horak and J. C. Trinkle, "On the similarities and differences among contact models in robot simulation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 493–499, 2019.

[15] É. Delassus, "Mémoire sur la théorie des liaisons finies unilatérales," *Ann. Sci. Ec. Norm. Super. (4)*, vol. 34, pp. 95–179, 1917.

[16] G. De Saxcé and Z.-Q. Feng, "The bipotential method: A constructive approach to design the complete contact law with friction and improved numerical algorithms," *Mathematical and Computer Modelling*, vol. 28, no. 4, pp. 225–245, 1998. Recent Advances in Contact Mechanics.

[17] J. Carpentier, L. Montaut, and Q. L. Lidec, "From compliant to rigid contact simulation: a unified and efficient approach," *Robotics: Science and Systems*, 2024.

[18] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery* (Y. Payan, ed.), vol. 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pp. 283–321, Springer, 2012.

[19] F. Jourdan, P. Alart, and M. Jean, "A Gauss-Seidel like algorithm to solve frictional contact problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 155, no. 1, pp. 31–47, 1998.

[20] E. Coumans and Y. Bai, "PyBullet, a python module for physics simulation for games, robotics and machine learning." http://pybullet.org, 2016.

[21] CEA LIST, "XDE physics simulator." https://www-list.cea.fr/, 2020. Simulation engine for multibody dynamics with contact and friction, developed by CEA.

[22] M. Macklin, K. Storey, M. Lu, P. Terdiman, N. Chentanez, S. Jeschke, and M. Müller, "Small steps in physics simulation," in *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 1–7, 2019.

[23] I. Huang, Y. Narang, C. Eppner, B. Sundaralingam, M. Macklin, T. Hermans, and D. Fox, "DefGraspSim: Simulation-based grasping of 3D deformable objects," *arXiv preprint 2107.05778*, 2021.

[24] C. Yu, W. Du, Z. Zong, A. Castro, C. Jiang, and X. Han, "A convex formulation of material points and rigid bodies with GPU-accelerated async-coupling for interactive simulation," *arXiv preprint 2503.05046*, 2025.

[25] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makoviychuk, "Non-smooth Newton methods for deformable multi-body dynamics," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–20, 2019.

[26] T. Howell, S. Le Cleac'h, Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable physics engine for robotics," *arXiv preprint 2203.00806v3*, 2022.

[27] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics." https://drake.mit.edu, 2019.

[28] E. Todorov, "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6054–6061, IEEE, 2014.

[29] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 1–9, IEEE, 2018.

[30] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Current Robotics Reports*, vol. 2, no. 3, pp. 231–238, 2021.

[31] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2023.

[32] E. Todorov, "Goal directed dynamics," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2994–3000, 2018.

[33] S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Transactions on Robotics*, vol. 40, pp. 1617–1629, 2024.

[34] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *The International Journal of Robotics Research*, p. 02783649241273645, 2024.

[35] V. Kurtz, A. Castro, A. O. Onol, and H. Lin, "Inverse dynamics trajectory optimization for contact-implicit model predictive control," *The International Journal of Robotics Research*, p. 02783649251344635, 2025.

[36] A. Aydinoglu, A. Wei, W.-C. Huang, and M. Posa, "Consensus complementarity control for multicontact mpc," *IEEE Transactions on Robotics*, vol. 40, pp. 3879–3896, 2024.

[37] B. Mirtich and J. Canny, "Impulse-based simulation of rigid bodies," in *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, I3D '95, (New York, NY, USA), pp. 181–ff., Association for Computing Machinery, 1995.

[38] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *2011 IEEE international conference on robotics and automation*, pp. 1085–1090, IEEE, 2011.

[39] J. Kim and N. S. Pollard, "Direct control of simulated non-human characters," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 56–65, 2011.
[40] J. Kim and N. S. Pollard, "Fast simulation of skeleton-driven deformable body characters," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 5, pp. 1–19, 2011.
[41] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 46–53, 2018.
[42] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1413–1419, 2017.
[43] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE international conference on robotics and automation*, pp. 3982–3987, IEEE, 2013.
[44] B. Caasenbrood, A. Pogromsky, and H. Nijmeijer, "Sorotoki: A MATLAB toolkit for design, modeling, and control of soft robots," *IEEE Access*, vol. 12, pp. 17604–17638, 2024.
[45] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, "DiffTaichi: Differentiable programming for physical simulation," in *International Conference on Learning Representations*, 2020.
[46] Y. Hu, J. Liu, A. E. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "Chainqueen: A real-time differentiable physical simulator for soft robotics," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6265–6271, 2018.
[47] T. Navez, E. Ménager, P. Chaillou, O. Goury, A. Kruszewski, and C. Duriez, "Modeling, embedded control, and design of soft robots using a learned condensed FEM model," *IEEE Transactions on Robotics*, vol. 41, pp. 2441–2459, 2025.
[48] S. Redon, A. Kheddar, and S. Coquillart, "Gauss' least constraints principle and rigid body simulations," in *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, vol. 1, pp. 517–522, IEEE, 2002.
[49] E. Coevoet, A. Escande, and C. Duriez, "Soft robots locomotion and manipulation control using FEM simulation and quadratic programming," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 739–745, 2019.
[50] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014.
[51] D. Ma, K. L. Judd, D. Orban, and M. A. Saunders, "Stabilized optimization via an NCL algorithm," in *Numerical Analysis and Optimization* (M. Al-Baali, L. Grandinetti, and A. Purnama, eds.), (Cham), pp. 173–191, Springer International Publishing, 2018.
[52] R. J. Vanderbei, "Symmetric quasidefinite matrices," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 100–113, 1995.
[53] D. Ruiz, "A scaling algorithm to equilibrate both rows and columns norms in matrices," Tech. Rep. RAL-TR-2001-034, Rutherford Appleton Laboratory, Oxon, England, 2001.
[54] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.
[55] J. Carpentier, F. Valenza, N. Mansard, *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems." https://stack-of-tasks.github.io/pinocchio, 2015–2021.
[56] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.
[57] J. Pan, S. Chitta, D. Manocha, J. Mirabel, J. Carpentier, and L. Montaut, "Coal - an extension of the Flexible Collision Library." https://github.com/coal-library/coal, 2025.
[58] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, no. 3, pp. 429–455, 2016.
[59] A. Jordana, S. Kleff, J. Carpentier, N. Mansard, and L. Righetti, "Force Feedback Model-Predictive Control via Online Estimation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11503–11509, May 2024.
[60] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2026.
[61] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
[62] S. E. Navarro, S. Nagels, H. Alagi, L.-M. Faller, O. Goury, T. Morales-Bieze, H. Zangl, B. Hein, R. Ramakers, W. Deferme, G. Zheng, and C. Duriez, "A model-based sensor fusion approach for force and shape estimation in soft robotics," *IEEE Robotics and automation letters*, vol. 5, no. 4, p. 5621–5628, 2020.
[63] M. Garstka, M. Cannon, and P. Goulart, "COSMO: A conic operator splitting method for convex conic problems," *arXiv preprint 1901.10887v2*, 2019.

APPENDIX

A. Projection onto the nonlinear complementarity set

Recall the nonlinear complementarity set, defined using the De Saxcé correction:

$$\mathbb{S}_c \coloneqq \left\{(\boldsymbol{\lambda}, \boldsymbol{\sigma}) \,\middle|\, \mathcal{K}_\mu \ni \boldsymbol{\lambda} \perp \boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma}) \in \mathcal{K}_\mu^*\right\}.$$

The NCP set can be rewritten as the disjoint union of three *modes*, over which the complementarity $\boldsymbol{\lambda} \perp \boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma})$ holds:

(M1) **Sticking:** $\boldsymbol{\lambda} \in \operatorname{int} \mathcal{K}_\mu$,
(M2) **Breaking:** $\boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma}) \in \operatorname{int} \mathcal{K}_\mu^*$,
(M3) **Sliding:** $\boldsymbol{\lambda} \in \partial\mathcal{K}_\mu$ *and* $\boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma}) \in \partial\mathcal{K}_\mu^*$.

We can, in each mode, characterize the value of either variable $(\boldsymbol{\lambda}, \boldsymbol{\sigma})$ *or* the relationship between them.

Proposition 1 (Description of NCP modes)**.** *Let* $(\boldsymbol{\lambda}, \boldsymbol{\sigma}) \in \mathbb{S}_c$. *Then, exactly one of the following holds:*

(M1) $\boldsymbol{\lambda} \in \operatorname{int} \mathcal{K}_\mu$ *and* $\boldsymbol{\sigma} = \mathbf{0}$.
(M2) $\sigma_N > 0$ *and* $\boldsymbol{\lambda} = \mathbf{0}$.
(M3) $\boldsymbol{\lambda} \in \partial\mathcal{K}_\mu$, $\sigma_N = 0$ *and* $\boldsymbol{\lambda}_T = -\mu\lambda_N \boldsymbol{\sigma}_T / \|\boldsymbol{\sigma}_T\|_2$ *(with* $\boldsymbol{\sigma}_T \neq 0$*)* or $\boldsymbol{\lambda} = \boldsymbol{\sigma} = \mathbf{0}$.

Proof. Obviously modes (M1) and (M2) are each disjoint from (M3). We continue the proof by disjunction.

Sticking mode (M1). Then, we have the strict inequality $\mu\lambda_N > \|\boldsymbol{\lambda}_T\|_2 \geq 0$. Writing out the complementarity condition and using the Cauchy-Schwarz inequality:

$$\begin{aligned} 0 &= \lambda_N(\sigma_N + \mu\|\boldsymbol{\sigma}_T\|_2) + \boldsymbol{\lambda}_T^\top \boldsymbol{\sigma}_T \\ &\geq \lambda_N(\sigma_N + \mu\|\boldsymbol{\sigma}_T\|_2) - \|\boldsymbol{\lambda}_T\|_2\|\boldsymbol{\sigma}_T\|_2 \\ &= \underbrace{\lambda_N}_{>0}\sigma_N + \underbrace{(\mu\lambda_N - \|\boldsymbol{\lambda}_T\|_2)}_{>0}\|\boldsymbol{\sigma}_T\|_2 \geq 0, \end{aligned}$$

which $\boldsymbol{\sigma} = \mathbf{0}$ excluding mode (M2).

Breaking mode (M2). $\boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma}) \in \operatorname{int} \mathcal{K}_\mu^*$ if and only if $\sigma_N > 0$. Once again using Cauchy-Schwarz leads to $\lambda_N\sigma_N = 0$, hence $\lambda_N = 0$ and $\boldsymbol{\lambda} = \mathbf{0}$. Projection onto this mode reduces to projecting $\boldsymbol{\sigma}$ onto the set $\{(\boldsymbol{y}_T, y_N) \mid y_N \geq 0\}$.

Sliding mode (M3). If $\boldsymbol{\lambda} \in \partial\mathcal{K}_\mu$, then $\mu\lambda_N = \|\boldsymbol{\lambda}_T\|_2$, $\sigma_N = 0$, and the following holds

$$0 = \mu\lambda_N\|\boldsymbol{\sigma}_T\|_2 + \boldsymbol{\lambda}_T^\top\boldsymbol{\sigma}_T \implies \boldsymbol{\lambda}_T^\top\boldsymbol{\sigma}_T = -\|\boldsymbol{\lambda}_T\|_2\|\boldsymbol{\sigma}_T\|_2,$$

which is the equality case for Cauchy-Schwarz, thus there exists $\alpha \geq 0$ such that $\boldsymbol{\lambda}_T = -\alpha\boldsymbol{\sigma}_T$. If $\boldsymbol{\sigma}_T \neq \mathbf{0}$, taking the norm leads to a single value of α. Otherwise $\boldsymbol{\lambda} = \boldsymbol{\sigma} = \mathbf{0}$. □

B. Monitoring inner convergence

The AL minimization step (i) requires reaching a prescribed stationarity tolerance ε_k for the minimization of $\mathcal{L}_\theta(\cdot,\cdot,\boldsymbol{y}_a,\boldsymbol{y}_c,\boldsymbol{y}_\sigma)$. Writing $\boldsymbol{v} \coloneqq (\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_c,\boldsymbol{\sigma}_c)$, $\widehat{\boldsymbol{v}} \coloneqq (\widehat{\boldsymbol{\lambda}}_a,\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c)$ and $\boldsymbol{y} \coloneqq (\boldsymbol{y}_a,\boldsymbol{y}_c,\boldsymbol{y}_\sigma)$ for short, we have

$$\nabla_{\boldsymbol{\lambda}_a}\mathcal{L}_\theta(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y}) = \nabla_{\boldsymbol{\lambda}_a}F(\boldsymbol{v}) + \boldsymbol{y}_a + \xi[\boldsymbol{\lambda}_a - \widehat{\boldsymbol{\lambda}}_a], \tag{28a}$$
$$\nabla_{\boldsymbol{\lambda}_c}\mathcal{L}_\theta(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y}) = \nabla_{\boldsymbol{\lambda}_c}F(\boldsymbol{v}) + \boldsymbol{y}_c + \eta[\boldsymbol{\lambda}_c - \widehat{\boldsymbol{\lambda}}_c], \tag{28b}$$
$$\nabla_{\boldsymbol{\sigma}_c}\mathcal{L}_\theta(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y}) = \nabla_{\boldsymbol{\sigma}_c}F(\boldsymbol{v}) + \boldsymbol{y}_\sigma + \rho[\boldsymbol{\sigma}_c - \widehat{\boldsymbol{\sigma}}_c], \tag{28c}$$
$$\partial_{\widehat{\boldsymbol{\lambda}}_a}\mathcal{L}_\theta(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y}) = \partial_{\widehat{\boldsymbol{\lambda}}_a}\mathcal{I}_{\mathbb{S}}(\widehat{\boldsymbol{v}}) - \boldsymbol{y}_a - \xi[\boldsymbol{\lambda}_a - \widehat{\boldsymbol{\lambda}}_a], \tag{28d}$$
$$\partial_{\widehat{\boldsymbol{\lambda}}_c}\mathcal{L}_\theta(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y}) = \partial_{\widehat{\boldsymbol{\lambda}}_c}\mathcal{I}_{\mathbb{S}}(\widehat{\boldsymbol{v}}) - \boldsymbol{y}_c - \eta[\boldsymbol{\lambda}_c - \widehat{\boldsymbol{\lambda}}_c], \tag{28e}$$
$$\partial_{\widehat{\boldsymbol{\sigma}}_c}\mathcal{L}_\theta(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y}) = \partial_{\widehat{\boldsymbol{\sigma}}_c}\mathcal{I}_{\mathbb{S}}(\widehat{\boldsymbol{v}}) - \boldsymbol{y}_\sigma - \rho[\boldsymbol{\sigma}_c - \widehat{\boldsymbol{\sigma}}_c]. \tag{28f}$$

Upon execution of steps (i.i)–(i.ii) by AltMin, it is clear that, for each j, the stationarity measure for minimizing $\mathcal{L}_{\theta_k}(\cdot,\cdot,\boldsymbol{y}^k)$ with respect to $\widehat{\boldsymbol{v}}$ evaluated at $(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j+1})$ is zero—directly by (i.ii). Moreover, owing to (i.i) we have

$$\begin{aligned}
0 &= \nabla_{\boldsymbol{\lambda}_a}\mathcal{L}_{\theta_k}(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j},\boldsymbol{y}^k),\\
0 &= \nabla_{\boldsymbol{\lambda}_c}\mathcal{L}_{\theta_k}(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j},\boldsymbol{y}^k),\\
0 &= \nabla_{\boldsymbol{\sigma}_c}\mathcal{L}_{\theta_k}(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j},\boldsymbol{y}^k).
\end{aligned}$$

Then, subtracting these zero contributions from the expressions in (28a)–(28c), we obtain

$$\begin{aligned}
\nabla_{\boldsymbol{\lambda}_a}\mathcal{L}_{\theta_k}(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j+1},\boldsymbol{y}^k) &= \xi_k(\widehat{\boldsymbol{\lambda}}_a^{k,j+1} - \widehat{\boldsymbol{\lambda}}_a^{k,j}),\\
\nabla_{\boldsymbol{\lambda}_c}\mathcal{L}_{\theta_k}(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j+1},\boldsymbol{y}^k) &= \eta_k(\widehat{\boldsymbol{\lambda}}_c^{k,j+1} - \widehat{\boldsymbol{\lambda}}_c^{k,j}),\\
\nabla_{\boldsymbol{\sigma}_c}\mathcal{L}_{\theta_k}(\boldsymbol{v}^{k,j+1},\widehat{\boldsymbol{v}}^{k,j+1},\boldsymbol{y}^k) &= \rho_k(\widehat{\boldsymbol{\sigma}}_c^{k,j+1} - \widehat{\boldsymbol{\sigma}}_c^{k,j}).
\end{aligned}$$

This leads to the inner residual defined in (23) for monitoring the convergence of AltMin.

C. Lagrangian derivation of stationarity conditions

Aiming at the stationarity conditions presented in Sec. III-G, we first rewrite (11) in the equivalent form

$$\begin{aligned}
\underset{\boldsymbol{v},\widehat{\boldsymbol{v}}}{\text{minimize}} \quad & f(\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_c) + \mathcal{I}_{\mathbb{B}}(\widehat{\boldsymbol{\lambda}}_a) + \mathcal{I}_{\mathbb{S}_c}(\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c) \qquad (29)\\
\text{subject to} \quad & \boldsymbol{\lambda}_a = \widehat{\boldsymbol{\lambda}}_a, \quad \boldsymbol{\lambda}_c = \widehat{\boldsymbol{\lambda}}_c, \quad \boldsymbol{\sigma}_c = \widehat{\boldsymbol{\sigma}}_c,\\
& \boldsymbol{\sigma}_c = C\boldsymbol{\lambda}_a + D\boldsymbol{\lambda}_c + \boldsymbol{g}.
\end{aligned}$$

Pairing each equality constraint with a multiplier, the Lagrangian function for (29) is given by

$$\begin{aligned}
\mathcal{L}(\boldsymbol{v},\widehat{\boldsymbol{v}},\boldsymbol{y},\boldsymbol{y}_{\text{eq}}) \coloneqq\ & f(\boldsymbol{\lambda}_a,\boldsymbol{\lambda}_c) + \mathcal{I}_{\mathbb{B}}(\widehat{\boldsymbol{\lambda}}_a) + \mathcal{I}_{\mathbb{S}_c}(\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c)\\
& + \boldsymbol{y}_a^\top(\boldsymbol{\lambda}_a - \widehat{\boldsymbol{\lambda}}_a) + \boldsymbol{y}_c^\top(\boldsymbol{\lambda}_c - \widehat{\boldsymbol{\lambda}}_c) + \boldsymbol{y}_\sigma^\top(\boldsymbol{\sigma}_c - \widehat{\boldsymbol{\sigma}}_c)\\
& + \boldsymbol{y}_{\text{eq}}^\top(C\boldsymbol{\lambda}_a + D\boldsymbol{\lambda}_c + \boldsymbol{g} - \boldsymbol{\sigma}_c)
\end{aligned}$$

and the first-order optimality conditions for (29) can be obtained by differentiation of $\mathcal{L}$ with respect to primal-dual variables:

$$\begin{aligned}
0 &= \nabla_{\boldsymbol{\lambda}_a}f(\boldsymbol{\lambda}_a^\star,\boldsymbol{\lambda}_c^\star) + \boldsymbol{y}_a^\star + C^\top\boldsymbol{y}_{\text{eq}}^\star, && [\text{from } \nabla_{\boldsymbol{\lambda}_a}]\\
0 &= \nabla_{\boldsymbol{\lambda}_c}f(\boldsymbol{\lambda}_a^\star,\boldsymbol{\lambda}_c^\star) + \boldsymbol{y}_c^\star + D^\top\boldsymbol{y}_{\text{eq}}^\star, && [\text{from } \nabla_{\boldsymbol{\lambda}_c}]\\
0 &= \boldsymbol{y}_\sigma^\star - \boldsymbol{y}_{\text{eq}}^\star, && [\text{from } \nabla_{\boldsymbol{\sigma}_c}]\\
0 &\in \mathcal{N}_{\mathbb{B}}(\widehat{\boldsymbol{\lambda}}_a^\star) - \boldsymbol{y}_a^\star, && [\text{from } \partial_{\widehat{\boldsymbol{\lambda}}_a}]\\
0 &\in \mathcal{N}_{\mathbb{S}_c}^{\lim}(\widehat{\boldsymbol{\lambda}}_c^\star,\widehat{\boldsymbol{\sigma}}_c^\star) - (\boldsymbol{y}_c^\star,\boldsymbol{y}_\sigma^\star), && [\text{from } \partial_{(\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c)}]\\
0 &= \boldsymbol{\lambda}_a^\star - \widehat{\boldsymbol{\lambda}}_a^\star, && [\text{from } \nabla_{\boldsymbol{y}_a}]\\
0 &= \boldsymbol{\lambda}_c^\star - \widehat{\boldsymbol{\lambda}}_c^\star, && [\text{from } \nabla_{\boldsymbol{y}_c}]\\
0 &= \boldsymbol{\sigma}_c^\star - \widehat{\boldsymbol{\sigma}}_c^\star, && [\text{from } \nabla_{\boldsymbol{y}_\sigma}]\\
0 &= C\boldsymbol{\lambda}_a^\star + D\boldsymbol{\lambda}_c^\star + \boldsymbol{g} - \boldsymbol{\sigma}_c^\star, && [\text{from } \nabla_{\boldsymbol{y}_{\text{eq}}}]
\end{aligned}$$

where $\mathcal{N}_{\mathbb{S}_c}^{\lim}(\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c)$ denotes the limiting normal cone to $\mathbb{S}_c$ at $(\widehat{\boldsymbol{\lambda}}_c,\widehat{\boldsymbol{\sigma}}_c)$, as in [13, §2.2]. Solving for the trivial equalities

$$(\widehat{\boldsymbol{\lambda}}_a^\star,\widehat{\boldsymbol{\lambda}}_c^\star,\widehat{\boldsymbol{\sigma}}_c^\star,\boldsymbol{y}_{\text{eq}}^\star) = (\boldsymbol{\lambda}_a^\star,\boldsymbol{\lambda}_c^\star,\boldsymbol{\sigma}_c^\star,\boldsymbol{y}_\sigma^\star)$$

and substituting back, we find the stationarity conditions for the constrained problem (10), without splitting variables:

$$\begin{aligned}
0 &= \nabla_{\boldsymbol{\lambda}_a}f(\boldsymbol{\lambda}_a^\star,\boldsymbol{\lambda}_c^\star) + \boldsymbol{y}_a^\star + C^\top\boldsymbol{y}_\sigma^\star,\\
0 &= \nabla_{\boldsymbol{\lambda}_c}f(\boldsymbol{\lambda}_a^\star,\boldsymbol{\lambda}_c^\star) + \boldsymbol{y}_c^\star + D^\top\boldsymbol{y}_\sigma^\star,\\
\boldsymbol{y}_a^\star &\in \mathcal{N}_{\mathbb{B}}(\boldsymbol{\lambda}_a^\star),\\
(\boldsymbol{y}_c^\star,\boldsymbol{y}_\sigma^\star) &\in \mathcal{N}_{\mathbb{S}_c}^{\lim}(\boldsymbol{\lambda}_c^\star,\widehat{\boldsymbol{\sigma}}_c^\star),\\
\boldsymbol{\sigma}_c^\star &= C\boldsymbol{\lambda}_a^\star + D\boldsymbol{\lambda}_c^\star + \boldsymbol{g}.
\end{aligned}$$

By definition of f from Sec. III-A, the first two conditions yield dual feasibility (25). The remaining conditions directly imply primal feasibility (24), while a careful case-by-case inspection of the (limiting) normal cone leads to the complementarity conditions.

D. Preconditioning via Ruiz equilibration

To improve numerical stability and reduce the condition number of the linear system (18), we perform a diagonal preconditioning step inspired by Ruiz equilibration [53]. We use an averaging scheme similar to a modified Ruiz equilibration applied to conic problems [63, §3.5].

This scaling is applied only for the resolution of the linear system and projection. All variables involved in the optimization (primal and dual) are scaled accordingly during the computation and unscaled afterward before evaluating optimality or stationary conditions.