On Solving the Dynamics of Constrained Rigid Multi-Body Systems with Kinematic Loops

Vassilios Tsounis, Ruben Grandia, and Moritz Bächer

***Abstract*—This technical report provides an in-depth evaluation of both established and state-of-the-art methods for simulating constrained rigid multi-body systems with hard-contact dynamics, using formulations of Nonlinear Complementarity Problems (NCPs). We are particularly interest in examining the simulation of highly coupled mechanical systems with multitudes of closed-loop bilateral kinematic joint constraints in the presence of additional unilateral constraints such as joint limits and frictional contacts with restitutive impacts. This work thus presents an up-to-date literature survey of the relevant fields, as well as an in-depth description of the approaches used for the formulation and solving of the numerical time-integration problem in a maximal coordinate setting. More specifically, our focus lies on a version of the overall problem that decomposes it into the forward dynamics problem followed by a time-integration using the states of the bodies and the constraint reactions rendered by the former. We then proceed to elaborate on the formulations used to model frictional contact dynamics and define a set of solvers that are representative of those currently employed in the majority of the established physics engines. A key aspect of this work is the definition of a benchmarking framework that we propose as a means to both qualitatively and quantitatively evaluate the performance envelopes of the set of solvers on a diverse set of challenging simulation scenarios. We thus present an extensive set of experiments that aim at highlighting the absolute and relative performance of all solvers on particular problems of interest as well as aggravatingly over the complete set defined in the suite.**

I. Introduction

Simulating complex mechanical systems such as multi-limb robots that can walk and manipulate is a key element in today's workflows for the development and testing of such systems. Indeed the availability of fast and accurate physical simulation has recently proved to be fundamental in applying state-of-the-art methods such as Deep Reinforcement Learning (DRL) for their control [1], [2]. One important challenge the field currently faces is being able to design and control robots with highly coupled mechanical assemblies that exhibit intrinsic closed kinematic loops [3]. Such configurations are often used as means for transferring power and motion with reduced requirements on actuation compared to serial chains.

Traditionally, however, the latter approach has been employed and the systems have been designed to be as simple as possible, both mechanically and morphologically. Examples include: a) actuating all joints, i.e. having no passive Degrees-of-Freedom (DoF), b) simplifying mechanical power transmission by placing actuators directly at or as near to the joint DoF as possible, and c) avoiding the use of parallel actuation that induces closed-loop kinematics. Besides the obvious benefits in terms of hardware design, in large part, the aforementioned conventions have also served to simplify simulating and controlling walking systems. In the presence of singularities and hyperstaticity, the system dynamics can become exceptionally challenging to solve, whether it be their forward dynamics for simulation, or inverse dynamics for control. As more advanced control methods that circumvent these challenges by employing nonlinear optimization and/or DRL have become widely available, it remains an open problem of how can we make our physics engines capable of accurately and efficiently simulating systems with arbitrary mechanical assemblies.

Such a capability would have the potential to revolutionize the kinds of robots that could be developed. In the case of DRL in particular, which makes no assumptions on the model of the system, it is only limited in practice by the throughput of the simulation due to the large sample complexity required for convergence [1], [4], [5]. Therefore if physically accurate and fast simulation for complex mechanisms can achieve an effective throughput on par with that of articulated systems, it would open up the possibility of applying DRL to any type of robot, and even more types of mechanical systems in general.

Broadly speaking, the requirements placed on such universal simulators could be considered in terms of two categories:

- Modeling requirements: these are the types of constraints that can be supported. These include bilateral kinematic constraints such as passive, actuated, binary or unary joints that are common in mechanical assemblies, closed-loop kinematics and redundancy, as well as unilateral constraints such as frictional contacts and joint limits.
- Performance requirements: physical plausibility[1] and the relative trade-off between accuracy and speed that a given approach would exhibit. In this context, accuracy is defined strictly in terms of constraint satisfaction and sensitivity of the resulting constraint reactions (i.e. forces and torques) w.r.t the state of the system. Simulation speed is often conflated with sample throughput. In this work we consider speed strictly in the single-instance sense, i.e. in the absence of mass parallelization, for example on a GPU.

Satisfying all of the aforementioned modeling requirements, however, bears significant numerical challenges. Firstly, the forward dynamics problem can be ill-conditioned in the presence of large mass ratios and hyperstaticity due to highly coupled constraints such as closed-loop kinematics and underactuation. Secondly, constraint satisfaction at configuration-level

All authors are with Disney Research Imagineering, Zurich, Switzerland. e-mail: *vassilios.tsounis@disney.com*

[1]Physical realism is a difficult assertion to make when simulating rigid-body systems with discrete contacts. A more conservative and fair view would be to assert physical plausibility in terms of respecting kinematic constraints and based on the regularity of the computed constraint forces and torques.

(i.e. joints breaking, limits being exceeded and body interpenetrations) can occur as a result of inaccurately estimating constraint reactions, and can accumulate prohibitively over multiple simulation steps. Moreover, another consequence of the ill-conditioning is the potential irregularity of the yielded constraint reactions, which can become extremely sensitive to small changes in the state of the system and thus change erratically over subsequent time-steps although the state has experienced changes on scales close to the numerical precision. Lastly, the stability of the solution can deteriorate when using larger time-steps, which is often necessary for increasing simulation throughput in applications such as DRL.

In this paper we present an empirical evaluation of existing approaches for solving the forward dynamics of constrained rigid-body systems. We consider cases that include bilateral kinematic constraints, of arbitrary connectivity and under/over actuation, as well as additional unilateral constraints that model restitutive impacts, frictional contacts and joint limits. Specifically, we compare both established and state-of-the-art algorithms for solving the Nonlinear Complementarity Problem (NCP) [6] derived from the *dual problem* of forward dynamics, i.e. the resolution of constraint reactions, and posited as a Nonlinear Second-Order Cone Program (NSOCP) [7].

Our evaluation consists of a benchmark suite of systems of varying complexity that range from primitive toy problems to full-scale robotic and *Audio-Animatronic*® systems. Moreover, we also evaluate techniques such such as constraint relaxation and proximal optimization in order to deal with the aforementioned types of ill-conditioning. Our comparisons are performed using set of performance metrics based on first principles as well as practical considerations, that summarily speaking, cover physical plausibility, accuracy, speed and robustness in as fair terms as possible.

As done in previous work [8], [9], [10], we employ a *maximal-coordinate* formulation of Constrained Rigid-Body Dynamics (CRBD). In contrast to using *minimal-coordinates*, this results in expressing kinematic constraints explicitly and makes it very straight-forward to model systems with kinematic loops and passive joints. In addition, it facilitates a versatile parameterization of the system kinematics that can support both unary and binary joints over a very extensive set of mechanically feasible joint types. Conversely, a potentially more computationally efficient approach could be presented in combining minimal-coordinates with constraints, i.e. Constrained Articulated-Body Dynamics (CABD) [11], [12], [13], [14], [15]. However, as the scope of this study lies mainly in comparing algorithms specialized for dynamics problems, using CRBD instead of CABD serves merely as a benchmark to better highlight differences between solvers. We assert that if a method can be made to work effectively in the more challenging case of CRBD, then employing CABD can only serve to further simplify the problem by reducing its overall dimensionality.

Contributions: This work thus makes the following contributions within the context of robotics and control:

1) We provide an extensive survey of the current state-of-the-art in physical simulation of rigid-body systems and evaluate the reproducibility of highly-cited works.
2) We define a suite of benchmark problems and performance metrics that can be used to systematically evaluate the relative performance of forward dynamics solvers. The suite consists of a set of problems that are relevant and meaningful within the context of robotics but have applicability to even broader classes of systems.
3) We present an extensive evaluation of relevant algorithms on the aforementioned benchmark suite and describe practical recommendations on using them to realize physics simulators for constrained systems.

II. Constrained Rigid-Body Mechanics

This section provides an overview of the mechanics of constrained rigid-body systems, and outlines the problems that arise within the context of physical simulation. The process of physical simulation itself, is most often viewed as solving a single high-level problem, comprised of three constituent sub-problems, namely: Event Detection (ED), Forward Dynamics (FD), and Time Integration (TI). Indeed, this decomposition is ubiquitous across all known physics engines [16], [17]. The first essentially corresponds to Collision Detection (CD), but can be viewed in the general sense of detecting events where the set of active constraints changes. In this way we can also incorporate detection of active joint limits, and possibly other elements that are evaluated at configuration-level. However, this work does not cover CD methods themselves. We recommend interested readers to consider [18], [19], [20], [21] for more details on the topic of CD. The FD and TI sub-problems are described in the next section.

A. Forward Dynamics

The forward dynamics problem is one of evaluating how the motion of the system will evolve at a certain time $t \in \mathbb{R}$ given its state. Broadly speaking, assuming the set of constraints is immutable, and for a given choice of generalized coordinates $\mathbf{s} := \mathbf{s}(t) \in \mathbb{R}^{n_s}$ and generalized velocities $\mathbf{u} := \mathbf{u}(t) \in \mathbb{R}^{n_u}$ with $n_s \geq n_u$, the FD problem can be stated generically as:

$$\textbf{Find} \quad \dot{\mathbf{u}} \in \mathbb{R}^{n_u} \ , \dot{\mathbf{s}} \in \mathbb{R}^{n_s} \quad \text{s.t.} :$$

$$\dot{\mathbf{s}} = \mathbf{H}(\mathbf{s})\, \mathbf{u} \tag{1a}$$

$$\mathbf{M}(\mathbf{s})\, \dot{\mathbf{u}} = \mathbf{h}(\mathbf{s}, \mathbf{u}) + \mathbf{J}(\mathbf{s})^T \boldsymbol{\lambda} \tag{1b}$$

$$\mathbf{f}(\mathbf{s}, \mathbf{u}) = 0 \tag{1c}$$

$$\mathcal{K}^* \ni \mathbf{g}(\mathbf{s}, \mathbf{u}) \perp \boldsymbol{\lambda} \in \mathcal{K} \quad . \tag{1d}$$

$\mathbf{H}(\mathbf{s}) \in \mathbb{R}^{n_s \times n_u}$ is a matrix mapping generalized velocities to the time-derivative of the generalized coordinates. Such a mapping is often necessary for floating-base systems, where $\dot{\mathbf{s}} \neq \mathbf{u}$ due to the parameterization of rotations, e.g. quaternions, Euler angles, etc. $\mathbf{M}(\mathbf{s}) \in \mathbb{S}^{n_u}_{++}$ is the generalized mass matrix, where $\mathbb{S}^{n_u}_{++}$ denotes the set of $n \times n$ symmetric positive-definite (SPD)[2] matrices. $\mathbf{h}(\mathbf{s}, \mathbf{u}) \in \mathbb{R}^{n_u}$ is the vector

[2]When $\mathbf{s}$ is defined with either minimal (independent) coordinates or maximal coordinates then $\mathbf{M}(\mathbf{s})$ is always SPD. However, if the coordinates of $\mathbf{s}$ are dependent or redundant, then it is possible for $\mathbf{M}(\mathbf{s})$ to be singular, and thus positive semi-definite. See [22] for a detailed analysis on the matter.

of non-linear generalized force terms, which, can include gravity, Coriolis, centrifugal, actuation and external effects. $\mathbf{f} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_e}$ and $\mathbf{g} : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_i}$ are respectively the set of equality and inequality constraints in implicit form. $\mathbf{J}(\mathbf{s}) \in \mathbb{R}^{n_d \times n_u}$ is the constraint Jacobian[3] matrix, where $n_d = n_e + n_i$. $\boldsymbol{\lambda} \in \mathcal{K} \subseteq \mathbb{R}^{n_d}$ is the vector of Lagrange multipliers corresponding to constraint forces, where $\mathcal{K}$ is a closed and convex cone and $\mathcal{K}^*$ denotes its dual.

Fundamentally, (1a-b) form a system of Differential Algebraic Equations (DAE). However, since $\boldsymbol{\lambda} \in \mathcal{K}$ is bound by set-valued force laws [23] (e.g. unilateral contact and Coulomb friction), the system becomes a Differential Inclusion (DI). Thus, (1) overall is a Variational Inequality (VI) [6], that due to the feasible set being a *cone*, forms a Complementarity Problem (CP). Lastly, due to the structure of the complementarity constraints (1d), which in general may be nonlinear, the FD problem (1) is therefore an NCP.

B. Impulsive Dynamics

One of the primary difficulties in modeling the contact dynamics of rigid bodies is that impacts and friction induce instantaneous changes to $\mathbf{u}(t)$ and $\dot{\mathbf{u}}(t)$ as well as to the set of active constraints. The discontinuity of the former thus impacts their integrability, as they cease to be continuous functions in t. In order to deal with this non-smoothness, we can employ techniques from *non-smooth dynamics* [23], [24]. Specifically, *differential measures* [25] provide a means to decompose the velocity differential into smooth and impulsive terms as

$$d\mathbf{u} = \dot{\mathbf{u}}(t)\, dt + (\mathbf{u}^+ - \mathbf{u}^-)\, d\eta \ , \tag{2}$$

where $\dot{\mathbf{u}}(t)$ is the smooth acceleration and $\mathbf{u}^-, \mathbf{u}^+$ are the pre- and post-event impulsive velocities, respectively. The key ingredient in this model is the impulsive differential $d\eta$. Assuming $n_p \in \mathbb{N}_{\geq 0}$ impulsive events occur at times $t_i \in [t_0, t]\ \forall\, i \in \{1, \dots, n_p\}$, the set of discrete impulsive events can be denoted as $I_p = \bigcup_{i=1}^{n_p} \{t_i\}$. Then, $d\eta$ represents a finite sum of so-called *Dirac point measures* over $t \in \mathbb{R}$:

$$d\eta := \sum_{t_i \in I_p} d\delta_{t_i} \ , \quad \int_{t_0}^{t} d\delta_{t_i} = \begin{cases} 1 \ , \ t_i \in [t_0,\, t] \\ 0 \ , \ t_i \notin [t_0,\, t] \end{cases} \ , \tag{3}$$

thus allowing the time-integration of the impulsive velocity as

$$\int_{t_0}^{t} \left(\mathbf{u}^+(\tau) - \mathbf{u}^-(\tau)\right)\, d\eta = \sum_{t_i \in I_p} \left(\mathbf{u}^+(t_i) - \mathbf{u}^-(t_i)\right) \ . \tag{4}$$

Equivalently, the differential measure over generalized forces

$$d\mathbf{p} = \mathbf{w}_s\, dt + \mathbf{w}_{ns}\, d\eta \ , \tag{5}$$

is often referred to as *percussion*, where $\mathbf{w}_s, \mathbf{w}_{ns}$ correspond to the smooth and non-smooth (i.e. impulsive) generalized forces, respectively. Thus, (2) and (5) enable the decomposition of the system dynamics into respective smooth and impulsive parts

$$\mathbf{M}(\mathbf{s})\, \dot{\mathbf{u}}\, dt = \mathbf{h}(\mathbf{s}, \mathbf{u})\, dt + \mathbf{J}_s(\mathbf{s})^T\, \boldsymbol{\lambda}_s\, dt \ , \tag{6a}$$

$$\mathbf{M}(\mathbf{s})\, \left(\mathbf{u}^+ - \mathbf{u}^-\right)\, d\eta = \mathbf{J}_{ns}(\mathbf{s})^T\, \boldsymbol{\lambda}_{ns}\, d\eta \ , \tag{6b}$$

where $\mathbf{w}_s = \mathbf{J}_s(\mathbf{s})^T\, \boldsymbol{\lambda}_s$ and $\mathbf{w}_{ns} = \mathbf{J}_{ns}(\mathbf{s})^T\, \boldsymbol{\lambda}_{ns}$ are necessarily distinct as the set of active constraints may differ between the continuous dynamics and those at the instants of discontinuity. Moreover, while $\boldsymbol{\lambda}_s$ represent forces, $\boldsymbol{\lambda}_{ns}$ now represent *impulses*. Herein we refer to both as *constraint reactions*.

Therefore, whenever an impulsive event occurs at some time t_i, the FD problem (1) must be rephrased to employ (6b) in-place of (6a). This leads to the impulsive FD problem

Find $\mathbf{u}^+ \in \mathbb{R}^{n_u}\ , \dot{\mathbf{s}} \in \mathbb{R}^{n_s}$ s.t. :

$$\dot{\mathbf{s}} = \mathbf{H}(\mathbf{s})\, \mathbf{u}^+ \tag{7a}$$

$$\mathbf{M}(\mathbf{s})\, \left(\mathbf{u}^+ - \mathbf{u}^-\right) = \mathbf{J}(\mathbf{s})^T\, \boldsymbol{\lambda} \tag{7b}$$

$$\mathbf{f}(\mathbf{s}, \mathbf{u}^-, \mathbf{u}^+) = 0 \tag{7c}$$

$$\mathcal{K}^* \ni \mathbf{g}(\mathbf{s}, \mathbf{u}^-, \mathbf{u}^+) \perp \boldsymbol{\lambda} \in \mathcal{K} \ , \tag{7d}$$

where all time-dependent configuration-level quantities are evaluated at time t_i, e.g. $\mathbf{s} := \mathbf{s}(t_i)$, while the pre-event generalized velocity $\mathbf{u}^- = \mathbf{u}^-(t_i)$ is corresponds to the lower limit value $\lim_{\tau \to 0} \mathbf{u}(t_i - \tau)$ resulting from the time-integration of the smooth dynamics (1). Crucially, (7) can be understood as a boundary-value problem w.r.t time, as it is only admissible at the discrete atomic time-points t_i. Overall, the impulsive FD problem (7), like its smooth counterpart (1), is also a NCP, but one that now yields the system's generalized velocities directly, instead of the generalized accelerations.

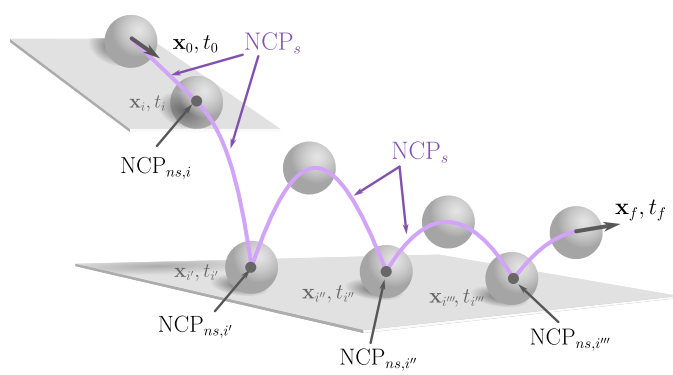

Fig. 1. A visual depiction of the non-smooth initial-value problem (8) using the example of a rigid sphere rolling-off of one plane and then bouncing on another. When the sphere drops the first plane, the impact would result in a contact that is initially open, then closes momentarily and opens once again. Although the position and orientation of the sphere would remain continuous functions of time, the linear and angular velocities would change instantaneously in order to satisfy the contact constraints. This means that the velocities are in effect not continuous, and nor are the positions and orientations smooth functions of time. Therefore, the sphere's trajectory consists of the piece-wise smooth segments (violet lines) and atomic points $\bigcup_i t_i$ at the moments of impact (gray nodes). The former is determined by the smooth dynamics represented by the acceleration-level $\mathrm{NCP}_{s,t}$ of (1), while the latter defined at each instance t_i is determined by the impulsive NCP_{ns,t_i} of (7). Direct time-integration is intractable analytically thus necessitating the application of time-discretization to approximate the integral in (8c).

[3] With the given form for $\mathbf{f}(\mathbf{s}, \mathbf{u})$ and $\mathbf{g}(\mathbf{s}, \mathbf{u})$, $\mathbf{J}(\mathbf{s}) = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} & \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \end{bmatrix}$. Alternatively, if the constraints were formed at configuration level, i.e. were in the form of $\mathbf{f}(\mathbf{s})$ and $\mathbf{g}(\mathbf{s})$, and $\mathbf{s}$ did not employ reduced parameterizations of rotation, then the Jacobian would be defined as $\mathbf{J}(\mathbf{s}) = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{s}} & \frac{\partial \mathbf{g}}{\partial \mathbf{s}} \end{bmatrix}$.

C. Time Integration

Given means to evaluate the forward dynamics of the constrained rigid-body system, pointwise for any time t, simulation amounts to time-integrating the former over some finite time interval $I = [t_0, t_f]$. This corresponds to solving an *Initial Value Problem* (IVP) in the form of

$$\textbf{For } t \in I := [t_0, t_f], \text{ and } I_p := \bigcup_{i=1}^{n_p} \{t_i\} \subset I$$

$$\textbf{With } \mathbf{x}(t) := \begin{bmatrix} \mathbf{s}(t) \\ \mathbf{u}(t) \end{bmatrix} \tag{8a}$$

$$\textbf{Find } \mathbf{x}(t)\,, \text{ s.t.}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{8b}$$

$$\mathbf{x}(t) = \mathbf{x}_0 + \begin{bmatrix} \displaystyle\int_{t_0}^{t} \mathbf{H}(\mathbf{s}(\tau))\,\mathbf{u}(\tau)\,d\tau \\ \displaystyle\sum_{t_i \in I_p} \left(\mathbf{u}^+(t_i) - \mathbf{u}^-(t_i)\right) + \int_{t_0}^{t} \dot{\mathbf{u}}(t)\,d\tau \end{bmatrix} \tag{8c}$$

$$\text{NCP}_{ns,t_i}\,(7) \begin{cases} \mathbf{M}\left(\mathbf{u}^+(t_i) - \mathbf{u}^-(t_i)\right) = \mathbf{J}_{ns}^T \boldsymbol{\lambda}_{ns} & \text{(8d)} \\ \mathbf{f}_{ns}(\mathbf{s}(t_i), \mathbf{u}^+(t_i)) = 0 & \text{(8e)} \\ \mathcal{K}_{ns}^* \ni \mathbf{g}_{ns}(\mathbf{s}(t_i), \mathbf{u}^+(t_i)) \perp \boldsymbol{\lambda}_{ns} \in \mathcal{K}_{ns} & \text{(8f)} \end{cases}$$

$$\text{NCP}_{s,t}\,(1) \begin{cases} \mathbf{M}\,\dot{\mathbf{u}}(t) = \mathbf{h} + \mathbf{J}_s^T \boldsymbol{\lambda}_s & \text{(8g)} \\ \mathbf{f}_s(\mathbf{s}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t)) = 0 & \text{(8h)} \\ \mathcal{K}_s^* \ni \mathbf{g}_s(\mathbf{s}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t)) \perp \boldsymbol{\lambda}_s \in \mathcal{K}_s & \text{(8i)} \end{cases}$$

Fig. 1 provides a visual reference for understanding the structure of the IVP (8). Essentially, it involves three distinct sub-problems. The first and second, we recognize to be the aforedescribed FD problems NCP_{ns,t_i} (7) and $\text{NCP}_{s,t}$ (1), which yield $\dot{\mathbf{u}}(\tau)$ and $\mathbf{u}^+(t_i)$ pointwise for any t and t_i, respectively, and the third corresponds to evaluating the time-integrals in (8c). However, as the NCP_{ns,t_i} and $\text{NCP}_{s,t}$ can only be evaluated pointwise, they cannot be solved in closed-form to yield $\dot{\mathbf{u}}(\tau)$ and $\mathbf{u}(t)$ as integrable functions. Therefore, the IVP (8) is analytically intractable in general, and solving it necessitates the use of time-discretization and the application of numerical techniques.

III. Problem Formulation

This section outlines the derivation of a computationally tractable approximation of the IVP (8) summarized previously. Specifically, we describe the set of approximations, w.r.t time-discretization, that are required to transcribe the integration and NCP sub-problems, and introduce a reduced form of the latter that is amenable to numerical optimization techniques.

A. Time Discretization

In order to render the IVP (8) tractable, appropriate time-discretization and time-integration schemes are required. Among the most widely adopted are the so-called *time-stepping methods* [25], [26]. These progress the state over series of pre-determined time-intervals and evaluate the system dynamics only on impulse-velocity level. In doing so, every simulation step is able to handle all types of impulsive events in a unified fashion. They are also able to resolve phenomena such as Painlevé's paradox [27] that occur when evaluating the dynamics at force-acceleration level.

Assuming a fixed time-step $\Delta t \in \mathbb{R}_{\geq 0}$, and considering all impulsive events occurring within the closed interval $[t_0, t_0 + \Delta t]$ to happen simultaneously, the smooth and impulsive dynamics of (6) can be combined as

$$\mathbf{M}\,\left(\mathbf{u}^+ - \mathbf{u}^-\right) = \Delta t\,\mathbf{h} + \mathbf{J}^T \boldsymbol{\lambda}\,, \tag{9}$$

and the IVP over a single-step interval $[t_0, t_0 + \Delta t]$ becomes

$$\textbf{For } t \in I := [t_0, t_0 + \Delta t]$$

$$\textbf{Find } \mathbf{x}(t_0 + \Delta t)\,, \text{ s.t.}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{10a}$$

$$\mathbf{x}(t_0 + \Delta t) = \begin{bmatrix} \mathbf{s}(t_0) \boxplus \mathbf{H}(\mathbf{s}(t_0))\,\mathbf{u}^+(t_0 + \Delta t) \\ \mathbf{u}^+(t_0 + \Delta t) \end{bmatrix} \tag{10b}$$

$$\text{NCP}_{\Delta t} \begin{cases} \mathbf{M}\left(\mathbf{u}^+ - \mathbf{u}^-\right) = \Delta t\,\mathbf{h} + \mathbf{J}^T \boldsymbol{\lambda} & \text{(10c)} \\ \mathbf{f}(\mathbf{s}, \mathbf{u}^+) = 0 & \text{(10d)} \\ \mathcal{K}^* \ni \mathbf{g}(\mathbf{s}, \mathbf{u}^+) \perp \boldsymbol{\lambda} \in \mathcal{K}\,, & \text{(10e)} \end{cases}$$

where the $\boxplus$ denotes the addition operator specific to the parameterization of rotations used to define $\mathbf{s}$. In this work we make use of the *exponential map* to compute integrals of orientation as the concatenation of rotation operators [28].

Thus the single-step IVP (10) formulated over the interval $[t_0, t_0 + \Delta t]$, amounts to solving a single NCP to yield the generalized velocities $\mathbf{u}(t_0 + \Delta t)$ and then forward integrating to render the next-step generalized coordinates $\mathbf{s}(t_0 + \Delta t)$. This integration scheme is often referred to as the *semi-implicit Euler* method, as velocities are computed implicitly via the NCP, while generalized positions are integrated explicitly.

Although in this work we only consider the first-order forward integration of the generalized coordinates, higher-order integration methods such as Runge-Kutta can be conceived, that would solve the NCP at multiple internal steps to render more accurate, and possibly more stable state integrators. Furthermore, another integration method commonly used in non-smooth mechanics formulations is the mid-point time-stepping scheme of J.J. Moreau [25]. It is similar to the semi-implicit Euler method, except that all intermediate quantities, including event detection, are evaluated from the state at the middle of the time-step computed using backward Euler.

B. The Dual Problem

A common strategy for solving the NCP sub-problem (10c), is by taking an alternative view that rewrites the problem in terms of the constraint reactions $\boldsymbol{\lambda}$. Generally speaking, this alternative formulation can be referred to as the *dual*

problem of forward dynamics. Doing so, the FD problem can be solved by simple back-substitution to the respective equations of motion (9), to yield $\mathbf{u}^+$. Although a multitude of approaches exist for deriving the dual problem, we consider the most principled to be that which employs *Gauss' principle of duality* and the *extended principle of least constraint* [29], [23]. At this stage, we will only outline the formulation of the dual FD problem, while its construction will be detailed in Sec. V, and a complete derivation is provided in Appendix. B.

Projecting (10c) to the *space of the constraints*, the dynamics of the system are summarized by the following quantities:

1) the inverse apparent inertia, a.k.a the *Delassus* matrix
$$\mathbf{D} := \mathbf{J}^T \mathbf{M}^{-1} \mathbf{J} \tag{11}$$
2) the unconstrained, a.k.a free, velocity
$$\mathbf{v}_f := \mathbf{J} \left(\mathbf{u}^- + \Delta t \, \mathbf{M}^{-1}\mathbf{h}\right) + \mathbf{v}^* \tag{12}$$
3) post-event constraint velocity
$$\mathbf{v}^+(\boldsymbol{\lambda}) := \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f \tag{13}$$
4) augmented post-event constraint velocity
$$\hat{\mathbf{v}}(\boldsymbol{\lambda}) := \mathbf{v}^+(\boldsymbol{\lambda}) + \boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) \tag{14}$$

where $\mathbf{v}^*$ is an auxiliary bias term that allows us to introduce additional elements such as a model of impacts and constraint stabilization, and $\boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda}))$ is the non-linear De Saxcé correction [30], [31]. The latter is an operator over the post-event constraint velocity and is the principle source of non-linearity in the problem, as detailed in Sec. IV-D. Thus, the NCP formulated for the dual FD is stated, rather concisely, as

$\text{NCP}(\mathbf{D}, \mathbf{v}_f, \mathcal{K})$:

$$\textbf{Find } \boldsymbol{\lambda}, \text{ s.t. } \quad \mathcal{K}^* \ni \hat{\mathbf{v}}(\boldsymbol{\lambda}) \perp \boldsymbol{\lambda} \in \mathcal{K} \ . \tag{15}$$

Furthermore, (13)-(15) correspond to the KKT conditions of a Nonlinear Second-Order Cone Program (NSOCP). This perspective motivates the application of relevant techniques which can be used to solve the NCP via the transcription

$\text{NSOCP}(\mathbf{D}, \mathbf{v}_f, \mathcal{K})$:

$$\textbf{Find } \boldsymbol{\lambda} = \underset{\mathbf{x}\in\mathcal{K}}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{x}^T \mathbf{D}\,\mathbf{x} + \mathbf{x}^T \left(\mathbf{v}_f + \boldsymbol{\Gamma}(\mathbf{v}^+(\mathbf{x}))\right) \tag{16}$$

Carpentier et al provided a coarse proof of the equivalence between the NCP and the NSOCP in [32], and it has also been outlined in other works such as [7]. Given this context, we herein identify the constraint reactions $\boldsymbol{\lambda}$ and the constraint velocities $\mathbf{v}^+$ as the primal and dual variables, respectively, of the dual FD problem. We will also find it useful to decompose the objective function of (16) into the individual terms

$$f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T \mathbf{D}\,\mathbf{x} + \mathbf{x}^T \mathbf{v}_f \ , \tag{17a}$$
$$f_{\text{DS}}(\mathbf{x}) := \mathbf{x}^T \boldsymbol{\Gamma}(\mathbf{v}^+(\mathbf{x})) \ , \tag{17b}$$
$$f_{\text{NCP}}(\mathbf{x}) := f(\mathbf{x}) + f_{\text{DS}}(\mathbf{x}) \ . \tag{17c}$$

The total NSOCP objective (17c) thus consists of the purely quadratic objective (17a) as well as the non-linear De Saxcé term (17b). This decomposition will enable a relaxation of the problem that will be discussed next.

C. Problem Relaxations

Observing the decomposition of the NSOCP objective function defined in (17), we may consider the case where the non-linear term (17a) is omitted. Doing so turns out to be equivalent to approximating the original problem with a Cone Complementarity Problem (CCP) in the form of

$\text{CCP}(\mathbf{D}, \mathbf{v}_f, \mathcal{K})$:

$$\textbf{Find } \boldsymbol{\lambda}, \text{ s.t. } \quad \mathcal{K}^* \ni \mathbf{v}^+(\boldsymbol{\lambda}) \perp \boldsymbol{\lambda} \in \mathcal{K} \ . \tag{18}$$

The primary effect of this relaxation is that complementarity is now asserted directly between the constraint velocities and the constraint reactions. Equivalently, a transcription of (18) as an optimization problem results in a convex Second-Order Cone Program (SOCP) which can be stated concisely as

$\text{SOCP}(\mathbf{D}, \mathbf{v}_f, \mathcal{K})$:

$$\textbf{Find } \boldsymbol{\lambda} = \underset{\mathbf{x}\in\mathcal{K}}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{x}^T \mathbf{D}\,\mathbf{x} + \mathbf{x}^T \mathbf{v}_f \tag{19}$$

This alternate formulation of the FD problem can be traced back to the work of Redon et al in [33], Anitescu et al in [34], and Drumwright et al in [35], although not stated as concisely as in the now established form of (19). Undoubtedly, the CCP approach as we know it today, has mainly been popularized by the work of Todorov in [36] in developing the MuJoCo simulator [37]. More information on the CCP and its implications can be found in [12], [17], [38].

However, the CCP formulation as been the source of significant debate among researchers, as discussed in [39], [35]. Due to the mathematical rigor by which the NCP is derived from VI theory, the CCP is widely thought to be an approximation that results in undesirable artifacts such as forces acting at a distance [34], [40], [38]. Conversely, Chaterjee [39] and Drumwright [35], as well as the authors of MuJoCo [41], note that NCP complementarity does not necessarily agree with physical experiments, as the rigid-body model is also fundamentally an approximation of real physical bodies. In any case, the CCP model has proven to be a useful mechanism to realize physics engines with impressive throughput and versatility, such as MuJoCo and project Chrono [42], [43].

D. First-Order Dual Solvers

Broadly speaking, the vast majority of algorithms used in the various popular physics engines and simulators, are *first-order* methods and follow a common pattern. To outline their commonalities, we can coarsely fit them to the template defined in Alg. 1 that presents a generic form of first-order algorithm for solving the dual problem. From a high-level perspective, the major common components of these include:

- $\mathbf{D}, \mathbf{v}_f, \mathcal{K}$: definition of the dual problem
- $f_{BLAS}(\cdot)$: operator to represent the linear system solver[4]

[4]BLAS: Basic Linear Algebra Subprograms

Algorithm 1 Generic First-Order Dual Solver

Require: $\boldsymbol{\lambda}^0$, $\mathbf{D}$, $\mathbf{v}_f$, $\mathcal{K}$, $f_{BLAS}(\cdot)$ $\mathcal{P}^{\mathcal{K}}(\cdot)$, $f_{stop}(\cdot)$

for $i = 1$ to N **do**

$\mathbf{s}^i \leftarrow \boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda}^{i-1}))$ ▷ estimate nonlinearity

$\hat{\mathbf{v}}^i \leftarrow \mathbf{D}\,\boldsymbol{\lambda}^{i-1} + \mathbf{v}_f + \mathbf{s}^i$ ▷ compute gradient

$\boldsymbol{\lambda}_0^i \leftarrow f_{BLAS}(\mathbf{D}\,,\ -\hat{\mathbf{v}}^i)$ ▷ perform descent step

$\boldsymbol{\lambda}^i \leftarrow \mathcal{P}^{\mathcal{K}}(\boldsymbol{\lambda}_0^i)$ ▷ project to feasible set

$\boldsymbol{\lambda}^* \leftarrow f_{stop}(\boldsymbol{\lambda}^i)$ ▷ check termination criteria

end for

return $\boldsymbol{\lambda}^*$

- $\mathcal{P}^{\mathcal{K}}(\cdot)$: operator to project iterates to the feasible set $\mathcal{K}$
- $f_{stop}(\cdot)$: operator to represent the *termination criteria*
- $\boldsymbol{\lambda}^0$: (optional) initial guess to *warmstart* the solution

The generic BLAS operator $f_{BLAS}(\cdot)$ expresses the method used to approximately perform the inversion of the Delassus matrix $\mathbf{D}$. It often is the case that this is not realized in a literal sense, but rather, it represents a factorization of $\mathbf{D}$, e.g. Cholesky or LU decompositions, that facilitate computing an *unconstrained solution* $\boldsymbol{\lambda}_0^i$ at each iteration.

In every iteration, the projection operator $\mathcal{P}^{\mathcal{K}}(\cdot)$, a.k.a *projector*, projects the unconstrained solution $\boldsymbol{\lambda}_0^i$ onto the feasible set $\mathcal{K}$, yielding a solution iterate $\boldsymbol{\lambda}^i$ that is feasible according to $\mathcal{K}$. Such an operation is common in optimization algorithms that deal with inequality constraints. It therefore attempts to ensure that each iterate $\boldsymbol{\lambda}^{i+1}$, and therefore also the final solution $\boldsymbol{\lambda}^*$, satisfy the inequalities as well as other conditions that define $\mathcal{K}$. $\mathcal{P}^{\mathcal{K}}(\cdot)$ is often one of the most important components of algorithms that follow the template of Alg. 1. As we will see in Sec. VI-C, choosing how to realize it depends on the contact model used, and thus plays a crucial role in solving the dual problem (15) or (18).

Each algorithm necessarily defines its own specialized termination operator $f_{stop}(\cdot)$ that is closely tied to the numerical scheme used. Common elements among various dual solvers include the use of absolute and/or relative numerical error between sequential solution iterates. But it does not, and sometimes cannot, be based solely on such criteria. In addition, algorithms must also check for constraint satisfaction, i.e. whether solution iterates lie within the feasible set, and by how much they violate the respective constraints. This ensures that solvers will converge only when both the numerical error as well as constraint violation are within desired bounds.

Lastly, the use of an initial guess $\boldsymbol{\lambda}^0$ is a common trait of many efficient numeric solvers. In most cases it significantly speeds up convergence to a solution, however, choosing how to construct one can prove challenging [44], [45], [38]. If the contact configuration has not changed, i.e. the dimensionality of the constraint forces is the same, the most common approach is to use the solution of the previous time-step. Otherwise, initialization to zero is necessary, and is the default behavior when $\boldsymbol{\lambda}^0$ is unspecified. A more robust approach would be to compute a constraint-invariant initial guess based on the current state of the system [44], [45].

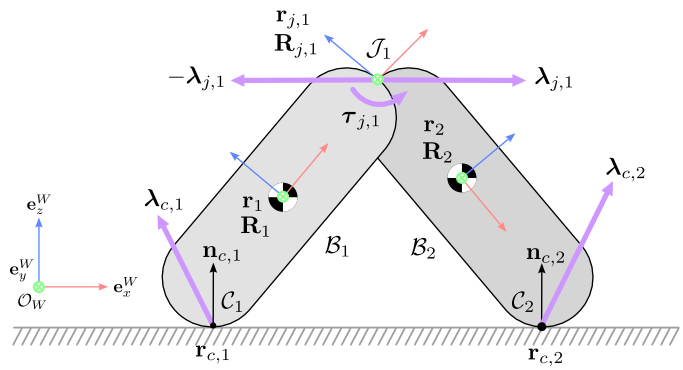

Fig. 2. Free-body diagram of a simple constrained system with: (1) $n_b = 2$ rigid bodies $\mathcal{B}_1$ and $\mathcal{B}_2$, (2) $n_j = 1$ joints $\mathcal{J}_1$ with revolute constraints, and (3) $n_c = 2$ contacts $\mathcal{C}_1$ and $\mathcal{C}_2$ between the bodies and the ground.

IV. Modeling Constrained Systems

Until this point we have been able to state the FD problem in rather generic terms, without being specific about what is involved in the transcription of the NCP (15). This section introduces the elements of physical modeling that will provide the foundation from which we will later build the NCP concretely in Sec. V. Specifically, we will detail the maximal-coordinate CRBD formulation and the modeling of constraints, i.e. joints, limits and contacts.

For this purpose, we introduce some additional notation to denote the *index sets* of the system's DoFs and constraints [23]. Consider a constrained rigid-body system, such as the one depicted in Fig. 2. It can be defined as the collection of bodies $\mathcal{B}_i\,, i \in \{1, \ldots, n_b\}$, joints $\mathcal{J}_j\,, j \in \{1, \ldots, n_j\}$, limits $\mathcal{L}_l\,, l \in \{1, \ldots, n_l\}$ and contacts $\mathcal{C}_k\,, k \in \{1, \ldots, n_c\}$. The symbols $\mathcal{B}_i$, $\mathcal{J}_j$, $\mathcal{L}_l$, $\mathcal{C}_k$ are used to refer abstractly to each element; implying its dimensionality, associated quantities, and relevant local coordinate frames.

Moreover, to remove ambiguity from representing quantities in different Cartesian frames of reference we will employ the following conventions: Global coordinates are defined in the world frame W. Given two local reference frames A and B, moving w.r.t W, the local relative position of B w.r.t A expressed in B is denoted as ${}_B\mathbf{r}_{AB}$, and the relative orientation from B to A as $\mathbf{R}_{AB}$. For absolute quantities, i.e. those expressed in global coordinates, the W subscript is omitted for brevity, e.g. ${}_W\mathbf{r}_{WA} \equiv \mathbf{r}_A$ and $\mathbf{R}_{WA} \equiv \mathbf{R}_A$. Thus, Cartesian coordinate transformations are expressed in the form $\mathbf{r}_B = \mathbf{r}_A + \mathbf{r}_{AB} = \mathbf{r}_A + \mathbf{R}_A\,{}_A\mathbf{r}_{AB} = \mathbf{r}_A + \mathbf{R}_A\,\mathbf{R}_{AB}\,{}_B\mathbf{r}_{AB}$.

A. Rigid Bodies

Each rigid body $\mathcal{B}_i$, using a body-fixed coordinate frame located at its respective Center-of-Mass (CoM), is parameterized by the following collection of intrinsic and extrinsic properties:

- mass m_i
- moment-of-inertia ${}_i\mathbf{I}_i$
- surface geometry $\mathcal{G}_{\mathcal{B},i}$
- CoM position $\mathbf{r}_i \in \mathbb{R}^3$
- CoM orientation $\mathbf{R}_i \in \mathrm{SO}(3)$
- CoM linear velocity $\mathbf{v}_i \in \mathbb{R}^3$

- CoM angular velocity $\boldsymbol{\omega}_i \in \mathbb{R}^3$

Each body is intrinsically characterized by its inertial and geometric properties. The former comprise its mass m_i and moment-of-inertia ${}_i\mathbf{I}_i$ expressed in local body-fixed coordinates. The latter is represented by the parameterized set $\mathcal{G}_{\mathcal{B},i}$[5] containing all points belonging to its exterior surface geometry.

The body's extrinsic property is its mutable state, comprising its Cartesian pose and twist defined by the aforementioned CoM-centric properties $\mathbf{r}_i, \mathbf{R}_i, \mathbf{v}_i, \boldsymbol{\omega}_i$. In addition, we may option for a more concise parameterization of the body's orientation such as quaternions. Specifically, using Hamiltonian unit quaternions $\mathbf{q}_i \in \mathbb{H}$, we can express the rotation matrix as $\mathbf{R}_i = \mathbf{R}(\mathbf{q}_i)$. This choice does not affect the derivation of the dynamics in later sections; it only impacts the representation of configuration-dependent and configuration-level constraints. The pose and twist vectors of $\mathcal{B}_i$ are respectively defined as

$$\mathbf{s}_i = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{q}_i \end{bmatrix} \in \mathbb{S}\,, \;\; \mathbf{u}_i = \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \in \mathbb{R}^6\,, \tag{20}$$

where the compound set $\mathbb{S} := \mathbb{R}^3 \times \mathbb{H}$ is constructed using the *Cartesian product*[6] of the sets of positions and quaternions. $\mathbb{S}$ in this case is merely a quaternion-based parameterization of Cartesian poses, which, are technically elements of $\mathrm{SE}(3)$.

B. Joints

Each joint $\mathcal{J}_j$ introduces a set of bilateral (i.e. equality) constraints that restricts the motion of the bodies it acts upon. These can either take the form of unary constraints acting on a single body, anchoring it to the world, or binary constraints coupling the motion between a body pair. By convention, we designate base B_j and follower F_j coordinate frames for each $\mathcal{J}_j$. For unary joints $B_j \equiv W$ and F_j is coincident with the body-fixed frame of the body. For binary joints, both B_j and F_j are coincident with the body-fixed CoM frames of the corresponding bodies. With the aforementioned conventions, a universal parameterization of joints consists of the:

- relative position w.r.t the base body ${}_B\mathbf{x}_{Bj} \in \mathbb{R}^3$
- relative position w.r.t the follower body ${}_F\mathbf{x}_{Fj} \in \mathbb{R}^3$
- frame axes $\mathbf{X}_j \in \mathrm{SO}(3)$
- selection matrix $\mathbf{S}_j \in \mathbb{R}^{6\times 6}$
- constraint dimensions $m_j \in [1, 6] \subset \mathbb{N}_+$
- DoF dimensions $d_j = 6 - m_j$
- position $\mathbf{r}_j \in \mathbb{R}^3$
- coordinate frame $\mathbf{R}_j \in \mathrm{SO}(3)$
- generalized configuration $\mathbf{q}_j \in \mathbb{R}^{d_j}$

In a maximal-coordinate setting, the intrinsic parameters of a joint are the constraint dimensions m_j, the relative positions ${}_B\mathbf{x}_{Bj}$, ${}_B\mathbf{x}_{Bj}$, the frame axes $\mathbf{X}_j$, and the selection matrix $\mathbf{S}_j$. m_j is effectively the number of constraint equations introduced, and together with the $\mathbf{X}_j$ and $\mathbf{S}_j$ determine the kinematic DoFs it encodes, i.e. prismatic, revolute, spherical, etc. All of the aforementioned quantities are essentially constants that fully specify its parameterization. Thus, all other quantities such as the absolute position $\mathbf{r}_j$, orientation $\mathbf{R}_j$, and generalized configuration $\mathbf{q}_j$ are derived from the joint's parameters and the configurations of the associated bodies.

According to its kinematic type, each joint introduces a set of m_j configuration-level implicit equations in the form of

$$\mathbf{f}_j(\mathbf{s}_{B_j}, \mathbf{s}_{F_j}) = 0, \tag{21}$$

where $\mathbf{f}_j : \mathbb{S} \times \mathbb{S} \to \mathbb{R}^{m_j}$, and $\mathbf{s}_{B_j}, \mathbf{s}_{F_j}$ are the poses of the base and follower bodies, respectively. It will be necessary to also express the constraints at velocity-level in Pfaffian form as

$$\dot{\mathbf{f}}_j(\mathbf{s}_{B_j}, \mathbf{s}_{F_j}, \mathbf{u}_{B_j}, \mathbf{u}_{F_j}) = 0. \tag{22}$$

Taking the first-order derivative w.r.t time of (21) is referred to as *index reduction* in the context of DAEs [46], and allows us to express the constraints as functions of the body twists. Moreover, (22) also admits an interpretation that is fundamental to formulating constrained dynamics: it defines the so-called *constraint-space velocities* $\mathbf{v}_j = \dot{\mathbf{f}}_j(\mathbf{s}_{B_j}, \mathbf{s}_{F_j}, \mathbf{u}_{B_j}, \mathbf{u}_{F_j})$, which are closely related to the concept of *virtual displacements* [47].

The constraints are enforced by corresponding generalized forces that are represented by a vector of Lagrange multipliers $\boldsymbol{\lambda}_j \in \mathbb{R}^{m_j}$ that, according to the principle of d'Alembert, should conserve the energy of the system. The generalized forces acting along the DoFs of the joint are defined as the vector $\boldsymbol{\tau}_j \in \mathbb{R}^{d_j}$. Crucially, both $\boldsymbol{\lambda}_j$ and $\boldsymbol{\tau}_j$ are quantities defined in the local coordinates of the joint frame defined by $\mathbf{R}_j$. However, in order to define the dynamics of a joint, we need to express the forces and torques, i.e. *wrenches*, enacted upon the bodies. This is the principle function of the selection matrix $\mathbf{S}_j$ and the frame axes $\mathbf{X}_j$. Specifically, the former maps the constraint and actuation generalized force vectors to 6D wrenches as

$${}_j\mathbf{w}_j = \mathbf{S}_j \begin{bmatrix} \boldsymbol{\lambda}_j \\ \boldsymbol{\tau}_j \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{c,j} & \mathbf{S}_{a,j} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_j \\ \boldsymbol{\tau}_j \end{bmatrix}\,, \tag{23}$$

$\mathbf{S}_{c,j} \in \mathbb{R}^{6\times m_j}$ and $\mathbf{S}_{a,j} \in \mathbb{R}^{6\times d_j}$ are the component selection matrices that respectively map the generalized forces along the constraint and DoF dimensions. If the joint is passive then $\boldsymbol{\tau}_j$ is always zero, $\mathbf{S}_j \equiv \mathbf{S}_{c,j}$ and (23) effectively reduces to

$${}_j\mathbf{w}_j = \mathbf{S}_j\, \boldsymbol{\lambda}_j\,. \tag{24}$$

The base-follow convention defines the joint wrench as acting on the follower F_j by the base B_j at position $\mathbf{r}_j$. To compute the body-wise wrenches about their respective CoMs we must first retrieve the joint wrench, acting at $\mathbf{r}_j$ and expressed in world coordinates, using the joint's axes and frame matrices:

$$\mathbf{w}_j = \bar{\mathbf{R}}_j\, \bar{\mathbf{X}}_j\, {}_j\mathbf{w}_j\,. \tag{25}$$

Finally, in order to express the wrench effected upon each body, screw transformation matrices must be applied. These are quantities that transform 6D wrenches and twists from one point and frame of application to another. In the context of this work, we only need to consider screw transformations of wrenches, so that we can express the effect of joint wrench $\mathbf{w}_j$

[5] Parametric geometry sets can take the form of primitive geometric shapes such as boxes, spheres etc, which are parameterized by their respective constants (width, height, length and radius, respectively), or more explicit forms such as triangle-based meshes which are collections of vertices.

[6] A Cartesian product of two sets A and B, is defined as the set of all ordered pairs formed from their elements, i.e. $A \times B := \{(a, b) \,|\, a \in A\,,\; b \in B\}$. Effectively $A \times B$ is a mechanism through which we can construct compound sets/spaces over multiple scalar fields, vector spaces etc.

when acting on body $\mathcal{B}_i$, that may correspond to the base B_j or follower F_j. Referring to this wrench as $\mathbf{w}_{i,j}$, and denoting the skew-symmetric operator as $[\cdot]_\times$, we can express it as

$$\mathbf{w}_{i,j} = \mathbf{W}_{i,j}(\mathbf{r}_j, \mathbf{r}_i)\, \mathbf{w}_j \tag{26a}$$

$$\mathbf{W}_{i,j}(\mathbf{r}_j, \mathbf{r}_i) = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_3 \\ [\mathbf{r}_i - \mathbf{r}_j]_\times & \mathbb{I}_3 \end{bmatrix} \ . \tag{26b}$$

C. Limits

Each joint limit $\mathcal{L}_l$ introduces an additional unilateral constraint that is defined by the kinematic limits of the associated joint $\mathcal{J}_j$, that may represent mechanical end-stops or other restrictions to the motion along the admissible DoFs. Each $\mathcal{L}_l$ is thus explicitly dependent on the parameterization of the corresponding $\mathcal{J}_j$, but in addition requires the specification of:

- the DoF selection vector $\mathbf{s}_l \in \mathbb{R}^6$
- the DoF configuration $q_l \in \mathbb{R}$
- the minimum DoF limit $q_l^{min} \in \mathbb{R}$
- the maximum DoF limit $q_l^{max} \in \mathbb{R}$

Fundamentally, each joint with prescribed configuration limits defines a set implicit inequalities in the form of

$$\mathbf{g}_j(\mathbf{s}_{B_j}, \mathbf{s}_{F_j}) \geq 0 \ , \tag{27}$$

where $\mathbf{g}_j : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}^{m_{j,l}}$ and $m_{j,l} \leq 2\, d_j$, because they may be imposed on either lower and/or upper orthant of each joint DoF. However, only one side may ever be active at any point in time, so the dimensionality of limit constraints effectively reduces to $m_{j,l} \leq d_j$. To understand why we must consider how limits can become active in the first place. Given the instantaneous joint DoF configuration $\mathbf{q}_j \in \mathbb{R}^{m_j}$ and lower/upper DoF limits denoted as $\mathbf{q}_j^{min}, \mathbf{q}_j^{max} \in \mathbb{R}^{m_j}$, we can define the respective *joint-limit gap functions*

$$\mathbf{g}_j^{min}(\mathbf{q}_j) = \mathbf{q}_j - \mathbf{q}_j^{min} \tag{28a}$$

$$\mathbf{g}_j^{max}(\mathbf{q}_j) = \mathbf{q}_j^{max} - \mathbf{q}_j \ . \tag{28b}$$

If either $g_{j,i}^{min}(q_{j,i}) \leq 0$ or $g_{j,i}^{max}(q_{j,i}) \leq 0$, then the lower, or respectively upper, limit of the joint has been reached and thus the corresponding limit constraint becomes active. Thus for every joint DoF coordinate $i \in [1, d_j]$ of $\mathcal{J}_j$ where a limit is active, a single limit entity $\mathcal{L}_l$ is defined with $q_l := q_{j,i}$, $q_l^{min} := q_{j,i}^{min}$ and $q_l^{max} := q_{j,i}^{max}$. Moreover, employing an index mapping in the form of $g_l^{min}(q_l) := g_{j,i}^{min}(q_{j,i})$ and $g_l^{max}(q_l) := g_{j,i}^{max}(q_{j,i})$, we can define the unilateral constraint introduced by each $\mathcal{L}_l$ as the implicit inequality

$$g_l(q_l) := \begin{cases} g_l^{min}(q_l) & , \ g_l^{min}(q_l) \leq g_l^{max}(q_l) \\ g_l^{max}(q_l) & , \ g_l^{max}(q_l) \leq g_l^{min}(q_l) \end{cases} \ . \tag{29}$$

Note that (29) does not actually express a piecewise continuous function, rather it just expresses the fact that the constraint corresponds to the gap function of only one of the lower/upper limits at any point in time.

The corresponding constraint reaction of $\mathcal{L}_l$ is represented by the Lagrange multiplier $\lambda_l \geq 0$, and together with (29), define the complementarity conditions

$$g_l(q_l) \geq 0 \ \perp \ \lambda_l \geq 0 \ . \tag{30}$$

By slightly abusing notation, we can denote the constraint-space velocity of $\mathcal{L}_l$ as $\mathrm{v}_l(\dot{q}_l) := \dot{f}_l(\mathbf{s}_{B_j}, \mathbf{s}_{F_j}, \mathbf{u}_{B_j}, \mathbf{u}_{F_j})$, which enables us to express the velocity-level formulation of (30) as

$$\mathrm{v}_l(\dot{q}_l) \geq 0 \ \perp \ \lambda_l \geq 0 \ . \tag{31}$$

Lastly, we must express the wrenches enacting on the bodies associated with $\mathcal{L}_l$. Fortunately, this works out to be exceptionally straightforward since $\mathcal{L}_l$ is by definition acting along only a single DoF of the corresponding joint $\mathcal{J}_j$. Invoking once more the index remapping used to define q_l, the DoF selection vector of each $\mathcal{L}_l$ is exactly the i-th column of the joint's DoF selection matrix $\mathbf{S}_{a,j}$, i.e. $\mathbf{s}_l := (\mathbf{S}_{j,a})_i$. Thus the wrench applied by an active joint limit, acting at and about the joint position $\mathbf{r}_j$ and expressed in world coordinates, is

$$\mathbf{w}_l = \mathbf{R}_j\, \mathbf{X}_j\, \mathbf{s}_l\, \lambda_l \ , \tag{32}$$

D. Contacts

Each discrete contact $\mathcal{C}_k$, introduces a set of unilateral (i.e. inequality) constraints that act to simultaneously: (a) prevent inter-penetration between it's associated bodies when they collide, and (b) define the material interactions between them in the form of *friction* and *restitution*. Contact constraints are therefore more complicated than joints and limits, as they act on both the kinematics and dynamics of the system in a way that depends on the material properties of the bodies; not just their geometry and inertial properties. Similarly to joints, they may also be unary or binary, depending on whether a collision occurs between a body and the world or between bodies. Like limits, they introduce set-valued force laws, i.e. constraints on the contact reactions themselves.

In order to specify the constraints contributed by each contact element $\mathcal{C}_k$, let's first define the intrinsic quantities that parameterize it. Namely, these are the:

- position $\mathbf{r}_{c,k} \in \mathbb{R}^3$
- normal vector $\mathbf{n}_{c,k} \in \mathbb{R}^3, \ \|\mathbf{n}_{c,k}\|_2 = 1$
- gap distance $d_{c,k} \in \mathbb{R}$
- coefficient of friction $\mu_k \in \mathbb{R}_+$
- coefficient of restitution $\epsilon_k \in [0, 1]$

These quantities are thus the minimal amount of information required to define a model of contacts and their corresponding constraint sets. We will now detail how we can define these constraints precisely and the physical phenomena that they model, so that we can incorporate them into our CRBD model.

***Collisions**:* Although often used synonymously, *collisions* and *contacts* correspond to different aspects of the interactions between bodies. For real physical bodies, each collision defines a continuum of body contact distributed over potentially multiple surfaces. However, for simulating multi-body systems, and especially those with rigid bodies, the most straight-forward way to model such interactions is using *discrete contacts*, i.e. geometrically discretizing these surfaces as collections of points. Therefore, a collision occurring between a pair of bodies, characterizes the geometry of the interaction, and is the progenitor of the set of discrete contacts that we use to represent it. If the colliding bodies have convex geometries,

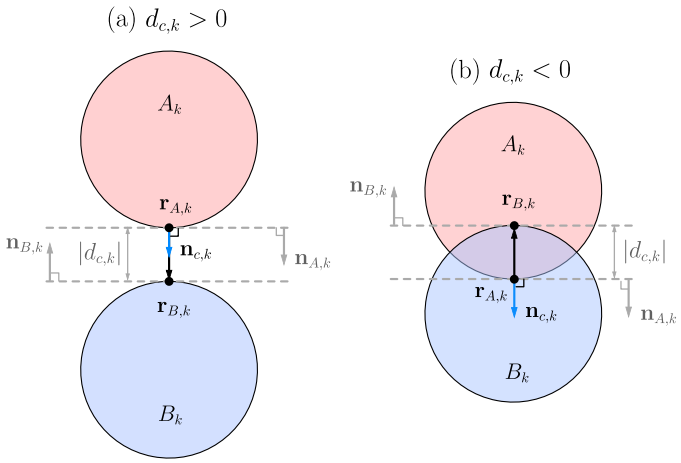

Fig. 3. Binary collisions between bodies A_k and B_k for arbitrary contact with index k. In all configurations, the gap-function (33) can be used to compute the signed distance $d_{c,k}\,\mathbf{n}_{c,k}$ (black arrows) of B_k w.r.t. A_k, where $\mathbf{n}_{c,k}$ is the normal vector (blue arrows) w.r.t the plane that is tangent to the two contacting points of the bodies. Choosing A_k as the reference body is a merely a matter of convention. When $d_{c,k} > 0$ (case (a)), then there is no interpenetration, and the distance is interpreted as the distance between the nearest points between the bodies. Conversely, when $d_{c,k} \leq 0$ (case (b)), interpenetration occurs, and $d_{c,k}$ becomes the penetration depth measured between the most interpenetrating points.

then a single contact can represent the collision, otherwise multiple may be required. Moreover, particularly challenging yet important, are the cases of multiple co-planar contacts, e.g. a box on a plane, as the geometry is indeed convex, yet multiple contacts are required to represent the interaction. Within this context, we will describe how collisions between bodies define the geometric properties of contacts.

A crucial aspect of defining the contact position $\mathbf{r}_{c,k}$ is that it does not necessarily coincide with the exact position on each body where the collision occurs. Consider the case depicted in Fig.3b, where the two (convex) bodies A_k and B_k are interpenetrating. If we define the contact normal to lie along the line connecting the inner-most penetrating points on the bodies, then the contact should be defined using both points. Instead, we arbitrarily designate one of these as *the contact position* $\mathbf{r}_{c,k} := \mathbf{r}_{A,k}$ and determine the location of the other using the surface normal $\mathbf{n}_{c,k}$ and penetration depth $d_{c,k}$ as $\mathbf{r}_{B,k} = \mathbf{r}_{c,k} + d_{c,k}\,\mathbf{n}_{c,k}$. The relative position between bodies A_k and B_k is used to define the so-called *gap function*

$$\mathbf{g}_k(\mathbf{r}_{A,k}, \mathbf{r}_{B,k}) = \mathbf{r}_{B,k} - \mathbf{r}_{A,k} = d_{c,k}\,\mathbf{n}_{c,k} \tag{33}$$

of each contact, where $\mathbf{g}_k : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$. Essentially, the relative position expressed via (33) corresponds to a vector-valued signed-distance function, and defines the configuration-level constraint function of $\mathcal{C}_k$.

Contact Modes: Noting how the penetration depth $d_{c,k}$ can take zero, positive and negative values, we can use this to define the so-called *contact mode* m_k, a discrete valued scalar that summarizes the state of $\mathcal{C}_k$. When $d_{c,k} > 0$, we define the contact to be *open* ($m_k =$ Open) and $d_{c,k}$ is interpreted to be the minimum Euclidean distance between the two bodies. Conversely, when $d_{c,k} \leq 0$ the contact is defined as *closed* ($m_k =$ Stick $\vee$ Slip) and $d_{c,k}$ becomes the penetration depth. Therefore, for convex geometries, $\mathbf{r}_{A,k}, \mathbf{r}_{B,k}$ are either the nearest points between the two bodies or the most-penetrating, respectively, and are always functions of the corresponding body poses $\mathbf{s}_{A,k}$ $\mathbf{s}_{B,k}$. When the geometries are not convex, or multiple contacts per body pair are admissible, the situation is slightly more complicated[7].

Contact Space: In this work we only consider isotropic 3D Coulomb-type friction, where the contact constraint reaction and respective constraint-space velocity are respectively 3D vectors, i.e. $\boldsymbol{\lambda}_k$, $\mathbf{v}_{c,k} \in \mathbb{R}^3$. Although more advanced models of friction exist, such as those that can also incorporate torsional effects [48], [41], we consider the simpler case a sufficient baseline for the purposes of this evaluation.

Contact Frames: The space in which all contact-related quantities can most easily be represented is that of a local contact-specific coordinate frame. Employing such a frame allows for the re-parameterization of the contact reaction $\boldsymbol{\lambda}_k$ into a form that delineates between the normal N and tangent T components. By applying certain conventions, the normal vector $\mathbf{n}_{c,k}$, can be used to represent the corresponding tangent plane as the tangent and orthogonal (i.e. bi-normal) vectors $\mathbf{t}_{c,k}, \mathbf{o}_{c,k} \in \mathbb{R}^3$, respectively. The triplet $\mathbf{n}_{c,k}, \mathbf{t}_{c,k}, \mathbf{o}_{c,k}$ thus enables the construction of a rotation matrix $\mathbf{R}_k \in \mathrm{SO}(3)$ to represent the coordinate frame of $\mathcal{C}_k$. This construction is expressed by the contact-frame function $\mathbf{C} : \mathbb{R}^3 \to \mathrm{SO}(3)$ as

$$\mathbf{R}_k = \mathbf{C}(\mathbf{n}_{c,k}) =: \begin{bmatrix} \mathbf{t}_{c,k} & \mathbf{o}_{c,k} & \mathbf{n}_{c,k} \end{bmatrix}, \tag{34}$$

In this work we place the normal along the contact-local Z-axis, and the XY-plane as the tangent space. This results in parameterizing the contact reaction in local coordinates as

$$\boldsymbol{\lambda}_k := \begin{bmatrix} \boldsymbol{\lambda}_{T,k} \\ \lambda_{N,k} \end{bmatrix}, \quad \boldsymbol{\lambda}_{T,k} := \begin{bmatrix} \lambda_{t,k} \\ \lambda_{o,k} \end{bmatrix}, \tag{35}$$

where $\lambda_{N,k}$, $\lambda_{t,k}$, and $\lambda_{o,k}$ are respectively the normal, tangent, and orthogonal components. Thus $\boldsymbol{\lambda}_k$, and its normal-tangent decomposition, are expressed in global coordinates as

$${}_W\boldsymbol{\lambda}_k := \mathbf{R}_k\,\boldsymbol{\lambda}_k \tag{36a}$$

$${}_W\boldsymbol{\lambda}_{T,k} := \mathbf{R}_k \begin{bmatrix} \boldsymbol{\lambda}_{T,k} \\ 0 \end{bmatrix} \tag{36b}$$

$${}_W\boldsymbol{\lambda}_{N,k} := \lambda_{N,k}\,\mathbf{n}_{c,k}\,. \tag{36c}$$

Contact Wrench: Similarly to joints, the wrench applied to contacting bodies is also computed using the screw transformations described in (26). However, given the aforementioned assumption that contact reactions contain only linear 3D terms, only the left block-column of (26b) is needed. Thus, the contact wrench applied to body $\mathcal{B}_i$ by $\mathcal{C}_k$ is therefore

$$\mathbf{w}_{c,k,i} = \mathbf{W}_{c,k,i}(\mathbf{r}_k, \mathbf{r}_i)\,{}_W\boldsymbol{\lambda}_k \tag{37a}$$

[7]However, an easy way to work around this is by realizing that contacts are defined only once they occur (i.e. become closed) in the first-place, so then the consideration of $\mathbf{r}_{A,k}, \mathbf{r}_{B,k}$, only matters locally while the contact persists and until they open.

$$\mathbf{W}_{c,k,i}(\mathbf{r}_k, \mathbf{r}_i) = \begin{bmatrix} \mathbb{I}_3 \\ [\mathbf{r}_i - \mathbf{r}_k]_\times \end{bmatrix} \tag{37b}$$

Contact Motion: The duality of twists and wrenches allows us to also express the linear velocity of the point of contact on $\mathcal{B}_i$ as an affine function of the respective body twist in the form of

$$_W\mathbf{v}_{c,k,i} = \mathbf{W}_{c,k,i}(\mathbf{r}_k, \mathbf{r}_i)^T \, \mathbf{u}_i \tag{38}$$

However, the quantity we are actually interested in is the relative velocity computed at the point of contact $\mathbf{r}_k$, between the contacting bodies A_k and B_k. Referring to this simply as *the contact velocity*, (38)[8] is used to define it precisely using the relations

$$_W\mathbf{v}_{c,k,A} = \mathbf{W}_{c,k,A}(\mathbf{r}_k, \mathbf{r}_{A,k})^T \, \mathbf{u}_{A,k} \tag{39a}$$

$$_W\mathbf{v}_{c,k,B} = \mathbf{W}_{c,k,B}(\mathbf{r}_k, \mathbf{r}_{B,k})^T \, \mathbf{u}_{B,k} \tag{39b}$$

$$_W\mathbf{v}_{c,k} = {}_W\mathbf{v}_{c,k,A} - {}_W\mathbf{v}_{c,k,B} \tag{39c}$$

$$\mathbf{v}_{c,k} = \mathbf{R}_k^T \, {}_W\mathbf{v}_{c,k} \ . \tag{39d}$$

Signorini Conditions: Also known as the *unilateral contact hypothesis*, the Signorini conditions assert that contact reactions preventing interpenetration between bodies must act only while the contact is closed, i.e. when the gap constraint is active. Using the definition of the gap function from (33), this statement is formalized as the complementarity problem

$$g_{c,N,k} := \mathbf{n}_{c,k}^T \, \mathbf{g}_k(\mathbf{r}_{A,k}, \mathbf{r}_{B,k}) \geq 0 \tag{40a}$$

$$g_{c,N,k} \geq 0 \ \perp \ \lambda_{N,k} \geq 0 \ . \tag{40b}$$

However, in order to introduce these complementarity conditions to the formulation of the system dynamics, it is often necessary to state them at the velocity and acceleration level, [39], [23]. Denoting the first and second derivative of (40a) w.r.t time as $\mathrm{v}_{c,N,k} := \mathrm{v}_{c,N,k}(\mathbf{s}_{A,k}, \mathbf{s}_{B,k}, \mathbf{u}_{A,k}, \mathbf{u}_{B,k})$, and $a_{c,N,k} := a_{c,N,k}(\mathbf{s}_{A,k}, \mathbf{s}_{B,k}, \mathbf{u}_{A,k}, \mathbf{u}_{B,k}, \dot{\mathbf{u}}_{A,k}, \dot{\mathbf{u}}_{B,k})$, they become

$$\mathrm{v}_{c,N,k} \geq 0 \ \perp \ \lambda_{N,k} \geq 0 \ , \tag{41a}$$

$$a_{c,N,k} \geq 0 \ \perp \ \lambda_{N,k} \geq 0 \ . \tag{41b}$$

Coulomb Friction: Under the assumption of isotropic dry friction, both the static and kinetic frictional reactions are determined by a single friction coefficient μ_k. Thus, the two fundamental assumptions of the Coulomb friction model are: (a) static friction equaling $\mu_k \, \lambda_{N,k}$ must be overcome in order for motion to occur, and (b) kinetic friction takes the form of a tangential reaction that opposes non-zero contact velocity i.e.

$$\boldsymbol{\lambda}_{T,k} = -\mu_k \, \lambda_{N,k} \, \frac{\mathbf{v}_{c,T,k}}{\|\mathbf{v}_{c,T,k}\|} \ , \ \|\mathbf{v}_{c,T,k}\| > 0 \ . \tag{42}$$

These two conditions combined define the set inclusion

$$\boldsymbol{\lambda}_{T,k} \in \mathcal{D}(\mu_k \, \lambda_{N,k}) \ , \tag{43}$$

where $\mathcal{D}(\mu_k \, \lambda_{N,k})$ is a disk of radius $\mu_k \, \lambda_{N,k}$. Moreover, the requirement that contact reaction along the normal direction is only repulsive can be written as the set inclusion

$$\lambda_{N,k} \in \mathbb{R}_+ \ . \tag{44}$$

Fig. 5 depicts these *set-valued force laws* as non-linear functions w.r.t the gap function $g_{c,N,k}$ and contact velocity $\mathbf{v}_{c,k}$. Combining (43) and (44) forms the Coulomb friction cone

$$\boldsymbol{\lambda}_k \in \mathcal{K}_{\mu_k} \ , \tag{45a}$$

$$\mathcal{K}_{\mu_k} := \{\boldsymbol{\lambda} \in \mathbb{R}^3 \mid \|\boldsymbol{\lambda}_{T,k}\|_2 \leq \mu_k \, \lambda_{N,k} \ , \ \lambda_{N,k} \geq 0\} \ . \tag{45b}$$

The 3D Coulomb friction cone $\mathcal{K}_{\mu_k}$ is a symmetric second-order cone often referred to as a Lorentz cone [49], whose aperture is determined by the respective friction coefficient μ_k, i.e. $\theta_k = 2 \, tan^{-1}(\mu_k)$. $\mathcal{K}_{\mu_k}$ is a closed and convex set that bounds the corresponding 3D contact reaction to lie both within and and on its surface boundary $\partial\mathcal{K}_{\mu_k}$.

An important property of Lorentz cones is that they are self dual, i.e. they are equal to their conjugate sets. This property allows us to also define the *dual cone* of $\mathcal{K}_\mu$ as $\mathcal{K}_{\mu_k}^* := \mathcal{K}_{\mu_k^{-1}}$ and the *polar cone* as $\mathcal{K}_{\mu_k}^o := -\mathcal{K}_{\mu_k}^* = -\mathcal{K}_{\mu_k^{-1}}$. Their geometric relationships are depicted in Fig. 4. As $\mathcal{K}_\mu$ is defined in the space of contact reactions, the dual and polar cones are defined in the dual space of the contact velocities.

The Disjunctive Model: The Coulomb friction cone model can be combined with the Signorini conditions to define the contact mode m_k in terms of the contact reaction $\boldsymbol{\lambda}_k$ and velocity $\mathbf{v}_{c,k}$. This results in the so-called *disjunctive*

[8] An alternative derivation can be found, if instead, we form the first-order time-derivative of the gap function (33).

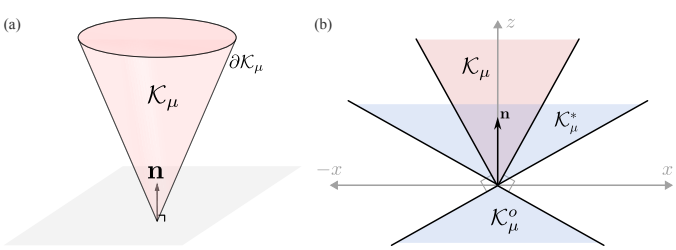

Fig. 4. The Coulomb friction cone $\mathcal{K}_\mu$ (a) and the relationships between the former to its dual cone $\mathcal{K}_\mu^*$ and polar cone $\mathcal{K}_\mu^o$ (b). As conjugates of $\mathcal{K}_\mu$, the dual and polar cones are those whose elements make obtuse angles with each $\boldsymbol{\lambda} \in \mathcal{K}_\mu$, i.e. $\mathbf{x} \in \mathcal{K}_\mu^*$, $\mathbf{x}^T \boldsymbol{\lambda} \geq 0$, and $\mathbf{x} \in \mathcal{K}_\mu^o$, $\mathbf{x}^T \boldsymbol{\lambda} \leq 0$.

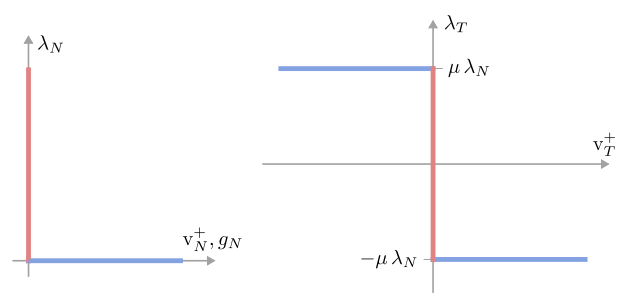

Fig. 5. The set-valued force laws along the normal and tangent directions. The red segments represent the sets of admissible values that the contact reactions λ_N, λ_T can take while the respective contact velocities are zero. The blue segments represent the admissible constant values when non-zero velocities are present in the respective directions.

formulation of the Signorini-Coulomb model:

$$m_k = \begin{cases} \text{Open} & , v_{c,N,k} > 0 \wedge \boldsymbol{\lambda}_k = 0 \\ \text{Stick} & , \|\mathbf{v}_{c,T,k}\|_2 = 0 \wedge \boldsymbol{\lambda}_k \in \mathcal{K}_{\mu_k} \\ \text{Slip} & , \|\mathbf{v}_{c,T,k}\|_2 > 0 \wedge \boldsymbol{\lambda}_k \in \partial\mathcal{K}_{\mu_k} \end{cases} . \tag{46}$$

The first case is trivial, and is equivalent to using $g_{c,N,k} > 0$. The second case corresponds to the friction-cone constraint to when the contact force lies strictly in the interior of the cone. The third corresponds to the case where the contact force lies solely on the boundary $\partial\mathcal{K}_{\mu_k}$ of the cone.

Maximal Dissipation: The Maximum Dissipation Principle (MDP), most often attributed to J.J. Moreau [50], asserts that for each active contact $\mathcal{C}_k$, the tangential contact reaction must be maximally dissipative, i.e. it should maximize mechanical power loss along the tangent plane. For given normal reaction $\lambda_{N,k} \in \mathbb{R}_+$ and tangential velocity $\mathbf{v}_{c,T,k} \in \mathbb{R}^2$, the tangential contact reaction according to the MDP is the solution to

$$\boldsymbol{\lambda}_{T,k} = \underset{\mathbf{x}}{\text{argmax}} \quad -\mathbf{x}^T \mathbf{v}_{c,T,k} \quad \text{s.t.} \quad \|\mathbf{x}\|_2 \leq \mu_k \lambda_{N,k} . \tag{47}$$

The MDP is a formulation of frictional contact as an optimization problem that generalizes (42) as it is also applicable to the case of multiple contacts. It is straight-forward to verify that in the single-contact case, (42) is in fact maximally dissipative. The power product would not obtain its maximum value unless $\boldsymbol{\lambda}_{T,k}$ is colinear and opposite to $\mathbf{v}_{c,T,k}$. Moreover, (47) is admissible even when $\lambda_{N,k} = 0$ and/or $\mathbf{v}_{c,T,k} = 0$, thus encompassing all contact modes. Furthermore, it motivates the bi-potential method [30] which can be used derive the general NCP. Since (42) is maximally dissipative, we can form the power-loss product to yield

$$\boldsymbol{\lambda}_{T,k}^T \mathbf{v}_{c,T,k} = -\mu_k \lambda_{N,k} \|\mathbf{v}_{c,T,k}\|_2 . \tag{48}$$

By moving all terms to the left-hand side, and trivially adding the product of normal components, we can form the expression

$$\lambda_{N,k} v_{c,N,k} + \boldsymbol{\lambda}_{T,k}^T \mathbf{v}_{c,T,k} + \mu_k \lambda_{N,k} \|\mathbf{v}_{c,T,k}\|_2 = 0 , \tag{49}$$

which like the MDP, can be verified to hold in all contact modes. By introducing the *De Saxcé correction operator*

$$\boldsymbol{\Gamma}_{\mu_k}(\mathbf{v}_{c,k}) := \begin{bmatrix} 0 \\ 0 \\ \mu_k \|\mathbf{v}_{c,T,k}\|_2 \end{bmatrix} , \tag{50}$$

and defining the *augmented contact velocity*

$$\hat{\mathbf{v}}_{c,k} := \mathbf{v}_{c,k} + \boldsymbol{\Gamma}_{\mu_k}(\mathbf{v}_{c,T,k}) , \tag{51}$$

(49) can be restated as the complementarity condition

$$\hat{\mathbf{v}}_{c,k} \in \mathcal{K}^*_{\mu_k} \perp \boldsymbol{\lambda}_k \in \mathcal{K}_{\mu_k} , \tag{52}$$

where the complementarity now requires that $\hat{\mathbf{v}}_{c,k}$ lie within the dual friction cone $\mathcal{K}^*_{\mu_k}$. This result is of paramount importance because it demonstrates that (52) must hold in order for $\boldsymbol{\lambda}_k$ to be maximally dissipative, as well as to ensure that the resulting contact velocity lie solely on the tangential plane when the contact is closed. The main difficulty with this formulation is that it makes the complementarity condition *non-associative*, since it is not stated directly in the contact velocities and reactions.

Newtonian Impacts: To model restitution effects, *Newton's model of impacts* can be incorporated in the velocity-level Signorini conditions (41a). Considering the impact event to occur instantaneously at time t_i, the twists of bodies A_k and B_k experience a discontinuity that is denoted by their pre/post-event values $\mathbf{u}^{-/+}_{A,k}$ and $\mathbf{u}^{-/+}_{B,k}$, respectively. According to Newton's model, the pre/post-event contact velocities $\mathbf{v}^-_{c,N,k} := \mathbf{v}_{c,N,k}(\mathbf{u}^-_{A,k}, \mathbf{u}^-_{B,k})$ and $\mathbf{v}^+_{c,N,k} := \mathbf{v}_{c,N,k}(\mathbf{u}^+_{A,k}, \mathbf{u}^+_{B,k})$ in the direction of the contact normal are coupled by the *restitution equations*

$$\mathbf{v}^+_{c,N,k} = -\epsilon_k \, \mathbf{v}^-_{c,N,k} , \tag{53}$$

with ϵ_k being the coefficient of restitution. In effect, the restitution equations (53) serve to augment the Signorini conditions by biasing the contact velocity along the normal direction. Thus the impact-augmented Newton-Signorini model becomes

$$0 \leq \mathbf{v}^+_{c,N,k} + \epsilon_k \, \mathbf{v}^-_{c,N,k} \perp \boldsymbol{\lambda}_k \in \mathcal{K}_{\mu_k} . \tag{54}$$

Lastly, combining (54) with (52), we can form the impact-augmented Newton-Coulomb-Signorini NCP in the form of

$$\hat{\mathbf{v}}_{c,k} + \epsilon_k \, \mathbf{v}^-_{c,N,k} \in \mathcal{K}^*_{\mu_k} \perp \boldsymbol{\lambda}_k \in \mathcal{K}_{\mu_k} . \tag{55}$$

E. Constrained Rigid-Body Dynamics

With the elements described in the previous sections in hand, we can now derive Equations-of-Motion (EoM) of the CRBD formulation in maximal-coordinates. We begin by formulating, for each body $\mathcal{B}_i$, the *Newton-Euler* equations

$$\mathbf{M}_i \, \dot{\mathbf{u}}_i + \begin{bmatrix} -m_i \, \mathbf{g} \\ [\boldsymbol{\omega}_i]_\times \, \mathbf{I}_i \, \boldsymbol{\omega}_i \end{bmatrix} = \mathbf{w}_{total,i} , \tag{56}$$

where $\mathbf{g}$ is the gravity acceleration vector, and $\mathbf{M}_i \in \mathbb{R}^{6\times 6}$ is the body-wise generalized mass matrix, which is defined as

$$\mathbf{M}_i = \begin{bmatrix} m_i \, \mathbb{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{i\,i}\mathbf{I}_i \, \mathbf{R}_i^T \end{bmatrix} , \tag{57}$$

and $\mathbf{I}_i$ is the body's MoI matrix expressed in world coordinates, which can be expressed using the body's orientation and the local MoI using the *parallel axis theorem* as

$$\mathbf{I}_i = \mathbf{R}_{i\,i}\mathbf{I}_i \, \mathbf{R}_i^T . \tag{58}$$

To further simplify the expression for (56), we can denote the non-linear terms of gravity and Coriolis wrenches as

$$\mathbf{w}_{gc,i} := \begin{bmatrix} m_i \, \mathbf{g} \\ -[\boldsymbol{\omega}_i]_\times \, \mathbf{I}_i \, \boldsymbol{\omega}_i \end{bmatrix} , \tag{59}$$

and arrive at a concise expression of the body-wise EoM

$$\mathbf{M}_i \, \dot{\mathbf{u}}_i = \mathbf{w}_{gc,i} + \mathbf{w}_{total,i} . \tag{60}$$

Stacking (60) of each body, we can form the system EoM

$$\mathbf{M} \, \dot{\mathbf{u}} = \mathbf{w}_{gc} + \mathbf{w}_{total} , \tag{61}$$

where $\mathbf{M}$ is the block-diagonal generalized mass matrix first introduced in Sec. II, and $\mathbf{w}_{gc}$ and $\mathbf{w}_{total}$ are the stacked gravity-Coriolis and total wrenches, respectively.

In order to introduce constraints explicitly, we must decompose $\mathbf{w}_{total}$ into the individual terms which account for all forces applied to the bodies and that are not part of $\mathbf{w}_{gc}$:

$$\mathbf{w}_{total} = \mathbf{w}_e + \mathbf{w}_a + \mathbf{w}_J + \mathbf{w}_L + \mathbf{w}_C \tag{62}$$

We have also included the external wrenches $\mathbf{w}_e$ to account for purely external factors such as external disturbances and forces we may wish to apply ad-hoc. Next we will express $\mathbf{w}_a$, $\mathbf{w}_J$, $\mathbf{w}_L$ and $\mathbf{w}_C$ as functions of the respective constraint reactions and generalized actuation forces. To this end, we can apply the screw transforms (26) to (25) and (32) combined with (23), as well as apply (36a) to (37). Essentially, doing so renders the Jacobian matrices $\mathbf{J}_a$, $\mathbf{J}_J$, $\mathbf{J}_L$, and $\mathbf{J}_C$, corresponding to actuated joint DoFs, joints, limits, and contacts, respectively.

Proceeding element-wise over each $\mathcal{J}_j$, $\mathcal{L}_l$, and $\mathcal{C}_k$ present in the system, we can define the matrix-blocks of the corresponding Jacobian for each associated body. For each joint $\mathcal{J}_j$ attached to body $\mathcal{B}_i$, the corresponding matrix-block of $\mathbf{J}_J$ is

$$\mathbf{J}_{J,i,j}^T = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_3 \\ [\mathbf{r}_i - \mathbf{r}_j]_\times & \mathbb{I}_3 \end{bmatrix} \mathbf{R}_j \, \mathbf{X}_j \, \mathbf{S}_{c,j} \ , \tag{63}$$

and if the joint is actuated, the matrix-block of $\mathbf{J}_a$ is

$$\mathbf{J}_{a,i,j}^T = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_3 \\ [\mathbf{r}_i - \mathbf{r}_j]_\times & \mathbb{I}_3 \end{bmatrix} \mathbf{R}_j \, \mathbf{X}_j \, \mathbf{S}_{a,j} \ . \tag{64}$$

For each limit $\mathcal{L}_l$ the matrix-block of $\mathbf{J}_L$ is

$$\mathbf{J}_{L,i,j,l}^T = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_3 \\ [\mathbf{r}_i - \mathbf{r}_j]_\times & \mathbb{I}_3 \end{bmatrix} \mathbf{R}_j \, \mathbf{X}_j \, \mathbf{s}_l \ , \tag{65}$$

and for each contact $\mathcal{C}_k$ the matrix-block of $\mathbf{J}_C$ is

$$\mathbf{J}_{c,i,k}^T = \begin{bmatrix} \mathbb{I}_3 \\ [\mathbf{r}_i - \mathbf{r}_k]_\times \end{bmatrix} \mathbf{R}_k \ . \tag{66}$$

We will also denote the total constraint Jacobian as

$$\mathbf{J} := \begin{bmatrix} \mathbf{J}_J^T & \mathbf{J}_L^T & \mathbf{J}_C^T \end{bmatrix}^T \ , \tag{67}$$

Next we will define stacked vectors over all joint DoF actuation forces and constraint reaction Lagrange multipliers corresponding to each constraint group, in the form of

$$\boldsymbol{\tau}_a := \begin{bmatrix} \boldsymbol{\tau}_{a,1} \\ \vdots \\ \boldsymbol{\tau}_{a,n_j} \end{bmatrix} \ , \quad \boldsymbol{\lambda}_J := \begin{bmatrix} \boldsymbol{\lambda}_{j,1} \\ \vdots \\ \boldsymbol{\lambda}_{j,n_j} \end{bmatrix} \ ,$$

$$\boldsymbol{\lambda}_L := \begin{bmatrix} \boldsymbol{\lambda}_{l,1} \\ \vdots \\ \boldsymbol{\lambda}_{l,n_l} \end{bmatrix} \ , \quad \boldsymbol{\lambda}_C := \begin{bmatrix} \boldsymbol{\lambda}_{c,1} \\ \vdots \\ \boldsymbol{\lambda}_{c,n_c} \end{bmatrix} \ , \tag{68}$$

as well as the vector of all constraint reaction multipliers

$$\boldsymbol{\lambda} := \begin{bmatrix} \boldsymbol{\lambda}_J^T & \boldsymbol{\lambda}_L^T & \boldsymbol{\lambda}_C^T \end{bmatrix}^T \ . \tag{69}$$

Having defined the Jacobian matrices as well as the DoF actuation and constraint reaction vectors, we can now concisely express the total wrenches applied to the system of bodies as

$$\mathbf{w}_{total} = \mathbf{w}_e + \mathbf{J}_a^T \boldsymbol{\tau}_a + \mathbf{J}_J^T \boldsymbol{\lambda}_J + \mathbf{J}_L^T \boldsymbol{\lambda}_L + \mathbf{J}_C^T \boldsymbol{\lambda}_C \tag{70}$$

$$= \mathbf{w}_e + \mathbf{J}^T \boldsymbol{\lambda} \ . \tag{71}$$

Moreover, by re-arranging some terms, we may define the vector of non-linear generalized force terms

$$\mathbf{h} := \mathbf{w}_{gc} + \mathbf{w}_e + \mathbf{J}_a^T \boldsymbol{\tau}_a \ , \tag{72}$$

and thus arrive at the acceleration-level EoM

$$\mathbf{M} \, \dot{\mathbf{u}} = \mathbf{h} + \mathbf{J}^T \boldsymbol{\lambda} \ . \tag{73}$$

In (73) we recognize the smooth (i.e. continuous) version of the system dynamics. Using the derivation described in Sec. II-B, it is now quite straightforward to derive the time-stepping version of the system EoM and render

$$\mathbf{M} \left(\mathbf{u}^+ - \mathbf{u}^- \right) = \Delta t \, \mathbf{h} + \mathbf{J}^T \boldsymbol{\lambda} \ . \tag{74}$$

V. Constructing The Dual Problem

This section describes the construction of the dual FD NCP (15) using the individual modeling elements described in Sec. IV. First, we will detail how the dual NCP can be constructed to incorporate all the elements described in Sec. IV-D for modeling contacts with restitutive impacts and friction. Second, we will describe two augmentations that will enable us realize constraint stabilization and constraint softening.

A. Origins

The dual FD NCP transcribed as the NSOCP (16) was first introduced by Cadoux [51]. To the best of our knowledge, this is the first instance of employing the *bi-potential method* of De Saxcé [30] to incorporate the non-linear term in the optimization objective. This augmentation is crucial to the NSOCP representing a complete model of contact dynamics, but as the De Saxcé operator $\boldsymbol{\Gamma}_\mu(\cdot)$ is highly non-linear, it makes the problem NP-hard. The former was first introduced by De Saxcé et al in [52] and subsequently elaborated upon in [30]. The latter provided a complete theoretical treatment that both justified its validity as well as established its connection to the concept of *superpotentials* of J.J. Moreau [50], that have long been a key part of the non-smooth dynamics foundations. We recommend the seminal text of Glocker et al [23] for more insight on this topic. The NSOCP as we know it today, as well as the methods to approximate it, were introduced by Acary et al in [31] where it was derived from the primal FD problem.

B. Problem Definition

We begin by describing the construction of the three quantities defining the dual NCP, namely, the Delassus matrix $\mathbf{D}$, the free-velocity $\mathbf{v}_f$ and the feasible set $\mathcal{K}$ of admissible constraint reactions. However, regarding the Delassus matrix in particular, we do not actually need to state anything further. Having defined the generalized mass matrix $\mathbf{M}$ and the constraint Jacobian $\mathbf{J}$ in Sec. IV, we may simply apply (11) to compute it directly. Thus, at this stage we can focus solely on $\mathbf{v}_f$ and $\mathcal{K}$ as these require some work to define.

The principle mechanism we must employ is projection onto the constraint-space. Exploiting the duality induced by the constraint Jacobian $\mathbf{J}$, we can express the pre/post-event constraint velocities using the relations

$$\mathbf{v}^{-/+} = \mathbf{J}\,\mathbf{u}^{-/+} \quad . \tag{75}$$

In addition, we will re-arrange the time-stepping EoM (74) to express the post-event generalized velocity $\mathbf{u}^+$ as a function of the constraint reactions $\boldsymbol{\lambda}$ as

$$\mathbf{u}^+(\boldsymbol{\lambda}) = \mathbf{u}^- + \Delta t\,\mathbf{M}^{-1}\,\mathbf{h} + \mathbf{M}^{-1}\,\mathbf{J}^T\,\boldsymbol{\lambda} \quad . \tag{76}$$

Combining (75) and (76) yields the vector of post-event constraint-space velocity (13) introduced in Sec. III-B:

$$\mathbf{v}^+(\boldsymbol{\lambda}) = \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f \quad .$$

In Sec. IV-D we described the Coulomb-Signorini-Newton model (55) which asserts complementarity conditions on the impact-augmented and De Saxcé-corrected constraint-space velocity of each contact $\mathcal{C}_k$. We can generalize this statement to include all constraints by forming the NCP

$$\mathcal{K}^* \ni \mathbf{v}^+(\boldsymbol{\lambda}) + \boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) + \mathbf{E}\,\mathbf{v}^- \perp \boldsymbol{\lambda} \in \mathcal{K} \quad , \tag{77}$$

where, and with slight abuse of notation, we can denote the system-level De Saxcé operator using its contact-wise definition (50) to only act along the contact constraint dimensions

$$\boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) := \begin{bmatrix} \mathbf{0}_{n_{jd}} \\ \mathbf{0}_{n_l} \\ \boldsymbol{\Gamma}_{\boldsymbol{\mu}}(\mathbf{v}_C^+(\boldsymbol{\lambda})) \end{bmatrix} \quad , \tag{78a}$$

$$\boldsymbol{\Gamma}_{\boldsymbol{\mu}}(\mathbf{v}_C^+(\boldsymbol{\lambda})) := \begin{bmatrix} \boldsymbol{\Gamma}_{\mu_1}(\mathbf{v}_{c,1}^+(\boldsymbol{\lambda})) \\ \vdots \\ \boldsymbol{\Gamma}_{\mu_{n_c}}(\mathbf{v}_{c,n_c}^+(\boldsymbol{\lambda})) \end{bmatrix} \quad . \tag{78b}$$

Moreover, $\mathbf{E} \in \mathbb{R}^{n_d \times n_d}$ is the *restitution matrix*, constructed using the coefficients of restitution of all contacts

$$\mathbf{E} := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{E}_C \end{bmatrix} \tag{79a}$$

$$\mathbf{E}_C := \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \epsilon_{c,1} \end{bmatrix} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \epsilon_{c,n_c} \end{bmatrix} \end{bmatrix} \quad . \tag{79b}$$

As it turns out, as the restitution relations are only considered along the contact normal directions[9], the impact augmentation term can be absorbed into the free velocity $\mathbf{v}_f$ as a component of the bias velocity, i.e. $\mathbf{v}^* = \mathbf{E}\,\mathbf{v}^-$, which admits the redefinition $\mathbf{v}_f \leftarrow \mathbf{v}_f + \mathbf{E}\,\mathbf{v}^-$. Effectively, this redefinition reduces (77) to the standard definition of the NCP (15). We provide a brief proof of this property in Appendix. A-B. Thus the free-velocity is simply constructed using

$$\mathbf{v}_f = \mathbf{J}\,\left(\mathbf{u}^- + \Delta t\,\mathbf{M}^{-1}\mathbf{h}\right) + \mathbf{E}\,\mathbf{v}^- \tag{80}$$

In practice, however, can avoid constructing $\mathbf{E}$ and performing all the unnecessary matrix-multiplications with zeros by simply appending $\epsilon_{c,k}\,\mathrm{v}_k^-$ to the row of $\mathbf{v}_f$ corresponding to the normal direction of each respective contact $\mathcal{C}_k$. Thus the only actual computational cost lies in computing $\mathbf{v}^- = \mathbf{J}\,\mathbf{u}^-$, which we do so anyway in (80).

Next, we will define the feasible set $\mathcal{K}$ of admissible constraint reactions. Luckily this is rather straight-forward to do, as it works out to simply be the *Cartesian product* over the element-wise feasibility sets, which is often denoted as

$$\mathcal{K} = \prod_{i}^{n_j+n_l+n_c} \mathcal{K}_i \ . \tag{81}$$

A crucial property of this composition is that, assuming each $\mathcal{K}_i$ is a proper cone, then so too is the total feasible set $\mathcal{K}$ [49] . For each joint, limit and contact, we must invoke the appropriate set-valued force law to define the corresponding component set $\mathcal{K}_i$ of the aforedescribed product. Thus, for each joint $\mathcal{J}_j$, $\mathcal{K}_j = \mathbb{R}^{m_j}$, i.e. the constraint reactions are unconstrained and therefore span the entire constraint-space of the joint. For each limit $\mathcal{L}_i$, $\mathcal{K}_l = \mathbb{R}_+$, i.e. the force is constrained to the positive half-space, a.k.a. the nonnegative orthant, of the corresponding joint DoF. Lastly, for each contact $\mathcal{C}_k$, $\mathcal{K}_k = \mathcal{K}_{\mu_k}$, i.e. the set is the Coulomb friction cone $\mathcal{K}_{\mu_k}$ whose aperture is determined by the contact-specific friction coefficient μ_k. This results in the composite cone

$$\mathcal{K} := \mathbb{R}^{n_{jd}} \times \mathbb{R}^{n_l} \times \mathcal{K}_{\boldsymbol{\mu}} \ , \tag{82}$$

where $\boldsymbol{\mu} := [\mu_1, \dots, \mu_{n_c}]^T$ denotes the vector of all friction coefficients, and $\mathcal{K}_{\boldsymbol{\mu}}$ is the composite cone formed over the Coulomb friction cones of all active contacts.

Lastly, we will form the *total configuration-level constraint function* of the system. To construct it, we can use the definitions of the configuration-level implicit functions of all elements $\mathcal{J}_j$, $\mathcal{L}_l$ and $\mathcal{C}_k$, using (21), (29) and (40a), respectively. Stacking all individual elements of each constraint group, we first form the respective group-wise constraint functions

$$\mathbf{f}_J(\mathbf{s}) = \begin{bmatrix} \mathbf{f}_1(\mathbf{s}) \\ \vdots \\ \mathbf{f}_{n_j}(\mathbf{s}) \end{bmatrix}, \mathbf{f}_L(\mathbf{s}) = \begin{bmatrix} g_{l,1}(\mathbf{s}) \\ \vdots \\ g_{l,n_l}(\mathbf{s}) \end{bmatrix}, \mathbf{f}_C(\mathbf{s}) = \begin{bmatrix} g_{n,1}(\mathbf{s}) \\ \vdots \\ g_{n,n_c}(\mathbf{s}) \end{bmatrix} . \tag{83}$$

The total configuration-level implicit constraint function results from simply combining the components in (83) to form

$$\mathbf{f}(\mathbf{s}) = \begin{bmatrix} \mathbf{f}_J(\mathbf{s}) \\ \mathbf{f}_L(\mathbf{s}) \\ \mathbf{f}_C(\mathbf{s}) \end{bmatrix} = 0 \ . \tag{84}$$

Although not directly part of the definition of dual FD NCP (15), we will use it to realize the augmentations in the sections that follow, as well as later in the definition of performance metrics in Sec. VII-D.

[9]It is technically feasible to include restitution effects along the contact tangential directions, as is described in [53], [54] for deriving the PROX/SORprox algorithm. However, we are not aware of any materials whose physical interactions would justify the construction of such a model.

C. Constraint Stabilization

Without employing a numerical integration scheme that can, at each discrete time-step, update the state of the system in a way that satisfies all constraints exactly, it is inevitable that numerical drift will lead to violations of the configuration-level constraints (21), (27) and (40). Such is the case when using simpler time-stepping integrators such as semi-implicit Euler (10b) or Moreau's midpoint scheme [25]. In addition, performing CD at discrete time-steps unavoidably leads to some interpenetration of the bodies, which, is directly dependent on the step size Δt and the potential velocities of the bodies.

These simulation artifacts are also exacerbated by the imprecision that may be exhibited by dual solvers. Essentially, we can understand why by realizing that the solutions to the NCP/CCP they render are computed with the goal of reducing the constraint *velocity* to zero along the bilateral and unilateral constraint dimensions (i.e. not considering frictional components). This means that dual solvers have no direct impact on the constraints at configuration-level, and any under/over estimation of the constraint reactions will lead to further numerical drift, compounding that of the integration scheme.

However, configuration-level constraint violation can be alleviated by employing so-called *constraint stabilization* techniques, such as that of Baumgarter [55]. In this work we followed an approach to realizing Baumgarter-like constraint stabilization based on the work of Preclik et al [56] as well as the recent technical guide [17]. The end result will be an augmentation of the problem that follows the same procedure described previously in Sec. V-B for modeling frictional contacts with restitutive impacts. However, for illustrative purposes, it will be easier to explain the approach by first considering the simpler case where only bilateral joint constraints are present in the system.

The first step involves using (75) and (76) to express the velocity-level bilateral constraints as a function of the constraint reactions $\boldsymbol{\lambda}$ via the definition of the post-event constraint velocity. Doing so yields the constraints

$$\mathbf{v}^{+}(\boldsymbol{\lambda}) = \mathbf{J}\,\mathbf{u}^{+}(\boldsymbol{\lambda}) = 0 \tag{85}$$

We will consider an augmentation of (85) that introduces control feedback terms that are functions of the instantaneous constraint violation. The latter is represented by the vector of constraint residuals computed from the configuration-level constraint functions (83) of the joints, in the form of

$$\mathbf{r}_J = \mathbf{f}_J(\mathbf{s}) \ . \tag{86}$$

Thus the *stabilized* bilateral constraints can be stated as

$$\mathbf{v}^{+}(\boldsymbol{\lambda}) + \epsilon\,\boldsymbol{\lambda} = -\frac{v}{\Delta t}\,\mathbf{r}_J \quad , \tag{87}$$

where ϵ and v are often referred to as the Constraint Force Mixing (CFM) constant and the Error Reduction Parameter (ERP), respectively. To the best of our knowledge, these names are attributed to the ODE physics engine [57]. We refer interested readers to [17], [41] for a more in depth explanation and interpretation of these parameters. In effect, they serve to respectively soften and dampen the constraint-space dynamics of the system. As our objective is to introduce such a constraint-stabilization scheme to the dual problem without compromising its structure as an NCP, we will necessarily omit the CFM term $\epsilon\,\boldsymbol{\lambda}$ for now. We do so because CFM introduces compliance to the system which effectively softens the constraints, and is the topic of Sec. V-D.

The next step, therefore, is to introduce ERP-type constraint stabilization to the NCP. Similar to how incorporating restitutive impacts augments the complementarity conditions (52) to yield (55), the constraint stabilization of (87) can be introduced via an additional bias in the form of

$$\mathcal{K}^{*} \ni \hat{\mathbf{v}}(\boldsymbol{\lambda}) + \mathbf{v}_B \perp \boldsymbol{\lambda} \in \mathcal{K} \quad , \tag{88}$$

where $\mathbf{v}_B$ is the *constraint stabilization bias velocity* that we need to construct. Moreover, as stabilization acts exclusively on the unilateral components of limit and contact constraints (i.e. contact normal directions), we can once more exploit the invariance of the De Saxcé operator w.r.t the bias $\mathbf{v}_B$ and include it as an additional component of the total bias to the free-velocity, i.e. $\mathbf{v}^{*} = \mathbf{E}\,\mathbf{v}^{-} + \mathbf{v}_B$. This ensures that the NCP retains the structure of the un-augmented problem.

Now we can proceed to construct $\mathbf{v}_B$. First of all, we will specify distinct ERPs $\alpha_j, \beta_l, \gamma_k \in \mathbb{R}_{+}$ for each joint $\mathcal{J}_j$, limit $\mathcal{L}_l$ and contact $\mathcal{C}_k$ constraint subset, respectively. For each joint $\mathcal{J}_j$, the bilateral constraint residual and stabilization bias are

$$\mathbf{r}_{J,j} = \mathbf{f}_{J,j}(\mathbf{s}) \tag{89a}$$

$$\mathbf{v}_{B,J,j} = \frac{\alpha_j}{\Delta t}\,\mathbf{r}_{J,j} \quad . \tag{89b}$$

For each limit $\mathcal{L}_l$, as the constraints are unilateral, we will need to additionally constrain the residuals to act in the orthant that satisfies the complementarity conditions. Thus, the corresponding residual and bias velocity in this case are

$$r_l = g_l(q_l) \tag{90a}$$

$$\mathrm{v}_{B,L,l} = \frac{\beta}{\Delta t}\,\min(0\,,\ r_l) \quad . \tag{90b}$$

For the unilateral contact constraints of each active contact $\mathcal{C}_k$, we do something similar as for limits. Recalling our conventions for contacts described in Sec.IV-D, body penetration occurs when $d_{c,k} \leq 0$. However, some CD methods and software can render potential collisions, i.e. contacts with positive penetration, in order to allow for preemptive actions. For this reason, Preclik et al [56] proposed a double-sided method which ensures that the contact reaction remains zero if the contact would not close within the time-step. Thus, for each $\mathcal{C}_k$, this approach takes the form of

$$r_k = d_{c,k} + \delta_c \tag{91a}$$

$$\mathrm{v}_{B,c,k} = \gamma_k\,\min\left(0\,,\ \frac{r_{c,k}}{\Delta t}\right) + \max\left(0\,,\ \frac{r_{c,k}}{\Delta t}\right) \quad , \tag{91b}$$

where the constant $\delta_c \in \mathbb{R}$ is the so-called *contact penetration margin*. The purpose of the penetration margin is to either serve as a tolerance of permissible contact penetration ($\delta_c \geq 0$), or to additionally inflate the constraint boundary ($\delta_c < 0$).

The constant scaling factors $\alpha_j/\Delta t$, $\beta_l/\Delta t$, and $\gamma_j/\Delta t$ thus bias the constraint reactions to push the bodies in directions that drive the corresponding residuals to zero, e.g. $\mathbf{r}_{J,j} \to 0$.

Setting $\alpha_j, \beta_l, \gamma_j = 1$, amounts to correcting the residual in a single step. However, one must be careful with using this value, because it can lead to excessive oscillations, as the problem can become very stiff, since the correction behaves equivalently to a linear spring. Some physics engines like NVIDIA PhysX, apply an ad-hoc capping of the this bias, that is referred to as the *maximum depenetration velocity*. In our current implementation we have not yet considered such additional effects, but these may be considered in future work.

Now we can construct the bias velocity vectors for each constraint type by simply stacking the individual contributions of each constraint element using (89), (90) and (91), to form

$$\mathbf{v}_{B,J} = \begin{bmatrix} \mathrm{v}_{B,J,1} \\ \dots \\ \mathrm{v}_{B,J,n_j} \end{bmatrix}, \mathbf{v}_{B,L} = \begin{bmatrix} \mathrm{v}_{B,L,1} \\ \dots \\ \mathrm{v}_{B,L,n_l} \end{bmatrix}, \mathbf{v}_{B,C} = \begin{bmatrix} \mathrm{v}_{B,c,1} \\ \dots \\ \mathrm{v}_{B,c,n_c} \end{bmatrix}, \quad (92)$$

and then combine them to form the total bias velocity

$$\mathbf{v}_B = \begin{bmatrix} \mathbf{v}_{B,J}^T & \mathbf{v}_{B,L}^T & \mathbf{v}_{B,C}^T \end{bmatrix}^T . \quad (93)$$

Finally, the constraint-stabilized free-velocity is computed as

$$\mathbf{v}_f = \mathbf{J} \left(\mathbf{u}^- + \Delta t \, \mathbf{M}^{-1} \mathbf{h} \right) + \mathbf{E} \, \mathbf{v}^- + \mathbf{v}_B . \quad (94)$$

D. Constraint Softening

The dual FD problem poses significant numerical difficulties. First of all, in either NCP (15) or NSOCP (16) forms, the complementarity constraint to be enforced is non-convex and non-smooth, and the De Saxcé operator is highly non-linear. Second, even if were to consider a relaxation of the in the form of the CCP (18) or the SOCP (19), the problem would still not be strictly convex as the Delassus matrix is in general only positive semi-definite i.e. it is rarely positive definite and often ill-conditioned. To address these challenges, we will consider another type of relaxation that regularizes the problem by softening the constraints.

Specifically, we follow the approach introduced by Todorov et al [12] for MuJoCo [58] that introduces a diagonal regularizer $\mathbf{R} \in \mathbb{S}_{++}^{n_d}$ to the Delassus matrix as well as an additional regulation bias velocity $\mathbf{v}_r \in \mathbb{R}^{n_d}$ to the free-velocity vector. However, as the particulars of how MuJoCo realizes this method has evolved significantly since the initial publication of [12], we must clarify that we our implementation is mostly based on that described in the official documentation [41] of MuJoCo. Thus, in this section we will describe how MuJoCo's constraint softening scheme can be applied to both the NCP and CCP formulations of the FD problem. We will provide a brief derivation to clarify how this is possible and we refer readers to [41] for further details.

The constraint softening scheme essentially serves to augment the NSOCP (16) introduced in Sec. III-B to render the Regularized NSOCP (RNSOCP) that takes the form of

$$\begin{aligned} &\text{RNSOCP}(\mathbf{D}\,,\ \mathbf{R}\,,\ \mathbf{v}_f\,,\ \mathbf{v}_r\,,\ \mathcal{K}) : \\ &\textbf{Find } \boldsymbol{\lambda} = \underset{\mathbf{x} \in \mathcal{K}}{\operatorname{argmin}} \ \frac{1}{2} \mathbf{x}^T \left(\mathbf{D} + \mathbf{R} \right) \mathbf{x} \\ &\qquad\qquad + \mathbf{x}^T \left(\mathbf{v}_f + \mathbf{v}_r + \boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) \right) . \end{aligned} \quad (95)$$

Equivalently, we can also state the CCP-type version as the Regularized Second-Order Cone Program (RSOCP)

$$\begin{aligned} &\text{RSOCP}(\mathbf{D}\,,\ \mathbf{R}\,,\ \mathbf{v}_f\,,\ \mathbf{v}_r\,,\ \mathcal{K}) : \\ &\textbf{Find } \boldsymbol{\lambda} = \underset{\mathbf{x} \in \mathcal{K}}{\operatorname{argmin}} \ \frac{1}{2} \mathbf{x}^T \left(\mathbf{D} + \mathbf{R} \right) \mathbf{x} + \mathbf{x}^T \left(\mathbf{v}_f + \mathbf{v}_r \right) \end{aligned} \quad (96)$$

which, is *strictly convex* since the augmented Delassus matrix $\mathbf{D} + \mathbf{R}$ is symmetric and positive definite (PD). Thus, the combined effects of the CCP relaxation and constraint softening yield a dual FD problem that can be solved much more efficiently than the original NCP problem. However, one limitation of this approach is that it does not resolve the issue of ill-conditioning of $\mathbf{D}$, and can in fact lead to instabilities. This phenomena will be demonstrated later in the experiments presented Sec. VIII-D, but we will outline a potential reason for this at the end of this section.

To derive and construct the constraint softening terms we will once again start by assuming only bilateral joint constraints are present in the system. Fundamentally, employing the diagonal regularizer $\mathbf{R}$ amounts to introducing mechanical compliance to each constraint dimension, which is akin to introducing multiple spring-mass-like elements to the system. We can define the potential energy of this generalized "spring" element, as a function of the generalized state using the definition of the equality constraints functions (22), to form

$$U_C(\mathbf{s}, \mathbf{u}^+) = \frac{1}{2} \left\| \dot{\mathbf{f}}_J(\mathbf{s}, \mathbf{u}^+) \right\|_{\mathbf{K}}^2 , \quad (97)$$

where $\mathbf{K} \in \mathbb{S}_{++}^{n_d}$ is a PD stiffness matrix. Note how we have used the velocity-level constraints as opposed to the configuration-level variants (21). This is possible by virtue of index reduction, and is necessary since we require the potential and the resulting force to be functions of the post-event generalized velocities $\mathbf{u}^+$. Taking the partial derivative w.r.t the former yields the generalized force of the spring as

$$\boldsymbol{\tau}_C := -\frac{\partial U_C(\mathbf{s}, \mathbf{u}^+)}{\partial \mathbf{u}^+} = -\mathbf{J}_J^T \, \mathbf{K} \, \dot{\mathbf{f}}_J(\mathbf{s}, \mathbf{u}^+) . \quad (98)$$

Observing the multiplicand of the Jacobian matrix $\mathbf{J}_J$, we can equate this quantity to the force of a Hookean linear spring model in constraint-space, where the residuals play the role of spring deflections. These spring forces are thus exactly the constraint reactions, and thus we can state the equivalence as

$$\boldsymbol{\lambda} = -\mathbf{K} \, \dot{\mathbf{f}}_J(\mathbf{s}, \mathbf{u}^+) \ \Leftrightarrow \ \dot{\mathbf{f}}_J(\mathbf{s}, \mathbf{u}^+) = -\mathbf{R} \, \boldsymbol{\lambda} \ , \quad (99)$$

where now we define the regularizer as $\mathbf{R} := \mathbf{K}^{-1}$ which makes it exactly equivalent to a mechanical compliance. Combining (99) with the time-stepping EoM (74) to form

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_J^T \\ \mathbf{J}_J & -\mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{u}^+ \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \Delta t \, \mathbf{h} + \mathbf{M} \, \mathbf{u}^- \\ -\mathbf{v}^* \end{bmatrix} . \quad (100)$$

Note how we have included the bias velocity $\mathbf{v}^*$ in the RHS of (100). This is to account for the fact that any augmentations considered besides constraint softening would necessarily be included in the definition of the velocity-level constraints, i.e. $\dot{\mathbf{f}}_J(\mathbf{s}, \mathbf{u}^+) := \mathbf{J}_J \, \mathbf{u}^+ - \mathbf{v}^*$. The system (100) demonstrates that the inclusion of the compliance in the joints can render the system solvable with direct methods, otherwise its absence

would result in a zero matrix in-place of $\mathbf{R}$ and thus potentially admit multiple or no solution at all. Moreover, (100) can also result from expressing the KKT conditions of (96), which in the absence of unilateral constraints is identical to (95).

Taking an optimization perspective, we can express the Lagrangian of the original primal FD problem as

$$\mathcal{L}(\mathbf{u}^+, \boldsymbol{\lambda}) := \frac{1}{2}\|\mathbf{u}^+ - \mathbf{u}_f\|_{\mathbf{M}} + \boldsymbol{\lambda}^T \left(\mathbf{J}_J \mathbf{u}^+ + \mathbf{v}^*\right) \quad , \tag{101}$$

and observe the resulting saddle-point problem

$$\min_{\mathbf{u}^+} \max_{\boldsymbol{\lambda}} \mathcal{L}(\mathbf{u}^+, \boldsymbol{\lambda}) - \frac{1}{2} \| \boldsymbol{\lambda} \|^2_{\mathbf{R}} \quad . \tag{102}$$

The additional term in the Lagrangian constitutes what is referred to in the optimization literature as *Tikhonov regularization* [59]. Applying such a regularization is therefore guaranteed to yield a unique minimum-norm solution, which is a significant benefit of constraint softening, since it stabilizes the resulting contact reactions w.r.t successive time-steps.

However, this regularization can result in significant underestimation of the constraint forces, leading to intolerable constraint violation. Physically, this means that joints break, bodies drift away from one another, and contacting bodies exhibit significant interpenetration. Todorov et al [12] proposed a solution to this predicament: construct $\mathbf{v}_r$ as a special form the constraint stabilization described previously in Sec. V-C. In a nutshell, it causes the constraint-space dynamics to behave as a multi-dimensional, decoupled, and critically-damped spring-mass system. Moreover, by carefully crafting a non-linear modulation of the constraint residual, the effective compliance, damping and stiffness can be adapted automatically according to the magnitude of the constraint violation. Proceeding element-wise using index j for each scalar constraint equation, the MuJoCo approach can be outlined as follows:

1) Define shaping functions that modulate the residuals r_j.
2) Use the shaping functions to compute impedance d_j.
3) Use each impedance d_j to compute corresponding compliance c_j, damping b_j and stiffness k_i coefficients.
4) Construct $\mathbf{R}$ and $\mathbf{a}_r$ using the triplets $c_j, b_j, k_i \, \forall j$.

At the core of this approach lies a family of shaping functions, which we'll generically denote as $f_s(x)$, where $f_s : \mathbb{R} \rightarrow \mathbb{R}$. The shaping functions are reflected sigmoids (i.e. symmetric w.r.t the domain) that modulate the residuals r_j. Parameterized by the midpoint value m_j and power factor p_j, they take the form of the linear and non-linear variants

$$\mathrm{f}_{s,L}(x) = x \tag{103a}$$

$$\mathrm{f}_{s,NL}(x) = \begin{cases} \frac{x^{p_j}}{m_j^{1-p_j}} & , x \leq m_j \\ 1 - \frac{(1-x)^{p_j}}{(1-m_j)^{1-p_j}} & , x > m_j \end{cases} \quad . \tag{103b}$$

We can choose between either depending on whether we select $p_j = 1$ to use the linear, or $p_j > 1$ for the nonlinear. Having chosen a variant, the constraint impedance is then computed using the piece-wise continuous pseudo-function

$$x_j = \frac{|r_j - m_j|}{w_j} \tag{104a}$$

$$d_j := \begin{cases} \frac{1}{2}\left(d_{0,j} + d_{w,j}\right) & , d_{0,j} = d_{w,j} \vee w_j \approx 0 \\ d_{w,j} & , x_j \geq 1 \\ d_{0,j} & , x_j \leq 0 \\ d_{0,j} + f_s(x_j)\left(d_{w,j} - d_{0,j}\right) & , \text{otherwise} \end{cases} \quad , \tag{104b}$$

where x_j is a shorthand for the absolute of the scaled and offset residual r_j, w_j is the width parameter that defines the interval $[-w_j, w_j]$ within which the impedance is shaped by r_j, $d_{0,j} = d(0)$ is the minimum impedance when $x_j = 0$, and $d_{w,j} = d(w_j)$ is the maximum impedance when $r_j = w_j$. Note that this operation effectively bounds the residual-to-impedance mapping to the ranges $r_j \in [-w_j, w_j]$ and $d_j \in [d_{0,j}, d_{w,j}]$. Moreover, as we require that both impedance limits $d_{0,j}$ and $d_{w,j}$ (with $d_{0,j} \leq d_{w,j}$) are selected within the $[0, 1]$ range, the computed impedance also lies in the range $d_j \in [0, 1]$. Fig.6 demonstrates examples of the residual-to-impedance mappings for various shaping function parameters. Therefore, given impedance d_j computed as just described, the compliance, damping and stiffness coefficients are respectively

$$c_j = \frac{1 - d_j}{d_j} \ , \ b_j = \frac{2}{d_{w,j}\, T_j} \ , \ k_j = \frac{d_j}{d_{w,j}^2\, T_j^2\, \beta_j^2} \tag{105}$$

where T_j and β_j are the time-constant and the damping ratio, respectively, of the effective mass-spring-damper system of the j-th constraint. The time-constant is essentially the inverse of the natural frequency of the system multiplied by the damping ratio. Finally, the coefficients of the compliance matrix and reference acceleration vector are respectively computed as

$$R_{jj} = c_j\, D_{jj} \ , \quad v_{r,j} = \Delta t \left(b_j\, v_j^- + k_j\, r_j\right) \tag{106}$$

where D_{jj} is the j-th diagonal coefficient of the Delassus matrix $\mathbf{D}$, and v_j is the j-th coefficient of constraint-space velocity vector $\mathbf{v}$. This formulation therefore proceeds element-wise and allows us to select distinct values of the parameter set $\{m, p, w, d_0, d_w, T, \beta\}_j$ for each constraint j. This affords us great flexibility in selecting those that achieve the best results possible, but with the disadvantage of not being automated.

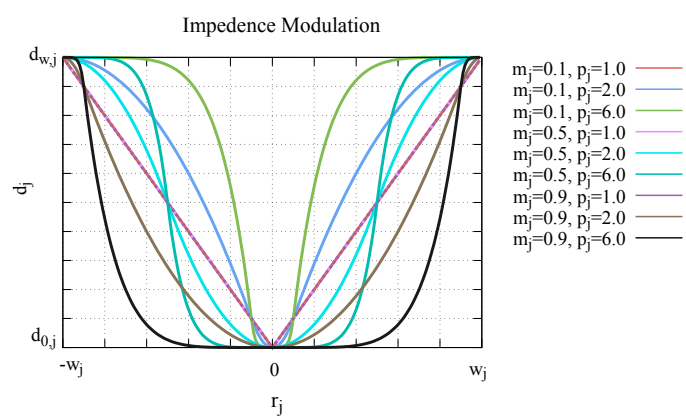

Fig. 6. Examples of impedance shaping using the sigmoidal shaping functions $f_s(x)$, for various settings of the midpoint value m_j and power factor p_j parameters. The 2D plots are agnostic to the absolute impedance values, and are shown only relative to the minimum and maximum impedance defined by $d_{0,j}$ (dmin) and $d_{0,j}$ (dmax) respectively. The X-axis is clipped to the range defined by the width parameter w_j.

The final step involves constructing the regulation velocity $\mathbf{v}_r$. Although in practice we actually add each term directly to each dimension of $\mathbf{v}_f$, for illustrative purposes, we can define this more clearly by expressing the diagonal matrices

$$\mathbf{K}_r = \mathrm{diag}([k_1, \ldots, \ k_{n_d}]^T) \tag{107}$$

$$\mathbf{B}_r = \mathrm{diag}([b_1, \ldots, \ b_{n_d}]^T) \tag{108}$$

$$\mathbf{R} = \mathrm{diag}([c_1 \, D_{11}, \ldots, \ c_{n_d} \, D_{n_d \, n_d}]^T) \quad . \tag{109}$$

The regulation velocity $\mathbf{v}_r$ is thus computed as

$$\mathbf{v}_r = \Delta t \, \left(\mathbf{B}_r \, \mathbf{v}_j^- + \mathbf{K}_r \, \mathbf{r} \right) \quad . \tag{110}$$

Having outlined what MuJoCo-type constraint-softening terms, we can briefly analyze what they actually represent, and why they may lead to instabilities. As the regularized Delassus matrix is positive definite due to the presence of $\mathbf{R}$, it is thus also invertible, admitting the closed-form solution

$$\boldsymbol{\lambda}_0 := (\, \mathbf{D} + \mathbf{R} \,)^{-1} \, (\, \mathbf{v}_f + \mathbf{v}_r \,) \; . \tag{111}$$

Using (111) renders the constraint-space velocity

$$\mathbf{v}_0^+ = \mathbf{D} \, (\mathbf{D} + \mathbf{R})^{-1} \, \mathbf{v}_f + \mathbf{R} \, (\mathbf{D} + \mathbf{R})^{-1} \, \mathbf{v}_r \; , \tag{112}$$

which, reveals the element-wise constraint-space dynamics

$$v_{0,j}^+ = d_j \, (b_j \, v_j^- + k_j \, r_j) + (1 - d_j) \, v_{f,j} \quad . \tag{113}$$

This demonstrates that the system will behave as multiple decoupled spring-mass-damper elements. However, this can lead to significant oscillations if the parameters that determine the non-linear impedance are not chosen carefully. Therefore, although the regularization and regulation of the problem is somewhat automated, it still requires the user to choose the aforedescribed parameters appropriately.

One final clarification that is necessary to make before concluding this part, is that the MuJoCo-like constraint softening method is mostly, but not exactly, compatible with the NSOCP formulation (95). As we described previously in Sec. V-B and Sec. V-D, an augmentation that biases the free-velocity $\mathbf{v}_f$ via $\mathbf{v}^*$ must satisfy the invariance property of the De Saxcé operator. However, as $\mathbf{v}_r$ includes non-zero components in the frictional dimensions due to the damping terms, it violates this requirement. By virtue of approximation, we can proceed to apply it in the same manner as all other augmentations.

VI. Dual Solvers

Given a complete definition, and construction, of the dual FD NCP (15), we now address how it can be solved numerically. This section therefore describes the set of first-order dual FD algorithms that fulfill the template of Alg.1. For brevity, we henceforth refer to them simply as *dual solvers*. Following the taxonomies described in [60], [7], [17], [38], we will summarize how they are related as well as their distinguishing factors. Of course, one may note that their respective derivations can be based on many seemingly different perspectives of the dual FD problem, i.e. *complementarity systems*, *projected dynamics*, *differential inclusions* and *constrained optimization*. However, the underlying theory of *convex analysis* can be used to establish their equivalence as demonstrated by Brogliato [61].

As Moreau first established in his seminal work [25], a unifying view of these first-order methods can be attained via the *augmented Lagrange method* (ALM) [62], and *proximal operators* [63]. We will first define the theoretical framework that justifies the template of Alg. 1 and then proceed to describe the set of computational building-blocks that lead to the construction of concrete dual solvers. Effectively, the solvers resulting from this perspective are those that can be described as variants of the celebrated PGS algorithm, some which can be seen as generalizations and improvements thereof. In addition, we also consider a version of a state-of-the-art solver based on the primal-dual Alternating Direction Method of Multipliers (ADMM) algorithm [64], that was recently proposed by Carpentier et al in [32]. This ADMM method is also derived using ALM and proximal operators and thus also fulfills the template of Alg. 1, further broadening their applicability as the underlying foundation. The complete list of solvers we have evaluated is presented in Tab. I, and includes relevant implementation details.

A. Preliminaries

We begin with a brief review of elements from *convex analysis* and *proximal optimization* that will be required in the continuation. We refer the reader to [65], [63] for details.

The *indicator function* of a closed and convex set $\mathcal{C} \subset \mathbb{R}^n$ evaluated at some point $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\Psi_{\mathcal{C}}(\mathbf{x}) := \begin{cases} 0 & , \ \mathbf{x} \in \mathcal{C} \\ \infty & , \ \mathbf{x} \notin \mathcal{C} \end{cases} \; , \tag{114}$$

and its conjugate is the so-called *support function*

$$\mathcal{S}_{\mathcal{C}}(\mathbf{x}) = \Psi_{\mathcal{C}}^*(\mathbf{x}) := \sup_{\mathbf{y}} \{ \mathbf{y}^T \, \mathbf{x} \, : \, \mathbf{y} \in \mathcal{C} \} \; . \tag{115}$$

It is important to note at this point, that as $\mathcal{C}$ is convex, then so too are $\Psi_{\mathcal{C}}(\mathbf{x})$ and $\mathcal{S}_{\mathcal{C}}(\mathbf{x})$. In general, given a convex function $f : \mathbb{R}^n \to \mathbb{R}$, then its *subdifferential* $\partial f(\mathbf{x})$ is defined as

$$\partial f(\mathbf{x}) := \{ \mathbf{y} \, : \, \mathbf{y}^T \, (\mathbf{z} - \mathbf{x}) \leq (f(\mathbf{z}) - f(\mathbf{x})) \; , \; \forall \mathbf{z} \} \; . \tag{116}$$

Thus given that the indicator and support functions are convex, we can apply (116) to define their respective subdifferentials as

$$\partial \Psi_{\mathcal{C}}(\mathbf{x}) := \{ \mathbf{y} \, : \, \mathbf{y}^T \, (\mathbf{z} - \mathbf{x}) \leq 0 \, , \; \forall \mathbf{z} \in \mathcal{C} \} \quad , \tag{117}$$

$$\partial \mathcal{S}_{\mathcal{C}}(\mathbf{x}) := \{ \mathbf{y} \, : \, \mathbf{y}^T \, \mathbf{z} \leq 0 \, , \; \forall \mathbf{z} \in \partial \, \Psi_{\mathcal{C}}(\mathbf{x}) \} \quad . \tag{118}$$

We now observe that, the aforedefined subdifferentials correspond to the normal and tangent cones of $\mathcal{C}$ at point $\mathbf{x} \in \mathcal{C}$, respectively, i.e. $\mathcal{N}_{\mathcal{C}}(\mathbf{x}) = \partial \, \Psi_{\mathcal{C}}(\mathbf{x})$ and $\mathcal{T}_{\mathcal{C}}(\mathbf{x}) = \partial \, \mathcal{S}_{\mathcal{C}}(\mathbf{x})$. This property establishes the duality (i.e. conjugacy) between the two sets and their elements, in the form of

$$\mathbf{y} \in \mathcal{N}_{\mathcal{C}}(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \mathcal{T}_{\mathcal{C}}(\mathbf{y}) \Leftrightarrow \mathbf{x} \in \mathcal{C} \, , \; \mathbf{x}^T \mathbf{y} = \mathcal{S}_{\mathcal{C}}(\mathbf{y}) \; . \tag{119}$$

Moreover, when the set under consideration is a closed and convex cone $\mathcal{K}$, then in addition, the support function is identical to the indicator function of the dual cone $\mathcal{K}^*$, i.e.

$\mathcal{S}_{\mathcal{K}}(\mathbf{x}) = \Psi_{\mathcal{K}^*}(\mathbf{x})$. This property, together with the duality defined in (119) results in the complementarity property

$$\mathbf{y} \in \mathcal{N}_{\mathcal{K}}(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \mathcal{N}_{\mathcal{K}^*}(\mathbf{y}) \Leftrightarrow \mathcal{K} \ni \mathbf{x} \perp \mathbf{y} \in \mathcal{K}^* \ . \quad (120)$$

Next, let us introduce the *proximal operator*, which, defines a generalized notion of projection. For any weighted metric norm $\|\mathbf{x}\|_{\mathbf{A}} := \sqrt{\mathbf{x}^T \mathbf{A}\, \mathbf{x}}$, where $\mathbf{A} \in \mathbb{S}^n_{++}$, a proximal-point to a convex function $f(\mathbf{x})$ at some $\mathbf{x} \in \mathbb{R}^n$ can be defined as

$$\text{prox}^{\mathbf{A}}_{\mathbf{f}}(\mathbf{x}) := \underset{\mathbf{y}}{\text{argmin}}\ \mathbf{f}(\mathbf{y}) + \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{A}} \ . \quad (121)$$

The resulting $\mathbf{z} = \text{prox}^{\mathbf{A}}_{\mathbf{f}}(\mathbf{x})$ is often called the *proximal-point* to the convex function f at $\mathbf{x}$, and $\mathbf{z} \in \mathbf{dom} f$. In addition, we can define a mechanism to quantify the distance of the proximal projection via the *proximal vector distance function*

$$\text{dist}^{\mathbf{A}}_{\mathbf{f}}(\mathbf{x}) := \mathbf{x} - \text{prox}^{\mathbf{A}}_{\mathbf{f}}(\mathbf{x}) \ . \quad (122)$$

Furthermore, proximal operators are also applicable to define projections onto convex sets. Applying (121) to the indicator function (114) of a convex set $\mathcal{C}$, the proximal projection and distance operators take the form of

$$\text{prox}^{\mathbf{A}}_{\mathcal{C}}(\mathbf{x}) := \underset{\mathbf{y}\in\mathcal{C}}{\text{argmin}}\ \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{A}} \quad (123a)$$

$$\text{dist}^{\mathbf{A}}_{\mathcal{C}}(\mathbf{x}) := \mathbf{x} - \text{prox}^{\mathbf{A}}_{\mathcal{C}}(\mathbf{x}) \ . \quad (123b)$$

Lastly, the subdifferential of the proximal operator also yields an inclusion to the normal cone $\mathcal{N}_{\mathcal{C}}(\mathbf{x})$. Expanding (123a) using the definition of the indicator function $\Psi_{\mathcal{C}}(\mathbf{x})$ as

$$\mathbf{x}^* = \text{prox}^{\mathbf{A}}_{\mathcal{C}}(\mathbf{x}) = \underset{\mathbf{y}}{\text{argmin}}\ \underbrace{\Psi_{\mathcal{C}}(\mathbf{y}) + \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{A}}}_{f(\mathbf{y})} \ , \quad (124)$$

and evaluating the subdifferential at the proximal-point yields

$$\partial f(\mathbf{x}^*) = \partial\, \Psi_{\mathcal{C}}(\mathbf{x}^*) - \mathbf{A}\,(\mathbf{x}-\mathbf{x}^*) \ni 0 \ . \quad (125)$$

Thus we can see that the proximal operator renders a set inclusion to the normal cone of $\mathcal{C}$ at point $\mathbf{x}$ in the form of

$$\mathbf{x} \in \left(\mathcal{N}_{\mathcal{C}}(\mathbf{x}^*) + \mathbf{A}^{-1}\mathbf{x}^*\right) \ . \quad (126)$$

B. The Proximal Perspective

To enunciate how and why projective first-order methods follow the template in Alg. 1, we provide a brief derivation that highlights the underling common structure. This derivation, based on the augmented Lagrange method and proximal operators, has its roots in the early seminal work of J.J Moreau in [25] and has been elaborated upon extensively by others as well. The derivation provided here is principally based on the work of Studer et al in [66], [67] and Erleben in [54].

As first introduced in Sec. III, in order to tackle the NCP (15), the principle tool will be the optimization perspective via the NSOCP (16). Due to the nonlinear and non-differentiable De Saxcé operator, it is very difficult to solve directly. One method that has shown great success is that proposed by Cadoux [51] and further developed by Acary et al in [31], both taking inspiration from the original *predictor-corrector* approach of De Saxcé in [30]. This method attains a convexification of (16) via fixing the De Saxcé term to the image of its argument and proceeding by successive approximation of this value. As we will show, it renders the problem convex and differentiable, fitting well with the derivation of projective first-order methods. This convexified NCP takes the form of

$$\text{SOCP}(\mathbf{D}, \mathbf{v}_f, \mathcal{K}) :$$

$$\textbf{Given } \mathbf{s} = \mathbf{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) \quad (127)$$

$$\textbf{Find } \boldsymbol{\lambda} = \underset{\mathbf{x}\in\mathcal{K}}{\text{argmin}}\ \frac{1}{2}\mathbf{x}^T\,\mathbf{D}\,\mathbf{x} + \mathbf{x}^T\,(\mathbf{v}_f + \mathbf{s})$$

To simplify notation, we denote (127) with the shorthand

$$\begin{aligned} \min_{\mathbf{x}} \quad & h(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{f}(\mathbf{x}) \in \mathcal{K} \end{aligned} \ . \quad (128)$$

The first step involved rewriting the inclusion constraint as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & h(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{A}\,(\mathbf{f}(\mathbf{x}) - \mathbf{y}) = 0 \\ & \mathbf{y} \in \mathcal{K} \end{aligned} \ , \quad (129)$$

where $\mathbf{y} \in \mathbb{R}^{n_d}$ is a vector of *slack variables* and $\mathbf{A} \in \mathbb{S}^{n_d}_{++}$ is a matrix that will serve as a weighted metric norm. Since $\mathbf{A}$ is PD, the equality constraint can only be satisfied when $\mathbf{f}(\mathbf{x}) - \mathbf{y} = 0$. The augmented Lagrangian of (129) is

$$\mathcal{L}^A_{\rho,\mathbf{A}}(\mathbf{x},\mathbf{y},\mathbf{z}) := h(\mathbf{x}) + \mathbf{z}^T\,\mathbf{A}\ (\mathbf{f}(\mathbf{x}) - \mathbf{y}) + \frac{\rho}{2}\|\mathbf{f}(\mathbf{x}) - \mathbf{y}\|^2_A \quad (130a)$$

$$= h(\mathbf{x}) - \frac{1}{2\rho}\mathbf{z}^T\,\mathbf{A}\,\mathbf{z} + \frac{\rho}{2}\|\rho^{-1}\,\mathbf{z} + \mathbf{f}(\mathbf{x}) - \mathbf{y}\|^2_A \quad (130b)$$

$$= h(\mathbf{x}) - \frac{\rho^{-1}}{2}\|\mathbf{z}\|^2_A + \frac{\rho}{2}\|\rho^{-1}\,\mathbf{z} + \mathbf{f}(\mathbf{x}) - \mathbf{y}\|^2_A \ , \quad (130c)$$

where $\mathbf{z} \in \mathbb{R}^m$ is the vector of Lagrange multipliers, $\rho \in \mathbb{R}$ is the additional penalty parameter, and $\|\mathbf{x}\|^2_A = \mathbf{x}^T\,\mathbf{A}\,\mathbf{x}$ denotes taking the norm in the metric space defined by $\mathbf{A}$. Next, we eliminate the slack variables $\mathbf{y}$ from (129) by observing that $\mathcal{L}^A_{\rho,\mathbf{A}}(\mathbf{x},\mathbf{y},\mathbf{z})$ is minimized w.r.t $\mathbf{y} \in \mathcal{K}$ at the point

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{K}}{\text{argmin}}\ \|\rho^{-1}\,\mathbf{z} + \mathbf{f}(\mathbf{x}) - \mathbf{y}\|^2_A = \text{prox}^{\mathbf{A}}_{\mathcal{K}}(\rho^{-1}\,\mathbf{z} + \mathbf{f}(\mathbf{x})) \ , \quad (131)$$

yielding the $\mathbf{y}$-minimized augmented Lagrangian

$$\mathcal{L}^A_{\rho,\mathbf{A}}(\mathbf{x},\mathbf{y}^*,\mathbf{z}) = h(\mathbf{x}) - \frac{\rho^{-1}}{2}\|\mathbf{z}\|^2_A + \frac{\rho}{2}\|\text{dist}^{\mathbf{A}}_{\mathcal{K}}(\rho^{-1}\mathbf{z} + \mathbf{f}(\mathbf{x}))\|^2_A \ . \quad (132)$$

The elimination of $\mathbf{y}$ renders the reduced saddle-point problem

$$\min_{\mathbf{x}}\ \max_{\mathbf{z}}\ \mathcal{L}^A_{\rho,\mathbf{A}}(\mathbf{x},\mathbf{y}^*,\mathbf{z}) \ . \quad (133)$$

Forming the KKT optimality conditions

$$\nabla_{\mathbf{x}}\,\mathcal{L}_{\rho,\mathbf{A}}(\mathbf{x}^*,\mathbf{y}^*,\mathbf{z}^*) = 0 \ ,\ \nabla_{\mathbf{z}}\,\mathcal{L}_{\rho,\mathbf{A}}(\mathbf{x}^*,\mathbf{y}^*,\mathbf{z}^*) = 0 \ , \quad (134)$$

for (132), results in the system of equations

$$\nabla_{\mathbf{x}}\,h(\mathbf{x}) + \rho\,\mathbf{A}\,\text{dist}^{\mathbf{A}}_{\mathcal{K}}(\rho^{-1}\mathbf{z} + \mathbf{f}(\mathbf{x}))\,\nabla_{\mathbf{x}}\,\mathbf{f}(\mathbf{x}) = 0 \quad (135a)$$

$$-\rho^{-1}\,\mathbf{A}\,\mathbf{z} + \mathbf{A}\,\text{dist}^{\mathbf{A}}_{\mathcal{K}}(\rho^{-1}\mathbf{z} + \mathbf{f}(\mathbf{x})) = 0 \ , \quad (135b)$$

and with a bit of refactoring we can arrive at

$$\nabla_{\mathbf{x}} h(\mathbf{x}) + \mathbf{A}\,\mathbf{z}\,\nabla_{\mathbf{x}}\,\mathbf{f}(\mathbf{x}) = 0 \tag{136a}$$

$$\mathbf{f}(\mathbf{x}) = \text{prox}_{\mathcal{K}}^{\mathbf{A}}(\mathbf{f}(\mathbf{x}) + \rho^{-1}\,\mathbf{z}) \ . \tag{136b}$$

Now we can plug in the definition of the SOCP (127) to (136), and observe that the Lagrange multipliers of the dual problem correspond to $\mathbf{z} := -\hat{\mathbf{v}}$, i.e. negative augmented post-event constraint velocities. The resulting system is

$$\hat{\mathbf{v}} = \mathbf{A}\ (\mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \mathbf{s}) \tag{137a}$$

$$\boldsymbol{\lambda} = \text{prox}_{\mathcal{K}}^{\mathbf{A}}(\boldsymbol{\lambda} - \rho^{-1}\,\hat{\mathbf{v}}) \ , \tag{137b}$$

which, when combined form the *proximal-point equation*

$$\boldsymbol{\lambda} = \text{prox}_{\mathcal{K}}^{\mathbf{A}}(\boldsymbol{\lambda} - \rho^{-1}\,\mathbf{A}^{-1}\,(\mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \mathbf{s})) \ . \tag{138}$$

A crucial property of the proximal operator is that its solutions are also *fixed-points* of the projection operation, i.e.

$$\boldsymbol{\lambda}^* = \text{prox}_{\mathcal{K}}^{\mathbf{A}}(\boldsymbol{\lambda}^*) \ . \tag{139}$$

Therefore, interpreting (139) as a recursive procedure, and starting from some initial point $\boldsymbol{\lambda}^0$, successively applying the proximal operator should converge to one of its fixed-points in a finite number of steps. This is the so-called *proximal-point algorithm* and can be conceptualized as

$$\boldsymbol{\lambda}^* \approx \boldsymbol{\lambda}^n = \text{prox}_{\mathcal{K}}^{\mathbf{A}} \circ ... \circ \text{prox}_{\mathcal{K}}^{\mathbf{A}}(\boldsymbol{\lambda}^0) \ . \tag{140}$$

The final step involves realizing the $\text{prox}_{\mathcal{K}}^{\mathbf{A}}(\cdot)$ operator. The specifics of how this is done is exactly that which distinguishes most algorithms from one another. To recapitulate, in order to construct a proximal operator we need to choose:

- an appropriate metric norm $\mathbf{A}$
- an appropriate representation of ρ
- a method to realize the projection
- an iterative procedure

Moreover, another crucial choice is whether to realize the projection as one *global* operation accounting for the whole problem at once, or break it up into chunks that can be solved *locally*, depending of course on the structure of $\mathbf{A}$. The latter is what is referred to in the optimization literature as *block-coordinate descent* [68]. We will leave the global approach for Sec. VI-F where it is used in the ADMM-based solver. For now, we will focus on the block-wise methods that includes PGS and its variants. If $\mathbf{A}$ is block-diagonal, then we can express the block-wise operation

$$\begin{aligned} \mathbf{z}_j^i &= \left(\sum_{m=1}^{n_j+n_l+n_c} \mathbf{D}_{mj}\,\boldsymbol{\lambda}_m^i \right) + \mathbf{v}_{f,j} + \mathbf{s}_j \\ \boldsymbol{\lambda}_j^{i+1} &= \text{prox}_{\mathcal{K}_j}^{\mathbf{A}_{jj}} \left(\boldsymbol{\lambda}_j^i - \tfrac{1}{\rho}\,\mathbf{A}_{jj}^{-1}\,\mathbf{z}_j^i \right) \end{aligned} \ , \tag{141}$$

and, if $\mathbf{A}$ is purely diagonal, the coordinate-wise version is

$$\begin{aligned} z_j^i &= \textstyle\sum_{m=1}^{n_d} D_{mj}\,\lambda_m^i + v_{f,j} + s_j \\ \lambda_j^{i+1} &= \text{prox}_{\mathcal{K}_j}^{a_j} \left(\lambda_j^i - \rho^{-1}\,a_j^{-1}\,z_j^i \right) \end{aligned} \ . \tag{142}$$

We should now recognize that (141) and (142) correspond to a single Jacobi iteration, followed by a projection onto the feasible set of each coordinate. We can therefore alternatively use a Gauss-Seidel (GS) iteration [69], in the form of

$$\begin{aligned} \mathbf{z}_j^i &:= \sum_{m=1}^{m\leq j} \mathbf{D}_{mj}\boldsymbol{\lambda}_m^{i+1} + \sum_{m=j+1}^{n_j+n_l+n_c} \mathbf{D}_{mj}\boldsymbol{\lambda}_m^i + \mathbf{v}_{f,j} + \mathbf{s}_j \\ \boldsymbol{\lambda}_j^{i+1} &= \text{prox}_{\mathcal{K}_j}(\,\boldsymbol{\lambda}_j^i - \rho^{-1}\,\mathbf{A}_{jj}^{-1}\,\mathbf{z}_j^i\,) \end{aligned} \ , \tag{143}$$

$$\begin{aligned} z_j^i &:= \sum_{m=1}^{m\leq j} D_{mj}\lambda_m^{i+1} + \sum_{m=j+1}^{n_d} D_{mj}\lambda_m^i + v_{f,j} + s_j \\ \lambda_j^{i+1} &= \text{prox}_{\mathcal{K}_j}(\,\lambda_j^i - \rho^{-1}\,A_{jj}^{-1}\,z_j^i\,) \end{aligned} \ . \tag{144}$$

We can also use a Successive-Over-Relaxation (SOR) update

$$\boldsymbol{\lambda}_j^{i+1} = (1-\omega)\boldsymbol{\lambda}_j^i + \omega\boldsymbol{\lambda}_j^{i+1} \ , \tag{145}$$

after every block-wise or coordinate-wise proximal projection. $\omega \in [0, 2]$ is the so-called relaxation factor that linearly interpolates between the previous and current iterates. This additional operation presents some interesting theoretical and practical properties, and is often used to accelerate convergence. However, it can potentially lead to divergence if not selected carefully, and choosing a value that generalizes across all problems can prove difficult. Choosing $\omega = 1$ corresponds to the GS iteration and thus this additional mechanism is a very versatile and easy extension to apply.

Essentially, all of the aforementioned *local* approaches treat each sub-problem as though all other constraint reactions are fixed constants. Moreover, the structure of the proximal-point algorithm is akin to *projected gradient descent*. As indicated by (136), the velocity term corresponds to an estimate of the gradient, $\mathbf{A}$ serves as an approximate Hessian and ρ can be understood as a step-size. Lastly, the projection onto the feasible set via the proximal operator exclusively serves to satisfy the set-valued and inequality constraints. We have thus laid the foundation which unifies the set for tackling the dual FD NCP. It remains now to choose how to realize the projection and select ρ and $\mathbf{A}$, and will be covered next.

C. Projectors & Local Solvers

This section describes the set of projection operators that are used to realize the proximal operator (123b). As the latter can be considered more of a theoretical construct as opposed to a literal operation, we will denote an *implementation* thereof as a projection operator, i.e. *projector*, $\mathcal{P}^{\mathcal{K}}(\cdot)$ introduced in the template of Alg.1. Many of them are derived analytically in closed-form, while some require solving an inner optimization problem. Recalling the construction (81) used for the composite set $\mathcal{K}$ of admissible constraint reactions, we can equivalently construct a composite projector as

$$\mathcal{P}^{\mathcal{K}}(\mathbf{x}) := \begin{bmatrix} \mathcal{P}^{\mathcal{K}_1}(\mathbf{x}_1) \\ \vdots \\ \mathcal{P}^{\mathcal{K}_{n_i}}(\mathbf{x}_{n_i}) \end{bmatrix} \ , \tag{146}$$

where n_i is the number of inequality constraint sets. Note that, in general, $n_i \neq n_l + n_c$, as the constraint sets can differ depending on the contact model e.g. decoupled or spatial.

Euclidean Projectors: The first set consists of geometric projectors admitting analytical expressions. Specifically, they realize Euclidean projections onto their respective cones, as they are derived assume the Euclidean metric norm. These include the:

- projector to the nonnegative orthant $\mathbb{R}_+$

$$\mathcal{P}^{\mathbb{R}_+}(x) := \max(0, x) \ . \tag{147}$$

- projector to a 2D disk $\mathcal{D}(r) \subset \mathbb{R}^2$ of radius $r \in \mathbb{R}_+$:

$$\mathcal{P}^{\mathcal{D}(r)}(\mathbf{x}) := \begin{cases} \mathbf{x} & , \|\mathbf{x}\|_2 \leq r \\ r \, \frac{\mathbf{x}}{\|\mathbf{x}\|_2} & , \|\mathbf{x}\|_2 > r \end{cases} \tag{148}$$

- projector to a generic 3D Lorentz cone $\mathcal{K}_3 \subset \mathbb{R}^3$:

$$\mathcal{P}^{\mathcal{K}_3}(\mathbf{x}) := \begin{cases} 0 & , \|\mathbf{x}_{xy}\|_2 \leq -\mathrm{x}_z \\ \mathbf{x} & , \|\mathbf{x}_{xy}\|_2 \leq \mathrm{x}_z \\ \frac{1}{2}\left(1 + \frac{x_z}{\|\mathbf{x}_{xy}\|_2}\right) \begin{bmatrix} \mathbf{x}_{xy} \\ \|\mathbf{x}_{xy}\|_2 \end{bmatrix} & , \|\mathbf{x}_{xy}\|_2 > |\mathrm{x}_z| \end{cases} \tag{149}$$

- projector to a Coulomb friction cone $\mathcal{K}_\mu \subset \mathbb{R}^3$

$$\mathcal{P}^{\mathcal{K}_\mu}(\mathbf{x}) := \begin{cases} 0 & , \|\mathbf{x}_t\|_2 \leq -\mu^{-1}\,\mathrm{x}_n \\ \mathbf{x} & , \|\mathbf{x}_t\|_2 \leq \mu\,\mathrm{x}_n \\ \frac{\mu\,\|\mathbf{x}_t\|_2 + \mathrm{x}_n}{\mu^2+1} \begin{bmatrix} \frac{\mu\,\mathbf{x}_t}{\|\mathbf{x}_t\|_2} \\ 1 \end{bmatrix} & , \|\mathbf{x}_t\|_2 > \mu\,|\mathrm{x}_n| \end{cases} , \tag{150}$$

- projector to the dual Coulomb friction cone $\mathcal{K}^*_\mu := \mathcal{K}_{\frac{1}{\mu}}$

$$\mathcal{P}^{\mathcal{K}^*_\mu}(\mathbf{x}) := \begin{cases} 0 & , \|\mathbf{x}_t\|_2 \leq -\mu\,\mathrm{x}_n \\ \mathbf{x} & , \|\mathbf{x}_t\|_2 \leq \mu^{-1}\,\mathrm{x}_n \\ \frac{\|\mathbf{x}_t\|_2 + \mu\,\mathrm{x}_n}{\mu^2+1} \begin{bmatrix} \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|_2} \\ \mu \end{bmatrix} & , \|\mathbf{x}_t\|_2 > \mu^{-1}\,|\mathrm{x}_n| \end{cases} . \tag{151}$$

Note that the Coulomb cone and its dual are essentially specializations of the 3D Lorentz cone (149). Moreover, although omitted from the aforedescribed list, $\mathbb{R}^n$ also admits a projector in the form of the identity function $\mathcal{P}^{\mathbb{R}^n}(\mathbf{x}) = \mathbf{x}$.

Local Solvers: The second set consists of projectors that are additionally embedded with a contact model, in contrast to the aforedescribed primitives which are purely geometric. They can explicitly incorporate the Signorini-Coulomb contact model, resulting in projections to the Coulomb friction cone that satisfy the complementarity conditions. They are therefore solvers of the single-contact dual problem and are used in constraint-wise iteration schemes.

The first such projectors we consider are $\mathcal{P}_{CCP}(\cdot)$ and $\mathcal{P}_{NCP}(\cdot)$ that are respectively defined in Alg. 2 and Alg. 3. These are very simple approaches to realizing the proximal projection that involve nothing more than a single-step predictor-corrector operation. Specifically, $\mathcal{P}_{CCP}(\cdot)$ can be interpreted as assuming $\mathbf{A} = \mathbb{I}_{n_d}$ and ρ set to the inverse of the average diagonal of the local Delassus matrix (i.e. respective diagonal block). A subtlety of this approach is that if an estimate of the local De Saxcé correction is omitted, then the dual variable is effectively the contact velocity $\mathbf{v}_j^+$, which leads to non-zero solutions along the contact normal. This is because it only enforces complementarity between primal and dual variables, without explicit consideration of the reaction and velocity along the contact normal.

Conversely, $\mathcal{P}_{NCP}(\cdot)$ addresses this issue by decomposing the problem along the normal and tangent directions as two separate problems. This ensures zero contact velocity along the contact normal by forcing the normal contact reaction to be non-negative regardless of the presence of a De Saxcé correction term. For this projector, again $\mathbf{A} = \mathbb{I}_{n_d}$ but ρ is different for the normal and tangential directions. For the normal direction, it is taken directly as the inverse of the respective diagonal component of $\mathbf{D}$, while for both tangential components, the smallest of the two diagonal components is used. However, this projector can lead to contact reactions that are not maximally dissipative since the decoupling cannot account for the couplings induced by other constraints.

Both $\mathcal{P}_{CCP}(\cdot)$ and $\mathcal{P}_{NCP}(\cdot)$ can be considered as *naive* projectors. A more holistic approach would be to consider solving each local problem exactly, accounting for all other couplings brought by the local dual variable $\mathbf{z}_j^i$. This approach has been described in works such as that of Studer in [67], Bonnefon et al in [70], and Todorov in [12]. These methods solve a single-contact problem by taking the geometric perspective described in Sec. IV-D and treating the open, stick and sliding cases explicitly. For the first two cases, the projection is trivially zero and the unconstrained solution of the linear system (i.e. initially assuming the dual variable is zero). For the sliding case, the solution is found by considering the conic section formed by the Coulomb cone and plane of maximum compression. Finding the correct point amounts to finding a point on a 2D ellipse which can be rendered as a solution to

Algorithm 2 CCP-Type Local Solver $\mathcal{P}_{CCP}(\cdot)$

Require: $\boldsymbol{\lambda}_j$, $\mathbf{D}_{jj}$, $\mathbf{v}_{f,j}$, $\mathbf{s}_j$, μ ▷ local problem definition

1: $\mathbf{z}_j \leftarrow \mathbf{D}_{jj}\,\boldsymbol{\lambda}_j + \mathbf{v}_{f,j} + \mathbf{s}_j$ ▷ compute local velocity

2: $\boldsymbol{\lambda}_j \leftarrow \boldsymbol{\lambda}_j - \dfrac{3}{\mathbf{tr}(\mathbf{D}_j)}\,\mathbf{z}_j$

3: $\boldsymbol{\lambda}_j \leftarrow \mathcal{P}^{\mathcal{K}_\mu}(\boldsymbol{\lambda}_j)$ ▷ Coulomb cone projector (150)

4: **return** $\boldsymbol{\lambda}_j$

Algorithm 3 NCP-Type Local Projector $\mathcal{P}_{NCP}(\cdot)$

Require: $\boldsymbol{\lambda}_j$, $\mathbf{D}_{jj}$, $\mathbf{v}_{f,j}$, $\mathbf{s}_j$, μ ▷ local problem definition

1: $\mathbf{z}_j \leftarrow \mathbf{D}_{jj}\,\boldsymbol{\lambda}_j + \mathbf{v}_{f,j} + \mathbf{s}_j$ ▷ compute local velocity

2: $\lambda_{j,N} \leftarrow \lambda_{j,N} - \dfrac{1}{D_{j,NN}}\,z_{j,N}$

3: $\lambda_{j,N} \leftarrow \mathcal{P}^{\mathbb{R}_+}(\lambda_{j,N})$ ▷ orthant projector (147)

4: $\boldsymbol{\lambda}_{j,T} \leftarrow \boldsymbol{\lambda}_{j,T} - \dfrac{1}{\min(D_{j,TT,x}, D_{j,TT,y})}\,\mathbf{z}_{j,T}$

5: $\boldsymbol{\lambda}_{j,T} \leftarrow \mathcal{P}^{\mathcal{D}(\mu\,\lambda_{j,N})}(\boldsymbol{\lambda}_{j,T})$ ▷ disk projector (148)

6: **return** $\boldsymbol{\lambda}_j$

a Quadratic Cone Program (QCP). In the approaches of [67], [70], the QCP was used to derive a semi-analytical solution in the form of a root-finding problem of a *quartic polynomial*. In this work we will denote such QCP-based local solvers as the Nonlinear-Block projector $\mathcal{P}_{NB}(\cdot)$, defined in Alg. 4. Such a projector can used to realize a form of dual solver some authors refer to as the Nonlinear-Block Gauss Seidel method [71], [72], [12]. However, the derivation of the QCP can vary depending on certain modeling assumptions.

In this work we have realized the variants proposed by Preclik et al in [56] and by Hwangbo et al in [40], respectively denoted as $\mathcal{P}_{MDP}(\cdot)$ and $\mathcal{P}_{BS}(\cdot)$. Both of these projectors are based on the derivation originally described by Preclik et al in [56], but differ in how the QCP is solved. In the case of the former, the single-contact problem is transcribed as a single-variable QCP that is solved as a quartic polynomial, similar to [67], [70], [12]. It differs to the aforementioned, however, in that the QCP is derived using a model that employs an alternate form of the disjunctive Signorini-Coulomb model (46), in conjunction with a objective function derived from the MDP (47). This approach was later adapted by Hwangbo et al in [40], who revised it by solving the QCP using a bisection-search (BS) method, thusly employing it in the RaiSim simulator. A recent analysis of the former by Lidec et al in [73], identified that the QCP formulation was actually violating the MDP when strong couplings between tangential and normal reactions occur, and proposed an amendment based on approximating the De Saxcé correction term (50).

Algorithm 4 Nonlinear-Block Local Solver $\mathcal{P}_{NB}(\cdot)$

Require: $\boldsymbol{\lambda}_j$, $\mathbf{D}_{jj}$, $\mathbf{v}_{f,j}$, $\mathbf{s}_j$, μ ▷ local problem definition
$\boldsymbol{\lambda}_j^0 \leftarrow -\mathbf{D}_{jj}^{-1}\,(\mathbf{v}_{f,j}+\mathbf{s}_j)$ ▷ unconstrained solution
if $\mathrm{v}_{f,n} < 0$ **then**
  if $\mu = 0$ **then**
    $\boldsymbol{\lambda}_j = [0,\,0,\,-D_{nn}^{-1}\,v_{f,n}]^T$ ▷ contact is frictionless
  else
    if $\boldsymbol{\lambda}_0 \in \mathcal{S} \cap \mathcal{K}_\mu$ **then**
      $\boldsymbol{\lambda}_j = \boldsymbol{\lambda}_0$ ▷ contact is sticking
    else
      if $\mathrm{v}_{f,n} = 0$ **then**
        if $r_{d/\mu} > 1$ **then**
          $\boldsymbol{\lambda}_j = 0$ ▷ degenerate elliptic
        else
          $\boldsymbol{\lambda}_i$ from (224) ▷ degenerate para/hyperbolic
        end if
      else
        $\boldsymbol{\lambda}_j$ from (222) ▷ contact is slipping
      end if
    end if
  end if
else
  $\boldsymbol{\lambda}_j = 0$ ▷ contact is open
end if
return $\boldsymbol{\lambda}_j$

We will briefly summarize the definition of the projector through the lens of proximal operators, while a complete derivations can be found in Appendix. C. Readers are also referred to [56], [40], [73] for further details. In our context, the local-contact QCP derived by Preclik that also incorporates the De Saxcé correction, takes the form of the problem

$$\begin{aligned} \textbf{Find} \quad \boldsymbol{\lambda}_j = \underset{\mathbf{x}}{\operatorname{argmin}} \quad & \tfrac{1}{2}\,\mathbf{x}^T\,\mathbf{D}_{jj}\,\mathbf{x} + \mathbf{x}^T\,(\mathbf{v}_{f,j}+\mathbf{s}_j) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{K}_{\mu_j} \\ & \mathbf{D}_{jj}\,\mathbf{x} + \mathbf{v}_{f,j} \geq 0 \\ & \mathbf{D}_{jj}\,\mathbf{x} \geq 0 \end{aligned} \quad . \tag{152}$$

The additional inequality constraints correspond to the Signorini condition on the post-event contact velocity, and to the requirement that the contact reaction does not induce a velocity bias along the plane normal. Although the former is typically understood to be a condition of optimality, including it explicitly admits its geometric reinterpretation as finding a point on a conic section. Keeping to the broader view, we will express (152) as a proximal operator. Denoting the inequality constraints as inclusions to the sets

$$\mathbf{n} := [0,\,0,\,1]^T\ , \tag{153a}$$

$$\mathcal{V}_{\mathbf{n}} := \{\mathbf{x}\,:\,\mathbf{n}^T\,(\mathbf{D}_{jj}\,\mathbf{x}+\mathbf{v}_{f,j}) \geq 0\}\ , \tag{153b}$$

$$\Lambda_{\mathbf{n}} := \{\mathbf{x}\,:\,\mathbf{n}^T\,\mathbf{D}_{jj}\,\mathbf{x} \geq 0\} \tag{153c}$$

we can recast (152) as a proximal projection in the form of

$$\operatorname{prox}^{\mathbf{D}}_{\mathcal{V}_{\mathbf{n}}\cap\mathcal{K}_\mu}(\mathbf{x}) := \underset{\mathbf{y}}{\operatorname{argmin}}\, f_{\text{MDP}}(\mathbf{y},\mathbf{x}) \tag{154a}$$

$$\begin{aligned} f_{\text{MDP}}(\mathbf{x},\mathbf{y}) := {} & \Psi_{\mathcal{K}_{\mu_j}}(\mathbf{y}) + \Psi_{\mathcal{V}_{\mathbf{n}}}(\mathbf{y}) \\ & + \Psi_{\Lambda_{\mathbf{n}}}(\mathbf{y}) + \frac{1}{2}\,\|\mathbf{x}-\mathbf{y}\|^2_{\mathbf{D}_{jj}}\ . \end{aligned} \tag{154b}$$

Note that this equivalence holds only if $\mathbf{x} := -\mathbf{D}_{jj}\,(\mathbf{v}_{f,j}+\mathbf{s}_j)$. Plugging in the unconstrained solution to (154b), yields (152). This projector therefore corresponds to a proximal operator with the single-contact Delassus matrix as a weighted metric norm, i.e. $\mathbf{A} = \mathbf{D}_{jj}$ and $\rho = 1$. The realization of (154b) as a projector applicable to per-constraint iteration is thus

$$\mathcal{P}_{\text{MDP}}(\mathbf{x}) := \begin{cases} 0 & ,\bar{\mathrm{v}}_{f,n} < 0 \\ \mathbf{x} & ,\bar{\mathrm{v}}_{f,n} \geq 0,\ \mathbf{x} \in \mathcal{K}_\mu \\ \text{QCP}(\mathbf{x},\bar{\mathbf{D}},\bar{\mathbf{v}}_f,\mathcal{K}_\mu) & ,\bar{\mathrm{v}}_{f,n} \geq 0,\ \mathbf{x} \notin \mathcal{K}_\mu \end{cases} . \tag{155}$$

D. Convergence

This section describes how we may realize the termination criteria functions $f_{stop}(\cdot)$ of Alg. 1. Although such operations do not directly affect the convergence trajectories of the corresponding algorithms, they are critical to the quality of the resulting solutions and useful for assessing solver performance. Thus, special care must be taken in selecting which quantities should be evaluated in order to terminate a given solver, as the criteria must reflect the effect of the algorithm.

Given the transcription of the problem as the NSOCP (16), the most relevant measures to consider, would be the *primal,*

dual and complementarity residuals, defined respectively as

$$\mathbf{r}_p(\boldsymbol{\lambda}) := \boldsymbol{\lambda} - \mathcal{P}^{\mathcal{K}}(\boldsymbol{\lambda}) \tag{156}$$

$$\mathbf{r}_d(\mathbf{v}) := \mathbf{v} - \mathcal{P}^{\mathcal{K}^*}(\mathbf{v}) \ , \tag{157}$$

$$\mathbf{r}_{cp}(\boldsymbol{\lambda}, \mathbf{v}) := [\, \boldsymbol{\lambda}_i^T \, \mathbf{v}_i \,]_{n_l+n_c} \ , \tag{158}$$

where $[x_i]_n := [x_1, \cdots, x_1]^T$. Note how the primal and dual residuals are conveniently expressed using the projectors described in Sec. VI-C. This property reflects the fact that the primal and dual variables of the problem should satisfy the normal-cone inclusions (120). Thus, a measure of the error corresponds to the proximal distance operator (122).

There is a crucial subtlety regarding the formulation of the dual and complementarity residuals compared to the primal. While, the primal residual (156) is a universal measure that is applicable in all cases, (157) and (158) must be evaluated according to whether the algorithm is solving a NCP or CCP formulation of the dual problem. For the NCP, they would be evaluated using the augmented constraint velocity $\hat{\mathbf{v}}^+$ that includes the De Saxcé correction (14), while for the CCP, it would be the post-event constraint velocity $\mathbf{v}^+$ (13).

Moreover, we can also consider the *iterate residual*

$$\mathbf{r}_{iter}(\boldsymbol{\lambda}^i, \boldsymbol{\lambda}^{i-1}) := \boldsymbol{\lambda}^i - \boldsymbol{\lambda}^{i-1} \ , \tag{159}$$

i.e. the relative difference between successive iterates $\boldsymbol{\lambda}^i$ and $\boldsymbol{\lambda}^{i+1}$ of the solution. This alternative measure can be used in several ways. Firstly, some approaches described in the literature employ (159) as a computationally cheaper method of determining convergence [7], when the evaluation of (156-158) may prove costly. Second, it can be used in conjunction with problem residuals as an auxiliary criteria for detecting solver stagnation, and thus invoke early stopping of the algorithm.

Some simulators like MuJoCo [37], option for using alternative convergence criteria based on the objective function. Specifically, in the case of MuJoCo's GPGS solver, the quadratic objective (17a) is evaluated over successive solution iterates in order to compute a *relative improvement* of the objective function, that is scaled by the total system inertia:

$$r_f := I_{total}^{-1} \, |f(\boldsymbol{\lambda}^i) - f(\boldsymbol{\lambda}^{i-1})| \ , \tag{160}$$

where $I_{total} > 0$ is the total diagonal inertial, i.e. the sum of diagonal terms of the generalized mass matrix $\mathbf{M}$. Essentially, this heuristic aims at identifying plateaus in the optimization in order to perform early stopping, thus preventing the algorithm from iterating without significant effect. The scaling using the total diagonal inertia, can be understood to improve conditioning w.r.t large mass ratios present in the system.

In addition to the aforedefined metrics, another useful element from VI theory is the *natural map function*

$$\mathbf{F}_{\text{VI}}^{\text{nat}}(\mathbf{x}) := \mathbf{x} - \mathcal{P}^{\mathcal{K}}\left(\mathbf{x} - \mathbf{F}(\mathbf{x})\right) \ , \tag{161}$$

which can be stated generically for any NCP of the form

$$\mathcal{K}^* \ni \mathbf{F}(\mathbf{x}) \perp \mathbf{x} \in \mathcal{K}^* \ . \tag{162}$$

This functional stems from a rather intuitive geometric interpretation of (162) that is depicted in Fig.7. In [74], [6], [60], it is proven that $\mathbf{F}_{\text{VI}}^{\text{nat}}(\mathbf{x}^*) = 0$ for some vector $\mathbf{x}^* \in \mathcal{K}$, is a necessary and sufficient condition for $\mathbf{x}^*$ to be a solution to the NCP (162). The utility of this metric is often overlooked in the robotics and graphics fields, but as Acary et al demonstrate in [7], it can serve as a reliable metric for qualifying if the output of solvers actually correspond to solutions of the NCP/CCP. In effect, it subsumes the primal, dual and complementarity residuals, summarizing them into a single expression, albeit one that is significantly more expensive to compute. In the case of the dual FD problems (15) and (18), the functional $\mathbf{F}(\cdot)$ corresponds to $\hat{\mathbf{v}}^+$ and $\mathbf{v}^+$, respectively. Thus, the natural-map functions are for each problem are

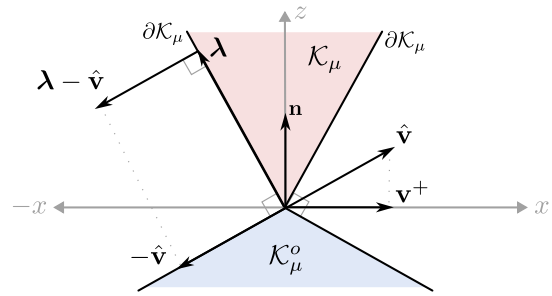

Fig. 7. A depiction of the geometric interpretation of the natural-map function. The VI functional is defined to be the augmented constraint velocity $\hat{\mathbf{v}} := \mathbf{F}(\boldsymbol{\lambda})$. If $\boldsymbol{\lambda} \in \partial\mathcal{K}_\mu$ and $-\mathbf{F}(\boldsymbol{\lambda}) \in \partial\mathcal{K}^o$ with $\boldsymbol{\lambda} \perp \mathbf{F}(\boldsymbol{\lambda})$, then the vector $\boldsymbol{\lambda} - \mathbf{F}(\boldsymbol{\lambda})$ originating at $\boldsymbol{\lambda}$, is exactly perpendicular to the surface of $\mathcal{K}_\mu$. This means that its projection onto the cone should yield $\mathbf{x}$, i.e. $\mathbf{F}_{\text{VI}}^{\text{nat}}(\boldsymbol{\lambda}) = 0$.

$$\mathbf{F}_{\text{NCP}}^{\text{nat}}(\boldsymbol{\lambda}) := \boldsymbol{\lambda} - \mathcal{P}^{\mathcal{K}}\left(\boldsymbol{\lambda} - \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \boldsymbol{\Gamma}\left(\mathbf{v}^+(\boldsymbol{\lambda})\right)\right) \ , \tag{163}$$

$$\mathbf{F}_{\text{CCP}}^{\text{nat}}(\boldsymbol{\lambda}) := \boldsymbol{\lambda} - \mathcal{P}^{\mathcal{K}}\left(\boldsymbol{\lambda} - \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f\right) \ . \tag{164}$$

Lastly, we must clarify how the aforedescribed vector-valued residuals and metrics may be used in practice. The two most prominent approaches employed in the majority of numerical methods are the averaged L_2 and L_∞ norms. The former computes $n_d^{-1}\|\cdot\|_2$, and can serve as a measure of the average error. The latter computes $\|\cdot\|_\infty$ and effectively renders the worst-case error. In this work, however, employ the L_∞ norm as it scales better w.r.t problem dimensions.

E. Projective Splitting Methods

With all the aforedescribed building-blocks in hand, we can now define the first set of concrete algorithms we will use to solve the FD dual NCP. We will refer to these as *Projective Splitting Methods*, in accordance with the taxonomy described in [7]. Generally speaking, these algorithms all are block-wise versions of the SOR method described in Sec. VI-B. SOR is the most general of the so-called fixed-point iteration methods based on matrix splitting[10]. This categorization is also justified by the observation that in the case of only bilateral constraints being present, the solvers are exactly equivalent to an indirect method for solving the linear system formed by the EoM (74) and the velocity-level constraints (22).

As a matter of convention, however, we can think of these as variants of the classic Projected Gauss-Seidel (PGS) [75],

[10]SOR, Gauss-Seidel and the Jacobi method are so-called *splitting methods*. These are a family of the more broader class of fixed-point iteration methods that also includes *Krylov subspace methods* such as the linear Conjugate Gradient (CG) algorithm, and the Minimum Residual (MINRES) method.

[76], [77], which is ubiquitous in the realm of physical simulation. Indeed it is present in most, if not all, of the popular physics engines such MuJoCo[58], PhysX [78], Bullet [20], RaiSim [40] and ODE [57]. However, the variants that are most commonly distinguished from "vanilla" PGS are PSOR [79], GPGS [37], [12], NBGS [72]. Moreover, a generalization of all of these algorithms, and arguably the most theoretically founded, is the PROX algorithm [66], [54] (a.k.a. SORProx). In addition to being the closest to the exact structure of the block-wise proximal-point algorithm (141), it selects $\mathbf{A}$ and ρ exactly according to the conditioning of the problem and performs an automatic adaptation of these parameters over successive iterations. However, purely due to time considerations, the authors regretfully have not been able to include the PROX algorithm in this evaluation, but we hope to do so in the future.

All projective splitting-based solvers are realized for the multi-constrained multi-body FD problem case using the scheme defined in Alg. 5. PGS-CCP by Tasora et al [80] is a blocked variant of PGS that solves the CCP (18) formulation of the dual problem using Euclidean projection, i.e. $\mathbf{A} = \mathbb{I}_{n_d}$ and an ALM penalty equal to the average of the diagonal terms of each local Delassus matrix. PGS-NCP is a version of the classic PGS adapted to 3D Coulomb friction cones that uses decoupled normal-tangent projections to enforce the NCP constraints. NBGS by Preclik et al [56] that uses an blocked version of PGS that solves each local per-contact problem exactly using a semi-analytical solution based on a decision tree and a quartic polynomial. RAISIM by Hwangbo et al [40] is fundamentally identical to NBGS except that it uses the $\mathcal{P}_{BS}(\cdot)$ projector based on the bisection-search method. RAISIM-DS by Lidec et al [73] is an enhanced version of RAISIM that also incorporates successive approximations of the De Saxcé correction in the local solver of per-contact constraints. RAISIM-DS-ES, an Early Stopping (ES) version of RAISIM-DS that replaces the constraint-based termination criteria with the relative objective function improvement (160).

F. Alternating Direction Method of Multipliers

In the context of mechanics, ADMM has only recently began to be explored as an alternative to first-order methods such as PGS and its variants. Tasora et al in [45], applied ADMM to the relaxed CCP (18) of the dual FD problem transcribed as the convex SOCP (19). This work demonstrated impressive versatility in being able to simulate both constrained rigid-body systems such as a robotic manipulator as well as stacks of flexible bodies such as deformable bars. Similarly, others have also applied ADMM to the primal FD problem with equally impressive demonstrations of its capabilities. Daviet in [81], [82] used it to solve a softened version of the problem, rendering exceptionally realistic simulations of compliant hair fibers. Very recently, Lee et al in [83] used it to solve an NCP-type formulation for simulating articulated robots, demonstrating competitive accuracy-speed tradeoffs compared to a state-of-the-art Newton-based method.

The second set of solvers we have evaluated are based on the ADMM algorithm of Boyd et al [64]. Specifically, we have realized two ADMM-based methods to solve the dual FD problem that respectively tackle the CCP (18) and NCP (15). The first, here referred to as ADMM-CCP, is based on the algorithm proposed by Lidec et al in [38], that similarly to that of Tasora et al [45], constitutes a direct application of ADMM to the SOCP (19). The second and more advanced solver, here referred to as ADMM-NCP, is based on the proximal formulation proposed by Carpentier et al in [32] that combines ADMM with proximal algorithms in a form that can solve the NCP via the convex SOCP (127). In both cases, we have adapted the original algorithms to the maximal-coordinate CRBD formulation in order to also incorporate bilateral joint and unilateral limit constraints.

As it turns out however, we found that both ADMM-CCP and ADMM-NCP are so similar, that they can share a common implementation. They only difference between them, in fact, is that ADMM-NCP includes two additional terms: (a) the estimate of the De Saxcé correction (78), and (b) an additional proximal regularization objective. Thus, to bring these solvers into the context of this work, we provide a brief outline of the proximal formulation of ADMM-NCP, describe the computations that are required to realize it, and clarify how it is relaxed to form ADMM-CCP. For further details regarding ADMM-NCP and ADMM in general, we respectively refer the interested reader to [64] and [32]. We highly recommend these original works, as well as [45], as these proved very insightful in better understanding how and why ADMM can be applied to such problems.

Proximal ADMM: As yet another first-order method adhering to the template of Alg. 1, ADMM-NCP shares a near-identical derivation to splitting-based methods. Thus our starting point to deriving it is a direct sequel to Sec. VI-B. The first step is recognizing that ADMM is in fact a global solver that accounts for all constraint simultaneously, as opposed to PGS and its variants that use the per-constraint iteration scheme in conjunction with a local single-contact solver.

Thus, ADMM-NCP effectively computes a direct approximation of the global proximal-point projection

$$\boldsymbol{\lambda} = \text{prox}_{\mathcal{K}}^{\mathbf{A}}(\boldsymbol{\lambda} - \rho^{-1}\,\mathbf{A}^{-1}\,(\mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \mathbf{s}))\ .$$

Rewriting the objective function of the convex SOCP (127) as

$$h(\mathbf{x}, \mathbf{s}) := \frac{1}{2}\,\mathbf{x}^T\,\mathbf{D}\,\mathbf{x} + \mathbf{x}^T(\mathbf{v}_f + \mathbf{s})\ , \tag{165}$$

and setting the metric tensor $\mathbf{A} = \mathbb{I}_{n_d}$ to employ the Euclidean norm, we can define the augmented Lagrangian

$$\begin{aligned}\mathcal{L}^A_{\rho,\eta}(\mathbf{s},\,\mathbf{x},\,\mathbf{y},\,\mathbf{z}) := \quad & h(\mathbf{x},\,\mathbf{s}) + \Psi_{\mathcal{K}}(\mathbf{y}) \\ & + \tfrac{\eta}{2}\|\mathbf{x} - \mathbf{x}^-\|_2^2 - \tfrac{1}{2\rho}\|\mathbf{z}\|_2^2 \quad , \\ & + \tfrac{\rho}{2}\|\mathbf{x} - \mathbf{y} - \rho^{-1}\mathbf{z}\|_2^2\end{aligned} \tag{166}$$

where η is an additional proximal parameter, and $\mathbf{x}^-$ is the previous estimate of the primal variable. Typically, the additional proximal parameter is set to $\eta = 10^{-6}$. Now note how(166) is otherwise identical to (130c), except for the additional proximal-point term $\frac{\eta}{2}\|\mathbf{x} - \mathbf{x}^-\|_2^2$. However

simple it may seem, this single term was one of the important insights of [32], as it provides a special form of regularization to the Delassus matrix that renders the Hessian of (165) strictly convex while totally avoiding any biasing effects in the solution. Another way to interpret the effect of this term is that the Lagrangian (166) implies a double proximal projection, one to the feasible set and another to the previous iterate.

A crucial shortcut that ADMM takes from the typical ALM derivation described in Sec. VI-B, is that it defines the alternating directions that update the primal variables $\mathbf{x}$ and $\mathbf{y}$. Thus each iteration of ADMM-NCP decomposes the problem into a cascade the of sub-problems

$$\mathbf{s}^i = \mathbf{\Gamma}(\mathbf{v}^+(\mathbf{x}^{i-1})) \ , \tag{167a}$$

$$\mathbf{x}^i = \underset{\mathbf{x}}{\operatorname{argmin}} \ \mathcal{L}^A_{\rho,\eta}(\mathbf{s}^i, \mathbf{x}, \mathbf{y}^{i-1}, \mathbf{z}^{i-1}) \ , \tag{167b}$$

$$\mathbf{y}^i = \underset{\mathbf{y}}{\operatorname{argmin}} \ \mathcal{L}^A_{\rho,\eta}(\mathbf{s}^i, \mathbf{x}^i, \mathbf{y}, \mathbf{z}^{i-1}) \ , \tag{167c}$$

$$\mathbf{z}^i = \mathbf{z}^{i-1} - \rho\,(\mathbf{x}^i - \mathbf{y}^i) \ . \tag{167d}$$

The first step (167a) performs the convex relaxation of the NSOCP (16) via the iterate estimates of the De Saxcé correction. The second step (167b) corresponds to an unconstrained QP that renders the unconstrained solution $\mathbf{x}^i$ directly as

$$\mathbf{x}^i = -\mathbf{D}^{-1}_{\rho,\eta}\left(\mathbf{v}_f + \mathbf{s}^i - \eta\,\mathbf{x}^k - \rho\,\mathbf{y}_k - \mathbf{z}_k\right) \ , \tag{168}$$

where $\mathbf{D}_{\rho,\eta} := \mathbf{D} + (\eta + \rho)\,\mathbb{I}_{n_d}$ denotes Delassus matrix regularized by the proximal point parameter and the ALM penalty. The third sub-problem in (167c) corresponds to the proximal projection onto the feasible set, that is realized using the Euclidean projector onto the cone $\mathcal{K}$, i.e.

$$\mathbf{y}^i = \operatorname{prox}^{\rho}_{\mathcal{K}}\left(\mathbf{x}^i - \rho^{-1}\mathbf{z}^{i-1}\right) = \mathcal{P}^{\mathcal{K}}\left(\mathbf{x}^i - \rho^{-1}\mathbf{z}^{i-1}\right) \ . \tag{169}$$

The last sub-problem (167d) corresponds to the update of the dual variable $\mathbf{z}$ of ADMM, and is akin to performing a simple gradient descent step with the penalty ρ acting as a step-size, and the *consensus error* between the primal variables acting as the gradient. Lastly, to extract ADMM-CCP from the aforedescribed derivation, we can simply neglect (167a), i.e. $\mathbf{s}^i = 0$, $\forall i$ and omit $\frac{\eta}{2}\|\mathbf{x} - \mathbf{x}^-\|_2^2$ from (166).

Convergence Criteria: The typical definition of ADMM involves a convergence criteria based solely on the primal-dual residuals r_p and r_d, as defined in (156) and (157), respectively. However, as ADMM-NCP is solving an NCP, it is crucial that (158) must also be evaluated in order to ensure that solutions satisfy the physical constraints of the problem. However, compared to the other splitting-based solvers, ADMM-NCP does not need to evaluate the costly computations described in Sec. VI-F, which would require several matrix-vector multiplications to evaluate. Instead, for ADMM-NCP and ADMM-CCP, we can evaluate r_p and r_d directly using the iterate values of the primal-dual variables $\mathbf{x}^i$, $\mathbf{y}^i$, $\mathbf{z}^i$ as

$$\mathbf{r}^i_p := \mathbf{x}^i - \mathbf{y}^i \tag{170a}$$

$$\mathbf{r}^i_d := \eta\,(\mathbf{x}^i - \mathbf{x}^{i-1}) + \rho\,(\mathbf{y}^i - \mathbf{y}^{i-1}) \tag{170b}$$

$$\mathbf{r}^i_{cp} := [\,\mathbf{x}^{i\,T}_j\ \mathbf{z}^i_j\,]_{n_l+n_c} \tag{170c}$$

Penalty Adaptation: The final component of ADMM-NCP involves the determination of the ALM penalty parameter ρ. Similarly to the description of splitting-based methods and how each variant elects to choose ρ, there several options exist for ADMM as well. In particular, we have realized three:

1) A fixed-penalty (FP), where it is set to a user-specified constant, i.e. $\rho = \rho^0 > 0$, which by default, we have found $\rho^0 = 1$ to work well across all problems.

2) A *Linear Adaptation* (LA) scheme in the form of

$$\rho^i = R_{LA}(\rho^{i-1}) := \begin{cases} \rho^{i-1}\,\tau_{inc} & , r^i_p/r^i_d \geq \alpha \\ \rho^{i-1}\,\tau_{dec} & , r^i_p/r^i_d \leq \alpha^{-1} \\ \rho^{i-1} & , \alpha^{-1} < r^i_p/r^i_d < \alpha \end{cases} \ , \tag{171}$$

where r^i_p and r^i_d are respectively the primal and dual residual of the current iteration, $\alpha > 1$ is the a threshold on the ratio of primal-dual residuals, and $\tau_{inc}, \tau_{dec} > 0$ are the linear increment/decrement factors. The latter essentially modulate the penalty parameter only when the ratio of primal-dual residuals exceeds the threshold α, attempting to maintain them both relatively close to each other. Typical values for these parameters are $\alpha = 10.0$ and $\tau_{inc}, \tau_{dec} = 1.5$.

3) Based on an analysis by Nishihara et al in [84] on the of the convergence properties of ADMM, Carpentier et al proposed a *Spectral Adaptation* (SA) scheme that both initializes and adapts the penalty ρ based on the spectral properties of the Delassus matrix $\mathbf{D}$, i.e. the conditioning of the problem. Denoting the largest and smallest eigenvalues of $\mathbf{D}$ respectively as $L := \lambda_{max}(\mathbf{D})$ and $m := \lambda_{min}(\mathbf{D})$, and subsequently the condition number as $\kappa = \kappa(D) =: L/m$, the SA scheme is

$$\rho^0 = \sqrt{L_0\,m_0}\,\kappa_0^{\tau_0} \tag{172a}$$

$$\rho^i = R_{SA}(\rho^{i-1}) := \begin{cases} \rho^{i-1}\,\kappa_0^{\tau} & , r^i_p/r^i_d \geq \alpha \\ \rho^{i-1}\,\kappa_0^{-\tau} & , r^i_p/r^i_d \leq \alpha^{-1} \\ \rho^{i-1} & , \alpha^{-1} < r^i_p/r^i_d < \alpha \end{cases} \tag{172b}$$

where $\tau_0, \tau > 0$ are the initialization and adaptation factors. Typical values for SA are $\alpha = 10.0$, $\tau_0 = 0.2$, $\tau = 0.05$. A final remark regarding the values of L, m in the SA scheme, is that it is not actually necessary to compute $m := \lambda_{min}(\mathbf{D})$, since η ensures that it will be its worst-case value. Thus we can always set $m = \eta$. Moreover, the value of L need not be exact, meaning that we can employ computationally cheap estimates that can be rendered using techniques such as the Power-Iteration method [85]. Sec. VII-B provides more details on the spectral properties L, m, κ and how they are determined by the properties of the system.

TABLE I
Dual Solvers: Definition of each solver considered in this evaluation.

Solver ID		Scheme	Projector	De Saxcé	Termination	Notes
PGS-CCP		SOR (Alg. 5)	CCP (Alg. 2)	yes	(156)(157)(158)	
PGS-NCP		SOR (Alg. 5)	NCP (Alg. 3)	no	(156)(157)(158)	
NBGS		SOR (Alg. 5)	NB (Alg. 4) + MDP	no	(156)(157)(158)	
RAISIM		SOR (Alg. 5)	NB (Alg. 4) + BS	no	(156)(157)(158)	
RAISIM-DS		SOR (Alg. 5)	NB (Alg. 4) + BS	yes	(156)(157)(158)	
RAISIM-DS-ES		SOR (Alg. 5)	NB (Alg. 4) + BS	yes	(160)	
ADMM-CCP		ADMM (Alg. 6)	Euclidean (150)	no	(156)(157)(158)	fixed penalty
ADMM-NCP-*	FP	ADMM (Alg. 6)	Euclidean (150)	yes	(156)(157)(158)	fixed penalty
	LA	ADMM (Alg. 6)	Euclidean (150)	yes	(156)(157)(158)	linear adapt. (171)
	SA	ADMM (Alg. 6)	Euclidean (150)	yes	(156)(157)(158)	spectral adapt. (172)

Algorithm 5 Projective Splitting-based Solver

Require: N_{max}, ω^0, ω_{min}, γ, ϵ_p, ϵ_d, ϵ_{cp} ▷ solver parameters
Require: $\mathbf{D}$, $\mathbf{v}_f$, $\mathcal{K}$, $\mathcal{P}(\cdot)$, $f_{stop}(\cdot)$ ▷ problem definition
Require: $\boldsymbol{\lambda}^0$, $\mathbf{v}^{+0}$ ▷ optional warmstart
▷ initialization

$\omega \leftarrow \omega^0$
for $i = 1$ to N_{max} **do** ▷ solver iteration

  for $j = 1$ to n_j **do** ▷ joint iteration
    $\bar{\mathbf{v}}_{f,j} \leftarrow \mathbf{v}_{f,j} - \sum_{m<j} \mathbf{D}_{jm}\, \boldsymbol{\lambda}_m^i - \sum_{m>j} \mathbf{D}_{jm}\, \boldsymbol{\lambda}_m^{i-1}$
    $\boldsymbol{\lambda}_{j,0}^i \leftarrow -\mathbf{D}_{jj}^{-1}\, \bar{\mathbf{v}}_{f,j}$
    $\boldsymbol{\lambda}_j^i \leftarrow (1-\omega)\, \boldsymbol{\lambda}_j^{i-1} + \omega\, \boldsymbol{\lambda}_{j,0}^i$
  end for

  for $l = 1$ to n_l **do** ▷ limit iteration
    $\bar{v}_{f,l} \leftarrow v_{f,l} - \sum_{m<l} \mathbf{D}_{lm}\, \boldsymbol{\lambda}_m^i - \sum_{m>j} \mathbf{D}_{lm}\, \boldsymbol{\lambda}_m^{i-1}$
    $\lambda_{l,0}^i \leftarrow -D_{ll}^{-1}\, \bar{v}_{f,l}$
    $\lambda_l^i \leftarrow \mathcal{P}^{\mathbb{R}_+}(\lambda_{l,0}^i)$
    $\lambda_l^i \leftarrow (1-\omega)\, \lambda_l^{i-1} + \omega\, \lambda_l^i$
  end for

  for $k = 1$ to n_c **do** ▷ contact iteration
    $\bar{\mathbf{v}}_{f,k} \leftarrow \mathbf{v}_{f,k} - \sum_{m<k} \mathbf{D}_{km}\, \boldsymbol{\lambda}_m^i - \sum_{m>k} \mathbf{D}_{km}\, \boldsymbol{\lambda}_m^{i-1}$
    $\mathbf{s}_i \leftarrow \boldsymbol{\Gamma}(\mathbf{D}_{kk}\, \boldsymbol{\lambda}_k^i + \bar{\mathbf{v}}_{f,k})$
    $\boldsymbol{\lambda}_k^i \leftarrow \mathcal{P}(\boldsymbol{\lambda}_k^i, \mathbf{D}_{kk}, \bar{\mathbf{v}}_{f,k}, \mathbf{s}_i, \mu_k)$
    $\boldsymbol{\lambda}_k^i \leftarrow (1-\omega)\, \boldsymbol{\lambda}_k^{i-1} + \omega\, \boldsymbol{\lambda}_k^i$
  end for

  $\mathbf{v}^+ \leftarrow \mathbf{D}\, \boldsymbol{\lambda}^i + \mathbf{v}_f$ ▷ update solution
  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}^i$

  if $f_{stop}(\boldsymbol{\lambda}^i, \mathbf{v}^i, \epsilon_p, \epsilon_d, \epsilon_{cp}) = \text{true}$ **then**
    break
  end if
  $\omega \leftarrow \min(\omega_{min}, \gamma\, \omega)$ ▷ relaxation decay

end for

return $\boldsymbol{\lambda}$, $\mathbf{v}^+$

Algorithm 6 Proximal ADMM-based Solver

Require: N_{max}, ρ^0, η, ω , ϵ_p, ϵ_d, ϵ_{cp} ▷ solver parameters
Require: $\mathbf{D}$, $\mathbf{v}_f$, $\mathcal{K}$, $\mathcal{P}^{\mathcal{K}}(\cdot)$ ▷ problem definition
Require: $\boldsymbol{\lambda}^0$, $\mathbf{v}^{+0}$ ▷ optional warmstart

$\mathbf{x}^0 \leftarrow \mathbf{y}^0 \leftarrow \boldsymbol{\lambda}^0$ ▷ initialization
$\mathbf{z}^0 \leftarrow \mathbf{v}^{+0}$

for $i = 1$ to N_{max} **do** ▷ solver iterations

  $\mathbf{s}^i \leftarrow \boldsymbol{\Gamma}(\mathbf{z}^{i-1})$ ▷ De Saxcé estimate

  $\mathbf{v}^i \leftarrow \mathbf{v}_f + \mathbf{s}^i - \eta\, \mathbf{x}^{i-1} - \rho^i\, \mathbf{y}^{i-1} - \mathbf{z}^{i-1}$ ▷ offset update
  $\mathbf{x}^i \leftarrow -\left(\mathbf{D} + (\eta + \rho^i)\, \mathbb{I}_{n_d}\right)^{-1} \mathbf{v}^i$ ▷ primal update
  $\mathbf{x}^i \leftarrow (1-\omega)\, \mathbf{y}^{i-1} + \omega\, \mathbf{x}^i$ ▷ over-relaxation
  $\mathbf{y}^i \leftarrow \mathcal{P}^{\mathcal{K}}\left(\mathbf{x}^i - \frac{1}{\rho^i}\, \mathbf{z}^{i-1}\right)$ ▷ slack update
  $\mathbf{z}^i \leftarrow \mathbf{z}^{i-1} - \rho^i\, \left(\mathbf{x}^i - \mathbf{y}^i\right)$ ▷ dual update

  $\boldsymbol{\lambda} \leftarrow \mathbf{y}^i$ ▷ update solution
  $\mathbf{v}^+ \leftarrow \mathbf{z}^i - \boldsymbol{\Gamma}(\mathbf{z}^i)$

  $r_p^i \leftarrow \|\mathbf{x}^i - \mathbf{y}^i\|_\infty$ ▷ residuals
  $r_d^i \leftarrow \|\eta\, (\mathbf{x}^i - \mathbf{x}^{i-1}) + \rho^i\, (\mathbf{y}^i - \mathbf{y}^{i-1})\|_\infty$
  $r_{cp}^i \leftarrow \|r_{cp}(\mathbf{x}^i\,,\ \mathbf{z}^i)\|_\infty$ ▷ from (158)

  if $(r_p^i < \epsilon_p) \wedge (r_d^i < \epsilon_d) \wedge (r_{cp}^i < \epsilon_{cp})$ **then**
    break
  end if

  $\rho^i \leftarrow R\left(\rho^i\,,\ r_p^i\,,\ r_d^i\right)$ ▷ from (171) or (172)

end for

return $\boldsymbol{\lambda}$, $\mathbf{v}^+$

VII. Benchmarking Framework

In this section we proposed a general-purpose framework for evaluating FD solvers. Fundamentally, our approach involves three core elements. Firstly, we have compiled an extensive suite of problems involving constrained rigid multi-body systems ranging from primitive toy problems to full-scale complex mechanical assemblies such as robotic systems and *Audio-Animatronics*® figures. Secondly, we define a scheme to categorize dual problems based on the relative constraint dimensionality and conditioning of each, which enables a systematic characterization of solver performance based on the problem difficulty. Third, the we prescribe a set of quantitative performance metrics based on first-principles that are indicative of solver accuracy, robustness and speed. The presented approach is greatly inspired by the past work of Acary et al in [86], [7], and is in effect, a direct extension of the former to cases of complex mechanical assemblies such as robotic systems. Within the scope of this work, we utilized it to only compare the specific set of algorithms described in Sec. VI, as will be presented in Sec. VIII.

A. Benchmark Suite

The suite of simulation problems, depicted in Fig. 8, consists of twelve constrained rigid multi-body systems, spanning a wide range of ill-conditioned and numerically challenging scenarios. The toy problems principally serve to establish performance baselines of each solver on the distinct types of ill-conditioning emerging in the more complex systems, and thus facilitate systematic analyses of their qualitative behaviors. The full-scale systems are used to asses how algorithm performance scales with problem dimensionality and conditioning directly in cases of immediate practical relevance. It is thus organized into three categories of increasing system complexity and difficulty, namely:

1) *Primitives*: a set of toy problems, each exhibiting a specific case of ill-conditioning to be tackled in isolation.
2) *Robotics*: a set of problems involving robotic systems that are of moderate to high difficulty. They are all articulated systems without kinematic loops, and collisions involve only geometric primitives or minimal-size convex meshes, as is typical in robotics applications.
3) *Audio-Animatronics*®: a set of complex mechanical systems, each exhibiting multiple kinematic loop-closures. While most are affixed w.r.t the world via a unary fixed-joints one is also free-floating. For these systems collisions are evaluated solely using the raw true mesh-based geometry of the mechanical assembly. The geometry is also multi-layered, as the internal mechanisms are enclosed by exterior *shells*. Thus potential collisions include combinations between the world, shells and internal components.

B. Characterizing Ill-conditioning

It is important to first define what exactly ill-conditioning is and how it arises in the context of mechanics. Generally speaking, given a function or operator defined by the mapping $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, we say that the former is *ill-conditioned*, if its so-called *condition number* $\kappa_f \in (0, \infty)$ exhibits large values, i.e. many orders of magnitude larger than unity. The precise definition of κ_f depends on the nature of f and the choice of metric norm used to measure distances in the domain and image of the mapping, but effectively, it is always some measure of the sensitivity of the input-output mapping $\mathbf{y} = f(\mathbf{x})$. In our case we will employ definitions pertaining to *convex functions* and *matrices*, which, are closely related through the theory of *monotone operators* [87].

Specifically, if the operator is a convex function $\mathbf{f} : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$, where $\bar{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$, the condition number of $y = f(\mathbf{x})$ is defined as $\kappa_f = L/m$, where $m > 0$ is the *convexity parameter* and $L > 0$ is the *Lipschitz constant* of f. See [87] for more details. Intuitively, m and L are respectively equivalent to the smallest and largest singular values of the Hessian $\nabla^2 f(\mathbf{x})$ if f were twice continuously-differentiable. This equivalence naturally leads to second definition regarding matrices: in the case of affine functions of the form $f(x) = \mathbf{A}\,\mathbf{x} + \mathbf{b}$, the equivalent quantity is the *matrix condition number* $\kappa_{\mathbf{A}} := \sigma_{max}(\mathbf{A})/\sigma_{min}(\mathbf{A})$, i.e. the ratio of its largest and smallest singular values $\sigma_{max}(\mathbf{A}) := L$, $\sigma_{min}(\mathbf{A}) := m$ of matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$. This also means that the Lipschitz constant corresponds to the spectral radius of $\mathbf{A}$, i.e. $L = \rho(\mathbf{A})$. Moreover, for symmetric matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$, i.e. $\mathbf{A} = \mathbf{A}^T$, $\kappa_{\mathbf{A}} := \lambda_{max}(\mathbf{A})/\lambda_{min}(\mathbf{A})$ where now the condition number can be defined using its largest and smallest eigenvalues.

With these definitions in hand, let us now consider how ill-conditioning arises within the mechanical context of rigid-body dynamics. The two key elements at play here are the Jacobian matrix $\mathbf{J}$ and the generalized mass matrix $\mathbf{M}$. Therefore, when either of the two (or both) exhibit large condition numbers, then the overall FD problem is said to be ill-conditioned, and can occur in cases of:

- *Hyperstaticity*: this corresponds to rank-deficiency in the Jacobian matrix $\mathbf{J}$ due to constraint coupling. It can occur in the following ways: (a) intrinsically due to kinematic loops occurring in the joint morphology, (b) extrinsically due to multiple contacts acting on the same body or on the system overall, and (c) when the system approaches a kinematic singularity. Rank-deficiency therefore amounts to $\mathbf{J}$ exhibiting one or more zero-valued eigenvalues, i.e. $\sigma_{min}(\mathbf{A}) = 0$, resulting in $\kappa \rightarrow \infty$.
- *Inertial disparity*: this is when the system's mass ratio $r_m = m_{max}/m_{min} \gg 1$, i.e. that of the largest and smallest mass present in the system is significantly larger than unity. How much larger than unity is of course relative. Practically speaking, and in terms of orders-of-magnitude, a forward dynamics problem can be ill-conditioned if $r_m \approx 10^3$. Although $\mathbf{M}$ is necessarily positive-definite, its inversion with a high r_m multiplied by the effective lever-arms of $\mathbf{J}$ can result in exceptionally large $\kappa_{\mathbf{D}}$, the condition number of the Delassus matrix.

Therefore, the combined effects of hyperstaticity and inertial disparity can prove detrimental to both the rate of convergence of dual solvers as well as the stability of the resulting constraint reactions. The former leads to excessive iterations being

required to solve the problem, while the latter causes erratic changes to the forces rendered between successive iterations without substantial changes to the system's state.

C. Sample Categories

Given the aforedescribed definition of ill-conditioning, we aim to systematically characterize and categorize time-wise samples extracted from each problem in the suite. In this context, a *sample* is defined as a single dual problem in the form of the NCP (15) or CCP (18). Each sample can therefore be solved by a set of solver instances varying in terms of the algorithm or hyper-parameters. This aspect of our work has been inspired by the tremendously thorough technical reports of Acary et al on the FCLib [86] and numerical solver comparisons in [7], as well as that of Ledic et al in their comparisons of contact models in [38].

Employing a similar categorization as in [7], we characterize each sample problem based on so-called *constraint densities* $d_j = n_{jd}/n_{bd}$, $d_c = 3\,n_c/6\,n_b$, $d_{jlc} = n_d/6\,n_b$ corresponding to that of the joints, contacts and total constraint sets, respectively, where $n_{jd} := \sum_{j=1}^{n_j} m_j$ denotes the total number of constraint dimensions of the joints. These densities express the ratio of number of constraints to body DoFs, and whenever they are greater than unity, express the *potential* for rank-deficiency of $\mathbf{J}$. For this reason we also measure the actual matrix rank $r_{\mathbf{J}} = \text{rank}(\mathbf{J})$ of the system Jacobian. Note that we have omitted the isolated constraint density of joint limits as it, by definition, cannot exceed the joint constraint density d_j, and so does not really contribute much information on its own. It is, however, included in the total constraint density measurement d_{jlc}. Thus, using these densities and marix rank, we categorize each sample as either:

- Independent Joints: $n_c = 0$, $n_j \geq 1$, and $r_{\mathbf{J}} = n_{jd}$.
- Redundant Joints: $n_c = 0$, $n_j \geq 1$, $r_{\mathbf{J}} < n_{jd}$, and $d_j < 1$
- Dense Joints: $n_c = 0$, $n_j \geq 1$, $r_{\mathbf{J}} < n_{jd}$, and $d_j \geq 1$
- Single Contact: $n_c = 1$
- Sparse Contacts: $n_c \in [2\,,\ 2\,n_b]$ (i.e. $d_c < 1$)
- Dense Contacts: $n_c > 2\,n_b$ and $d_c > 1$
- Dense Constraints: $n_c \geq 1$, $n_j \geq 1$ and $d_{jlc} > 1$

The first case of *Independent Joints* is used to establish a baseline of nominal performance for samples that can admit a full-row rank $\mathbf{J}$. Note also the special case of *Single Contact* $n_c = 1$. Although seemingly trivial, for problems with large mass ratios and/or large number of joints, the single-contact case can inform us about the capacity of a particular solver to correctly propagate and distribute constraint reactions throughout the system.

D. Performance Metrics

Selecting appropriate performance metrics is crucial to evaluating dual solvers w.r.t their physical plausibility, accuracy, speed, and robustness, both in relative and absolute terms. Moreover, our objective must be to state such comparisons as fairly as possible, but also in a manner that is representative of the actual needs of relevant applications. Overall, our approach for evaluating dual solvers consists of a comparative framework that combines and extends the work of Acary et al [7] and Lidec et al [38], [32]. Specifically, it involves two aspects: a) a set of sample-wise quantitative performance metrics, and b) a schema for representing aggregate performance using statistical measures computed over multiple samples.

Accuracy: We will first define metrics for physical plausibility and accuracy. In this regard, comparing dual solvers in systematic manner is a rather non-trivial task. As the algorithms can differ significantly in terms of the employed numerical scheme and the contact model, this precludes the use of their respective convergence criteria as comparative measures. Therefore, we must resort to using metrics which can generalize to all cases, e.g. a CCP or NCP-based contact model etc. In particular, we propose to use:

1) the *Penetration Residual*

$$r_{\text{pen}} =: \|\,\mathbf{f}(\mathbf{s})\,\|_\infty\ , \tag{173}$$

where $\mathbf{f} : \mathbb{S}^{n_b} \rightarrow \mathbb{R}^{n_{sd}}$ is the total configuration-level constraint function defined in (84). This metric effectively indicates the worst-case configuration-level constraint violation occurring as a byproduct of the computed constraint reactions.

2) the *NCP Dual Residual*

$$r_{\text{dual}} =: \|\mathbf{r}_{\text{d}}(\mathbf{v}^+ + \boldsymbol{\Gamma}(\mathbf{v}^+))\|_\infty\ . \tag{174}$$

This quantity is useful because it indicates if non-zero velocities are present along directions that violate the constraint. For joints this simply amounts to the total constraint-space velocities, which should ideally be zero at all times. For limits and contacts, it corresponds to non-zero velocities along the unilateral constraint dimension, e.g. the contact normal. A proof of this property is provided in Appendix. A.

3) the *NCP Complementarity Residual*

$$r_{\text{ncp}} =: \|\mathbf{r}_{\text{cp}}(\boldsymbol{\lambda}\,,\ \mathbf{v}^+ + \boldsymbol{\Gamma}(\mathbf{v}^+))\|_\infty\ , \tag{175}$$

which serves as a measure of adherence to the MDP. Essentially, non-zero values indicate a misalignment between constraint reactions and corresponding contact velocities along the tangential planes, i.e. the deviation from maximal dissipativity, as well as erroneous power distributions along contact normals.

4) the *NCP Natural-Map Residual*

$$r_{\text{nat}} =: \|\mathbf{F}^{\text{nat}}_{\text{NCP}}(\boldsymbol{\lambda}, \mathbf{v}^+)\|_\infty\ , \tag{176}$$

which indicates how near the computed contact reaction is to a solution of the NCP. As described in Sec. VI-D, it combines the effects of the primal, dual and complementarity residuals into a single value, providing a convenient short-hand for both solver convergence and solution validity.

Speed: Next, we will define a set of metrics for assessing the speed and convergence behavior of each algorithm. These will serve as proxies to estimating the effective computational

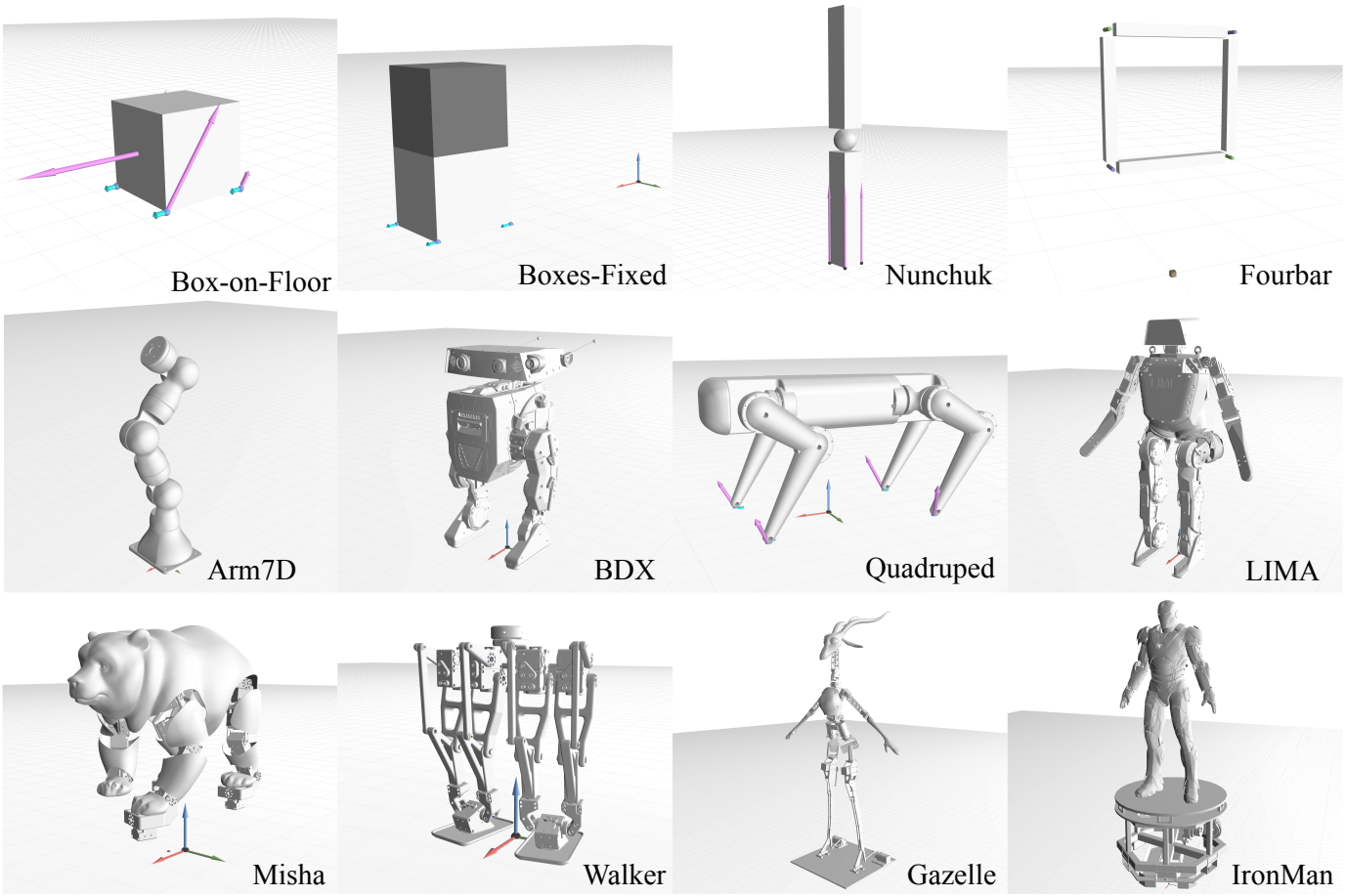

Fig. 8. *Benchmark Suite*: a set of twelve constrained rigid multi-body systems spanning three categories. The top, middle and bottom rows respectively depict the *Primitive*, *Robotics*, and *Audio-Animatronics®* problem categories, and the system complexity is increasing from left to right. The first category contains a set of toy problems that are often used to establish a baseline of behavior in specific cases of ill-conditioning. The second category involving robotic systems serves to evaluate solver performance on well established problems in encountered in robotics applications. These problems do not exhibit any inherent ill-conditioning, but the presence of joint limits and contacts, in addition to the use of meshes as the collision geometry, significantly increase the difficulty of simulating them. The third category involves mostly fixed-bases systems with the exception of Walker which is floating-based. All of these problems exhibit multiple intrinsic kinematic loops with a relatively large number of bodies, and present the biggest challenge, as the systems are inherently ill-conditioned due to the combined effects of: mesh collisions, inherent kinematic loops, passive (i.e. unactuated) joints, presence of unary joints, significantly larger problem dimensions, and large disparity in body masses.

TABLE II
Benchmark Suite: The intrinsic dimensions of each system, i.e. without active limit and contact constraints.

System	Base	Bodies	Joints	Total DoFs	Passive DoFs	Actuated DoFs	Constraints	Mass Ratio
Box-on-Plane	Free	1	0	6	0	0	0	1.00
Boxes-Fixed	Free	2	1	6	0	0	6	1000.00
Nunchuk	Free	3	2	12	6	0	6	1.00
Fourbar (fixed)	Fixed	4	5	1	0	1	26	1.00
Fourbar (free)	Free	4	4	7	5	1	20	1.00
Arm7D	Fixed	8	7	7	0	7	41	19.54
Quadruped	Free	13	12	18	6	12	60	29.07
BDX	Free	15	14	14	6	14	70	42.49
LIMA	Free	21	20	26	6	20	100	321.39
Misha	Fixed	22	24	23	0	23	121	90.67
Walker	Free	31	36	42	18	24	180	99.55
Gazelle	Fixed	38	45	60	38	22	214	1012.54
IronMan	Fixed	54	67	100	80	20	304	25544.44

complexity exhibited by each solver on the set of problems:

1) the *Number of Iterations* $i_{stop} \leq N_{max}$, which reflects a solver's convergence to a solution. $N_{max} \in \mathbb{N} \cup \{\infty\}$ is the maximum iterations permitted for a particular simulation, and can represent the computational budget afforded to a dual solver in terms of the number of operations executed on each problem.
2) the *Total Solve Time* $t_{solve} \leq t_{max}$, which represents the total computational cost incurred by a solver on a particular problem. $t_{max} \in \mathbb{R}_+ \cup \{\infty\}$ is the maximum permissible time-duration, and equivalently to N_{max}, can represent the computational budget in terms of the absolute wall-clock time.
3) the *Mean Iteration Time* t_{iter}, which approximates the computational cost of the per-iteration operations of a particular algorithm. Given the similar overall structure of all dual solvers, t_{iter} is indicative of the computational cost incurred by the specifics of each algorithms, such as the numerical scheme, the employed contact model and the residuals used as termination criteria.

Robustness: The last aspect of solver performance to consider is robustness. In this regard, we will asses how a solver behaves in the presence of ill-conditioning. Specifically, for each sample problem, our goal is to examine if a solver can: (a) converge at all, (b) converge within a specified computational budget, (c) produce stable motions despite not converging within the afforded budget, and (d) diverge by resulting in `NaN`, `Inf` or arbitrarily large values. To this end, two methods often employed in benchmarks for numerical optimization methods is to evaluate whether $t_{solve} \geq t_{max}$ or $i_{stop} \geq N_{max}$. If either case holds on a given problem, we can consider the solver to have failed to solve it. In this work we will exclusively employ the latter check w.r.t $i_{stop} \geq N_{max}$, as our evaluation focuses mainly on the numerical properties of each algorithm, as opposed to their absolute time complexity.

Profiles: Lastly, we employ the methodology of *performance profiles* by Dolan et al [88] in order to statistically evaluate the relative performance of each solver, w.r.t the aforedefined metrics. Such an approach has nowadays become standard practice for benchmarking numerical optimization algorithms, and is slowly gaining favor in application domains such as physical simulation as well [7], [32].It enables us to aggregate performance metrics collected over individual or multiple simulations, and render cumulative probability distributions that highlight the relative strengths and weaknesses of each solver/algorithm in a systematic way.

In this setting, we assume a set $\mathcal{P}$ consisting of $n_{dp} := |\mathcal{P}|$ dual problems, as well as a set $\mathcal{S}$ of $n_{ds} := |\mathcal{S}|$ dual solvers. For each index pair $p, s \in \mathcal{P} \times \mathcal{S}$, we execute solver s on problem p and collect the set of metrics $\mathcal{M}_{p,s} := \{r_{pen}\ ,\ r_d\ ,\ r_{ncp}\ ,\ r_{nat}\ ,\ i_{stop}\ ,\ t_{solve}\ ,\ t_{iter}\}$ defined previously. It is important to note that all performance metrics in $\mathcal{M}_{p,s}$ admit an increasing partial ordering, i.e. lowest is best, and it holds that $m_{p,s} \geq 0\ ,\ \forall m_{p,s} \in \mathcal{M}_{p,s}$. For every metric sample $m_{p,s}$ we compute a corresponding relative *performance ratio*

$$r_{p,s} := \frac{m_{p,s}}{\max\{m_{p,s} : s \in \mathcal{S}\} + \epsilon}\ ,\ r_{p,s} \in [1\ ,\ r_M]\ , \tag{177}$$

where $m_p^* = \max\{m_{p,s} : s \in \mathcal{S}\}$ is the *baseline* metric for problem p, i.e. the performance metric of the best performing solver for the specific problem. ϵ is an infinitesimal constant added to the denominator to protect against divisions by zero, and r_M is a parameter that we can set ad-hoc to represent the ratio corresponding to failure to converge, e.g. $r_M = \infty$.

A *performance profile* of each solver s w.r.t a specific metric is defined as the cumulative distribution function

$$\rho_s(\tau) := \frac{1}{n_p}\ \text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}\ , \tag{178}$$

over values τ of the performance ratio. Essentially, each value $\rho_s(\tau)$ expresses the probability that solver $s \in \mathcal{S}$ is withing a factor τ of the best performing solver over all problems in the set $\mathcal{S}$. Two important limit cases are to compare the values $\rho_s(1)$ and $\rho_s(r_M)$ of each solver. The former represents the number of problems in which s exhibited the best performance, and the latter indicates the number of problems that s solved successfully (i.e. converged in).

However, there lies a crucial subtlety in computing performance ratios for samples that can contain values smaller than the machine precision $\epsilon(d)$[11], such as when $m_{p,s}$ corresponds to one of the aforedescribed residuals. Using (177) directly in such cases may result in divisions with zero or arbitrarily minuscule values that can skew or completely distort the generation of each performance profile using (178). To overcome this problem, we utilize the augmentation proposed by Dingle & Higham [89], which replaces each sample as

$$\bar{m}_{p,s} = \begin{cases} m_{p,s} & ,\ m_{p,s} > m_{min} \\ m_{p,s}\ \frac{m_{max} - m_{min}}{m_{min}} + m_{min} & ,\ m_{p,s} \leq m_{min} \end{cases} \tag{179}$$

where $m_{max} = 0.5 \cdot \epsilon(d)$ and $m_{min} = 10^{-2} \cdot \epsilon(d)$ are the upper and lower limit metric values. Using (179) essentially re-normalizes values approaching $\epsilon(d)$ while preserving the strict ordering of samples and thus the monotonicity of each performance profile (178).

VIII. Experiments

This section presents a series of experiments designed to evaluate the algorithms described in Sec.VI using the benchmark suite defined in Sec. VII. The primitive problems will be used to analyze the behavior of the dual solvers and determine *if*, *when* and *how* they can exhibit characteristics required for physically plausible simulations. The full-scale problems will be used to extract performance statistics and quantify the relative differences between solvers at a larger scale.

A. Setup

The software facilitating our evaluation is implemented entirely in C++17 and employs the Eigen3 [90] library as a front-end for all BLAS operations. One aspect that is crucial

[11] For 64-bit double-precision floating-point, $\epsilon(d) = 2^{-52} \approx 2.2 \cdot 10^{-16}$.

to any numerical method is whether the problem data is represented using *sparse* and/or *dense* matrices and vectors. In our current implementation we have only used dense representations as our focus lies mainly in the evaluation of the numerical properties of the relevant algorithms. We do, however, plan to extend our implementation to sparse algebra in future work that will include other runtime performance enhancements as well. All results presented herein were executed on two distinct hardware set-ups. The first is a high-end desktop PC with an Intel Core i9-14900K 32-core CPU and 128GB of RAM, running Ubuntu 20.04. The code in this setup uses Intel MKL as the BLAS backend to Eigen3, and is compiled with `GCC 10` (`Ubuntu 10.5.0-1ubuntu1 20.04`). The second setup is a 2023 MacBook Pro with an Apple M3 Max CPU and 64GB RAM running macOS Sonoma 14.6, and uses with apple-specific `clang 15` (`arm64-apple-darwin-23.6.0`). Having both set-ups allows us to evaluate the potential disparity induced by hardware and compiler-specific quirks in the numerical output of all solvers.

B. A Note on Collision Detection

In all of the presented experiments, the contact configuration is *never assumed*, and in all cases, is generated based on the configuration of the system at the begging of each time-step. Regarding CD with geometric primitives and meshes, we use the library provided by the ODE [57]. In addition, we found it necessary to perform a post-processing of the set of generated contact points. Firstly, we established a robust grouping of geometries in order to control broad-phase collision checks between neighboring bodies, i.e. those connected by joints, and to be able to toggle internal collisions within the rigid-body system. Second, in the near-phase check, we cull redundant points that lie within a minimum distance of each other. This was particularly crucial to handling collisions involving mesh-based geometries, where potentially hundreds of points can be generated for each body without culling. Lastly, in order to deal with artifacts such as *contact chattering*, we added a mechanism to set a configurable collision detection margin that facilitates a penetration tolerance between bodies. Unlike other CD libraries such as FCL [19] and Coal [21] (a.k.a. HPP-FCL), ODE does not inherently support collision margins.

C. Primitive Problems

The first phase of experiments presents a fine-grained analysis of the set of solvers defined in Sec.VI on the subset of primitive problems. The objective of this phase is to examine how each solver behaves in the presence of certain types of ill-conditioning, and observe the physical artifacts that can manifest. All primary experiments involve independent simulations executed with each solver, for the purposes of observing the distinct motions they generate. These will be referred to as *independent runs*. However, in cases where performance profiles will be used to quantify relative performance between solvers, we necessarily need to compare them on the same data, i.e. identical dual problems. In such cases we use ADMM-NCP as a reference solver to generate a sequence of problems that are then re-solved by all others. We refer to these as *curated runs*. All solvers are by default configured for *high-precision*; $N_{max} = 10^4$ and $\epsilon_{abs} = 10^{-12}$ used for all residuals. In certain cases, which will be stated explicitly, we will use *high-throughput* settings with $N_{max} = 10^3$ and $\epsilon_{abs} = 10^{-6}$. Unless otherwise specified, all contact interactions will involve coefficients of friction $\mu_c = 0.7$ and restitution $\epsilon_c = 0$, and default values for constraint softening are $cs_j = \{d_0 = 0.9,\, d_w = 0.95,\, w = 0.001,\, m = 0.5,\, p = 2.0,\, T = 0.02,\, \beta = 1.0\}_j$

***Box-on-Plane**:* This first experiment involves a box placed on an a flat plane. A similar setup has been demonstrated in other works, such as those of Drumwright et al [91] and Lidec et al [38]. The box is a single cuboid body with mass $m = 1\,\mathrm{kg}$, sides of dimension $d = 20\,\mathrm{cm}$. The objective here is to observe: (a) the distribution of reaction forces among the $n_c \leq 4$ contacts, (b) stick-slip transitions, and (c) the overall motion of the body. The box is initialized resting undisturbed on the plane for the first two seconds. An external force is applied thereafter about the box's CoM along the global X-axis, with a linearly increasing magnitude of $0\,\mathrm{Nm}$ to $f_{max} = 2\,\mu_c\, m\, g\,\mathrm{Nm}$ over a duration of $\Delta t_{ext} = 6\,\mathrm{s}$. The force is removed at $t = 8\,\mathrm{s}$ and the box is allowed to decelerate due to friction until it comes to rest, where it is once again is left undisturbed. The entire procedure lasts a total of $t_f = 10\,\mathrm{s}$. The experiment is executed over two distinct run. In the first run, the dual problem is configured for hard contacts, and in the second, it is configured with constraint softening enabled.

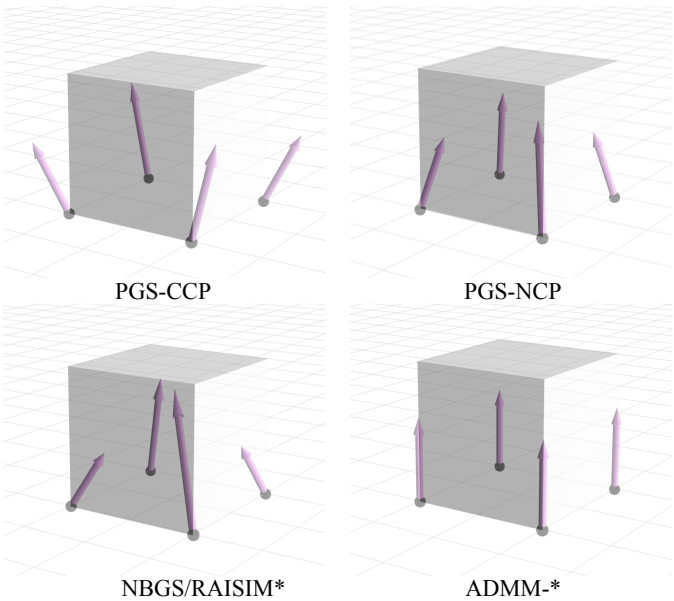

Fig. 9. The different force distributions exhibited by each solver on the hard-contact problem. Only the ADMM solvers yield a minimum-norm solution that evenly distributes contact forces without exhibiting internal forces, while all splitting-based solvers result in uneven and internal forces. This is expected however, since the per-contact iteration scheme is known to suffer from a lack of symmetry [17]. This artifact is due to the order in which the constraints are accessed, which, inevitably induces bias in the local solutions, especially when using a cold-start. Conversely, ADMM can account for all couplings simultaneously and eliminates internal forces.

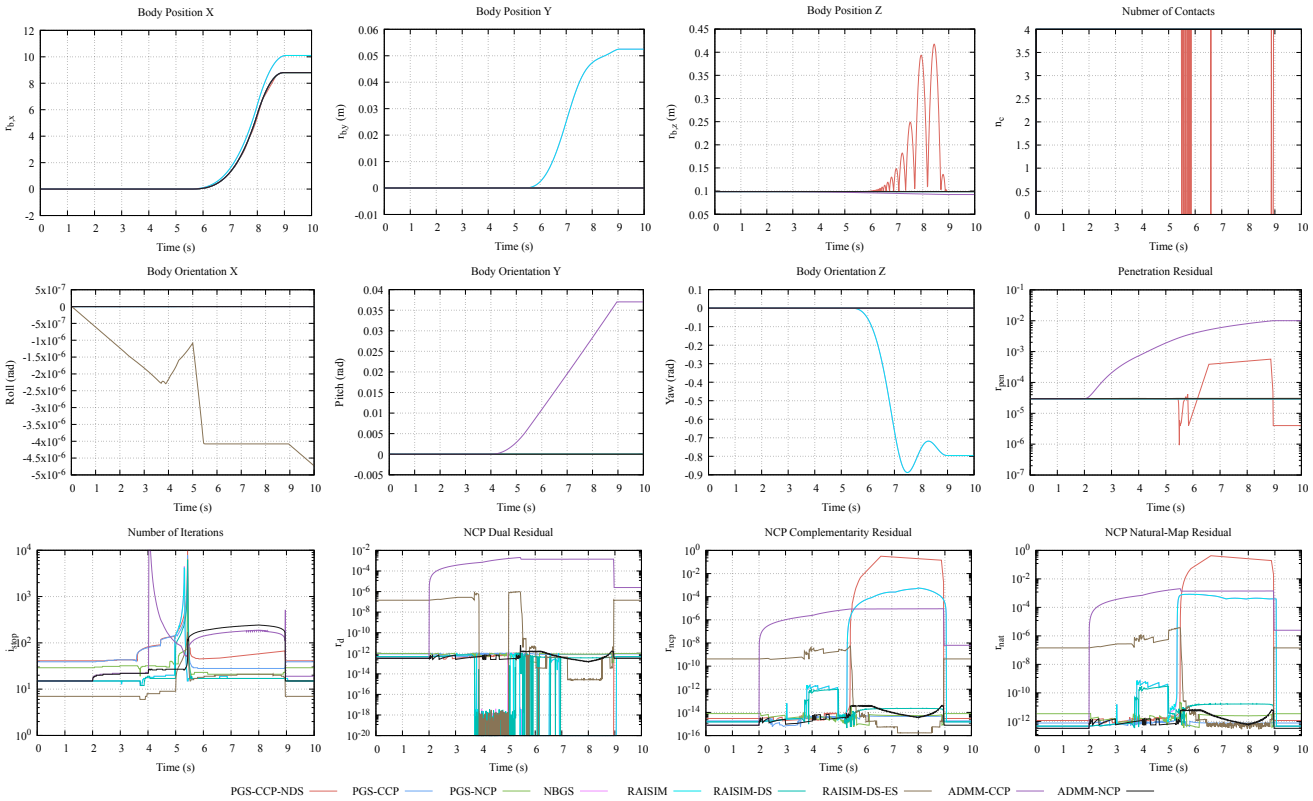

Fig. 10. *Box-on-Plane*: Time-series plots of the first run on the hard-contact problem. The PGS-CCP, PGS-NCP, RAISIM-DS and ADMM-NCP solvers yield similar motions, while the others exhibit significant translational and rotational deviations along the Y and Z-axes. When sliding, the PGS-CCP-NDS results in positive normal velocities thus leading to jumping, as can be seen in the right middle plot (red).

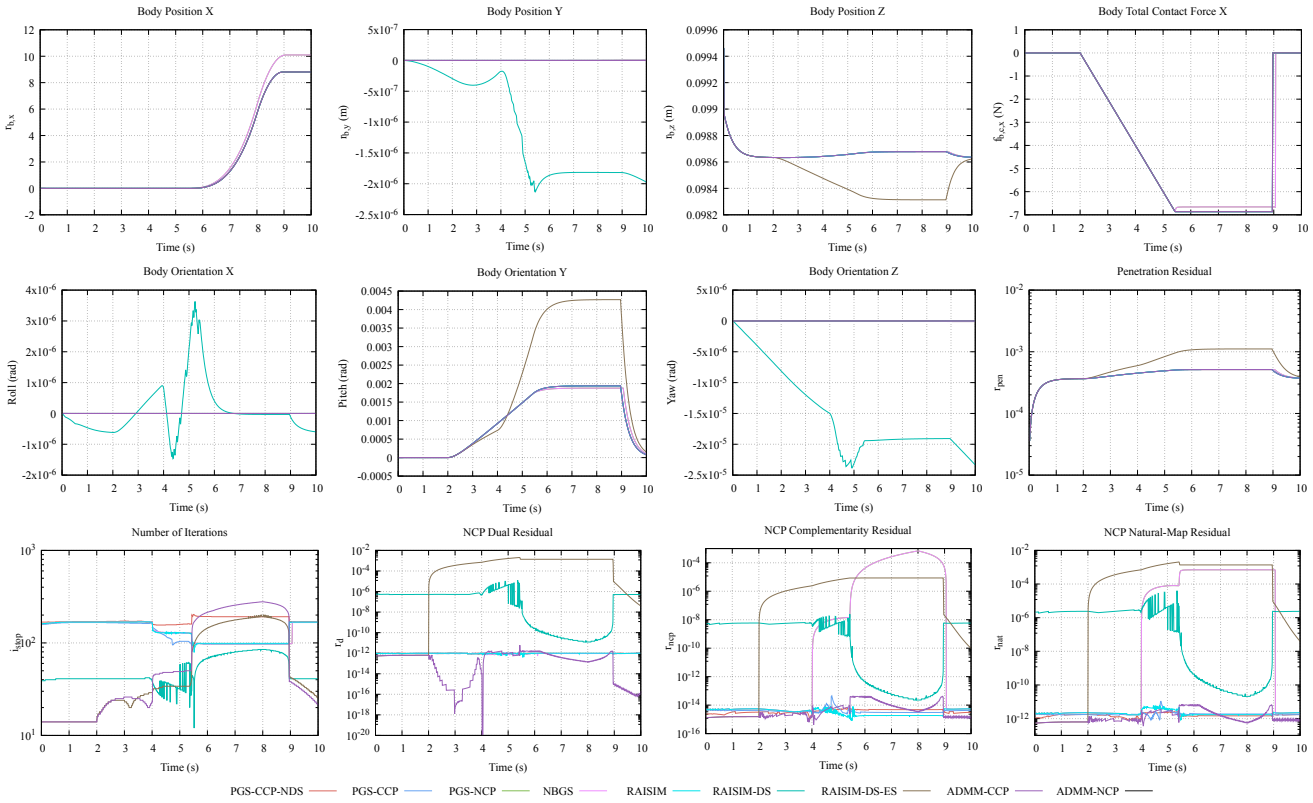

Fig. 11. *Box-on-Plane*: Time-series plots of the second run with MuJoCo-like constraint softening enabled. All solvers render minimum-norm contact forces and the motions exhibit significantly less deviations due to the improved symmetry of the former. NBGS and RAISIM still overshoot the trajectory of the other solvers due to the forces not being maximally dissipative.

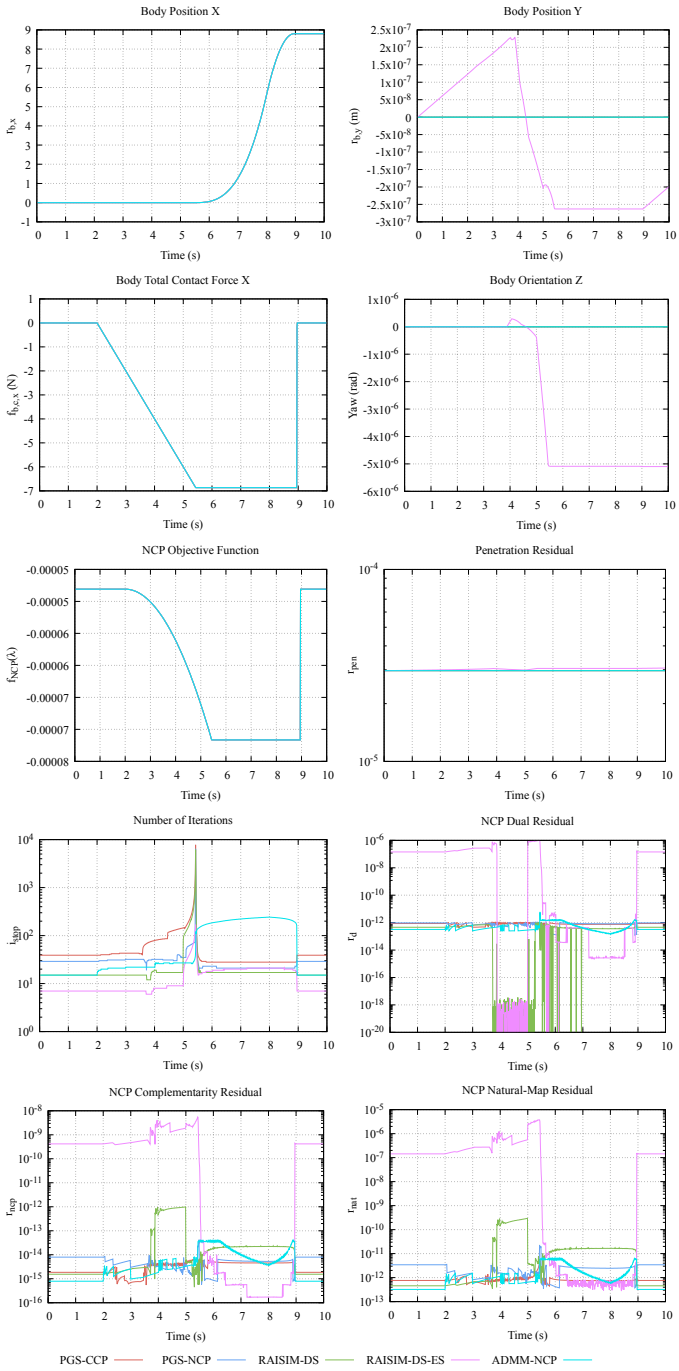

Fig. 12. *Box-on-Plane*: A comparison focusing on the NCP-type solvers. While the overall motions, forces and optimization objectives generated by all solvers can seem very similar, a detailed view of the NCP dual, NCP complementarity and NCP natural-map residuals can exemplify the subtle differences in their solutions.

The first run highlights the differences in the generated forces and motions. Fig. 9 depicts the contact force distributions rendered by each solver while the box is unforced, and Fig. 10 presents time-series data over the interval $[0, t_f]$. For this run only, we also include PGS-CCP-NDS, a version of PGS-CCP without the De Saxcé correction. The results indicate that on this particular problem, the motion of the box is nearly identical for all NCP-type solvers, i.e. PGS-CCP, PGS-NCP, RAISIM-DS, and ADMM-NCP, while all others exhibit significant discrepancies. To complement Fig. 10, we also provide an isolated comparison of the NCP-type solvers in Fig. 12. ADMM-CCP is similar to ADMM-NCP, but the absence of the De Saxcé correction leads significant underestimation of the normal contact forces when the external force is applied, leading to a gradual sinking of the box as it slides. Although the force distributions of PGS-NCP and RAISIM-DS are uneven while the contacts are sticking, when sliding occurs they do not incur any translation along the Y-axis or rotation about the Z-axis. Contrary to this, the NBGS and RAISIM solvers do not render maximally dissipative frictional forces, as the respective motion trajectories overshoot those of the NCP-type solvers. Moreover, they result in translational and rotational deviations along the Y-axis and Z-axis, respectively.

The PGS-CCP-NDS solver results in contact velocities with positive vertical components that lead to a jumping-like behavior. This artifact is a very clear demonstration of the effects of the CCP model, where duality is being enforced between contact reactions and velocities directly. Interestingly though, despite the jumping, the forces seem to be maximally dissipative, as the overall motion along the X and Y axes remains close to that yielded by NCP-type solvers. Comparatively, the inclusion of the De Saxcé correction in PGS-CCP proves indeed capable of eliminating the jumping artifacts, and results in the same overall behavior as the other NCP-type solvers. Also noteworthy is the fact that the forces rendered by PGS-CCP at rest point outwards as opposed to all other splitting-based solvers which point inwards[12].

The second run demonstrates the effects of the regularization induced by constraint softening, and is summarized in Fig. 11. While the box is unforced, the softening leads to virtually the same minimum-norm force solution as that of the ADMM solvers. Similarly to the first run, the NBGS and RAISIM solvers fail to yield maximally dissipative frictional forces and thus overshoot the NCP-type motion trajectories. PGS-CCP-NDS once again leads to jumping but now the motion becomes even more erratic, causing rolling and even larger deviations along the Y-axis. For this reason we have excluded it from the comparisons presented Fig. 11. Interestingly, the NCP-type exhibit nearly identical trajectories as when configured for hard-contacts. This outcome indicates that constraint softening is indeed capable of producing a very near approximation of the NCP solution, given an appropriate selection of the parameters described in Sec. V-D.

In addition, we executed an auxiliary curated run to directly highlight the relative performance between solvers on this problem. Performance profiles resulting from this are presented in Fig. 13, and exemplify several subtleties in comparing solvers. First, they demonstrate that all performance

[12]This result is quite counter-intuitive; at rest, the only term influencing the contact reactions is the gravitational force acting upon the box's CoM, which, via the lever-arms defined between the CoM and the contact positions defines the global free-velocity $\mathbf{v}_f$. From the perspective of each local problem, the gravity-induced wrench would cause outward sliding in the absence of all other contact forces. This means that when using a cold-start, in the first iteration we would expect the forces to point inwards opposing the sliding velocity, which would then be corrected by the increasing influence of the other contacts over multiple solver iterations. This is exactly the behavior observed for PGS-NCP, NBGS and RAISIM solvers, but not for PGS-CCP.

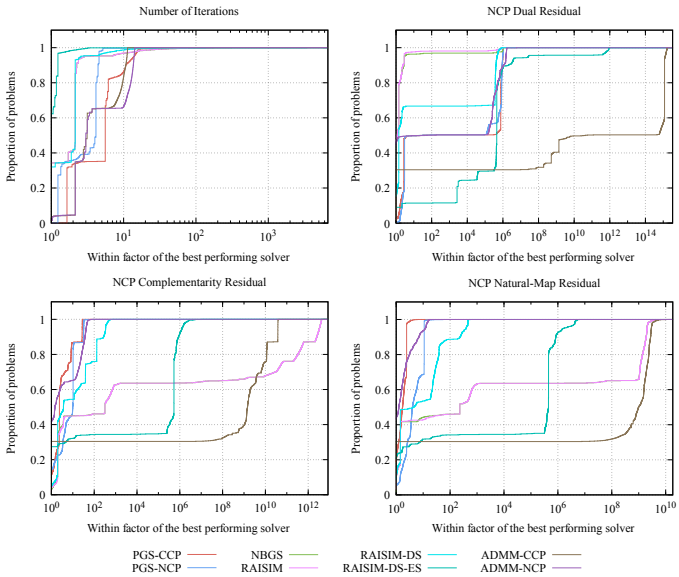

Fig. 13. *Boxes-on-Plane*: Performance profiles showing the relative performance between solvers. While ADMM-NCP demonstrates the best behavior and performance overall, it is still noteworthy that PGS-CCP, PGS-NCP and RAISIM-DS come very close in terms of the accuracy metrics r_{ncp} and r_{nat} (bottom row). Interestingly, NBGS and RAISIM exhibit better performance overall in terms of r_{dual} (top right), which is consistent with the fact that they demonstrated the least penetration error. However, their significantly reduced performance in terms of r_{ncp} and r_{nat} demonstrates that all accuracy metrics must be taken in to account simultaneously in order to summarize the overall accuracy of a solver.

metrics from Sec. VII-D provide distinct and equally important views of constraint satisfaction. Case in point, the NCP dual and natural-map residuals are equally as important metrics as NCP complementarity. Second, the performance profiles can directly quantify how similar the NCP-type solvers actually are. Fig. 13 shows that PGS-CCP, PGS-NCP, RAISIM-DS and ADMM-NCP differ by less than an order of magnitude to each other on all accuracy residuals, when compared head-to-head on the same problems. These differences are not so apparent in the time-series data of the independent runs.

***Boxes-Fixed**:* The second experiment involves a constrained system with a large mass-ratio being dropped and then dragged on a flat plane. Specifically, the system consists of two boxes rigidly attached via a fixed joint, where the bottom box has mass $m_b = 0.1\,\mathrm{kg}$ and the top $m_t = 10^3\,\mathrm{kg}$ resulting in a mass-ratio of $r_m = 10^4$. Both boxes have sides of dimension $d = 20\,\mathrm{cm}$. A similar setup has been described by Erleben in [92] and Lidec in [38], but they stacked the two unequally weighted boxes atop one another as opposed to using a fixed joint. We chose the jointed configuration as our focus mainly lies in the interaction between joints and contacts. Note that collision detection was necessarily disabled between the two bodies since they were jointed. The objective here is to observe the combined effects of inertial-disparity and rank-deficiency on solver convergence. We executed three runs with all solvers set to high-precision: (a) start at rest and apply a linearly increasing force, (b) drop the system from an initial height of $z_b = 0.5\,\mathrm{m}$ of the bottom box, and (c) combine both scenarios (a) and (b) and enable constraint stabilization

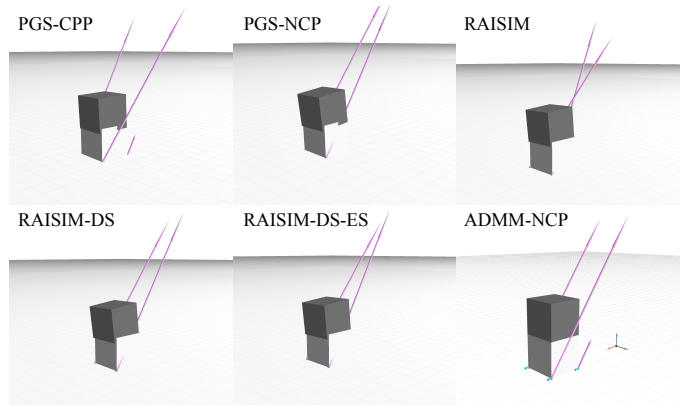

Fig. 14. *Boxes-Fixed*: Examples of the constraint violation accumulated by each solver as well as the forces rendered. The translational deviation of the box causes a shifting of the overall CoM of the system that leads to the rear contacts opening while sliding.

with $\alpha_j, \gamma_k = 0.1$. Note that ADMM-CCP was excluded from this experiment as it often diverged during our initial testing.

In the first run, the setup is similar to that described in Sec. VIII-C, in that the external force applied to the bottom box is ramped-up from $0\,\mathrm{Nm}$ to $f_{max} = 1.01\,\mu_c\,(m_b + m_t)\,g\,\mathrm{Nm}$[13] over a duration of approximately $\Delta t_{ramp} = 2\,\mathrm{s}$, and held at the maximum value for another $\Delta t_{max} = 6\,\mathrm{s}$. Once the external force is removed, the system is left to decelerate and rest over a final period of $\Delta t_{free} = 2\,\mathrm{s}$, with and thus total duration of the experiment is $t_f = 12\,\mathrm{s}$. The time-series plots of this run is shown in Fig. 15. Importantly, we observed that all solvers failed to converge within N_{max} iterations. However, despite the lack of convergence, ADMM-NCP was still able to exhibit the expected behavior. All others where able to initially render correct constraint reactions, but with the application of the external force, they underestimated the joint reactions leading to the top box drifting away from the nominal pose relative to the bottom box. A screen capture showing this effect is depicted in Fig. 14. Thus, one important observation from this run is that, for practical applications, solvers can still output usable constraint reactions despite a lack of convergence. This is a somewhat a trait of robustness, since non-convergence does not mean that forces are infeasible. However, this also demonstrates how choosing an appropriate value for N_{max} forces us to make a compromise between accuracy and speed.

In the second run, the system is dropped from an initial height to evaluate the capacity of the solvers to correctly propagate impact forces through the joints. The corresponding time-series plots are shown in Fig. 16. Despite the large mass-ratio, constraint violation was initially minimal after the impact for all solvers. The splitting-based solvers, however, accumulated significant joint constraint violation over time. Of particular note is that NBGS and the variants of RAISIM, all were able to keep contact penetration values even smaller than those of ADMM-NCP. This demonstrates the efficacy of using a holistic local contact model and solver such as (152), as opposed to those of PGS-CCP and PGS-NCP.

[13]The 1.01 factor in the computation of the maximum external force is used to induce slipping by only marginally exceeding the stiction force.

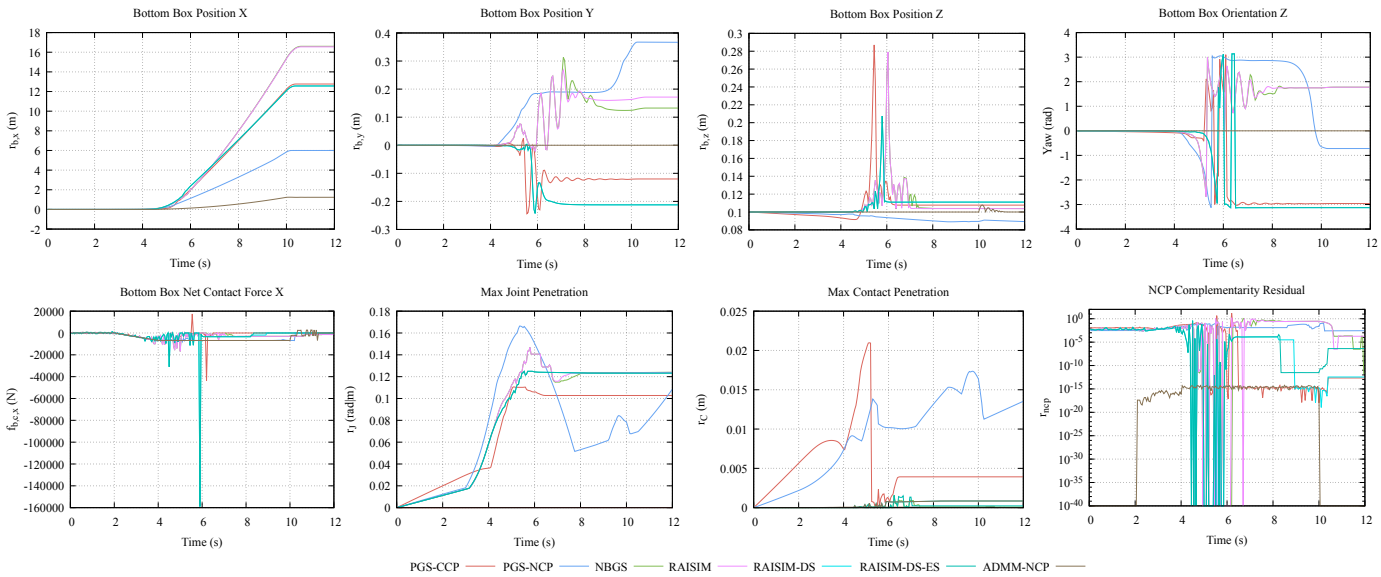

Fig. 15. *Boxes-Fixed*: Time-series plots of the motions, forces and performance metrics for the dragging run. The top box is of mass $m_t = 10^3$ kg while the bottom box $m_b = 0.1$ kg with a system mass-ratio of $r_m = 10^4$. Despite not converging, ADMM-NCP was still able to realize the expected behavior with minimal constraint violation. All other solvers (PGS-*, NBGS and RAISIM-*) grossly underestimated the constraint reaction forces leading to a collapsing of the joint and body interpenetration.

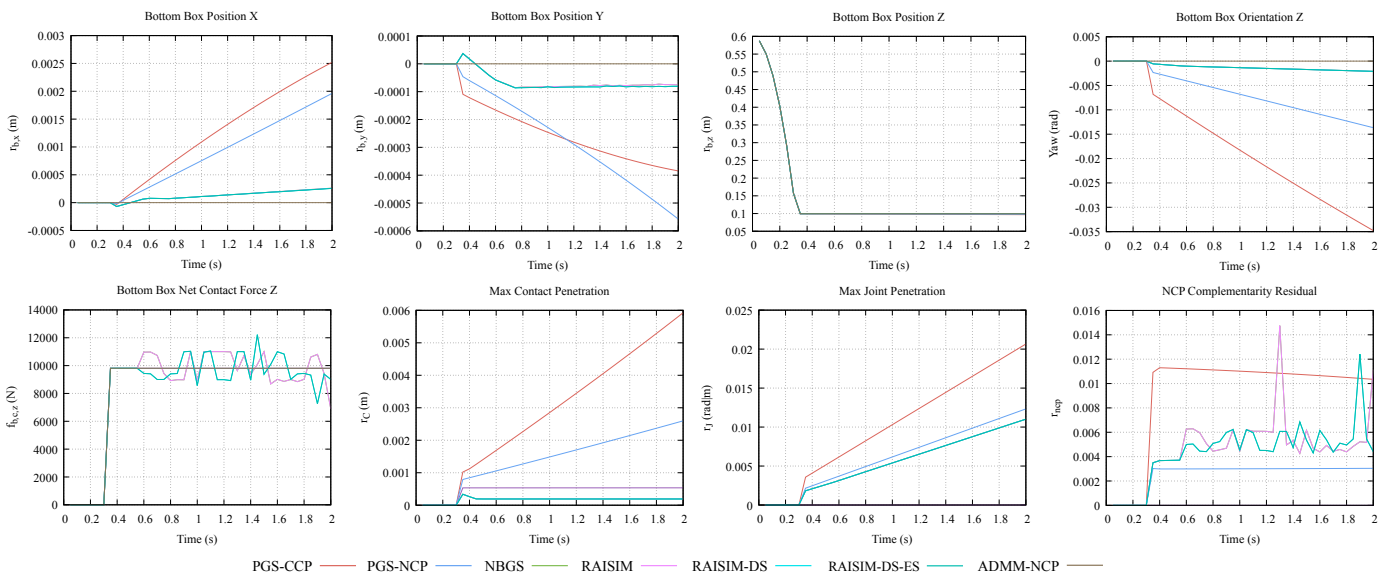

Fig. 16. *Boxes-Fixed*: Time-series plots of the dropping run showing the effects of the impact on the constraint violation (center-bottom) and overall motion (top row). These effects are also apparent in the NCP complementarity residual (bottom-right) were the solvers with large constraint violation also exhibit large values thereof. While NBGS and RAISIM-* variants exhibit relatively good performance w.r.t velocity constraint satisfaction, the contact forces along the normal (bottom-left) exhibit very large fluctuations w.r.t the correct solution attained by PGS-* and ADMM-NCP.

The third run that introduces constraint stabilization is depicted Fig. 17. The stabilization brought a significant reduction in constraint violation for all solvers. As observed previously, the application of the external force causes NBGS and vanilla RAISIM to insufficiently propagate the forces to the joint when sliding begins, causing the motion to differ w.r.t. the other solvers. However, this is somewhat of a false negative, as the deviation is due to stack of boxes tipping over during the dragging phase. See Fig. 17 for a more detailed description of this discrepancy. Overall, we observed that constraint stabilization is capable of reducing constraint violation while retaining the properties of the NCP[14].

In addition, motivated by the results in the three primary runs, we also evaluated the effects of varying the mass-ratio. To more thoroughly analyze how mass-ratios impact convergence, we performed a grid-search over combinations of masses in the range $m_{bt}, m_{bt} \in [10^{-6}, 10^{6}]$, generating a set of single-step problems: one at rest and another while the bottom box was pushed along the X-axis with f_{max}. Solving both sets with $N_{max} = 10^5$ to allow for convergence to occur reveals

[14]This in part, can be considered experimental verification of the invariance property of the De Saxcé operator, as mentioned in Sec.V-C and proven analytically in Appendix. A-B

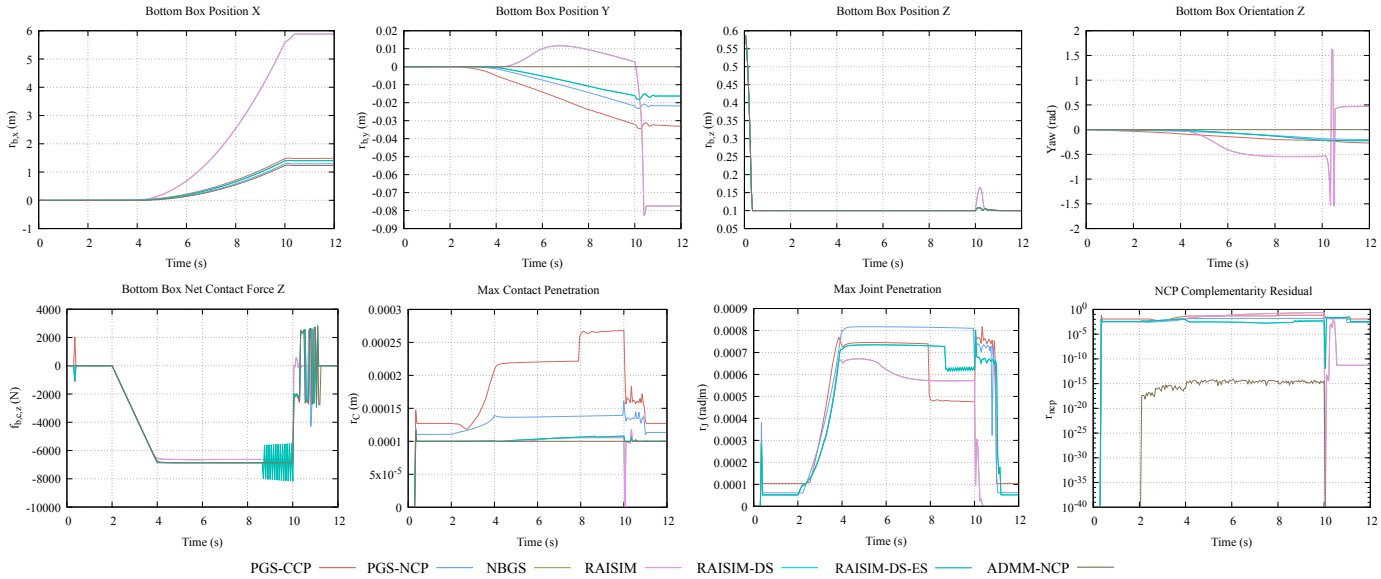

Fig. 17. *Boxes-Fixed*: Time-series plots of the combined drop and dragging run that also enables constraint stabilization. The augmentation to the problem leads to significant improvement w.r.t constraint violation (center-bottom) for all solvers. In the case of the NBGS (green) and vanilla solvers RAISIM (magenta) the large deviation from the correct solution (brown, ADMM-NCP) is somewhat circumstantial. The initial impact causes a dislodging of the joint, which leads to a shifting of the total center of mass. This leads to a tip-over occurred, which induced a short flight-phase wherein the external force dominated the reduced number of contacts, therefore dragging the system excessively.

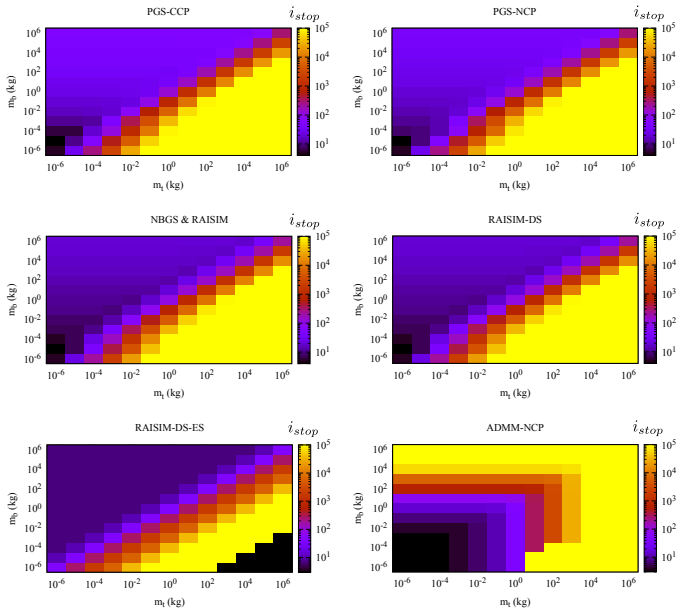

Fig. 18. *Boxes-Fixed*: A heatmap-like rendering of solver iterations i_{stop} as functions of the masses m_b and m_t of the bottom and top bodies, respectively. All splitting methods perform better when the masses are equal (left-to-right diagonal elements) as well as when the top body is lighter (upper-left diagonals). Note also the varying scales and respective color gradients of each solver. We were unable to use a common scale as the gradients became incomprehensible. Most revealing is the experimental verification of the fundamental differences between ADMM and splitting w.r.t inertial disparity (i.e. mass-rations). The latter all suffer when the relative mass ratio is larger than $\gtrapprox 1000$ and the bottom body is heavier. This shows how splitting-based solvers struggle to propagate contact reactions through the joints to the top body. Conversely, ADMM-NCP is more robust to large mass ratios, and in most cases manages to converge within $N_{max} = 10^4$ iterations. However it seems to be more sensitive to the absolute scale of the heaviest body, which, directly determines the spectral radius $L = \rho(\mathbf{D})$ of the Delassus matrix.

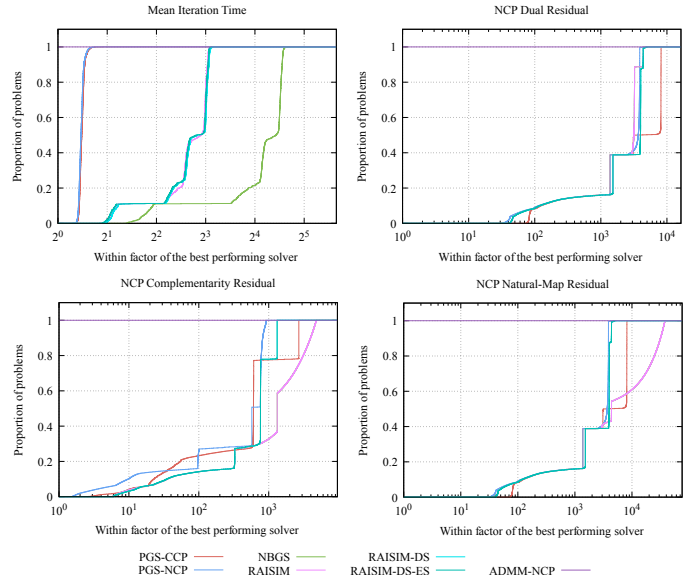

Fig. 19. *Boxes-Fixed*: performance profiles showing the relative performance between solvers. In this scenario ADMM-NCP is the outright winner, as indicated by being the best solver on 100% of the samples across all metrics. It was the only solver to ever be able to converge on some samples, and dominated all other solvers by more than two orders of magnitude on all accuracy metrics on the majority of samples.

very interesting properties that distinguishes the splitting-based solvers from ADMM-NCP. The full extent of this analysis can be found in Fig. 39 and Fig. 40 in Appendix. D, but an exemplary extract is provided in Fig. 18. These figures present heat-map-like renderings of the solver iterations and accuracy metrics as functions of the two body masses. Particularly revealing is how they show that ADMM-NCP is more sensitive to the absolute numerical scale of the actual masses, as

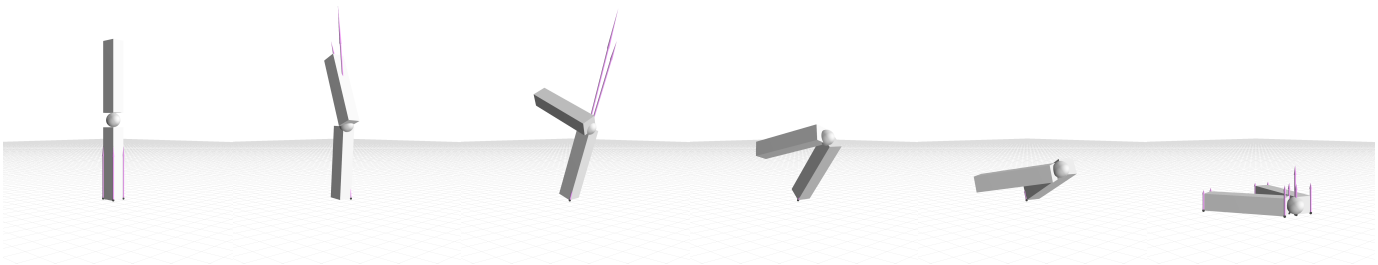

Fig. 20. *Nunchaku*: A rendering that showcases the motions of the system in the corresponding set of experiments. The system is initialized to drop from a fixed initial height. After impacting the plane, all four corners of the bottom box are in contact, and the system is able to remain vertical (left-most frame) for several seconds until numerical drift from the integrator eventually leads to the top box collapsing into the intermediate sphere and onto the bottom box (left middle frame). As collisions are disabled between the boxes and the spheres, only the boxes between them can render internal contacts. The collapse indeed leads to this case, and the top and bottom boxes remain in internal contact until the entire system contacts the plane (right-most frame). This particular rendering was generated using the ADMM-NCP solver, which can generate minimum-norm contact reactions without internal couplings (left/right-most frames).

opposed to their ratio. More specifically, these renderings, and especially those depicting i_{stop} in Fig. 18, indicate that the convergence of ADMM-NCP actually depends more on the spectral radius $\rho(\mathbf{D})$ than on the condition number $\kappa_{\mathbf{D}}$. This behavior seems to be inline with the analysis presented by Nishihara et al in [84] regarding the convergence properties of ADMM. This also motivates the use of *preconditioning strategies*, as was proposed by Tasora [45], which we intend to investigate further in the future.

Lastly, we executed a curated run using the initial mass distributions in order to render the performance profiles depicted in Fib. 19. On this particular problem ADMM-NCP clearly dominated, as it was the only one able to converge at all on certain samples, and exhibited the best accuracy metrics on all samples. Most importantly, w.r.t the accuracy metrics, all splitting-based trailed by a factor of at least one, and up to three, orders of magnitude, essentially corresponding to significant digits of accuracy. This degradation of accuracy is therefore consistent with the observed underestimation of the computed constraint reactions.

***Nunchaku**:* The third primitive experiment involves a system we have fondly named *Nunchaku*, consisting of two boxes and a sphere connected by two spherical joints. Each box is of dimensions $0.1\,\mathrm{m} \times 0.1\,\mathrm{m} \times 0.5\,\mathrm{m}$, the sphere is of radius $r = 0.05\,\mathrm{m}$, and all bodies are of equal mass $m = 1\,\mathrm{kg}$. The objective of this experiment is to observe the effects of hyperstaticity when multiple bodies come into contact with the environment (i.e. external collisions) as well as with each other (i.e. internal collisions). In addition, we also wanted to observe the efficacy of using constraint softening and stabilization. In this scenario, we do not apply any external force on the system. Instead, we vertically align all bodies, suspend the system at a fixed height above the ground plane and let it drop. The collapsing motion of the constrained system leads to a large number of internal and external contacts, thus resulting in a multitude of constraint loops, as exemplified in Fig. 20. This experiment is executed in the form of an ablation of constraint softening and stabilization over four runs: (a) as a hard-constrained problem, (b) with only constraint stabilization enabled, (c) with only constraint softening enabled (d) both constraint softening and stabilization enabled.

In the first run all solvers behaved reasonably, with minimally perceptible constraint violation. The time-series data, depicted in Fig. 21, indicate that most solvers are indeed capable of keeping constraint violation to within a few millimeters for both joints and contacts, with PGS-NCP and and ADMM-CCP performing the worst. Observing the trends of the number iterations required for each solve, it becomes clear that the aforementioned solvers reached the max iterations exactly at times of increased violation. Thus, similarly to the effects of a high mass-ratios observed in the Boxes-Fixed, the rate of convergence on Nunchaku is impacted by the hyperstaticity due to significant rank-deficiency in the contact Jacobian $\mathbf{J}_C$.

The second run introducing constraint stabilization is depicted in Fig. 41, and demonstrates a drastic reduction in constraint violation across all solvers, even in cases where the solvers fail to converge before the maximum iterations. The third run is depicted in Fig. 42 and shows that MuJoCo-like constraint softening scheme brings a significant decrease in the number of iterations and the overall constraint violation. In addition, it bears no discernible impact on the complementarity and natural-map residuals, indicating that the constraint softening mostly retains physical plausibility and accuracy.

The fourth run that includes both constraint softening and stabilization is shown in Fig. 43. Combining the two augmentations significantly increases overall performance for all solvers, both in terms of constraint violation and the number of iterations. One can argue that if the parameters of constraint softening are set appropriately for the problem then additional constraint stabilization is redundant. Our goal, however, is to evaluate the individual effects, as well as the interplay, of these augmentations when using recommended default parameters. Although an extensive tuning of these parameters can indeed be beneficial, it is often quite laborious since, to the best of our knowledge, no automatic procedure exists at present time.

An final curated run on the hard-contact problem is used to generate the performance profiles depicted in Fig. 22, and summarize the comparison of all solvers applied to this system. Interestingly, these show that on this system RAISIM and RAISIM-DS seem to outperform even ADMM-NCP w.r.t the complementarity residual r_{ncp} accuracy metric, even if only by a small margin. Although this result is surprising, it can be understood by considering that the vast majority of

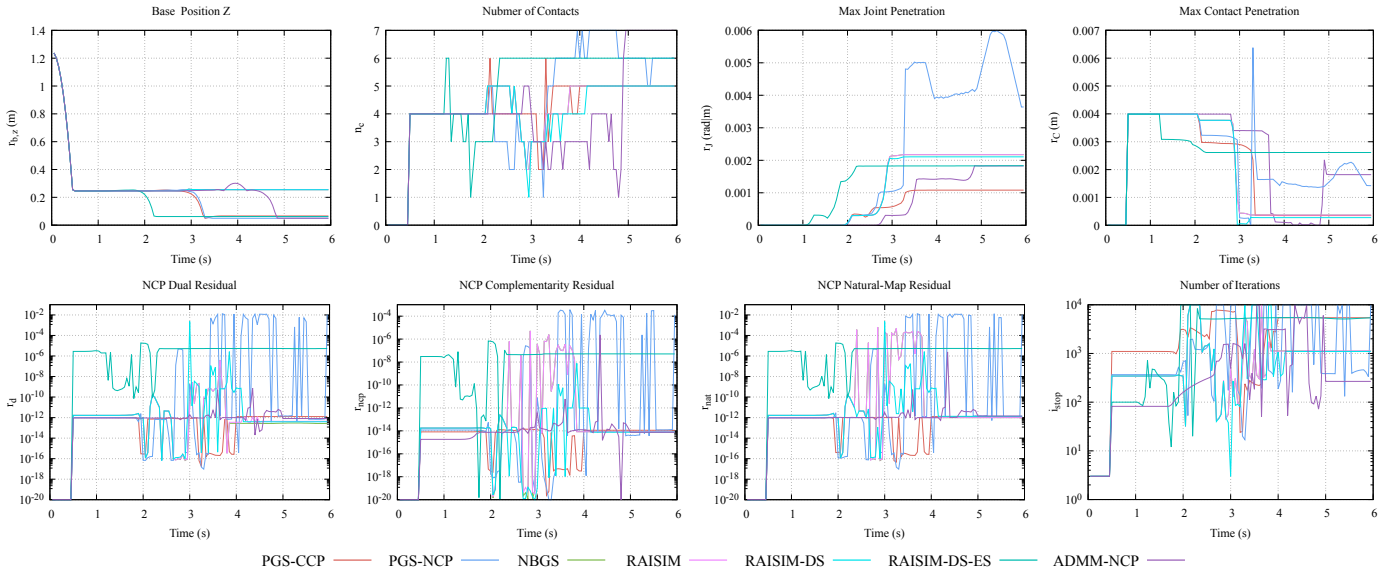

Fig. 21. *Nunchaku*: Summary of the first run, solving the hard-contact problem without augmentations. All solvers exhibit non-zero constraint penetration, i.e. configuration-level constraint residuals. Considering the absence of constraint stabilization, their performance is rather good on such a small-scale problem.

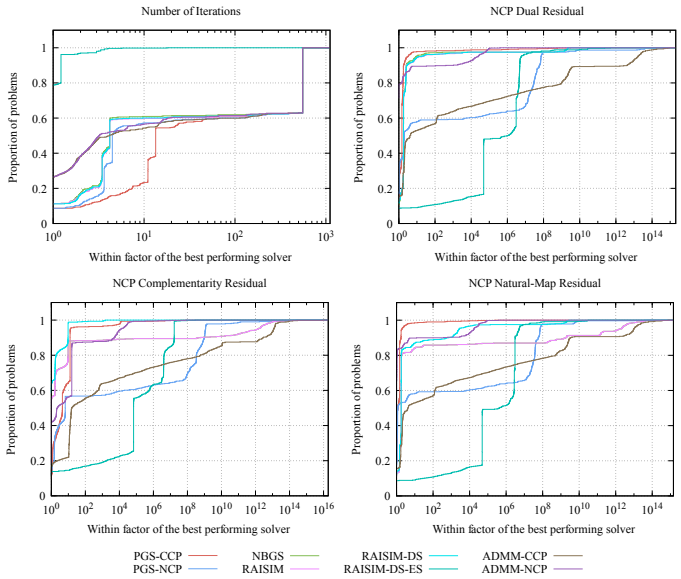

Fig. 22. *Nunchaku*: Performance profiles showing the relative performance between solvers on this problem. While PGS-CCP is the least performing in terms of number of iterations (top left), it demonstrates accuracy comparable to ADMM-NCP. RAISIM and RAISIM-DS also seem to perform best w.r.t NCP complementarity (bottom left). The least accurate, but fastest to converge is the early-stopping RAISIM-DS-ES.

samples include sticking or opening contacts, whereas one of the most prominent advantages of ADMM-NCP over the other solvers lies in cases of frictional sliding. Moreover, PGS-CCP also demonstrates good accuracy performance comparable to that of ADMM-NCP, but is the slowest of all in terms of convergence. RAISIM-DS-ES was the fastest solver overall on this problem, but also the least accurate, trailing all others by multiple orders of magnitude w.r.t the accuracy metrics.

***Fourbar**:* The fourth and final primitive experiment involves a classic planar four-bar linkage, which we will simply refer to as *Fourbar* for brevity. Such systems form the basis of many types of mechanisms that transfer motion and power. Indeed they are ubiquitous in mechanical assemblies, ranging from Ackermann steering mechanisms [93] in automobiles to autonomous excavators [94]. The Fourbar problem we employ in this work is categorized as a planar parallelogram-type with an RRRR joint arrangement, where each 'R' indicates a 1D revolute (a.k.a. hinge) joint. This system provides a versatile reference problem for evaluating all components of physics simulators such the dual solvers we evaluate in this work, as well as numerical integration schemes [10].

All four bodies are of dimensions $0.01\,\mathrm{m} \times 0.01\,\mathrm{m} \times 0.1\,\mathrm{m}$ and mass $m = 1\,\mathrm{kg}$. Of the four revolute joints, two are actuated and two are passive with the actuated joints opposing each other w.r.t one of the diagonals of the parallelogram. All four joints define their rotation w.r.t the rest state when the Fourbar is a square and are subject to limits at $q_j \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. The bottom link is the designated *base* body and is initialized at a height $z_b = 10\,\mathrm{cm}$ above a plane aligned to the world origin. The system is first dropped in order to impact the plane, from where it eventually begins to roll to one side due to numerical drift from state integration. Due to the imposed joint limits, it locks onto a rhombus-like state and is left to stabilize until a force amounting to the total mass of the system is applied to whichever body is the highest at time $t = 10\,\mathrm{s}$. The push causes the Fourbar to fall onto the plane where all four bodies contact the ground. The objective of this experiment is to therefore evaluate the full set of solvers in scenarios where all constraints types are present, i.e. joints, limits and contacts, and while they all act simultaneously in the presence of an intrinsic kinematic loop. As done for the previous primitive systems, the experiments are executed over multiple runs to perform an ablation on constraint stabilization and softening. Moreover, in order to also evaluate the effects

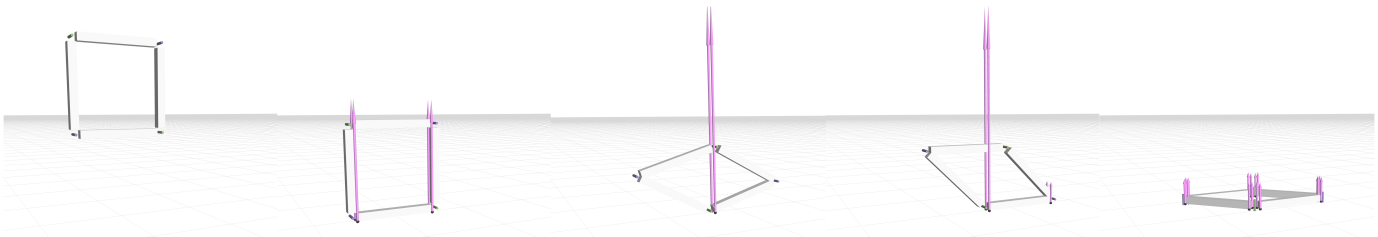

Fig. 23. *Fourbar*: Renderings of the motion of the system and active contact forces along the different phases of the experiment. The four-bar is dropped from an initial height from which it begins to shift onto one side due to gradually accumulating numerical drift due to the semi-implicit Euler integrator. From there it rolls over to the right lateral link, where it is pushed by an external force so that it can collapse. The external disturbance serves to push the system into a configuration where all bodies are contacting the ground, each at 4 co-planar points. In many cases this occurs while the joint limits are still active. This in this final phase, a maximal number of active constraints are present, in addition to the intrinsic kinematic loop of the system.

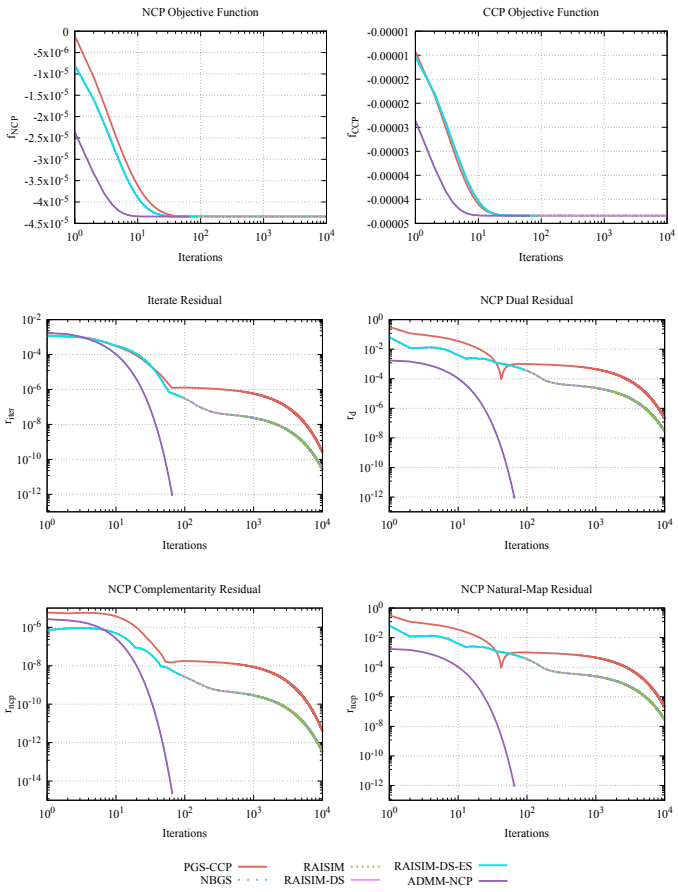

Fig. 24. *Fourbar*: Convergence profiles of the most prominent solvers, with both X and Y axes are rendered in *log*10 scale. The NCP and CCP objectives (top row) are juxtaposed with the accuracy performance metrics to highlight the crucial differences between them. RAISIM-DS-ES relies on the former to determine convergence, and thus traverses the same trajectory as NBGS, RAISIM and RAISIM-DS, but stops much sooner than these. Meanwhile, PGS-CCP presents the slowest convergence of all, with a notable *notch* in the NCP dual and natural-map residuals at around iteration $i = 70$. ADMM-NCP presents essentially a linear convergence rate that is approximately two orders of magnitude larger than the splitting-based solvers.

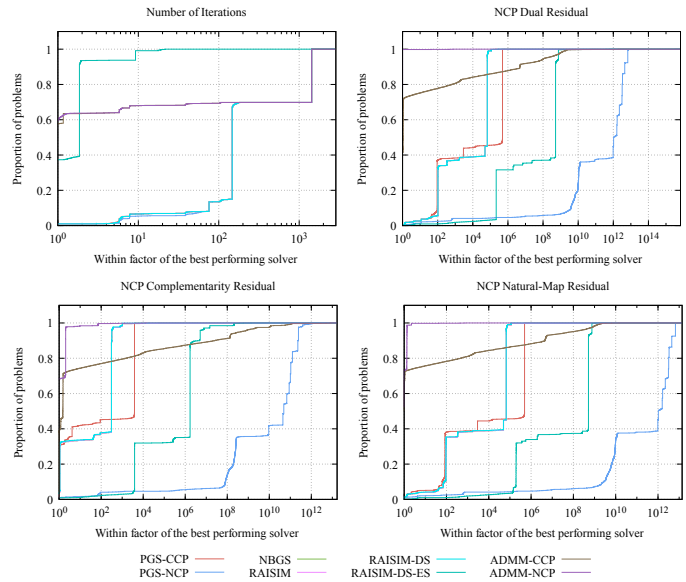

Fig. 25. *Fourbar*: Performance profiles of the full set of solvers. In contrast to the other primitive problems where the differences between solvers were less pronounced, in this case ADMM-NCP proves to be the best solver overall by a wide margin. Even w.r.t the number of iterations i_{stop}, ADMM-NCP was best on approximately %60 of the problems, and %70 w.r.t the accuracy metrics, while being within a factor of 10 close to the best solver on %100 of the problems.

of inertial disparity, we execute an additional run where the masses of the lateral (left-right) bodies are set to $m_L = 1\,\text{kg}$ and $m_H = 1000\,\text{kg}$. This final runs thus evaluates the solvers in a scenario where all types of ill-conditioning are present.

The first run is depicted in Fig. 26. ADMM-NCP and RAISIM-DS-ES were the only solvers capable of converging within the maximum number of iterations in the majority of the time-steps. To elaborate on this result, convergence profiles are provided in Fig. 24, corresponding to a time-step where the Fourbar has impacted the plane. Moreover, NBGS, RAISIM and RAISIM-DS performed on par with ADMM-NCP w.r.t. constraint violation of both joints and contacts. Meanwhile, the early-stopping RAISIM-DS-ES exhibits very high joint constraint violation compared to the former. Although the early-stopping termination criteria presents a significant aid in avoiding excessive iterations of the solver, it results in significant underestimation of the constraint reactions.

The second run with constraint stabilization is shown in Fig. 44 and demonstrates a drastic reduction in constraint violation for all solvers. ADMM-NCP and RAISIM-DS-ES were mostly unaffected in terms of convergence, except when all bodies contacted the ground simultaneously. We hypothesize that the additional bias of the stabilization term causes an effect similar to the wedging example described by Erleben

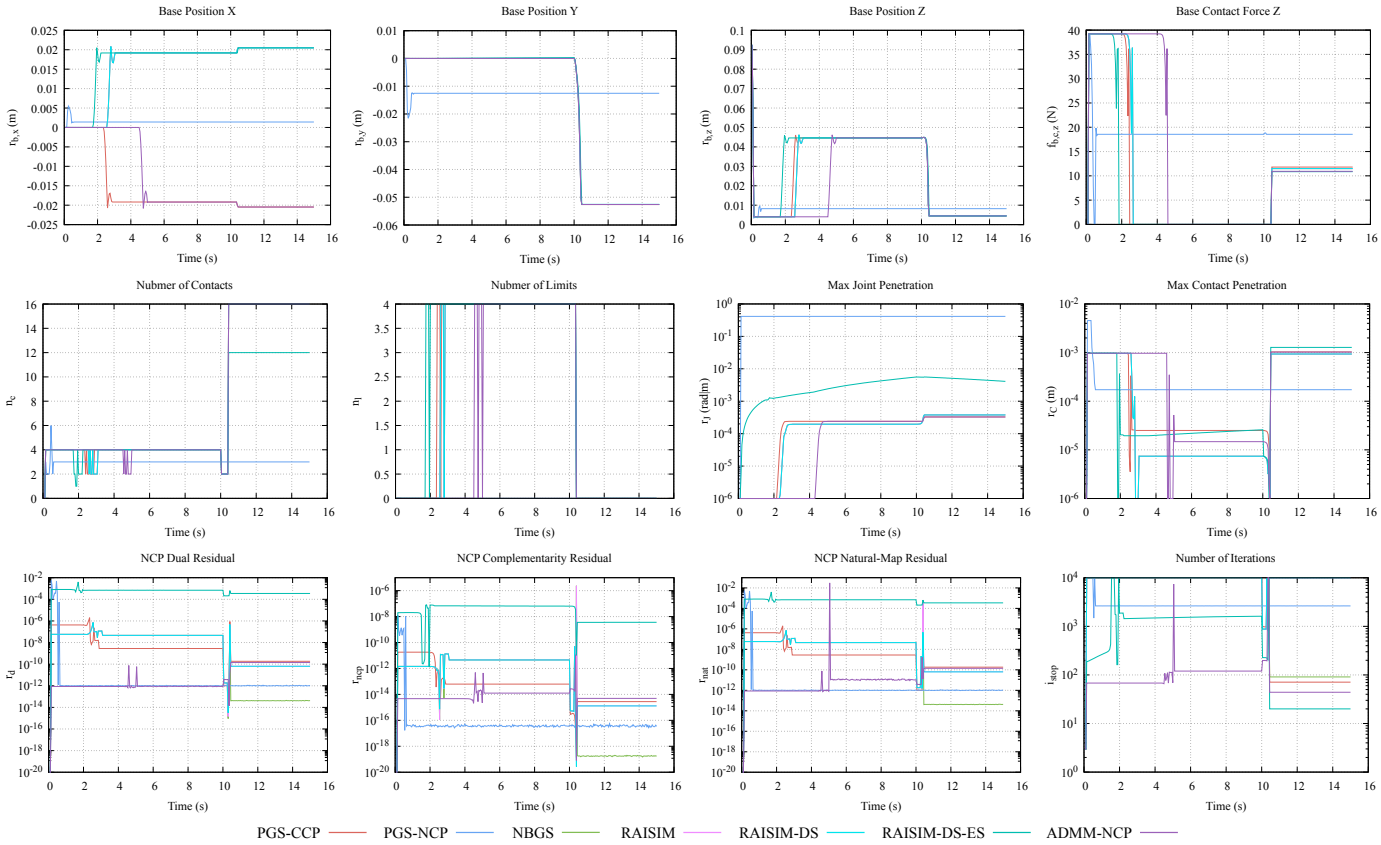

Fig. 26. *Fourbar*: Summary of the first run on the un-augmented hard-contact problem. The top row depicts the absolute position of the designated base link, i.e. the lowest of the two horizontal bodies. The second row depicts the net contact forces acting on the base along the respective coordinates. The system is initialized at a fixed height (top-right) from where it falls until impacting the ground at about $t \simeq 0.2\,\text{s}$. The system shifts to one of two sides along the global X-axis, depending on the numerical drift introduced by semi-implicit integrator, which depends on the precision of the forces computed by each solver. For some solvers, the system begins shifting left, while for others right. ADMM-NCP being the most precise, accumulates drift the least, thus shifting the latest of all, while RAISIM-DS-ES shifts the earliest due to the imprecision induced by the early-stopping termination criteria. The latter also exhibits the most amount of configuration-level constraint violation for joints (lower-middle-right in cyan). At the end of the shifting motion, the limits of all four joints activate resulting in the Fourbar rolling over to one side, and at $t = 10\,\text{s}$ the external force acting on the top-most body along -Y-axis pushes the system onto the ground where all bodies are simultaneously contact it. This results in $n_c = 16$ total simultaneous contacts, thus making the system significantly over-constrained. The plot showing the contact force along the Y-axis (top-middle-center) shows how the early-stopping results in significant internal forces due to the hyperstaticity of the system. Conversely, ADMM-NCP presents the least (virtually zero) coupling.

in [92] and Horak et al in [95]. Essentially when multiple constraints are acting along approximately co-linear and opposing directions, the solver ends up in a ping-pong effect where the contribution of each constraint cancels that of another, causing it to make slow or no progress towards a solution. In this case the opposing biases are those of the contact stabilization and the constraint dimensions of the joints. As convergence is conditioned on the required tolerance, in this case set to $\epsilon_{abs} = 10^{-12}$, the fact that ADMM-NCP still renders correct forces is an indication that the ping-pong effect most likely manifests at smaller scales of the relevant residuals.

The third run, executed with constraint softening applied to all constraints, is depicted in Fig. 45. This augmentation results in all solvers being able to converge well within N_{max}. ADMM-NCP and RAISIM-DS-ES exhibited the most significant improvement, with the latter being able to do so within $i_{stop} < 100$. However, this acceleration came at a significant cost of increased constraint violation for all solvers, which was close to an order of magnitude larger compared to the previous runs. Comparatively, the fourth run depicted in Fig. 46 shows that combining both constraint stabilization and softening yields the best results overall.

The fifth run, that introduces a large mass-ratio to the system and with constraint stabilization, is depicted in Fig. 47. In this scenario, all solvers fail to converge with $N_{max} = 10^4$. However, most solvers exhibit a relatively low degree of constraint violation, except the splitting-based methods when initial impact with the plane occurs. For these, all the performance metrics pertaining to physical accuracy are significantly degraded w.r.t. previous runs. This is consistent with the Boxes-Fixed experiment where all splitting-based solvers were incapable of correctly propagating forces and satisfying both configuration-level and dynamics constraints.

Lastly, we must remark that initial testing caused us to exclude PGS-NCP and ADMM-CCP from this experiment as they resulted in erroneous behaviors. The former caused significant rotation of the base link upon first impact from the drop, even when constraint stabilization was enabled. The latter diverged moments after all bodies contacted the ground after the push at $t = 10\,\text{s}$. These motions distorted the time-

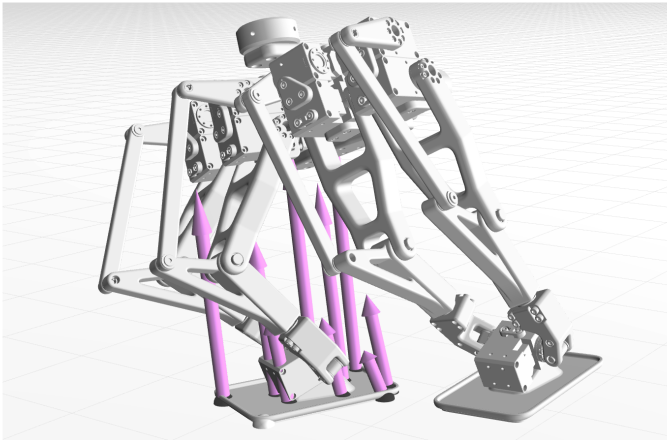

Fig. 27. *Walker*: The forces distributed over the non-convex collision geometry of the system's planar feet rendered by the ADMM solver. Black and white spheres indicated contacts that are sticking and opening, respectively. The collection of contacts on the heel of the left foot is consistent with the CoM of the system. Importantly, ADMM is able to generate minimum-norm contact reactions that exhibit minimal to no internal cross-couplings, despite the large number of co-planar contacts being generated in this case.

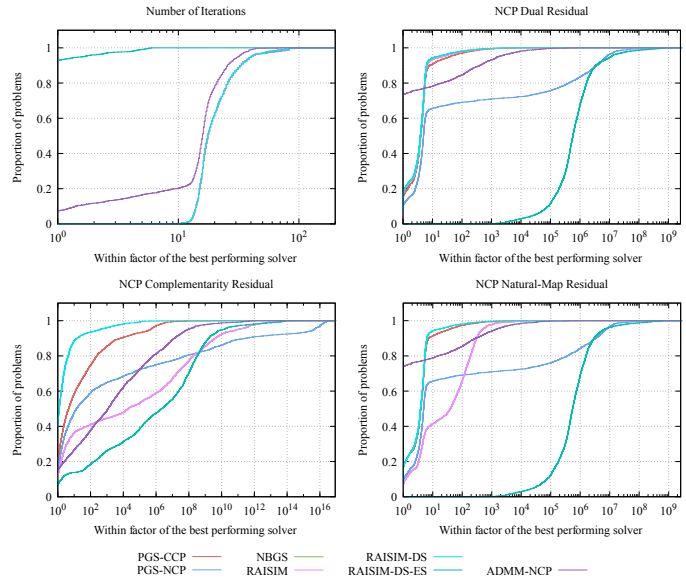

Fig. 28. *Walker*: Performance profiles on the set of solvers. ADMM-CCP has been excluded is it often diverged on this problem resulting in invalid solutions. On this particular problem RAISIM-DS-ES is by a wide margin the fastest solver (top left), while ADMM-NCP is the most accurate overall. However, RAISIM-DS and PGS-CCP prove to be better w.r.t the accuracy metric of the complementarity residual r_{ncp} (bottom left), and within a factor of 10 close to ADMM-NCP w.r.t the NCP dual and natural-map residuals (right column). This result indicates that RAISIM-DS and PGS-CCP, can be effective in rendering realistic simulations w.r.t the accuracy metrics.

series plots in the aforementioned figures to such a degree that we opted to remove them from the comparison.

D. Full-Scale Systems

The second set of experiments evaluates the dual solvers on a set of full-scale systems. With these, we demonstrate the capabilities and limitations of each solver on problems that are representative of practical applications for controlling robotic systems and *Audio-Animatronics*® figures.

Walker: The first full-scale system is the bipedal Walker [96]. Although we categorized it as belonging to the *Audio-Animatronics*® group, it is in fact a hybrid between the former and a robotic system. It is a multi-limb floating-based system with planar feet, intended for evaluating bipedal locomotion of highly over-and-under-constrained systems. Indeed, each leg exhibits three intrinsic kinematic loops, forming seven overall when both feet are in contact with the ground. As mentioned in Sec. VII-A, collision detection is preformed using the raw mechanical meshes imported from the mechanical designs of CAD, i.e. we have not applied any post-processing to convexify or simplify them. This system, however, does not impose joint limits. Locomotion is realized naively in open-loop by using joint-space PID control to track reference trajectories of duration $t_f \approx 23\,\mathrm{s}$. These references are retargeted from an animation generated for a simpler joint morphology, as detailed in [97]. The objectives of this experiment, corresponding one-to-one to each run, evaluate solvers in the cases of:

1) the un-augmented system free-floating without gravity, in order to establish a baseline for performance
2) the un-augmented system while it is locomoting on the ground with gravity and contacts enabled
3) when including constraint stabilization
4) when including constraint softening only for contacts[15]
5) simulation stability when using high-throughput settings for all solvers, with $N_{max} = 500$ and $\epsilon_{abs} = 10^{-4}$

However, we have had to exclude the ADMM-CCP solver from all but the first run, as it once again diverged in certain time-steps, causing significant distortion of all time-series plots. Regardless, since the experiment includes the primary solvers of interest, the comparisons can proceed without being significantly affected by this absence.

The first run is depicted in Fig. 48. Since in this case essentially only the bilateral joint constraints are active, the only relevant metric is that of the dual residual r_d. In this case, this metric represents the residual velocities in constraint-space, which should ideally be zero at all times. As we do not include any constraint stabilization, all solvers accumulate significant configuration-level constraint violation in the form of the joint penetration residual. RAISIM-DS-ES exhibits more than an order of magnitude larger compared to other solvers. ADMM-NCP performs the best in this case, with the PGS and RAISIM variants performing comparatively similar.

The second run introduces contacts and gravity, and is summarized in Fig. 49. In this case RAISIM-DS-ES and PGS-NCP lead to the biped falling over within the first few steps, while all other solvers manage to render simulations were locomotion proceeds until the end of the reference trajectory. It becomes apparent that as these also exhibit the largest amount

[15]Initial testing, conducted to set up these experiments, lead us to conclude that it is not feasible to apply constraint stabilization to joints on this system. The induced under-determination of the forces resulted in almost instantaneous dislocation of the joints and all bodies flying apart.

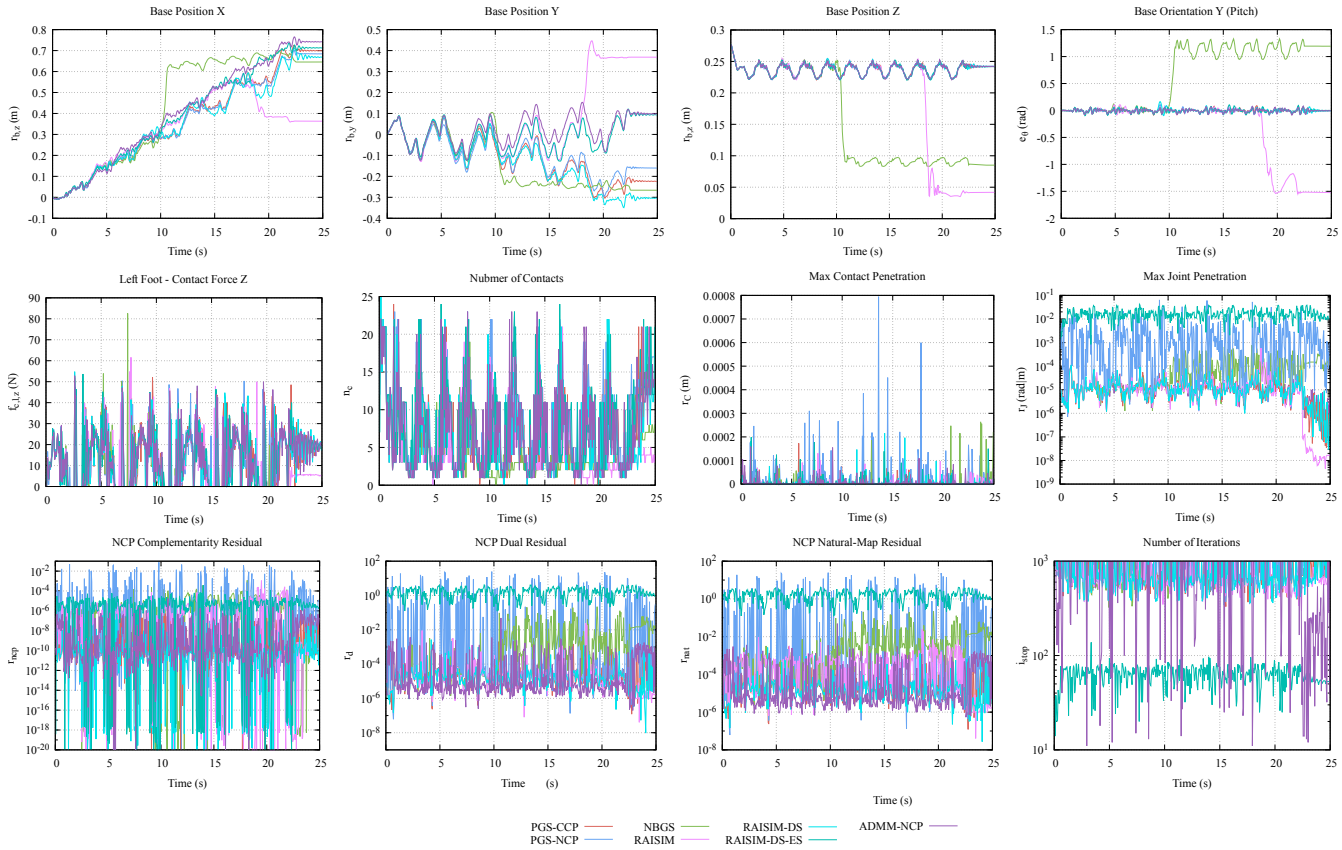

Fig. 29. *Walker*: Time-series plots of the third run where stabilization is applied to all constraint sets. ADMM-NCP exhibits the best overall performance on this problem with significantly better performance w.r.t the accuracy metrics of the NCP dual, complementarity and natural-map residuals. RAISIM-DS and PGS-CCP perform similarly well, and the motions they generate remain close to that of the former. Meanwhile, RAISIM-DS-ES, although reasonably performance, exhibits excessive constraint violation close to unity, due to the early-stopping termination criteria. PGS-NCP, NBGS and vanilla RAISIM are the least performing solvers.

of violation in the configuration-level joint constraints, and thus the rapid accumulation of error at the ankle joints leads to the tip-over. This scenario is of course not representative of actual applications, but illustrates the rate and scale at which the solvers deviate in the presence of multiple contacts that exacerbate the ill-conditioning of the problem.

The third run with constraint stabilization enabled is presented in Fig. 29. In this case, most solvers and including RAISIM-DS-ES and PGS-NCP, can render simulations that keep constraint violation to reasonably small values. In this case, NBGS and vanilla RAISIM lead to tip-overs, but we consider this a false negative as it is a mere artifact of their respective simulations. In fact, by setting the collision margin to $\delta_c = 10^{-5}$ we were able to make all solvers reach the end of the animation without falling. The most important outcome of this run is that most solvers were sufficiently aided by the constraint stabilization in maintaining violation low. The degree by which this was accomplished, renders simulations whose output is usable in actual applications.

The fourth run is executed with constraint stabilization on all constraints and constraint softening applied only for contacts. Fig. VIII-D depicts a summary of this run. The combination of both augmentations proves effective in rendering simulations that look very similar across all solvers. The softening provides a significant boost to the convergence of all splitting-based solvers but ADMM-NCP seems mostly unaffected. Moreover, for all but PGS-NCP and RAISIM-DS-ES, constraint violation is mostly the same as the first run, demonstrating that both augmentations can be combined effectively on such problems, similar to what was observed for Nunchaku and Fourbar.

The fifth run, depicted in Fig. 50, shows the effects of the high-throughput settings. In this case RAISIM-DS-ES was unable to maintain constraint violation, both kinematic and dynamic, to sufficient levels and very rapidly leads to a tip-over. All other solvers seemed mostly unaffected by this relaxation. ADMM-NCP, in particular, performed nearly identically to the second and third runs, thus exhibiting exceptional robustness w.r.t N_{max} and ϵ_{abs}.

A final curated run on the constraint-stabilized problem was used to generate the performance profiles presented in Fig. 28. From these we can conclude that although ADMM-NCP is once again the best solver overall, RAISIM-DS-ES, RAISIM-DS and PGS-CPP, present surprisingly comparable performance, in certain aspects. Specifically, RAISIM-DS-ES proved best overall in terms of number of iterations, RAISIM-DS proved best w.r.t. the NCP complementarity accuracy metric, and PGS-CPP proved to always come within a factor of 10 of the best solver w.r.t the NCP dual and natural-map metrics. Thus, although ADMM-NCP is once again the

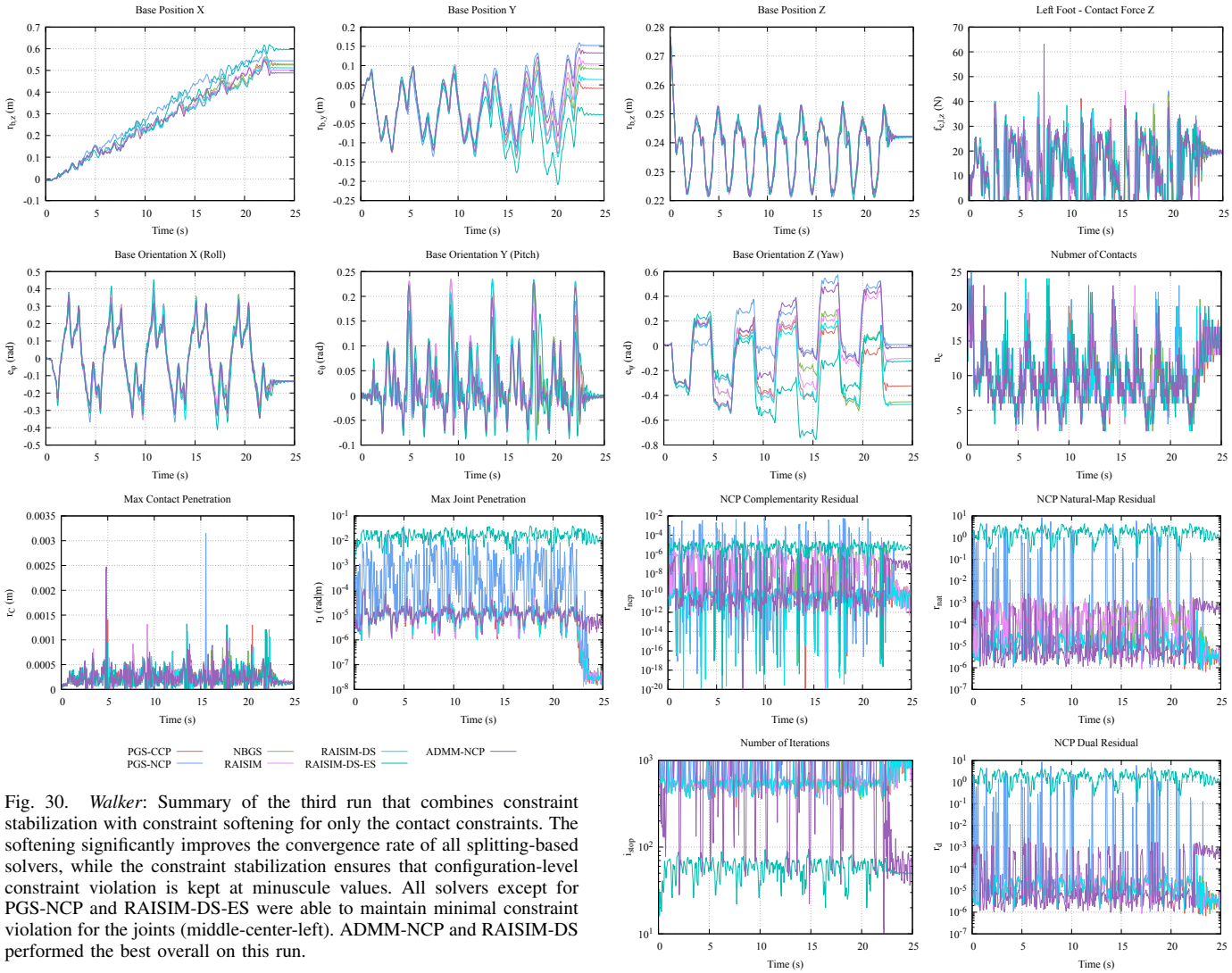

Fig. 30. *Walker*: Summary of the third run that combines constraint stabilization with constraint softening for only the contact constraints. The softening significantly improves the convergence rate of all splitting-based solvers, while the constraint stabilization ensures that configuration-level constraint violation is kept at minuscule values. All solvers except for PGS-NCP and RAISIM-DS-ES were able to maintain minimal constraint violation for the joints (middle-center-left). ADMM-NCP and RAISIM-DS performed the best overall on this run.

overall winner, the results indicate that using the problem augmentations with appropriate setting of parameters can prove effective in reducing the performance gap between the former and splitting-based solvers.

Lastly, we must remark on the influence of the PID controller in terms of simulation stability. We observed that despite the overall robustness of most solvers, setting the derivative gain of the PID controller to large values lead to a rapid explosion of the bodies, even when using ADMM-NCP. This made it exceptionally challenging to tune the controller gains, but most importantly, it is indicative that the present formulation is sensitive to actuator dynamics. Many physics engines alleviate this problem by introducing an implicit PD controller directly into the formulation of the forward dynamics problem [98], [99]. However, although this effect is well known, from our investigations, it seems to not be well covered in the known literature, and we intend to investigate this aspect further in future work.

***IronMan**:* The second full-scale problem is the highly-complex *IronMan © MARVEL Audio-Animatronics®* figure realized using the state-of-the-art A1000[16] platform and currently being exhibited at Disneyland Paris, France. It is by far the most challenging system in the benchmark suite, and undoubtedly exceeds the mechanical complexity of most humanoid robots by a large factor, with its 54 links and 67 joints. By comparison, the most complicated humanoid figures known to date, such as the first generation Atlas [100] of Boston Dynamics, consist of 29 links and 28 joints. Fig. 31 provides a split view that reveals the exceptionally complex mechanical assembly concealed by the exterior shells.

For this final systems, we will not proceed as exhaustively as before. Our objective here is merely to evaluate how solver performance scales for systems of this size, and to observe potential simulation artifacts. To this end, we executed two runs where: (a) the system is left to *rag-doll*[17], i.e. collapse due to no actuation being active, and (b) use a joint-space PID controller to track reference motions retargeted from an animation. In both runs, all solvers are set to the high-

[16] https://sites.disney.com/waltdisneyimagineering/a-1000/

[17] This is a reference to the term used computer animations and video games.

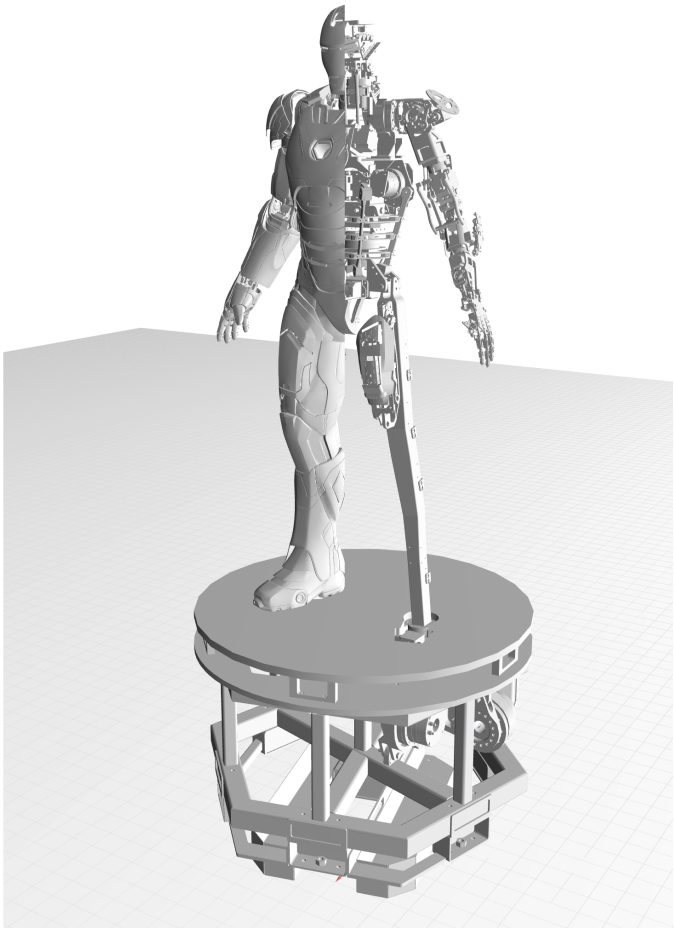

Fig. 31. *IronMan* © *MARVEL*: the *Audio-Animatronics*® figure, featured in real-life at Disneyland Paris in France, and shown here with (left) and without (right) the external shell geometries. The internal mechanical assembly reveals an exceptionally complex humanoid system.

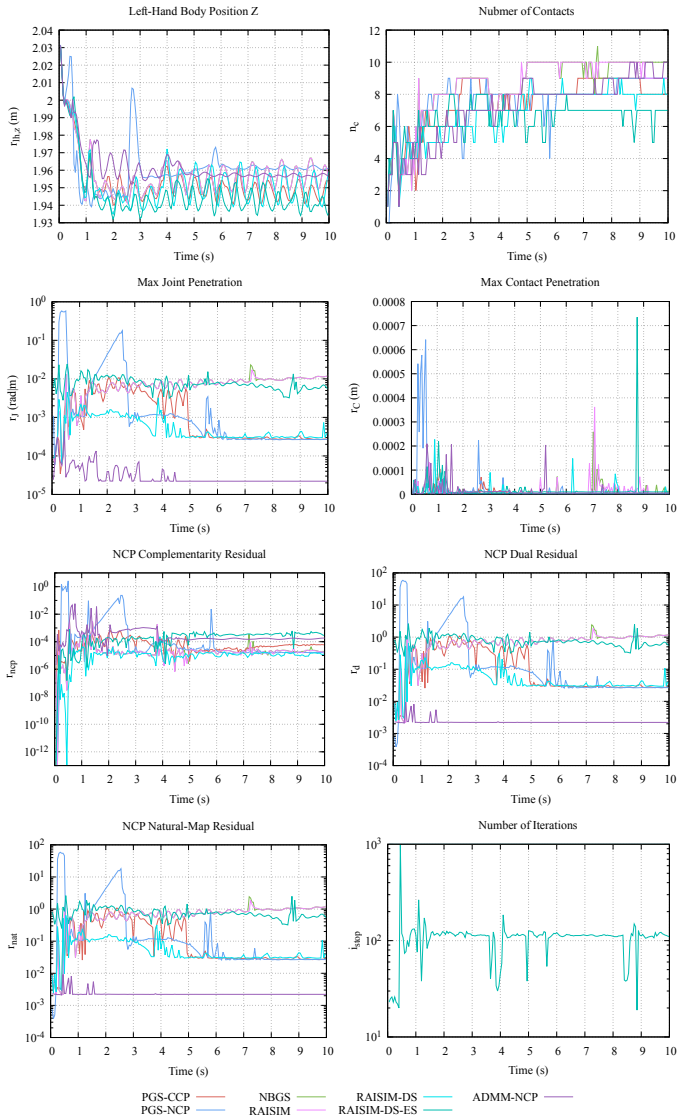

Fig. 32. *IronMan*: Summary of the first run where the system is unactuated and left to rag-doll. While joint contact penetration remain reasonable low for most solvers, and on the order of a few millimeters, the dynamic constraints represented by the accuracy metrics of the NCP dual, complementarity and natural-map residuals exhibit much larger values compared to other systems in the benchmark suite. In this regard, ADMM-NCP is the best performing one overall, and RAISIM-DS-ES the least.

throughput setting with $N_{max} = 1000$ and $\epsilon_{abs} = 10^{-6}$.

The first run where the system is allowed to rag-doll is summarized in Fig. 32. Despite the size of the system, a relatively low number of contacts occur during the motion, due to the fixed-base. Each solver overall performed comparatively similar to the Walker system, with ADMM-NCP exhibiting the best performance on all metrics, despite the lack of convergence. Although RAISIM-DS-ES was the only solver to converge on this problem, it performed the worst, yet the corresponding simulation proved sufficiently stable and with no visible artifacts. Most surprising on this system is that although configuration-level constraint violation is kept relatively low, the dynamic constraints paint a different picture. The dual r_d and natural-map r_{nat} residuals approach scales near unity, indicating that some constraints contain significant residual constraint-space velocities. However, given the large inertial disparity of the system, coupled with the lack of solver convergence, these values are most likely due to the joints that couple very massive bodies with relatively small ones, such as the ones in the base of the system. Such a phenomena is justification for considering a pre-conditioning of the problem, and/or employing vector-valued tolerances, so to ensure that residuals are evaluated in a scale-invariant manner.

The second run including the effects of the joint-space PID controller is depicted in Fig. 33. In this scenario most solvers performed similarly to the first run, with the exception of PGS-NCP which lead to erratic oscillations propagating throughout the system. Indeed, as shown in Fig. 33, PGS-NCP resulted in joint constraint violations up to $1\,\mathrm{m}$ at certain instances. This result, together with all other systems described previously, causes us to conclude that PGS-NCP is the least performing solver of the set we evaluated and least likely to be considered for use in actual applications.

Finally, an additional curated run on the actuated setup results in the performance profiles depicted in Fig. 34. From these we see that ADMM-NCP performs well only on the NCP dual and natural-map accuracy metrics, while failing

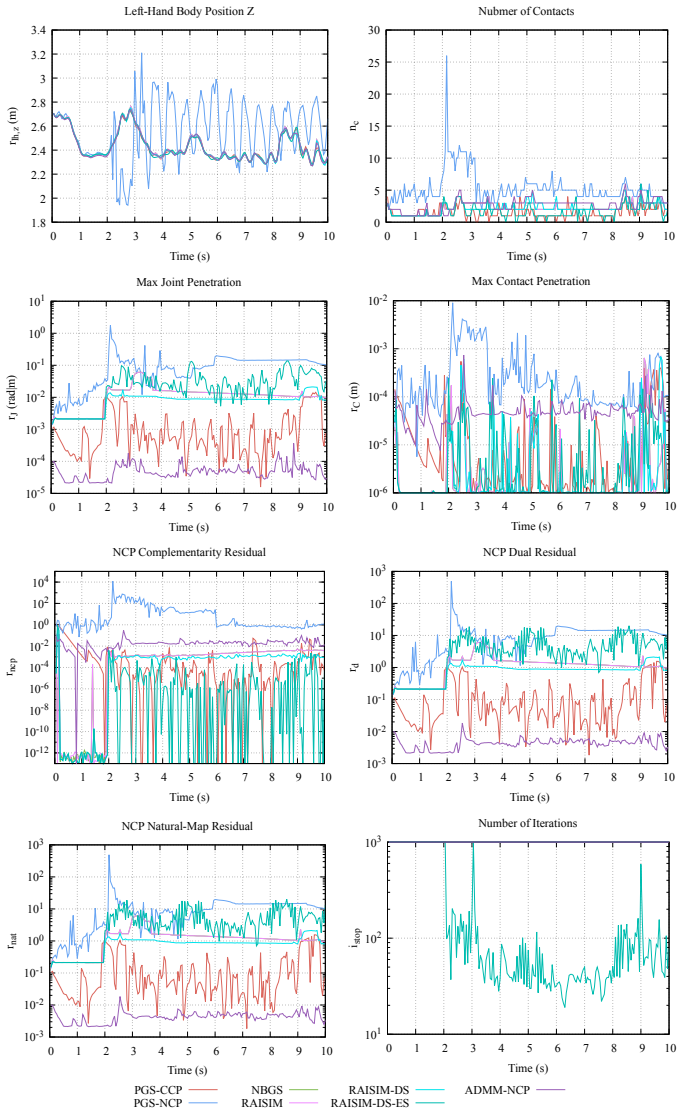

Fig. 33. *IronMan*: Summary of the second run in which the system is actuated to track a reference trajectories using a naive joint-space PID controller.

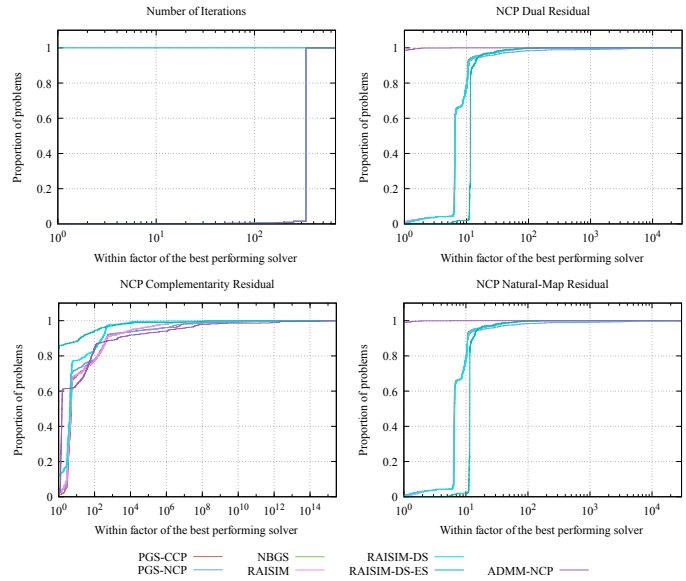

Fig. 34. *IronMan*: performance profiles

TABLE III
Benchmarks: Data contents of each dual problem sample.

Type	Entry	Description
Problem Definition	$\mathbf{D}$	Delassus matrix
	$\mathbf{v}_f$	free-velocity vector
	$\boldsymbol{\mu}$	friction coefficients
	dt	time-step
Problem Dimensions	n_b	no. of bodies
	n_j	no. of joints
	n_l	no. of active limits
	n_c	no. of active contacts
	$\{d_j\}$	joint dimensions
	$\{i_j\}$	joint index offset
	$\{i_l\}$	limit index offset
Problem Properties	c	sample category
	$\mathbf{rank}(\mathbf{J})$	Jacobian rank
	r_m	mass-ratio

to beat RAISIM-DS-ES and RAISIM-DS in terms of the number of iterations and NCP dual residual. The degree of this disparity between accuracy metrics is very surprising, even considering the respective comparison made for Walker in Fig. 28. However, the absolute performance presented in Fig. 32 and Fig. 33, provide context. The relative performance, although seemingly large, is only up to a factor of 100 lower for ADMM-NCP w.r.t residual-based metric on the majority of sample problems. This in combination with the fact that ADMM-NCP and PGS-CCP do not manage to converge, thus makes it difficult for this comparison to be conclusive.

E. Benchmarks

The third and final set of experiments we present in this work, involves quantifying the aggregate relative performance of each solver over multiple systems contained in the benchmark suite of Sec. VII-A. However, realizing this aggregation naively using curated runs over each problem can prove error prone and cumbersome. The most straightforward solution to this predicament is to compile a fixed set of of dual problems that can be stored and retrieved for offline solver execution, in a systematic manner and on-demand. This is the approach pioneered by Acary et al with the development and publication of the FCLIB [86], presenting a set of frictional-contact problems for various large-scale mechanical systems, generated using the Siconos [101] and LMGC90 [102] software platforms. Other similar efforts also exist in the literature, notably that of Lu et al in [103]. However, the problems contained in the aforementioned works cater to systems like granular media and static structures, with a clear absence of those with highly-coupled bilateral kinematic constraints that are most relevant to our application domain.

In order to realize such a setup, we executed a single curated run of ADMM-NCP on each of the twelve systems, sampling dual problems at certain frames (i.e. time-steps). The sampling criteria is based on defining sample *buckets*

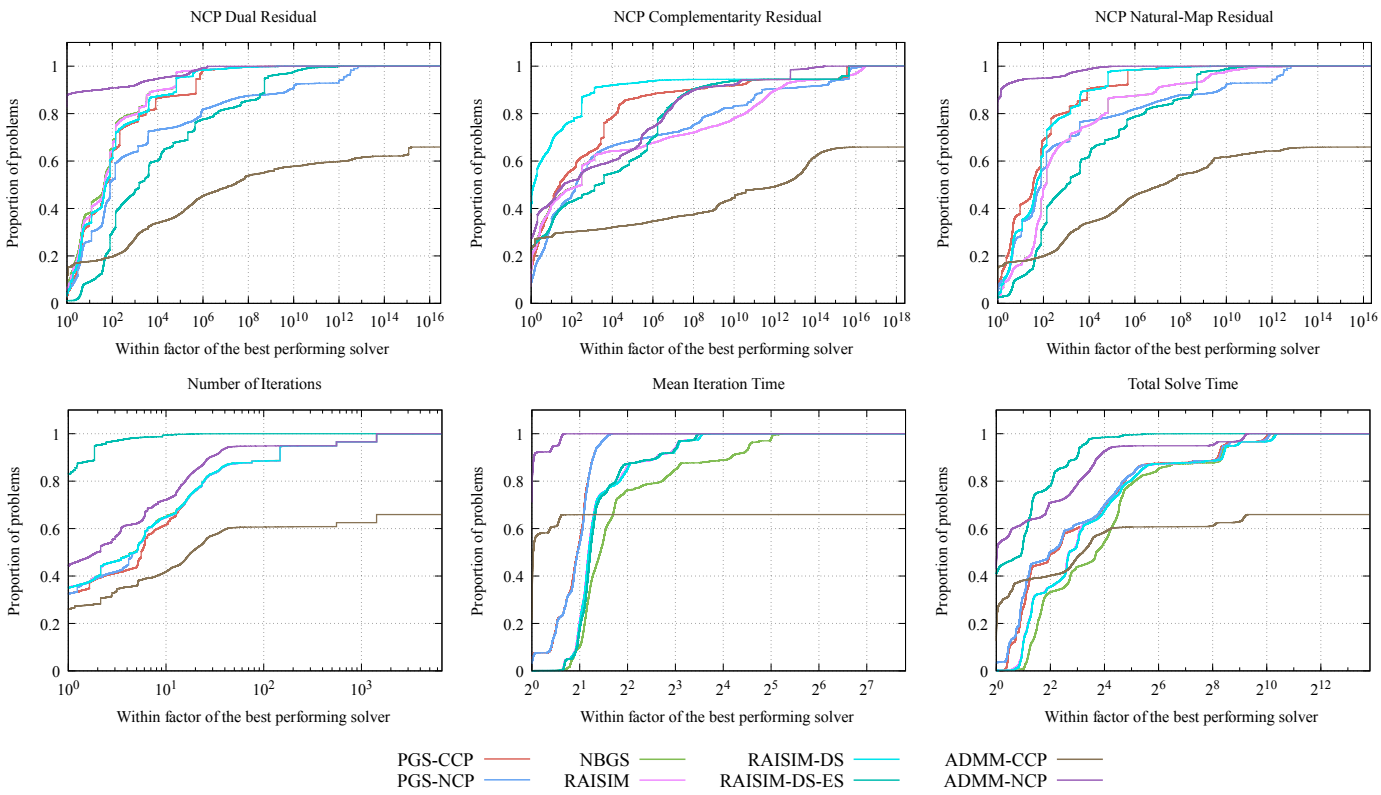

Fig. 35. *Benchmarks*: performance profiles generated over all problems in the suite.

of fixed-size corresponding to each of the categories defined in Sec. VII-C. This ensures that the sampling process collects a diverse set of samples spanning all cases of ill-conditioning. All captured frames were compiled into a fixed data-set of $N_p = 13298$ dual problems stored in the HDF5 file format [104], similar to FCLIB. Each problem entry p consists of a problem definition tuple $\{\mathbf{D}, \mathbf{v}_f, \boldsymbol{\mu}\}_p$ as well as additional meta-data that includes its sample categorization defined in Sec. VII-C, its ill-conditioning properties in terms of those defined in Sec. VII-B, and individual constraint dimensionalities. Table III outlines the data contents of each dual problem sample. The total relative performance of each solver aggregated over all systems in the suite is presented in Fig. 35. Moreover, we provide additional aggregations per system categories and sample categories in Appendix. D.

F. Discussion

We will conclude this section with a summary of the observations made throughout the course of this evaluation.

Collision Detection:

- One aspect of ODE that we found particularly challenging to deal with is that *box-to-plane* collisions yield contacts that are spuriously re-assigned at each time-step-wise query. This resulted in a re-assignment of the index order of each contact constraint, leading to respectively erratic contact reactions for the splitting-based solvers, which are sensitive to this ordering. We thus conclude that when using this type of solver, one must pay close attention to the aforementioned behavior and ensure that a CD pipeline can render the same contact over sequential invocations.

- Employing a collision detection margin is crucial to stabilizing contacts. One challenge of co-planar contacts, such as those present in the Primitive systems but in Walker as well, is that minuscule contact velocities along the respective normal directions will cause separation of the bodies and contact chattering. Defining a collision detection margin, either in the CD pipeline itself, or as a threshold in the Baumgarte-like contact constraint stabilization term, can alleviate this effect. By permitting some marginal body interpenetration, e.g. $\delta = 10^{-6}$, contact chattering can be eliminated without significantly impacting the overall results of the simulation.

- When evaluating collisions between bodies with mesh-based geometries an inordinate number of contacts may be rendered, which can be detrimental to simulation performance. A systematic method is thus required that can reduce these to a minimal number per-body pair. However, it is open problem as to how to define an minimal number of contacts that optimally represent the collision. We found that a naive approach using an ad-hoc filtering works very well in practice, but this can still render more contacts than required, as demonstrated on the Walker system. There, when each foot contacted the ground, up to $n_c = 20$ contacts occurred on each planar foot, reduced from approximately $n_c = 200$ using

the minimum-distance rejection sampling-based scheme.

Splitting-Based Solvers:

- Of all splitting-based solver, RAISIM-DS and RAISIM-DS-ES performed the best, with significant differences w.r.t vanilla RAISIM, and NBGS on which they are based. This result indicates that the inclusion of the De Saxcé correction significantly improves the quality of the simulation w.r.t all accuracy metrics. Moreover, this result also speaks in favor of preferring holistic local contact models and solvers instead of simpler blocked methods such as PGS-CCP and PGS-NCP.

- RAISIM-DS-ES was the clear overall winner among splitting-based solvers. Although the early-stopping criteria can transiently manifest artifacts due to under-estimation of the contact reactions, using constraint stabilization can provide sufficient correction to render results nearly indistinguishable from the more accurate solvers that ran for more iterations. We must credit the authors of MuJoCo for this particular improvement, as we did not encounter it in the various other works we reviewed. A more in-depth analysis for why it works so well is warranted, however.

- The inclusion of the De Saxcé correction in the PGS-CCP was exceptionally easy to incorporate into the original PGS-CCP solver described by Tasora et al in [80], and rendered overall better results than PGS-NCP. This indicates that a decoupling of the normal and tangential dynamics of each local contact model is not as effective as enforcing duality between the augmented contact velocity $\hat{\mathbf{v}}$ and the contact reaction $\boldsymbol{\lambda}$.

  Furthermore, the DS-corrected PGS-CCP solver seems to perform almost as well as RAISIM-DS w.r.t the dual r_d and natural-map r_{nat} accuracy metrics, while being computationally cheaper. This means that PGS-CCP is still a viable candidate for applications where real-time performance requirements may outweigh solver accuracy.

- As note by Tasora et al in [80] and Sheldon et al in [17], the order in which constraints are updated in splitting-based solvers is crucial for convergence and solution fidelity. We experimented with various combinations for ordering the constraint sets of joints, limits and contacts, and found that the best results were rendered by the order defined in Alg. 5, where first bilateral joint constraints are updated, then limits and contact constraints last.

- Sheldon et al in [17] also describe additional augmentations to the per-constraint iteration, in the form of dual forward-backward and randomization of the indexing. We experimented with these modifications by respectively augmenting Alg. 5. In the former case, we observed no substantive improvement to convergence or difference in the rendered solution. In the latter case, we also randomized the indices of each constraint subset, and did not observe any tangible improvement. In fact, the randomization leads to contact force instabilities as the directions of the corresponding forces along the respective tangential directions changed erratically due to the reordering.

ADMM:

- ADMM-NCP proved to be the best performing solver overall. Indeed, it was the only one capable of rendering the expected, and most physically plausible, simulations for the primitive systems. Most importantly, it demonstrated the best performance on the highly-coupled systems like Fourbar, Walker and IronMan and w.r.t all accuracy metrics. Moreover, it was the most robust w.r.t using low tolerances and restricting the number to iterations so that simulations could be run in real-time.

- The proximal regularization in ADMM-NCP proved to be exceptionally effective against even severe ill-conditioning of the dual problem, particularly hyperstatic cases where the system Jacobian was significantly rank-deficient. As an added benefit, the regularization also eliminates internal forces to a very large degree, rendering minimum-norm contact reactions, which are crucial aspects of physically plausible simulations in which direct measurements of these might be necessary for control applications.

- ADMM-NCP was also the most robust w.r.t inertial disparity large mass-ratios. Our analysis in Sec. VIII-C demonstrated that, in fact, ADMM-NCP is sensitive mainly w.r.t the spectral radius $L = \rho(\mathbf{D})$, of the Delassus matrix, which is also the Lipschitz constant of the NSOCP (16) objective function. This result indicates that its convergence properties could be further improved given a pre-conditioning of the problem, such as that proposed by Tasora et al in [45]. Lastly, we remark that the spectral adaptation strategy of the penalty parameter proposed by Carpentier et al in [32] provide an additional mechanism to accelerate convergence, that could potentially be combined with the former.

- Unfortunately, our analysis caused us to conclude that ADMM-CCP is unpredictably unstable. Although performing comparably close to ADMM-NCP on most problems, it diverged in particular instances on several of the applications-relevant systems such as Fourbar and Walker. This led us to exclude it altogether from several of the aforedescribed experiments. We are unsure as to why and when instabilities arise, as the exclusion of the De Saxcé correction should not, in principle, impact the solver directly. This warrants further investigation.

Solver Comparisons:

- Although ADMM-NCP was the best solver overall,

RAISIM-DS, RAISIM-DS-ES and PGS-CCP proved to perform comparatively well. Our experiments demonstrate that in terms of raw simulation throughput, RAISIM-DS-ES was the best, and is a viable candidate for lower-fidelity simulations where accuracy is not as important as speed.
On many problems RAISIM-DS and PGS-CCP performed better than ADMM-NCP w.r.t the NCP complementarity residual metric r_{ncp}. We provide such an example of these differences in Sec. VIII-D were we presented experiments on the Walker system. This result was rather surprising, given the overall better behavior of ADMM-NCP and failure of the others to perform well in cases of severe ill-conditioning. We speculate that this may be due to the lack of samples in the large-scale systems that include sliding contacts. We know from the primitive experiments involving slipping contacts that all solvers exhibit both degraded convergence and small residual velocities along contact normal directions, with ADMM-NCP performing the best in these cases, and RAISIM-DS behaving comparably close.

- Our results are generally in strong agreement with those presented by Lidec et al in [73] and [38]. Moreover, the former work inspired us to consider the introduction of the De Saxcé correction to all compatible solvers. We argue that such a trivial modification should definitely be a considered for applications where differentiability is not necessary.

Constraint Stabilization & Softening:

- Experiments that included constraint stabilization and softening to the dual problem were in strong agreement with the theoretical result presented in Sec. V-C, were we should the necessary conditions for an augmentation to retain the properties of the NCP. Our results show that the augmentations are effective means to improve simulation throughput and fidelity in such a minimal-coordinate formulation.

- When the mass ratios are relatively low, or close to unity, the MuJoCo-based constraint softening proves exceptionally effective in improving convergence and therefore simulation throughput. However, as we saw with the Boxes-Fixed experiment, high mass-ratios make it effectively impossible to use such a relaxation, even with extraneous tuning of the softening parameters.

- Our results indicate that it is both viable and effective to combine classic Baumgarte-like constraint stabilization with the MuJoCo-based constraint softening. Indeed, their combination can bring the benefits of each and thus accelerate simulation throughput while minimizing configuration-level constraint violation.

IX. Conclusion

This work has presented an extensive analysis of constrained rigid multi-body simulation realized using the architecture of separately solving the forward dynamics problem and performing time-integration of the state. We have provided a rigorous literature survey to build as complete as possible of a picture of past and recent recent directions. In addition, we have employed fundamental and advanced principles of mechanics to derive and construct the dual problem of forward dynamics in the form of a NCP that encompasses a majority of the complex mechanical systems that one would need to simulate in the domains of robotics and *Audio-Animatronics*® . Through the lens of convex analysis, we have employed the theory of the augmented Lagrange method and proximal operators to depict a unifying perspective of strategies for solving the dual problem of forward dynamics, and have presented a summary of established and current state-of-the-art methods.

We have contributed a framework for performing comparative analyses of the aforementioned solver strategies, founded on fundamental physical principles and an broad inclusion of systems that are relevant for practical applications. Lastly, we have presented an extensive set of experiments aiming to elucidate both qualitative and quantitative differences between solvers, on both toy and application-level problems, and present our findings to the community. We hope that this work, as the ones that inspired it, can guide others in developing the next generation of physics simulators that can further the fields of robotics, character animation, and computer graphics.

Our experiments have lead us to conclude that both state-of-the-art methods such as ADMM as well as established approaches based on splitting-methods such as Gauss-Seidel iteration, can be made competitive on all forms of problems, given small modifications to the formulation of the problem, augmentations to stabilize it, and a careful selection of algorithm termination criteria. Most notable is how the inclusion of the so-called De Saxcé correction is nearly trivial, yet exceptionally effective in improving simulation accuracy and fidelity, and with negligible added computational cost.

Although we attempted to incorporate as many of the important aspects of realizing forward dynamics solvers for physical simulation, there were aspects which we did not have yet consider. These include introcution of an implicit PD controller, the application of warm-starting, pre-conditioning, and parameter adaptation that can accelerate the numerical convergence of all solvers for whom these are applicable. Moreover, we did not consider the effects of sparse matrix algebra, and intra-operation parallelization, e.g. graph-coloring and parallelization, as means to accelerate the computational time-complexity of our implementations. Lastly, we did not consider the family of approaches tackling the *primal* formulation of the forward dynamics problem, and most notably those employing Newton-based second-order methods such as those employed in Drake [105] and MuJoCO. All of the aforementioned, as well as an implementation capable of executing on the GPU, are part of future work.

Appendix A
Properties of the De Saxcé Operator

Definition A.1. In the constraint-space of each contact, we can decompose any vector $\mathbf{v} \in \mathbb{R}^3$ into its 2D tangential and scalar normal components in the form of

$$\mathbf{v} := \begin{bmatrix} \mathbf{v}_T \\ \mathrm{v}_N \end{bmatrix} . \tag{180}$$

Thus, we can decompose the post-event contact velocity as

$$\mathbf{v}^+ := \begin{bmatrix} \mathbf{v}_T^+ \\ \mathrm{v}_N^+ \end{bmatrix} , \tag{181}$$

and the augmented contact velocity as

$$\begin{aligned} \hat{\mathbf{v}} &:= \mathbf{v}^+ + \boldsymbol{\Gamma}_\mu(\mathbf{v}^+) \\ &= \begin{bmatrix} \mathbf{v}_T^+ \\ \mathrm{v}_N^+ \end{bmatrix} + \begin{bmatrix} 0 \\ \mu \left\| \mathbf{v}_T^+ \right\|_2 \end{bmatrix} . \end{aligned} \tag{182}$$

A. Velocity Constraint Equivalence

Lemma A.1. *The post-impact velocity $\mathbf{v}^+$ lies in the half-space defined by the unit vector $\mathbf{n} \in \mathbb{R}^3$ of the contact normal, if and only if, the augmented velocity lies in the dual cone $\mathcal{K}^*$, i.e. the following conditions are equivalent:*

$$v_N^+ \geq 0 \;\Leftrightarrow\; \hat{\mathbf{v}} \in \mathcal{K}_\mu^* \tag{183}$$

Proof. Since the dual cone of the Coulomb friction cone $\mathcal{K}_\mu$ is $\mathcal{K}^* \equiv \mathcal{K}_{\mu^{-1}}$, then any vector $\mathbf{x} \in \mathcal{K}^*$ satisfies the relations

$$\left\| \mathbf{x}_T \right\|_2 \leq \frac{1}{\mu} \mathrm{x}_N \,,\; \mathrm{x}_N \geq 0 \;. \tag{184}$$

Applying (184) to the decomposition (182) of $\hat{\mathbf{v}}$, we see that

$$\begin{aligned} \hat{\mathbf{v}} \in \mathcal{K}^* \;\Leftrightarrow\; & \left\| \mathbf{v}_T^+ \right\|_2 \leq \tfrac{1}{\mu} \left(\mathrm{v}_N^+ + \mu \left\| \mathbf{v}_T^+ \right\|_2 \right) \\ & ,\; \mathrm{v}_N^+ + \mu \left\| \mathbf{v}_T^+ \right\|_2 \geq 0 \end{aligned} . \tag{185}$$

The $\mu \left\| \mathbf{v}_T^+ \right\|_2$ terms cancel out in the RHS of the first inequality, leading to a simplification in the form of

$$\begin{aligned} \hat{\mathbf{v}} \in \mathcal{K}^* \;\Leftrightarrow\; & 0 \leq \mathrm{v}_N^+ \\ & ,\; \mathrm{v}_N^+ + \mu \left\| \mathbf{v}_T^+ \right\|_2 \geq 0 \;. \end{aligned}$$

Noting that the second inequality is now trivially satisfied, it becomes redundant, and removing it yields the equivalence

$$\hat{\mathbf{v}} \in \mathcal{K}_\mu^* \;\Leftrightarrow\; \mathrm{v}_N^+ \geq 0$$

□

Remark. Interpreting these relations as constraints, the equivalence implies that they must yield the same solution.

B. Invariance of the De Saxcé Operator

Lemma A.2. *The De Saxcé operator is invariant to differences along the normal direction, i.e. for vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$ differing by an offset in the form of*

$$\mathbf{v}_2 := \mathbf{v}_1 + \mathbf{v}_b \;,\; \mathbf{v}_b = \begin{bmatrix} 0 \\ v_{b,N} \end{bmatrix} ,$$

where $v_{b,N} \in \mathbb{R}$, then it holds that

$$\boldsymbol{\Gamma}_\mu(\mathbf{v}_1^+) = \boldsymbol{\Gamma}_\mu(\mathbf{v}_2^+) \tag{186}$$

Proof. Using the decomposition (180), the vectors $\mathbf{v}_1, \mathbf{v}_2$ are expressed using their tangential and normal components as

$$\mathbf{v}_2 = \begin{bmatrix} \mathbf{v}_{2,T} \\ \mathrm{v}_{2,N} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{1,T} \\ \mathrm{v}_{1,N} + \mathrm{v}_{b,N} \end{bmatrix} ,$$

and applying the De Saxcé operator to the first, yields

$$\boldsymbol{\Gamma}_\mu(\mathbf{v}_2) = \begin{bmatrix} 0 \\ \mu \left\| \mathbf{v}_{2,T} \right\|_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mu \left\| \mathbf{v}_{1,T} \right\|_2 \end{bmatrix} = \boldsymbol{\Gamma}_\mu(\mathbf{v}_1) \,.$$

□

Corollary A.2.1. *Lemma A.2 allows all biases to the augmented contact velocity $\hat{\mathbf{v}}$ to be absorbed as biases to the free-velocity $\mathbf{v}_f$. Expressing the post-impact contact velocity as a function of the constraint reaction $\boldsymbol{\lambda} \in \mathbb{R}^3$ in the form of*

$$\mathbf{v}^+(\boldsymbol{\lambda}) := \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f \;,$$

and subsequently the augmented contact velocity as

$$\hat{\mathbf{v}}(\boldsymbol{\lambda}) := \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) \;,$$

then any bias velocity included in the definition of the complementarity conditions asserted between $\hat{\mathbf{v}}(\boldsymbol{\lambda})$ and $\boldsymbol{\lambda}$, can be transferred to $\mathbf{v}_f$. For example, the introduction of Newton's model of restitutive impacts, incurs the bias

$$\mathbf{v}_N^- = \begin{bmatrix} 0 & 0 & v_N^- \end{bmatrix}^T \;,$$

which we use to re-state the Signorini-Coulomb-Newton NCP model (55), here recapitulated as

$$\mathcal{K}_\mu^* \ni \hat{\mathbf{v}}(\boldsymbol{\lambda}) + \mathbf{v}_N^- \perp \boldsymbol{\lambda} \in \mathcal{K}_\mu \;.$$

Lemma A.2 essentially allows us to re-state the problem as

$$\begin{aligned} \mathbf{v}'^+(\boldsymbol{\lambda}) &:= \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \mathbf{v}_N^- \\ \hat{\mathbf{v}}'(\boldsymbol{\lambda}) &:= \mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f + \boldsymbol{\Gamma}(\mathbf{v}'^+(\boldsymbol{\lambda})) \\ \mathcal{K}_\mu^* &\ni \hat{\mathbf{v}}'(\boldsymbol{\lambda}) \perp \boldsymbol{\lambda} \in \mathcal{K}_\mu \;. \end{aligned}$$

Appendix B
Derivation of the Dual Forward Dynamics NCP

Theorem B.1. *The dual FD problem stated as the NSOCP*

$$\boldsymbol{\lambda} = \underset{\mathbf{x} \in \mathcal{K}}{\operatorname{argmin}} \quad \tfrac{1}{2} \mathbf{x}^T \mathbf{D}\,\mathbf{x} + \mathbf{x}^T \mathbf{v}_f + \mathbf{x}^T \boldsymbol{\Gamma}(\mathbf{v}^+(\mathbf{x})) \tag{187}$$

is the Lagrange dual of the primal FD problem

$$\begin{aligned} \mathbf{u}^+ = \underset{\mathbf{x}}{\operatorname{argmin}} \; & \frac{1}{2} \|\mathbf{x} - \mathbf{u}_f\|_{\mathbf{M}} \\ \textit{s.t.} \quad & \mathbf{J}_J\,\mathbf{x} = 0 \\ & \mathbf{J}_L\,\mathbf{x} \geq 0 \\ & \mathbf{J}_{C,N}\,\mathbf{x} \geq 0 \quad , \end{aligned} \tag{188}$$

when the constraint reactions are additionally required to adhere to the Coulomb friction cone model. $\mathbf{u}_f :=$

$\mathbf{M}^{-1}\,(\mathbf{u}^- + \Delta t\,\mathbf{h})$*, and* $\mathbf{J}_{C,N}$ *is a sub-matrix of the contact constraint Jacobian matrix* $\mathbf{J}_C$ *constructed from the rows corresponding to the contact normal directions.*

Remark. The primal problem (188) is a re-formulation of the forward dynamics NCP (10c) as an optimization problem, that is transcribed using the *extended principle of least constraint* and *the duality principle of Gauss*. See Glocker [23] for more details regarding these principles. The resulting primal FD problem is a linearly-constrained QP directly in the post-impact generalized velocities of the system, and is more general than (187), as it holds for any model of friction.

Proof. Starting from (188), we first apply Lemma A.2 to each individual unilateral contact constraint to form the equivalence

$$\mathbf{J}_{C,N}\,\mathbf{u}^+ = \mathbf{v}^+_{C,N} \geq 0 \;\Leftrightarrow\; \hat{\mathbf{v}} \in \mathcal{K}^*_{\boldsymbol{\mu}} \quad , \tag{189}$$

which enables the substitution

$$\begin{aligned} \mathbf{u}^+ = \underset{\mathbf{x}}{\operatorname{argmin}}\; & \frac{1}{2}\|\mathbf{x} - \mathbf{u}_f\|_{\mathbf{M}} \\ \text{s.t.} \quad & \mathbf{J}_J\,\mathbf{x} = 0 \\ & \mathbf{J}_L\,\mathbf{x} \geq 0 \\ & \mathbf{J}_C\,\mathbf{x} + \boldsymbol{\Gamma}(\mathbf{J}_C\,\mathbf{x}) \in \mathcal{K}^*_{\boldsymbol{\mu}} \quad . \end{aligned} \tag{190}$$

Remark. This step has implicitly introduced the Coulomb friction model via the definition of the augmented velocity.

The next step involves the technique of introducing *slack variables* to decouple the velocity terms from the inequalities and set-inclusions. This yields the equivalent problem

$$\begin{aligned} \mathbf{u}^+ = \underset{\mathbf{x},\mathbf{y},\mathbf{z}}{\operatorname{argmin}}\; & \frac{1}{2}\|\mathbf{x} - \mathbf{u}_f\|_{\mathbf{M}} \\ \text{s.t.} \quad & \mathbf{J}_J\,\mathbf{x} = 0 \\ & \mathbf{y} = \mathbf{J}_L\,\mathbf{x} \\ & \mathbf{y} \in \mathbb{R}^{n_l}_+ \\ & \mathbf{z} = \mathbf{J}_C\,\mathbf{x} + \boldsymbol{\Gamma}(\mathbf{J}_C\,\mathbf{x}) \\ & \mathbf{z} \in \mathcal{K}^*_{\boldsymbol{\mu}} \quad , \end{aligned} \tag{191}$$

where $\mathbf{y}\,,\ \mathbf{z}$ are the slack variables corresponding to the limit and contact constraints. This decoupling enables us to remove the inequalities and set-inclusions using indicator functions as

$$\begin{aligned} \mathbf{u}^+ = \underset{\mathbf{x},\mathbf{y},\mathbf{z}}{\operatorname{argmin}}\; & \frac{1}{2}\|\mathbf{x} - \mathbf{u}_f\|_{\mathbf{M}} + \Psi_{\mathbb{R}^{n_l}_+}(\mathbf{y}) + \Psi_{\mathcal{K}^*_{\boldsymbol{\mu}}}(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{J}_J\,\mathbf{x} = 0 \\ & \mathbf{y} - \mathbf{J}_L\,\mathbf{x} = 0 \\ & \mathbf{z} - \mathbf{J}_C\,\mathbf{x} - \boldsymbol{\Gamma}(\mathbf{J}_C\,\mathbf{x}) = 0 \quad . \end{aligned} \tag{192}$$

The primal problem (192) now only contains equality constraints, for which defining the vector Lagrange multipliers $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_J^T\ \boldsymbol{\lambda}_L^T\ \boldsymbol{\lambda}_C^T]^T$, we form the Lagrangian functional

$$\begin{aligned} \mathcal{L}(\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\lambda}) := \;& \tfrac{1}{2}\|\mathbf{x} - \mathbf{u}_f\|_{\mathbf{M}} + \Psi_{\mathbb{R}^{n_l}_+}(\mathbf{y}) + \Psi_{\mathcal{K}^*_{\boldsymbol{\mu}}}(\mathbf{z}) \\ & - \boldsymbol{\lambda}_J^T\,\mathbf{J}_J\,\mathbf{x} - \boldsymbol{\lambda}_L^T\,(\mathbf{J}_L\,\mathbf{x} - \mathbf{y}) \\ & - \boldsymbol{\lambda}_C^T\,(\mathbf{J}_C\,\mathbf{x} + \boldsymbol{\Gamma}(\mathbf{J}_C\,\mathbf{x}) - \mathbf{z}) \quad , \end{aligned} \tag{193}$$

and thus the post-event generalized velocities $\mathbf{u}^+$ are solutions of the *saddle-point problem*

$$\mathbf{u}^+ = \operatorname{argm}\min_{\mathbf{x},\mathbf{y},\mathbf{z}}\max_{\boldsymbol{\lambda}}\ \mathcal{L}(\mathbf{x},\mathbf{y},\mathbf{z},\boldsymbol{\lambda})\ . \tag{194}$$

At this point we will begin to introduce what we already know from mechanics in order to start removing variables and terms from (193). First of all, we know that the EoM (74) result from the gradient $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*,\mathbf{y},\mathbf{z},\boldsymbol{\lambda}) = 0$. Exploiting this fact, we substitute $\mathbf{x}^* = \mathbf{u}^+$ in (193), and use (74) to expand all quadratic terms. This results in a crucial property of the objective function (188), which conveniently reduces it to

$$\frac{1}{2}\|\mathbf{u}^+ - \mathbf{u}_f\|_{\mathbf{M}} = \frac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} = \frac{1}{2}\|\boldsymbol{\lambda}\|_{\mathbf{D}} \quad . \tag{195}$$

Substituting (195) in (193), the Lagrangian becomes

$$\begin{aligned} \mathcal{L}(\mathbf{u}^+,\mathbf{y},\mathbf{z},\boldsymbol{\lambda}) \;& = \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} + \Psi_{\mathbb{R}^{n_l}_+}(\mathbf{y}) + \Psi_{\mathcal{K}^*_{\boldsymbol{\mu}}}(\mathbf{z}) \\ & - \boldsymbol{\lambda}_J^T\,\mathbf{J}_J\,\mathbf{u}^+ - \boldsymbol{\lambda}_L^T\,(\mathbf{J}_L\,\mathbf{u}^+ - \mathbf{y}) \\ & - \boldsymbol{\lambda}_C^T\,(\mathbf{J}_C\,\mathbf{u}^+ + \boldsymbol{\Gamma}(\mathbf{J}_C\,\mathbf{u}^+) - \mathbf{z}) \end{aligned} \quad . \tag{196}$$

Now comes one of the most difficult and subtle parts of this derivation. By duality of the cones $\mathbb{R}^n_+ \leftrightarrow \mathbb{R}^n_+$ and $\mathcal{K}_{\boldsymbol{\mu}} \leftrightarrow \mathcal{K}^*_{\boldsymbol{\mu}}$, as well as the definition (115) of their respective *support functions*, when $\Psi_{\mathcal{K}^*_{\boldsymbol{\mu}}}(\mathbf{z}) = 0$ and $\Psi_{\mathcal{K}_{\boldsymbol{\mu}}}(\boldsymbol{\lambda}_C) = 0$, then necessarily $\min_{\mathbf{z}} \boldsymbol{\lambda}_C^T\,\mathbf{z} = 0$. These observations result in the following equivalences between the primal and dual sub-problems for the limit and contact constraints:

$$\begin{aligned} \min_{\mathbf{y}} \Psi_{\mathbb{R}^{n_l}_+}(\mathbf{y}) \;&\Leftrightarrow\; \max_{\boldsymbol{\lambda}_L} -\Psi_{\mathbb{R}^{n_l}_+}(\boldsymbol{\lambda}_L) \\ \min_{\mathbf{z}} \Psi_{\mathcal{K}^*_{\boldsymbol{\mu}}}(\mathbf{z}) + \boldsymbol{\lambda}_C^T\,\mathbf{z} \;&\Leftrightarrow\; \max_{\boldsymbol{\lambda}_C} -\Psi_{\mathcal{K}_{\boldsymbol{\mu}}}(\boldsymbol{\lambda}_C) \end{aligned} \tag{197}$$

Thus, when $\mathbf{y}^* = \mathbf{v}_L^+$ and $\mathbf{z}^* = \hat{\mathbf{v}}_C$, and exploiting (197), the problem is effectively equivalent to

$$\begin{aligned} \mathcal{L}(\mathbf{u}^+,\mathbf{v}_L^+,\hat{\mathbf{v}}_C,\boldsymbol{\lambda}) \;& = \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} \\ & -\Psi_{\mathbb{R}^{n_l}_+}(\boldsymbol{\lambda}_L) - \Psi_{\mathcal{K}_{\boldsymbol{\mu}}}(\boldsymbol{\lambda}_C) \\ & - \boldsymbol{\lambda}_J^T\,\mathbf{J}_J\,\mathbf{u}^+ - \boldsymbol{\lambda}_L^T\,\mathbf{J}_L\,\mathbf{u}^+ \\ & - \boldsymbol{\lambda}_C^T\,\mathbf{J}_C\,\mathbf{u}^+ - \boldsymbol{\lambda}_C^T\boldsymbol{\Gamma}(\mathbf{J}_C\,\mathbf{u}^+) \quad . \end{aligned} \tag{198}$$

We can simplify (198) by grouping all linear terms using the definition of the total Jacobian matrix $\mathbf{J}$ to form

$$\begin{aligned} \mathcal{L}(\mathbf{u}^+,\mathbf{v}_L^+,\hat{\mathbf{v}}_C,\boldsymbol{\lambda}) \;& = -\tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} \\ & -\Psi_{\mathbb{R}^{n_l}_+}(\boldsymbol{\lambda}_L) - \Psi_{\mathcal{K}_{\boldsymbol{\mu}}}(\boldsymbol{\lambda}_C) \\ & -\boldsymbol{\lambda}^T\,\mathbf{J}\,\mathbf{u}^+ - \boldsymbol{\lambda}_C^T\boldsymbol{\Gamma}(\mathbf{v}_C^+(\boldsymbol{\lambda})) \quad , \end{aligned} \tag{199}$$

and then use (76) to substitute $\mathbf{u}^+$, resulting in

$$\begin{aligned}\mathcal{L}(\mathbf{u}^+, \mathbf{v}_L^+, \hat{\mathbf{v}}_C, \boldsymbol{\lambda}) &= \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} \\ &\quad -\Psi_{\mathbb{R}_+^{n_l}}(\boldsymbol{\lambda}_L) - \Psi_{\mathcal{K}_{\boldsymbol{\mu}}}(\boldsymbol{\lambda}_C) \\ &\quad -\boldsymbol{\lambda}^T\,\mathbf{D}\,\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\,\mathbf{v}_f \\ &\quad -\boldsymbol{\lambda}_C^T\boldsymbol{\Gamma}(\mathbf{v}_C^+(\boldsymbol{\lambda}))\ \ ,\end{aligned} \tag{200}$$

which is now only a function of the Lagrange multipliers $\boldsymbol{\lambda}$. Before proceeding with the last step, simplifying the quadratic terms, brings us to the dual maximization problem

$$\begin{aligned}\max_{\boldsymbol{\lambda}} \quad & -\tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\,\mathbf{v}_f - \boldsymbol{\lambda}_C^T\boldsymbol{\Gamma}(\mathbf{v}_C^+(\boldsymbol{\lambda})) \\ & -\Psi_{\mathbb{R}_+^{n_l}}(\boldsymbol{\lambda}_L) - \Psi_{\mathcal{K}_{\boldsymbol{\mu}}}(\boldsymbol{\lambda}_C)\ \ .\end{aligned} \tag{201}$$

The final step involves using the definition of the total cone

$$\mathcal{K} := \mathbb{R}^{n_{jd}} \times \mathbb{R}_+^{n_l} \times \mathcal{K}_{\boldsymbol{\mu}}\ ,$$

and inversion of the sign of the objective function to render a minimization problem. These at last yield the Lagrange dual of the primal FD problem, in the form of

$$\min_{\boldsymbol{\lambda}\in\mathcal{K}} \quad \tfrac{1}{2}\boldsymbol{\lambda}^T\mathbf{D}\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\,\mathbf{v}_f + \boldsymbol{\lambda}^T\boldsymbol{\Gamma}(\mathbf{v}^+(\boldsymbol{\lambda})) \tag{202}$$

□

Appendix C
Solving of the Single-Contact Problem

A. A Semi-Analytical Approach

This derivation is based on that presented by Preclik et al in [72], [56], and assumes only a single contact is present. Thus we can assume that $\mathbf{D} \in \mathbb{S}_{++}^3$, $\mathbf{v}_f \in \mathbb{R}^3$, and $\mathcal{K}_\mu \subset \mathbb{R}^3$. The dual FD problem can be stated

$$\begin{aligned}\min_{\boldsymbol{\lambda}} \quad & \tfrac{1}{2}\,\boldsymbol{\lambda}^T\,\mathbf{D}\,\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\,(\mathbf{v}_f + \mathbf{s}) \\ \text{s.t.} \quad & \boldsymbol{\lambda} \in \mathcal{K}_\mu \\ & \mathrm{v}_{c,N}^+(\boldsymbol{\lambda}) \geq 0 \\ & \lambda_N^T\,\mathrm{v}_{c,N}^+(\boldsymbol{\lambda}) = 0\end{aligned}\ , \tag{203}$$

where the Signorini conditions are to be enforced explicitly, as opposed to be assumed to hold only at the solution. Since we are only dealing with a single contact, we can express the problem definition using the coefficients of the Delassus matrix and the free velocity as

$$\mathbf{D} := \begin{bmatrix} D_{tt} & D_{to} & D_{tn} \\ D_{ot} & D_{oo} & D_{on} \\ D_{nt} & D_{no} & D_{nn}\end{bmatrix}\ ,\ \mathbf{v}_f := \begin{bmatrix}\mathrm{v}_{f,t} \\ \mathrm{v}_{f,o} \\ \mathrm{v}_{f,n}\end{bmatrix} \tag{204}$$

The total, tangential and normal contact reactions expressed in local coordinates are respectively

$$\boldsymbol{\lambda} := \begin{bmatrix}\lambda_t \\ \lambda_o \\ \lambda_n\end{bmatrix}\ ,\ \boldsymbol{\lambda}_T := \begin{bmatrix}\lambda_t \\ \lambda_o \\ 0\end{bmatrix}\ ,\ \boldsymbol{\lambda}_N := \begin{bmatrix}0 \\ 0 \\ \lambda_n\end{bmatrix}\ . \tag{205}$$

Next we will project the Delassus matrix and the free velocity onto the local axis of the contact normal $\mathbf{n}$, here denoted as $e_n = [0\,0\,1]^T$, to form

$$\mathbf{d}_n := e_n^T\,\mathbf{D} = \begin{bmatrix} D_{tn} \\ D_{on} \\ D_{nn}\end{bmatrix} \quad \mathrm{v}_{f,n} = e_n^T\,\mathbf{v}_f\ , \tag{206}$$

to simplify the expression of the post-event contact velocity

$$\mathbf{v}_{c,N}^+ = e_n^T\,(\mathbf{D}\,\boldsymbol{\lambda} + \mathbf{v}_f) = \mathbf{d}_n^T\,\boldsymbol{\lambda} + \mathrm{v}_{f,n} \geq 0\ . \tag{207}$$

Thus, the single-contact dual problem is rewritten as

$$\begin{aligned}\min_{\boldsymbol{\lambda}} \quad & \tfrac{1}{2}\,\boldsymbol{\lambda}^T\,\mathbf{D}\,\boldsymbol{\lambda} + \boldsymbol{\lambda}^T\,\mathbf{v}_f \\ \text{s.t.} \quad & \lambda_n \geq 0 \\ & \mathbf{d}_n^T\,\boldsymbol{\lambda} \geq 0 \\ & \mathbf{d}_n^T\,\boldsymbol{\lambda} + \mathrm{v}_{f,n} \geq 0 \\ & \|\boldsymbol{\lambda}_T\|_2 \leq \mu\,\lambda_n\ \ .\end{aligned} \tag{208}$$

A key aspect of this derivation is the consideration of the geometry that is implied by (208), as depicted in Fig.36. The friction cone constraint $\|\boldsymbol{\lambda}_T\|_2 \leq \mu\,\lambda_n$ defines a conic surface boundary that extends in both half-spaces of the normal axis. The unilateral force constraint $\lambda_n \geq 0$ requires us to exclude the cone below the tangential plane. The inequality $\mathbf{d}_n^T\,\boldsymbol{\lambda} + \mathrm{v}_{f,n} \geq 0$ of the unilateral contact hypothesis defines a plane that forms a closed upper bound on the contact reaction along the normal axis. If we know that the contact will remain closed, or in slip, then we can expect the contact reaction to necessarily lie on this plane, because (207) becomes an equality constraint (i.e. the constraint is active). The parameters (206) fully specify this plane, as $\mathbf{d}_n$ is parallel to the plane normal, and $\boldsymbol{\lambda}_{T=0} = [0\,0\,D_{nn}^{-1}\,\mathrm{v}_{f,n}]^T$ defines the point at which the plane intersects the normal axis. A mechanical interpretation of this plane serving as an upper-bound on the normal contact reaction gives us cause to refer to it as the *plane of maximum compression* $\mathcal{P}_{mc}$.

Now, let us also consider when and how this plane intersects the friction cone $\mathcal{K}_\mu$. If the contact is sliding, then we know that the contact reaction will lie on the boundary $\partial\mathcal{K}_\mu$. This means that the *conic section* $\mathcal{P}_{mc} \cap \partial\mathcal{K}_\mu$ defines the feasible set for this case. Thus, we can examine the possible contact states m of the solution to (208), and what they imply geometrically. For this analysis, we also need the unconstrained solution

$$\boldsymbol{\lambda}_0 = -\mathbf{D}^{-1}\,\mathbf{v}_f\ , \tag{209}$$

Note that (209) always holds, since in the single-contact case, $\mathbf{D}$ is always invertible[18]. The three contact states defined in Sec.IV-D are:

- m = Open: The Signorini conditions become $\mathbf{v}_{c,N}^+ > 0$, $\boldsymbol{\lambda}_N = 0$, and therefore $\boldsymbol{\lambda}_T = 0$ as well. $\boldsymbol{\lambda} \in \mathcal{K}_\mu$ (i.e. in the interior of the cone), and thus $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$. Using (207), we can reduce this case to $\mathrm{v}_{f,n} > 0$.

- m = Stick: The Signorini conditions become $\mathbf{v}_{c,N}^+ = 0$, $\boldsymbol{\lambda}_N > 0$, and due to Coulomb's law $\boldsymbol{\lambda}_T = 0$. $\boldsymbol{\lambda} \in \mathcal{K}_\mu$

[18] i.e. as there are no couplings that can cause it to become rank-deficient. Only the nonphysical cases of zero mass, singular moment of inertia, or an infinite lever-arm can cause it to become singular.

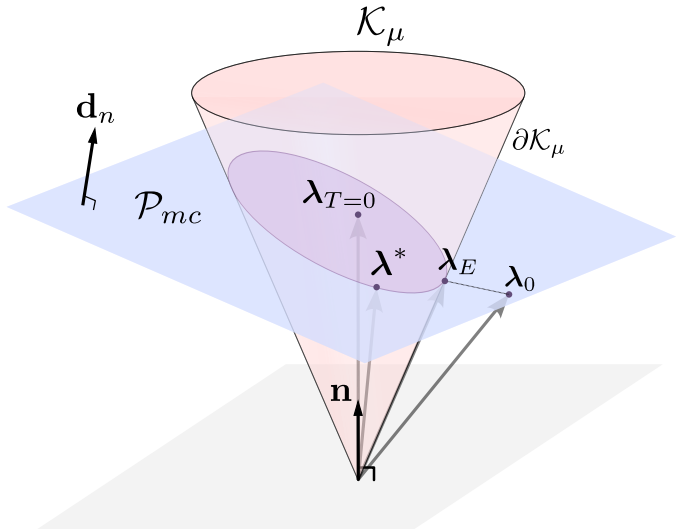

Fig. 36. A visualization of the geometry implied by the constraints of the single-contact dual problem (208). It depicts the contact normal $\mathbf{n}$, the tangential plane (gray), the friction cone $\mathcal{K}_\mu$ (red) with boundary $\partial\mathcal{K}_\mu$, the plane of maximum compression $\mathcal{P}_{mc}$ (blue), and the elliptic conic-section $\mathcal{P}_{mc} \cap \partial\mathcal{K}_\mu$ (purple). $\mathcal{P}_{mc}$ intersects the local normal axis at $\boldsymbol{\lambda}_{T=0}$. The unconstrained solution $\boldsymbol{\lambda}_0$ is projected by the MDP as $\boldsymbol{\lambda}^*$, while the naive projection using the Euclidean minimum-distance metric is denoted as $\boldsymbol{\lambda}_E$.

and thus $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$. Using (207), this reduces to $\mathrm{v}_{f,n} \leq 0$.

- m = Slip: The Signorini conditions become $\mathbf{v}^+_{c,N} = 0$, $\boldsymbol{\lambda}_N > 0$, and because of Coulomb's law $\|\boldsymbol{\lambda}_T\|_2 \leq \mu\,\lambda_n$. Therefore, the only conditions on the constrained and unconstrained solution are respectively $\boldsymbol{\lambda} \in \mathcal{P}_{mc} \cap \partial\mathcal{K}_\mu$ and $\boldsymbol{\lambda}_0 \in \mathcal{P}_{mc}$.

We now have a partial sketch of the semi-analytical solution. The open and sticking contact cases are solely determined by the given $\mathrm{v}_{f,n}$, where in both $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$. The only non-trivial case is that of slipping contacts. Employing the approach of Preclik et al from [56], we can apply the MDP, together with the geometry of conic sections, to re-phrase (208) for the slipping case and yield the maximally dissipative contact reaction.

As conic-sections can be expressed succinctly using *polar coordinates*, let's re-parameterize the contact reaction as

$$\lambda_t = r\,\cos\phi \tag{210a}$$

$$\lambda_o = r\,\sin\phi \tag{210b}$$

$$\lambda_n = -D_{nn}^{-1}\left(\mathrm{v}_{f,n} + D_{nt}\,r\,\cos\phi + D_{no}\,r\,\sin\phi\right)\,, \tag{210c}$$

$$\boldsymbol{\lambda}_p(r,\phi) = \begin{bmatrix}\lambda_t & \lambda_o & \lambda_n\end{bmatrix}^T \tag{210d}$$

where $r \in \mathbb{R}_+$ and $\phi \in \mathbb{R}$ are the radial and angular coordinates, respectively. Using (210) in $\|\boldsymbol{\lambda}_T\|_2 \leq \mu\,\lambda_n$ yeilds

$$r\left(D_{nn} + \mu\,D_{nt}\,\cos\phi + \mu\,D_{no}\,\sin\phi\right) \leq -\mu\,\mathrm{v}_{f,n}\,. \tag{211}$$

The multiplicand of r is the *radial scaling* function

$$f_r(\phi) := D_{nn} + \mu\,D_{nt}\,\cos\phi + \mu\,D_{no}\,\sin\phi \tag{212a}$$

$$= D_{nn} + \mu\sqrt{D_{nt}^2 + D_{no}^2}\,\cos\left(\phi - \tan^{-1}\left(\frac{D_{no}}{D_{nt}}\right)\right)\,, \tag{212b}$$

where $f_r : \mathbf{R} \to \mathbb{R}$. When $f_r(\phi) \leq 0$, (211) holds trivially. However, when $f_r(\phi) > 0$, we need to use (212b) to determine the *feasible range* of angles $\phi \in \mathcal{I}_F + 2\pi\mathbb{N}$, for which (211) holds, i.e.

$$\mathcal{I}_F = \begin{cases}(-\Delta\phi, \Delta\phi] + \phi_0, & ,D_{nn} \leq \mu\sqrt{D_{nt}^2 + D_{no}^2}\\ [0, 2\pi) & ,\textit{otherwise}\end{cases}\,, \tag{213a}$$

$$\Delta\phi = \cos^{-1}\frac{-D_{nn}}{\mu\sqrt{D_{nt}^2 + D_{no}^2}}\,,\ \phi_0 = \tan^{-1}\frac{D_{no}}{D_{nt}}\,. \tag{213b}$$

According to (211), $f_r(\phi)$ determines an upper-bound on r:

$$0 < r \leq \frac{-\mu\,\mathrm{v}_{f,n}}{\max(0, f_r(\phi))}\,. \tag{214}$$

By defining the polar radius upper-bound as the function

$$\bar{r}(\phi) := \frac{-\mu\,\mathrm{v}_{f,n}}{f_r(\phi)}\,, \tag{215}$$

we can express the tangential contact reaction as

$$\boldsymbol{\lambda}_T(\phi) := \bar{r}(\phi)\begin{bmatrix}\cos\phi\\ \sin\phi\end{bmatrix}\,. \tag{216}$$

Projecting the $\mathbf{D}$ and $\mathbf{v}_f$ onto the tangential plane as

$$\hat{\mathbf{D}} := \begin{bmatrix}D_{tt} - D_{nn}^{-1}\,D_{nt}^2 & D_{to} - D_{nn}^{-1}\,D_{nt}\,D_{no}\\ D_{to} - D_{nn}^{-1}\,D_{nt}\,D_{no} & D_{oo} - D_{nn}^{-1}\,D_{no}^2\end{bmatrix} \tag{217}$$

$$\hat{\mathbf{v}} := \begin{bmatrix}\mathrm{v}_{f,t} - D_{nn}^{-1}\,D_{nt}\,\mathrm{v}_{f,n}\\ \mathrm{v}_{f,o} - D_{nn}^{-1}\,D_{no}\,\mathrm{v}_{f,n}\end{bmatrix}\,, \tag{218}$$

we can thus state the *single-contact polar-angle dual problem*

$$\begin{aligned}\min_{\phi}\quad & \tfrac{1}{2}\,\boldsymbol{\lambda}_T(\phi)^T\,\hat{\mathbf{D}}\,\boldsymbol{\lambda}_T(\phi) + \boldsymbol{\lambda}_T(\phi)^T\,\hat{\mathbf{v}}\\ \text{s.t.}\quad & \phi \in \mathcal{I}_F\end{aligned}\,, \tag{219}$$

where now the only constraint is that the solution must lie in the interval $\mathcal{I}_F$. The optimality condition on the gradient of the objective function of (219) is therefore

$$\mathbf{t}(\phi)^T\left(\hat{\mathbf{D}}\,\boldsymbol{\lambda}_T(\phi) + \hat{\mathbf{v}}\right) = 0 \tag{220a}$$

$$\mathbf{t}(\phi) := \begin{bmatrix}\delta(\phi)\,\cos\phi - \bar{r}(\phi)\,\sin\phi\\ \delta(\phi)\,\sin\phi + \bar{r}(\phi)\,\cos\phi\end{bmatrix} \tag{220b}$$

$$\delta(\phi) := \frac{\bar{r}(\phi)}{\mathrm{v}_{f,n}}\sqrt{D_{nt}^2 + D_{no}^2}\,\sin\left(\phi - \tan^{-1}\left(\frac{D_{no}}{D_{nt}}\right)\right) \tag{220c}$$

At this point, plugging-in (210) to (220) and multiplying by $\frac{\mu^2\,\mathrm{v}_{f,n}^2}{\bar{r}(\phi)^3}$, results in the *polar-angle trigonometric equation*

$$c_0 + c_1\,\cos(\phi + \phi_1) + c_2\,\cos(2\,\phi + \phi_2) = 0 \tag{221}$$

The second-last step involves transforming the trigonometric equation to a complex polynomial. This requires defining the complex *phasor* $x = e^{i\,\phi}$ and applying it to (221). The result is the *complex quartic polynomial equation*

$$c_2\,\beta^*\,x^4 + c_1\,\alpha^*\,x^3 + 2\,c_0\,x^2 + c_1\,\alpha\,x + c_2\,\beta = 0\,, \tag{222}$$

where $\alpha = e^{i\,\phi_1}$ and $\alpha = e^{i\,\phi_2}$ are complex phasor constants, and α^* and β^* are their respective complex conjugates. We have therefore reduced the dual problem (208), for the case of

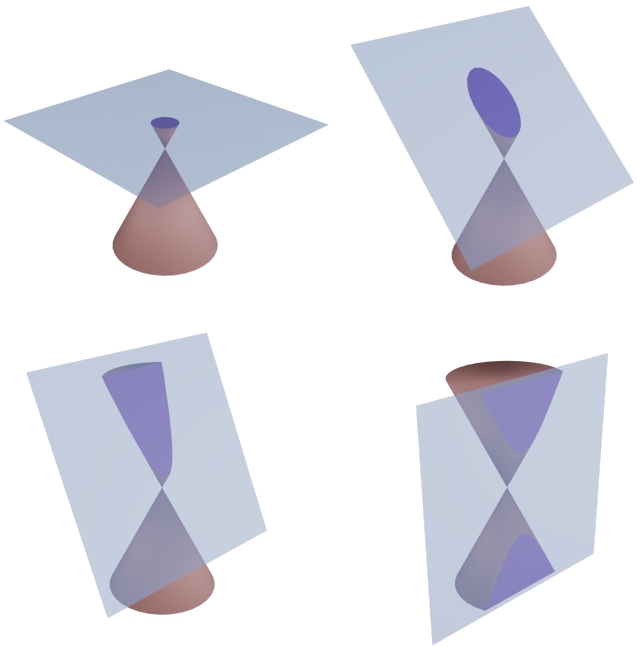

Fig. 37. Conic-section types: (top left) a circle when $r_{d/\mu} \to \infty$, i.e. the plane of maximum compression $\mathcal{P}_{mc}$ is exactly parallel to the contact normal $\mathbf{n}$, resulting in a circle. (top right) an ellipse when $r_{d/\mu} > 1$, i.e. $\mathbf{d}_n$ always points in directions that do not intersect the cone boundary $\partial\mathcal{K}_\mu$. (bottom left) a parabola when $r_{d/\mu} = 1$, i.e. $\mathbf{d}_n$ is parallel to $\partial\mathcal{K}_\mu$. (bottom right) a hyperbola when $r_{d/\mu} < 1$, i.e. when $\mathcal{P}_{mc}$ intersects $\partial\mathcal{K}_\mu$ in both half-spaces.

a sliding contact, to a polynomial equation (221). Solving[19] it, however, often yields multiple solutions due its quartic order, but only one of these can be maximally dissipative. To determine the one we seek, we can trivially exclude those that do not lie in the interval $\mathcal{I}_F$, as well as those that do not satisfy (220) and (221). From those that remain, we select the one that minimizes the objective function of (219). This is necessarily unique because we know that the original problem in Cartesian coordinates is strictly convex. The maximally dissipative contact reaction λ^* is computed using (210) for the optimal polar-coordinate values ϕ^* and $r^* = \bar{r}(\phi^*)$.

There are a few corner cases to mention though. In the geometric interpretation, the conic section exhibits degenerate cases that occur when $\mathrm{v}_{f,n} = 0$, i.e. $\mathcal{P}_{mc}$ intersects the origin. To examine these cases, we'll utilize a quantity which tells us how $\mathcal{P}_{mc}$ is oriented w.r.t $\mathcal{K}_\mu$. Using the definition of $\Delta\phi$ from (213b), we can define the *conic-section inclination ratio*

$$r_{d/\mu} = (\cos\Delta\phi)^2 = \frac{D_{nn}^2}{\mu^2\left(D_{nt}^2 + D_{no}^2\right)} \in \mathbb{R}_+ \ . \tag{223}$$

Thus, we can use the values of $r_{d/\mu}$ to identify the four types of conic-sections that $\mathcal{P}_{mc} \cap \partial\mathcal{K}_\mu$ can form. These are depicted in Fig.37, and are defined precisely by the cases:

1) $r_{d/\mu} \to \infty$: *circular*, when $\mathbf{D}$ is diagonal and/or $\mu = 0$.
2) $r_{d/\mu} > 1$: *elliptic*, $\mathcal{P}_{mc}$ intersects $\partial\mathcal{K}_\mu$ in one half-space
3) $r_{d/\mu} = 1$: *parabolic*, $\mathcal{P}_{mc}$ is parallel to $\partial\mathcal{K}_\mu$
4) $r_{d/\mu} < 1$: *hyperbolic*, $\mathcal{P}_{mc}$ intersects $\partial\mathcal{K}_\mu$ in both half-spaces

[19]In our implementation we have used `Eigen::PolynomialSolver`.

When $\mathrm{v}_{f,n} = 0 \wedge r_{d/\mu} > 1$, $\mathcal{P}_{mc}$ intersects $\partial\mathcal{K}_\mu$ uniquely at the origin, and the solution is trivially $\lambda^* = 0$. This is the case of a degenerate elliptic conic-section. Conversely, when $\mathrm{v}_{f,n} = 0 \wedge r_{d/\mu} < 1$, the problem admits a closed-form solution that is conditioned on whether the unconstrained solution lies within the friction cone:

$$\phi^* = \begin{cases} \phi_0 - \Delta\phi & , \lambda_0 \in \mathcal{K}_\mu \\ \phi_0 + \Delta\phi & , \textit{otherwise} \end{cases} \ , \tag{224}$$

$r^* = \max(0, \bar{r}(\phi^*))$, and λ^* is computed from (210). This is the case of a degenerate hyperbolic conic-section. For the special case of degenerate parabolic conic-section where $\mathrm{v}_{f,n} = 0 \wedge r_{d/\mu} = 1$, (224) still applies, but with $\Delta\phi = \pi$, because $\mathbf{d}_n$ is exactly perpendicular to the surface of $\mathcal{K}_\mu$. The final degenerate case is when $\mu = 0$ (i.e. frictionless contact), where $r_{d/\mu} \to \infty$ and the solution is trivially

$$\lambda^*_{\mu=0} = \begin{bmatrix} 0 \\ 0 \\ -D_{nn}^{-1}\,\mathrm{v}_{f,n} \end{bmatrix} \tag{225}$$

In summary, we have seen that $\mathbf{d}_n$, $\mathrm{v}_{f,n}$ and μ uniquely determine the geometry of the single-contact problem, i.e. $\mathcal{P}_{mc} \cap \mathcal{K}_\mu$. Moreover, only one particular combination requires solving the quartic polynomial (222), while all other cases can be handled analytically.

B. The Bisection-Search Method

This derivation is based on that presented by Hwangbo et al in [40] that describes the formulation used by RaiSim. Instead of solving the quartic polynomial (222) to handle the slipping case, Hwangbo et al solved (219) by employing the well established *bisection-method* to search for a zero-crossing that corresponds to the optimal polar-angle. In doing so, we can avoid performing most of the expensive trigonometric operations that semi-analytical approach requires. This alternate method significantly accelerates the operation that yields the maximally dissipative projection of a given contact reaction. This acceleration also comes without cost of accuracy, as the results are identical to former up to reasonable numerical tolerances, and thus serves as its drop-in replacement.

Recalling elements of the derivation of Appendix. C-A, the contact reaction necessarily lies on the plane of maximum compression $\mathcal{P}_{mc}$ defined by the quantities

$$\mathbf{d}_n := e_n^T\,\mathbf{D} = \begin{bmatrix} D_{tn} & D_{on} & D_{nn} \end{bmatrix}^T \ , \ \mathrm{v}_{f,n} = e_n^T\,\mathbf{v}_f \ ,$$

where $\mathbf{d}_n$ is colinear to plane normal $\mathbf{n}$ and $\boldsymbol{\lambda}_{T=0} = [0\,0\,D_{nn}^{-1}\,\mathrm{v}_{f,n}]^T$ is the point at which it intersects the axis $e_n = [0\,0\,1]^T$. When the contact is sliding, we seek solutions on the set $\mathcal{P}_{mc} \cap \partial\mathcal{K}_\mu$ that represents the conic-section formed by $\mathcal{P}_{mc}$ as it intersects the boundary of the friction cone $\partial\mathcal{K}_\mu$. Exploiting the geometry of conic-sections, we parameterize the contact reaction using polar coordinates in the form of

$$\lambda_t = r\,\cos\phi \ , \quad \lambda_o = r\,\sin\phi$$

$$\lambda_n = -D_{nn}^{-1}\left(\mathrm{v}_{f,n} + D_{nt}\,r\,\cos\phi + D_{no}\,r\,\sin\phi\right) \ ,$$

$$\boldsymbol{\lambda}_p(r,\phi) = \begin{bmatrix} \lambda_t & \lambda_o & \lambda_n \end{bmatrix}^T \ ,$$

The conic-section induces a coupling between the polar coordinates such that the radius becomes a function of the angle, i.e.

$$r(\phi) := \frac{-\mu\, \mathrm{v}_{f,n}}{D_{nn} + \mu\, D_{nt}\, \cos\phi + \mu\, D_{no}\, \sin\phi} \ .$$

Depending on how $\mathcal{P}_{mc} \cap \partial\mathcal{K}_\mu$ is formed, the set of admissible values for the polar angle ϕ, is defined as the feasible range

$$\mathcal{I}_F = \begin{cases} (-\Delta\phi, \Delta\phi] + \phi_0, & , D_{nn} \leq \mu\sqrt{D_{nt}^2 + D_{no}^2} \\ [0, 2\pi) & , otherwise \end{cases} ,$$

where the central angle ϕ_0 and half-range $\Delta\phi$ are defined in (213b). In the case of a sliding contact ($m =$ Slip), the single-contact dual problem (219) can be formulated solely in terms of the polar-angle as

$$\begin{aligned} \min_{\phi} \quad & \tfrac{1}{2}\, \boldsymbol{\lambda}_T(\phi)^T\, \hat{\mathbf{D}}\, \boldsymbol{\lambda}_T(\phi) + \boldsymbol{\lambda}_T(\phi)^T\, \hat{\mathbf{v}} \\ \text{s.t.} \quad & \phi \in \mathcal{I}_F \end{aligned} ,$$

where $\boldsymbol{\lambda}_T(\phi) \in \mathbb{R}^2$ is the polar-angle-dependent tangential contact reaction, and $\hat{\mathbf{D}} \in \mathbb{R}^{2\times 2}$, $\hat{\mathbf{v}}_f \in \mathbb{R}^2$, are respectively, the Delassus matrix and a free velocity projected onto $\mathcal{P}_{mc}$.

The conic-section, and thus the range of admissible angles, depends solely on the normal projection of the Delassus matrix and the friction cone aperture. Second, where along the conic-section, solutions actually form is also dependent on the free velocity, as it determines the unconstrained solution $\boldsymbol{\lambda}_0$. As sliding friction necessarily opposes motion, we can expect that the maximally dissipative contact reaction to lie on a section of the curve that shares some properties with $\boldsymbol{\lambda}_0$. Thus, the motivation behind this approach lies in three insightful observations regarding how and where solutions form:

1) the single-contact polar-angle dual problem (219) is strictly convex with a unique global minimum
2) the feasible region $\mathcal{I}_F$ is a bounded interval
3) when multiple solutions (i.e. zeros) of the stationarity condition (220a) exist, only the unique global minimum may lie in region of the conic-section curve that is within line-of-sight from the unconstrained solution (see p.31 of [72] for a formal proof of this property)

From these, Hwangbo et al extrapolated the following ideas. Firstly, as the worst-case search interval $\mathcal{I}_F$ is finite, a direct search for zeros of (220a) is possible. Secondly, the line-of-site region presents a reasonable heuristic to both begin the search from as well as to narrow the effective search interval. Moreover, the Euclidean (i.e. minimum-distance) projection of the unconstrained solution provides an automatic mechanism to define a starting point for the search. Lastly, as floating-point trigonometric computations are expensive, an iterative method that only requires simple algebraic expressions can potentially reduce the time complexity of the search.

Next we describe how the bisection-method can be used to solve this polar-angle dual problem without solving the quartic polynomial (222). The objective function of the dual problem (208), and its gradient are

$$h_0(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T\, \mathbf{D}\, \boldsymbol{\lambda} + \boldsymbol{\lambda}^T\, \mathbf{v}_f \tag{227a}$$

$$\nabla_{\boldsymbol{\lambda}} h_0(\boldsymbol{\lambda}) = \mathbf{D}\, \boldsymbol{\lambda} + \mathbf{v}_f = \bar{\mathbf{D}}\, \boldsymbol{\lambda}_T + \bar{\mathbf{v}}_f \ , \tag{227b}$$

where we've used 3D extensions of the tangentially-projected Delassus matrix and free velocity vector in the form of

$$\bar{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}} \\ 0 \end{bmatrix} \in \mathbb{R}^{3\times 2} \quad , \quad \bar{\mathbf{v}}_f = \begin{bmatrix} \hat{\mathbf{v}}_f \\ 0 \end{bmatrix} \in \mathbb{R}^3 \ . \tag{228}$$

Next, the constraint imposed by $\mathcal{P}_{mc}$ and its gradient are

$$h_1(\boldsymbol{\lambda}) = \mathbf{d}_n^T\, \boldsymbol{\lambda} + \mathrm{v}_{f,n} = 0 \tag{229a}$$

$$\nabla_{\boldsymbol{\lambda}} h_1(\boldsymbol{\lambda}) = \mathbf{d}_n \ , \tag{229b}$$

and equivalently, those of the friction cone $\mathcal{K}_\mu$ are

$$h_2(\boldsymbol{\lambda}) = \lambda_t^2 + \lambda_o^2 - \mu^2\, \lambda_n^2 = 0 \tag{230a}$$

$$\nabla_{\boldsymbol{\lambda}} h_2(\boldsymbol{\lambda}) = \begin{bmatrix} 2\,\lambda_t & 2\,\lambda_o & -2\,\mu^2\,\lambda_n \end{bmatrix}^T \ . \tag{230b}$$

Geometrically speaking, we know that $\nabla_{\boldsymbol{\lambda}} h_1(\boldsymbol{\lambda})$ and $\nabla_{\boldsymbol{\lambda}} h_2(\boldsymbol{\lambda})$ are always perpendicular to their respective constraint surfaces. We can exploit this to define a vector that is always tangent to the conic-section curve by forming an orthogonal projection to the aforementioned constraint vectors. This results in the *tangent vector*

$$t(\boldsymbol{\lambda}) := \nabla_{\boldsymbol{\lambda}} h_1(\boldsymbol{\lambda}) \times \nabla_{\boldsymbol{\lambda}} h_2(\boldsymbol{\lambda}) \ . \tag{231}$$

Furthermore, the KKT conditions require that the gradient of the objective function $\nabla_{\boldsymbol{\lambda}} h_0(\boldsymbol{\lambda})$ lie within the space spanned by the constraint gradients $\nabla_{\boldsymbol{\lambda}} h_1(\boldsymbol{\lambda})$ and $\nabla_{\boldsymbol{\lambda}} h_1(\boldsymbol{\lambda})$. Thus, the stationarity condition required on the Lagrangian of the dual problem can be expressed as

$$\nabla_{\boldsymbol{\lambda}} h_0(\boldsymbol{\lambda}^*)^T\, t(\boldsymbol{\lambda}^*) = 0 \ , \tag{232}$$

which is exactly (220a) from Sec.C-A. We denote the function that evaluates this gradient product as

$$g(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} h_0(\boldsymbol{\lambda})^T\, t(\boldsymbol{\lambda}) \ . \tag{233}$$

At this point we can introduce the bisection-method and explain how it's applicable. It operates by taking some initial interval $[\phi_{\min}, \phi_{\max}]$, and progressively dividing it in half while checking for changes in the sign of the function. The sign determines from which end of the interval the half-interval will be taken, i.e. start or end, and in this way, it very quickly converges to any detected zero-crossing. In our case, the function it operates on is the gradient product defined in (233). Thus, we require two more operations to realize this bisection-based search. First is the function that yields the sign of the gradient product, which we'll denoted as

$$s_g(\boldsymbol{\lambda}) = \mathrm{sign}(\, g(\boldsymbol{\lambda})\,) \ . \tag{234}$$

Second, we require an operation that can provide a starting point for the search. We can use the radial projector

$$\phi_E = \tan^{-1} \frac{\lambda_{0,o}}{\lambda_{0,t}} \ , r_E = \bar{r}(\phi_E) \tag{235a}$$

$$\mathcal{P}_{\mathrm{E}}^{\mathcal{S}\cap\mathcal{K}}(\boldsymbol{\lambda}_0) := \boldsymbol{\lambda}_p(\phi_E, \bar{r}(\phi_E)) \ . \tag{235b}$$

where $\mathcal{P}_{\mathrm{E}}^{\mathcal{S}\cap\mathcal{K}} : \mathbb{R}^3 \to \mathbb{R}^3$[20].

[20] $\mathcal{P}_{\mathrm{E}}^{\mathcal{S}\cap\mathcal{K}}(\boldsymbol{\lambda}_0)$ is a legitimate contact-consistent projector in its own right. However, in contrast to $\mathcal{P}_{\mathrm{MDP}}^{\mathcal{S}\cap\mathcal{K}}(\boldsymbol{\lambda}_0)$ and $P_{\mathrm{BS}}^{\mathcal{S}\cap\mathcal{K}}(\boldsymbol{\lambda}_0)$, it does not yield maximally dissipative reactions.

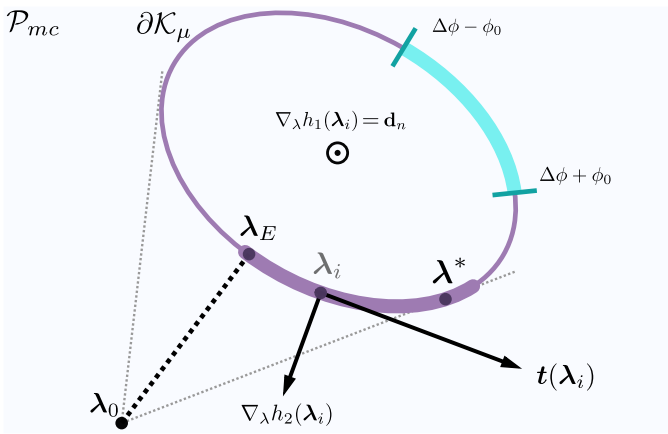

Fig. 38. The bisection-method applied to a sample conic-section (purple) defined by the intersection of the plane of maximum compression $\mathcal{P}_{mc}$ and the boundary of the friction cone $\partial\mathcal{K}_\mu$, depicted using $\mathcal{P}_{mc}$ as a cross-section via the plane normal $\mathbf{d}_n$. The unconstrained solution $\boldsymbol{\lambda}_0$ is first projected onto the conic-section as the Euclidean projection $\boldsymbol{\lambda}_E$. From the perspective of $\boldsymbol{\lambda}_0$, the line-of-sight region (gray dotted lines) extends onto each end of the contour of the curve, and the side opposite the gradient is taken as the zero-crossing search interval (think purple curve segment). At each step of the interval-search phase, the iterate $\boldsymbol{\lambda}_i$ is used to compute the tangent vector $t(\boldsymbol{\lambda}_i)$ w.r.t the local contour via the orthogonality of the constraint gradients $\nabla_{\boldsymbol{\lambda}} h_1(\boldsymbol{\lambda}_i)$ and $\nabla_{\boldsymbol{\lambda}} h_2(\boldsymbol{\lambda}_i)$. Thus, the search interval is progressively extended towards the clock-wise direction until the end reaching the end of the line-of-sight region, and while avoiding the infeasible region $[0, 2\pi) \setminus \mathcal{I}_F$ (cyan).

We now have all the elements necessary to construct the bisection-method algorithm to realize $P_{\text{BS}}^{\mathcal{S}\cap\mathcal{K}}(\boldsymbol{\lambda}_0)$. The full algorithm is detailed in Alg.7, and a visual depiction using a mock example is provided in Fig.38. However, it's worth taking a moment to outline the key steps involved and provide some context. These are:

1) compute $\boldsymbol{\lambda}_E$ using the Euclidean projection of $\boldsymbol{\lambda}_0$
2) compute the line-of-sight region $\mathcal{I}_{LoS}$
3) compute the feasible region $\mathcal{I}_F$ using μ and $\mathbf{d}_n$
4) determine the smallest feasible interval $\mathcal{I}_{BS} \subseteq \mathcal{I}_F \cap \mathcal{I}_{LoS}$
5) perform bisection on the interval $\mathcal{I}_{BS}$
6) if no solution was found, perform bisection on $\mathcal{I}_F$

All but step (4) are straight-forward operations on the problem constants. The most delicate, and most important, is step (4) that seeks the smallest feasible interval to perform the bisection-search on. In a nutshell, this procedure proceeds as follows: starting from the angle ϕ_E corresponding to the Euclidean projection of the unconstrained solution $\boldsymbol{\lambda}_0$, we take make a series of exponentially increasing steps opposite of the direction (i.e. sign) of the projected gradient at ϕ_E, and re-compute this gradient sign at each step until it inverts or until we've reached some margin beyond the line-of-sight region. The first case signifies that a zero-crossing has been detected, and the bisection search interval $[\phi_{\text{min}}, \phi_{\text{max}}]$ is collapsed to the previous and current angle of the interval search procedure.

However, there is a critical caveat to this procedure. As the interval necessarily needs to be a compact set for bisection to succeed, we must carefully handle the cases where the infeasible region $[0, 2\pi) \setminus \mathcal{I}_F$, overlaps with or is very close to, the line-of-sight interval $\mathcal{I}_{LoS}$. In fact, we found that the original algorithm described in [40] sometimes fails in such cases. To robustify it, we've incorporated a few modifications that ensure that: (a) we don't accidentally stop the interval expansion within or at the limit of the infeasible region, and (b) if this interval search fails, we use step (6) to perform a brute-force search of all zero-crossings and determine the optimal angle via exclusion similarly to how it was handled for the quartic polynomial solutions in SecC-A. The resulting operation, even in the worst case of resorting to step (6), drastically reduces the average time-complexity of the projection.

Algorithm 7 Bisection-Based Optimal Polar-Angle Search

Require: $\boldsymbol{\lambda}_0 \in \mathbb{R}^3,\ \mathbf{D} \in \mathbb{R}^{3\times 3},\ \mathbf{v}_f \in \mathbb{R}^3,\ \mu \in \mathbb{R}_+$
Require: $N,\ \beta_1,\ \beta_2,\ \beta_3,\ \epsilon_{BS}$

$\phi_E \leftarrow \text{atan2}(\lambda_{0,o}, \lambda_{0,t})$ ▷ Euclidean projection
$\boldsymbol{\lambda}_E \leftarrow \boldsymbol{\lambda}_p(\phi_E, r(\phi_E))$ ▷ using (210)
$g_0 \leftarrow g(\boldsymbol{\lambda}_E)$ ▷ using (233)
$s_0 \leftarrow s_g(\boldsymbol{\lambda}_E)$ ▷ using (234)

$\Delta\phi_{LoS} \leftarrow \text{acos}\left(\frac{\|\boldsymbol{\lambda}_{0,T}\|_2}{r_E}\right)$ ▷ line-of-sight range
$\mathcal{I}_{LoS} \leftarrow [-\Delta\phi_{LoS}\,,\ \Delta\phi_{LoS}] + \phi_E$

$\mathcal{I}_F \leftarrow$ from $\mathbf{d}_n$ and μ using (213) ▷ feasible range

$\phi_0 \leftarrow \phi_E$ ▷ initial angle
$\alpha \leftarrow -\min(|\Delta\phi_{LoS}|, \beta_1)\, s_0$ ▷ initial step-size
found $\leftarrow$ false

for $i = 1$ to N **do** ▷ find smallest feasible search interval
  $\phi_i \leftarrow \phi_{i-1} + \alpha$
  $r_i \leftarrow r(\phi_i)$
  $\boldsymbol{\lambda}_i \leftarrow \boldsymbol{\lambda}_p(\phi_i, r(\phi_i)))$
  $g_i \leftarrow g(\boldsymbol{\lambda}_i)$
  $s_i \leftarrow s(\boldsymbol{\lambda}_i)$
  $p_i \leftarrow |\phi_E - \phi_i| - \Delta\phi_{LoS}$

  if $r_i > 0 \wedge g_0 \cdot g_i < 0$ **then**
    $\phi_f \leftarrow \phi_i$
    found $\leftarrow$ true ▷ zero-crossing detected
    break
  end if

  if $p_i < 0 \vee r_i < 0$ **then**
    $\phi_i \leftarrow \phi_{i-1}$ ▷ reset interval
    $\alpha \leftarrow \alpha\,\beta_2$ ▷ expand step-size
  else
    $\alpha \leftarrow -\beta_3\, s_0$ ▷ step conservatively past line-of-sight
    $g_0 \leftarrow g_i$
    $s_0 \leftarrow s_i$
  end if
end for

if found **then**
  $\phi_{\min} \leftarrow \min(\phi_0\,,\ \phi_f)$
  $\phi_{\max} \leftarrow \max(\phi_0\,,\ \phi_f)$
  $\mathcal{I}_{BS} \leftarrow [\phi_{\min}\,,\ \phi_{\max}]$ ▷ bisection interval
  $\phi^* \leftarrow$ from Alg.8 using $\mathcal{I}_{BS}$ ▷ bisection operation
else
  $\Phi_{zc} \leftarrow$ zero-crossings in $\mathcal{I}_F$ using Alg.9 ▷ brute-force search
  $\phi^* \leftarrow \arg\min_{\phi\in\Phi_{zc}} h_0(\boldsymbol{\lambda}_p(\phi, r(\phi))))$ ▷ using (227a)
end if

$\boldsymbol{\lambda}^* \leftarrow \boldsymbol{\lambda}_p(\phi^*, r(\phi^*)))$

return $\boldsymbol{\lambda}^*$

Algorithm 8 Bisection-Method Search Operation

Require: $\phi_{\min} \in \mathbb{R},\ \phi_{\max} \in \mathbb{R},\ \epsilon_{BS} \in \mathbb{R}_+$

$g_0 \leftarrow g(\boldsymbol{\lambda}_p(\phi_{\min}, r(\phi_{\min}))))$ ▷ using (233)

repeat ▷ bisection procedure

  $\phi_{\text{mid}} \leftarrow \frac{1}{2}\,(\phi_{\min} + \phi_{\max})$ ▷ midpoint
  $r_{\text{mid}} \leftarrow r(\phi_{\text{mid}})$ ▷ using (215)
  $\boldsymbol{\lambda}_{\text{mid}} \leftarrow \boldsymbol{\lambda}_p(\phi_{\text{mid}}, r(\phi_{\text{mid}})))$ ▷ using (210)

  if $g(\boldsymbol{\lambda}_{\text{mid}}) \cdot g_0 < 0$ **then** ▷ check for zero-crossing
    $\phi_{\min} \leftarrow \phi_{\text{mid}}$ ▷ contract left
  else
    $\phi_{\max} \leftarrow \phi_{\text{mid}}$ ▷ contract right
  end if

  $\boldsymbol{\lambda}_{\min} \leftarrow \boldsymbol{\lambda}_p(\phi_{\min}, r(\phi_{\min})))$
  $\boldsymbol{\lambda}_{\max} \leftarrow \boldsymbol{\lambda}_p(\phi_{\max}, r(\phi_{\max})))$

until $\|\boldsymbol{\lambda}_{\min} - \boldsymbol{\lambda}_{\max}\|_2 < \epsilon_{BS}$ ▷ check for convergence

return ϕ_{mid}

Algorithm 9 Feasible Region Bisection-Search

Require: $\mathcal{I}_F,\ N \in \mathbb{N}_+$

$\Phi_{zc} \leftarrow \varnothing$ ▷ initialize set of zero-crossings
$\mathcal{I} \leftarrow \varnothing$ ▷ initialize set of intervals

$\phi_0 \leftarrow \min(\mathcal{I}_F)$ ▷ start
$\Delta\phi \leftarrow |\phi_f - \phi_0|$ ▷ range
$\alpha \leftarrow \frac{\Delta\phi}{N}$ ▷ step-size
$g_0 \leftarrow g(\boldsymbol{\lambda}_p(\phi_0, r(\phi_0)))$

for $i = 1$ to N **do** ▷ zero-crossing interval construction
  $\phi_i \leftarrow \phi_{i-1} + \alpha$
  $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_p(\phi_i, r(\phi_i))$
  $g_i \leftarrow g(\boldsymbol{\lambda}_i)$
  if $g_i \cdot g_{i-1} < 0$ **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{[\phi_i - \alpha\,,\ \phi_i]\}$ ▷ extend interval set
  end if
end for

for $i = 1$ to n_{zc} **do** ▷ interval-wise bisection
  $\phi_i \leftarrow$ from Alg.8 using $\mathcal{I}_i$ ▷ bisection operation
  $\Phi_{zc} \leftarrow \Phi_{zc} \cup \{\phi_i\}$ ▷ extend zero-crossing set
end for

return Φ_{zc}

APPENDIX D
ADDITIONAL EXPERIMENT FIGURES

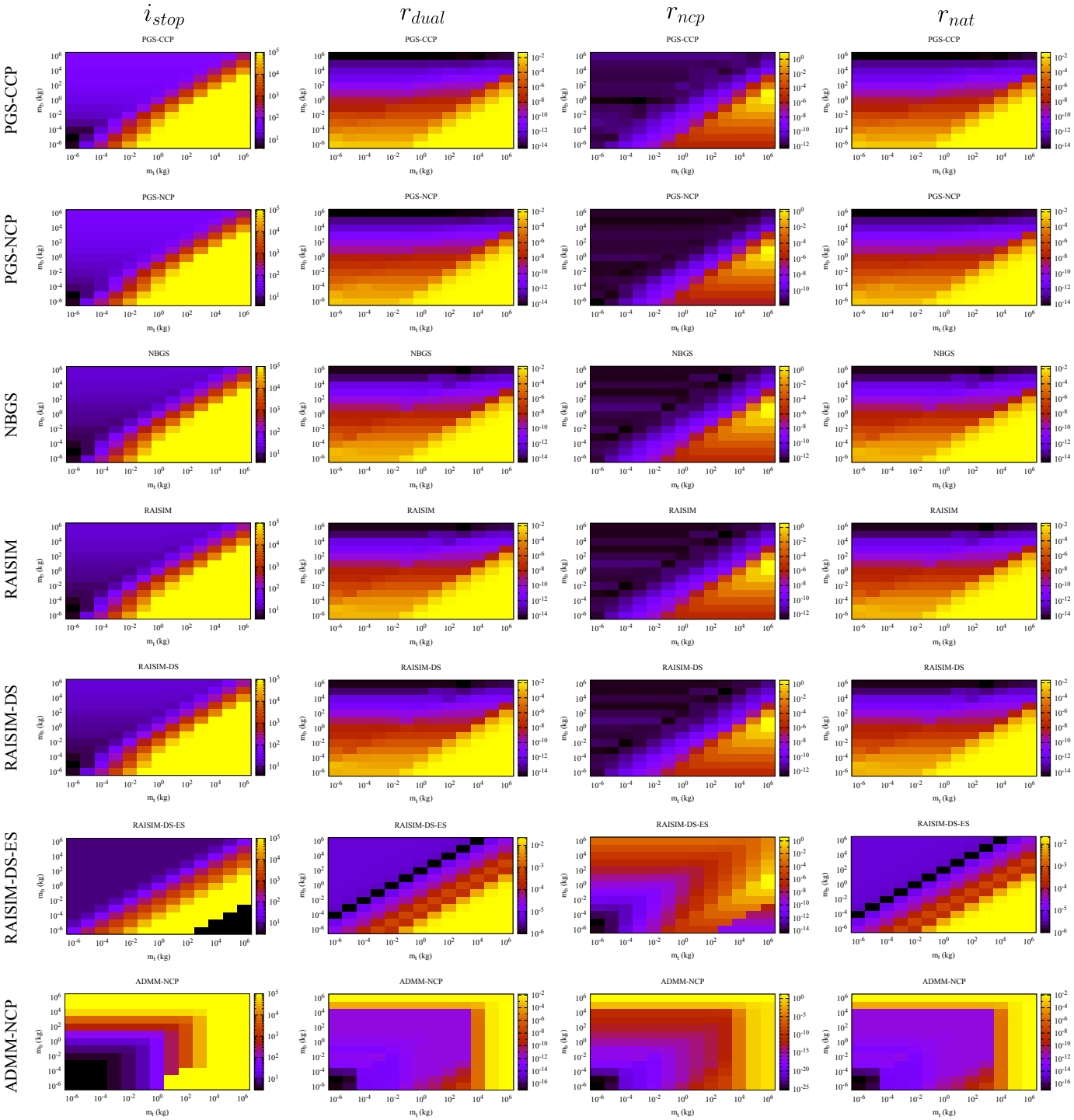

Fig. 39. *Boxes-Fixed*: A heatmap-like rendering of solver iterations i_{stop} and the complementarity residual r_{ncp} as functions of the bottom and top bodies with respective masses m_b and m_t. The data is generated while the bottom box is at rest on the plane.

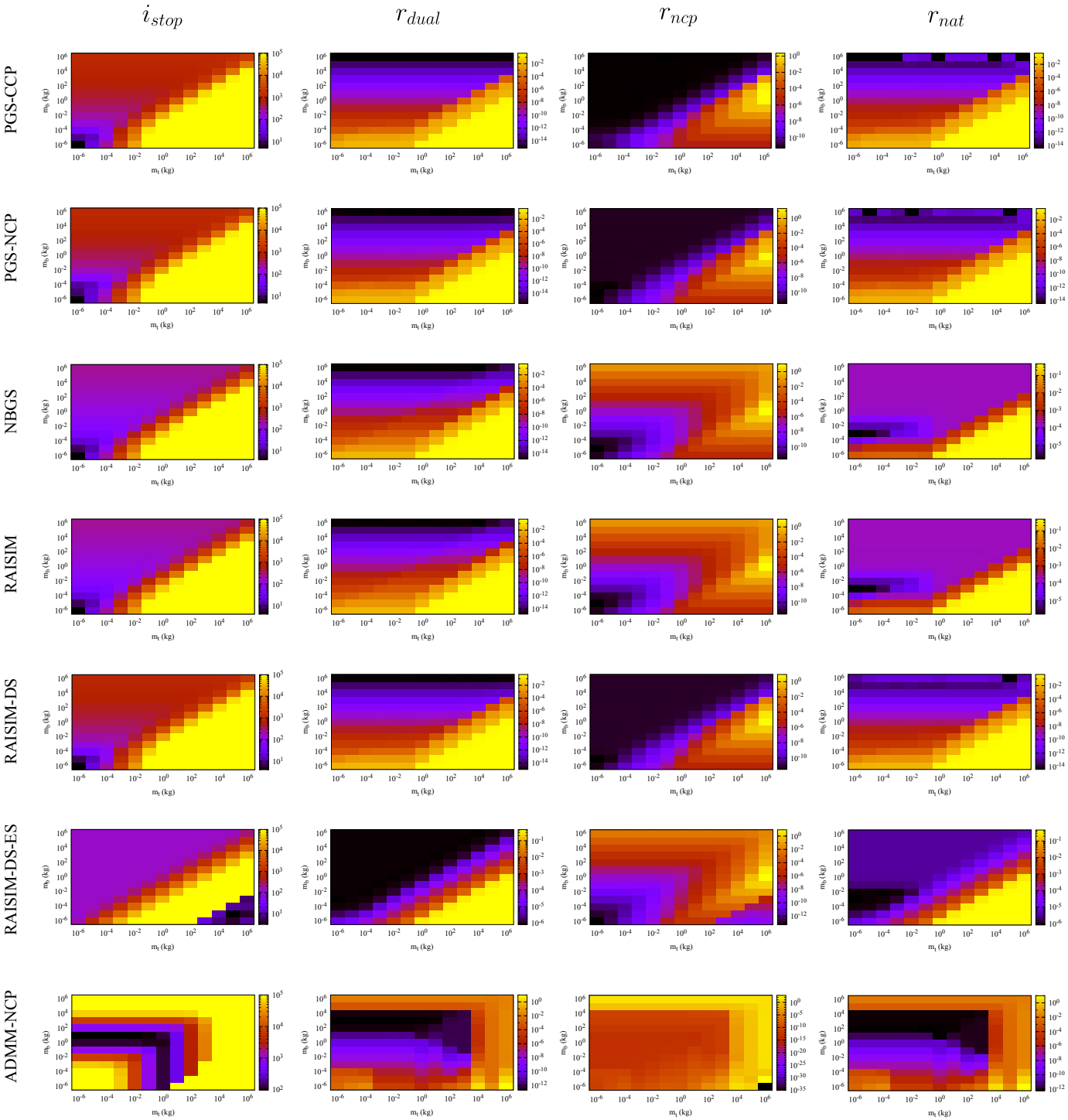

Fig. 40. *Boxes-Fixed*: A heatmap-like rendering of solver iterations i_{stop} and the complementarity residual r_{ncp} as functions of the bottom and top bodies with respective masses m_b and m_t. The data is generated while the bottom box is pushed by a force marginally exceeding the necessary stiction force.

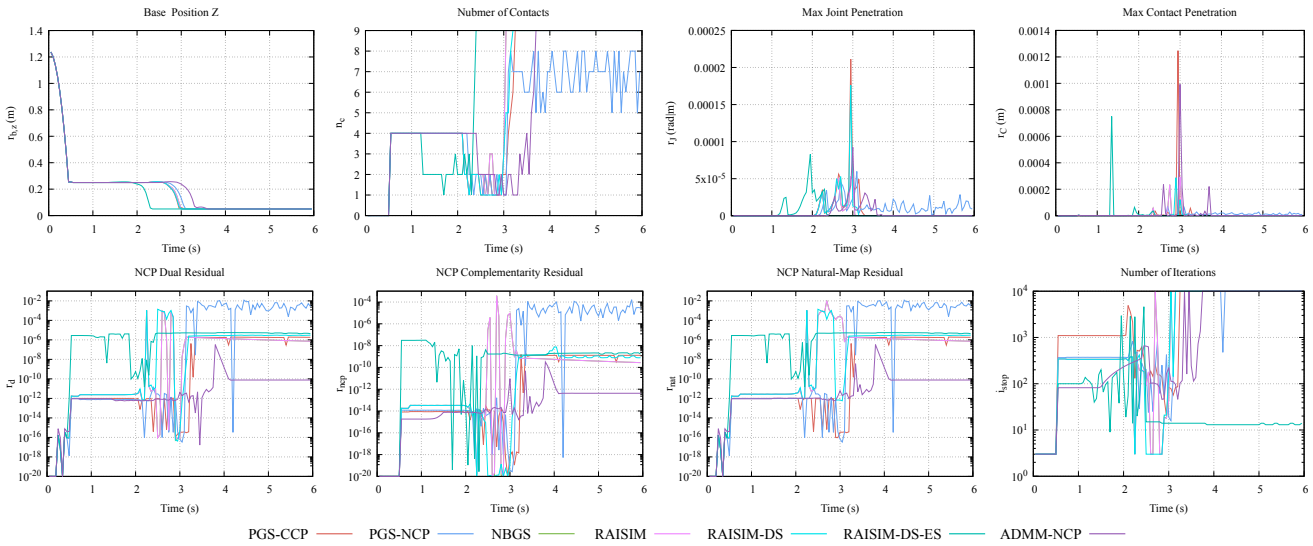

Fig. 41. *Nunchaku*: Summary of the second run that enables constraint stabilization.

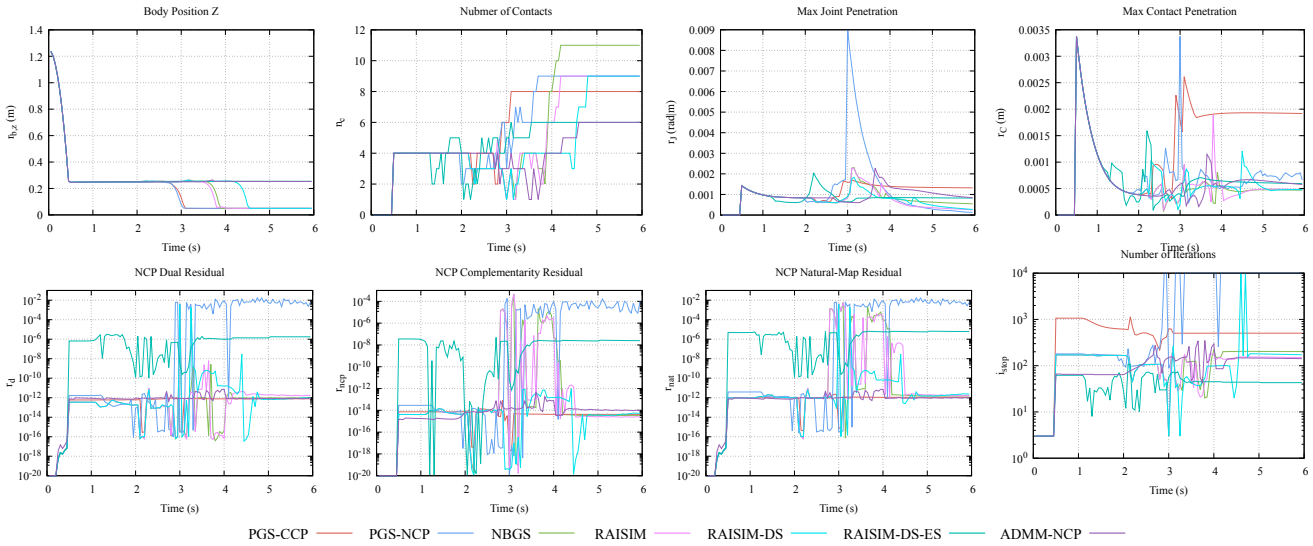

Fig. 42. *Nunchaku*: Summary of the third run that enables constraint softening.

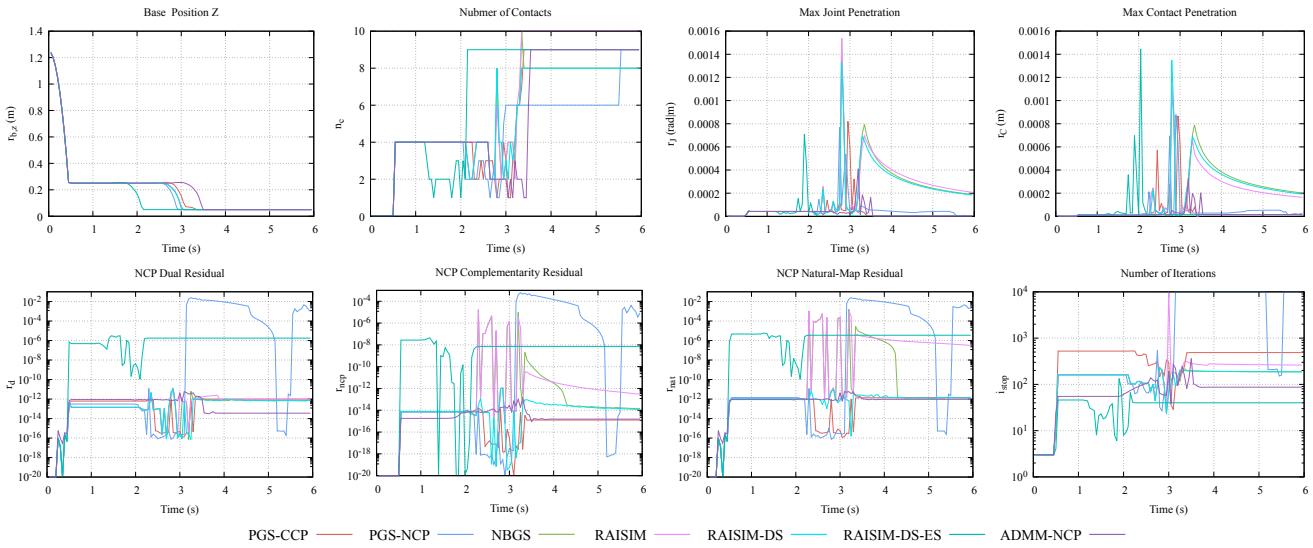

Fig. 43. *Nunchaku*: Summary of the fourth run that enables both constraint stabilization and softening.

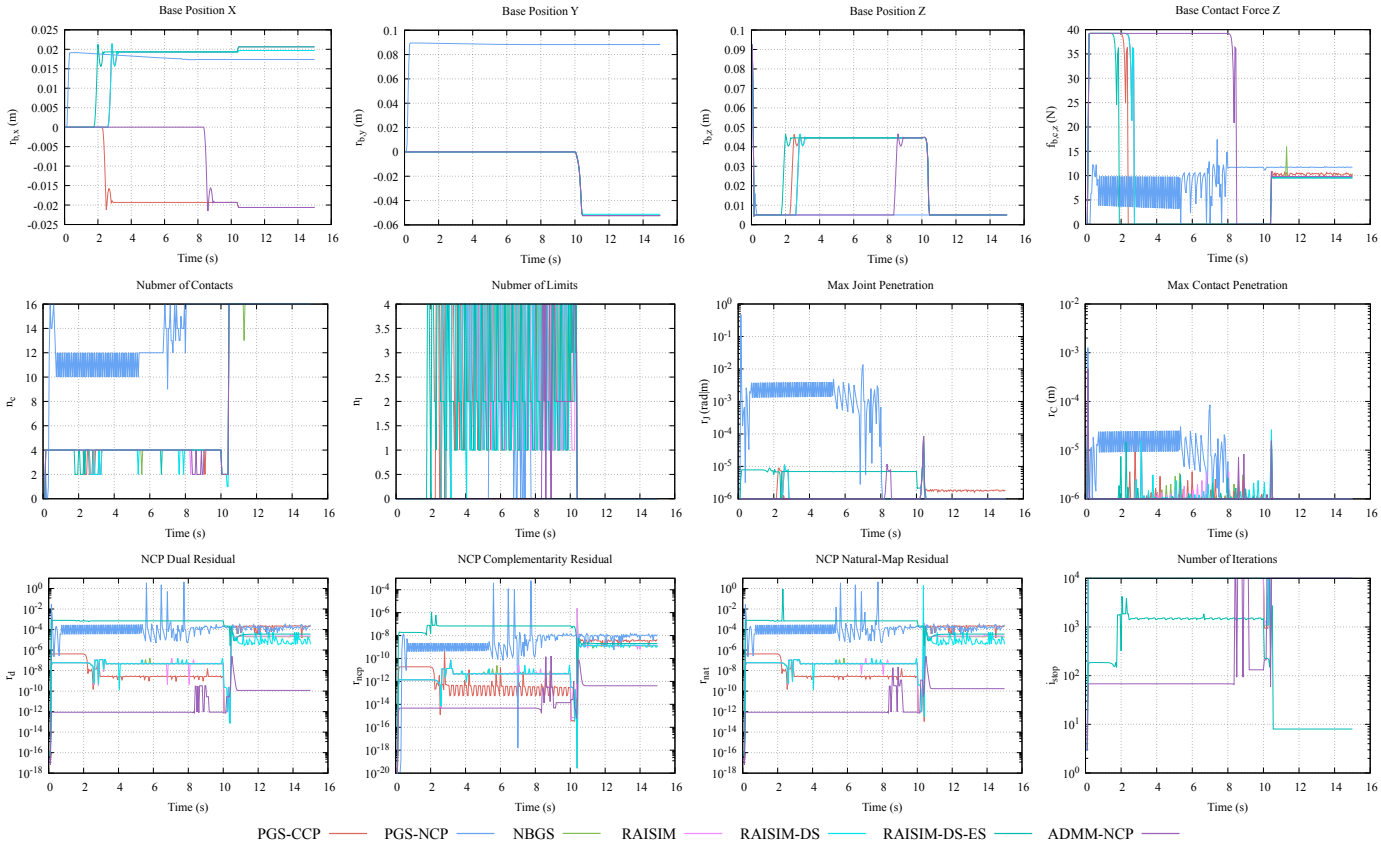

Fig. 44. *Fourbar*: Summary of the second run that enables constraint stabilization.

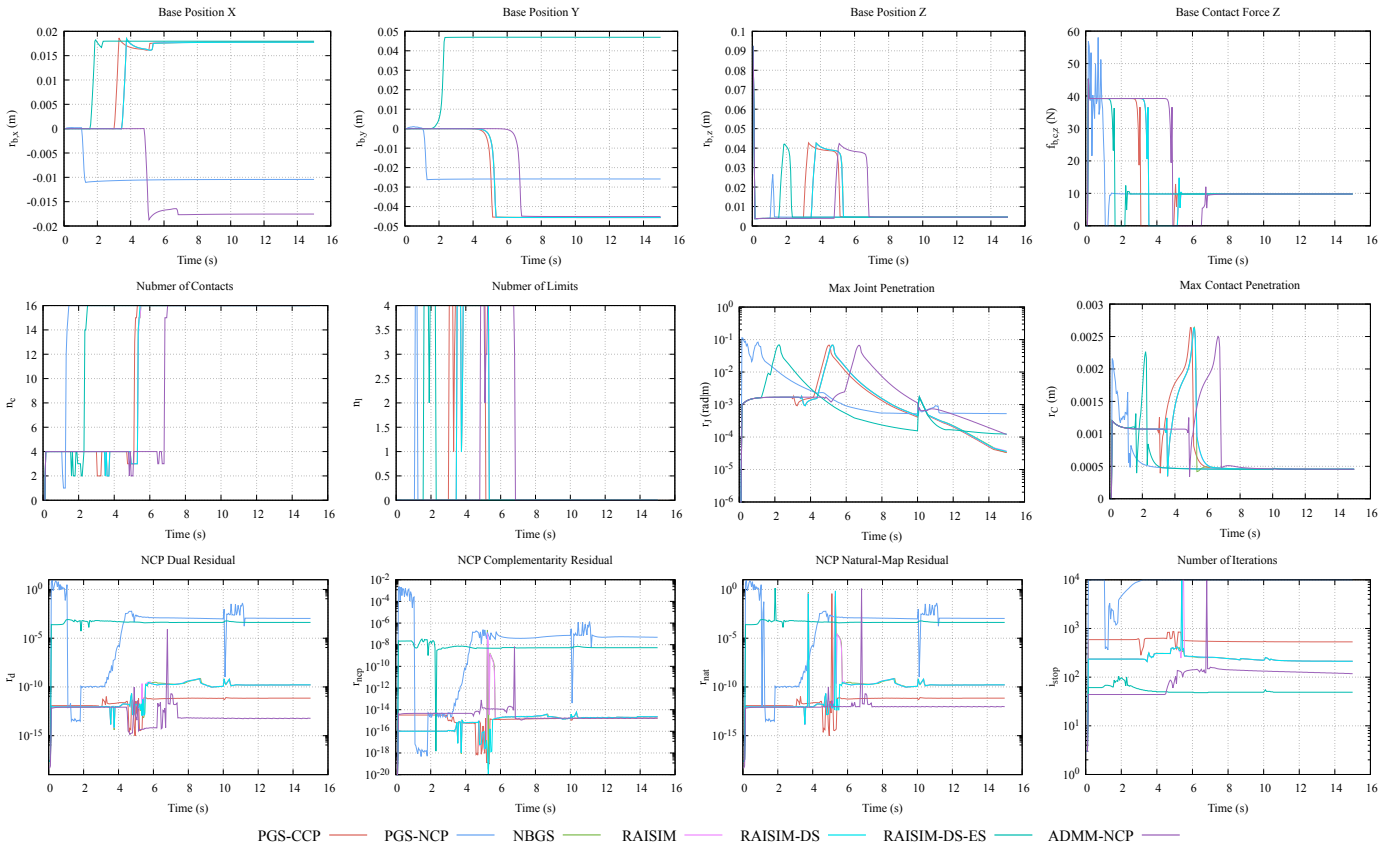

Fig. 45. *Fourbar*: Summary of the third run that enables constraint softening.

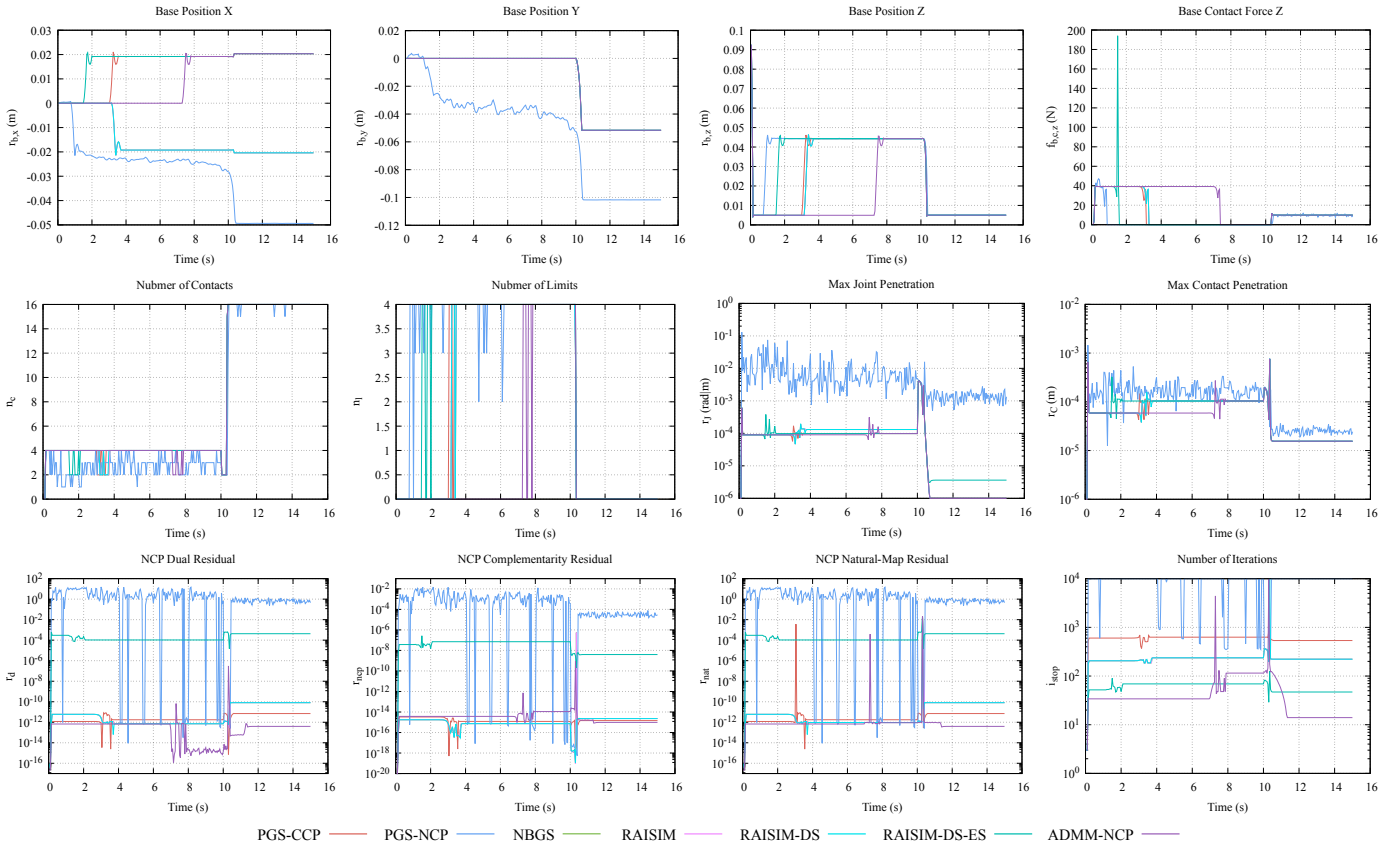

Fig. 46. *Fourbar*: Summary of the fourth run that enables both constraint stabilization and softening.

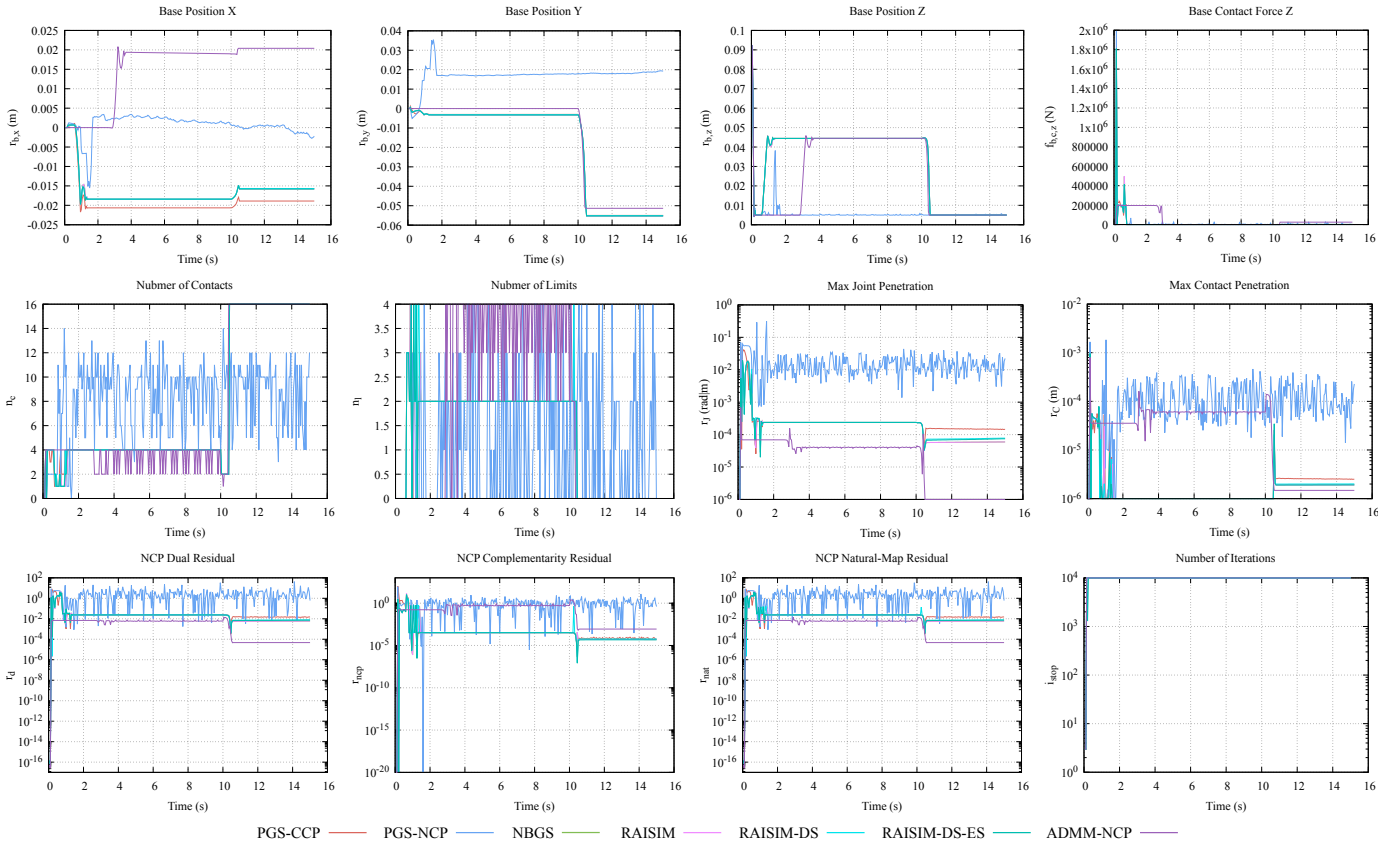

Fig. 47. *Fourbar*: Summary of solving the un-augmented problem in the presence of a large mass-ratio.

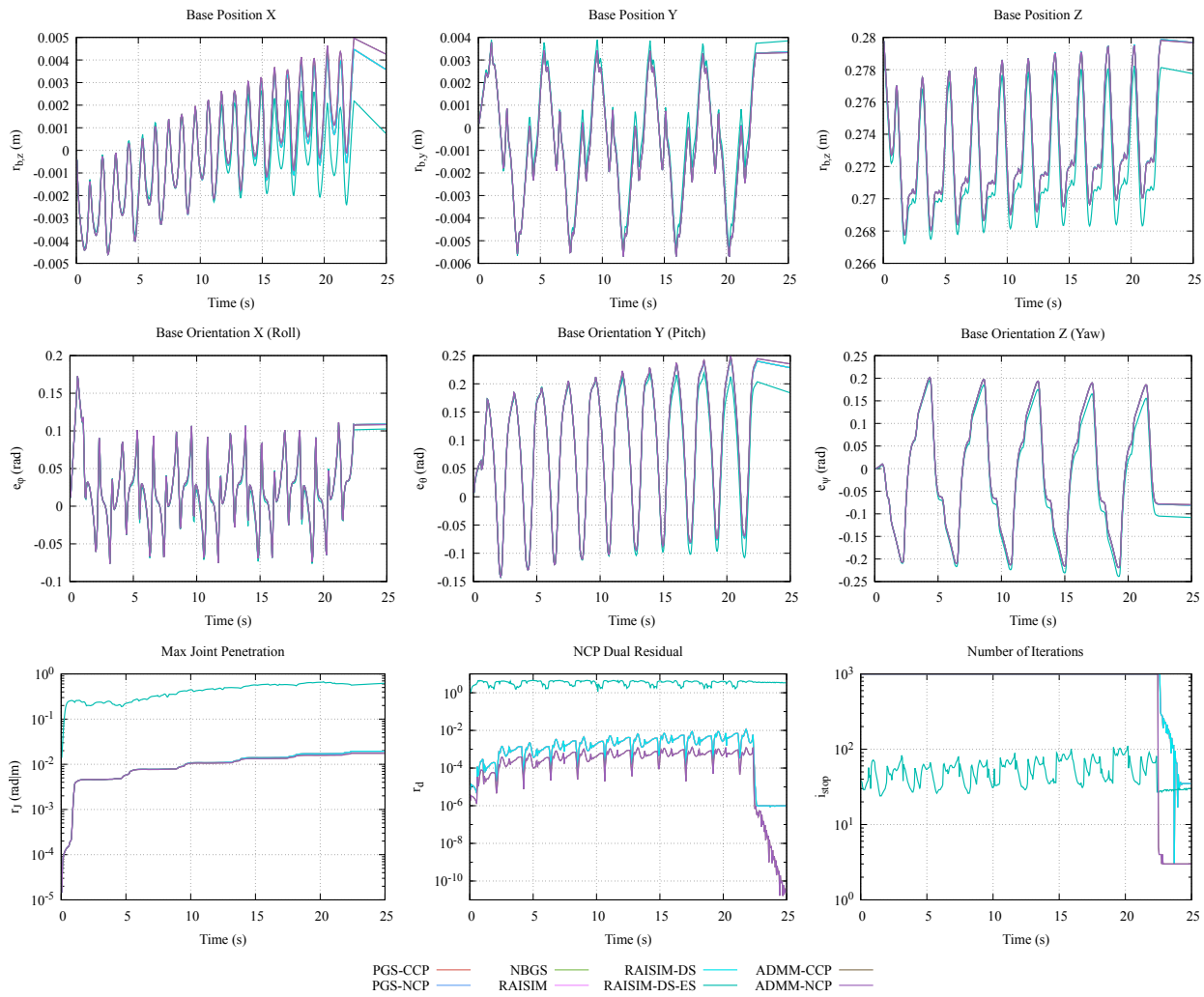

Fig. 48. *Walker*: Time-series plots of solving the un-augmented problem with environment collisions and gravity disabled.

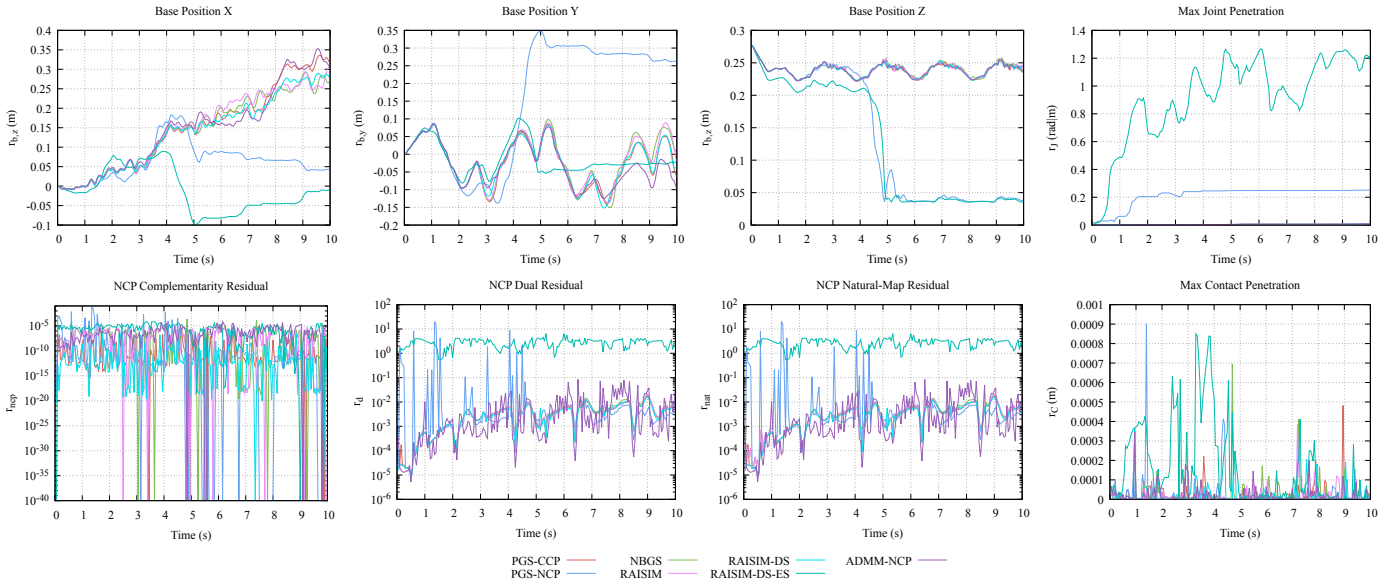

Fig. 49. *Walker*: Time-series plots of solving the un-augmented problem with external collisions and gravity enabled.

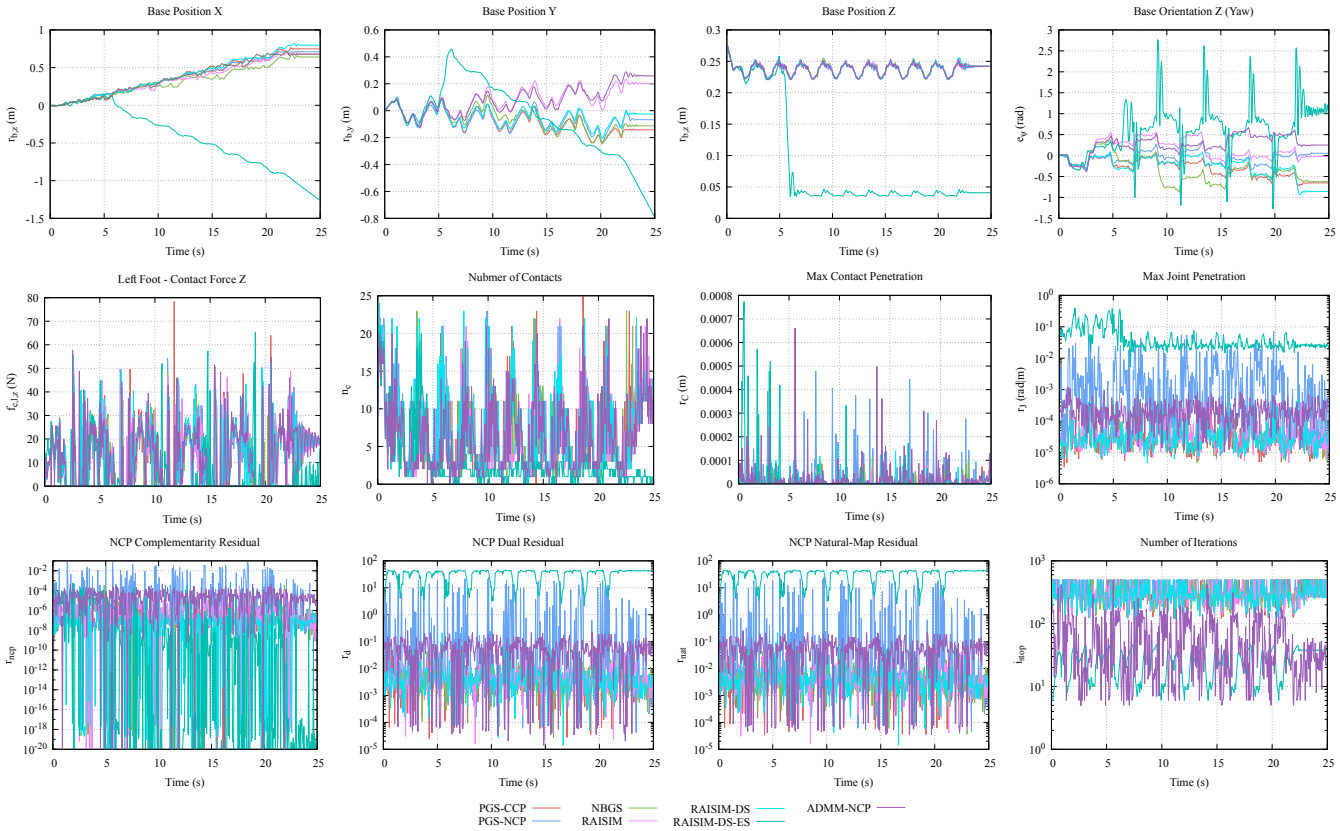

Fig. 50. *Walker*: Time-series plots of solving the stabilized problem using high-throughput solver settings.

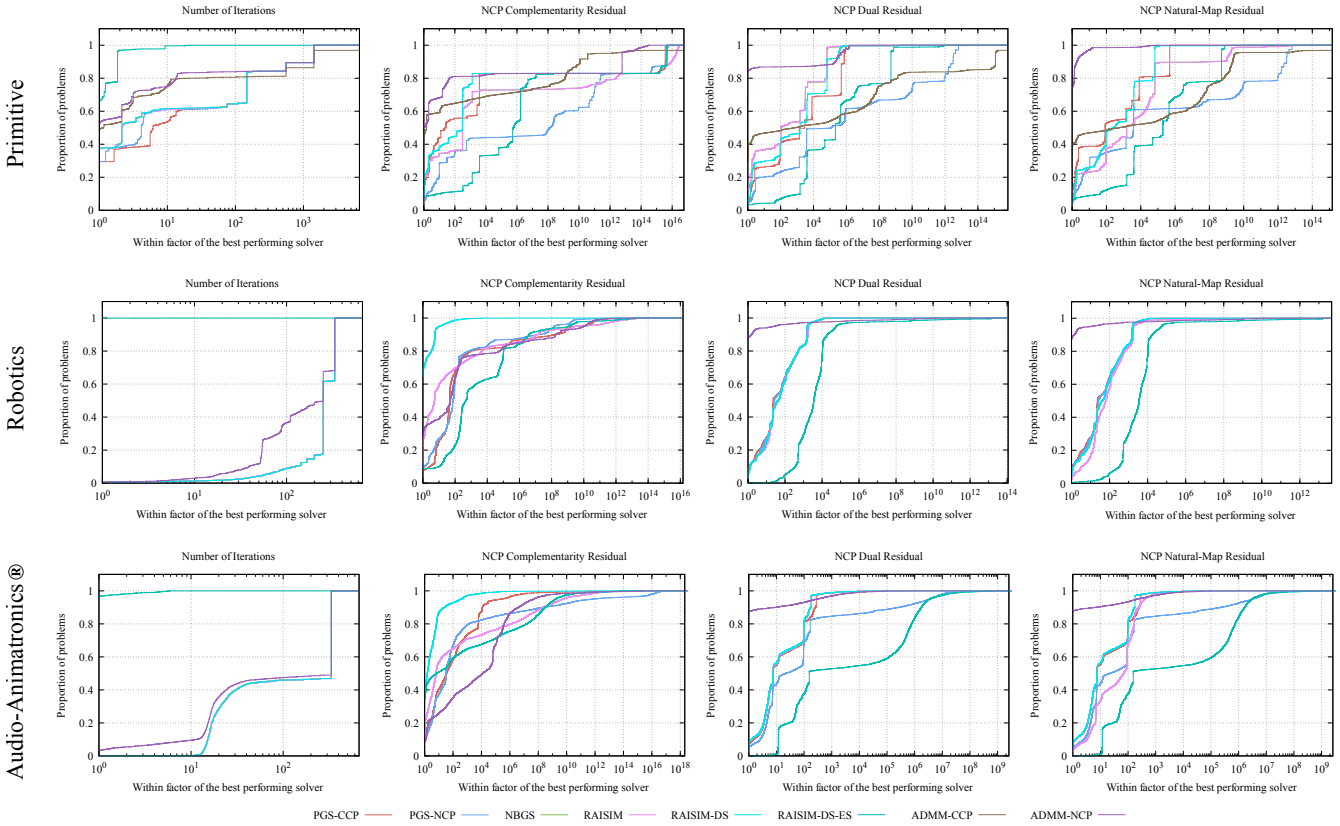

Fig. 51. Performance profiles w.r.t each benchmark system category.

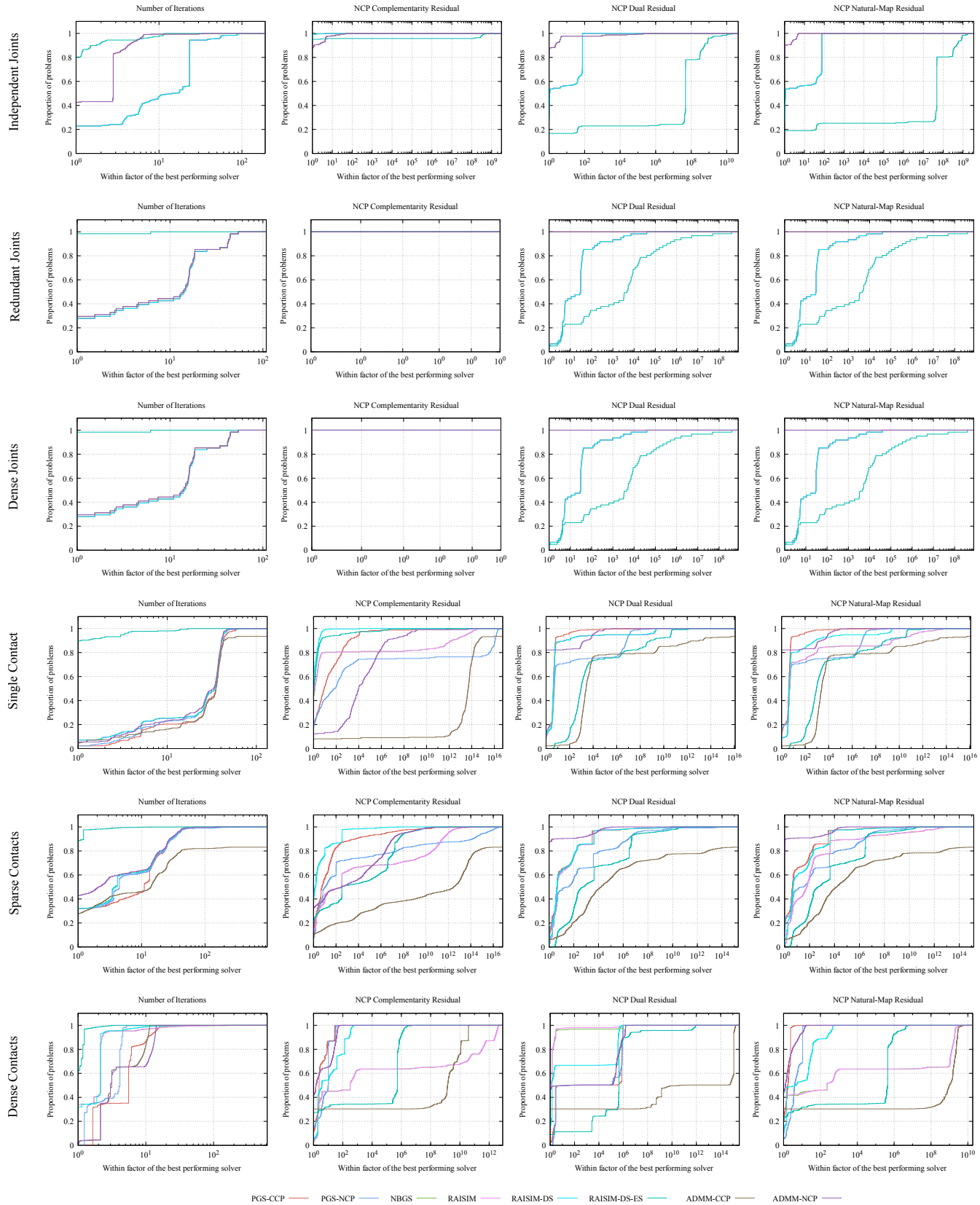

Fig. 52. Performance profiles w.r.t each benchmark sample category.

References

[1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aau5872

[2] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," *CoRR*, vol. abs/2108.10470, 2021. [Online]. Available: https://arxiv.org/abs/2108.10470

[3] L. de Matteis, V. Batto, J. Carpentier, and N. Mansard, "Optimal Control of Walkers with Parallel Actuation," Mar. 2025, working paper or preprint. [Online]. Available: https://hal.science/hal-04716938

[4] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 91–100. [Online]. Available: https://proceedings.mlr.press/v164/rudin22a.html

[5] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adi7566

[6] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

[7] V. Acary, M. Brémond, and O. Huber, "On solving contact problems with Coulomb friction: formulations and numerical comparisons," in *Advanced Topics in Nonsmooth Dynamics - Transactions of the European Network for Nonsmooth Dynamics*, S. I. Publishing, Ed., Jun. 2018, pp. 375–457. [Online]. Available: https://inria.hal.science/hal-01878539

[8] C. Schumacher, E. Knoop, and M. Bächer, "A versatile inverse kinematics formulation for retargeting motions onto robots with kinematic loops," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 943–950, 2021.

[9] G. Maloisel, C. Schumacher, E. Knoop, R. Grandia, and M. Bächer, "Optimal design of robotic character kinematics," *ACM Trans. Graph.*, vol. 42, no. 6, Dec. 2023. [Online]. Available: https://doi.org/10.1145/3618404

[10] ——, "A versatile quaternion-based constrained rigid body dynamics," *ACM Transactions on Graphics (SIGGRAPH)*, 2025.

[11] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[12] 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)*, 2014, pp. 6054–6061.

[13] J. Carpentier, R. Budhiraja, and N. Mansard, "Proximal and sparse resolution of constrained dynamic equations," in *Robotics: Science and Systems 2021*, 2021.

[14] A. S. Sathya and J. Carpentier, "Constrained articulated body dynamics algorithms," *IEEE Transactions on Robotics*, 2024.

[15] ——, "Constrained articulated body dynamics algorithms," *Trans. Rob.*, vol. 41, p. 430–449, Jan. 2025. [Online]. Available: https://doi.org/10.1109/TRO.2024.3502515

[16] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4397–4404.

[17] S. Andrews, K. Erleben, and Z. Ferguson, "Contact and friction simulation for computer graphics," in *ACM SIGGRAPH 2022 Courses*, ser. SIGGRAPH '22. New York, NY, USA: Association for Computing Machinery, 2022.

[18] C. Ericson, *Real-time collision detection*. Crc Press, 2004.

[19] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3859–3866.

[20] E. Coumans, "Bullet physics sdk: real-time collision detection and multi-physics simulation for vr, games, visual effects, robotics, machine learning etc." https://github.com/bulletphysics/bullet3, 2022.

[21] J. Pan, S. Chitta, D. Manocha, F. Lamiraux, J. Mirabel, J. Carpentier, L. Montaut *et al.*, "Coal: an extension of the flexible collision library," https://github.com/coal-library/coal, 2015–2024.

[22] F. E. Udwadia and P. Phohomsiri, "Explicit equations of motion for constrained mechanical systems with singular mass matrices and applications to multi-body dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 462, no. 2071, pp. 2097–2117, 2006.

[23] C. Glocker, "Set-valued force laws, ser. lecture notes in applied mathematics, f. pfeiffer, ed," 2001.

[24] D. E. Stewart, *Dynamics with Inequalities*. Society for Industrial and Applied Mathematics, 2011. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611970715

[25] J. J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," in *Nonsmooth mechanics and Applications*. Springer, 1988, pp. 1–82.

[26] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[27] F. Génot and B. Brogliato, "New results on painlevé paradoxes," *European Journal of Mechanics - A/Solids*, vol. 18, no. 4, pp. 653–677, 1999.

[28] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nuetzi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart, "A primer on the differential calculus of 3d orientations," 2016. [Online]. Available: https://arxiv.org/abs/1606.05285

[29] F. E. Udwadia and R. E. Kalaba, "A new perspective on constrained motion," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1906, pp. 407–410, 1992. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0158

[30] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717798001198

[31] V. Acary, F. Cadoux, C. Lemaréchal, and J. Malick, "A formulation of the linear discrete coulomb friction problem via convex optimization," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 91, no. 2, pp. 155–175, 2011.

[32] J. Carpentier, Q. Le Lidec, and L. Montaut, "From Compliant to Rigid Contact Simulation: a Unified and Efficient Approach," in *20th edition of the "Robotics: Science and Systems" (RSS) Conference*, Delft, Netherlands, Jul. 2024. [Online]. Available: https://hal.science/hal-04588906

[33] 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, 2002, pp. 517–522 vol.1.

[34] M. Anitescu, "Optimization-based simulation of nonsmooth rigid multibody dynamics," *Mathematical Programming*, vol. 105, no. 1, pp. 113–143, 2006. [Online]. Available: https://doi.org/10.1007/s10107-005-0590-7

[35] E. Drumwright and D. A. Shell, *Modeling Contact Friction and Joint Friction in Dynamic Robotic Simulation Using the Principle of Maximum Dissipation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 249–266. [Online]. Available: https://doi.org/10.1007/978-3-642-17452-0_15

[36] E. Todorov, "A convex, smooth and invertible contact model for trajectory optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1071–1076.

[37] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE. IEEE, 2012, pp. 5026–5033.

[38] 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.

[39] A. Chatterjee, "On the realism of complementarity conditions in rigid body collisions," *Nonlinear Dynamics*, vol. 20, no. 2, pp. 159–168, 1999. [Online]. Available: https://doi.org/10.1023/A:1008397905242

[40] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[41] DeepMind, "Online documentation for the mujoco physics simulator." https://mujoco.readthedocs.io/en/stable/computation/index.html, 2024.

[42] H. Mazhar, T. Heyn, D. Negrut, and A. Tasora, "Using nesterov's method to accelerate multibody dynamics with friction and contact," *ACM Trans. Graph.*, vol. 34, no. 3, May 2015. [Online]. Available: https://doi.org/10.1145/2735627

[43] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *High Performance Computing in Science and Engineering*, T. Kozubek, R. Blaheta, J. Šístek, M. Rozložník, and M. Čermák, Eds. Cham: Springer International Publishing, 2016, pp. 19–49.

[44] D. Wang, M. Servin, and T. Berglund, "Warm starting the projected gauss–seidel algorithm for granular matter simulation," *Computational Particle Mechanics*, vol. 3, pp. 43–52, 2016.

[45] A. Tasora, D. Mangoni, S. Benatti, and R. Garziera, "Solving variational inequalities and cone complementarity problems in nonsmooth dynamics using the alternating direction method of multipliers," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 16, pp. 4093–4113, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6693

[46] E. Griepentrog, *Index reduction methods for differential-algebraic equations*. Humboldt-Univ., Fachbereich Mathematik, Informationsstelle, 1991.

[47] H. Baruh, *Analytical dynamics*, ser. McGraw-Hill international editions. WCB/McGraw-Hill, 1999. [Online]. Available: https://cir.nii.ac.jp/crid/1130282272764601984

[48] R. Leine and C. Glocker, "A set-valued force law for spatial coulomb–contensou friction," *European Journal of Mechanics - A/Solids*, vol. 22, no. 2, pp. 193–216, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0997753803000251

[49] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[50] J. J. Moreau, "Application of convex analysis to some problems of dry friction," in *Trends in applications of pure mathematics to mechanics*. Pitman, 1977, pp. 263–280.

[51] F. Cadoux, "Optimization and convex analysis for nonsmooth dynamics," *HAL*, vol. 2009, no. 0, 2009. [Online]. Available: http://dml.mathdoc.fr/item/tel-00440798

[52] G. de Saxcé and Z.-Q. Feng, "New inequality and functional for contact with friction: the implicit standard material approach," *Journal of Structural Mechanics*, vol. 19, no. 3, pp. 301–325, 1991.

[53] C. Studer, *Numerics of unilateral contacts and friction: modeling and numerical time integration in non-smooth dynamics*. Springer Science & Business Media, 2009, vol. 47.

[54] K. Erleben, "Rigid body contact problems using proximal operators," in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, ser. SCA '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3099564.3099575

[55] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer methods in applied mechanics and engineering*, vol. 1, no. 1, pp. 1–16, 1972.

[56] T. Preclik, S. Eibl, and U. Rüde, "The maximum dissipation principle in rigid-body dynamics with inelastic impacts," *Computational Mechanics*, vol. 62, no. 1, pp. 81–96, 2018. [Online]. Available: https://doi.org/10.1007/s00466-017-1486-0

[57] R. Smith, "Open dynamics engine," 2008, http://www.ode.org/. [Online]. Available: http://www.ode.org/

[58] DeepMind, "Mujoco: A general purpose physics simulator for multi-joint dynamics with contact," https://github.com/google-deepmind/mujoco, 2024.

[59] A. Tikhonov, A. Goncharsky, V. Stepanov, and A. Yagola, "Numerical methods for the solution of ill-posed problems," *Netherlands: Kluwer Academic*, 1995.

[60] V. Acary and B. Brogliato, *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*. Springer Science & Business Media, 2008.

[61] B. Brogliato, A. Daniilidis, C. Lemarechal, and V. Acary, "On the equivalence between complementarity systems, projected systems and differential inclusions," *Systems & Control Letters*, vol. 55, no. 1, pp. 45–51, 2006.

[62] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[63] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. [Online]. Available: http://dx.doi.org/10.1561/2400000003

[64] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[65] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[66] C. Studer and C. Glocker, "Solving normal cone inclusion problems in contact mechanics by iterative methods," *Journal of System Design and Dynamics*, vol. 1, no. 3, pp. 458–467, 2007.

[67] C. Studer, "Augmented time-stepping integration of non-smooth dynamical systems," Ph.D. dissertation, ETH Zurich, 2008.

[68] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[69] G. Strang, *Linear Algebra And Its Aapplications*. Belmont, CA: Thomson, Brooks/Cole, 2006. [Online]. Available: http://www.amazon.com/Linear-Algebra-Its-Applications-Edition/dp/0030105676

[70] O. Bonnefon and G. Daviet, "Quartic formulation of Coulomb 3D frictional contact," INRIA, Technical Report RT-0400, Jan. 2011. [Online]. Available: https://inria.hal.science/inria-00553859

[71] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[72] T. Preclik, *Models and algorithms for ultrascale simulations of non-smooth granular dynamics*. Friedrich-Alexander-Universitaet Erlangen-Nuernberg (Germany), 2014.

[73] Q. Le Lidec and J. Carpentier, "Reconciling raisim with the maximum dissipation principle," *IEEE Transactions on Robotics*, vol. 40, pp. 3638–3641, 2024.

[74] B. C. Eaves, "On the basic theorem of complementarity," *Mathematical Programming*, vol. 1, no. 1, pp. 68–75, 1971. [Online]. Available: https://doi.org/10.1007/BF01584073

[75] M. Jean and J. J. Moreau, "Unilaterality and dry friction in the dynamics of rigid body collections," in *1st Contact Mechanics International Symposium*, Lausanne, Switzerland, 1992, pp. 31–48. [Online]. Available: https://hal.science/hal-01863710

[76] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782597001370

[77] M. Anitescu and G. D. Hart, "A fixed-point iteration approach for multibody dynamics with contact and small friction," *Mathematical Programming*, vol. 101, pp. 3–32, 2004.

[78] NVIDIA, "Nvidia physx: a scalable, multi-physics sdk for simulating and modeling physics in robotics, autonomous vehicles, and vfx workflows." https://github.com/NVIDIA-Omniverse/PhysX, 2024.

[79] K. Erleben, "Velocity-based shock propagation for multibody dynamics animation," *ACM Trans. Graph.*, vol. 26, no. 2, p. 12–es, Jun. 2007. [Online]. Available: https://doi.org/10.1145/1243980.1243986

[80] A. Tasora and M. Anitescu, "A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 5, pp. 439–453, 2011.

[81] G. Daviet, "Simple and scalable frictional contacts for thin nodal objects," *ACM Trans. Graph.*, vol. 39, no. 4, Aug. 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392439

[82] ——, "Interactive hair simulation on the gpu using admm," in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3588432.3591551

[83] J. Lee, M. Lee, S. Park, J. Yun, and D. Lee, "Variations of augmented lagrangian for robotic multi-contact simulation," 2025. [Online]. Available: https://arxiv.org/abs/2502.16898

[84] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, "A general analysis of the convergence of admm," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 343–352.

[85] R. Mises and H. Pollaczek-Geiringer, "Praktische verfahren der gleichungsauflösung." *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 1, pp. 58–77, 1929.

[86] V. Acary, M. Brémond, T. Koziara, and F. Pérignon, "FCLIB: a collection of discrete 3D Frictional Contact problems," INRIA, France, Technical Report RT-0444, Feb. 2014. [Online]. Available: https://inria.hal.science/hal-00945820

[87] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. comput. math*, vol. 15, no. 1, pp. 3–43, 2016.

[88] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002. [Online]. Available: https://doi.org/10.1007/s101070100263

[89] N. J. Dingle and N. J. Higham, "Reducing the influence of tiny normwise relative errors on performance profiles," *ACM Trans. Math. Softw.*, vol. 39, no. 4, Jul. 2013. [Online]. Available: https://doi.org/10.1145/2491491.2491494

[90] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[91] E. Drumwright and D. A. Shell, "An evaluation of methods for modeling contact in multibody simulation," *2011 IEEE International Conference on Robotics and Automation*, pp. 1695–1701, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:9550173

[92] K. Erleben, "Stable, robust, and versatile multibody dynamics animation," *Ph. D. Thesis, University of Copenhagen, Copenhagen*, pp. 1–241, 2004.

[93] W. C. Mitchell, A. Staniforth, and I. Scott, "Analysis of ackermann steering geometry," SAE Technical Paper, Tech. Rep., 2006.

[94] P. Egli and M. Hutter, "A general approach for the automation of hydraulic excavator arms using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5679–5686, 2022.

[95] 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.

[96] K. G. Gim, J. Kim, and K. Yamane, "Design and fabrication of a bipedal robot using serial-parallel hybrid leg mechanism," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5095–5100.

[97] C. Schumacher, E. Knoop, and M. Bächer, "A versatile inverse kinematics formulation for retargeting motions onto robots with kinematic loops," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 943–950, 2021.

[98] J. Tan, K. Liu, and G. Turk, "Stable proportional-derivative controllers," *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 34–44, 2011.

[99] Z. Yin and K. Yin, "Linear time stable pd controllers for physics-based character animation," *Computer Graphics Forum*, vol. 39, no. 8, pp. 191–200, 2020.

[100] 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, pp. 429–455, 2016.

[101] V. Acary and F. Pérignon, "Siconos: A software platform for modeling, simulation, analysis and control of nonsmooth dynamical systems," *SNE Simulation News Europe*, vol. 17, no. 3/4, pp. 19–26, 2007.

[102] F. Dubois and R. Mozul, "Lmgc90," in *11e colloque national en calcul des structures*, 2013.

[103] Y. Lu, *A framework for comparison of methods for solving complementarity problems that arise in multibody dynamics*. Rensselaer Polytechnic Institute, 2016.

[104] The HDF Group, "Hierarchical Data Format, version 5," [Online]. Available: https://github.com/HDFGroup/hdf5.

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