

From Compliant to Rigid Contact Simulation: a Unified and Efficient Approach

Justin Carpentier*

Inria, École normale supérieure
CNRS, PSL Research University
75005 Paris, France
Email: justin.carpentier@inria.fr

Quentin Le Lidec*

Inria, École normale supérieure
CNRS, PSL Research University
75005 Paris, France
Email: quentin.le-lidec@inria.fr

Louis Montaut*

Inria, École normale supérieure
CNRS, PSL Research University
75005 Paris, France
Email: louis.montaut@inria.fr

Abstract—Whether rigid or compliant, contact interactions are inherent to robot motions, enabling them to move or manipulate things. Contact interactions result from complex physical phenomena, that can be mathematically cast as Nonlinear Complementarity Problems (NCPs) in the context of rigid or compliant point contact interactions. Such a class of complementarity problems is, in general, difficult to solve both from an optimization and numerical perspective. Over the past decades, dedicated and specialized contact solvers, implemented in modern robotics simulators (e.g., Bullet, Drake, MuJoCo, DART, Raisim) have emerged. Yet, most of these solvers tend either to solve a relaxed formulation of the original contact problems (at the price of physical inconsistencies) or to scale poorly with the problem dimension or its numerical conditioning (e.g., a robotic hand manipulating a paper sheet). In this paper, we introduce a unified and efficient approach to solving NCPs in the context of contact simulation. It relies on a sound combination of the Alternating Direction Method of Multipliers (ADMM) and proximal algorithms to account for both compliant and rigid contact interfaces in a unified way. To handle ill-conditioned problems and accelerate the convergence rate, we also propose an efficient update strategy to adapt the ADMM hyperparameters automatically. By leveraging proximal methods, we also propose new algorithmic solutions to efficiently evaluate the inverse dynamics involving rigid and compliant contact interactions, extending the approach developed in MuJoCo. We validate the efficiency and robustness of our contact solver against several alternative contact methods of the literature and benchmark them on various robotics and granular mechanics scenarios. Overall, the proposed approach is shown to be competitive against classic methods for simple contact problems and outperforms existing solutions on more complex scenarios, involving tens of contacts and poor conditioning. Our code is made open-source at <https://github.com/Simple-Robotics/Simple>.

I. INTRODUCTION

Contact interactions are the substrate of movement. When performing locomotion or manipulation tasks, one naturally makes and breaks contact with the environment. Providing robots with such abilities would require them to precisely apprehend the physics of contact. In this respect, simulators have been key components for various robotics applications. Reinforcement Learning (RL) policies for manipulation [12, 24] or locomotion [35] often rely on years of simulated trajectories during training. Alternatively, Model Predictive

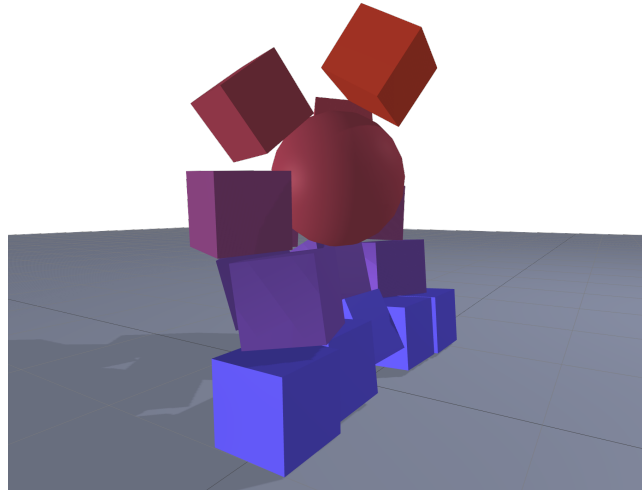


Fig. 1. **Simulation of an ill-conditioned stack of cubes.** Our approach robustly solves high-dimensional and ill-conditioned problems such as the ones involved when simulating a stack of 15 cubes involving more than 60 contact points and objects’ masses varying from 1 kg (bottom row, in light blue) to 10.000 kg (highest row, in bright red). Simulation videos at https://youtu.be/fi_qg9cTx0NY?si=NGtx1tiYrIGtHXSK.

Control (MPC) techniques call the simulator and its derivatives at high frequency at runtime to achieve a reactive behavior [32, 30].

Simulating rigid contact interactions with friction requires solving a Nonlinear Complementarity Problem (NCP) which has been recognized as a hard problem both from an optimization and numerical perspective [2]. For this reason, simulators proceed to tradeoffs between physical realism, robustness, and efficiency. In this respect, each physics engine embeds its own contact model and contact solver, which come with their underpinning physical hypotheses and numerical capabilities.

Earlier simulators like ODE [49] and Bullet [14] have been developed for graphical purposes. Graphical applications mainly require the physics engine to provide visually consistent trajectories. Nowadays, the requirements in robotics applications are different: the physics of contacts should be as close as possible to reality, and the contact solver should be numerically efficient. Indeed, this would reduce training time and enhance the transferability of RL while making MPC more reactive.

More recently, robotics-oriented engines were developed

*Equal contribution. Authors are listed in alphabetical order.

with a focus on efficiency. MuJoCo [54] introduces a physical relaxation that induces a convex contact model with distant and compliant interactions. This choice enables the use of powerful algorithms from the optimization literature and leads to improved stability of the simulation. Its robustness has made MuJoCo the standard testbed for RL algorithms, but its reality gap still limits its use in robotics. Drake [53] leverages a similar contact model [11] and proposes a procedure to set the compliance parameter more realistically, according to the physical parameters of the simulated objects. This results in an improved sim-to-real transfer capability on manipulation tasks [29]. RaiSim [27] fixes MuJoCo’s artifacts due to distant and artificially compliant contact forces but at the expense of less robust numerics [33]. The resulting contact model drove success in quadrupedal locomotion in challenging setups [28, 35, 39].

Recent works [25, 33] compare the different existing approaches and their impact. Although they highlight the significant improvements in contact simulation driven by the efforts of the robotics community, they also reveal current simulators are all subject to trade-offs and, thus, some barriers are still to be removed. Some experimental works [20, 3] also evaluate contact models from current simulators against real-world data. They show that they still fail to capture accurately the physical phenomena involved in contact interactions. Inspired by these studies, we aim to improve the physicality of current simulators while preserving their efficiency.

We make the following contributions:

- we introduce a new formulation and related algorithm for solving forward dynamics problems with contacts that can efficiently deal with compliant and rigid contacts without making any physical relaxation,
- we propose a formulation and an algorithmic solution for computing inverse dynamics with rigid and compliant contacts, the reciprocal of forward dynamics, extending the seminal work of Todorov [54],
- we provide an open-source and efficient C++ implementation leveraging the Eigen library [23], and that can be easily plugged into existing simulation frameworks,
- we extensively evaluate the proposed solutions on both mechanics and robotics scenarios of the literature.

The paper is organized as follows. We first provide the necessary background on constrained dynamics, how it connects to proximal optimization, the NCP appearing during the simulation of frictional contacts and the existing solver from the robotics community (Sec. II). Second, we propose an ADMM-based algorithm for contact solving (Sec. III). Third, we introduce a similar algorithm for the problem of inverse dynamics in the presence of contacts (Sec. IV). Finally, we evaluate our approach against standard problems of the robotics and mechanics communities (Sec. V) before discussing how it compares to previous works (Sec. VI).

II. BACKGROUND

In this section, we recall the principles underlying the simulation of systems under rigid and compliant constraints. To prepare the ground for the proposed algorithm, we introduce proximal operators and how they can be interpreted from a mechanics point of view. Finally, we introduce the NCP underlying the simulation of physical systems in the presence of frictional contacts and the existing approaches to solve it.

A. Equality-constrained dynamics

The free motion of a poly-articulated system is governed by the so-called Lagrangian dynamics of the form

$$M(\mathbf{q})\dot{\mathbf{v}} + \mathbf{b}(\mathbf{q}, \mathbf{v}) = \boldsymbol{\tau} \quad (1)$$

where we denote by $\mathbf{q} \in \mathcal{Q} \cong \mathbb{R}^{n_q}$, $\mathbf{v} \in \mathcal{T}_q \mathcal{Q} \cong \mathbb{R}^{n_v}$ and $\boldsymbol{\tau} \in \mathcal{T}_q^* \mathcal{Q} \cong \mathbb{R}^{n_v}$, the joint configuration vector, the joint velocity vector and the joint torque vector. $M(\mathbf{q})$ is the joint-space inertia matrix and $\mathbf{b}(\mathbf{q}, \mathbf{v})$ includes terms related to the gravity, Coriolis, and centrifugal effects. Considering constraints such as kinematic loop closures or anchor points can be done by adding implicit equations of the form

$$\mathbf{f}_c(\mathbf{q}) = 0, \quad (2)$$

where $\mathbf{f}_c : \mathcal{Q} \mapsto \mathbb{R}^m$ is the constraint function of dimension m . Such constraints act on the system via the constraint forces $\boldsymbol{\lambda} \in \mathbb{R}^m$ that are spanned by the transpose of its Jacobian, denoted by $J_c \in \mathbb{R}^{m \times n_v}$. Thus, the constrained equation of motion reads

$$M(\mathbf{q})\dot{\mathbf{v}} + \mathbf{b}(\mathbf{q}, \mathbf{v}) = \boldsymbol{\tau} + J_c^\top \boldsymbol{\lambda}. \quad (3)$$

In order to simulate the constrained system, given a torque input $\boldsymbol{\tau}$ and knowing \mathbf{q} , \mathbf{v} , one needs to solve (3) and (2) for the unknowns $\dot{\mathbf{v}}$ and $\boldsymbol{\lambda}$. To do so, it is more convenient to first rewrite (2) as a function of $\dot{\mathbf{v}}$ by proceeding to index reduction. Thus, differentiating (2) twice with respect to time yields

$$J_c \dot{\mathbf{v}} + \underbrace{\dot{J}_c \mathbf{v}}_{\gamma(\mathbf{q}, \mathbf{v})} = 0. \quad (4)$$

For index reduction to be valid, it implicitly assumes that the preceding once differentiated and the original systems (2) are verified at the current time-step. Due to numerical reasons, such a condition is never exactly met, and a corrective term is added to γ to avoid numerical drift as done in the Baumgarte stabilization techniques. Combining the equations (3) and (4) leads to

$$\begin{pmatrix} M & J_c^\top \\ J_c & 0_{m \times m} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{v}} \\ -\boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} - \mathbf{b}(\mathbf{q}, \mathbf{v}) \\ -\gamma(\mathbf{q}, \mathbf{v}) \end{pmatrix}. \quad (5)$$

At this stage, it is worth noting that (5) is often not invertible as cases where J_c is rank deficient i.e. hyper-static systems, are common in robotics.

B. Compliant constraints

In the previously described constrained dynamics, rigid constraints were modeled via (2) which enforces f_c to be null. When modeling compliant constraints, the function f_c is allowed to be non-null but now defines a potential energy $U(\mathbf{q}) = \frac{1}{2}\|f_c(\mathbf{q})\|_{R^{-1}}^2$ where $\|x\|_A = \sqrt{x^\top A x}$ for any $x \in \mathbb{R}^m$ and any $A \in \mathbb{R}^m$ positive semi-definite. Here, $R \in \mathbb{R}^{m \times m}$ denotes compliance, which is a physical property of the system. Such potential energy leads to a constraint torque τ_c

$$\tau_c = -\nabla_{\mathbf{q}} U(\mathbf{q}) = -J_c^\top \underbrace{R^{-1} f_c(\mathbf{q})}_{-\lambda}, \quad (6)$$

which induces (2) to be replaced by a linear mapping between constraint forces and constraint violation

$$f_c(\mathbf{q}) = -R\lambda. \quad (7)$$

As previously described, proceeding to index reduction with (7) and adding (3), simulation of compliant constraints leads to the following system

$$\begin{pmatrix} M & J_c^\top \\ J_c & -R \end{pmatrix} \begin{pmatrix} \dot{\mathbf{v}} \\ -\lambda \end{pmatrix} = \begin{pmatrix} \tau - b(\mathbf{q}, \mathbf{v}) \\ -\gamma \end{pmatrix}, \quad (8)$$

whose only difference with (5) lies in the null lower right block being replaced by the compliance matrix R . Numerically, this difference appears to be critical. Indeed, R being positive definite makes the system (8) always well-defined and invertible [55].

C. Constrained dynamics from an optimization perspective

All the previous equations could be alternatively derived from an optimization standpoint. Indeed, one could observe that (5) coincides with the Karush–Kuhn–Tucker (KKT) conditions of the following Quadratic Programming (QP):

$$\begin{aligned} \min_{\dot{\mathbf{v}}} \quad & \frac{1}{2} \|\dot{\mathbf{v}} - \dot{\mathbf{v}}_f\|_M^2 \\ \text{s.t.} \quad & J_c \dot{\mathbf{v}} + \gamma = 0 \end{aligned} \quad (9)$$

where $\dot{\mathbf{v}}_f = M^{-1}(\tau - b(\mathbf{q}, \mathbf{v}))$ is the so-called joint free acceleration of the unconstrained system (1). Problem (9) corresponds to the formulation of the least constraint principle [7] and is equivalent to the following saddle-point problem:

$$\min_{\dot{\mathbf{v}}} \max_{\lambda} \mathcal{L}(\dot{\mathbf{v}}, \lambda) \quad (10)$$

with $\mathcal{L}(\dot{\mathbf{v}}, \lambda) = \frac{1}{2} \|\dot{\mathbf{v}} - \dot{\mathbf{v}}_f\|_M^2 - \lambda^\top (J_c \dot{\mathbf{v}} + \gamma)$ being the Lagrangian associated to (9).

Similarly, the compliant system (8) could be retrieved from the regularized problem:

$$\min_{\dot{\mathbf{v}}} \max_{\lambda} \mathcal{L}(\dot{\mathbf{v}}, \lambda) - \frac{1}{2} \|\lambda\|_R^2. \quad (11)$$

where the deformation energy acts as a Tikhonov regularization on the dual variables (i.e., the constraint forces) of the rigid constrained problem (10).

D. Proximal algorithms

Proximal algorithms [45] are a general class of optimization techniques, ubiquitous in convex optimization, and are essentially rooted around the notion of proximal operators. The proximal operator [40] of a convex function f is defined by

$$\text{prox}_{\rho, f}(x) = \arg \min_y f(y) + \frac{\rho}{2} \|x - y\|_2^2 \quad (12)$$

where $\rho \in \mathbb{R}_{>0}$ is homogeneous to the inverse of a step size. A specific property of this operator is that its fixed points coincide with the set of minimizers of f . The proximal point algorithm [45] uses the corresponding fixed point iteration

$$x^{k+1} = \text{prox}_{\rho, f}(x^k), \quad (13)$$

and is known to converge towards a minimizer of f .

Considering the particular case of QP, one seeks to solve

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^\top H x + g^\top x \\ \text{s.t.} \quad & A x = b \end{aligned} \quad (14)$$

where $H \in \mathbb{R}^{q \times q}$ is a symmetric positive semi-definite matrix, $A \in \mathbb{R}^{p \times q}$, $g \in \mathbb{R}^q$ and $b \in \mathbb{R}^p$. The Lagrangian of this problem writes,

$$\mathcal{L}(x, z) = \frac{1}{2} x^\top H x + g^\top x - z^\top (A x - b), \quad (15)$$

where $z \in \mathbb{R}^p$ is the dual variable. The solution to the problem verifies

$$x^*, z^* = \arg \min_x \max_z \mathcal{L}(x, z) \quad (16)$$

and the KKT conditions

$$\begin{pmatrix} H & A^\top \\ A & 0_{p \times p} \end{pmatrix} \begin{pmatrix} x^* \\ -z^* \end{pmatrix} = \begin{pmatrix} -g \\ b \end{pmatrix}. \quad (17)$$

When applied to the dual part of the QP (14), the proximal point algorithm can be written as

$$x^{k+1}, z^{k+1} = \arg \min_x \max_z \mathcal{L}(x, z) - \frac{\rho}{2} \|z - z^k\|_2^2, \quad (18)$$

where the *minus sign* in front of the term $-\frac{\rho}{2} \|z - z^k\|_2^2$ comes from the maximization over the dual variable z . The solution of the proximal iteration defined in Eq. (18) is obtained by solving the following system of equations

$$\begin{pmatrix} H & A^\top \\ A & -\rho \text{Id} \end{pmatrix} \begin{pmatrix} x^{k+1} \\ -z^{k+1} \end{pmatrix} = \begin{pmatrix} -g \\ b - \rho z^k \end{pmatrix}. \quad (19)$$

This approach is widely used in optimization and at the core of several optimization solvers [51, 4] and algorithms [10] for solving constrained dynamical problems.

Link between proximal and compliance terms. Going back to the simulation of constrained systems, one should notice the similarity between (8) and (19). Interestingly, drawing parallels between these two systems, the proximal regularization acts as a numerical compliance, thus making the iterates (19) always well-defined. However, due to different regularization terms in (11) and (18), the effect of the proximal operator differs

from a standard mechanical compliance. This translates into a shift in the right-hand side of (19) through the term $-\rho z^k$. As further discussed in [45], this term cancels out over the iterations, making the proximal solution converge towards the solution of the original rigid problem (17). Therefore, proximal regularization can be seen as a numerical technique akin to vanishing compliance.

E. Modeling frictional unilateral contacts: the Nonlinear Complementary Problem

Previously described equations govern the motion of systems under equality constraints, *i.e.*, bilateral constraints. However, punctual contact interactions are subject to three main modeling hypotheses: the unilateral contact constraint, the Coulomb friction law, and the Maximum Dissipation Principle (MDP). Next, we detail these principles and the problem they induce.

Unilateral contact hypothesis. Through the unilateral hypothesis, one assumes that objects cannot interpenetrate, yielding the following inequality:

$$\Phi(\mathbf{q})_N \geq 0, \quad (20)$$

where $\Phi(\mathbf{q}) \in \mathbb{R}^{3n_c}$ denotes the separation vector [19] defined as the minimum norm translation vector putting two shapes at a null distance, n_c being the number of contact points, and the subscripts N and T refer to the normal and tangential indices, respectively.

By duality, such a constraint induces contact forces $\boldsymbol{\lambda} \in \mathbb{R}^{3n_c}$ that can only act in a repulsive fashion and when objects are in contact, *i.e.*, when $\Phi(\mathbf{q})_N = 0$. These constraints are summarized by the *Signorini condition* [48]

$$0 \leq \boldsymbol{\lambda}_N \perp \Phi(\mathbf{q})_N \geq 0. \quad (21)$$

where $a \perp b$ for vectors a and b means $a^\top b = 0$.

We use an impulse-based formulation to deal with rigid dynamics and impacts. Applying the Euler symplectic scheme to discretize (3) leads to

$$M(\mathbf{v}^{t+1} - \mathbf{v}_f) = J_c^\top \boldsymbol{\lambda} \quad (22)$$

where $\boldsymbol{\lambda}$ now denotes an impulse, $\mathbf{v}_f = \Delta t \dot{\mathbf{v}}_f$, Δt being the time step, and $J_c \in \mathbb{R}^{3n_c \times n_v}$ is the Jacobian of Φ . Here, the choice of integration scheme only requires one costly evaluation of M , J_c , \mathbf{v}_f at the current time step \mathbf{q}^t , \mathbf{v}^t . More involved implicit integrators could be envisaged to improve stability, but they would also need to evaluate these operators multiple times. In particular, they would require multiple calls to the contact-detection and rigid-body algorithms, which can be computationally expensive. As was done earlier, the *Signorini condition* can be written in impulse via index reduction [41]

$$0 \leq \boldsymbol{\lambda}_N \perp (J_c \mathbf{v}^{t+1} + \gamma)_N \geq 0 \quad (23)$$

where γ is equal to $\Phi(\mathbf{q}^t)$ plus previously mentioned corrective terms. We note $\mathbf{c} = J_c \mathbf{v}^{t+1} \in \mathbb{R}^{3n_c}$ the contact point velocities.

Similarly to the bilateral case from Sec. II, any contact constraint can be made compliant by modifying the *Signorini condition*

$$0 \leq \boldsymbol{\lambda}_N \perp (J_c \mathbf{v}^{t+1} + \gamma + R\boldsymbol{\lambda})_N \geq 0. \quad (24)$$

The additional compliance term of (24) introduces possible interpenetration which can be used to model soft bodies. More involved deformation models based on the finite element method [46] are also used in robotics [36] but requires an extra computational cost.

Coulomb's law of friction is generally employed to model dry friction via

$$\forall i, \lambda^{(i)} \in \mathcal{K}_{\mu^{(i)}} \quad (25)$$

where $\mathcal{K}_{\mu^{(i)}} = \{\lambda^{(i)} \in \mathbb{R}^3 \mid \|\lambda_T^{(i)}\|_2 \leq \mu^{(i)} \lambda_N^{(i)}\}$ is the second-order friction cone, $\mu = (\mu^{(1)}, \dots, \mu^{(n_c)}) \in \mathbb{R}_{\geq 0}^{n_c}$ is the vector of coefficient of frictions and the superscript (i) refers to the indices associated with the i -th contact point. By noting the Cartesian product $\mathcal{K}_\mu = \prod_{i=1}^{n_c} \mathcal{K}^{(i)}$, (25) can be aggregated into $\boldsymbol{\lambda} \in \mathcal{K}_\mu$.

Maximum dissipation principle. In addition, the maximum dissipation principle (MDP) [41] states that, whenever a contact point is sliding, friction impulses should maximize the dissipated power

$$\forall i, \lambda_T^{(i)} = \arg \min_{\beta_T, \|\beta_T\| \leq \mu^{(i)} \lambda_N^{(i)}} \beta_T^\top \mathbf{c}_T^{(i)}. \quad (26)$$

where c is the contact point velocity previously defined.

Nonlinear complementary problem. Finally, combining equations (22), (24), (25), (26), $\boldsymbol{\lambda}$ verifies the following NCP

$$\mathcal{K}_\mu \ni \boldsymbol{\lambda} \perp \boldsymbol{\sigma} + \Gamma(\boldsymbol{\sigma}) \in \mathcal{K}_\mu^* \quad (27)$$

where $\mathcal{K}_\mu^* = \mathcal{K}_{\frac{1}{\mu}}$ is the dual friction cone of \mathcal{K}_μ , $G = J_c M^{-1} J_c^\top$ is the so-called Delassus matrix [16], $\mathbf{g} = J_c \mathbf{v}_f + \gamma$ is the free contact point velocities plus corrective terms, and we use the shorthand notation $\boldsymbol{\sigma} = (G + R)\boldsymbol{\lambda} + \mathbf{g}$. We use the notation $\Gamma(\boldsymbol{\sigma}) = (\Gamma^{(1)}(\boldsymbol{\sigma}^{(1)}) \dots \Gamma^{(n_c)}(\boldsymbol{\sigma}^{(n_c)})) \in \mathbb{R}^{3n_c}$ with $\Gamma^{(i)}(\boldsymbol{\sigma}^{(i)}) = \begin{pmatrix} 0 & 0 & \mu^{(i)} \|\boldsymbol{\sigma}_T^{(i)}\| \end{pmatrix} \in \mathbb{R}^3$ denoting the De Saxcé correction [15].

F. Existing solvers

Due to the non-convexity and nonsmoothness of the complementarity constraint, the nonlinearity of the DeSaxcé correction, and the ill-conditioning of the Delassus matrix G , the NCP (27) is known to be a numerically hard problem to solve in general [2].

A first class of simulators, *e.g.*, ODE [49], PhysX [37], DART[34], proceed by linearizing the friction cones to get an approximate but more tractable Linear Complementarity Problem (LCP). LCPs are well-studied [13] and can be solved with the Projected Gauss-Seidel (PGS) algorithm. As a first-order algorithm, PGS is not affected by null eigenvalues due

TABLE I
CHARACTERISTICS OF THE CONTACT SOLVERS IN ROBOTICS.

Physics engine	Complementarity problem	Contact type	Algorithm
ODE[49], PhysX[37], DART [34]	LCP	hard	PGS
Bullet[14]	NCP	hard	PGS
MuJoCo[54], Drake [53]	CCP	soft	non-smooth Newton
RaiSim[27]	-*	hard	per-contact bisection
Dojo[26]	NCP	hard	Interior Point
Ours	NCP	hard & soft	ADMM

*RaiSim does not model the contact problem as a complementarity problem.

to hyperstaticity but is sensitive to the conditioning of the Delassus matrix G , thus hindering robustness. In its recent versions, Bullet [14] implements a similar PGS algorithm working on the original second-order friction cone.

An alternative approach, notably adopted in MuJoCo[54] and Drake[53], consists in ignoring the DeSaxcé correction in (27), which has the effect of relaxing the *Signorini condition* [33]. This leads to a Cone Complementarity Problem (CCP) which is equivalent to a convex Second-Order Cone Programming (SOCP) problem [6]. Such a problem can be solved via robust off-the-shelf optimization algorithms benefiting from strong guarantees. In this respect, a previous work [52] uses an ADMM algorithm to solve this problem. In MuJoCo [54] and Drake [53] simulators, the hyperstatic cases are handled by systematically adding compliance to the problem, thus making it impossible to simulate purely rigid systems.

RaiSim [27] uses a specific contact model enforcing the *Signorini condition*, which combines favorably with its custom per-contact bisection algorithm. Despite its computational efficiency, this approach inherits the sensitivity to conditioning from Gauss-Seidel-like techniques.

Dojo [26] avoids any physical relaxation and uses an Interior Point (IP) algorithm to solve (27). As a second-order algorithm, Dojo's solver can handle ill-conditioned problems. However, just like IP algorithms, the resulting algorithm is difficult to warm-start and requires an expensive Cholesky computation at each iteration, which makes the approach time-consuming, and thus limits its range of applications, notably in the context of real-time control scenarios.

III. EFFICIENT SOLVING OF FRICTIONAL CONTACT DYNAMICS

In this section, we detail the central contribution of this paper, namely a novel algorithm to solve NCPs of the form of problem (27). At the core of our approach, is the development of a new ADMM method and update strategy, enabling us to solve complex contact problems, that might be poorly conditioned, as it might occur in contact mechanics problems.

A. NCP as a cascade of optimization problems

The NCP formulated in (27) corresponds to the solving of the interweaving problems of the form

$$\lambda = \arg \min_{f \in \mathcal{K}_\mu} \frac{1}{2} f^\top (G + R) f + f^\top (\Gamma(\sigma) + g), \quad (28a)$$

and

$$\sigma = (G + R)\lambda + g. \quad (28b)$$

Indeed, if we denote by $z \in \mathcal{K}_\mu^*$ the dual variable associated to the primal variable $\lambda \in \mathcal{K}_\mu$, the Lagrangian of Problem (28) reads

$$\mathcal{L}(f, z) = \frac{1}{2} f^\top (G + R) f + f^\top (\Gamma(\sigma) + g) - f^\top z. \quad (29)$$

The optimality conditions are thus given by canceling the gradient of \mathcal{L} w.r.t. f at the optimal solution (λ, z^*)

$$\nabla_f \mathcal{L}(\lambda, z^*) = (G + R)\lambda + g + \Gamma(\sigma) - z^* = 0, \quad (30)$$

and the fact that

$$\mathcal{K}_\mu \ni \lambda \perp z^* \in \mathcal{K}_\mu^*. \quad (31)$$

Injecting (30) into (31), leads to the nonlinear complementarity conditions of (27).

Interestingly, (28a) appears to be a convex problem which proves to be useful as it allows sub-steps of our ADMM approach to be performed efficiently. However, the interweaving of (28a) and (28b) through the non-smooth de Saxcé function Γ makes the problem (27) difficult to solve in general by standard optimization techniques [2]. As discussed in [2], one approach consists in incorporating estimates of $\Gamma(\sigma)$ (which is equal to $\Gamma(z^*)$), via the updates of a variable named s , inside the solving of (28a). More specifically, this results in a cascade of optimization problems and is done by setting, at the k^{th} iterate, $s_k = \Gamma(z_k)$ (Alg. 1, line 2), corresponding to the nonlinear term in (28) considered as constant, then solving (28a) (Alg. 1, line 4) and updating z_k (Alg. 1, line 6) from the new optimal force vector f_k . The solving of this cascade of problems continues until the primal and dual optimal convergence criteria have been met up to a certain numerical tolerance ϵ_{abs} .

This cascaded strategy has shown to be effective in practice [2]. A critical part involves solving the inner optimization problem (28a) efficiently, which is nothing more than a Quadratically Constrained Quadratic Program (QCQP). One approach could thus consist in leveraging existing SOCP

solvers, such as ECOS [18] or SCS [43]. Yet, because of their high level of versatility, such off-the-shelf solvers tend to be less efficient when considering a specific class of problems such as the ones occurring in contact mechanics [2]. Following this line of thought, we instead propose to develop a dedicated and efficient ADMM approach for solving 28a, particularly suited for solving ill-conditioned problems.

B. Proximal ADMM formulation

By introducing a slack variable \mathbf{y} such that $\mathbf{y} = \mathbf{f}$, the QCQP (28a) is made separable as follows

$$\min_{\mathbf{f}, \mathbf{y}} h_1(\mathbf{f}) + h_2(\mathbf{y}) \quad (32a)$$

$$\text{s.t. } \mathbf{f} = \mathbf{y}, \quad (32b)$$

where $h_1(\mathbf{f}) := \frac{1}{2} \mathbf{f}^\top (G + R) \mathbf{f} + \mathbf{f}^\top (s + g)$ is a smooth convex function with s being the current estimate of $\Gamma(\sigma)$, and $h_2(\mathbf{y}) := \mathcal{I}_{\mathcal{K}_\mu}(\mathbf{y})$ is the nonsmooth indicator function associated with the convex cone \mathcal{K}_μ defined as

$$\mathcal{I}_{\mathcal{K}_\mu}(\mathbf{y}) = \begin{cases} 0, & \text{if } \mathbf{y} \in \mathcal{K}_\mu \\ +\infty, & \text{otherwise} \end{cases}.$$

By naming \mathbf{z} the dual variable associated with (32b), the augmented Lagrangian of problem (32) reads

$$\mathcal{L}_\rho^A(\mathbf{f}, \mathbf{y}, \mathbf{z}) = h_1(\mathbf{f}) + h_2(\mathbf{y}) - \mathbf{z}^\top (\mathbf{f} - \mathbf{y}) + \frac{\rho}{2} \|\mathbf{f} - \mathbf{y}\|_2^2 \quad (33)$$

where ρ is the augmented Lagrangian penalty term [5]. \mathcal{L}_ρ^A defined in (33) can be equivalently rewritten as

$$\mathcal{L}_\rho^A(\mathbf{f}, \mathbf{y}, \mathbf{z}) = h_1(\mathbf{f}) + h_2(\mathbf{y}) + \frac{\rho}{2} \left\| \mathbf{f} - \mathbf{y} - \frac{\mathbf{z}}{\rho} \right\|_2^2 - \frac{1}{\rho} \|\mathbf{z}\|_2^2. \quad (34)$$

Finally, to make the sub-problem associated to h_1 strongly convex w.r.t. \mathbf{f} , we propose to add the proximal term $\frac{\eta}{2} \|\mathbf{f} - \mathbf{f}^-\|_2^2$ to \mathcal{L}_ρ^A , leading to

$$\begin{aligned} \mathcal{L}_{\rho, \eta}^A(\mathbf{f}, \mathbf{y}, \mathbf{z}) &= h_1(\mathbf{f}) + \frac{\eta}{2} \|\mathbf{f} - \mathbf{f}^-\|_2^2 + h_2(\mathbf{y}) \\ &\quad + \frac{\rho}{2} \left\| \mathbf{f} - \mathbf{y} - \frac{\mathbf{z}}{\rho} \right\|_2^2 - \frac{1}{\rho} \|\mathbf{z}\|_2^2, \end{aligned} \quad (35)$$

where \mathbf{f}^- is the previous estimate of \mathbf{f} . This proximal term affects the conditioning of the problem by adding a regularization to the Hessian of h_1 , which will play an essential role in the update strategy further detailed in Sec. III-F. η is fixed and typically set to a value of 10^{-6} .

C. Pseudocode of the ADMM updates

Solving the saddle-point problem associated with the proximal augmented Lagrangian $\mathcal{L}_{\rho, \eta}^A$ defined in (35) can be efficiently done via ADMM iterates. The updates are defined by the following successive optimization steps:

$$\mathbf{f}_k = \arg \min_{\mathbf{f}} \mathcal{L}_{\rho, \eta}^A(\mathbf{f}, \mathbf{y}_{k-1}, \mathbf{z}_{k-1}) \quad (36a)$$

$$\mathbf{y}_k = \arg \min_{\mathbf{y}} \mathcal{L}_{\rho, \eta}^A(\mathbf{f}_k, \mathbf{y}, \mathbf{z}_{k-1}) \quad (36b)$$

$$\mathbf{z}_k = \mathbf{z}_{k-1} - \rho(\mathbf{f}_k - \mathbf{y}_k). \quad (36c)$$

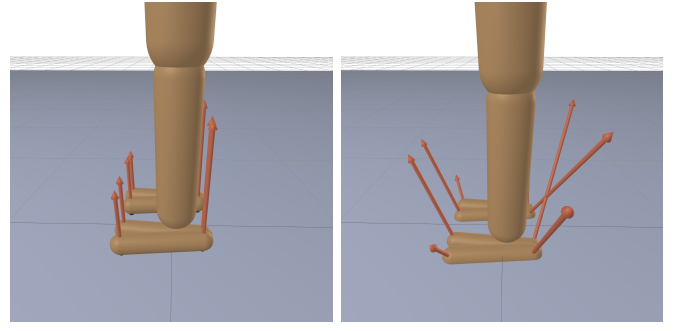


Fig. 2. **Compliant contacts.** Our contact solver can handle both purely rigid (left) and compliant contacts (right).

We now detail the content of each sub-computations of (36). The first equation (36a) corresponds to the solving of an unconstrained quadratic problem, whose solution is given by:

$$\mathbf{f}_k = -(G + R + (\eta + \rho)\text{Id})^{-1} (g + s_k - \eta \mathbf{f}_{k-1} - \rho \mathbf{y}_{k-1} - \mathbf{z}_{k-1}). \quad (37)$$

It is worth noting that (37) is always well-defined because of the proximal regularization. This allows our ADMM approach to handle both compliant and purely rigid contacts as illustrated in Fig.2. The second equation (36b) can be explicitly written as:

$$\mathbf{y}_k = \arg \min_{\mathbf{y} \in \mathcal{K}_\mu} \frac{\rho}{2} \left\| \mathbf{f}_k - \frac{\mathbf{z}_{k-1}}{\rho} - \mathbf{y} \right\|_2^2. \quad (38)$$

It corresponds to the orthogonal projection of the vector $\mathbf{f}_k - \frac{\mathbf{z}_{k-1}}{\rho}$ on the Cartesian product of friction cones \mathcal{K}_μ , also denoted by:

$$\mathbf{y}_k = \mathcal{P}_{\mathcal{K}_\mu}^{\text{Id}} \left(\mathbf{f}_k - \frac{\mathbf{z}_{k-1}}{\rho} \right) \quad (39)$$

The third equation (36c) corresponds to the classic augmented Lagrangian multiplier updates [5].

D. Primal and dual convergence criteria

As classically done in the ADMM settings, the primal and dual residuals associated with the augmented Lagrangian function $\mathcal{L}_{\rho, \eta}^A$ at the k^{th} iterates are respectively given by

$$\mathbf{r}_k^{\text{prim}} = \mathbf{f}_k - \mathbf{y}_k, \quad (40)$$

and

$$\mathbf{r}_k^{\text{dual}} = \eta(\mathbf{f}_k - \mathbf{f}_{k-1}) + \rho(\mathbf{y}_k - \mathbf{y}_{k-1}). \quad (41)$$

In addition, we use a contact complementarity residual defined as

$$\mathbf{r}_k^{\text{comp}} = \left(|\mathbf{f}_k^{(1)\top} \mathbf{z}_k^{(1)}| \quad \dots \quad |\mathbf{f}_k^{(n_c)\top} \mathbf{z}_k^{(n_c)}| \right). \quad (42)$$

The iterations of ADMM have converged to precision ϵ_{abs} when:

$$\|\mathbf{r}_k^{\text{prim}}\|_\infty, \|\mathbf{r}_k^{\text{dual}}\|_\infty, \|\mathbf{r}_k^{\text{comp}}\|_\infty \leq \epsilon_{\text{abs}} \quad (43)$$

A typical set of values for ϵ_{abs} which offers a good compromise between realistic simulation (compared to

alternative resolution methods) and computation times to solve the NCP problems is $[10^{-4}; 10^{-6}]$. We choose to use the infinity norm ($\|\cdot\|_\infty$) as it is independent of the problem dimensions. As explained in [45], it is also possible to introduce relative convergence criterion, as done in many optimization solvers of the literature [43, 4], to account for the potential stagnation of the optimization variables due to the numerics.

E. Exploiting problem sparsity

Over the three main steps of the ADMM recursion (36), the last two (36b) and (36c) are cheap operations of linear complexity w.r.t. the problem dimensions. The most complex operation lies in the resolution of the linear system in Eq. (36a), and detailed in Eq. 37. It notably requires the inversion of the augmented Delassus matrix, denoted by $G_{\rho,\eta}^R := G + R + (\eta + \rho)\text{Id}$ in the sequel. Interestingly, the additional terms $R + (\eta + \rho)\text{Id}$ only modify the diagonal of the Delassus matrix G with positive elements. In other words, both the sparsity of the Delassus matrix and the positivity are preserved, allowing to directly call sparse or dense Cholesky methods to decompose the augmented Delassus $G_{\rho,\eta}^R$.

Additionally, in the case of kinematics algorithms composed of multiple joints including loop-closure, one can leverage branch-inducing sparsity algorithms [22] to efficiently evaluate the Delassus matrix [21, 47] or directly obtain its Cholesky decomposition at a reduced cost [10]. Finally, it is worth noticing that, as soon as the ADMM penalty term ρ is updated, it requires the full Cholesky refactorization of $G_{\rho,\eta}^R$. In other words, to lower the computational footprint induced by the successive Cholesky factorizations, it is essential to lower the number of updates of ρ . This motivates the introduction of the spectral update rule for ρ , detailed in the next subsection, which is experimentally validated in Sec. V.

F. ADMM parameters update strategies

The convergence rate of the ADMM methods is directly related to the value of the augmented penalty term ρ [45]. Choosing this value directly depends on input problem values (G, R, g and \mathcal{K}_μ). There is no automatic procedure to choose the best ρ which will lower the number of iterations to reach the desired primal/dual accuracy ϵ_{abs} .

Linear update rule. From a given initial value ρ , a well-known strategy is to linearly update ρ according to the ratio between primal and dual residuals $r_k^{\text{prim}}/r_k^{\text{dual}}$, following the update rule at the k^{th} iterate

$$\rho^{\text{new}} = \begin{cases} \tau^{\text{inc}} \rho & \text{if } \|r_k^{\text{prim}}\|_\infty \geq \alpha \|r_k^{\text{dual}}\|_\infty \\ \rho / \tau^{\text{dec}} & \text{if } \|r_k^{\text{dual}}\|_\infty \geq \alpha \|r_k^{\text{prim}}\|_\infty \\ \rho, & \text{otherwise,} \end{cases} \quad (44)$$

where $\tau^{\text{inc}} > 1$ and $\tau^{\text{dec}} > 1$ are increment/decrement factors, and $\alpha > 1$ is the ratio parameter between primal and dual residuals. The overall idea is to maintain the trajectory of primal and dual residual norms within a tube of diameter α . Yet, if the problem is poorly conditioned, this linear update

rule will often trigger many updates of ρ , thus requiring each time to recompute the Cholesky factorization associated with the nonsingular augmented Delassus matrix $G_{\rho,\eta}^R$.

Spectral update rule. To overcome the inherent limitations of the standard linear ADMM update rule, we introduce a new update strategy that accounts for the spectral properties of the augmented Delassus matrix $G_{\rho,\eta}^R$. More precisely, our approach is inspired by the work of Nishihara et al. [42] which provides a convergence analysis of a generic class of ADMM formulations, including ours depicted by (32). Their analysis assumes that the ADMM penalty parameter ρ is of the form

$$\rho := \sqrt{mL} \kappa^p, \quad (45)$$

where m is the strong convexity parameter of h_1 and L is the Lipschitz constant associated with ∇h_1 , where h_1 refers to the smooth convex function in the ADMM formulation (32). As h_1 is a quadratic function, m and L respectively correspond to the lowest and largest eigenvalues of $G_{\rho,\eta}^R$. The ratio $\kappa := L/m$ is the condition number of h_1 and $G_{\rho,\eta}^R$. p is the free exponent parameter directly balancing the contribution of the condition number κ in the choice of ρ .

While in [42] p is assumed to be constant, we suggest adjusting its value according to the ratio between primal and dual residuals $\|r_k^{\text{prim}}\|_\infty / \|r_k^{\text{dual}}\|_\infty$, in the spirit of the linear update rule recalled previously. More precisely, we propose this selection strategy

$$p^{\text{new}} = \begin{cases} p + p^{\text{inc}} & \text{if } \|r_k^{\text{prim}}\|_\infty \geq \alpha \|r_k^{\text{dual}}\|_\infty, \\ p - p^{\text{dec}} & \text{if } \|r_k^{\text{dual}}\|_\infty \geq \alpha \|r_k^{\text{prim}}\|_\infty, \\ p, & \text{otherwise,} \end{cases} \quad (46)$$

$$\rho^{\text{new}} = \sqrt{mL} \kappa^{p^{\text{new}}},$$

where p^{inc} and p^{dec} are increments on the exponent parameter p . A typical value is $p^{\text{inc}} = p^{\text{dec}} = 0.05$. As for the linear update rule, $\alpha > 1$ is the ratio parameter between primal and dual residuals, forcing the primal and dual residual norms to lie within a tube of diameter α . To the best of the author's knowledge, this spectral update strategy is novel and directly scales the ADMM update parameter according to the smoothness of the NCP problem.

It is worth noticing that the proposed solution only considers the lowest and largest eigenvalues of $G_{\rho,\eta}^R$, which can be easily estimated from the power iteration algorithm for instance, which, in practice, converges in very few iterations compared to the problem dimensions. Finally, thanks to the presence of the proximal term added in (35), we have $m \geq \eta > 0$, which guarantees the well-posedness of the strategy.

G. Pseudocode

Algorithm 1 summarizes our ADMM-based approach for solving the NCP (27) problem of frictional contacts simulation. It takes as inputs the contact problem parameters, such as the Delassus matrix G , the free contact point velocities g , the Cartesian product of the friction cones \mathcal{K}_μ , the compliance matrix R , as well as a desired precision ϵ_{abs} . The outputs of

Alg. 1 at line 13 correspond to both the optimal contact forces λ and the contact point velocities σ . This last quantity can be directly obtained from the dual variable z of the NCP problem, which corresponds to the sum of the contact point velocity $\sigma = (G + R)\lambda + g$ and the DeSaxé corrective term $\Gamma(\sigma)$.

Algorithm 1: Pseudocode of the ADMM algorithm for NCP for rigid and compliant contacts.

Input: Delassus matrix G , free contact point velocities g , friction cones \mathcal{K}_μ , compliance R , desired precision ϵ_{abs} .

Output: Contact impulses λ and contact velocities σ

```

1 for  $k = 1$  to  $n_{\text{iter}}$  do
  /* ADMM updates */
2    $s_k \leftarrow \Gamma(z_{k-1})$ ;
3    $f_k \leftarrow -(G + R + (\eta + \rho)I_d)^{-1}$ 
4      $(g + s_k - \eta f_{k-1} - \rho y_{k-1} - z_{k-1})$ ;
5    $y_k \leftarrow \mathcal{P}_{\mathcal{K}_\mu}^{\text{Id}}\left(f_k - \frac{z_{k-1}}{\rho}\right)$ ;
6    $z_k \leftarrow z_{k-1} - \rho(f_k - y_k)$ ;
  /* Primal/dual criteria evaluation */
7   compute the primal/dual convergence criteria given
     by (40), (41) and the stopping criteria (43);
8   if converged then
9     | break;
10  end
  /* Update  $\rho$  */
11  Update  $\rho$  according to the spectral strategy
     described in Sec. III-F;
12 end
13  $\lambda \leftarrow y_k$  and  $\sigma \leftarrow z_k - \Gamma(z_k)$ ;

```

IV. INVERSE DYNAMICS

We now consider the inverse dynamics problem, corresponding to the search of the torque τ and the contact impulses λ that induce a given joint velocity v_{ref} . Starting from the NCP (27) formulation and enforcing the contact point velocities to be equal to $J_c v_{\text{ref}}$ yields the inverse dynamics problem,

$$\begin{aligned} \mathcal{K}_\mu \ni \lambda \perp \sigma + \Gamma(\sigma) \in \mathcal{K}_\mu^* \\ \sigma = R\lambda + J_c v_{\text{ref}} + \gamma. \end{aligned} \quad (47)$$

We recall that, from KKT conditions, a solution of (47) should minimize

$$\min_{\lambda \in \mathcal{K}_\mu} \frac{1}{2} \|\lambda + R^{-1}(J_c v_{\text{ref}} + \gamma + s)\|_R^2 \quad (48)$$

where $s = \Gamma(\sigma)$. This corresponds to the projection on \mathcal{K}_μ under the metric induced by R , so we note $\mathcal{P}_{\mathcal{K}_\mu}^R(-R^{-1}(J_c v_{\text{ref}} + \gamma + s))$ the minimizer of (48). We observe the formulation of (48) becomes ill-defined for the purely rigid case $R = 0$. Indeed, as described previously (Sec. II), the rigidity often makes the problem of contact

impulses under-determined and thus non-invertible. As was done in the case of forward dynamics (Sec. III), we aim to preserve the rigid contact hypothesis by leveraging proximal optimization. For the case of inverse dynamics, iterating the proximal operator associated to (48) (Alg. 2, line 3) allows to find a λ verifying (47) even in the rigid case.

At this stage, it is worth noting that (47) has the same structure as (27) and is also an NCP. Therefore, one could have used the algorithm introduced in Sec. III (Alg. 1) to solve it. However, one notable difference between (27) and (47) is the absence of the Delassus matrix in the latter, which has been replaced by the compliance matrix R . Most often, the compliance matrix has a diagonal structure of the form $R = \text{Diag}(R_T^{(1)}, R_T^{(1)}, R_N^{(1)}, \dots, R_T^{(n_c)}, R_T^{(n_c)}, R_N^{(n_c)})$ which can be exploited to design a more efficient algorithm. Indeed, in the case of a diagonal matrix $D \in \mathbb{R}^{3n_c \times 3n_c}$, the operator $\mathcal{P}_{\mathcal{K}_\mu}^D$ can be computed analytically via the equality $\mathcal{P}_{\mathcal{K}_\mu}^D(x) = D^{-\frac{1}{2}} \mathcal{P}_{\mathcal{K}_\mu}^{\text{Id}}(D^{\frac{1}{2}}x)$ with $\tilde{\mu} = \left(\sqrt{\frac{D_T^{(1)}}{D_N^{(1)}}} \mu^{(1)} \quad \dots \quad \sqrt{\frac{D_T^{(n_c)}}{D_N^{(n_c)}}} \mu^{(n_c)} \right)$ which makes the iterations of Alg. 2 computationally cheap.

Once contact forces are determined, the torque τ can be retrieved with a call to the Recursive Newton-Euler Algorithm (RNEA) (Alg. 2, line 5). It is worth noticing that the contribution of the contact torque $\tau_c = J_c^\top \lambda$ can be accounted for through the backward sweep of the RNEA algorithm, thus efficiently exploiting the sparsity induced by the kinematic tree, in the spirit of rigid-body dynamics algorithms [22].

Algorithm 2: Pseudo-code of the proximal algorithm for Inverse Dynamics.

Input: Joint velocity v_{ref} , compliance R , contact Jacobian J_c , friction cones \mathcal{K}_μ

Output: Torque τ , contact impulses λ

```

1 for  $k = 1$  to  $n_{\text{iter}}$  do
2    $s \leftarrow \Gamma(R\lambda + J_c v_{\text{ref}} + \gamma)$ ;
3    $\lambda \leftarrow \mathcal{P}_{\mathcal{K}_\mu}^{R+\rho \text{Id}}(-(R + \rho \text{Id})^{-1}(J_c v_{\text{ref}} + \gamma + s - \rho \lambda))$ ;
4 end
5  $\tau \leftarrow \text{RNEA}(q, v, v_{\text{ref}}) - J_c^\top \lambda$ ;

```

V. EVALUATIONS AND BENCHMARKS

In this section, we evaluate our approach to challenging robotics scenarios and benchmarks from the computational mechanics community. In particular, we measure performances in terms of the number of iterations required to converge and computational timings. We also validate our inverse dynamics algorithm by performing a control task involving contact interactions with a UR5 arm. Additional results are available in the companion video (https://youtu.be/i_qg9cTx0NY?si=NGtx1tiYrIGtHXSK).

A. Simulation benchmarks: standard robotics systems

Our solver has been implemented in C++, leveraging the Eigen library [23] for efficient linear algebra, the Pinocchio

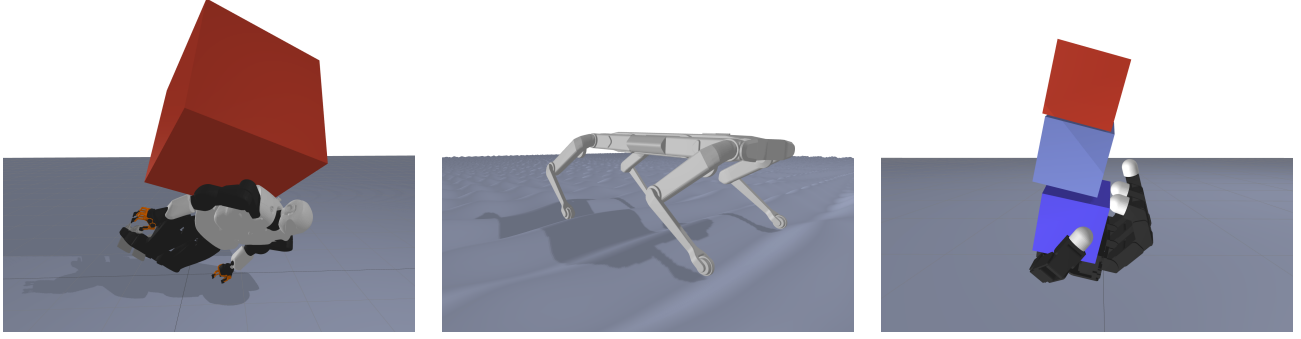


Fig. 3. **Robotics systems.** We evaluate our approach on common robotics scenarios occurring during locomotion, e.g., Talos falling on the ground (**left**), Solo walking on steep terrain (**center**), and manipulation, e.g., Allegro hand interacting with cubes (**right**).

framework [8, 9] for efficient rigid body algorithms, which comes with the HPP-FCL library [44] for fast collision detection. As baselines, we have also implemented the over-relaxed PGS algorithm [31] and used the state-of-the-art SCS solver [43] to solve the CCP relaxation. Our experiments are done on a MacBook Pro with a M1 Max CPU. For all the benchmarks, we use a time step Δt of 1ms and set the proximal value η to 10^{-6} .

Convergence analysis. We evaluate the convergence speed of our algorithm by monitoring the evolution of the convergence criteria $\|r_k^{\text{dual}}\|_\infty$ and $\|r_k^{\text{comp}}\|_\infty$ across iterations ($\|r_k^{\text{prim}}\|_\infty$ being null due to projection steps). The study is done on two different contact problems. The first one is a stack of rigid boxes of different masses (between 1 and 10^4 kg) hit by a ball of 10^3 kg (Fig. 1). Stacking objects of high mass ratios induces bad conditioning of the Delassus matrix G (these ratios are of 10 for the Talos scene and 10^4 for the wall of cubes). Therefore, this scene allows us to evaluate the numerical stability of the solvers. For the second one, we study a problem obtained with the humanoid robot Talos [50] falling on the ground, whose 36-dof kinematic chain induces a complex inertial coupling between all the contact points through the Delassus matrix. Additionally, we use these two scenarios to perform an ablation study on the benefits of using the spectral update rule for the ADMM parameter adaption.

As shown by Fig. 4, the per-contact strategy of PGS loops; this undesired behavior is explained by the fact that these problems involve high mass ratios and strongly coupled contact forces. On the opposite, our ADMM-based algorithm exploits the Cholesky decomposition of $G_{\rho,\eta}^R$ which allows it to be insensitive to the conditioning and to capture the coupling between individual contact problems. Figure 4 also demonstrates how leveraging the spectral information to update adapt ρ leads to an improved convergence rate w.r.t. the linear update rule. Therefore, our approach can efficiently solve the two problems at a high precision threshold $\epsilon_{\text{abs}} = 10^{-9}$.

Timings. When evaluating timings, we cover a range of contact types that aim to be wide enough to represent robotics

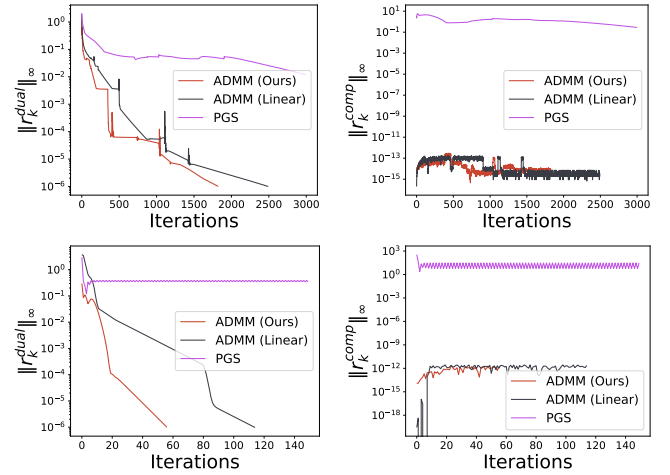


Fig. 4. **Convergence analysis.** Convergence of the dual feasibility (**left**) and complementarity (**right**) is monitored across iterations on the contact problems obtained from simulating a stack of cubes (**top row**) and the Talos robot (**bottom row**).

applications. To do so, we consider four distinct scenarios (Fig. 1,3). For the first two, we reuse the setups of the convergence analysis: a stack of boxes of different masses (around 60 contact points) and a humanoid falling on the ground (around 10 contact points). As a third experiment, we simulate an Allegro hand holding a stack of cubes, a classical setup in manipulation applications. Finally, we evaluate our approach on a more dynamic task: a quadruped moving on a steep terrain (Fig. 3). The walking motion is generated via an MPC controller, which produces reactive behavior and results in a wide range of contact types, notably many breaking and sliding contacts.

For each benchmark, we average on a trajectory the computational time necessary to reach a fixed precision ($\epsilon_{\text{abs}} = 10^{-4}$) or to hit the maximum number of iterations ($n_{\text{iter}} = 10^3$). Figure 5 summarizes our evaluation of timings on robotics systems. We observe that our algorithm performs consistently well in different scenarios compared to the PGS and SCS algorithms, even when they operate on the relaxed CCP problem. In particular, we note a significant performance

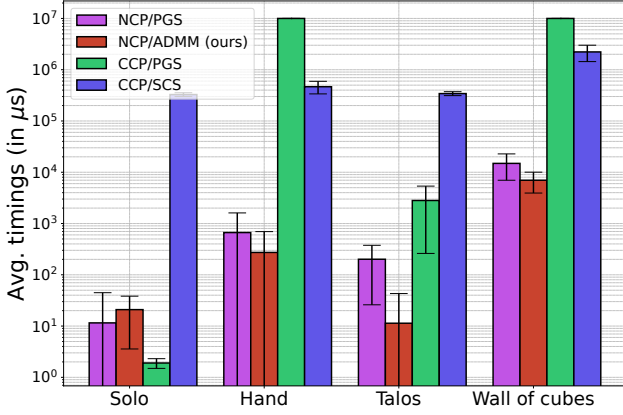


Fig. 5. **Timings on robotics systems.** Thanks to its combination of automatic spectral adaptation (Sec.III-F) and sparse Cholesky update (Sec. III-E), our algorithm (in red) yields a stable behavior across the various robot simulations.

gap between ADMM and PGS when the complexity of the structure of the inertia grows, as is the case for the hand, the humanoid, and the stack of cubes. As shown during the convergence analysis, the complex coupling induced by G can slow down per-contact approaches and even hinder convergence. It is also worth noting that our algorithm does not require any hyperparameter tuning across the various considered scenarios. Indeed, ρ is automatically scaled via the spectral rule, which induces a reduced computational overhead when combined with rigid body algorithms for the Cholesky updates.

B. Comparison against state-of-the-art physics engines

We implemented our algorithm in a C++ simulation loop, building on top of Pinocchio [8] for rigid body dynamics and HPP-FCL [44] for collision detection. We evaluate our approach against the state-of-the-art simulators MuJoCo, Drake, and Bullet on three scenarios of increasing complexity in terms of degrees of freedom: a UR5 robotic arm, a Cassie robot, and the MuJoCo humanoid. For each benchmark, we simulate the robot for 2 seconds without any actuation or damping in the joints. For all simulators, joint limits are not taken into account. The MuJoCo Humanoid is made of capsules for collisions, while the UR5 and Cassie robots use convex meshes (which can reach up to 1000 vertices per body). All possible pairs of bodies are considered for collision detection, except the successive pairs in the kinematic chain. Performance is measured by the number of time steps per second on a single thread of an Apple M1 Max CPU. The results are reported in Fig. 6. The ADMM solver embedded in a simulation loop depicts competitive results compared to alternative solutions in the robotics literature.

It is worth noticing that this is a preliminary benchmark. A more detailed one (involving additional metrics, more physical systems, and comparisons to real data) would benefit the robotics community but would require a dedicated study. We leave this benchmarking study as future work.

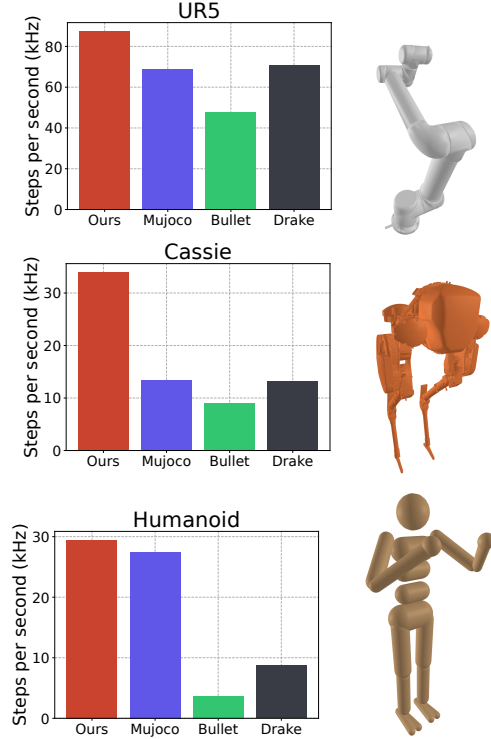


Fig. 6. **Comparison against SOTA physics simulators.** Performances are reported for a UR5 arm (top), and a Cassie robot (middle), the MuJoCo simple humanoid (bottom).

C. Complex scenarios of computational mechanics

In this section, we leverage the collection of complex contact problems provided in the FCLIB benchmark collections [1]. These problems correspond to a set of various mechanical problems identified as challenging by the computational mechanics community. We focus on three main categories of problems: *BoxesStack*, *Chain*, and *Capsules*. The size and properties of this dataset are reported in Tab. II.

Timings. We benchmark our solver against PGS on the three aforementioned scenarios. We request a precision of $\epsilon_{\text{abs}} = 10^{-6}$ on both solvers. Due to the size of the problems, we exploit the sparse Cholesky solver coming with Eigen to account for the sparsity of the contact problems.

We use performance-profile distributions [17], a metric to fairly compare optimization solvers in the optimization community. More precisely, for a given solver, it measures the ratio of problems solved which are τ times slower than the best solver for a given problem. We refer to [17] for further details. The performance-profile distributions are reported in Fig. 7 for the three categories. These plots show that our solver significantly outperforms PGS on the complex contact problems collected in the FCLIB dataset. On problems from *BoxesStack* and *Capsules*, we observe that PGS is not able to converge to the desired accuracy and reach the maximum number of iterations n_{iter} set to 20,000.

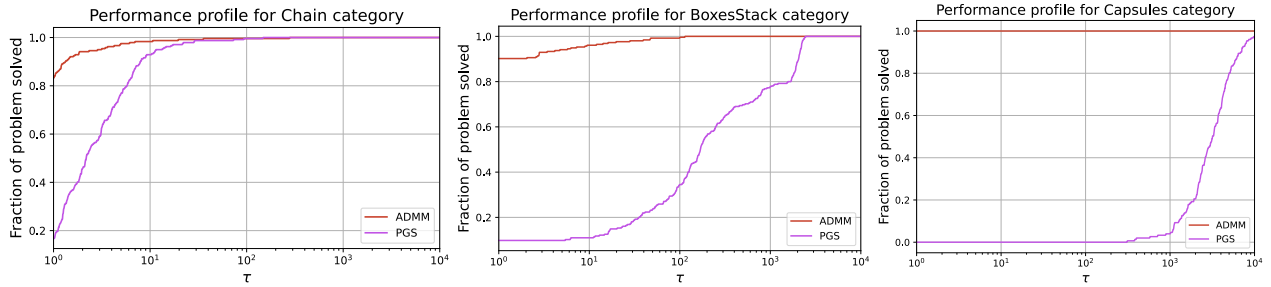


Fig. 7. **Performance profiles** curves on a subset of the FCLIB dataset.

Category	# Problems	n_c	Dofs	friction μ
<i>BoxesStack</i>	255	[0:200]	[6:300]	0.7
<i>Chain</i>	242	[8:28]	[48:60]	0.3
<i>Capsules</i>	249	[0:200]	[6:300]	0.7

TABLE II
PROBLEMS OF THE FCLIB DATASET [1].

Empirical evaluation of the spectral and linear ADMM updates. In section III-F, we have motivated the introduction of the spectral update strategy as a way to reduce the number of Cholesky factorization updates involved in the ADMM algorithm 1, which is the most computationally demanding part of the ADMM solver. We evaluate this assertion by evaluating the mean number and the standard deviation of Cholesky updates over the *BoxesStack*, *Chain*, and *Capsules* problem categories. We notably vary the linear parameter τ and spectral parameter p .

The results are reported in Tab. III. Depending on the problem category, the linear update strategy tends to produce higher standard deviations and mean numbers of Cholesky updates than the spectral rule. The spectral rule provides the most consistent statistics with lower updates and is less sensitive to the class of problems to solve, with an average of a dozen Cholesky updates. We also report in Tab. IV the mean timings to solve a given problem. These timings are consistent with the mean number of Cholesky updates reported in Tab. III, in the sense that best timings are obtained for the spectral update rule depicting the lowest number of Cholesky updates.

D. Inverse dynamics

We evaluate our approach for inverse dynamics on the control problem consisting of finding the torque to slide the end effector of a robotic arm (UR5) on a wall where the contact is assumed to be purely rigid $R = 0$ (Fig. 8). The reference joint velocity \mathbf{v}_{ref} is such that the contact point velocity is $\mathbf{c}_{\text{ref}} = \mathbf{J}_c \mathbf{v}_{\text{ref}}$ is tangent to the wall, i.e., $\mathbf{c}_{\text{ref}N} = 0$.

When setting ρ to 10^{-8} , the approach proposed in Sec. IV requires only one iteration to find a solution with a precision of $\epsilon_{\text{abs}} = 10^{-6}$. One should note that the proximal regularization is necessary due to the rigid contact hypothesis. On Fig. 8, we evaluate the benefits of incorporating the iterative De Saxcé

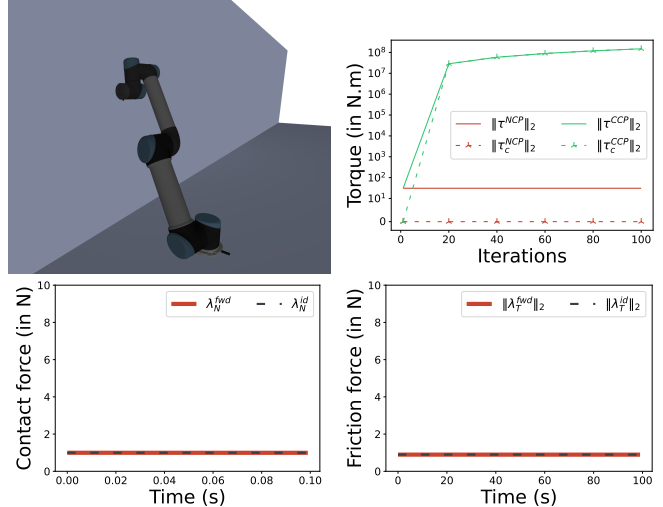


Fig. 8. **Inverse dynamics** is used to slide the end-effector of a UR5 on a wall (**top left**). The desired tangential contact velocity is infeasible with the CCP model which causes divergence of both the joint (τ) and contact (τ_c) torques, while our approach robustly converges for the NCP formulation (**top right**). The normal (**bottom left**) and tangential (**bottom right**) contact forces computed by the inverse dynamics algorithm are consistent with the forward dynamics solution.

correction (Alg. 2, line 2) and work directly with the NCP (27) model. When setting this corrective term to 0 which exactly corresponds to the CCP contact model, we observe that the contact torque τ_c and thus the actuation torque τ diverge. Indeed, for a sliding motion, $\mathbf{c}_{\text{ref}} \notin \mathcal{K}_\mu^*$ which causes the desired motion to be infeasible in the sense of CCP.

VI. DISCUSSION

The closest to our work are the approaches of MuJoCo [54] and Drake [11], which tend to solve linear systems coupling all the contact points together, by inverting a matrix containing the Delassus contact matrix. Unlike per-contact approaches [49, 14], we exploit the Cholesky decomposition of the Delassus matrix to jointly update contact forces, which improves robustness. As explained in Sec. III, such Cholesky decomposition can be efficiently computed by leveraging the sparsity pattern induced by the kinematic trees of robots composing the scene. Moreover, we alleviate the computational footprint by reducing the number of these Cholesky re-factorizations via a spectral adaptation of

TABLE III
COMPARISON OF THE NUMBER OF CHOLESKY UPDATES BETWEEN LINEAR AND SPECTRAL UPDATE STRATEGIES

Category	Linear				Spectral		
	$\tau = 2$	$\tau = 4$	$\tau = 8$	$\tau = 16$	$p = 0.01$	$p = 0.05$	$p = 0.08$
<i>BoxesStack</i>	9.20 ± 2.48	6.12 ± 1.75	5.85 ± 2.47	6.12 ± 2.40	11.26 ± 4.99	5.02 ± 2.40	3.97 ± 1.76
<i>Chain</i>	5.74 ± 1.53	8.76 ± 69.93	33.7 ± 212.77	25.48 ± 154.88	7.87 ± 6.55	2.76 ± 1.65	3.61 ± 20.48
<i>Capsules</i>	4.58 ± 1.71	3.09 ± 2.01	3.60 ± 2.94	4.6 ± 2.06	5.57 ± 4.85	2.30 ± 1.50	2.12 ± 1.01

TABLE IV
COMPARISON OF THE MEAN TIMINGS, IN MS, BETWEEN LINEAR AND SPECTRAL UPDATE STRATEGIES

Category	Linear				Spectral		
	$\tau = 2$	$\tau = 4$	$\tau = 8$	$\tau = 16$	$p = 0.01$	$p = 0.05$	$p = 0.08$
<i>BoxesStack</i>	2.16 ± 1.80	1.73 ± 1.55	1.76 ± 1.80	1.95 ± 1.87	2.56 ± 2.03	1.55 ± 1.60	1.51 ± 1.76
<i>Chain</i>	0.527 ± 0.417	0.637 ± 0.373	1.61 ± 7.84	1.34 ± 6.82	0.445 ± 0.53	0.321 ± 0.304	0.494 ± 0.301
<i>Capsules</i>	2.86 ± 1.90	2.50 ± 1.92	1.95 ± 1.35	1.69 ± 1.20	2.17 ± 1.38	1.72 ± 1.09	1.63 ± 1.10

the ADMM parameter. In terms of physical realism, our improvement is twofold: we do not relax the *Signorini condition*, i.e., our simulator does not exhibit forces at distance, and we make it possible to simulate purely rigid systems ($R = 0$). Optionally, by setting De Saxcé corrective term to zero, we robustly solve the convex formulation of these simulators and retrieve convergence guarantees. For future work, further experiments should be conducted to evaluate to what extent avoiding simulation artifacts related to relaxations [25, 33] tightens the reality gap of simulators [20, 36] and how a reduced gap impacts downstream tasks.

Our approach also influences the formulation of the inverse dynamics. As shown by Todorov [54], in the case of MuJoCo and Drake, the convex relaxation and the compliance make the dynamics invertible. Because we also encompass rigid contacts, we cannot assume the mapping between joint velocities and contact forces to be uniquely defined (e.g., hyperstatic scenarios). Using a proximal algorithm allows handling ill-defined cases (i.e., nonunique solution) and converges toward one possible solution of the inverse problem on forces. Moreover, via the iterative DeSaxcé correction, we preserve the NCP formulation. This enables us to invert the dynamics of motions that were previously infeasible in the sense of the CCP, for instance for sliding motions.

In a parallel line of research, a recent growing effort has been made to port classical simulators such as MuJoCo and PhysX to hardware accelerators e.g. GPUs and TPUs, which resulted in MuJoCo XLA and Isaac Gym [38]. These architectures provide high parallelization capabilities, but they impose hard constraints on the design of contact models and algorithms. The approach introduced in this paper focuses on exploiting the versatility and efficiency of modern CPUs to achieve physically accurate simulation at competitive rates. However, it seems promising for future work to adapt it in order to leverage hardware accelerators.

VII. CONCLUSION

In this paper, we have introduced an ADMM-based algorithm to solve the NCP associated with the simulation

of dynamics involving rigid frictional contacts. We have evaluated our approach to challenging benchmarks from both the robotics and computational mechanics communities. Our rich set of experiments demonstrates that we can robustly simulate a wide range of scenarios while keeping a limited computational burden and avoiding physical relaxation.

The current approach could still be improved by gaining timings on the collision detection routine corresponding to the bottleneck. Similarly, rigid-body dynamics algorithms for constrained dynamical systems still represent an active area of research whose improvements would directly affect contact solvers and, thus, physics simulation in robotics. Although we did not observe cases causing our algorithm to diverge, working towards theoretical convergence guarantees could be an interesting research direction. Moreover, our algorithm for inverse dynamics could be generalized to account for the underactuation and the unfeasible reference accelerations it can induce.

Finally, we believe these promising results are a further step towards more computationally efficient and physically consistent simulators which, due to their centrality, could positively impact the overall robotics community and related fields where efficient and reliable simulation matters.

ACKNOWLEDGEMENTS

This work was supported in part by the French government under the management of Agence Nationale de la Recherche (ANR) as part of the "Investissements d'avenir" program, references ANR-19-P3IA-0001 (PRAIRIE 3IA Institute) and ANR-22-CE33-0007 (INEXACT), the European project AGIMUS (Grant 101070165), the Louis Vuitton ENS Chair on Artificial Intelligence and the Casino ENS Chair on Algorithmic and Machine Learning. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

- [1] Vincent Acary, Maurice Brémond, Tomasz Koziara, and Franck Pérignon. FCLIB: a collection of discrete 3D

- Frictional Contact problems. Technical Report RT-0444, INRIA, February 2014. URL <https://inria.hal.science/hal-00945820>.
- [2] Vincent Acary, Maurice Brémond, and Olivier Huber. On solving contact problems with Coulomb friction: formulations and numerical comparisons. Research Report RR-9118, INRIA, November 2017. URL <https://hal.inria.fr/hal-01630836>.
 - [3] Brian Acosta, William Yang, and Michael Posa. Validating robotics simulators on real-world impacts. *IEEE Robotics and Automation Letters*, 7(3):6471–6478, 2022.
 - [4] Antoine Bambade, Sarah El-Kazdadi, Adrien Taylor, and Justin Carpentier. Prox-qp: Yet another quadratic programming solver for robotics and beyond. In *RSS 2022-Robotics: Science and Systems*, 2022.
 - [5] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
 - [6] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
 - [7] Herman Bruyninckx and Oussama Khatib. Gauss’ principle and the dynamics of redundant and constrained manipulators. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, pages 2563–2568. IEEE, 2000.
 - [8] Justin Carpentier, Florian Valenza, Nicolas Mansard, et al. Pinocchio: fast forward and inverse dynamics for poly-articulated systems. <https://stack-of-tasks.github.io/pinocchio/>, 2015–2021.
 - [9] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiriaux, Olivier Stasse, and Nicolas Mansard. The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*, 2019.
 - [10] Justin Carpentier, Rohan Budhiraja, and Nicolas Mansard. Proximal and sparse resolution of constrained dynamic equations. In *Proc. Robot., Sci. Syst.*, 2021.
 - [11] Alejandro M Castro, Frank N Permenter, and Xuchen Han. An unconstrained convex formulation of compliant contact. *IEEE Transactions on Robotics*, 39(2):1301–1320, 2022.
 - [12] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022.
 - [13] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem*. Society for Industrial and Applied Mathematics, 2009. doi: 10.1137/1.9780898719000. URL <https://epubs.siam.org/doi/abs/10.1137/1.9780898719000>.
 - [14] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>, 2016–2021.
 - [15] Géry 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*, 28(4-8):225–245, August 1998. doi: 10.1016/S0895-7177(98)00119-8. URL <https://hal.archives-ouvertes.fr/hal-03883288>.
 - [16] Étienne Delassus. Mémoire sur la théorie des liaisons finies unilatérales. In *Annales scientifiques de l’École normale supérieure*, volume 34, pages 95–179, 1917.
 - [17] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91:201–213, 2002.
 - [18] Alexander Domahidi, Eric Chu, and Stephen Boyd. Ecos: An socp solver for embedded systems. In *2013 European control conference (ECC)*, pages 3071–3076. IEEE, 2013.
 - [19] Christer Ericson. *Real-time collision detection*. Crc Press, 2004.
 - [20] Nima Fazeli, Elliott Donlon, Evan Drumwright, and Alberto Rodriguez. Empirical evaluation of common contact models for planar impact. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3418–3425. IEEE, 2017.
 - [21] Roy Featherstone. Exploiting sparsity in operational-space dynamics. *The International Journal of Robotics Research*, 29(10):1353–1368, 2010.
 - [22] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.
 - [23] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
 - [24] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5977–5984. IEEE, 2023.
 - [25] Peter C Horak and Jeff C Trinkle. On the similarities and differences among contact models in robot simulation. *IEEE Robotics and Automation Letters*, 4(2):493–499, 2019.
 - [26] Taylor A Howell, Simon Le Cleac’h, J Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable simulator for robotics. *arXiv preprint arXiv:2203.00806*, 9, 2022.
 - [27] Jemin Hwangbo, Joonho Lee, and Marco Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018. URL www.raisim.com.
 - [28] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.
 - [29] Toyota Research Institute. Scooping (sim vs reality side-by-side). <https://youtu.be/5aVDWjWd0EU?si=R511Doxi7Xh-V07W>.

- [30] Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. DTC: Deep Tracking Control. *Science Robotics*, 9(86):eadh5401, 2024.
- [31] Franck Jourdan, Pierre Alart, and Michel Jean. A Gauss-Seidel like algorithm to solve frictional contact problems. *Computer methods in applied mechanics and engineering*, 155(1-2):31–47, 1998.
- [32] Jonas Koenemann, Andrea Del Prete, Yuval Tassa, Emanuel Todorov, Olivier Stasse, Maren Bennewitz, and Nicolas Mansard. Whole-body model-predictive control applied to the hrp-2 humanoid. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3346–3351. IEEE, 2015.
- [33] Quentin Le Lidec, Wilson Jallet, Louis Montaut, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Contact models in robotics: a comparative analysis.
- [34] Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):500, Feb 2018. doi: 10.21105/joss.00500. URL <https://doi.org/10.21105/joss.00500>.
- [35] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.
- [36] Daolin Ma, Elliott Donlon, Siyuan Dong, and Alberto Rodriguez. Dense tactile force estimation using gelslim and inverse fem. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5418–5424. IEEE, 2019.
- [37] Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapon Chentanez, Stefan Jeschke, and Matthias Müller. Small steps in physics simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366779. doi: 10.1145/3309486.3340247. URL <https://doi.org/10.1145/3309486.3340247>.
- [38] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.
- [39] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.
- [40] Jean Jacques Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255:2897–2899, 1962.
- [41] Jean Jacques Moreau. Unilateral Contact and Dry Friction in Finite Freedom Dynamics. In Moreau J.J. and Panagiotopoulos P.D., editors, *Nonsmooth Mechanics and Applications*, volume 302 of *International Centre for Mechanical Sciences (Courses and Lectures)*, pages 1–82. Springer, 1988. doi: 10.1007/978-3-7091-2624-0_1. URL <https://hal.archives-ouvertes.fr/hal-01713847>.
- [42] Robert Nishihara, Laurent Lessard, Ben Recht, Andrew Packard, and Michael Jordan. A general analysis of the convergence of ADMM. In *International conference on machine learning*, pages 343–352. PMLR, 2015.
- [43] Brendan O'Donoghue. Operator splitting for a homogeneous embedding of the linear complementarity problem. *SIAM Journal on Optimization*, 31:1999–2023, August 2021.
- [44] Jia Pan, Sachin Chitta, Dinesh Manocha, Florent Lamiriaux, Joseph Mirabel, Justin Carpentier, et al. HPP-FCL: an extension of the Flexible Collision Library. <https://github.com/humanoid-path-planner/hpp-fcl>, 2015–2022.
- [45] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.
- [46] Junuthula Narasimha Reddy. An introduction to the finite element method. *New York*, 27:14, 1993.
- [47] Ajay Suresha Sathya, Wilm Decre, and Jan Swevers. Pv-osimr: A lowest order complexity algorithm for computing the delassus matrix, 2023.
- [48] Antonio Signorini. Questioni di elasticità non linearizzata e semilinearizzata. *Rendiconti di Matematica e delle sue applicazioni*, 18(5):95–139, 1959.
- [49] Russell Smith. Open dynamics engine, 2008. URL <http://www.ode.org/>. <http://www.ode.org/>.
- [50] Olivier Stasse, Thomas Flayols, Rohan Budhiraja, Kevin Giraud-Esclasse, Justin Carpentier, Joseph Mirabel, Andrea Del Prete, Philippe Souères, Nicolas Mansard, Florent Lamiriaux, et al. Talos: A new humanoid research platform targeted for industrial applications. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 689–695. IEEE, 2017.
- [51] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. Osqp: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020.
- [52] Alessandro Tasora, Dario Mangoni, Simone Benatti, and Rinaldo 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*, 122(16):4093–4113, 2021.
- [53] Russ Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019. URL <https://drake.mit.edu>.
- [54] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[55] Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. Stable constrained dynamics. *ACM*

Transactions on Graphics (TOG), 34(4):1–10, 2015.