Differentiable simulation for physical system identification

Quentin Le Lidec[1], Igor Kalevatykh[1], Ivan Laptev[1], Cordelia Schmid[1] and Justin Carpentier[1]

***Abstract*—Simulating frictional contacts remains a challenging research topic in robotics. Recently, differentiable physics emerged and has proven to be a key element in model-based Reinforcement Learning (RL) and optimal control fields. However, most of the current formulations deploy coarse approximations of the underlying physical principles. Indeed, the classic simulators loose precision by casting the Nonlinear Complementarity Problem (NCP) of frictional contact into a Linear Complementarity Problem (LCP) to simplify computations. Moreover, such methods deploy non-smooth operations and cannot be automatically differentiated. In this paper, we propose (i) an extension of the staggered projections algorithm for more accurate solutions of the problem of contacts with friction. Based on this formulation, we introduce (ii) a differentiable simulator and an efficient way to compute the analytical derivatives of the involved optimization problems. Finally, (iii) we validate the proposed framework with a set of experiments to present a possible application of our differentiable simulator. In particular, using our approach we demonstrate accurate estimation of friction coefficients and object masses both in synthetic and real experiments.**

***Index Terms*—Contact Modeling, Simulation and Animation, Optimization and Optimal Control, Calibration and Identification**

I. Introduction

PHYSICAL simulation, as it allows for both training and testing control policies, appears to be a key element in robotics. Rigid Body Algorithms [1] provide an efficient way to compute the forward dynamics of multi-body rigid systems when there is no frictional contact. It is also possible to differentiate the quantities simulated with these algorithms with respect to the state and the control variables of the system. Using analytical derivatives (instead of Automatic Differentiation or finite differences) allows for efficient computation [2]. Differentiable physics has proven to be very useful for gradient-based algorithms for optimal control and trajectory optimization [3], [4], [5], [6], [7]. However, the case of simulation with frictional contacts remains a challenging problem for the control community [8].

In the same vein, simulation of frictional contacts is a crucial point when training Reinforcement Learning (RL) agents to achieve complex control tasks involving contact interactions. Indeed, RL is a powerful tool to learn control policies but often requires millions of samples generated in simulation, which is the reason why the simulator has to be efficient. Moreover, the learned policies tend to exploit the artifacts of the simulators due to approximations of the underlying physics which leads to unrealistic motions that are difficult to transfer to real systems. This mismatch between reality and simulation, known as the reality gap [9], highly limits the ability to transfer simulation-learnt policies to real robots. Hence, simulators should be both fast and accurate.

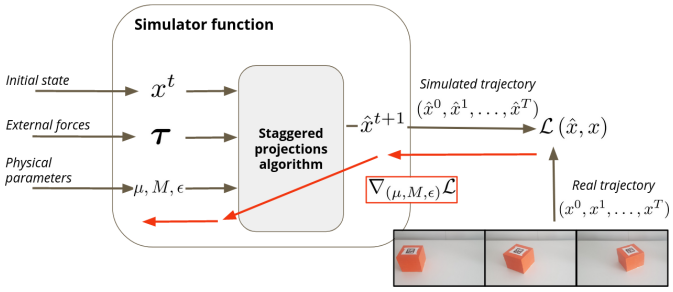


Fig. 1: Overview of our differentiable simulator. The differentiability of the simulator allows to integrate it into a larger learning architecture and infer physical parameters such as friction coefficients μ and mass M of the objects, from real trajectories of these objects.

Modeling frictional contacts is one of the most challenging aspect of physical simulations given the non-linearity and non-convexity of complementarity constraint and the maximum dissipation principle. These underlying physical laws of rigid contact dynamics are typically simplified (spring-damper [10]), approximated [11] or relaxed [12] in classic physics engines. These choices aim to increase computational efficiency but may also result in non-realistic behaviors in simulation [13].

Simulating a system requires accurate values of its physical parameters, such as masses and friction coefficients of objects. Given the difficulty of estimating these parameters, however, their values are often randomized [14]. As result, such an approach often leads to the imprecise simulation.

In this paper, we propose an approach that guarantees the differentiability of the simulator and also avoids error-prone approximations of complementarity constraints and the maximum dissipation principle. To this end, we extend the staggered projections algorithm [15] to deal with the friction cone constraint. In addition, we use techniques from the field of sensitivity analysis to differentiate the result of the simulation with respect to the physical parameters. This allows us to design a process to infer unknown physical parameters

Manuscript received: October 15, 2020; Revised: December 6, 2020; Accepted: January 31, 2021.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments.

[1]Inria, Ecole normale suprieure, CNRS, PSL Research University, 75005 Paris, France. *corresponding author: quentin.le-lidec@inria.fr*

Digital Object Identifier (DOI): 10.1109/LRA.2021.3062323

of a system that are essential for simulations but complex to measure in practice (such as friction coefficients), directly from real data.

The core contributions of this work are as follows:

1) We extend the work in [15] by formulating the frictional contact problem as a sequence of Quadratically Constrained Quadratic Programming (QCQP) problems without approximating any of the underlying physics principles and taking elastic collisions into account.
2) We propose computation of analytical derivatives of the solution of a QCQP as well as the efficient and robust implementation of the solver and its derivatives.
3) We demonstrate applications of our differentiable simulation to system identification by inferring physical properties of objects from videos of dynamical scenes.

This paper is organized as follows. Section II proposes an overview of the work done in the area of differentiable simulation and physical system identification. In III-A, we introduce the mathematical framework of the problem of frictional contacts and solve it by extending the staggered projection algorithm. In III-B, we expose the analytical derivatives of a QCQP which allow us to derive a differentiable and accurate simulator. In Section IV, we validate our method by applying it to the task of physical system identification and discuss the issue of parameter observability. This leads us to present some future research directions in Section V.

II. Related Work

Physical simulation algorithms. The problem of contacts without frictions can be formulated as a Linear Complementarity Problem (LCP) [16] and can be solved for instance using the Projected Gauss-Seidel (PGS) algorithm. This formulation can be adapted to the frictional case by approximating the friction cone with a four sided pyramid [17] as done in Bullet [11] or [8]. The algorithm of staggered projections [15] introduces a formulation of the frictional contacts problem as a fix point of coupled projections. By also using the pyramidal approximation of Coulomb's law, this method achieves simulating a system after solving a cascade of Quadratic Programming (QP) problems. Some others approaches [18] relax the complementarity constraint in order to transform the frictional contact problem into a single and simple optimization one [12]. However, this relaxation may lead to physically implausible behaviors such as object interactions without objects being in contact [13]. In this paper, we extend the formulation in [15] to laws of multiple elastic collisions [19]. In addition, we adapt it to account for conic constraints (conic friction constraint represented as ice-cream cones). This makes it possible to write explicitly the problem of frictional contact as a sequence of optimization problems, where the problems become QCQP problems. The use of intermediate QCQP problems enables us to exploit techniques from sensitivity analysis to differentiate the solution of the problem by back-propagating the solution over the cascade of convex problems.

Differentiable optimization and differentiable physics engine. Given that the solution of the frictional contact problem is a solution of a sequence of optimization problems, its differentiation requires derivatives of the solution of an optimization problem with respect to its parameters. In the case of an unconstrained optimization problem, a solution introduced in [20] consists in replacing the argmin operator by an approximation with an optimization procedure such as gradient descent. In this case, the number of gradient steps is fixed and each step represents an operation into the computational graph of the layer. Then, the gradient descent can be unrolled to compute the gradient with respect to the parameters of the optimization problem. However, this technique can lead to large computational graphs when the number of required gradient steps is important, increasing the computational cost when performing the backpropagation. Moreover, it is not possible to proceed this way when considering a constrained optimization problem because the optimization procedures often require projection steps which cannot be differentiated. Thus, implicit argmin differentiation which relies on the differentiation of optimality conditions appears to be a way to deal with constrained problems [21]. Although the implementations of this approach allows to solve very general constrained optimization problems and get the derivatives of the solution, they also lose efficiency in the process. More specialized solvers [22] use an equivalent implicit approach while taking advantages of the structure of the problem they are solving to gain efficiency. Simulators like [8] adapted this solver to be able to solve the LCP problem resulting from the approximation of the friction cone, to build a differentiable simulator. Although our work is closely related to [8], we avoid making any approximation by exploiting our extension of the formulation from [15] and the chain rule to differentiate the output of our simulator by differentiating through a sequence of optimization problems.

Generative physics model for system identification. The field of system identification [23] intends to build a mathematical model of a dynamical system from its measurements. The related work [8], [24], [25], [26] identifies parameters of physical systems using simulators as generative models. In each case, the identification is done by simulating the physical system and then optimizing the physical parameters so that the simulations are fitting to the real scenes. In this work, we adopt an approach close to [8], by relying on the differentiation of our physical model to estimate its physical parameters. However, because we avoid some of the approximations made in [8], we are able to consider not only 2D but also 3D systems. This allows us to apply our approach to the concrete task of inferring physical parameters from videos [24].

III. Differentiable Simulation

In this section, we show how the staggered projection algorithm [15] can be adapted to handle both the friction cone and elastic collisions. Then, we introduce the analytical derivatives of the QCQP problem appearing in this formulation, and propose a robust implementation that leads to a differentiable simulator.

A. Solving the frictional contact problem

Simulating a physical system corresponds to computing the next system state $(\boldsymbol{q}^{t+1}, \boldsymbol{v}^{t+1})$ and the current contact

forces $\boldsymbol{\lambda}$, given the initial state $(\boldsymbol{q}^t, \boldsymbol{v}^t)$, where $\boldsymbol{q} \in \mathbb{R}^{n_q}$ and $\boldsymbol{v} \in \mathbb{R}^{n_v}$ are the vectors of generalized position and velocities, n_c being the number of contact points[1]. To compute these quantities, our method relies on three main physical laws that we are going to introduce: the Euler-Lagrange equation of motion, the complementarity constraint between contact normal accelerations and forces, and the Maximum Dissipation Principle (MDP) from Coulomb's law of friction. From the classical Lagrangian dynamics, we get the following generalized equations of motion in continuous time for an unconstrained system:

$$M\boldsymbol{a} = \boldsymbol{\tau}(\mathbf{q}, \boldsymbol{v})$$

where $M \in \mathbb{R}^{n_v \times n_v}$ is the inertia matrix of the system, $\boldsymbol{a} \in \mathbb{R}^{n_v}$ is the generalized acceleration and $\boldsymbol{\tau}$ the vector of generalized forces which contains Coriolis and centrifugal effects, actuation, gravity and external forces. Moreover, when the dynamical system interacts with other objects, an additional term $J^T\boldsymbol{\lambda}$ has to be gathered to represent the effect of contact interaction forces:

$$M\boldsymbol{a} = \boldsymbol{\tau}(\mathbf{q}, \boldsymbol{v}) + J^T\boldsymbol{\lambda} \tag{1}$$

where $J \in \mathbb{R}^{3n_c \times n_v}$ is the Jacobian of the contact points and $\boldsymbol{\lambda}$ contains both the normal forces and the tangential friction forces. Following the approach proposed by [15] we consider separately the normal $\boldsymbol{\lambda}_n^{t+1} \in \mathbb{R}^{n_c}$ and tangential $\boldsymbol{\lambda}_t^{t+1} \in \mathbb{R}^{2n_c}$ components of $\boldsymbol{\lambda}$. We denote by $J_n \in \mathbb{R}^{n_c \times n_v}$ and by $J_t \in \mathbb{R}^{2n_c \times n_v}$ the projections of J on the normal and tangent directions of contacts respectively. Thus, after discretizing (1) with a time step Δt, we obtain:

$$M\left(\boldsymbol{v}^{t+1} - \boldsymbol{v}^t\right) = \Delta t\ \boldsymbol{\tau}(\mathbf{q}^t, \boldsymbol{v}^t) + J_t^T\boldsymbol{\lambda}_t^{t+1} + J_n^T\boldsymbol{\lambda}_n^{t+1}. \tag{2}$$

In this paper, we prefer to exploit this velocity-based formulation to be able to deal with discontinuities appearing during collisions, as we will see later. This is why we will now talk about impulses instead of forces when evoking the contact interaction quantity $\boldsymbol{\lambda}$.

Integrating the complementary constraint that (i) rigid bodies can not interpenetrate each other while (ii) contact impulses can act only to separate them when they are in contact, leads to the complementarity constraint:

$$0 \leq J_n\boldsymbol{v}^{t+1} \perp \boldsymbol{\lambda}_n^{t+1} \geq 0$$

Considering the law for multiple collision points [19] leads to the slightly modified constraint[2]:

$$0 \leq J_n\left(\boldsymbol{v}^{t+1} + \epsilon\boldsymbol{v}^t\right) \perp \boldsymbol{\lambda}_n^{t+1} \geq 0 \tag{3}$$

where ϵ is the coefficient of restitution quantifying the elasticity of the impact (when $\epsilon = 1$ the impact occurs with full restitution while $\epsilon = 0$ is a completely inelastic impact).

To model frictional contacts, we adopt Coulomb's law of friction. It imposes that the contact impulse $\boldsymbol{\lambda}$ lies into a cone whose tightness is determined by the coefficient of friction μ. At this stage, it is worth noting that μ takes two different values depending on if the object in contact are static (μ_{stat}) or in relative motion (μ_{kin}) and $\mu_{\text{stat}} \geq \mu_{\text{kin}}$. This constraint combined with MDP gives:

$$\boldsymbol{\lambda}_t^{t+1} = \underset{\boldsymbol{\lambda}_t\ s.t.\ \|\boldsymbol{\lambda}_{t(i)}\|_2 \leq \mu_i\boldsymbol{\lambda}_{n(i)}^{t+1}}{\operatorname{argmax}} \left(-\left(J_t^T\boldsymbol{\lambda}_t\right)^T\boldsymbol{v}^{t+1}\right) \tag{4}$$

where $\boldsymbol{\lambda}_{n,t(i)}$ corresponds to the contact impulses of the i$^{\text{th}}$ contact point. We note that the MDP (4) actually corresponds to the dual of the least action principle, which guarantees it to remain valid even at stiction.

Let A be a convex set, we denote by:

$$P_A(x) = \underset{z \in A}{\operatorname{argmin}} \frac{1}{2}(x - z)M^{-1}(x - z)$$

the operator of projection on the set A under the metric induced by the inertia matrix M. We also note respectfully the sets $C = \{J_n^T\boldsymbol{\lambda}_n,\ \lambda_n \geq 0\}$ and $F_{(\boldsymbol{\lambda}_n)} = \{J_t^T\boldsymbol{\lambda}_t,\ \forall i\ \|\boldsymbol{\lambda}_{t(i)}\|_2 \leq \mu_i\boldsymbol{\lambda}_{n(i)}\}$, the sets of admissible normal and tangential contact impulses. For the following, we also note $\boldsymbol{v}^p \in \mathbb{R}^{n_v}$ the contact-free velocity which verifies $M\left(\boldsymbol{v}^p - \boldsymbol{v}^t\right) = \Delta t\ \boldsymbol{\tau}(\mathbf{q}^t, \boldsymbol{v}^t)$.

Finally, (2), (3) and (4) correspond to the three physical principles we consider to simulate our system and compute the three unknowns $\boldsymbol{v}^{t+1}$, $\boldsymbol{\lambda}_t^{t+1}$ and $\boldsymbol{\lambda}_n^{t+1}$. As demonstrated in [15], the $\boldsymbol{\lambda}_t^{t+1}$ and $\boldsymbol{\lambda}_n^{t+1}$ solving these three equations equivalently verify the following staggered projections :

$$\begin{aligned} P_C(-M\left(\boldsymbol{v}^p + \epsilon\boldsymbol{v}^t\right) - J_t^T\boldsymbol{\lambda}_t^{t+1}) &= J_n^T\boldsymbol{\lambda}_n^{t+1} \\ P_{F(\boldsymbol{\lambda}_n^{t+1})}(-M\boldsymbol{v}^p - J_n^T\boldsymbol{\lambda}_n^{t+1}) &= J_t^T\boldsymbol{\lambda}_t^{t+1} \end{aligned}$$

and expanding the P_C and $P_{F(\boldsymbol{\lambda}_n^{t+1})}$ operators leads to the interdependant QP and QCQP:

$$\begin{aligned} \boldsymbol{\lambda}_n^{t+1} &= \underset{\lambda \geq 0}{\operatorname{argmin}}\ \frac{1}{2}\lambda^T G_n \lambda + \lambda^T g_n \\ \boldsymbol{\lambda}_t^{t+1} &= \underset{\|\lambda_{t(i)}\|_2 \leq \mu_i\boldsymbol{\lambda}_{n(i)}^{t+1}}{\operatorname{argmin}}\ \frac{1}{2}\lambda^T G_t \lambda + \lambda^T g_t \end{aligned} \tag{5}$$

where:

$$\begin{aligned} G_n &= J_n M^{-1} {J_n}^T,\ g_n = J_n\left(\boldsymbol{v}^p + \epsilon\boldsymbol{v}^t\right) + G_{nt}\boldsymbol{\lambda}_t^{t+1} \\ G_t &= J_t M^{-1} {J_t}^T,\ g_t = J_t\boldsymbol{v}^p + {G_{nt}}^T\boldsymbol{\lambda}_n^{t+1} \end{aligned}$$

with $G_{nt} = J_n M^{-1} J_t^T$. The formulation of (5) naturally induces a fix point algorithm to solve for $\boldsymbol{\lambda}_t^{t+1}$ and $\boldsymbol{\lambda}_n^{t+1}$. Finally, by fixing the number of fix point iterations to n_{step} (a convergence analysis similar to [15] finds $n_{\text{step}} \in [3, 10]$ to have reasonable computation time and a precision sufficient for most of applications) , $\boldsymbol{\lambda}_t^{t+1}$ and $\boldsymbol{\lambda}_n^{t+1}$ can be computed by solving a sequence of optimization problems alternating between a QP and a QCQP (at lines 8 and 10 of Algo. 1). To compute the argmin operation, we avoid classic Primal Dual Interior Point Method solvers and rely on a regularized ADMM algorithm. As we detail it in Appendix V-A, this allows us to solve QCQPs in a way requiring as much as computation as QPs, but also to deal with over-constrained situations making the problems (5) ill conditioned while requiring only tens of iterations to converge.

[1]Here, we also considered that the configuration vector and its related velocity vector may have different dimensions.

[2]In the same way, Baumgarte's stabilization can be used to avoid the point "drift" issue, with $0 \leq J_n\left(\boldsymbol{v}^{t+1} + \epsilon\boldsymbol{v}^t + \boldsymbol{v}^B\right) \perp \boldsymbol{\lambda}_n^{t+1} \geq 0$ where $J_n\boldsymbol{v}^B = -Ke$ and e the penetration error.

Algorithm 1: Extended staggered projection

Input: Initial state, physical parameters and external forces: $\boldsymbol{v}^t, M, J, \mu, \boldsymbol{\tau}(\mathbf{q}^t, \boldsymbol{v}^t)$

Output: Next state: $\boldsymbol{v}^{t+1}$

$\boldsymbol{v}^p \leftarrow \boldsymbol{v}^t + M^{-1}\Delta t \boldsymbol{\tau}(\mathbf{q}^t, \boldsymbol{v}^t)$;
$\boldsymbol{\lambda}_t \leftarrow \boldsymbol{\lambda}_t^t$;
$G_n \leftarrow J_n M^{-1} J_n^T$; $G_t \leftarrow J_t M^{-1} J_t^T$;
$G_{nt} \leftarrow J_n M^{-1} J_t^T$;
for $i \leftarrow 0$ **to** n_{step} **do**
  $g_n \leftarrow J_n\left(\boldsymbol{v}^p + \epsilon \boldsymbol{v}^t\right) + G_{nt}\boldsymbol{\lambda}_t$;
  $\boldsymbol{\lambda}_n \leftarrow \underset{\lambda \geq 0}{\text{argmin}}\ \frac{1}{2}\lambda^T G_n \lambda + \lambda^T g_n$;
  $g_t \leftarrow J_t \boldsymbol{v}^p + {G_{nt}}^T \boldsymbol{\lambda_n}$;
  $\boldsymbol{\lambda}_t \leftarrow \underset{\|\lambda_{t(i)}\|_2 \leq \mu \boldsymbol{\lambda}_{n(i)}}{\text{argmin}}\ \frac{1}{2}\lambda^T G_t \lambda + \lambda^T g_t$;
end
$\boldsymbol{\lambda}_t^{t+1} \leftarrow \boldsymbol{\lambda}_t$; $\boldsymbol{\lambda}_n^{t+1} \leftarrow \boldsymbol{\lambda}_n$;
$\boldsymbol{v}^{t+1} \leftarrow \boldsymbol{v}^p + M^{-1}\left(J_t^T \boldsymbol{\lambda}_t^{t+1} + J_n^T \boldsymbol{\lambda}_n^{t+1}\right)$

We can already notice that using the chain rule allows to differentiate the computed velocity $\boldsymbol{v}^{t+1}$ and contact impulses $\boldsymbol{\lambda}_t^{t+1}$ and $\boldsymbol{\lambda}_n^{t+1}$ as long as we know how to differentiate the successive argmin operators.

B. Differentiating the solution

As explained in III-A, differentiating the outputs $\boldsymbol{v}^{t+1}$, $\boldsymbol{\lambda}_t^{t+1}$, $\boldsymbol{\lambda}_n^{t+1}$ of the simulation requires to compute the derivatives of the solution of the QCQP and QP problems that are involved in the algorithm 1. Thus, in the same way as it is done in [22], we implemented the function solving the particular case of QCQP (we do not detail the QP case as it is already studied in [22]) appearing during the step of projection onto $F_{(\boldsymbol{\lambda_n})}$. This function can be written as:

$$z_{i+1} = \underset{\|z_{t(i)}\|_2 \leq \mu(z_i)\|\lambda_{n(i)}(z_i)\|_2}{\text{argmin}} \frac{1}{2} z^T G(z_i) z + g(z_i)^T z \quad (6)$$

where z_i is an input variable (μ, J, M, $\boldsymbol{\lambda}_n$, $\boldsymbol{\tau}$, $\boldsymbol{v}^t$ in our case) parameterizing the QCQP and $z_{i+1} \in \mathbb{R}^{2n_c}$ its solution. Using the implicit differentiation approach [27], we implemented the analytical derivatives that allow to compute $\frac{\partial z_{i+1}}{\partial z_i}$ which is necessary when performing a backward pass. The Karush-Kuhn-Tucker optimality conditions of the QCQP can be written:

$$\begin{aligned} &\text{Diag}(\|z_{i+1 t(i)}\|_2^2 - \mu_i^2 \lambda_{n(i)}^2)\gamma = 0 \\ &G z_{i+1} + g + 2\text{Diag}\left(\Gamma\gamma\right) z_{i+1} = 0 \end{aligned}$$

where $\gamma \in \mathbb{R}^{n_c}$ corresponds to the dual solution associated to the constraints and where $\Gamma \in \mathbb{R}^{2n_c \times n_c}$:

$$\Gamma = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & \dots & 0 & 1 \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix}$$

Then, it is possible to differentiate these equations to get the following system where the unknowns are the variations of the primal and dual solutions $\mathrm{d}z_{i+1}$ and $\mathrm{d}\gamma$:

$$\Delta \begin{pmatrix} \mathrm{d}\gamma \\ \mathrm{d}z_{i+1} \end{pmatrix} = \delta \quad (7)$$

where:

$$\Delta = \begin{pmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{pmatrix},\ \delta = \begin{pmatrix} \delta_1 \mathrm{d}\mu + \delta_2 \mathrm{d}\lambda_n \\ -\mathrm{d}G z_{i+1} - \mathrm{d}g \end{pmatrix}$$

and:

$$\begin{aligned} \Delta_{11} &= \text{Diag}(\|z_{i+1 t(i)}\|_2^2 - \mu_i^2 \lambda_{n(i)}^2), \\ \Delta_{12} &= 2\text{Diag}(\gamma)\Gamma^T \text{Diag}(z_{i+1}), \\ \Delta_{21} &= 2\text{Diag}(z_{i+1})\Gamma, \\ \Delta_{22} &= G + 2\text{Diag}\left(\Gamma\gamma\right), \\ \delta_1 &= 2\text{Diag}(\gamma_i \mu_i \lambda_{n(i)}^2),\ \delta_2 = 2\text{Diag}(\gamma_i \mu_i^2 \lambda_{n(i)}) \end{aligned}$$

Solving (7) allows to compute the derivatives $\mathrm{d}z_{i+1}$ of the solution of our QCQP with respect to G, g, μ and λ_n. It is important to notice that those derivatives remain true as long as the matrix Δ is invertible. For instance, when all constraints are inactive and G has an high condition number, it is not possible to invert Δ. In this case we use iterative refinement as it is introduced in [28], to solve the system. This allows to solve approximately systems like $Ax = b$ even when A is ill-conditioned. To do so, we intend to solve the problem $\min_x \frac{1}{2}\|Ax - b\|_2^2$, but the solution of this problem requires to compute the pseudo-inverse of A by applying a shift to the original problem to regularize it. Instead, iterative refinement uses an iterative process defined by : $x^{k+1} = \underset{x}{\text{argmin}}\ \frac{1}{2}\|Ax - b\|_2^2 + \frac{\rho}{2}\|x - x^k\|_2^2$ which converges to the solution of the least squares problem and only requires the computation of a regularized pseudo-inverse.

However, in many cases such as in IV-B, we do not want to compute the variations of the primal and dual variables $\mathrm{d}z_{i+1}$, $\mathrm{d}\gamma$ but rather the gradient of a loss $\mathcal{L}$ formed with z_{i+1} that we are minimizing with respect to the parameters z_i. As done in [22], we proceed by directly computing the product with the previous backward pass vector $\frac{\partial \mathcal{L}}{\partial z_{i+1}}$ as follows:

$$\begin{aligned} \mathrm{d}\mathcal{L} &= \frac{\partial \mathcal{L}}{\partial z_{i+1}} \mathrm{d}z_{i+1} \\ &= \begin{pmatrix} 0 \\ \left(\frac{\partial \mathcal{L}}{\partial z_{i+1}}\right)^T \end{pmatrix}^T \begin{pmatrix} \mathrm{d}\gamma \\ \mathrm{d}z_{i+1} \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{b}_\gamma \\ \mathrm{b}_{z_{i+1}} \end{pmatrix}^T \delta \end{aligned}$$

where $\begin{pmatrix} \mathrm{b}_\gamma \\ \mathrm{b}_{z_{i+1}} \end{pmatrix} = \Delta^{-T} \begin{pmatrix} 0 \\ \left(\frac{\partial \mathcal{L}}{\partial z_{i+1}}\right)^T \end{pmatrix}$. Using the expression of δ, we get:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda_n} &= \delta_2^T \mathrm{b}_\gamma,\ \frac{\partial \mathcal{L}}{\partial \mu} = \delta_1^T \mathrm{b}_\gamma, \\ \frac{\partial \mathcal{L}}{\partial g} &= -\mathrm{b}_{z_{i+1}},\ \frac{\partial \mathcal{L}}{\partial G} = -\mathrm{b}_{z_{i+1}} z_{i+1}^T \end{aligned}$$

Fig. 2: Experimental setup to determine the physical properties of the cube. We considered two different scenes : in the first one, the unknown cube is sliding on the floor, starting with a given initial velocity $\boldsymbol{v}^0$; in the second setup, the same cube collides with a second cube whose characteristics are known.

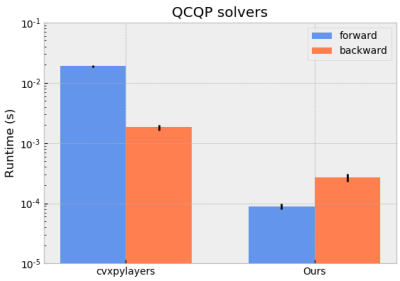


Fig. 3: Comparisons of runtime performances between [21] and ours, on randomly generated QCQPs of the form (6)

And finally, the gradient we are interested in $\frac{\partial \mathcal{L}}{\partial z_i}$ is obtained with the chain rule:

$$\frac{\partial \mathcal{L}}{\partial z_i} = \frac{\partial \mathcal{L}}{\partial \lambda_n}\frac{\partial \lambda_n}{\partial z_i} + \frac{\partial \mathcal{L}}{\partial \mu}\frac{\partial \mu}{\partial z_i} + \frac{\partial \mathcal{L}}{\partial g}\frac{\partial g}{\partial z_i} + \frac{\partial \mathcal{L}}{\partial G}\frac{\partial G}{\partial z_i}.$$

Eventually, we observe on Fig. 3, that on our particular QCQP/QP problems, our solver is efficient and robust during both the forward and backward passes. Indeed, the regularized ADMM of V-A and the iterative refinement respectively allow to solve ill-conditioned problems and compute their derivatives. This point is determinant as it makes it possible to deal with the case of G only being positive semi-definite, which is occurring often in robotics when G is Delassus' matrix for over-constrained systems. The code of our solver is publicly available at https://github.com/quentinll/diffqcqp.

IV. Experiments

In this section, we show through experiments how the differentiability of our simulator can be exploited to retrieve the physical parameters of a system from its trajectories. In addition, we show that it is possible only under the condition that the trajectory contains enough information to avoid any ambiguity, which leads us to some experiments on the observability issue. A video illustrating our work is available at https://youtu.be/d248IWMLW9o.

A. Experimental setup

In our experiments, we intend to estimate the physical parameters of a cube from simulated and real dynamical scenes. The second part of experiments involves another cube whose properties were known (Fig. 2).

For our simulator, we used the Pinocchio library [29], [30] for the implementation of the rigid body algorithms from [1] and for collisions detection algorithm, and PyTorch [31] for the implementation of backward Automatic Differentiation.

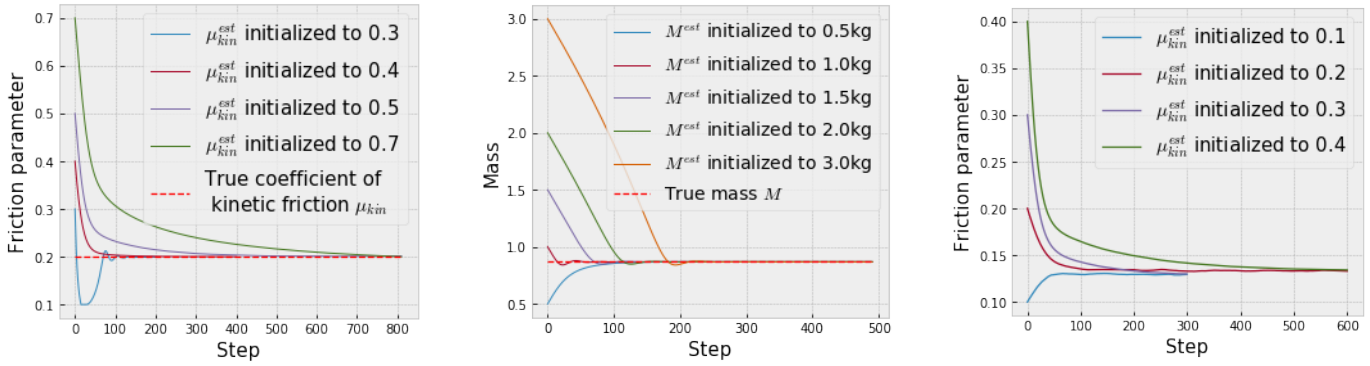
B. Physical parameters inference from trajectories

To exhibit the new ability of our simulator we will consider the scenario of an object interacting with the floor (a cube sliding on the floor but it could be a more complicated scenario like a walking robot), where every parameters (the inertias M, external forces $\boldsymbol{\tau}$) are known except for the coefficient of kinetic friction μ_{kin} of the object with the floor (we could do the same with others parameters). We will also consider that we dispose of a trajectory $(x^0, x^1, \dots, x^T)$ of this object interacting with the floor, where $x = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{a})$. Here, we cover two cases: either (i) x is generated in simulation so we know precisely the ground truth parameters of the system (Fig. 4a,4b,5,6), or (ii) the trajectory x is extracted with a pose estimation algorithm from videos of real experiments (Fig. 4c). It is worth noting that the simulated trajectories were generated with an algorithm (PGS-NCP from [13]) and a time step different from the ones of the differentiable simulator used for the inference, and that we also added white noise (variance of 10^{-3}m) to make sure that results from simulations do not depend on the way trajectories are simulated.

We note the "simulator function" $g_\mu : x^t \mapsto \hat{x}^{t+1}$ whose computational graph corresponds to the Algorithm 1 and whose only unknown parameter is μ. Then, we can defined the MSE loss:

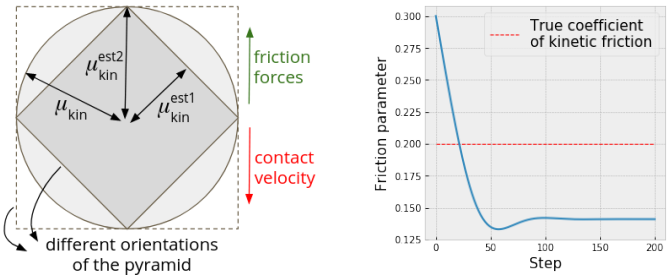
$$\mathcal{L}(\mu) = \sum_{t=1}^{T} \|x^t - \hat{x}^t\|_2^2 = \sum_{t=1}^{T} \|x^t - g_\mu(x^{t-1})\|_2^2$$

which is the sum of the errors made by the simulator at each time step. Using the differentiability of the "simulator function" g_μ with respect to μ, it is possible to compute $\nabla_\mu \mathcal{L}$ by backpropagating the loss using the Automatic Differentiation tool of PyTorch [31], as illustrated on Fig. 1. Then, we minimize $\mathcal{L}$ with respect to μ using Adam algorithm [32] .



(a) (b) (c)

Fig. 4: Results of the inference process of physical parameters from simulated (Fig. 4a,4b) or real (Fig. 4c) trajectories. When the inference is done from simulated trajectories, our method always converges to the ground truth value. When trajectories come from real experiments (Fig. 4c), the ground truth value of μ_{kin} is not available but we observe that our system converges to $\mu_{\text{kin}} = 0.13$ for every initialization and this value is consistent with the tables [33] and the coefficient of static friction we measured $\mu_{\text{stat}} = 0.18$.



(a) (b)

Fig. 5: Limitations of simulators approximating the friction cone with a pyramid [8],[11],[24],[26]. On Fig. 5a, the darker pyramid corresponds to the worst case, when the pyramid is rotated with an angle $\pi/4$ with respect to the contact velocity. As predicted, Fig. 5b shows that the inferred value $\mu_{\text{kin}}^{\text{est1}}$ converges towards $0.141 = \mu_{\text{kin}}/\sqrt{2}$.

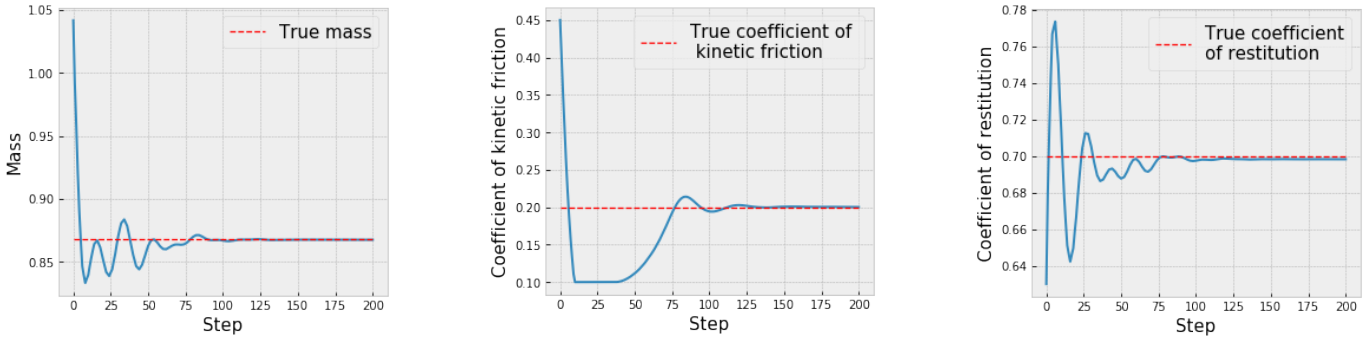
Proceeding this way allows to retrieve the coefficient of kinetic friction μ_{kin} from both simulated (Fig. 4a) or real (Fig. 4c) trajectories when all others parameters are known. The same method makes it possible to also infer the mass of the cube M (Fig. 4b) or any other physical parameter (external forces, initial state, etc).

Moreover, Fig. 5 demonstrates why our choice of modeling the friction as an ice-cream cone (instead of a pyramid) is determinant to ensure the success of the inference. Indeed, when such a pyramidal approximation is made, the value of $\mu_{\text{kin}}^{\text{est}}$ depends on the choice of orientation between the axes of the pyramid and the contact point velocity, as illustrated by Fig. 5a. Thus in 3D, because pyramidal cones are not isotropic, a same friction coefficient value may lead to two different simulated trajectories when using formulations based on this kind of approximation [8],[11],[24],[26]. Similarly, in the context of friction estimation, the same observed motion may then lead to two different friction values depending on the orientation of the frictional pyramid (Fig. 5a,5b). This effect could be limited by approximating the cone by a polyhedron with more faces, which also comes at the cost of a larger computational time.

C. Parameters observability

In the context of inferring several physical parameters at the same time, we aim at minimizing $\mathcal{L}(\mu, M, \epsilon) = \sum_{t=1}^{T} \|x^t - g_{\mu,M,\epsilon}(x^{t-1})\|_2^2$. When optimizing the model parameters, our approach would allow to get one of the possible combinations of parameters (in the sense that several combinations of parameters may lead to the observed trajectories), but it may not be the true one. This limitation directly comes from the observability of the physical parameters. In the same way, for instance, it would be impossible to infer the friction coefficient of the cube with the floor if there was no contact between the cube and the floor in the given trajectory because, in this case, any value of μ_{kin} would be possible. Thus, the trajectory given for the inference process needs to make the desired parameters observable by excluding others possible values. This leads to other very interesting questions: How to generate a trajectory that allows to expose some particular properties of the system? Is it possible to infer physical characteristics of an object using information only coming from trajectories?

This last question refers to the case where the number of parameters we try to infer becomes important (which can happen when inferring shapes for instance) and leads to ambiguities that would require adding visual or material information to be solved. However, in this work, we only address the first question. We show that using more complex trajectories where the object whose characteristics are unknown is interacting with other known objects allows to avoid some of the possible ambiguities. We proceed by using a setup similar to the previous, except that the unknown cube collides with a known one during the experiment. We observe on Fig. 6 that this enables us to infer the mass of the cube M, together with its friction coefficient μ and the elasticity parameter ϵ at the same time. Although the collision introduce a new parameter



(a) (b) (c)

Fig. 6: The collision with an object, whose physical characteristics are known, allows to solve the ambiguity and retrieve the true parameters μ and M simultaneously, which was not possible previously. In the same time, we are also able to find the value of the coefficient of restitution involved in the collision.

ϵ, it does not make the parameters harder to observe because ϵ can be determined independently using $\epsilon = -v_{\text{rel}}^{+}/v_{\text{rel}}^{-}$.

D. Limitations of the approach

Even if they were not apparent during the previous experiments, we noticed two limitations to our framework that are inherent to the staggered projections algorithm [15]. Indeed, as shown in [15], the algorithm is not monotone, thus, it does not have theoretical convergence guarantees. Demonstrating possible guarantees for the staggered projections algorithm would be an interesting work to be done. In addition, our approach also requires to solve a cascade of optimization problems at each step which is why it is accurate, but it can also appear costly compared to algorithms linearizing the friction cone and solving only one LCP.

V. Discussion and Future Work

In this work, we extended the formulation proposed by Kaufman et al. of the frictional contact problem that allows to write the contact impulses as a solution of a sequence of convex optimization problems. Then we introduced the analytical derivatives of the various optimization problems that are involved in this formulation, and, proposed a simple but efficient implementation for these solvers and their analytical derivatives. We showed experimentally that our approach is able to infer physical parameters directly from videos of the evolution of interacting rigid body systems. Our experiments also allowed to demonstrate the importance of the observability issue when addressing this kind of a task. More generally, we believe that the efficiency and robustness of our differentiable simulator can lead to concrete applications involving real physical scenes and large amount of data, in particular in the context of robotic dexterous manipulation.

In future work, we intend to extend our framework to also include the inference of the position of contact points and the shape of objects directly from videos. Learning these quantities can require the introduction of many additional parameters, so we expect the observability issue to be central. In the present work, we only used videos to retrieve objects' trajectories, and we can expect that using additional advanced computer vision algorithms will provide precious information to solve this issue. Finally, exploiting the differentiable dynamics introduced in this paper in the frame of model-based control approaches (e.g. optimal control or model-based reinforcement learning) appears as another exciting research direction.

Acknowledgment

This work was supported in part by the HPC resources from GENCI-IDRIS(Grant AD011011342), the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and Louis Vuitton ENS Chair on Artificial Intelligence.

References

[1] R. Featherstone, *Rigid Body Dynamics Algorithms*, 01 2008.
[2] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and Systems*, 2018.
[3] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
[4] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *International Journal of Robotics Research*, 2014.
[5] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *IEEE Transactions on Robotics*, 2018.
[6] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, "Trajectory optimization with implicit hard contacts," *IEEE Robotics and Automation Letters*, 2018.
[7] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
[8] F. de Avila Belbute-Peres, K. A. Smith, K. R. Allen, J. B. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *NeurIPS*, 2018.
[9] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.
[10] A. M. Castro, A. Qu, N. Kuppuswamy, A. Alspach, and M. Sherman, "A transition-aware method for the simulation of compliant contact with regularized friction," *IEEE Robotics and Automation Letters*, Apr 2020.

[11] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[12] 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*, 2012.

[13] P. C. Horak and J. C. Trinkle, "On the similarities and differences among contact models in robot simulation," *IEEE Robotics and Automation Letters*, 2019.

[14] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017.

[15] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai, "Staggered projections for frictional contact in multibody systems," *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, no. 5, 2008.

[16] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The linear complementarity problem*. Siam, 1992.

[17] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *NONLINEAR DYNAMICS*, 1997.

[18] M. Anitescu and A. Tasora, "An iterative approach for cone complementarity problems for nonsmooth dynamics," *Computational Optimization and Applications*, vol. 47, 10 2010.

[19] T. Giang, G. Bradshaw, and C. OSullivan, "Complementarity based multiple point collision resolution," in *Proc. Fourth Irish Workshop on Computer Graphics*, 2003.

[20] J. Domke, "Generic methods for optimization-based modeling," in *Artificial Intelligence and Statistics*. PMLR, 2012.

[21] A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. Moursi, "Differentiating through a cone program," *Journal of Applied and Numerical Optimization*, vol. 1, no. 2, 2019.

[22] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*. PMLR, 2017.

[23] L. Ljung, *System Identification: Theory for the User*. Pearson Education, 1998.

[24] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[25] S. Purushwalkam, A. Gupta, D. M. Kaufman, and B. Russell, "Bounce and learn: Modeling scene dynamics with real-world bounces," 2019.

[26] C. Song and A. Boularias, "Learning to Slide Unknown Objects with Differentiable Physics Simulations," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020.

[27] A. Griewank and A. Walther, *Evaluating Derivatives*, 2nd ed. Society for Industrial and Applied Mathematics, 2008.

[28] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, Jan. 2014.

[29] J. Carpentier, F. Valenza, N. Mansard, *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," https://stack-of-tasks.github.io/pinocchio, 2015–2019.

[30] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] D. Atack and D. Tabor, "The friction of wood," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1958.

[34] 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, 01 2011.

[35] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. Jordan, "A general analysis of the convergence of admm," in *International Conference on Machine Learning*. PMLR, 2015.

APPENDIX

A. Projecting on the set of frictional impulses with ADMM

In order to improve the performances of our extension of the staggered projections algorithm we implemented with PyTorch [31] a solver for the specific QCQP problem appearing during the projection step on $F_{(\boldsymbol{\lambda}_n)}$. To solve this problem we used the ADMM algorithm from [34]. The problem can be re-written:

$$\min_{x,z} \quad f(x) + g(z) \quad s.t. \quad x = z$$

where $f(x) = \frac{1}{2}x^T P x + q^T x$ and $g(z) = \mathcal{I}_{\mathcal{C}}(z)$, with $\mathcal{I}_{\mathcal{C}}$ the characteristic function of $\mathcal{C} = \{z,\ \forall i\ \|z_{t(i)}\|_2 \leq \mu_i \lambda_{n(i)}\}$. Thus, the ADMM algorithm can be written:

$$\begin{aligned} x^{k+1} &= \underset{x}{\text{argmin}} \quad \mathcal{L}_\rho(x, z^k, y^k) \\ z^{k+1} &= \underset{z}{\text{argmin}} \quad \mathcal{L}_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho(x^{k+1} - z^{k+1}) \end{aligned}$$

where y is the dual variable of the problem and $\mathcal{L}_\rho(x, z, y) = f(x) + g(z) + y^T(x - z) + \frac{\rho}{2}||x - z||_2^2$ is the associated augmented Lagrangian. We also chose to add the term $\frac{\alpha}{2}||x - x^k||_2^2$ to the objective function f, which corresponds to a proximal regularization. Indeed, this term modifies $\tilde{P} = P + \alpha\,\mathrm{Id}$ and $\tilde{q}_k = q - \alpha\, x^k$. This regularization of P allows to handle ill-conditioned cases. In addition, it also induces that the smallest eigenvalue of $\tilde{P}$ is equal to α, and, using the work from [35], we can automatically scale the parameter ρ of the augmented Lagrangian, with $\rho = \sqrt{L\ \alpha}\ \left(\frac{L}{\alpha}\right)^{0.4}$ where L is the biggest eigenvalue of P. Moreover, we observe that the step $z^{k+1} = \underset{z}{\text{argmin}} \quad \mathcal{L}_\rho(x^{k+1}, z, y^k)$ can be seen as a projection step on the convex set $\mathcal{C}$. We adapt ρ during the optimization, in the way proposed by [34]. That way, when the ratio between the primal and dual residual is over a threshold (we fixed it to 10), we correspondingly adapt ρ by multiplying or dividing by the conditioning number $\left(\frac{L}{\alpha}\right)^{0.1}$. Due to this automatic scaling of ρ, the ADMM algorithm allows to solve the QCQP problem in a very efficient way and stable way for a large class of rigid body systems. Because the dual variable of the problem is computed iteratively, another advantage of the ADMM algorithm is that it induces a natural optimality criterion which is the verification of the KKT optimality conditions on the gradient of the Lagrangian $\|Px + q + y\|_\infty < \epsilon$.