Reconciling RaiSim with the Maximum Dissipation Principle

Quentin Le Lidec, Justin Carpentier

Reconciling RaiSim with the Maximum Dissipation Principle

Quentin Le Lidec[1,†] and Justin Carpentier[1]

***Abstract*—Recent progress in reinforcement learning (RL) in robotics has been obtained by training control policy directly in simulation. Particularly in the context of quadrupedal locomotion, astonishing locomotion policies depicting high robustness against environmental perturbations have been trained by leveraging RaiSim simulator. While being more realistic than its counterparts, it has been shown recently that RaiSim does not obey the maximum dissipation principle, a fundamental principle when simulating rigid contact interactions. In this note, we detail these relaxations and propose an algorithmic correction of the RaiSim contact algorithm to handle the maximum dissipation principle adequately. Our experiments empirically demonstrate our approach leads to more physically-consistent simulation.**

I. Introduction

Over the past few years, RaiSim simulator [1] gave rise to successful applications of policy learning to solve real-hardware robotic locomotion tasks in uncontrolled environments with a remarkable agility [2], [3], [4]. For all these applications, the control policies achieve good performances in practice, even if learned purely in simulation. A central ingredient that might explain this success is the level of physical realism of RaiSim, particularly its capacity to accurately simulate complex contact interactions between a robot and its environment. However, in a recent study [5], we have shown RaiSim is still making some approximations, particularly in the way it fulfills the so-called maximum dissipation principle (MDP) introduced by Jean-Jacques Moreau [6]. In this note, we propose a correction to the vanilla contact formulation of RaiSim to mitigate these approximations via a simple and computationally free correction. This note is organized as follows. In Section II, we quickly revisit the hypotheses of contact modeling and, in particular, of RaiSim's contact model. Section III depicts the main limitation of the RaiSim contact model and extends it to account for the MDP adequately. Finally, this correction is evaluated through various illustrative examples in Section IV. This note is based on a more general study of the most common contact models employed in robotics simulators [5].

II. Background

Contact modelling. Using generalized coordinates $q \in \mathcal{Q} \cong \mathbb{R}^{n_q}$ and the joint velocity $v \in \mathcal{T}_q\mathcal{Q} = \mathbb{R}^{n_v}$ to describe the state of the system, the discrete Lagrangian equations of motion write:

$$Mv^{t+1} = Mv^t + \left(\tau - b(q^t, v^t)\right)\Delta t. \tag{1}$$

where M is the joint space inertia matrix, τ the joint torque vector while $b(q, v)$ accounts for the centrifugal and Coriolis effects, and for the generalized gravity, and Δt is simulation time-step. We denote $v_f \overset{\text{def}}{=} v^t + M^{-1}\left(\tau - Cv^t - g\right)\Delta t$ as the free velocity corresponding to the solution of (1) w.r.t. v^{t+1}. When contacts occur *i.e.* the normal component of the separation vector $\Phi(q)$ (Fig. 1) between the objects in contact is non positive, (1) is modified according to Gauss' least constraint principle:

$$Mv^{t+1} = Mv_f + J^\top \lambda \tag{2}$$

where $J = {}^{\partial\Phi}/_{\partial q}$ is the contact Jacobian which can be computed efficiently via the rigid body dynamics algorithms [7].

[1]Inria - Département d'Informatique de l'École normale supérieure, PSL Research University. Email: firstname.lastname@inria.fr
[†]Corresponding author

To avoid interpenetration, the normal contact point velocities, and contact forces are both required to be positive in order to avoid interpenetration and to allow only repulsive contact forces. Moreover, contact forces can act on rigid bodies only when in contact. These hypotheses lead to the so-called *Signorini condition* which can be written at the velocity level:

$$\forall i,\ 0 \leq \lambda_N^{(i)} \perp c_N^{(i)} - c_N^{(i)*} \geq 0. \tag{3}$$

where the super-script i refers to the i$^{\text{th}}$ contact point, the sub-script N accounts for the normal component, $c = Jv^{t+1}$ is the contact points velocity and c^* is the reference velocity of the contact points.

To model friction, the phenomenological Coulomb's law of friction is classically adopted. It states that contact forces should lie inside a second-order cone:

$$\lambda \in K_\mu = \prod_{i=1}^{n_c} K_{\mu^{(i)}} \tag{4}$$

where $K_{\mu^{(i)}} = \left\{\lambda^{(i)} \in \mathbb{R}^3 | \lambda_N^{(i)} \geq 0, \|\lambda_T^{(i)}\|_2 \leq \mu^{(i)}\lambda_N^{(i)}\right\}$ and the sub-script T accounts for the tangential components.

In addition, according to the Maximum Dissipation Principle (MDP), frictions should maximize the dissipated energy, and, combined with Coulomb's set of admissible friction forces, this writes:

$$\forall i,\ \lambda_T^{(i)} = -\mu^{(i)}\lambda_N^{(i)}\frac{c_T^{(i)}}{\|c_T\|_2}, \text{if } \|c_T^{(i)}\|_2 > 0. \tag{5}$$

Reworking the equations (2) (3) (4) (5) leads to the following nonlinear complementarity problem (NCP):

$$\forall i, K_{\mu^{(i)}} \ni \lambda^{(i)} \perp c^{(i)} + \Gamma\left(c^{(i)}, \mu^{(i)}\right) \in K^*_{\mu^{(i)}}. \tag{6}$$

where $c = G\lambda + g$ is the contact point velocity, $G = JM^{-1}J^\top$ is the so-called Delassus matrix, $g = Jv^f$ is the free velocity of contact points and Γ is the de Saxcé function defined by $\Gamma : (c, \mu) \in \mathbb{R}^3 \times \mathbb{R} \mapsto [0, 0, \mu\|c_T\|_2]$. The deviation from the physical principles can be measured via the primal and dual errors, respectively $\epsilon_{\text{p}}^{(i)} = \text{dist}_{K_{\mu^{(i)}}}\left(\frac{\lambda^{(i)}}{\Delta t}\right)$ and $\epsilon_{\text{d}}^{(i)} = \text{dist}_{K^*_{\mu^{(i)}}}\left(c^{(i)} + \Gamma\left(c^{(i)}, \mu^{(i)}\right)\right)$, and the complementarity $\epsilon_{\text{c}}^{(i)} = \left|\langle\frac{\lambda^{(i)}}{\Delta t}, c^{(i)} + \Gamma\left(c^{(i)}, \mu^{(i)}\right)\rangle\right|$. In our experiments, we use ϵ_{abs}, defined as the maximum of ϵ_{p}, ϵ_{d} and ϵ_{c}, to quantify the physical accuracy.

A more exhaustive introduction to contact models in robotics can be found in [5].

RaiSim uses a contact model inspired by MuJoCo's Cone Complementarity Problem (CCP). CCP relaxes the NCP (6) by ignoring de Saxcé corrective term, thus, transforming the contact problem into a Quadratically Constrained Quadratic Program (QCQP). This results in the violation of the *Signorini condition* whenever a contact

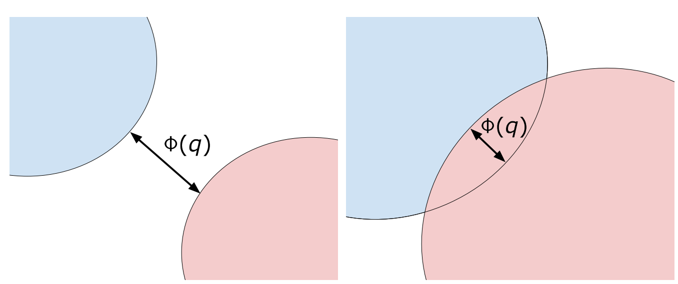

Figure 1. **The separation vector** Φ represents the displacement of minimal norm which puts objects in contact.

point is sliding. RaiSim intents to fix this by enforcing λ to lie on the hyperplane of null normal velocity $V_N^{(i)} = \{\lambda^{(i)} | G_N^{(ii)}\lambda^{(i)} + \tilde{g}_N^{(i)} = 0\}$ where $\tilde{g}^{(i)} = g^{(i)} + \sum_{j \neq i} G^{(ij)}\lambda^{(j)}$ is the i$^{\text{th}}$ contact point velocity as if it were free, and where we generalized the superscript (resp., subscript) notation to block operations on matrices with the first superscript (resp., subscripts) denoting the indexes of the rows while the second one refers to columns. This choice induces solutions conforming to the *Signorini condition*. When a contact point is sliding, the bisection algorithm is used to solve the following QCQP:

$$\min_{\lambda \in \partial K_{\mu^{(i)}} \cap V_N^{(i)}} \frac{1}{2}\lambda^\top G^{(ii)}\lambda + \tilde{g}^{(i)\top}\lambda \tag{7}$$

The set $\partial K_{\mu^{(i)}} \cap V_N^{(i)}$ being an ellipse, RaiSim leverages its analytical expression and the associated polar coordinates to write (7) as a 1D optimization problem on the angle θ before solving it via a dichotomous algorithm. We refer to [1] and [5] for a more detailed description of the Gauss-Seidel bisection algorithm.

III. Limits and extensions of RaiSim's contact model

RaiSim's limitations. Writing the Karush-Kuhn-Tucker (KKT) optimality conditions of the problem (7) yields:

$$G^{(ii)}\lambda + \tilde{g}^{(i)} + \gamma_1 G_N^{(ii)} + \gamma_2 \begin{bmatrix} \frac{\lambda_T^{(i)}}{\|\lambda_T^{(i)}\|_2} \\ -\mu^{(i)} \end{bmatrix} = 0 \tag{8a}$$

$$\|\lambda_T^{(i)}\|_2 = \mu^{(i)}\lambda_N^{(i)} \tag{8b}$$

$$G_N^{(ii)}\lambda^{(i)} + \tilde{g}_N^{(i)} = 0 \tag{8c}$$

where $\gamma_{1,2}$ are the dual variables associated to (7). Recalling that $c^{(i)} = G^{(i)}\lambda + g^{(i)}$ and injecting (8b),(8c) into (8a) gives:

$$\gamma_1 G_{NN}^{(ii)} - \gamma_2\mu^{(i)} = 0 \tag{9a}$$

$$c_T^{(i)} + \gamma_1 G_{NT}^{(ii)} + \gamma_2 \frac{\lambda_T^{(i)}}{\|\lambda_T^{(i)}\|_2} = 0 \tag{9b}$$

Finally, using (9a) to express γ_1 in (9b) leads to the following:

$$c_T^{(i)} \propto -\lambda_T^{(i)} - \frac{{\mu^{(i)}}^2\lambda_N^{(i)}}{G_{NN}^{(ii)}} G_{NT}^{(ii)} \tag{10}$$

which indicates that the contact model (7) proposed in [1] violates the MDP (5) whenever the Delassus matrix is not decoupled *i.e.* $G_{NT}^{(ii)}$ is not null.

De Saxcé correction. To retrieve the MDP, we propose to reintroduce de Saxcé's correction [8], which was neglected in the CCP formulation (Fig. 2). Because this correction is nonlinear in λ, directly incorporating it would make the algorithmic complex. In a Gauss-Seidel spirit, the corrective term can be approximated by a constant value using the latest estimate of the contact forces λ^-, leading to:

$$K_{\mu^{(i)}} \ni \lambda \perp G^{(ii)}\lambda + \tilde{g}^{(i)} + \Gamma(c^-, \mu^{(i)}) \in K_\mu^* \tag{11}$$

which constitutes the KKT conditions of the following problem:

$$\min_{\lambda \in K_{\mu^{(i)}}} \frac{1}{2}\lambda^\top G^{(ii)}\lambda + \left(\tilde{g}^{(i)} + \Gamma(c^-, \mu^{(i)})\right)^\top \lambda \tag{12}$$

and where $c^- = G^{(ii)}\lambda^- + \tilde{g}^{(i)}$. At convergence $c^- \to c^{(i)}$ and (11) becomes exactly (6).

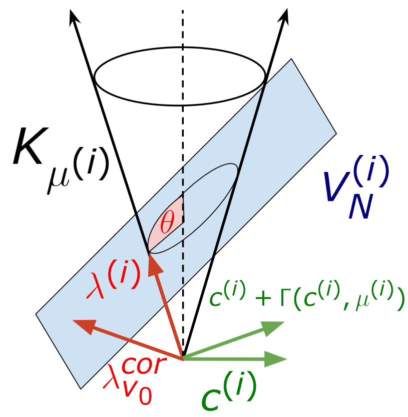

Figure 2. **Bisection algorithm.** When the contact point is sliding, RaiSim solves (7) while adding de Saxcé's correction (12) allows to retrieve the original NCP. The corrected problem (12) is equivalent to finding the element of $\partial K_{\mu^{(i)}} \cap V_N^{(i)}$ which is the closest to $\lambda_{v_0}^{cor}$ (Alg. 1, line 11). In both cases, the constraint set is an ellipse, which boils down to a 1D problem on θ using polar coordinates. This figure is inspired by Fig. 2 of [1].

Algorithm 1: De Saxcé correction of the Gauss-Seidel bisection algorithm

Input: Delassus matrix: G, free velocity: g, friction cones K_μ
Output: Contact forces: λ
1 **for** $k = 1$ **to** M **do**
2   **for** $i = 1$ **to** n_c **do**
3         $\tilde{g}^{(i)} \leftarrow g^{(i)} + \sum_{j \neq i} G^{(ij)}\lambda^{(j)}$;
4         $\lambda_{v_0}^{(i)} \leftarrow -{G^{(ii)}}^{-1}\tilde{g}^{(i)}$;
5   **if** $\tilde{g}_N^{(i)} > 0$ **then**
            // takeoff
6             $\lambda^* \leftarrow 0$;
7   **else if** $\lambda_{v_0}^{(i)} \in K_{\mu^{(i)}}$ **then**
            // stiction
8             $\lambda^* \leftarrow \lambda_{v_0}^{(i)}$;
9   **else**
            // sliding
10             $\tilde{g}^{cor} \leftarrow \tilde{g}^{(i)} + \Gamma(G^{(ii)}\lambda^{(i)} + \tilde{g}^{(i)}, \mu^{(i)})$;
11             $\lambda_{v_0}^{cor} \leftarrow -{G^{(ii)}}^{-1}\tilde{g}^{cor}$;
12             $\lambda^* \leftarrow \text{bisection}(G^{(ii)}, \tilde{g}^{cor}, K_{\mu^{(i)}}, \lambda_{v_0}^{cor})$;
13   **end**
14         $\lambda^{(i)} \leftarrow (1-\alpha)\lambda^{(i)} + \alpha\lambda^*$;
15         $\alpha \leftarrow \gamma\alpha + (1-\gamma)\alpha_{min}$;
16   **end**
17 **end**

For a sliding contact, (11) involves that $\lambda \in \partial K_{\mu^{(i)}} \cap V_N^{(i)}$, and thus, when solving (12) one can restrain the search over this sub-set, leading to the following corrected variant of (7):

$$\min_{\lambda \in \partial K_{\mu^{(i)}} \cap V_N^{(i)}} \frac{1}{2}\lambda^\top G^{(ii)}\lambda + \left(\tilde{g}^{(i)} + \Gamma(c^-, \mu^{(i)})\right)^\top \lambda \tag{13}$$

Practically, the original per-contact bisection algorithm can be adapted to (13) by iteratively updating the latter correction (Alg. 1, lines 10, 11), in a similar way to [9]. The resulting algorithm is called RaiSim+DS as it consists in a variant of RaiSim which includes a de Saxcé correction.

It is worth noting that the proposed correction offers no convergence guarantees just like the original algorithm. In practice, over-relaxation

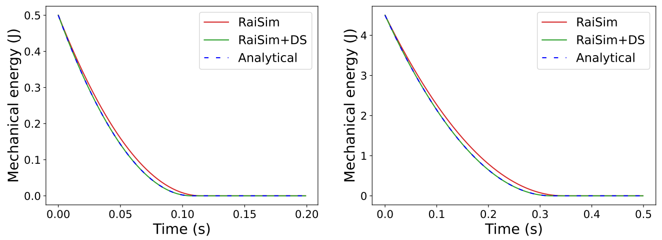

Figure 3. **A cube sliding on a plane**, with an initial tangential velocity of 1 m/s (**Left**) and 3 m/s (**Right**). Because RaiSim's contact model violates the MDP, the dissipated energy decreases more slowly than expected by the analytical solution governed by the maximum dissipation principle. On the contrary, with de Saxcé's corrective term, the contact model recovers the expected evolution.

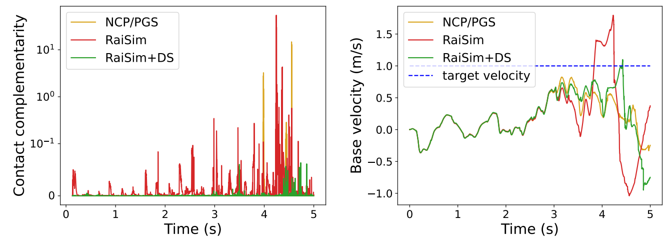

Figure 5. **MPC controller for Solo-12** is run in simulation with various contact models and solvers. The robot is operating on a bumpy and slippery terrain. The De Saxcé correction leads to a reduced contact complementarity error ϵ_{abs} w.r.t. the original RaiSim solver (**Left**). The correction also leads to very different controlled trajectories (**Right**).

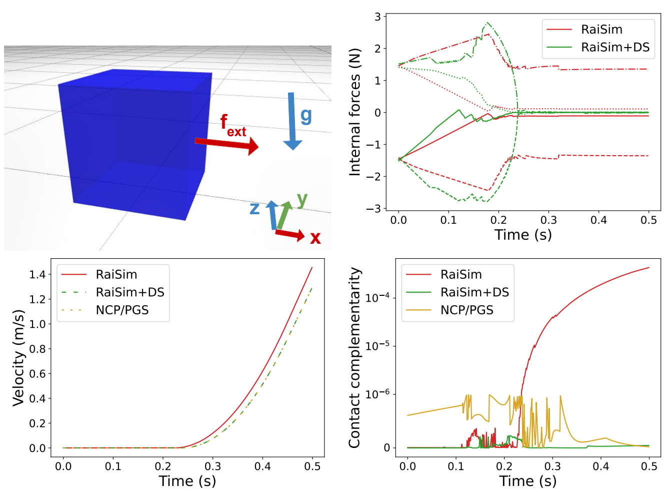

Figure 4. **Top left:** A cube is dragged on a plane by an increasing external tangential force along the x-axis. When the cube starts to slide along the x-axis, according to the MDP, the tangential contact forces should be coliinear to the same direction. **Top right:** The curves represent the internal force along the y-axis at each of the four contact points. RaiSim exhibits some remaining internal forces while our de Saxcé correction results in zero internal forces. **Bottom left:** because the dissipation is not maximal, the velocity simulated by RaiSim grows faster than what is obtained by our approach or a more classical PGS algorithm. **Bottom right:** the violation of the MDP is confirmed by a jump of the NCP criterion ϵ_{abs} when the cube is sliding, while the proposed correction keeps this criterion at a relatively low level.

with a decreasing α (Alg. 1, lines 14,15) is required to stabilize the algorithm and, thus, even at convergence correctness is not guaranteed.

IV. Experiments

The RaiSim algorithm and our proposed correction are implemented in ContactBench [5], a generic framework implementing contact solvers commonly used in robotics and leveraging the Pinocchio [10] and HPP-FCL [11] C++ libraries. ContactBench will be publicly available upon publication acceptance. Here, we run experiments inspired by the same previous work [5] in order to observe RaiSim's approximation pointed out in Section III and how our corrected contact model improves simulation.

A. Experiment 1: sliding cube

An experiment as simple as a cube sliding on a horizontal plane allows us to visualize how RaiSim's contact model violates the MDP, as evidenced by (10). Indeed, in this case, Fig. 3 exhibits a gap between the analytical evolution of the mechanical energy and its simulation via RaiSim's contact model. Moreover, the same experiment reveals that our approach using de Saxcé's correction allows us to bridge this gap.

B. Experiment 2: dragging a cube

For underdetermined cases, *e.g.* hyper-staticity, the contact problem can have an infinite set of solutions and so we denote by "internal forces", the contact force components deviating from the minimum norm solution. The second experiment, on a cube that is progressively dragged (Fig. 4), demonstrates that if both our approach and RaiSim lead to internal forces stretching the cube at stiction, the latter still induces to non-null internal components when the cube slides. The studied contact problem is under-determined at stiction, and internal forces do not affect the resulting trajectory. When the contact points are sliding, the friction forces are uniquely defined by the MDP (5) and should not exhibit any internal components. The correction results in a trajectory close to the one obtained via a Projected Gauss-Seidel (PGS) algorithm (see [5] for more details) which is an approach avoiding any physical approximation. This indicates that if RaiSim originally violates the MDP, adding a simple de Saxcé corrective term reconciles the simulator with this energetic principle.

C. Experiment 3: MPC for quadrupedal locomotion

Our last experiment aims at evaluating the impact of the de Saxcé correction on a more concrete application: Model Predictive Control (MPC) for quadrupedal locomotion with the Solo-12 robot. Running an MPC controller in a simulator using RaiSim's contact model and solvers with and without de Saxcé correction leads to different behaviors (Fig. 5). In this experiment, the number of iterations is fixed which is why the PGS algorithm cannot always control the contact complementarity error. RaiSim also leads to large errors but increasing the number of iterations would not help in this case because the MDP is inherently approximated. Eventually, RaiSim's algorithm when equipped with de Saxcé's correction is able to keep this error at a lower level.

V. Conclusion

In this note, following [5], we have shown that the RaiSim simulator can exhibit non-physical behaviors. Our study slightly modified the original algorithm to account for the maximum dissipation principle properly. Throughout our experiments, we empirically demonstrate our computationally-free modification can improve the physical consistency of simulation without additional computational burden. We hope this note will motivate new developments and progress in contact simulation in robotics.

Acknowledgments

We thank Jemin Hwangbo for providing useful details on the algorithm of the RaiSim simulator. This work was supported in part by L'Agence d'Innovation Défense, the French government under the management of Agence Nationale de la Recherche through the project INEXACT (ANR-22-CE33-0007-01) and as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), by the European Union through the AGIMUS project (GA no.101070165) and the Louis Vuitton ENS Chair on Artificial Intelligence. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[4] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[5] Q. Le Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: a comparative analysis," *Under Review*, 2023.

[6] J. J. Moreau, "Unilateral contact and dry friction in finite freedom dynamics," *Nonsmooth mechanics and Applications*, pp. 1–82, 1988.

[7] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[8] G. de Saxcé and Z.-Q. Feng, "The bipotential method: A constructive approach to design the complete contact law with friction and improved numerical algorithms," *Mathematical and Computer Modelling*, vol. 28, pp. 225–245, Aug. 1998.

[9] F. Cadoux, "An optimization-based algorithm for coulomb frictional contact," in *CANUM 2008-39e Congrès National d'Analyse Numérique*, vol. 27, pp. 54–69, EDP Science, 2008.

[10] 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 *2019 IEEE/SICE International Symposium on System Integration (SII)*, pp. 614–619, IEEE, 2019.

[11] J. Pan, S. Chitta, D. Manocha, F. Lamiraux, J. Mirabel, J. Carpentier, *et al.*, "HPP-FCL: an extension of the Flexible Collision Library." https://github.com/humanoid-path-planner/hpp-fcl, 2015–2023.