Towards Efficient Optimization Through Contact: Simulation, Gradients, and Algorithms

Simon Le Cleac'h
Stanford University
simonlc@stanford.edu

Contact is essential for a robot to interact with its environment. Contact interactions enable robots to navigate the world and to manipulate objects in their surroundings. That is why the ability to make and break contact with the environment safely and effectively is key to robot autonomy. Yet, contact dynamics presents a number of challenges to current learning and optimization-based methods. At a low level, controlling these systems involves high frequency impact disturbances which require quick reactive control [14, 5, 2]. It also introduces stiff dynamics equations that are hard to capture with current learning techniques [21]. At a higher level, long-horizon planning often requires dealing with discrete decisions [16, 18, 29]. Should I make contact with this object in order to grasp that one? Optimization over discrete variables is known to be challenging especially in time-constrained scenarios. Policy optimization and reinforcement learning (RL) techniques [9] have provided a way to control complex dynamical systems involving contact like the OpenAI's locomotion environments encoded in MuJoCo [4] and Isaac [15]. However, current methods often require an extremely large amount of samples. Indeed, it is common to see RL techniques requiring millions of simulation steps to obtain satisfactory performance [9, 26, 24].

The goal of my research lies in addressing these challenges to be able to **quickly and reliably generate robot behaviors that actively exploit contact interactions with the environment**. Towards this goal my research has focused on three key questions. **First**, how to effectively embed contact dynamics and physics engines in existing learning and optimization pipelines? **Second**, how do we leverage differentiable physics engines for real-time control of robots that make and break contact with their environments? **Third**, how can we combine sampling-based and gradient-based methods for contact-rich behavior learning?

A. Smoothly Differentiable Physics Engine

In order to effectively integrate physics engines into existing learning pipelines, we need to differentiate them efficiently. Historically, physics engines were providing gradients obtained using finite-difference schemes e.g. MuJoCo [28]. However, this method can be computationally costly and does not scale well with the gradient dimension. Recently, there as been a push for differentiable physics engines relying on automatic differentiation frameworks e.g. Brax [8], TinyDiff [11], Drake [27]. These approaches return exact gradients which are not informative about the broader dynamics landscape.

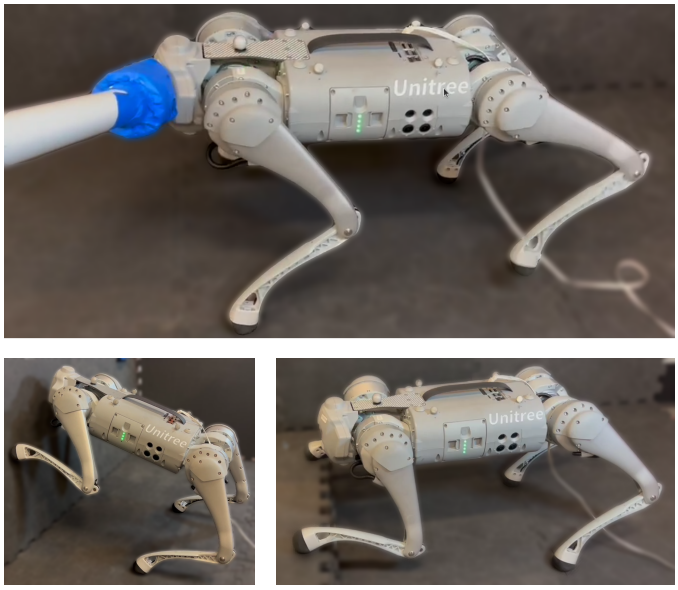

Fig. 1: The optimization-based policy queries and differentiates through the physics engine to find the sequence of controls that optimally tracks the reference trajectory. This policy is demonstrated on a Unitree Go1 quadruped: stable trotting while being pushed (top), transitioning from ground to standing against a wall (bottom left), and placing two feet onto a step (bottom right).

This makes them of little use when integrated in optimization pipelines. For instance, a gradient-based method relying on exact gradients would fail on a simple box-lifting task (Fig. 2) because the gradients are null initially. To resolve this issue, we introduced Dojo [13], a differentiable physics engine that can provide smooth gradients. These gradients carry meaningful information to efficiently solve downstream optimization tasks such as: policy optimization, planning, system identification and more. For instance, we can easily solve the box-lifting task by leveraging the smooth gradients since they capture the curvature of a broad dynamics landscape.

The method we introduced in Dojo to differentiate through contact dynamics relied on two key techniques. First, we formulated the problem of simulating contact between the robot and its environment as an optimization problem; specifically a nonlinear complementarity problem (NCP). To reliably solve the NCP, we devised a custom interior-point method [19, 3] building upon the Predictor-Corrector algorithm [17]. Second, we leveraged the implicit function theorem (IFT) [7] to differentiate through the NCP. This differentiation is computationally cheap and allows us to provide gradients with

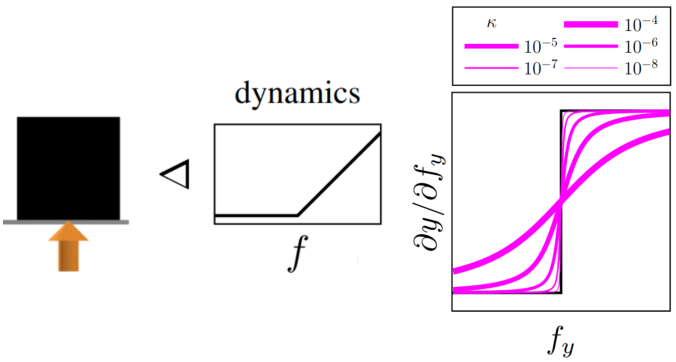

Fig. 2: We illustrate the gradient smoothness on a simple system: a box sitting on the floor (left). The contact dynamics is not smooth (center) because you need to overcome the force of gravity before you get any upward motion from the box. The exact dynamics gradients are discontinuous (right, black curve). Our approach [13] provides approximated gradients that can reach any desired level of smoothness by choosing κ the relaxation parameter (right, magenta).

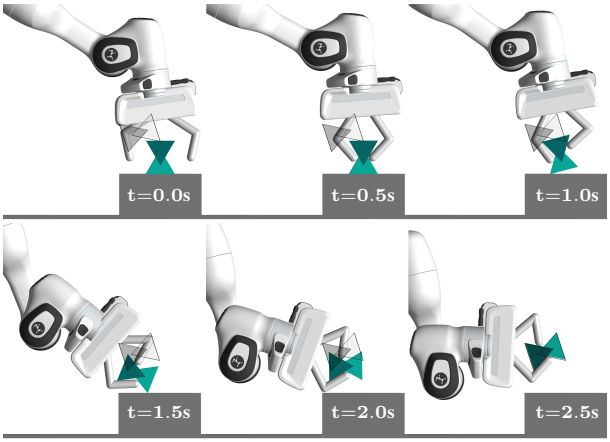

Fig. 3: We synthesize contact-rich manipulation behaviors through an approach combining gradient-based and sampling-based optimization. The robot arm is tasked with manipulating the green object to a desired pose (light gray overlay). Our simulation framework provides gradients of both the contact simulation and collision detection. The grasping plan shown is computed in 1 second on a laptop with an existing planner [20] requiring smooth gradients.

any desired level of smoothness by choosing the relaxation parameter used in the interior-point method (Fig. 2).

B. Real-Time Control Through Contact

In the previous section, we have provided a way to efficiently differentiate through contact dynamics. This allows us to integrate physics engines into learning and optimization pipelines. In this section, we embed a physics engine inside a real-time control policy. This results in a general control policy for systems that make and break contact with their environments. We introduce Contact-implicit model predictive control (CI-MPC) [5]. It generalizes linear MPC to contact-rich settings replacing linear dynamics with dynamics encoded by a simplified physics engine. In order to find the best sequence of controls, the MPC algorithm needs to evaluate and differentiate the dynamics constraints. This means querying the physics engine and differentiating through it. We have observed that the gradient smoothness was essential to ensure successful convergence of the MPC policy.

With this framework, we can track reference trajectories for a variety of systems involving contact: a Raibert hopper [23], a quadruped, and a planar biped [22]. We show that this policy is robust to model mismatch and can respond to disturbances by discovering and exploiting new contact modes across a variety of robotic systems in simulation. We also demonstrate real-time solution rates for CI-MPC and the ability to generate and track non-periodic behaviours in hardware experiments on a quadrupedal robot (Fig. 1).

C. Contact-Rich Behavior Optimization

In the previous section, we have seen how to leverage differentiable physics engines online to build real-time policies that work on hardware. In this section, we focus on learning offline contact-rich behavior optimization. First, we have learned simple locomotion policies for the half-cheetah [10, 30] and ant [25] robots from OpenAI Gym [4]. This was done by leveraging the dynamics gradients provided by Dojo [13]. Similarly to Xu's work [31], we have observed a significant decrease in the number of samples required to obtain a satisfactory locomotion behavior. Second, we have learned contact-rich manipulation plans where we manipulate an object to a desired location [6] (Fig. 3). We leveraged a sampling- and gradient-based algorithm proposed by Tao [20]. This algorithm leverages our ability to provide smooth gradient information through contact dynamics as well as through the collision detection routine. We obtained manipulation plans that featured a wide variety of low-level manipulation skills: sliding, tilting, reorienting, and lifting the object.

D. Future Work

In my ongoing and future research, I would like to expand on my previous work to develop learned MPC-based policies for robotic locomotion and manipulation. The recent success of sampling-based MPC policies [12] combined with policy optimization techniques [26, 24] could enable manipulation and locomotion behavior that successfully complete long-horizon tasks.

Additionally, I am eager to build a learned physics engine. This idea is driven by several motivations. First, it could better capture the physics by learning from real-world data, building accurate contact models without requiring time consuming parameter tuning. Second, when we simulate a specific robot for millions of steps, considering that each simulation step is solving an optimization problem, we are solving millions of highly related optimization problems. Leveraging amortized optimization [1] to learn components of the solver would, I believe, lead to substantial improvement in both reliability and speed. Furthermore, a learned physics engine would require simpler optimization and linear algebra routines than the current predictor-corrector interior-point solver. This would ease the deployment onto the latest GPU architectures unlocking key performance gains.

References

[1] Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022.

[2] Alp Aydinoglu and Michael Posa. Real-time multi-contact model predictive control via admm. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3414–3421. IEEE, 2022.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[5] Simon Le Cleac'h, Taylor Howell, Mac Schwager, and Zachary Manchester. Fast contact-implicit model-predictive control. *arXiv preprint arXiv:2107.05616*, 2021.

[6] Simon Le Cleac'h, Mac Schwager, Zachary Manchester, Vikas Sindhwani, Pete Florence, and Sumeet Singh. Single-level differentiable contact simulation. *arXiv preprint arXiv:2212.06764*, 2022.

[7] Ulisse Dini. *Lezioni di analisi infinitesimale*, volume 1. Fratelli Nistri, 1907.

[8] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax-A differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021.

[9] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[10] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. *Advances in neural information processing systems*, 28, 2015.

[11] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S. Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. *arXiv preprint arXiv:2011.04217*, 2020.

[12] Taylor Howell, Nimrod Gileadi, Saran Tunyasuvunakool, Kevin Zakka, Tom Erez, and Yuval Tassa. Predictive sampling: Real-time behaviour synthesis with mujoco. *arXiv preprint arXiv:2212.00541*, 2022.

[13] Taylor Howell, Simon Le Cleac'h, Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable physics engine for robotics. *arXiv preprint arXiv:2203.00806*, 2022. URL https://arxiv.org/abs/2203.00806.

[14] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.

[15] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[16] Tobia Marcucci, Robin Deits, Marco Gabiccini, Antonio Bicchi, and Russ Tedrake. Approximate hybrid model predictive control for multi-contact push recovery in complex environments. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 31–38. IEEE, 2017.

[17] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2 (4):575–601, 1992.

[18] Toki Migimatsu and Jeannette Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5(2):844–851, 2020.

[19] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

[20] Tao Pang, HJ Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *arXiv preprint arXiv:2206.10787*, 2022.

[21] Samuel Pfrommer, Mathew Halm, and Michael Posa. ContactNets: Learning discontinuous contact dynamics with smooth, implicit representations. *arXiv preprint arXiv:2009.11193*, 2020.

[22] Jerry Pratt and Gill Pratt. Intuitive control of a planar bipedal walking robot. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2014–2021, 1998.

[23] Marc H. Raibert, H. Benjamin Brown Jr., Michael Chepponis, Jeff Koechling, Jessica K. Hodgins, Diane Dustman, W. Kevin Brennan, David S. Barrett, Clay M. Thompson, John Daniell Hebert, Woojin Lee, and Borvansky Lance. Dynamically stable legged locomotion. Technical report, Massachusetts Institute of Technology Cambridge Artificial Intelligence Lab, 1989.

[24] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[25] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

[28] 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.

[29] Marc Toussaint, Jason Harris, Jung-Su Ha, Danny Driess, and Wolfgang Hönig. Sequence-of-constraints mpc: Reactive timing-optimal control of sequential manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13753–13760. IEEE, 2022.

[30] Pawel Wawrzynski. Learning to control a 6-degree-of-freedom walking robot. In *EUROCON 2007-The International Conference on" Computer as a Tool"*, pages 698–705. IEEE, 2007.

[31] Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. *arXiv preprint arXiv:2204.07137*, 2022.