Closing the Sim2Real Gap using Invertible Simulators

Eric Heiden
University of Southern California

I. Introduction

Employing robots in the real world to perform a large variety of tasks remains a great challenge to current perception, planning and control algorithms. Various specialized representations, such as for mapping or localization, have been proposed which are typically used in fixed pipelines that fuse perception, planning and control. These approaches are typically highly interpretable in a way that humans can reason about the prediction uncertainty of the system, or what additional measurements are necessary to improve the predictions. On the other hand, such static frameworks do not allow the robot to learn from experience or adapt to changing task requirements.

Learning-based approaches have found great success in domains where large amounts of labelled data is available. Many problems in robotics, however, do not belong to such regime where training data is easily obtainable. Instead, it is often only possible to provide few kinesthetic demonstrations, or rely entirely on self-supervised or reinforcement learning. While the longstanding motivation behind such learning approaches is to enable robots to improve by learning from their own experience, the current instantiations of state-of-the-art reinforcement learning (RL) algorithms, even model-based, require extensive amounts of interaction samples, such that, in most cases, simulators are necessary to provide a risk-free environment that runs orders of magnitude faster than real time. With regards to accountable AI, many of these approaches are not human-interpretable – they may achieve high performance on certain tasks but it remains an open research question how the training setup must be designed to guarantee performance throughout all metrics over the tasks of interest.

Whether control policies are learned through reinforcement learning, or feedback control laws are optimized – in most approaches simulators are used to validate the algorithms before deploying them on the real system. An inherent problem to such techniques is the disparity between the simulated and the real world, i.e., the sim2real gap. Various methods have been proposed to overcome this issue, such as domain randomization and domain adaption.

In this work, we approach the problem of visuomotor control from a different angle. Instead of learning a separate model or policy in a simulator, we use the simulator as the model that we can use to derive controllers and estimate the state of our system of interest. Simulators, such as physics engines, already encode our understanding of the world through the laws of physics and generalize well to a wide variety of application scenarios. Most quantities, such as the geometry of the objects, are human-interpretable and can be even verified through a variety of specialized tools. We propose to design the simulator from the ground up to be *invertible*, i.e., such that we can estimate the simulation settings from the observations of the real system.

II. Related Work

A variety of novel deep learning architectures have been proposed to learn *intuitive physics* models. Inductive bias has been introduced through graph neural networks (e.g. [17]), particularly interaction networks (e.g. [1]) that are able to learn rigid and soft body dynamics. Vision-based machine learning approaches to predict the future outcomes of the state of the world have been proposed e.g. in [19, 10]. Nonetheless, such models allow little room for interpretability and their parameters are not grounded in the laws of physics.

The idea of devising invertible, analytical models has been explored in computer graphics through differentiable rendering systems [15, 12, 14], and in the simulation of rigid-body dynamics [6, 4]. Various differentiable simulators have been implemented recently in the Taichi programming language [**?**]. We leverage similar ideas in our work, yet propose to make these analytical models more amenable to gradient-based optimization.

The approach of adapting the simulator to real world dynamics has been less explored. While many previous works have shown to adapt simulators to the real world using system identification and state estimation (e.g. [11]), few have shown adaptive model-based control schemes that actively close the feedback loop between the real and the simulated system [3].

III. Differentiable Simulators

In this section, we highlight our prior work on building invertible simulators that enable system identification and control through gradient-based optimization. To this end, we focused our research efforts on dynamics simulation of articulated rigid bodies [8], and sensor simulation in the form of a physics-based rendering approach to model optical sensors [9], such as LIDAR sensors. Following [5], we implement a physics engine for mechanisms, such as robot arms, consisting of rigid bodies that are connected via joints. We apply automatic differentiation which allows us to compute gradients of any quantity involved in the simulation of complex systems, opening avenues to state estimation, optimal control and system design.

A. Probabilistic System Identification

To reduce the mismatch between the simulated and the real-world behavior (sim2real gap), we can directly minimize the

ℓ^2-norm of the discrepancy between the simulated and the measured states from the real system. Minimizing this sum-of-squares error results in the approximation of the mean of the observed data [2]. However, the assumption that the simulation parameters are uniquely defined and can be estimated may not hold for complex models, or in cases where very few observations from the real system are available. Since we are interested in a general estimation approach that is able to capture potential couplings between model parameters and yield results under the presence of noisy observations, we investigate a multi-modal probabilistic estimation method to infer physical parameters [7]. We fit the distribution over the simulation parameters, such as the lengths of a three-link compound pendulum, via a Gaussian mixture model whose samples are input to our physics engine. Finally, we minimize the error between the simulated trajectories and the trajectories from the real system (which we collected from VICON motion capture measurements). The entire stochastic computation graph is end-to-end differentiable and our experiments demonstrate how multimodal simulation parameters can be estimated from a few samples of the real system, while accounting for the system identification uncertainty.

B. Adaptive MPC

Besides parameter estimation, a key benefit of differentiable physics is its applicability to optimal control algorithms. Trajectory optimization assumes that the dynamics model is accurate w.r.t the real world and generates sequences of actions that achieve optimal behavior toward a given goal state, leading to open-loop control. Model-predictive control (MPC) leverages trajectory optimization in a feedback loop. After some actions are executed in the real world and subsequent state samples are observed, *adaptive* MPC fits the dynamics model to these samples to align it closer with the real-world dynamics. We incorporate our dynamics model in such receding-horizon control algorithm to achieve swing-up motions of multi-link cartpoles in the MuJoCo simulator [18].

Within a handful of training episodes, adaptive MPC with ILQR [13] as trajectory optimizer infers the correct model parameters involved in the dynamics of a double cartpole, even when the initial parameterization is very different in the reference environment than our simulator. Leveraging our stochastic estimation approach, in future work we plan to design control algorithms that trade off exploration and exploitation to systematically close the sim2real gap through behaviors that excite part of the state space for which the model prediction uncertainty is high, while ensuring safe trajectories are followed.

C. Sensor modeling

So far we assumed we have access to the true state of the system we want to control, design, or identify. In our real-world experiments, we used a VICON motion capture system to accurately track the weights of a compound pendulum in order to infer its dynamical properties. However, to enable a robot to interact autonomously with the world, we cannot leverage such instrumentation but have to rely on sensor measurements to estimate the underlying system state. We start addressing the perception side of our framework by modeling light detection and ranging (LIDAR) sensors.

Extending our simulator-as-model framework, we start from first principles on how laser light interacts with surfaces under various conditions [16], and implement a physically plausible simulation that recovers various effects encountered with physical LIDAR, such as spurious measurements, reflection and refraction. Using automatic differentiation, we compute the derivatives of all parameters in our simulation with respect to the simulated measurements, and apply gradient-based optimizers to find the simulation parameters that most closely match the true observations. Among our experiments, we localize a LIDAR given its measurements, track a mirror, and infer material properties and calibration settings based on real laser scans. These tasks are typically infeasible by classical laser scan registration algorithms since they do not account for highly reflective or partially transparent materials. In addition, thanks to having end-to-end differentiable computation pipelines, we are able to include neural networks into our simulation that learn parts of the real world dynamics for which we have no analytical or only incomplete models. In the LIDAR simulation, we used such hybrid simulation approach to learn the effect of the radiance on the phase shift that the photodiode measured, and were able to recover various effects encountered on the real sensor. Future research is directed towards investigating how such hybrid approach can be employed in other domains as well, potentially leading to significant improvements in closing the sim2real gap while only requiring a few measurement samples from the real system, thanks to the strong inductive bias our simulation places on the learning problem.

IV. Conclusion

We have presented a framework where simulators are integrated into control and system identification pipelines. Through new approaches to modeling of rigid body dynamics and optical sensors, we propose invertible simulators whose parameters have physical meaning and can be estimated through uncertainty-aware system identification techniques. By constantly reducing the error between simulated and actual measurements in a feedback loop, the sim2real gap is directly minimized. We have outlined two main future research directions that we aim to pursue. First, we plan to incorporate techniques to soften hard constraints inside the simulation algorithms to make our method more amenable to local optimization, while integrating learnable residual models that accommodate unmodeled dynamics. Second, we design a feedback controller that uses the predicted level of uncertainty to explicitly trade off exploration and exploitation while following a task objective. Through these contributions, we aim to reach closer to the goal of a human-interpretable model that leads to complex behavior synthesis for robots interacting with the real world.

Acknowledgements

I thank my coauthors David Millard, Ziang Liu, Ragesh K. Ramachandran, Hejia Zhang, and my advisor Gaurav S. Sukhatme.

References

[1] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pages 4502–4510, 2016.

[2] Christopher M Bishop. Mixture density networks. Technical report, Aston University, 1994.

[3] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan D. Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *International Conference on Robotics and Automation*, 2019.

[4] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J. Zico Kolter. End-to-end differentiable physics for learning and control. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 7178–7189. Curran Associates, Inc., 2018.

[5] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 0387743146.

[6] Markus Giftthaler, Michael Neunert, Markus Stäuble, Marco Frigerio, Claudio Semini, and Jonas Buchli. Automatic differentiation of rigid body dynamics for optimal control and estimation. *Advanced Robotics*, 31(22):1225–1237, 2017. doi: 10.1080/01691864.2017.1395361.

[7] Eric Heiden, David Millard, and Gaurav S. Sukhatme. Real2sim transfer using differentiable physics. *R:SS Workshop on Closing the Reality Gap in Sim2real Transfer for Robotic Manipulation*, 2019.

[8] Eric Heiden, David Millard, Hejia Zhang, and Gaurav S. Sukhatme. Interactive differentiable simulation. *CoRR*, abs/1905.10706, 2019. URL http://arxiv.org/abs/1905.10706.

[9] Eric Heiden, Ziang Liu, Ragesh K. Ramachandran, and Gaurav S. Sukhatme. Physics-based simulation of continuous-wave LIDAR for localization, calibration and tracking. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[10] Michael Janner, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-centric models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJx9EhC9tQ.

[11] Svetoslav Kolev and Emanuel Todorov. Physically consistent state estimation and system identification for contacts. *International Conference on Humanoid Robots*, pages 1036–1043, 2015.

[12] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. In *SIGGRAPH Asia 2018 Technical Papers*, page 222. ACM, 2018.

[13] Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *International Conference on Informatics in Control, Automation and Robotics*, 2004.

[14] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. *arXiv preprint arXiv:1901.05567*, 2019.

[15] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[16] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[17] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In Jennifer Dy and Andreas Krause, editors, *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4470–4479, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/sanchez-gonzalez18a.html.

[18] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, pages 5026–5033, Oct 2012. doi: 10.1109/IROS.2012.6386109.

[19] Jiajun Wu, Ilker Yildirim, Joseph J. Lim, William T. Freeman, and Joshua B. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems*, pages 127–135, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969239.2969254.