Learning Dexterous Manipulation with Differentiable Physical Consistency

Philipp Ruppel
Department of Informatics
Universität Hamburg
Germany

Norman Hendrich
Department of Informatics
Universität Hamburg
Germany

Jianwei Zhang
Department of Informatics
Universität Hamburg
Germany

***Abstract*—Reinforcement learning and differentiable physics often suffer from local minima, particularly in contact-rich tasks. Some formulations also involve nested optimization, which can lead to further difficulties. To overcome these issues, we let the policy network propose contacts and simultaneously train for task goals and physical consistency. This leads to a single-level optimization problem, which we solve with efficient linearizations and matrix-free Gauss-Newton steps. Our prototype implementation can train neural networks for dexterous manipulation tasks within minutes on a single CPU core.**

I. Introduction

Deep reinforcement learning has become an extremely popular method in robotics [3]. Since it often requires many training episodes, it is typically performed in simulation. However, if a perfect model is already available in simulation, it should not even be necessary to try out random actions. We should be able to solve for a policy network directly.

A step in this direction is differentiable physics. We can propagate gradients through a physics simulator via automatic differentiation. However, most contact models are not differentiable. In some cases, we can still obtain useful gradients and even overcome small local minima by replacing discrete collision detection with a soft falloff (see figure 2, bottom). In many situations, however, this approach fails. If the materials are not sticky and we want to e.g. grasp a cube with the hand initially placed above the object (figure 2, top-left), the gradient would be zero, or point in the wrong direction.

For motion planning, contact-based formulations have been developed [1, 2], which overcome local minima by introducing additional free variables. When using a similar approach to train neural networks, the contact variables would have to be continuously re-optimized. We also want to add randomization (e.g. domain randomization) to obtain stable policies. Therefore, at each iteration, we would not only have to perform a cheap back propagation step, but we would first have to find new contact variables by solving constrained non-linear problems.

We overcome these issues by training a neural network to generate not only robot control signals but also contacts, simultaneously optimizing for task goals and physical consistency [4].

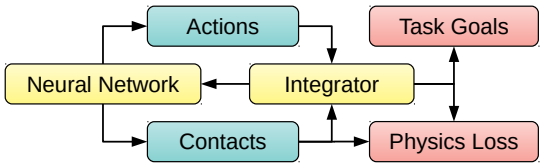

Fig. 1. We let the neural network propose contacts and simultaneously optimize for task goals and physical consistency.

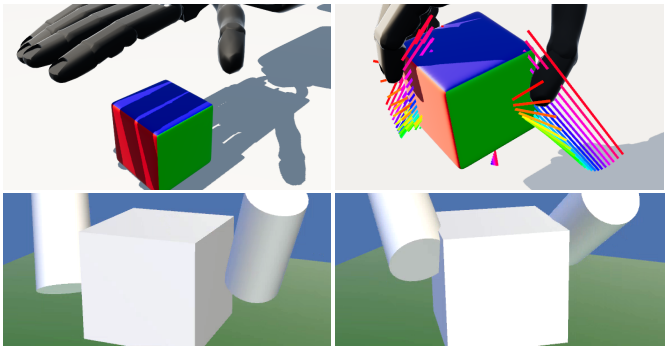

Fig. 2. Direct policy optimization with differentiable physics learns to reorient a cube with two fingers (bottom), but fails during a grasping task (top-left). We overcome (top) local minima by letting the neural network propose contacts (top-right, plotted over multiple time steps), simultaneously optimizing for task goals and physical consistency.

II. Formulation

A neural policy network receives state information as inputs and produces robot control signals as well as contact points and contact forces. We minimize a combined loss $L_t + L_p$ with task goals L_t and physical consistency terms L_p. The physical consistency loss L_p encourages the network to generate physically realistic contacts. For each time step $t \in T$, the network can output contact points $p_{a,b,t}$, $q_{a,b,t}$ and contact forces $f_{a,b,t}$ between bodies $a, b \in O$. We allow either small forces or small distances, while avoiding overlap $o_{a,b,t}$ and minimizing deviations $f_{a,b,t}$ from friction cones.

$$L_p = \sum_{a \in O} \sum_{b \in O} \sum_{t \in T} \left(\; |(p_{a,b,t} - q_{a,b,t}) \otimes f_{a,b,t}|^2 + f_{a,b,t}^2 + o_{a,b,t}^2 \; \right) \tag{1}$$

Robot commands, contact points and contact forces are integrated, and the states are fed back to the network as

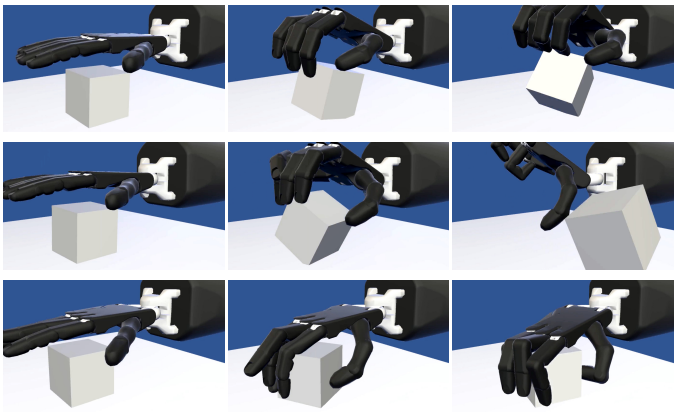

Fig. 3. Learning to rotate a cube around different axes (top, middle, bottom).

Fig. 4. Learning to adapt to different object positions.

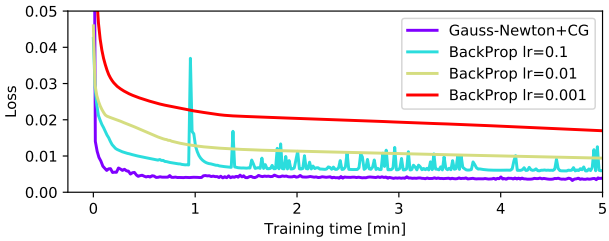

Fig. 5. Training curves with matrix-free Gauss-Newton and back propagation when learning to grasp a cube with a Shadow C6 hand.

inputs (see figure 1).

III. Solver

Artificial neural networks are usually trained via steepest gradient descent with back propagation, since the full gradient matrix would often be prohibitively large and dense. For many physics problems (e.g. contact dynamics), steepest gradient descent can be very slow, but the gradient matrices can be relatively small and sparse. These equations can be solved more efficiently with higher-order estimates, e.g. following the Gauss-Newton method. Our formulation combines artificial neural networks with contact dynamics in a single optimization problem. Instead of computing a gradient matrix, we generate linearized gradient programs via automatic program transformation. All non-linear operations are eliminated from the forward and reverse differentiation passes and moved into a separate linearization program. To efficiently handle spatial transformation, we introduce type conversions and linearizers. Each variable is associated with a value, a gradient and a linearizer. For a scalar, the linearizer is simply an implicit reference to the value. For a 3D pose, the value consists of a vector and a quaternion, the gradient is represented by a 6D vector screw and the linearizer stores a 3-by-3 matrix. We solve our multimodal policy optimization problem via Gauss-Newton steps as a sequence of linear equations. For each linear equation, we call the non-linear pass and the linearizers only once. A forward and a reverse gradient program are combined into a matrix replacement program, which only performs linear operations and is equivalent to a matrix-vector product with the Gauss-Newton matrix (but without having to compute said matrix). Using our matrix replacement, we perform a number of conjugate gradient descent steps.

IV. Experiments

We learn neural policies for a two-fingered reorientation task (figure 2, bottom), grasping a cube with a humanoid Shadow C6 hand mounted to a UR5 arm (figure 2, top), flipping a cube around different axes (figure 3), and turning a wheel (figure 4). Task goals are specified as squared errors between object poses and goal poses. In a first attempt, we implement a traditional rigid body physics simulator in our auto-differentiation framework. Since the fingers are initially not touching the object yet, the gradients are zero and policy optimization fails. We replace hard collision detection with a soft falloff over a signed distance function. We can now find successful policies for the two-fingered reorientation task, if the fingers start close to the object on opposite sides. However, this method still fails during all our experiments with multi-fingered hands. We now try our proposed method. It finds reasonable policies in less than five minutes for all tasks. By randomizing initial object poses, we can learn policies that adapt to the pose of the object (see figure 4). If we use traditional back propagation instead of our matrix-free Gauss-Newton method, progress is considerably slower (see figure 5).

V. Conclusion

We can overcome local minima and learn dexterous manipulation tasks via single-level gradient-based optimization by letting the neural network generate contacts and optimizing for physical consistency. The equations can be solved efficiently by generating linearizations through program transformation and performing matrix-free Gauss-Newton steps. Our prototype implementation learns dexterous manipulation tasks within minutes on a single CPU core.

References

[1] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proc. Eurographics*, pages 137–144, 2012.

[2] I. Mordatch, E. Todorov, and Z. Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics*, 2012.

[3] OpenAI. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 2018.

[4] P. Ruppel, N. Hendrich, and J. Zhang. Direct policy optimization with differentiable physical consistency for dexterous manipulation. In *ROBIO*, 2021.