Parametrization of Behavior Trees for Industrial Assembly Skills through Reinforcement Learning

Mayr, Matthias; Krueger, Volker

2019

Document Version:
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Mayr, M., & Krueger, V. (2019). *Parametrization of Behavior Trees for Industrial Assembly Skills through Reinforcement Learning*. Paper presented at SAIS 2019, 31st Workshop of the Swedish AI Society , Umeå, Sweden.

Total number of authors:
2

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Parametrization of Behavior Trees for Industrial Assembly Skills through Reinforcement Learning

Matthias Mayr* and Volker Krueger*

Department of Computer Science, Lund University, Lund, Sweden
{matthias.mayr,volker.krueger}@cs.lth.se

Keywords: Reinforcement Learning · Industrial Robotics · Assembly · Skills · Simulation

Problem Statement

It is an open question how to model complex robot skills in a simple but effective way. Behavior Trees (BT) are a hierarchical approach, that can be used to describe behaviors in a human readable way. Extended to also include the effects on the world state, they can also be used for task modeling [1].

In [2] BTs are combined with motion generators to model parametric robot movements for assembly tasks, that can require both accurate positioning and force-reactive execution. But whereas their representation is easier to understand and a structure can potentially be easily defined using a graphical user interface, choosing an appropriate parametrization is either tedious manual work or requires complex reasoning. Parameters in BTs are conditions for nodes and branches that model the execution, including recovery behaviors like a shake motion to solve peg in hole problems with inaccuracies. Additionally, there are motion generator parameters like attractor points on the trajectory, stiffness and damping. For example in the the experiment in [2], which solves a peg in hole problem with inaccuracies in millimeter-scale, more than 30 parameters had to be specified by hand. Especially when contact forces come into play it can be close to impossible to do this without a lot of trials and supervised real robot experiments. The goal is therefore to use reinforcement learning to learn both the policy parameters and the conditions for branches.

Approach and Research Questions

The use of reinforcement learning (RL) with real robot tasks however is challenging since especially model-free methods require long training times. In contrast to that, model-based methods are more time- and data-efficient [3] by several magnitudes. But this often has constraints regarding the policy representation (e.g. being differentiable) or the reward function (e.g. Euclidean distance). Black-box data-efficient policy search for robotics (Black-DROPS) is a model-based RL

* This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

approach that lifts those restrictions while being as data-efficient as a state-of-the art algorithm PILCO [4] and taking advantage of parallel computing to be as fast as analytical approaches [3]. Furthermore, in [5] it has been extended to scale to high dimensional state and action spaces as they typically appear in robotics.

In order to reduce real robot runtime even further and to reduce cost as well as wear of both production material and the robot, a sample trajectory will be demonstrated and the problem will be solved in simulation first. Leveraging those, some parts of the parametrization like free space movements can easily be determined in simulation, while for example contact forces are hard to model accurately. Therefore, uncertain parameters like friction coefficients can be subjected to a domain randomization as utilized in [6] to find a robust policy, that can handle both the real values and deviations of those.

This work aims to find an appropriate parametrization of a policy represented by a BT. The focus of a parallel PhD project lies on finding suitable BT structures that also define the parameters that need to be learned.

More specifically we want to explore these research questions:
1. Can a robot system automatically learn the parameters of a BT to reliably and robustly execute a robot skill? This includes using recovery behaviors where necessary. 2. Can model-based RL profit from being solved in simulation first and how does domain randomization lead to more robust policies? 3. Can the learning process be shortened or guided by incorporating knowledge about the task? E.g. maximum forces that can be applied to certain parts. 4. Can methods and techniques be developed that allow a transfer of a converged learning process to other tasks and other robots?

References

1. Rovida, F., Grossmann, B., Krueger, V.: Extended Behavior Trees for Quick Definition of Flexible Robotic Tasks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
2. Rovida, F., Wuthier, D., Grossmann, B., Fumagalli, M., Krueger, V.: Motion Generators Combined with Behavior Trees: A Novel Approach to Skill Modelling. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
3. Chatzilygeroudis, K., Rama, R., Kaushik, R., Goepp, D., Vassiliades, V., and Mouret, J.: Black-Box Data-efficient Policy Search for Robotics. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
4. Deisenroth, M.P., Fox, D., Rasmussen, C.E.: Gaussian Processes for Data-Efficient Learning in Robotics and Control. In: 2015 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).
5. Chatzilygeroudis, K. and Mouret, J.: Using Parameterized Black-Box Priors to Scale Up Model-Based Policy Search for Robotics. In: 2018 IEEE International Conference on Robotics and Automation (ICRA)
6. OpenAI, Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., Zaremba, W.: Learning Dexterous In-Hand Manipulation. arXiv:1808.00177 [cs, stat]. (2018).