# Learning to Hop using Guided Policy Search on Real Robot Hardware

Julian Viereck, Felix Grimmiger, Alexander Herzog, Ludovic Righetti
{firstname.lastname}@tuebingen.mpg.de
Autonomous Motion Department, Max Planck Institute of Intelligent Systems, Germany

## I. Motivation / Background

Today, many controllers for humanoid robots are manually fine tuned and require special adjustments to each robot. Adapting robots to different, fast changing environments and providing a robust controller for all scenarios a robot might face is challenging. Because the controllers are tuned to a specific hardware setup, this makes it hard to fast iterate on the hardware and change it.

This is different from animals and humans, which are able to learn fined tune motions from sensor data. Motivated by this, we are interested to see if current reinforcement learning techniques can be used to learn dynamic locomotion directly from data. We are also interested in understanding the inter-play between learning and the mechanical system. For instance, are there mechanical system on which learning is easier? Is there a leg design that can speed up learning?

To help us answer these questions, we build a 3D printed, torque controlled robot leg specially suited for running learning algorithms: The robot is easy to repair and can run for hours without manual reset. For the learning, we build on the Guided Policy Search (GPS) algorithm [1], which has been very successful at learning manipulation tasks for real robots recently. It was also shown that simulated locomotion could also be learned [2]. The algorithm has several interesting features including sample efficiency, it uses a combination of optimal control with learned local dynamic models, and it can generalize from local policies by using deep neural networks.

To our knowledge, the GPS framework has never been applied on real legged robots for locomotion tasks. In our work, we use the GPS framework to learn a hopping motion for a legged robot. In this simple setup, we aim to see if the algorithm can deal with contact rich hybrid dynamics. We are also interested to experimentally study the influence of sensory feedback (e.g. force sensing) and the mechanical leg design on the learning algorithm.

## II. Experiments & Hardware

In the following we present a brief overview on the GPS algorithm, what experiments we applied it to, and the real robot hardware we use to run experiments at the moment.

### A. GPS Algorithm

The GPS framework optimizes a set of local, task specific policies and uses them to learn a neural network policy. In our case, we optimize local policies for different initial configurations of our robot (height above the ground, leg position). The neural network policy learns not only to solve the tasks of the local policies but also how to solve new unseen tasks.

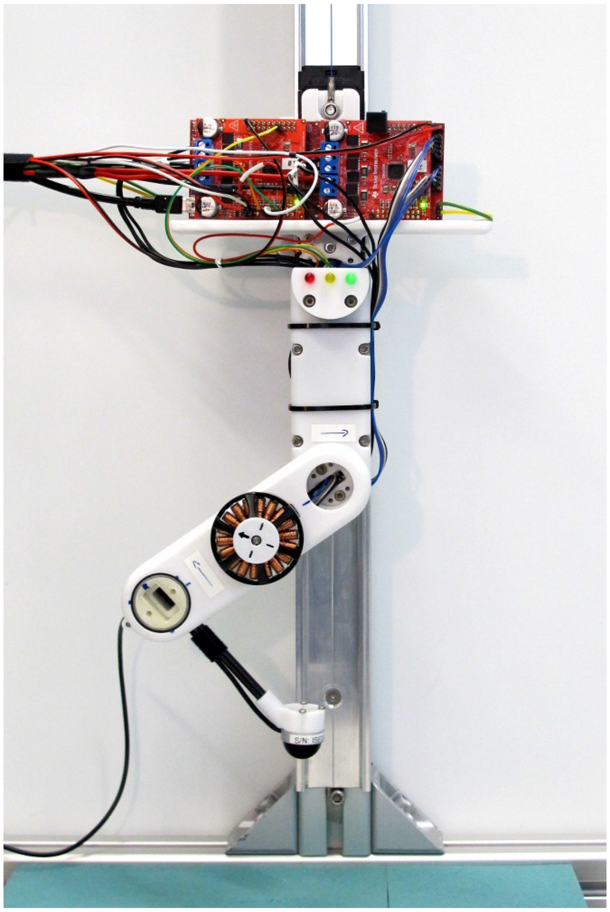

Fig. 1. 3D printed, 1D hopper robot.

To optimize the local policies, the GPS framework learns a local, linear model of the system dynamics along the policy's trajectory. For this, perturbed rollouts along the nominal policy trajectories are recorded and the local dynamics model parameters are fitted to the rollouts. A learned prior model helps to reduce the number of rollouts to get a good dynamics model. Once the local dynamics model is fitted, the local policies are optimized using iLQR with respect to a cost function. Eventually, the neural network policy is trained using the information from the local policies.

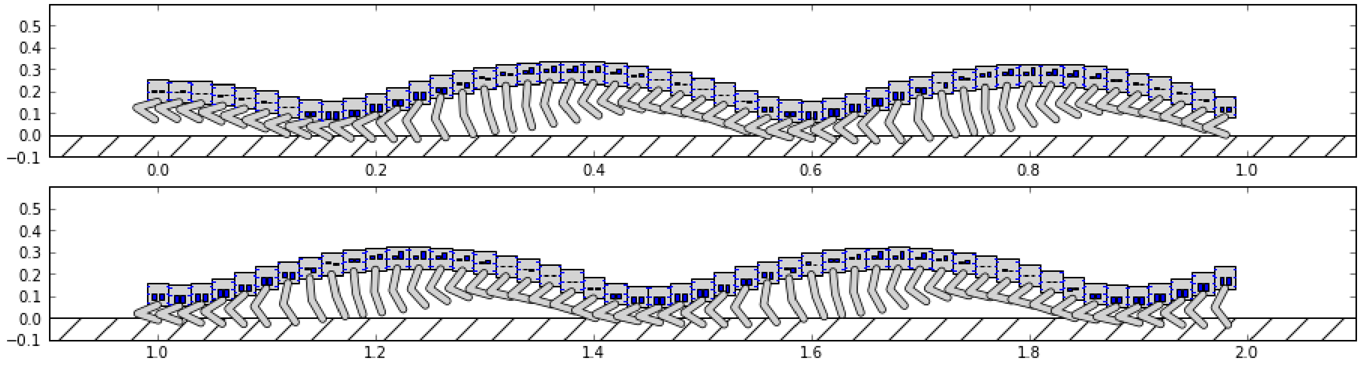

Fig. 2. Rollout of the neural network policy on the simulated hopper. The plot shows snapshots of the hopper over a 2 second trajectory.

The GPS framework modifies the optimization objective of the local policies and neural network policies such that they stay close to each other during optimization. This makes the local and neural network policy converge to the same behavior at the end of the optimization.

### B. Experiments

To see if the GPS framework can learn policies for contact-rich locomotion tasks, we aim to learn a hopping motion on robot with a single leg. The base of the robot is restricted to move along the verticle axis with fixed rotation axis. The robot has a torque controlled revolute joint for the hip and another one for the knee. In this abstract we present the results from simulation. We are currently testing the algorithm on a real robot. For the simulation, we model the forces acting on the robot using momentum preservation to get a realistic contact behavior.

Our learning procedure is as follows: we use two local policies. In the first policy the robot starts in a configuration high above the ground and in the second one the robot is closer to the ground. We use two cost functions to describe the hopping motion. One describes how to extend the leg of the robot when it is in contact with the ground and the other cost function describes the desired leg position for landing on the ground again. We begin the learning process by learning the system dynamics and optimizing the local policies. After five optimization iterations, we also start learning the neural network policy.

At the end of the optimization, the neural network policy is capable to reproduce the hopping trajectories from the local policies. The results are shown in Figure 2. In addition, we also analyze its stability with respect to noise applied to output torques.

In summary, we managed to learn a hopping motion using the GPS framework. This is remarkable, given the switching, contact rich, and non-linear system dynamics.

### C. Real Robot Hardware

We are now applying the same GPS learning framework on a real robot hardware. To support learning on a real robot hardware, we designed our custom 3D printed robot leg using cheap and easy to repair hardware components. Having this testbed is important as policies learned using reinforcement learning are often unstable and have aggressive exploration techniques with the potential to damage the robot. As reinforcement learning algorithms often need many rollouts, the hardware is designed to reset itself and run for hours without manual interaction.

To achieve this, we build the robot leg as shown in Figure 1. We restrict the motion along the vertical axis using a slider. The leg is equipped with two torque controlled brush-less motors - one for the hip and one for the knee joint. The last end-effector is equipped with a three axis force sensor to detect contact with the ground. In addition, we measure the hip-height above the ground, and the joint angles as well as joint velocities.

## III. Future Work

Using the GPS framework, we managed to learn a hopping motion in simulation. As next step, we started to apply the same framework on real hardware. More concrete, we want to investigate as future work how sensor feedback influences the learning, test the learning algorithm on different leg designs (e.g. legs with series elastic actuation), and apply the framework to a quadruped robot.

## References

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[2] S. Levine and V. Koltun. Guided policy search. In *ICML (3)*, pages 1–9, 2013.