Curricular Policy Search for Quadruped Jumping

Gautam Salhotra, Peter Englert, Gaurav S. Sukhatme

***Abstract*—In this work, we explore if curriculum learning can improve the performance of trajectory-based policy search on legged locomotion tasks. We evaluate a curricular policy search method that integrates the black-box optimization algorithm CMA-ES (Covariance Matrix Adaptation Evolution Strategy) into a curriculum learning framework that incrementally increases the task difficulty. We assume a deterministic episodic scenario and directly parameterize the action trajectory with target points of a radial basis function network. We show preliminary results on two simulated jumping tasks where the robot has to jump over tall objects. The curriculum starts with a flat terrain and slowly increases the height of the object in each iteration. The results indicate that this multi-stage learning scheme can find jumping motions that could not be found via traditional optimization techniques.**

I. Introduction

Legged locomotion tasks like walking, running, and jumping are challenging problems in robotics, which have been addressed in various ways. A common technique is nonlinear optimal control that optimizes a cost function with respect to dynamics constraints [1]. There exists a wide range of methods like indirect optimal control where only the actions are used as optimization variables or direct methods that also include the state trajectory. They produce locally optimal trajectories and achieve low computation times by using gradient-based optimization techniques. However, the performance of these methods strongly depends on the initial trajectory, which is difficult to obtain for complex tasks. Another approach to address locomotion problems is policy search [2], which computes a feedback control policy that maximizes a given utility function. Advantages of these methods is that they typically do not require gradients of the cost function and can handle high-dimensional inputs such as images. They require large amounts of samples and the resulting motions are typically not as smooth in comparison to optimal control. Evolutionary strategies like CMA-ES [3] are a certain class of policy search methods that use a single fitness evaluation of an episode. In this work, we use CMA-ES in a curriculum learning setting [4] that divides the learning into multiple stages where the difficulty of a task is incrementally increased in order to accelerate or improve learning.

II. Curricular Policy Search

We consider episodic tasks where the robot starts at an initial state x_0 and has to reach a goal configuration g. The episode terminates after a specified duration T. Our curricular policy search takes an environment and target task as input. It then creates a series of M interpolated environments between the base task (easiest to solve) and the target task (hardest to solve). This series serves as the curriculum for our learning method. The interpolation can capture various aspects of the environment, such as contact parameters (a task might be easier to solve with low friction), obstacle size (smaller obstacles might be a good stepping stone towards bigger obstacles), actuation limits (higher actuation limits make it easier to complete a tasks), etc. The idea is to use the solution of the previous task as initialization for the next stage.

Our method starts by applying CMA-ES on the base task. In each iteration of CMA-ES, a set of policy parameters is sampled based on the current distribution. For each parameter, the corresponding policy is applied on the system and a utility value is assigned to the observed state trajectory. Based on these values, CMA-ES updates its distribution to find the policy parameters that maximize the utility function. After CMA-ES converges, the difficulty of the environment is increased by one step and the mean of the next CMA-ES run is initialized to the best solution of the previous stage. This procedure is repeated until the optimization fails or the target task is reached.

We parameterize the action trajectory $u(t)$ of the robot with a set of N target points $\mathbf{u} = (u_1, u_2, \ldots, u_N)$ that are interpolated with a radial basis function network [5]. The points are evenly distributed along time $\mathbf{t} = (t_1, t_2, \ldots, t_N)$. The action at time t is then computed as the sum of the radial basis functions $u(t) = \sum_{i=1}^{N} w_i \phi(|t - t_i|)$ where we use a multiquadric kernel $\phi(t) = \sqrt{(t/\epsilon)^2 + 1}$. The weights $w = (w_1, \ldots, w_N)$ are estimated via linear least squares based on the current target points $(\mathbf{t}, \mathbf{u})$ such that $u(t_i) = u_i$. The parameter ϵ is set heuristically to the average duration between two points T/N. This policy parameterization reduces the number of optimization parameters and leads to a smooth action trajectory. Applying an action trajectory $u(t)$ on the system results in a corresponding sequence of states $(x_1, x_2, \ldots, x_N)$ that we use to specify a utility function $\mathcal{J}(\mathbf{u})$. In our experiments, we use the utility function $\mathcal{J}(\mathbf{u}) = -\sum_{t=1}^{N} ||x_t - g||$ that measures the negative distance of the robot state to the goal.

III. Experiments

We show experiments for quadruped jumping on two simulated environments. The first environment is the two-dimensional half-cheetah environment of OpenAI gym [6]. The half-cheetah robot has two legs, each with three degrees of freedom. Figure 1 shows a rollout of an optimized action trajectory given by CMA-ES on the target task. The second environment is a Laikago quadruped (Figure 2) by Unitree

The authors are with the Robotics and Embedded Systems Lab (RESL) at the University of Southern California. Correspondence to salhotra@usc.edu

Gaurav Sukhatme holds concurrent appointments as a Professor at USC and as an Amazon Scholar. This paper describes work performed at USC and is not associated with Amazon.

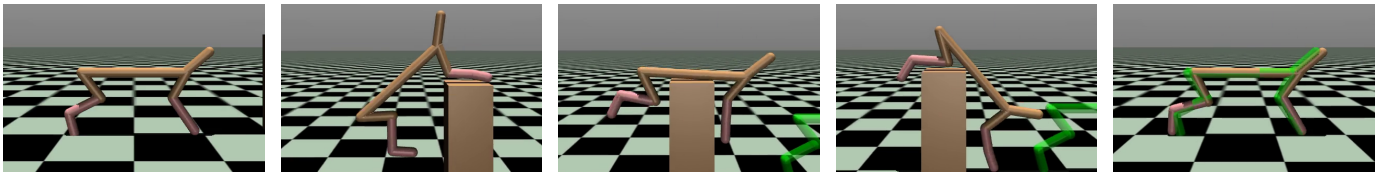

Fig. 1: Screenshots of the half cheetah jumping task. The goal location is shown in green. The obstacle is taller than the cheetah itself, so it first manages to jump on to the obstacle. Then, it uses its momentum and hind legs to go over the obstacle, before finally landing in the goal position.

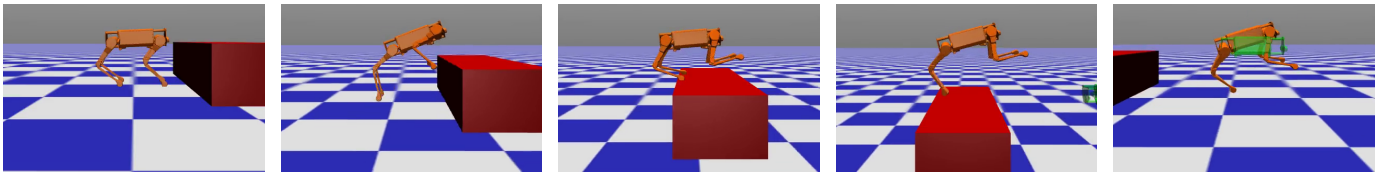

Fig. 2: Screenshots of the Laikago jumping task. The goal location of the torso is shown in green. The obstacle is almost as large as the robot itself, as seen in the first frame. The laikago jumps onto the obstacle, and performs a second jump to hop off of it, before its torso reaches the green goal position.

Robotics. The Laikago quadruped has four legs with three degrees of freedom each. In this experiment, we reduce the action space by mirroring the actions between the left and right legs. Thus, our action dimension reduces to six at each timestep. The full state dimension is 19, which includes 12 joint angles, three states of the torso position, and four elements of the quaternion. The duration of an episode is $T = 4$ seconds and we use $N = 20$ target points that parametrize the action trajectory. The robot controller runs at 500Hz. Our method creates a curriculum of $M = 9$ interpolated environments between a flat terrain and the target task. The box height is increased by 0.05m between two successive environments. In both experiments, the final box is nearly as tall as the robot itself. We set the number of CMA-ES iterations to 1000 for each stage, which takes about 9 minutes on 2 Intel Xeon Gold processors and 256GB RAM. We compare our method to a baseline that applies CMA-ES directly on the target task.

Figure 3 shows the learning curves of our policy search method across all stages. As it can be seen, the baseline method is not able to solve this task with the same amount of iterations. The trajectories for the baseline solution show the robot failing to jump above the box. Independent experiments show that the baseline could not even solve the task corresponding to stage 6. We were unable to solve the task with a direct trajectory optimization. Our method solves the task because the curricular approach provides the next stage with good initializations. Note the drop in the utility as we take the optimized values from the previous stage and apply it to the next stage. This implies that the solution of the previous stage does not directly work on the next stage. Hence, the optimized values serve as a mere initialization for the optimization at the current environment. Also note that the maximum utility achieved in a stage decreases with more difficult tasks. This shows that the environments are getting more difficult to solve and that the length of the motion from start to goal increases.

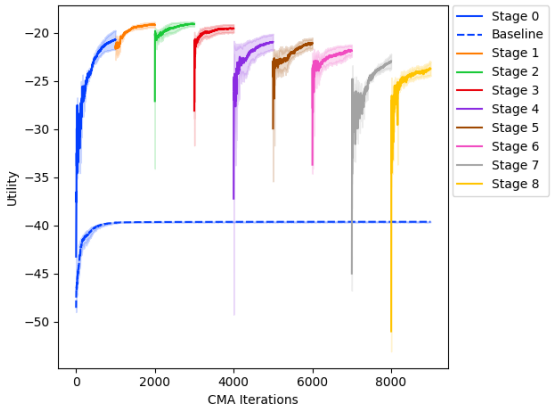

Fig. 3: Utility function over multiple stages. Dashed line indicates the baseline, solid line is our curricular policy search.

IV. Conclusion

We presented preliminary results of applying CMA-ES in a curriculum setting for quadruped jumping motions. Future work includes extending this method to tasks beyond jumping, creating a state-based policy with feedback control, and combining it with nonlinear optimal control to get smoother motions. We also intend to do further comparison of our approach to deep reinforcement learning and nonlinear optimal control.

References

[1] Arthur E Bryson and Ho Yu Chi. *Applied optimal control: optimization, estimation, and control*. Blaisdell Pub. Co., 1969.

[2] Olivier Sigaud and Freek Stulp. Policy search in continuous action domains: an overview. *Neural Networks*, 113:28–40, 2019.

[3] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[4] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[5] David S Broomhead and David Lowe. Radial basis functions, multivariable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv:1606.01540*, 2016.