Model-based Evolutionary Policy Search for Skill Learning in Continuous Domains

Jan Hendrik Metzen, University Bremen, Robotics Group, Robert-Hooke-Str. 5, 28359 Bremen, Germany

① Abstract

We investigate the utility of evolutionary policy search (EPS) for skill learning in a Hierarchical Reinforcement Learning architecture. While EPS is well suited for domains with **continuous state and action spaces**, it is susceptible to **non-stationarities** caused by concurrent learning of several skills and higher layers of the architecture. We hypothesize that (if **model-learning** is feasible) EPS should be used instead for **planning** based on trajectory sampling in a model which "smooths out" non-stationarities.

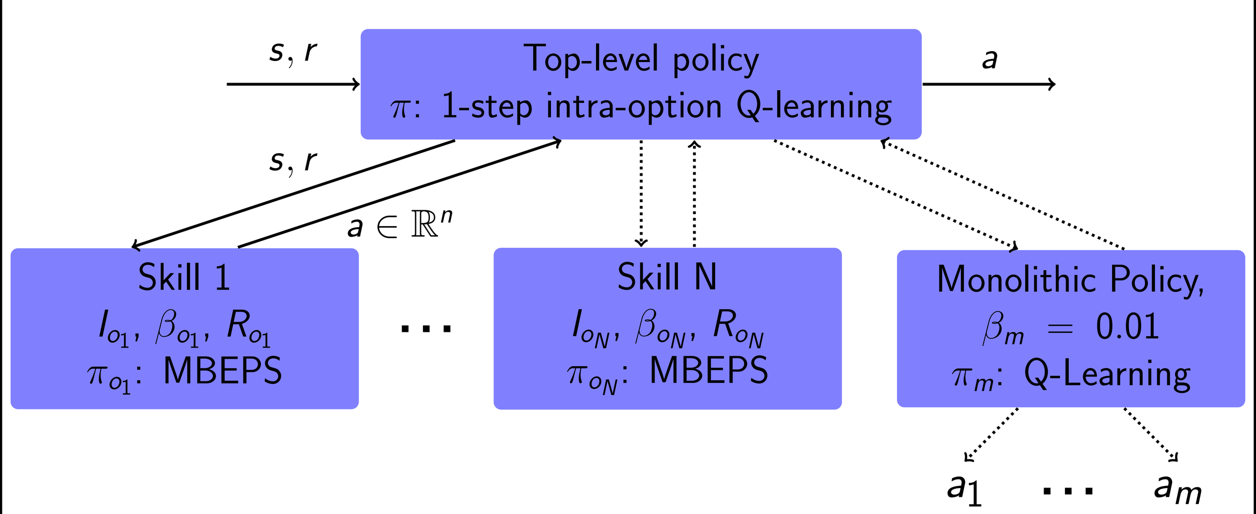
② Evolutionary Policy Search (EPS)

- Parametrized policy representation $\pi(\theta)$
- Population-based approach where each individual encodes parameter vector θ
- **Expected return of** $\pi(\theta)$:
  $J(\theta) = E[R(s_0)|s_0 \sim S_0, P^a_{ss'}, R^a_{ss'}, \pi(\theta)]$
- Fitness function $f(\theta)$ is sample-based approximation of $J(\theta)$
- Evolution Strategy used to find θ which maximizes $f(\theta)$

③ Using EPS for Skill Learning

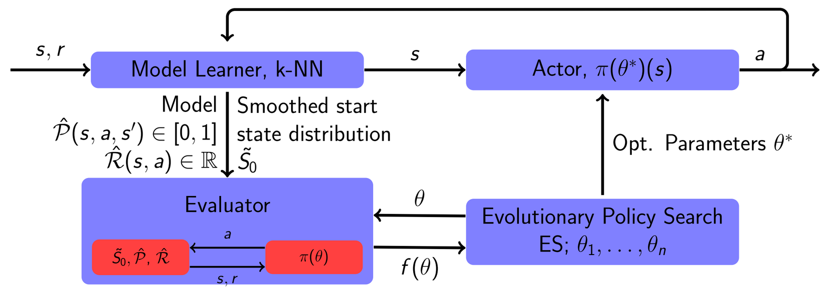
- **Motivation**: EPS is well-suited for problems with continuous state and action spaces [1, 2, 4]
- **Challenge**: Structured problems may require sophisticated policies with high dimensional parameter vector
- **Approach**: Hierarchical RL with autonomous skill discovery allows to split complex problems into simpler subproblems. For these subproblems, simple (e.g. linear) policies with a small number of parameters may suffice

Hierarchical RL Architecture



- **Challenge**: Concurrent learning of several skills and higher layers of the architecture makes option's start state distribution S_0 non-stationary
- **Approach**: Derive policy by planning (trajectory sampling) in learned model of environment in option's initiation set. Use EPS for planning in model; the start state distribution can be kept stationary during one planning iteration (i.e. one generation)

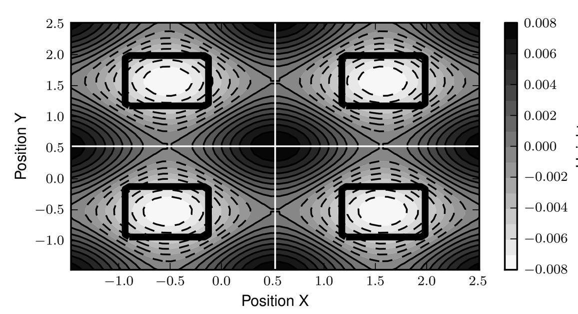
④ Model-based EPS



- Learn model of environment, e.g. using k-nearest neighbors (k-NN)
- Set $\tilde{S}_0$ to all states in which the skill was invoked recently
- $(\mu + \lambda)$ Evolution Strategy with self-adaptation of mutation strength σ
- Compute fitness $f(\theta)$ based on n_t trajectories sampled from the learned model while following $\pi(\theta)$
- Alternate between model-learning and policy improvement

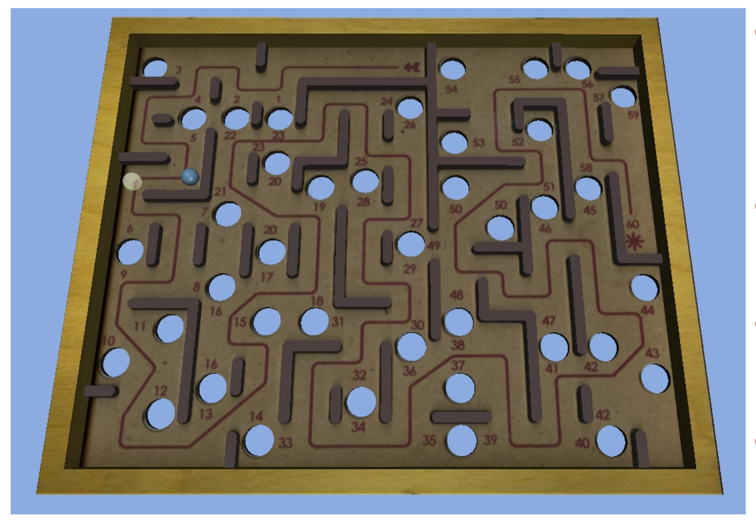
⑤ Scenarios

Multi-Valley Environment

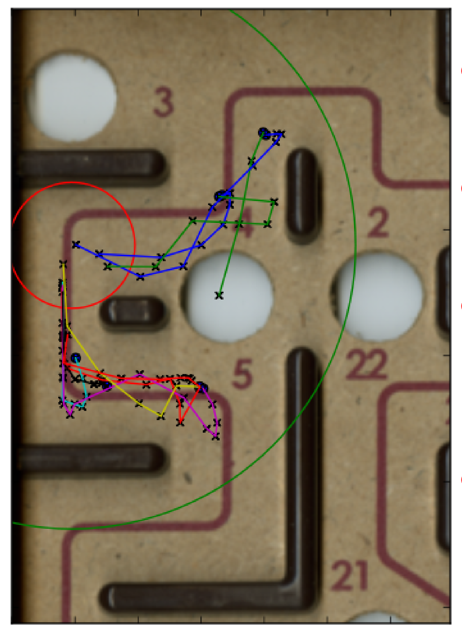


- 2D extension of Mountain Car with 4 valleys
- 4 tasks: reach bottom of certain valley with small velocity
- 5 state dimensions (position, velocity, task) and 2 action dimensions (accelerations)
- Skill Discovery using FIGE + OGAHC [3]

Skill Learning in Labyrinth Environment



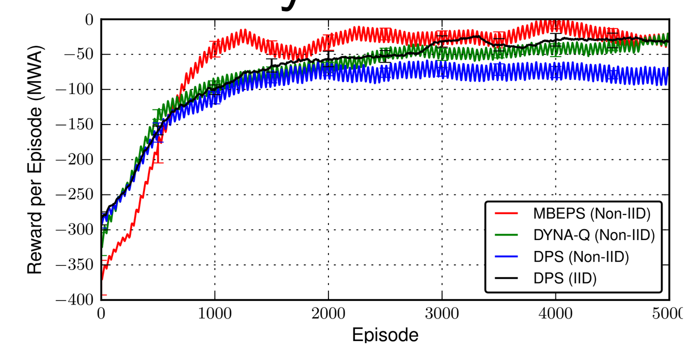
- 6 state dims: ball position (x, y), ball velocity (v_x, v_y), and board angle (ϕ_x, ϕ_y)
- 2 action dims: changes of board angles $(\triangle\phi_x, \triangle\phi_y)$, every 100ms
- Open Dynamics Engine (ODE)-based simulation
- Physical system with sensors and actuators exists



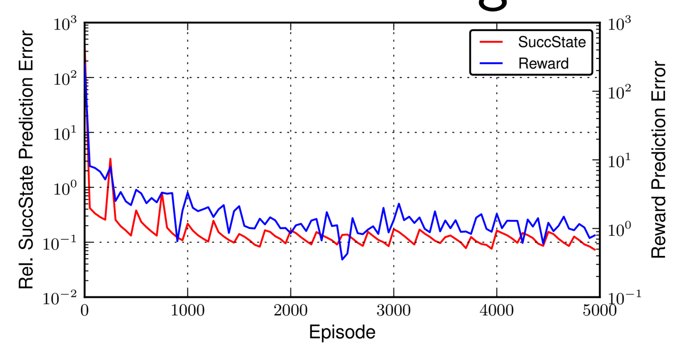
- Learning a predefined subtask: Reach red circle area from anywhere in green area
- Randomly start from 5 positions with random velocities and board angles
- Reward: -1 per time step, $+250$ for reaching goal area, -250 for leaving initiation set area, and -750 for falling into hole
- Policy parametrization:
  $\triangle\phi_x = 3\tanh(3(w_1|x|x + \sum_{p_0 \in \{-1,1\}^4} w_i \operatorname{rbf}(p_0, p, 0.5))) - \phi_x$
  with $p = (x, y, v_x, v_y)$ and $\operatorname{rbf}(p_0, p, b)) = \exp(-b||p - p_0||_2^2)$

⑥ Results

Labyrinth Environment - Skill Learning



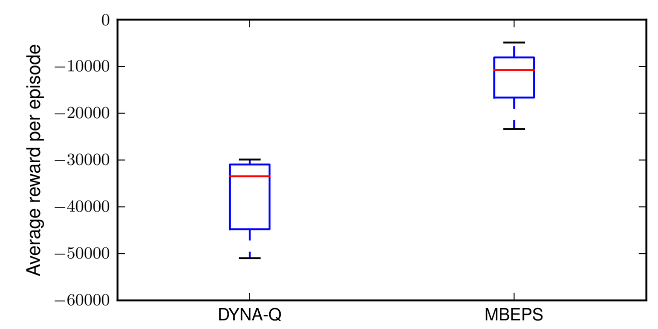
Learning performance



Model Error

- EPS: 10 + 6 Evolution strategy, $\sigma_0 = 0.8$, 25 evaluations per individual
- DYNA-Q: CMAC, 10 tilings, resolution $13^2 \times 2^3 \times 3^2$, $\gamma = 1.0$, $\epsilon = 0.01$, $\alpha = 0.25$
- Model-Learning: deterministic, 3-NN, offline optimization of distance function
- Planning: 5000 episodes trajectory sampling every 250 episodes
- Simulated non-stationarities in S_0 by cycling through 40 possible start states

MultiValley - Architecture



- **EPS**: 5 + 3 ES, linear policy $w_0x + w_1y + w_2v_x + w_3v_y + w_4$, $\sigma = 0.1$, 25 evals per individual
- **DYNA-Q**: CMAC, 5 tilings, resolution 9^4, $\gamma = 1.0$, $\epsilon = 0.01$, $\alpha = 0.75$
- Skill Discovery after 100000 steps; computation of skill policy directly after discovery (10000 sampled episodes per skill)
- Average reward obtained in the first 500 episodes

⑦ Outlook

- Comparison with other policy search approaches
- Using other model learning approaches; systematic exploration (RMax-like)
- Integrate Skill Discovery, Skill Learning, and Compositional Learning and evaluate in entire Labyrinth Environment

⑧ References

[1] Verena Heidrich-Meisner and Christian Igel.
Variable metric reinforcement learning methods applied to the noisy mountain car problem.
In *8th European Workshop on Reinforcement Learning (EWRL 2008)*, pages 136–150. Springer-Verlag, 2008.

[2] Shivaram Kalyanakrishnan and Peter Stone.
An empirical analysis of value function-based and policy search reinforcement learning.
In *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2009.

[3] Jan Hendrik Metzen.
Online skill discovery using graph-based clustering.
In *10th European Workshop on Reinforcement Learning (EWRL 2012)*, 2012.

[4] Julian Togelius, Tom Schaul, Daan Wierstra, Christian Igel, Faustino Gomez, and Jürgen Schmidhuber.
Ontogenetic and phylogenetic reinforcement learning.
Künstliche Intelligenz, (3/09):30–33, September 2009.

Universität Bremen

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
Robotics Innovation Center

Contact:
Jan Hendrik Metzen
University Bremen, Robotics Group
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
jhm@informatik.uni-bremen.de