A Comparison of Policy Search in Joint Space and Cartesian Space for Refinement of Skills

Alexander Fabisch

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Universität Bremen

28th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD 2019)

Outline

1. Motivation - Learning from Human Demonstration
2. Methods
    a. Movement Representation
    b. Approximation of Inverse Kinematics
    c. Policy Search
3. Experiments

Motivation

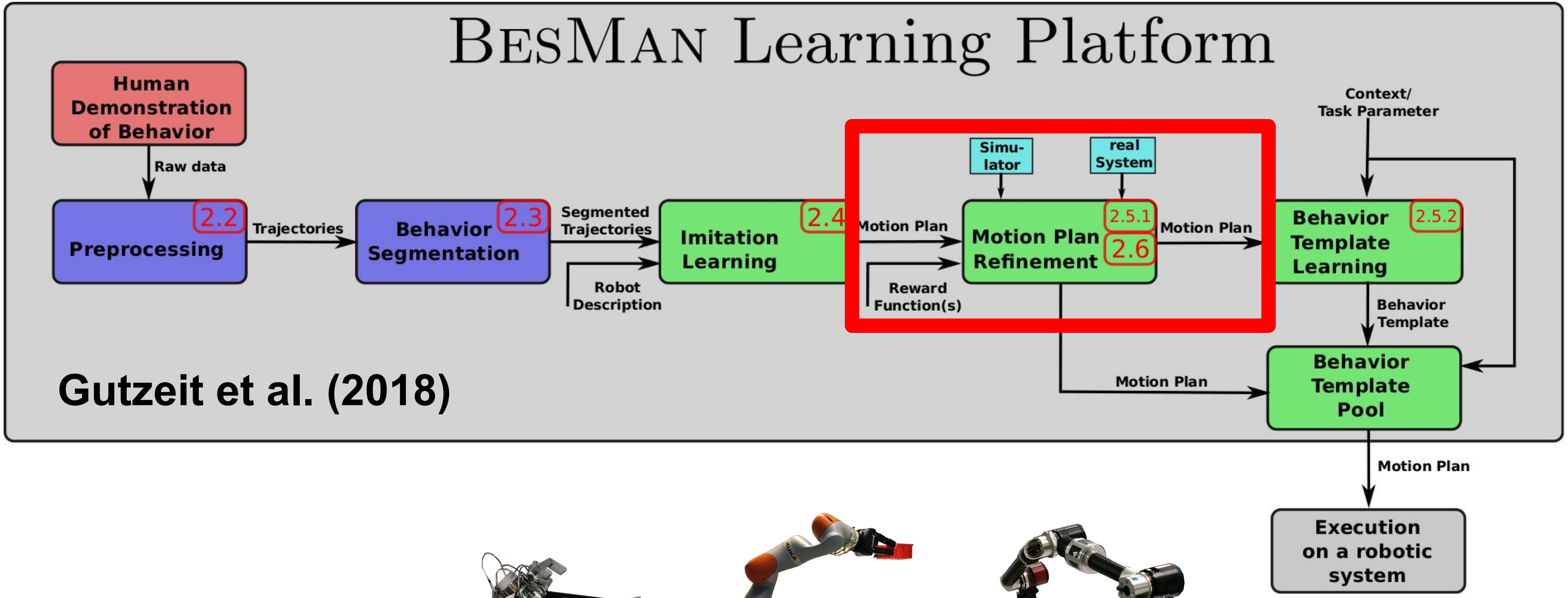
Big Picture - Robot Behavior Generation

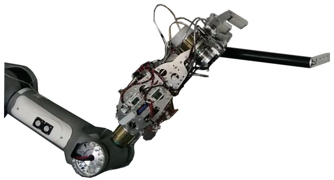
How can we combine prior knowledge with machine learning?

Reinforcement Learning:

- Initialization: imitation learning (programming by demonstration)
- Prestructured policies
- **Maybe (?): Learning in Cartesian space + inverse kinematics**

Motivation - Learning from Human Demonstration

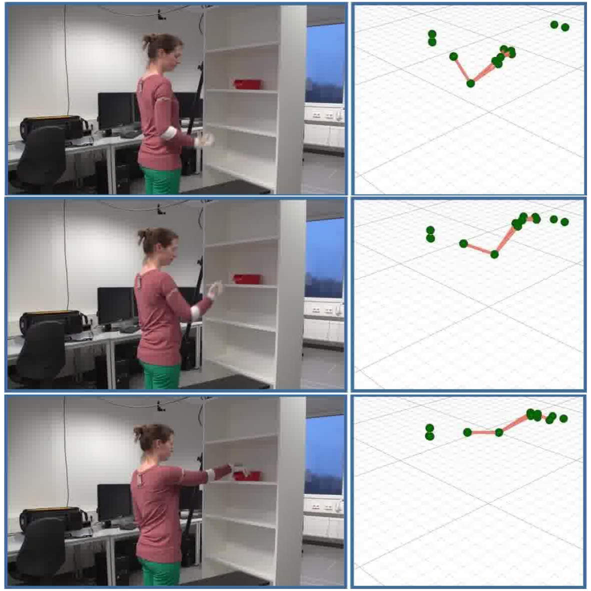




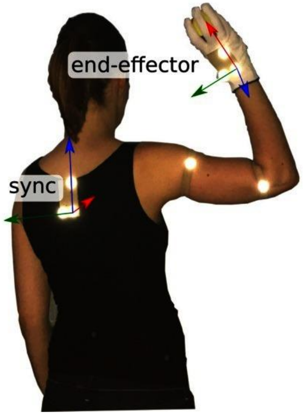
Coroma Deutsches Forschungszentrum für Künstliche Intelligenz GmbH Universität Bremen

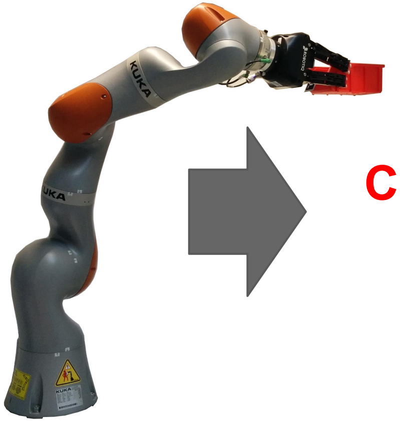
Motivation - Correspondence Problem

Motion Capture



Skill Transfer





Correspondence Problem

Reinforcement Learning

Gutzeit et al. (2018)

Methods

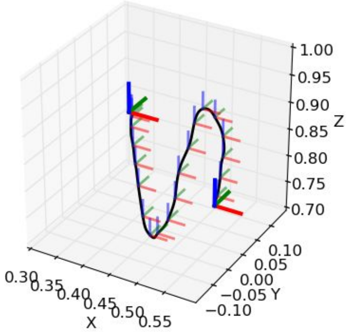
Methods - Prestructured Policy

Dynamical Movement Primitive (DMP)

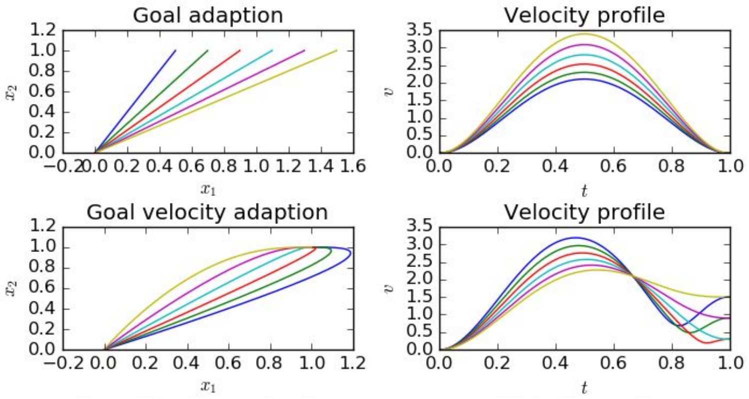
- Adaptable and learnable trajectories
- For imitation and reinforcement learning

Ijspeert et al. (2013); Ude et al. (2014)

3D



2D



Since demonstrations are transferred in Cartesian space, **is it better to refine trajectories in Cartesian space?**

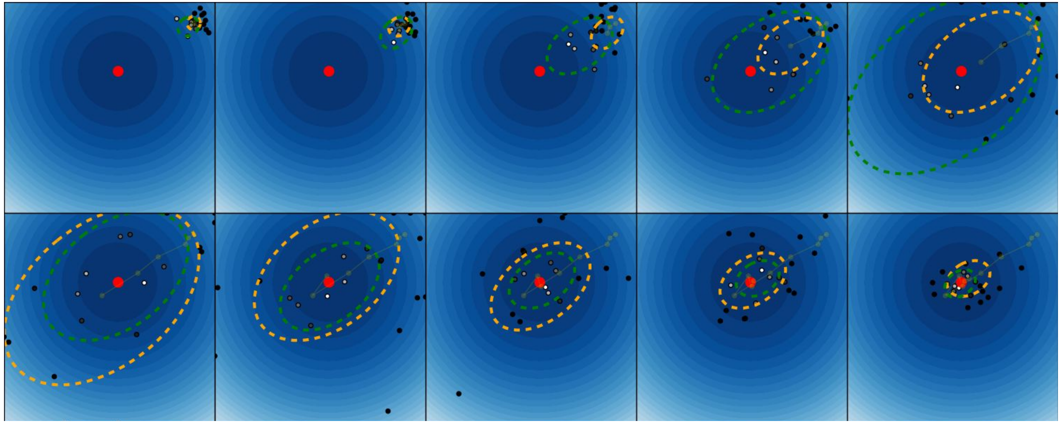
Candidates

We will compare reinforcement learning (policy search) with...

1. Joint space DMP (Joint space)
2. Cartesian DMP + IK that outputs the last reachable pose ("Exact IK")
3. Cartesian DMP + Approximation of IK (Approximate IK)

Methods - Skill Refinement / Policy Search

- Many policy search algorithms work similarly
- In this work we selected CMA-ES **(Hansen and Ostermeier, 2001)**
- CMA-ES only has a few critical hyperparameters
- One of them: initial step size; has been set to not make a difference in joint and Cartesian space



Methods - Numerical Inverse Kinematics

Forward Kinematics

$$f(\boldsymbol{q}_t) = \boldsymbol{p}_t$$

Indirect SQP

(IK in constraints)

$$\arg\min_{\boldsymbol{q}_t \in \mathbb{R}^n} (\boldsymbol{q}_{t-1} - \boldsymbol{q}_t)^T (\boldsymbol{q}_{t-1} - \boldsymbol{q}_t),$$
$$\text{s.t.} \quad g_i(\boldsymbol{q}_t) \leq b_i, \quad i = 1, \ldots, m$$

Direct SQP

(Beeson and Ames, 2015)

$$\arg\min_{\boldsymbol{q}_t \in \mathbb{R}^n} d(f(\boldsymbol{q}_t), \boldsymbol{p}_t^{\text{des}}), \quad \text{s.t.} \ g_i(\boldsymbol{q}_t) \leq b_i, \quad i = 1, \ldots, m$$

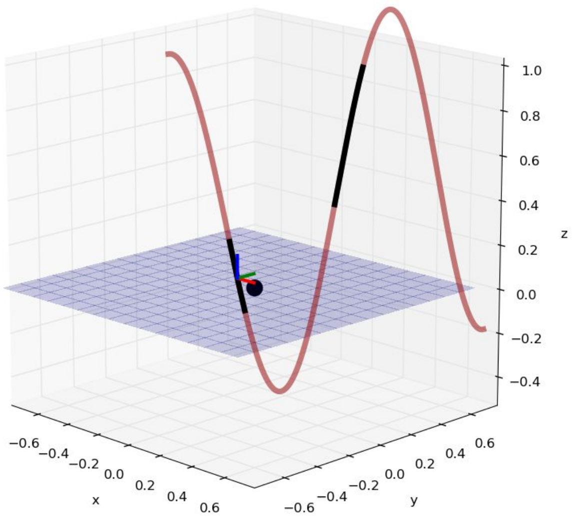
Approximation of IK

quaternion distance

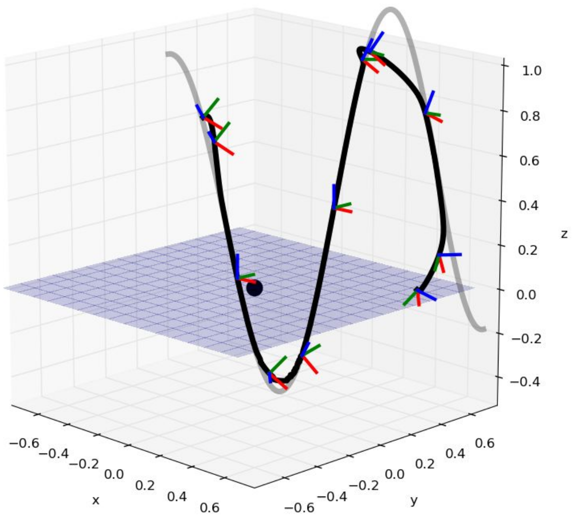
$$d(\boldsymbol{p}_1, \boldsymbol{p}_2) = \underline{w_{\text{pos}}} ||\boldsymbol{p}_{1,1:3} - \boldsymbol{p}_{2,1:3}||_2^2$$
$$+ \underline{w_{\text{rot}}} [\min(|| \log(\boldsymbol{p}_{1,3:7} * \overline{\boldsymbol{p}}_{2,3:7})||, 2\pi - || \log(\boldsymbol{p}_{1,3:7} * \overline{\boldsymbol{p}}_{2,3:7})||)]^2$$

Methods - Approximation of Inverse Kinematics

Red: exact solution not found

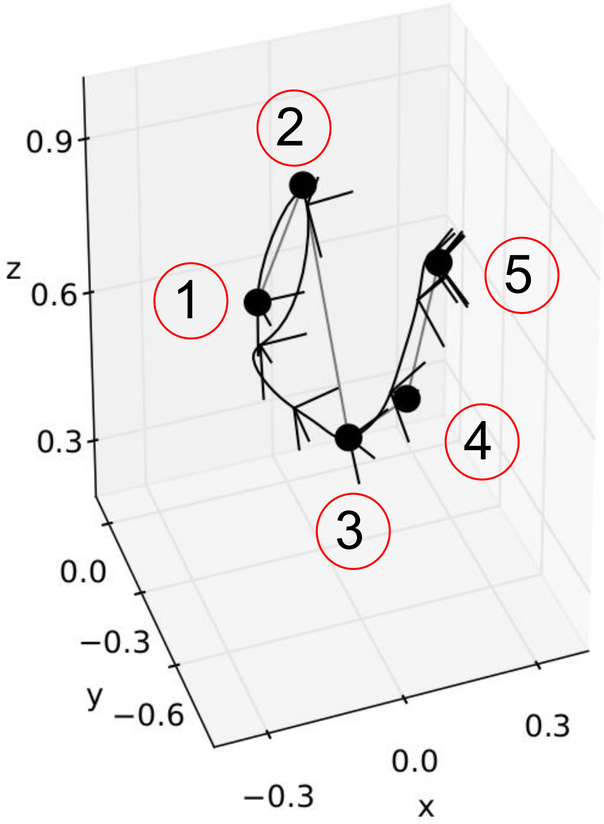


Approximation with low weight for orientation

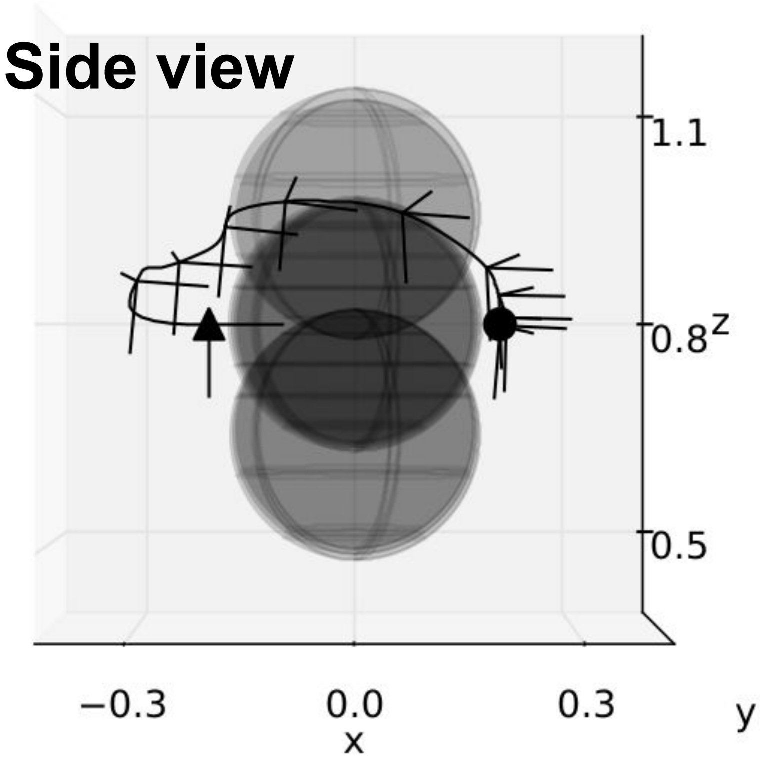


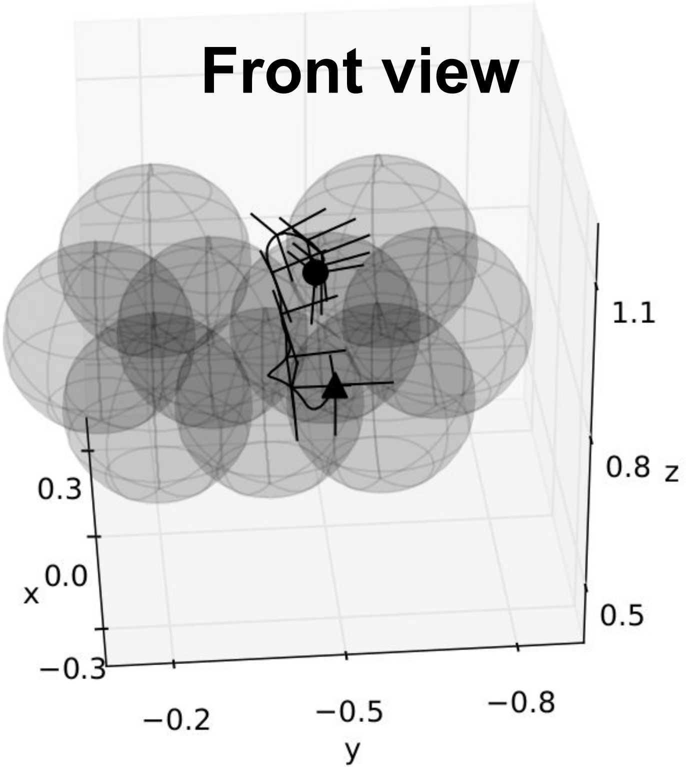
Experiments

Problem 1/4 - Viapoint

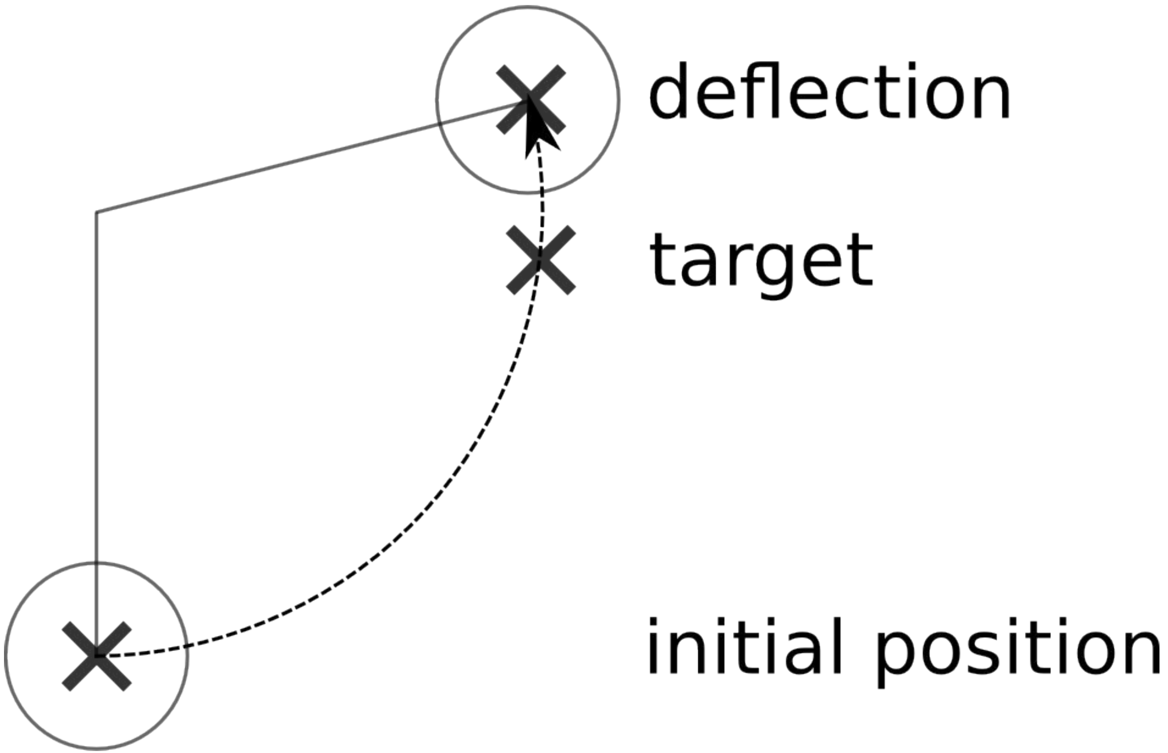


Problem 2/4 - Obstacle Avoidance

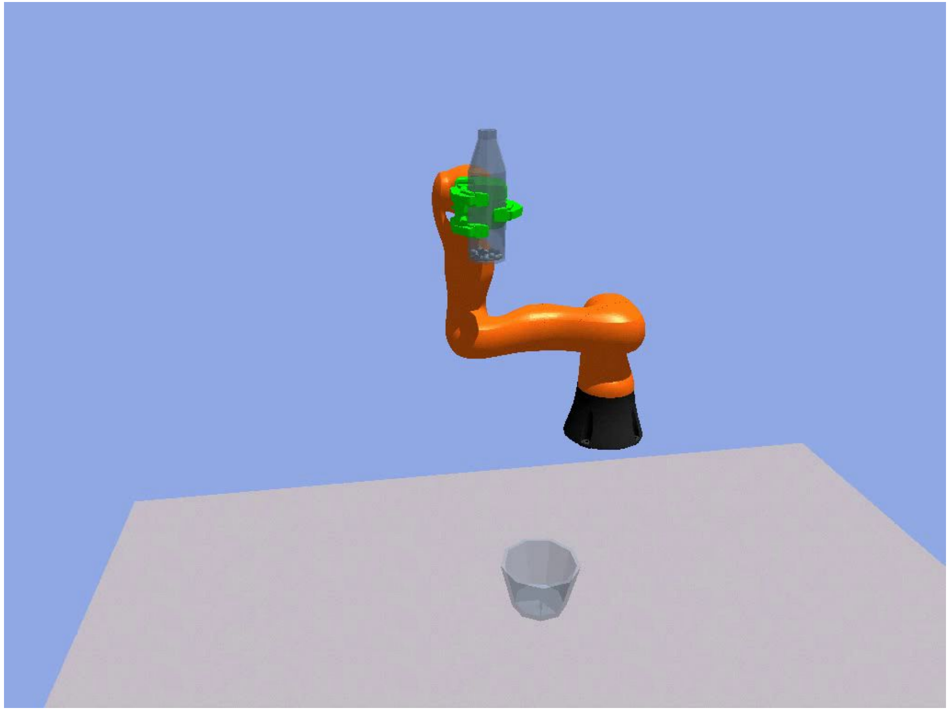




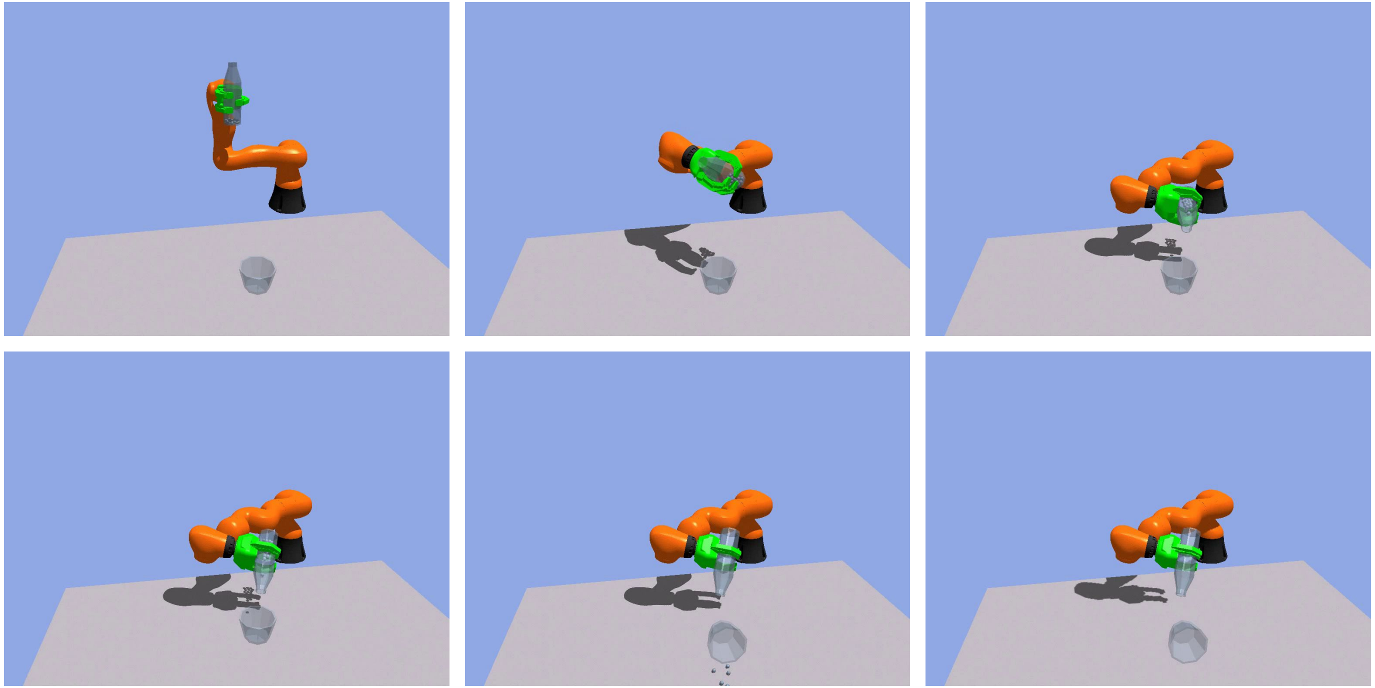
Problem 3/4 - Pendulum



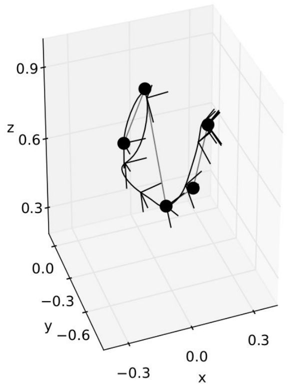
Problem 4/4 - Pouring

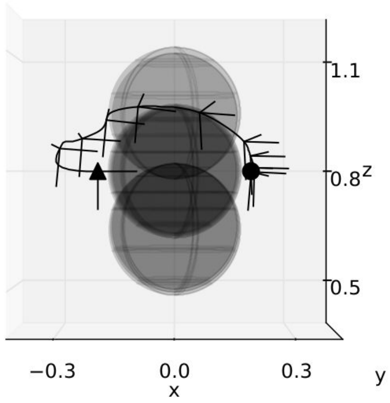


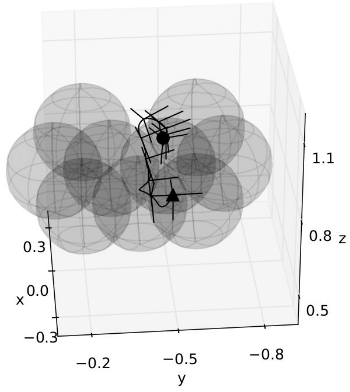
Pouring - Failure (Glass Tips Over)

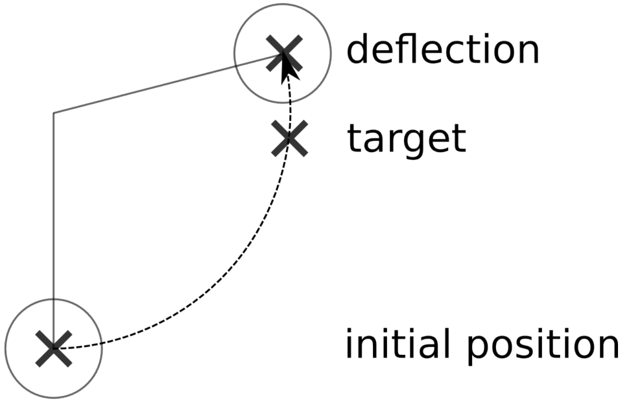


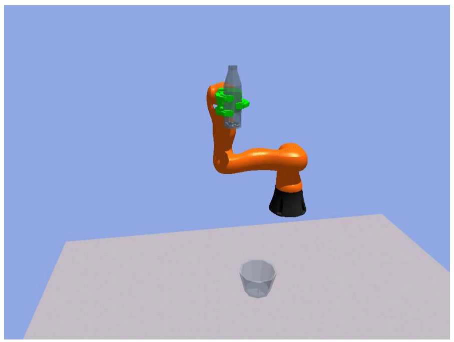
Problems











Difficulty

Coroma

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Universität Bremen

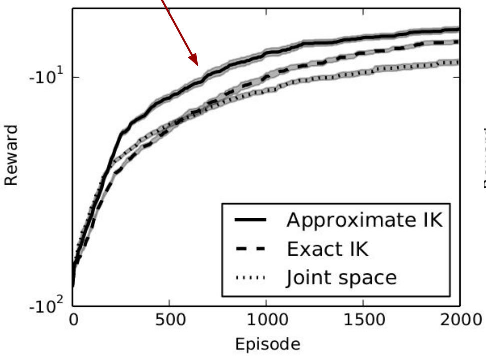
Results

2 Learning in Cartesian space can be faster, but depends on IK

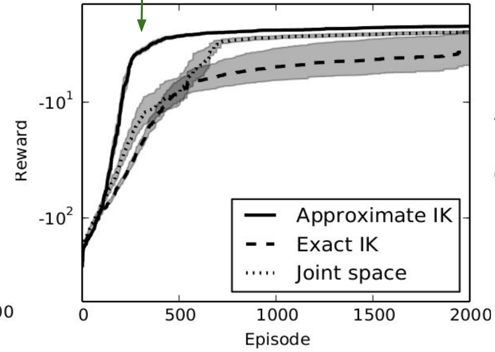
1 Learning in Cartesian space is clearly better

There is no significant difference here 3

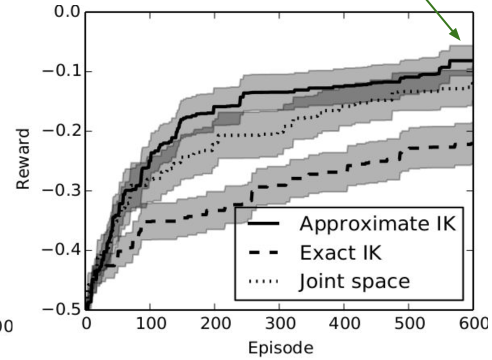
Viapoint



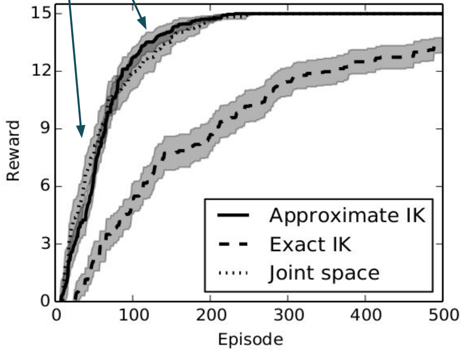
Obstacle



Pendulum

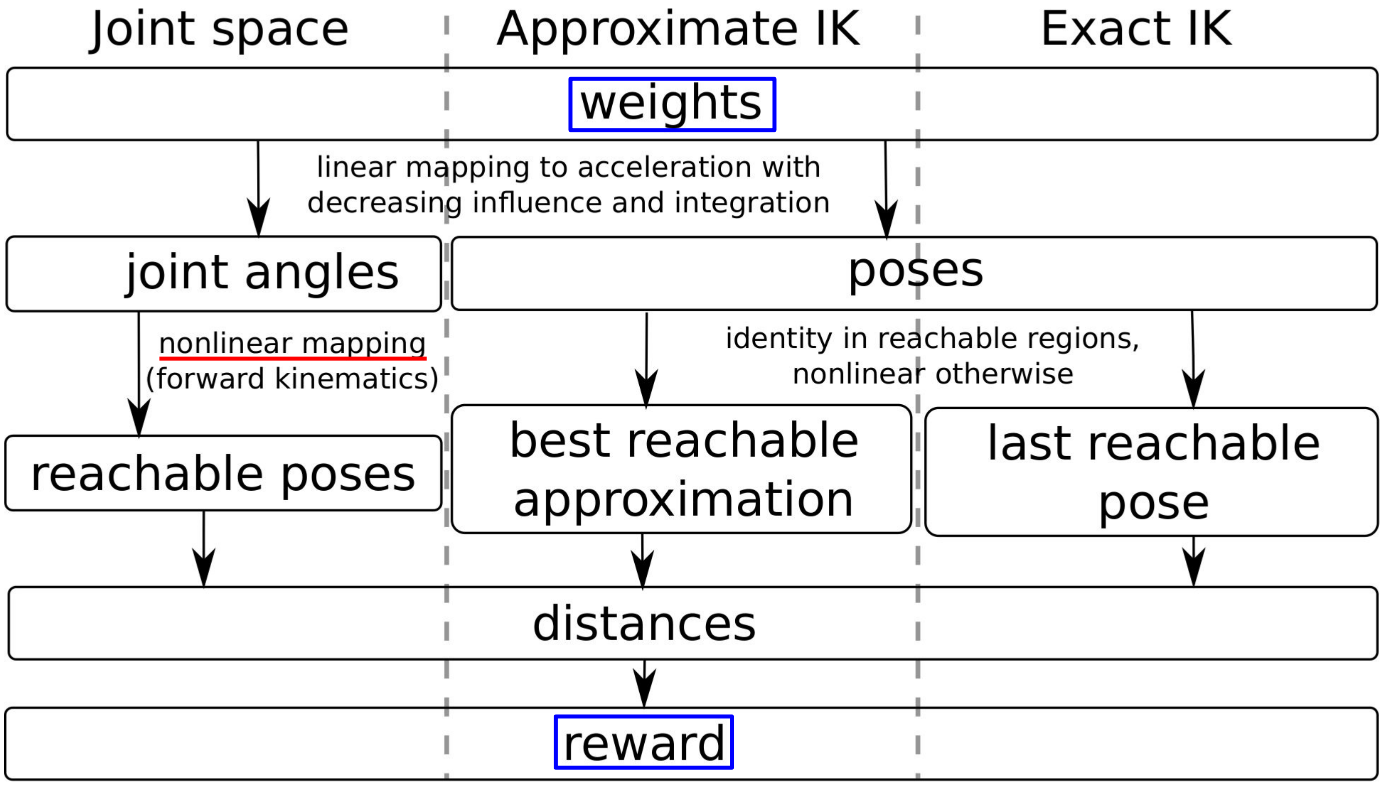


Pouring

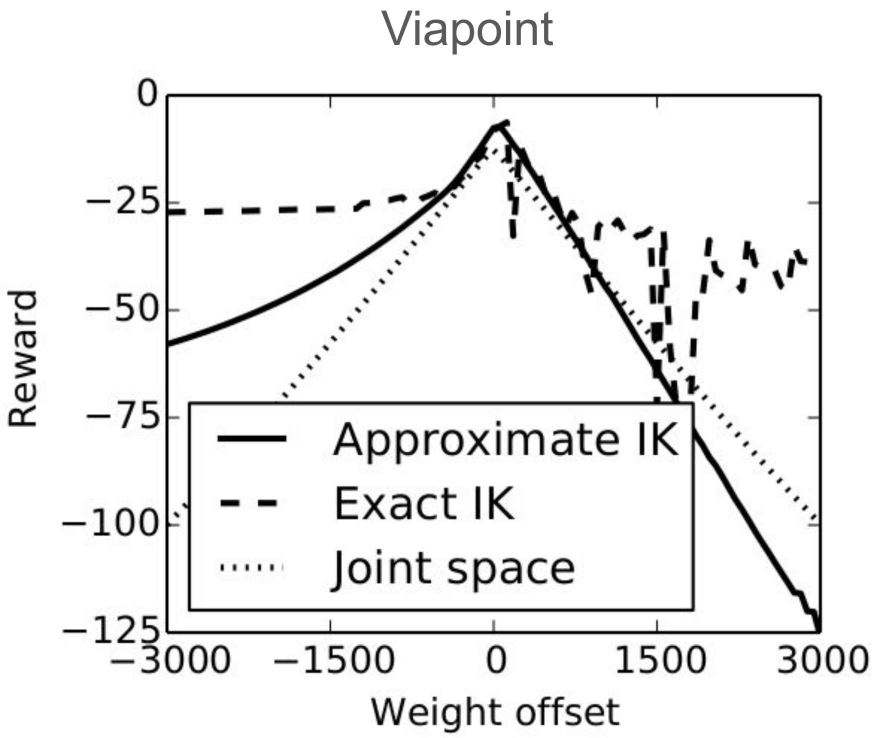


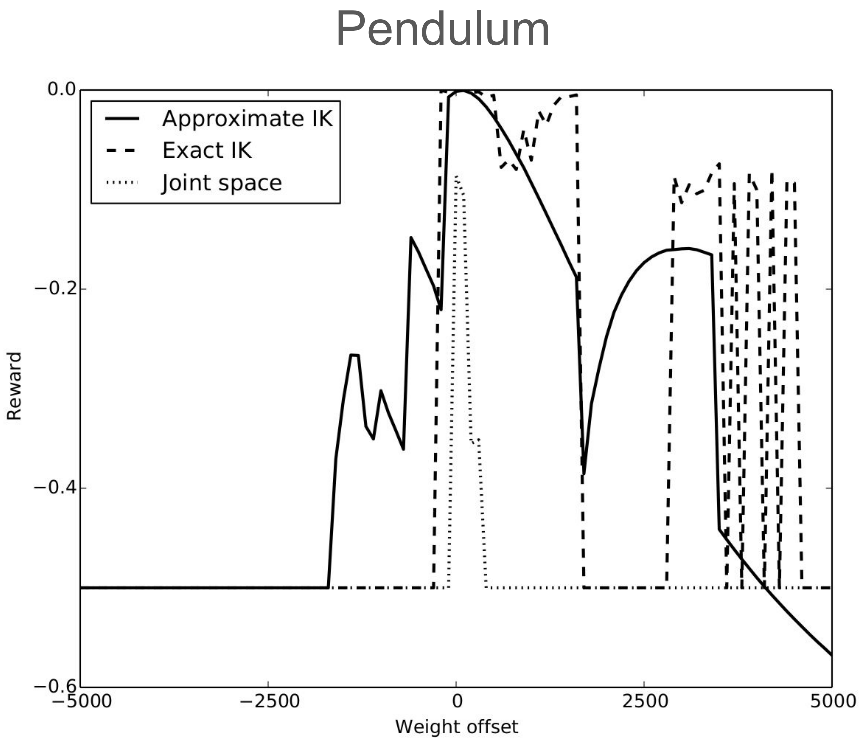
Difficulty

Separability in Cartesian Viapoint Problems



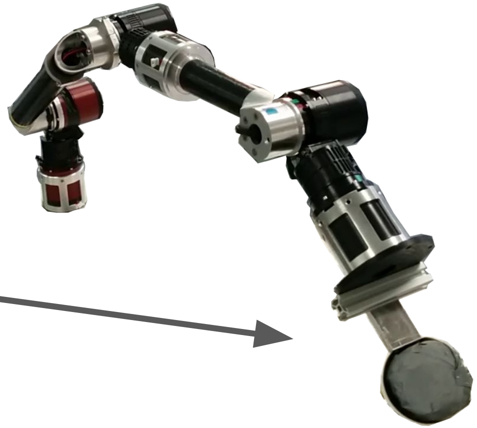
Reward Surface - Approximation vs. Exact IK



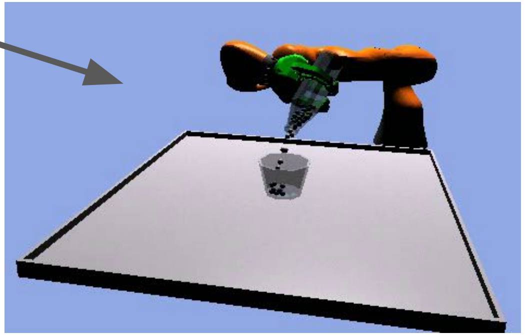


Conclusion

Conclusion



- Learn in the space in which the primary objective is defined
    - If you want smooth joint trajectories, learn in joint space
    - If you have a viapoint problem, learn in Cartesian space
    - If the relation between parameters and reward is too complex it does not make a difference
- Direct inverse kinematics formulation with approximation by SQP helps



Open Question

- What are the implications for other learning algorithms, e.g., Deep RL?

Literature

1. L. Gutzeit, A. Fabisch, M. Otto, J.H. Metzen, J. Hansen, F. Kirchner, and E.A. Kirchner. The BesMan learning platform for automated robot skill learning. *Frontiers in Robotics and AI*, 5, 2018.
2. A. Ude, B. Nemec, T. Petri, and J. Morimoto. Orientation in Cartesian space dynamic movement primitives. In *ICRA*, pages 2997-3004, 2014.
3. A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation*, 25:328-373, 2013.
4. P. Beeson and B. Ames. TRAC-IK: An open-source library for improved solving of generic inverse kinematics. In *Humanoids*, pages 928-935, 2015.
5. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159-195, 2001.

Code

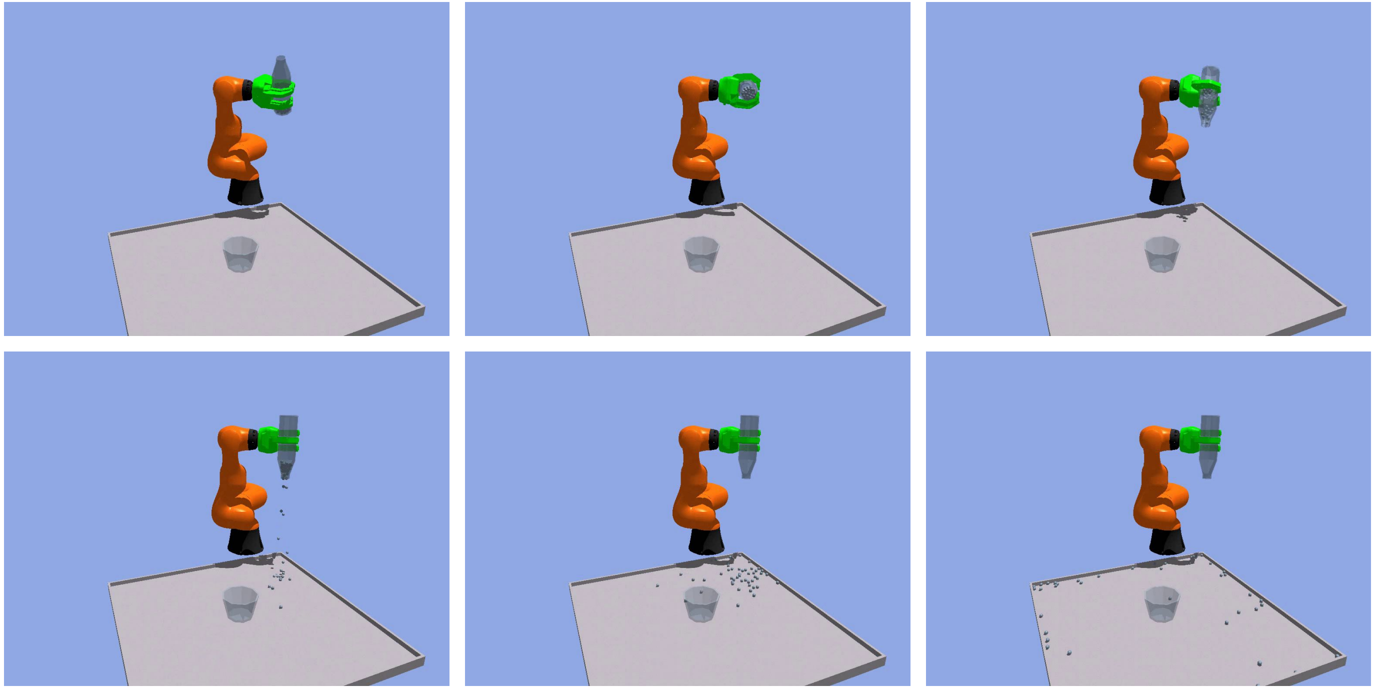
Code available at github.com/rock-learning/approxik

(except pouring environment)

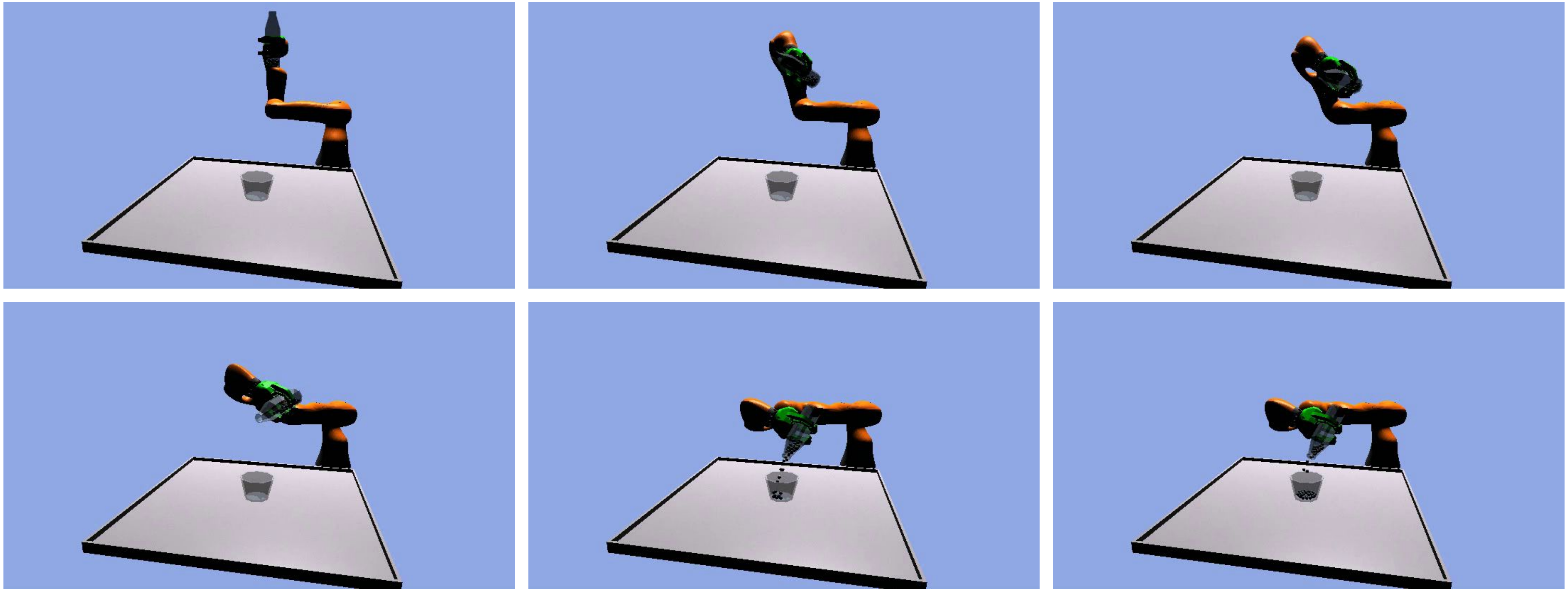
Why Is Pouring More Difficult?

- Slight action / parameter changes result in completely different rewards
  - Glass tips over
  - Marbles fall off the table
  - Collision with obstacle
- Temporal credit assignment is not possible with CMA-ES
  - Could be fixed with another policy search algorithm

Pouring - Failure (Missed Target)

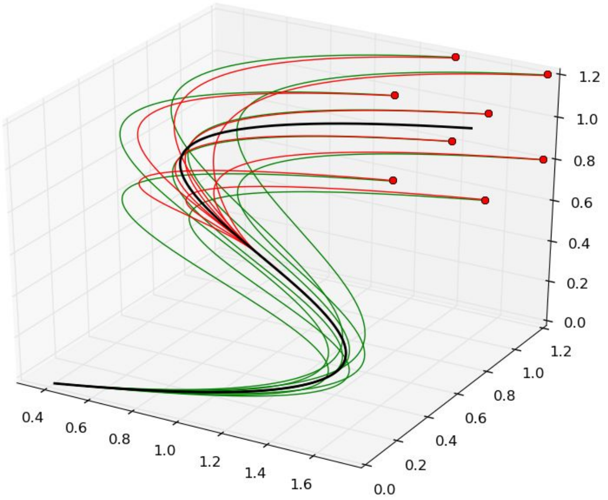


Pouring - Success



Dynamical Movement Primitives

Figure: Original DMP (black), modified goal (green), and modified goal online (red).



Ijspeert et al. (2013); Ude et al. (2014)

Reward Landscapes