A Graph-Search Based Approach for Movement Primitive Sequencing

Rudolf Lioutikov[1], Fabian Faller[1], Markus Sigg[1], Jan Peters[1,2], Guilherme Maeda[3]

***Abstract*— Demonstrating or pre-programming solutions to all possible tasks that a robot may encounter is unrealistic in scenarios that go beyond the structured industrial environment. Many tasks can be seen as sequences of simple sub-tasks. We propose an approach that allows robots to solve various tasks by sequencing elementary skills. These skills are represented by simple previously learned movement primitives. Each primitive only performs a limited action. Solutions for unseen problems are generated by combining these simple primitives into sequences, here using the A* graph search algorithm.**

Although work on imitation learning [1], [2] has been proposed as a method to avoid the need of an expert programmer, demonstrating new trajectories for every possible task may become a tedious process. While imitation learning has mostly focused on the individual improvement of primitives [3], it has neglected planning approaches such as simple, however powerful, graph searches that provide globally optimal solutions of an entire sequence. At the same time, the motion planning community has considered fully autonomous trajectory generation [4] where the demands on the world knowledge are usually hard to satisfy, efficient planning on kinodynamic state-spaces are non-trivial, and planning under holonomic assumptions involves post-processing solutions that may render infeasible trajectories.

This work explores a combination of Probabilistic Movement Primitives (ProMPs) [2] and the A* graph-search method to solve complex and unseen tasks. ProMPs consist of a trajectory generating model and a corresponding variable stiffness controller. In this work the sequencing of the primitives depends solely on the model, that is represented as normal distribution over the weights $\boldsymbol{w}$ over some basis functions $\boldsymbol{\Phi}$

$$\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \ \boldsymbol{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \delta\mathrm{I}\right)^{-1}\boldsymbol{\Phi}\boldsymbol{d}$$

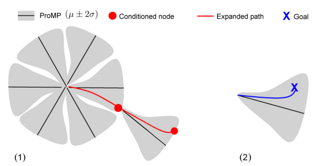
where $\boldsymbol{w}$ is learned from the demonstrations $\boldsymbol{d}$ using a ridge regression with regularizer δ. Given the representation of each primitive as a distribution the conditioning property of the ProMPs can now be used to either traverse the state space or to reach a goal state as illustrated in Figure 1a.

Each point in the state space can be considered a node of a graph and every primitive represents an edge connecting two nodes, namely the states at the beginning and the end of the primitive. Starting from the current state we now use an A* graph-search to find a path to the goal state.

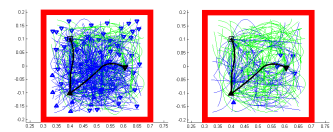
[1]Intelligent Autonomous Systems, TU Darmstadt, 64289 Darmstadt, Germany, {lioutikov, peters}@ias.tu-darmstadt.de

[2]Max Planck Institute for Intelligent Systems, 72070 Tübingen, Germany

[3]ATR Computational Neuroscience Labs, 2-2-2 Hikaridai Seika-sho, Soraku-gun, Kyoto 619-0288, Japan, g.maeda@atr.jp



(a) (1) An illustration of the use of ProMPs as an initial set of robot skills, with two expansions. In this case the set is composed of six elementary primitives, required to cover most of the search space from the current state. (2) A goal state can be easily reached with a movement primitive as long as the state lies inside the distribution.



(b) Expanded search tree for simple pick & place with unconstrained primitives for a pick and place task (left) and primitives requiring the object to be reachable (right). Blue lines denote movements without and green lines with having the object gripped. Upwards triangles indicate a pick and downward a place action. Bold lines and black markers show the computed solution.

However, some paths might be infeasible due to obstacles or other constraints. We again take advantage of the representation of the primitives as distributions and reduce the variance of each primitive such that the constraints are met, e.g. narrowing the distribution such that all drawn trajectories would avoid hitting an obstacle. Given that such constraints can eliminate entire primitives as possible successors, they significantly reduce the search space. Reversing this relation gives rise to the idea of requirements. We reduce the search space further by defining certain requirements for some primitives in order to be executed, e.g. only choose a pick or a place primitive if the object or the target is reachable. The significant benefit in reducing the search space can be seen in Figure 1b.

We evaluated the approach on a maze-pick-and-place task. The initial set of primitives consisted of simple ProMPs such as illustrated in Figure 1a including primitives for picking and placing. As shown in Figure 2 the presented approach successfully found a sequence of ProMPs, that navigated the robot through the maze while picking up an object at one location and placing it at the target location.

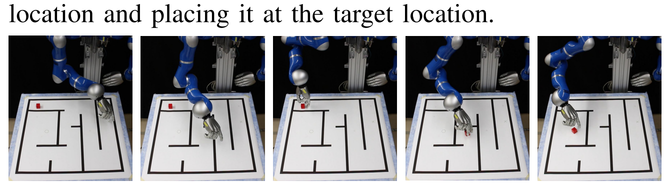


Fig. 2: The proposed approach found a sequence of simple primitives to successfully pick and place an object inside a maze environment.

[1] S. Schaal, "Dynamic movement primitives - a framework for motor control in humans and humanoid robots," in *The International Symposium on Adaptive Motion of Animals and Machines*, 2003.

[2] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots (AURO)*, accepted.

[3] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, 2013.

[4] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.