Fast Learning of Assembly Tasks using Dynamic Movement Primitives and Deterministic Policy Gradients

Fredrik Bagge Carlson*   Martin Karlsson

Introduction

- One-shot learning using DMP
- Update DMP using reinforcement learning
- Learn sensor-feedback controller with DMP as nominal controller

Introduction

We propose a framework for fast learning of robotic manipulation tasks that utilizes dynamic movement primitives, learned from human demonstration, to learn a nominal controller from a single demonstration.

An actor-critic framework is used to learn a nonlinear state and sensor feedback law, that acts around the nominal DMP controller.

Fast learning with this model-free approach is achieved by the DMP controller making use of the robot controllers internal dynamic model.

The off-policy characteristic of the proposed learning algorithm enables learning of the critic already in the human demonstration phase.

DMP equations

$$
\begin{aligned}
\tau^2 \ddot{q} &= \alpha_z(\beta_z(g - q) - \tau \dot{q}) + f_\theta(x) \qquad (1)\\
\tau \dot{x} &= -\alpha_x x\\
f_\theta(x) &= \phi(x)^T \theta
\end{aligned}
$$

A DMP can be considered a state feedback law that maps the state to reference positions and velocities $\mathcal{S} \times \mathcal{X} \rightarrow \mathcal{S} : \mu(q, \dot{q}, x)$.

To find a torque reference for the robot, the inverse model Eq. (2) may be used. This model is typically not available and is, for a robot with many degrees of freedom, hard to estimate from data.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^{-T}(q) f_{ext} \tag{2}$$

The external force/torque wrench present in assembly scenarios is especially hard to model in the presence of uncertainty and stiff environments.

A DMP can be considered a state feedback law that maps the state to reference positions and velocities $\mathcal{S} \times \mathcal{X} \to \mathcal{S} : \mu(q, \dot{q}, x)$.

To find a torque reference for the robot, the inverse model Eq. (2) may be used. This model is typically not available and is, for a robot with many degrees of freedom, hard to estimate from data.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^{-T}(q) f_{ext} \tag{2}$$

The external force/torque wrench present in assembly scenarios is especially hard to model in the presence of uncertainty and stiff environments.

A DMP can be considered a state feedback law that maps the state to reference positions and velocities $\mathcal{S} \times \mathcal{X} \rightarrow \mathcal{S} : \mu(q, \dot{q}, x)$.

To find a torque reference for the robot, the inverse model Eq. (2) may be used. This model is typically not available and is, for a robot with many degrees of freedom, hard to estimate from data.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + J^{-T}(q)f_{ext} \tag{2}$$

The external force/torque wrench present in assembly scenarios is especially hard to model in the presence of uncertainty and stiff environments.

Deterministic Policy Gradient

From the Bellman equation

$$Q^*(s,u) = r + \gamma Q^*(s_+, \mu_\theta(s_+))$$

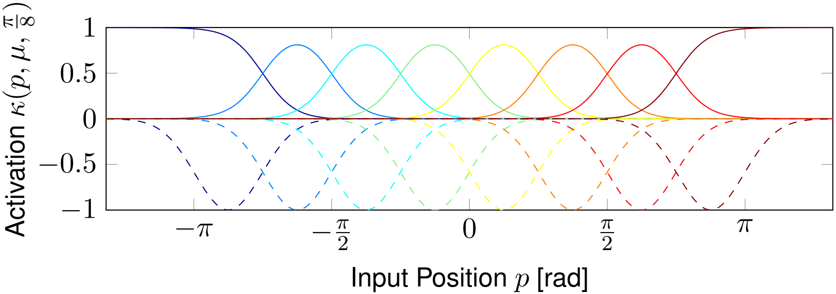
we get the temporal difference error δ associated with approximating the value function Q with Q^w. The DPG update equations then take on the (simplified) form[1]

$$\begin{aligned}
\delta &= r + \gamma Q^w(s_+, \mu_\theta(s_+)) - Q^w(s,u) \\
\theta_+ &= \theta + \nabla_\theta Q^w(s, \mu_\theta(s)) \\
w_+ &= w + \delta \nabla_w Q^w(s,u)
\end{aligned}$$

[1] David Silver et al. "Deterministic Policy Gradient Algorithms". In: *ICML*. Beijing, China, June 2014. URL: https://hal.inria.fr/hal-00938992.

I have been using simple function approximators.

$$Q^w(s,u) = u^T w + V^v(s)$$
$$V^v(s) = v^T \phi(s)$$
$$\mu_\theta(s) = \text{DMP} + \theta^T \phi(s)$$



Deep Learning

Replace shallow function approximators Q, V and μ with deep networks

$$\begin{aligned} Q^w(s,u) &= \text{Deep network} \\ V^v(s) &= \text{Deep network} \\ \mu_\theta(s) &= \text{DMP + Deep network} \end{aligned}$$

Efficient Exploration

Typical reinforcement learning frameworks employ more or less random exploration in order to gain knowledge of the environment and optimize the policy.

A notable exception is the Guided Policy Search (GPS) framework[2], in which a local, linear dynamics model is fit to recent data. A locally optimal linear control law is then calculated using iterative LQG[3] given the model and a specified cost function.

[2] Sergey Levine and Vladlen Koltun. "Guided Policy Search". In: *ICML '13: Proceedings of the 30th International Conference on Machine Learning*. `http://graphics.stanford.edu/projects/gpspaper`. 2013.

[3] Yuval Tassa, Nicolas Mansard, and Emo Todorov. "Control-limited differential dynamic programming". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. `https://homes.cs.washington.edu/~todorov/papers/TassaICRA14.pdf`. IEEE. 2014, pp. 1168–1175.

Efficient Exploration

Typical reinforcement learning frameworks employ more or less random exploration in order to gain knowledge of the environment and optimize the policy.

A notable exception is the Guided Policy Search (GPS) framework[2], in which a local, linear dynamics model is fit to recent data. A locally optimal linear control law is then calculated using iterative LQG[3] given the model and a specified cost function.

[2] Sergey Levine and Vladlen Koltun. "Guided Policy Search". In: *ICML '13: Proceedings of the 30th International Conference on Machine Learning*. http://graphics.stanford.edu/projects/gpspaper. 2013.

[3] Yuval Tassa, Nicolas Mansard, and Emo Todorov. "Control-limited differential dynamic programming". In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. https://homes.cs.washington.edu/~todorov/papers/TassaICRA14.pdf. IEEE. 2014, pp. 1168–1175.

Efficient Exploration continued

In GPS, the global actor is trained to reproduce a collection of recent trajectories obtained from executing the locally optimal linear controllers. As time progresses, the control law represented by the actor converges to the collection of recent linear controllers. A similar idea is being used here to accelerate learning through intelligent exploration.