

# Learning Neural Network Policies with Guided Policy Search under Unknown Dynamics

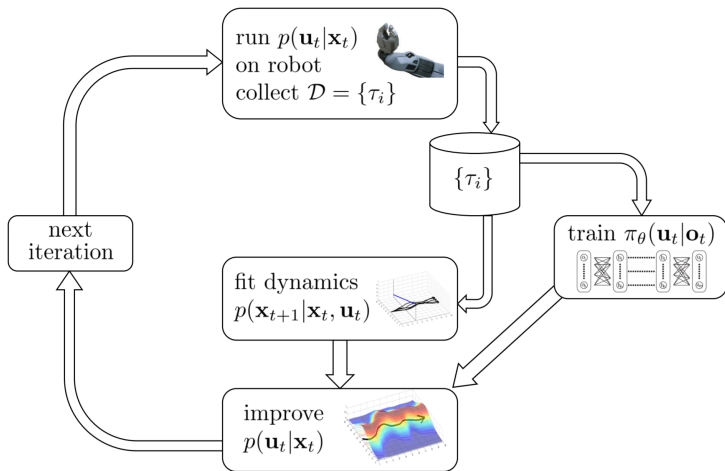
Sergey Levine and Pieter Abbeel  
Presenter: Jienan Yao

March 8, 2019

# Introduction

- ▶ Model-based methods
  - ▶ difficult for complex systems
- ▶ Model-free methods
  - ▶ require carefully designed, low-dimensional parameterizations
- ▶ Goal
  - ▶ learn dynamics of the system and locally valid optimal control at the same time
  - ▶ train a policy that is globally valid using learned local policies

# Overview



# Preliminaries

## Time-varying Gaussian Policy

- ▶  $p(\mathbf{u}_t|\mathbf{x}_t) = N(\mathbf{K}_t\mathbf{x}_t + \mathbf{k}_t, \mathbf{C}_t)$
- ▶ Could be efficiently optimized when the initial state distribution is narrow and approximately Gaussian

# Preliminaries

## Time-varying Gaussian Policy

- ▶  $p(\mathbf{u}_t|\mathbf{x}_t) = N(\mathbf{K}_t\mathbf{x}_t + \mathbf{k}_t, \mathbf{C}_t)$
- ▶ Could be efficiently optimized when the initial state distribution is narrow and approximately Gaussian

## iteratively linear-Gaussian regulator (iLQG)

- ▶ Iteratively construct locally optimal linear feedback controllers
- ▶ Computed by dynamic programming under linearization of dynamics and quadratic expansion of cost
- ▶ linear-Gaussian controller

$$p(\mathbf{u}_t|\mathbf{x}_t) = N(\hat{\mathbf{u}}_t + \mathbf{k}_t + \mathbf{K}_t(\mathbf{x}_t - \hat{\mathbf{x}}_t), Q_{\mathbf{u},\mathbf{u}t}^{-1})$$

# Trajectory Optimization under Unknown Dynamics

Dynamics  $N(f_{\mathbf{x}t}\mathbf{x}_t + f_{\mathbf{u}t}\mathbf{u}_t, \mathbf{F}_t)$  unknown

- ▶ Estimated from the samples generated from the real system under previous linear-Gaussian controllers
- ▶ Having estimated linear-Gaussian dynamics at each time step, run the preceding dynamic programming algorithm

Issue:

- ▶ Fitted dynamics is only valid in local region around the sample
- ▶ New controller could be very different from the old one
- ▶ Addressed by imposing KL-Divergence constraints between the old and new trajectory distribution

## KL-Divergence Constraints

- Modified cost function

$$\min_{p(\tau) \in N(\tau)} E_p[l(\tau)] \text{ s.t. } D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) \leq \epsilon$$

## KL-Divergence Constraints

- Modified cost function

$$\min_{p(\tau) \in N(\tau)} E_p[l(\tau)] \text{ s.t. } D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) \leq \epsilon$$

- Lagrangian of this problem ( $\eta$  dual variable)

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = E_p[l(\tau)] + \eta[D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) - \epsilon]$$



## KL-Divergence Constraints

- ▶ Modified cost function

$$\min_{p(\tau) \in N(\tau)} E_p[l(\tau)] \text{ s.t. } D_{\text{KL}}(p(\tau) \parallel \hat{p}(\tau)) \leq \epsilon$$

- ▶ Lagrangian of this problem ( $\eta$  dual variable)

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = E_p[l(\tau)] + \eta[D_{\text{KL}}(p(\tau) \parallel \hat{p}(\tau)) - \epsilon]$$

- ▶ Assuming  $p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \hat{p}(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(f_{\mathbf{x}t}\mathbf{x}_t + f_{\mathbf{u}t}\mathbf{u}_t, \mathbf{F}_t)$

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = \left[ \sum_t E_{p(\mathbf{x}_t, \mathbf{u}_t)} [l(\mathbf{x}_t, \mathbf{u}_t) - \eta \log \hat{p}(\mathbf{u}_t | \mathbf{x}_t)] \right] - \eta \mathcal{H}(p(\tau)) - \eta \epsilon$$

## KL-Divergence Constraints

- ▶ Modified cost function

$$\min_{p(\tau) \in N(\tau)} E_p[l(\tau)] \text{ s.t. } D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) \leq \epsilon$$

- ▶ Lagrangian of this problem ( $\eta$  dual variable)

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = E_p[l(\tau)] + \eta[D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) - \epsilon]$$

- ▶ Assuming  $p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \hat{p}(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(f_{\mathbf{x}t}\mathbf{x}_t + f_{\mathbf{u}t}\mathbf{u}_t, \mathbf{F}_t)$

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = \left[ \sum_t E_{p(\mathbf{x}_t, \mathbf{u}_t)} [l(\mathbf{x}_t, \mathbf{u}_t) - \eta \log \hat{p}(\mathbf{u}_t | \mathbf{x}_t)] \right] - \eta \mathcal{H}(p(\tau)) - \eta \epsilon$$

- ▶ Augmented cost function

$$\tilde{l}(\mathbf{x}_t, \mathbf{u}_t) = \frac{1}{\eta} l(\mathbf{x}_t, \mathbf{u}_t) - \log \hat{p}(\mathbf{u}_t | \mathbf{x}_t)$$

## KL-Divergence Constraints

- ▶ Modified cost function

$$\min_{p(\tau) \in N(\tau)} E_p[l(\tau)] \text{ s.t. } D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) \leq \epsilon$$

- ▶ Lagrangian of this problem ( $\eta$  dual variable)

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = E_p[l(\tau)] + \eta[D_{\text{KL}}(p(\tau) || \hat{p}(\tau)) - \epsilon]$$

- ▶ Assuming  $p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \hat{p}(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(f_{\mathbf{x}t}\mathbf{x}_t + f_{\mathbf{u}t}\mathbf{u}_t, \mathbf{F}_t)$

$$\mathcal{L}_{\text{traj}}(p(\tau), \eta) = \left[ \sum_t E_{p(\mathbf{x}_t, \mathbf{u}_t)} [l(\mathbf{x}_t, \mathbf{u}_t) - \eta \log \hat{p}(\mathbf{u}_t | \mathbf{x}_t)] \right] - \eta \mathcal{H}(p(\tau)) - \eta \epsilon$$

- ▶ Augmented cost function

$$\tilde{l}(\mathbf{x}_t, \mathbf{u}_t) = \frac{1}{\eta} l(\mathbf{x}_t, \mathbf{u}_t) - \log \hat{p}(\mathbf{u}_t | \mathbf{x}_t)$$

- ▶ Solved by dual gradient descent

## Dual Gradient Descent


$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s.t. } C(\mathbf{x}) = 0$$

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda C(\mathbf{x})$$

$$g(\lambda) = \mathcal{L}(\mathbf{x}^*(\lambda), \lambda)$$

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda)$$

$$\frac{dg}{d\lambda} = \frac{d\mathcal{L}}{d\lambda}(\mathbf{x}^*, \lambda)$$

- 
1. Find  $\mathbf{x}^* \leftarrow \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda)$
  2. Compute  $\frac{dg}{d\lambda} = \frac{d\mathcal{L}}{d\lambda}(\mathbf{x}^*, \lambda)$
  3.  $\lambda \leftarrow \lambda + \alpha \frac{dg}{d\lambda}$

## Background Dynamics Distribution

- ▶ Use priors to greatly reduce the number of samples required
- ▶ Gaussian Mixture Model (GMM) is a good choice for physical systems such as robots
  - ▶ dynamics reasonably approximated with piecewise linear functions
  - ▶ not necessarily good forward model but obtain prior for dynamics
- ▶ Refit the GMM at each iteration
- ▶ Infer the cluster weights for the samples
- ▶ Use the weighted mean and covariance of these clusters as the prior parameters to estimate a time-varying linear dynamics

## General Parameterized Policies

---

**Algorithm 1** Guided policy search with unknown dynamics

---

- 1: **for** iteration  $k = 1$  to  $K$  **do**
  - 2:   Generate samples  $\{\tau_i^j\}$  from each linear-Gaussian controller  $p_i(\tau)$  by performing rollouts
  - 3:   Fit the dynamics  $p_i(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$  to the samples  $\{\tau_i^j\}$
  - 4:   Minimize  $\sum_{i,t} \lambda_{i,t} D_{\text{KL}}(p_i(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t) \| p_i(\mathbf{x}_t, \mathbf{u}_t))$  with respect to  $\theta$  using samples  $\{\tau_i^j\}$
  - 5:   Update  $p_i(\mathbf{u}_t|\mathbf{x}_t)$  using the algorithm in Section 3 and the supplementary appendix
  - 6:   Increment dual variables  $\lambda_{i,t}$  by  $\alpha D_{\text{KL}}(p_i(\mathbf{x}_t)\pi_\theta(\mathbf{u}_t|\mathbf{x}_t) \| p_i(\mathbf{x}_t, \mathbf{u}_t))$
  - 7: **end for**
  - 8: **return** optimized policy parameters  $\theta$
-

## General Parameterized Policies

- ▶ Objective

$$\min_{\theta, p(\tau)} E_{p(\tau)}[l(\tau)] \text{ s.t. } D_{\text{KL}}(p(\mathbf{x}_t)\pi_{\theta}(\mathbf{u}_t|\mathbf{x}_t)||p(\mathbf{x}_t, \mathbf{u}_t)) = 0, \forall t$$

- ▶ Lagrangian of the problem

$$\mathcal{L}_{\text{GPS}}(\theta, p, \lambda) = E_{p(\tau)}[l(\tau)] + \sum_{t=1}^T \lambda_t D_{\text{KL}}(p(\mathbf{x}_t)\pi_{\theta}(\mathbf{u}_t|\mathbf{x}_t)||p(\mathbf{x}_t, \mathbf{u}_t))$$

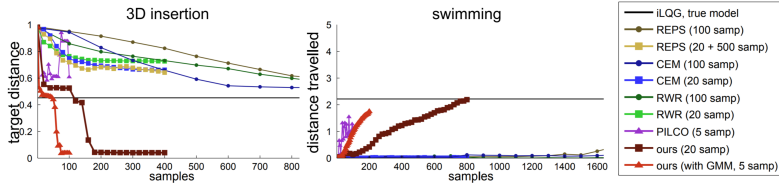
- ▶ Parameterized policy trained in a supervised fashion
- ▶ Trajectory optimization exploits structure of linear-Gaussian controllers, trained with fewer samples

## Experiments Conducted

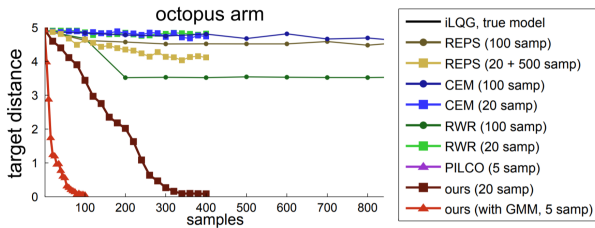
- ▶ 2D, 3D peg insertion (discontinuous dynamics)
- ▶ Octopus arm control (high-dimensional state and action space)
- ▶ Planar swimming (three-link snake)
- ▶ Walking (seven-link biped to maintain a target velocity)



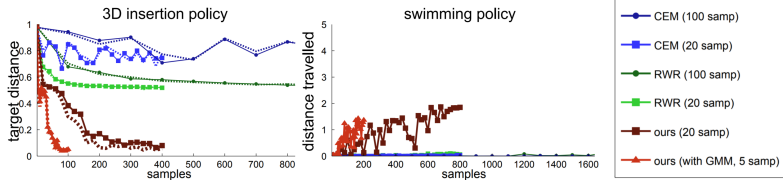
# Trajectory Optimization



# Trajectory Optimization



# Neural Network Policy Learning with GPS



## Conclusion and Discussion

- ▶ Optimize linear-Gaussian controllers under unknown dynamics
  - ▶ hybrid model-based and model-free approach
  - ▶ rely on a stronger assumption that time-varying linear-Gaussians are reasonable local approximation for the dynamics
- ▶ Train arbitrary parameterized policies (e.g. Neural Networks) within GPS framework
  - ▶ experiments show intelligent performance in partially observed environments, even for tasks that cannot be solved with direct model-free policy search
- ▶ Future directions
  - ▶ incorporate sensory information that is difficult to simulate but useful in partially observed domains