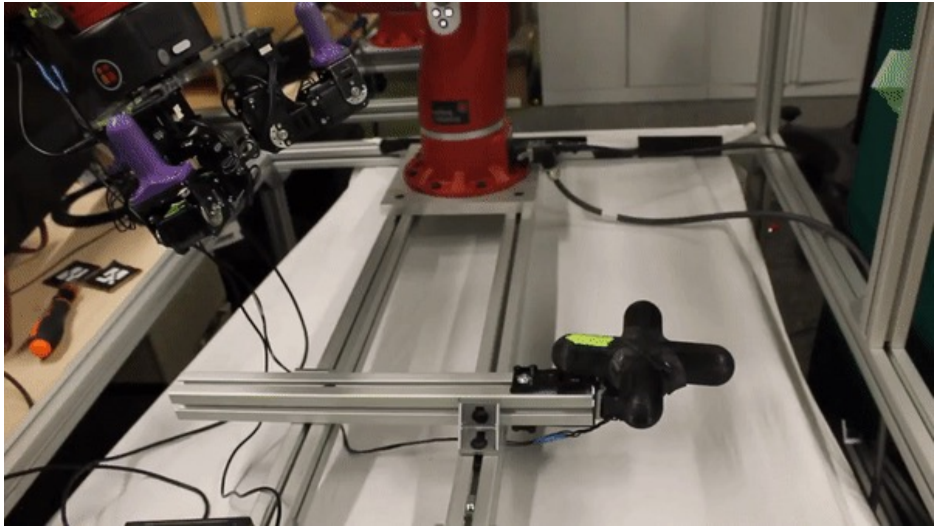
Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning

Presenter: Charles Nimo

October 14th 2021

Robotic Manipulators

Key Challenges



- No human in the loop ➡ Automatically learn from data
- Data-Efficient Learning
- Uncertainty: sensor noise, unknown processes, limited knowledge

Prior Work

- Policy Search for Motor Primitives in Robotics (Machine Learning, 2011)
    - A model-free policy learning method is presented which relies on rollouts sampled from the system.
- Gaussian Processes in Reinforcement Learning (NIPS, 2004)
    - Proposed algorithms that used Gaussian Process dynamics models in Reinforcement Learning setup
- Autonomous Helicopter Control using Reinforcement Learning Policy Search Methods (ICRA, 2001)
    - performs model-based reinforcement learning with certainty equivalence assumptions of latent system dynamics
- PILCO: A Model-Based and Data-Efficient Approach to Policy Search (ICML, 2011)
    - Introduces PILCO, a model-based policy search method aimed at reducing model bias

Central Problem

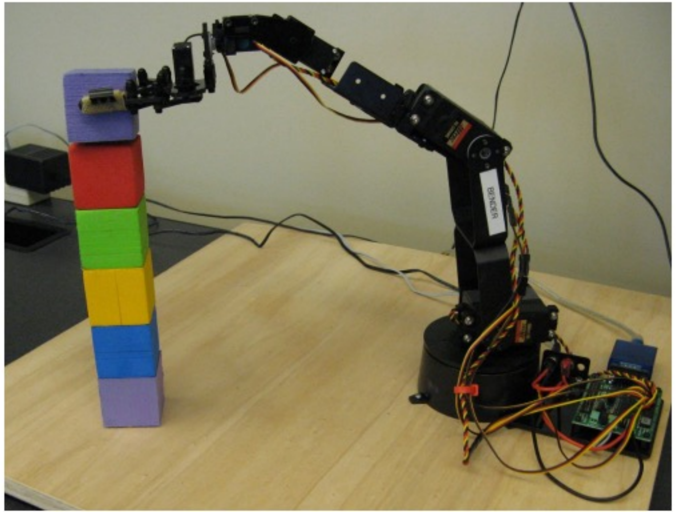
Can reinforcement learning be data efficient enough for robust manipulation with inexpensive hardware?

Central Problem

Data Efficient Reinforcement Learning

The ability to learn and make decisions in complex domains without requiring large quantities of data

Objective



Use data-efficient reinforcement learning to train a low precision robotic arm to stack a tower of foam blocks autonomously

The Task:

- No grasping
- Block Tracking with Kinect 640x480 RGB Camera
- Small number of interactions to prevent wear and tear
- No imitation learning – learns from scratch
- Cost Function

Probabilistic Inference for Learning Control (PILCO)

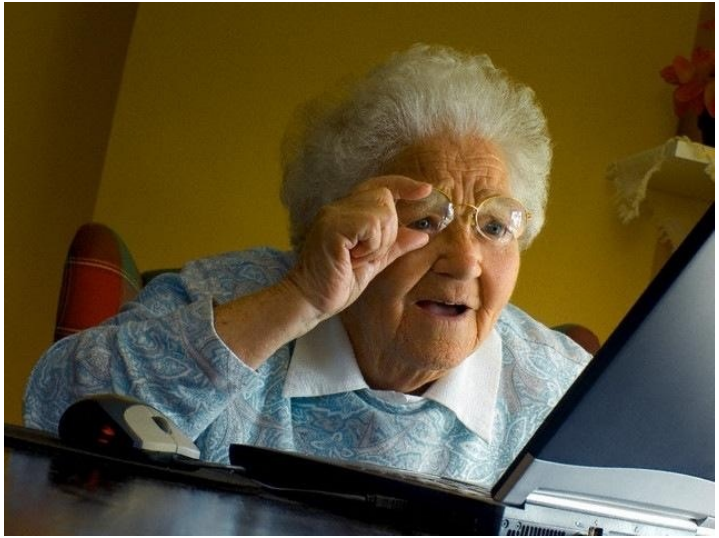
A framework for rapid model-based data-efficient reinforcement learning based on Gaussian Processes (GP).

Algorithm 1 PILCO

init: Set controller parameters ψ to random.
Apply random control signals and record data.
repeat
    Learn probabilistic GP dynamics model using all data
    repeat $\triangleright$ Model-based policy search
        Approx. inference for policy evaluation: get $J^{\pi}(\psi)$
        Gradients $\mathrm{d}J^{\pi}(\psi)/\mathrm{d}\psi$ for policy improvement
        Update parameters ψ (e.g., CG or L-BFGS).
    until convergence; **return** ψ^*
    Set $\pi^* \leftarrow \pi(\psi^*)$.
    Apply π^* to robot (single trial/episode); record data.
until task learned

Gaussian Processes

is a (potentially infinite) collection of random variables (RV) such that the joint distribution of every finite subset of RVs is multivariate Gaussian



PILCO Framework (High Level Steps)

Objective

Minimize expected long-term cost

$$J^{\pi} = \sum_{t=0}^{T} \mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)]$$

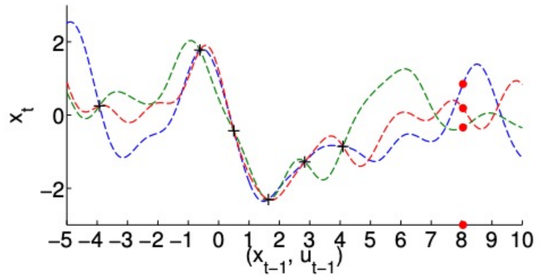
1. Probabilistic Model Learning (System Identification)
2. Long Term Planning/Prediction
3. Policy Search
4. Apply Policy to Robot

PILCO Framework

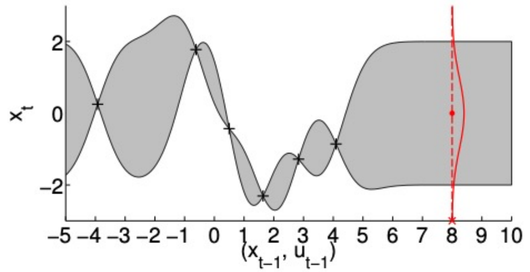
1. Probabilistic Model Learning

Apply Random Actions → Record Data → Learn GP Model

Task: find a (transition) function $f : (\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \mapsto \mathbf{x}_t$



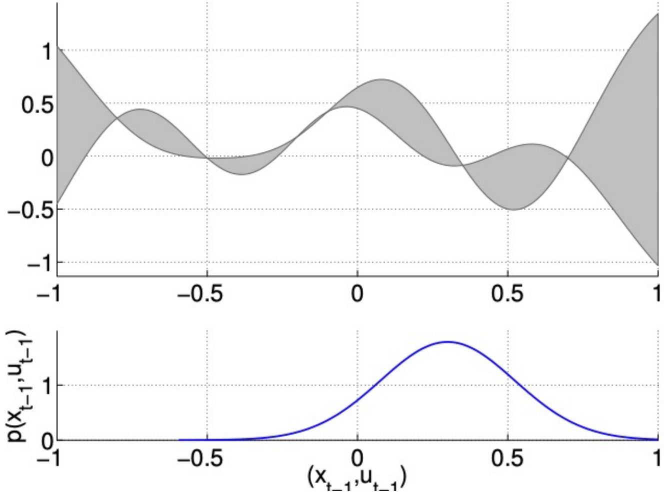
Plausible (deterministic) function approximators

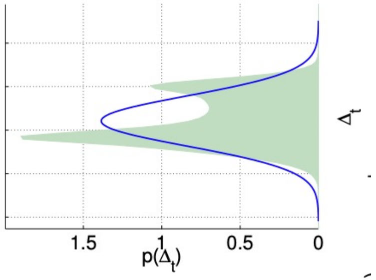


Probabilistic function approximator: distribution over plausible functions

PILCO Framework

1. Probabilistic Model Learning
2. Long Term Planning/Predictions

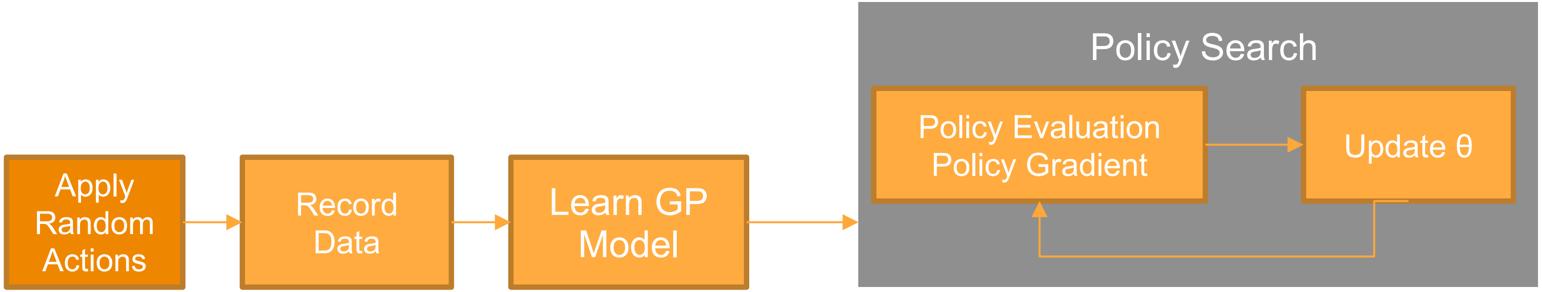




PILCO Framework

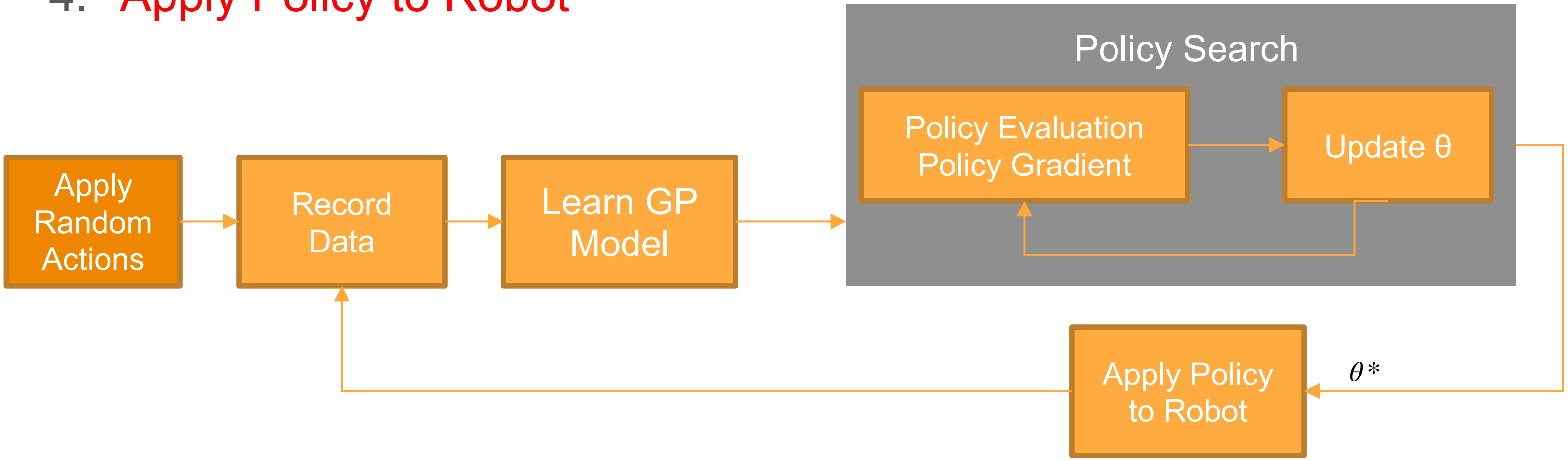
1. Probabilistic Model Learning
2. Long Term Planning/Predictions
3. Policy Search

$$\mathbb{E}_{\mathbf{x}_t}[c(\mathbf{x}_t)] = \int c(\mathbf{x}_t)\mathcal{N}\big(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\big)\, \mathrm{d}\mathbf{x}_t$$



PILCO Framework

1. Probabilistic Model Learning
2. Long Term Planning/Predictions
3. Policy Search
4. Apply Policy to Robot

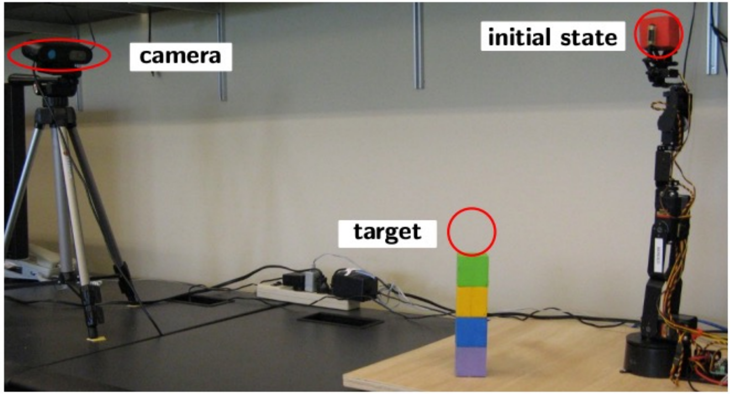


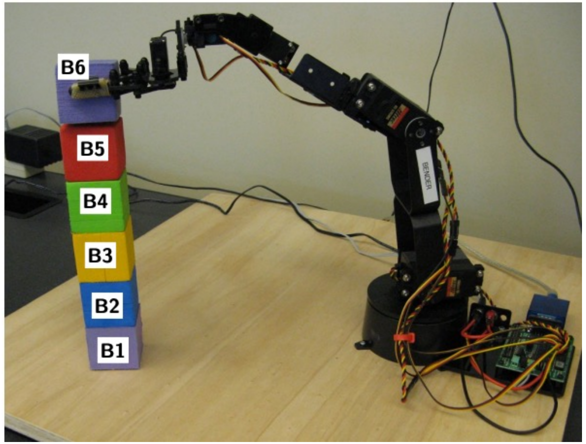
Experimental Validation

First Setup

Independent Controllers

- Independently trained controllers for each block (5)
- Total interaction time for stacking 5 blocks – 230 s (10 trials per block)



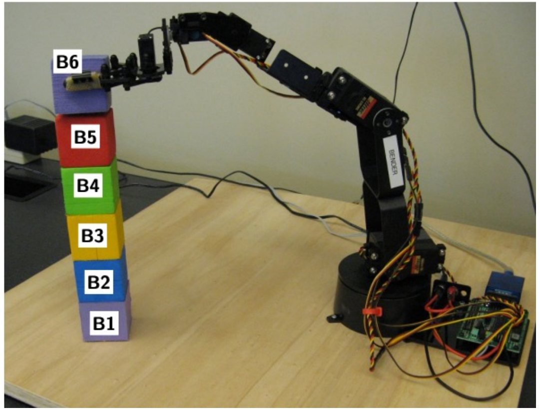


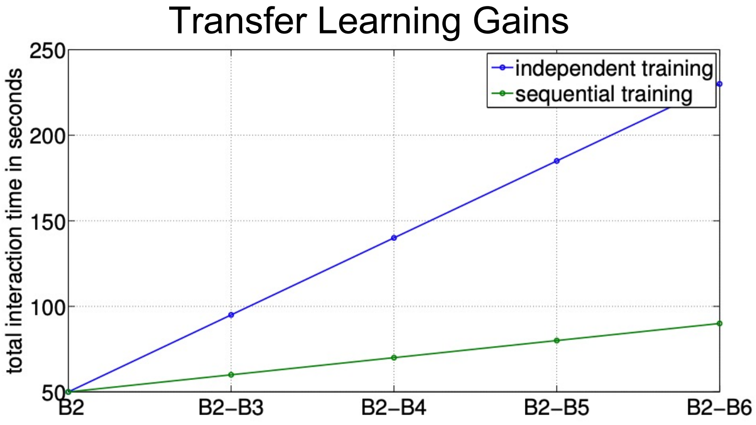
Experimental Validation

First Setup

Sequential Transfer Learning

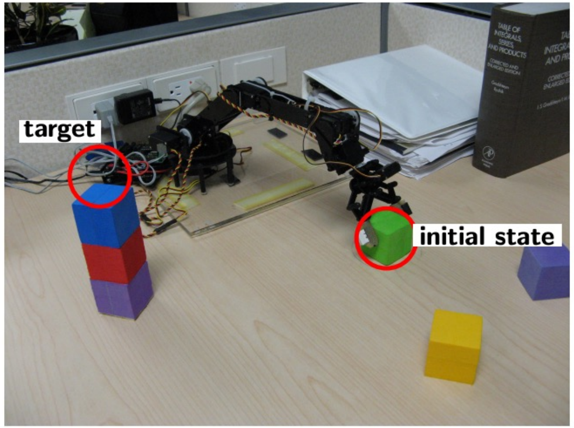
- Train independent controller
- Reuse the dynamics model and controller parameters for next block
- Learning to stack blocks required 90 s





The Task:

Experimental Validation



Second Setup

Collision Avoidance

- Collision is defined to occur when the robot arm collided with the tower of foam blocks
- Planning with state space constraints led to higher success rate
- Distances measured from block in gripper and target location

without collision avoidance	B2	B3	B4	B5	B6
collisions during training	12/40 (30%)	11/40 (27.5%)	13/40 (32.5%)	18/40 (45%)	21/40 (52.5%)
block deposit success rate	50%	43%	37%	47%	33%
distance (in cm) to target at time T	1.39 ± 0.81	0.73 ± 0.36	0.65 ± 0.35	0.71 ± 0.46	0.59 ± 0.34
with collision avoidance	B2	B3	B4	B5	B6
collisions during training	0/40 (0%)	2/40 (5%)	1/40 (2.5%)	3/40 (7.5%)	1/40 (2.5%)
block deposit success rate	90%	97%	90%	70%	97%
distance (in cm) to target at time T	0.89 ± 0.80	0.65 ± 0.33	0.67 ± 0.46	0.80 ± 0.37	1.34 ± 0.56

Limitations & Future Work

Limitations:

- PILCO is not optimal control
- Probabilistic models are only confident in areas of the space previously observed
- Does not take temporal correlation into account

Future Work:

- How could Neural Networks be used instead of Gaussian Processes?
- How does the PILCO framework perform to more complex tasks?

Summary

- Learning of Probabilistic Dynamics Model and Controller
- Incorporates model-uncertainty into long term planning
- Collision Avoidance during planning
- Does not rely on expert knowledge i.e., imitation learning or task specific prior knowledge
- Data Efficiency – learning from scratch is applicable to affordable, off-the-shelf robots

Extended Readings

- Gal, Yarin. Improving PILCO with Bayesian Neural Network Dynamics Models (ICML, 2016)
- Ebden, M. Gaussian Processes for Regression: A Quick Introduction (2008)
- Deisenroth, M.P. PILCO: A Model-Based and data efficient approach to Policy Search (ICML, 2011)

Thank you!