What can we learn from demonstrations?

Marc Toussaint

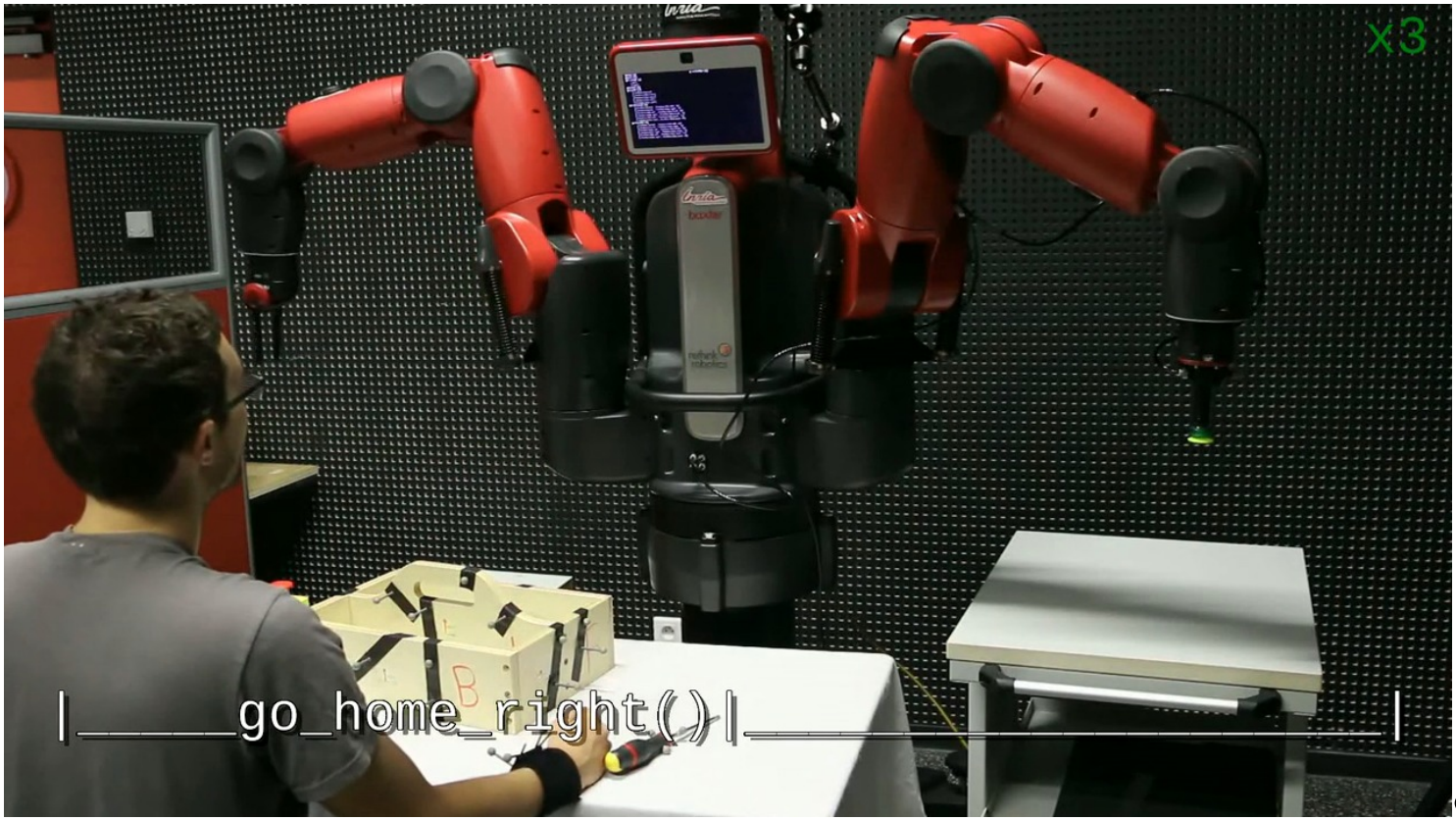
Machine Learning & Robotics Lab – University of Stuttgart

IROS workshop on ML in Robot Motion Planning, Okt 2018

Outline

- Previous work on learning from demonstration
  - Cooperative Manipulation Learning
  - Learning Manipulation Skills
- Recent work on Logic-Geometric Programming
- What can we learn from demonstrations?

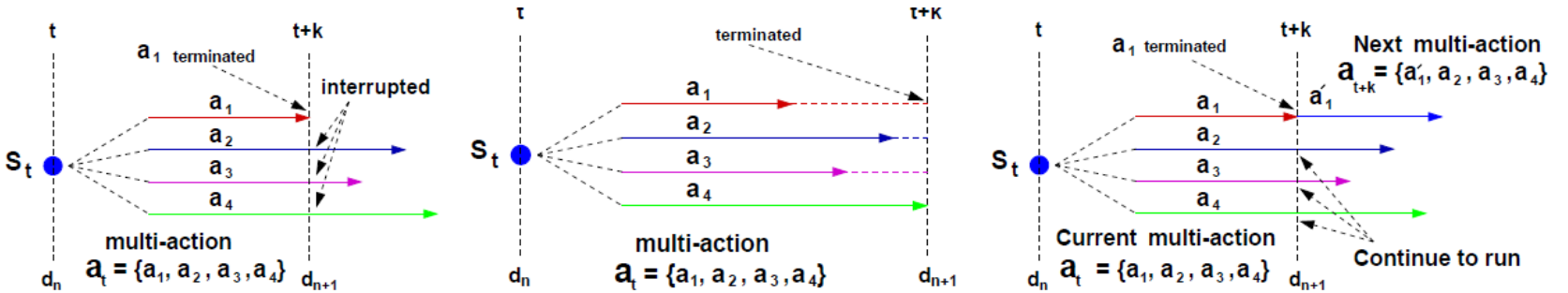
Cooperative Manipulation Learning



(In cooperation with Manuel Lopes, Jan Peters, Justus Piater; EU-Project *3rdHand*)

Process formalization

- Existing formulations: semi-MDPs over multi-actions
  - Concurrent Action Models (Rohanimanesh & Mahadevan); Concurrent MDPs & Probabilistic Temporal Planning (Mausam & Weld)
  - A certain episode times, the planner makes a multi-action decision $(a_1, a_2, .., a_n)$ for all n agents; the decision space becomes combinatorial



Rohanimanesh, Mahadevan (NIPS'02)

Relational Activity Processes (RAPs)

- actions $\rightarrow$ $\begin{cases} \text{activities, which are part of the } \textit{state} \\ \text{decisions (start/stop), which are instantaneous} \end{cases}$
- relational *state* lists the current activities:

```
(object Handle), (free humanLeft), (humanLeft graspingScrew)=1.0,
(humanRight grasped Handle), (Handle held), (robot releasing Long1)=1.5, ..
```

- **Reduction to a standard semi-MDP**
  - Standard methods for MCTS, direct policy learning & inverse RL become applicable in relational concurrent multi-agent domains

Toussaint, Munzer, Mollard & Lopes: *Relational Activity Processes for Modeling Concurrent Cooperation*. ICRA'16

Imitation learning & inverse RL for cooperative manipulation

- Great prior work:

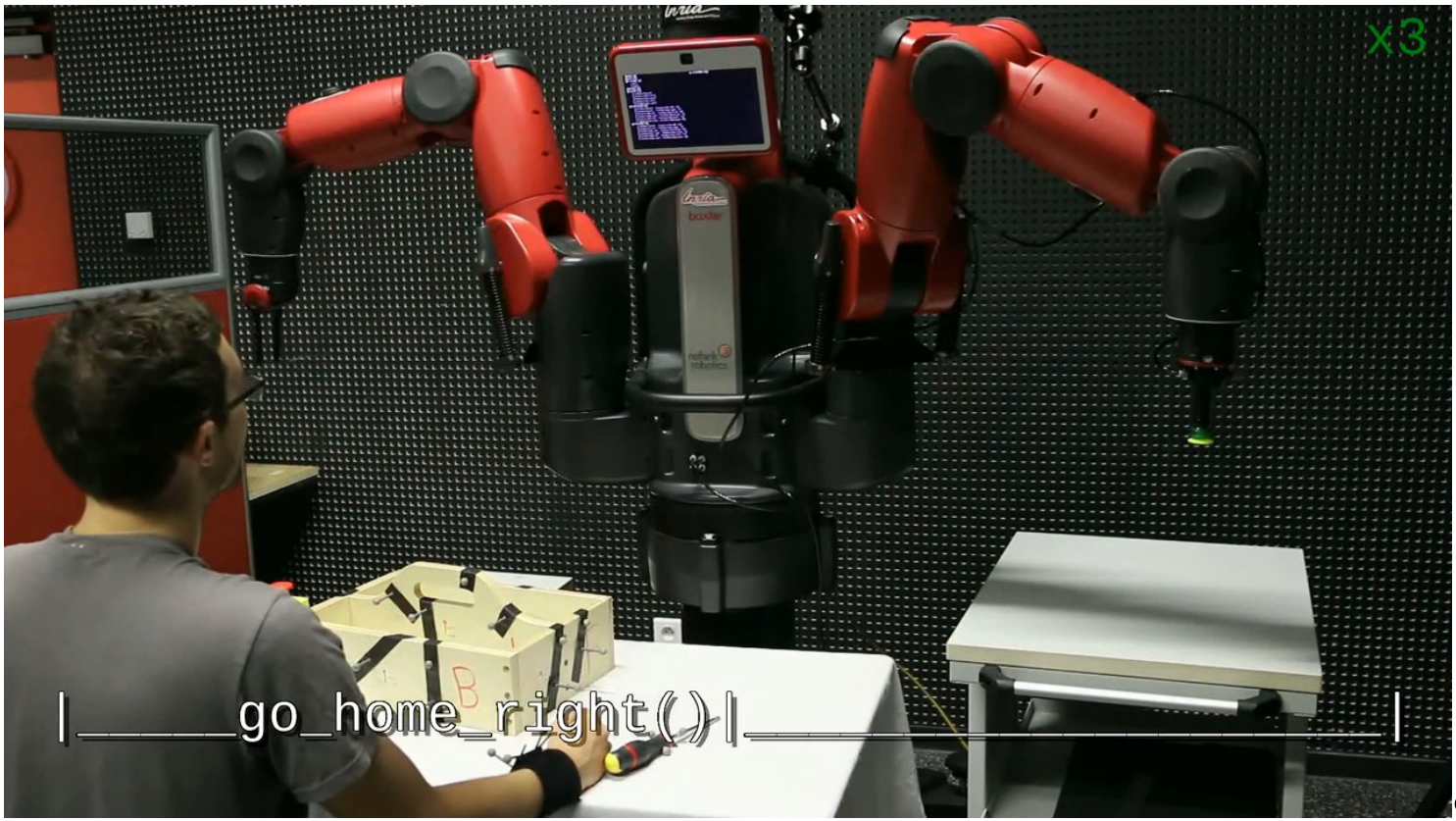
  Munzer et al.: *Inverse reinforcement learning in relational domains.* IJCAI'15

  - Imitation: Tree Boosted Relational Imitation Learning (TBRIL) to train a policy

    $$\pi(a \mid s) = \frac{e^{f(a,s)}}{\sum_{a' \in \mathcal{D}(s)} e^{f(a',s)}} \ , \quad f(a,s) = \psi(a,s)^\top \beta$$

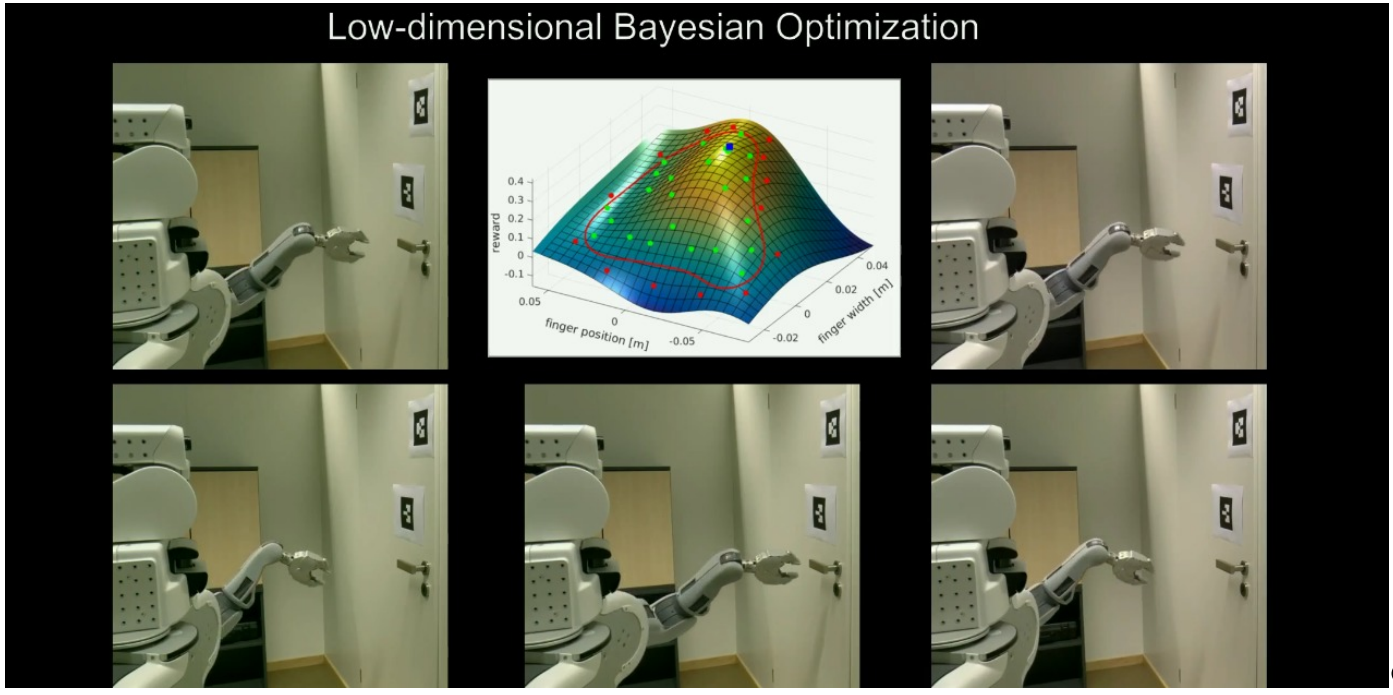
  - Use relational reward shaping and Cascaded Supervised IRL (CSI) to infer a relational reward function

- Directly translates to RAPs



Toussaint, Munzer, Mollard & Lopes: *Relational Activity Processes for Modeling Concurrent Cooperation*. ICRA'16

Learning Manipulation Skills



Peter Englert's work

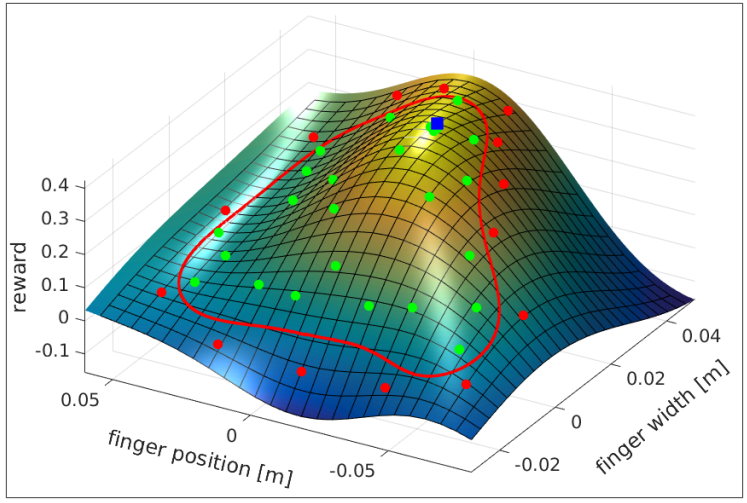
Combined Optimization and RL (CORL)

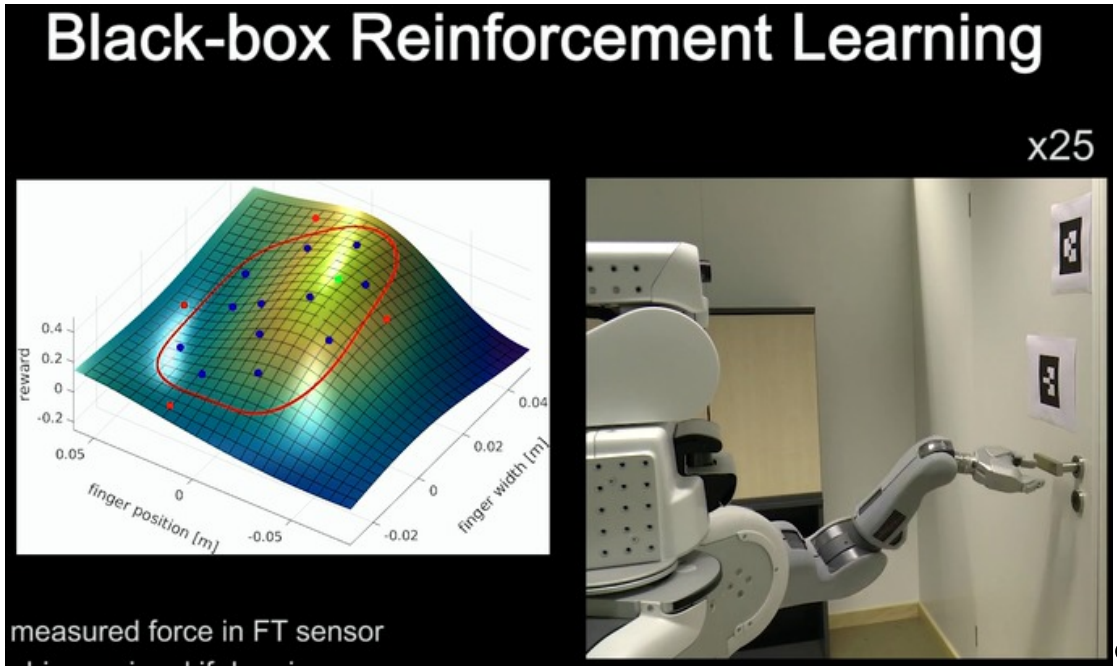
- CORL:
  - Policy parameters w
  - **projection**, implicitly given by a constraint $h(w, \theta) = 0$
  - **analytically known cost function** $J(w) = \mathsf{E}\{\sum_{t=0}^{T} c_t(x_t, u_t) \,|\, w\}$
  - **unknown black-box return function** $R(\theta) \in \mathbb{R}$
  - **unknown black-box success constraint** $S(\theta) \in \{0, 1\}$
  - Problem:

$$\min_{w,\theta} J(w) - R(\theta) \quad \text{s.t.} \quad h(w, \theta) = 0,\ S(\theta) = 1$$

- Alternate path optimization $\min_w J(w)$ s.t. $h(w, \theta) = 0$
  with Bayesian Optimization $\max_\theta R(\theta)$ s.t. $S(\theta) = 1$

Engert & Toussaint: *Combined Optimization and Reinforcement Learning for Manipulation Skills*. R:SS'16


reward
0.4
0.3
0.2
0.1
0
-0.1
0.05
0
-0.05
finger position [m]
0.04
0.02
0
-0.02
finger width [m]


Black-box Reinforcement Learning
x25
reward
0.4
0.2
0
-0.2
0.05
0
-0.05
finger position [m]
0.04
0.02
0
-0.02
finger width [m]
measured force in FT sensor

Inverse KKT to gain generalization

- Constrained optimization as the *generative assumption* of demonstrations

$$\min_{x_{0:T}} \quad \sum_{t=0}^{T} f_t(x_{t-k:t})^\top f_t(x_{t-k:t})$$
$$\text{s.t.} \quad \forall_t : \; g_t(x_{t-k:t}) \leq 0 \; , \quad h_t(x_{t-k:t}) = 0 \; .$$

- Problem:
  - Infer f_t from demonstrations
  - We assume f_t is parameterized by w
  - *Invert the KKT conditions* $\rightarrow$ QP over w's

Englert & Toussaint: *Inverse KKT – Learning Cost Functions of Manipulation Tasks from Demonstrations*. ISRR'15

Reduction to a Quadratic Program

$$\min_w \; w^\top \Lambda w \quad \text{s.t.} \quad w \geq 0$$

- Two ways to enforce a *non-singular* solution
  - Enforce positive-definiteness of Hessian at the demonstrations $\rightarrow$ maximize $\log |\nabla^2_x f(x)|$ (c.p. Levine & Koltun)
  - Add the constraint $\sum_i w_i \geq 1 \;\; \rightarrow \;\;$ Quadratic Program
- Even if $\Phi(x_{0:T}), g(x_{0:T}), h(x_{0:T})$ are a arbitrarily non-linear, this ends up a QP!

- Related work:
  - Levine & Koltun: Continuous inverse Optimal Control with Locally Optimal Examples. ICML'12
  - Puydupin-Jamin, Johnson & Bretl: A convex approach to inverse optimal control and its application to modeling human locomotion. ICRA'12
  - Jetchev & Toussaint: TRIC: Task space retrieval using inverse optimal control. Autonomous Robots, 2014.
  - Muhlig et al: Automatic selection of task spaces for imitation learning. IROS'09

Inverse KKT

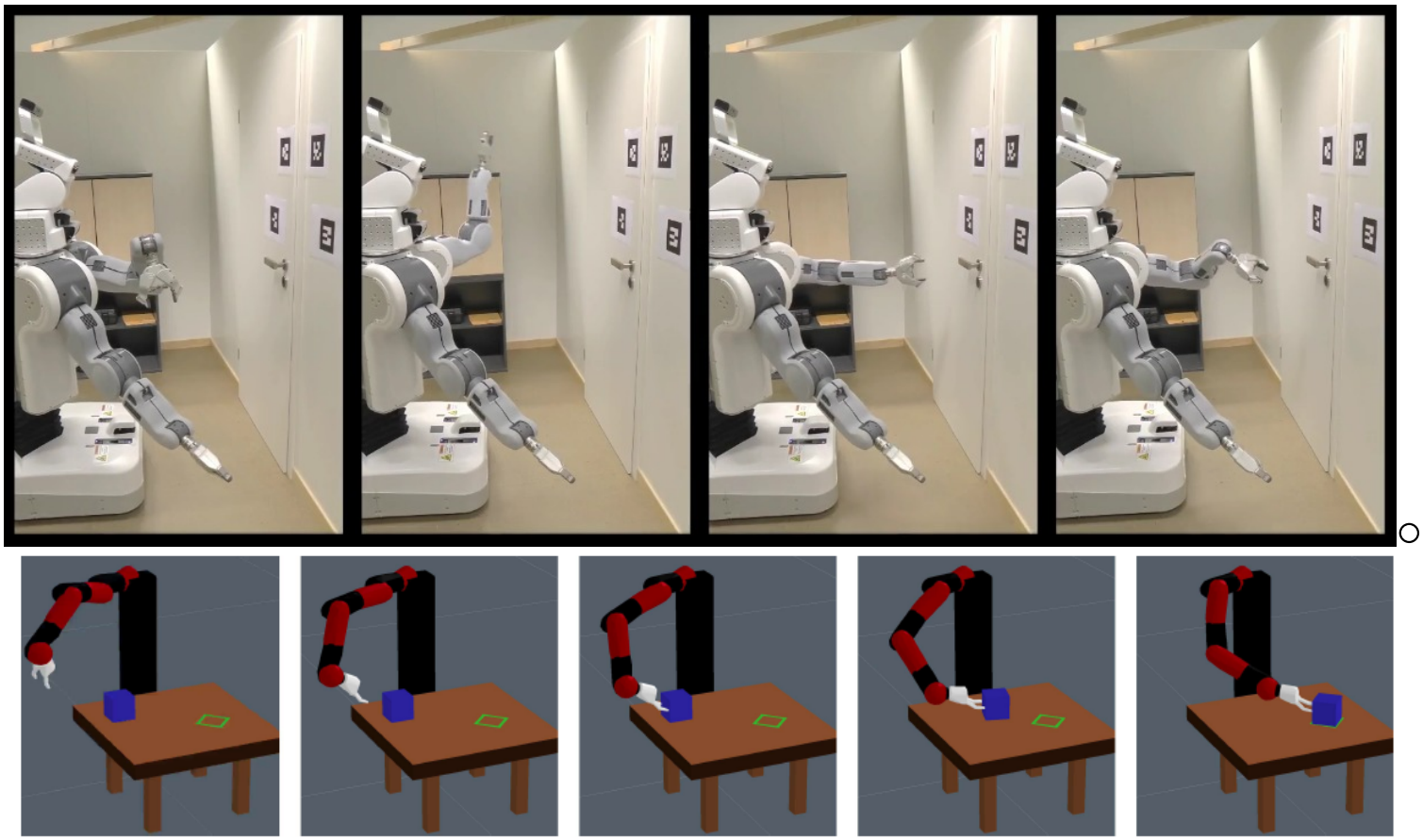


Fig. 3 These images show the box sliding motion of Section 5.2 where the goal of the task is to slide the blue box on the table to the green target region.

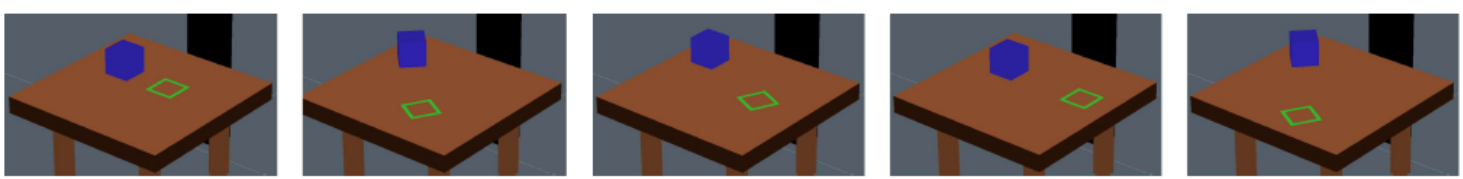


Fig. 4 Each image shows a different instance of the box sliding task. We were able to generalize to different initial box states (blue box) and to different final box targets (green area).

What can we learn from demonstrations?

What can we learn from demonstrations?

- Policies – imitation
- Goals, rewards, costs, values, preferences – IRL, IOC

What can we learn from demonstrations?

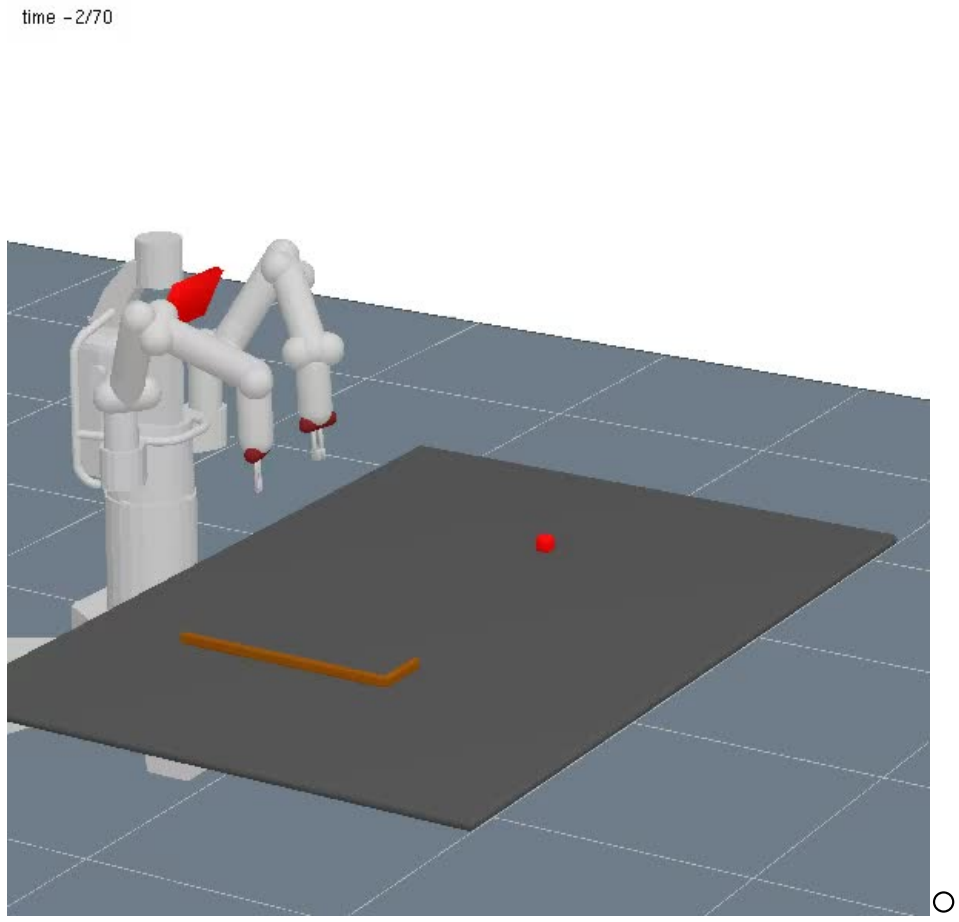
- Policies – imitation
- Goals, rewards, costs, values, preferences – IRL, IOC
- Models, dynamics?

What can we learn from demonstrations?

- Policies – imitation
- Goals, rewards, costs, values, preferences – IRL, IOC
- Models, dynamics?

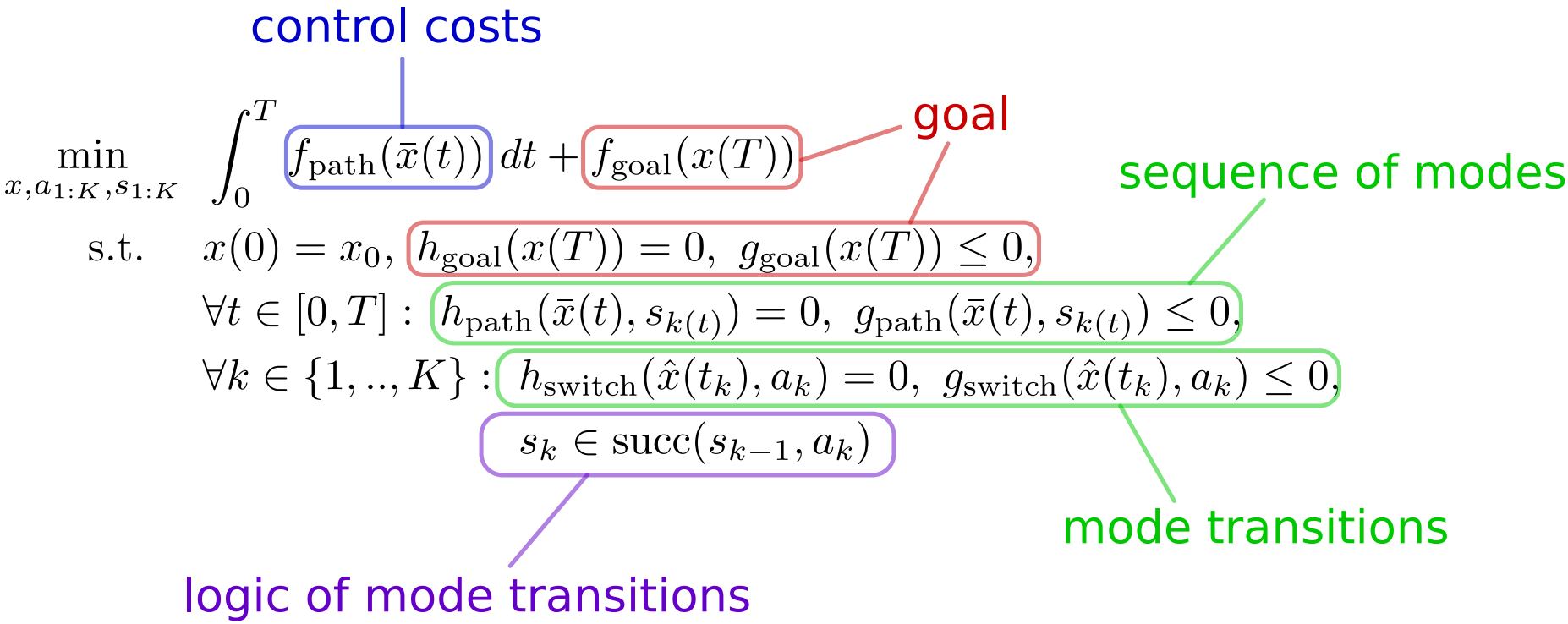
- anything that helps to make better decisions in the future
- anything that helps planning
  - Even when we know a (microscopic/analytical) model of the world, there are things to learn from demonstrations for more efficient planning

Physical Reasoning & Robot Manipulation



Toussaint, Allen, Smith, Tenenbaum: *Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning*. R:SS'18

Logic-Geometric Program



$$\min_{x, a_{1:K}, s_{1:K}} \int_0^T f_{\text{path}}(\bar{x}(t))\, dt + f_{\text{goal}}(x(T))$$

$$\text{s.t.} \quad x(0) = x_0,\ h_{\text{goal}}(x(T)) = 0,\ g_{\text{goal}}(x(T)) \le 0,$$

$$\forall t \in [0, T] : h_{\text{path}}(\bar{x}(t), s_{k(t)}) = 0,\ g_{\text{path}}(\bar{x}(t), s_{k(t)}) \le 0,$$

$$\forall k \in \{1, .., K\} : h_{\text{switch}}(\hat{x}(t_k), a_k) = 0,\ g_{\text{switch}}(\hat{x}(t_k), a_k) \le 0,$$

$$s_k \in \text{succ}(s_{k-1}, a_k)$$

- **Logic** to describe feasible sequences of modes
- **Modes** are grounded as differentiable constraints on the system dynamics
- Every *skeleton* $a_{1:K}$ defines a *smooth and tractable* NLP $\mathcal{P}(a_{1:K})$

"Logic of local optima"

Predicates to indicate modes

modes	(staFree X Y)	create stable free (7D) joint from X to Y
	(staOn X Y)	create stable 3D $xy\phi$ joint from X to Y
	(dynFree X)	create dynamic free joint from world to X
	(dynOn X Y)	create dynamic 3D $xy\phi$ joint from X to Y
	[impulse X Y]	impulse exchange equation
geometric	(touch X Y)	distance between X and Y equal 0
	(inside X Y)	point X is inside object Y $\rightarrow$ inequalities
	(above X Y)	Y supports X to not fall $\rightarrow$ inequalities
	(push X Y Z)	

dynFree, dynOn

$$M(q)\ddot{q}_q + F(q, \dot{q}) = 0$$

impulse

$$I_1\omega_1 - p_1 \times R = 0 \quad m_1 v_1 + m_2 v_2 = 0$$
$$I_2\omega_2 + p_2 \times R = 0 \quad (I - cc^\top)R = 0$$

Decision operators to sequence modes

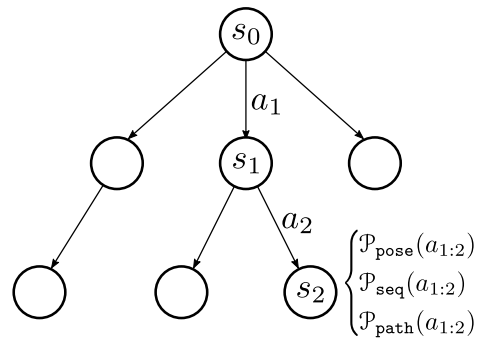
decisions	effects
grasp(X Y)	[touch X Y] (staFree X Y)
handover(X Y Z)	[touch Z Y] (staFree Z Y) !(staFree X Y)
place(X Y Z)	[above Y Z] (staOn Z Y) !(staFree X Y)
throw(X Y)	(dynFree Y) !(staFree X Y)
hit(X Y)	[touch X Y] [impulse X Y] (dynFree Y)
hitSlide(X Y Z)	[touch X Y] [impulse X Y] (above Y Z) (dynOn Y Z)
hitSlideSit(X Y Z)	“hitSlide(X Y Z)” “place(X Z)”
push(X, Y, Z)	komo(push X Y Z)

More predicates for preconditions: gripper, held, busy, animate, on, table

Multi-Bound Tree Search

- A NLP $\mathcal{P}$ describes $\min_x f(x)$ s.t. $g(x) \leq 0,\ h(x) = 0$
- **Definition:** $\hat{\mathcal{P}} \preceq \mathcal{P}$ (is lower bound) iff $[\mathcal{P}$ feas. $\Rightarrow\ \hat{\mathcal{P}}$ feas. $\wedge\ \hat{f}^* \leq f^*]$
- Every symbolic (sub-)sequence $s_{k:l}$ defines an NLP $\mathcal{P}(s_{k:l})$
- **Definition:** $\mathcal{P}$ seq. bounds itself iff $[s_{k:l} \subseteq s_{1:K} \Rightarrow \mathcal{P}(s_{k:l}) \preceq \mathcal{P}(s_{1:K})]$
- **Definition:** $(\mathcal{P}_1, .., \mathcal{P}_L)$ is a multi-bound iff $\forall_i :\ \mathcal{P}_i \preceq \mathcal{P}_{i+1}$ and $\mathcal{P}_i$ seq. bound

$\rightarrow$ Best-first search alternating over $\mathcal{P}_1, .., \mathcal{P}_L$



- Concrete bounds we use:

`sym`	symbolically feasible	$\ll 10$msec
`pose`	pose for last decision	$\sim 20 - 200$msec
`seq`	sequence of key poses for whole skeleton	$\sim 0.2 - 2$sec
`path`	full fine path for whole skeleton	~ 10sec

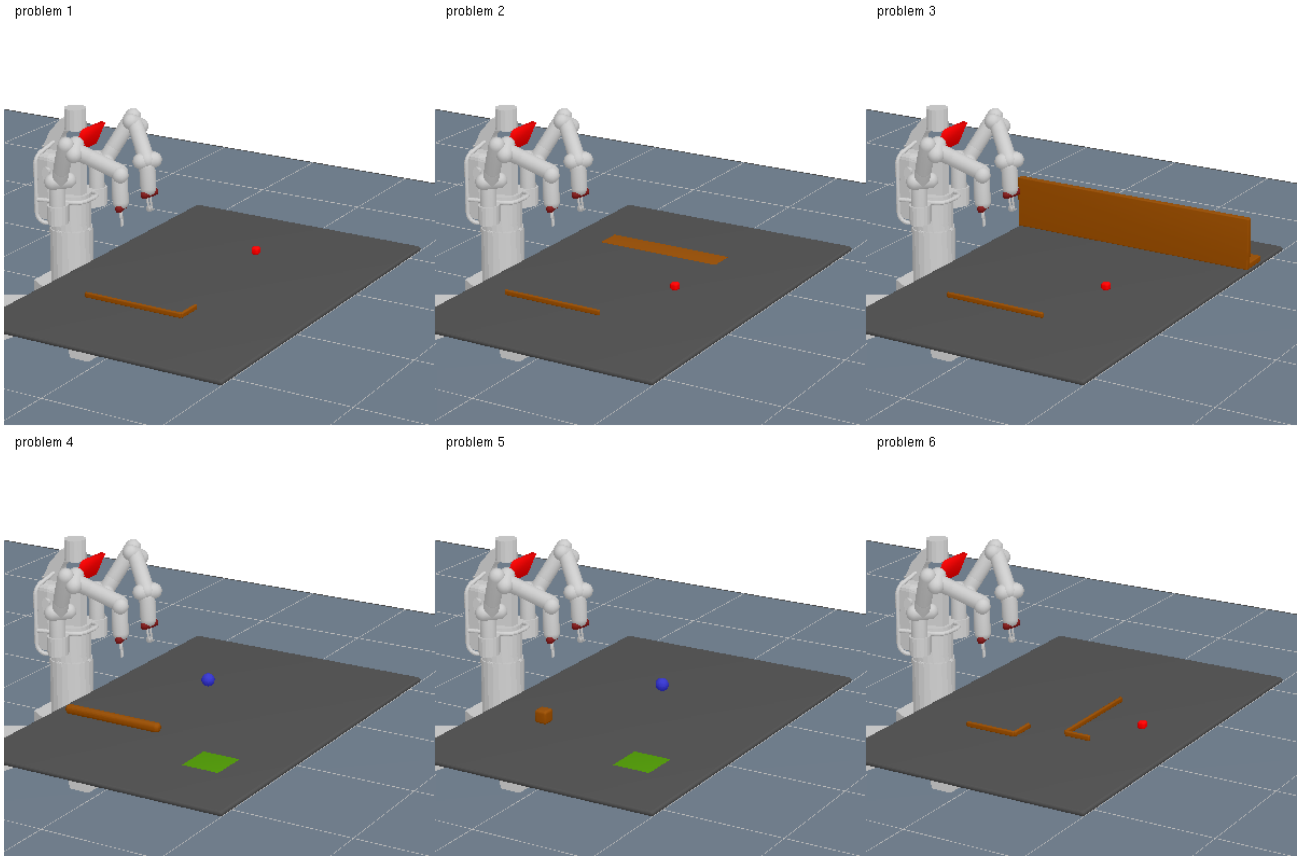
MBTS properties

- Optimality Guarantees? Yes, if...
  - we use strict best-first search to select on each level
  - we *could* solve the NLPs exactly (instead: mostly uni-modal, but no convexity guarantee)

- Possibilities to improve
  - novel cooperation with Erez Karpas (Technion, previously MIT)

    Karpaz et al: *Rational deployment of multiple heuristics in optimal state-space search*. AI 2018
  - integration with Fast Downward planning (STRIPS-stream; Garrett)
  - integration with Angelic Semantics (Marthi; Vega-Brown)

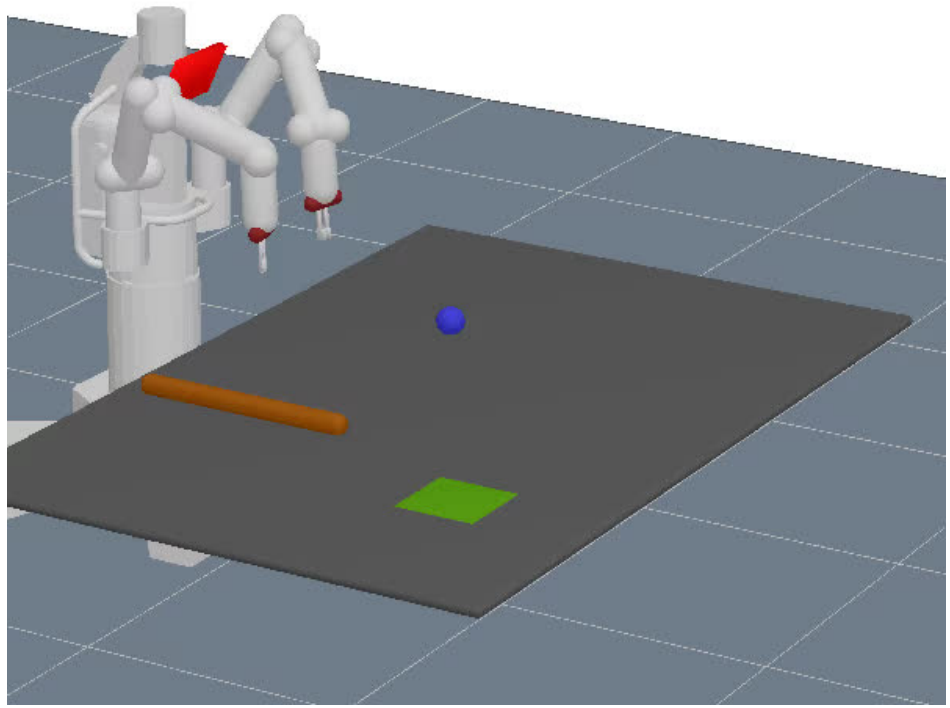
Experiments



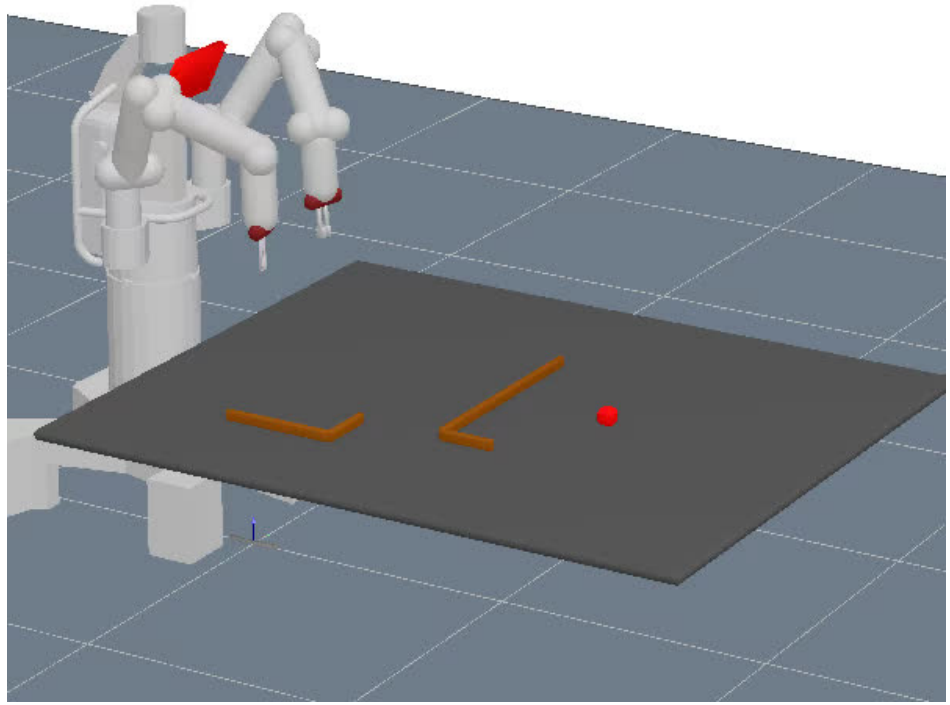
- Tree size at depth 4: (we limited logic for problem 6)

problem	1	2	3	4	5	6
tree size	12916	34564	7312	12242	12242	3386
branching	10.66	13.63	9.25	10.52	10.52	7.63

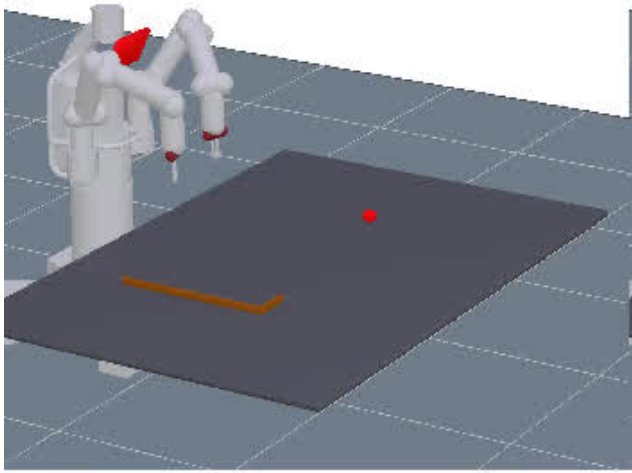
time -2/110



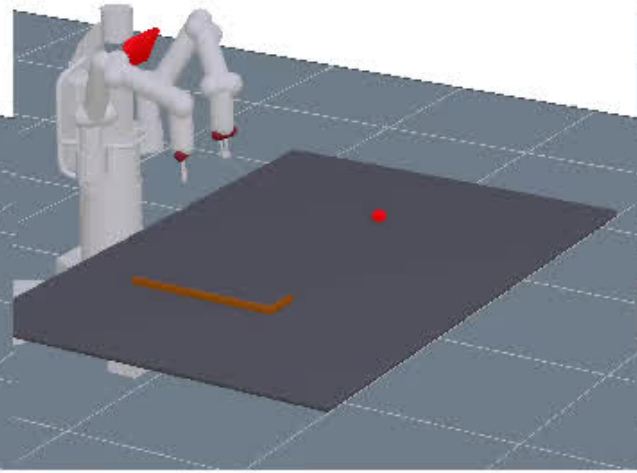
time - 2/130



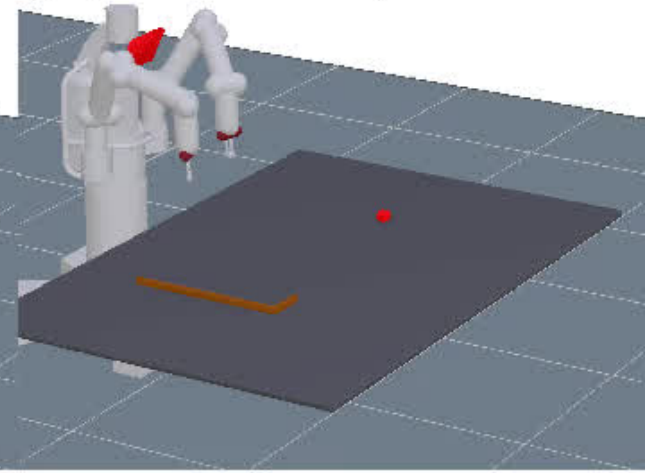
0:1: 0.3 1.14857 1.10575 2.07728 | 0.710069
(grasp baxterR stick)
(hitSlide stickTip redBall table1)
(grasp baxterL redBall)



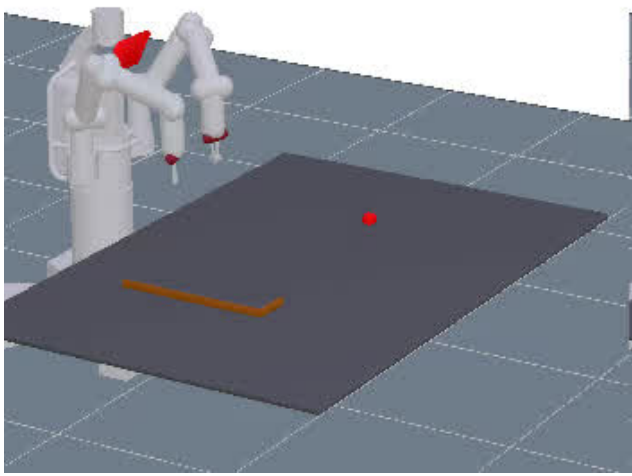
1:1: 0.3 1.02848 1.66055 2.42943 | 0.00944367
(grasp baxterR stick)
(push stickTip redBall table1)
(grasp baxterL redBall)



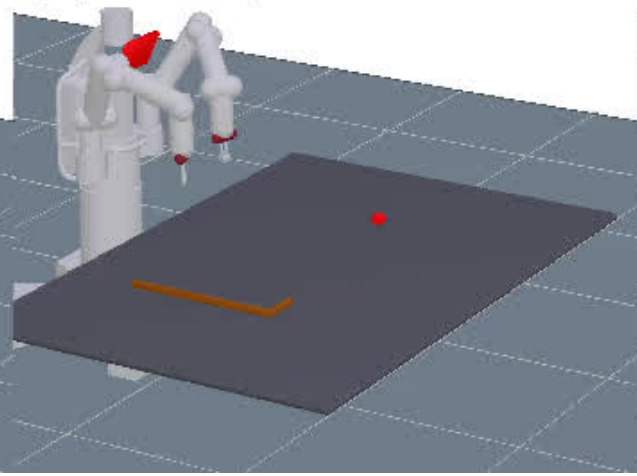
2:1: 0.4 1.16111 1.15196 2.48215 | 0.0207901
(grasp baxterR stick)
(handover baxterR stick baxterL)
(hitSlide stickTip redBall table1)
(graspSlide baxterR redBall table1)



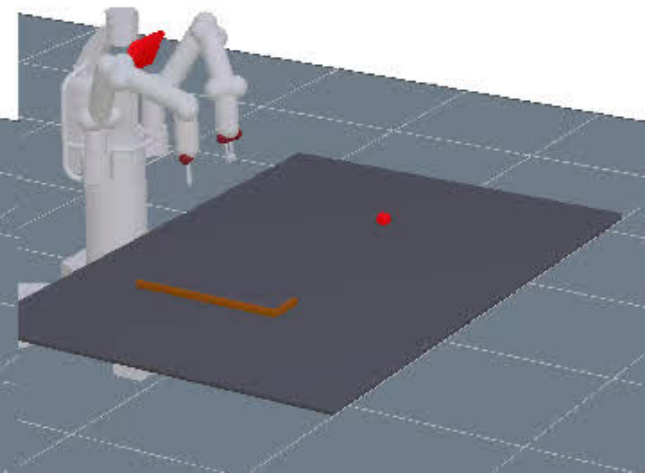
3:1: 0.3 1.14902 1.10464 2.54955 | 0.611458
(grasp baxterR stick)
(hitSlide stickTip redBall table1)
(graspSlide baxterL redBall table1)



4:1: 0.4 0.92368 2.01941 3.49634 | 0.0595839
(grasp baxterR stick)
(handover baxterR stick baxterL)
(push stickTip redBall table1)
(grasp baxterR redBall)

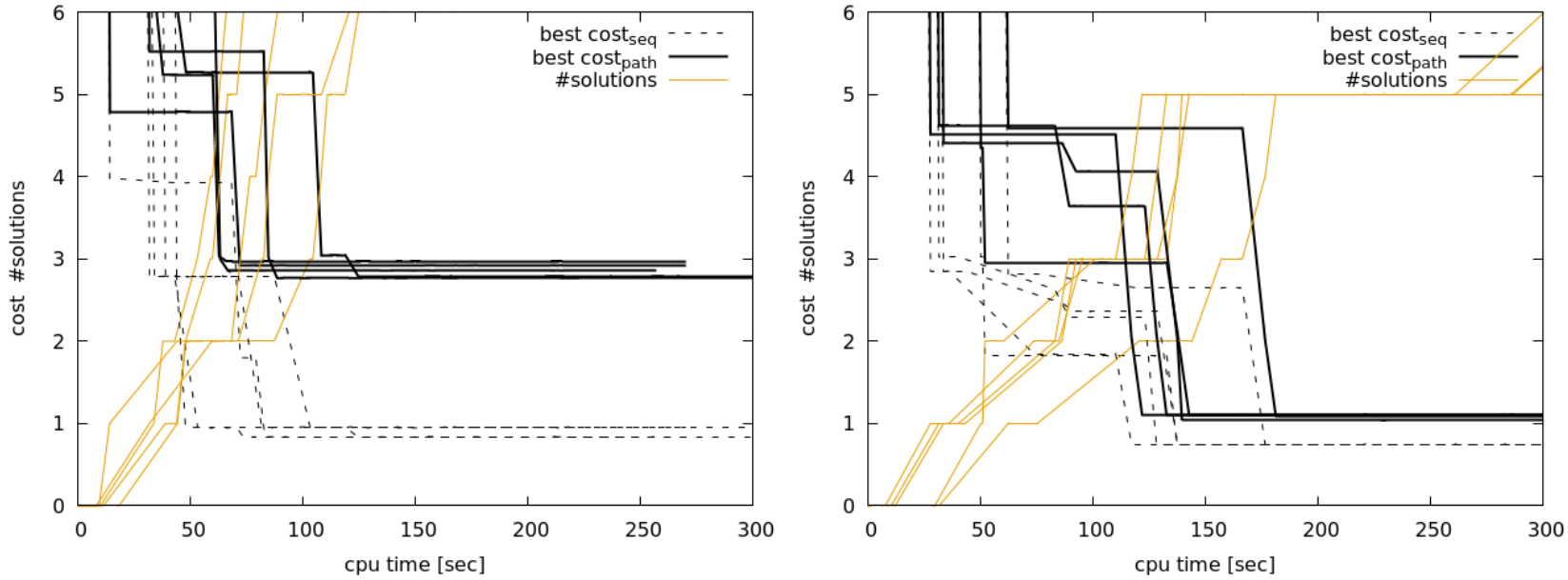


5:1: 0.3 1.14971 1.14327 2.7609 | 1.19
(graspSlide baxterR stick table1)
(hitSlide stickTip redBall table1)
(grasp baxterL redBall)



Run times

$$\sim 20 - 200\text{sec}$$



For 5 runs, cost of the best solution found, for bounds $\mathcal{P}_2$ and $\mathcal{P}_3$, over time

What is the common pattern?

What is the common pattern?

- 3D geometry
- objects
- objects either interact or not
  (interaction $\rightarrow$ contact)

- Different possible model of interaction:
  - Low-level analytical (forces, complementarities, contact physics)
  - quasi-static (stable relations)
  - kinematic & dynamic simplifications (pick, place, fly, slide)
  - effects (affordance learning)
  - neural networks (interaction networks, relation networks)

What can we learn from demonstrations?

- anything that helps planning
  - Even when we know a (microscopic/analytical) model of the world, there are things to learn from demonstrations for more efficient planning

What can we learn from demonstrations?

- anything that helps planning
  - Even when we know a (microscopic/analytical) model of the world, there are things to learn from demonstrations for more efficient planning

- Learn possible interactions models that are plannable

Summary

- Policies – imitation
- Goals, rewards, costs, values, preferences – IRL, IOC
- Models of interaction, even when we have an analytical model of the world

Summary

- Policies – imitation
- Goals, rewards, costs, values, preferences – IRL, IOC
- Models of interaction, even when we have an analytical model of the world

- Demonstrations may come from
  - Active exploration
  - Low-level, computationally heavy planning

Summary

- Policies – imitation
- Goals, rewards, costs, values, preferences – IRL, IOC
- Models of interaction, even when we have an analytical model of the world

- Demonstrations may come from
  - Active exploration
  - Low-level, computationally heavy planning
- Perception of interaction modes

Thanks

- *for your attention!*

- to co-authors & collaborators:

Peter Englert

Manuel Lopes
Thibaut Munzer
Yoan Mollard
Andrea Baisero
Baptiste Bush

Kelsey R Allen
Kevin A Smith
Josh B Tenenbaum
Tomas Lozano-Pérez
Leslie Pack Kaelbling