Learning and Reasoning in the Physical World

Marc Toussaint

Machine Learning & Robotics Lab – University of Stuttgart

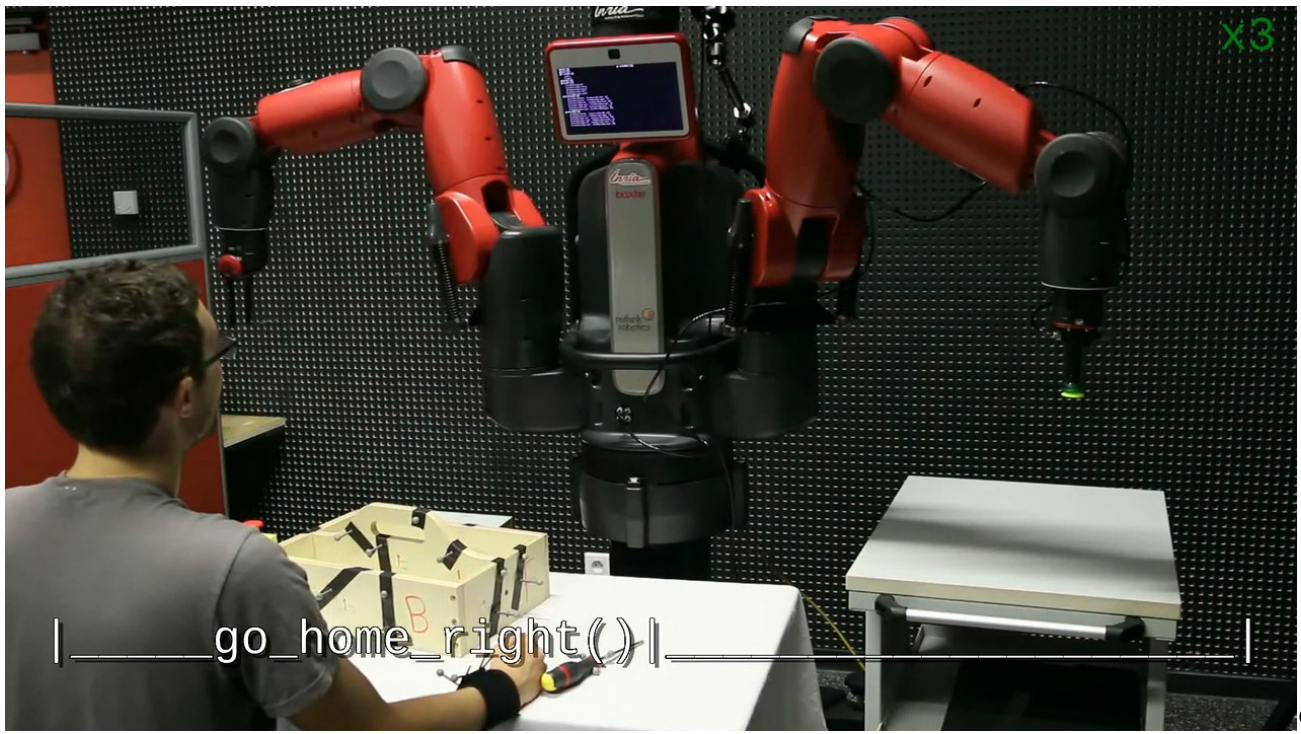
AI Lecture, Stuttgart, Jan 28, 2018

Outline

- Briefly: Work on
  - robot manipulation learning, learning from demonstration
  - relational MDPs
  - active learning

Learning from Few Samples

- Cooperative Manipulation Learning
- Relational imitation & inverse reinforcement learning



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

Busch, Toussaint, Lopes: *Planning Ergonomic Sequences of Actions in Human-Robot Interaction*. ICRA'18

Methods involved

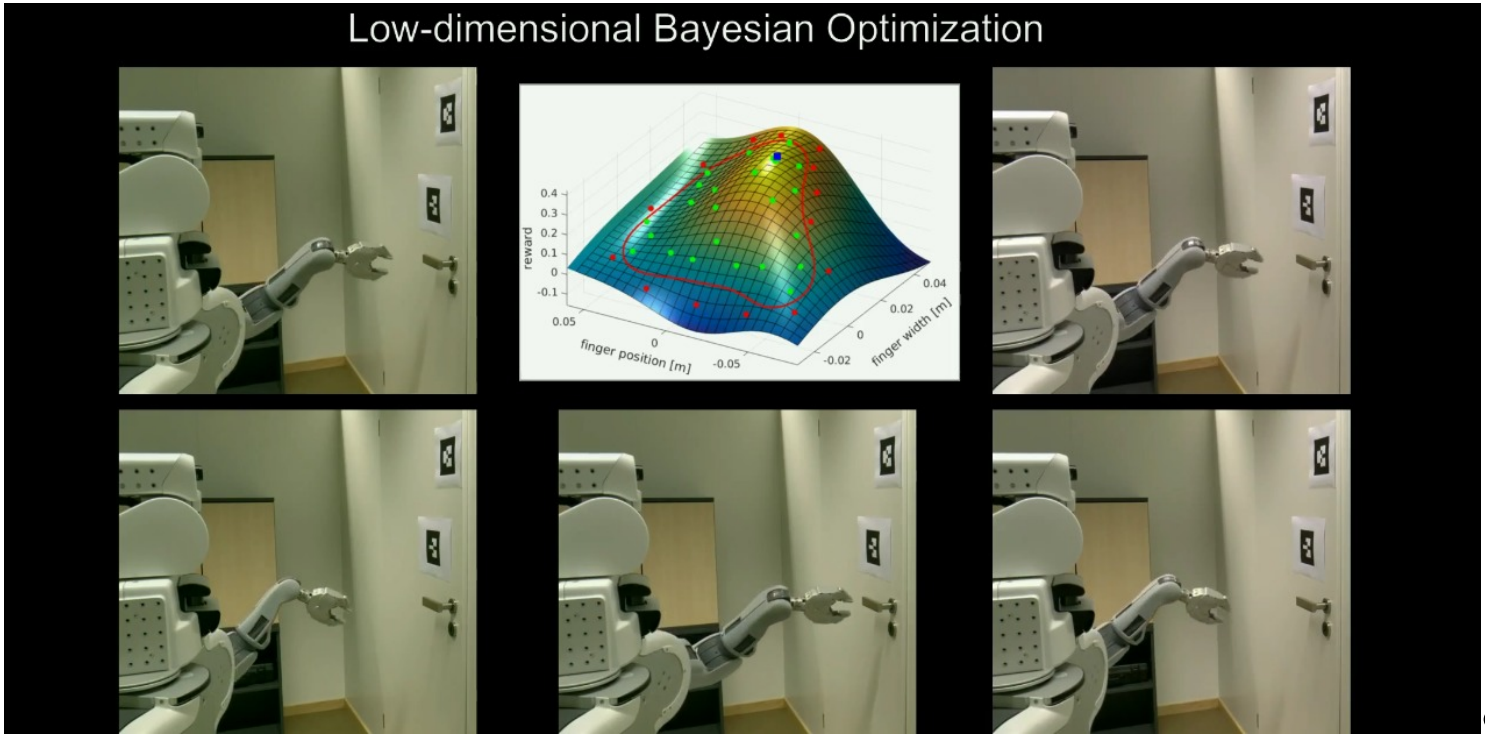
- Relational Activity Processes
  - The current *state* lists the current activities (relational (1st-order logic)):
    `(object Handle), (free humanLeft), (humanLeft graspingScrew)=1.0, (humanRight grasped Handle), (Handle held), (robot releasing Long1)=1.5,`
- This defines a decision process, which initiates, waits, and terminates activities of all agents, and predicts the effects.
- Tree Search to reasons about *decisions* (for all agents!)
- Reduction to relational semi-MDP to realize Inverse Reinforcement Learning (using Tree Boosted Relational Imitation Learning)

Munzer, Toussaint, Lopes: *Preference learning on the execution of collaborative human-robot tasks*. ICRA'17

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

Learning from Few Samples

- Combine analytical optimization with black-box BayesOpt
- Invert the KKT conditions to learn from demonstration



Englert, Vien, Toussaint: *Inverse KKT: Learning cost functions of manipulation tasks from demonstrations*. IJRR 2017

Englert, Toussaint: *Learning manipulation skills from a single demonstration*. IJRR 2018

Methods involved

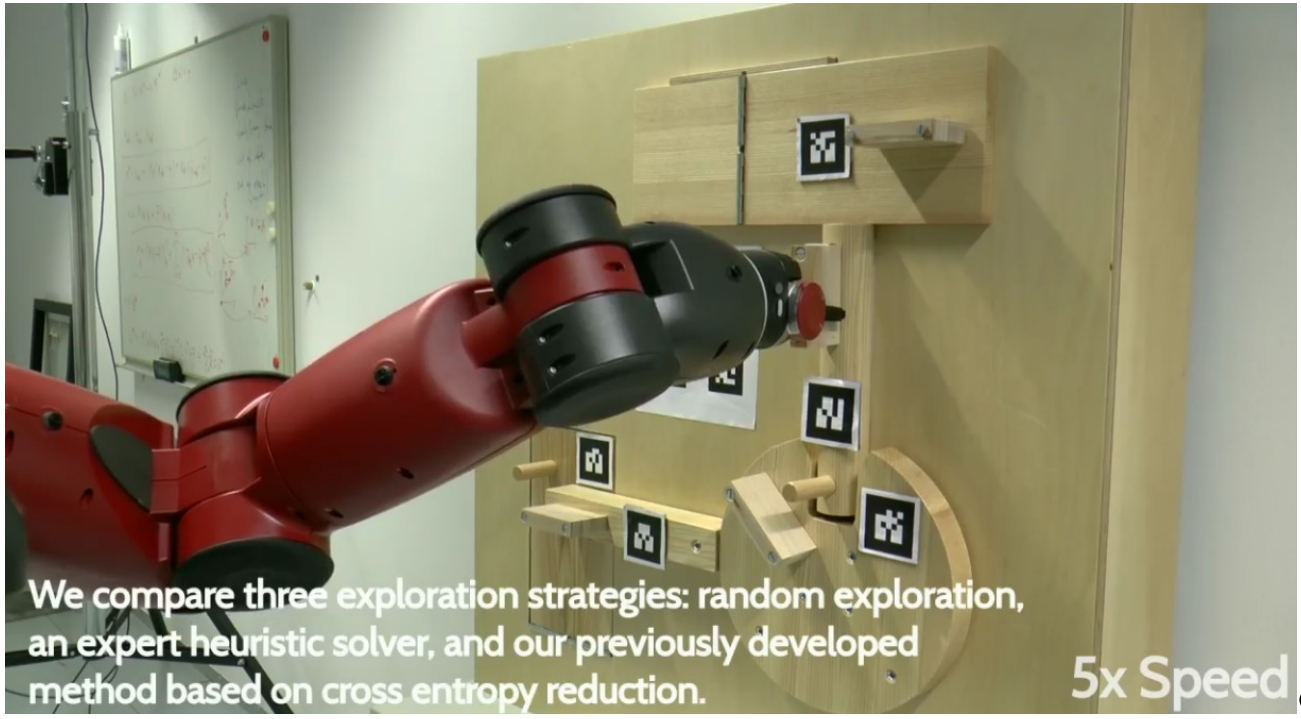
- Constrained optimization (KOMO) to generate motions
- Bayesian Optimization to search for good interaction parameters
- Inverting the KKT conditions for Inverse Reinforcement Learning

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

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

Learning from Few Samples

- Active Learning of Kinematic Mechanisms
- Bayesian inference over kinematic structures for active learning



Baum et al.: *Opening a Lockbox through Physical Exploration*. Humanoids'17

Kulick, Otte, Toussaint: *Active Exploration of Joint Dependency Structures*. ICRA'15

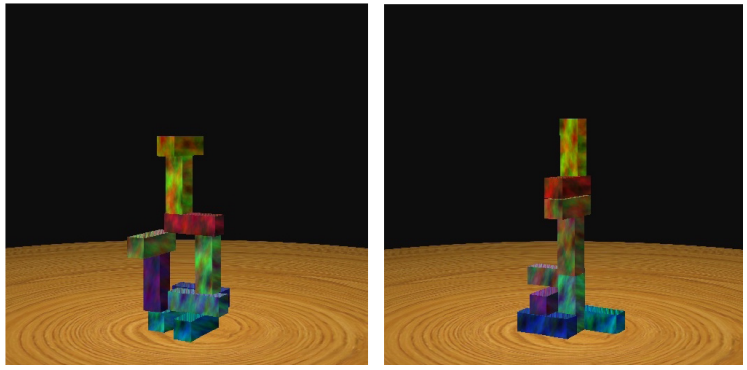
Methods involved

- Graphical Modes to represent what we know about the mechanism
- Probabilistic Inference to estimate information gain for potential next actions (active learning)

- All three lines of work exploit some understanding of the domain for sample efficiency

- But what is the fundamental structure of robot-world interaction?

Physical Reasoning & Manipulation



Battaglia, Hamrick & Tenenbaum, PNAS'13

(Wolfgang Köhler, 1917)

- What are computational models for physical reasoning?
- Reason about anything doable in a Newtonian world

Why is this interesting to study?

Why is this interesting to study?

- Physical Reasoning is under-researched
  - Lots of methodologies for physical modelling, but not reasoning
  - Focus of main-stream RL: specific skills $\rightarrow$ generalization to anything conceivable in a Newtonian world
  - Robotics: task and motion planning
  - Cognitive Science needs models

Why is this interesting to study?

- Physical Reasoning is under-researched
  - Lots of methodologies for physical modelling, but not reasoning
  - Focus of main-stream RL: specific skills $\rightarrow$ generalization to anything conceivable in a Newtonian world
  - Robotics: task and motion planning
  - Cognitive Science needs models

- Core challenge in robotics

Inverting Physics

- In analogy to inverting graphics
  Given desired outcomes, what inputs do we have to send to physics?

Inverting Physics

- In analogy to inverting graphics
  Given desired outcomes, what inputs do we have to send to physics?

- Differentiable Physics:
  - Todorov: A convex, smooth and invertible contact model for trajectory optimization. ICRA'11
  - de Avila Belbute-Peres & Kolter: A Modular Differentiable [..] Physics Engine. NIPS'17 workshop
  - Mordatch et al: Discovery of complex behaviors through contact-invariant optimization. TOG'12
  - Note: Local(!) differentiation through KKT conditions of constrained optimization

- *Gradients are powerful, but can they alone solve our problem?*

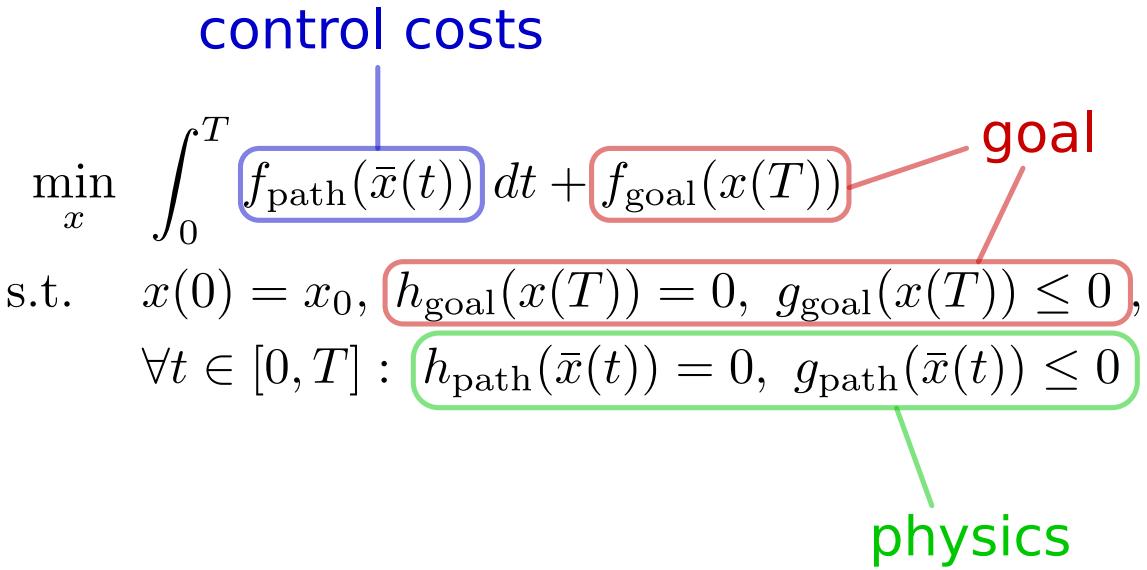
Inverting Physics

- In analogy to inverting graphics
  Given desired outcomes, what inputs do we have to send to physics?

- Differentiable Physics:
  - Todorov: A convex, smooth and invertible contact model for trajectory optimization. ICRA'11
  - de Avila Belbute-Peres & Kolter: A Modular Differentiable [..] Physics Engine. NIPS'17 workshop
  - Mordatch et al: Discovery of complex behaviors through contact-invariant optimization. TOG'12
  - Note: Local(!) differentiation through KKT conditions of constrained optimization

- *Gradients are powerful, but can they alone solve our problem?*
  - would contradict known complexity of task and motion planning
  - 'zero gradients' or local optima
  - discrete decisions translate to *combinatorics of local optima*

Unstructured Problem Formulation



- configuration space $\mathcal{X} = \mathbb{R}^n \times SE(3)^m$
- path $x : [0,T] \to \mathcal{X}$
- $\bar{x}(t) = (x(t), \dot{x}(t), \ddot{x}(t))$
- $(g,h)_{\text{path}}$: physics
- $(f,h,g)_{\text{goal}}$: objectives

Logic-Geometric Program

control costs

goal

sequence of modes

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

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

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

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

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

mode transitions

logic of mode transitions

- **Logic** to describe *possible* sequences of modes
- **Modes** are differentiable constraints on the path
- Every *skeleton* $a_{1:K}$ defines a *smooth and tractable* NLP

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

A Logic of Path Constraints

- The core categorical decision: (touch X Y)
- Finite types of interaction:
  - Stable relation
  - Inertial dynamics
  - Impulse or force exchange
  - etc

- Symbols to impose modes & constraints:

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

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

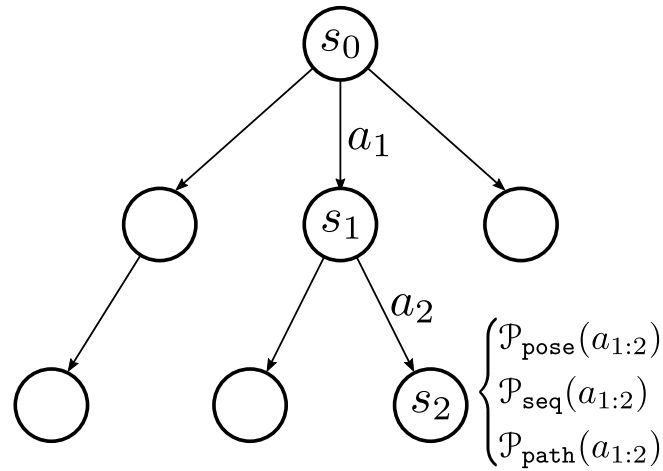
- Decision operators to sequence modes:

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



- Concrete bounds we use:

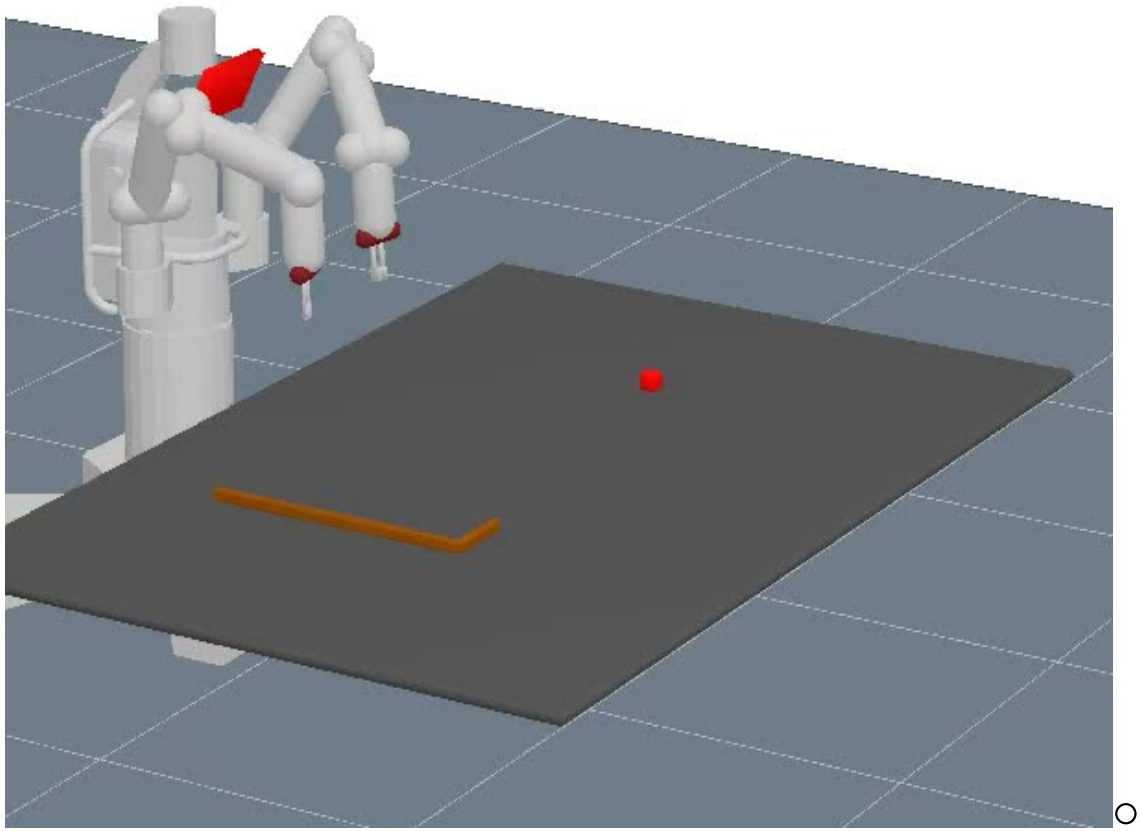
$\mathcal{P}_0$	`sym`	symbolically feasible	$\ll 10$msec
$\mathcal{P}_1$	`pose`	pose for last decision	$\sim 20 - 200$msec
$\mathcal{P}_2$	`seq`	sequence of key poses for whole skeleton	$\sim 0.2 - 2$sec
$\mathcal{P}_3$	`path`	full fine path for whole skeleton	~ 10sec

MBTS properties

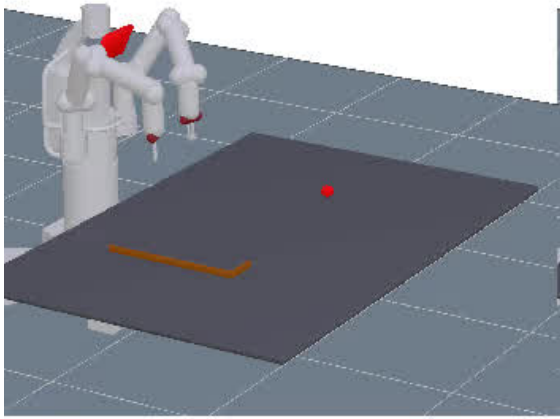
- Optimality Guarantees? Yes, if...
  - we *could* solve the NLPs exactly (instead: mostly uni-modal, but no convexity guarantee)

- Possibilities to improve
  - cooperation with Erez Karpas (Technion)

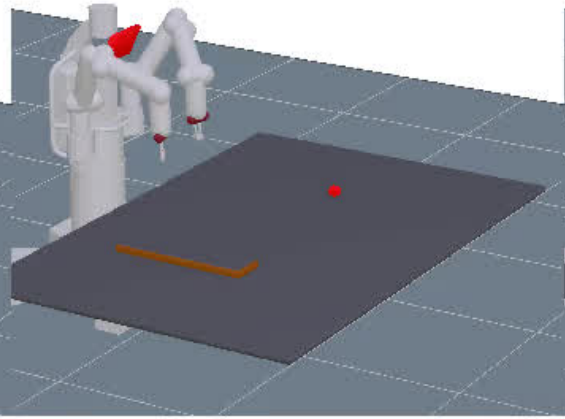
    Karpaz et al: *Rational deployment of multiple heuristics in optimal state-space search*. AI 2018
  - integration with Fast Downward planning (STRIPS-stream; Garrett)
  - integration with Angelic Semantics (Marthi; Vega-Brown)


time - 2/70

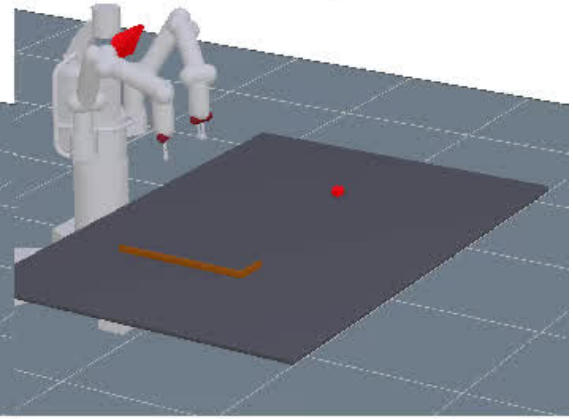
0:1: 0.3 1.14857 1.10575 2.07728 | 0.710069
(grasp baxterR stick)
(hitSlide stickTip redBall table1)
(grasp baxterL redBall)



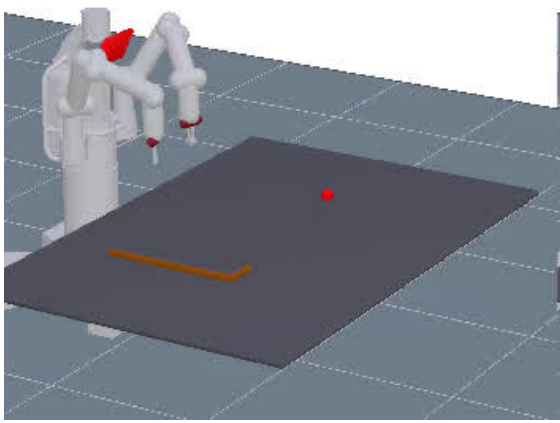
1:1: 0.3 1.02848 1.66055 2.42943 | 0.00944367
(grasp baxterR stick)
(push stickTip redBall table1)
(grasp baxterL redBall)



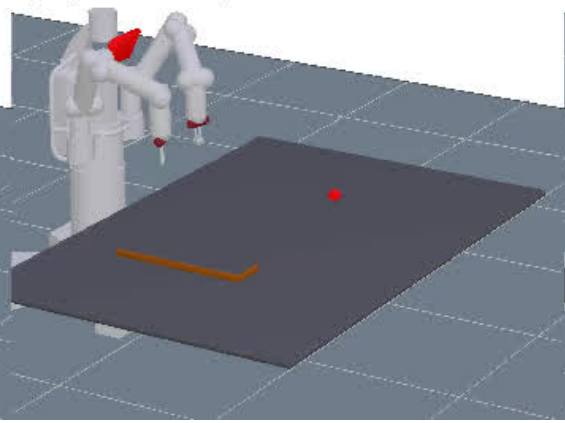
2:1: 0.4 1.16111 1.15196 2.48215 | 0.0207901
(grasp baxterR stick)
(handover baxterR stick baxterL)
(hitSlide stickTip redBall table1)
(graspSlide baxterR redBall table1)



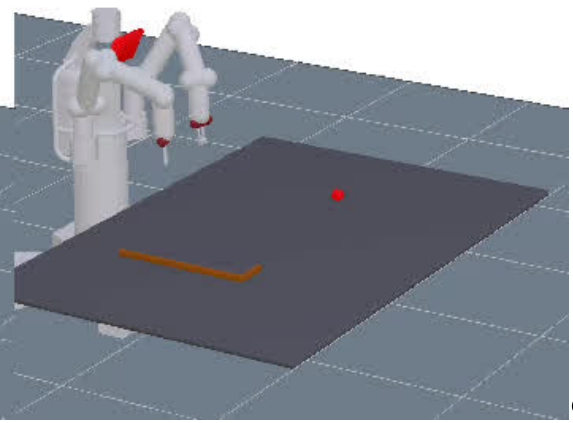
3:1: 0.3 1.14902 1.10464 2.54955 | 0.611458
(grasp baxterR stick)
(hitSlide stickTip redBall table1)
(graspSlide baxterL redBall table1)

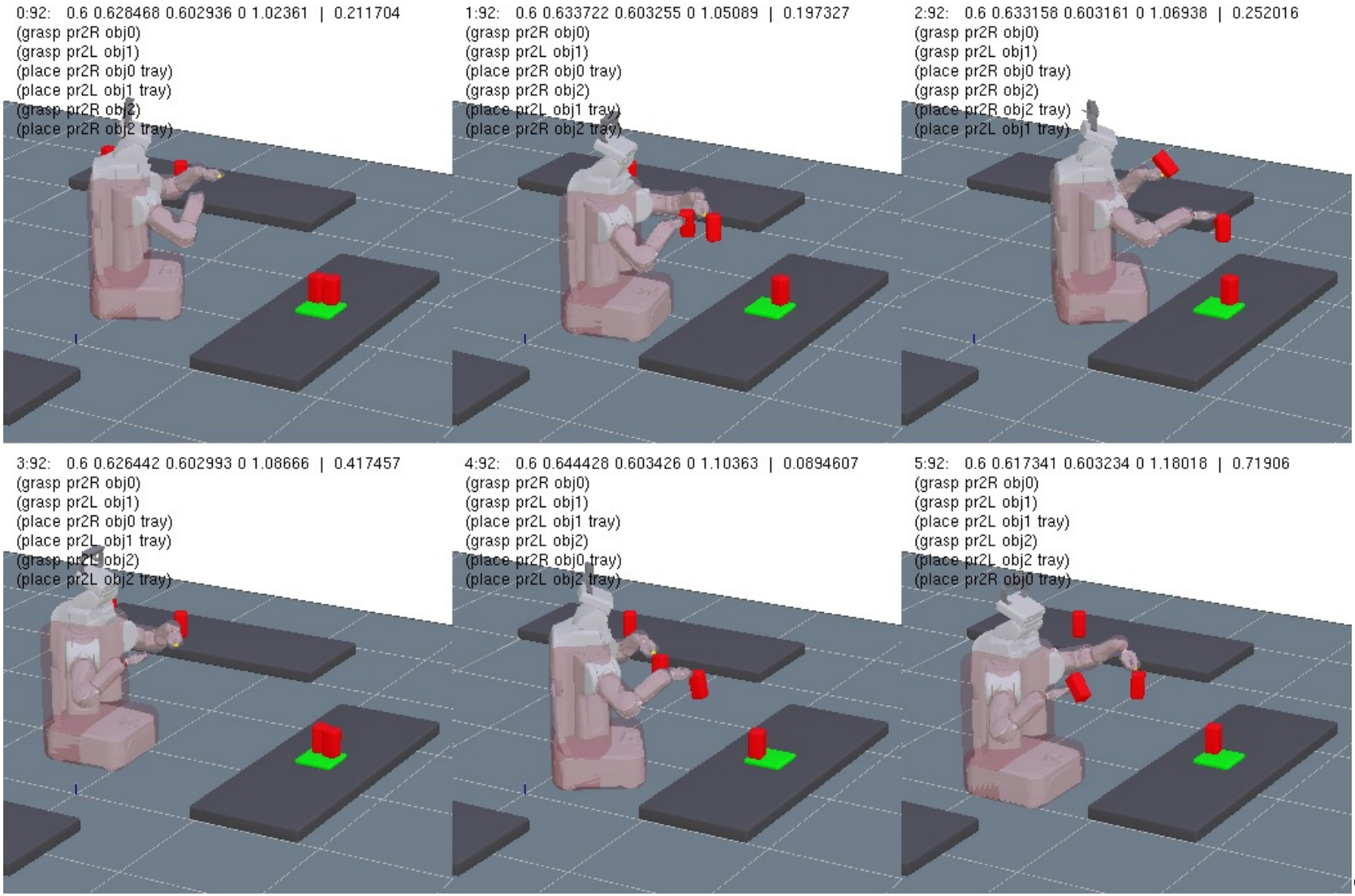


4:1: 0.4 0.92368 2.01941 3.49634 | 0.0595839
(grasp baxterR stick)
(handover baxterR stick baxterL)
(push stickTip redBall table1)
(grasp baxterR redBall)

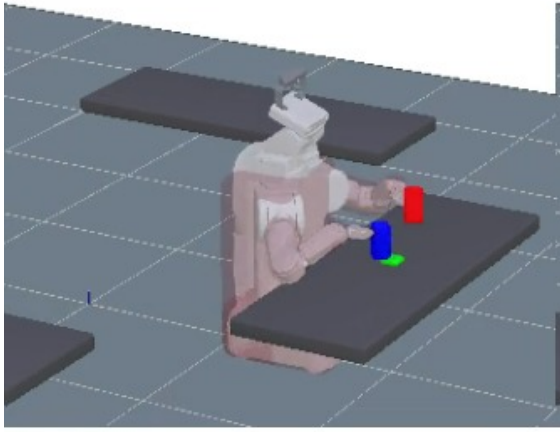


5:1: 0.3 1.14971 1.14327 2.7609 | 1.19
(graspSlide baxterR stick table1)
(hitSlide stickTip redBall table1)
(grasp baxterL redBall)

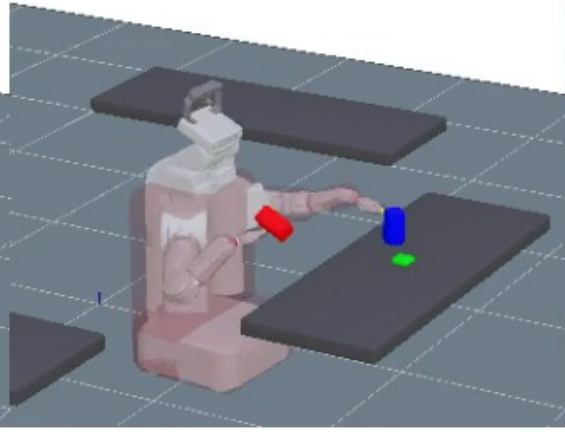



0:92: 0.6 0.628468 0.602936 0 1.02361 | 0.211704
(grasp pr2R obj0)
(grasp pr2L obj1)
(place pr2R obj0 tray)
(place pr2L obj1 tray)
(grasp pr2R obj2)
(place pr2R obj2 tray)
1:92: 0.6 0.633722 0.603255 0 1.05089 | 0.197327
(grasp pr2R obj0)
(grasp pr2L obj1)
(place pr2R obj0 tray)
(grasp pr2R obj2)
(place pr2L obj1 tray)
(place pr2R obj2 tray)
2:92: 0.6 0.633158 0.603161 0 1.06938 | 0.252016
(grasp pr2R obj0)
(grasp pr2L obj1)
(place pr2R obj0 tray)
(grasp pr2R obj2)
(place pr2R obj2 tray)
(place pr2L obj1 tray)
3:92: 0.6 0.626442 0.602993 0 1.08666 | 0.417457
(grasp pr2R obj0)
(grasp pr2L obj1)
(place pr2R obj0 tray)
(place pr2L obj1 tray)
(grasp pr2L obj2)
(place pr2L obj2 tray)
4:92: 0.6 0.644428 0.603426 0 1.10363 | 0.0894607
(grasp pr2R obj0)
(grasp pr2L obj1)
(place pr2L obj1 tray)
(grasp pr2L obj2)
(place pr2R obj0 tray)
(place pr2L obj2 tray)
5:92: 0.6 0.617341 0.603234 0 1.18018 | 0.71906
(grasp pr2R obj0)
(grasp pr2L obj1)
(place pr2L obj1 tray)
(grasp pr2L obj2)
(place pr2L obj2 tray)
(place pr2R obj0 tray)

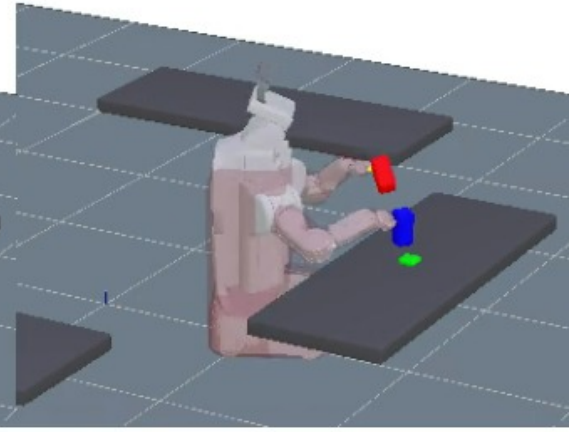
0:57: 0.3 0.350308 0.301882 0 0.469769 | 0.0812076
(grasp pr2R obj0)
(grasp pr2L obj3)
(place pr2R obj0 tray)



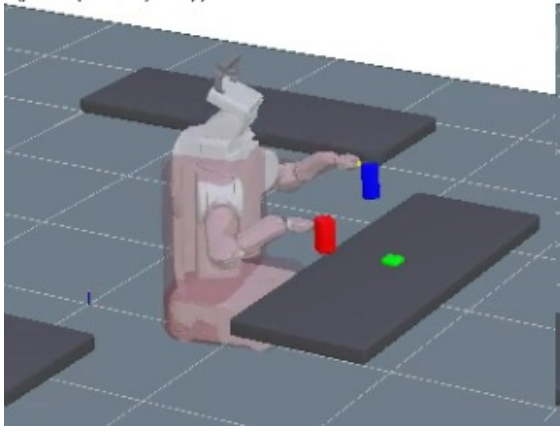
1:57: 0.3 0.307726 0.30273 0 0.508466 | 0.21674
(grasp pr2R obj3)
(grasp pr2L obj0)
(place pr2L obj0 tray)



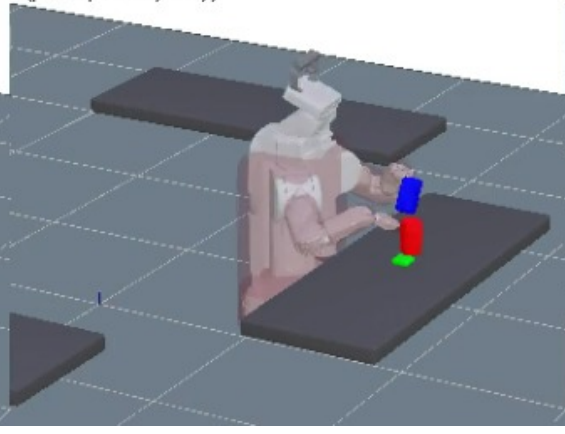
2:57: 0.3 0.311509 0.302527 0 0.547901 | 0.226081
(grasp pr2L obj3)
(grasp pr2R obj0)
(place pr2R obj0 tray)



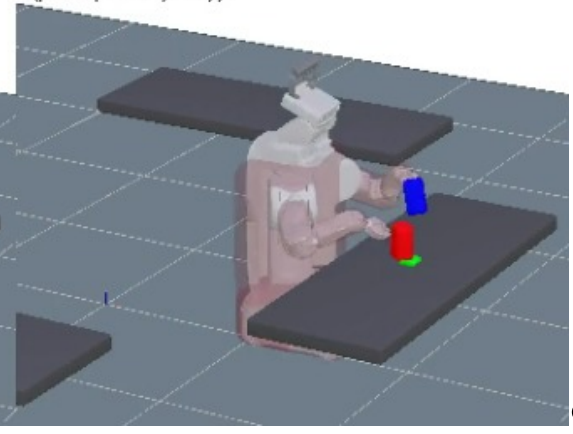
3:57: 0.4 0.414375 0.401737 0 0.56091 | 0.244107
(grasp pr2R obj3)
(grasp pr2L obj0)
(place pr2R obj3 table2)
(place pr2L obj0 tray)



4:57: 0.4 0.409768 0.401655 0 0.564126 | 0.469622
(grasp pr2L obj0)
(grasp pr2R obj3)
(place pr2R obj3 table2)
(place pr2L obj0 tray)

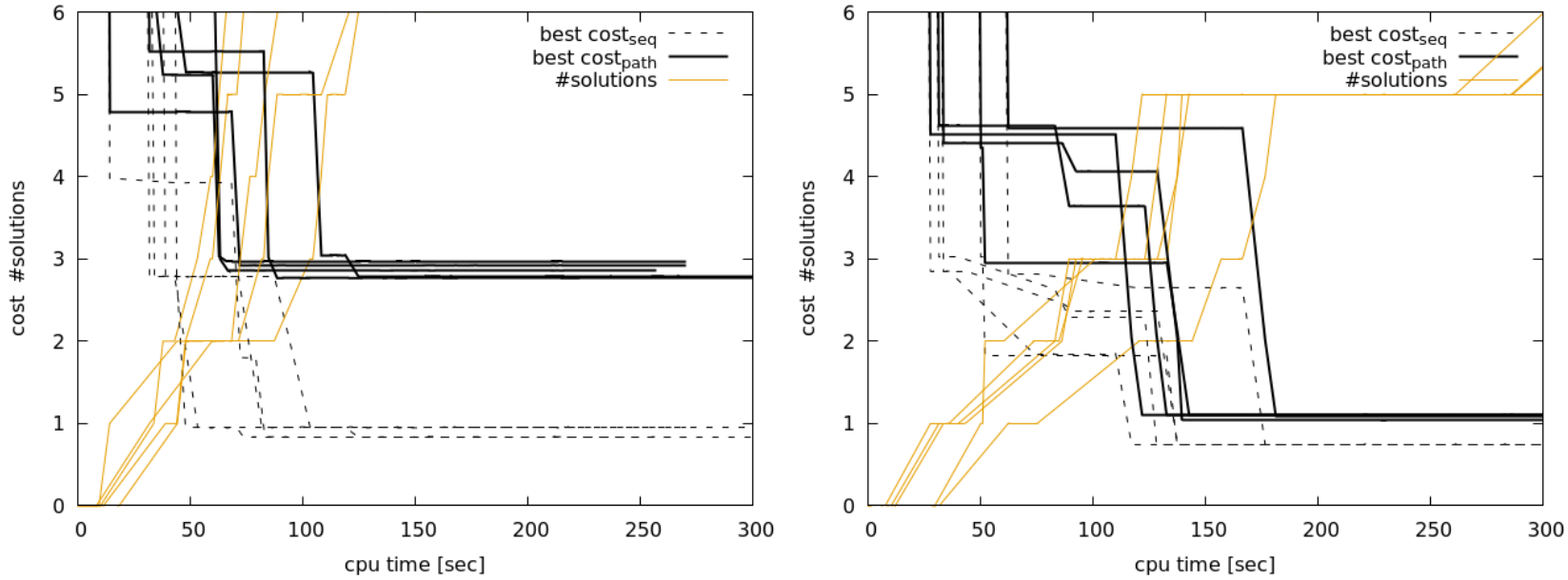


5:57: 0.4 0.409976 0.401518 0 0.56905 | 0.267901
(grasp pr2L obj0)
(grasp pr2R obj3)
(place pr2R obj3 tray)
(place pr2L obj0 tray)



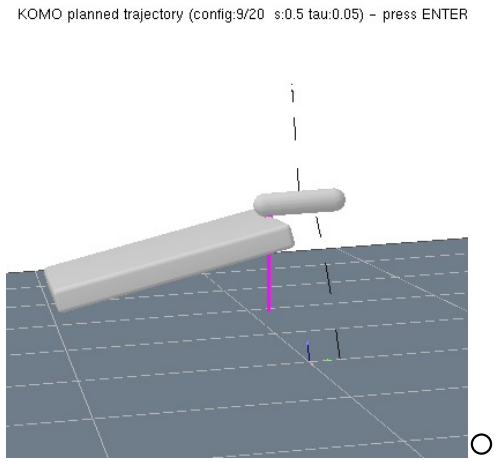
Run times

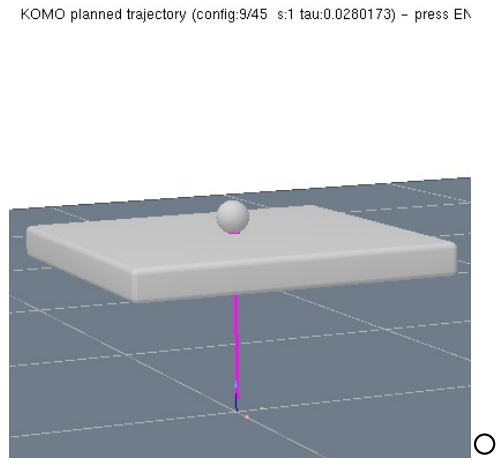
$$\sim 20 - 200\text{sec}$$

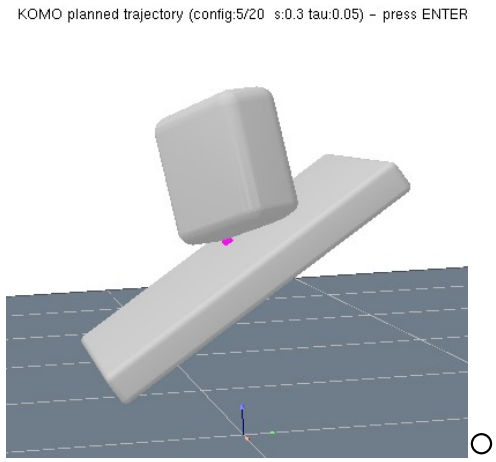


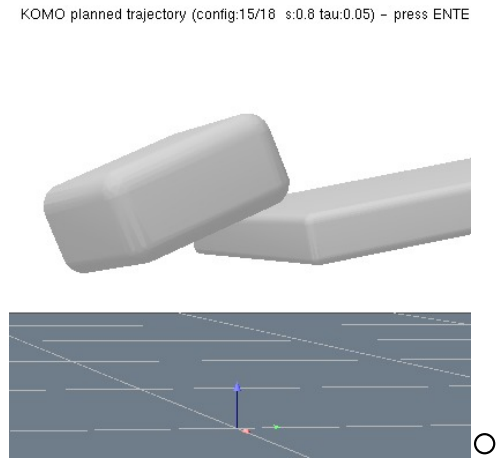
For 5 runs, cost of the best solution found, for bounds $\mathcal{P}_2$ and $\mathcal{P}_3$, over time

Other interaction types – all differentiable





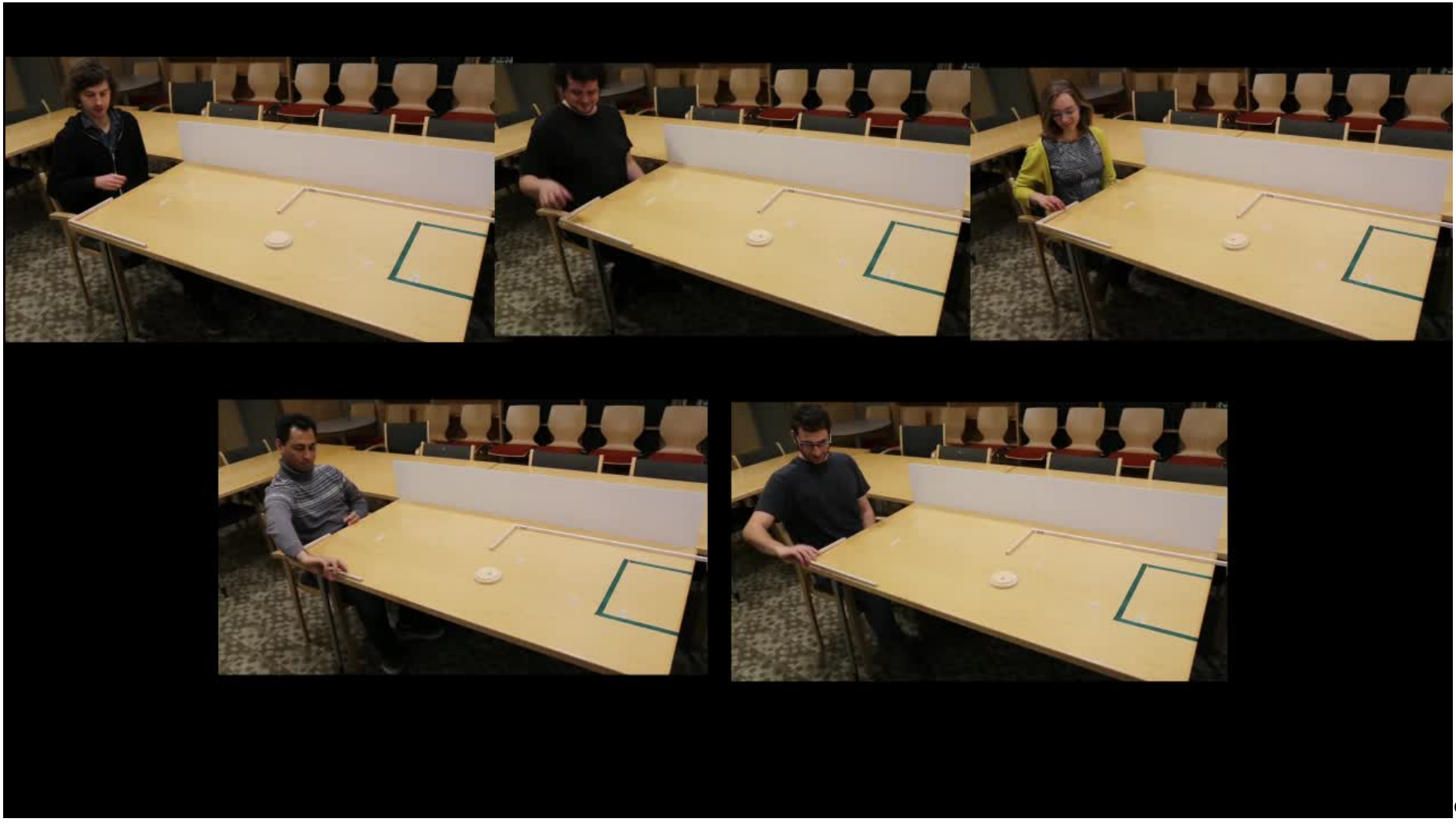




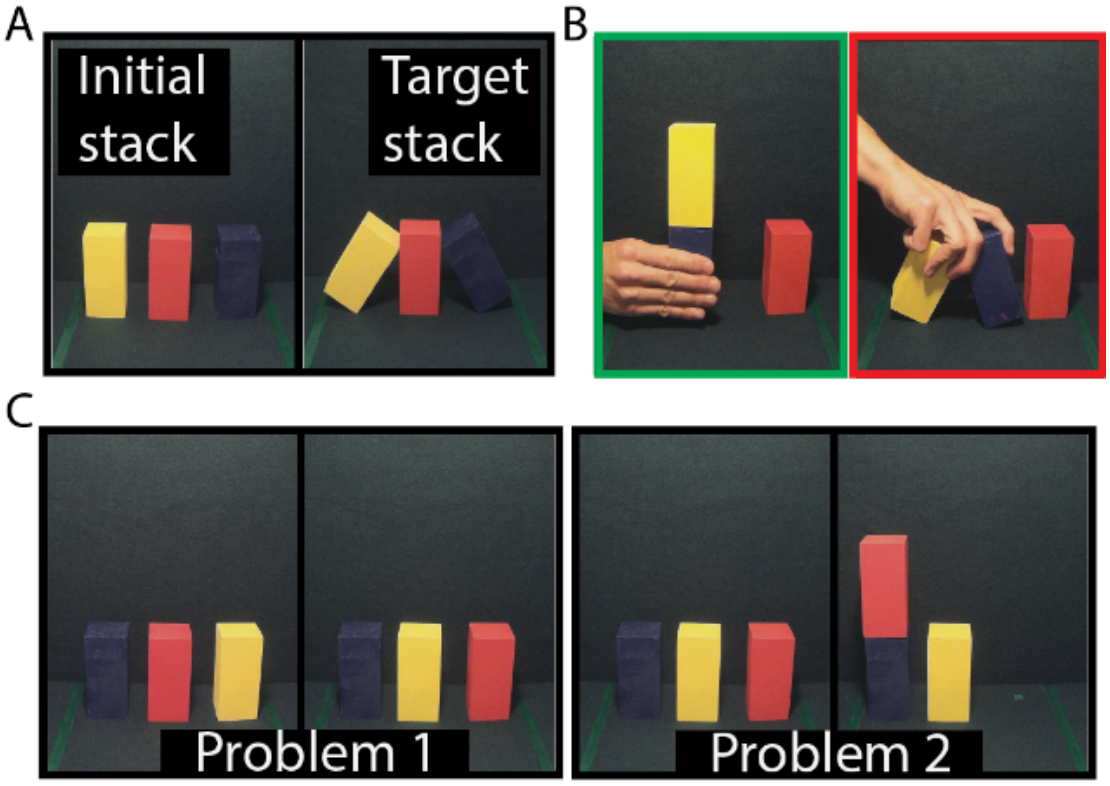
Relations to other areas

- Mixed-Integer Programming in Hybrid Control:
  $\rightarrow$ bridge to AI planning
- Differentiable Physics:
  $\rightarrow$ exploit differentiable modes; but introduce “logic of local optima”
- Dexterous Robot Manipulation:
  $\rightarrow$ represent manipulation modes to become AI-plannable
- Classical (sample-based) Task and Motion Planning:
  $\rightarrow$ optimization & physics
- Cognitive Science & Intuitive Physics
  $\rightarrow$ computational paradigm beyond MCMC

Human Experiments



Human Experiments



LGP as a model of human manipulation choice

Yildirim, Gerstenberg, Saeed, Toussaint, Tenenbaum: *Physical problem solving: Joint planning with symbolic, geometric, and dynamic constraints*. CogSci'17

What's next?

Planning $\rightarrow$ Execution

- So far, LGP only describes how to compute plans – execution of these plans is a different beast

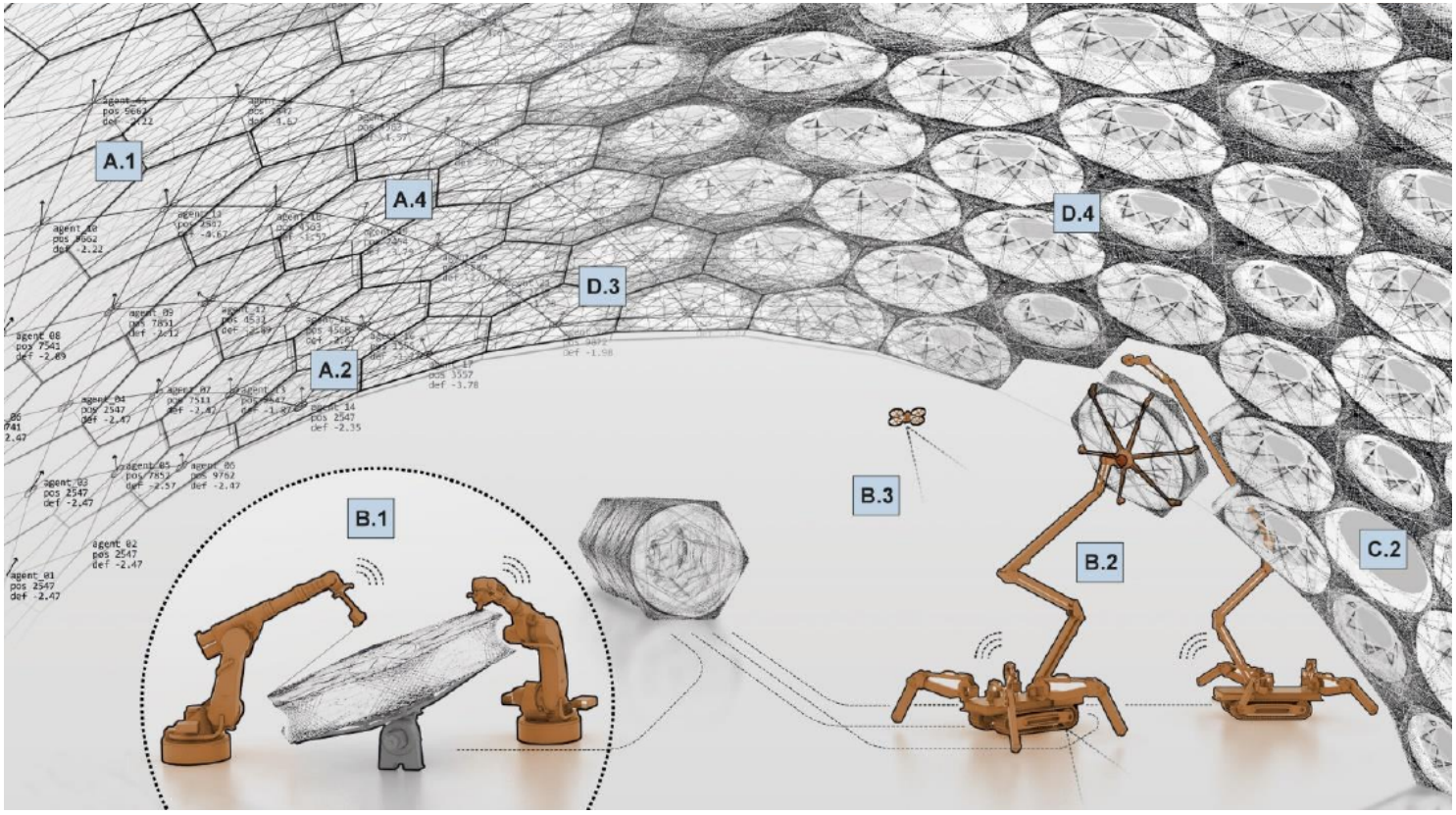
Planning $\rightarrow$ Execution

- A plan is only a *guess of what might be possible*
  (more rigorously, a *lower bound* of real-world execution)

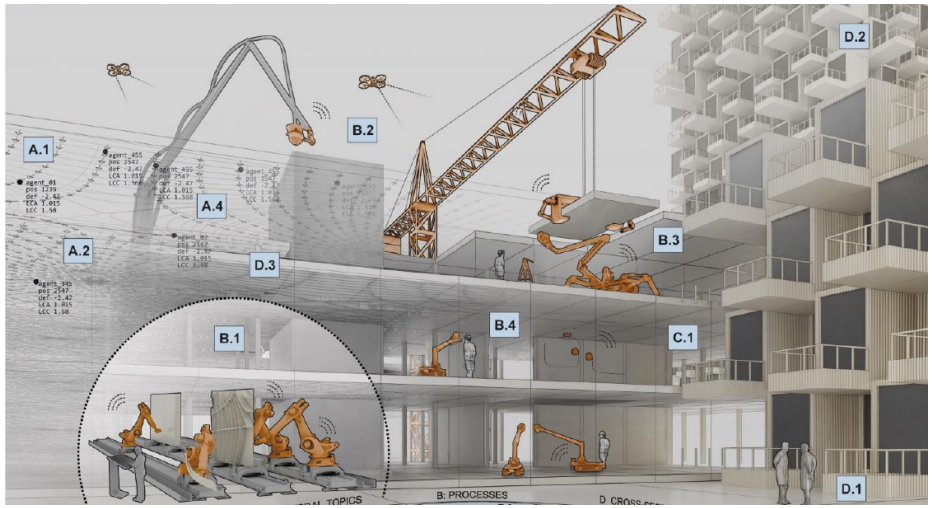
- Learn from failures:
  - We have a clear notion of failure; much more informative than reward
  - Sample-efficient RL to learn so choose, discard, and switch between plans

IntCDC

- Excellence Cluster in Integrated Computational Design and Construction



IntCDC



- Formalize the whole process (multi-robot construction, design, physics, etc) in a way so we can jointly reason over everything
  - Design so as to make it easier to construct
  - Design things that you didn't know could be constructed
  - Leverage simulations for large-scale exploration of designs