Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning

Marc Toussaint
Kelsey R. Allen
Kevin A. Smith
Joshua B. Tenenbaum

Their Contribution

They structured tool-use planning in domains that include physical interactions such as hitting and throwing to be modeled in Task and motion planning formulation.

They do this by formulating primitives in path optimization level (Can directly use motion planners to solve the task). This was done by imposing stable kinematics constraints and differential dynamics.

Betty the Crow

Differentiating a Simulator

- Efficient general inversion of simulator is implausible: discontinuities in effect space → local optima w.r.t. the global inverse problem.

- They introduce explicit predicates and logical rules that flexibly describe possible sequences of modes. All predicates are grounded as smooth and differentiable constraints on the system dynamics.

- They introduces these constraints as mathematical functions. Nonlinear solver solves the problem.

Problem Formulation

Cost of path. eg. Sum of joint accelerations.

Goal conditions. Eg. touch object with other object.

Physic Rules, eg. Kinematics

$$
\begin{aligned}
\min_{x} \quad & \int_0^T f_{\text{path}}(\bar{x}(t))\, dt + f_{\text{goal}}(x(T)) \\
\text{s.t.} \quad & x(0) = x_0,\ h_{\text{goal}}(x(T)) = 0,\ g_{\text{goal}}(x(T)) \le 0\,, \\
& \forall t \in [0, T]:\ h_{\text{path}}(\bar{x}(t)) = 0,\ g_{\text{path}}(\bar{x}(t)) \le 0\,. \quad (1)
\end{aligned}
$$

They ground all of these in mathematical equality constraints.

Mixed Logic Program

$a_{1:K}$: Action Skeleton

Every skeleton defines a problem($P(a_{1:K})$) to solve.

Logic(STRIPS) gives feasible skeletons

Since constraints on problems are smooth, each problem is ‘tractable’.

$$
\begin{aligned}
\min_{x, a_{1:K}, s_{1:K}} \quad & \int_0^T f_{\text{path}}(\bar{x}(t))\, dt + f_{\text{goal}}(x(T)) \\
\text{s.t.} \quad & x(0) = x_0,\ h_{\text{goal}}(x(T)) = 0,\ g_{\text{goal}}(x(T)) \le 0, \\
& \forall t \in [0, T]:\ h_{\text{path}}(\bar{x}(t), s_{k(t)}) = 0, \\
& \qquad\qquad\qquad g_{\text{path}}(\bar{x}(t), s_{k(t)}) \le 0 \\
& \forall k \in \{1, .., K\}:\ \boxed{\begin{aligned} & h_{\text{switch}}(\hat{x}(t_k), a_k) = 0, \\ & g_{\text{switch}}(\hat{x}(t_k), a_k) \le 0, \\ & s_k \in \text{succ}(s_{k\text{-}1}, a_k)\ . \end{aligned}} \qquad (2)
\end{aligned}
$$

Introduces smooth mode changes(As far as I understood.)

Optimization:

Logic makes problem tractable:

Mixed integer Program: By using integers, introduce if-else branches in mathematical problems.

LGP: Use logic instead of integers, allow adding mode changes in path optimization problem.

Notion of Modes

- Contact mode: Modeling interactions
- Kinematic mode: Modeling robots motions.
- Stable mode: Similar to Kinematic mode, used while manipulating objects. Stable refers to motion of object: transformation between robots-end effector, and object is constant(like adding a kinematic link between robot and object).

Constraints

TABLE I
PREDICATES TO IMPOSE CONSTRAINTS ON THE PATH OPTIMIZATION.

(touch X Y)	distance between X and Y equal 0
[impulse X Y]	ImpulseExchange eq & skip smoothness constraints on X Y
(staFree X Y)	create stable (constrained to zero velocity) free (7D) joint from X to Y
(staOn X Y)	create stable 3D $xy\phi$ joint from X to Y
(dynFree X)	create dynamic (constrained to gravitational inertial motion) free joint from world to X
(dynOn X Y)	create dynamic 3D $xy\phi$ joint from X to Y
(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)	(see text)

Actions

TABLE II
ACTION OPERATORS AND THE PATH CONSTRAINTS THEY IMPLY.

grasp(X Y)	[inside X Y] (staFree X Y)
handover(X Y Z)	[inside Z Y] (staFree Z Y)
place(X Y Z)	[above Y Z] (staOn Z Y)
throw(X Y)	(dynFree 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)

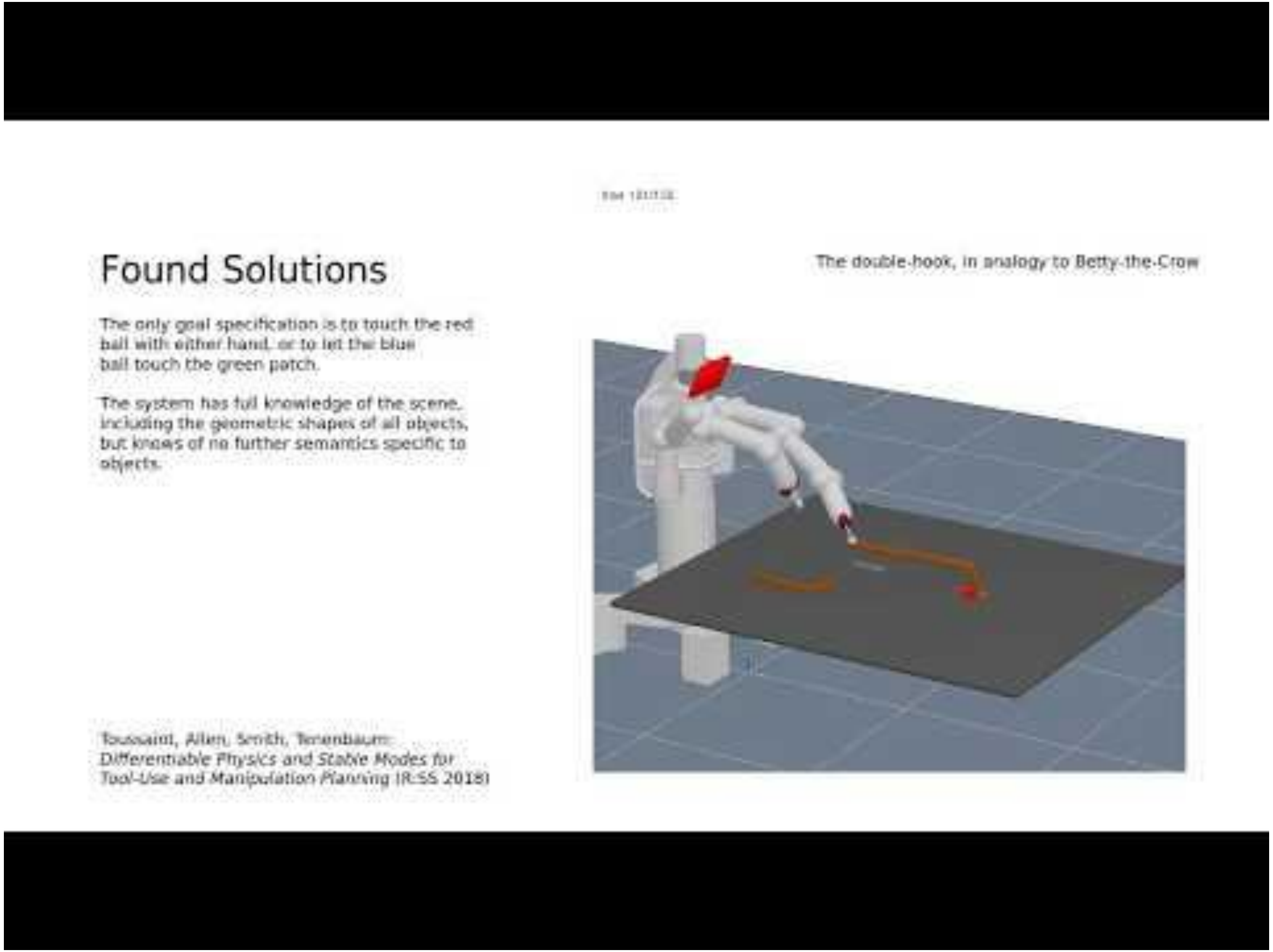
Solving The Problem

They use the existing Multi-Bound Tree Search(Marc Toussaint and Manuel Lopes, ICRA 2017) (MBTS) method to solve the resulting LGP problem.Logic described defines a decision tree.Every node in this decision tree is associated with a skeleton $a_{1:K}$ which defines the conditional path optimization problem $P(a_{1:K})$.

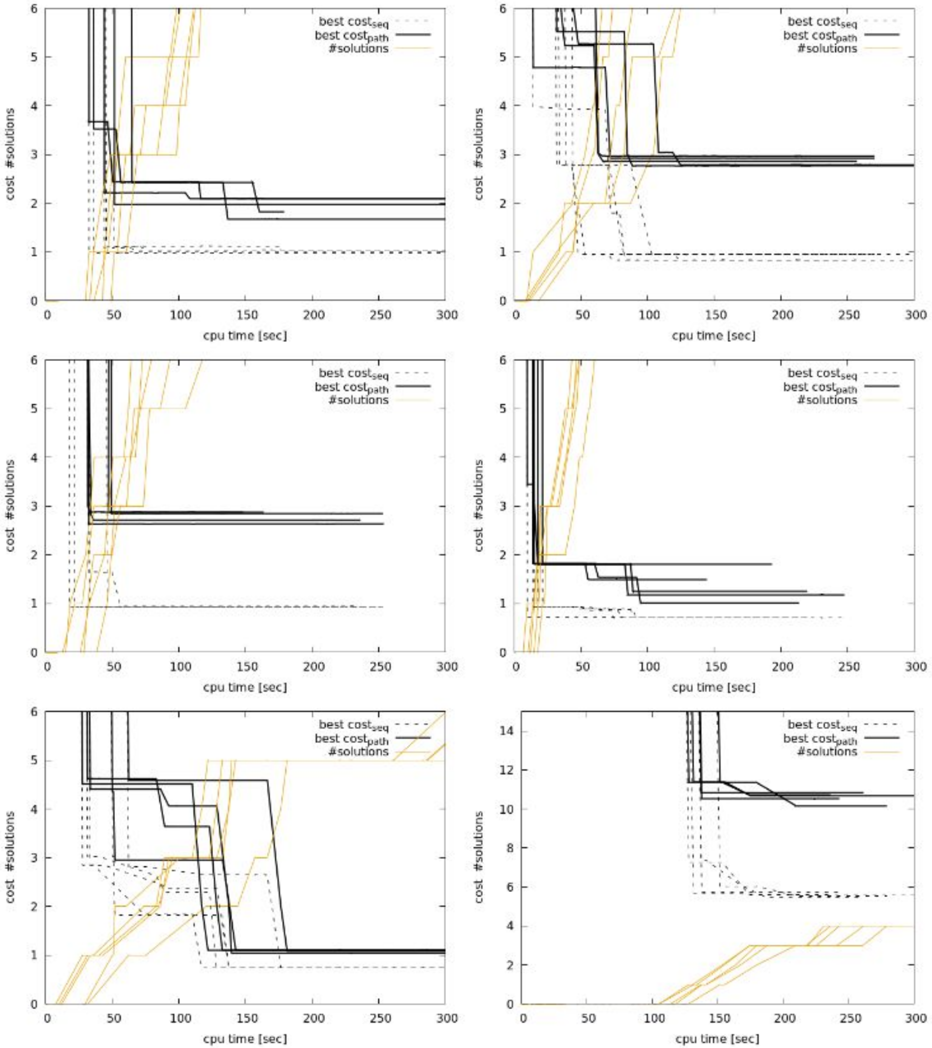
They create simplified nonlinear problems to find lower bounds of the cost associated with different action skeletons. They also create subproblems to optimize since if P is feasible P^ is feasible(P^ corresponds to subproblem.(From ICRA 2017 Paper)Eg. if you can't grasp a tool(P^), they you can not use that tool to pull some object (P)

- P1: Feasibility of only the last pose. Is IK solvable.(Not sure if for all actions or just last.)
- P2: For each action, they look at two time frames, one before one after. It shows whether is it feasible.
- P3: Discretized version of problem(Simpler)

Experiments



Results



best cost$_{seq}$
best cost$_{path}$
#solutions
cost #solutions
cpu time [sec]

Some References

https://www.ijcai.org/Proceedings/15/Papers/274.pdf : Original LGP formulation

https://ieeexplore.ieee.org/document/7989464 : ICRA 2017 for MBTS

https://brown.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=943259ba-f6be-44b1-93ec-a8c601422d3a : Nice Video that explains the stuff.