Models & Abstractions for Physical Reasoning

Marc Toussaint

Machine Learning & Robotics Lab – University of Stuttgart

NeurIPS 2018 workshop: Modelling the Physical World, Montreal, Dec 7, 2018

Physics is not differentiable

Sensitivity Analysis

- Ralph & Dempe. **Directional derivatives of the solution of a parametric nonlinear program. 1994**. Research Report.
- Fiacco & Kyparisis. **Sensitivity analysis in nonlinear programming** under second order assumptions. Lecture Notes in Control and Information Sciences, 74-97, **1985**.
- Kyparisis. Sensitivity analysis for nonlinear programs and variational inequalities with nonunique multipliers. Mathematics of Operations Research, 15:286298, 1990.
- Levy & Rockafellar. Sensitivity analysis of solutions to generalized equations. Trans. Amer. Math. Soc. 1993.
- Poliquin & Rockafellar. **Proto-derivative** formulas for basic **subgradient mappings** in mathematical programming. Set-valued Analysis, 2:275290, 1994.
- Levy & Rockafellar. Sensitivity of solutions in nonlinear programs with nonunique multiplier Recent Adv. in Nonsmooth Optimzation: 215-223, 1995

Sensitivity Analysis

– Ralph & Dempe. **Directional derivatives of the solution of a parametric nonlinear program. 1994**. Research Report.
– Fiacco & Kyparisis. **Sensitivity analysis in nonlinear programming** under second order assumptions. Lecture Notes in Control and Information Sciences, 74-97, **1985**.
– Kyparisis. Sensitivity analysis for nonlinear programs and variational inequalities with nonunique multipliers. Mathematics of Operations Research, 15:286298, 1990.
– Levy & Rockafellar. Sensitivity analysis of solutions to generalized equations. Trans. Amer. Math. Soc. 1993.
– Poliquin & Rockafellar. **Proto-derivative** formulas for basic **subgradient mappings** in mathematical programming. Set-valued Analysis, 2:275290, 1994.
– Levy & Rockafellar. Sensitivity of solutions in nonlinear programs with nonunique multiplier Recent Adv. in Nonsmooth Optimzation: 215-223, 1995

"We show under a standard constraint qualification, not requiring uniqueness of the multipliers, that the quasi-solution mapping is differentiable in a generalized sense, and we present a formula for its derivative."

Sensitivity Analysis

– Ralph & Dempe. **Directional derivatives of the solution of a parametric nonlinear program. 1994**. Research Report.
– Fiacco & Kyparisis. **Sensitivity analysis in nonlinear programming** under second order assumptions. Lecture Notes in Control and Information Sciences, 74-97, **1985**.
– Kyparisis. Sensitivity analysis for nonlinear programs and variational inequalities with nonunique multipliers. Mathematics of Operations Research, 15:286298, 1990.
– Levy & Rockafellar. Sensitivity analysis of solutions to generalized equations. Trans. Amer. Math. Soc. 1993.
– Poliquin & Rockafellar. **Proto-derivative** formulas for basic **subgradient mappings** in mathematical programming. Set-valued Analysis, 2:275290, 1994.
– Levy & Rockafellar. Sensitivity of solutions in nonlinear programs with nonunique multiplier Recent Adv. in Nonsmooth Optimzation: 215-223, 1995

"We show under a standard constraint qualification, not requiring uniqueness of the multipliers, that the quasi-solution mapping is differentiable in a generalized sense, and we present a formula for its derivative."

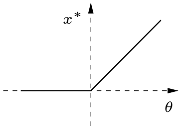
- Quasi-solution mapping: parameterized NLP $\mathcal{P}(\theta)$

$$S : \theta \mapsto \{x : \text{KKT hold for } \mathcal{P}(\theta)\}$$

Non-differentiable solution maps

- Basic example $(x, \theta \in \mathbb{R})$:

$$\min_x (x-\theta)^2 \quad \text{s.t.} \quad x \geq 0$$
$$S : \theta \mapsto x^* = \max\{0, x\}$$

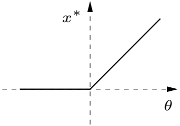


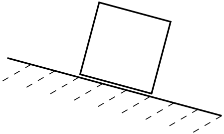
Non-differentiable solution maps

- Basic example $(x, \theta \in \mathbb{R})$:

$$\min_x (x-\theta)^2 \quad \text{s.t.} \quad x \geq 0$$

$$S : \theta \mapsto x^* = \max\{0, x\}$$





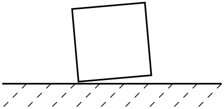
  – Discontinuous transition from stiction to sliding depending on θ

  – Bifurcation depending on contact or not

(discussion with Nima Fazeli, MIT)

Non-differentiable solution maps

- Jumping contact points:

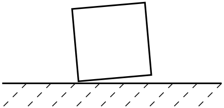


  - Chaotic system
  - Tiny change in initial condition (θ), huge change in outcome

(How often do we see only balls or capsules in demos?)

Non-differentiable solution maps

- Jumping contact points:



  - Chaotic system
  - Tiny change in initial condition (θ), huge change in outcome

  (How often do we see only balls or capsules in demos?)

- “We show *under a standard constraint qualification*,...”
  - Regularity conditions of constraints in vicinity of x^*
  - Typical technique for convergence proofs of NLP solvers

Gradients don't solve everything

Gradients don't solve everything

- Assume we have a local gradient

Gradients don't solve everything

- Assume we have a local gradient

- Case 1: We used strict constraints and complementarity formulations (gradients only hold within mode)
  $\rightarrow$ Zero gradient for every object the robot is not interacting with in the initialization
  $\rightarrow$ Gradient is completely useless to help deciding about interactions

Gradients don't solve everything

- Assume we have a local gradient

- Case 1: We used strict constraints and complementarity formulations (gradients only hold within mode)
  $\to$ Zero gradient for every object the robot is not interacting with in the initialization
  $\to$ Gradient is completely useless to help deciding about interactions

- Case 2: We're smoothing/relaxing interactions (Todorov)
  $\to$ combinatorics of local optima
  $\to$ Gradient doesn't help deciding about longer sequential interactions

Gradients don't solve everything

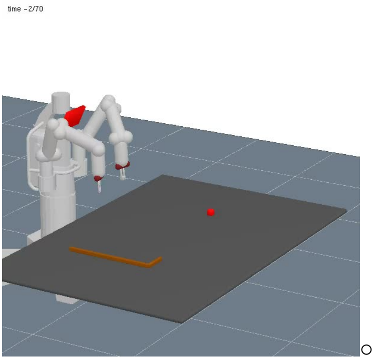
- Assume we have a local gradient

- Case 1: We used strict constraints and complementarity formulations (gradients only hold within mode)
  $\rightarrow$ Zero gradient for every object the robot is not interacting with in the initialization
  $\rightarrow$ Gradient is completely useless to help deciding about interactions

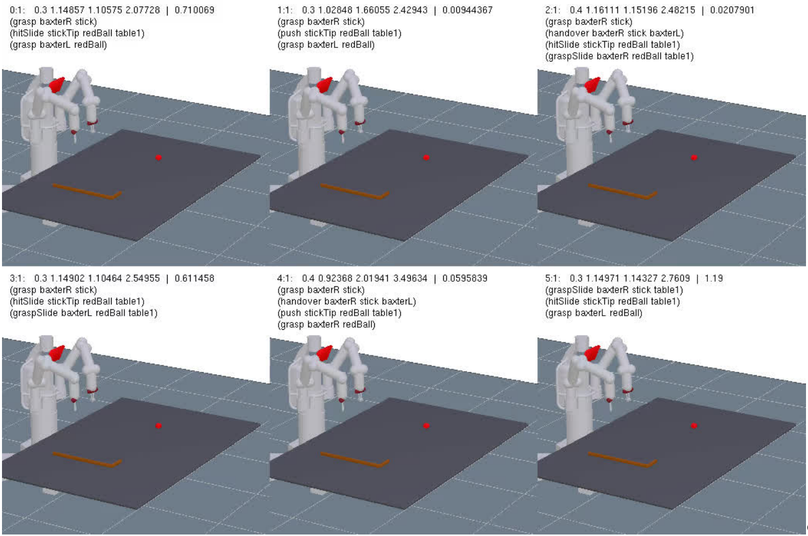
- Case 2: We're smoothing/relaxing interactions (Todorov)
  $\rightarrow$ combinatorics of local optima
  $\rightarrow$ Gradient doesn't help deciding about longer sequential interactions

- We knew that decisions about sequential interactions are NP hard
  Relaxation might sometimes help, but not really to solve inherently NP hard problems

Combining differentiable modes with logic



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


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)

- Logic-Geometric Program formulation:

control costs

goal

sequence of modes

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

mode transitions

logic of mode transitions

- Multi-Bound Tree Search as basic solver:
  - Every node in the LGP tree defines a skeleton (sequence of modes)
  - For every skeleton we have a hierarchy of NLPs $\mathcal{P}_1, .., \mathcal{P}_L$, which represent bounds of the full path problem
  - Do some kind of branch-and-bound

This talks: focus on discussion of mode models

- In LGP, we do not have to describe everything using high-fidelity physics models
- Different mode models: different simplifications/abstractions of physical interactions

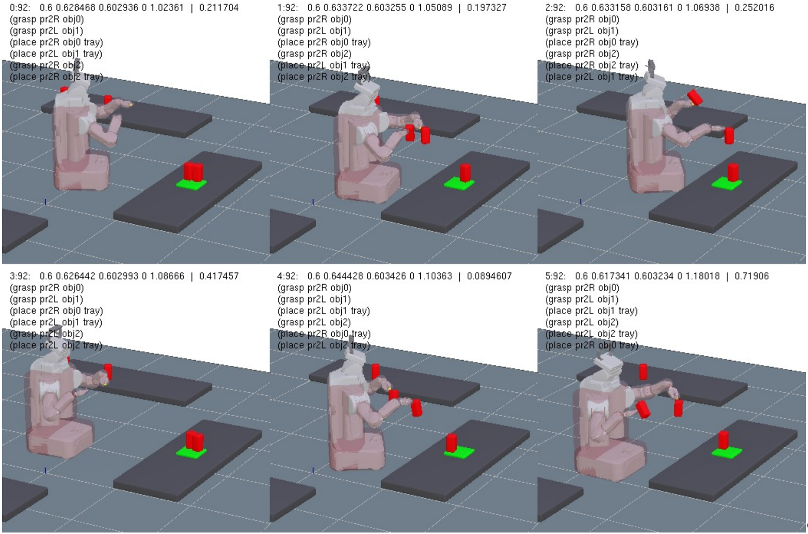
- Questions:
  Which mode models are sufficient to solve which tasks?
  Can the system make its own decisions on which abstractions to use to solve a task?

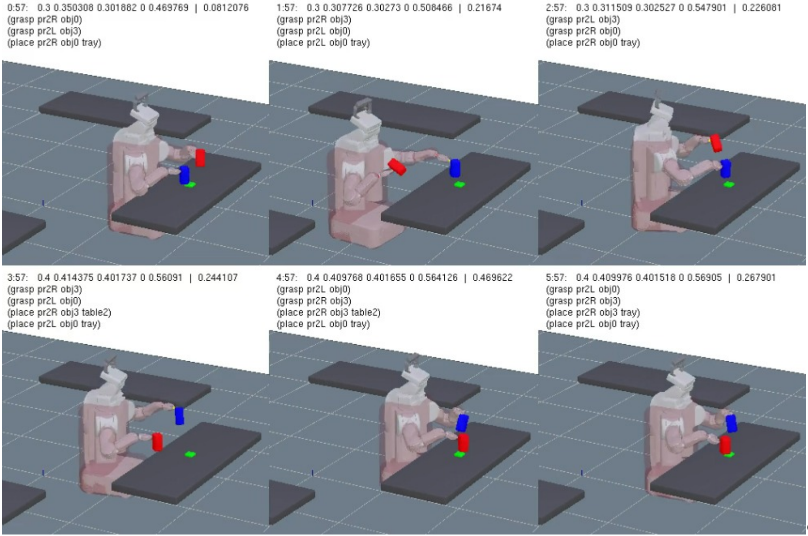
Stable Modes, Free Dynamics, Impulse Exchange

- Direct constraints on the path;
  - No additional decision variables
  - No representation of forces!
- (stable X Y) $\leftrightarrow$ relative pose of Y to X has zero velocity
  (dynamic X) $\leftrightarrow$ Newton-Euler acceleration law on object, so far without any force inputs: $\dot v = g, \dot\omega = 0$
  [impulse X Y] $\leftrightarrow$ direct constraint change in velocities: $R = m_1 \Delta v_1$:

$$\begin{aligned} &m_1 \Delta v_1 + m_2 \Delta v_2 = 0 && I_1 \Delta\omega_1 - p_1 \times R = 0 \\ &(I - cc^\top) R = 0 && I_2 \Delta\omega_2 + p_2 \times R = 0 \end{aligned}$$

- Straight-forward to make differentiable


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 0.301862 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)

Force-based interaction models

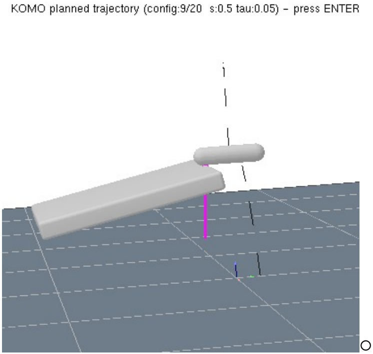
- (interact X Y) $\rightarrow$ introduce force decision variable $f \in \mathbb{R}^3$ into NLP
  - These directly enter the Newton-Euler equations for X and Y
- Possible constraints on f:

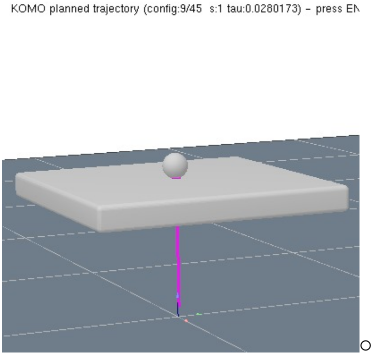
$-n^\top f \leq 0$	only pushing forces
$(\mathbf{I} - nn^\top)f = 0$	no tangential force – no friction
$df = 0$	force complementarity with distance
$f \approx 0$	force is small (small regularization)
$f - f' \approx 0$	force changes continuously (small regularization)
$V = 0$	no relative velocity (stiction & inelastic)
$n^\top V = 0$	no normal velocity (inelastic)
$n^\top V = -\beta n^\top V'$	normal vel is reflection of old vel (elasticity β)
$(\mathbf{I} - nn^\top)V = \alpha(\mathbf{I} - nn^\top)V'$	tangential velocity decreases exponentially

$V = [v_1 + w_1 \times (c - p_1)] - [v_2 + w_2 \times (c - p_2)]$, with c the contact point

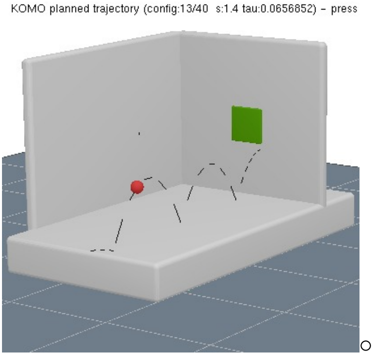
- Complementarity formulation similar to MPCC (Posa, Cantu & Tedrake), but not for friction cone
- Quasi-static reasoning: impose zero acceleration in all NE eqs. But regularize path length (1st-order problem)

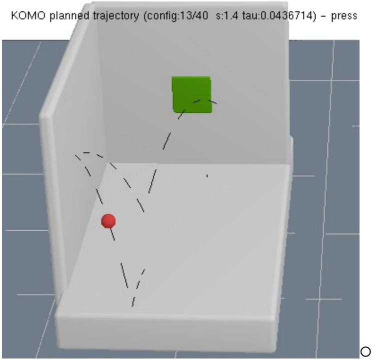
Fully passive scenarios (no goal, no control/actuation)



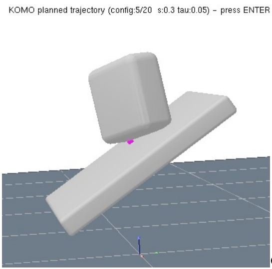


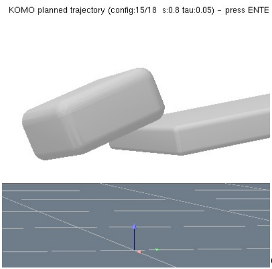
Same for goal-directed paths

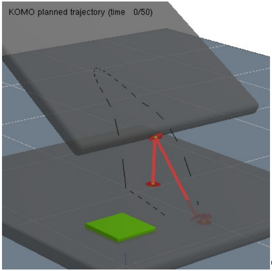




Friction & sliding







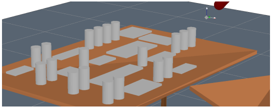
Challenges with this approach

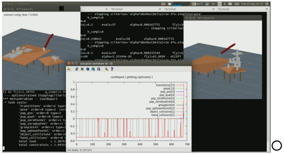
- Strengths:
  - Bridges between AI planning and physics, control, physical reasoning
  - Can integrate various levers of abstraction for reasoning
  - Important: Framework for formulating bounds to guide symbolic search

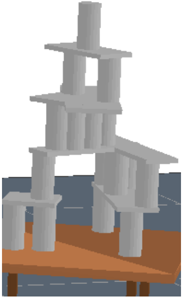
- Challenges:
  - Probabilistic Formulation, Stochastic Optimal Control, Execution, ...
  - Path optimization is tough for complex passive dynamics. Forward solving is just much easier than directly fitting a full path.
    (direct vs. indirect control)
  - Forward models integrate NLPs for each step to define forward dynamics—in contrast to having one big NLP over the path
  - But: The LGP framework can reason and optimize over future configurations before computing paths or forward shooting physics...

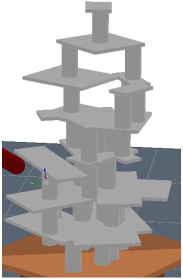
LGP & “Effective Kinematics”

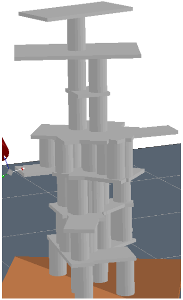
- “Effective kinematics” defines one of the bounds in tree search, that optimizes only over single frames; here over the final configuration

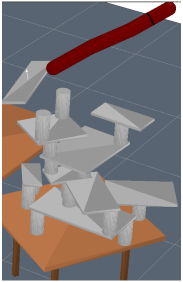












Toussaint: *Logic-geometric programming: An optimization-based approach to combined task and motion planning*. IJCAI’15

Discussion

- Physics is not differentiable
  Gradients don't solve everything
  $\rightarrow$ We need more than just differentiable models
  - Understand the structure of local optima and possible interaction sequences
  - Hybrid optimization, branch-and-bound, integrate logic

Discussion

- Physics is not differentiable
  Gradients don't solve everything
  $\rightarrow$ We need more than just differentiable models
  - Understand the structure of local optima and possible interaction sequences
  - Hybrid optimization, branch-and-bound, integrate logic

- "How do we choose among different paradigms for building and learning physical models?"
  $\rightarrow$ Exploit multiple levels/abstractions of physical interactions
  - stable, free, force interactions, quasi-static, complementarity

  $\rightarrow$ Go beyond mini-step forward models
  - Multi-scale, forward and backward, landmark-like

Discussion

- Standard forward simulators are not the only model of physics; and perhaps not the best for reasoning about long interaction sequences

- Scientific understanding of physical reasoning
  $\leftrightarrow$ More science on possible abstractions & models of physics

Thanks

- *for your attention!*

- special thanks to LIS & MIT:

  Leslie Pack Kaelbling, Tomas Lozano-Pérez, Russ Tedrake, Josh B Tenenbaum, Kelsey R Allen, Kevin A Smith, Ilker Yildirim, Nima Fazeli

  and to Toyota Research Institute

`https://github.com/MarcToussaint/rai-python`