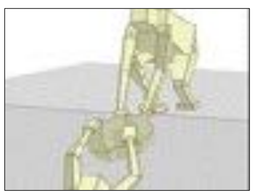
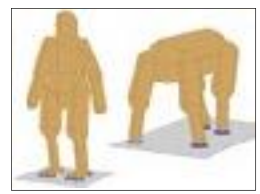
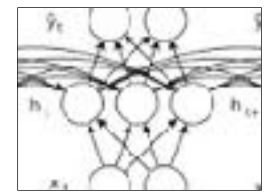
Optimization-Based Control:
Direct Collocation Methods for Trajectory and Policy Optimization





CS 287: Advanced Robotics, Fall 2019

Guest Lecture

Igor Mordatch

Overview

- **Previously:**
  - Locally optimal control (shooting vs. collocation)
  - Forward dynamics models and shooting (LQR, DDP)
- **Today:**
  - Direct collocation in detail (open-loop and policies)
  - *inverse* dynamics models
  - Solution methods for collocation problems
  - Optimization with contacts

Outline

- Trajectory optimization and direct collocation
- Inverse dynamics model
- Numerical optimization for collocation
- Optimizing dynamics with contact
- Collocation methods for policy learning

Trajectory Optimization

Trajectory Optimization

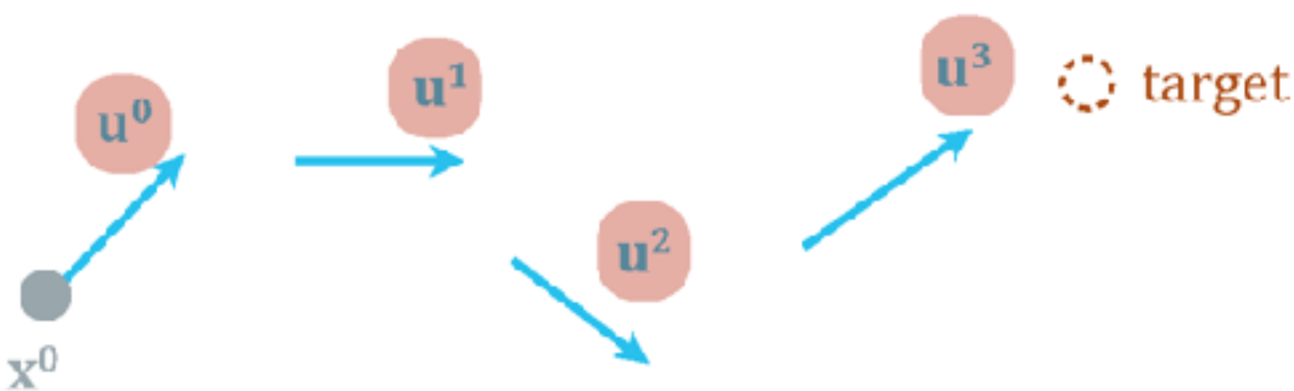
Forward Shooting:

$$\min_{\mathbf{u}^0 \dots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

Trajectory Optimization

Forward Shooting:

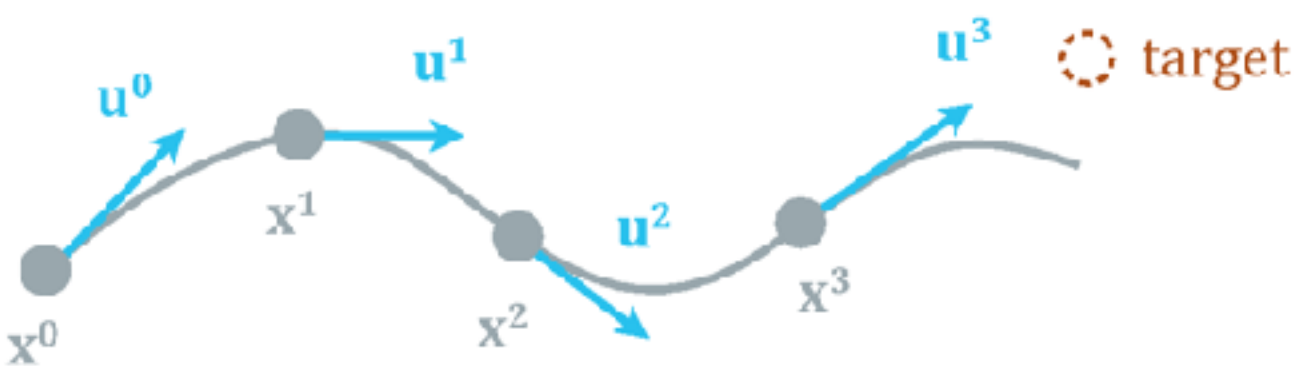
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Trajectory Optimization

Forward Shooting:

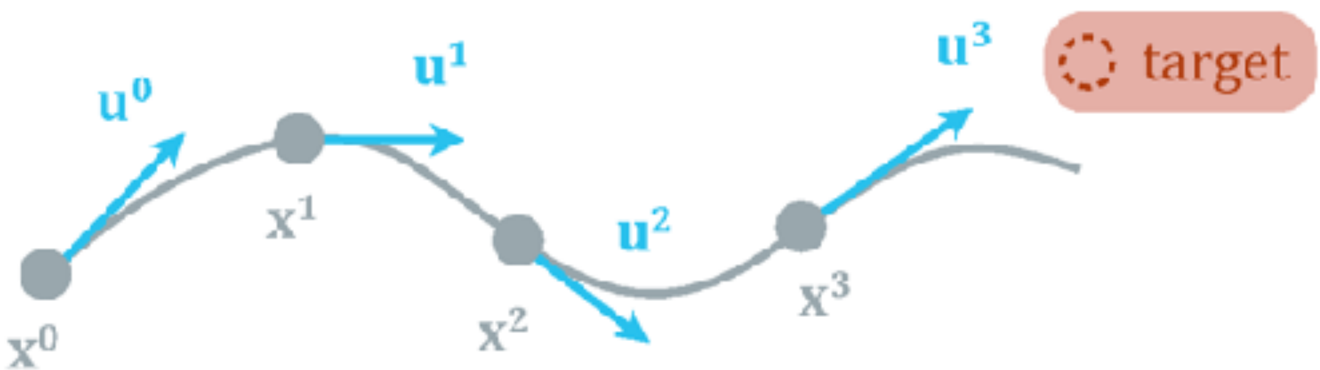
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Trajectory Optimization

Forward Shooting:

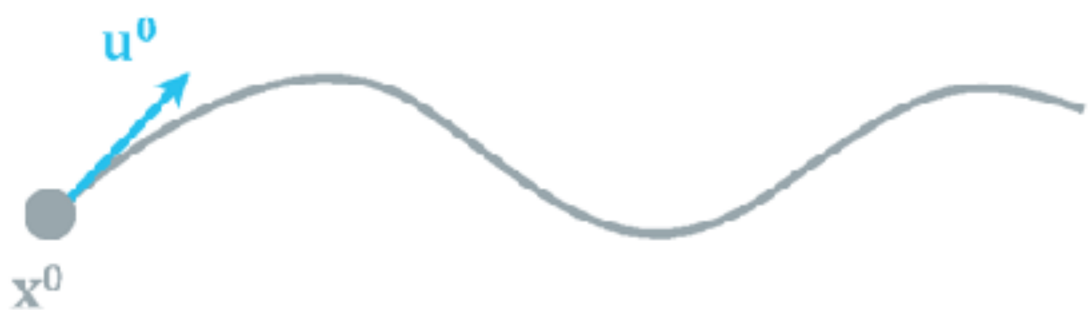
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Poor Conditioning

Forward Shooting:

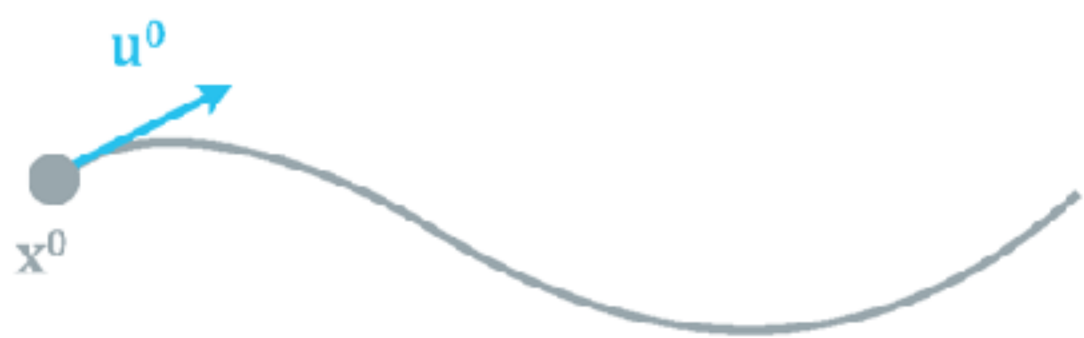
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Poor Conditioning

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Poor Conditioning

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

Poor Conditioning

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

$$\Updownarrow$$

$$\min_{\mathbf{u}_1, \ldots, \mathbf{u}_T} c(\mathbf{x}_1, \mathbf{u}_1) + c(f(\mathbf{x}_1, \mathbf{u}_1), \mathbf{u}_2) + \cdots$$

$$\cdots + c(f(f(\ldots)\ldots), \mathbf{u}_T)$$

Poor Conditioning

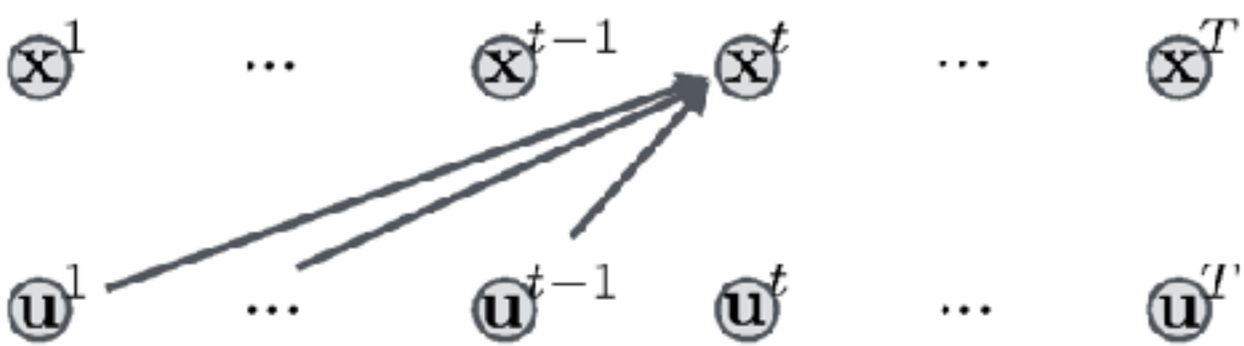
Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

$$\Updownarrow$$

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T} c(\mathbf{x}_1, \mathbf{u}_1) + c(f(\mathbf{x}_1, \mathbf{u}_1), \mathbf{u}_2) + \cdots$$

$$\cdots + c(f(f(\ldots)\ldots), \mathbf{u}_T)$$



Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

Narrow Feasible Region

Forward Shooting:

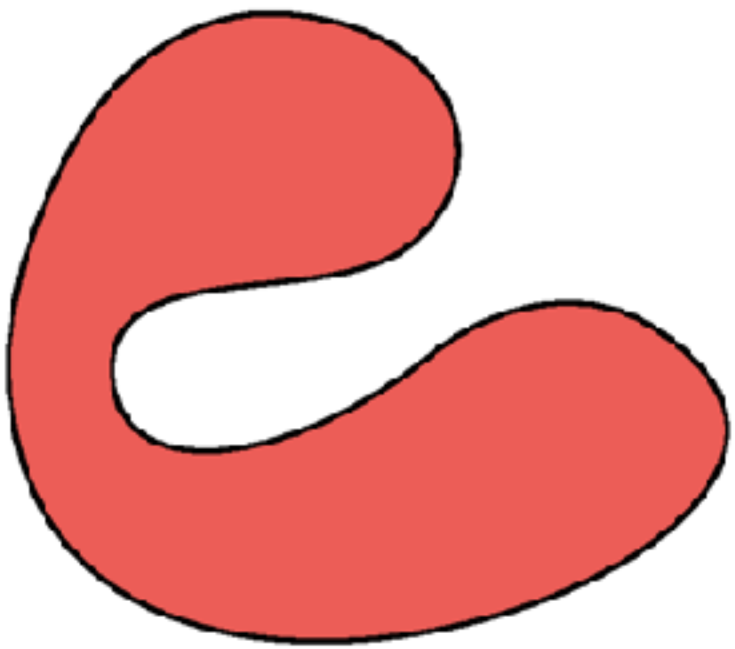
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

implicit hard constraint

Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

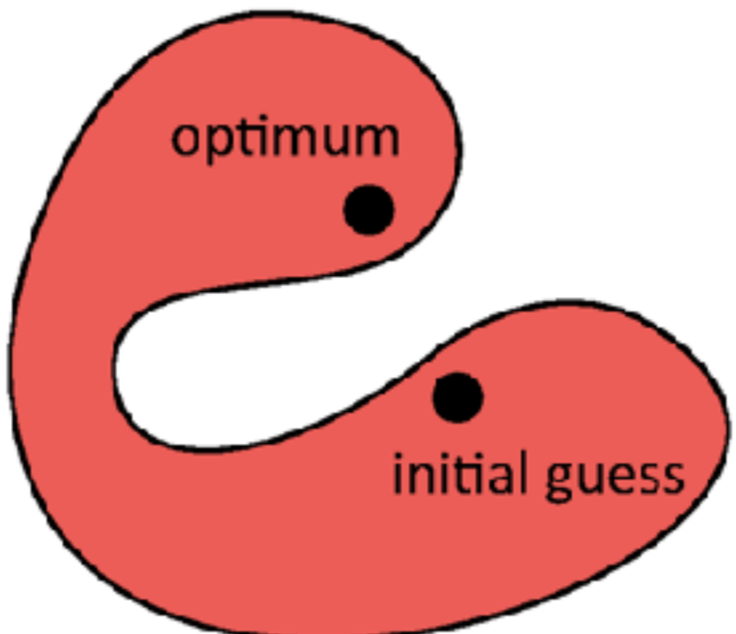


feasible region

Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

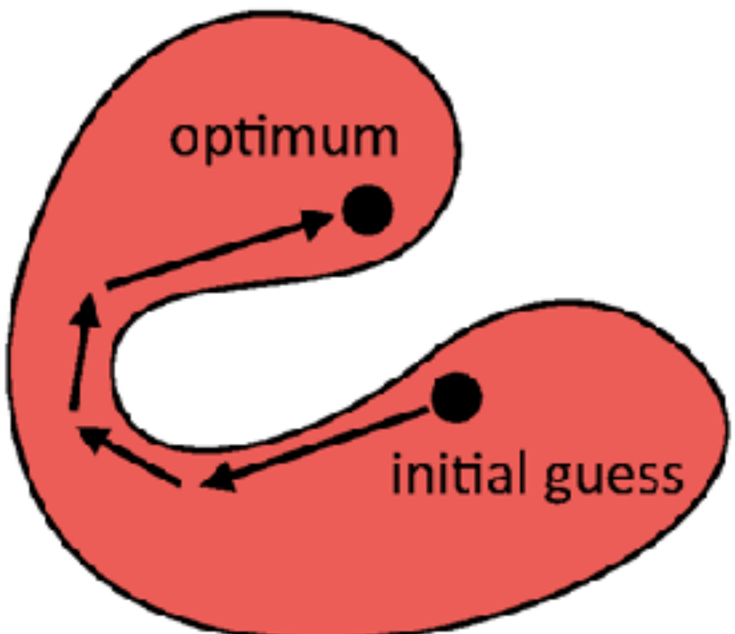


feasible region

Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

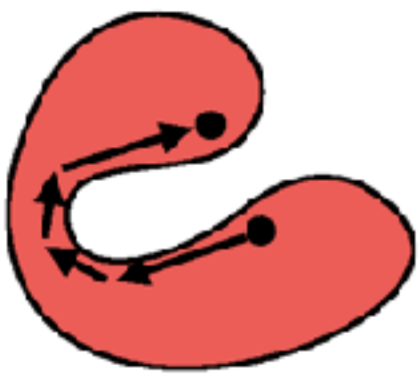


feasible region

Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



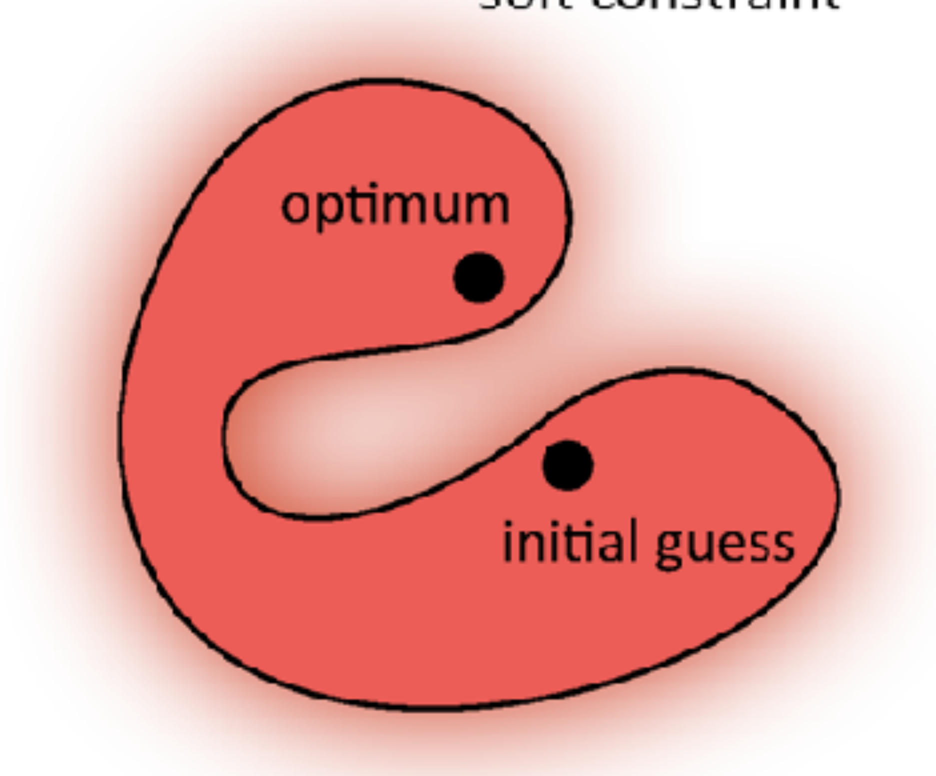
- Comes up as an issue in practice
  - collisions, falling down, etc…
- Prone to falling into local minima
- Makes solution sensitive to initial guess
- Initial guess from demonstrations and randomization helps

Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

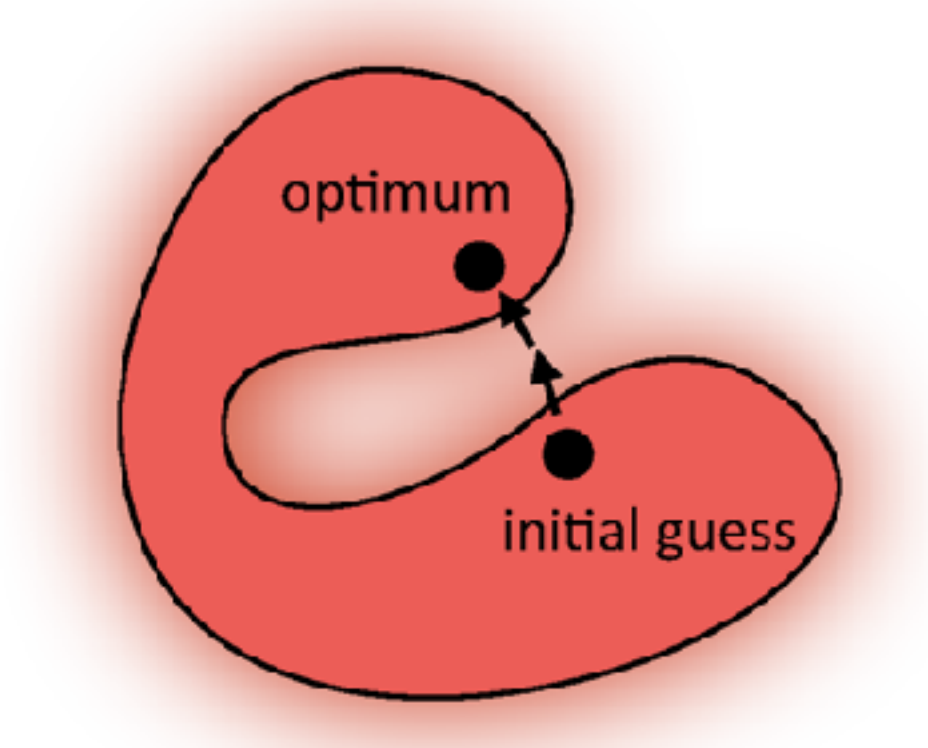
soft constraint



Narrow Feasible Region

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



From Last Lecture:

shooting	$\min_{u_0, u_1, \ldots, u_H} c(x_0, u_0) + c(f(x_0, u_0), u_1) + c(f(f(x_0, u_0), u_1), u_2) + \ldots$
collocation	$\min_{x_0, u_0, x_1, u_1, \ldots, x_H, u_H} \sum_{t=0}^{H} c(x_t, u_t)$ s.t. $x_{t+1} = f(x_t, u_t) \quad \forall t$

From Last Lecture:

shooting	$\min_{u_0, u_1, \ldots, u_H} c(x_0, u_0) + c(f(x_0, u_0), u_1) + c(f(f(x_0, u_0), u_1), u_2) + \ldots$
collocation	$\min_{x_0, u_0, x_1, u_1, \ldots, x_H, u_H} \sum_{t=0}^{H} c(x_t, u_t)$ s.t. $x_{t+1} = f(x_t, u_t) \quad \forall t$

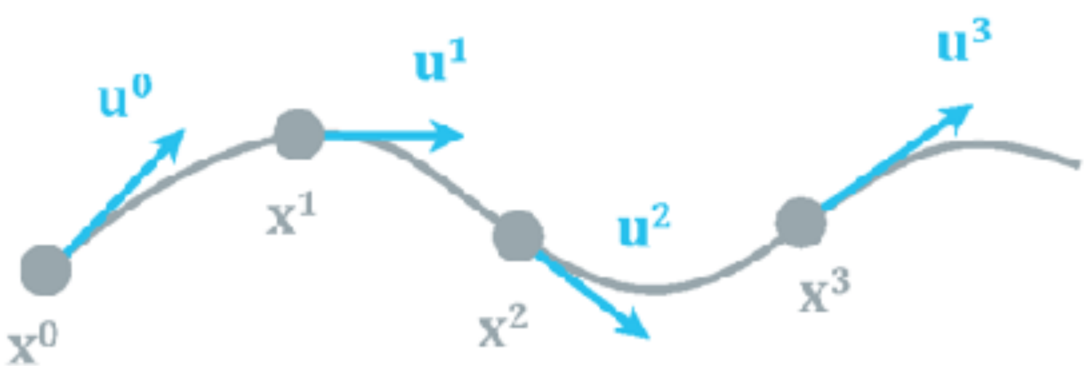
Direct Collocation

Forward Shooting:

$$\min_{\mathbf{u}^0 \dots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

Direct Collocation:

$$\min_{\mathbf{x}^0 \dots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \;\; st \;\; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$



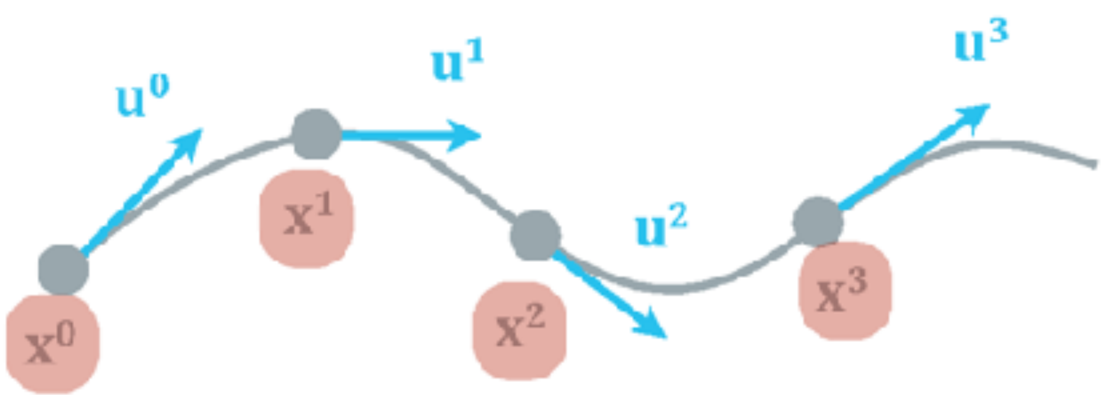
Direct Collocation

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

Direct Collocation:

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \ st \ f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$



From Last Lecture:

shooting	$\min_{u_0, u_1, \ldots, u_H} c(x_0, u_0) + c(f(x_0, u_0), u_1) + c(f(f(x_0, u_0), u_1), u_2) + \ldots$
collocation	$\min_{x_0, u_0, x_1, u_1, \ldots, x_H, u_H} \sum_{t=0}^{H} c(x_t, u_t)$ s.t. $x_{t+1} = f(x_t, u_t) \quad \forall t$

Direct Collocation

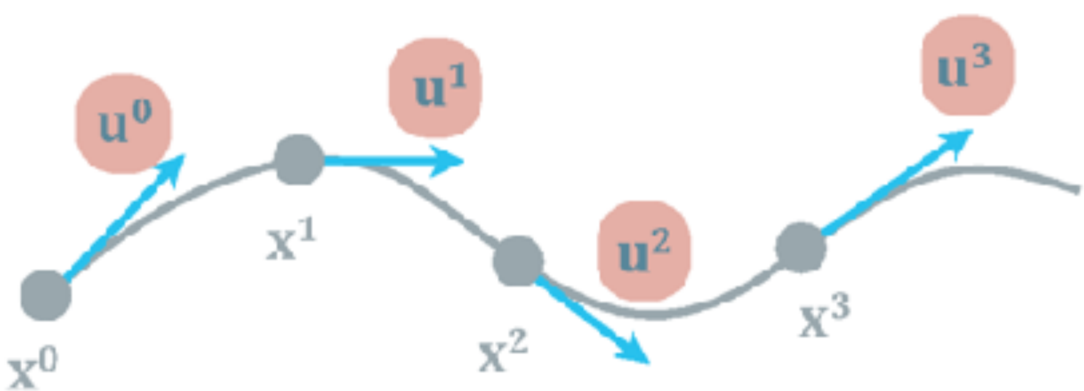
Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

inverse dynamics function

Direct Collocation:

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \;\; st \;\; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$



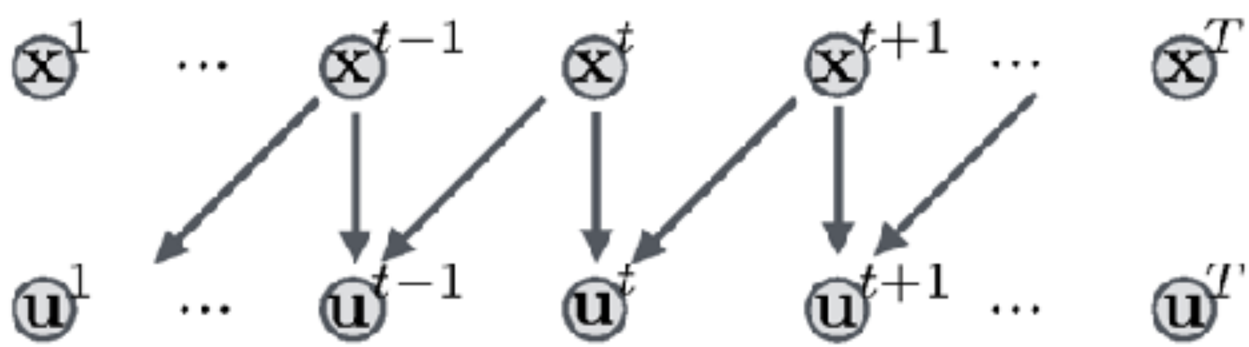
Direct Collocation

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

Direct Collocation:

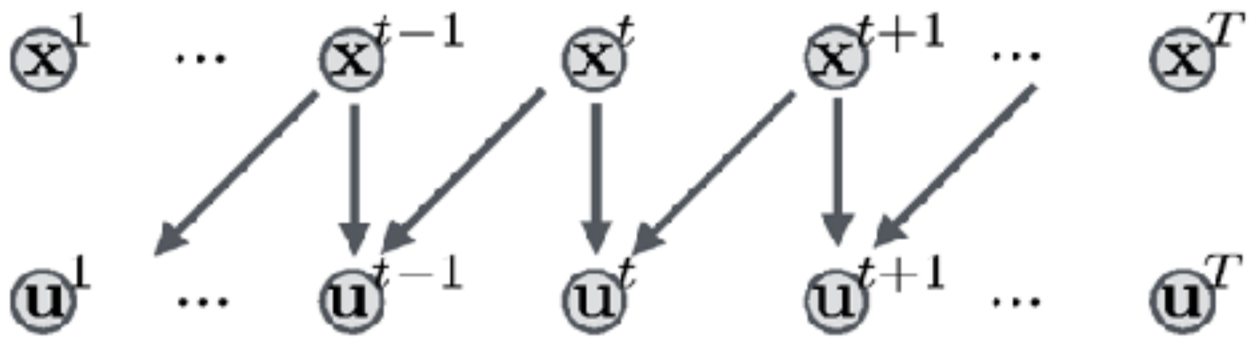
$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \ st \ f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$



Direct Collocation

Direct Collocation:

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \;\; st \;\; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$



- Only pairwise dependencies
- Good conditioning
  - changing x^1 has similar effect as changing x^T
- No forward integration instability

Direct Collocation

Direct Collocation:

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \; st \;\; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$

Explicit rather than implicit constraint

Direct Collocation

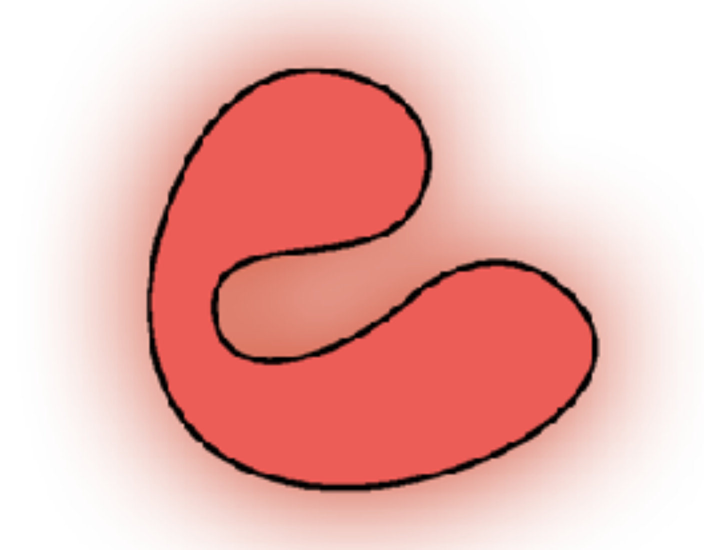
Direct Collocation:

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \; st \; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$

Explicit rather than implicit constraint

Can be hard or soft

Less prone to local minima



Shooting vs Direct Collocation

Forward Shooting:

$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$

- Optimize over controls
- State trajectory is implicit
- Dynamics is an implicit constraint (always satisfied)

Direct Collocation:

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \; st \; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$

- Optimize over states
- Controls and forces are implicit
- Dynamics is an explicit constraint (can be soft)

Outline

- Trajectory optimization and direct collocation
- Inverse dynamics model
- Numerical optimization for collocation
- Optimizing dynamics with contact
- Collocation methods for policy learning

Inverse Dynamics Model

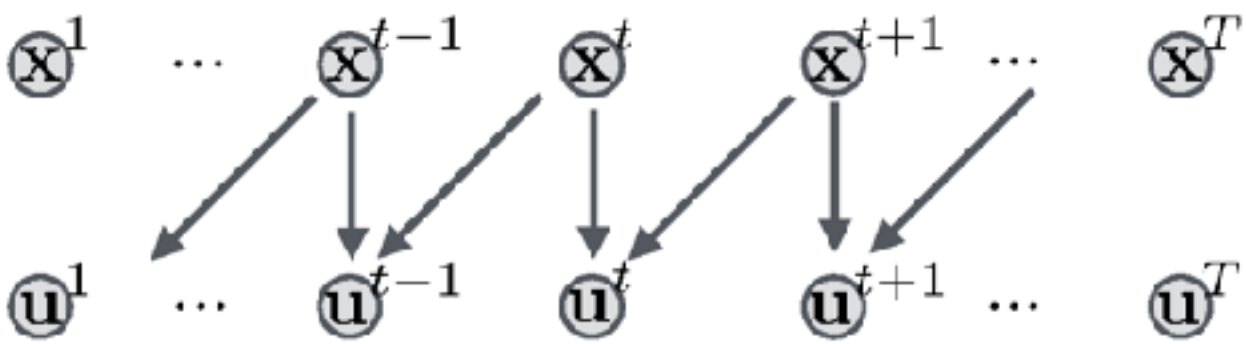
$$f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t$$

- Describes what controls and forces you apply when transitioning from x^t to x^{t+1}

Inverse Dynamics Model

$$f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t$$

- Describes what controls and forces you apply when transitioning from x^t to x^{t+1}
- Can be learned from data



Training data input: $\mathbf{x}^t$ $\mathbf{x}^{t+1}$

Target output: $\mathbf{u}^t$

Inverse Dynamics Model

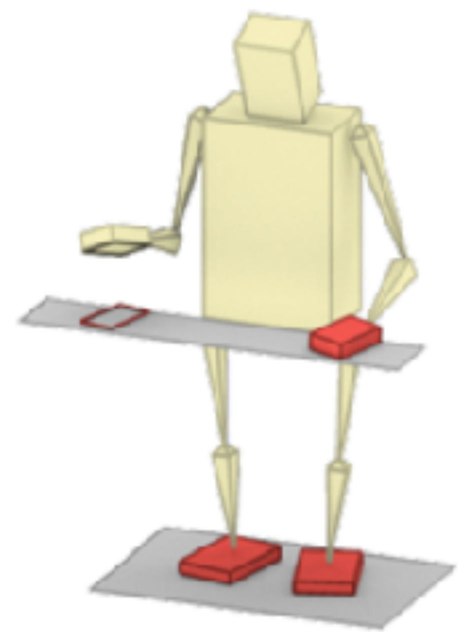
$$f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t$$

- Describes what controls and forces you apply when transitioning from x^t to x^{t+1}
- Can be learned from data
- For rigid multi-body dynamics, we can do better when we know system parameters

Rigid Multi-Body Dynamics

Generalized coordinates:

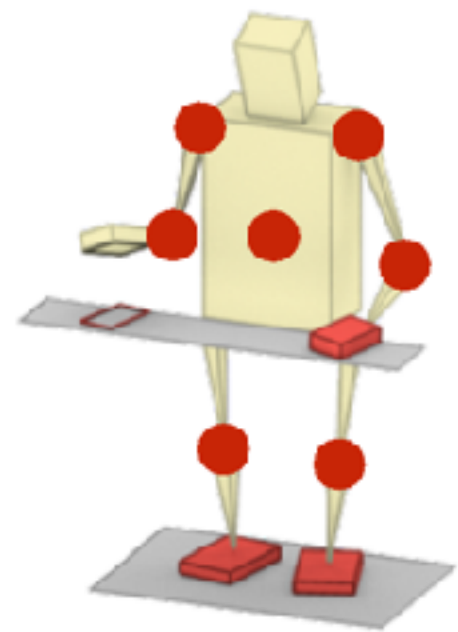
$$\mathbf{x}^t = \mathbf{q}^t$$



Rigid Multi-Body Dynamics

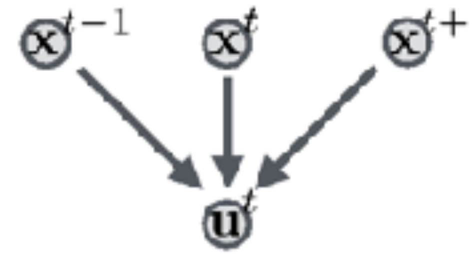
Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t$$



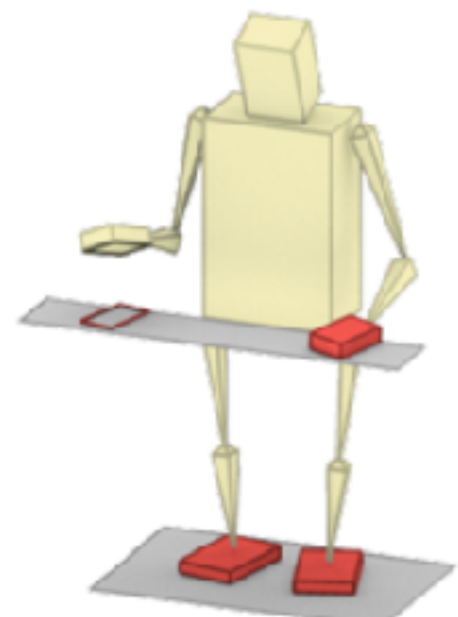
Rigid Multi-Body Dynamics

Generalized coordinates:



$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathrm{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathrm{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$

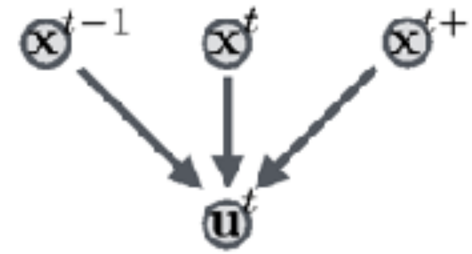
Calculate velocities and accelerations from nearby states



Rigid Multi-Body Dynamics

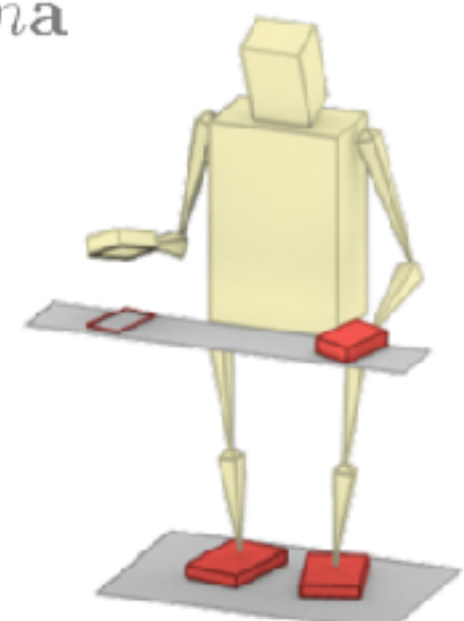
Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$



Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

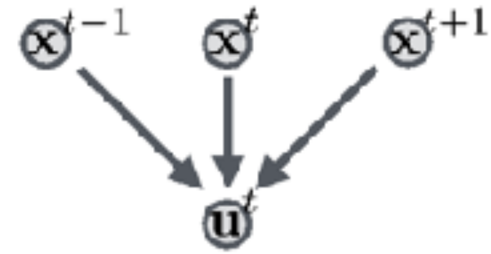
$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$



Rigid Multi-Body Dynamics

Generalized coordinates:

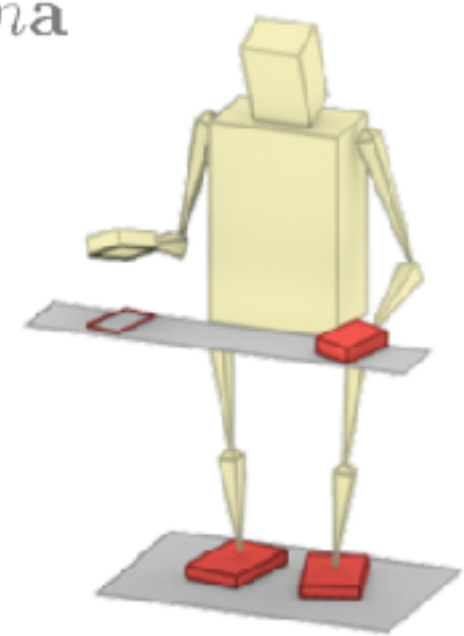
$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$



Dynamics equation: generalization of $\mathrm{f} = m\mathbf{a}$

$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

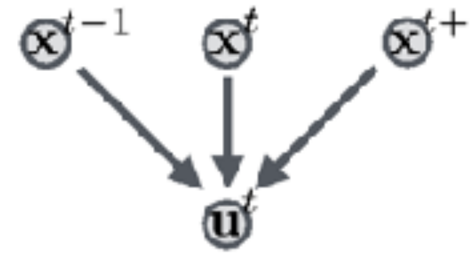
Generalized mass and Coriolis matrices



Rigid Multi-Body Dynamics

Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$

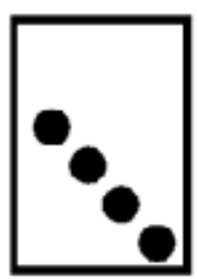


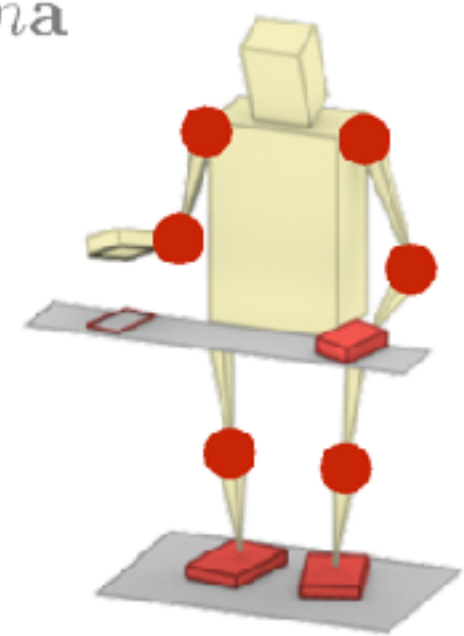
Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

Controls and actuation matrix

B

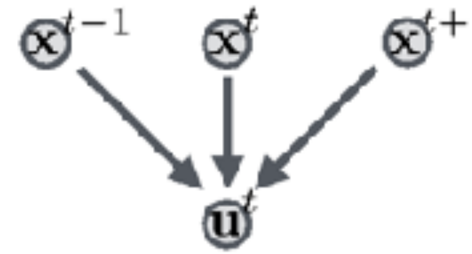




Rigid Multi-Body Dynamics

Generalized coordinates:

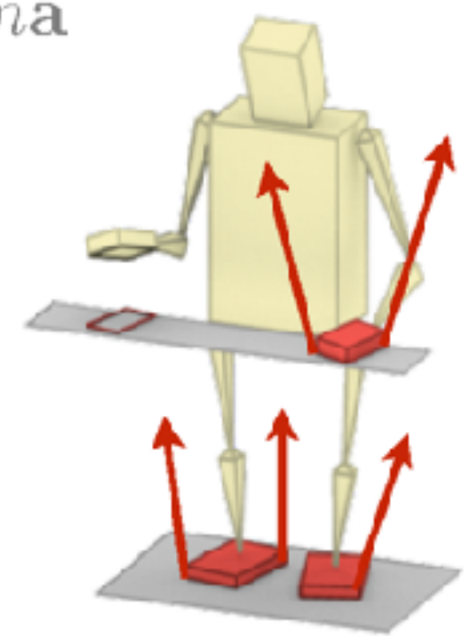
$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$



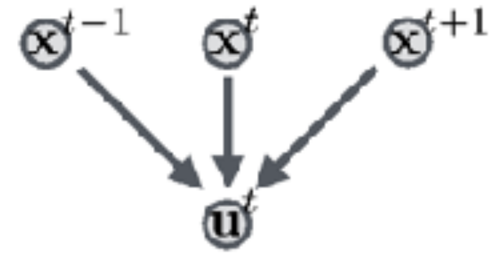
Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

Constraint forces and constraint Jacobian



Rigid Multi-Body Dynamics

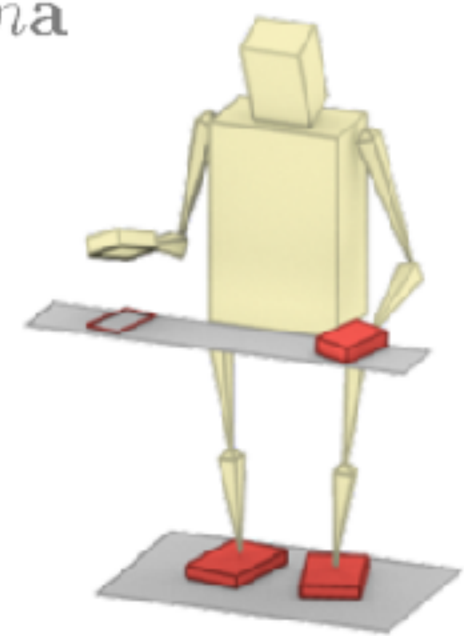


Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$

Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$



For more detail, see chapters 2 and 3 in

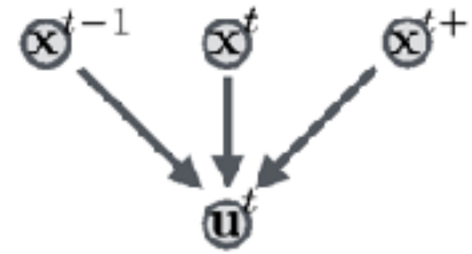
Springer Handbook of Robotics and

Analytical Dynamics: A New Approach

Rigid Multi-Body Dynamics

Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$

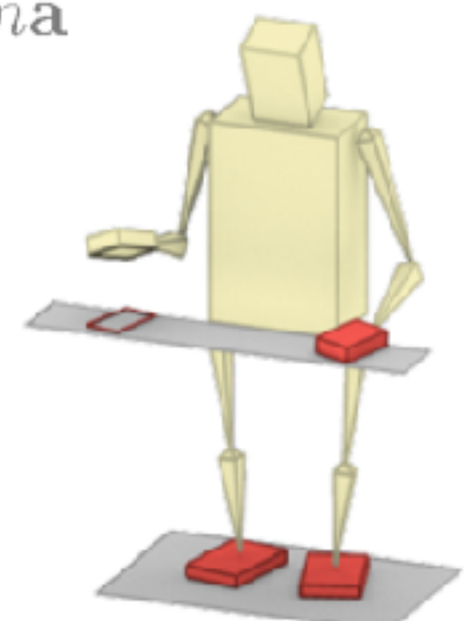


Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

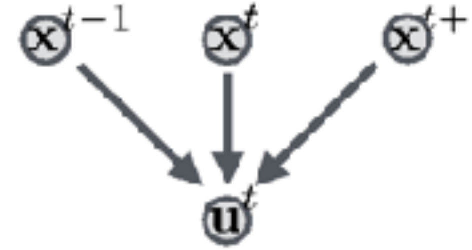
$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$

Inverse dynamics function:

$$f^{-1}(\mathbf{x}^{t-1}, \mathbf{x}^t, \mathbf{x}^{t+1}) = \arg\min_{\mathbf{u}\,\mathbf{f}} ||\ \cdot\ ||^2$$



Rigid Multi-Body Dynamics

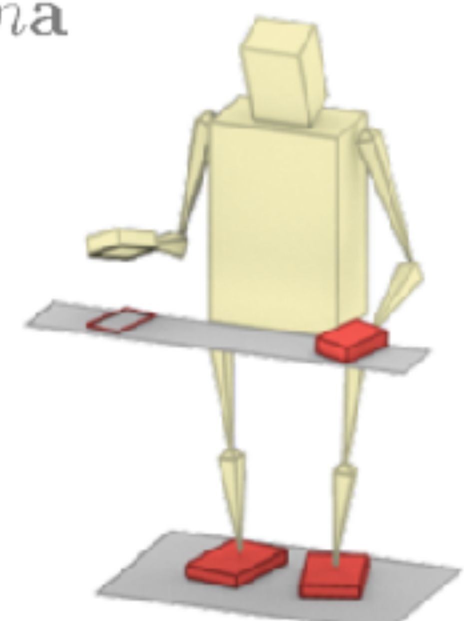


Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$

Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$

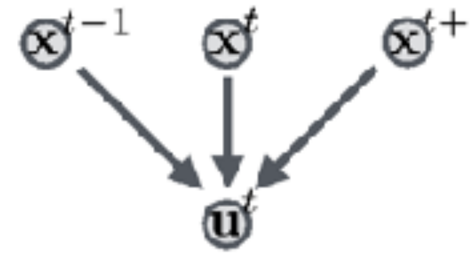


Inverse dynamics function:

$$f^{-1}(\mathbf{x}^{t-1}, \mathbf{x}^t, \mathbf{x}^{t+1}) = \arg\min_{\mathbf{u}\,\mathbf{f}} ||\;\downarrow\;||^2$$

can be solved numerically, or analytically [Todorov 14]

Rigid Multi-Body Dynamics

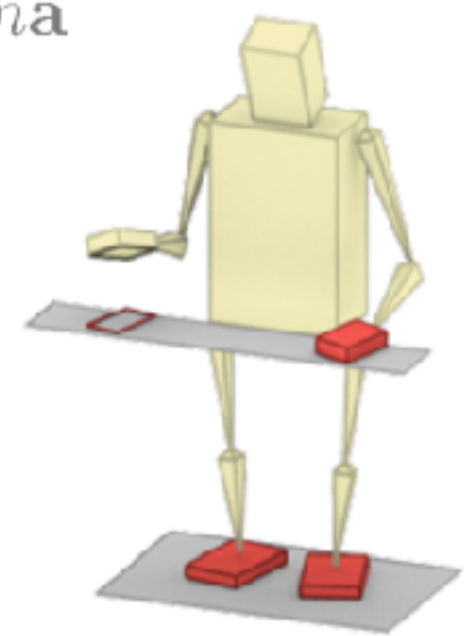


Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$

Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$



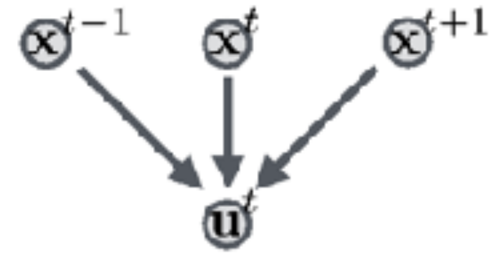
Inverse dynamics residual:

$$r(\mathbf{x}^{t-1}, \mathbf{x}^t, \mathbf{x}^{t+1}) = \min_{\mathbf{u}\,\mathbf{f}} \,||\ \downarrow\ ||^2$$

Rigid Multi-Body Dynamics

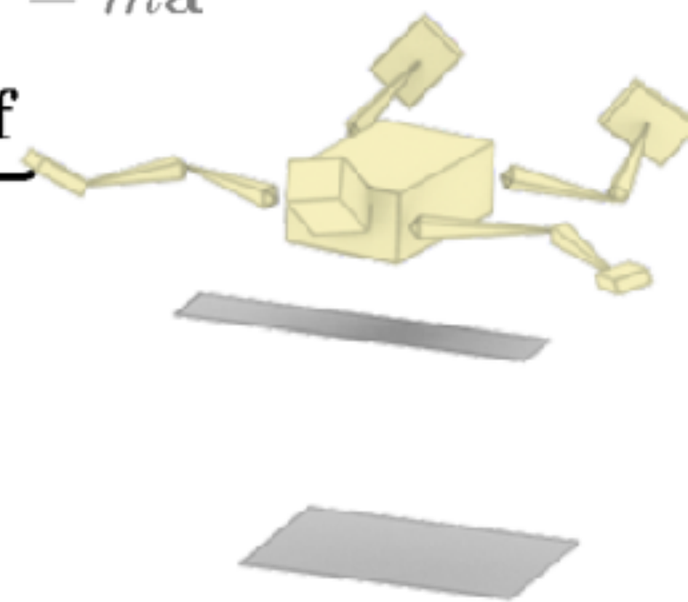
Generalized coordinates:

$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$



Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$



Inverse dynamics residual:

$$r(\mathbf{x}^{t-1}, \mathbf{x}^t, \mathbf{x}^{t+1}) = \min_{\mathbf{u}\ \mathbf{f}} ||\ \downarrow\ ||^2$$

Simple Particle Example

- Dynamics equation: $\mathbf{u} - \mathbf{g} = m\ddot{\mathbf{x}}$
- Inverse dynamics function:

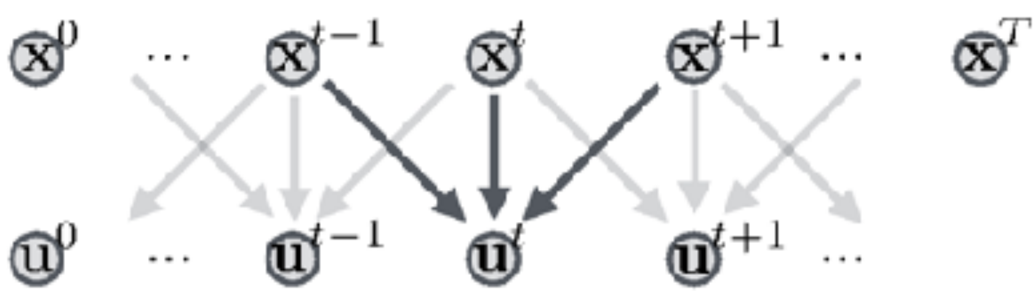
$$f^{-1}(\mathbf{x}^{t-1}, \mathbf{x}^{t}, \mathbf{x}^{t+1}) = \mathbf{u}^{t} = m(\mathbf{x}^{t-1} - 2\mathbf{x}^{t} + \mathbf{x}^{t+1})/\delta t + \mathbf{g}$$

- Cost: $C(\mathbf{x}) = ||\mathbf{x}||^2$
- Known:

  Initial state: $\mathbf{x}^0$ System parameters: m External forces: $\mathbf{g}$

- Optimization unknowns: $\mathbf{x}^1, ..., \mathbf{x}^T$
- Solution:

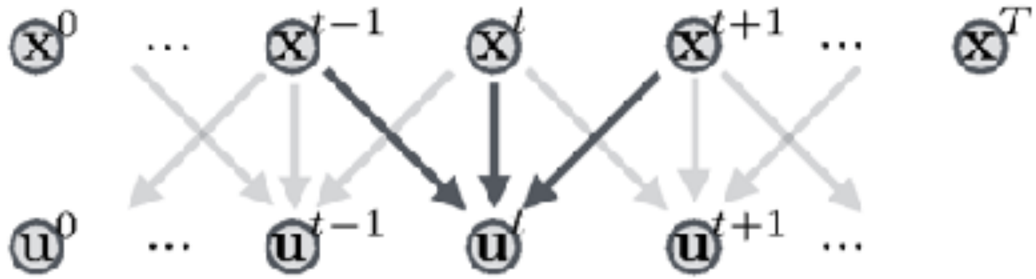
  States: $\mathbf{x}^1, ..., \mathbf{x}^T = \mathbf{0}$ Implicit controls: $\mathbf{u}^0, ..., \mathbf{x}^{T-1} = \mathbf{g}$



Outline

- Trajectory optimization and direct collocation
- Inverse dynamics model
- Numerical optimization for collocation
- Optimizing dynamics with contact
- Collocation methods for policy learning

Numerical Solutions for Direct Collocation Methods



- First Thought: Set up a TensorFlow graph and optimize with gradient descent
- For shooting methods we had 2nd order methods (Iterative LQR, DDP)
- For direct collocation we also can apply a truncated 2nd order method

Gauss-Newton Method

(recall Natural Gradient from lec. 6)

- Total trajectory cost is

$$C(\mathbf{X}) = \sum_t c(\phi^t(\mathbf{X}))$$

Gauss-Newton Method

- Total trajectory cost is

$$C(\mathbf{X}) = \sum_t c(\phi^t(\mathbf{X}))$$

includes inverse dynamics residual
and any cost function features

Gauss-Newton Method

- Total trajectory cost is

$$C(\mathbf{X}) = \sum_t c(\phi^t(\mathbf{X}))$$

- Its gradient and truncated Hessian are

$$C_{\mathbf{X}} = \sum_t c^t_{\phi} \phi^t_{\mathbf{X}}$$

$$C_{\mathbf{XX}} = \sum_t (\phi^t_{\mathbf{X}})^\top c^t_{\phi\phi} \phi^t_{\mathbf{X}} + c^t_{\phi} \phi^t_{\mathbf{XX}} \approx \sum_t (\phi^t_{\mathbf{X}})^\top c^t_{\phi\phi} \phi^t_{\mathbf{X}}$$

Gauss-Newton Method

- Total trajectory cost is

$$C(\mathbf{X}) = \sum_t c(\phi^t(\mathbf{X}))$$

- Its gradient and truncated Hessian are

$$C_{\mathbf{X}} = \sum_t c^t_{\phi} \phi^t_{\mathbf{X}}$$

$$C_{\mathbf{XX}} = \sum_t (\phi^t_{\mathbf{X}})^\top c^t_{\phi\phi} \phi^t_{\mathbf{X}} + c^t_{\phi} \phi^t_{\mathbf{XX}} \approx \sum_t (\phi^t_{\mathbf{X}})^\top c^t_{\phi\phi} \phi^t_{\mathbf{X}}$$

- Find optimal solution by iterative Gauss-Newton steps

$$\mathbf{X}^* = \mathbf{X}^* - C^{-1}_{\mathbf{XX}} C_{\mathbf{X}}$$

Gauss-Newton Method

- Total trajectory cost is

$$C(\mathbf{X}) = \sum_t c(\phi^t(\mathbf{X}))$$

- Its gradient and truncated Hessian are

$$C_{\mathbf{X}} = \sum_t c^t_{\phi} \phi^t_{\mathbf{X}}$$

$$C_{\mathbf{XX}} = \sum_t (\phi^t_{\mathbf{X}})^\top c^t_{\phi\phi} \phi^t_{\mathbf{X}} + c^t_{\phi} \phi^t_{\mathbf{XX}} \approx \sum_t (\phi^t_{\mathbf{X}})^\top c^t_{\phi\phi} \phi^t_{\mathbf{X}}$$

- Find optimal solution by iterative Gauss-Newton steps

$$\mathbf{X}^* = \mathbf{X}^* - C^{-1}_{\mathbf{XX}} C_{\mathbf{X}}$$

Typically use damped Hessian
(similar to Trust Region)

$$(C^{-1}_{\mathbf{XX}} + \lambda \mathrm{I}) C_{\mathbf{X}}$$

Recall Natural Gradient (Lec. 6). Can you see the commonalities?

Natural Gradient

- Consider a standard maximum likelihood problem: $$\max_\theta f(\theta) = \max_\theta \sum_i \log p(x^{(i)};\theta)$$

- Gradient: $$\frac{\partial f(\theta)}{\partial \theta_p} = \sum_i \frac{\partial \log p(x^{(i)};\theta)}{\partial \theta_p} = \sum_i \frac{\partial p(x^{(i)};\theta)}{\partial \theta_p} \frac{1}{p(x^{(i)};\theta)}$$

- Hessian: $$\frac{\partial^2 f(\theta)}{\partial \theta_q \partial \theta_p} = \sum_i \frac{\partial^2 p(x^{(i)};\theta)}{\partial \theta_q \partial \theta_p} \frac{1}{p(x^{(i)};\theta)} - \frac{\partial p(x^{(i)};\theta)}{\partial \theta_q} \frac{1}{p(x^{(i)};\theta)} \frac{\partial p(x^{(i)};\theta)}{\partial \theta_p} \frac{1}{p(x^{(i)};\theta)}$$

$$\nabla^2 f(\theta) = \sum_i \frac{\nabla^2 p(x^{(i)};\theta)}{p(x^{(i)};\theta)} - \left(\nabla \log p(x^{(i)};\theta)\right) \left(\nabla \log p(x^{(i)};\theta)\right)^\top$$

- Natural gradient: $$= \left(\sum_i \left(\nabla \log p(x^{(i)};\theta)\right) \left(\nabla \log p(x^{(i)};\theta)\right)^\top\right)^{-1} \left(\sum_i \nabla \log p(x^{(i)};\theta)\right)$$

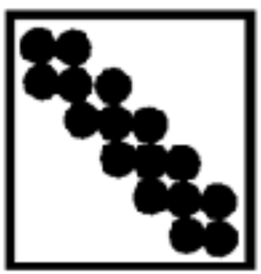
only keeps the 2nd term in the Hessian. Benefits: (1) faster to compute (only gradients needed); (2) guaranteed to be negative definite; (3) found to be superior in some experiments; (4) invariant to re-parameterization

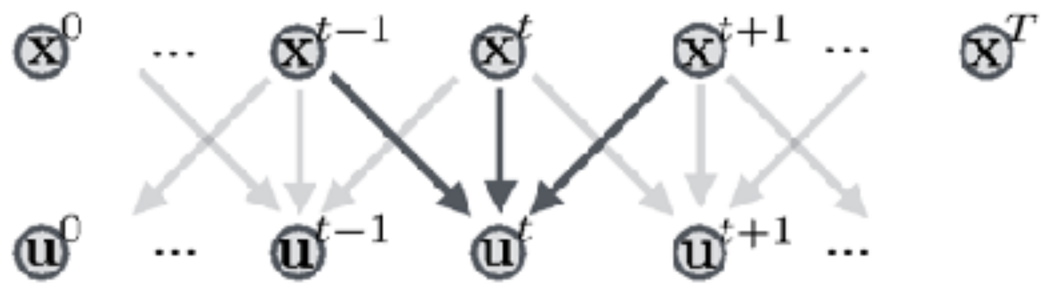
Gauss-Newton Method

- Requires inverting $(|\mathbf{x}|T) \times (|\mathbf{x}|T)$ Hessian every time???

$$\mathbf{X}^* = \mathbf{X}^* - C_{\mathbf{X}\mathbf{X}}^{-1} C_{\mathbf{X}}$$

- Hessian is block sparse





- Can use sparse linear system solvers
  - python: linalg.spsolve
  - Other methods possible (multigrid, projection, spectral?)
  - Constrained optimization possible (SQP) [Posa and Tedrake 12]

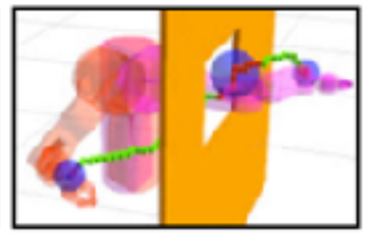
Outline

- Trajectory optimization and direct collocation
- Inverse dynamics model
- Numerical optimization for collocation
- Optimizing dynamics with contact
- Collocation methods for policy learning

Dynamics with Contact

- Both shooting and collocation methods can be applied to control of movement without contact
  - flying, driving, swimming robots, collision-free paths





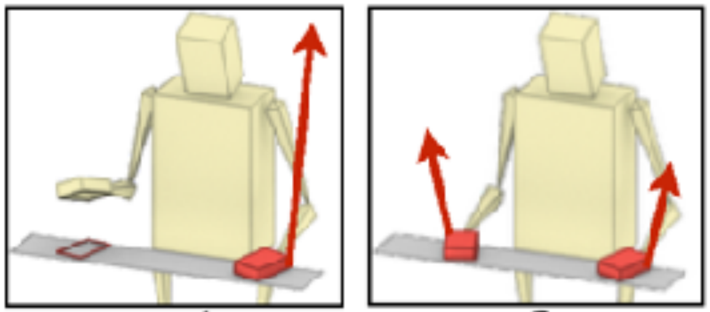
Dynamics with Contact

- Both shooting and collocation methods can be applied to control of movement without contact
  - flying, driving, swimming robots, collision-free paths

- With contact, it is difficult to apply either method
  - legged robots, manipulation

Dynamics with Contact

- Discontinuous jumps in contact forces (and their number)



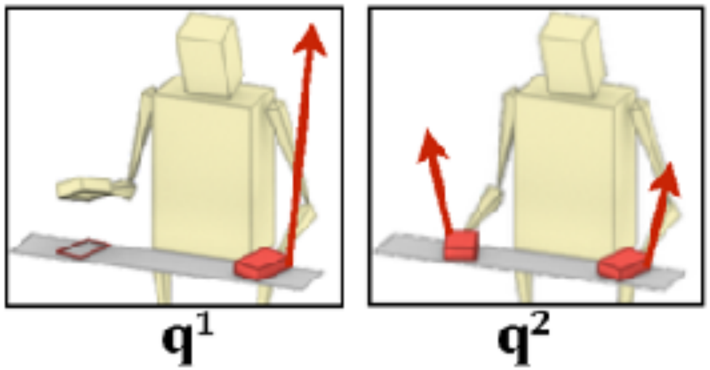
$\mathbf{q}^1$ $\mathbf{q}^2$

Dynamics equation:

$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

Dynamics with Contact

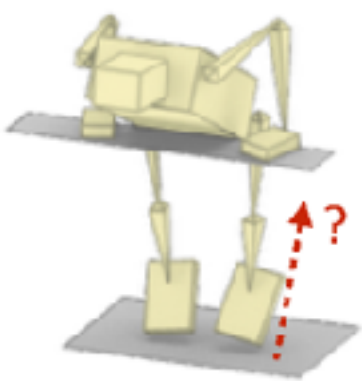
- Discontinuous jumps in contact forces (and their number)



Dynamics equation:

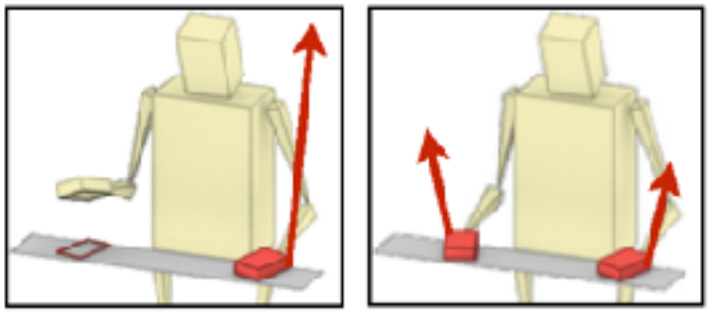
$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

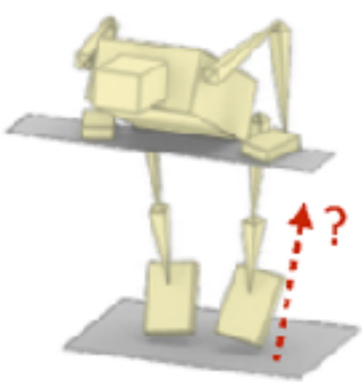
- No gradient information from inactive contacts



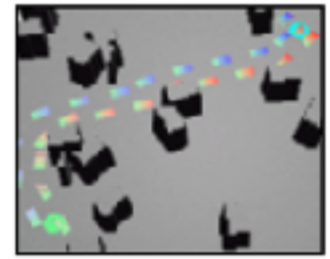
Can't anticipate being able to apply forces

Dynamics with Contact



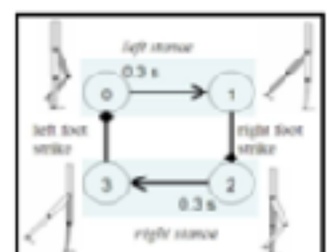


manual specification

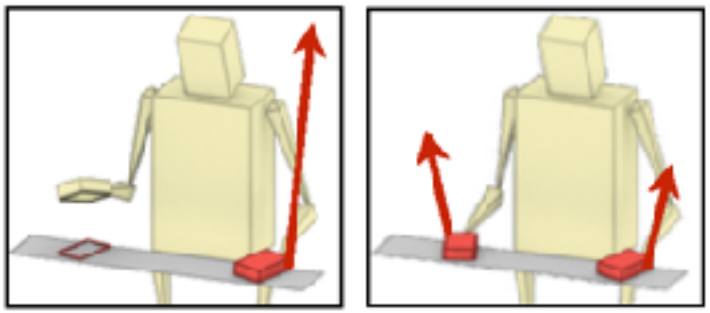


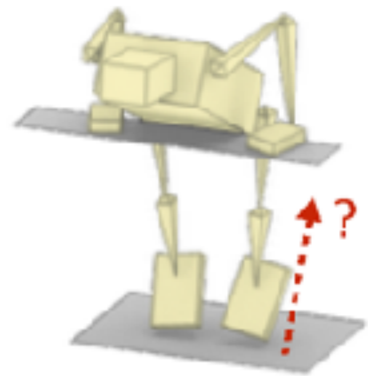
track demonstrations

motion structure



Dynamics with Contact





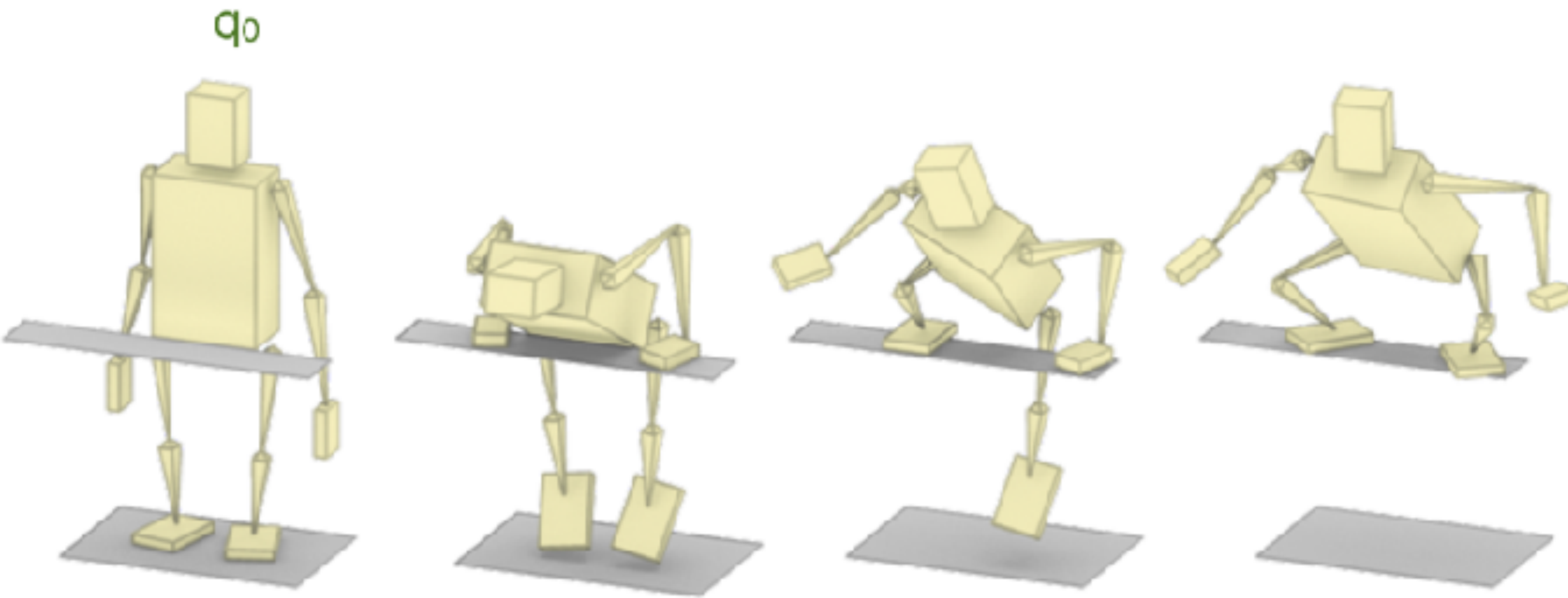
- Contact activity is an indirect function of state
- What if we make contact activity a direct optimization variable like we did for state?

Contact-Invariant Optimization

[Mordatch, Todorov, Popovic, SIGGRAPGH 2012]

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

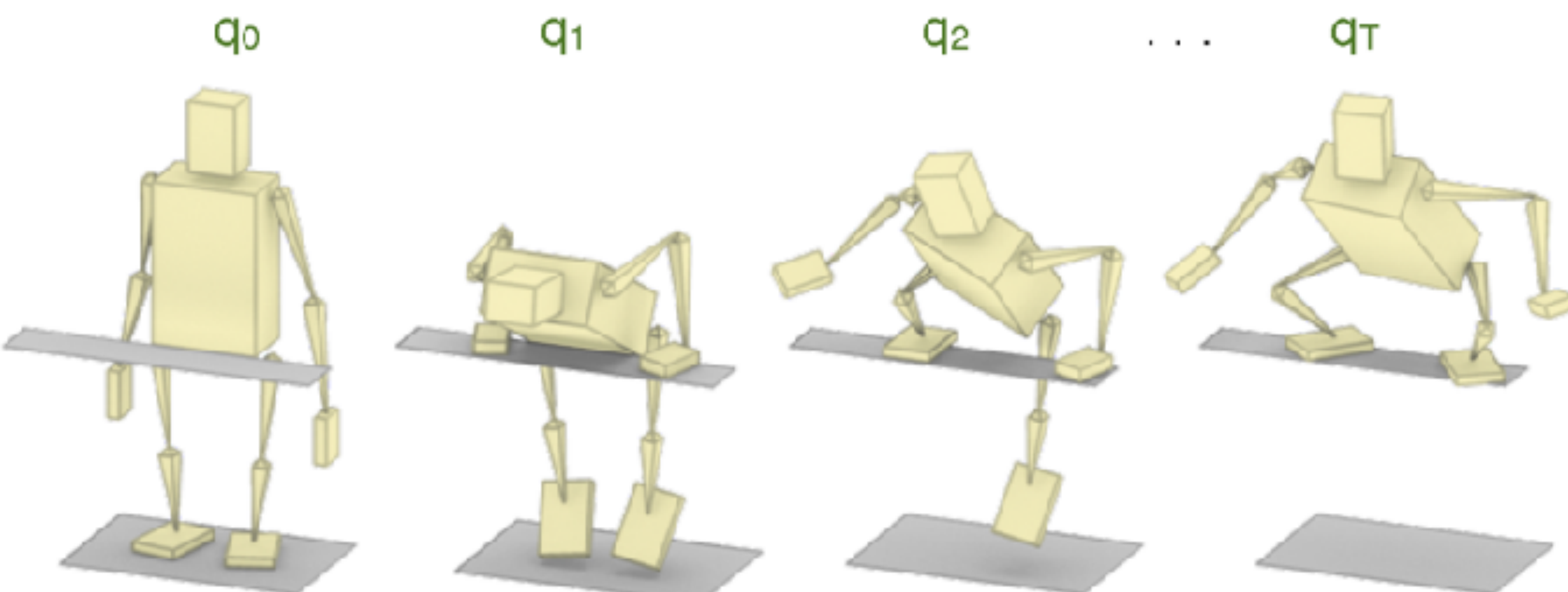
$\mathbf{x}$: [ ]



Contact-Invariant Optimization

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

$\mathbf{x}$: [q]

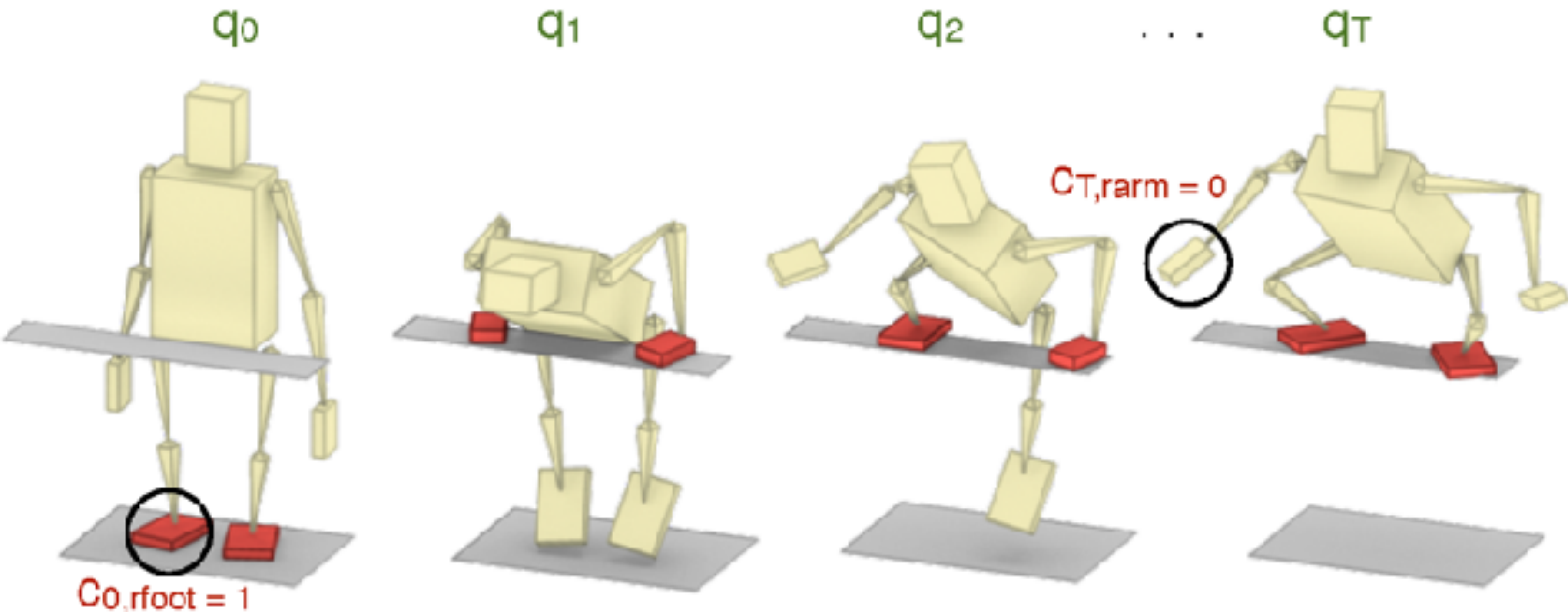


Contact-Invariant Optimization

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

x: [q c]

$c_{t,n} = 1$: foot/hand n is in contact with ground at time t



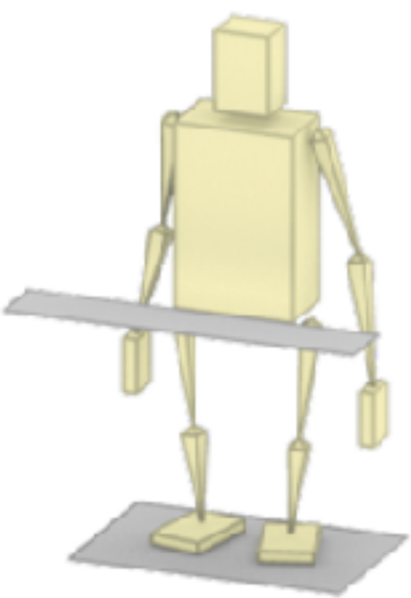
Contact-Invariant Optimization

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

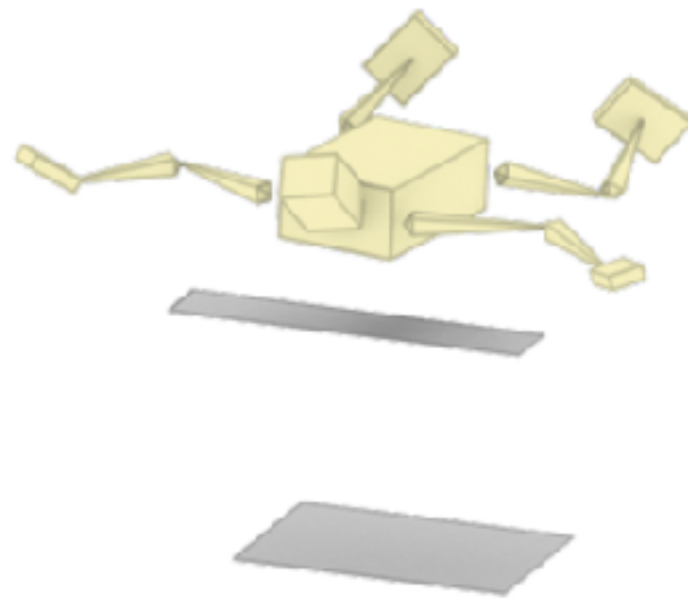
$\mathbf{x}$: [q c]

enforce **contact** and **dynamics** consistency between q and c

q_0



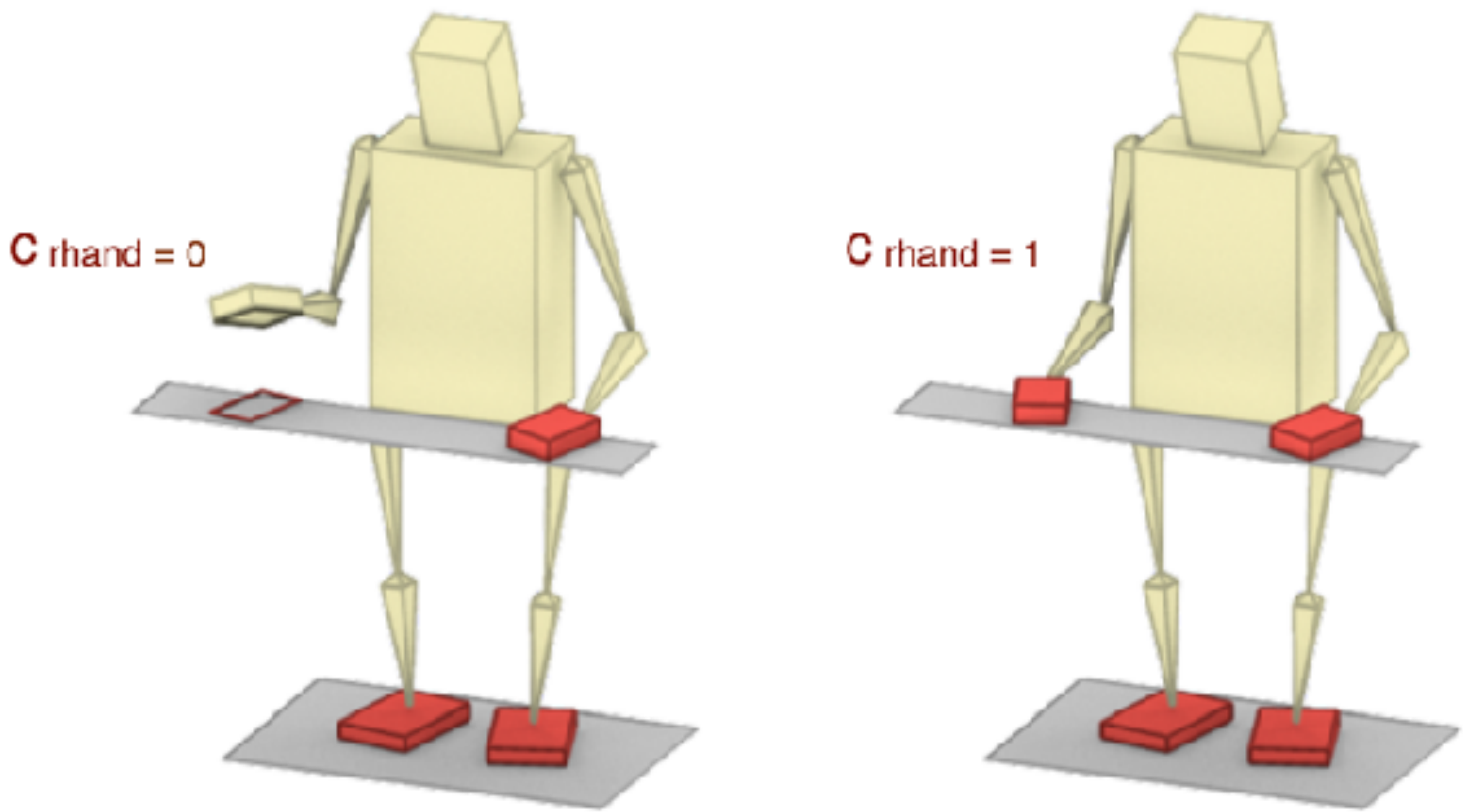
q_K



Contact Consistency

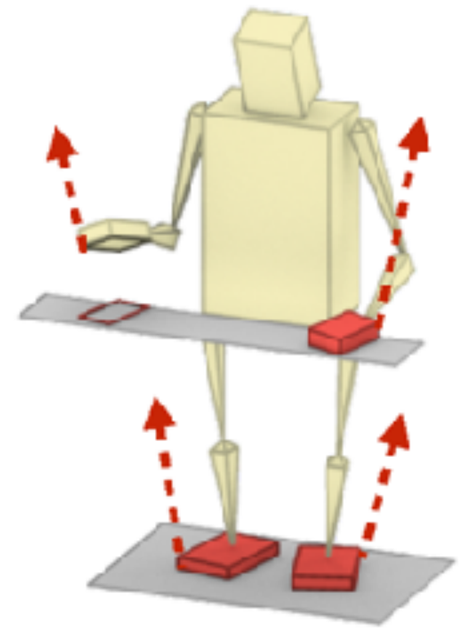
When c_n = 1 limb n must be touching ground and not sliding

When c_n = 0 limb n is unconstrained



Dynamics Consistency

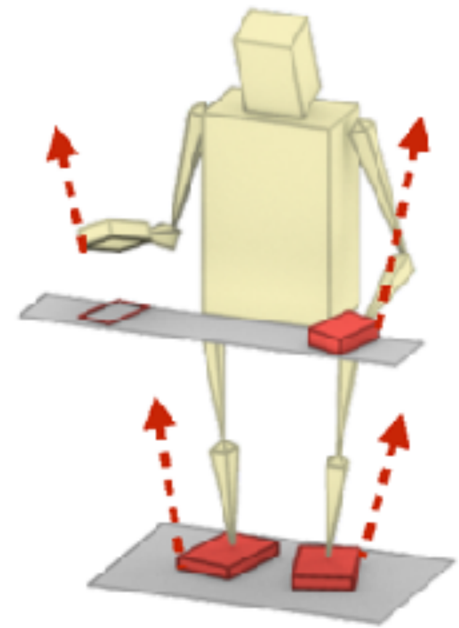
$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$



$$f^{-1}(\mathbf{x}^{t-1},\mathbf{x}^{t},\mathbf{x}^{t+1}) = \arg\min_{\mathbf{u}\,\mathbf{f}} ||\quad||^2$$

All forces are active (contact set is constant)

Dynamics Consistency



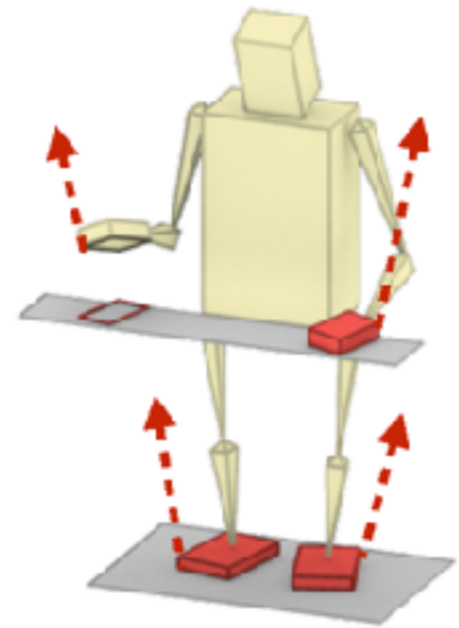
$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$

$$f^{-1}(\mathbf{x}^{t-1},\mathbf{x}^{t},\mathbf{x}^{t+1}) = \arg\min_{\mathbf{u}\,\mathbf{f}} \|\quad\|^2 + \sum_i \|\mathbf{f}_i\|^2/(c_i+\epsilon)$$

All forces are active (contact set is constant)

High penalties for using forces where c = 0

Dynamics Consistency



$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$

$$f^{-1}(\mathbf{x}^{t-1}, \mathbf{x}^{t}, \mathbf{x}^{t+1}) = \arg\min_{\mathbf{u}\,\mathbf{f}} ||\quad||^2 + \sum_i ||\mathbf{f}_i||^2/(c_i + \epsilon)$$

All forces are active (contact set is constant)

High penalties for using forces where c = 0

trajectory optimization guides inverse dynamics solver via c

Contact-Invariant Optimization

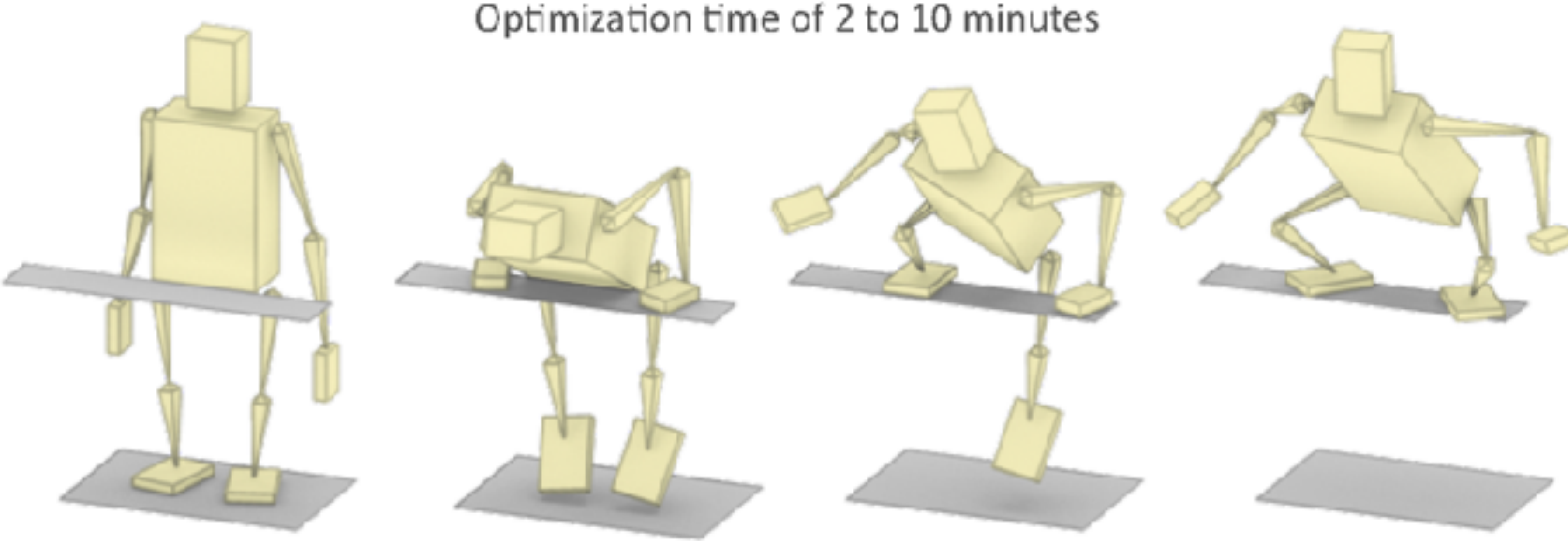
$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

$\mathbf{x}$: [q c]

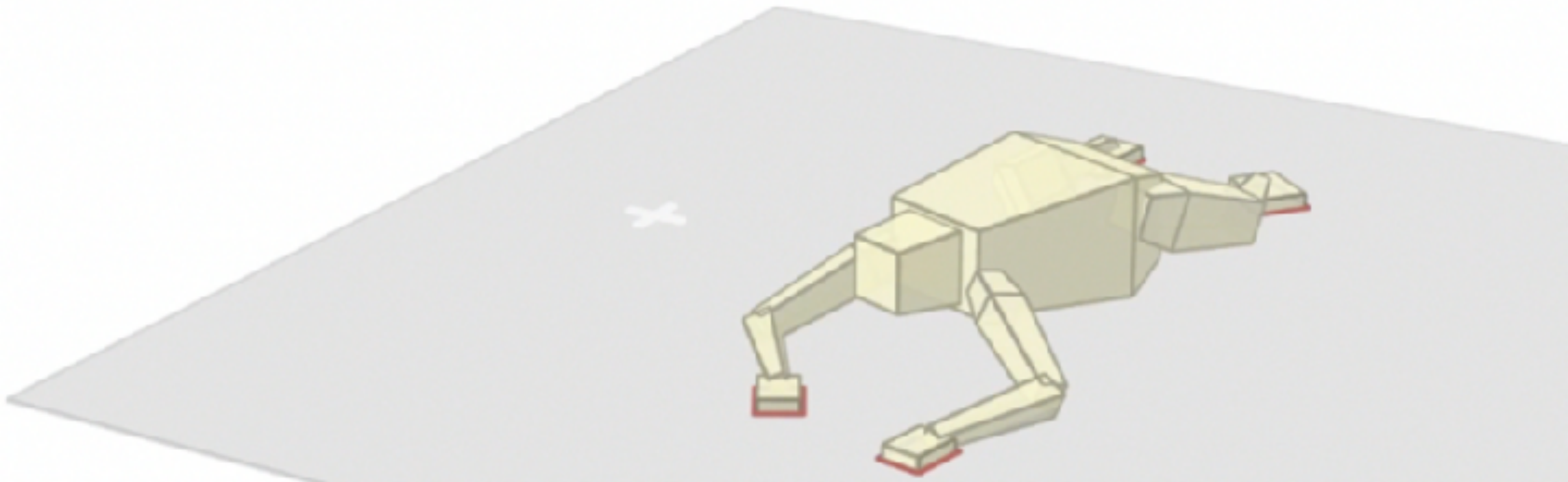
No contact discontinuities and always have a gradient

Solved with standard local optimization

Optimization time of 2 to 10 minutes

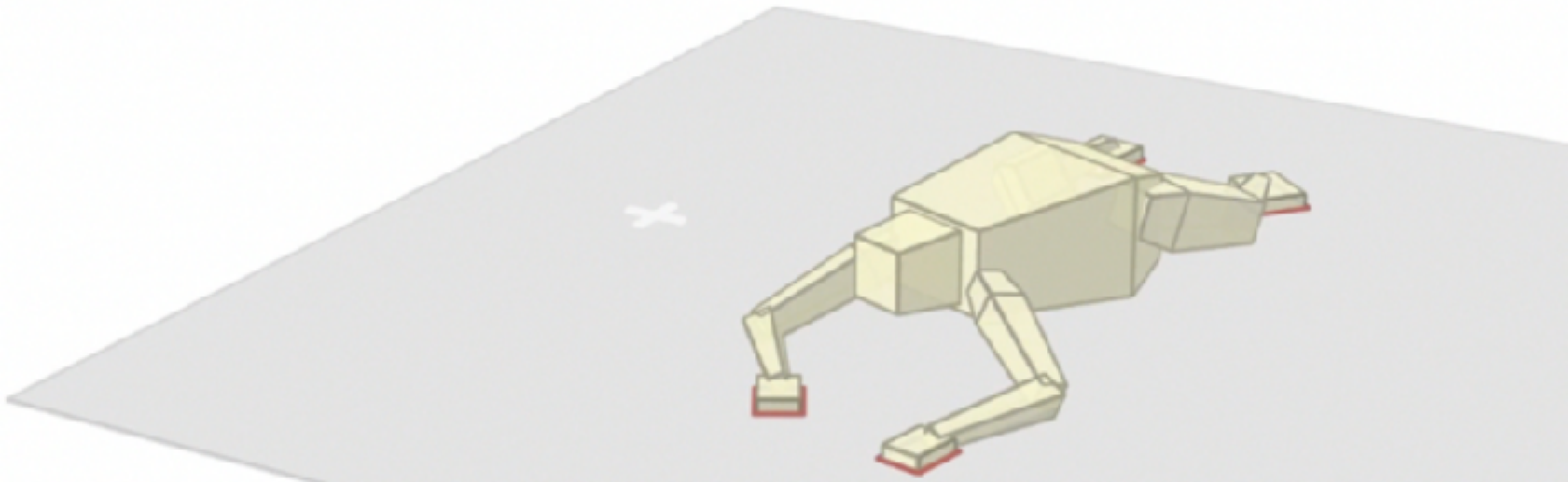


Optimization Progress



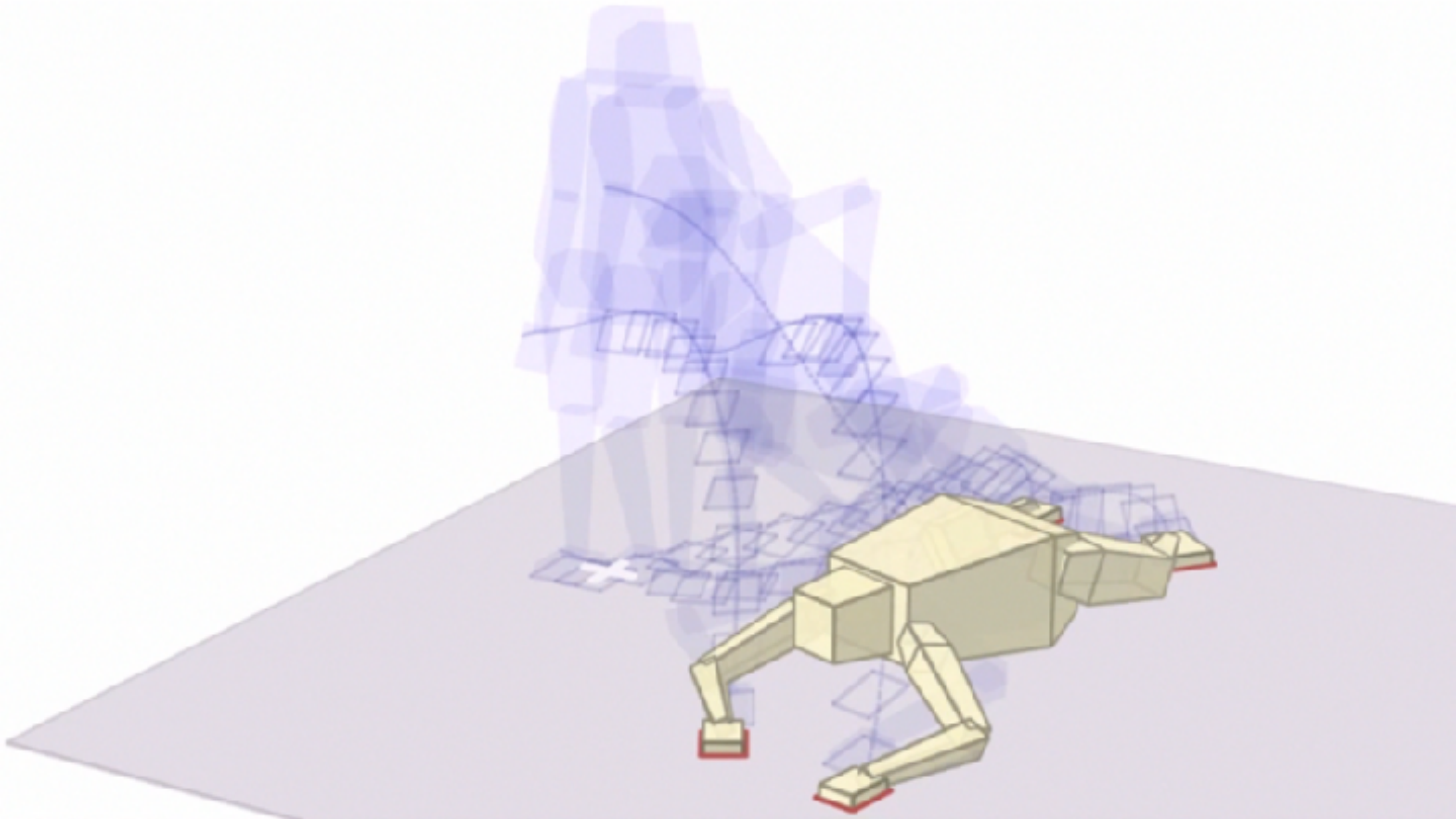
Optimization Progress

Stage 1



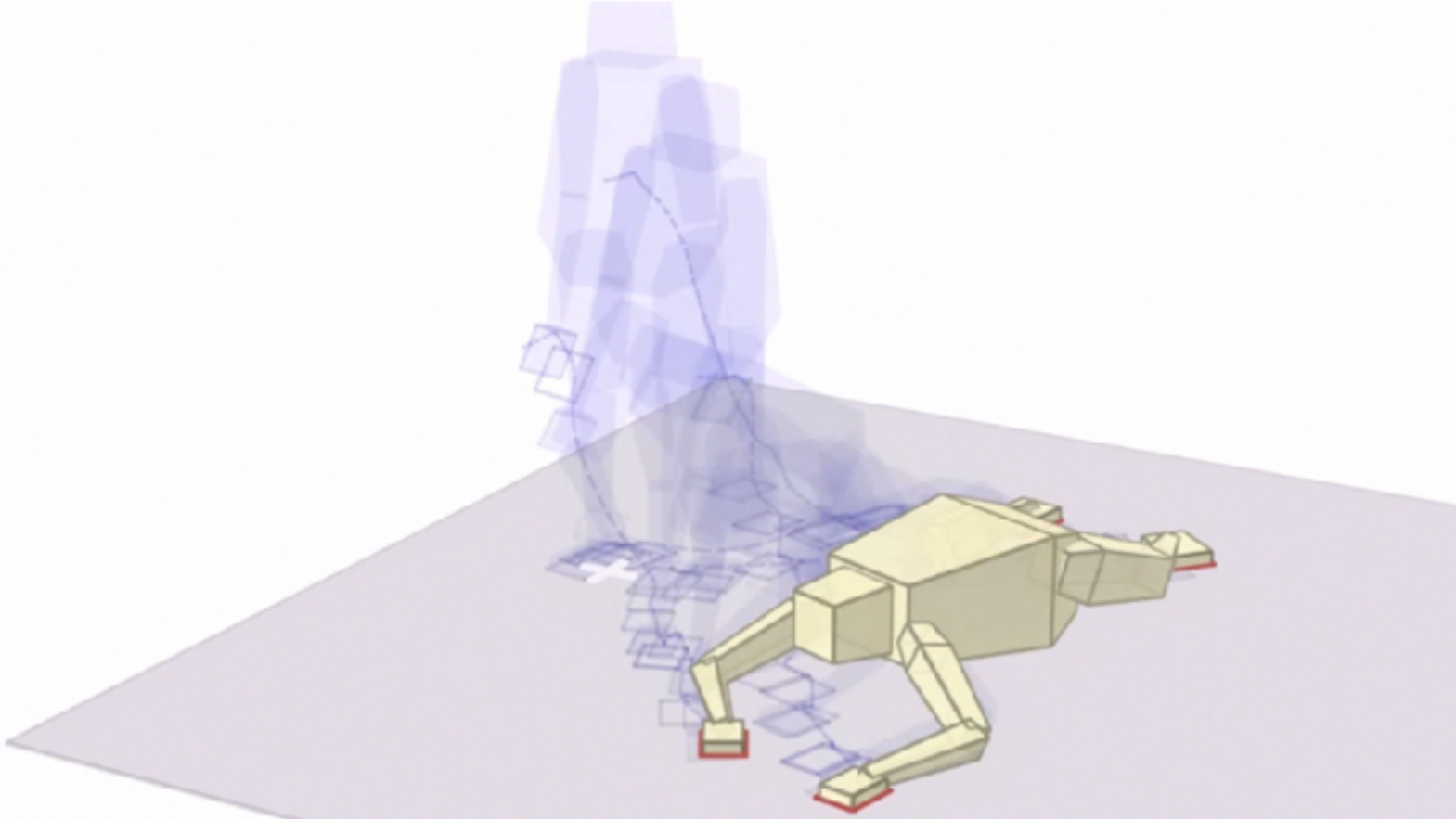
Optimization Progress

Stage 2

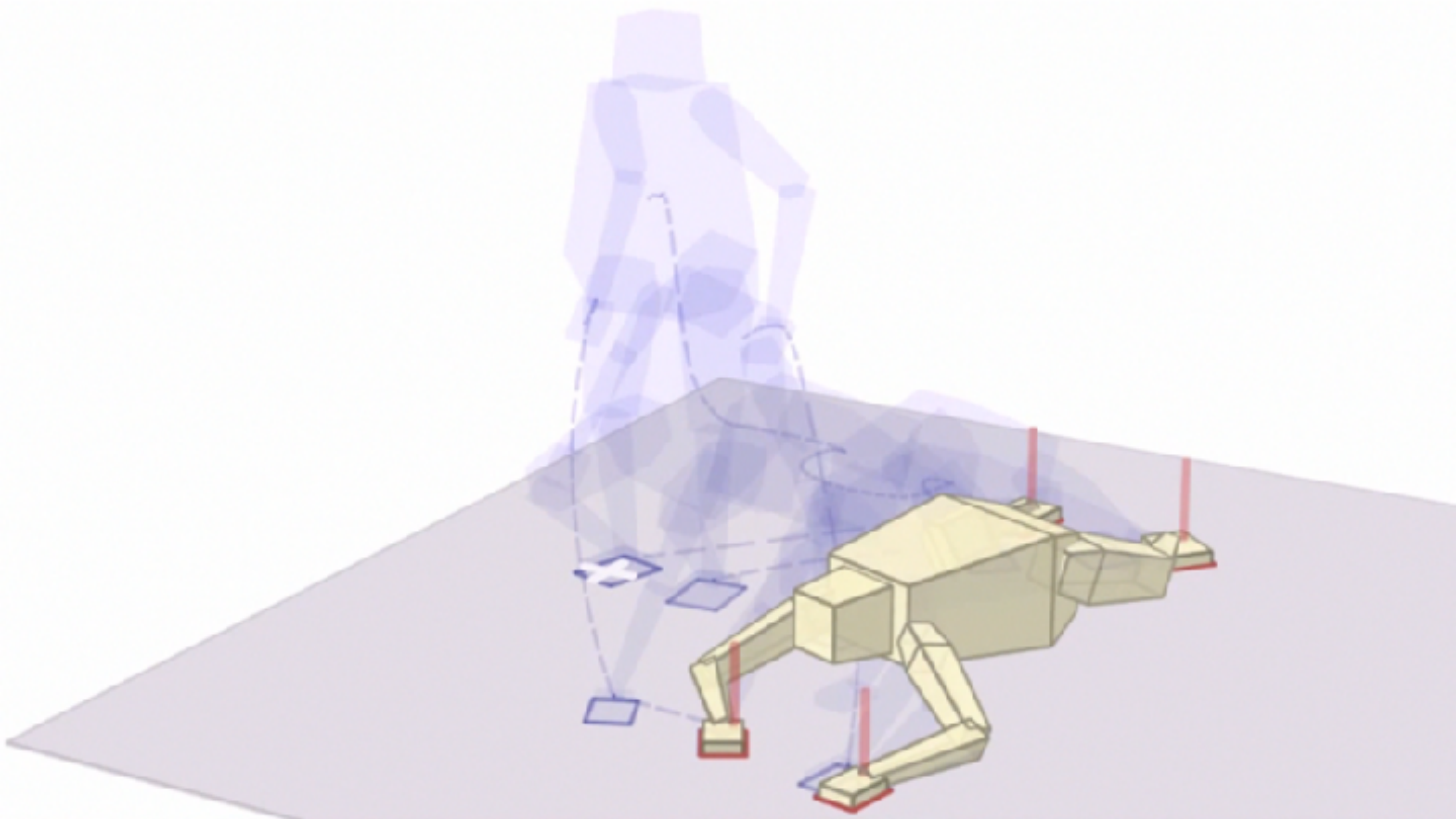


Optimization Progress

Stage 3



Optimization Result

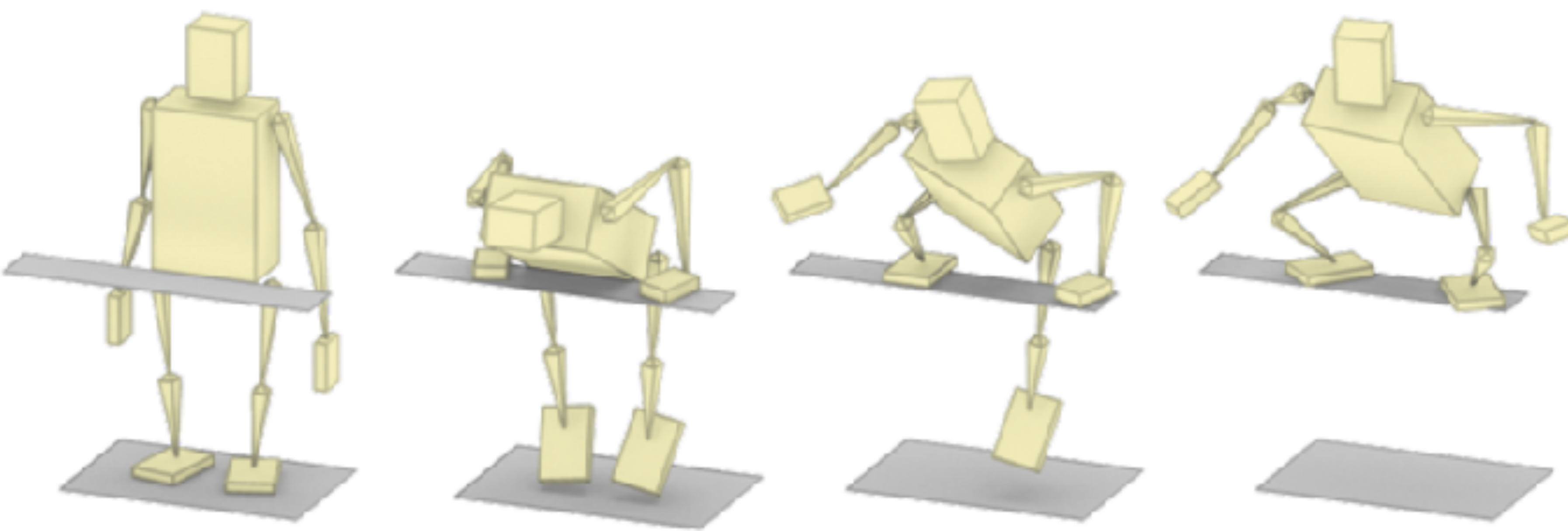


Idea

Add auxiliary variables

Softly enforce consistency between variables

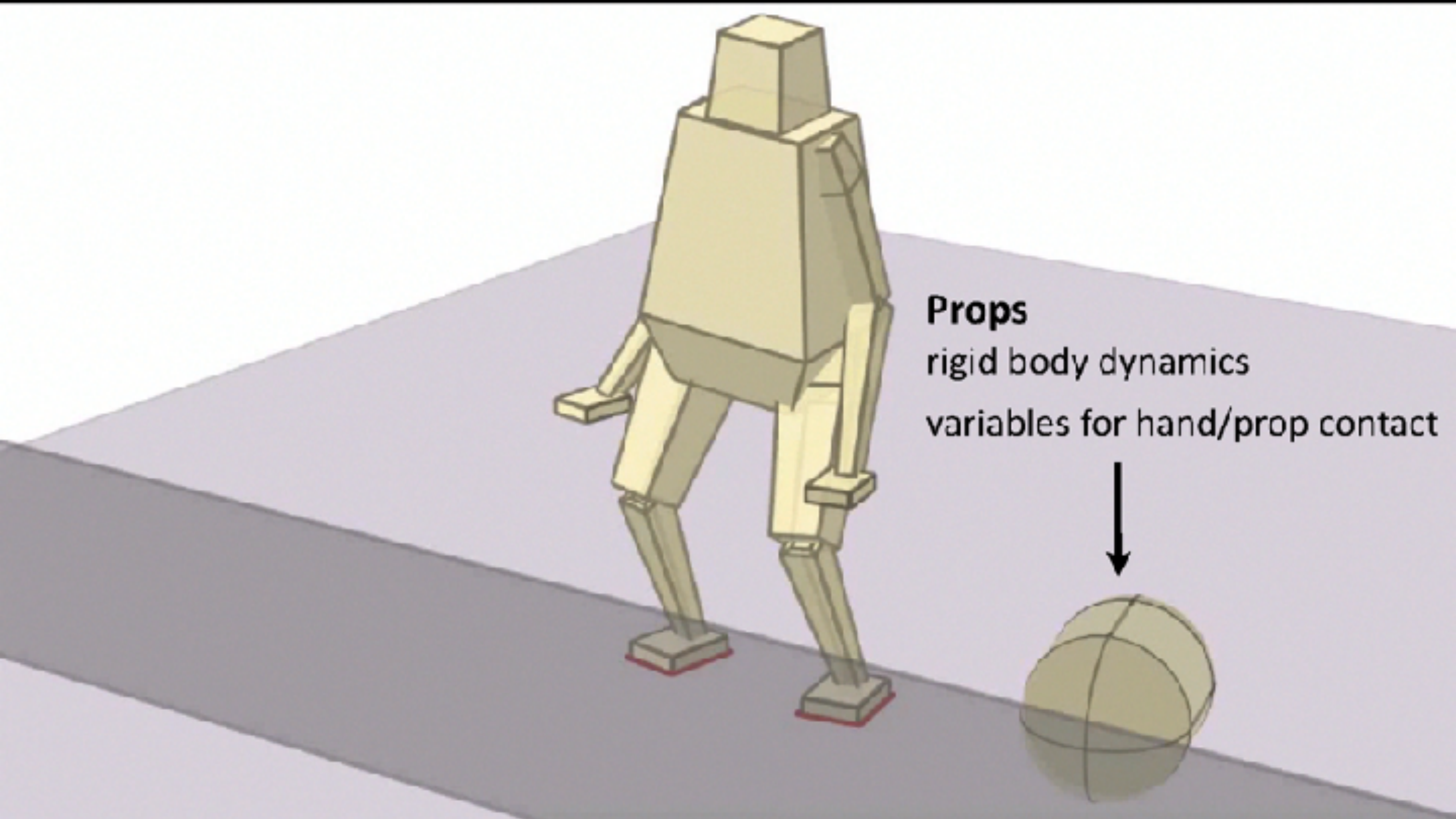
Search in larger, but easier to explore space

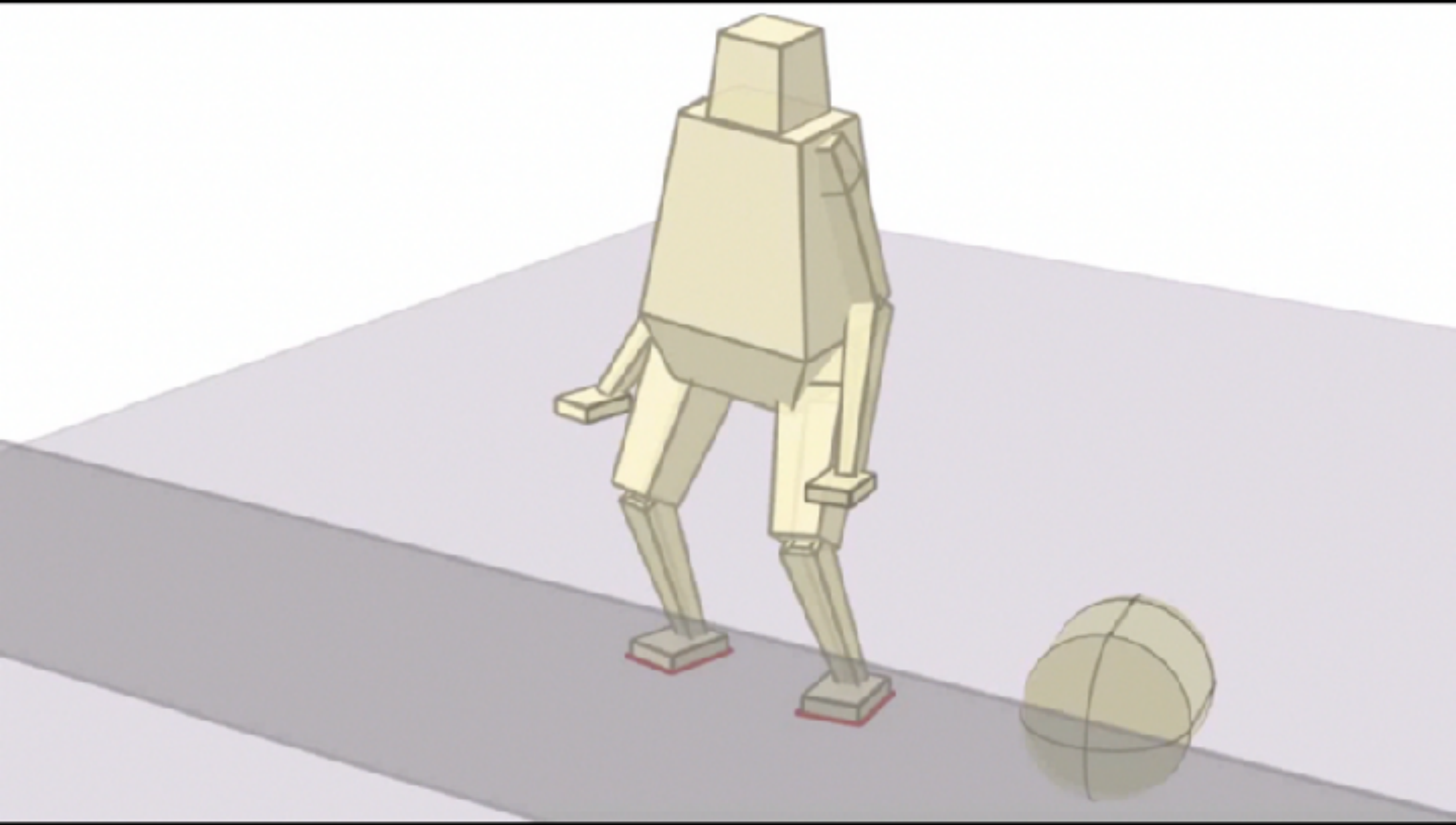


Interaction with Environment

Agile Behaviors

Non-Humanoid Character Morphologies


Props
rigid body dynamics
variables for hand/prop contact

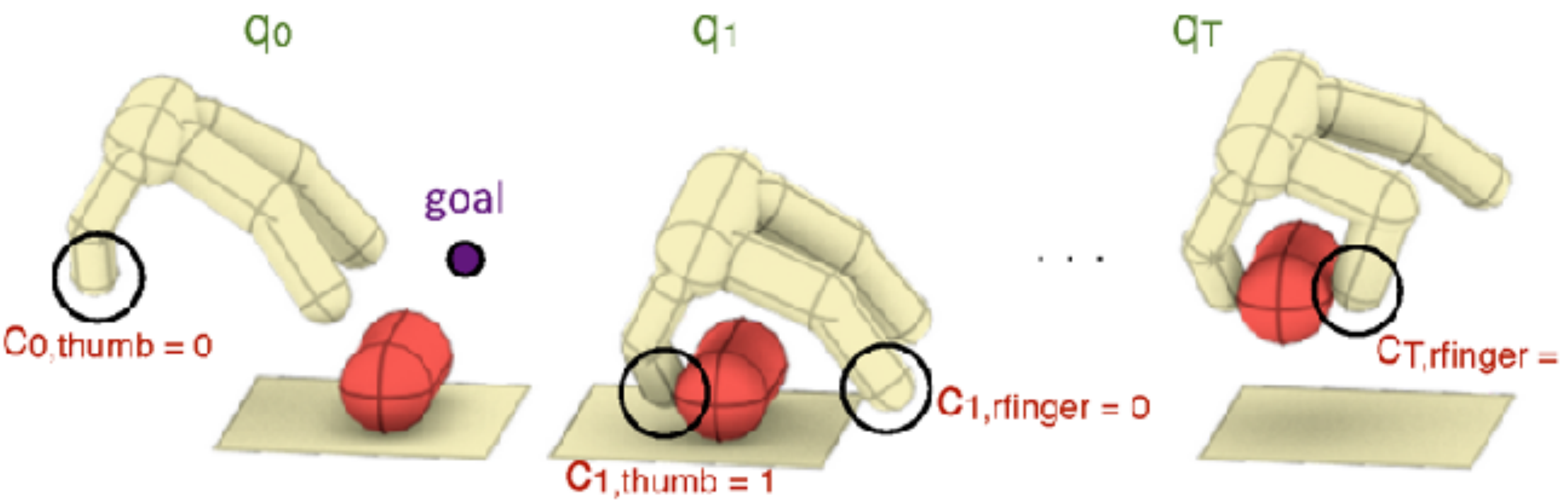


Interaction Between Multiple Characters

Hand Manipulation

$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

$\mathbf{x}$: [q c]

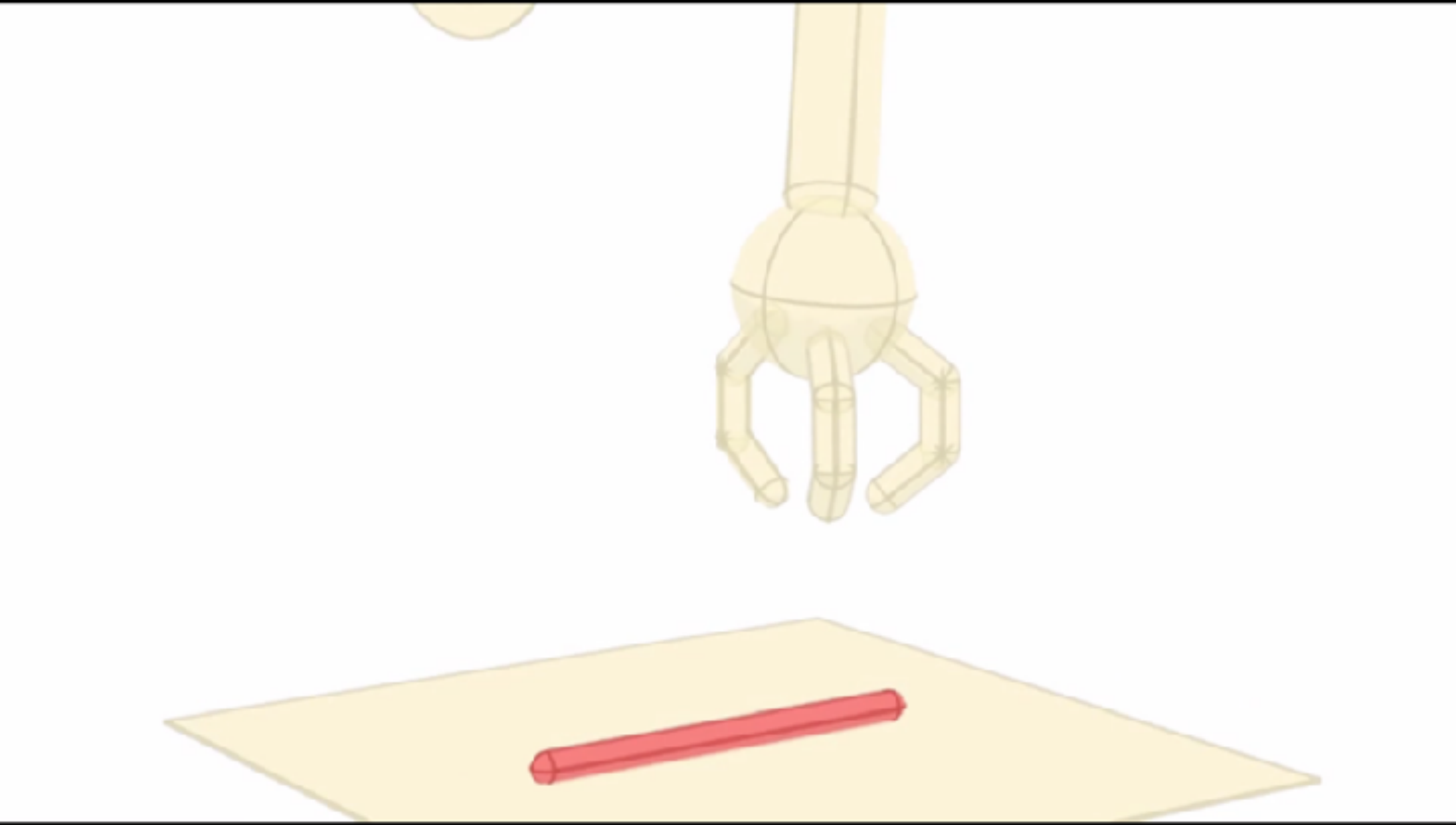


[Mordatch, Popovic, Todorov, SCA 2012]

Object Grasping

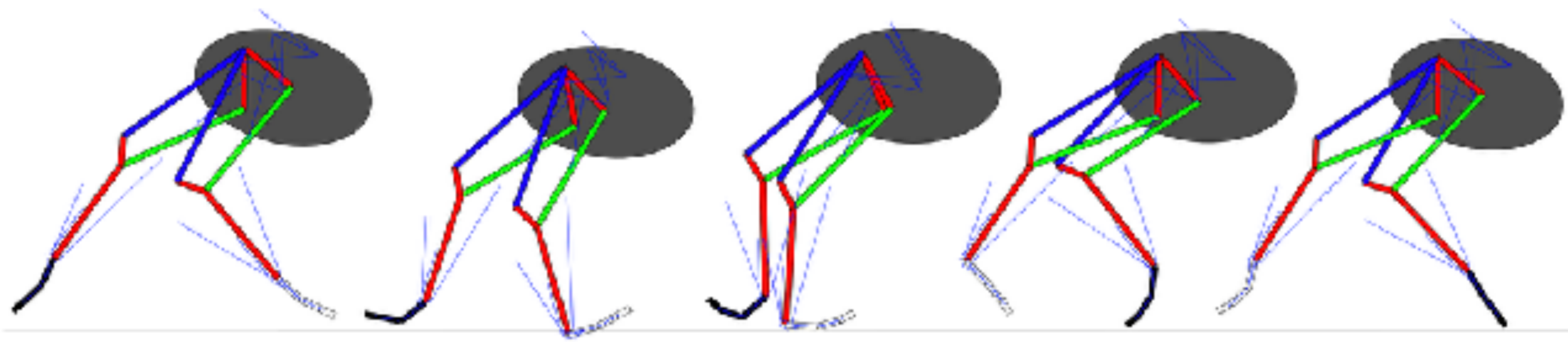
In-Hand Object Manipulation

Manipulation Tasks

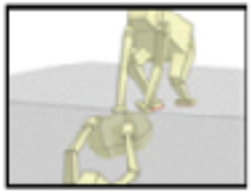


Two-Handed Manipulation

Direct Trajectory Optimization of Rigid Body Dynamical Systems Through Contact
Posa and Tedrake, 2012



$$\underset{\{h, x_0, \ldots, x_N, u_1, \ldots, u_N, \lambda_1, \ldots, \lambda_N\}}{\text{minimize}} g_f(x_N) + h \sum_{k=1}^{N} g(x_{k-1}, u_k)$$



Trajectory Optimization with Direct Collocation

Automatic and general approach

Optimization problem for each motion clip

Do we solve optimization problems to move?

No learning or reuse in optimization

Cannot deal with unexpected events

Instead of motion clips, find *policies*

Outline

- Trajectory optimization and direct collocation
- Inverse dynamics model
- Numerical optimization for collocation
- Optimizing dynamics with contact
- Collocation methods for policy learning

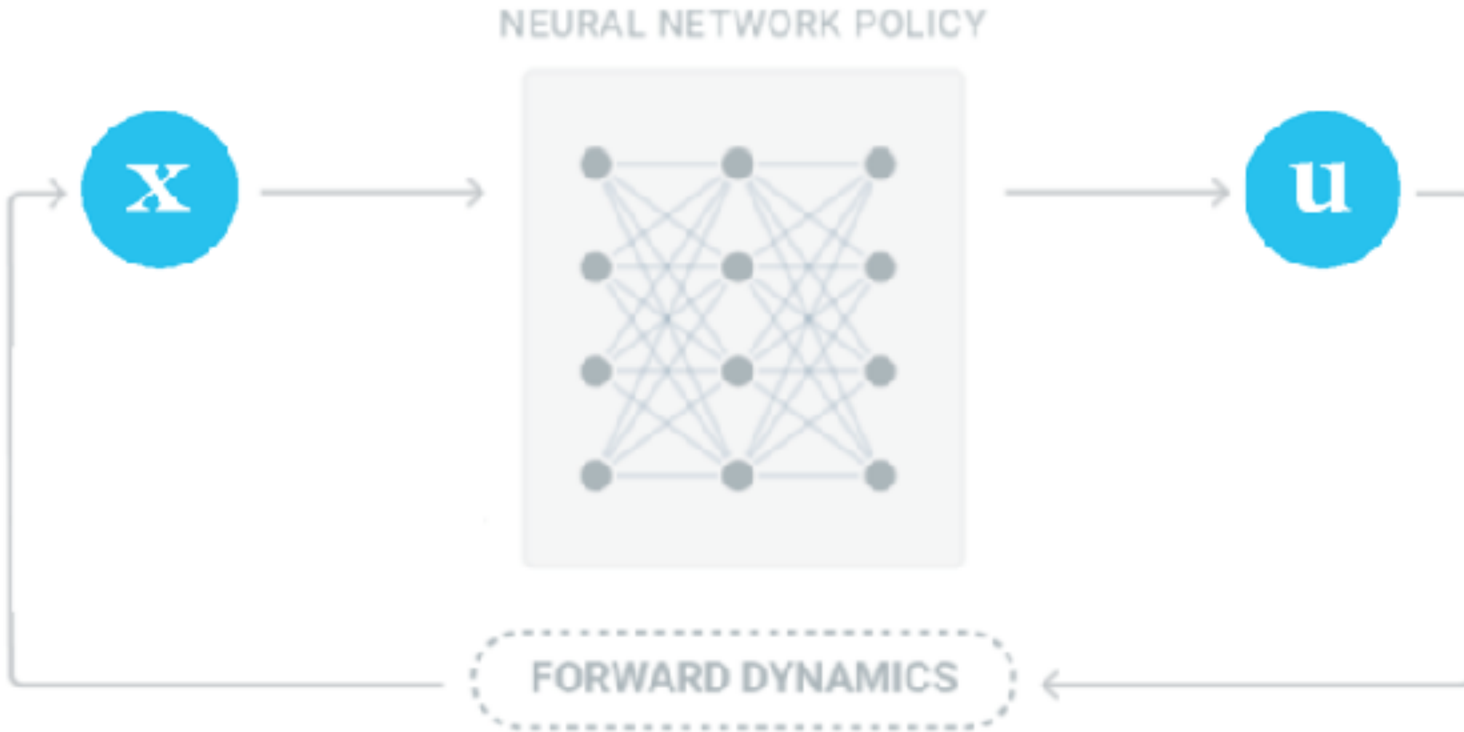
Recall from Last Lecture:

Optimal Control -- Approaches

	Return open-loop controls $u_0, u_1, \ldots, u_H$	Return feedback policy $\pi_\theta(\cdot)$ (e.g. linear or neural net)
shooting	$\min_{u_0,u_1,\ldots,u_H} c(x_0,u_0) + c(f(x_0,u_0),u_1) + c(f(f(x_0,u_0),u_1),u_2) + \ldots$	$\min_\theta c(x_0,\pi_\theta(x_0)) + c(f(x_0,\pi_\theta(x_0)),\pi_\theta(f(x_0,\pi_\theta(x_0)))) + \ldots$
collocation	$\min_{x_0,u_0,x_1,u_1,\ldots,x_H,u_H} \sum_{t=0}^H c(x_t,u_t)$ s.t. $x_{t+1} = f(x_t,u_t) \ \forall t$	$\min_{x_0,x_1,\ldots,x_H,\theta} \sum_{t=0}^H c(x_t,\pi_\theta(x_t))$ s.t. $x_{t+1} = f(x_t,\pi_\theta(x_t)) \ \forall t$ $\min_{x_0,u_0,x_1,u_1,\ldots,x_H,u_H,\theta} \sum_{t=0}^H c(x_t,u_t)$ s.t. $x_{t+1} = f(x_t,u_t) \ \forall t$, $u_t = \pi_\theta(x_t) \ \forall t$

Learning Control Policies

$$\pi_\theta : \mathbf{x} \mapsto \mathbf{u}$$



Learning Control Policies

$$\boldsymbol{\pi}_\theta \ : \ \mathbf{x} \mapsto \mathbf{u}$$

Learning Control Policies

$$\boldsymbol{\pi}_\theta \; : \; \mathbf{x} \mapsto \mathbf{u}$$

Forward Shooting:

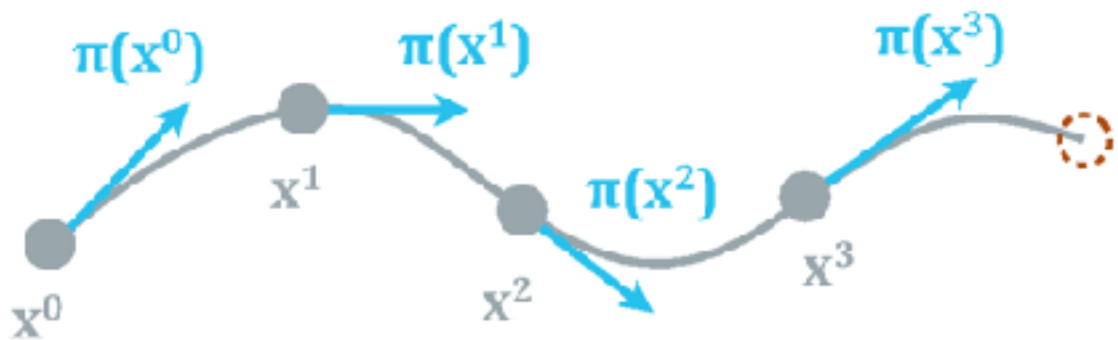
$$\min_{\theta} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \boldsymbol{\pi}_\theta(\mathbf{x}^t))$$

Learning Control Policies

$$\boldsymbol{\pi}_\theta \; : \; \mathbf{x} \; \mapsto \; \mathbf{u}$$

Forward Shooting:

$$\min_\theta \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \boldsymbol{\pi}_\theta(\mathbf{x}^t))$$

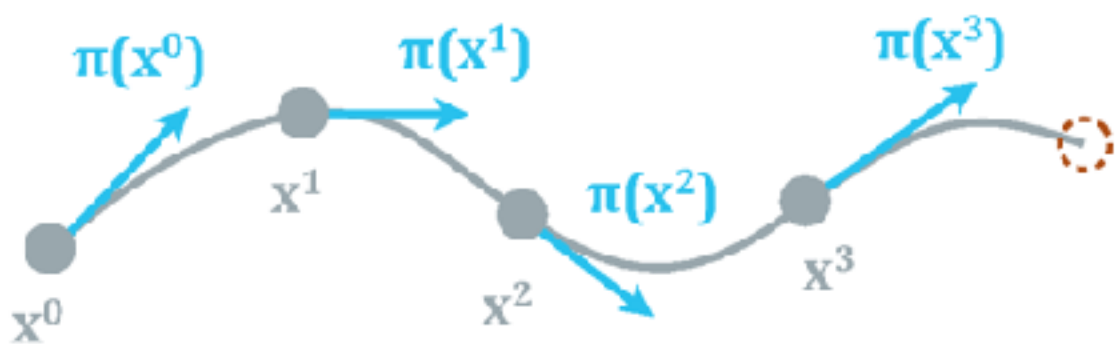


Learning Control Policies

$$\boldsymbol{\pi}_\theta \;:\; \mathbf{x} \;\mapsto\; \mathbf{u}$$

Forward Shooting:

$$\min_\theta \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \boldsymbol{\pi}_\theta(\mathbf{x}^t))$$



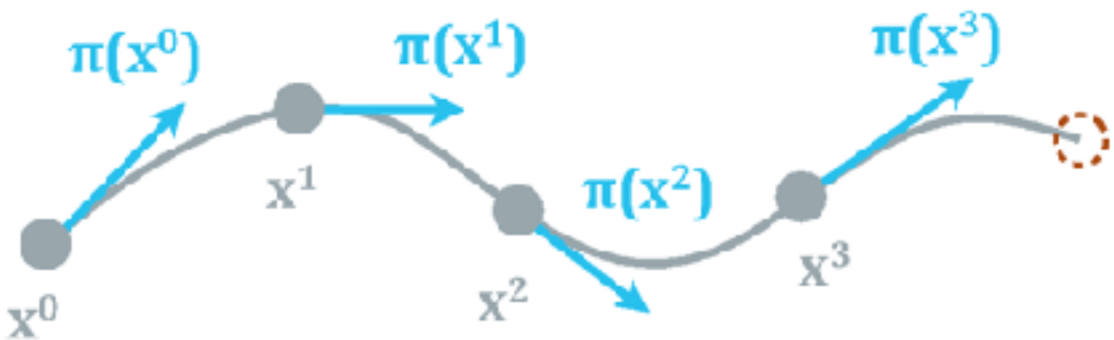
Learning Control Policies

$$\boldsymbol{\pi}_\theta \ : \ \mathbf{x} \ \mapsto \ \mathbf{u}$$

Forward Shooting:

$$\min_\theta \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \boldsymbol{\pi}_\theta(\mathbf{x}^t))$$

Poor Conditioning

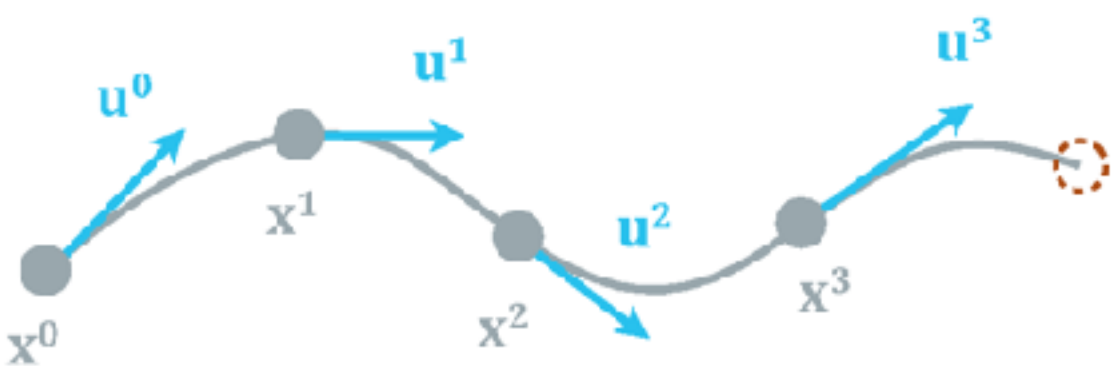


Learning Control Policies

$$\boldsymbol{\pi}_\theta \; : \; \mathbf{x} \mapsto \mathbf{u}$$

Forward Shooting:

Learning from Demonstrations:



Learning Control Policies

$$\boldsymbol{\pi}_\theta \;:\; \mathbf{x} \;\mapsto\; \mathbf{u}$$

Learning from Demonstrations:

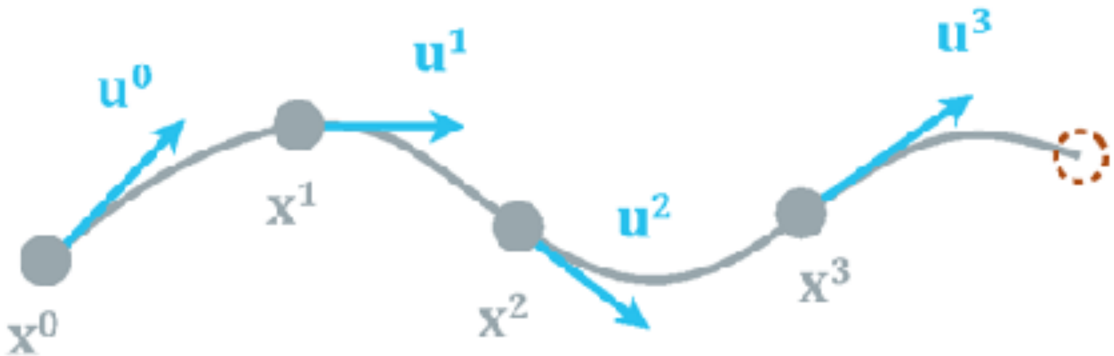
supervised learning

$$\min_\theta \sum_i ||\boldsymbol{\pi}_\theta(\mathbf{x}^i) - \mathbf{u}^i||^2$$

Training Data

input: $\mathbf{x}^i$

output: $\mathbf{u}^i$



Learning Control Policies

$$\boldsymbol{\pi}_\theta \ : \ \mathbf{x} \ \mapsto \ \mathbf{u}$$

Learning from Demonstrations:

supervised learning

$$\min_\theta \sum_i ||\boldsymbol{\pi}_\theta(\mathbf{x}^i) - \mathbf{u}^i||^2$$

Training Data

input: $\mathbf{x}^i$

output: $\mathbf{u}^i$

Where does training data come from?

- Human demonstration

Learning Control Policies

$$\boldsymbol{\pi}_\theta \; : \; \mathbf{x} \mapsto \mathbf{u}$$

Learning from Demonstrations:

supervised learning

$$\min_\theta \sum_i ||\boldsymbol{\pi}_\theta(\mathbf{x}^i) - \mathbf{u}^i||^2$$

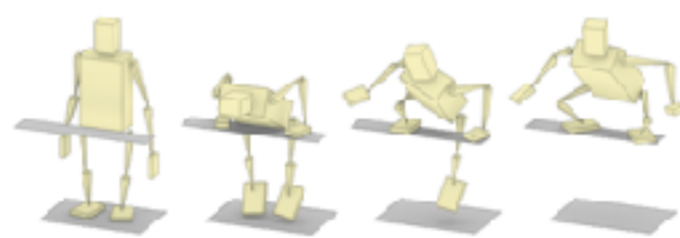
Training Data

input: $\mathbf{x}^i$

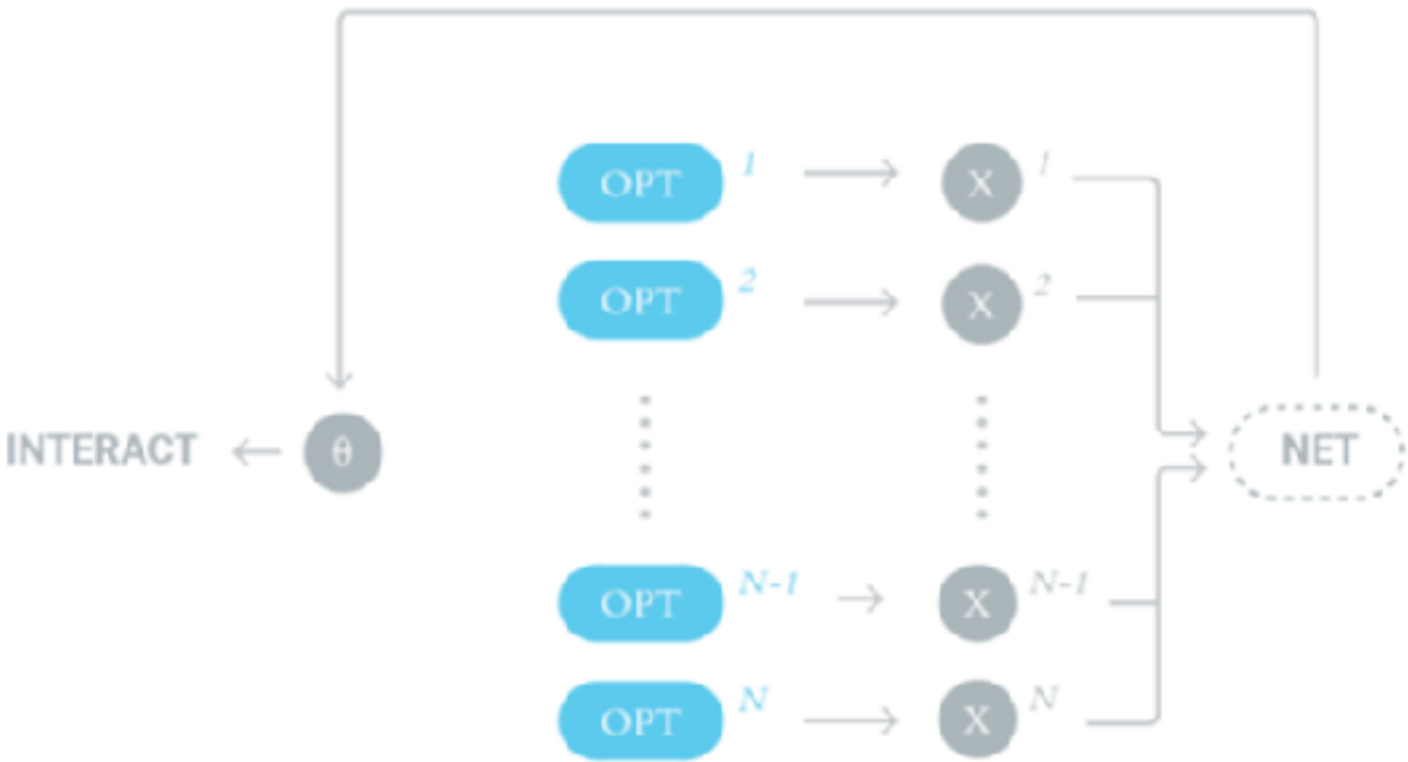
output: $\mathbf{u}^i$

Where does training data come from?

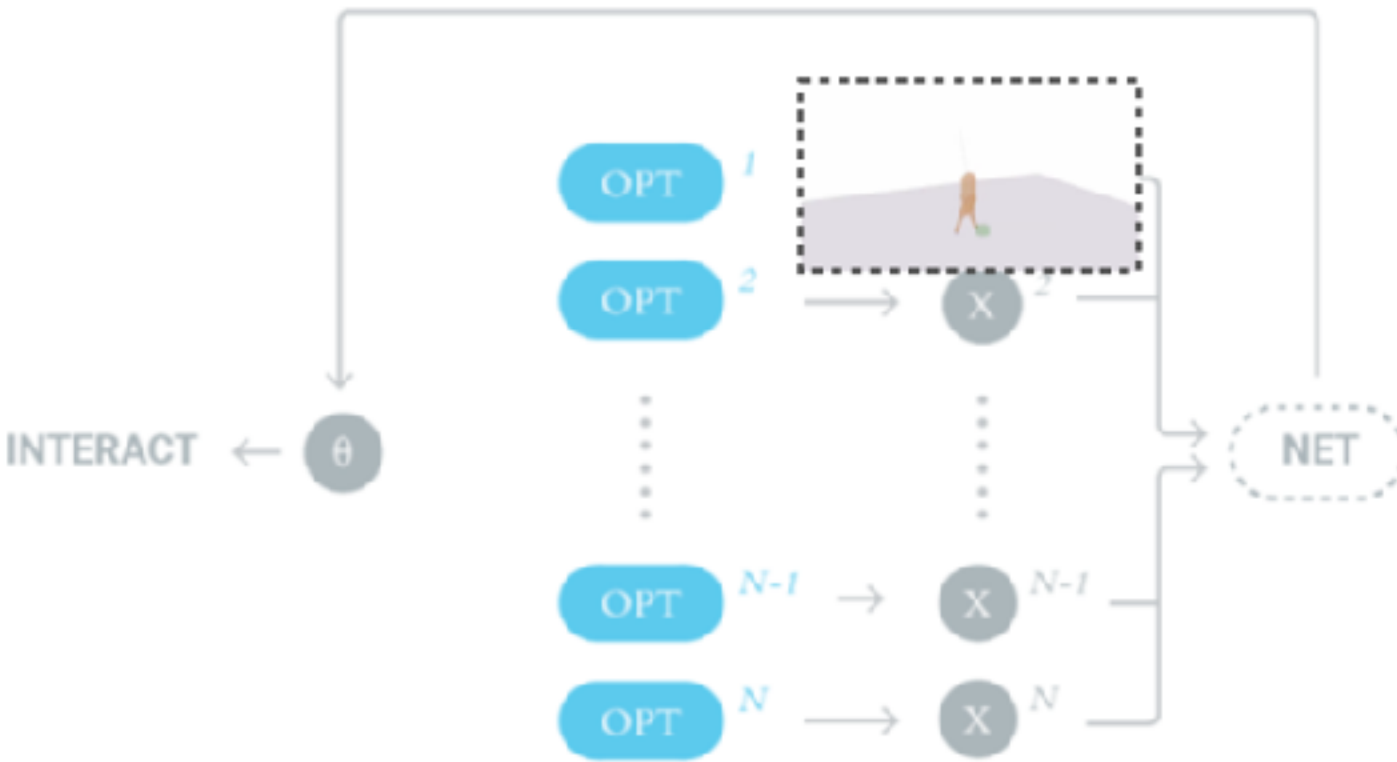
- Human demonstration
- Trajectory optimization



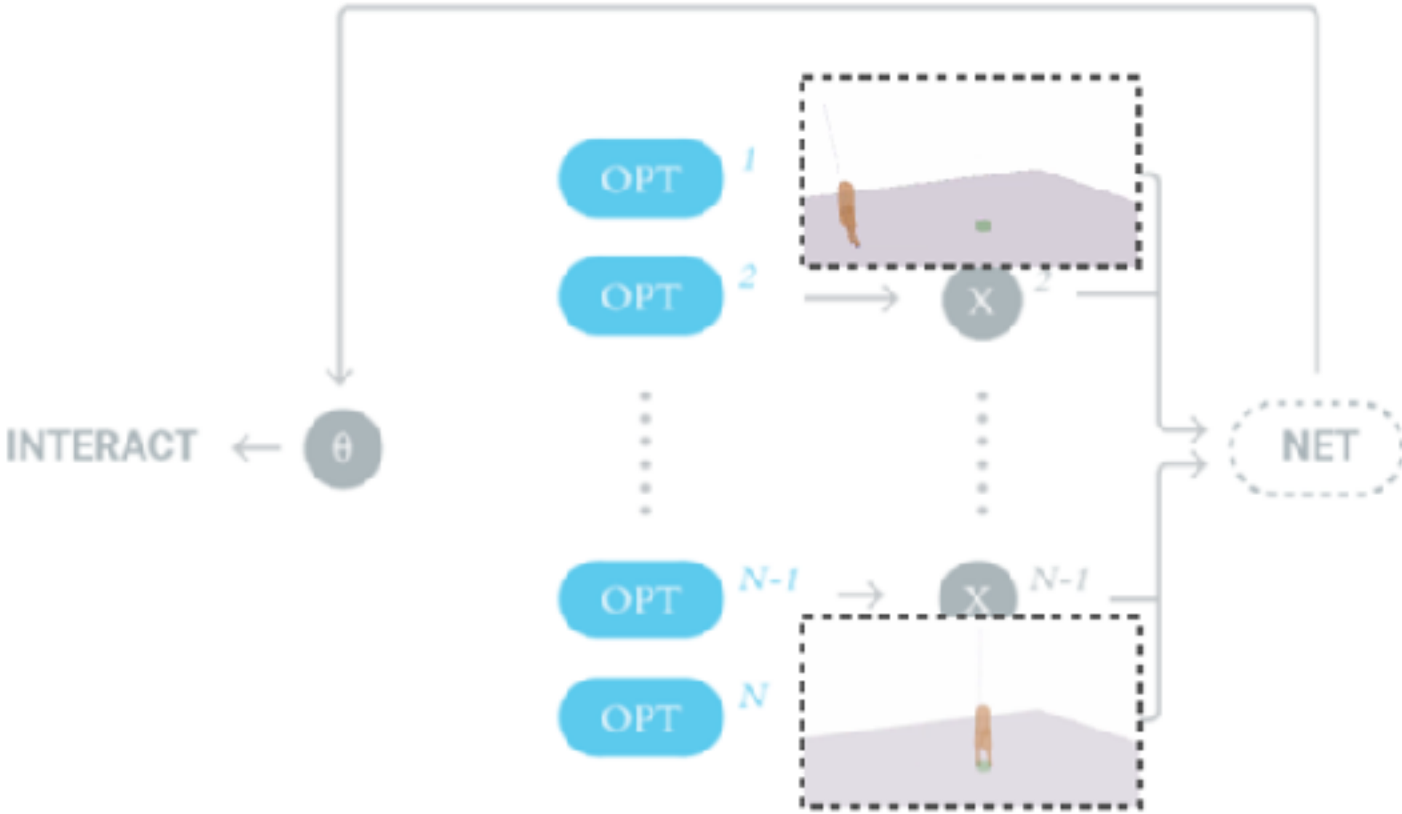
Learning Policies from Trajectory Optimization



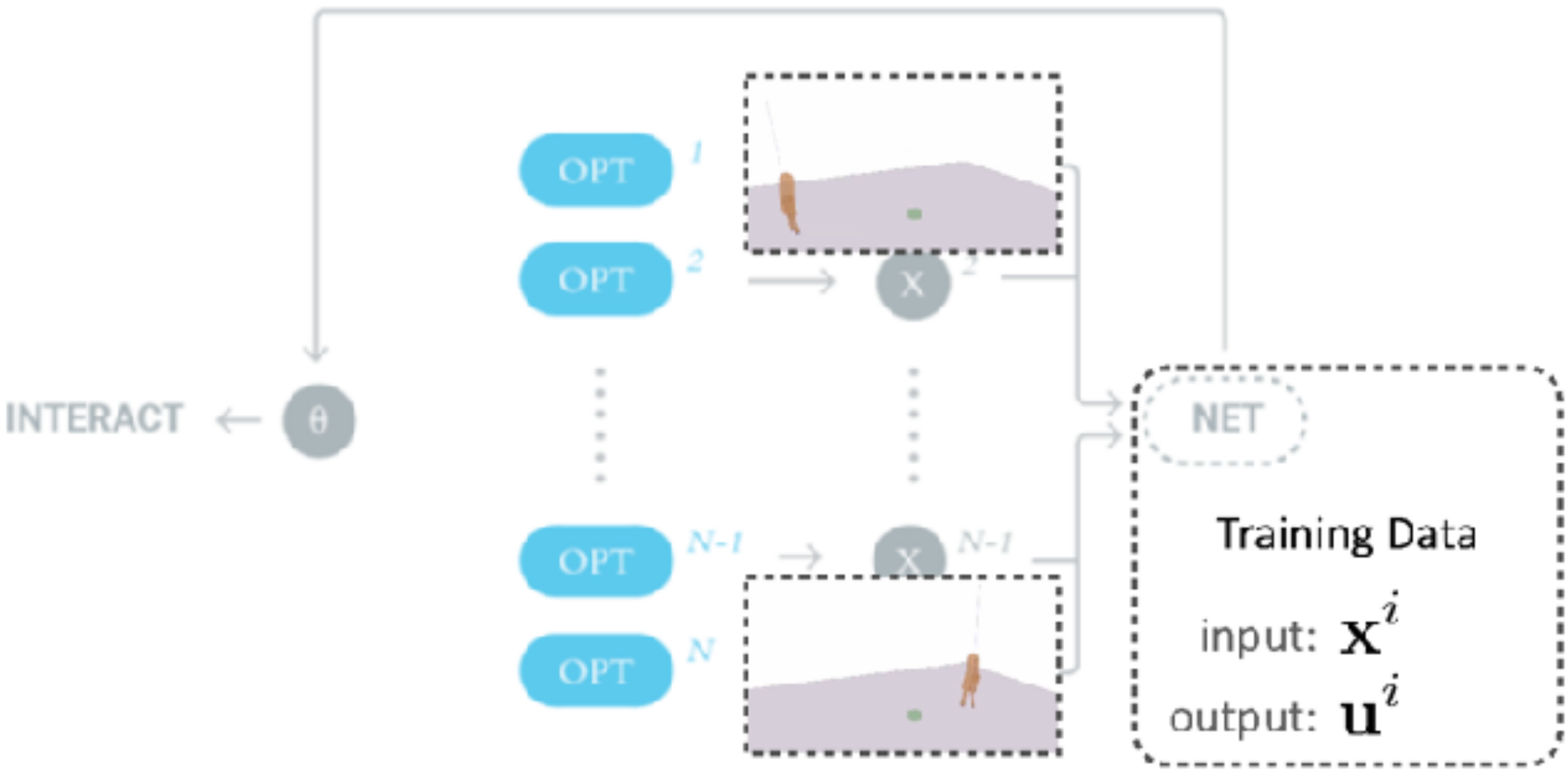
Learning Policies from Trajectory Optimization



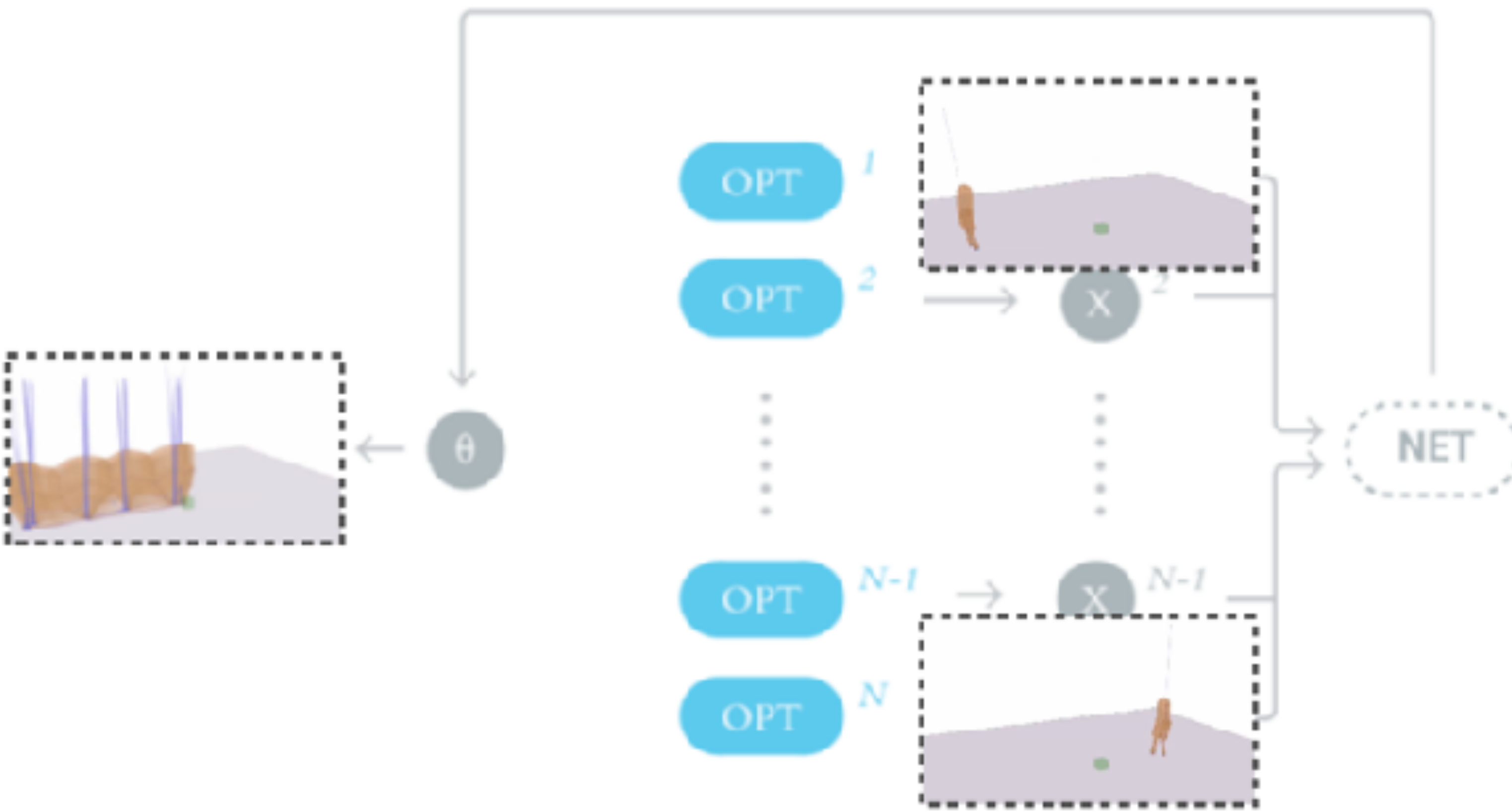
Learning Policies from Trajectory Optimization



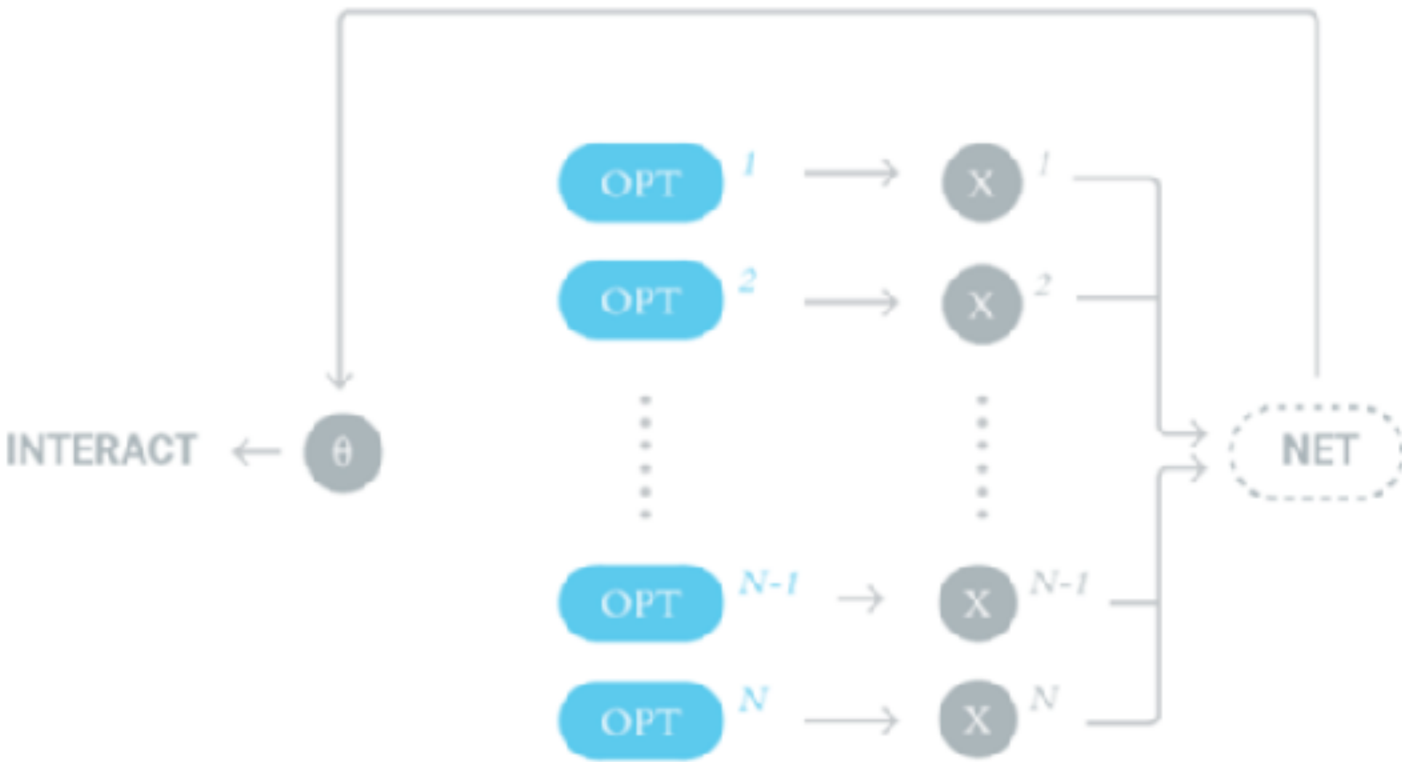
Learning Policies from Trajectory Optimization



Learning Policies from Trajectory Optimization

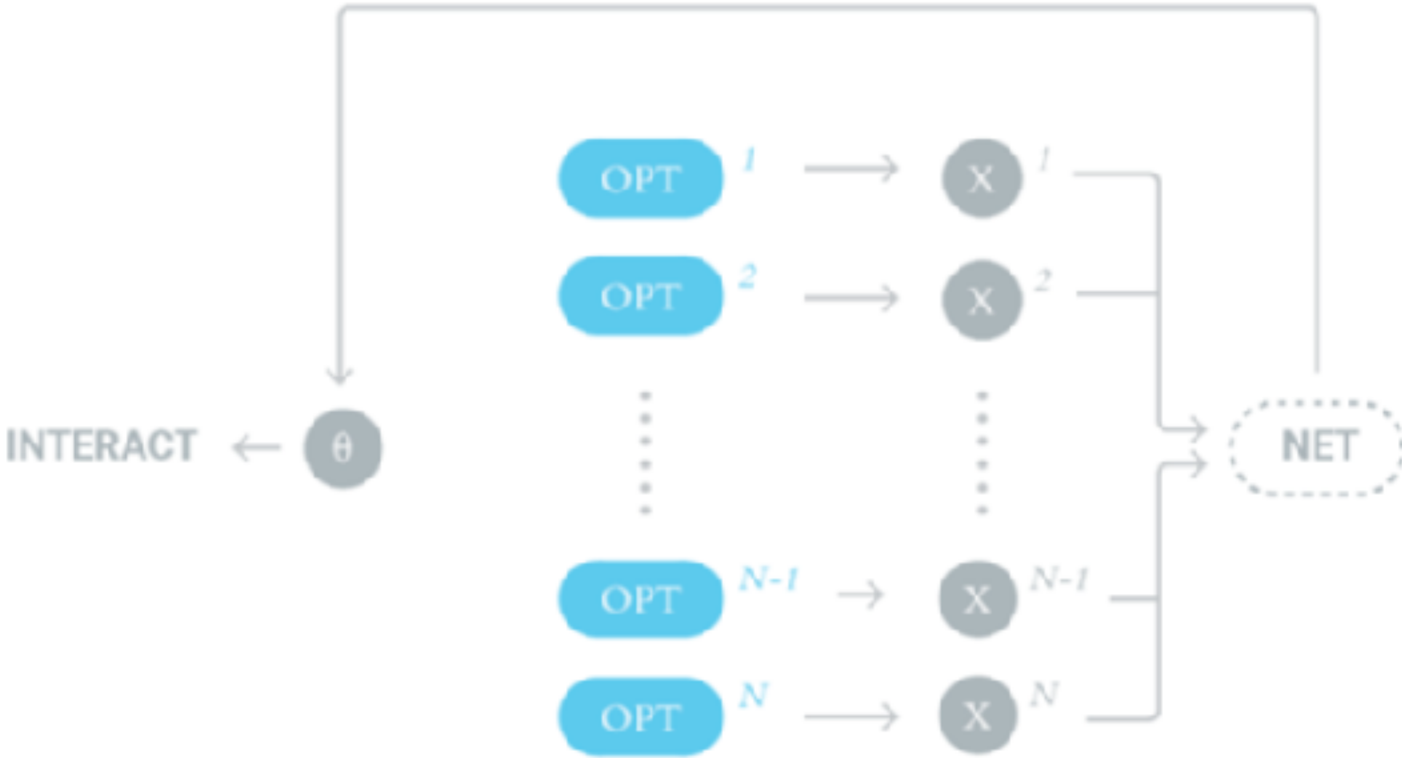


Learning Policies from Trajectory Optimization

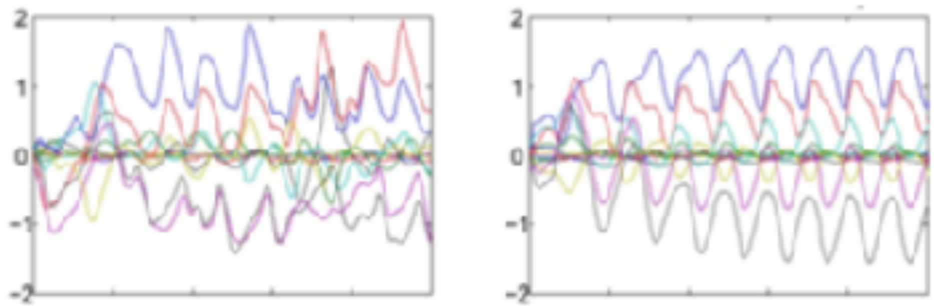


$\mathbf{X}^1, \ldots \mathbf{X}^N$ can be inconsistent or difficult to fit

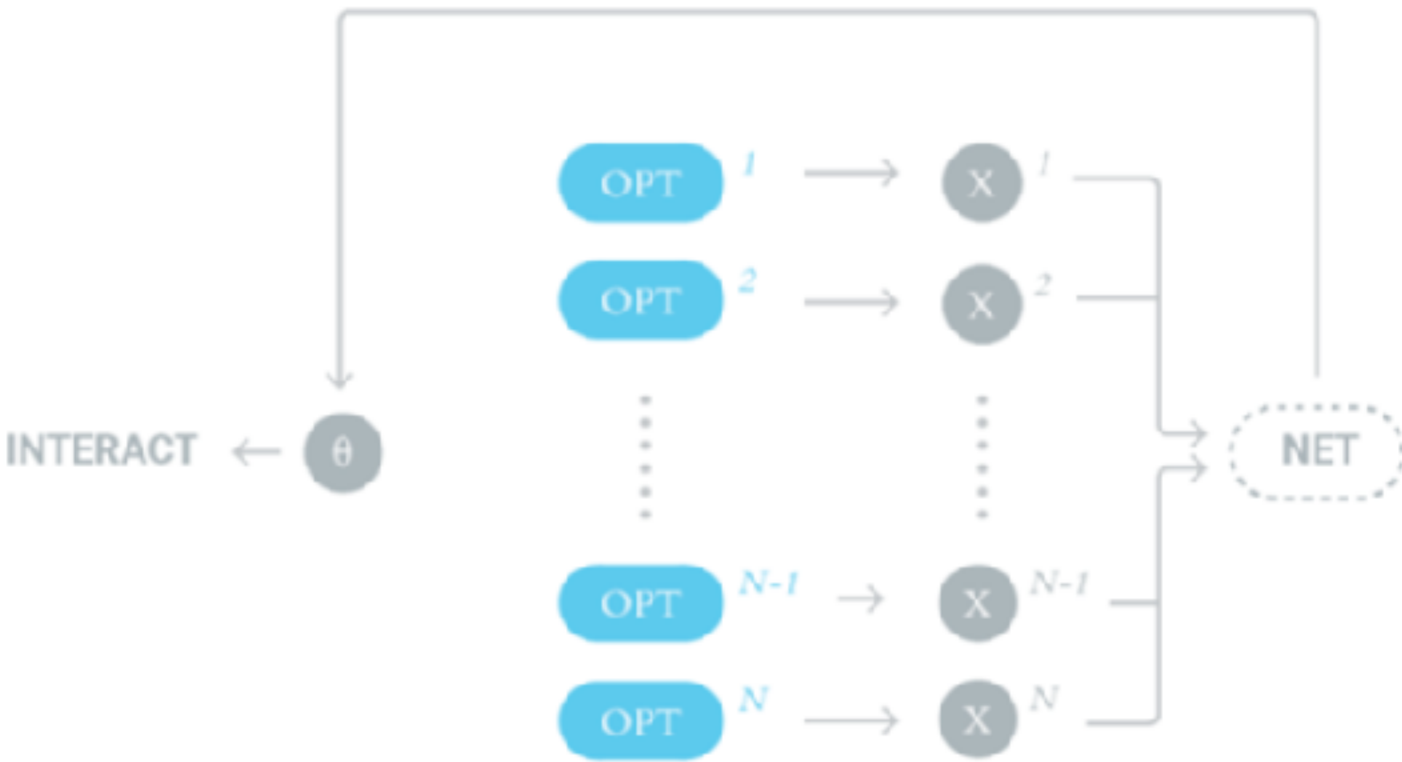
Learning Policies from Trajectory Optimization



$\mathbf{X}^1, \ldots \mathbf{X}^N$ can be inconsistent or difficult to fit

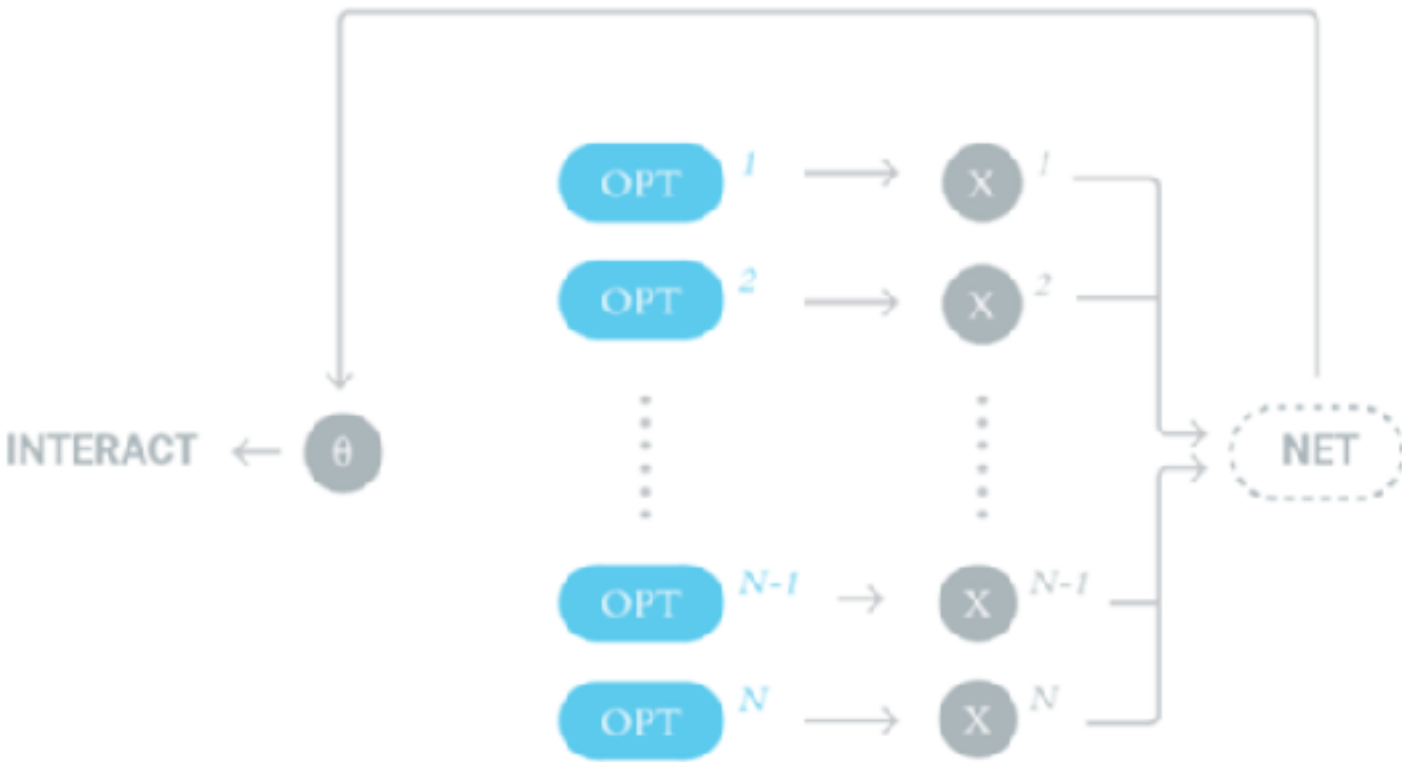


Learning Policies from Trajectory Optimization



$\mathbf{X}^1, \ldots \mathbf{X}^N$ can be inconsistent or difficult to fit

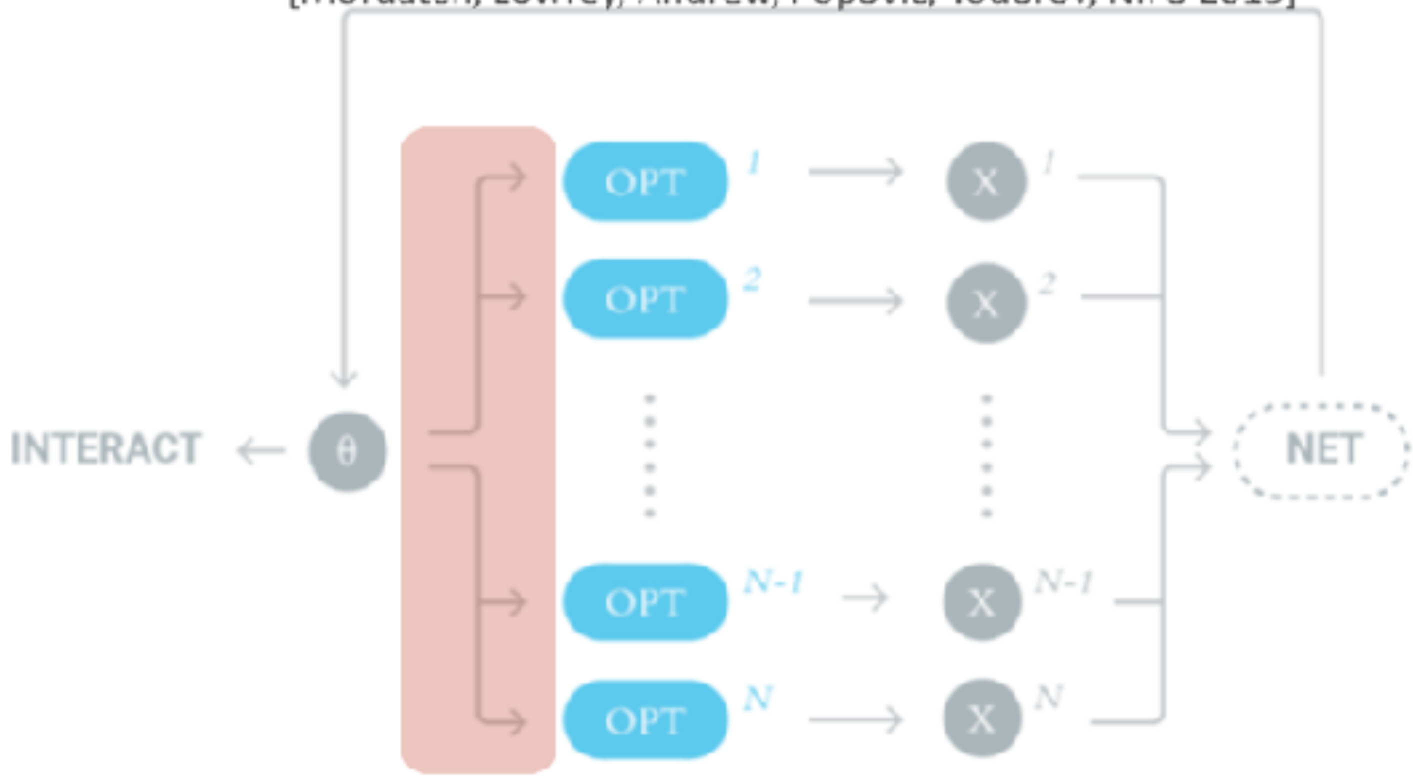
Learning Policies from Trajectory Optimization



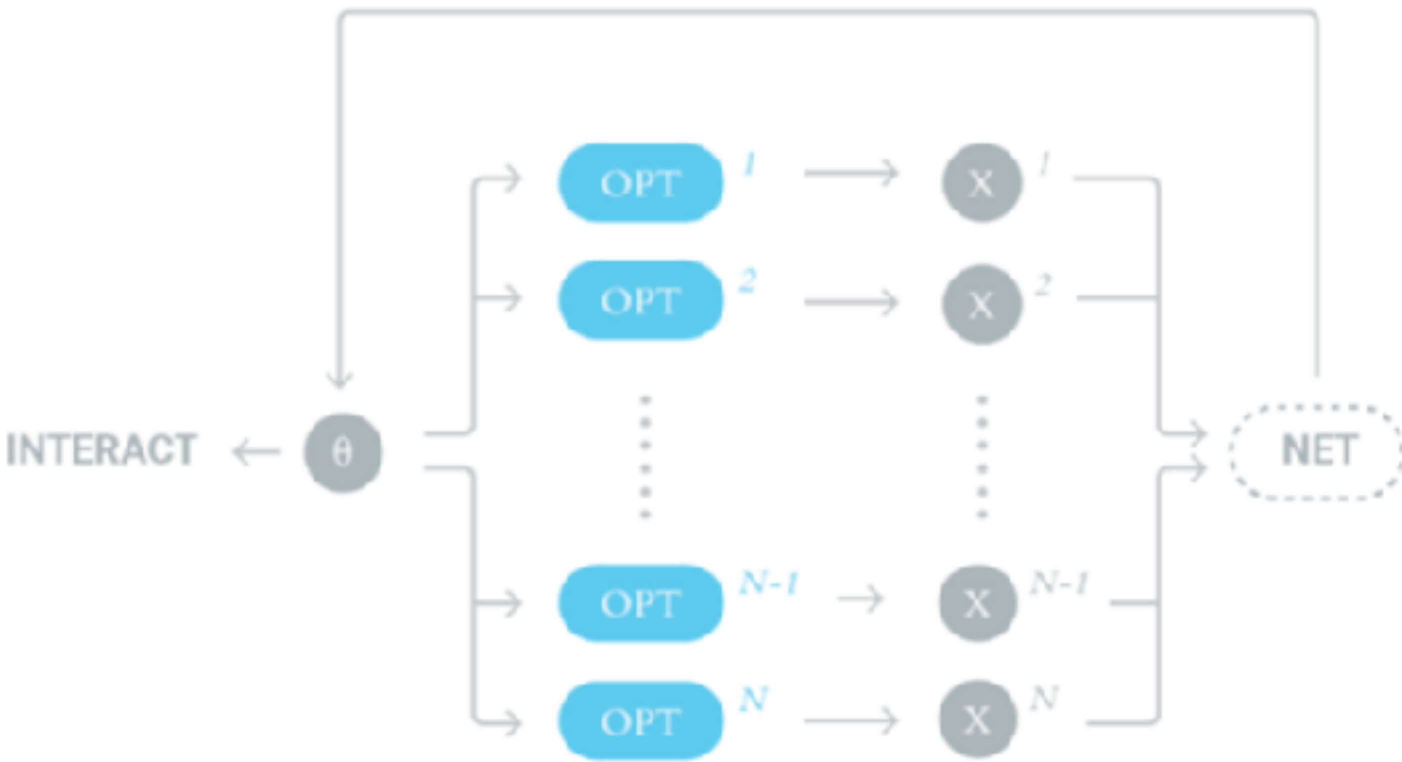
Joint Policy and Trajectory Optimization

[Mordatch, Todorov, RSS 2014]

[Mordatch, Lowrey, Andrew, Popovic, Todorov, NIPS 2015]

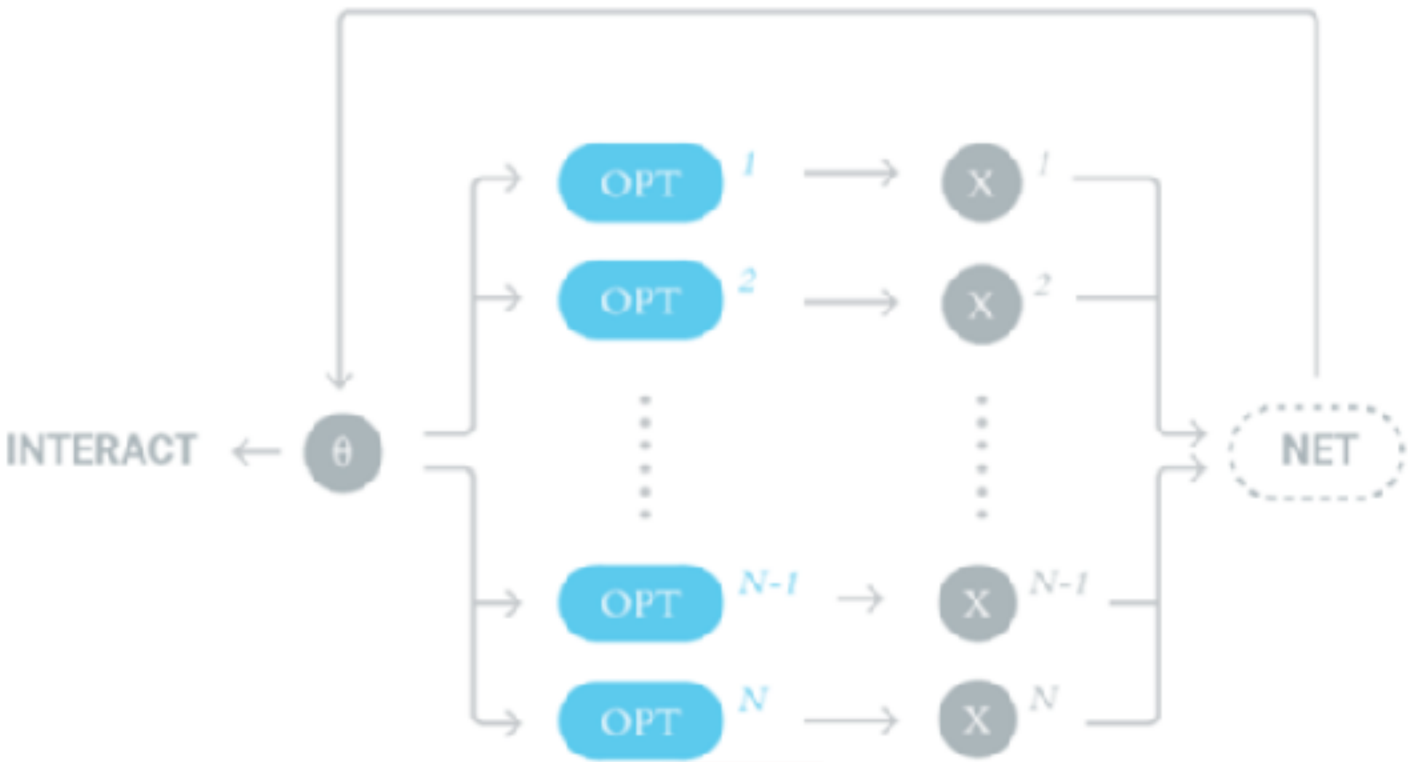


Joint Policy and Trajectory Optimization



$$\min_{\theta\ \mathbf{X}^1 \ldots \mathbf{X}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

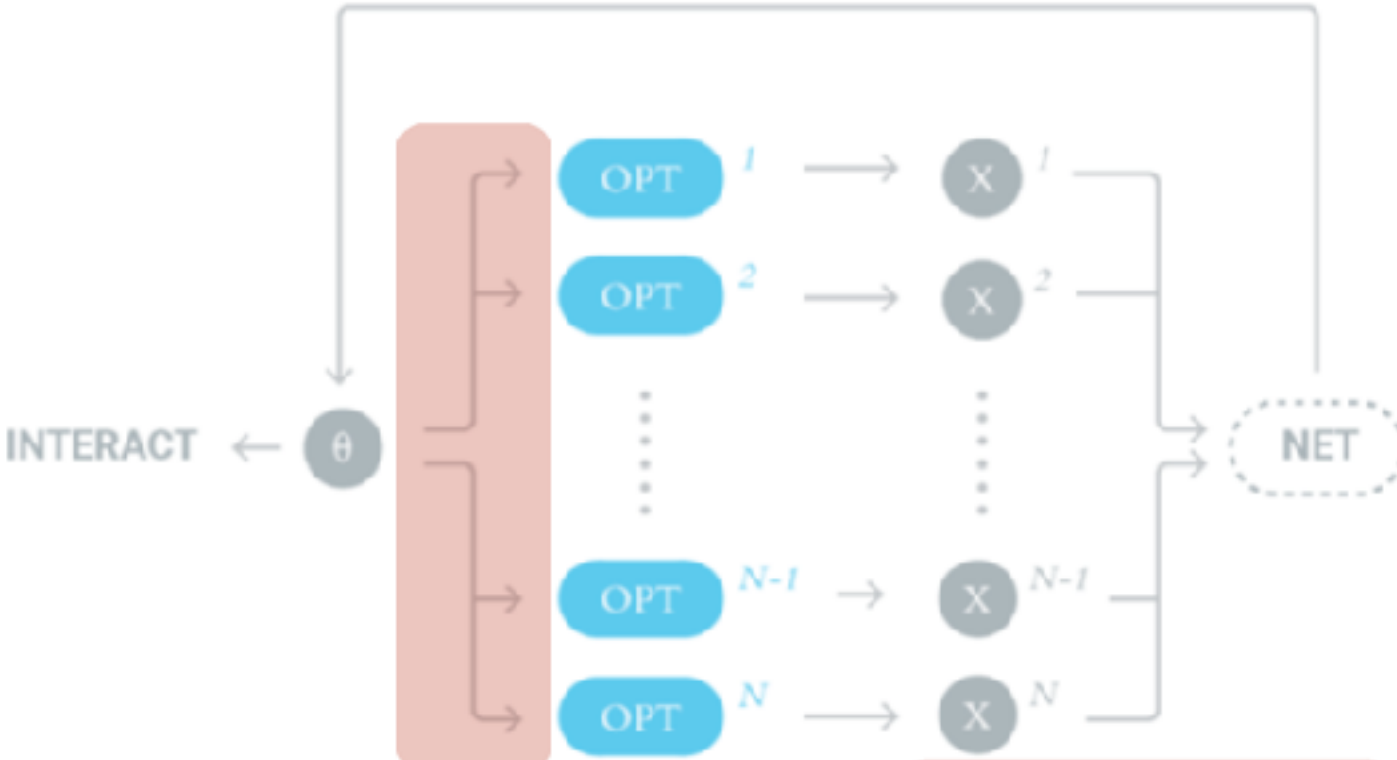
Joint Policy and Trajectory Optimization



$$\min_{\theta\ \mathbf{X}^1 \ldots \mathbf{X}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

traditional trajectory optimization

Joint Policy and Trajectory Optimization



$$\min_{\theta\, \mathbf{X}^1 \ldots \mathbf{X}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

Joint Policy and Trajectory Optimization

$$\min_{\theta\ \mathbf{x}^1 \ldots \mathbf{x}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

Joint Policy and Trajectory Optimization

$$\min_{\theta, \mathbf{x}^1 \dots \mathbf{x}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

Joint Policy and Trajectory Optimization

- Adjust policy
- Adjust trajectory

$$\min_{\theta,\, \mathbf{x}^1 \ldots \mathbf{x}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

Idea

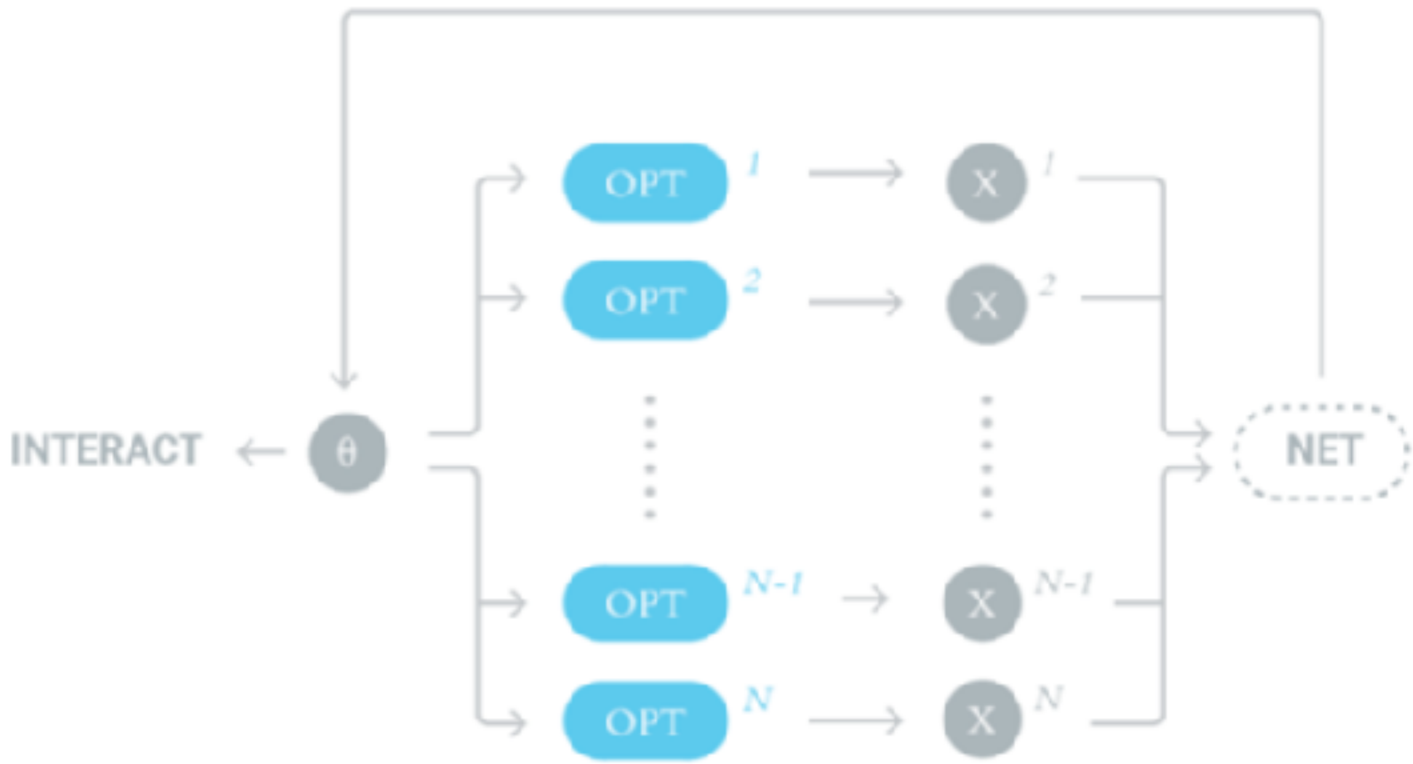
Add auxiliary variables

Softly enforce consistency between variables

Search in larger, but easier to explore space

$$\min_{\theta\ \mathbf{X}^1 \dots \mathbf{X}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\boldsymbol{\pi}_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

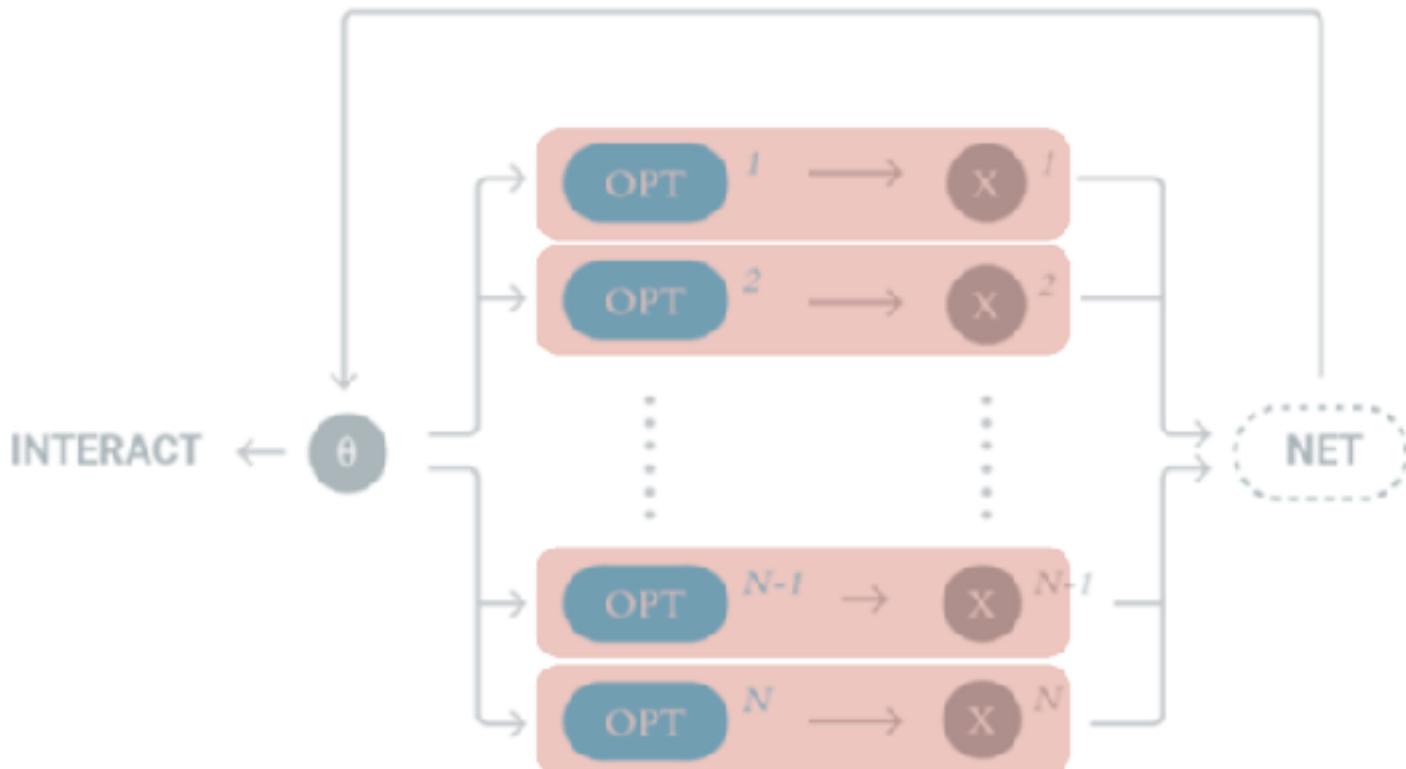
Alternating Optimization



Decompose into:

- trajectory optimizations
- regression

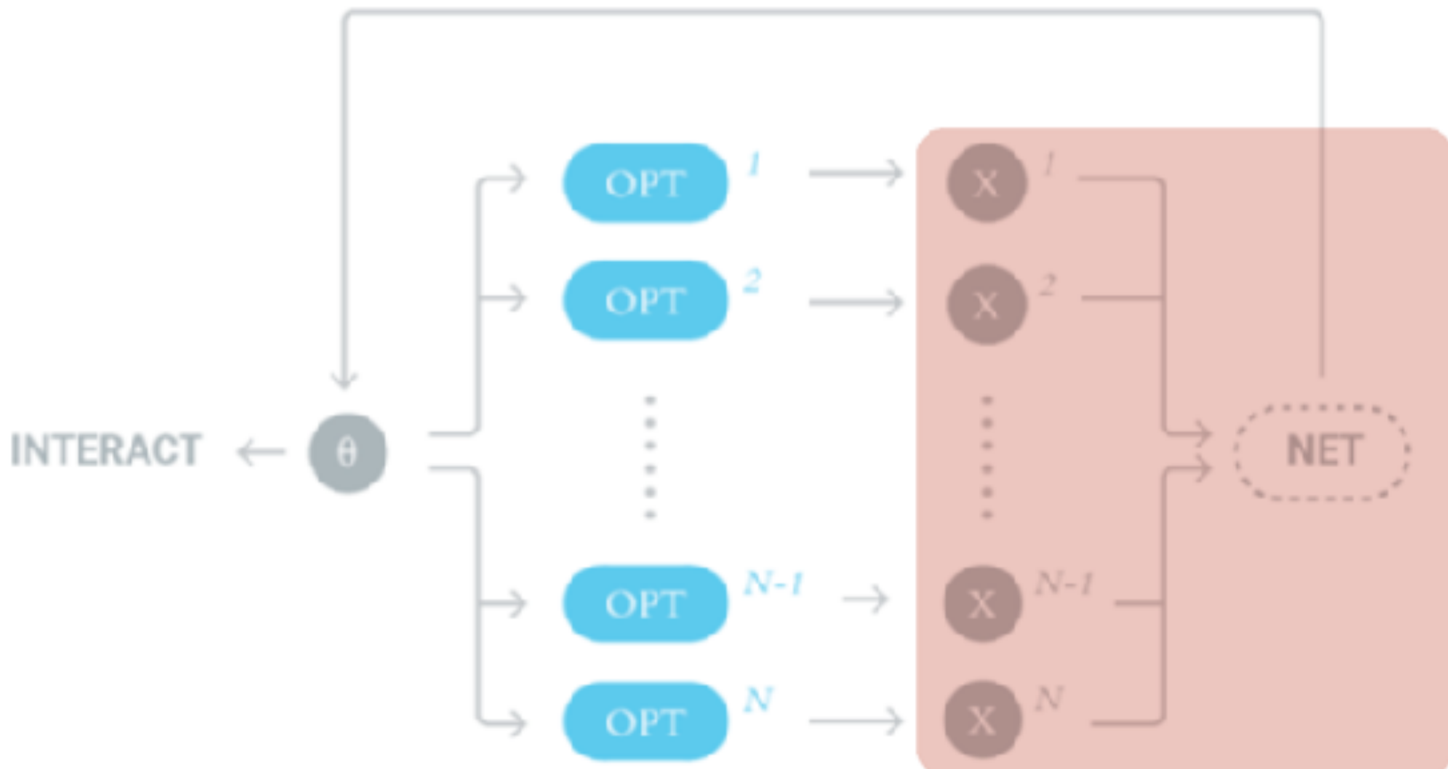
Alternating Optimization



Decompose into:

- trajectory optimizations $\min_{\mathbf{X}} \sum_t C(\mathbf{x}^t) + ||\pi_\theta(\mathbf{x}^t) - \mathbf{u}^t||^2$ "stay close to policy"
- regression

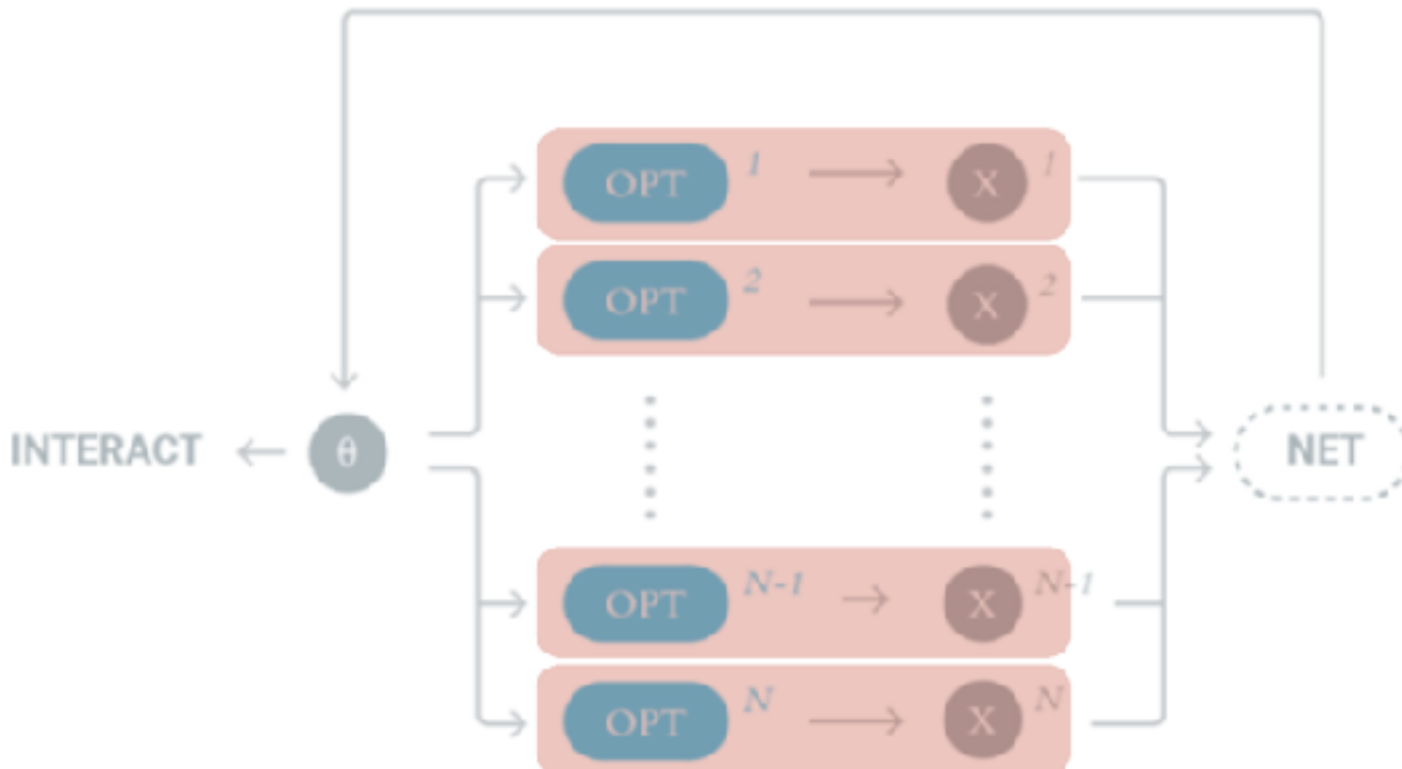
Alternating Optimization



Decompose into:

- trajectory optimizations
- regression $\min_{\theta} \sum_{i,t} ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$

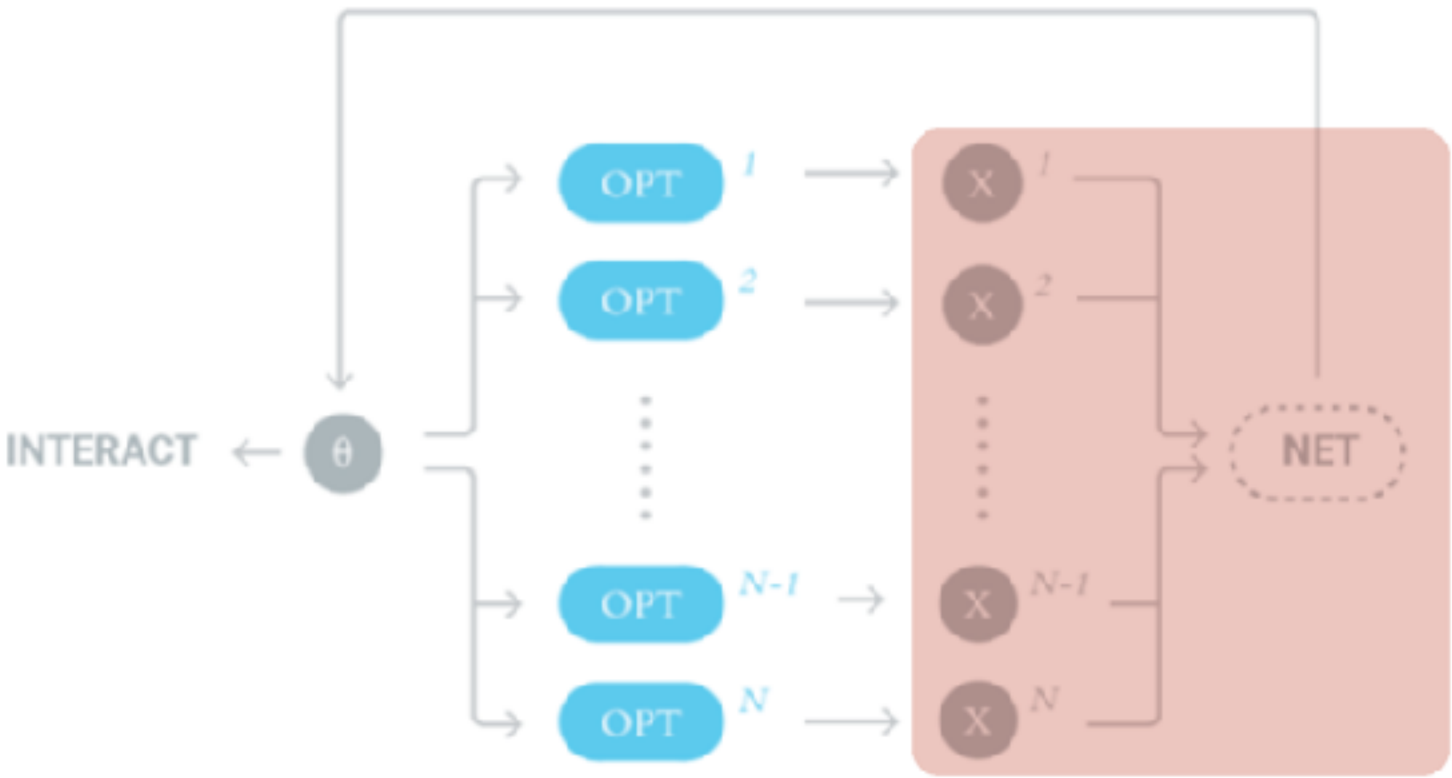
Alternating Optimization



Decompose into:

- trajectory optimizations $\min_{\mathbf{X}} \sum_t C(\mathbf{x}^t) + ||\boldsymbol{\pi}_\theta(\mathbf{x}^t) - \mathbf{u}^t||^2$
- regression

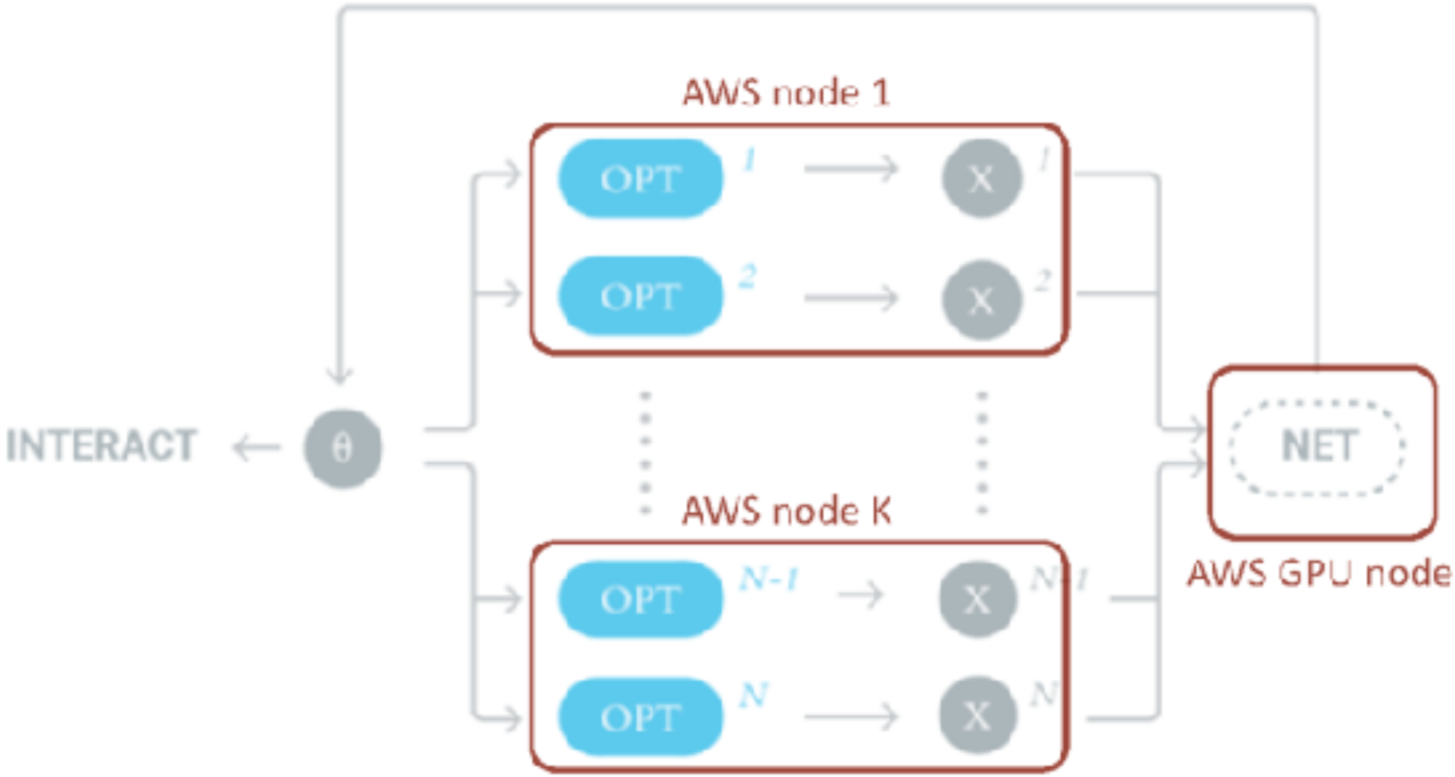
Alternating Optimization



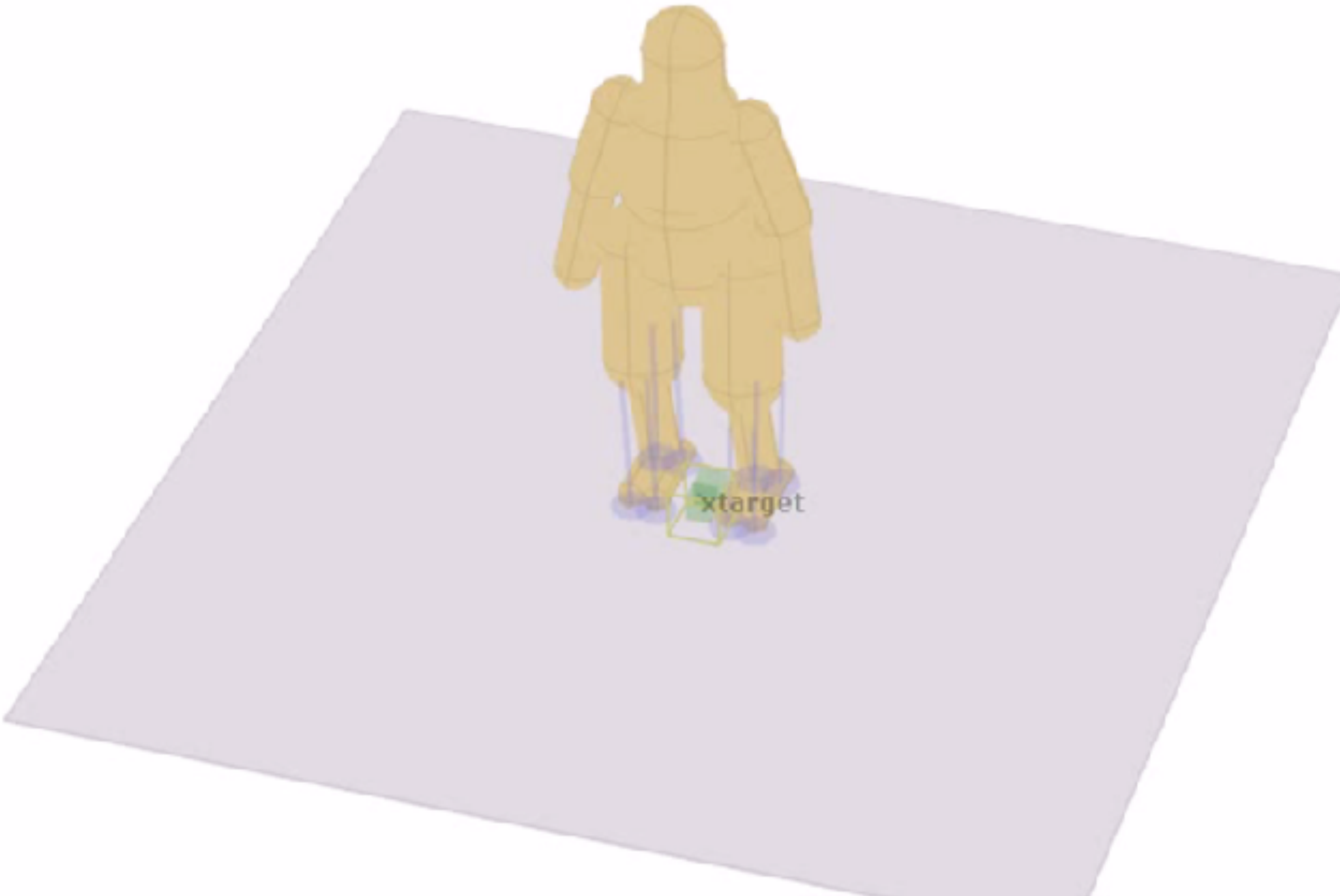
Decompose into:

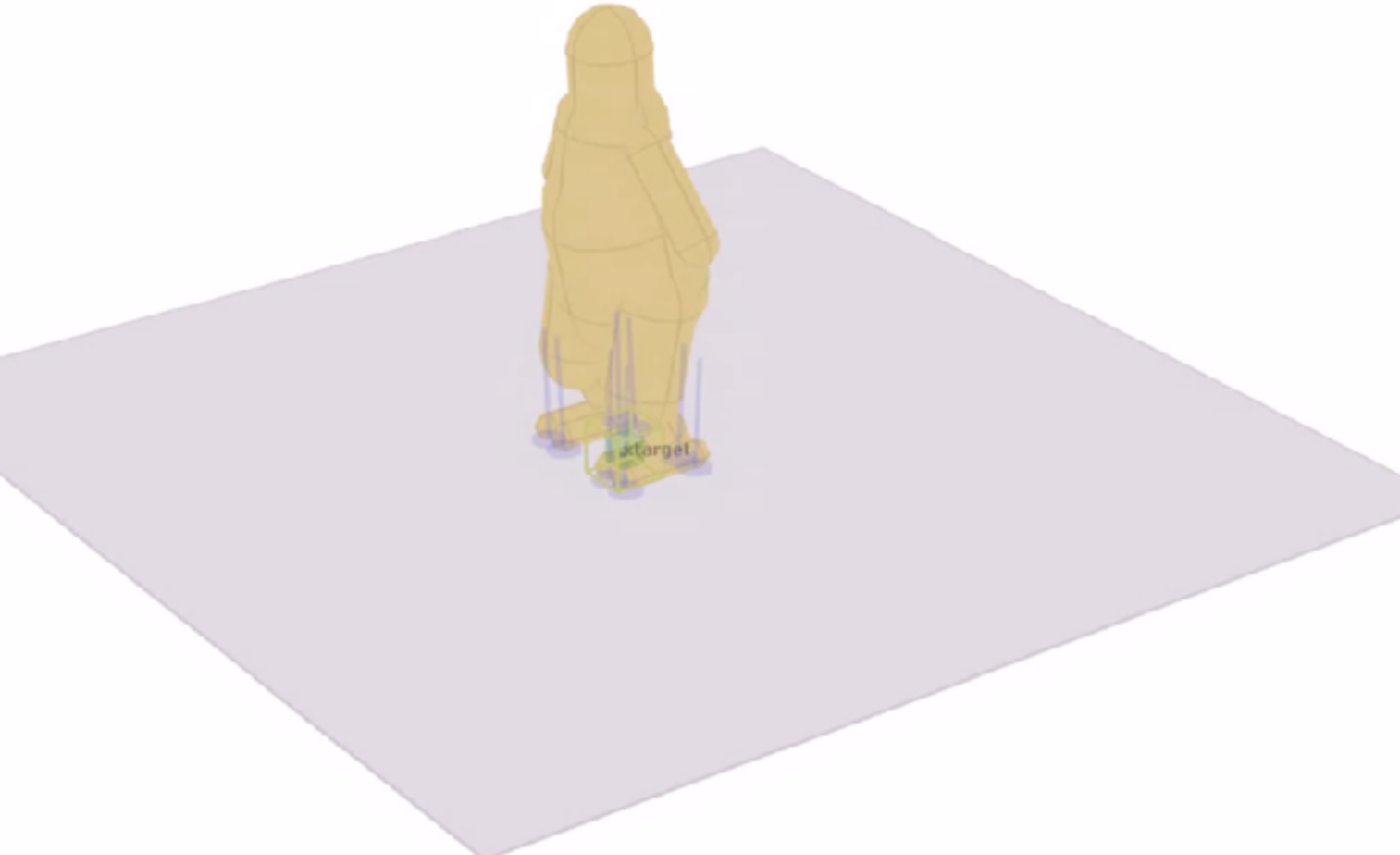
- trajectory optimizations
- regression $\min_{\theta} \sum_{i,t} ||\pi_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$

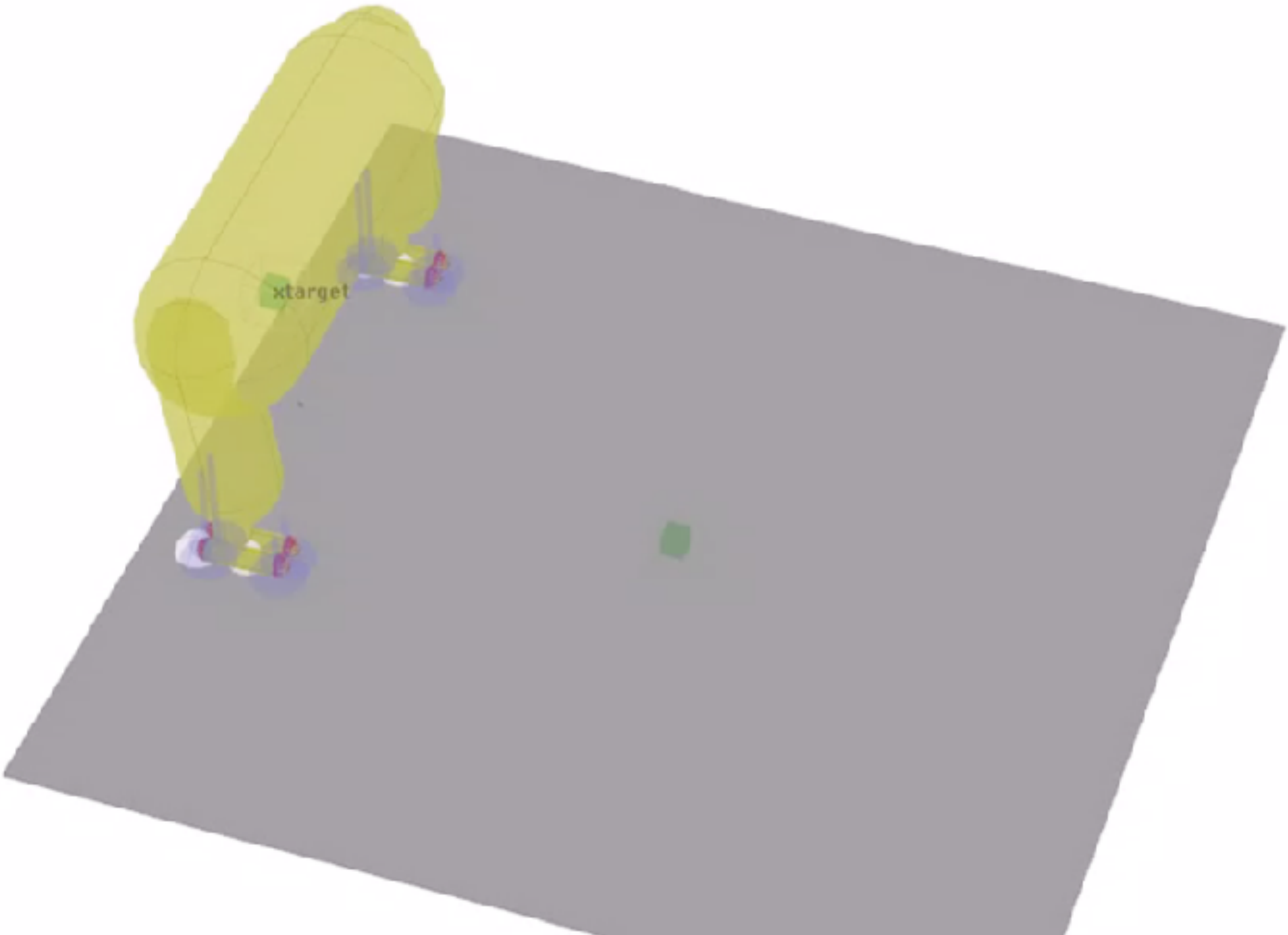
Scalable Implementation



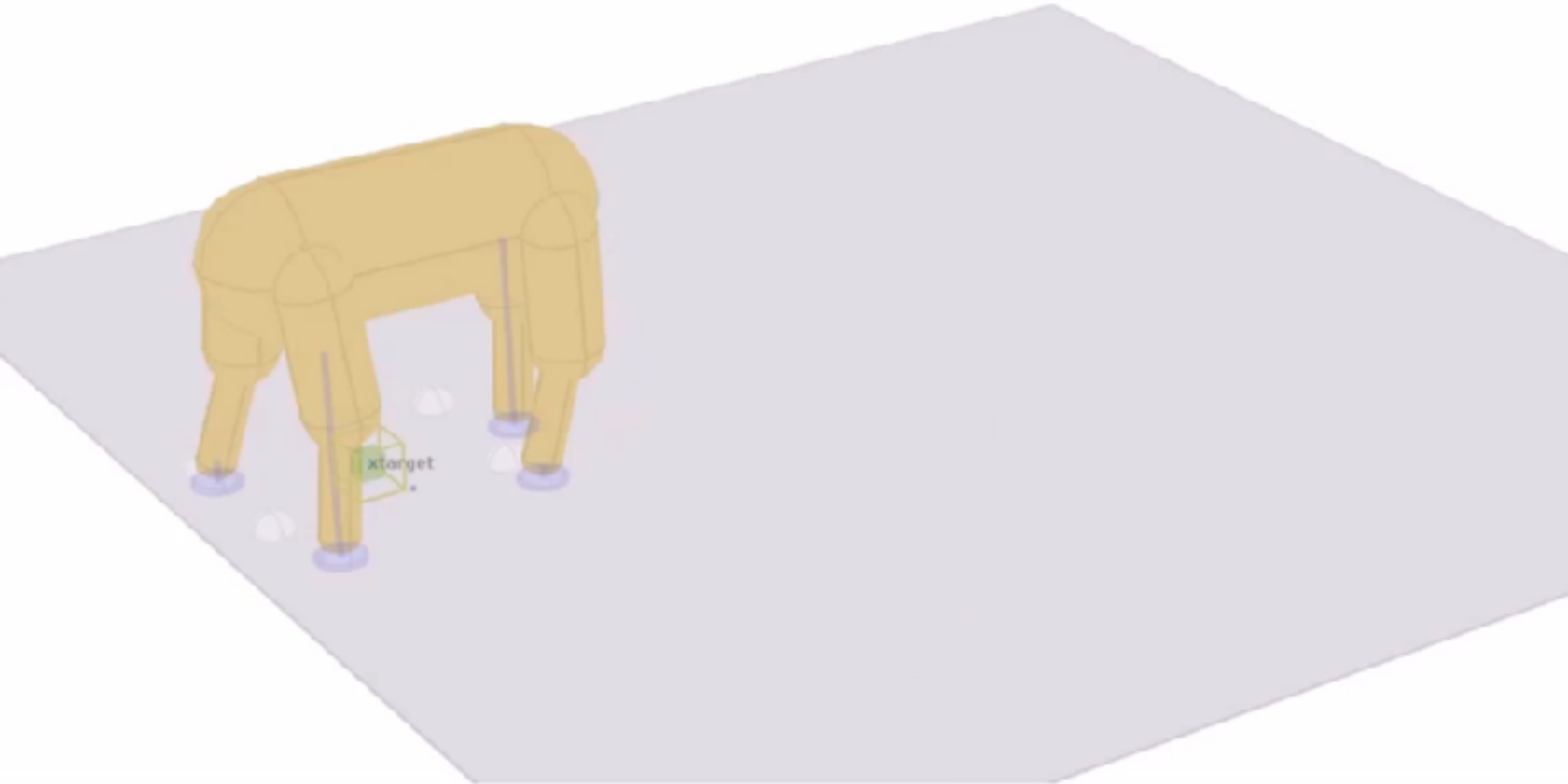
- asynchronous updates
- SGD network training
- Full dataset never loaded in memory

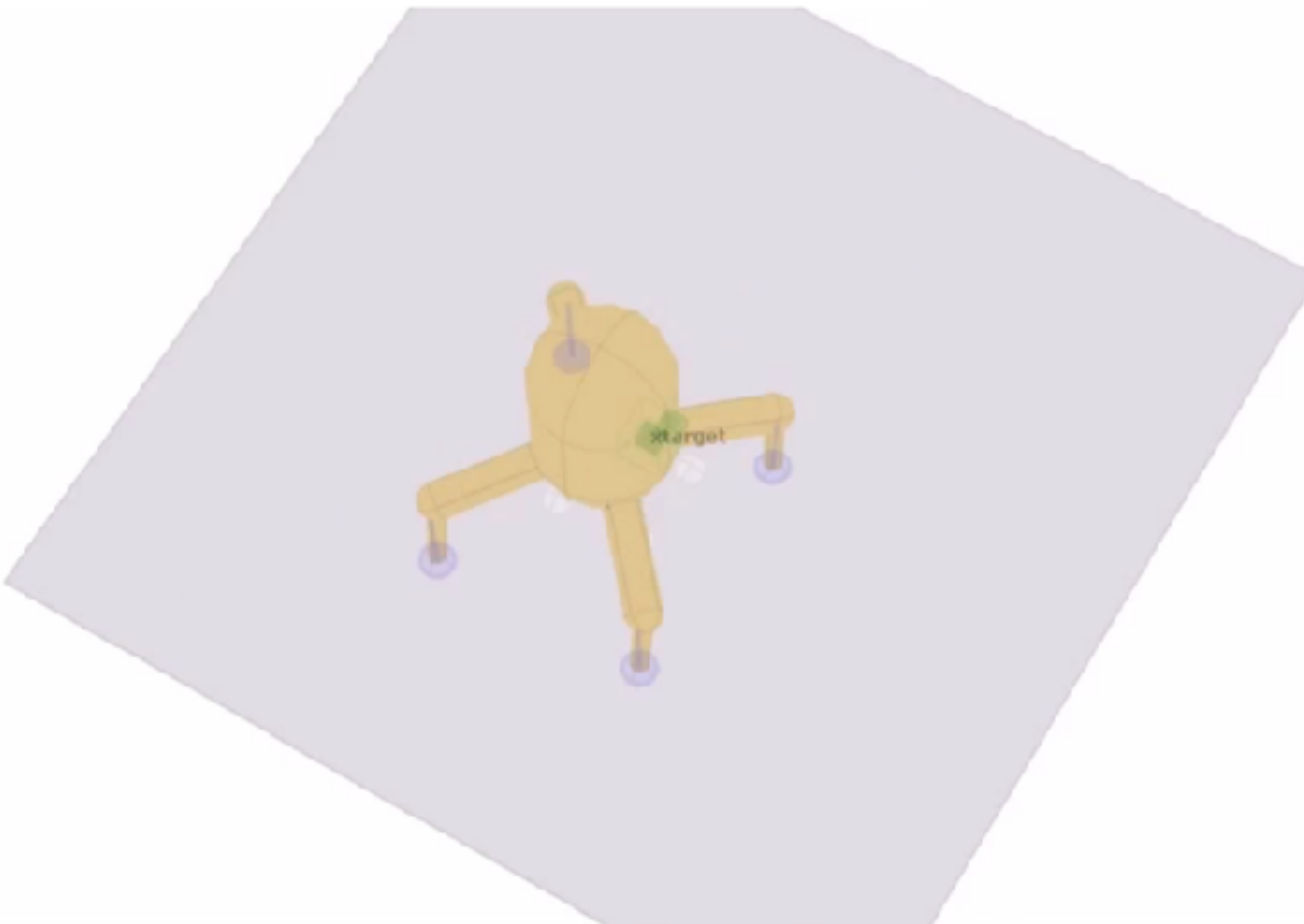

xtarget


xtarget


xtarget


target


xtarget


target

Future State Prediction