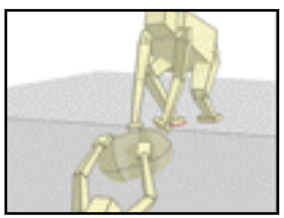
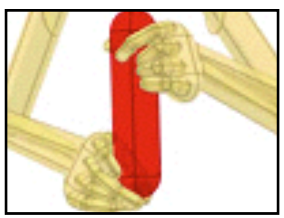
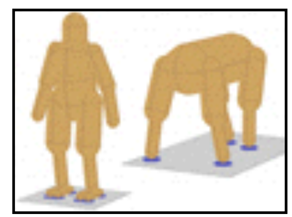
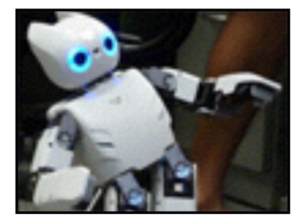
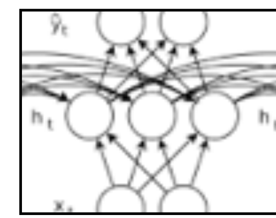
Direct Collocation Methods for Trajectory Optimization and Policy Learning

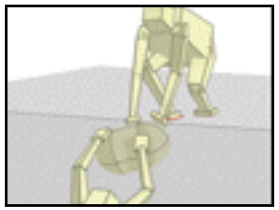
CS 294-112: Deep Reinforcement Learning

Guest Lecture

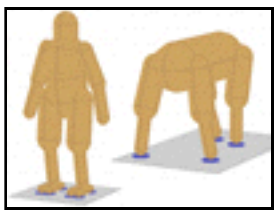
Igor Mordatch, OpenAI

Overview

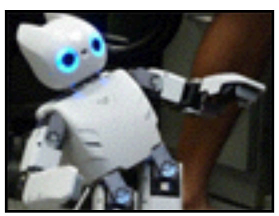
- **Last Time:** Learning dynamics models, optimal control, and policy learning
  - Focused on using forward dynamics models and shooting methods (LQR, DDP)
- **Today:**
  - Look at *inverse* dynamics models, direct collocation in detail (for optimal control and policy learning)
  - How to optimize complex movement with contacts
  - Dealing with unknown and uncertain dynamics
  - Applications to biomechanics



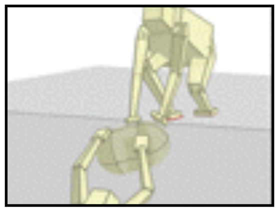
Trajectory Optimization with Direct Collocation



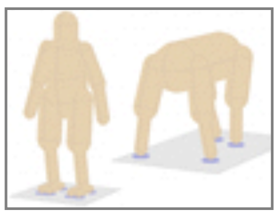
Learning Control Policies with Direct Collocation



Unknown/Uncertain Dynamics and Applications



Trajectory Optimization with Direct Collocation



Learning Control Policies with Direct Collocation

Unknown/Uncertain Dynamics and Applications

Trajectory Optimization

Trajectory Optimization

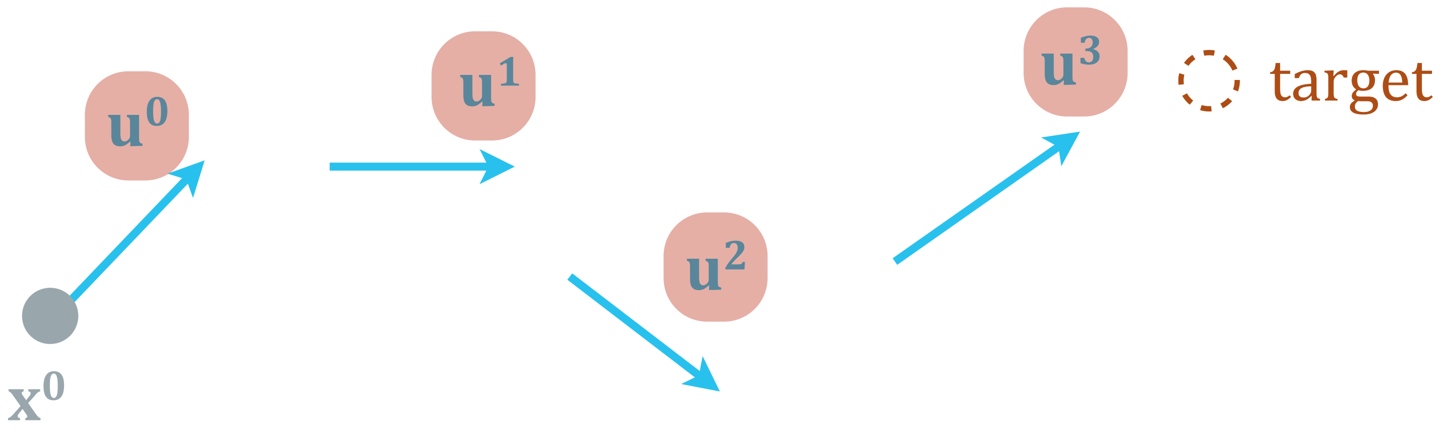
Forward Shooting:

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

Trajectory Optimization

Forward Shooting:

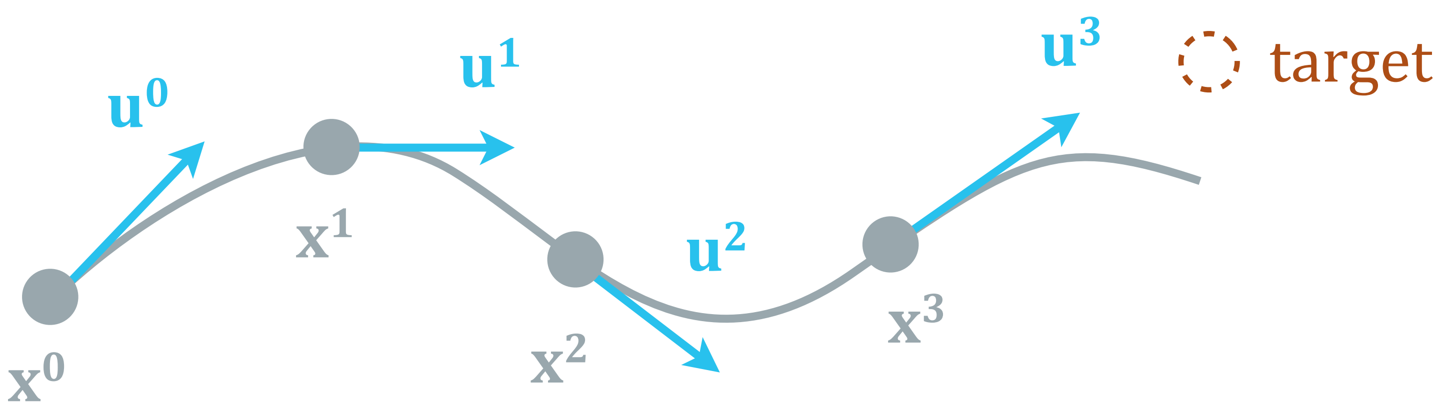
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Trajectory Optimization

Forward Shooting:

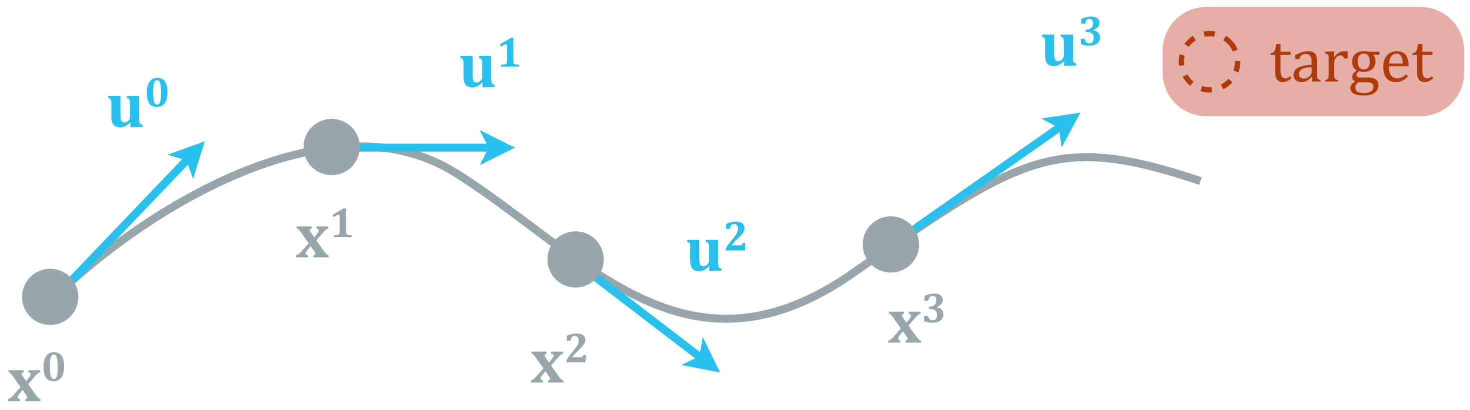
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \qquad \boxed{\mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)}$$



Trajectory Optimization

Forward Shooting:

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



Poor Conditioning

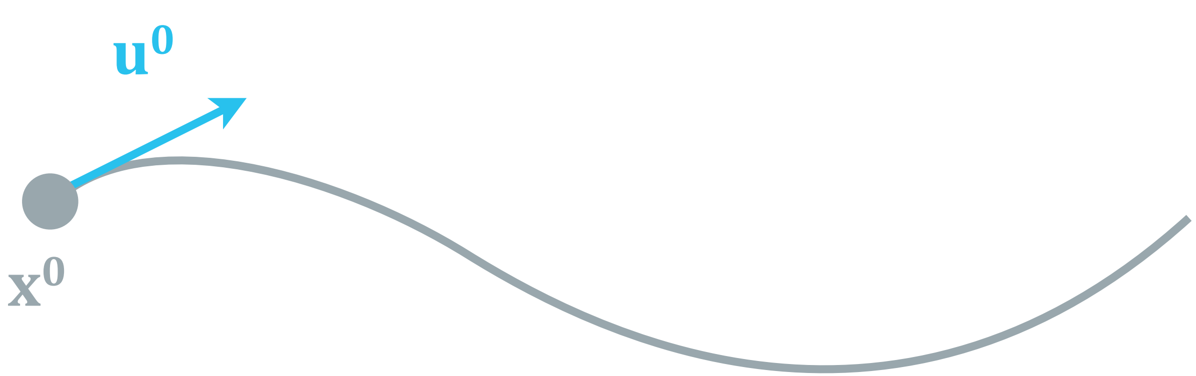
Forward Shooting:

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

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:

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

$$\Updownarrow$$

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

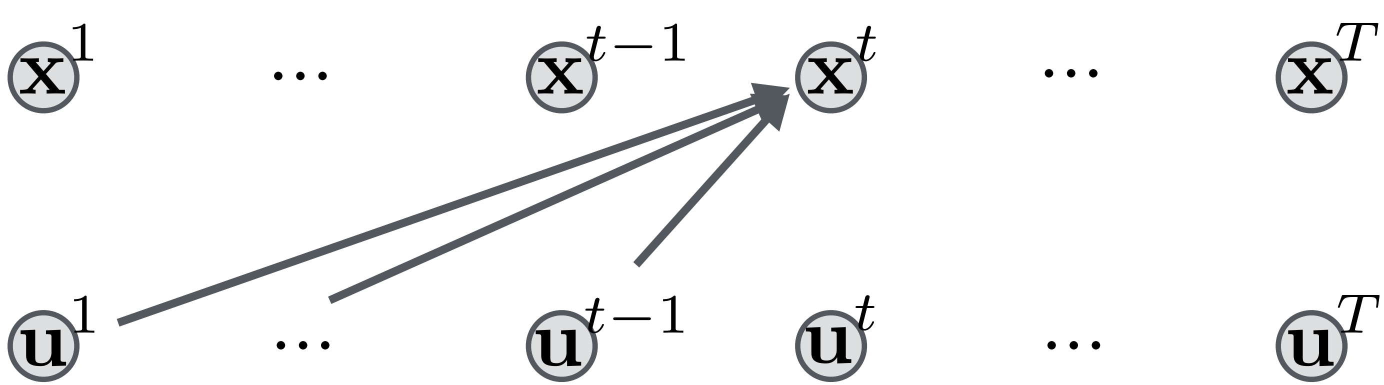
Poor Conditioning

Forward Shooting:

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

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

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



Narrow Feasible Region

Forward Shooting:

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

Narrow Feasible Region

Forward Shooting:

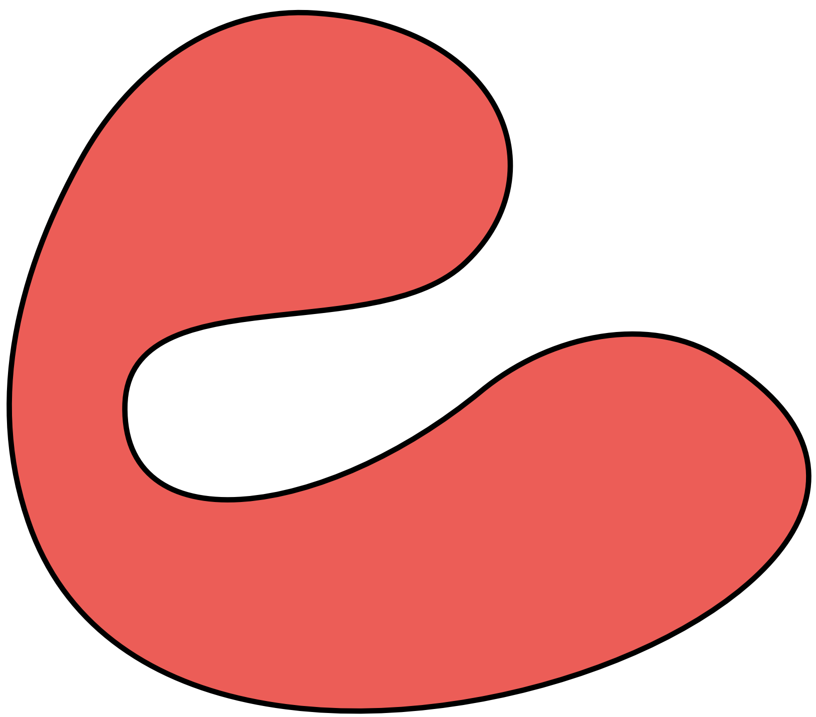
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \boxed{\mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)}$$

implicit hard constraint

Narrow Feasible Region

Forward Shooting:

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

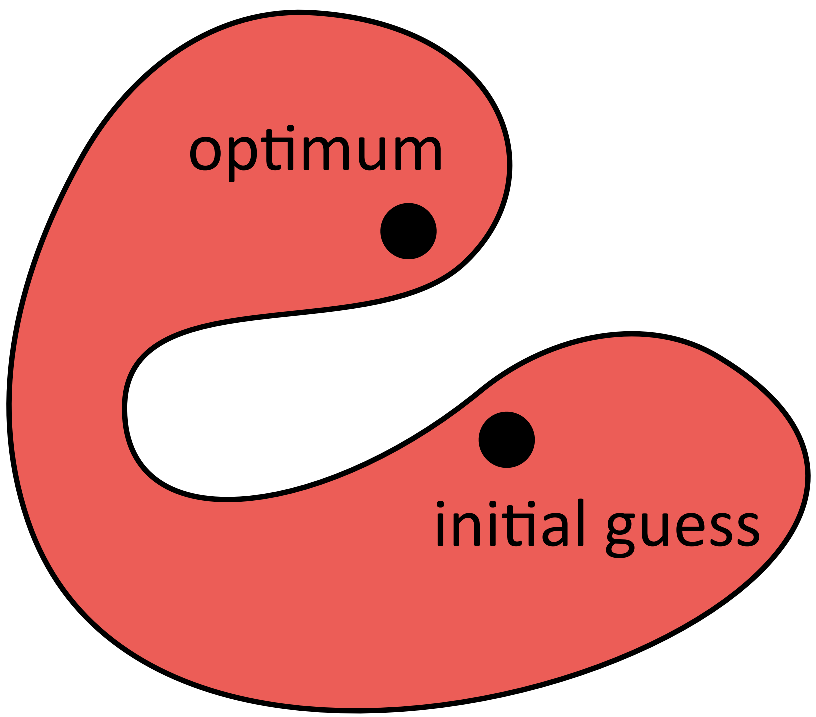


feasible region

Narrow Feasible Region

Forward Shooting:

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

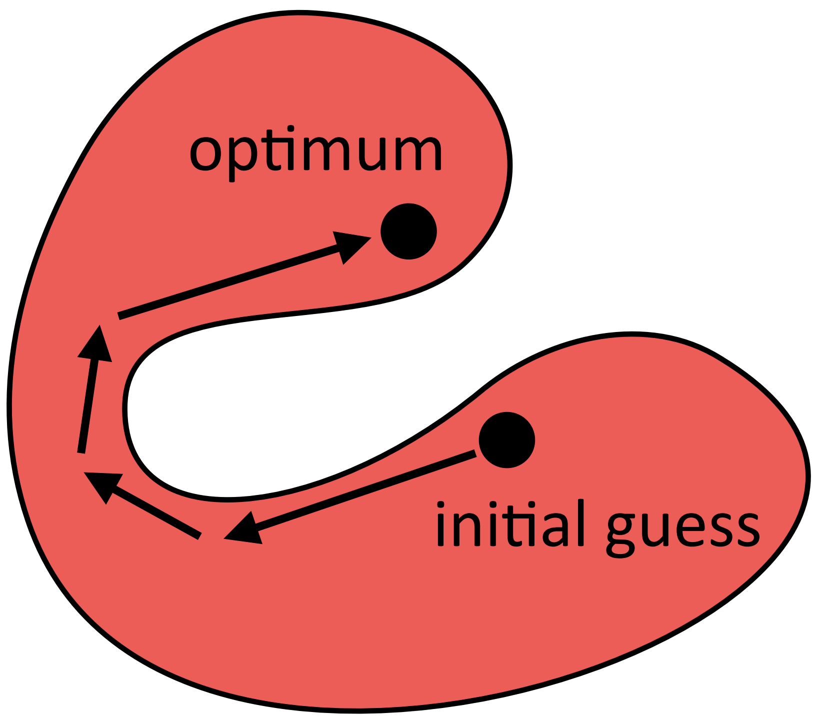


feasible region

Narrow Feasible Region

Forward Shooting:

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



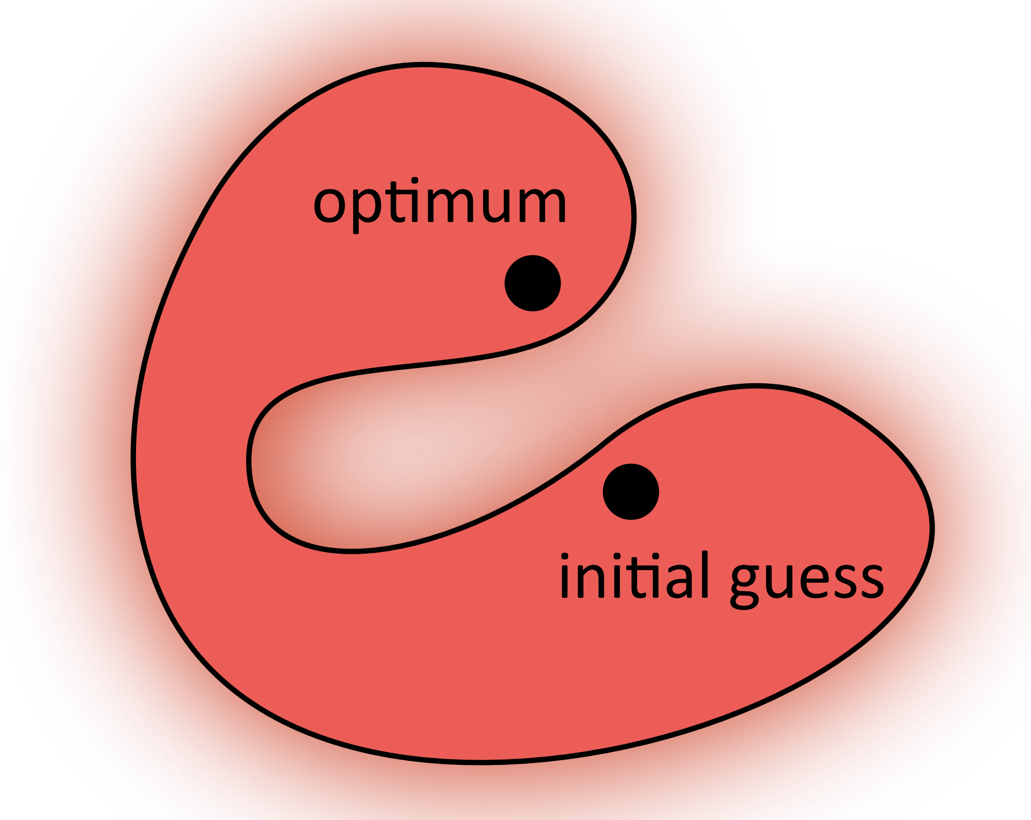
feasible region

Narrow Feasible Region

Forward Shooting:

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

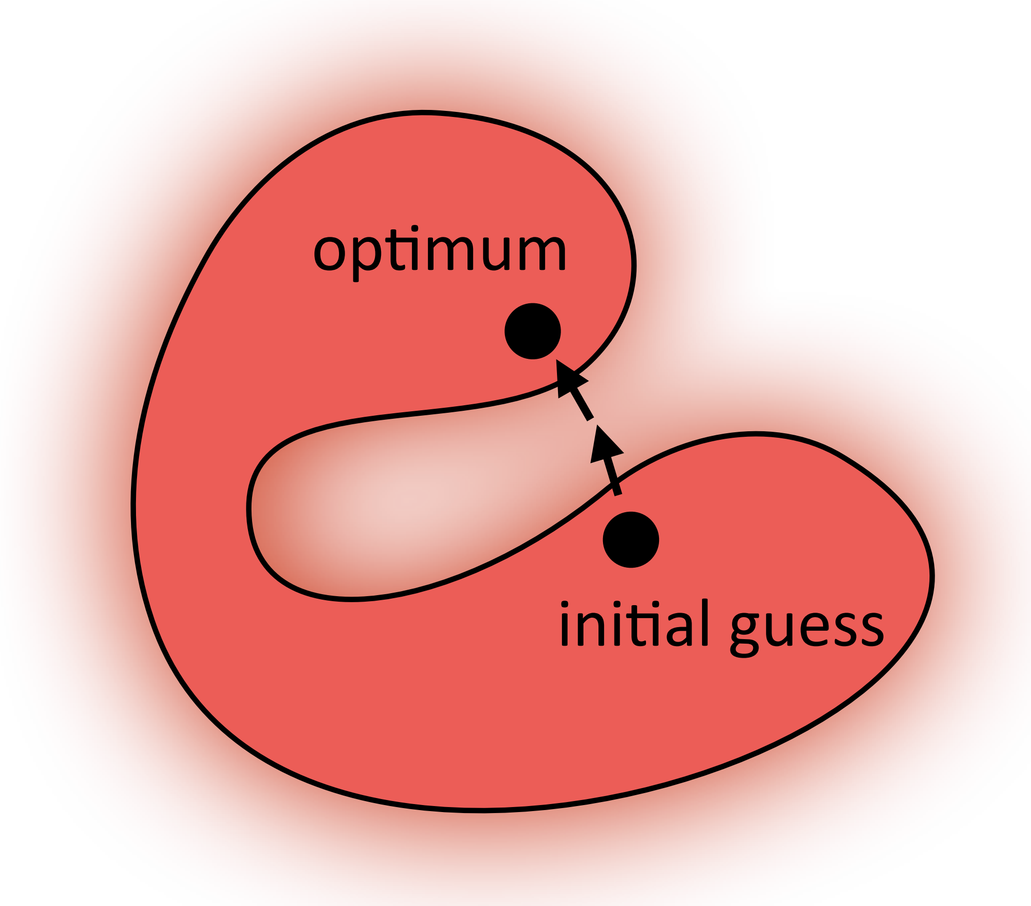
soft constraint



Narrow Feasible Region

Forward Shooting:

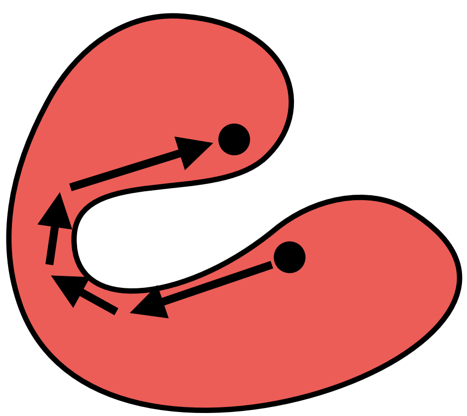
$$\min_{\mathbf{u}^0 \ldots \mathbf{u}^T} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \mathbf{u}^t)$$



Narrow Feasible Region

Forward Shooting:

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



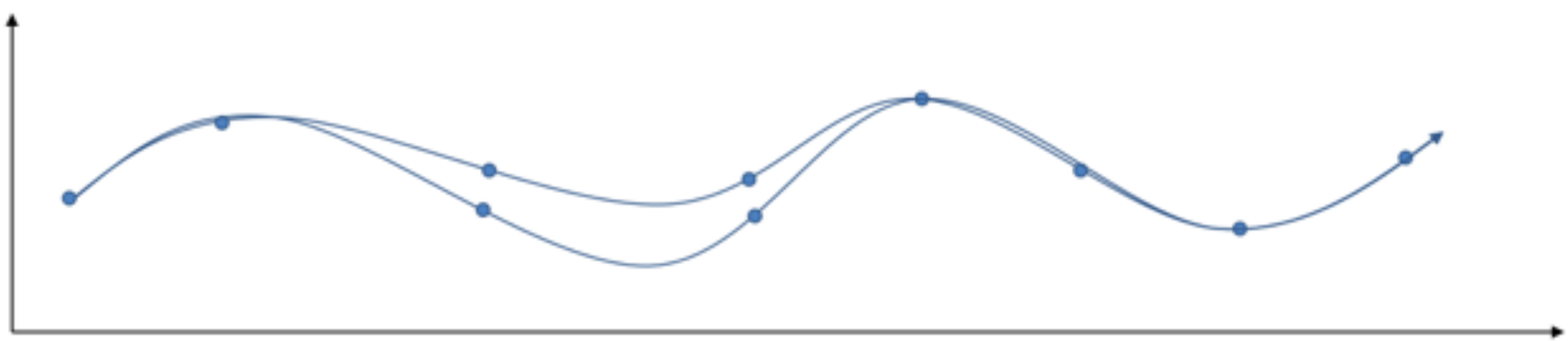
- Comes up as an issue in practice
  - collisions, falling down, etc…
- Prone to falling into local minima
- Makes solution sensitive to initial guess
- Initial guess from demonstrations and randomization helps

From Last Lecture:

Shooting methods vs collocation

collocation method: optimize over actions and states, with constraints

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\mathbf{x}_1,\ldots,\mathbf{x}_T} \sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t) \text{ s.t. } \mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$$

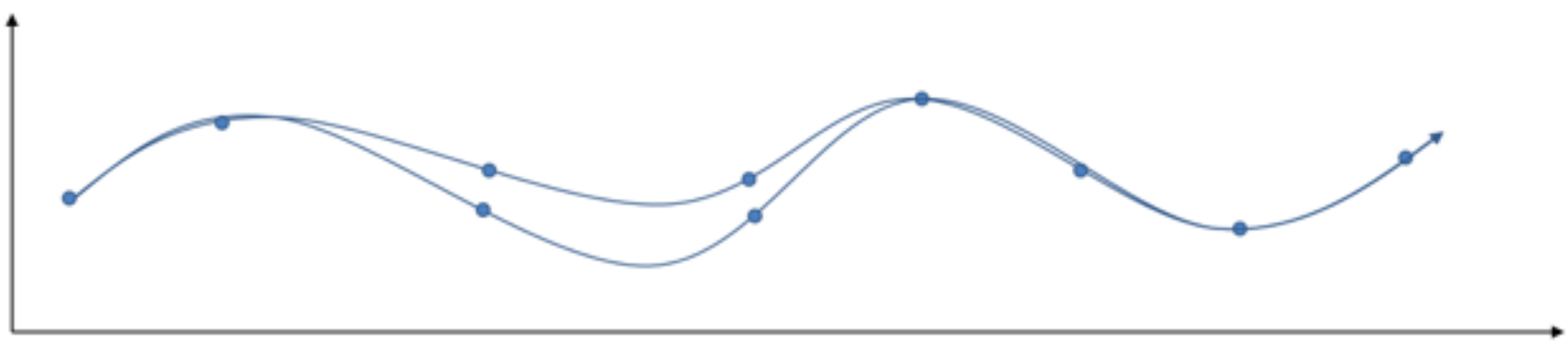


From Last Lecture:

Shooting methods vs collocation

collocation method: optimize over actions and states, with constraints

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\mathbf{x}_1,\ldots,\mathbf{x}_T} \sum_{t=1}^{T} c(\mathbf{x}_t,\mathbf{u}_t) \text{ s.t. } \mathbf{x}_t = f(\mathbf{x}_{t-1},\mathbf{u}_{t-1})$$

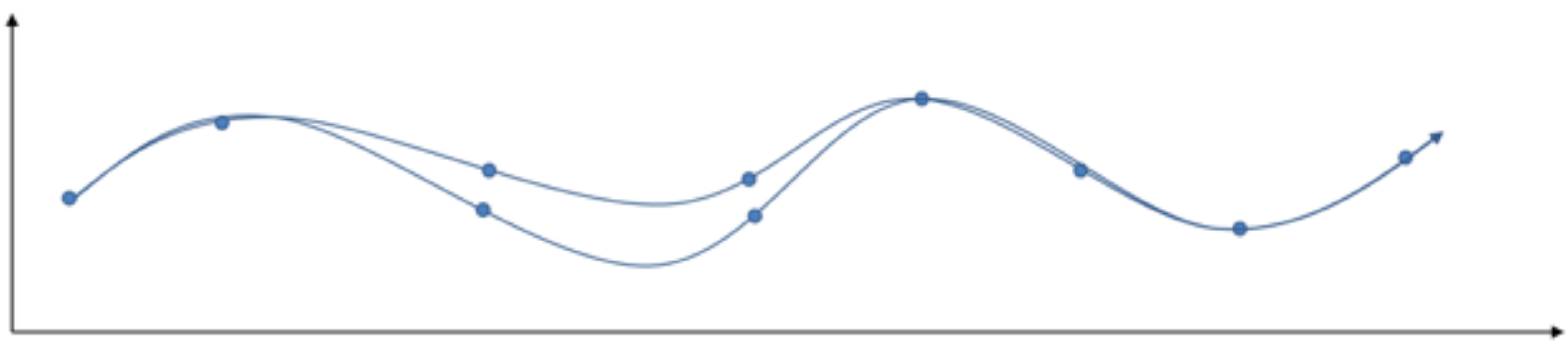


From Last Lecture:

Shooting methods vs collocation

collocation method: optimize over actions and states, with constraints

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\mathbf{x}_1,\ldots,\mathbf{x}_T} \sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t) \text{ s.t. } \mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$$



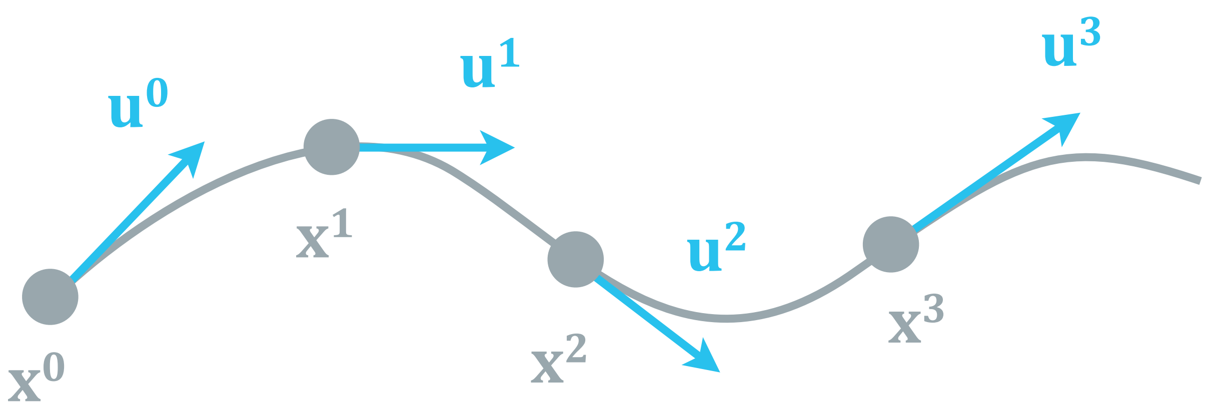
Direct Collocation

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:

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



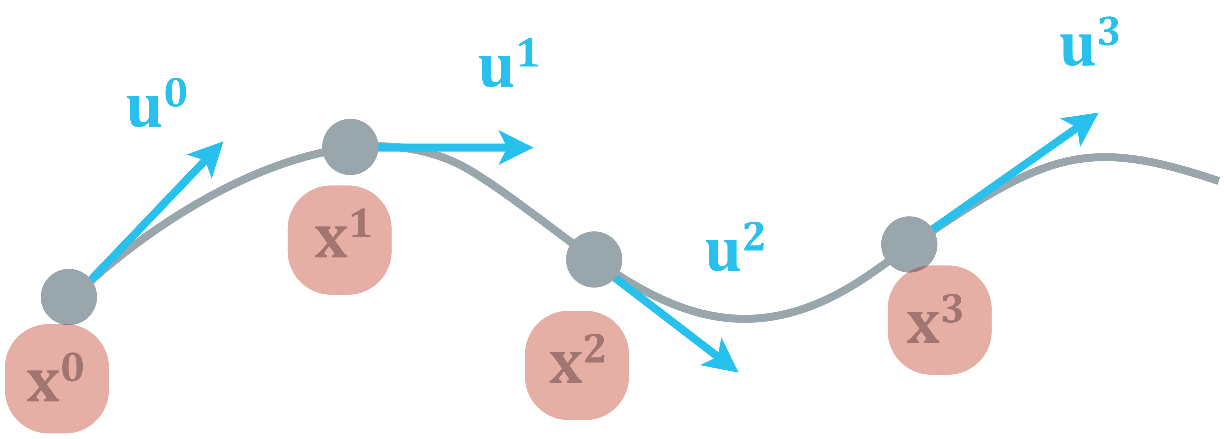
Direct Collocation

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:

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

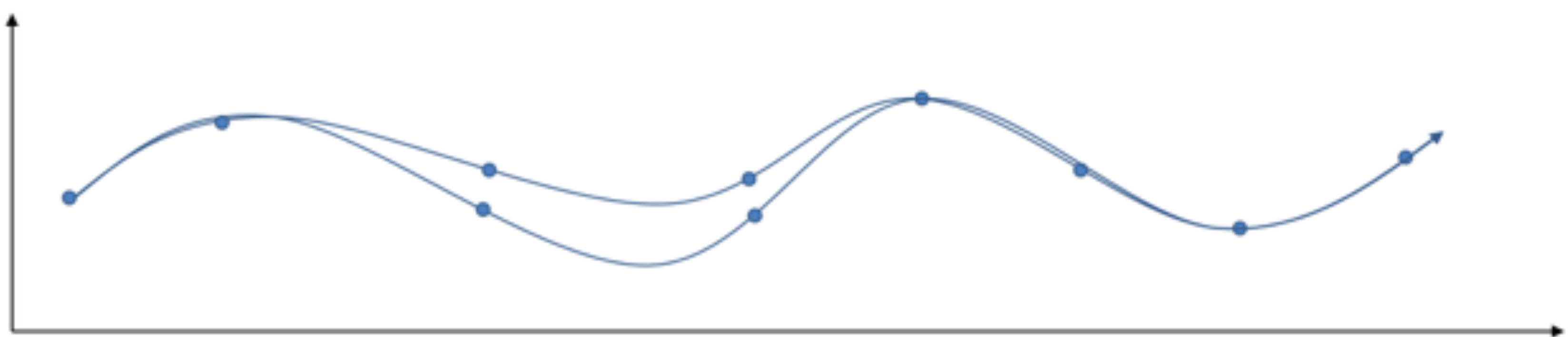


From Last Lecture:

Shooting methods vs collocation

collocation method: optimize over actions and states, with constraints

$$\min_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\mathbf{x}_1,\ldots,\mathbf{x}_T} \sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t) \text{ s.t. } \mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$$



Direct Collocation

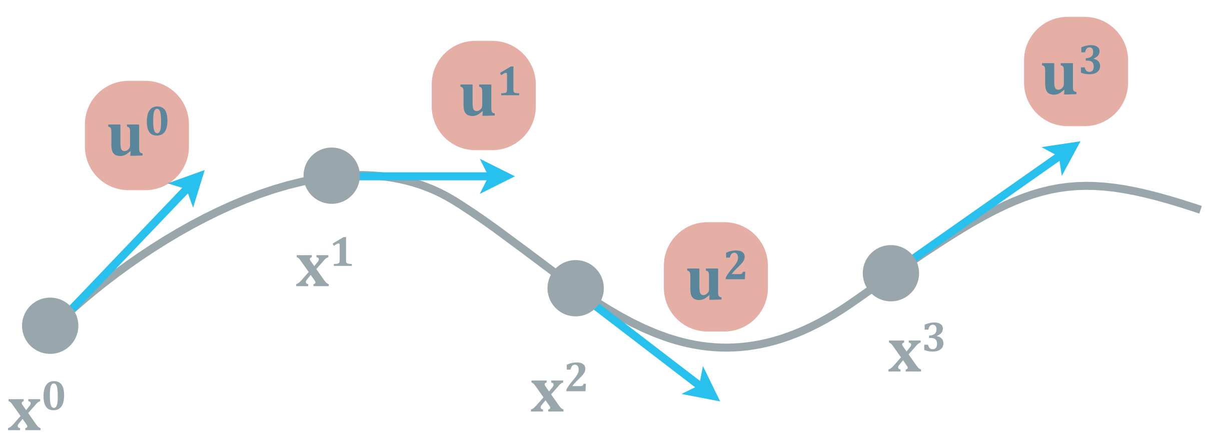
Forward Shooting:

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

inverse dynamics function

Direct Collocation:

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



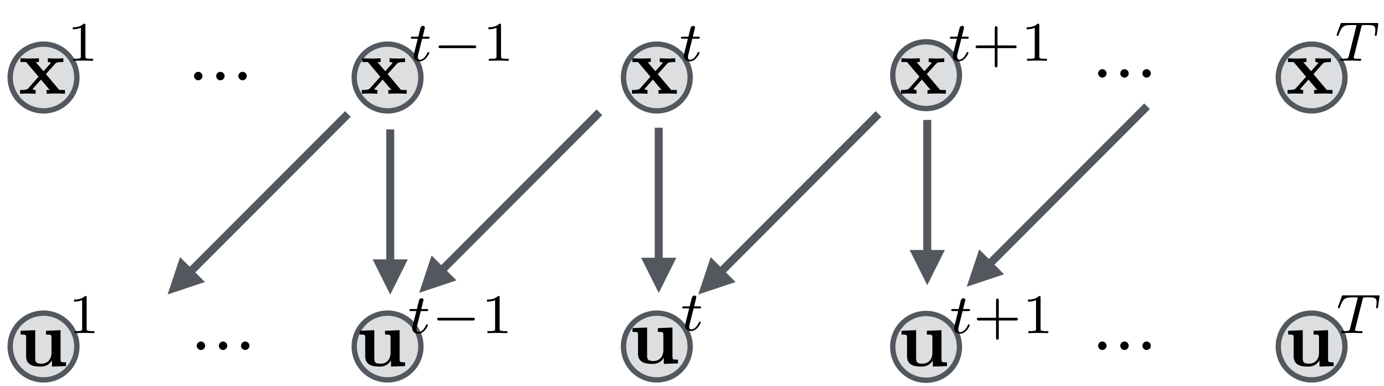
Direct Collocation

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:

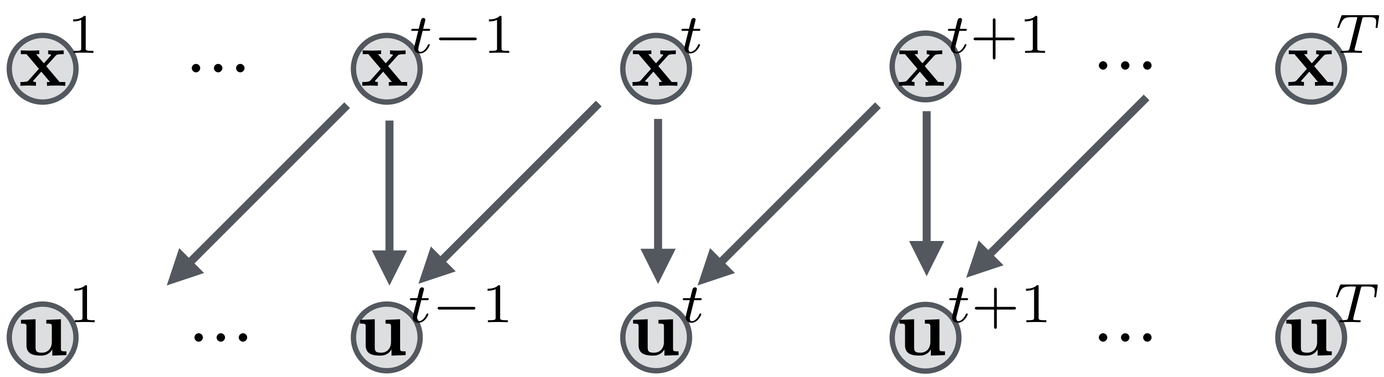
$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t), \;\; st \;\; f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t \in \mathcal{U}$$



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:

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

Explicit rather than implicit constraint

Direct Collocation

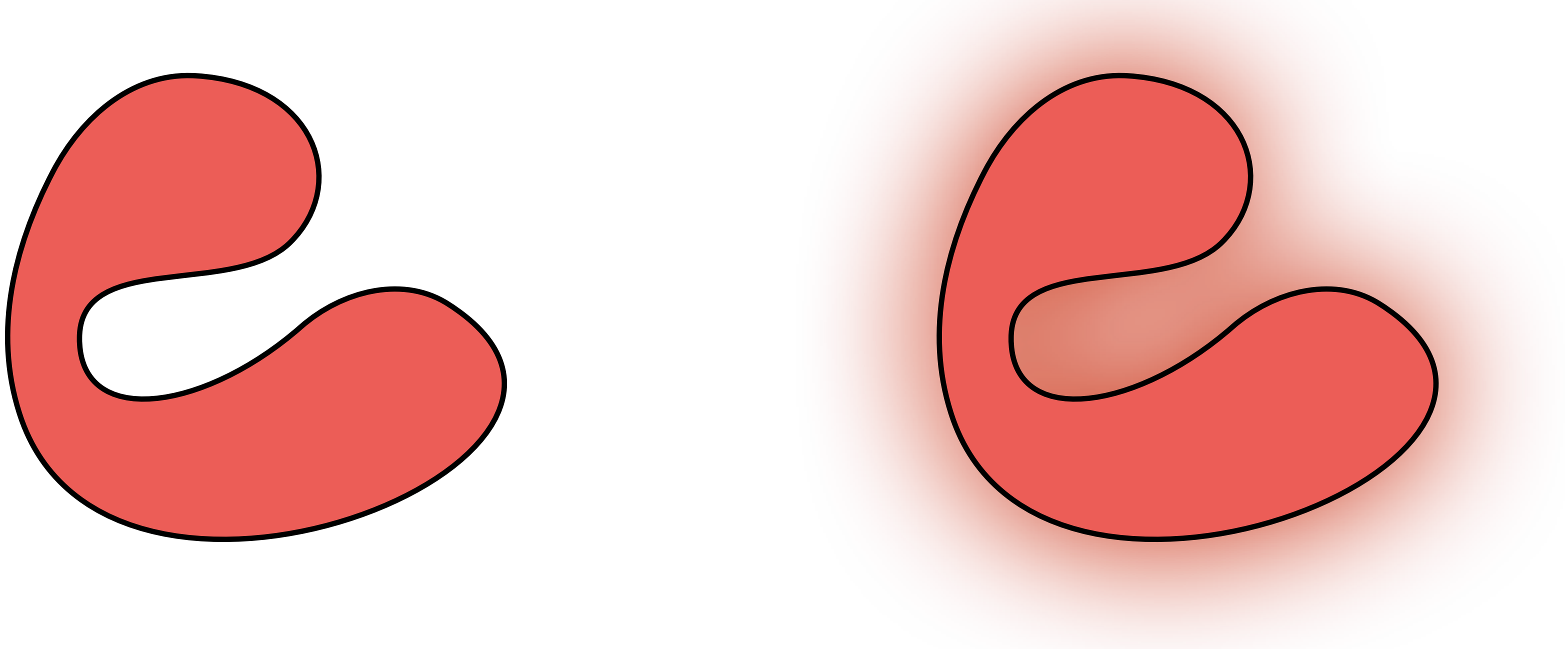
Direct Collocation:

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

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)

Inverse Dynamics Model

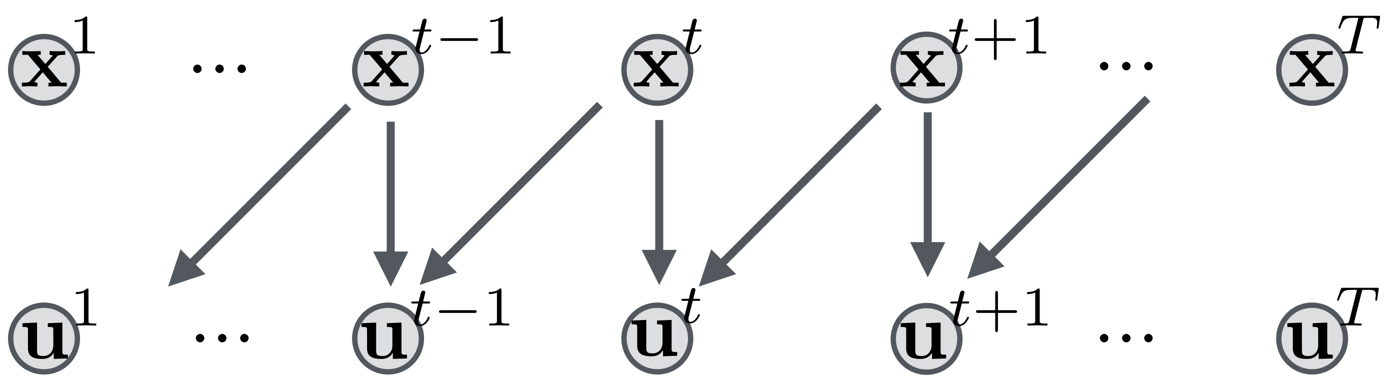
$$f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t$$

- 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

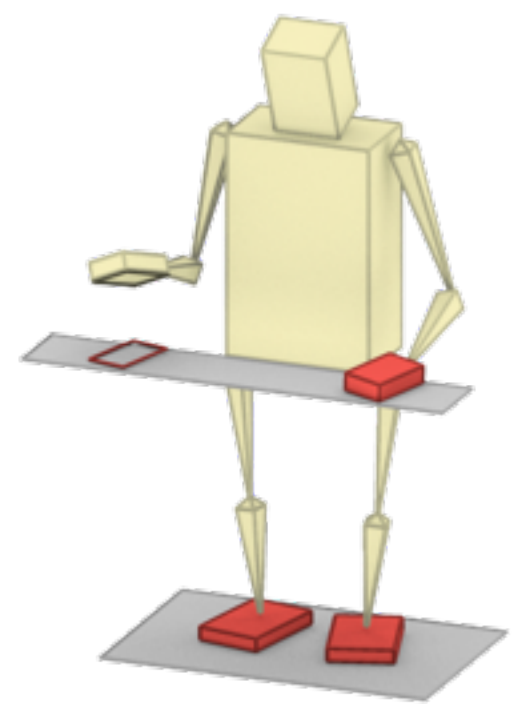
$$f^{-1}(\mathbf{x}^t, \mathbf{x}^{t+1}) = \mathbf{u}^t$$

- 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:

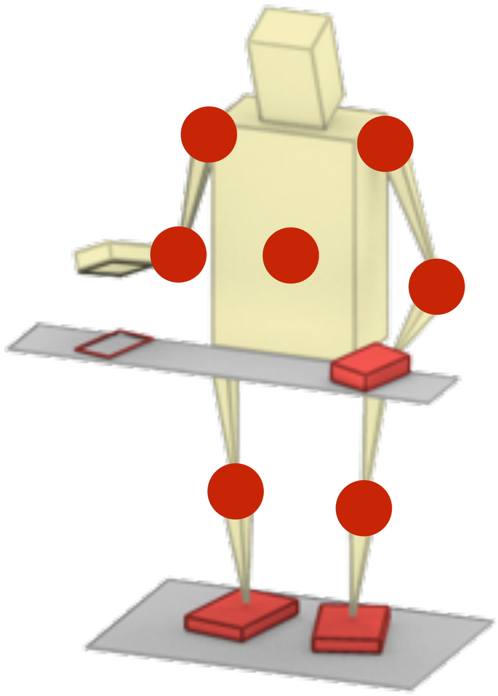
$$\mathbf{x}^t = \mathbf{q}^t$$



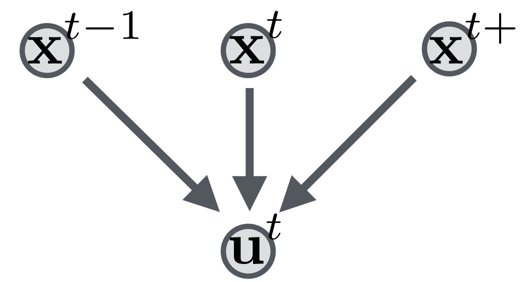
Rigid Multi-Body Dynamics

Generalized coordinates:

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



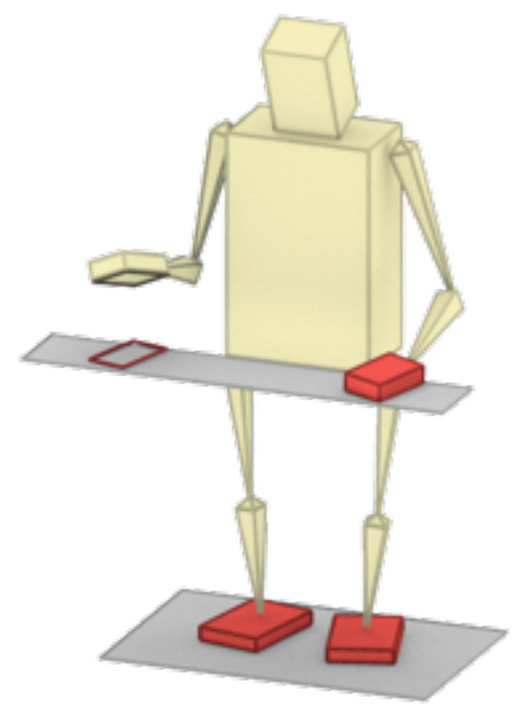
Rigid Multi-Body Dynamics



Generalized coordinates:

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

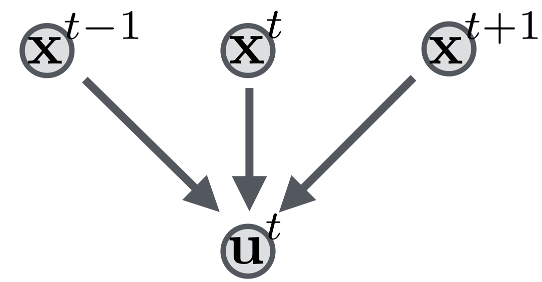
Calculate velocities and accelerations from nearby states



Rigid Multi-Body Dynamics

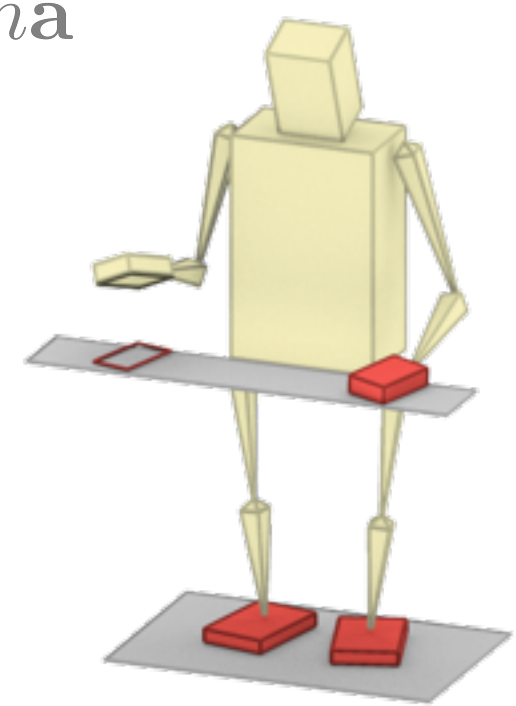
Generalized coordinates:

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



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

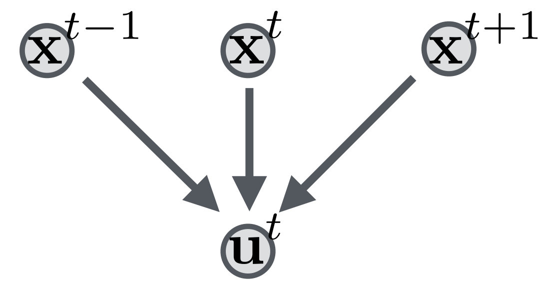
$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$



Rigid Multi-Body Dynamics

Generalized coordinates:

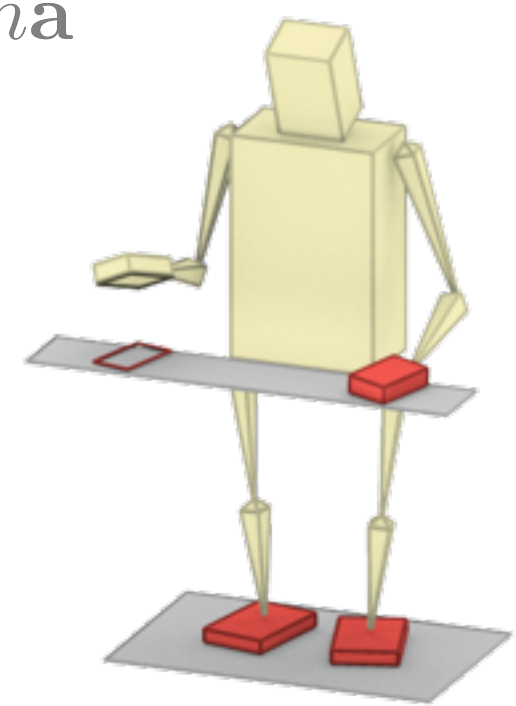
$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$



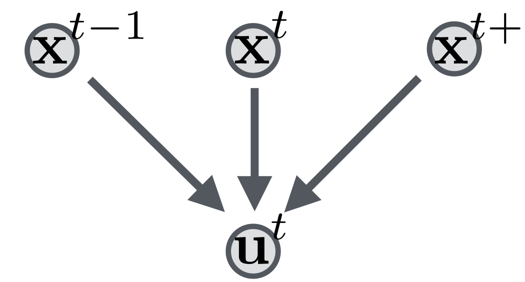
Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

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

Generalized mass and Coriolis matrices



Rigid Multi-Body Dynamics

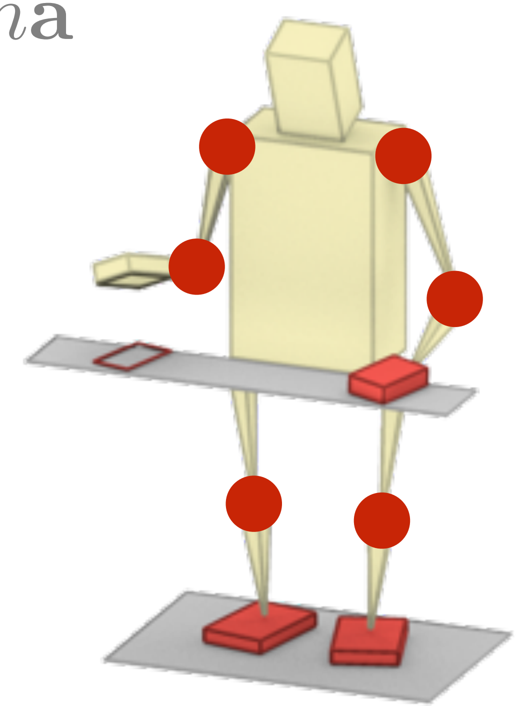


Generalized coordinates:

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

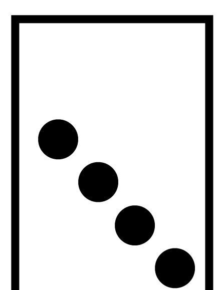
Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

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



Controls and actuation matrix

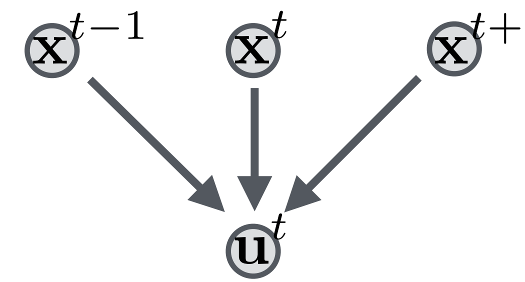
B



Rigid Multi-Body Dynamics

Generalized coordinates:

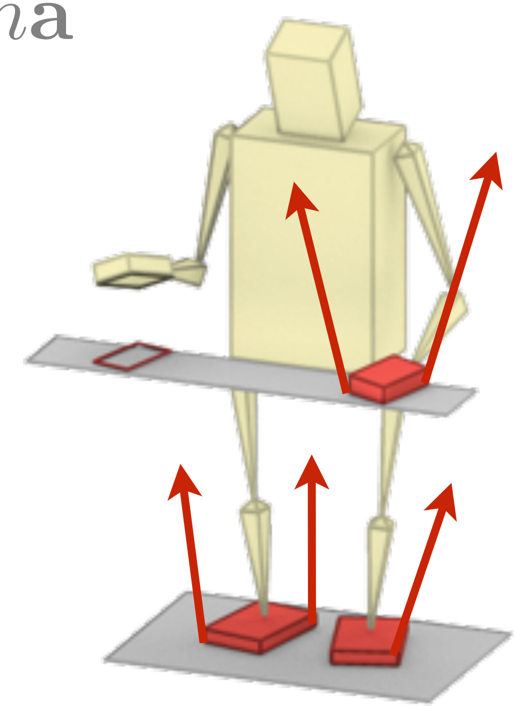
$$\mathbf{x}^t = \mathbf{q}^t \qquad \dot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - \mathbf{q}^t}{2\delta t} \quad \ddot{\mathbf{q}}^t = \frac{\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}}{\delta t^2}$$



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

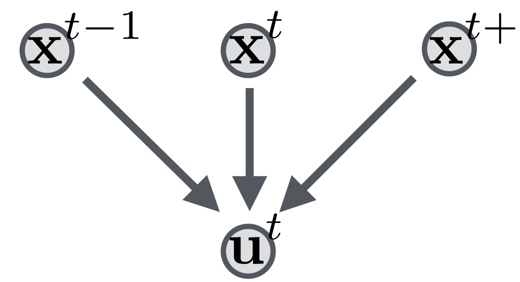
$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

Constraint forces and constraint Jacobian



Rigid Multi-Body Dynamics

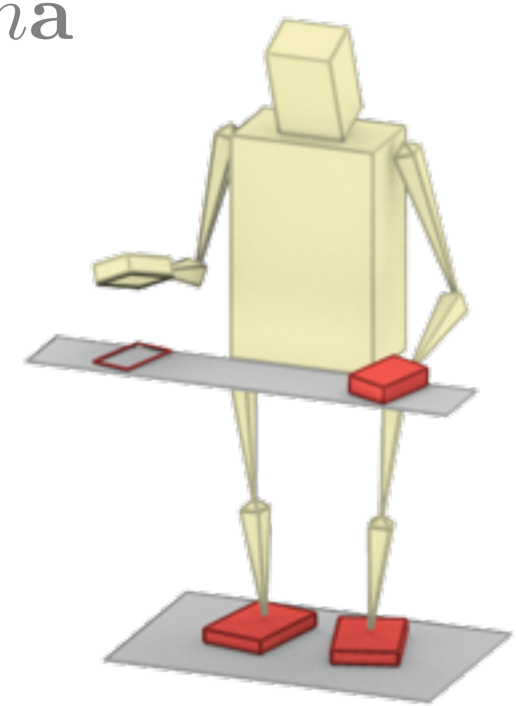
Generalized coordinates:



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

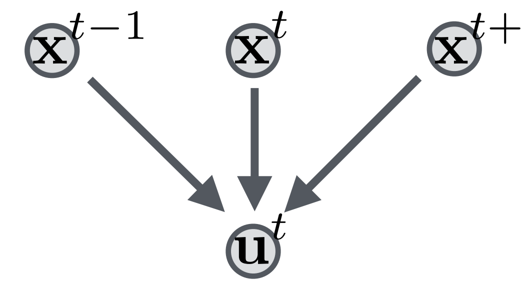
Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

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



For more detail, see chapters 2 and 3 in *Springer Handbook of Robotics* and *Analytical Dynamics: A New Approach*

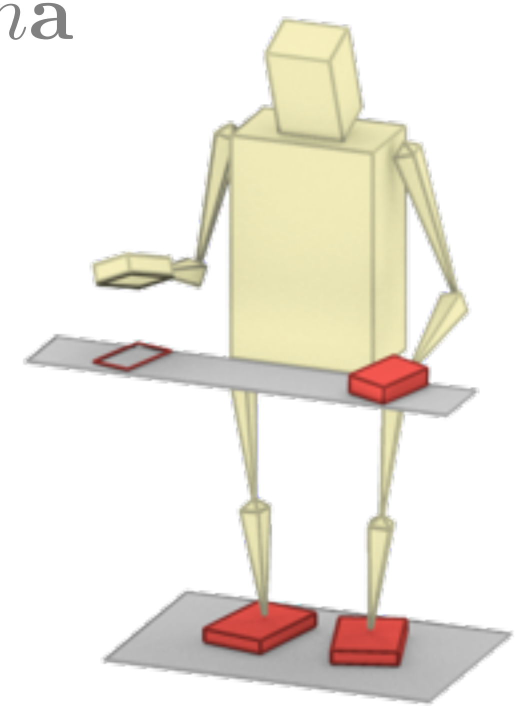
Rigid Multi-Body Dynamics



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 $\mathbf{f} = m\mathbf{a}$



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

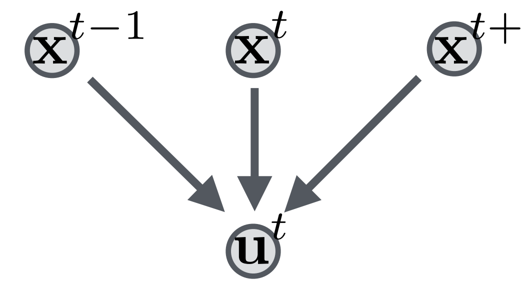
Inverse dynamics function:

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

Rigid Multi-Body Dynamics

Generalized coordinates:

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



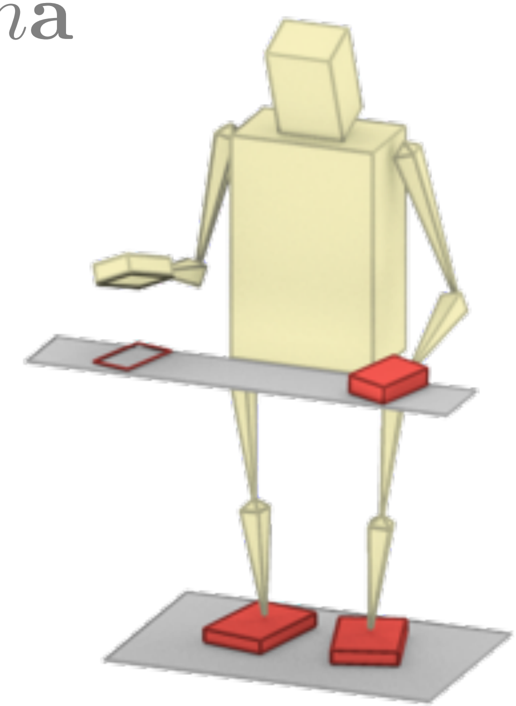
Dynamics equation: generalization of $\mathbf{f} = m\mathbf{a}$

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

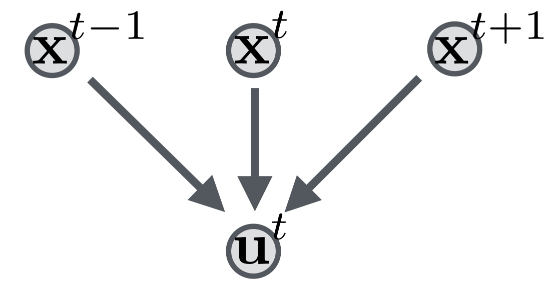
Inverse dynamics function:

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

can be solved numerically, or analytically [Todorov 14]



Rigid Multi-Body Dynamics

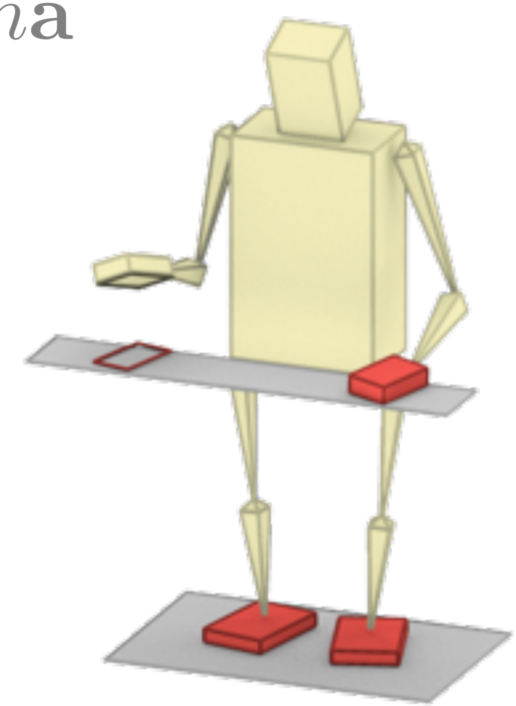


Generalized coordinates:

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

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

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

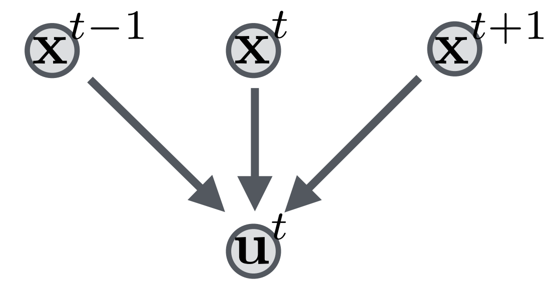


Inverse dynamics residual:

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

Rigid Multi-Body Dynamics

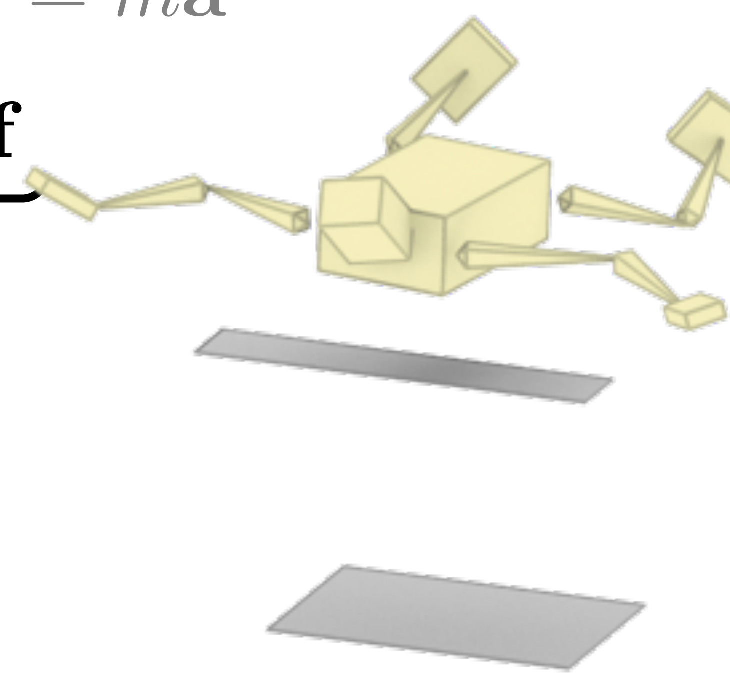
Generalized coordinates:



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

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:

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

Simple Particle Example

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

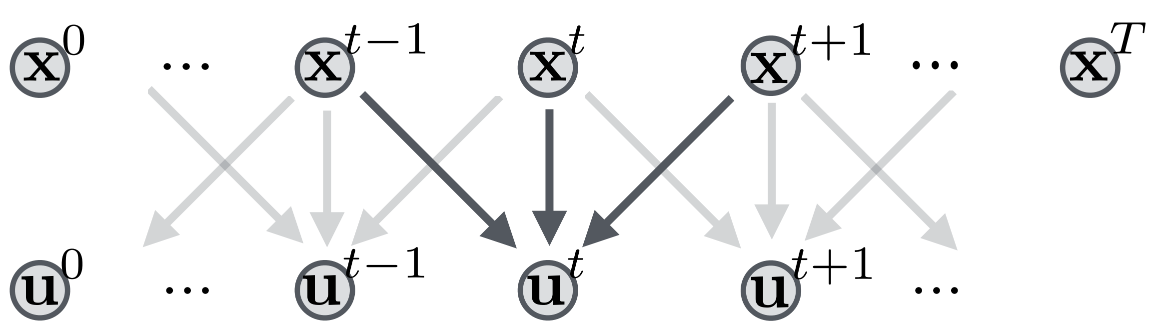
$$f^{-1}(\mathbf{x}^{t-1}, \mathbf{x}^{t}, \mathbf{x}^{t+1}) = \mathbf{u}^{t} = m(\mathbf{x}^{t-1} - 2\mathbf{x}^{t} + \mathbf{x}^{t+1})/\delta t + \mathbf{g}$$

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

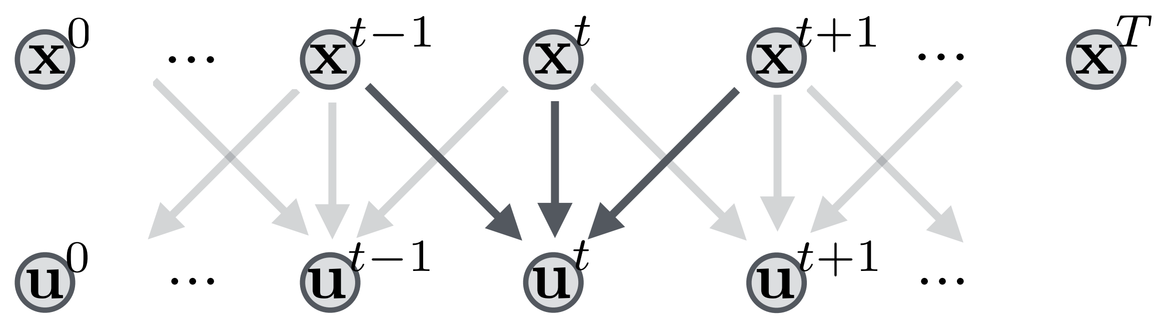
  Initial state: $\mathbf{x}^0$ System parameters: m External forces: $\mathbf{g}$

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

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



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

- Total trajectory cost is

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

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

$$C_{\mathbf{X}} = \sum_t c_\phi^t \phi_{\mathbf{X}}^t$$

$$C_{\mathbf{XX}} = \sum_t (\phi_{\mathbf{X}}^t)^\top c_{\phi\phi}^t \phi_{\mathbf{X}}^t + c_\phi^t \phi_{\mathbf{XX}}^t \approx \sum_t (\phi_{\mathbf{X}}^t)^\top c_{\phi\phi}^t \phi_{\mathbf{X}}^t$$

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

$$C_{\mathbf{X}} = \sum_t c_\phi^t \phi_{\mathbf{X}}^t$$

$$C_{\mathbf{XX}} = \sum_t (\phi_{\mathbf{X}}^t)^\top c_{\phi\phi}^t \phi_{\mathbf{X}}^t + c_\phi^t \phi_{\mathbf{XX}}^t \approx \sum_t (\phi_{\mathbf{X}}^t)^\top c_{\phi\phi}^t \phi_{\mathbf{X}}^t$$

- Find optimal solution by iterative Gauss-Newton steps

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

Typically use damped Hessian
(similar to Trust Region)

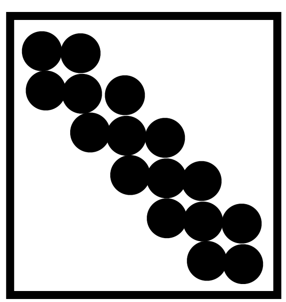
$$(C_{\mathbf{XX}}^{-1} + \lambda \mathbf{I}) C_{\mathbf{X}}$$

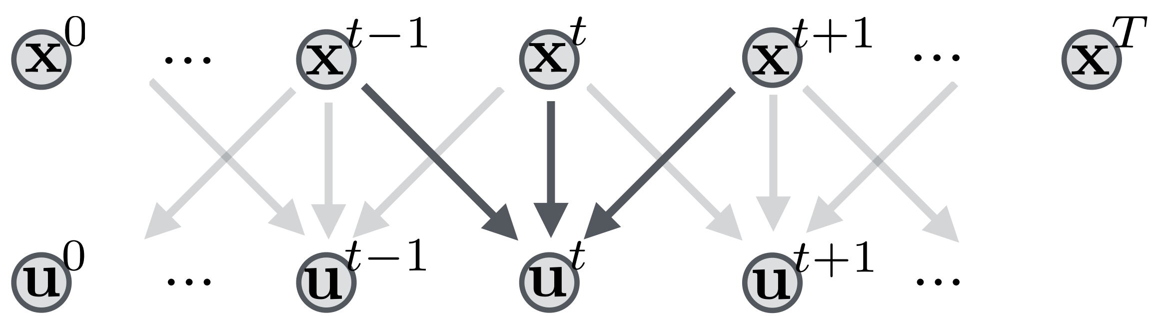
Gauss-Newton Method

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

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

- Hessian is block sparse



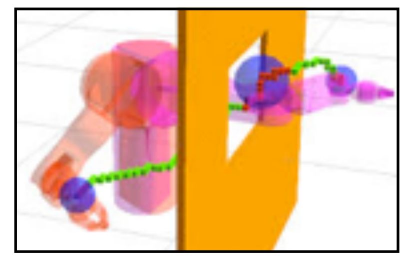


- Can use sparse linear system solvers
  - python: linalg.spsolve
  - Other methods possible (multigrid, projection, spectral?)
  - Constrained optimization possible (SQP) [Posa and Tedrake 12]

Dynamics with Contact

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

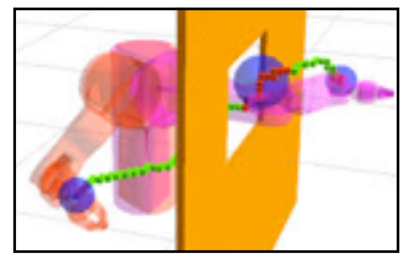





Dynamics with Contact

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



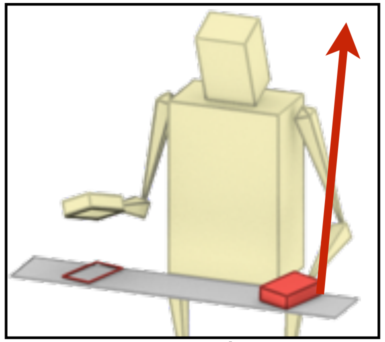



- With contact, it is difficult to apply either method
  - legged robots, manipulation

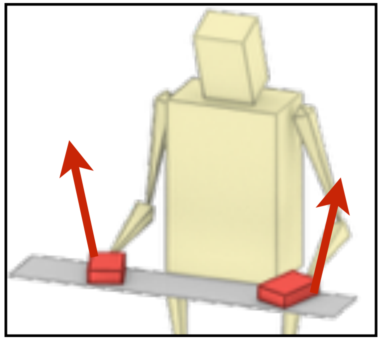



Dynamics with Contact

- Discontinuous jumps in contact forces (and their number)



$\mathbf{q}^1$



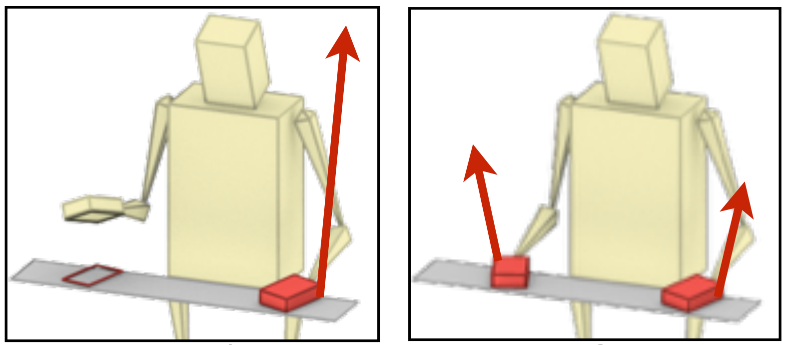
$\mathbf{q}^2$

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)

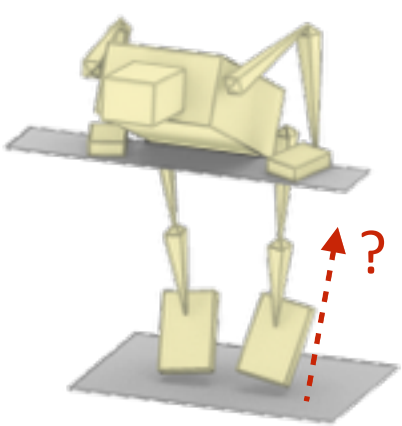


$\mathbf{q}^1$ $\mathbf{q}^2$

Dynamics equation:

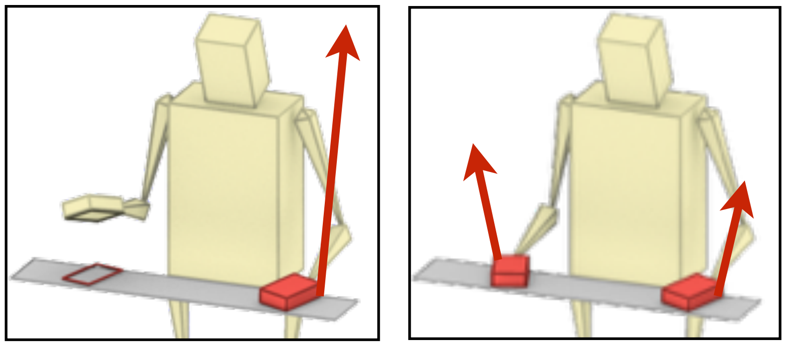
$$M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}$$

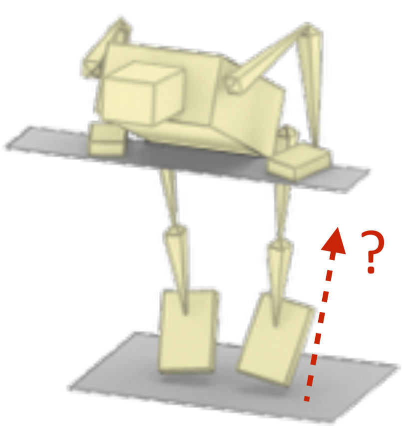
- No gradient information from inactive contacts



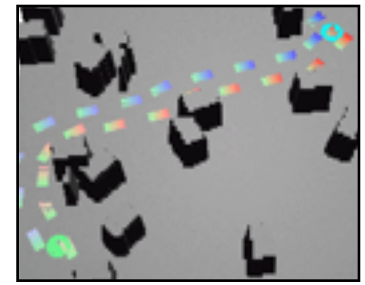
Can't anticipate being able to apply forces

Dynamics with Contact



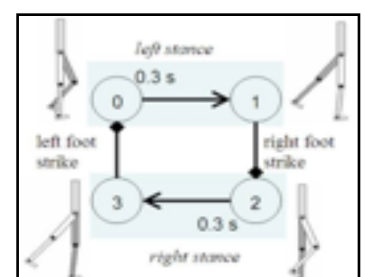


manual specification

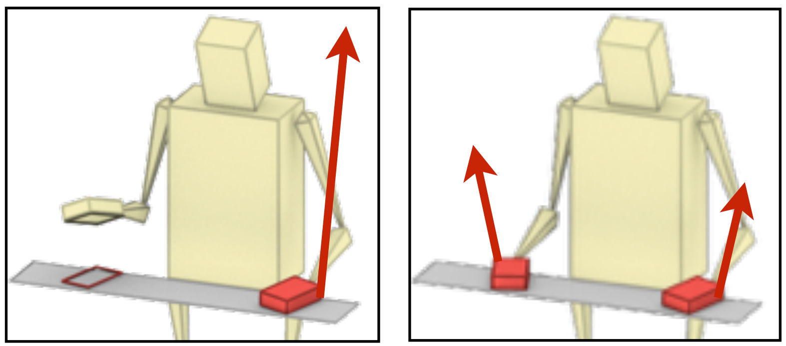


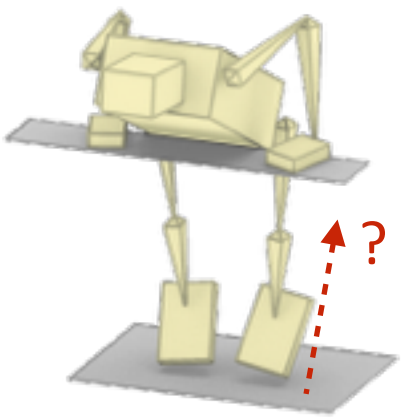
track demonstrations

motion structure



Dynamics with Contact





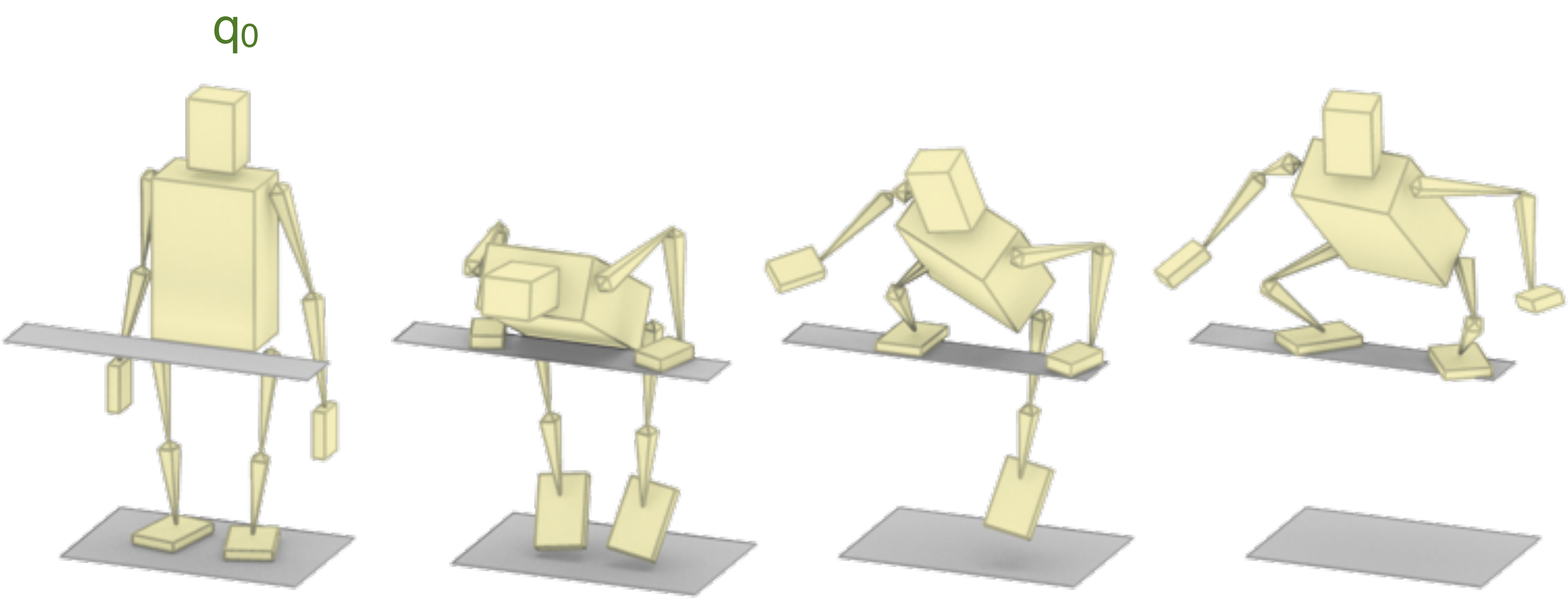
- Contact activity is an indirect function of state
- What if we make contact activity a direct optimization variable like we did for state?

Contact-Invariant Optimization

[Mordatch, Todorov, Popovic, SIGGRAPGH 2012]

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

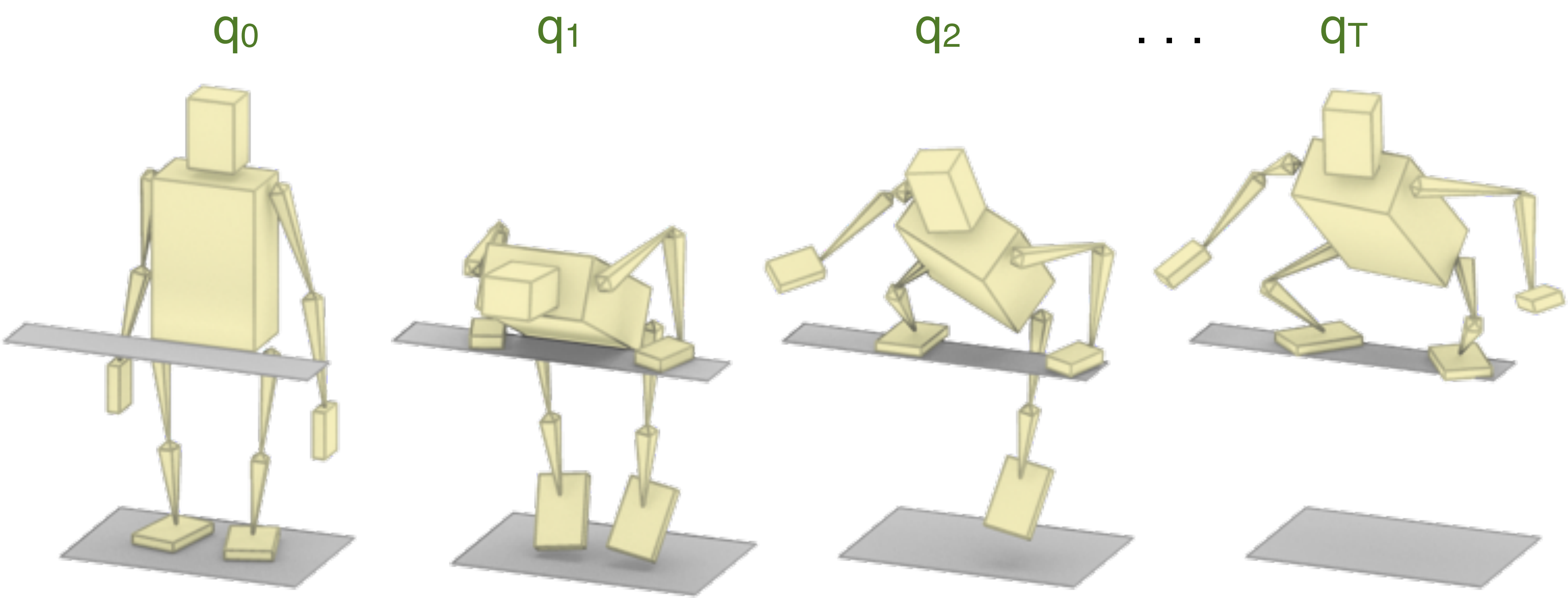
x: [ ]



Contact-Invariant Optimization

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

x: [q]

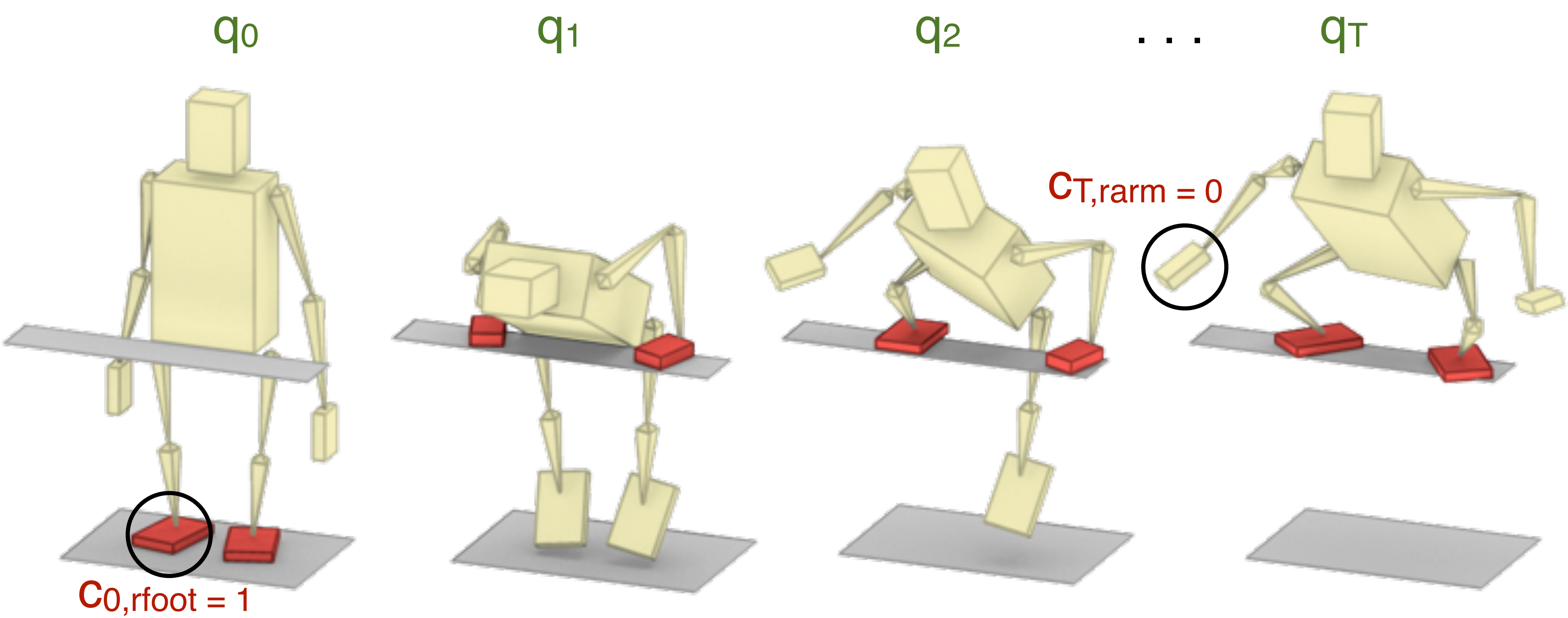


Contact-Invariant Optimization

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

x: [q c]

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



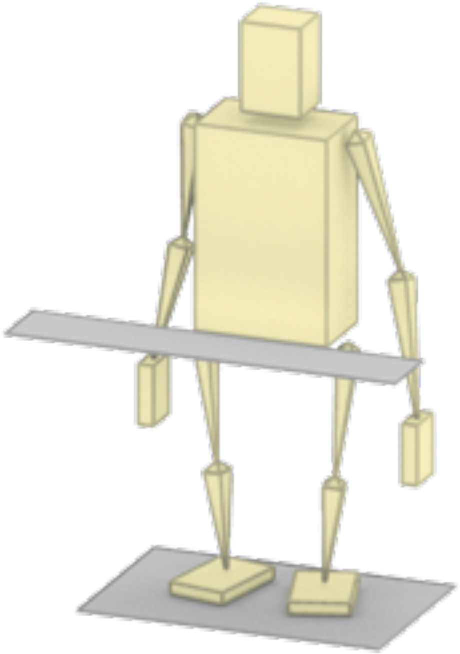
Contact-Invariant Optimization

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

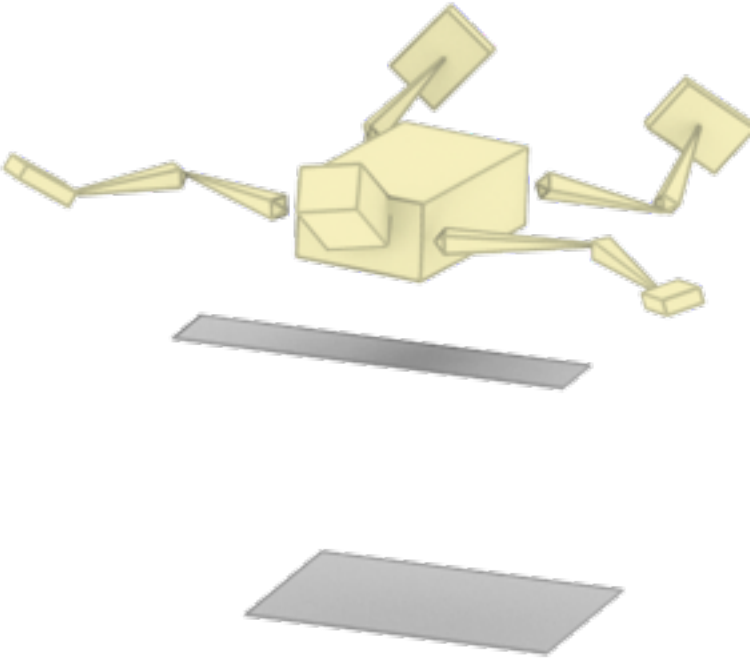
x: [q c]

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

q_0



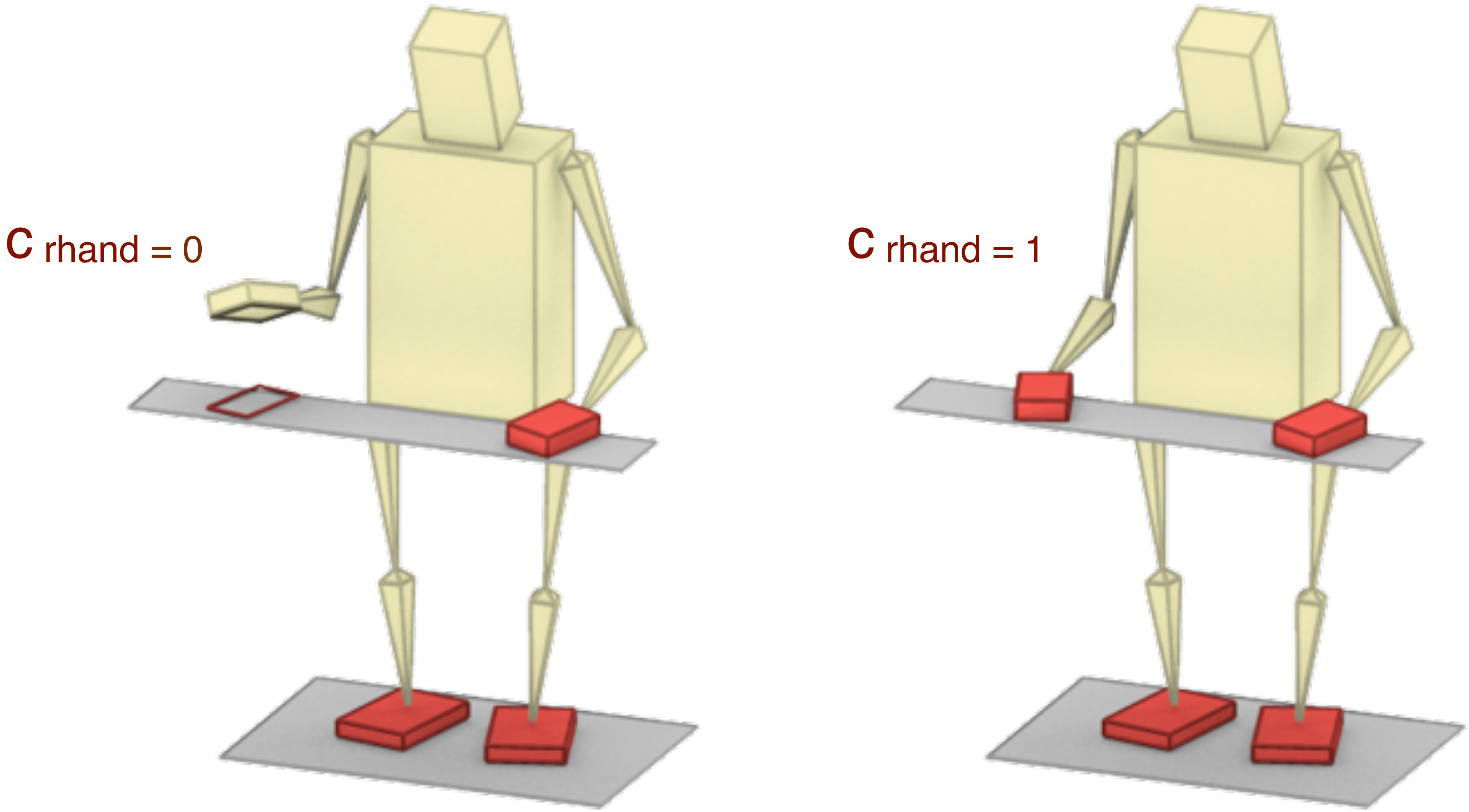
q_K



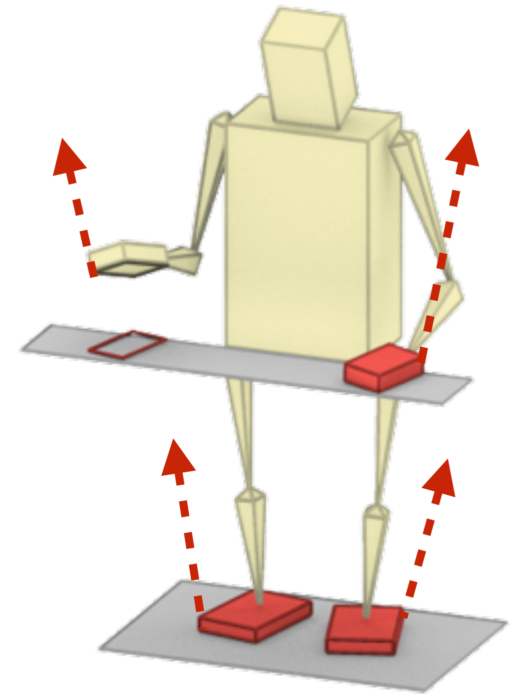
Contact Consistency

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

When $c_n = 0$ limb n is unconstrained



Dynamics Consistency

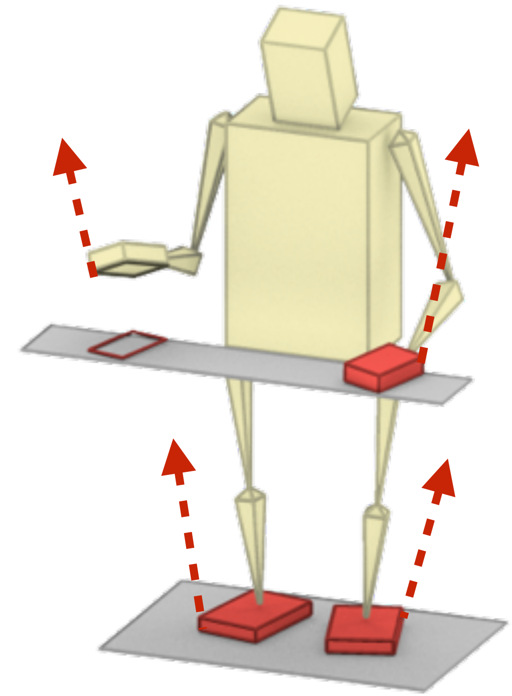


$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$

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

All forces are active (contact set is constant)

Dynamics Consistency



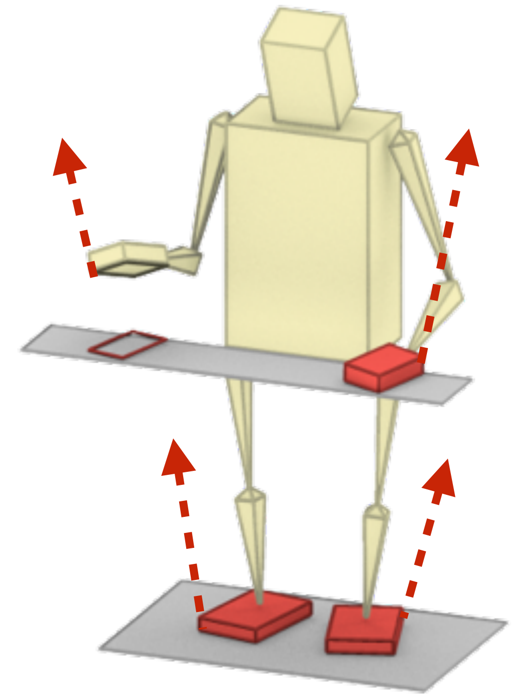
$$\underbrace{M(\mathbf{q})\,\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} = B\mathbf{u} + J(\mathbf{q})^T\,\mathbf{f}}$$

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

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

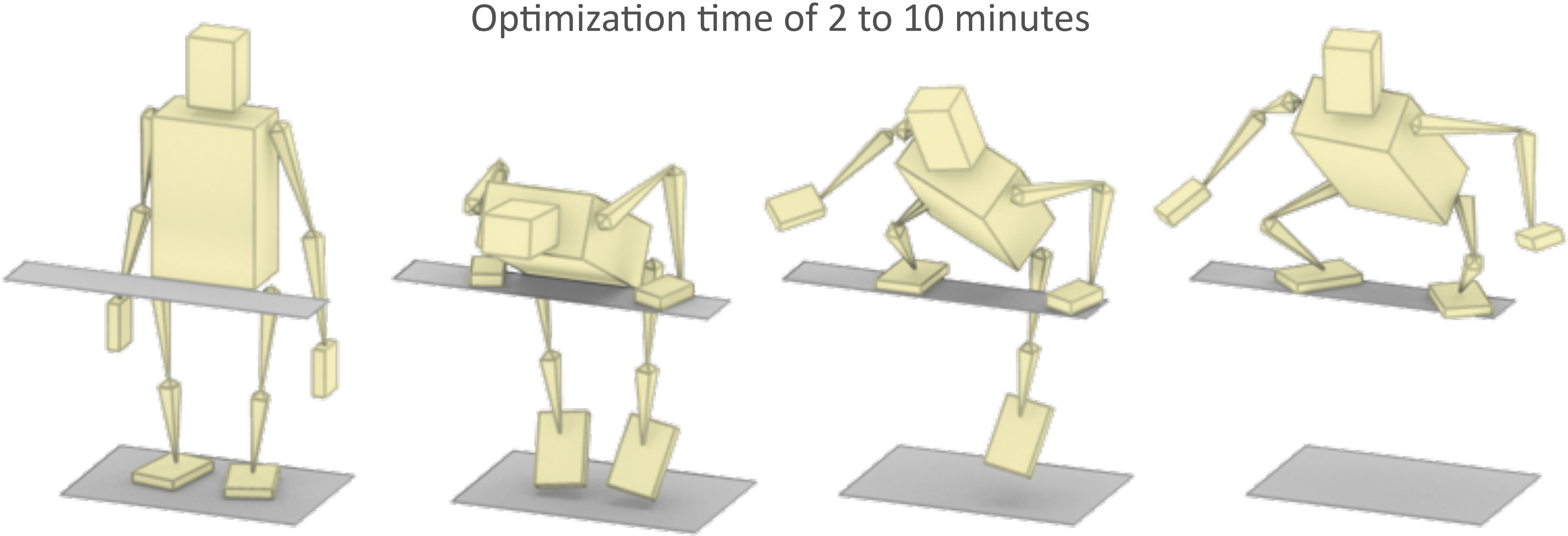
$$\min_{\mathbf{x}^0 \ldots \mathbf{x}^T} \sum_t C^t(\mathbf{x}^t)$$

x: [q c]

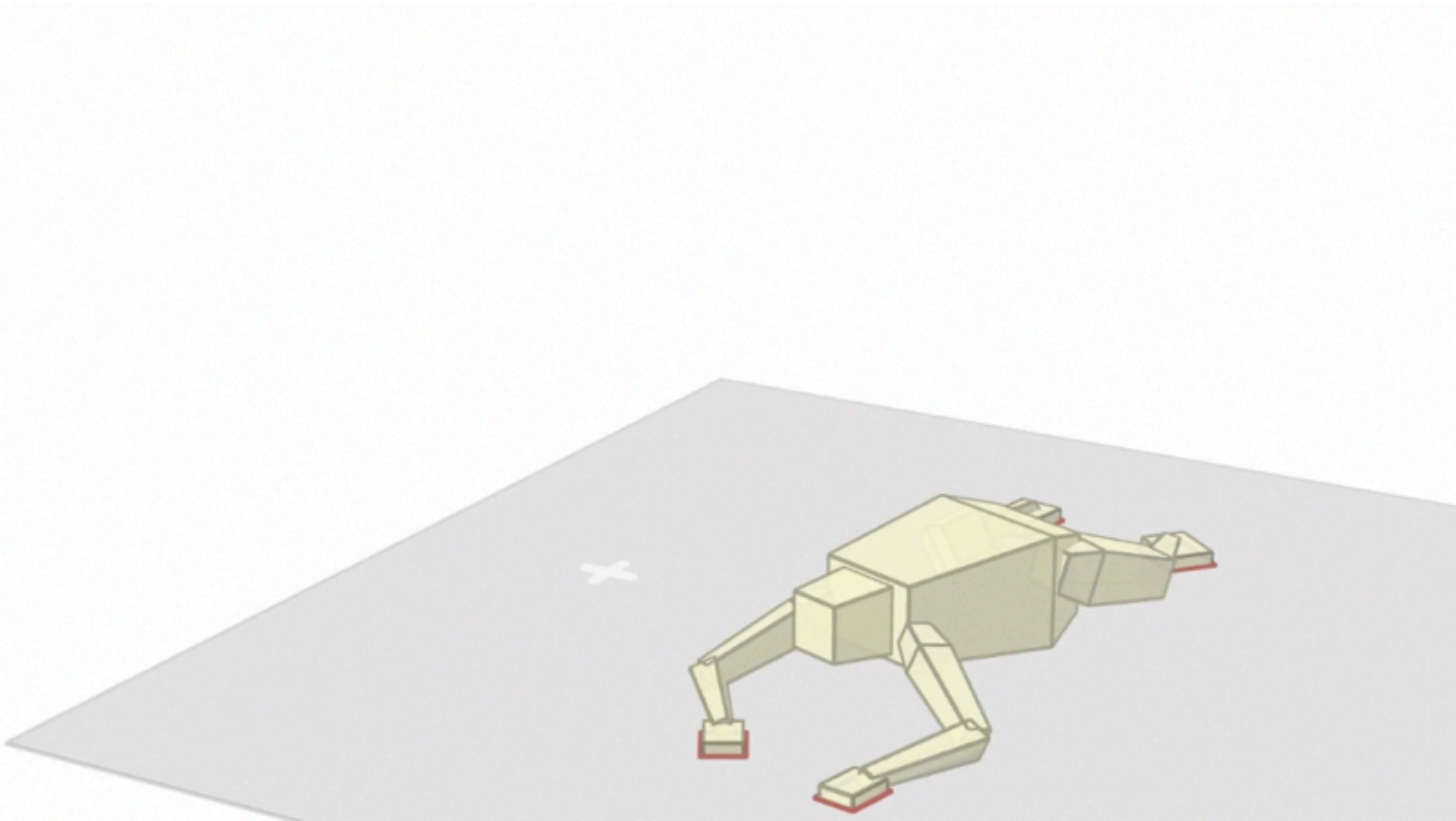
No contact discontinuities and always have a gradient

Solved with standard local optimization

Optimization time of 2 to 10 minutes

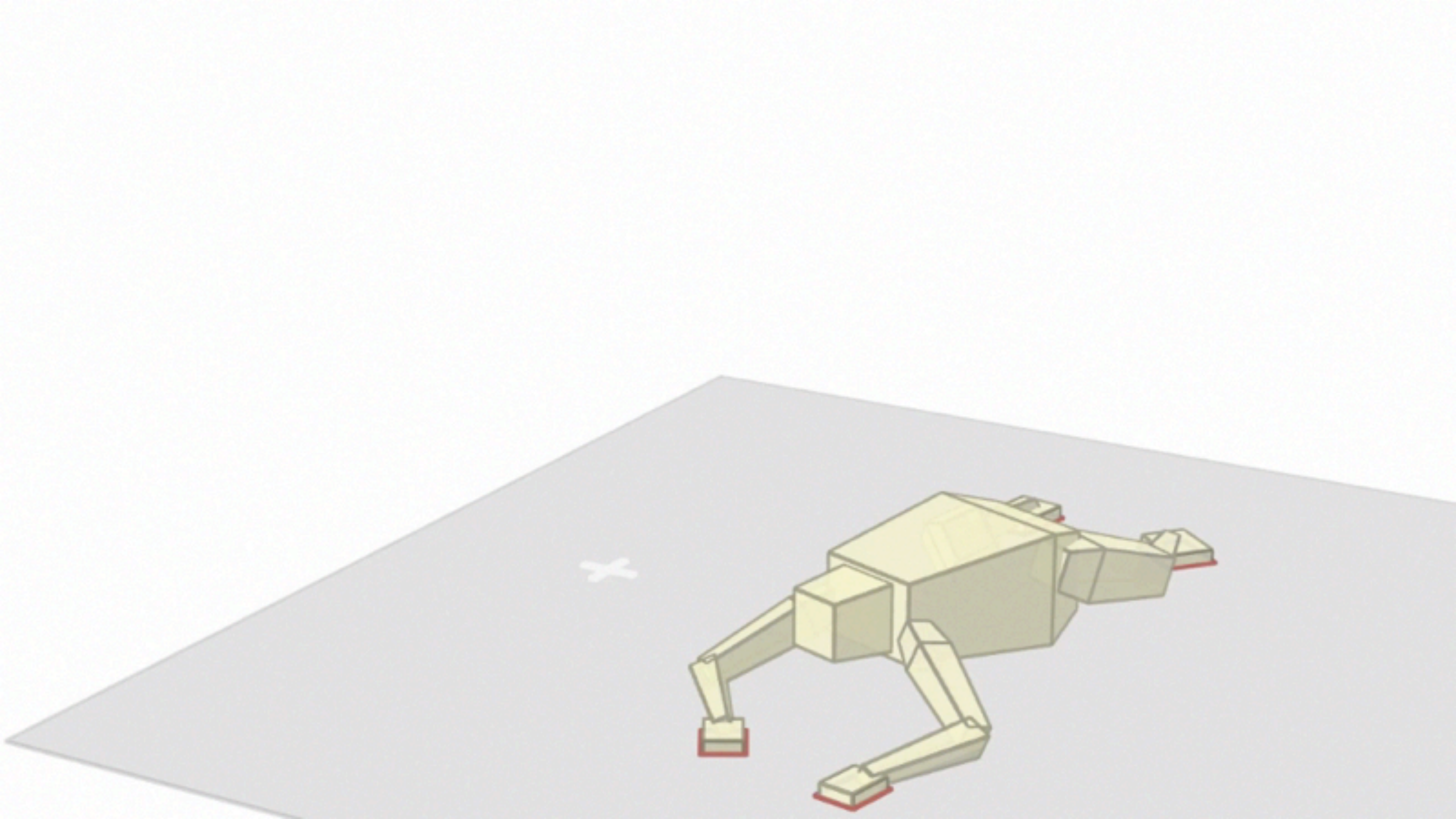


Optimization Progress



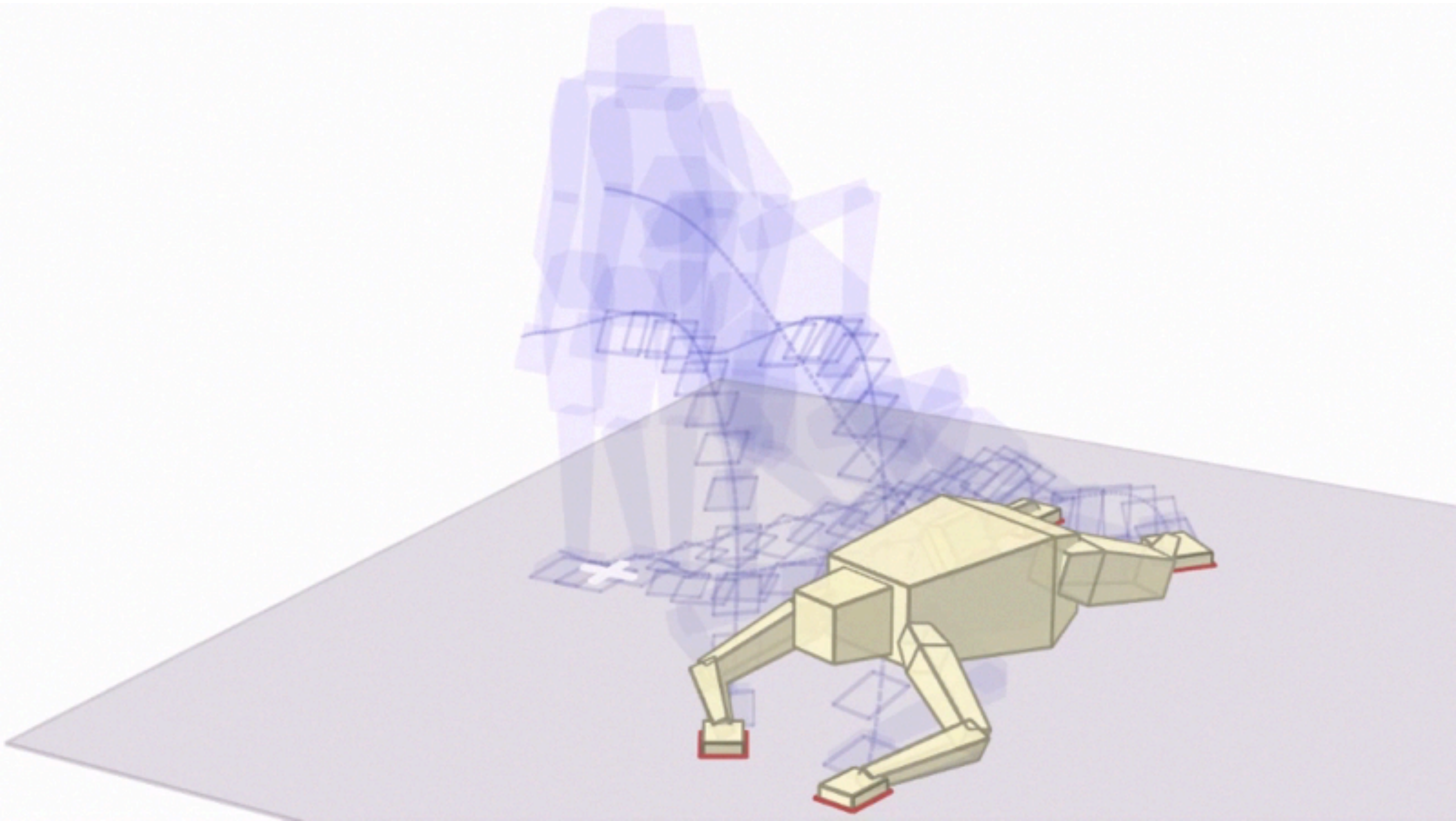
Optimization Progress

Stage 1



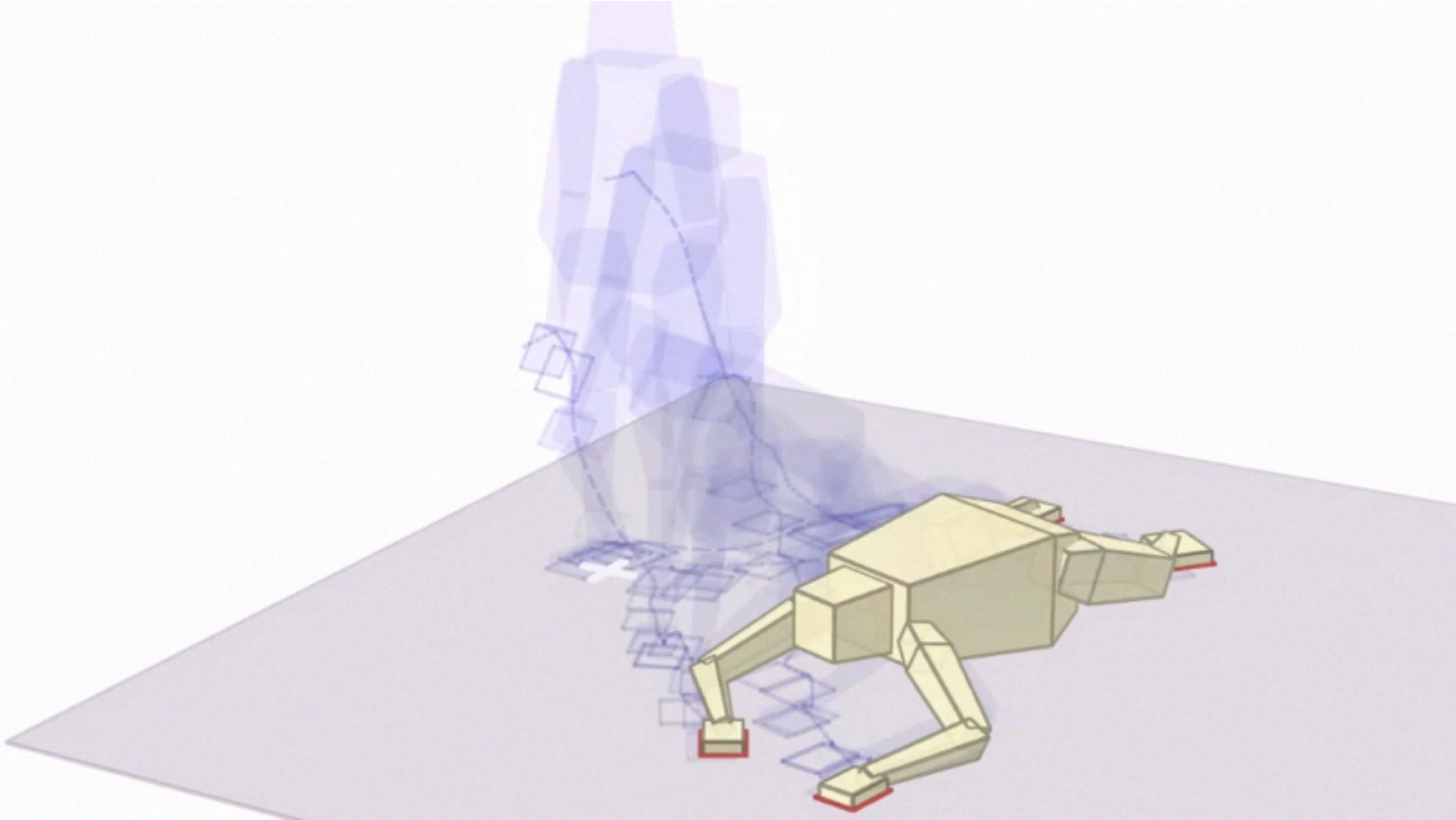
Optimization Progress

Stage 2

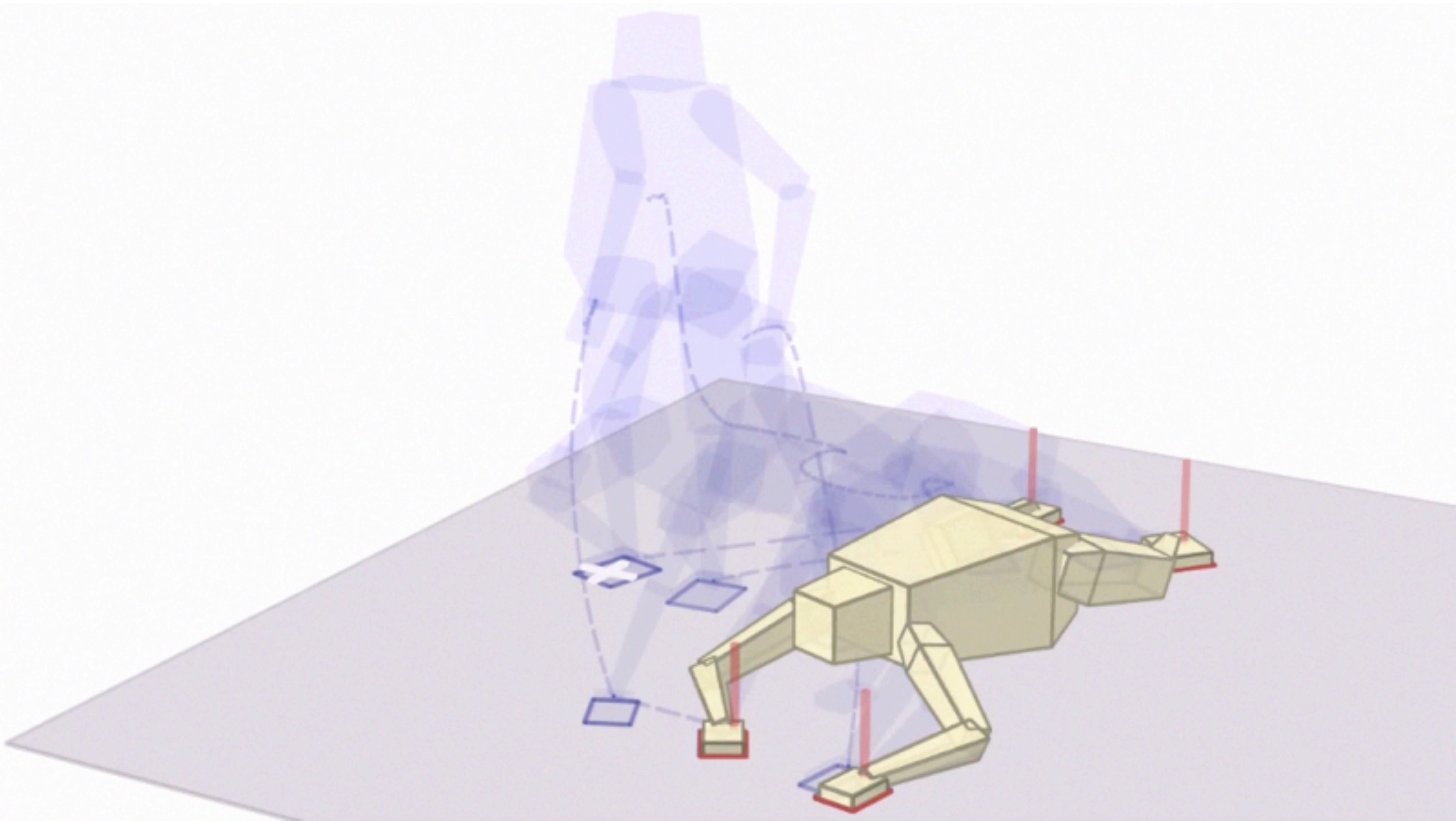


Optimization Progress

Stage 3



Optimization Result

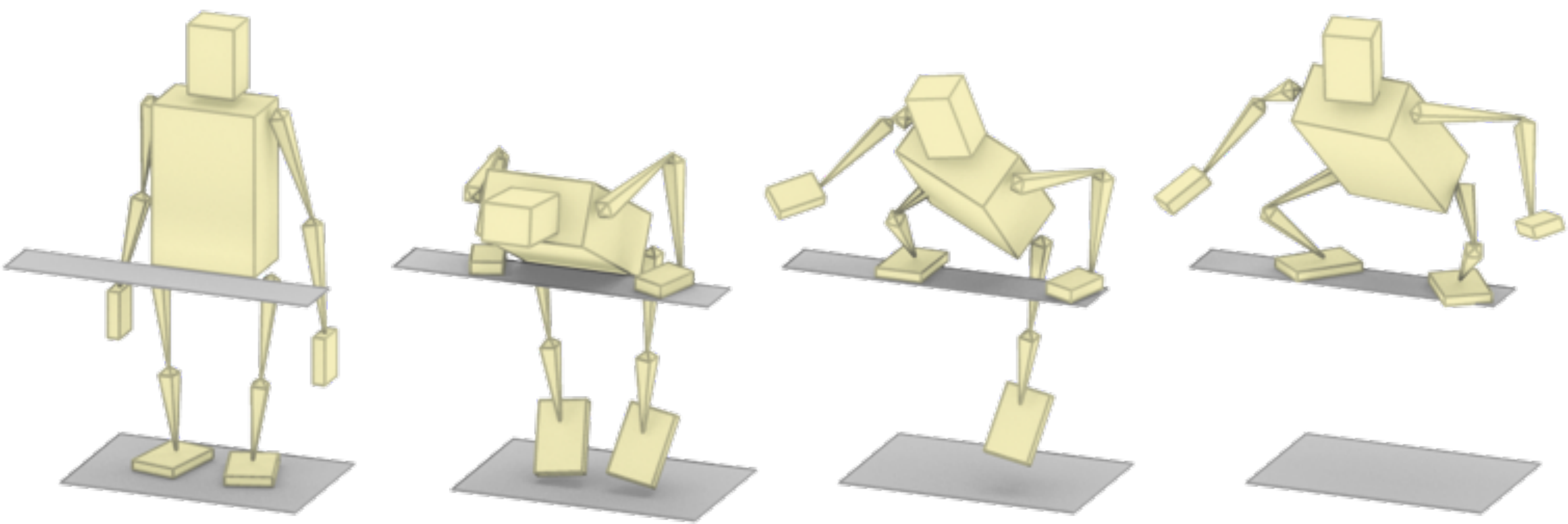


Idea

Add auxiliary variables

Softly enforce consistency between variables

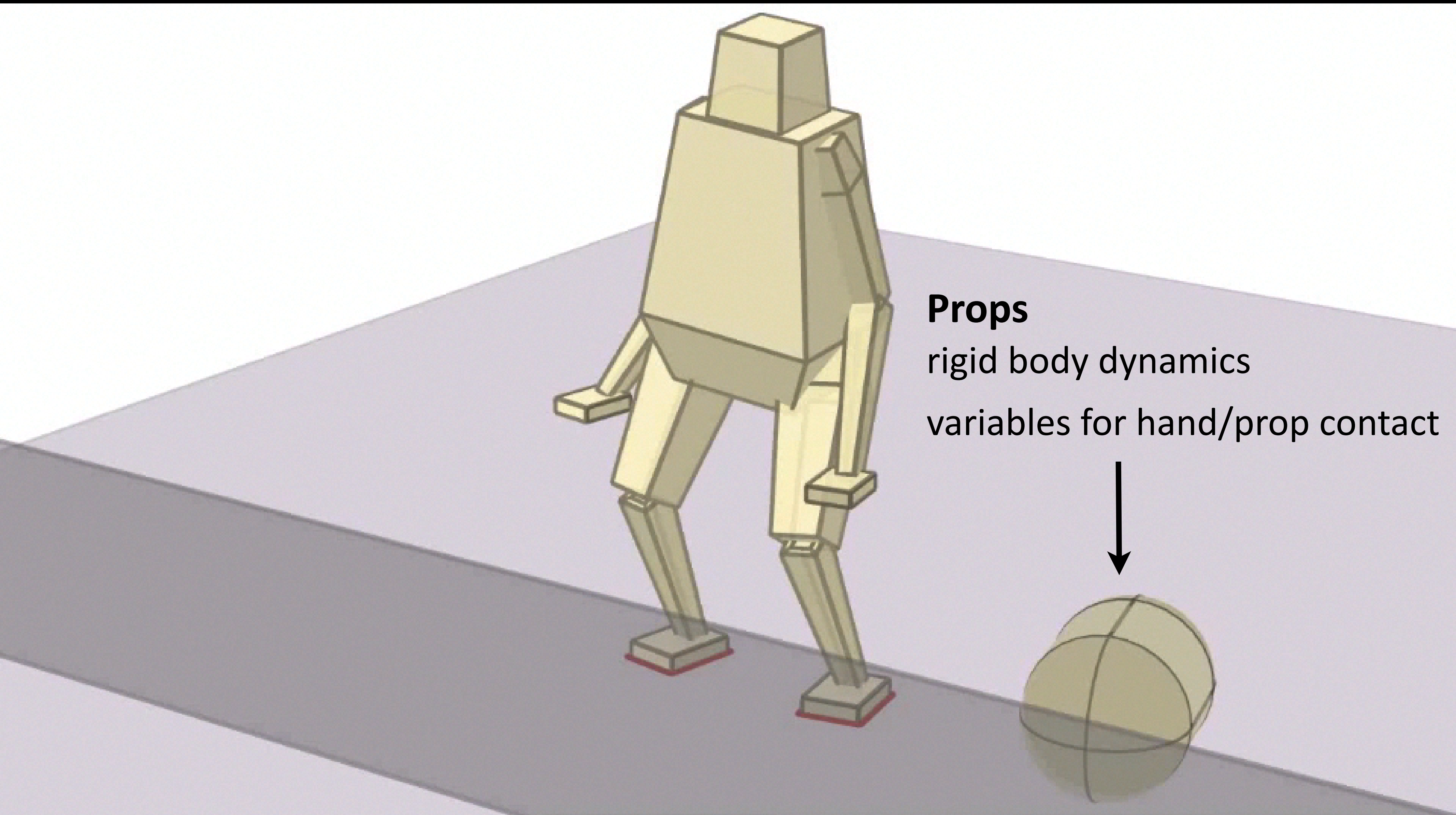
Search in larger, but easier to explore space

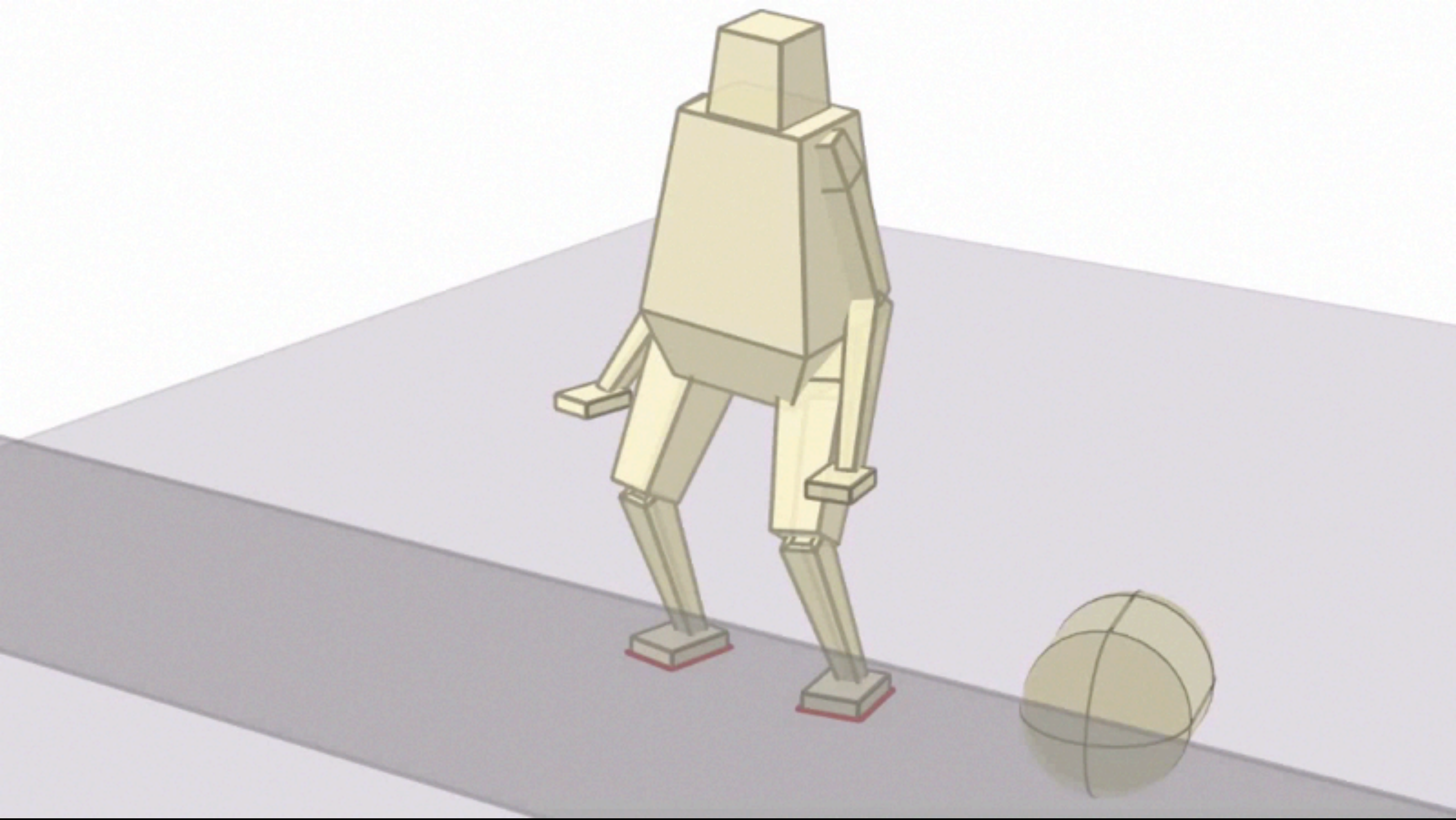


Interaction with Environment

Agile Behaviors

Non-Humanoid Character Morphologies


Props
rigid body dynamics
variables for hand/prop contact

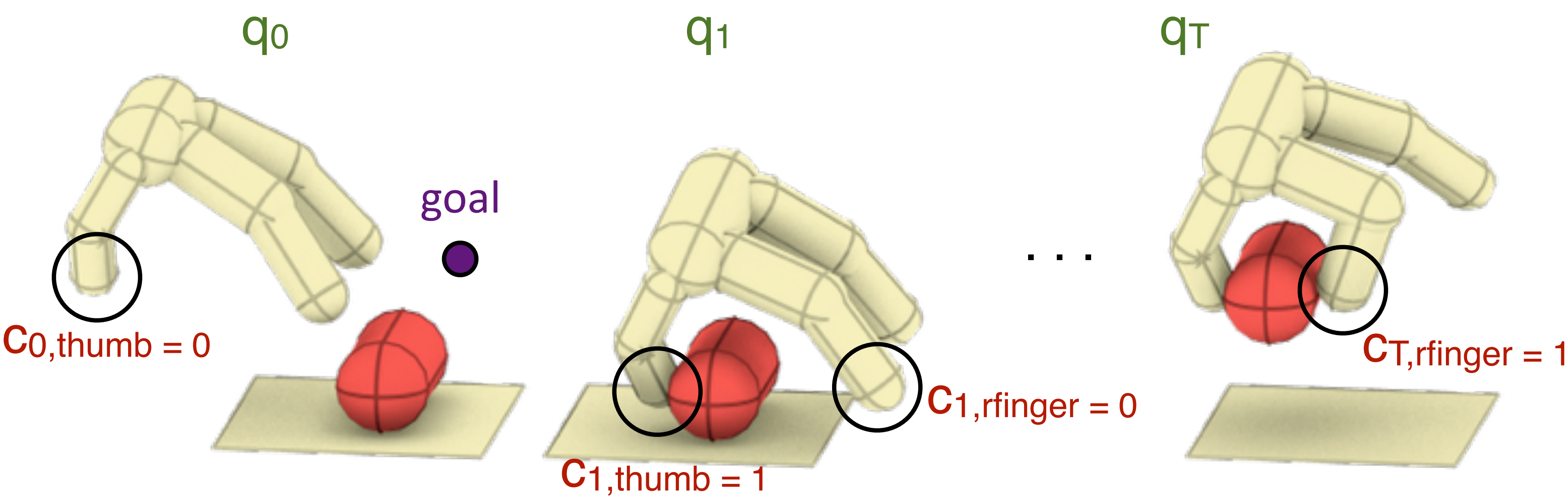


Interaction Between Multiple Characters

Hand Manipulation

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

x: [q c]

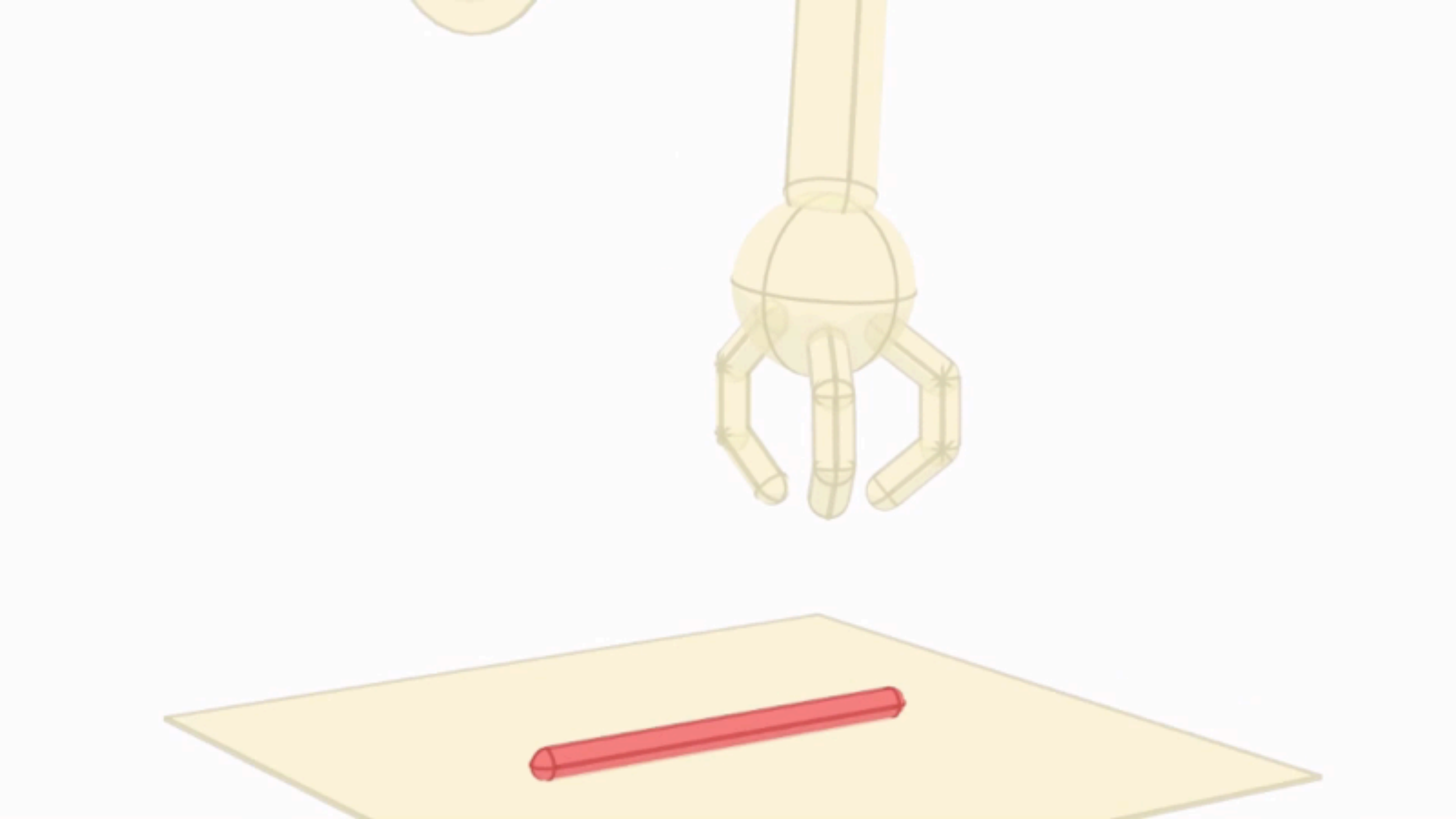


[Mordatch, Popovic, Todorov, SCA 2012]

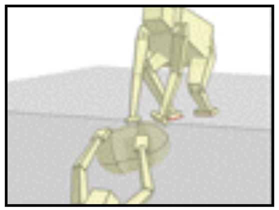
Object Grasping

In-Hand Object Manipulation

Manipulation Tasks



Two-Handed Manipulation



Trajectory Optimization with Direct Collocation

Automatic and general approach

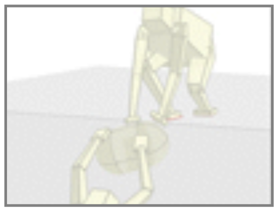
Optimization problem for each motion clip

Do we solve optimization problems to move?

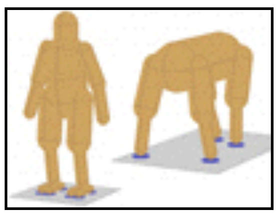
No learning or reuse in optimization

Cannot deal with unexpected events

Instead of motion clips, find *policies*



Trajectory Optimization with Direct Collocation

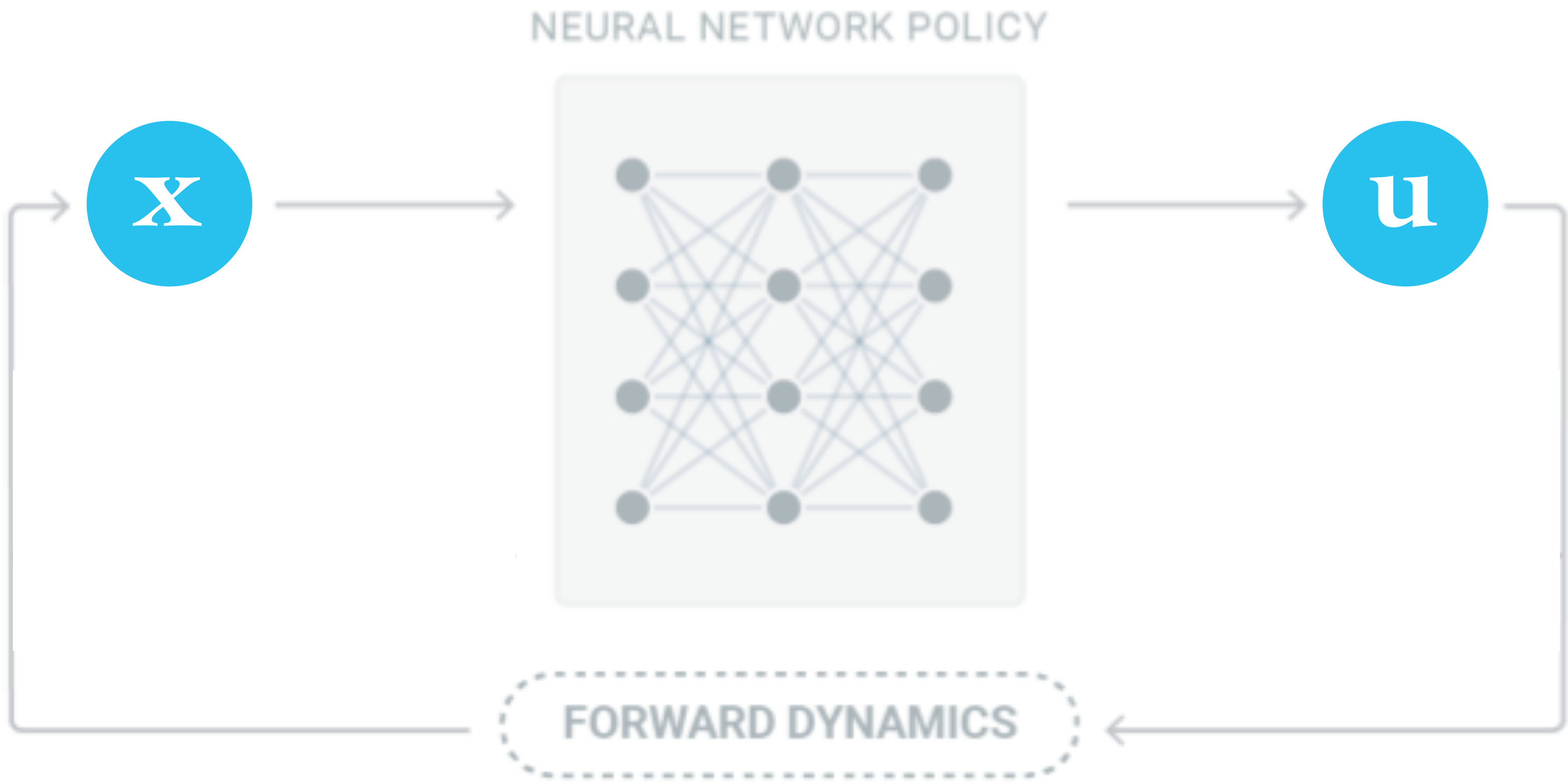


Learning Control Policies with Direct Collocation

Unknown/Uncertain Dynamics and Applications

Learning Control Policies

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



Learning Control Policies

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

Learning Control Policies

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

Forward Shooting:

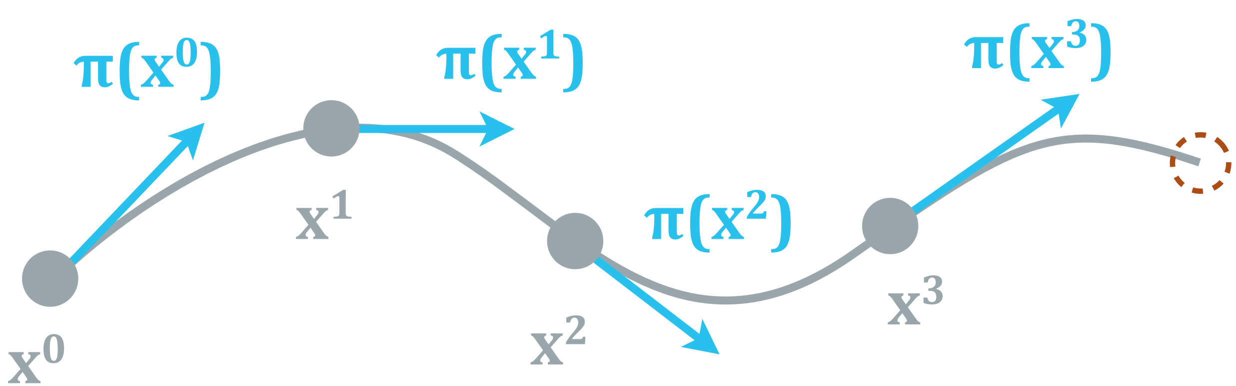
$$\min_{\theta} \sum_t C^t(\mathbf{x}^t), \quad \mathbf{x}^{t+1} = f(\mathbf{x}^t, \boldsymbol{\pi}_\theta(\mathbf{x}^t))$$

Learning Control Policies

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

Forward Shooting:

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

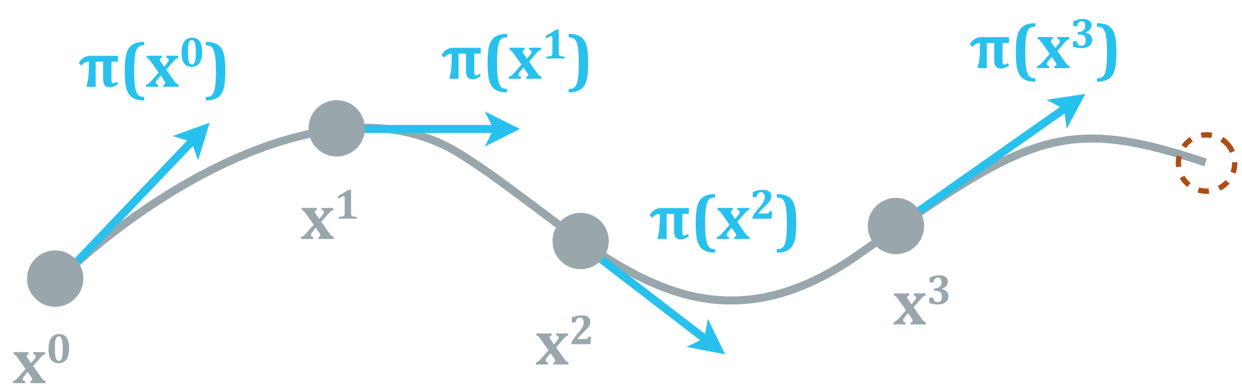


Learning Control Policies

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

Forward Shooting:

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



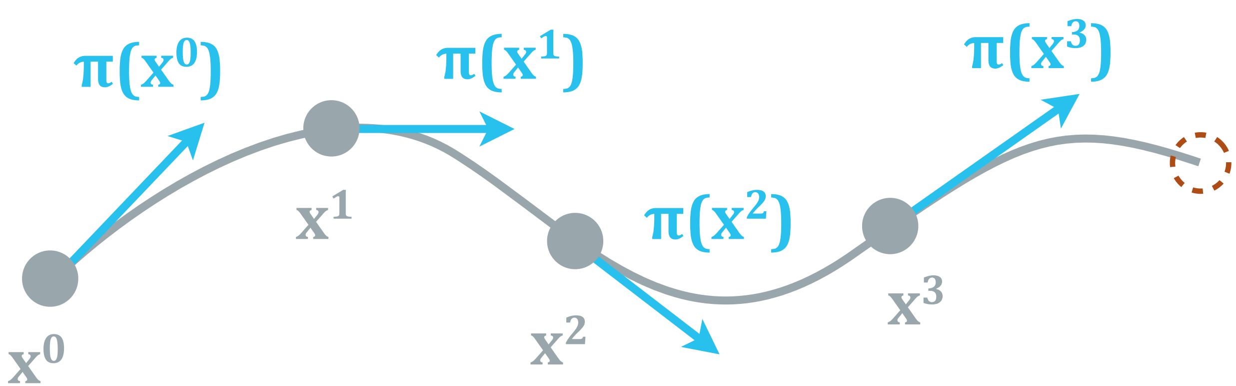
Learning Control Policies

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

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

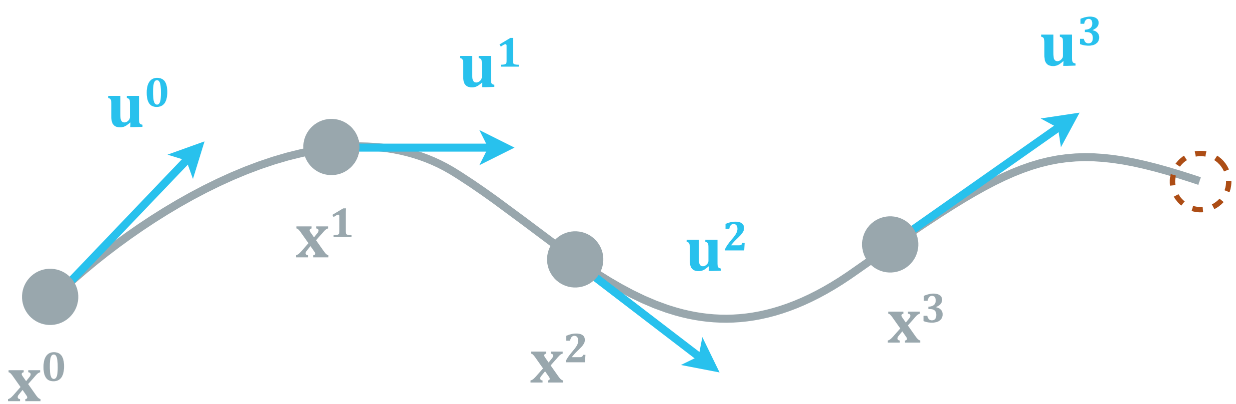


Learning Control Policies

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

Forward Shooting:

Learning from Demonstrations:



Learning Control Policies

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

Learning from Demonstrations:

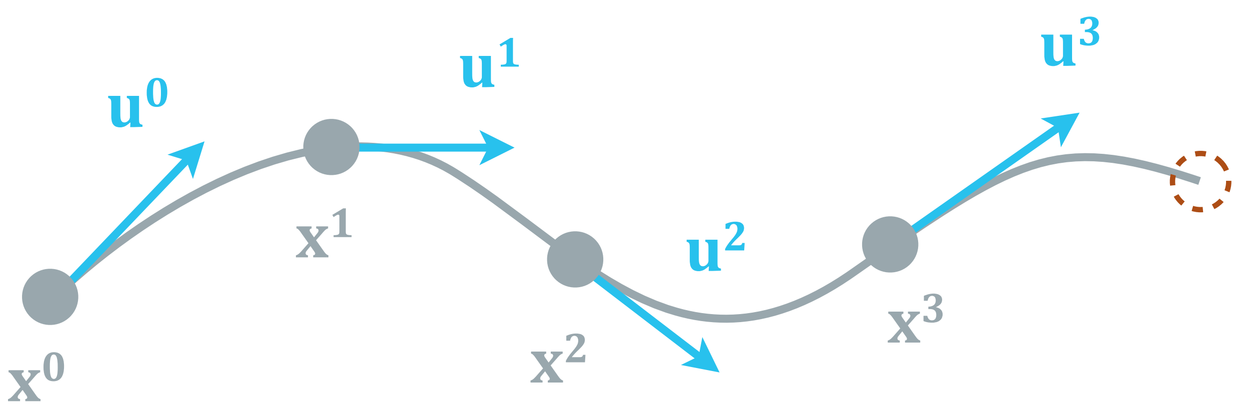
supervised learning

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

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

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

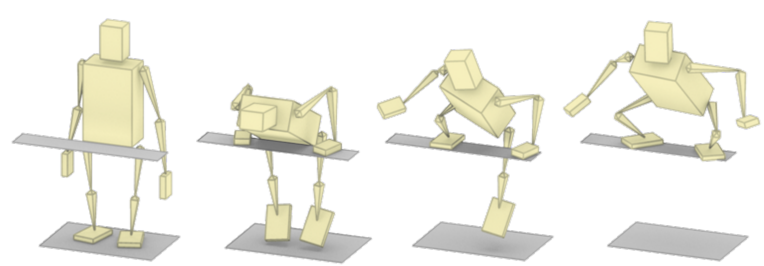
Training Data

input: $\mathbf{x}^i$

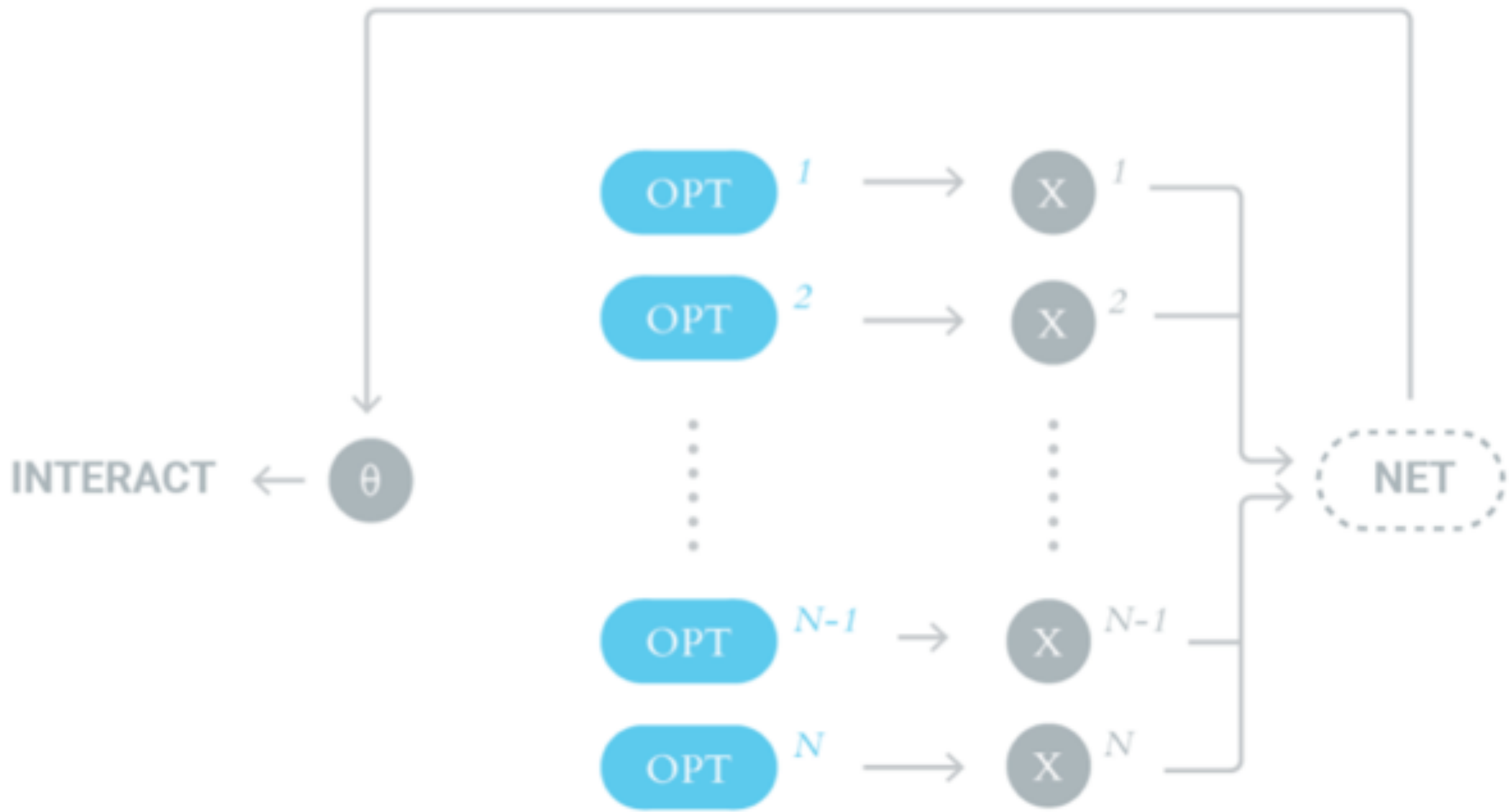
output: $\mathbf{u}^i$

Where does training data come from?

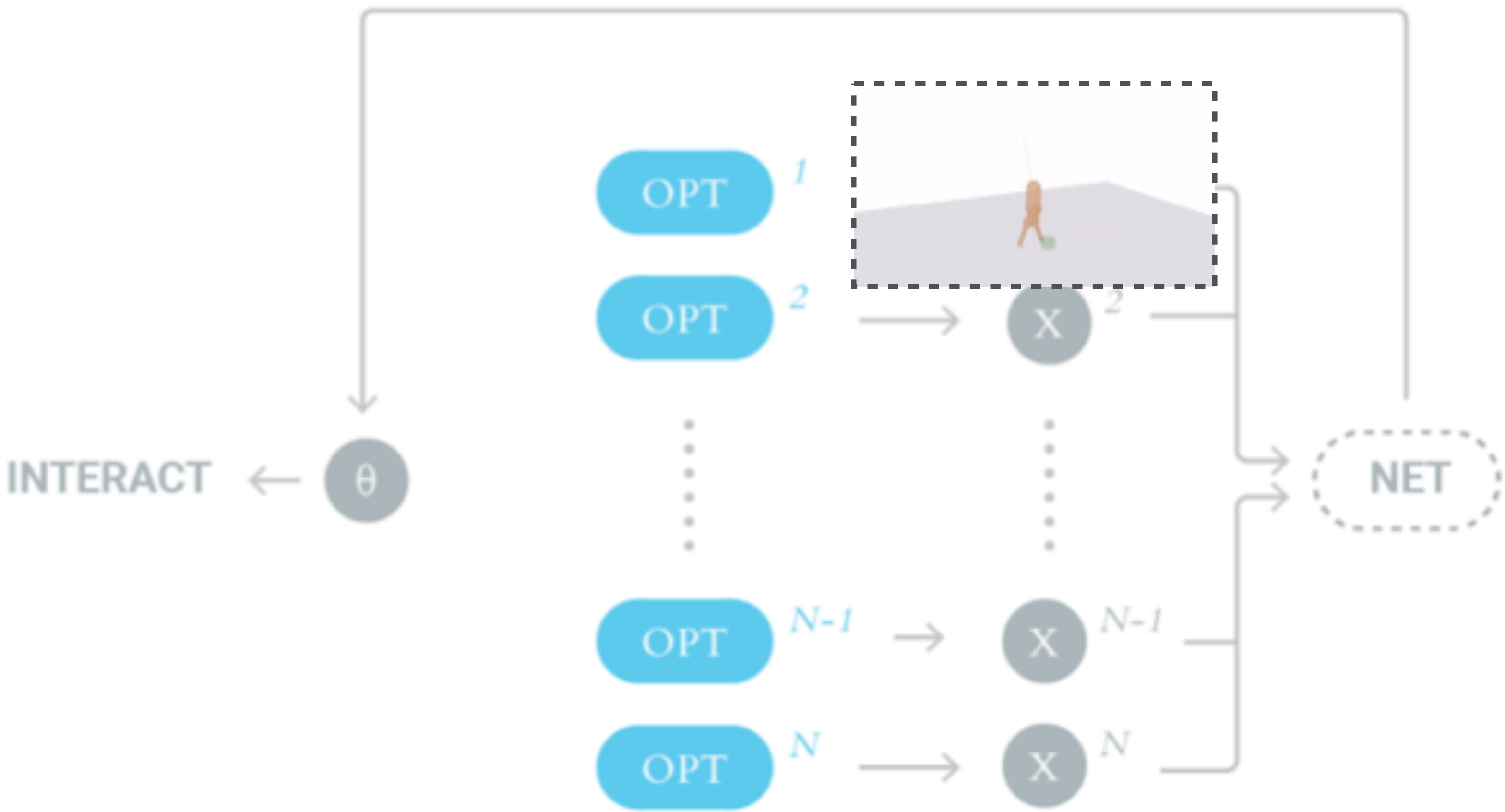
- Human demonstration
- Trajectory optimization



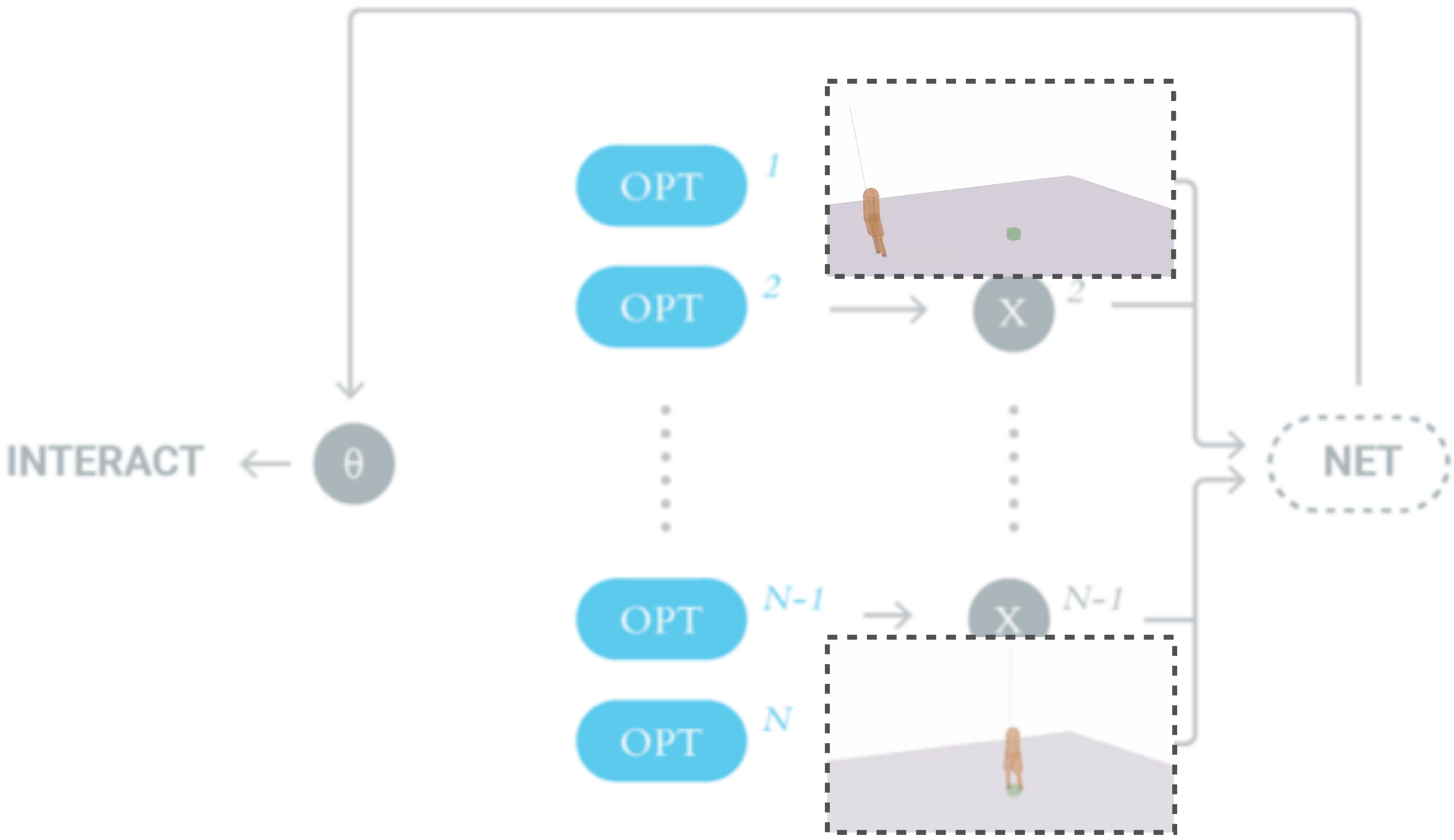
Learning Policies from Trajectory Optimization



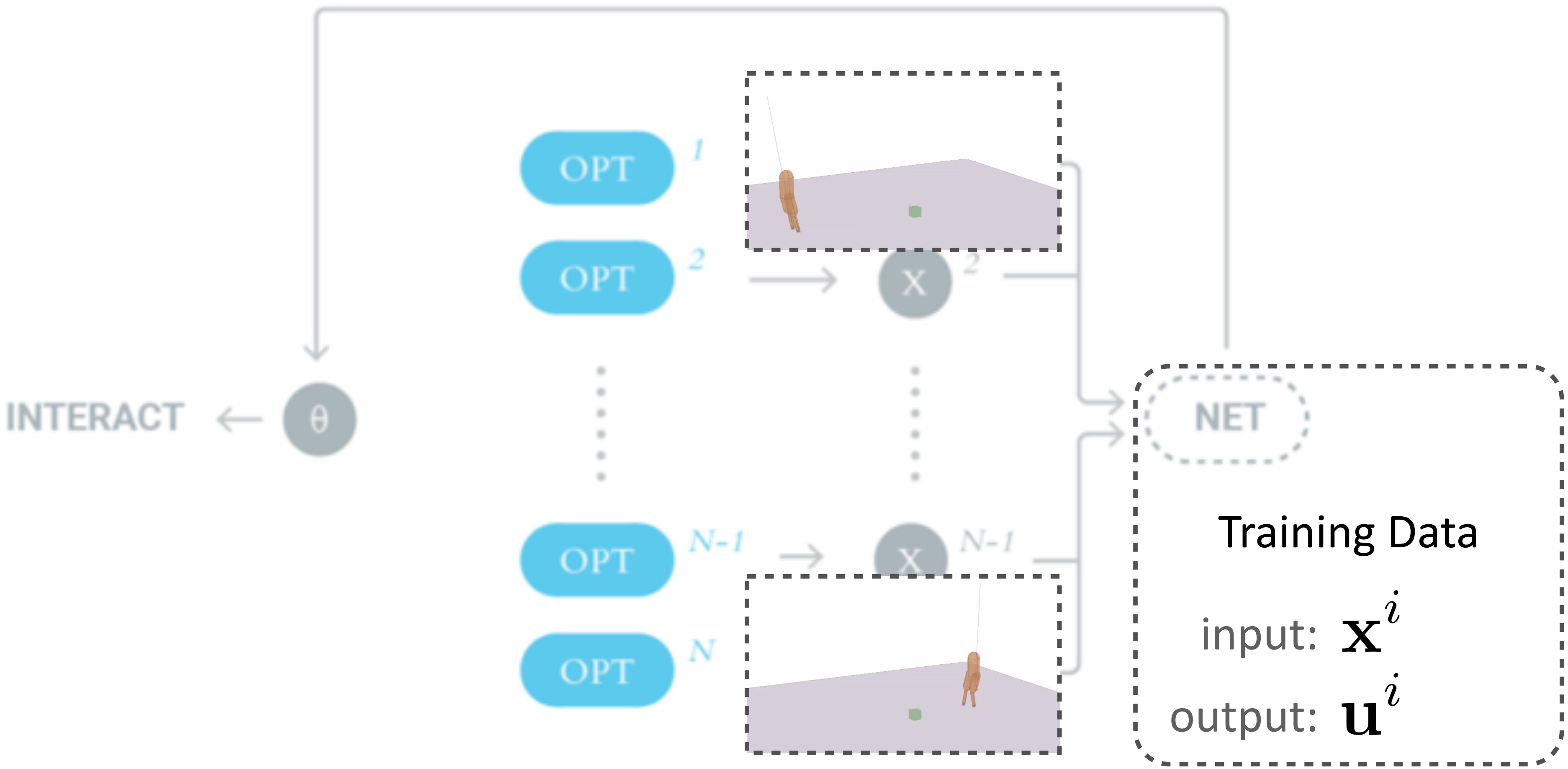
Learning Policies from Trajectory Optimization



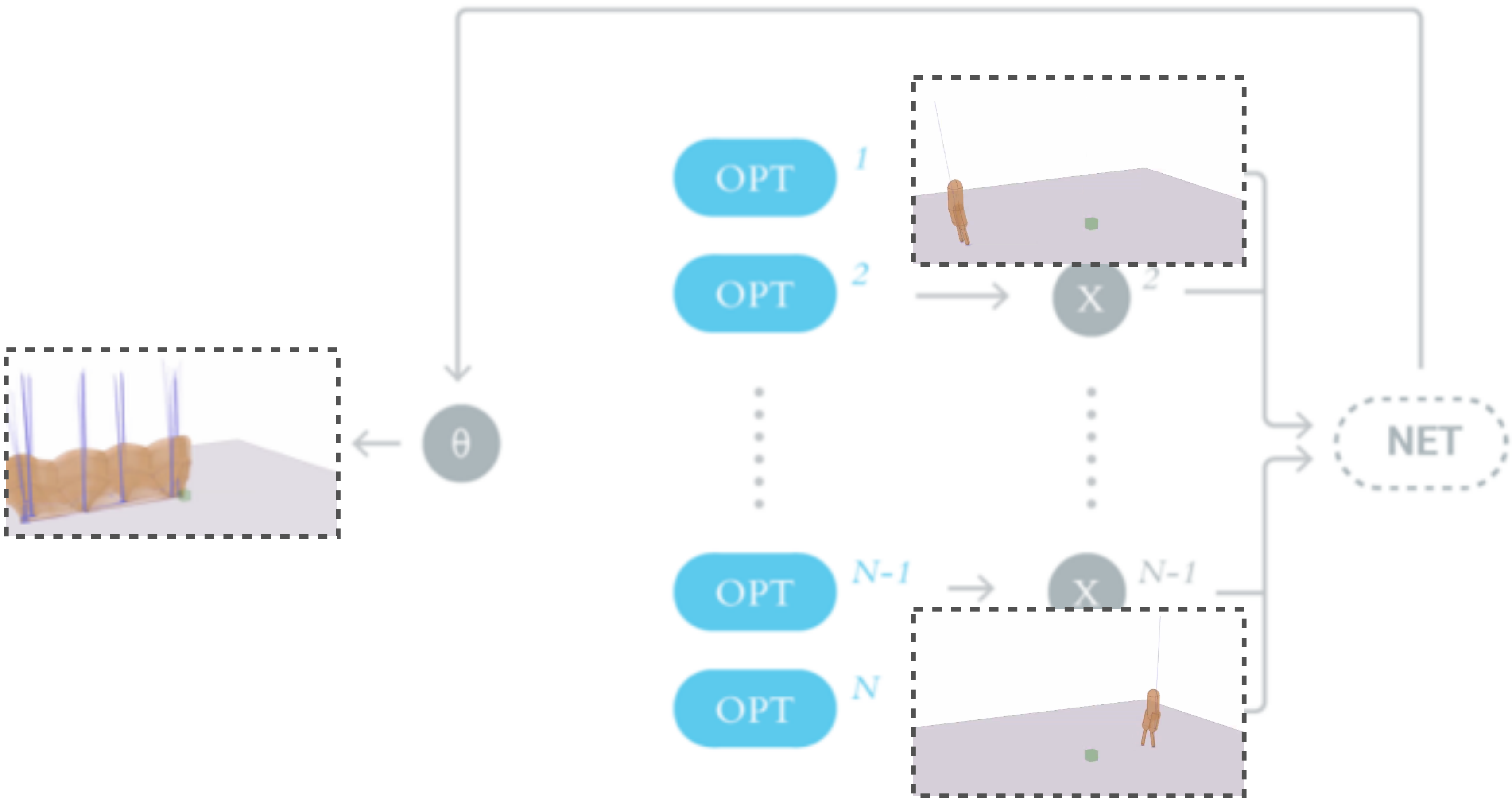
Learning Policies from Trajectory Optimization



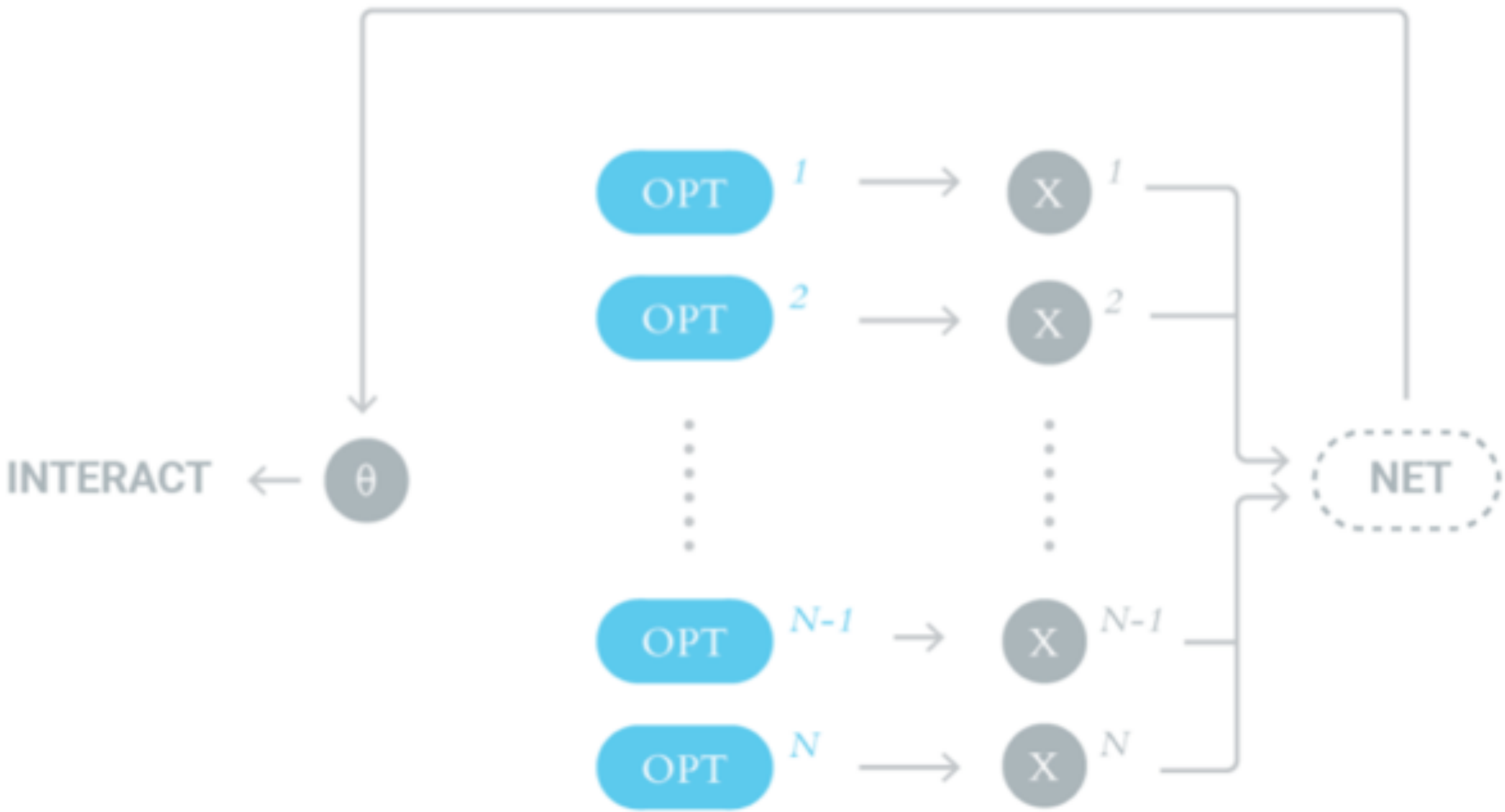
Learning Policies from Trajectory Optimization



Learning Policies from Trajectory Optimization

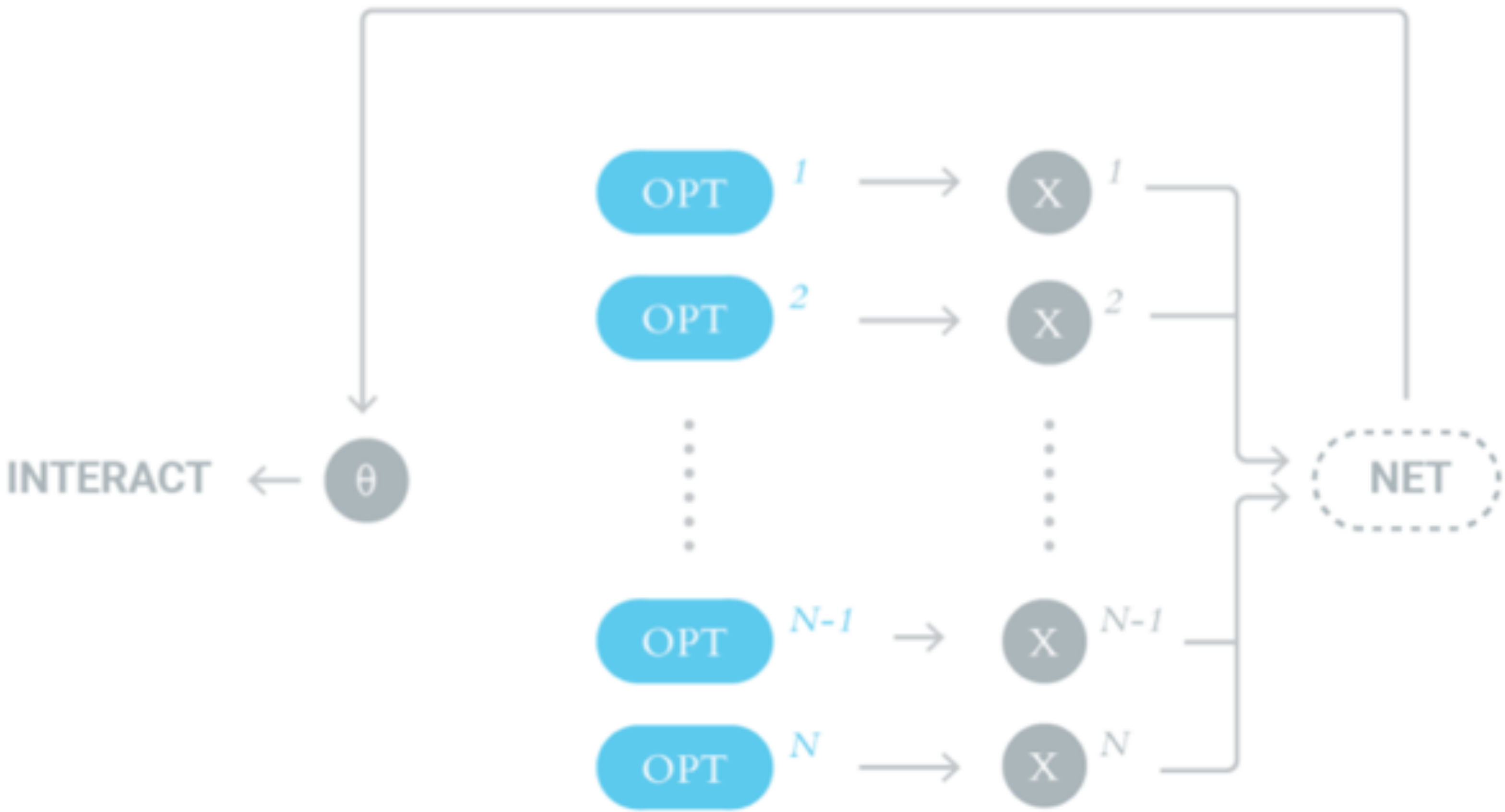


Learning Policies from Trajectory Optimization

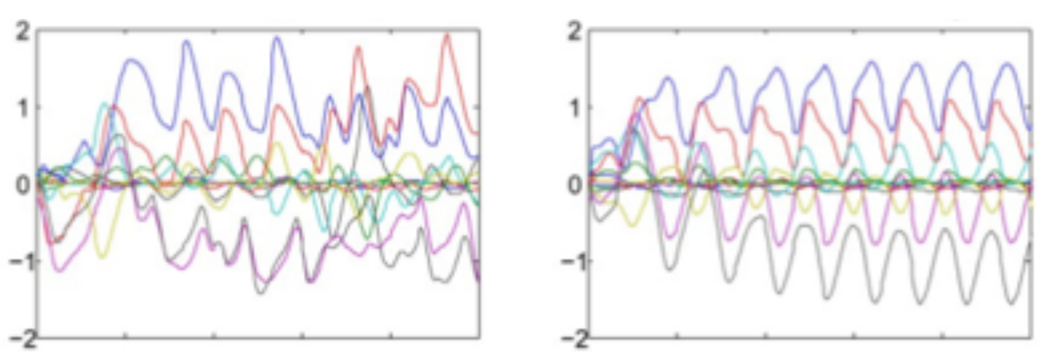


$\mathbf{X}^1, \ldots \mathbf{X}^N$ can be inconsistent or difficult to fit

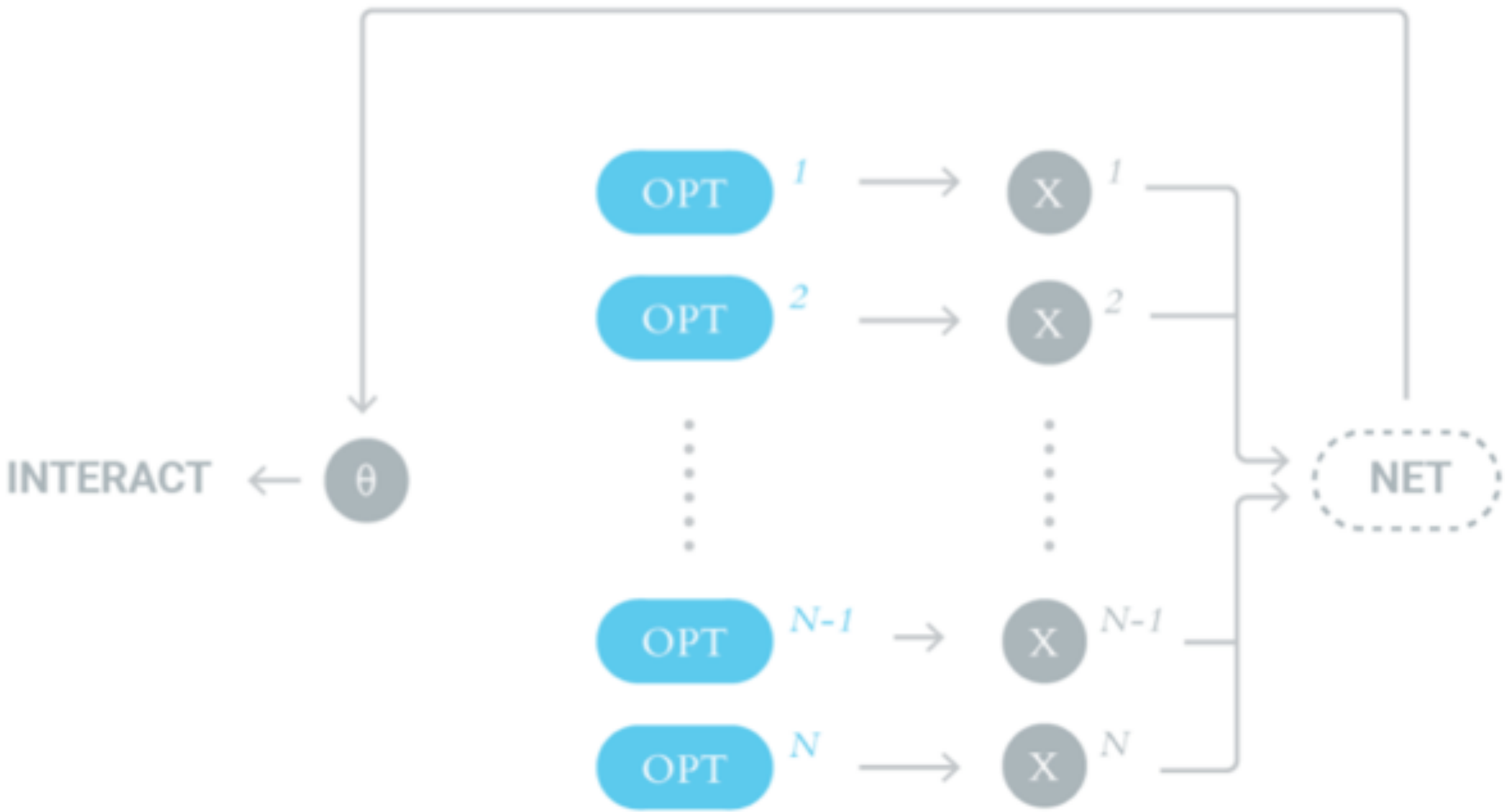
Learning Policies from Trajectory Optimization



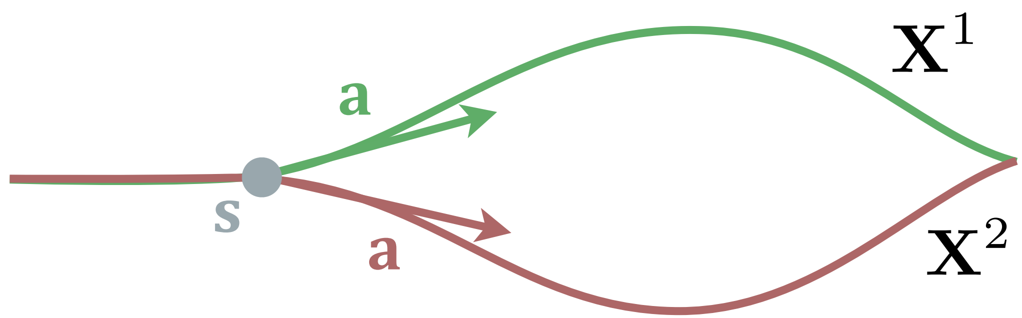
$\mathbf{X}^1, \ldots\ \mathbf{X}^N$ can be inconsistent or difficult to fit



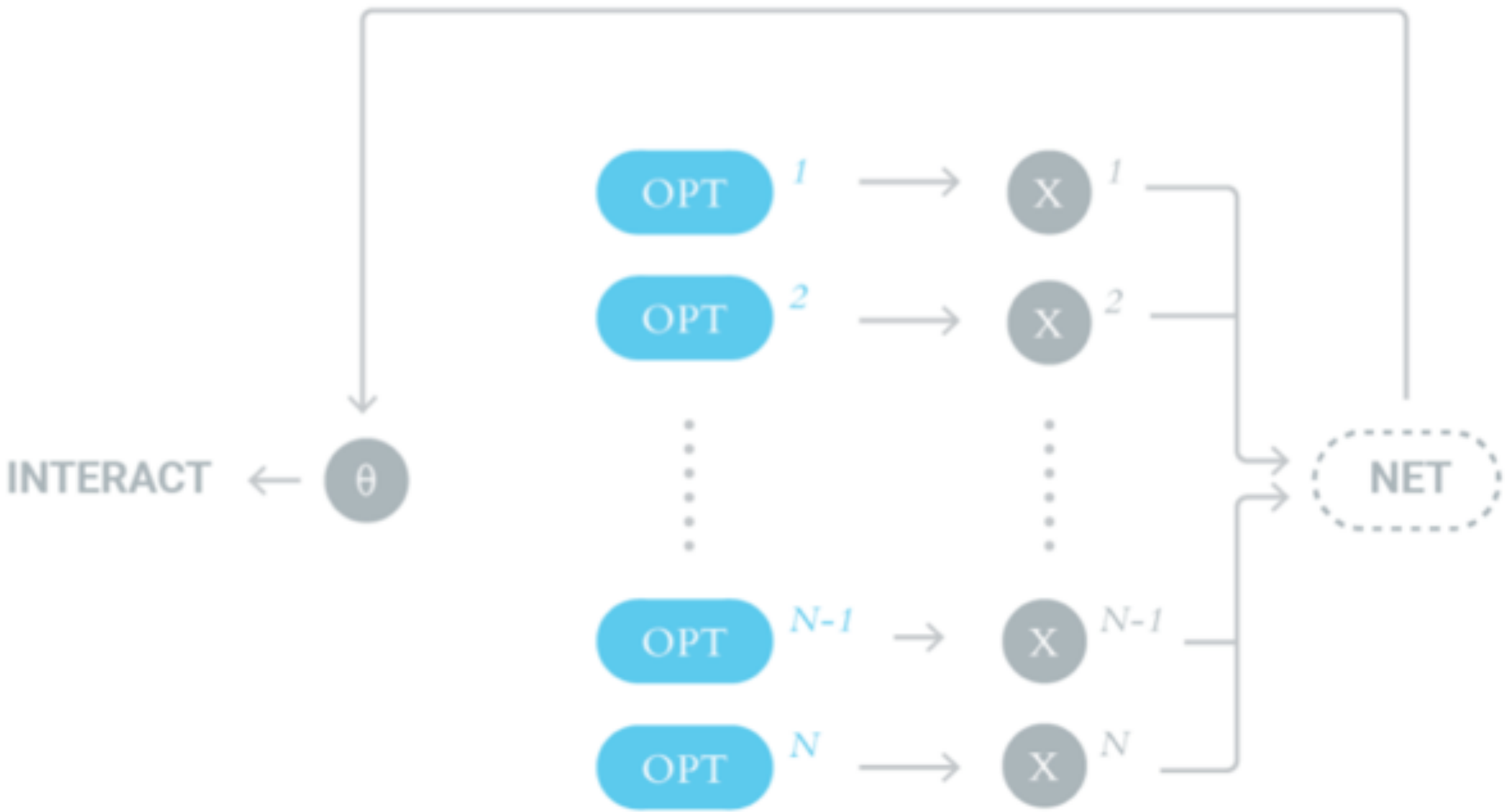
Learning Policies from Trajectory Optimization



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



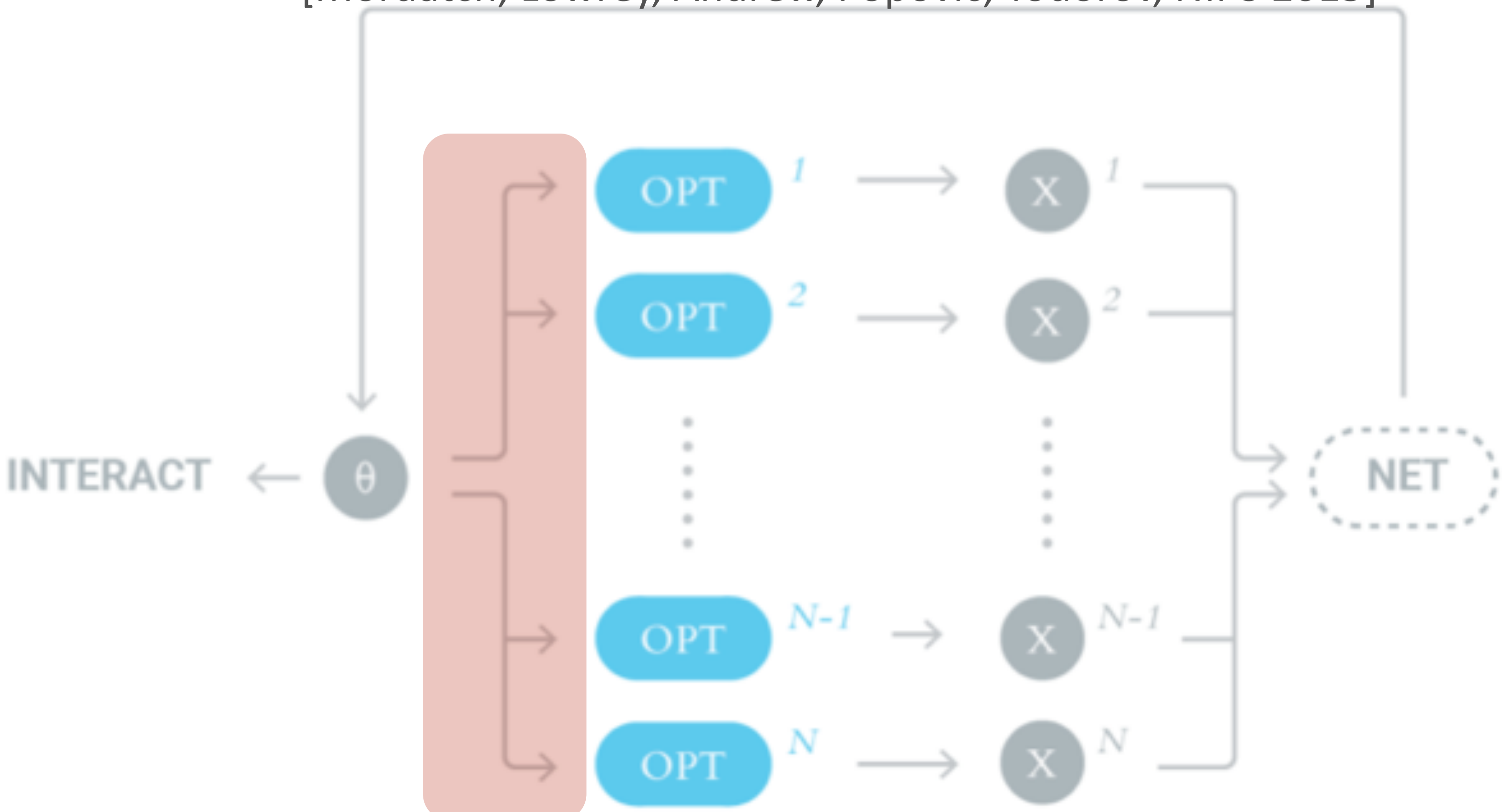
Learning Policies from Trajectory Optimization



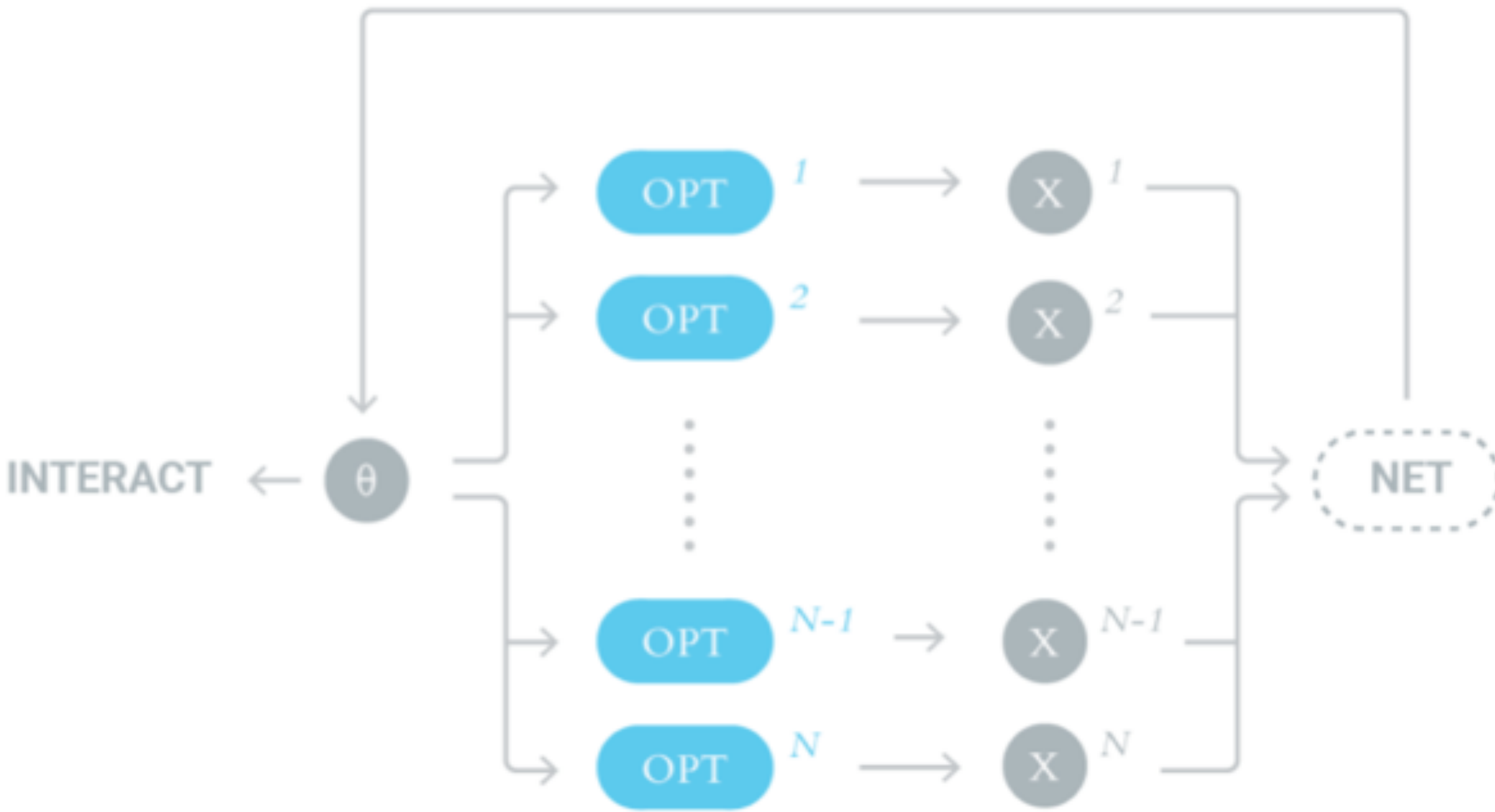
Joint Policy and Trajectory Optimization

[Mordatch, Todorov, RSS 2014]

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

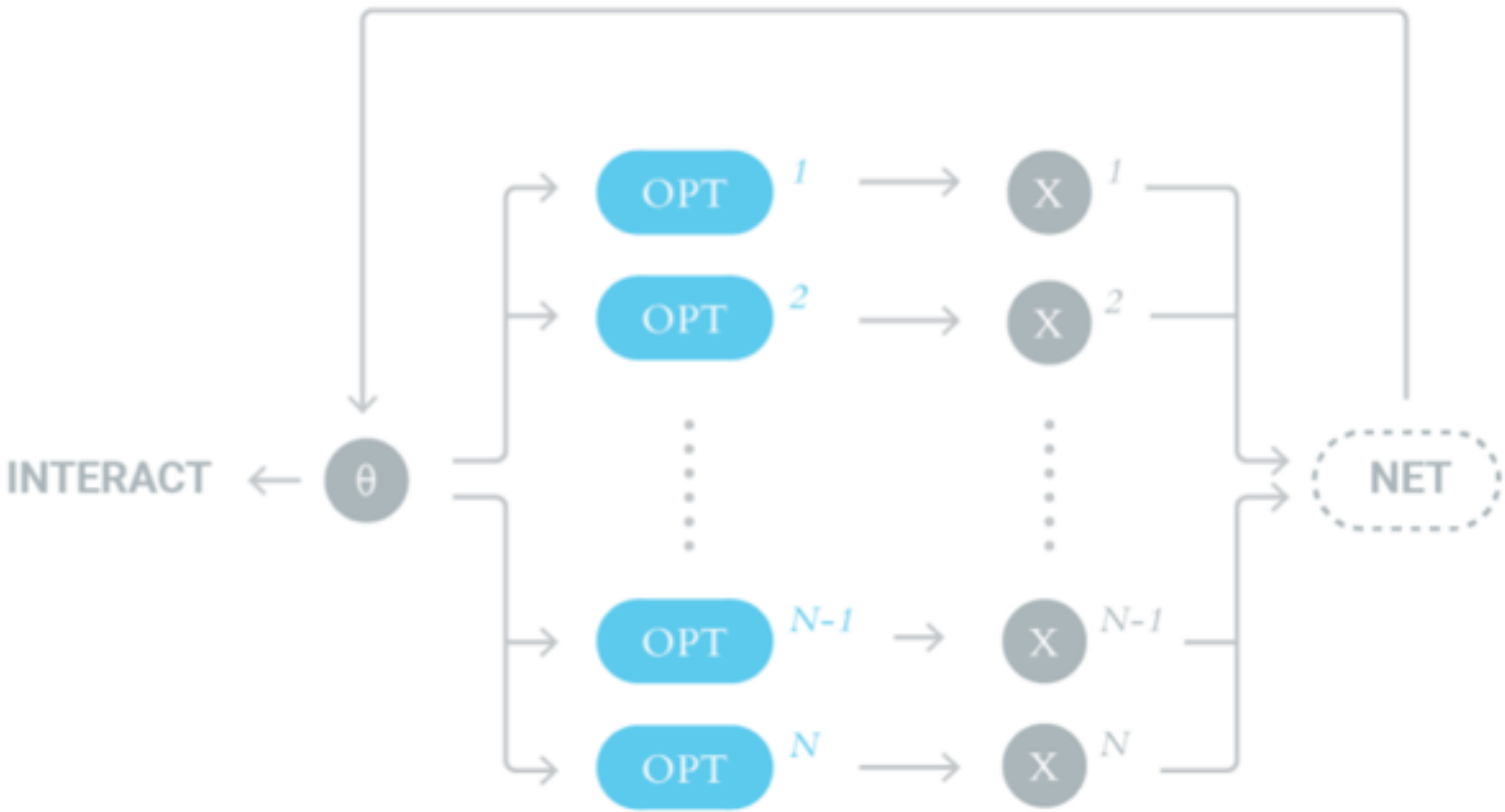


Joint Policy and Trajectory Optimization



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

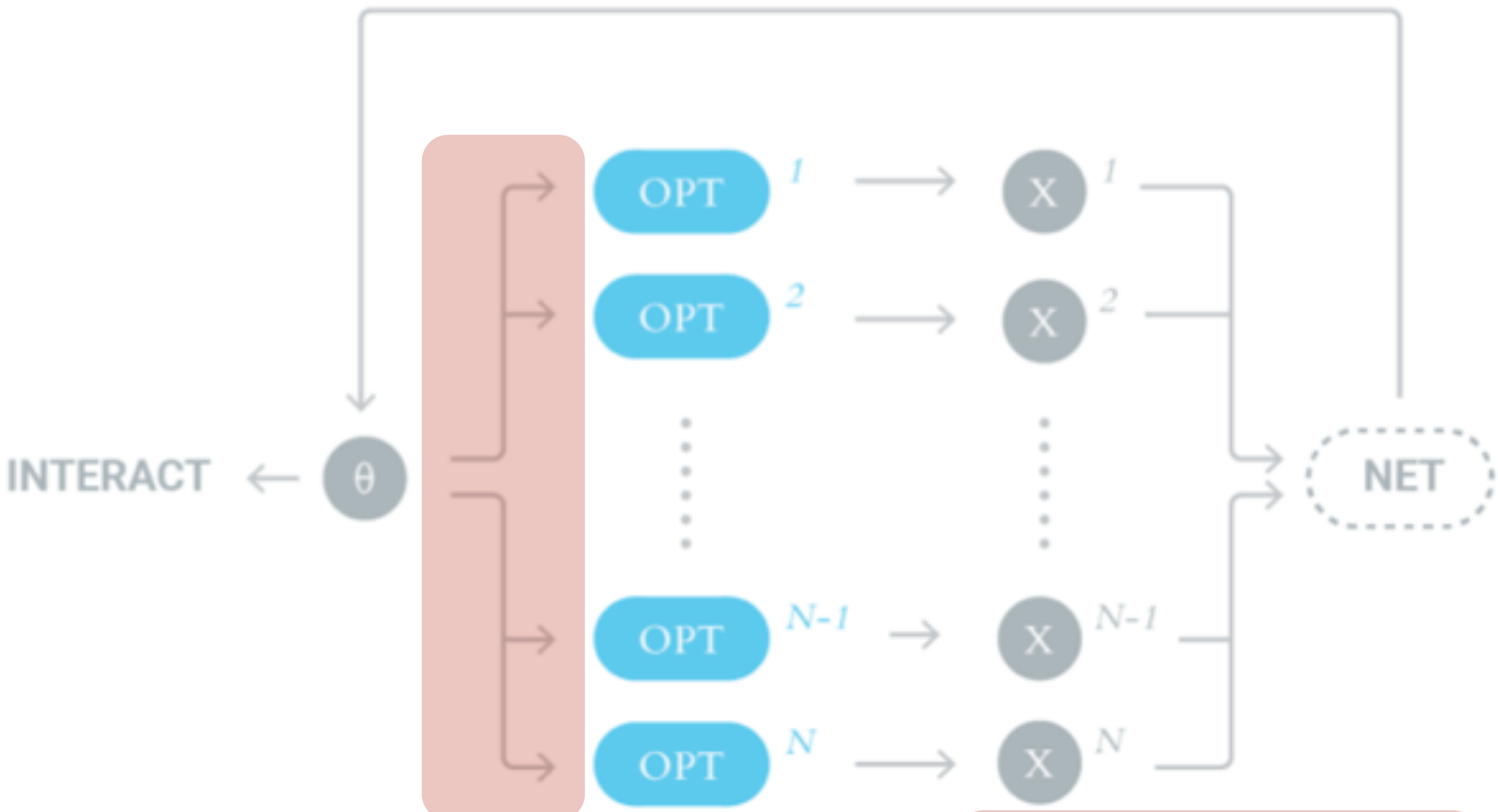
Joint Policy and Trajectory Optimization



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

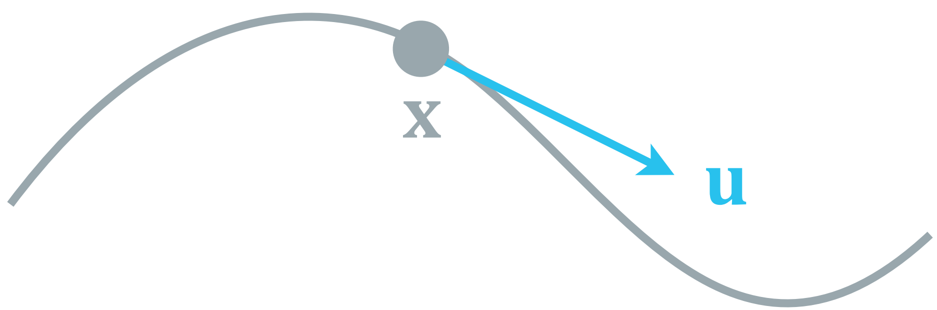
traditional trajectory optimization

Joint Policy and Trajectory Optimization



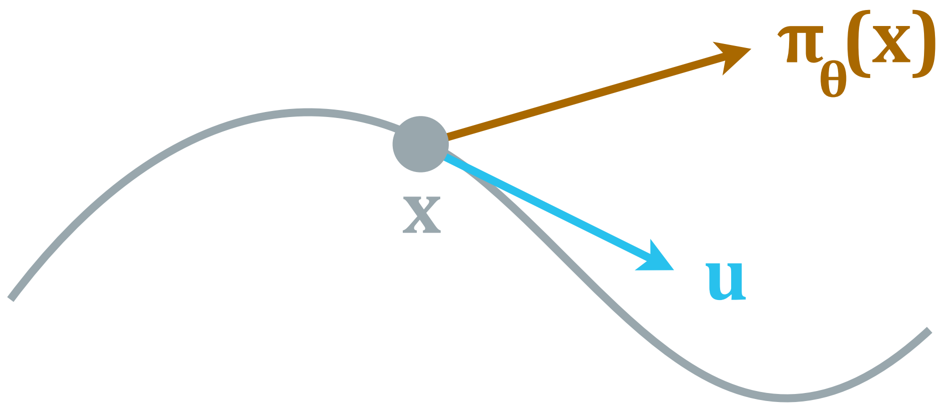
$$\min_{\theta\ \mathbf{X}^1\ \ldots\ \mathbf{X}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + \boxed{||\boldsymbol{\pi}_\theta(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2}$$

Joint Policy and Trajectory Optimization



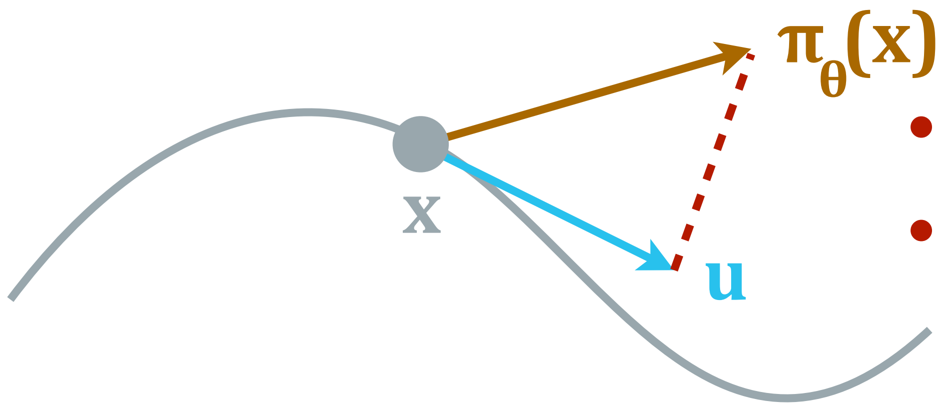
$$\min_{\theta\ \mathbf{x}^1 \dots \mathbf{x}^N} \sum_{i,t} C(\mathbf{x}^{i,t}) + ||\boldsymbol{\pi}_\theta(\mathbf{x}^{i,t}) - \boxed{\mathbf{u}^{i,t}}||^2$$

Joint Policy and Trajectory Optimization



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

Joint Policy and Trajectory Optimization



- Adjust policy
- Adjust trajectory

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

Idea

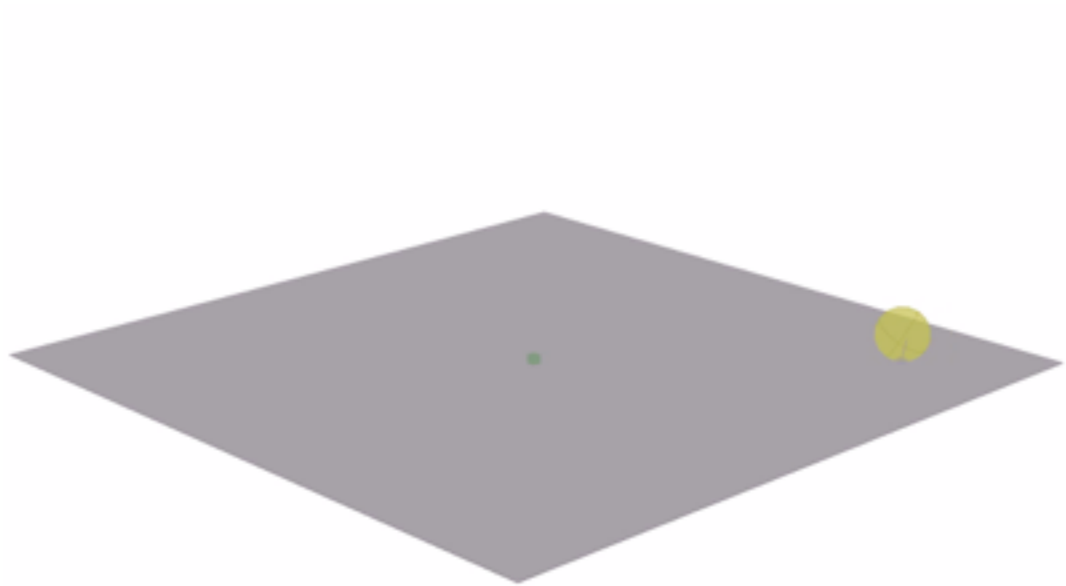
Add auxiliary variables

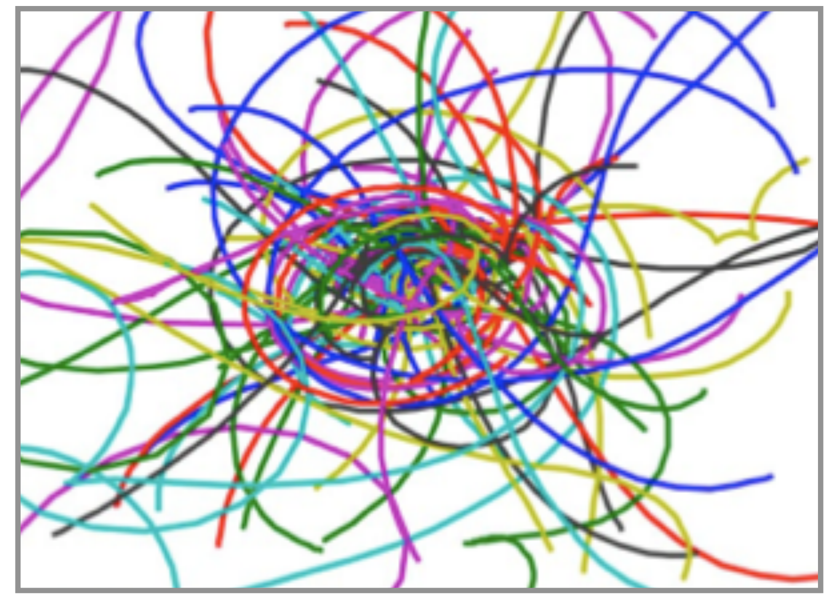
Softly enforce consistency between variables

Search in larger, but easier to explore space

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

Policy still not converging?





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

Policy Drift

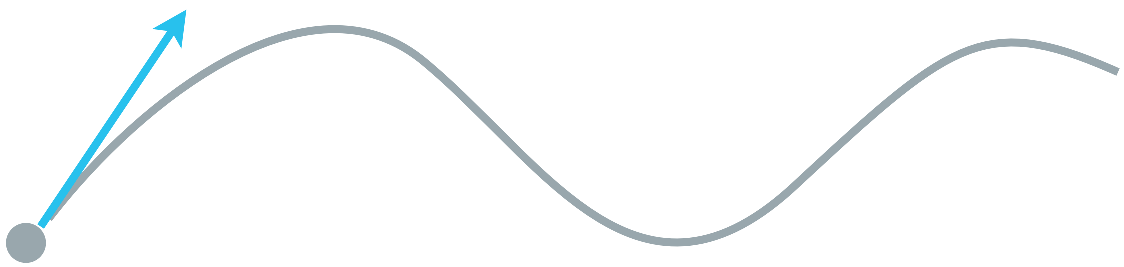
Traditional supervised learning at test time: independent errors

Policy at test time:

Policy Drift

Traditional supervised learning at test time: independent errors

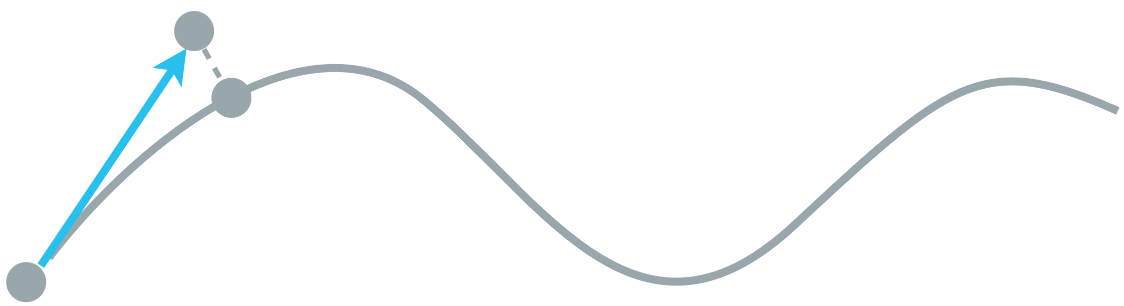
Policy at test time:



Policy Drift

Traditional supervised learning at test time: independent errors

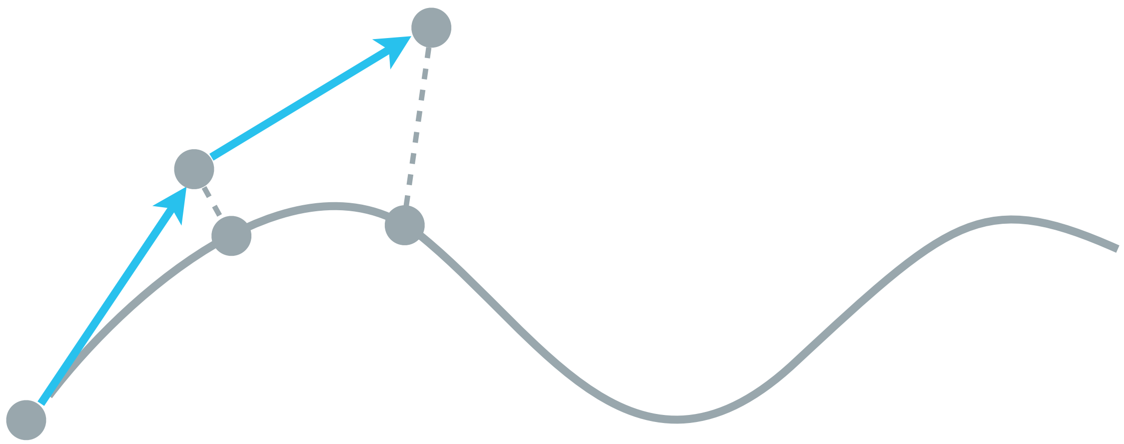
Policy at test time:



Policy Drift

Traditional supervised learning at test time: independent errors

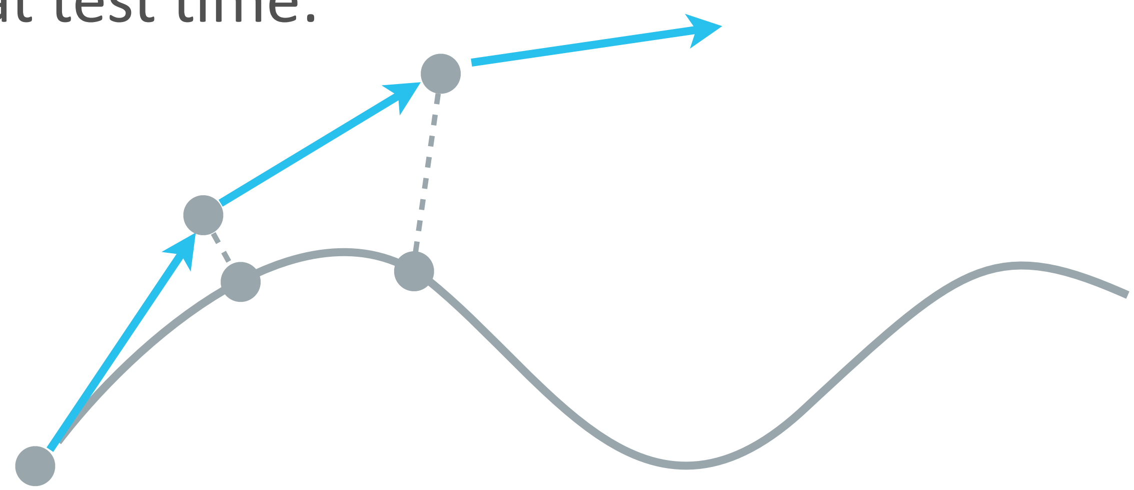
Policy at test time:



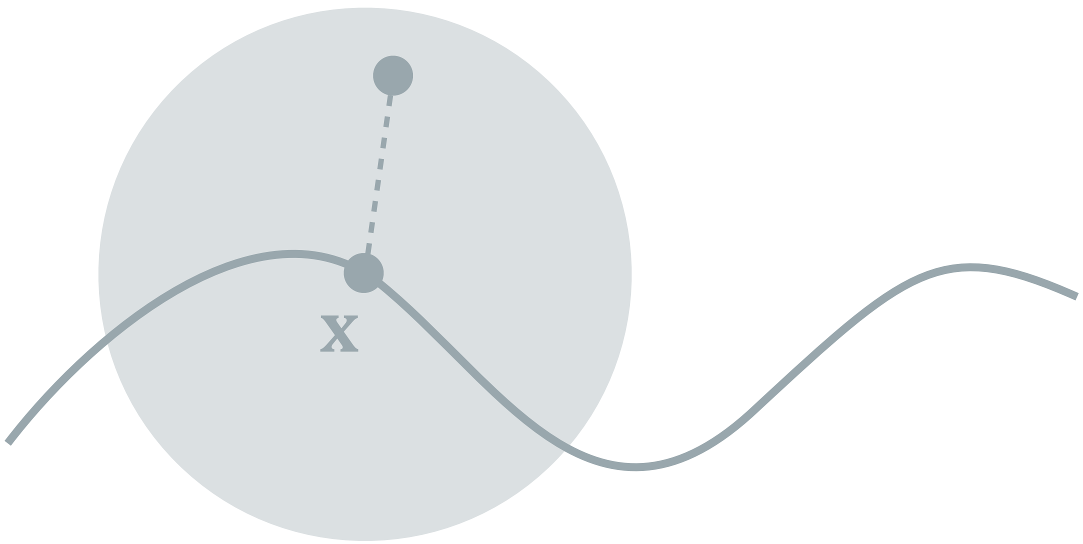
Policy Drift

Traditional supervised learning at test time: independent errors

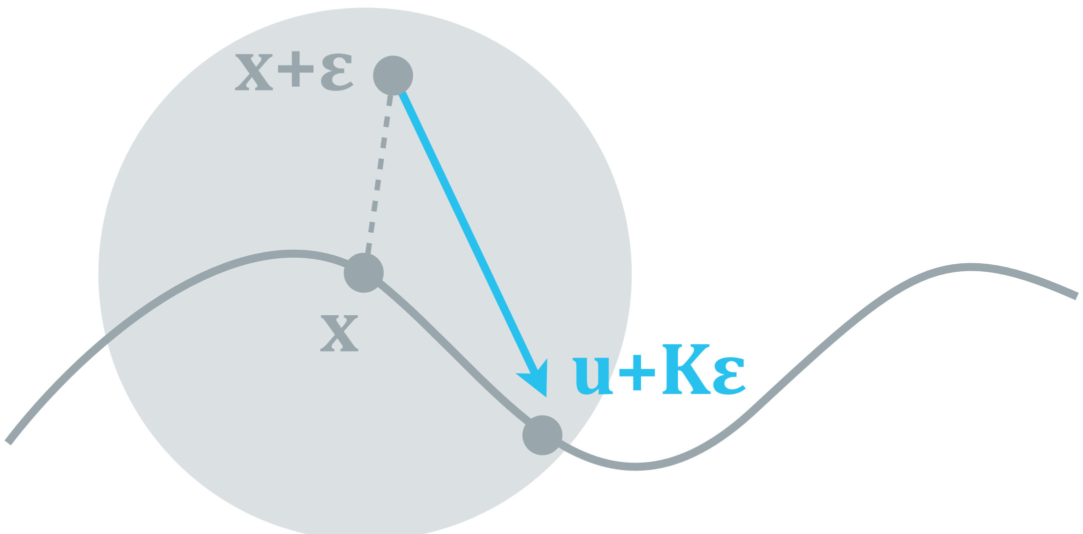
Policy at test time:



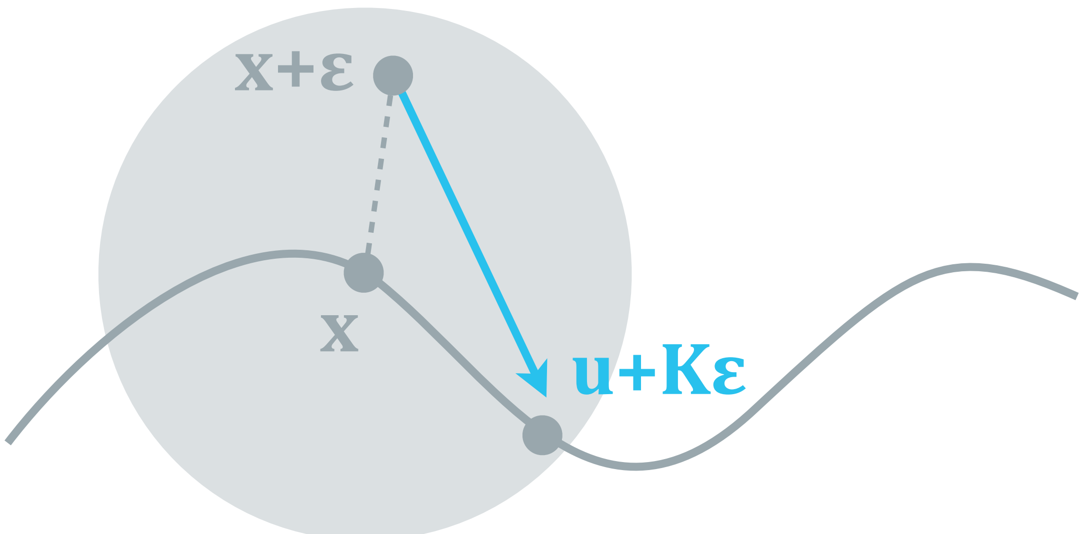
Policy Drift



Policy Drift



Injecting Network Noise

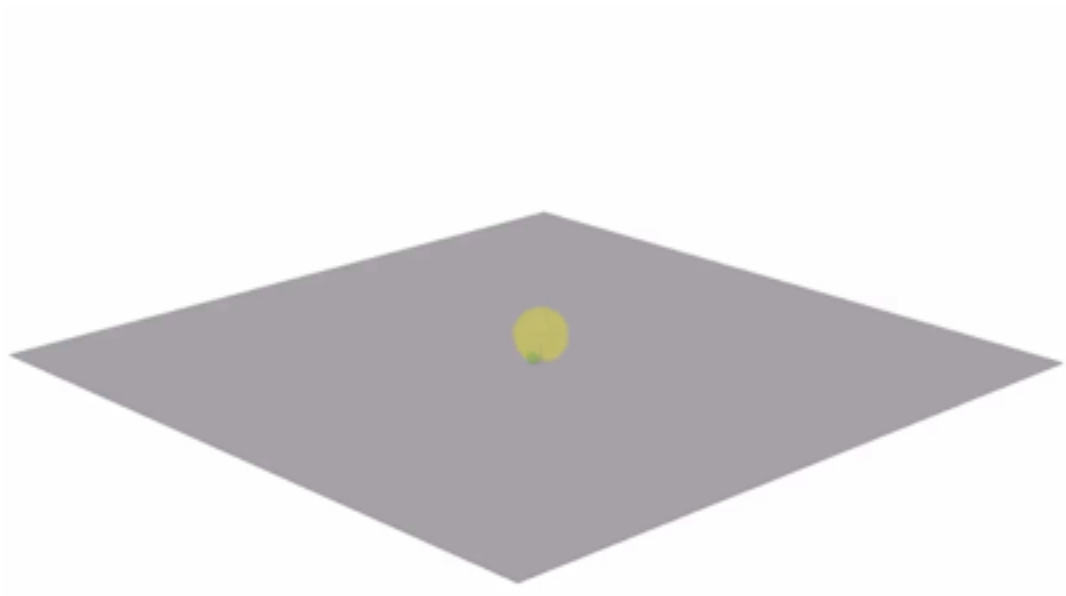


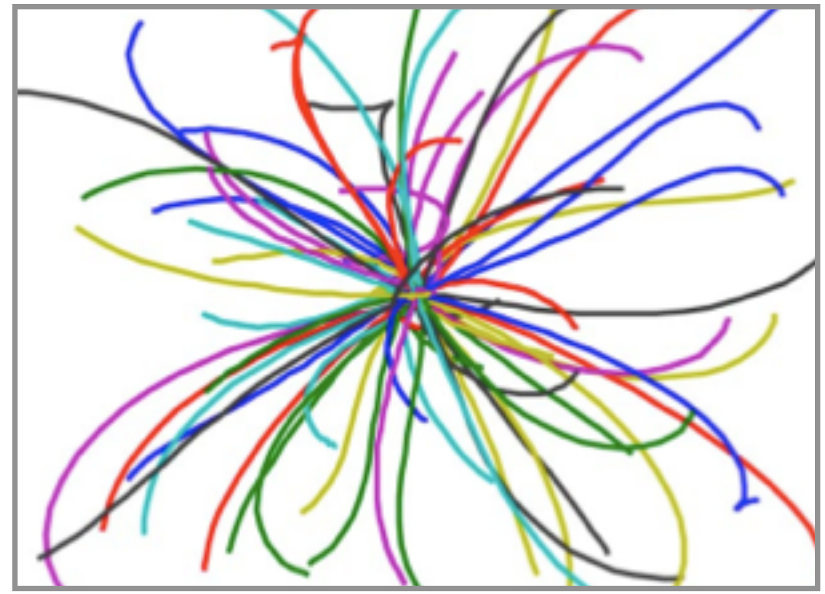
Noisy Training Data

input: $\mathbf{x} + \varepsilon$

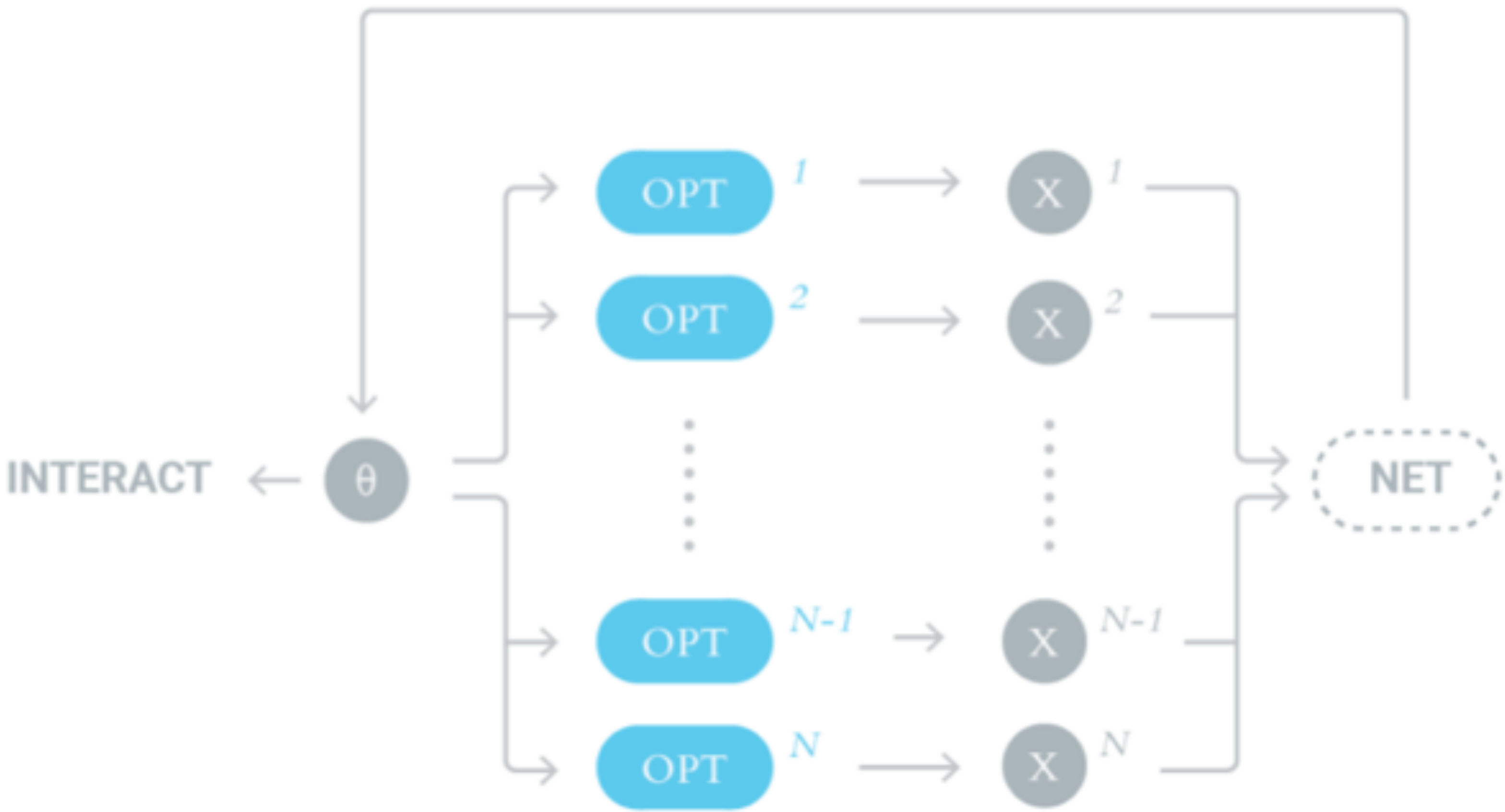
output: $\mathbf{u} + \mathbf{K}\varepsilon$

Injecting Network Noise





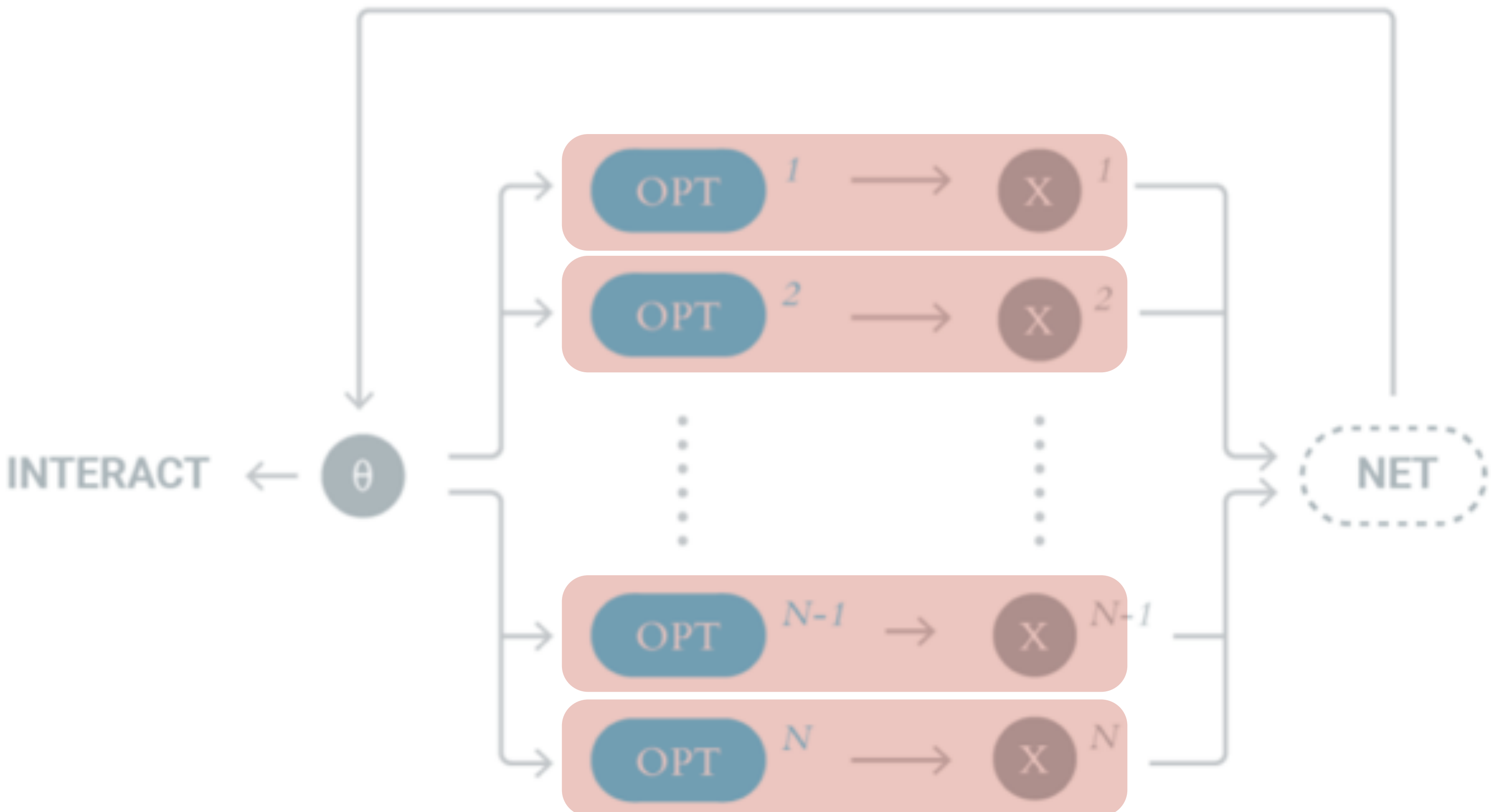
Alternating Optimization



Decompose into:

- trajectory optimizations
- regression

Alternating Optimization



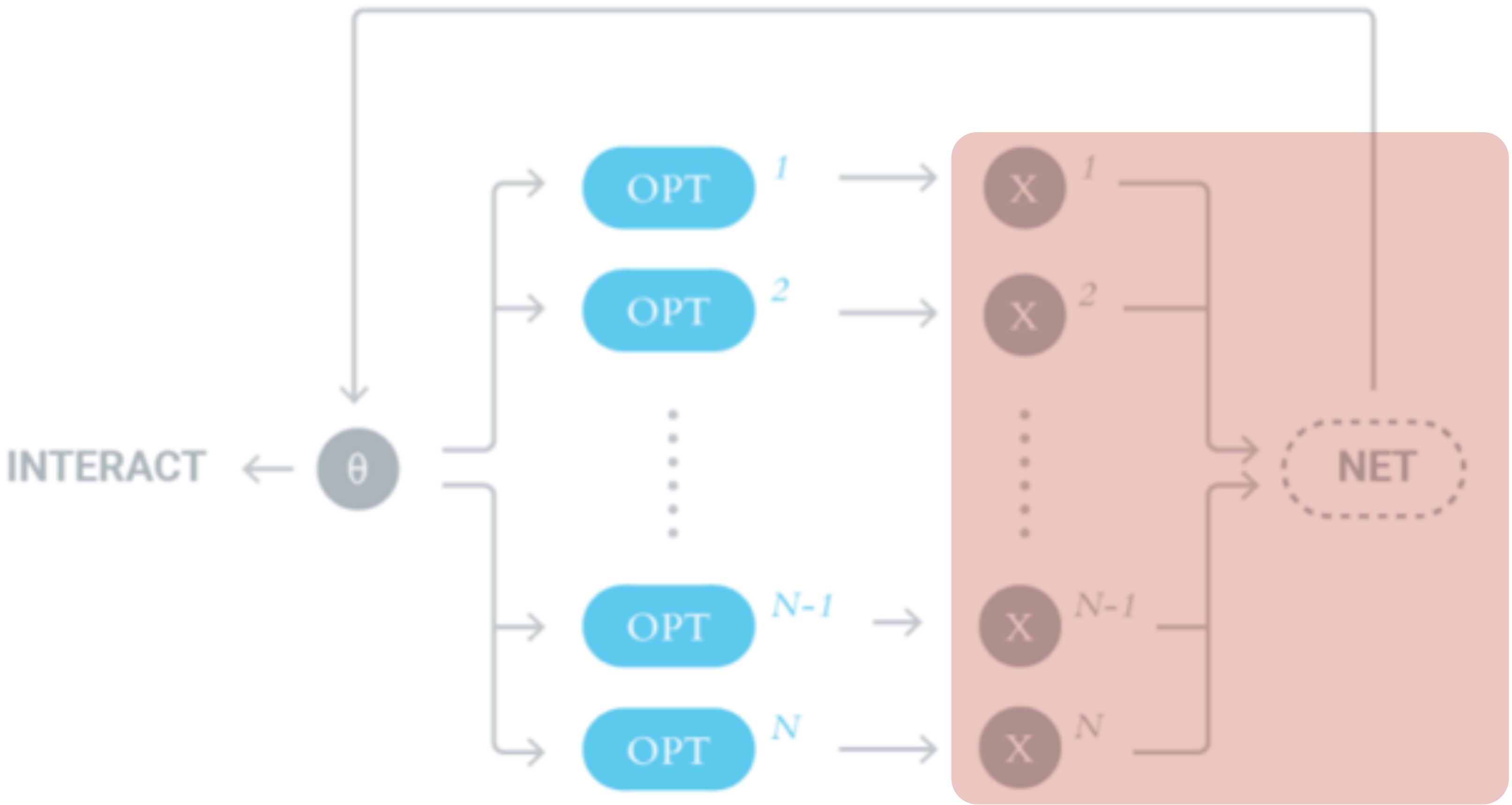
Decompose into:

- trajectory optimizations
- regression

“stay close to policy”

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

Alternating Optimization

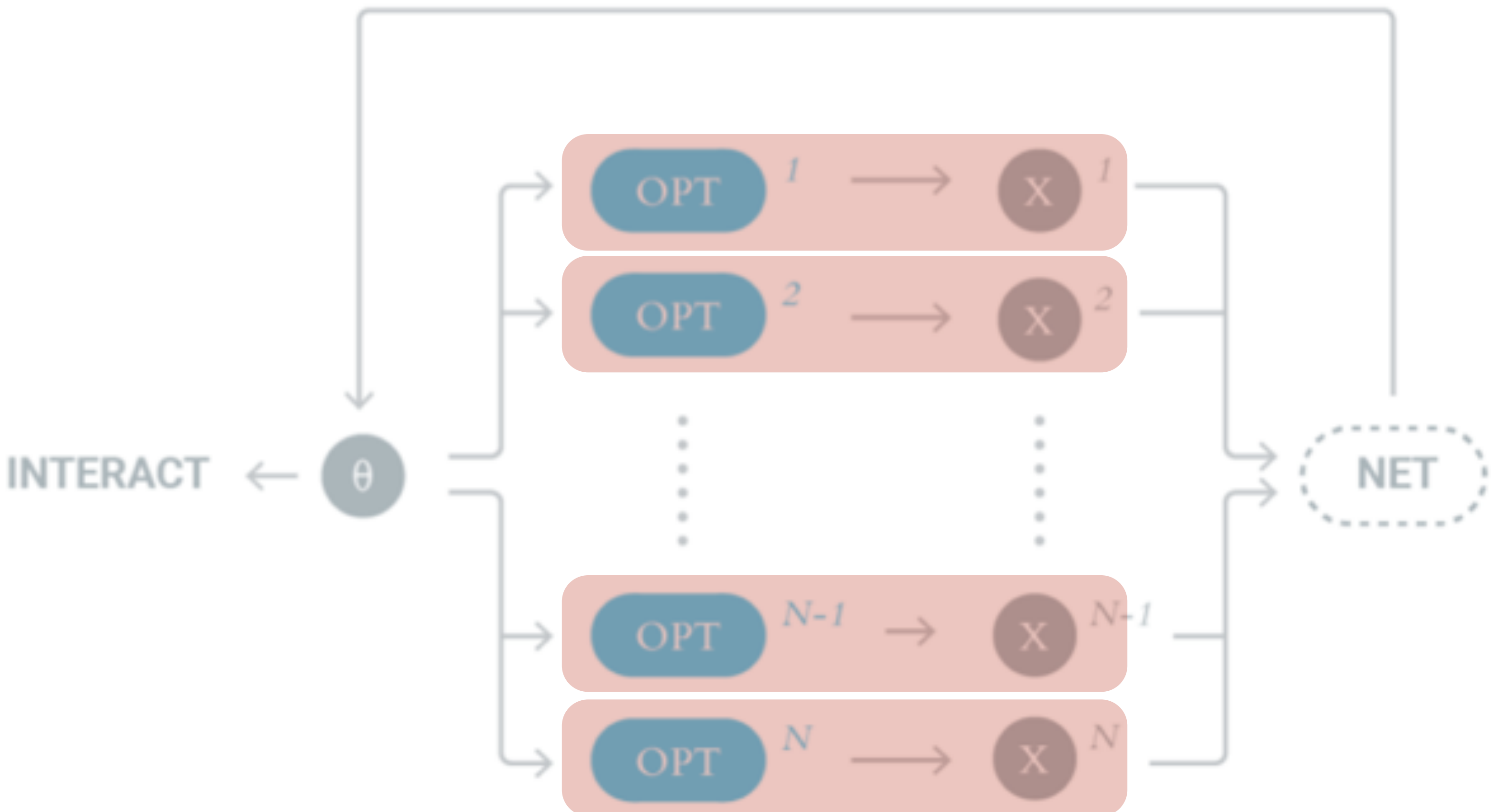


Decompose into:

- trajectory optimizations
- regression

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

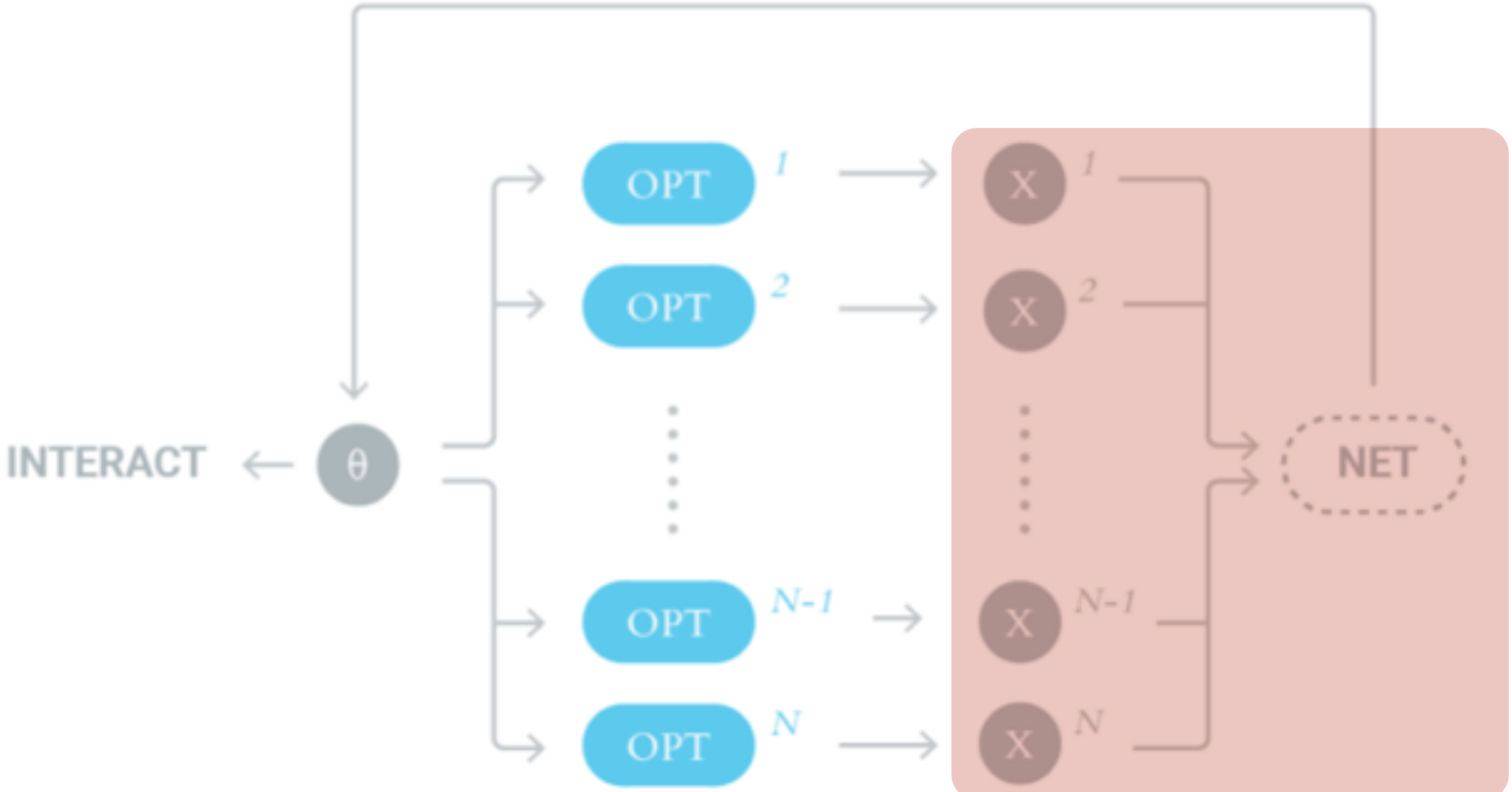
Alternating Optimization



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

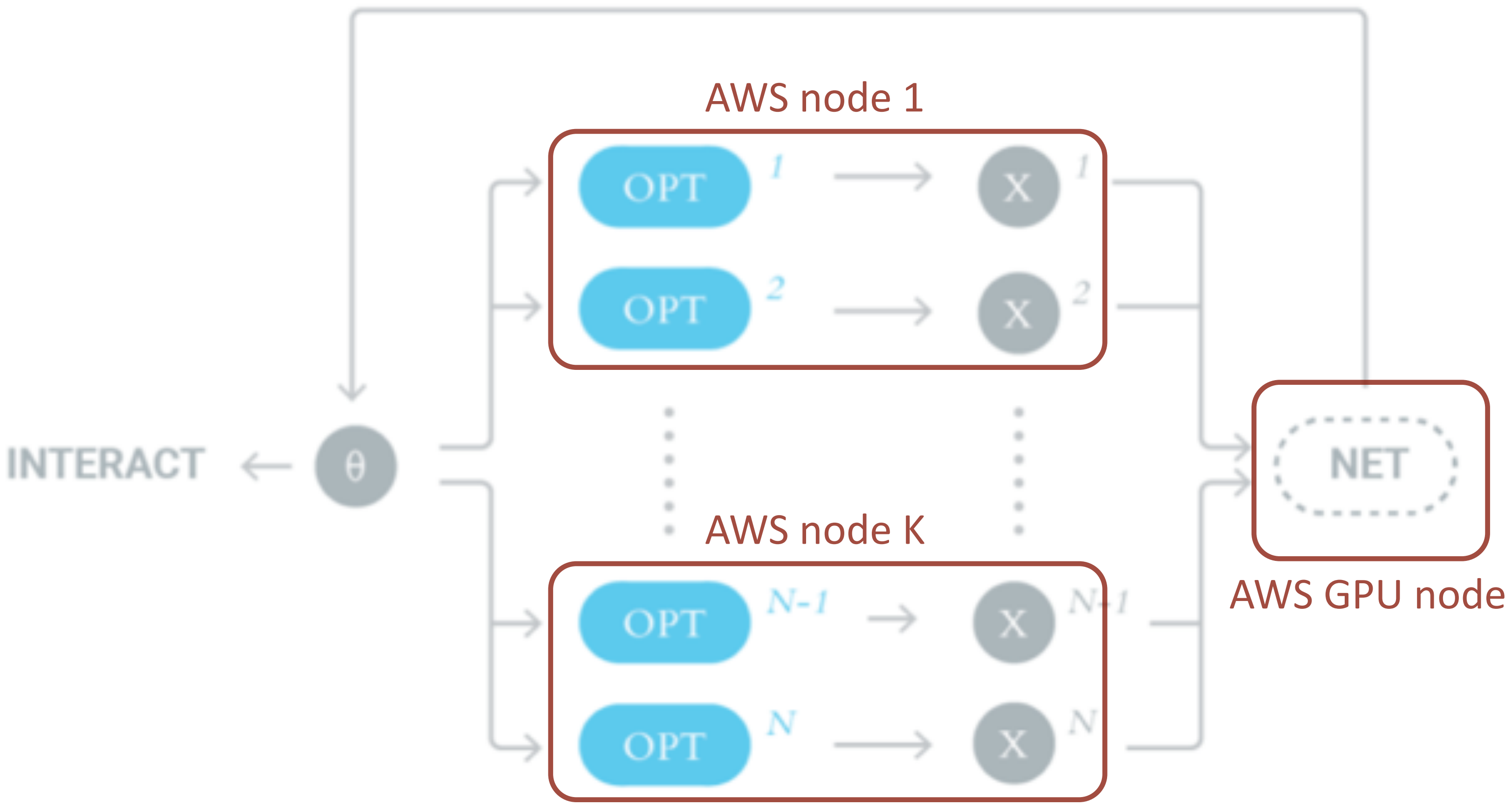


Decompose into:

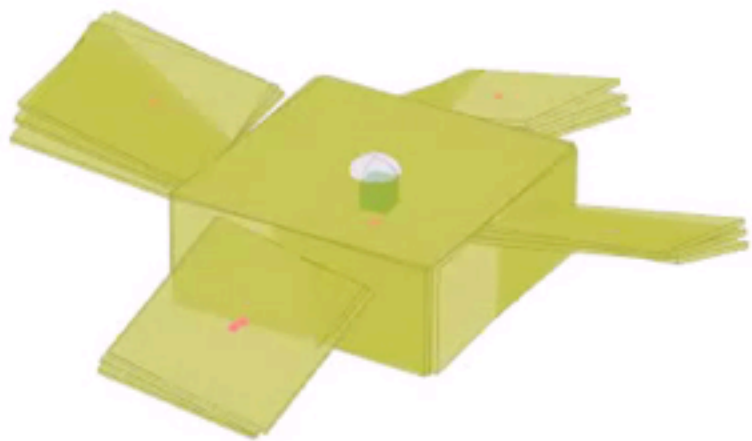
- trajectory optimizations
- regression

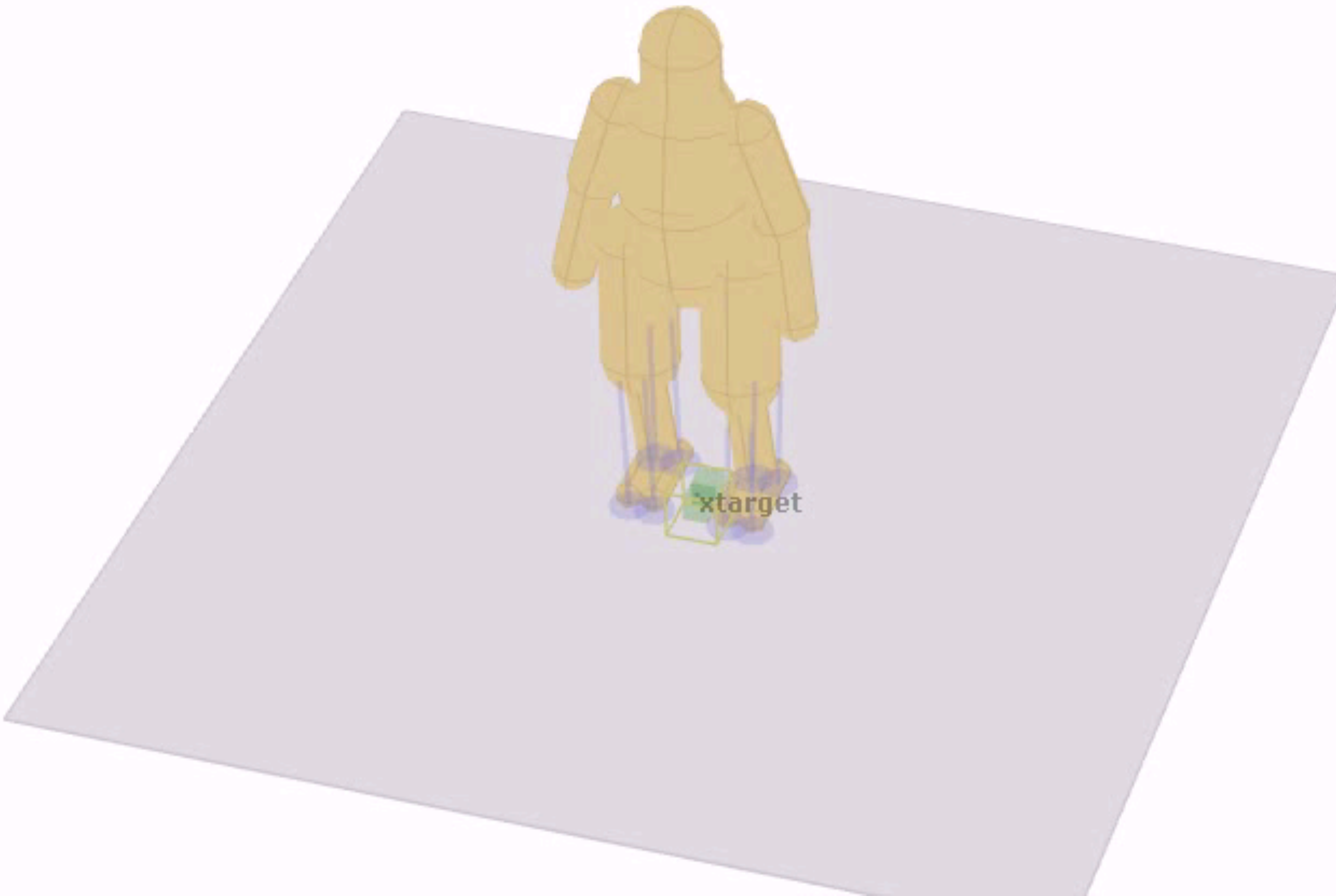
$$\min_{\theta} \sum_{i,t} ||\boldsymbol{\pi}_{\theta}(\mathbf{x}^{i,t}) - \mathbf{u}^{i,t}||^2$$

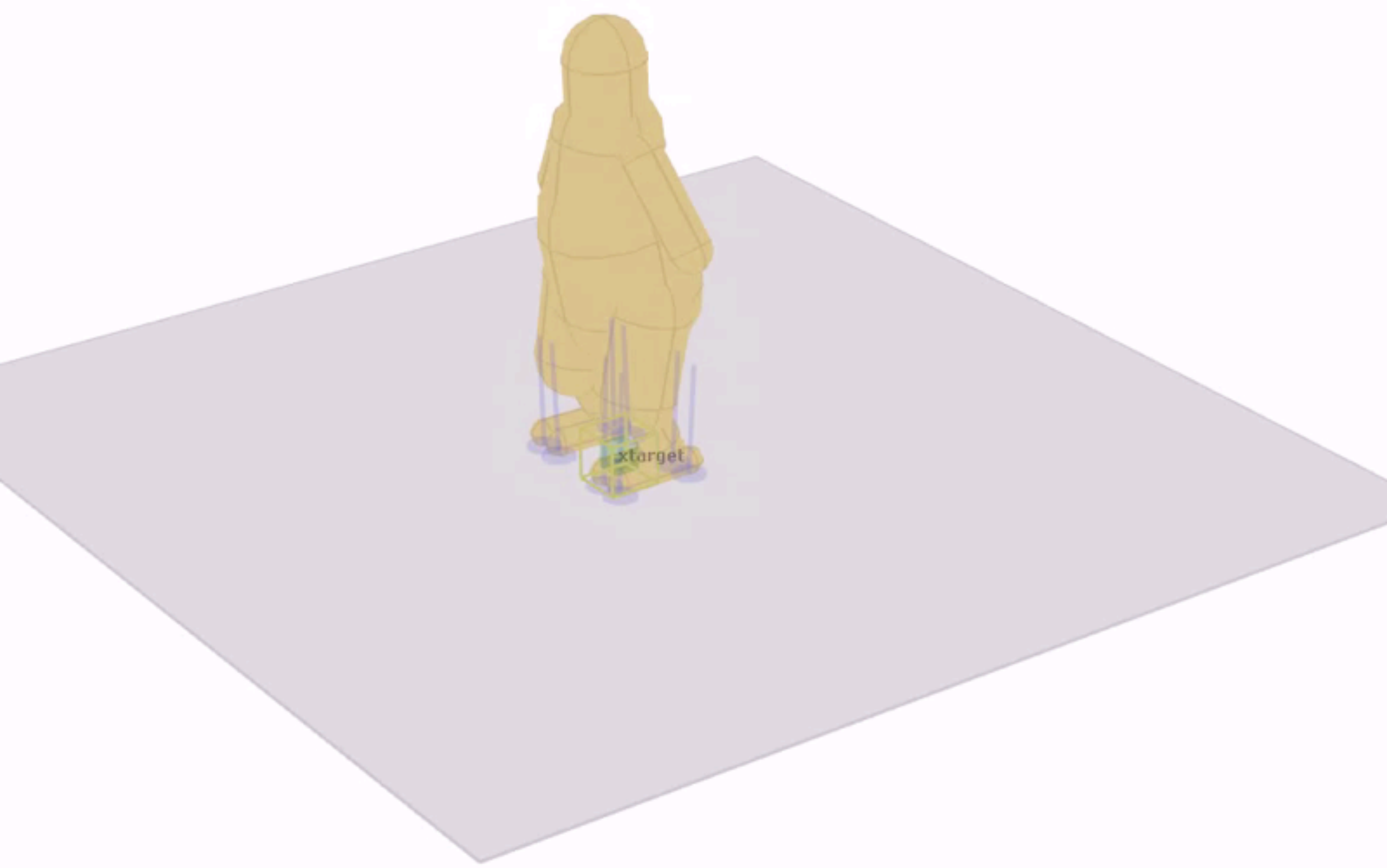
Scalable Implementation

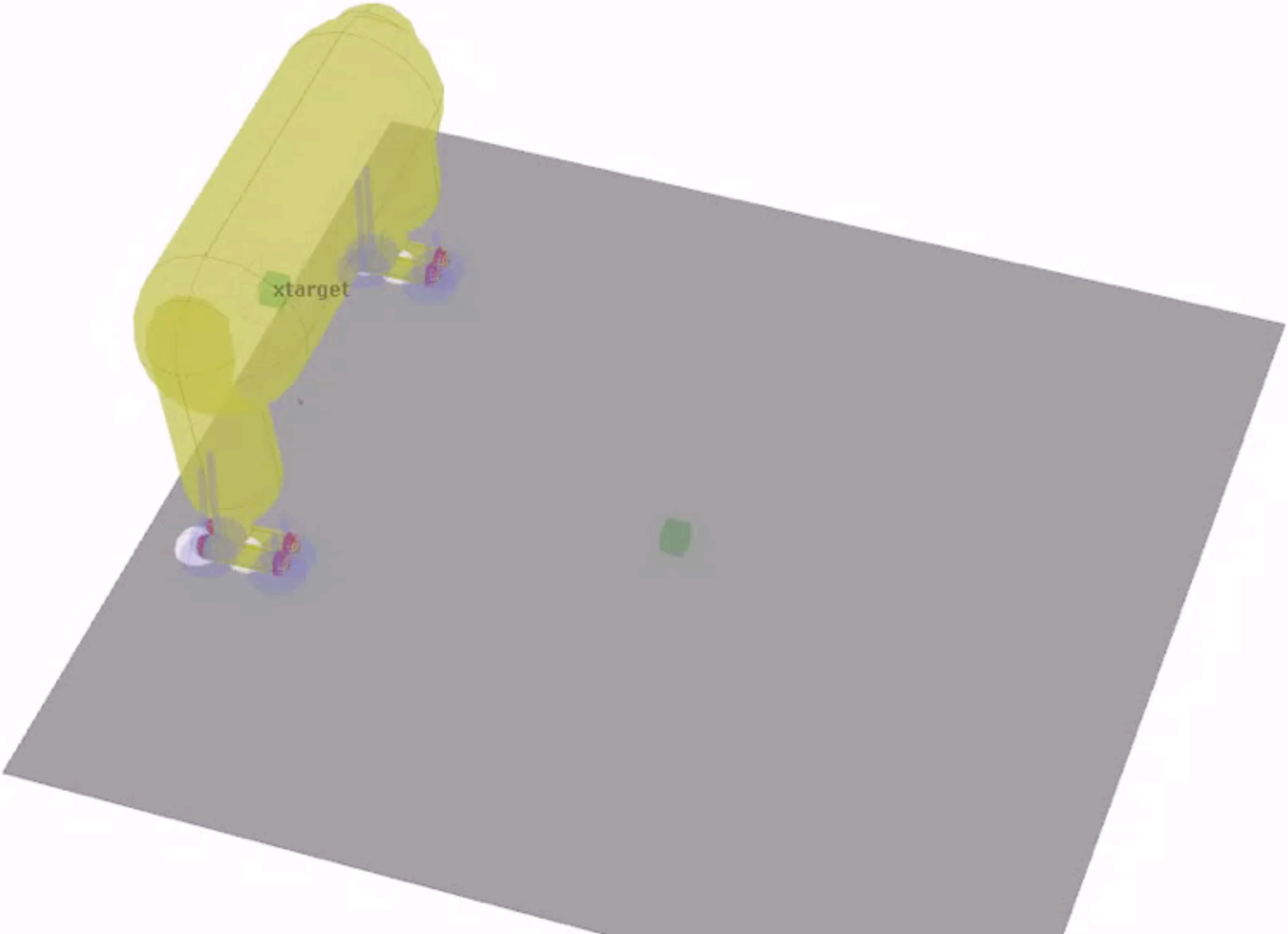


- asynchronous updates
- SGD network training
- Full dataset never loaded in memory

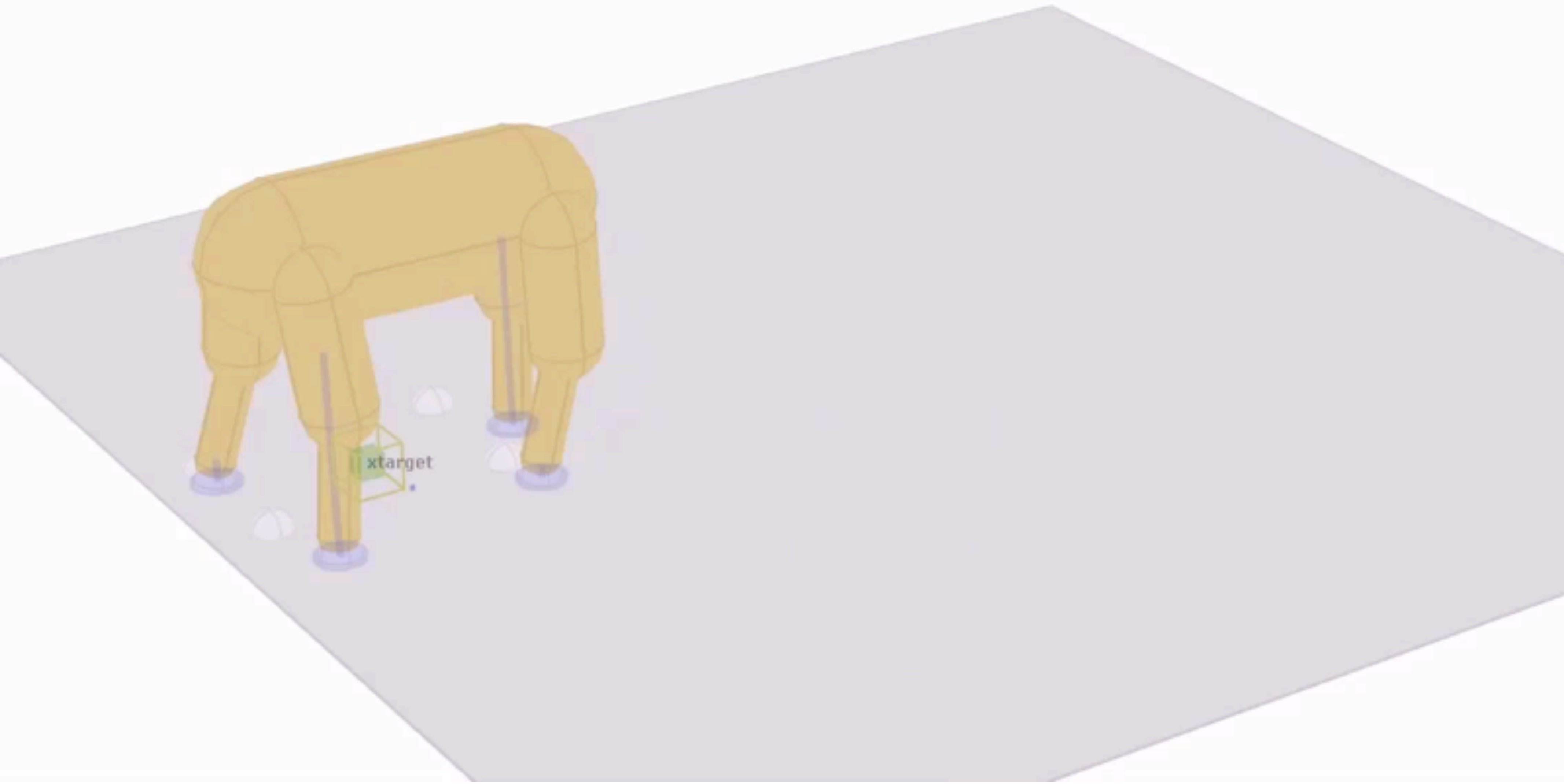


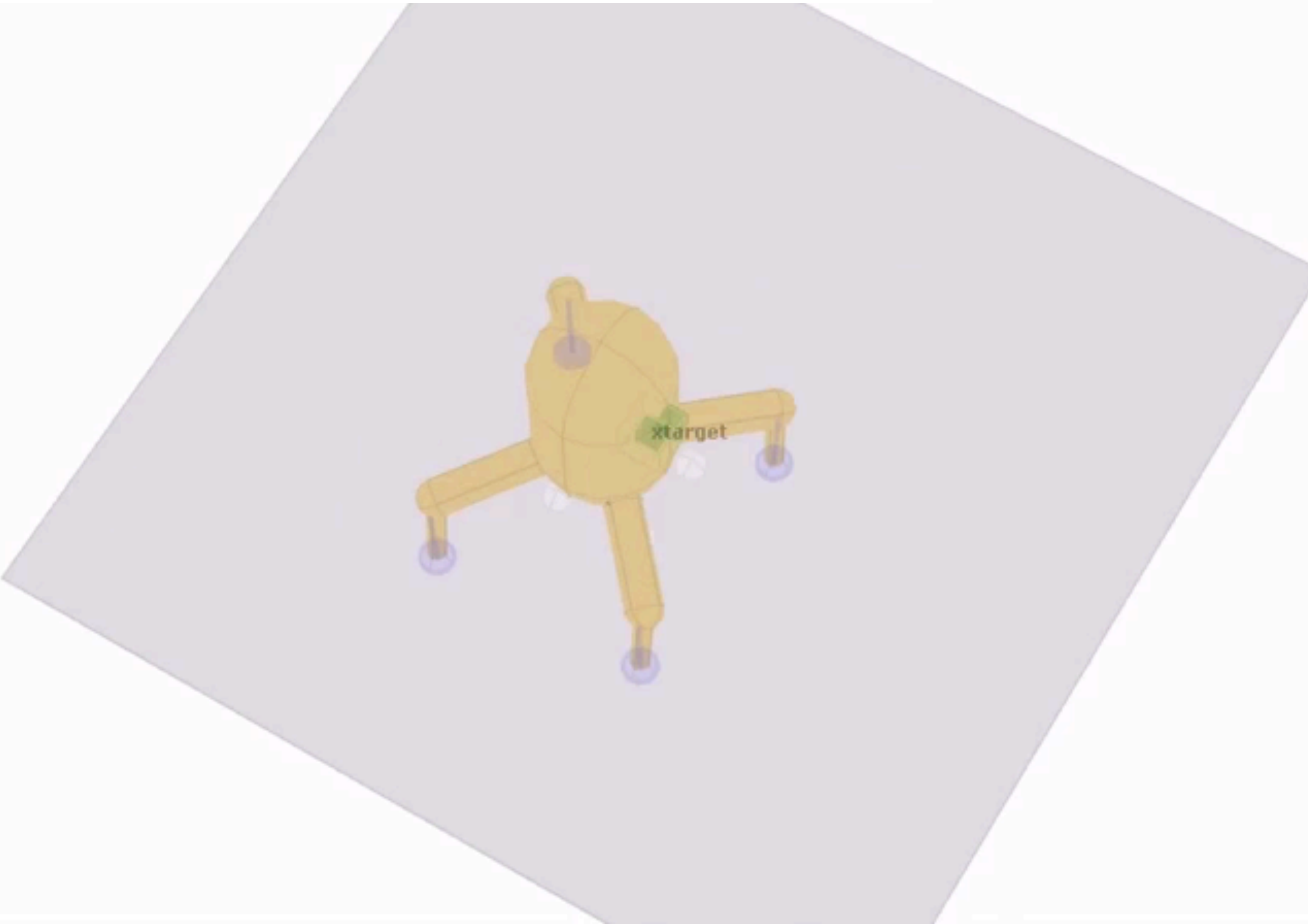

xtarget


xtarget

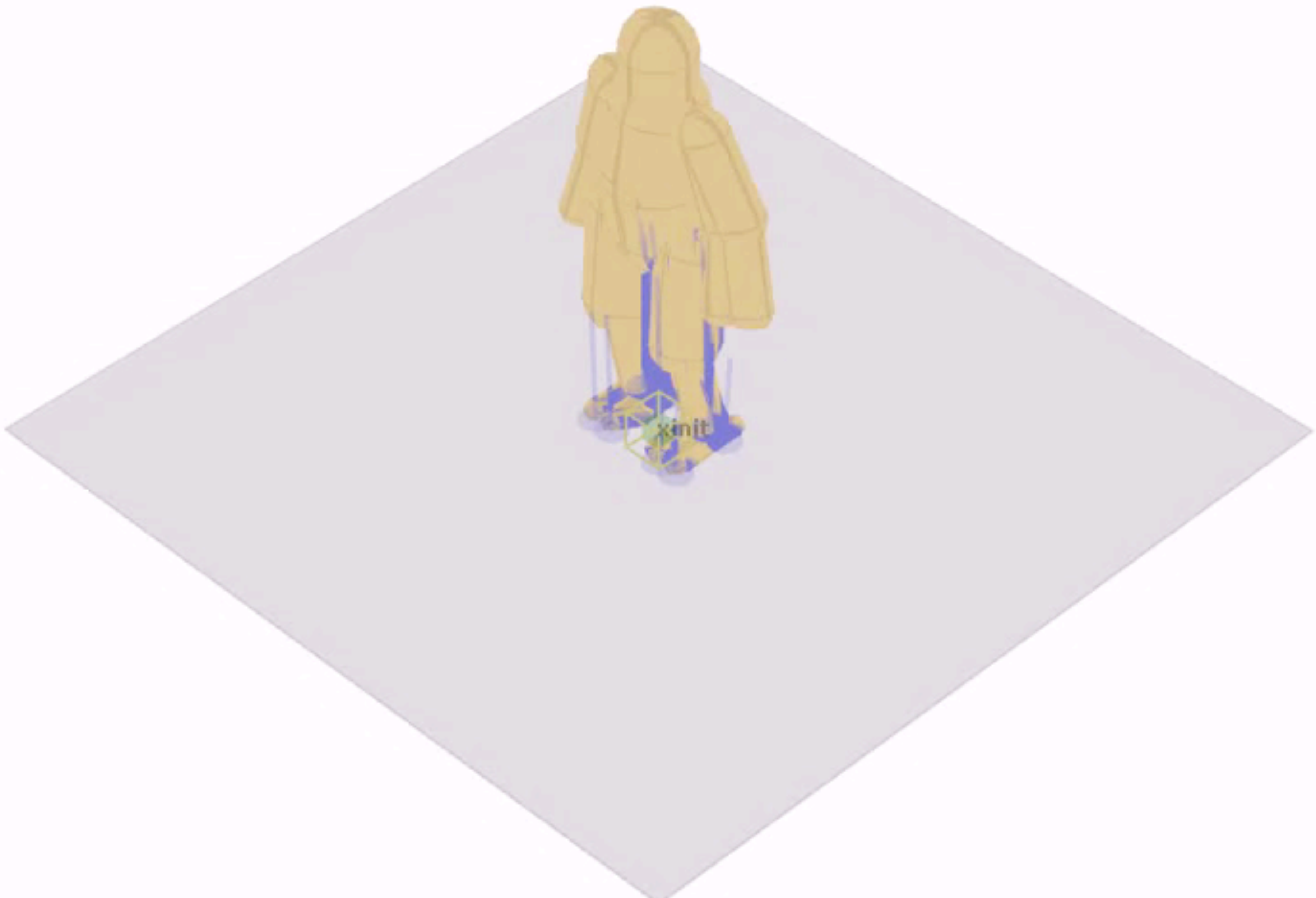

xtarget

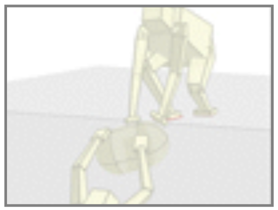

xtarget


xtarget

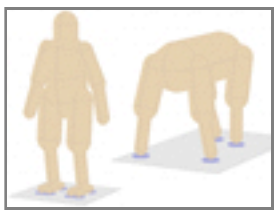

xtarget

Future State Prediction

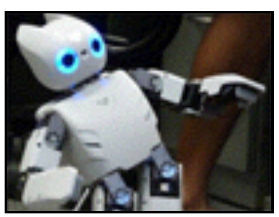




Trajectory Optimization with Direct Collocation

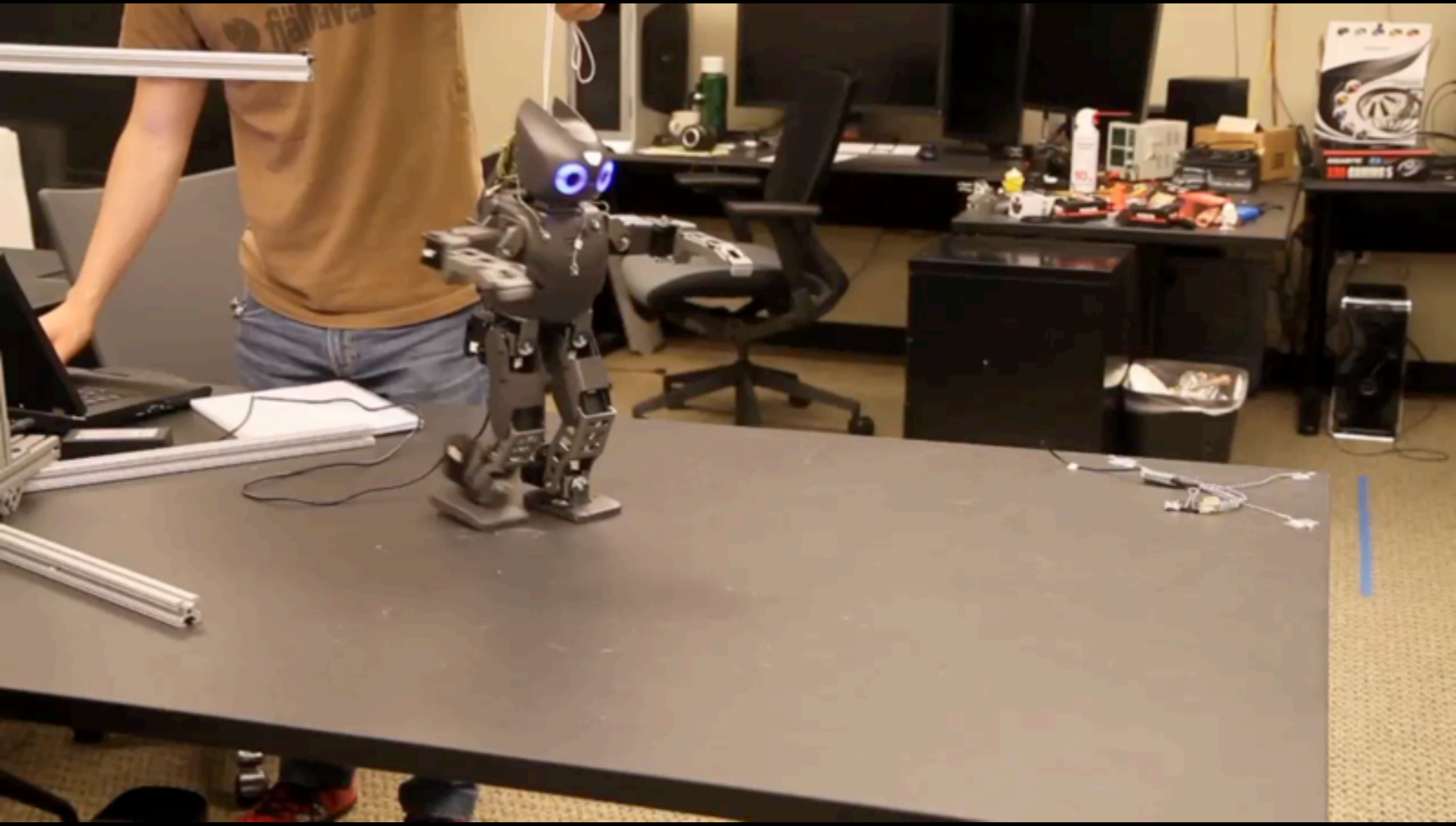


Learning Control Policies with Direct Collocation



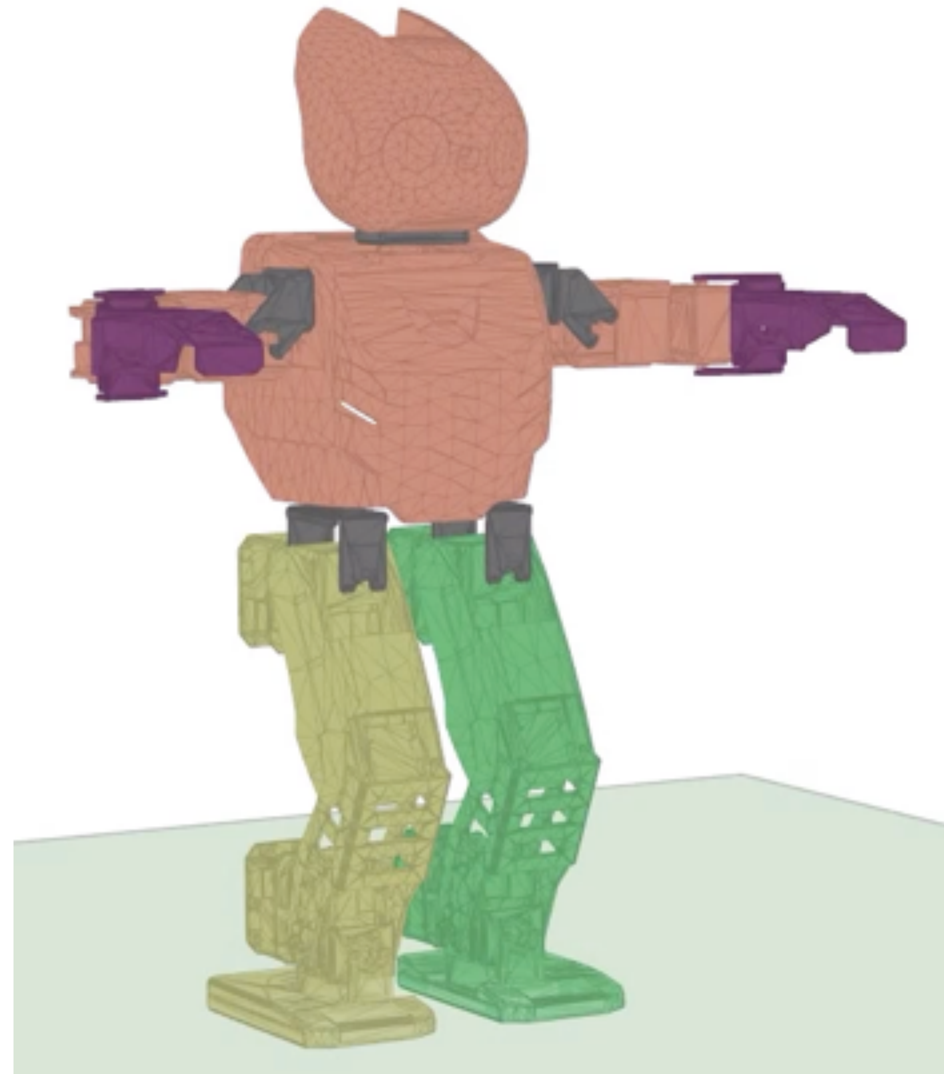
Unknown/Uncertain Dynamics and Applications

Executing optimal trajectories in open loop



What went wrong?

Darwin robot



Imperfect simulator model

Movement with Model Uncertainty

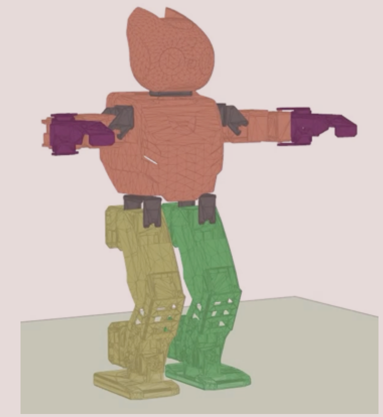
[Mordatch, Lowrey, Todorov, IROS 2015]

Movement with Model Uncertainty

[Mordatch, Lowrey, Todorov, IROS 2015]

Generate noisy models varying:

limb mass
limb center of mass
contact locations

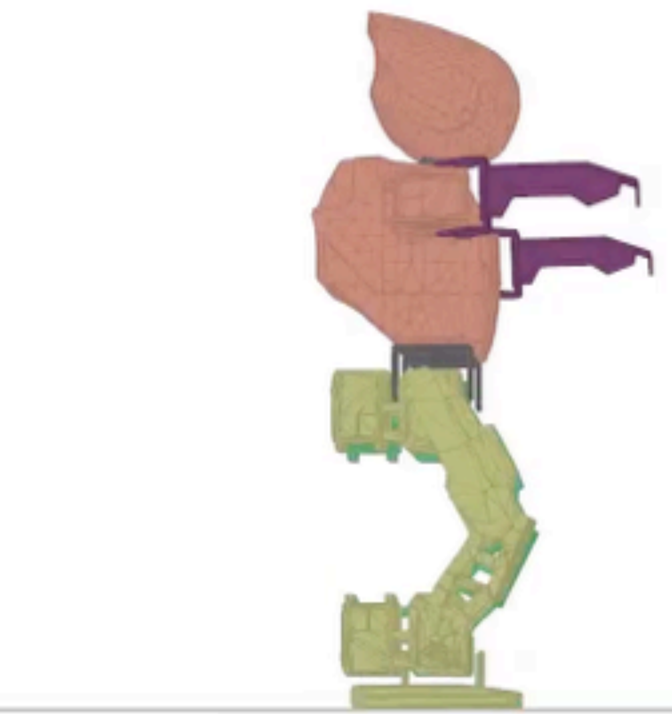


Optimize over multiple state trajectories

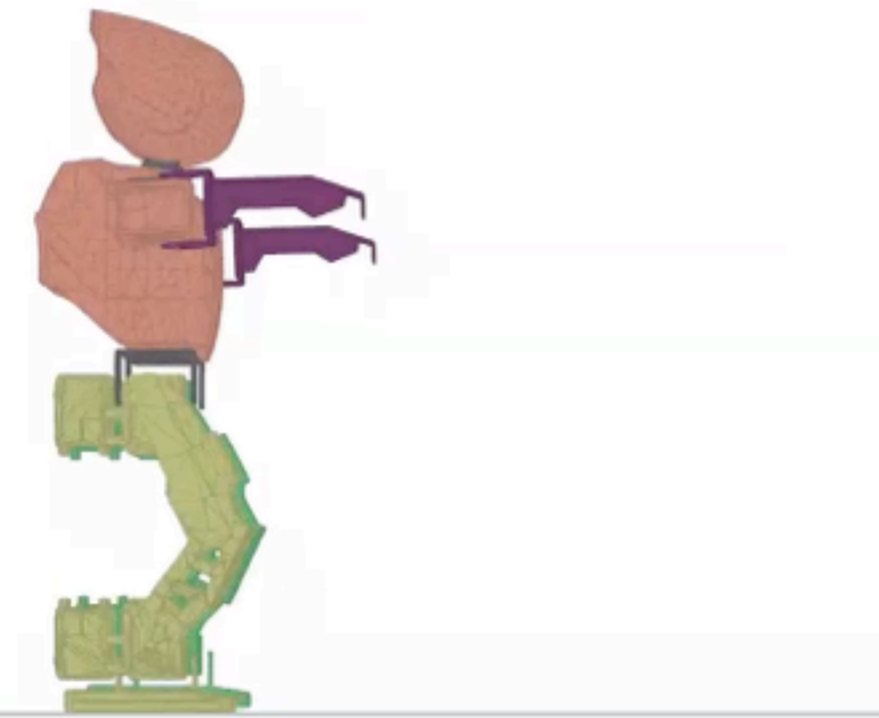
Single control trajectory

Execute control trajectory in open loop

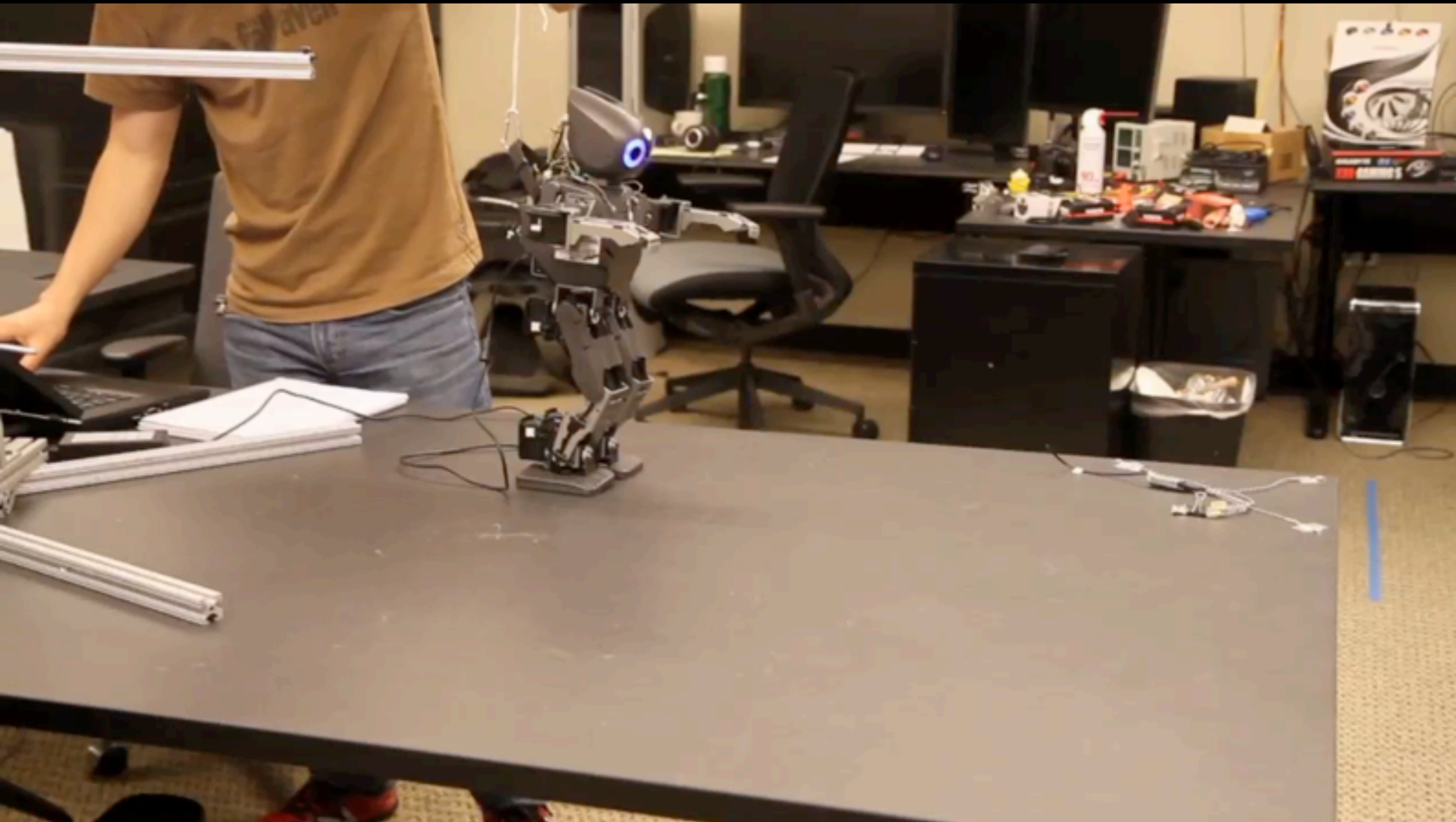
No Model Noise

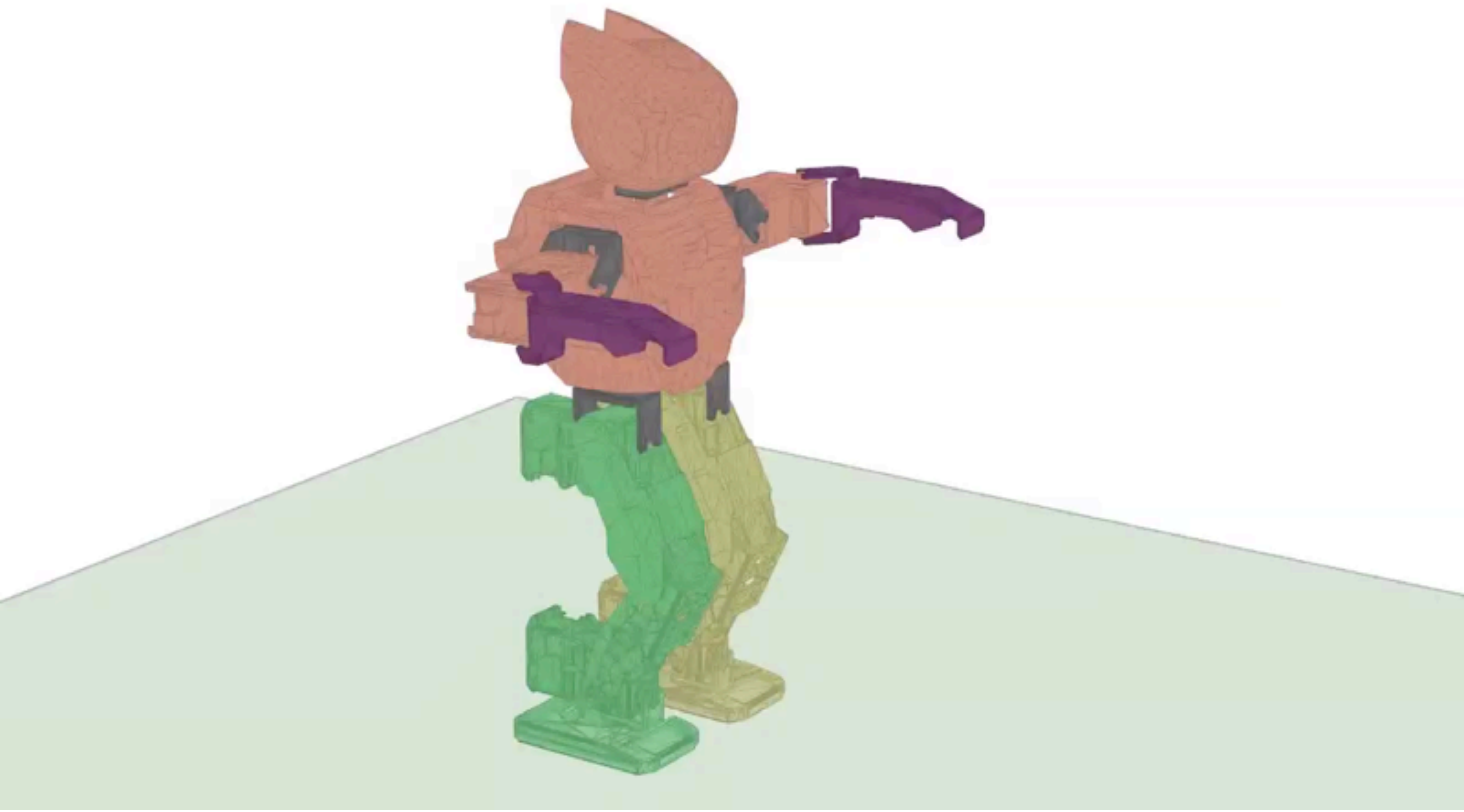


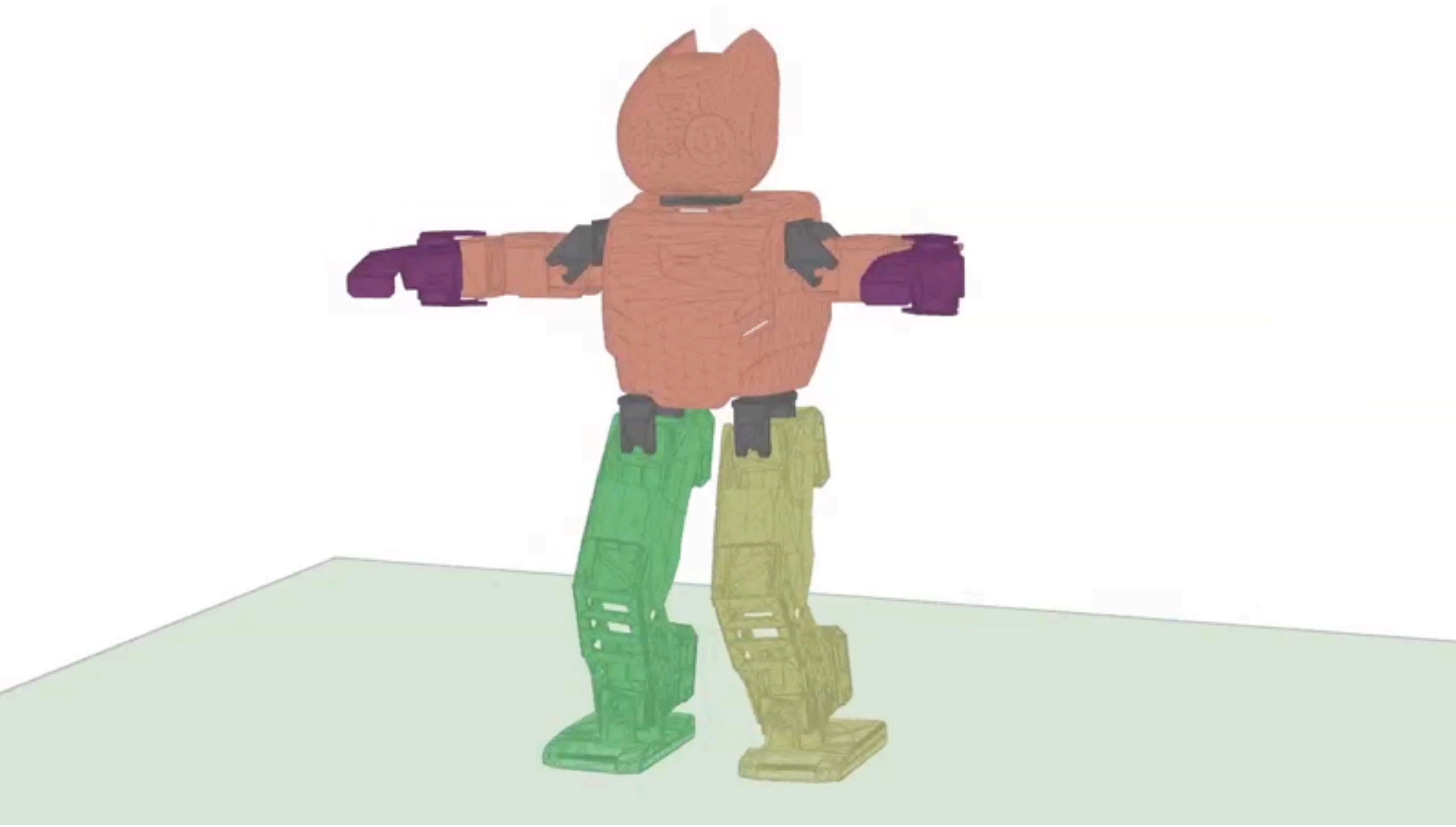
With Model Noise

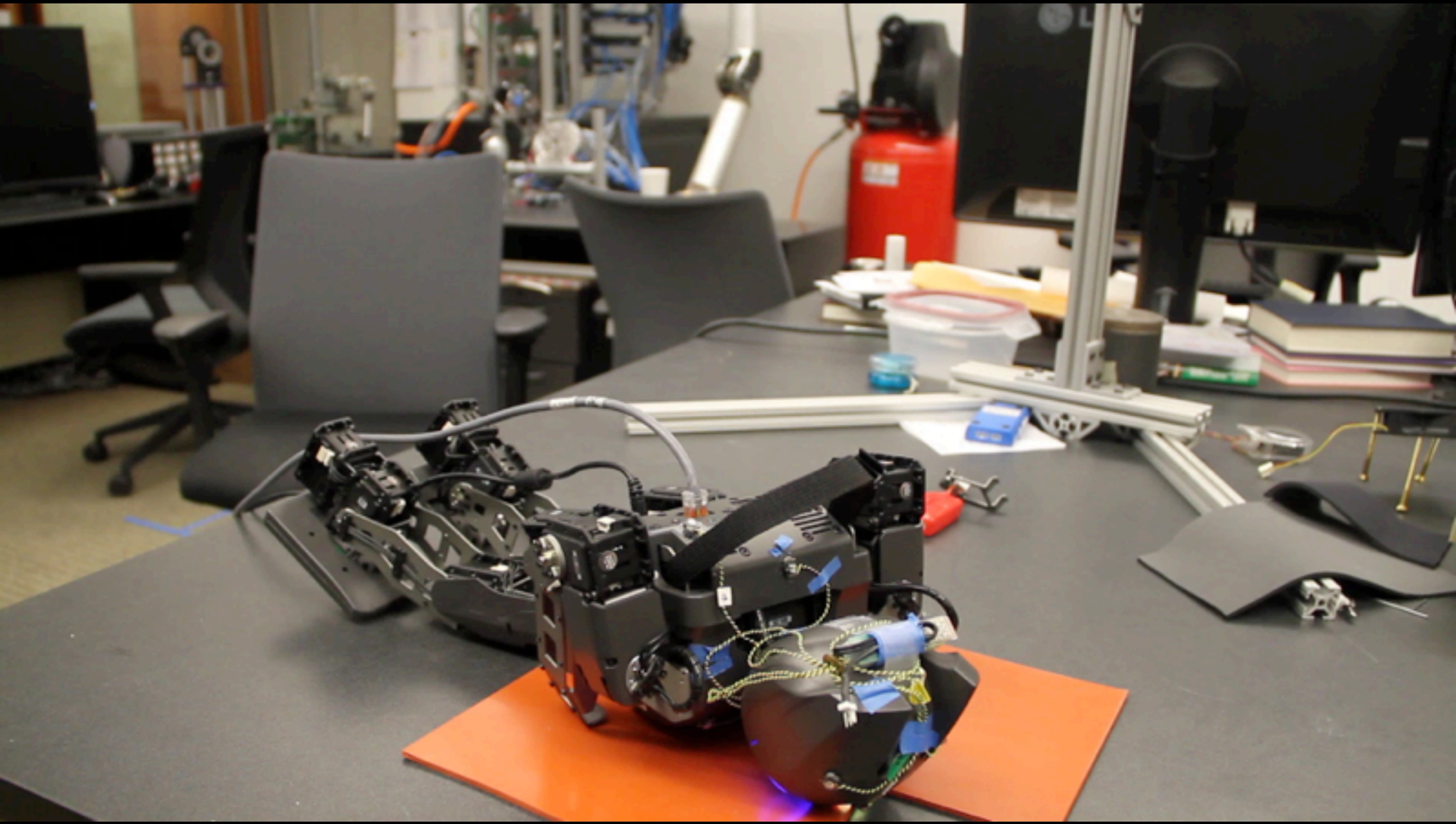


With Model Noise





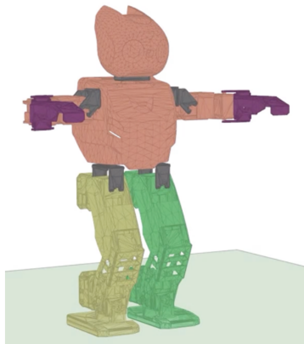




What if we don't want conservative motion?

Interactive Policies with Online Model Learning

[Mordatch, Mishra, Eppner, Abbeel, ICRA 2016]



Offline:

Train policy to output *desired* next state:

$$\bar{\mathbf{x}}^{t+1}$$ Joint Angles, IMU, Forces

At every timestep:

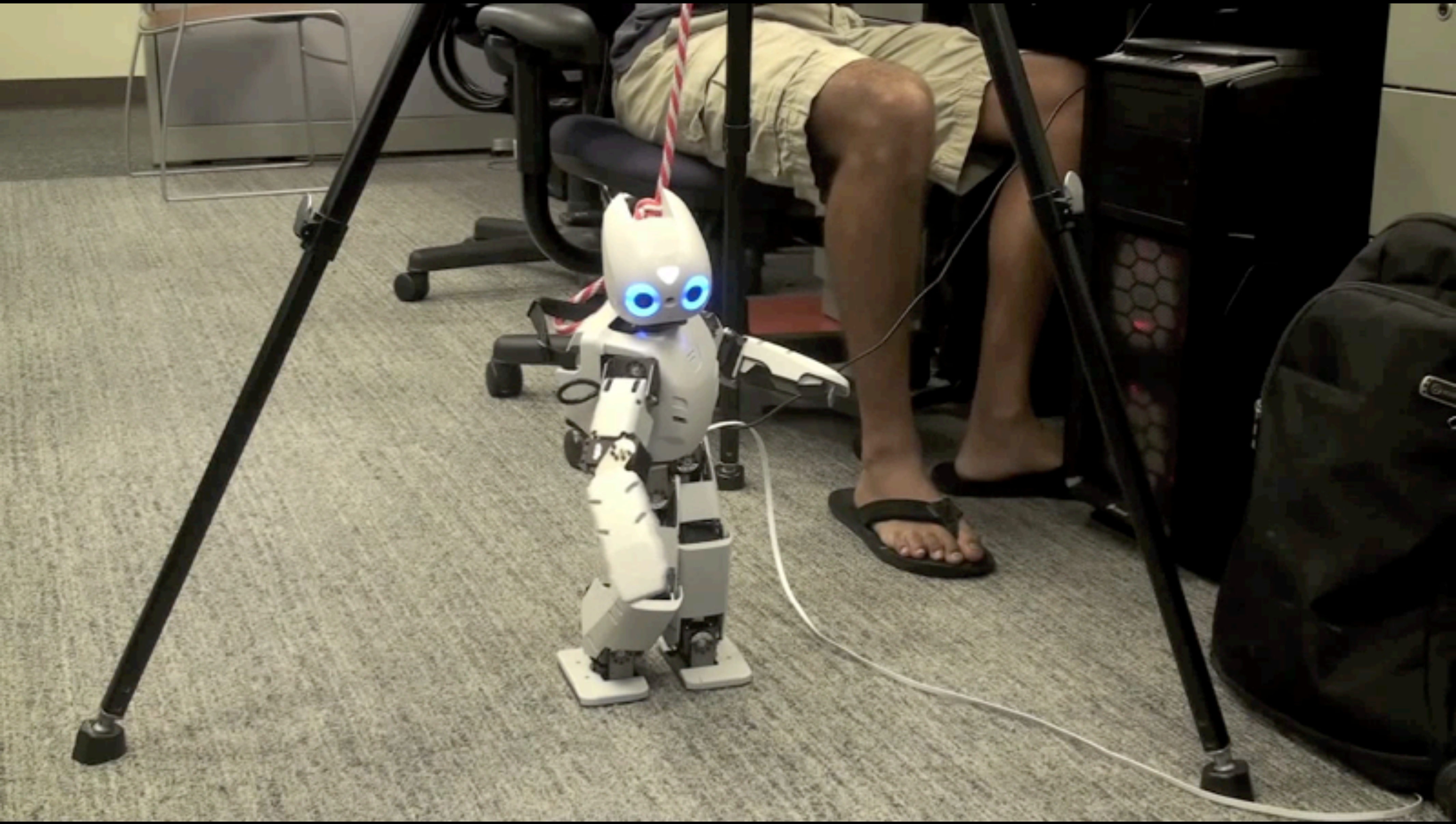
Learn robot dynamics on the fly from past observations

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

Query policy for $\bar{\mathbf{x}}^{t+1}$

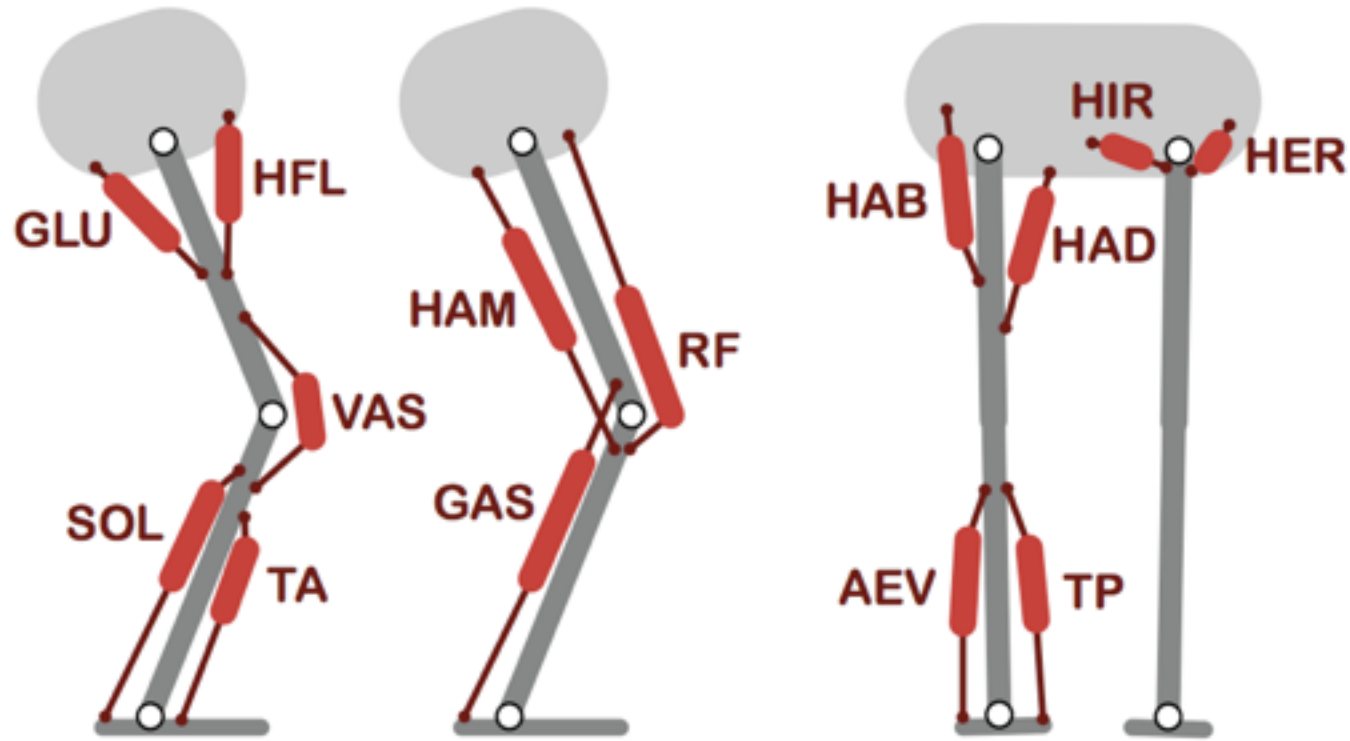
Solve for robot torques $\mathbf{u}^*$ such that

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

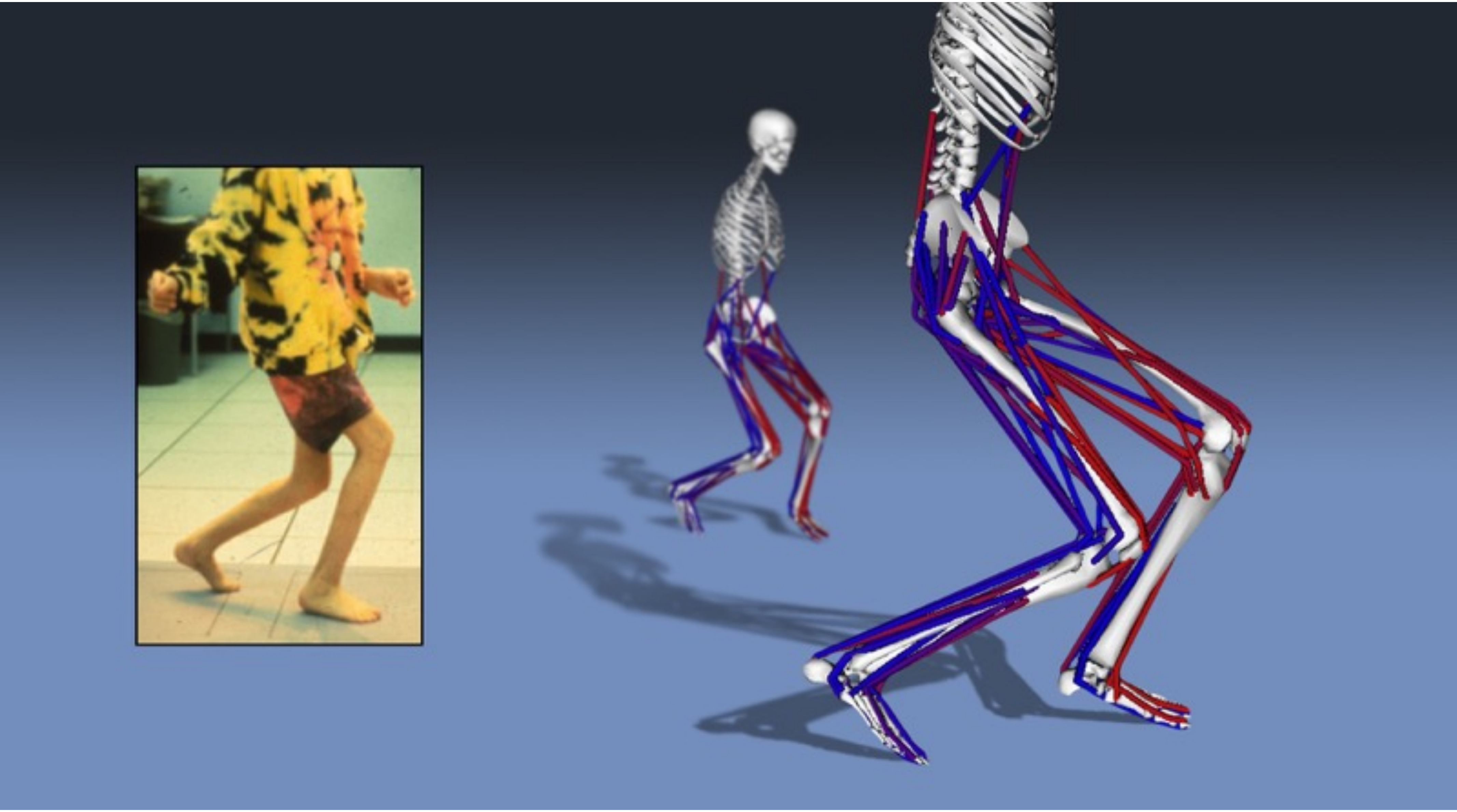


Detailed Humanoid Models

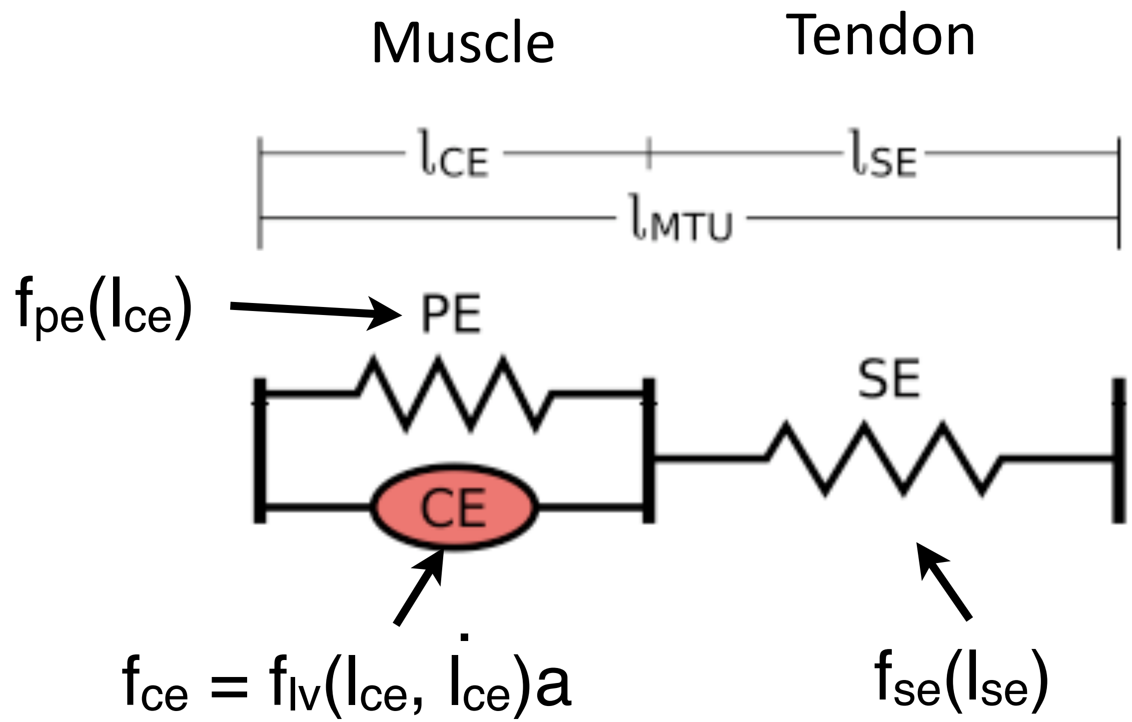
[Mordatch, Wang, Todorov, Koltun, SIGGRAPH Asia 2013]



Detailed Humanoid Models



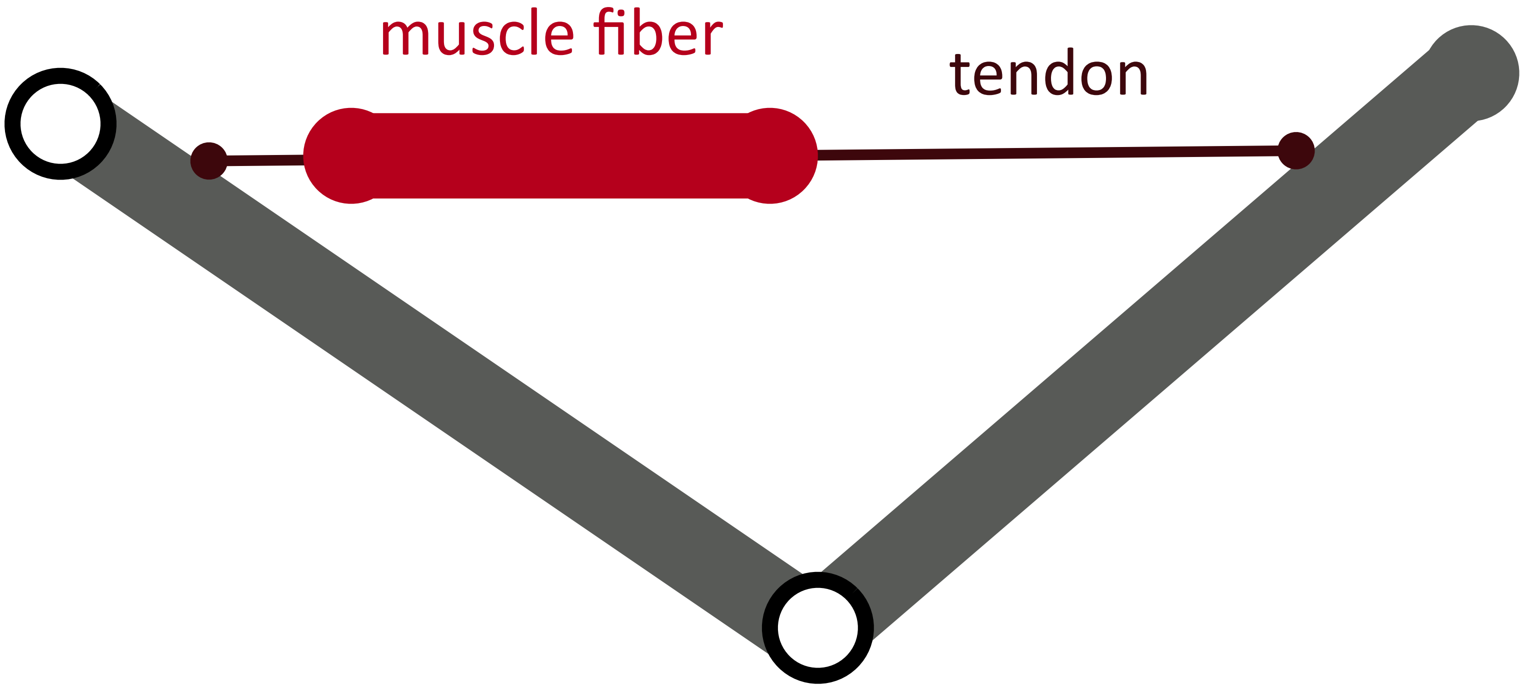
Musculotendon Actuator (Hill Model)



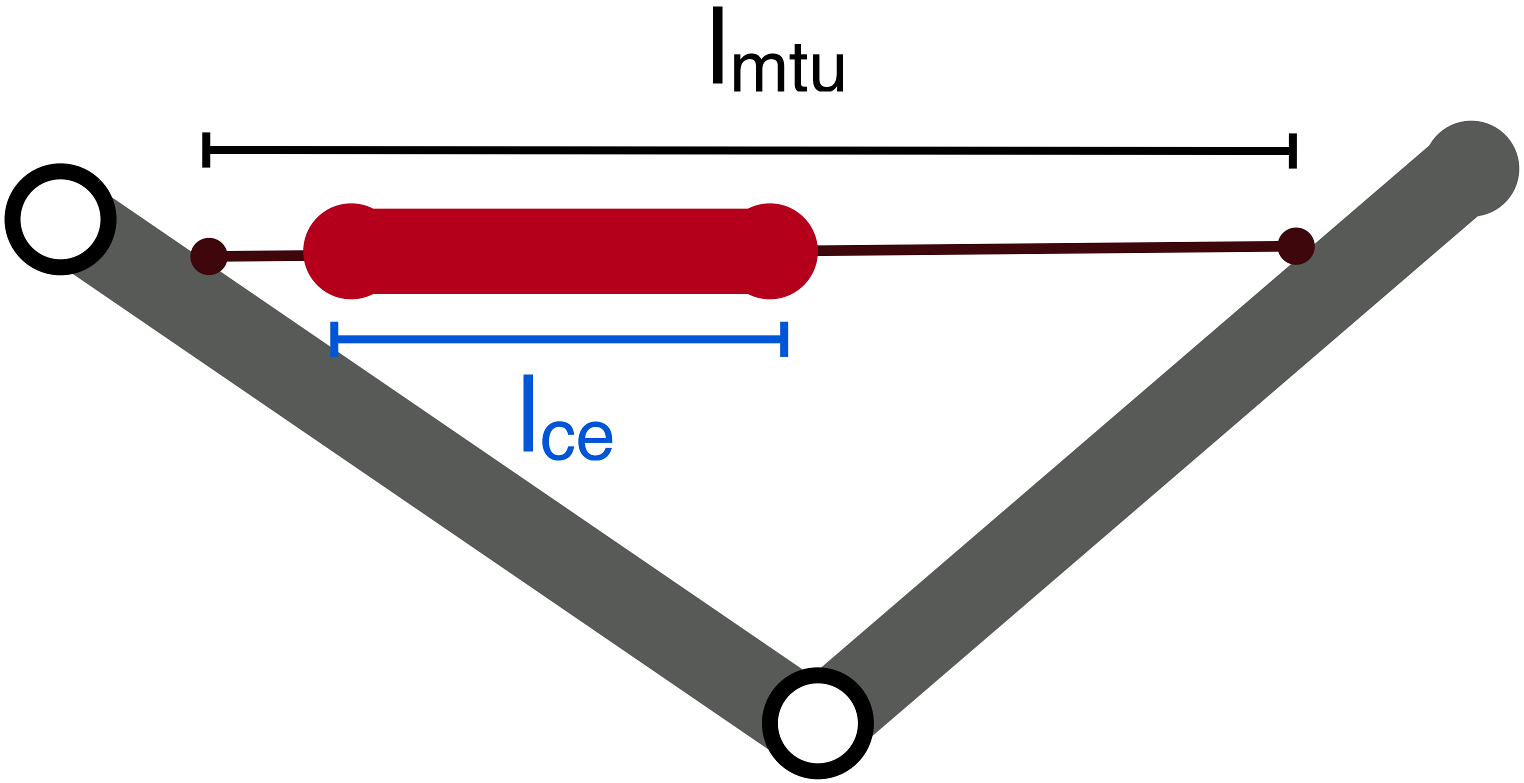
Equilibrium constraint: $f_{ce} + f_{pe} = f_{se}$

Model effort with metabolic energy expenditure [Anderson 99]

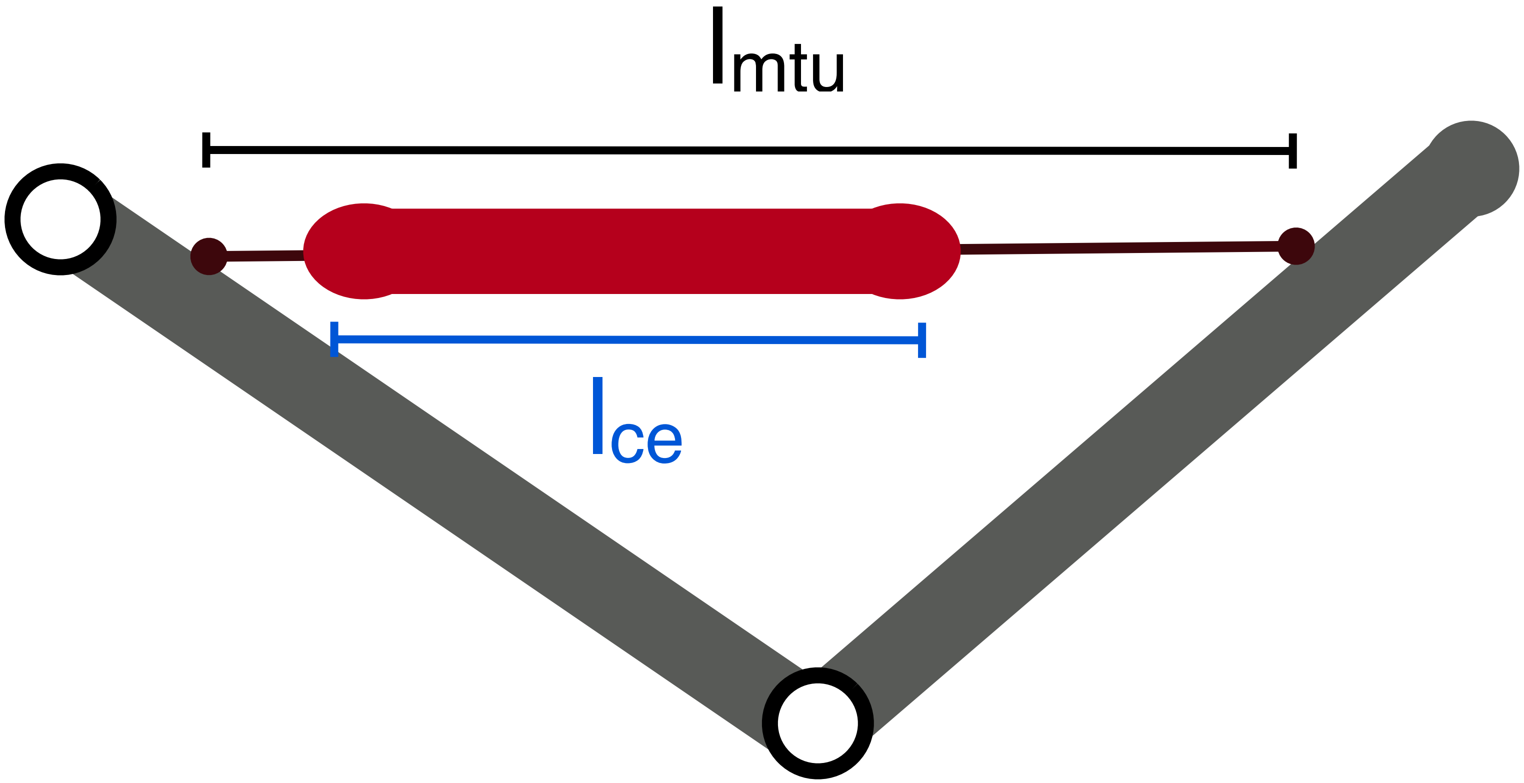
Musculotendon Unit (MTU) Actuator



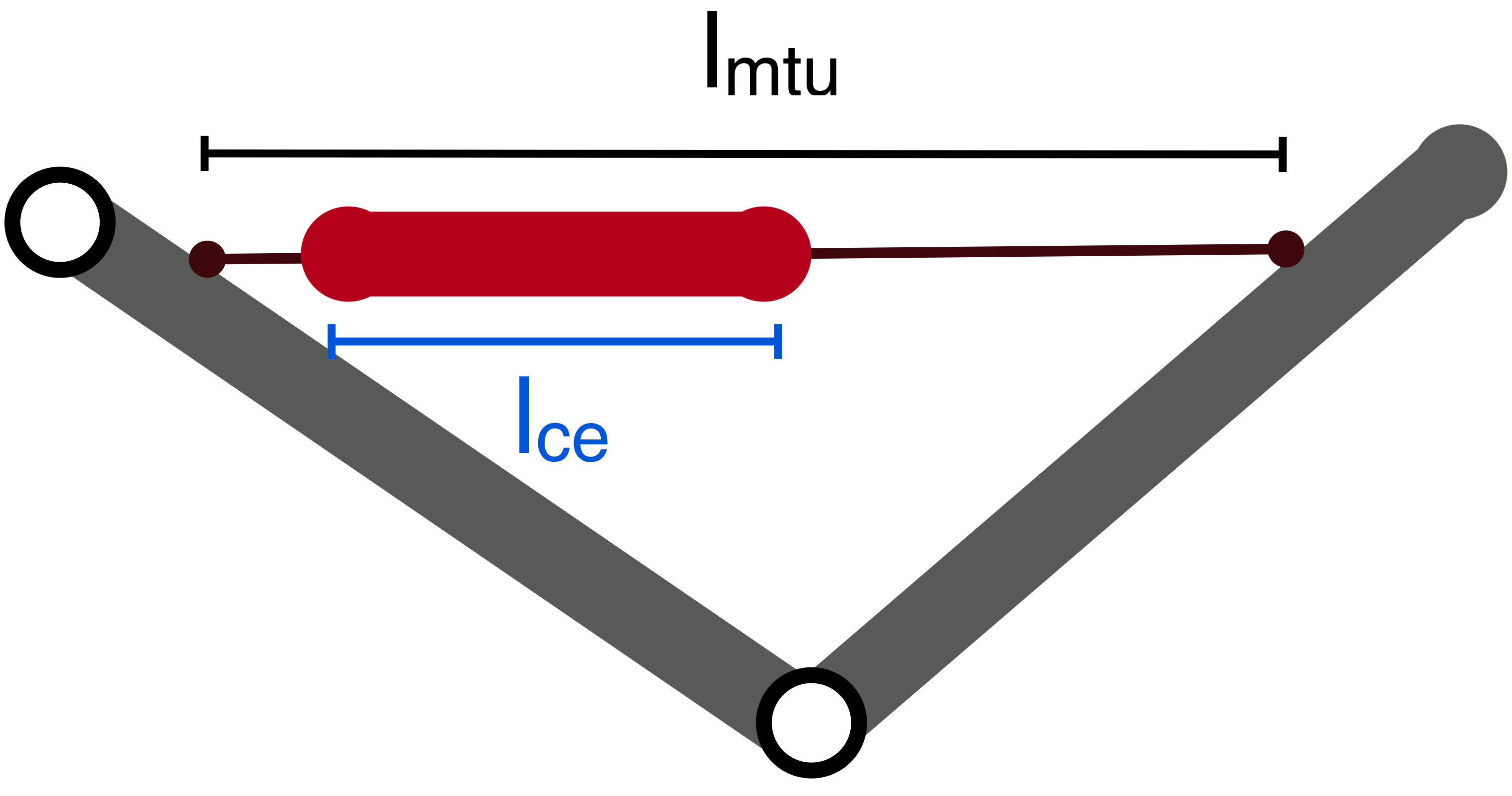
Musculotendon Unit (MTU) Actuator



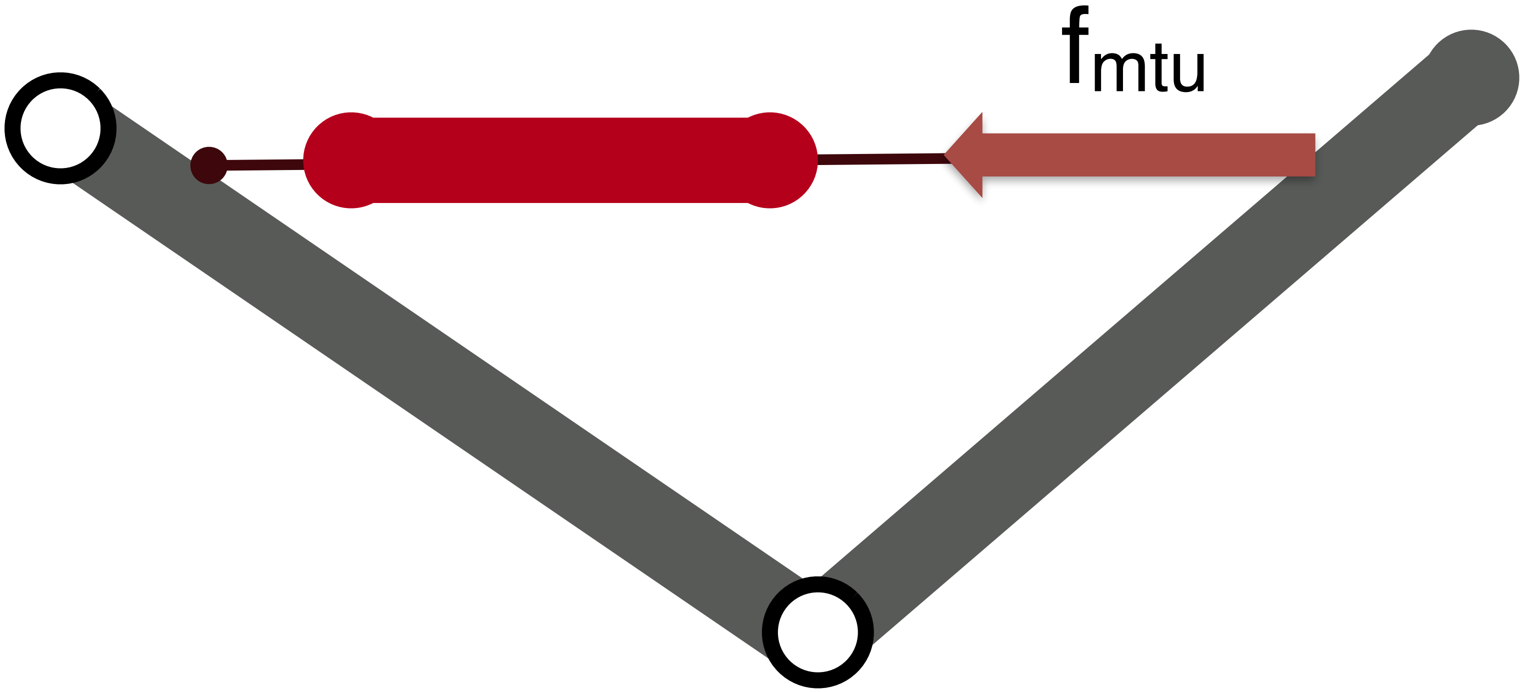
Musculotendon Unit (MTU) Actuator



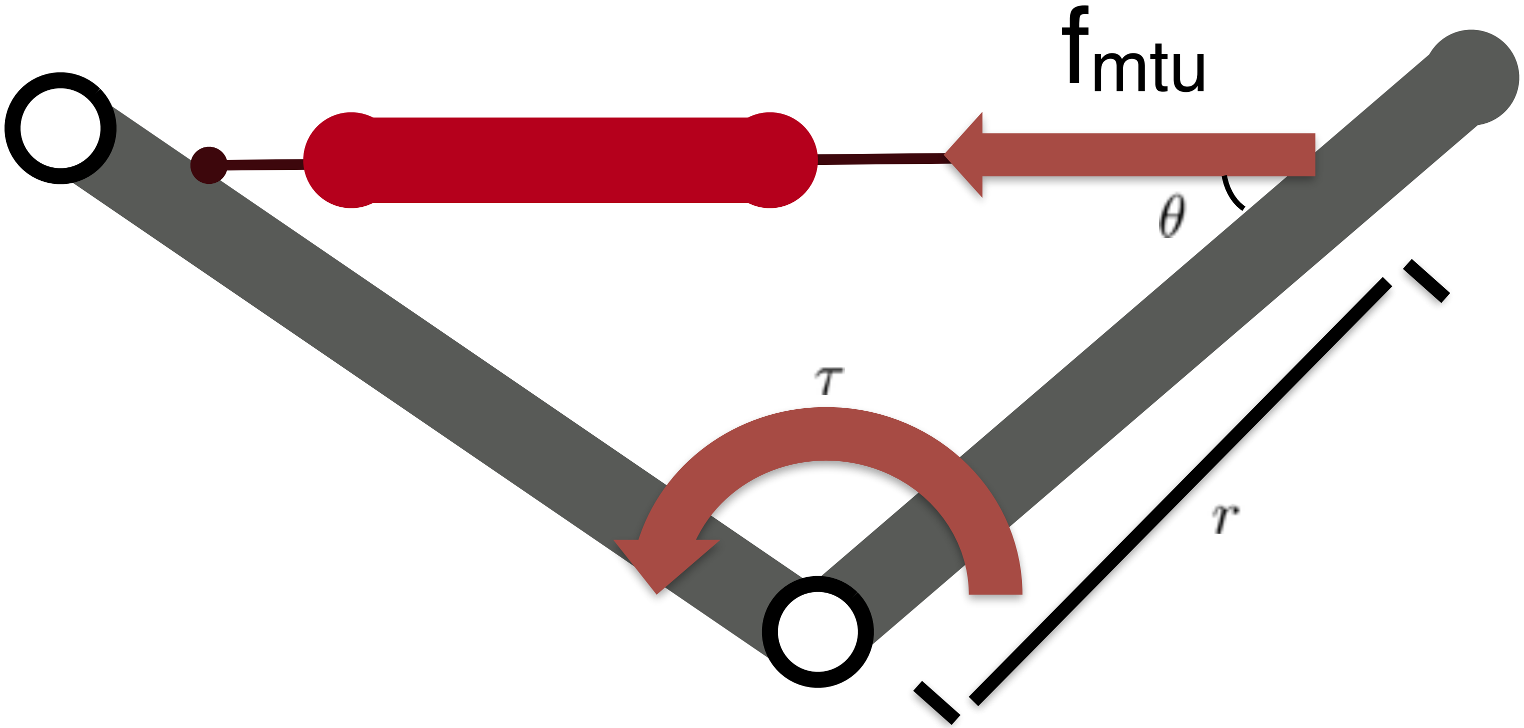
Musculotendon Unit (MTU) Actuator



Musculotendon Unit (MTU) Actuator

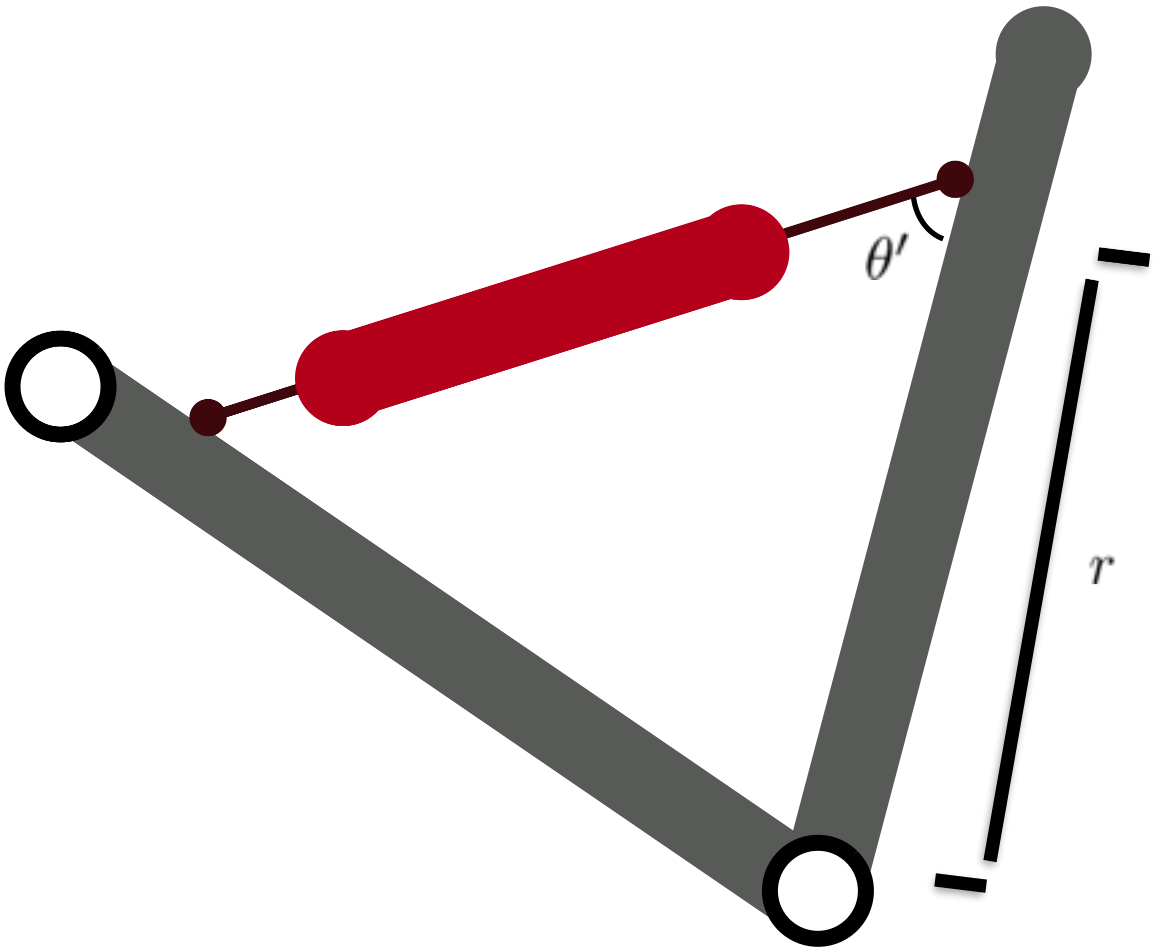


Musculotendon Unit (MTU) Actuator



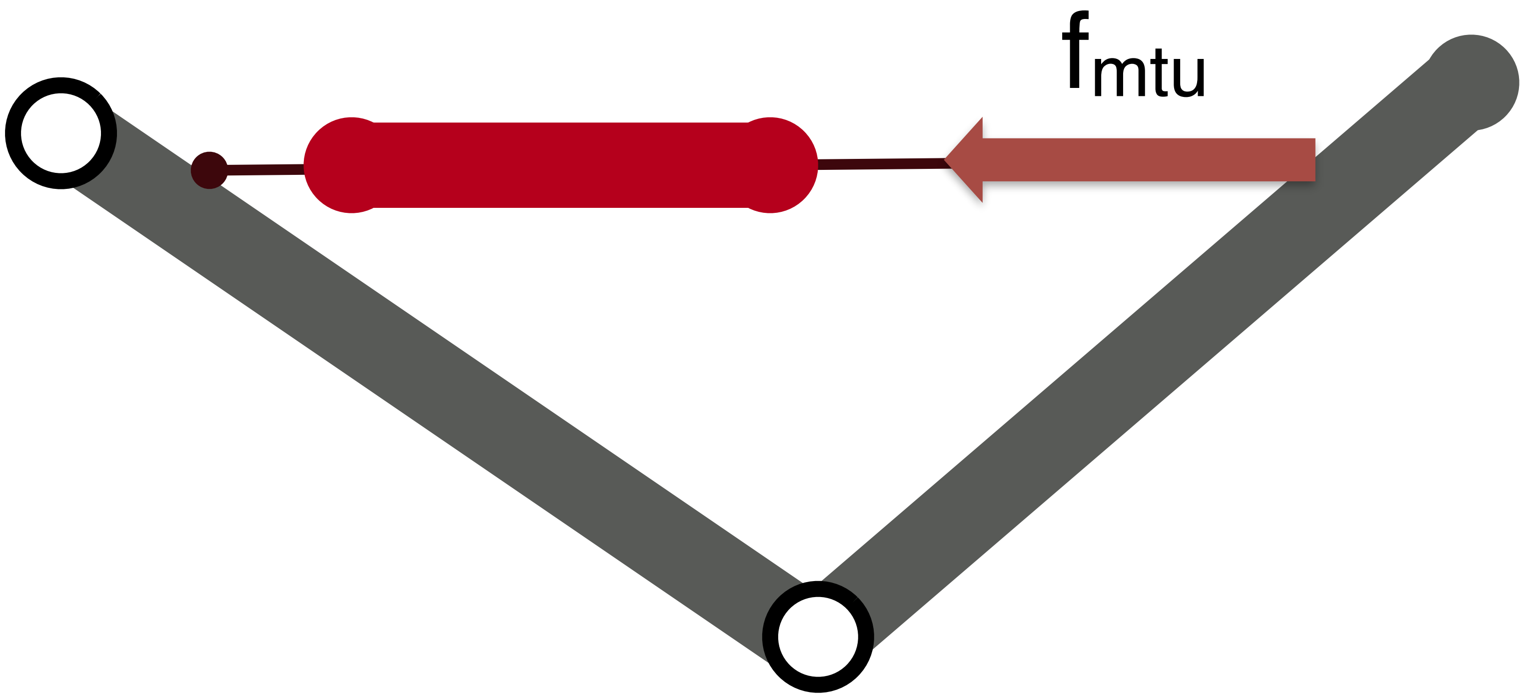
$$\tau = rF^{\mathrm{MTU}} \sin\theta$$

Musculotendon Unit (MTU) Actuator



$$\tau' = rF^{\mathrm{MTU}} \sin\theta'$$

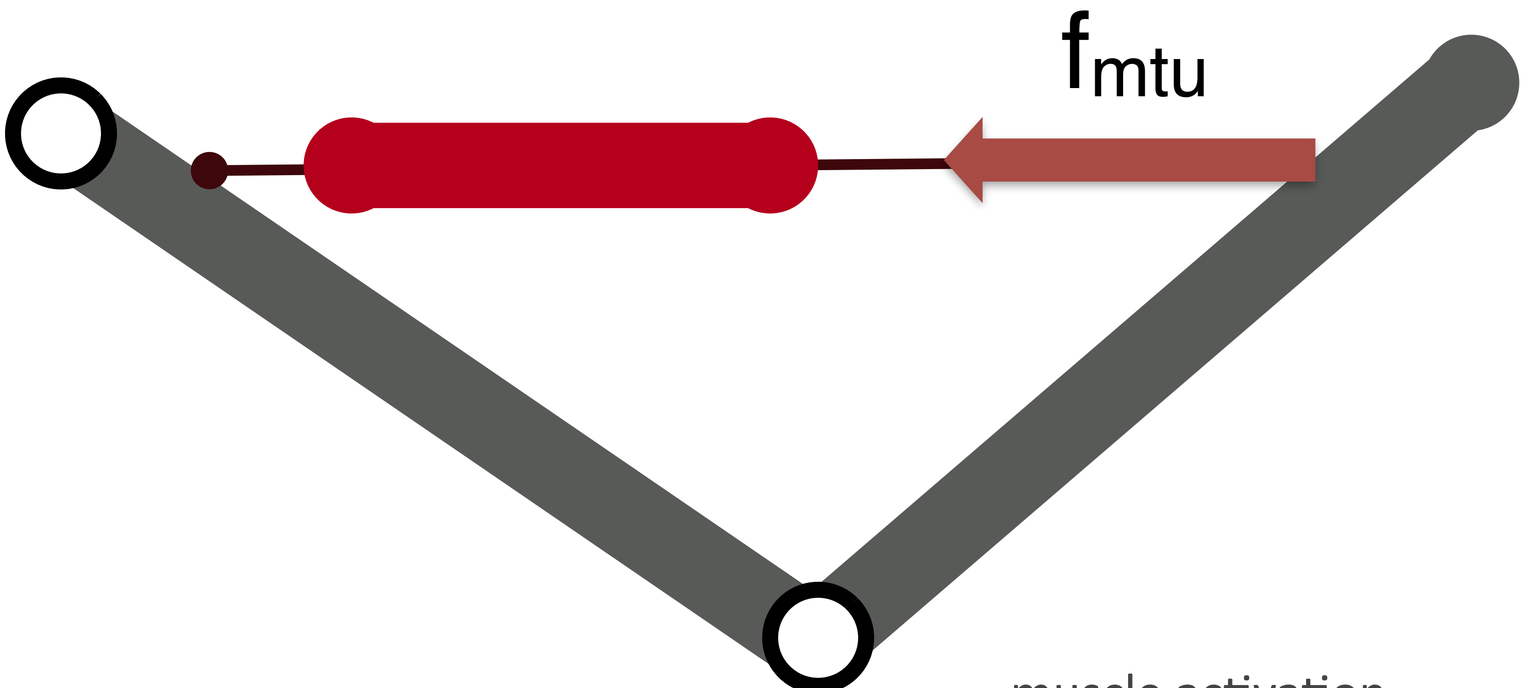
Musculotendon Unit (MTU) Actuator



$$f_{mtu} = f_{max}\, f_l(l_{ce})\, f_v(\dot{l}_{ce})\, a + f_{pe}$$

force-length and force-velocity relations

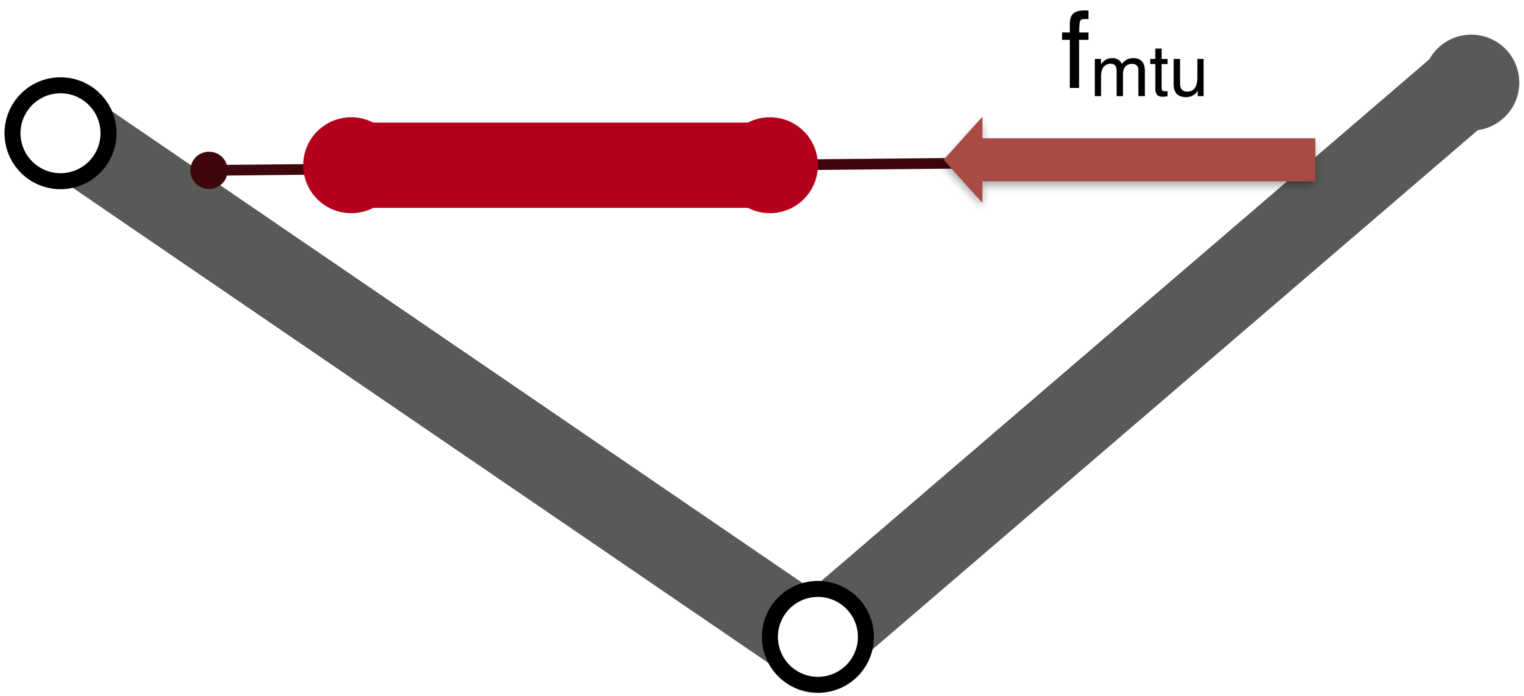
Musculotendon Unit (MTU) Actuator



muscle activation

$$f_{mtu} = f_{max}\, f_l(l_{ce})\ f_v(\dot{l}_{ce})\, a + f_{pe}$$

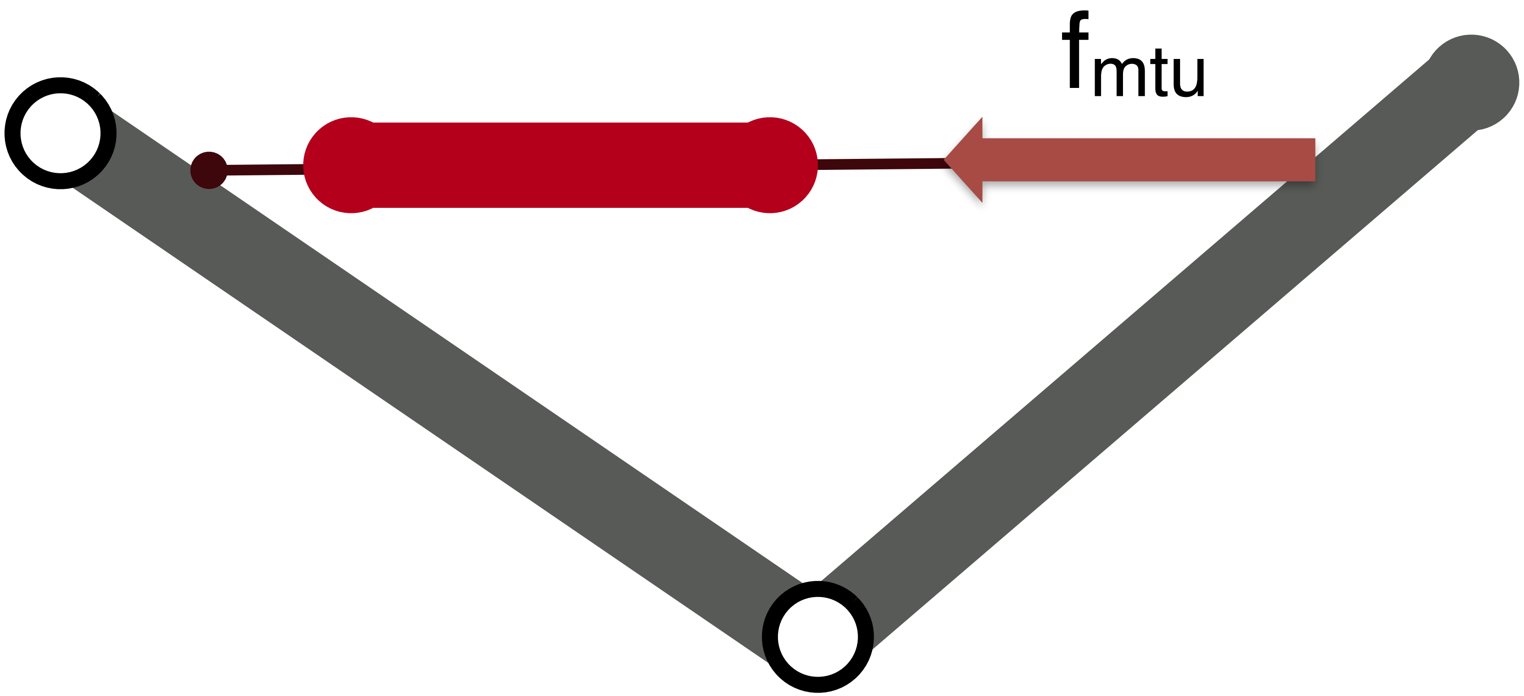
Musculotendon Unit (MTU) Actuator



$$f_{mtu} = f_{max}\, f_l(l_{ce})\, f_v(\dot{l}_{ce})\, a + f_{pe}$$

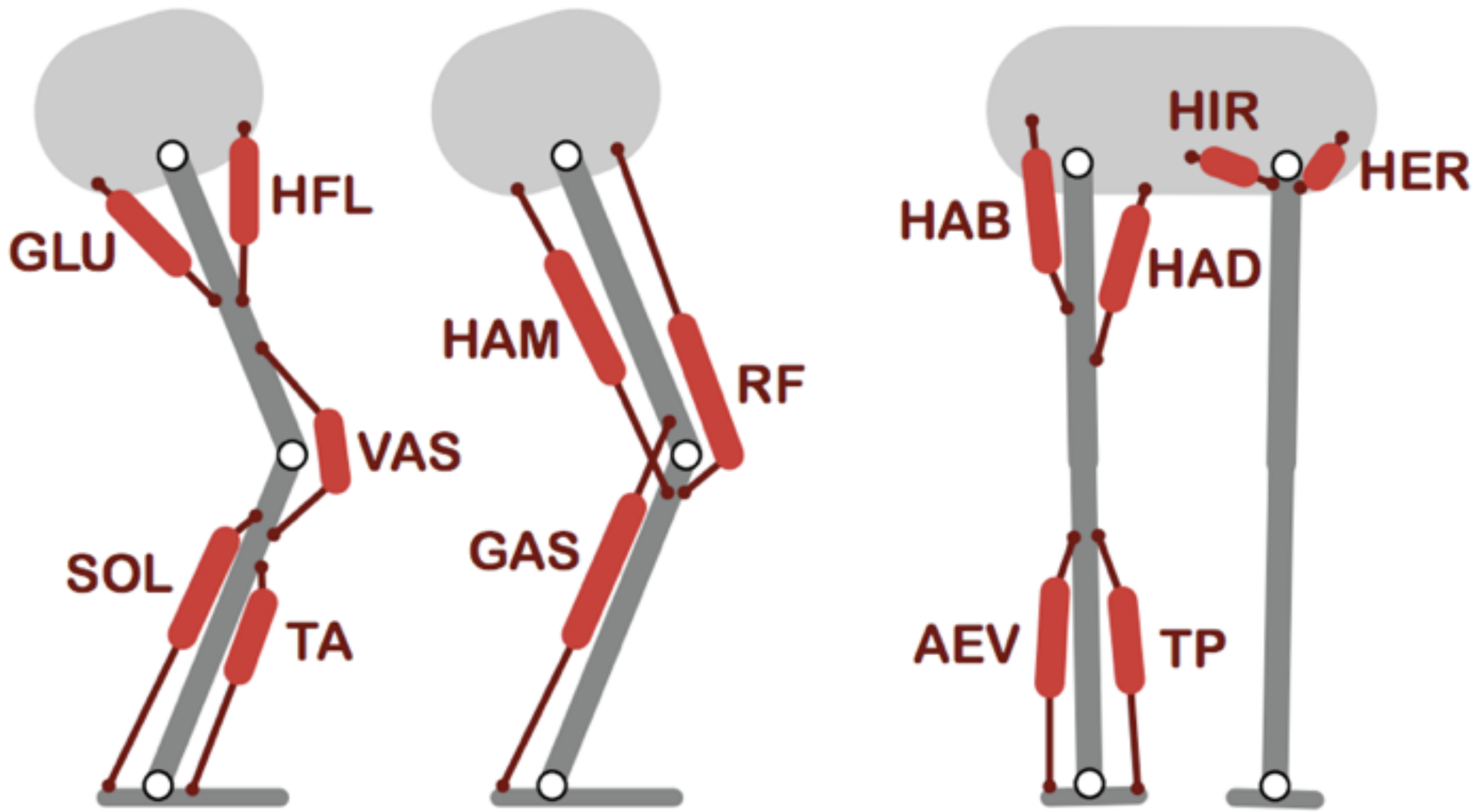
passive force

Musculotendon Unit (MTU) Actuator

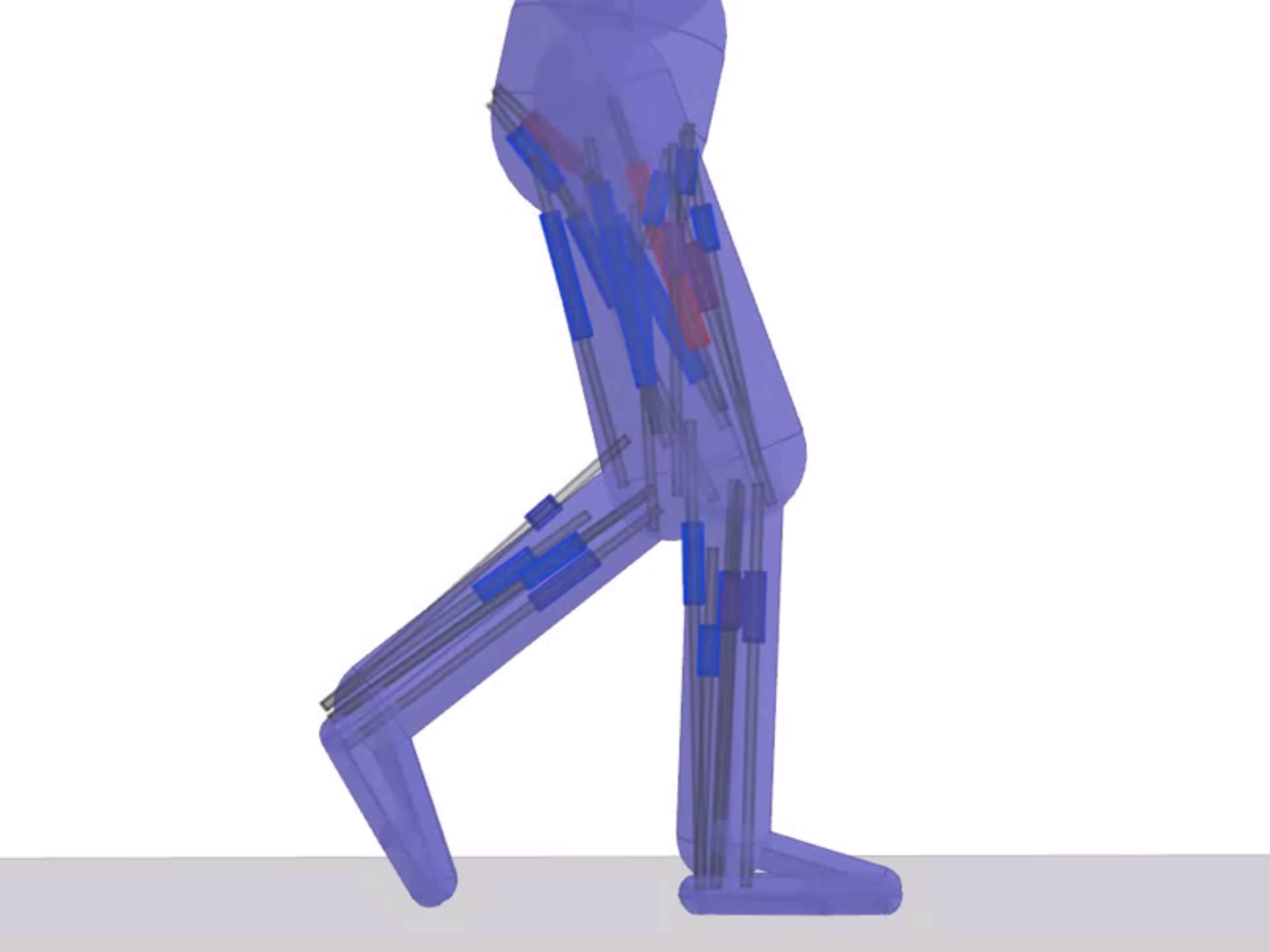


MTU usage effort modeled with metabolic energy expenditure
[Anderson 99]

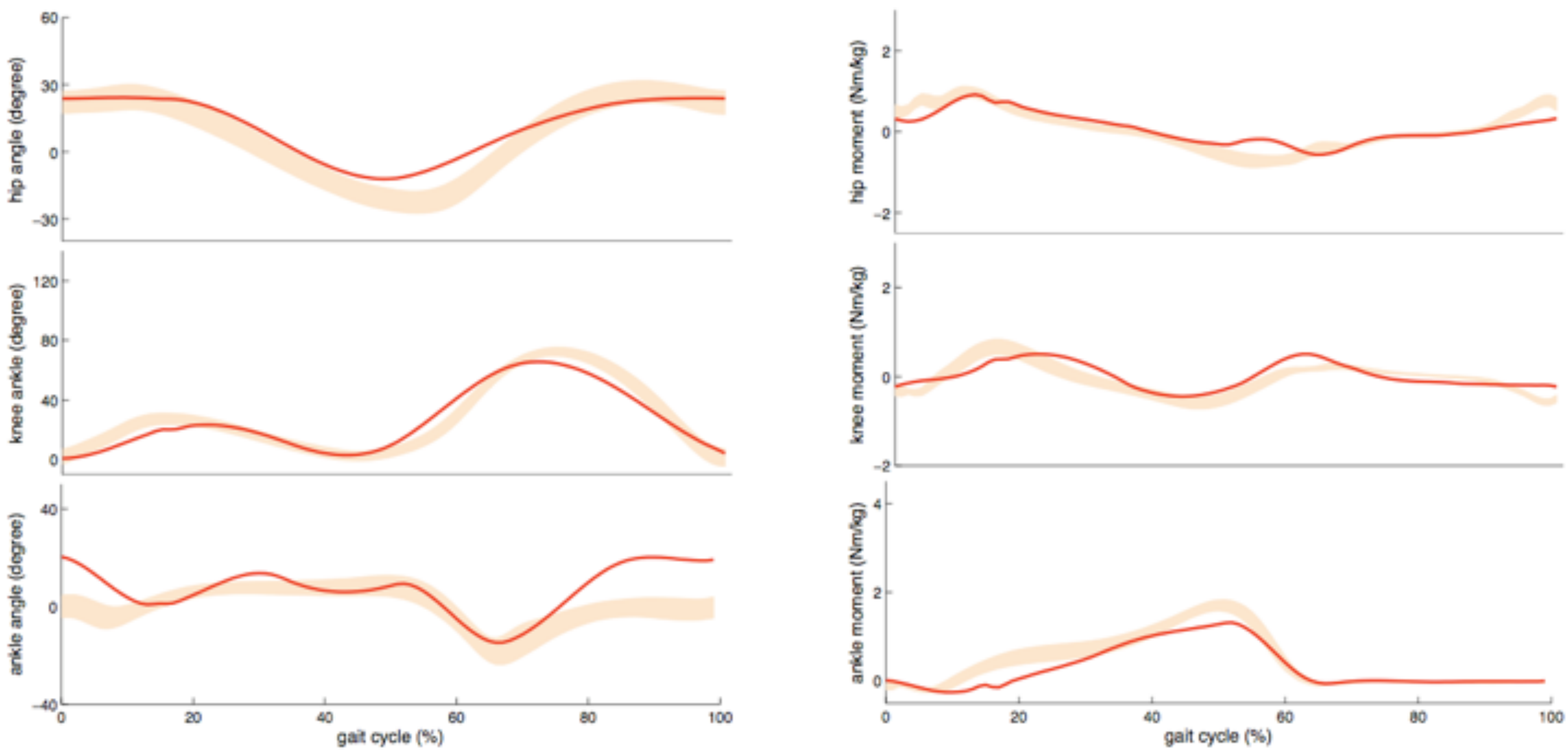
Humanoid Model with MTUs

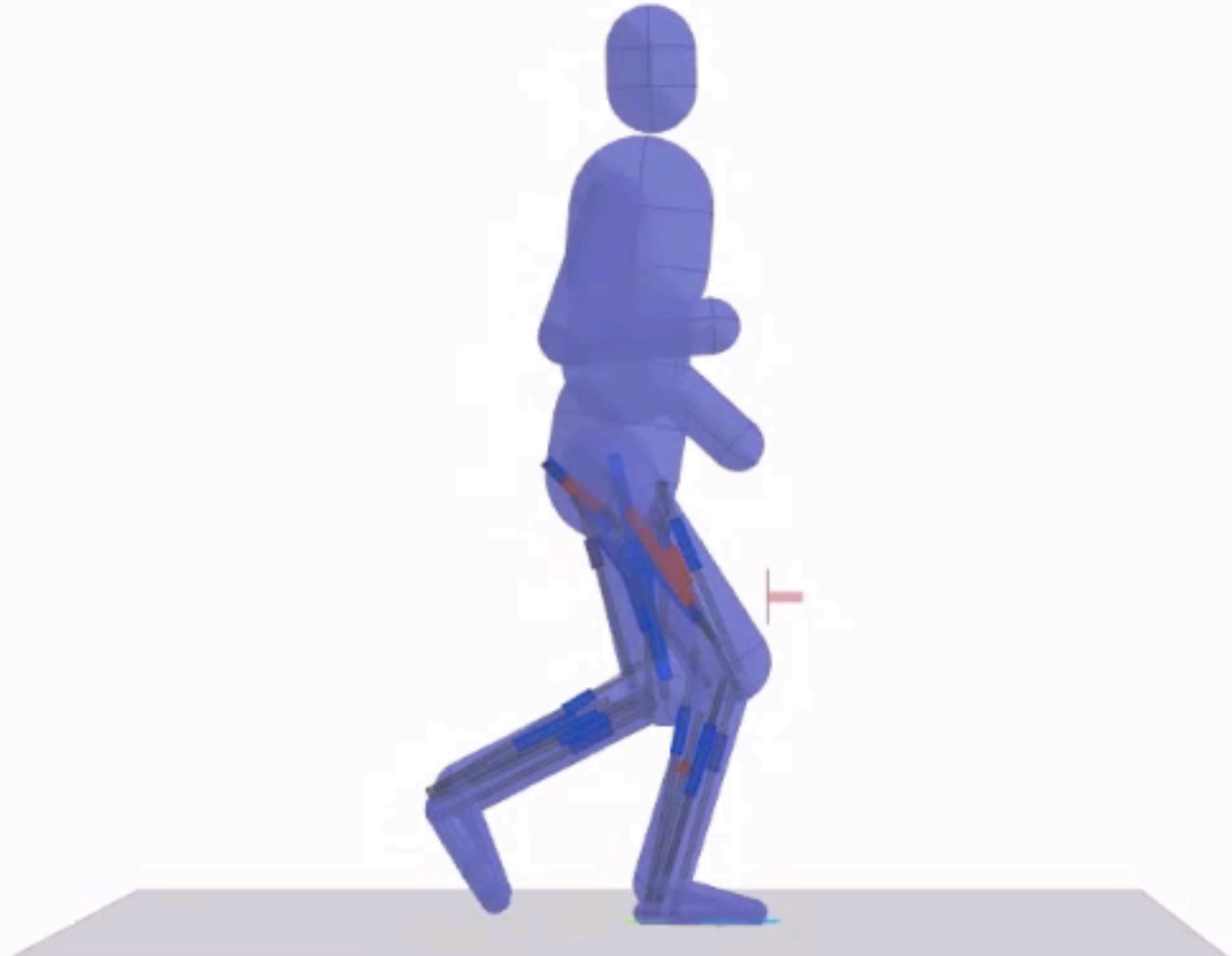


1.5m/s Walking

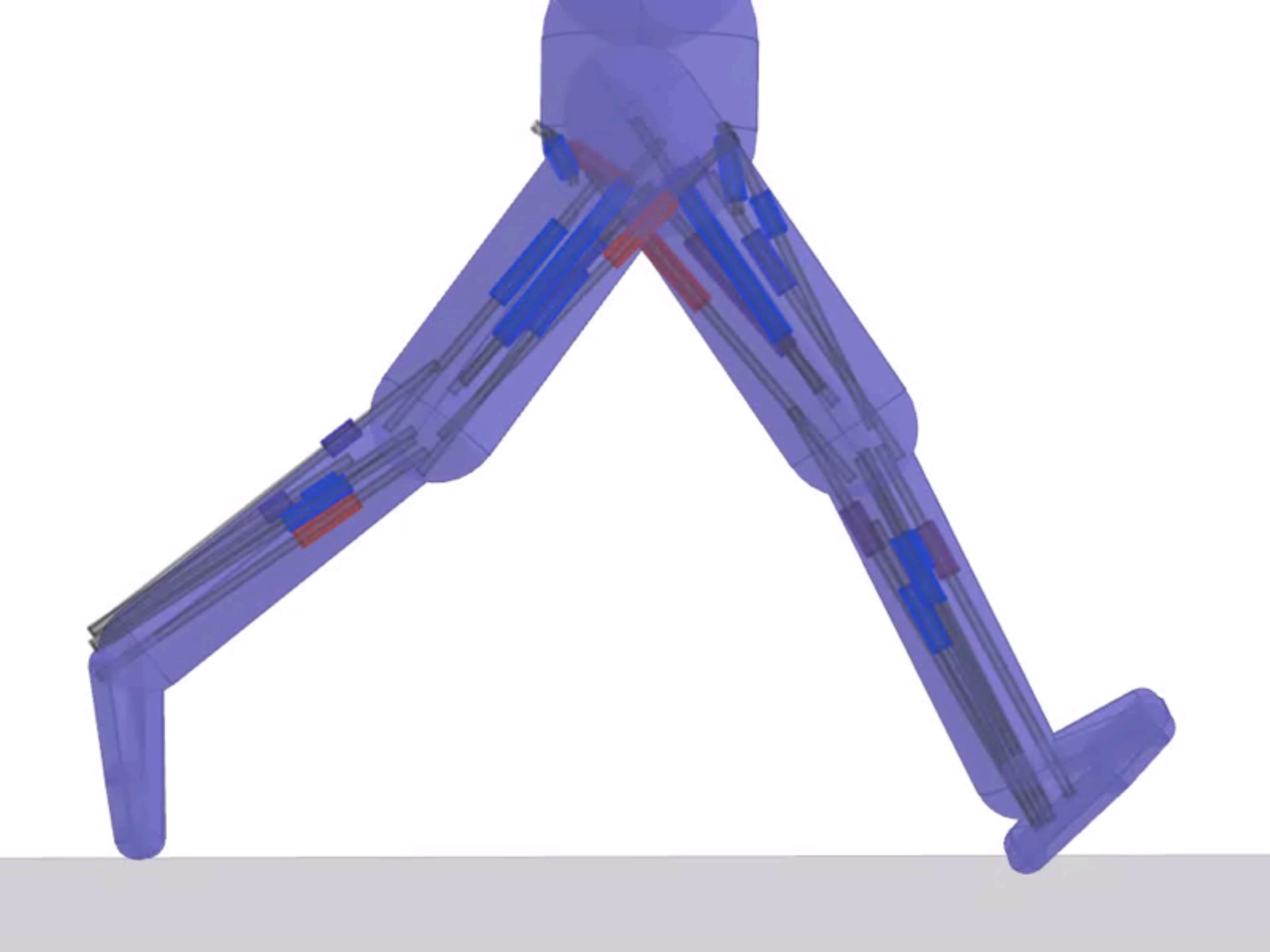


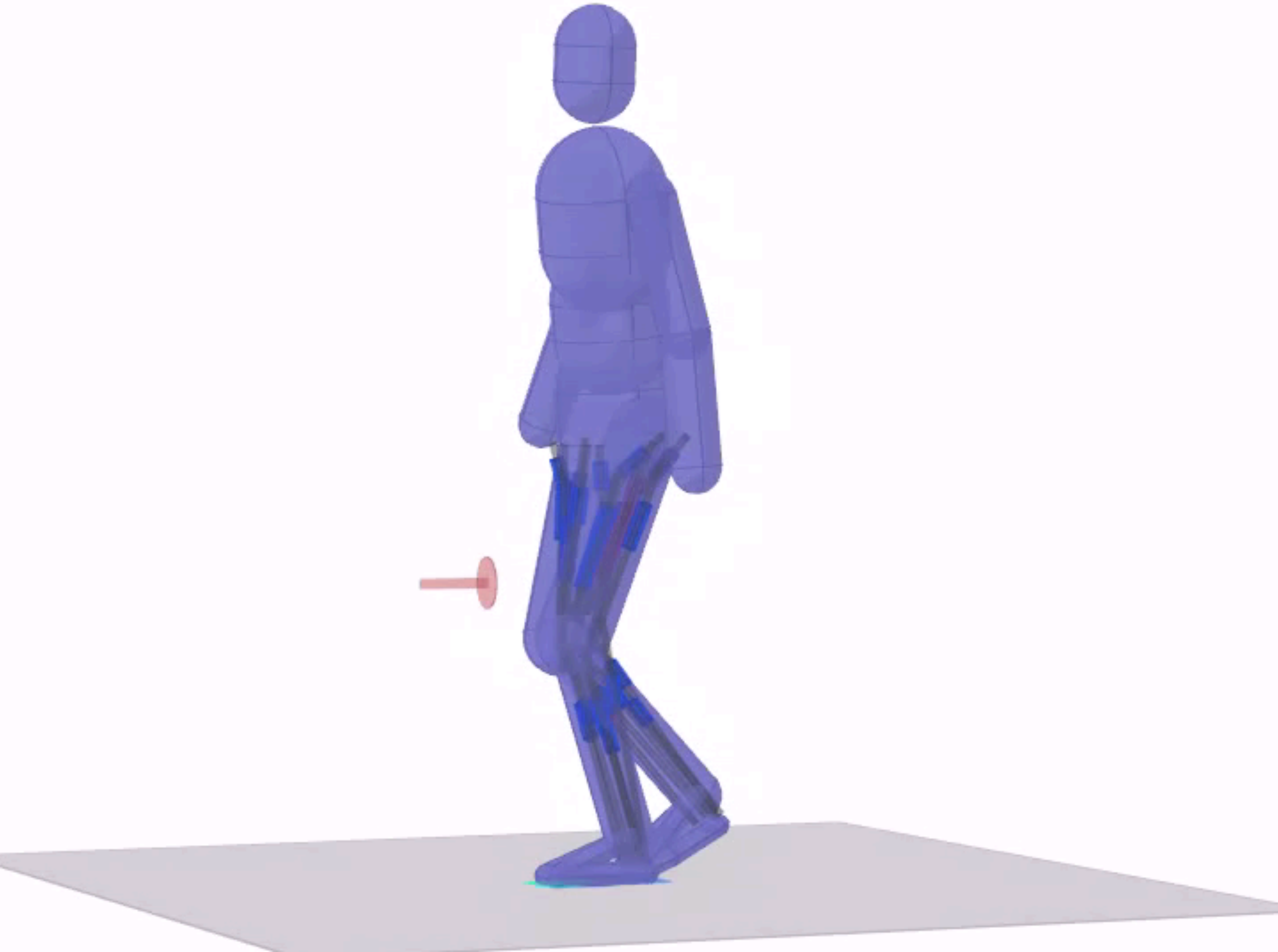
1.5m/s Walking Kinematics and Torques



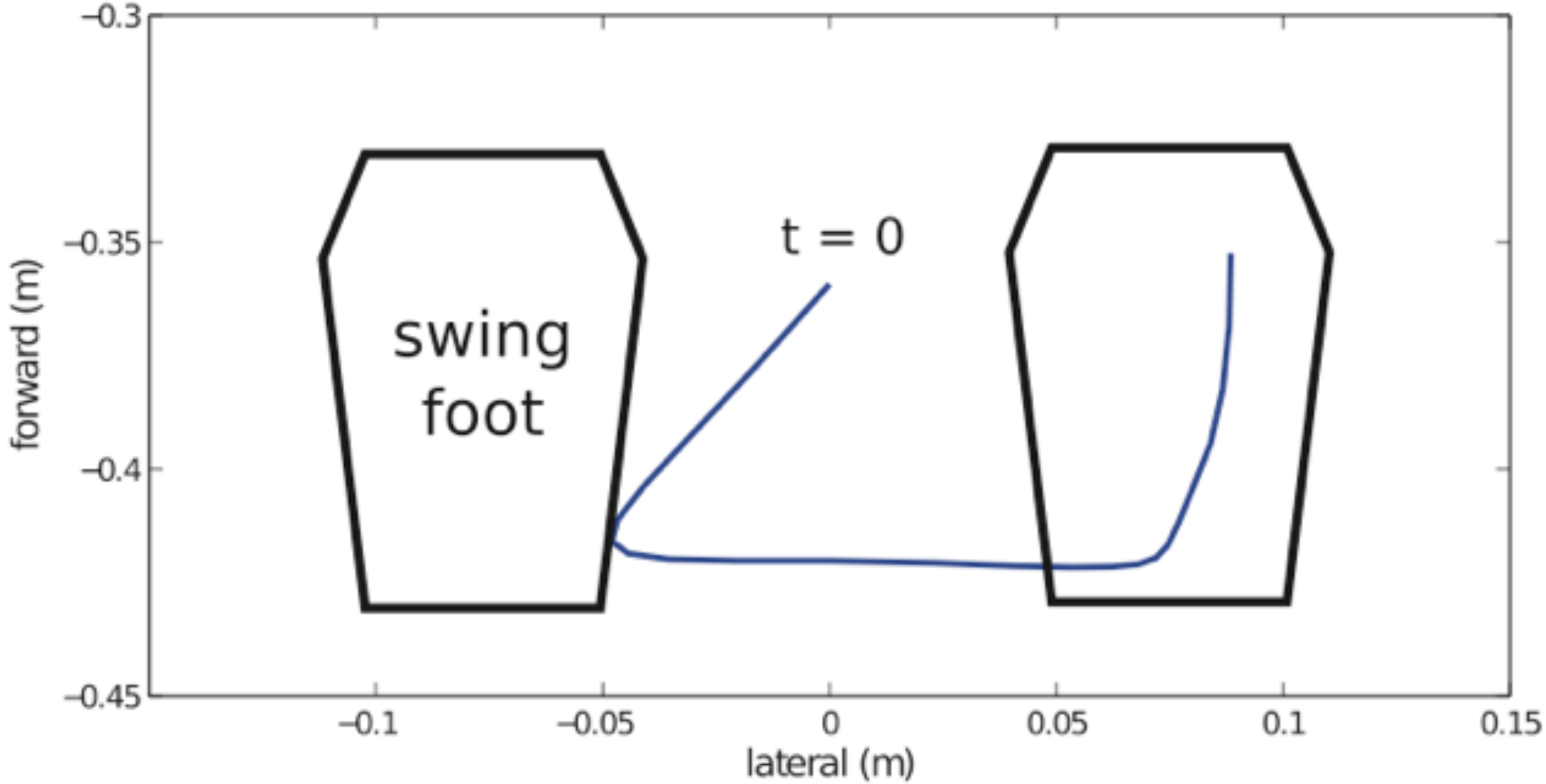


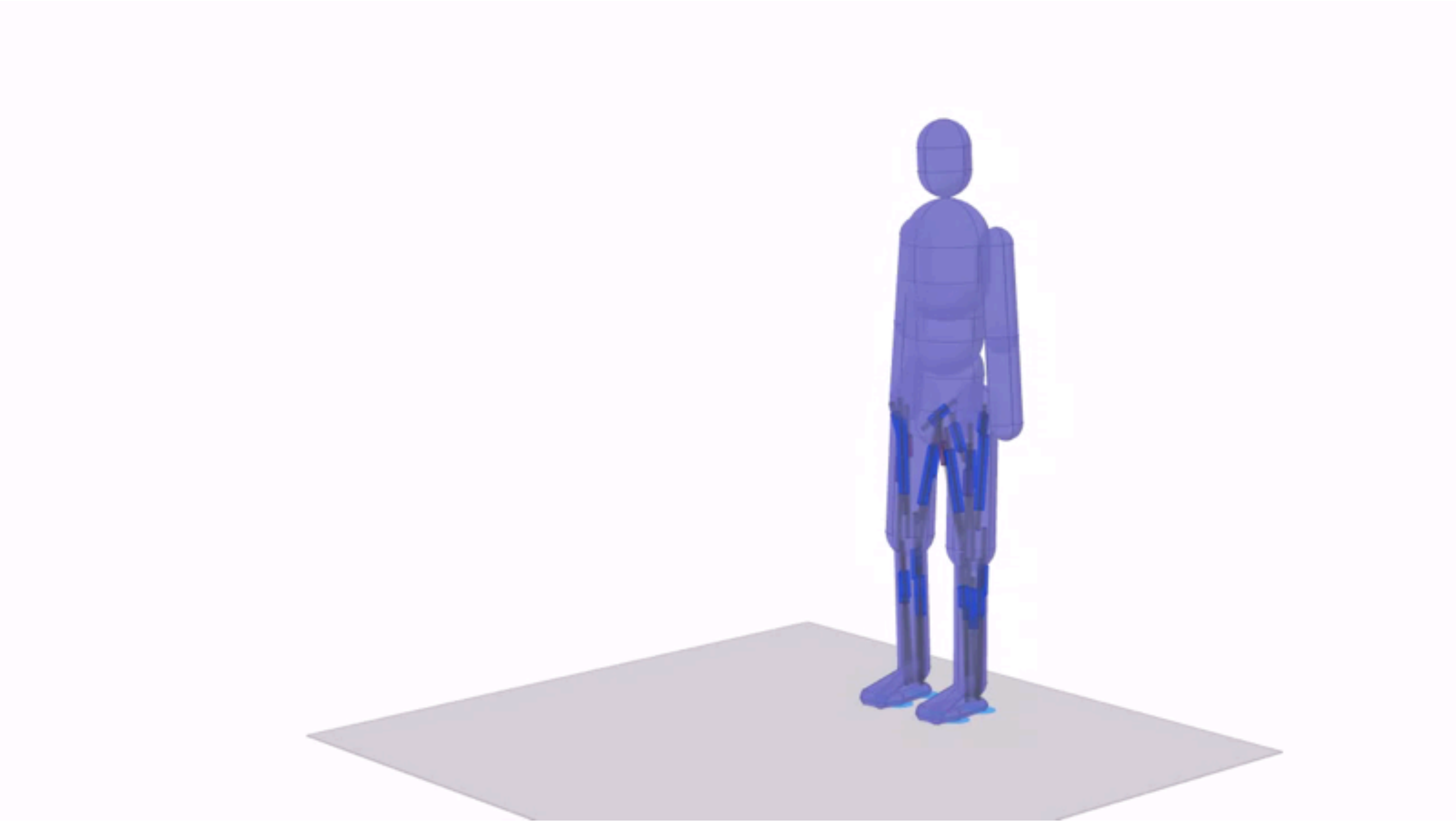
4m/s Running

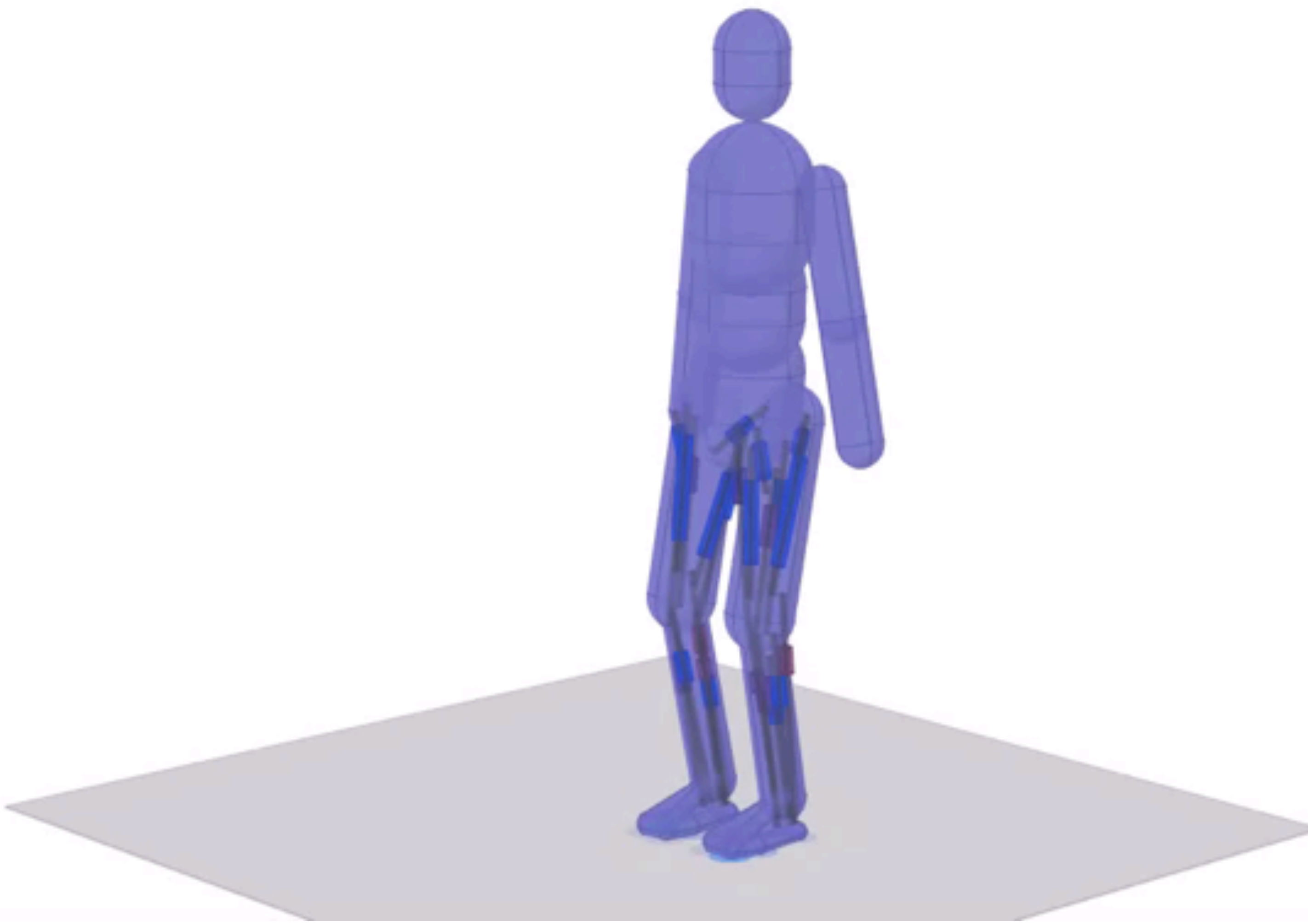


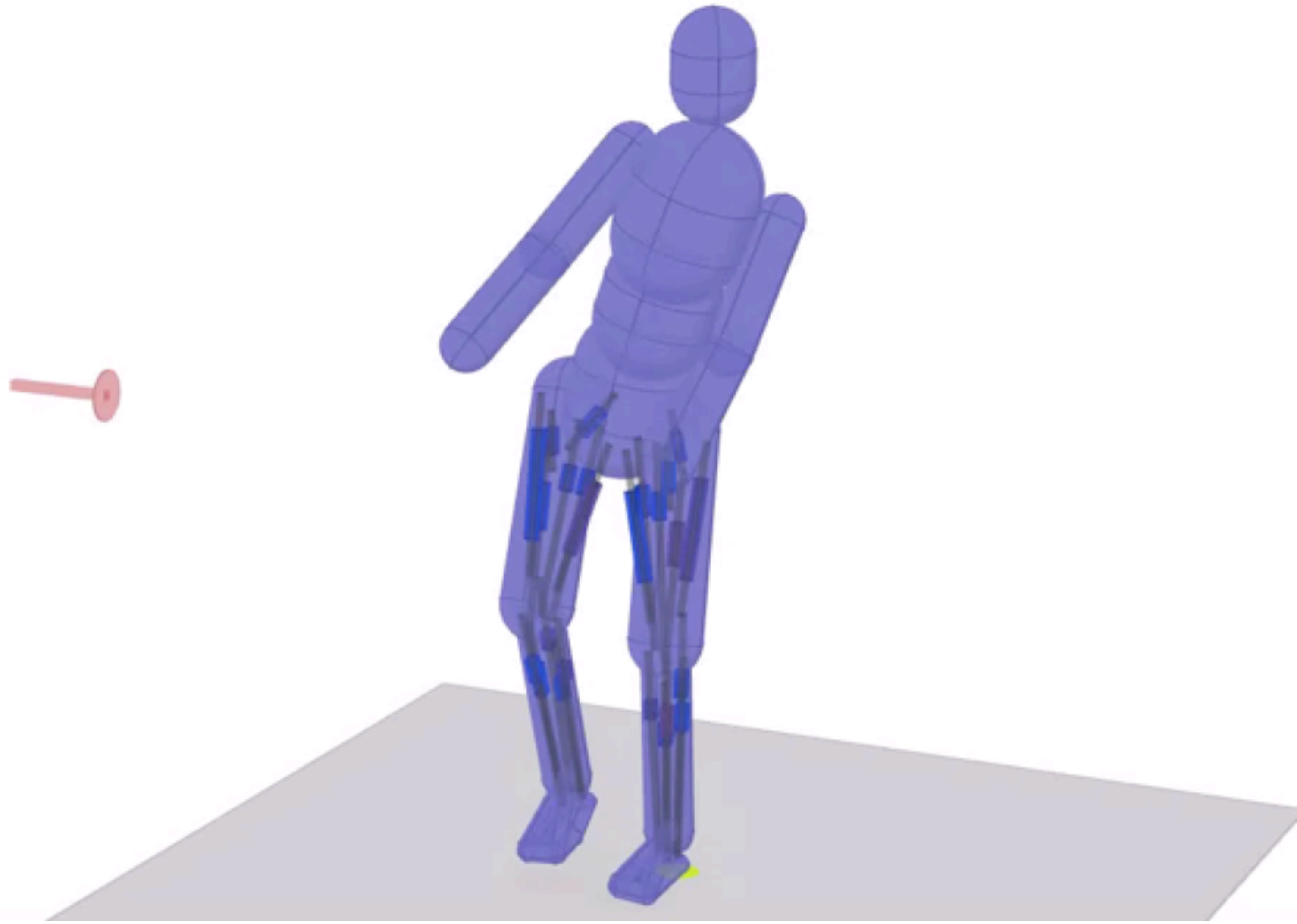


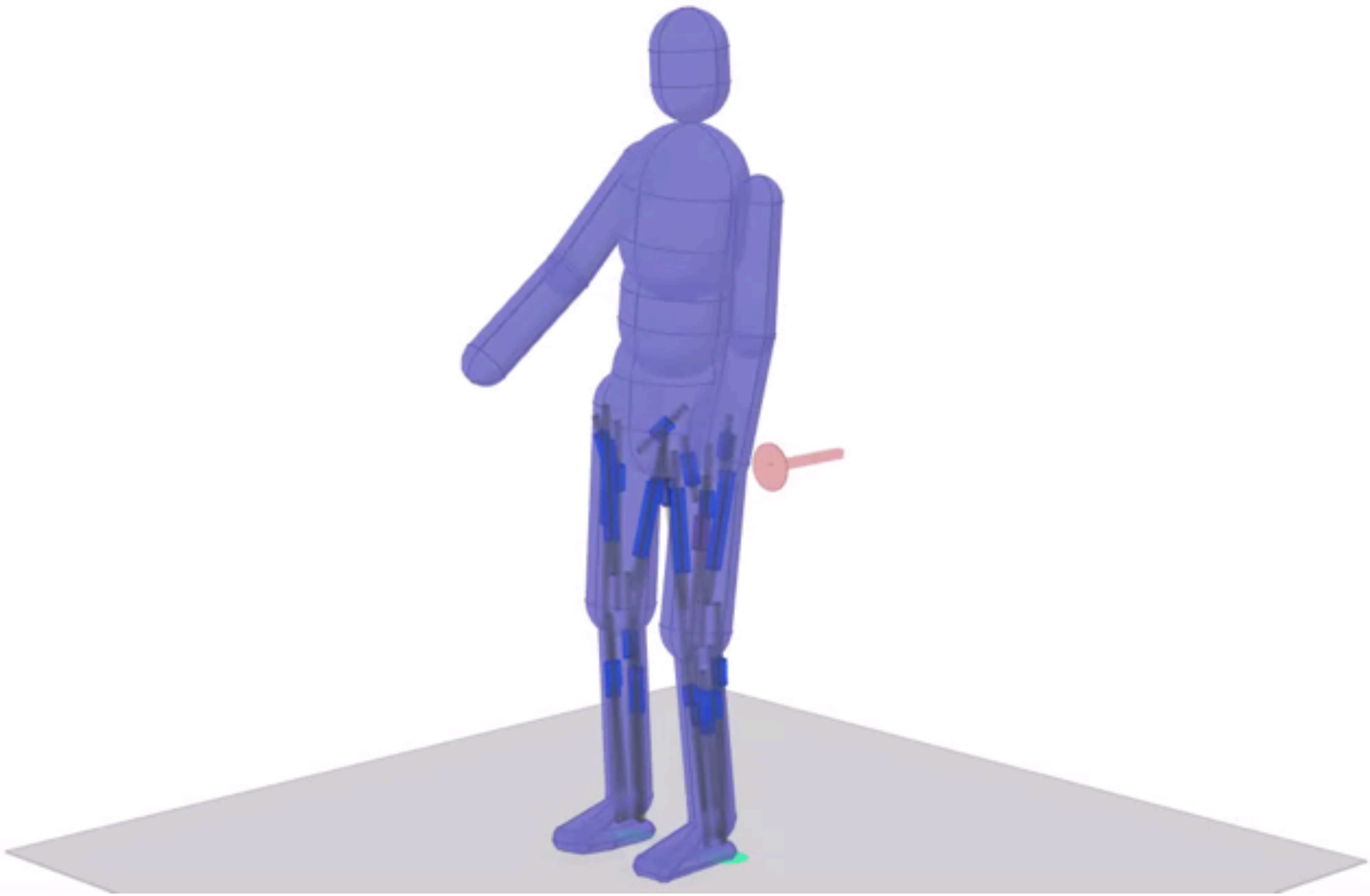
Gait Initiation COP Trajectory

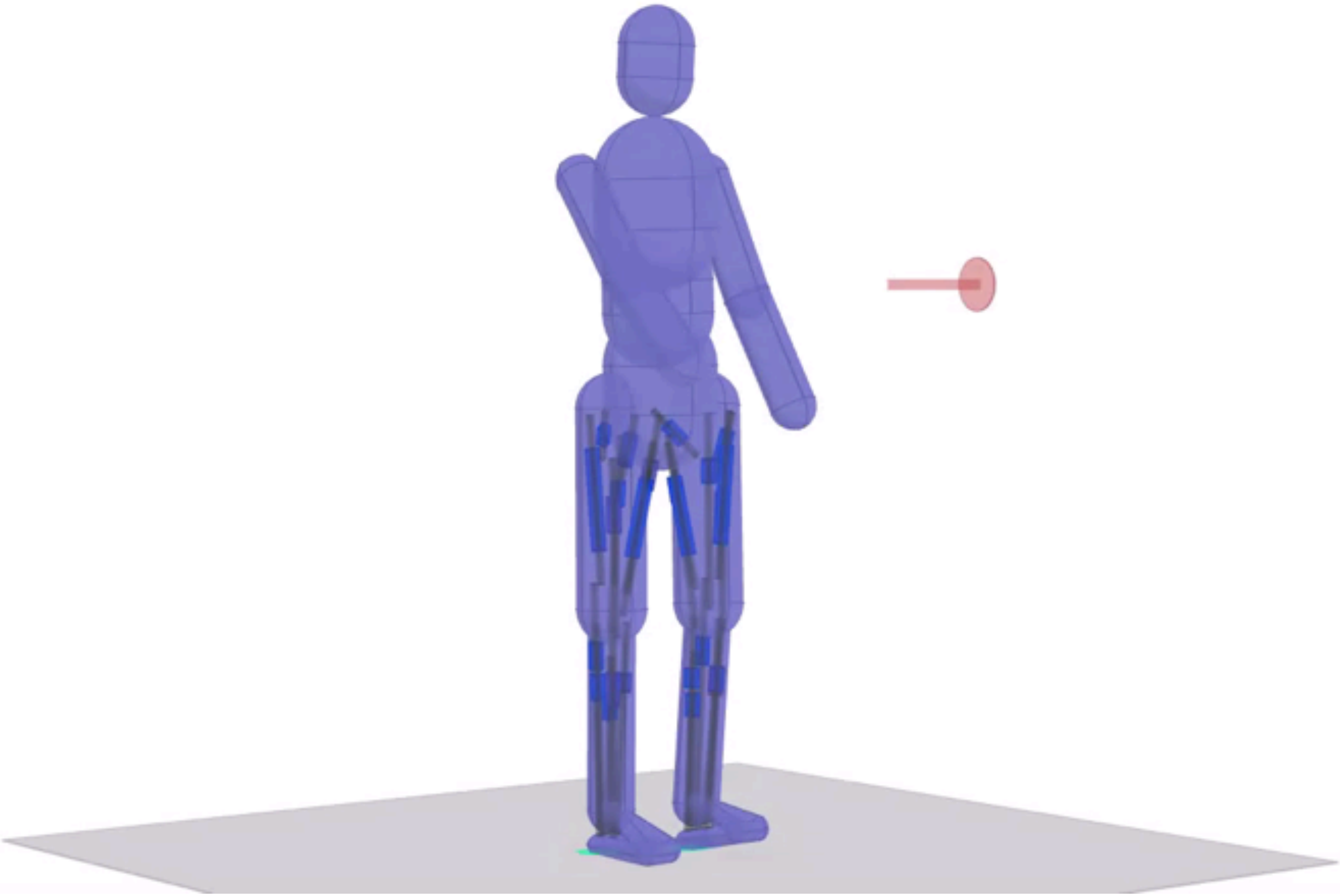




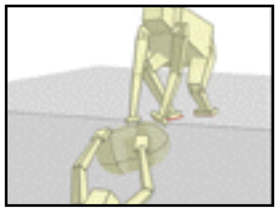




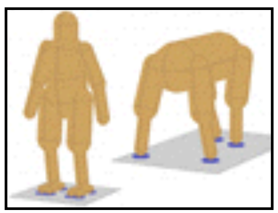




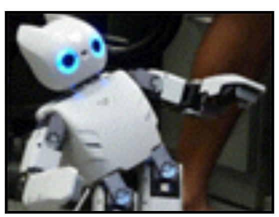
Trust Region Policy Optimization



Trajectory Optimization with Direct Collocation

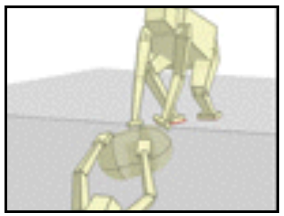
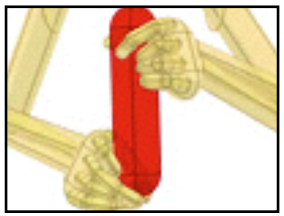
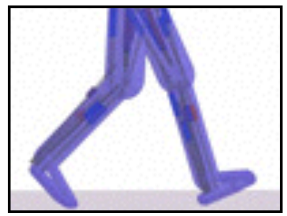
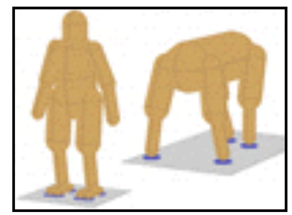
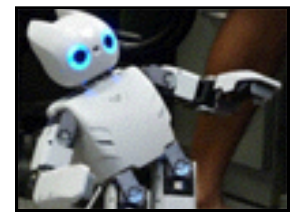
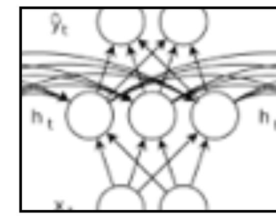


Learning Control Policies with Direct Collocation



Unknown/Uncertain Dynamics and Applications

Thank You!

Emo Todorov, Zoran Popovic, Pieter Abbeel, Vladlen Koltun, Aaron Hertzmann, Michael Kass, Jack Wang, Kendall Lowrey, Galen Andrew, Nikhil Mishra, Clemens Eppner, Nikita Kitaev, Sachin Patil, Martin DeLasa