How to Walk:

from a (mostly) optimal control perspective

John Zhang
Robotic Exploration Lab
Robotics Institute, Carnegie Mellon University

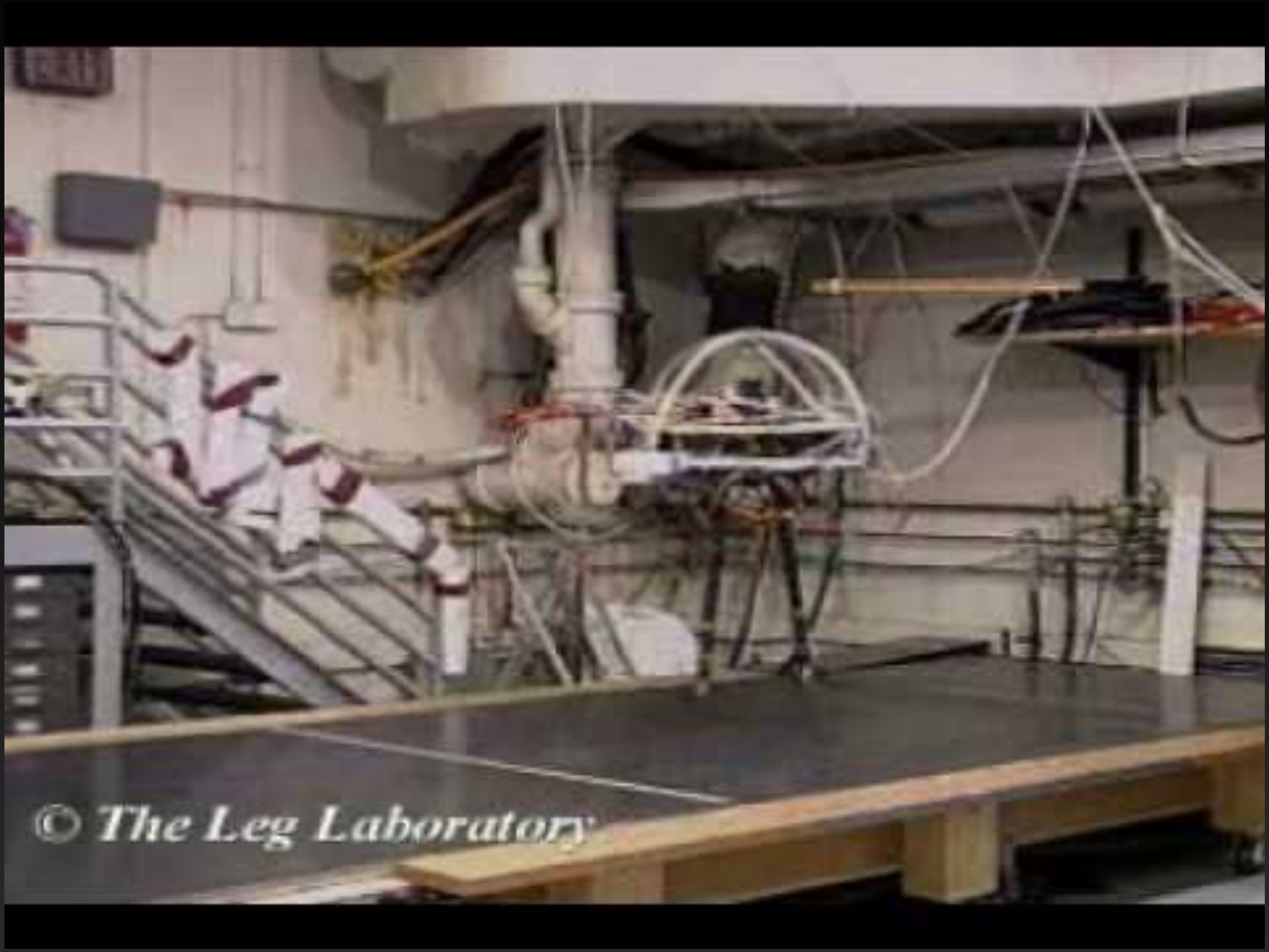
Optimal Control and Reinforcement Learning, 4.1.2025

logistics

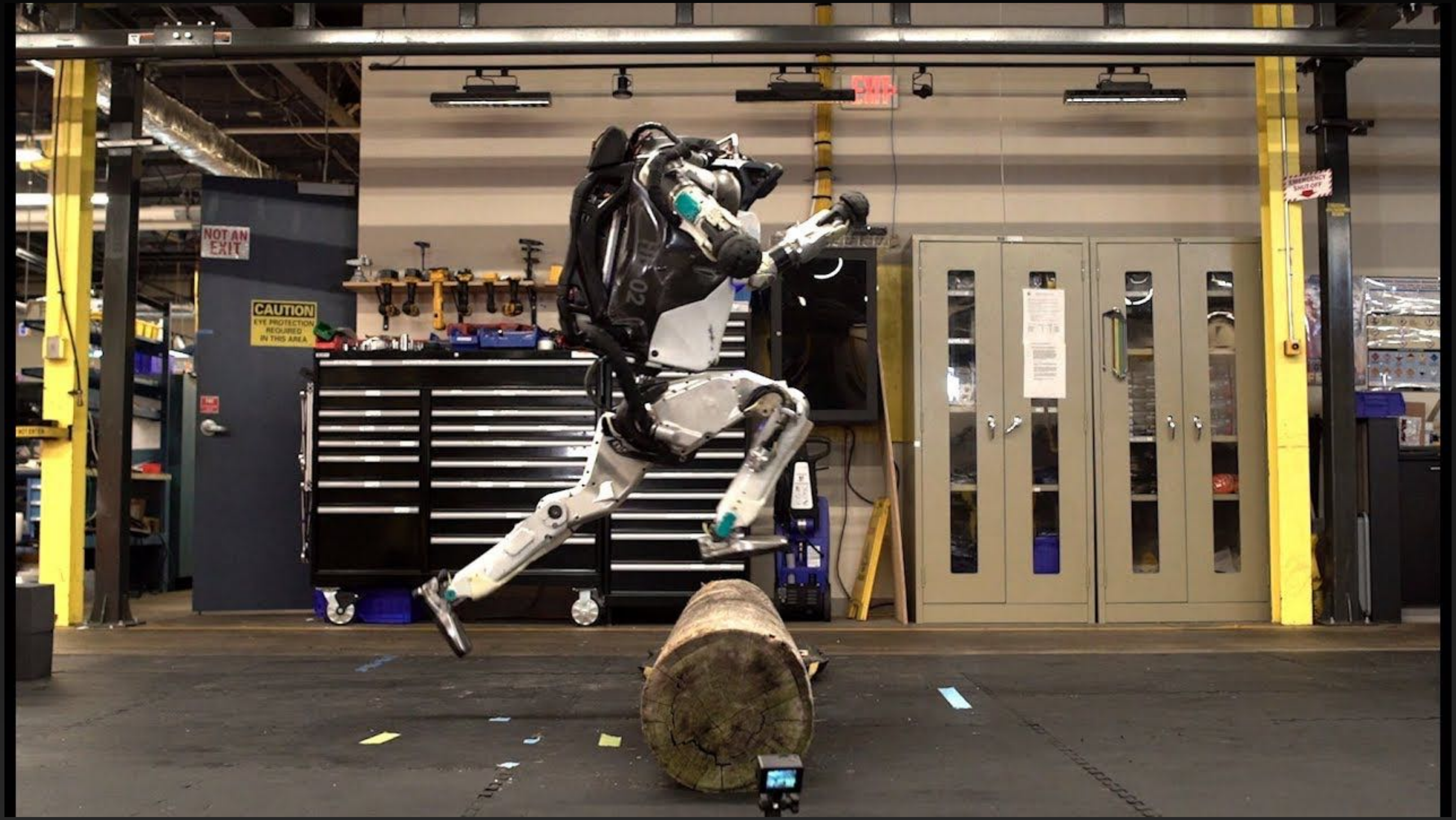
- guest lecture today, no class on Thursday
- no office hours this evening. I will host office hours after class in my office (NSH 1502)

1960s to early 2000s — Honda Asimo

1960s to early 2000s — CMU -> MIT Leg Lab



2010s — Boston Dynamics Atlas



2010s — MIT Cheetah

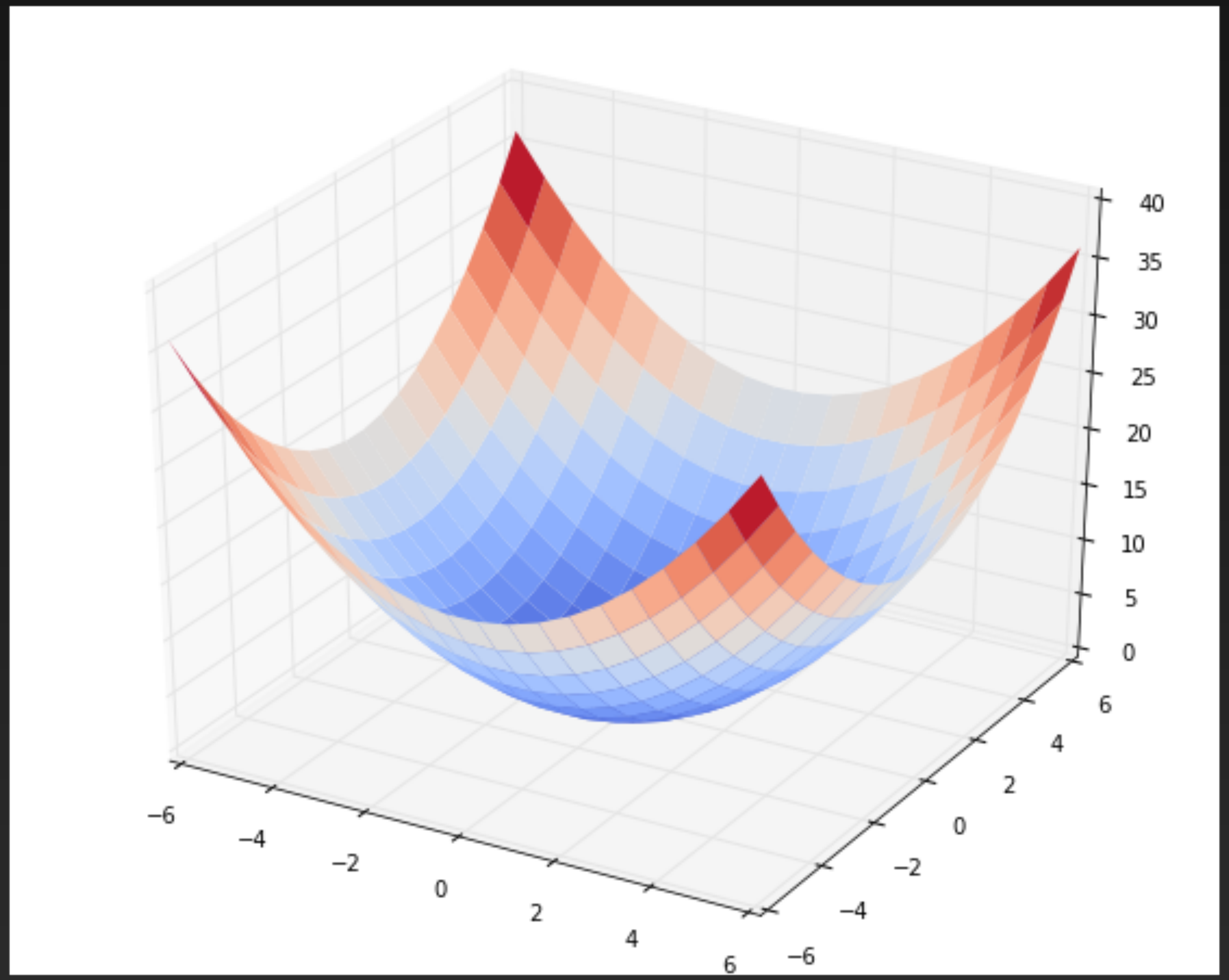
2025 — reinforcement learning

history of walking robots a summary

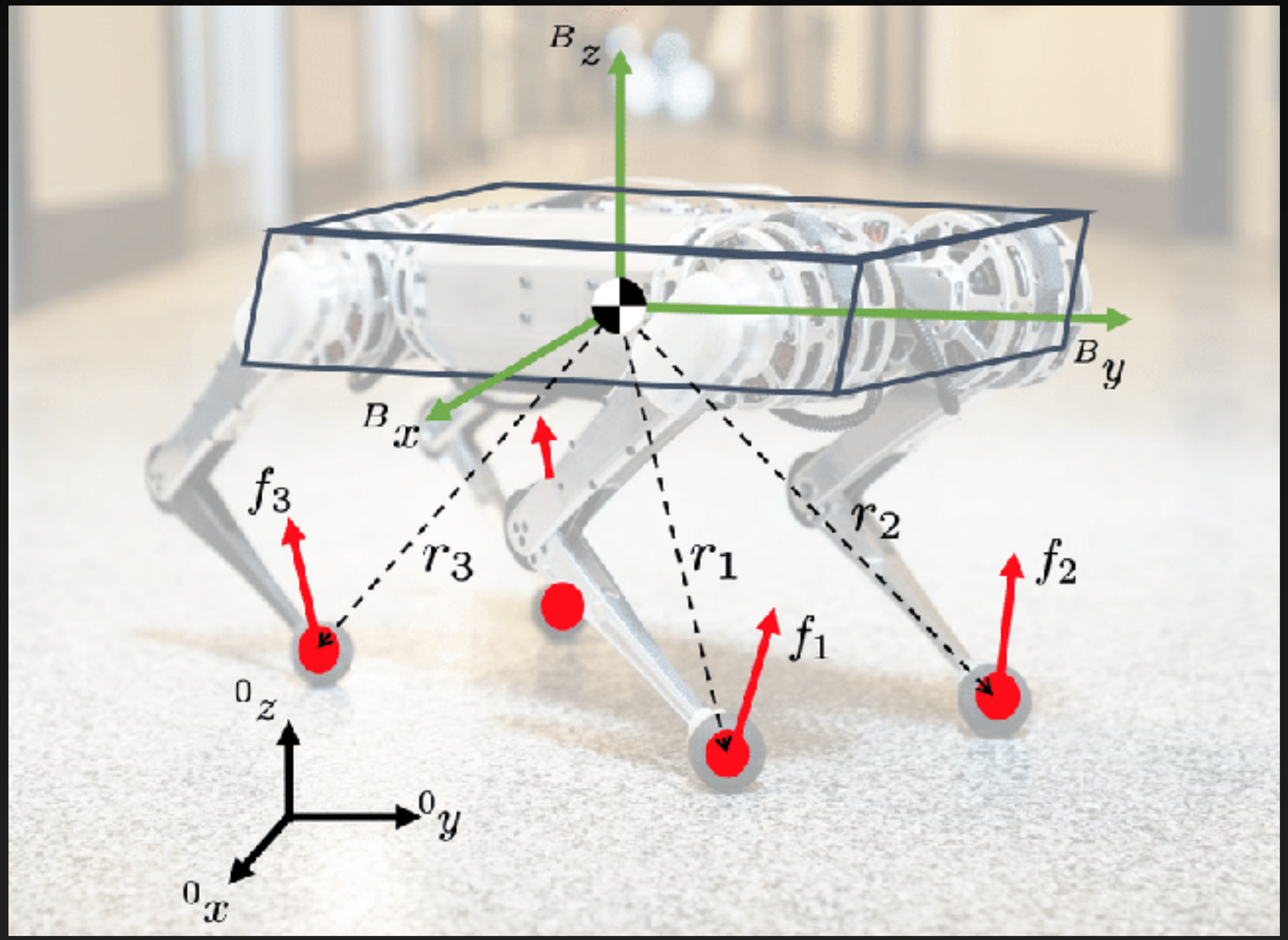
- first legged robots was built in ~1960s, serious research from ~1980s
- modeling approaches
  - inverted industrial arm: slow, quasi-static (Honda)
  - floating-based dynamics: dynamic locomotion (Raibert/CMU/MIT)
- actuation methods
  - hydraulic actuators used to be very common
  - these days direct-drive electric motors are much better
- control strategy
  - MPC was the state-of-the-art in the 2010s
  - since ~2021/2022 RL methods have become the standard

convex MPC

minimize u	quadratic cost
subject to	linear robot dynamics
	linear constraints

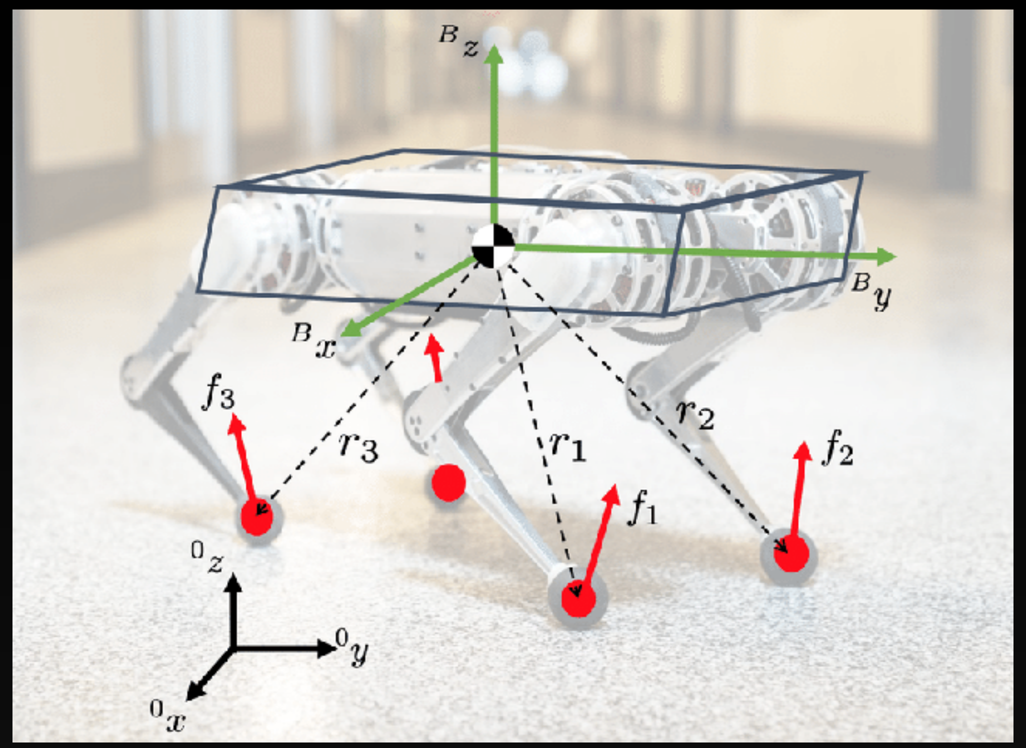


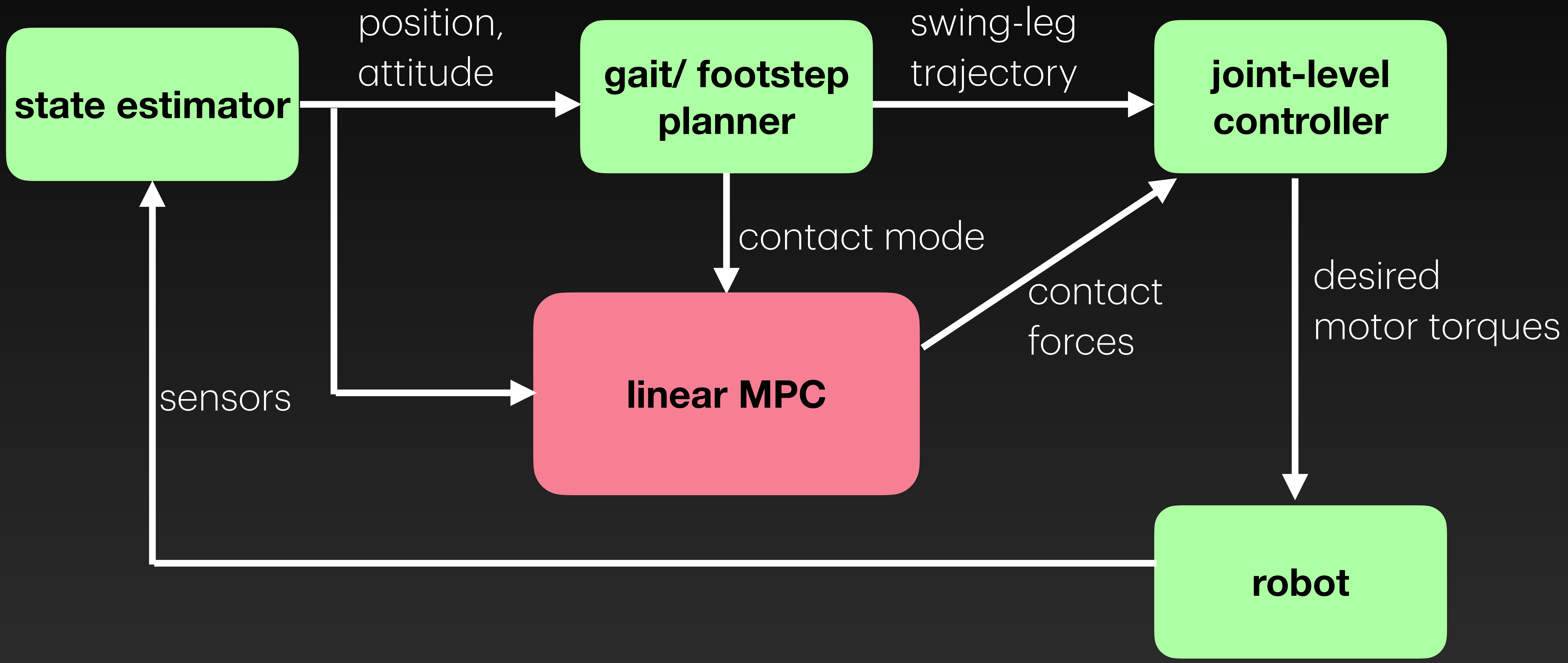
walking with convex MPC — single-rigid body model



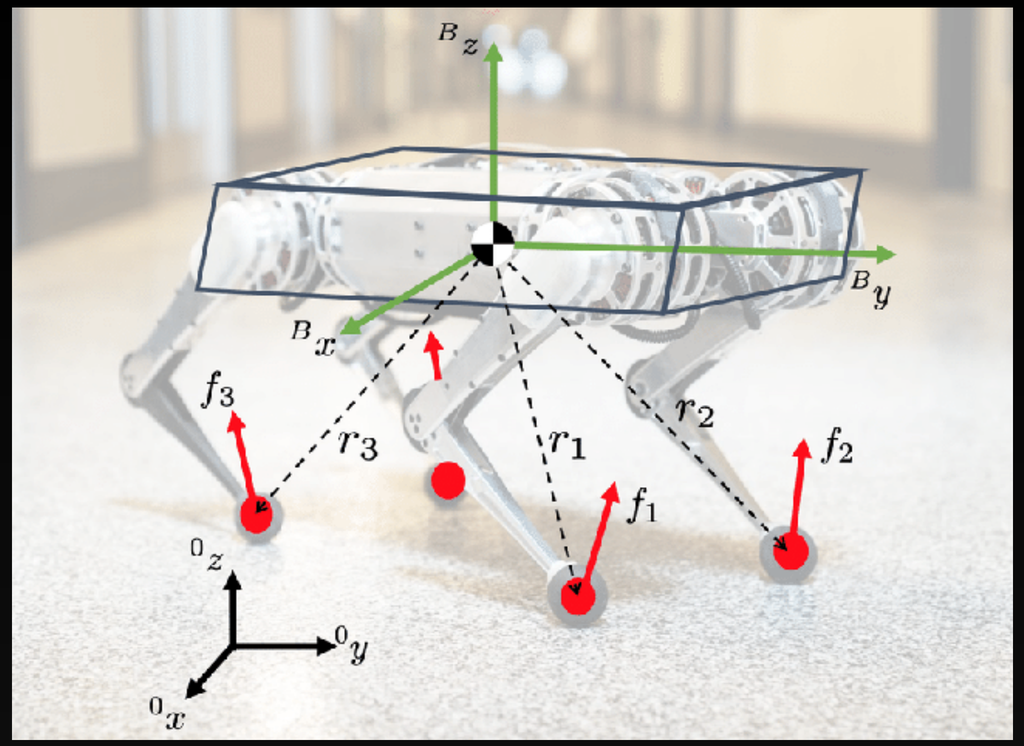
aka. lumped mass model/potato model etc.

walking with convex MPC





walking with convex MPC — assumptions



- leg mass and inertia are relative insignificant compare to the body
- body angular velocities and pitch/roll are small (small angle approximation)
- foot positions can track the reference pretty well

walking with convex MPC

- computationally inexpensive (hundreds of Hz on on-board computers)
- (linearized) single-rigid body assumptions can be limiting
- enhancements include using nonlinear single-rigid body or whole-body models (at the cost of more compute at run time, no longer convex)

walking with nonlinear MPC — iLQR

1x

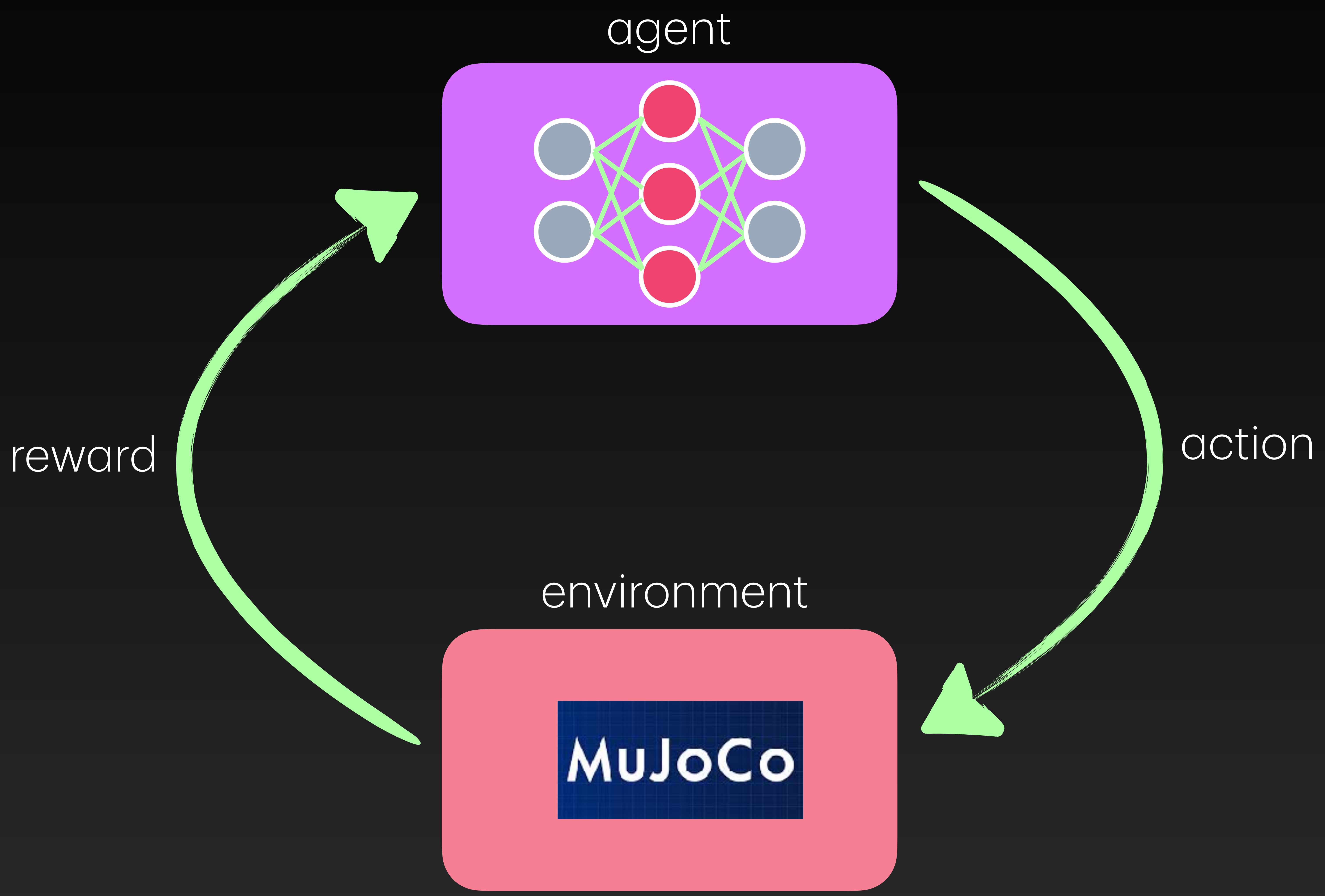
walking with iLQR — takeaways

- iLQR doesn't naturally handle additional constraints, contacts constraints are implicitly enforced through forward simulation with contact dynamics
- the algorithm dates back to the 1960s, computers just got faster
- we got a little better at modeling robots (rigid body and point-surface contact), but mostly computers got faster

walking with nonlinear MPC — still an active area of research

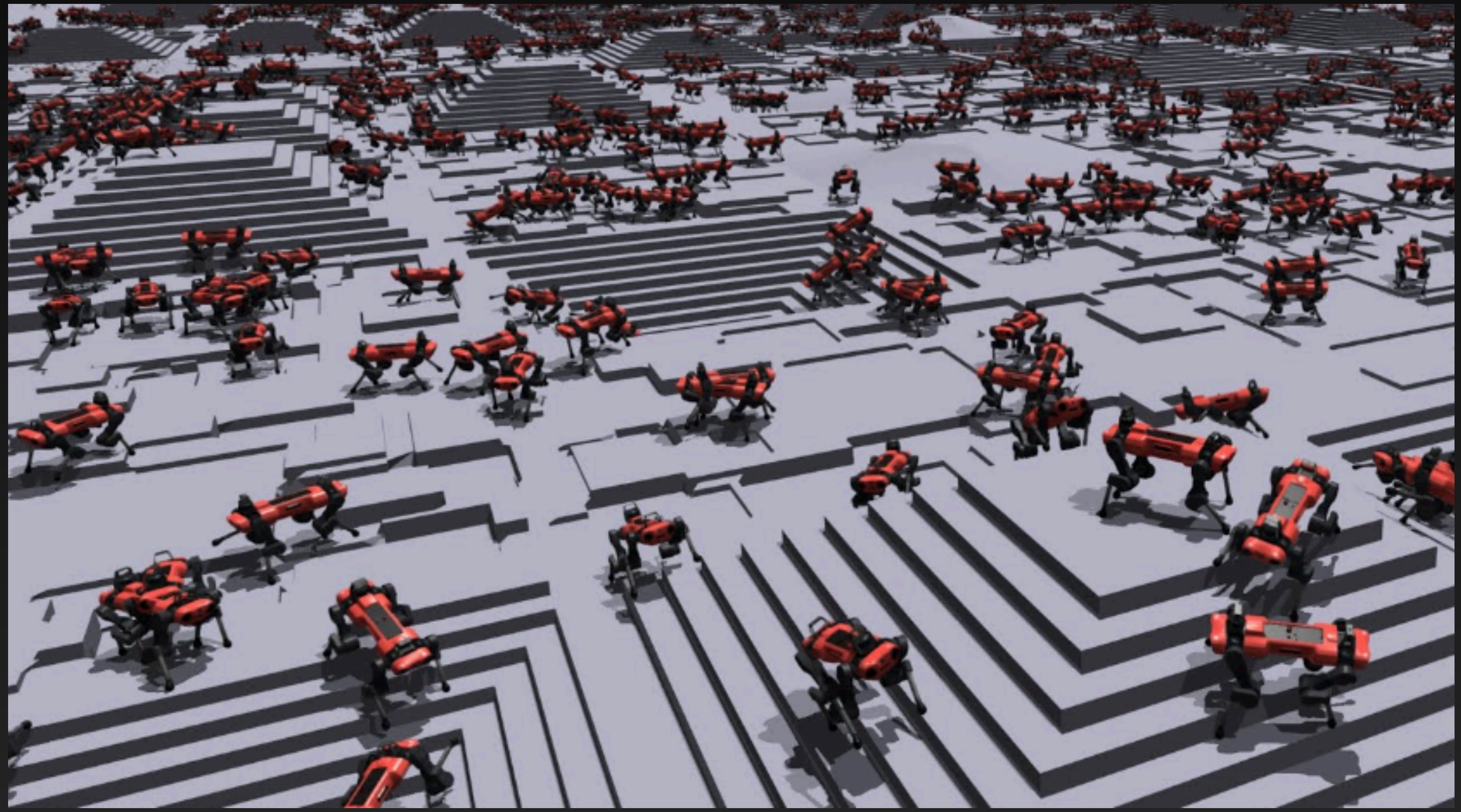
- nonlinear single-rigid body model, whole-body dynamics
- iLQR vs SQP
- differentiating through contact dynamics, reasoning over contact modes in the optimization problem (so called "contact-implicit" methods)

reinforcement learning



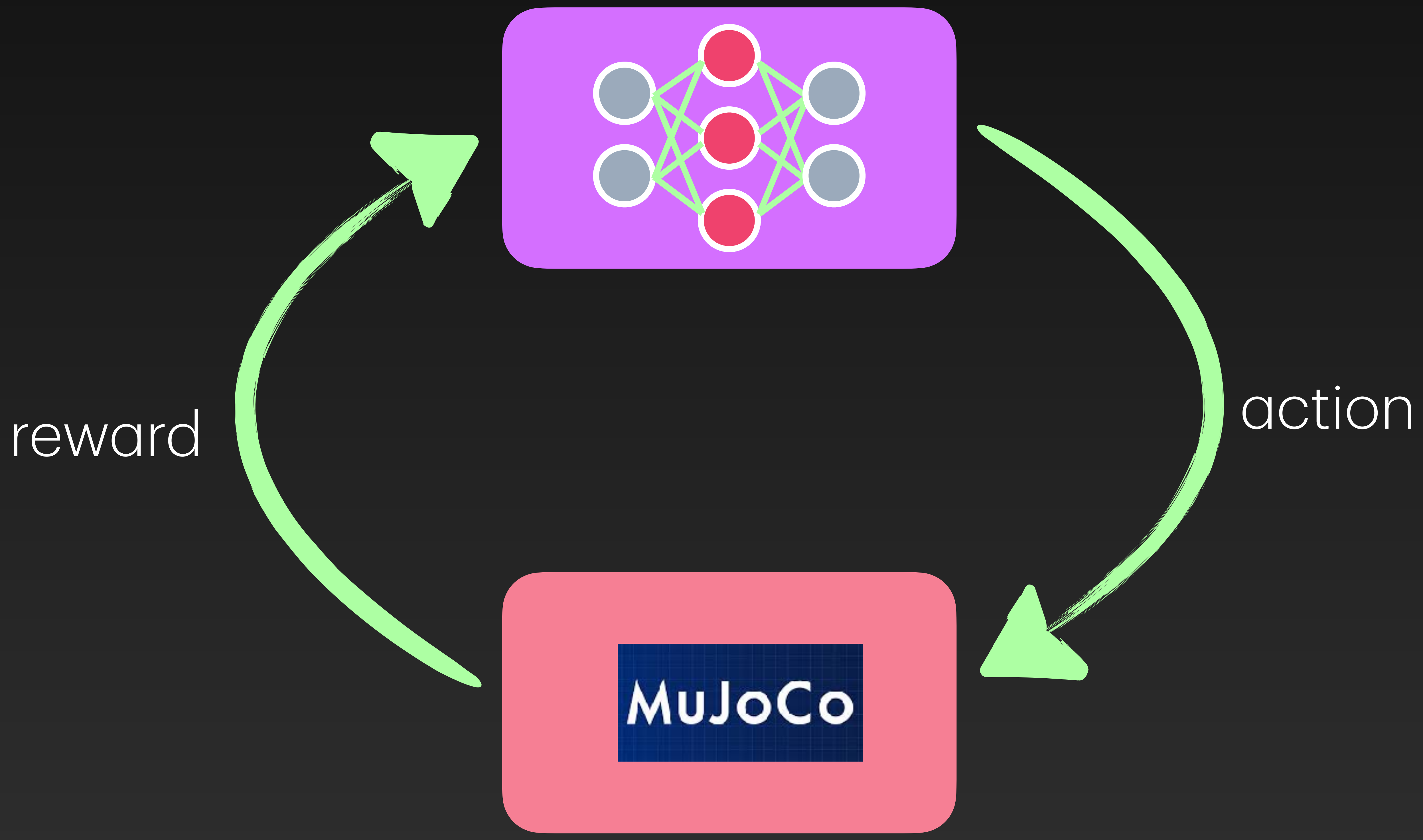
learning to walk with (sim-to-real) reinforcement learning

- massively parallel simulation on the GPU



learning to walk with (sim-to-real) reinforcement learning

- massively parallel simulation on the GPU
- modern robotics simulators are “good enough” for locomotion
- amortized compute offline, cheap to inference online



learning to walk with (sim-to-real) reinforcement learning

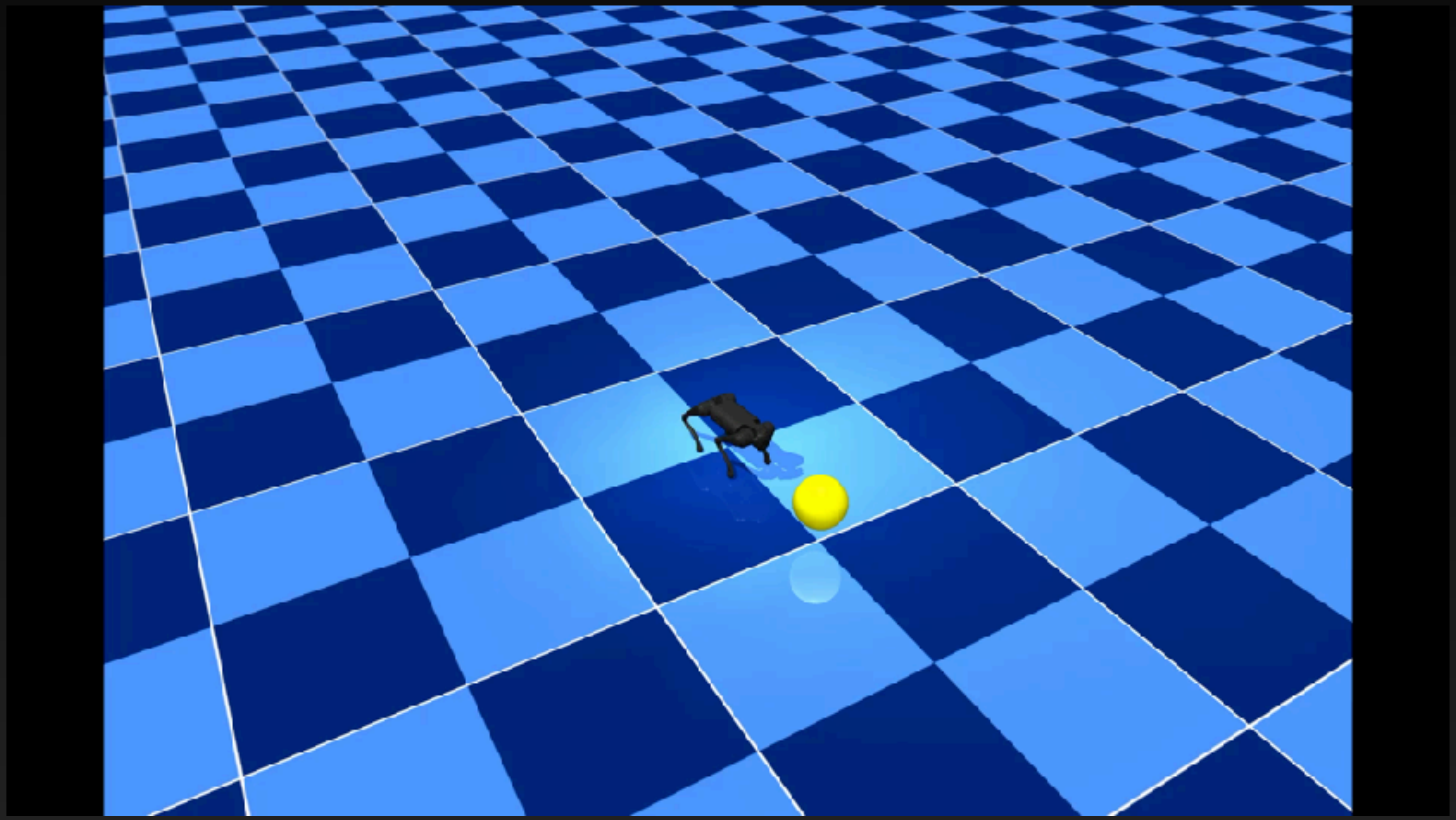
- massively parallel simulation on the GPU
- modern robotics simulators are "good enough" for locomotion
- amortized compute offline, cheap to inference online
- much richer representation (vision based, domain randomization, sensor to torques, etc.)

open research question:
how can we leverage parallel computation at test time?

model-predictive path integral control (MPPI)

MPPI algorithm:
- sample actions from some distribution
- evaluate the samples, pick the best one
- update the distribution mean

repeat



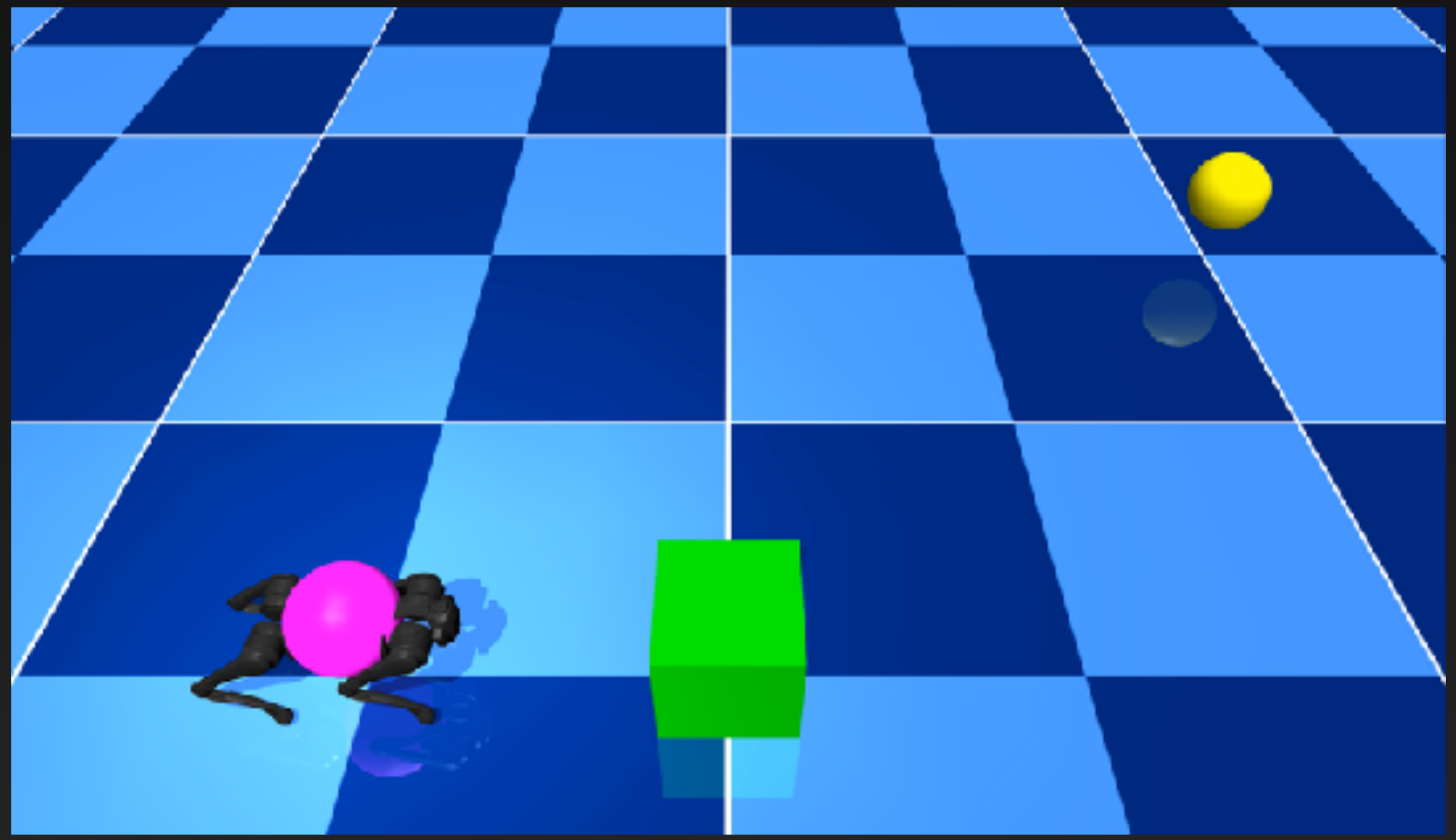
MPPI

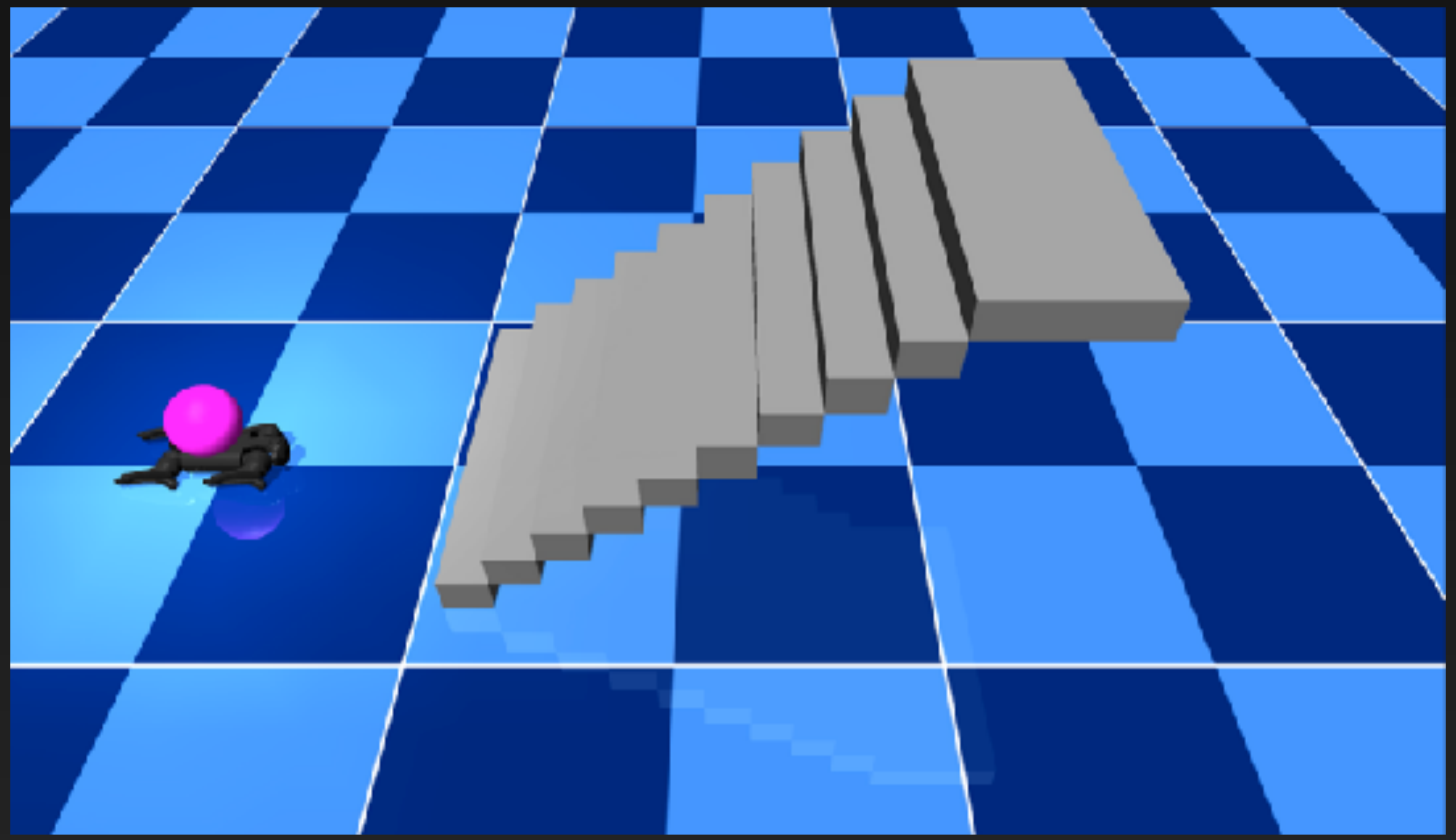
pro:
parallelization friendly
derivative free
naturally incorporates data-driven black-box models

con:
curse of dimensionality

walking (and manipulating) with MPPI

- simulators today give you (almost) everything you can model
- real-time reasoning over full-order models, nonlinear contact, and whole-body collision. without offline training

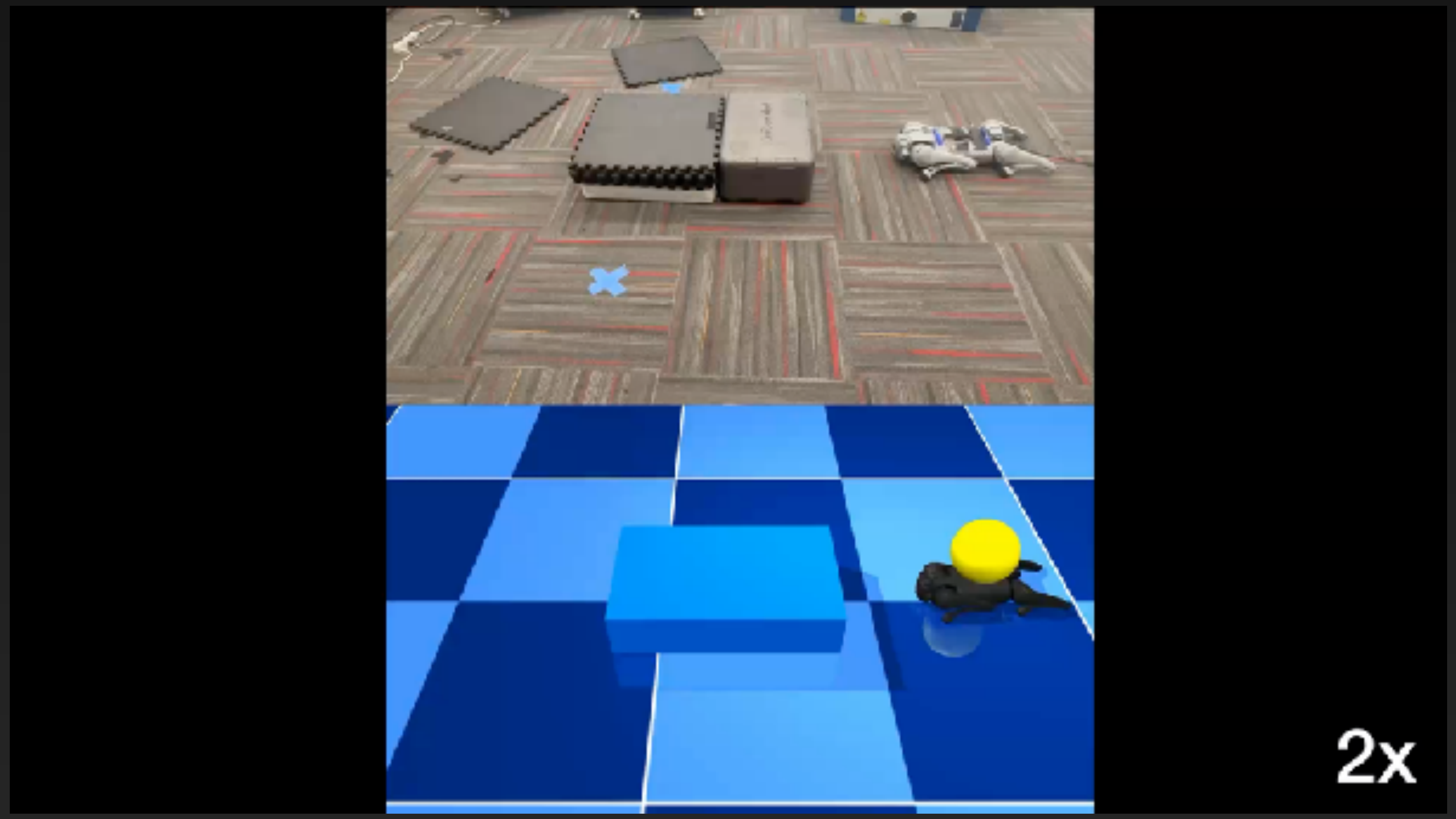


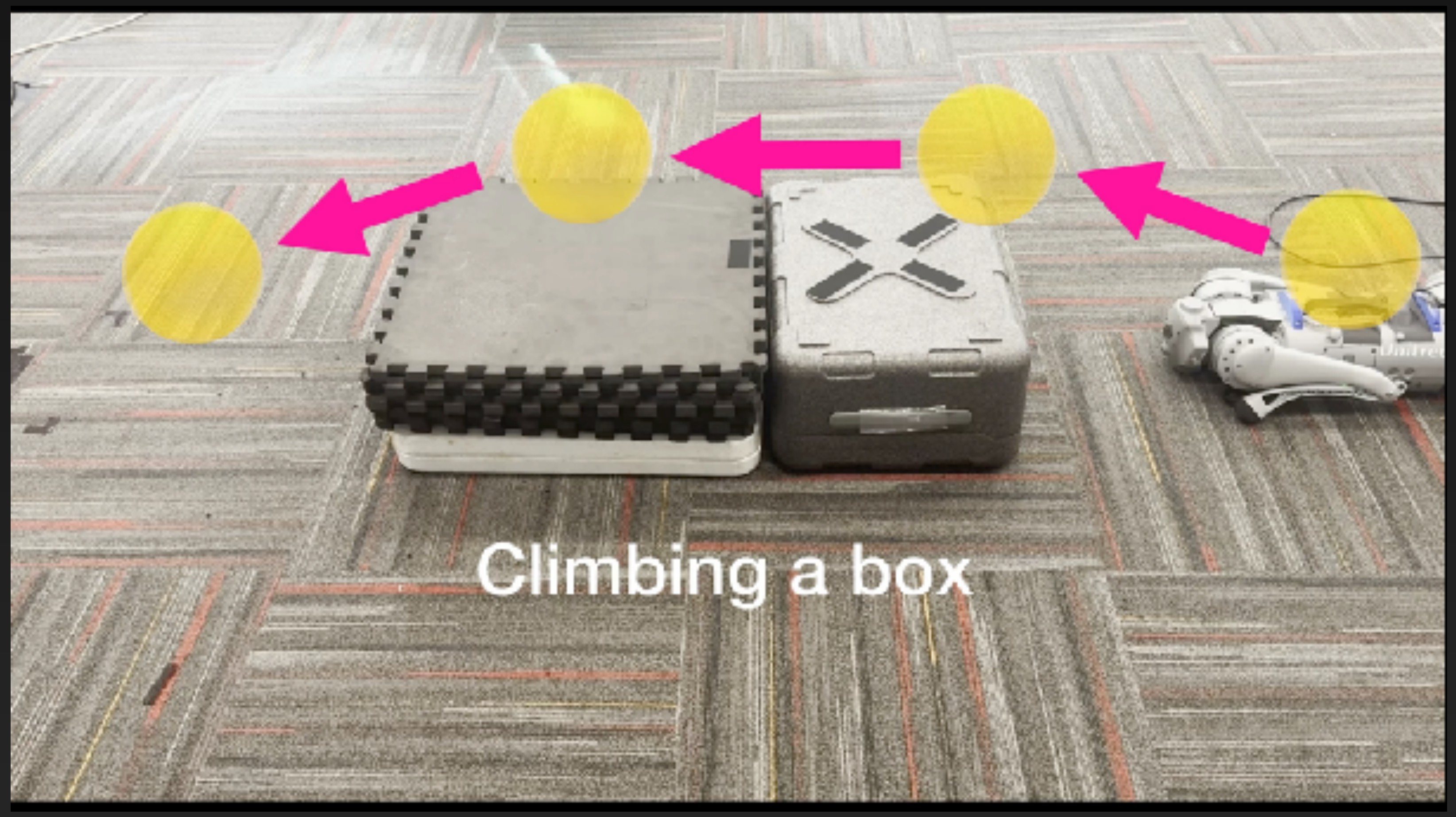


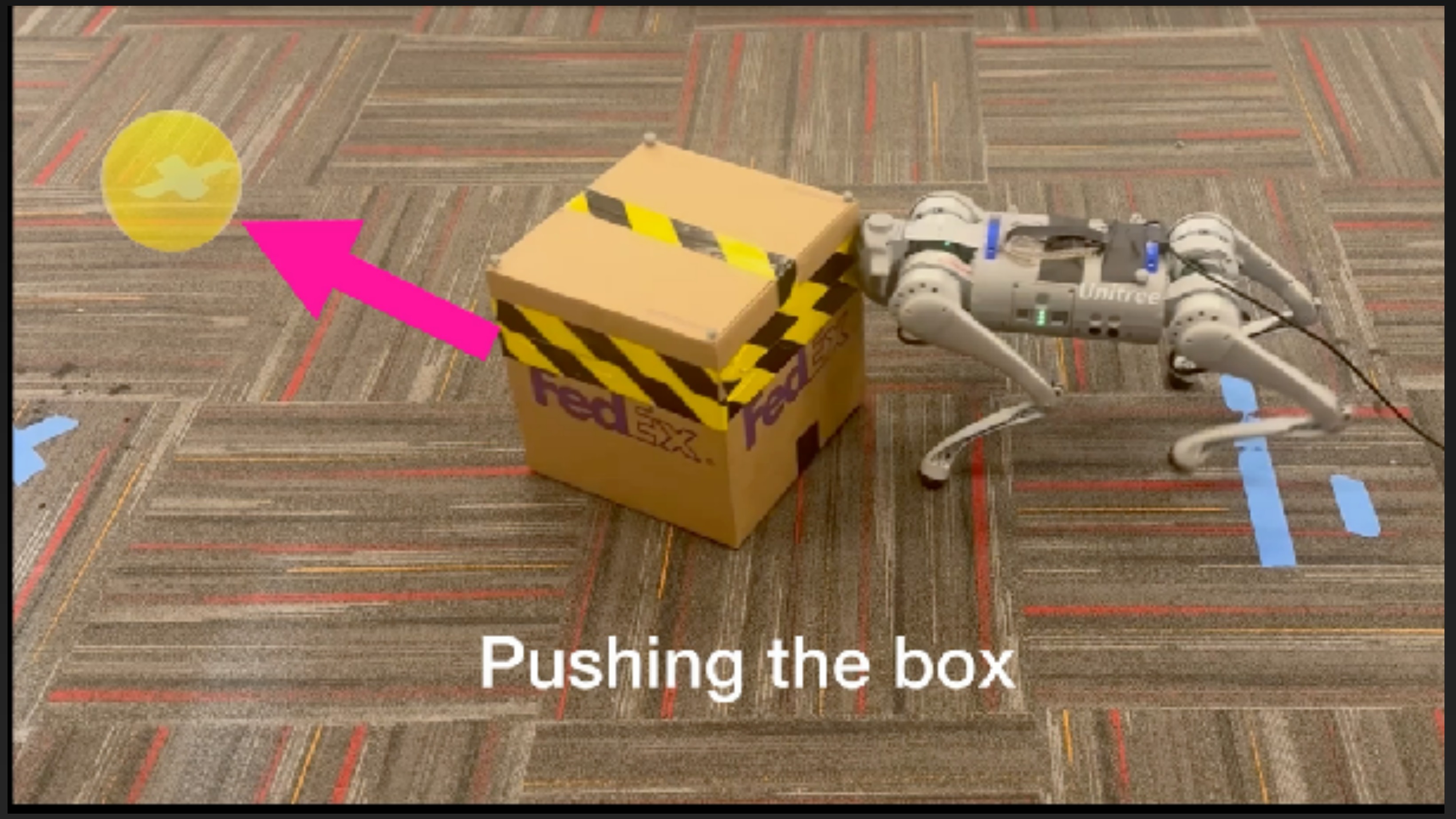
all solves in real time > 100 Hz

walking (and manipulating) with MPPI

- simulators today give you (almost) everything you can model
- real-time reasoning over full-order models, nonlinear contact, and whole-body collision. without offline training





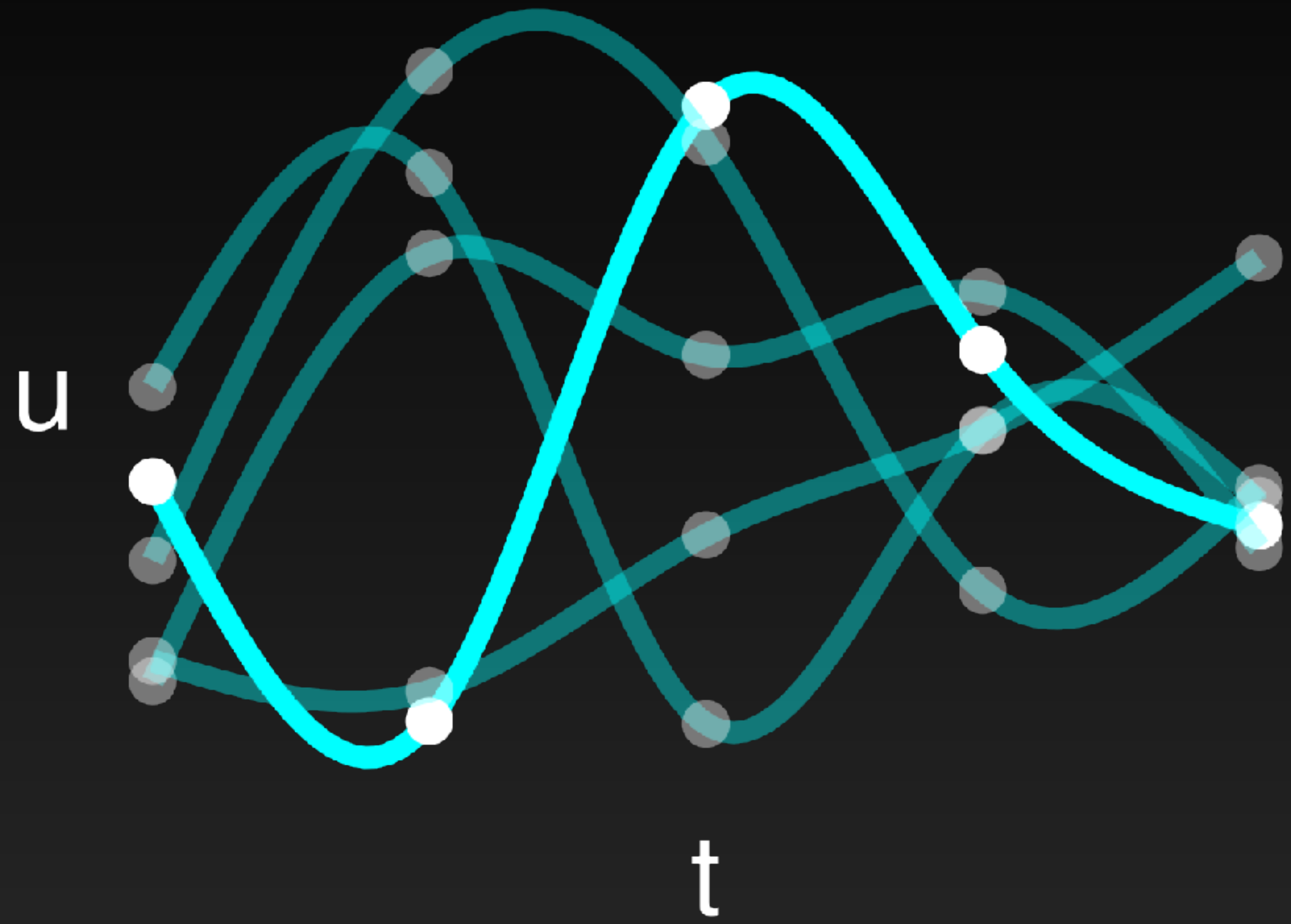


all solves in real time > 100 Hz   with 30 samples!

GPU simulation

CPU simulation

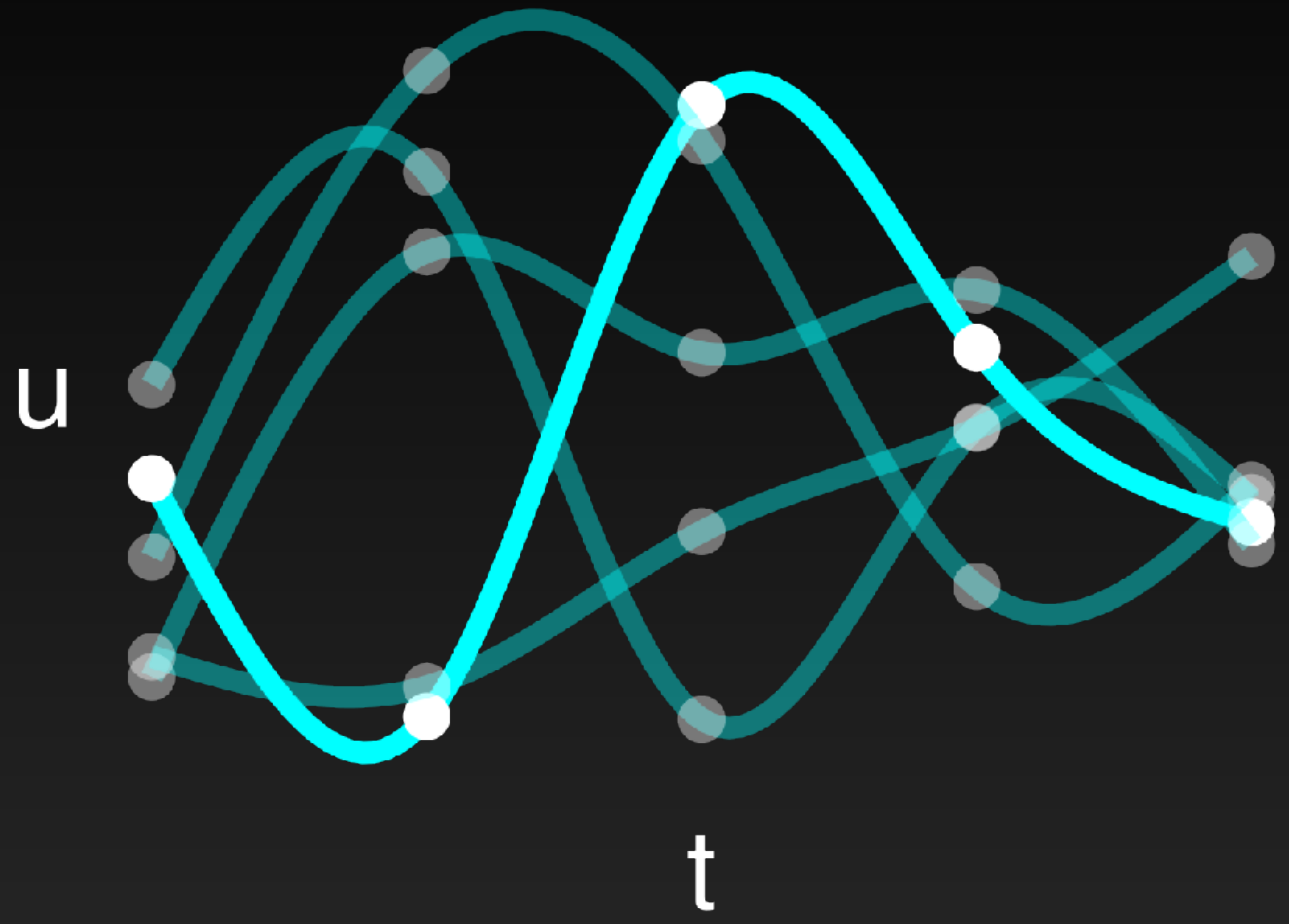
sampling over spline control points



reduce the search space by 10x

e.g.
direct sampling over controls:
50 time steps
vs.
sampling over splines:
5 control points

sampling over spline control points

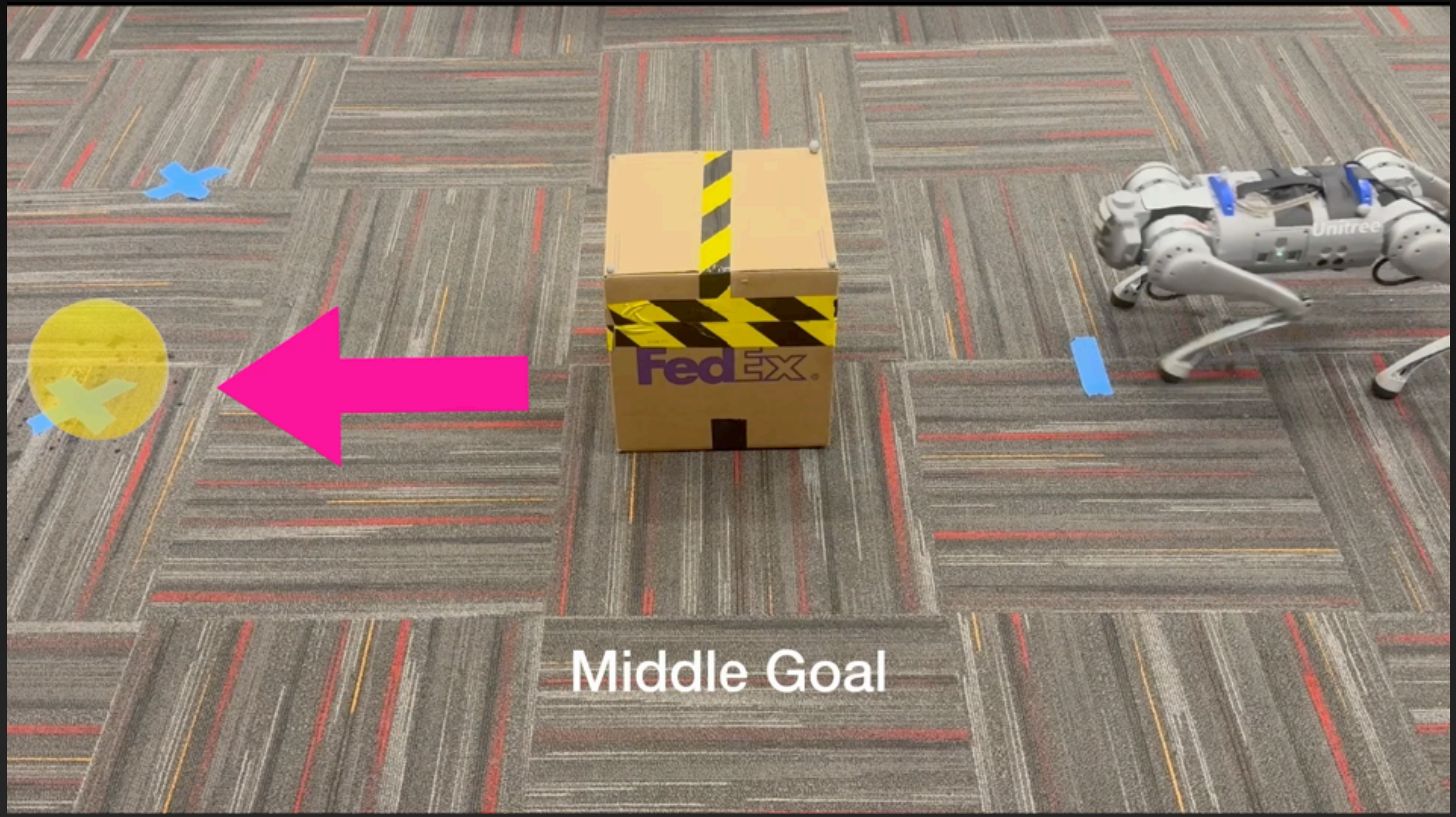


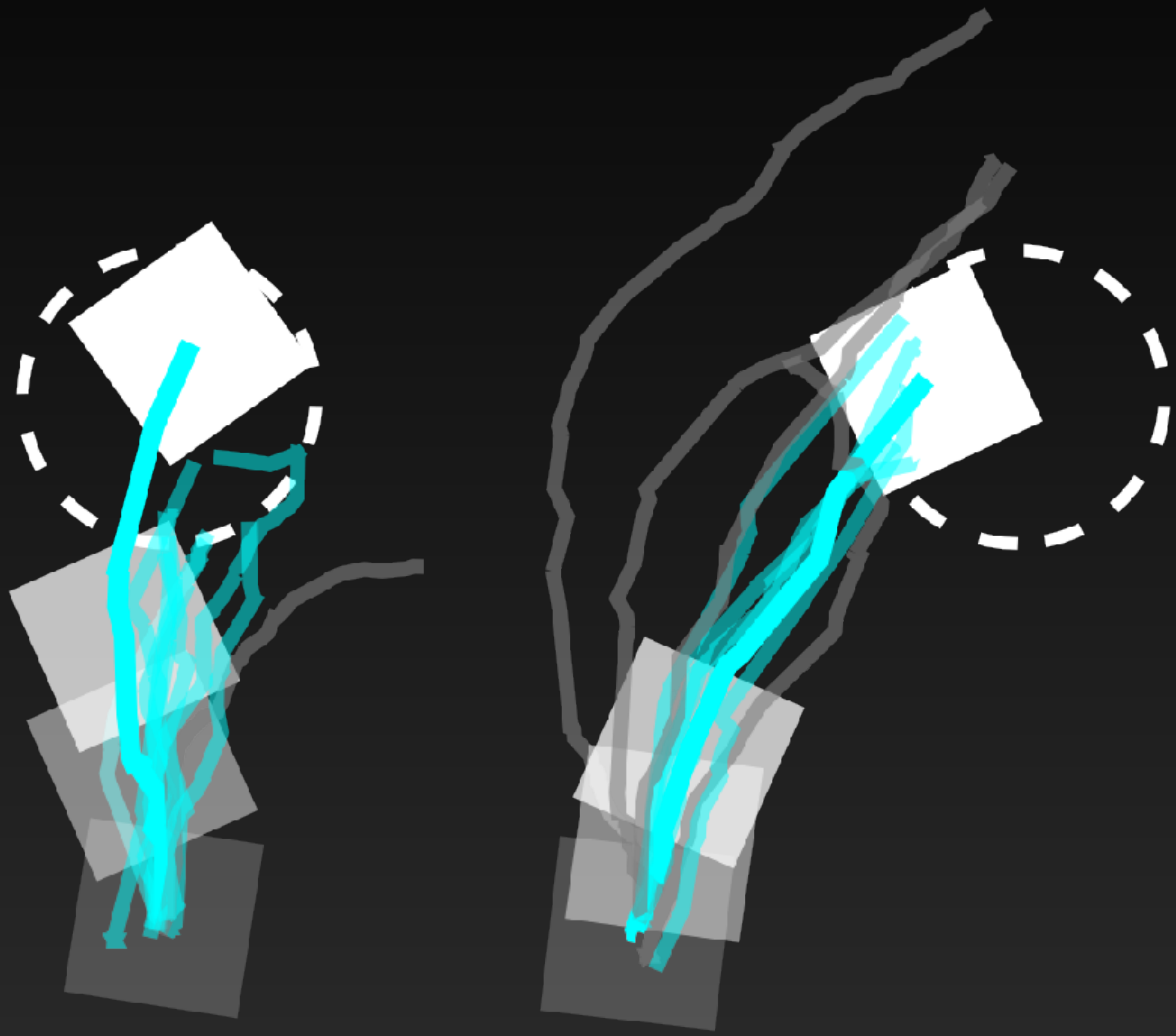
reduce the search space by 10x

actions are guaranteed to be smooth

MPPI — limitations

- many cases simulated physics diverge from real life

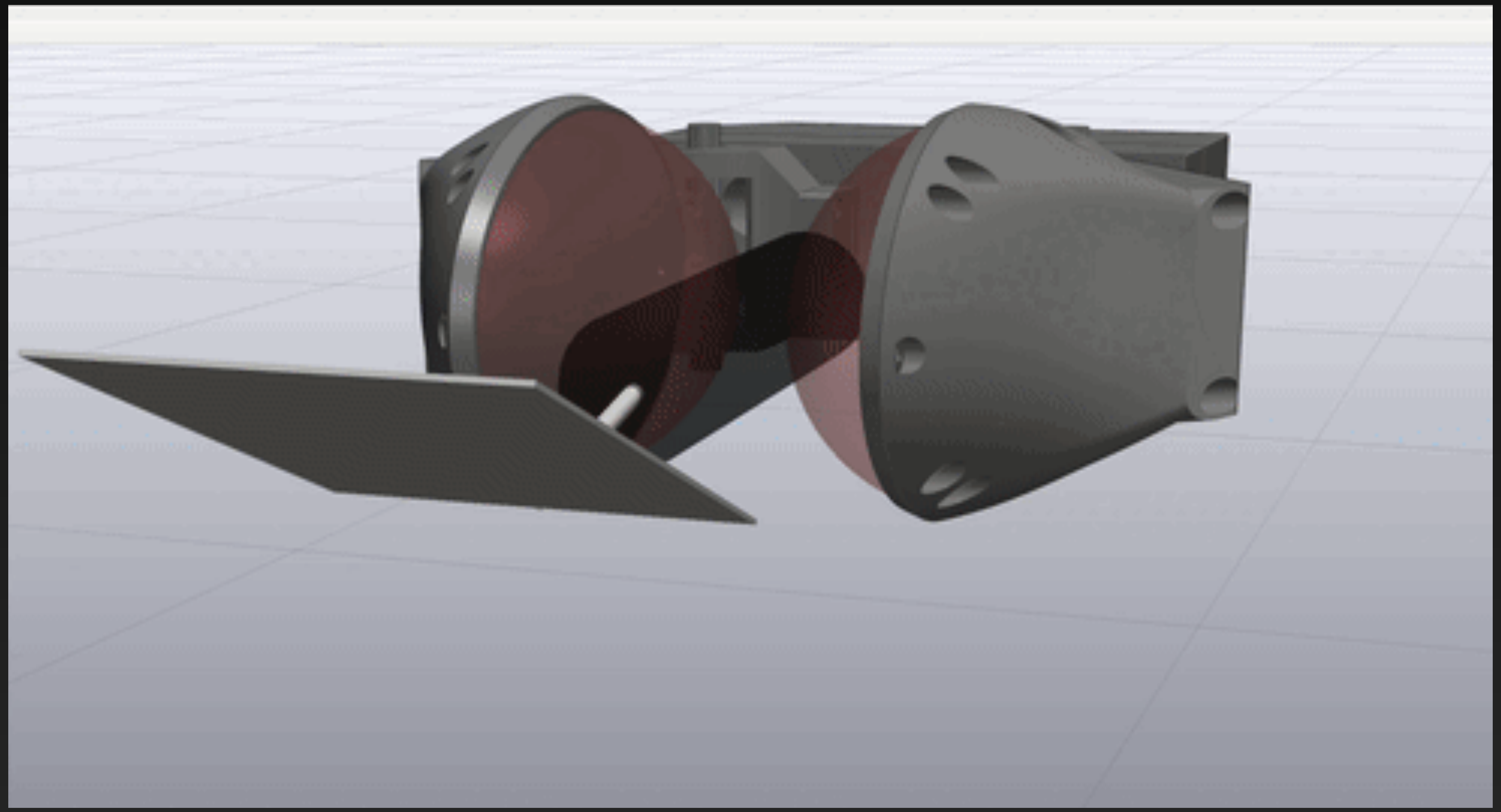




MPPI — limitations

- many cases simulated physics diverge from real life
- only control what you can simulate (fast and in parallel)

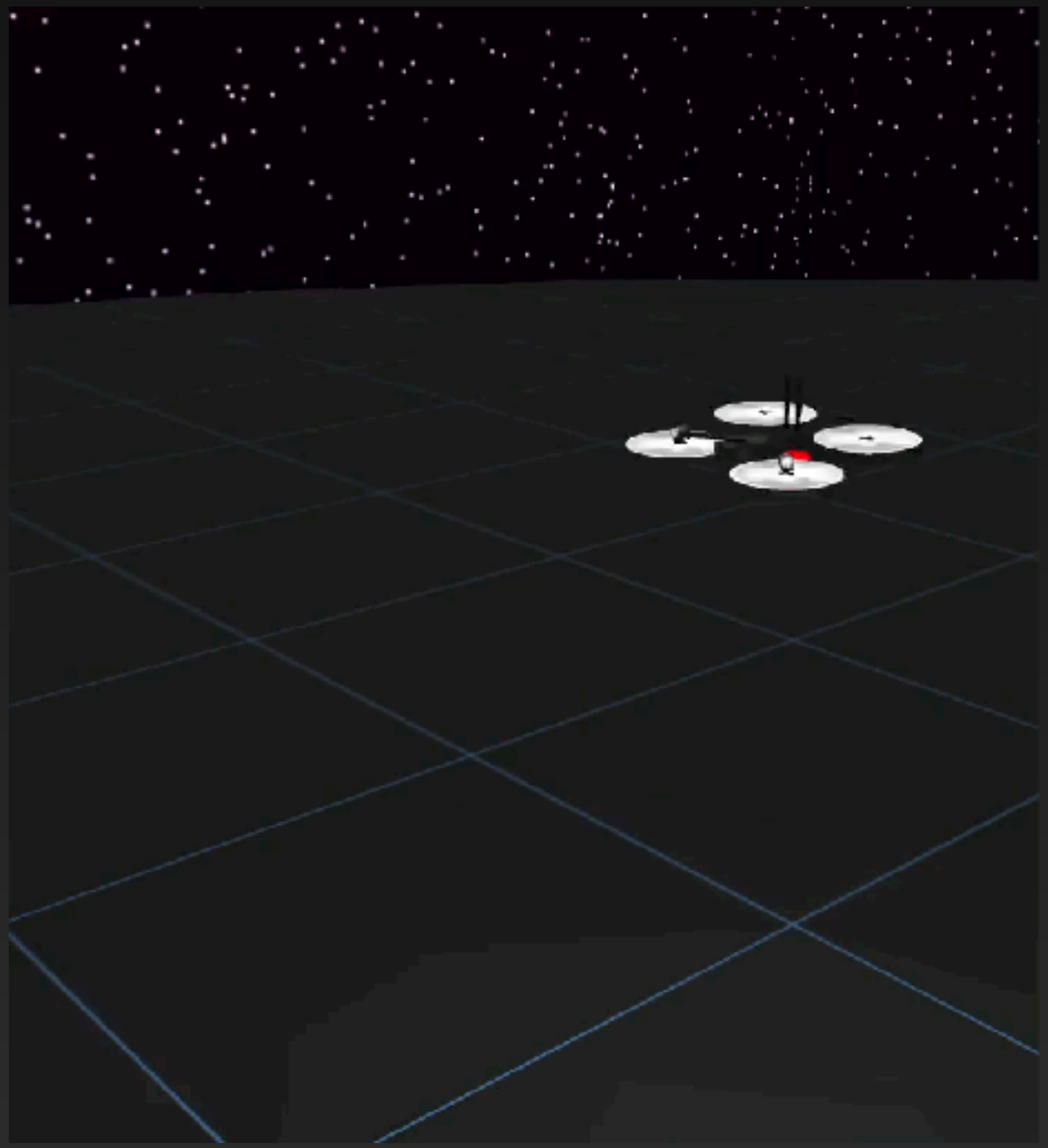
Aquarium, Lee et al.



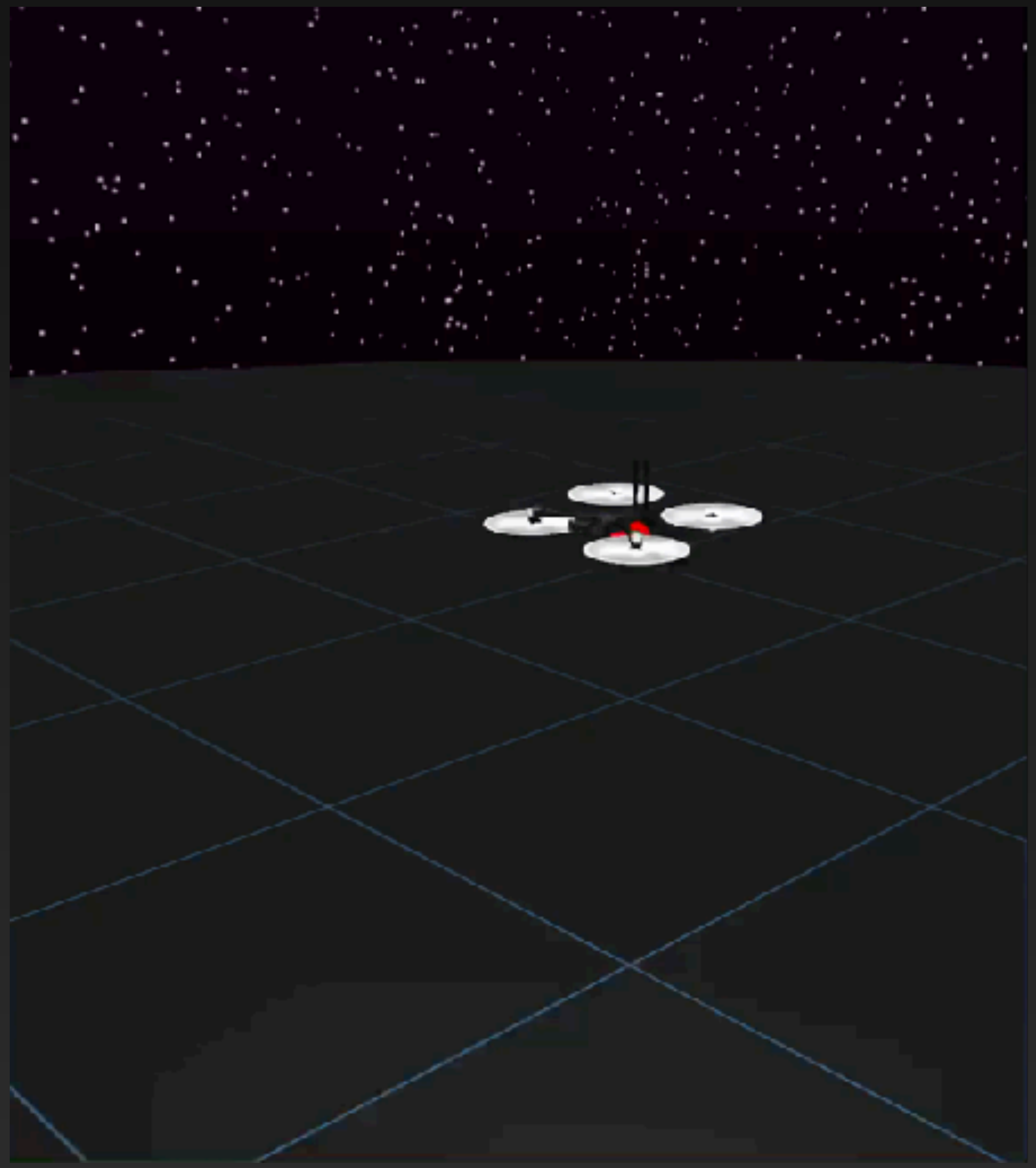
Drake, TRI.

MPPI — limitations

- many cases simulated physics diverge from real life
- only control what you can simulate (fast and in parallel)
- unstable systems



MPPI



LQR

parting thoughts

- legged locomotion is a largely solved problem
- the next frontier is locomotion + manipulation (so called “loco-manipulation”)

parting thoughts

- legged locomotion is a largely solved problem
- there are many open challenges in locomotion + manipulation for legged robots
- don’t worry about MPC vs RL, think about online vs offline computation
- robotics algorithms are deeply connected to the robot hardware and compute
- sim-to-real (or real-to-sim?) is still an open problem