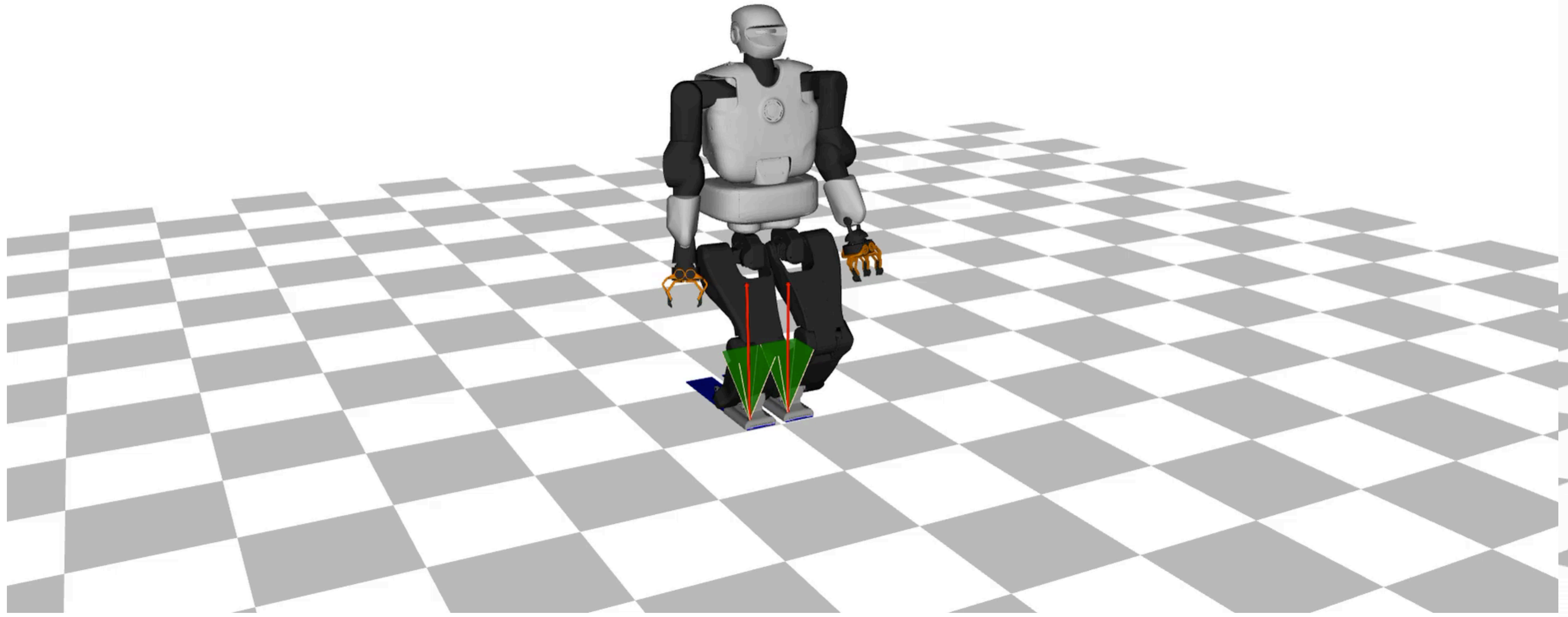
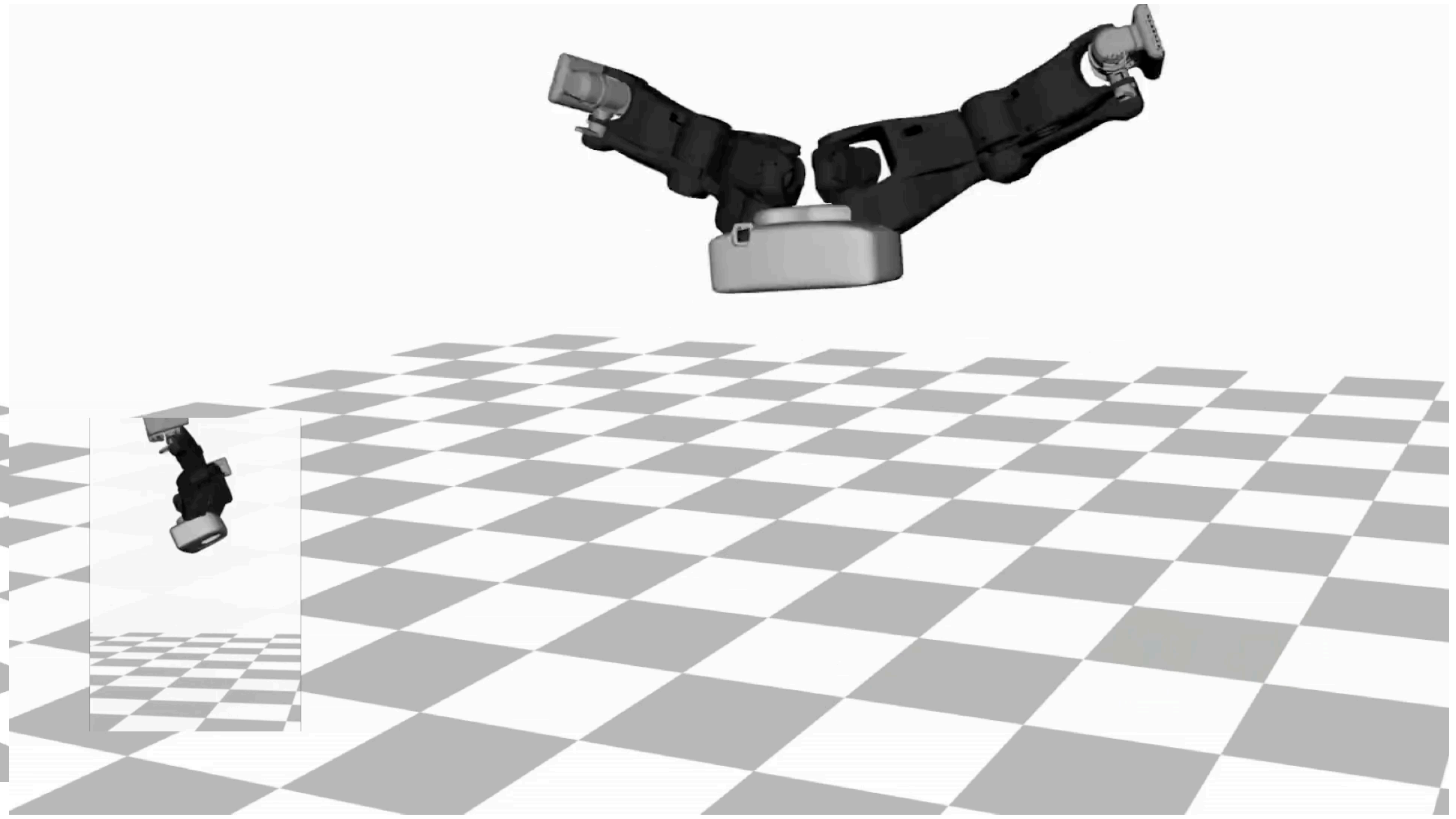
Crocoddyl:
Fast computations, Efficient Solvers, Receding Horizon, and Learning





Rohan Budhiraja, Amit Parag, Ewen Dantec, Justin Carpentier, Carlos Mastalli, Nicolas Mansard

LAAS CNRS

Crocoddyl: Contact RObot COntrol by Differential DYnamic Programming Library

Open-Source (BSD License) tool for Optimal Control, based on Differential Dynamic Programming based algorithms, and tailored for Legged Locomotion

https://hal.archives-ouvertes.fr/hal-02294059

GitHub

Numerical Optimal Control

Discretized, finite dimensional, non-linear problem

$$\min_{x^i, u^i \ldots} \sum_{i=0}^{N-1} l_s^i(x^i, u^i)$$

such that

$$x^{i+1} = \mathcal{I}_s(x^i, u^i, \delta t)$$ Dynamics Constraint

$$c_i(x^i, u^i) = 0$$ Equality constraints

$$h_i(x^i, u^i) \leq 0$$ Inequality constraints

Numerical Optimal Control

Discretized, finite dimensional, non-linear problem

$$\min_{x^i, u^i \dots} \sum_{i=0}^{N-1} l_s^i(x^i, u^i)$$

such that $\quad x^{i+1} = \mathcal{I}_s(x^i, u^i, \delta t) \quad$ Dynamics Constraint

$x^0 = \hat{x}^0 \quad$ Initial Condition

Legged Locomotion Problem

Main Challenges

- **Handling contact constraints**
  - to ensure feet placements are exactly satisfied.
- **Handling sparsity in the problem**
  - for faster resolution

LAAS CNRS

Legged Locomotion Problem

Main Challenges

- **Handling contact constraints**
  - to ensure feet placements are exactly satisfied.

- **Handling sparsity in the problem**
  - for faster resolution

Legged Locomotion Problem

Contact Constrained Dynamics

$$\begin{bmatrix} M & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ -\begin{bmatrix} f_k & \tau_k \end{bmatrix}^T \end{bmatrix} = \begin{bmatrix} S^T\tau - C\dot{\boldsymbol{q}} - g \\ -\dot{J}_k\dot{\boldsymbol{q}} \end{bmatrix}$$

… and the derivatives of the contact constrained dynamics

(Ongoing work discussed by Justin Carpentier)

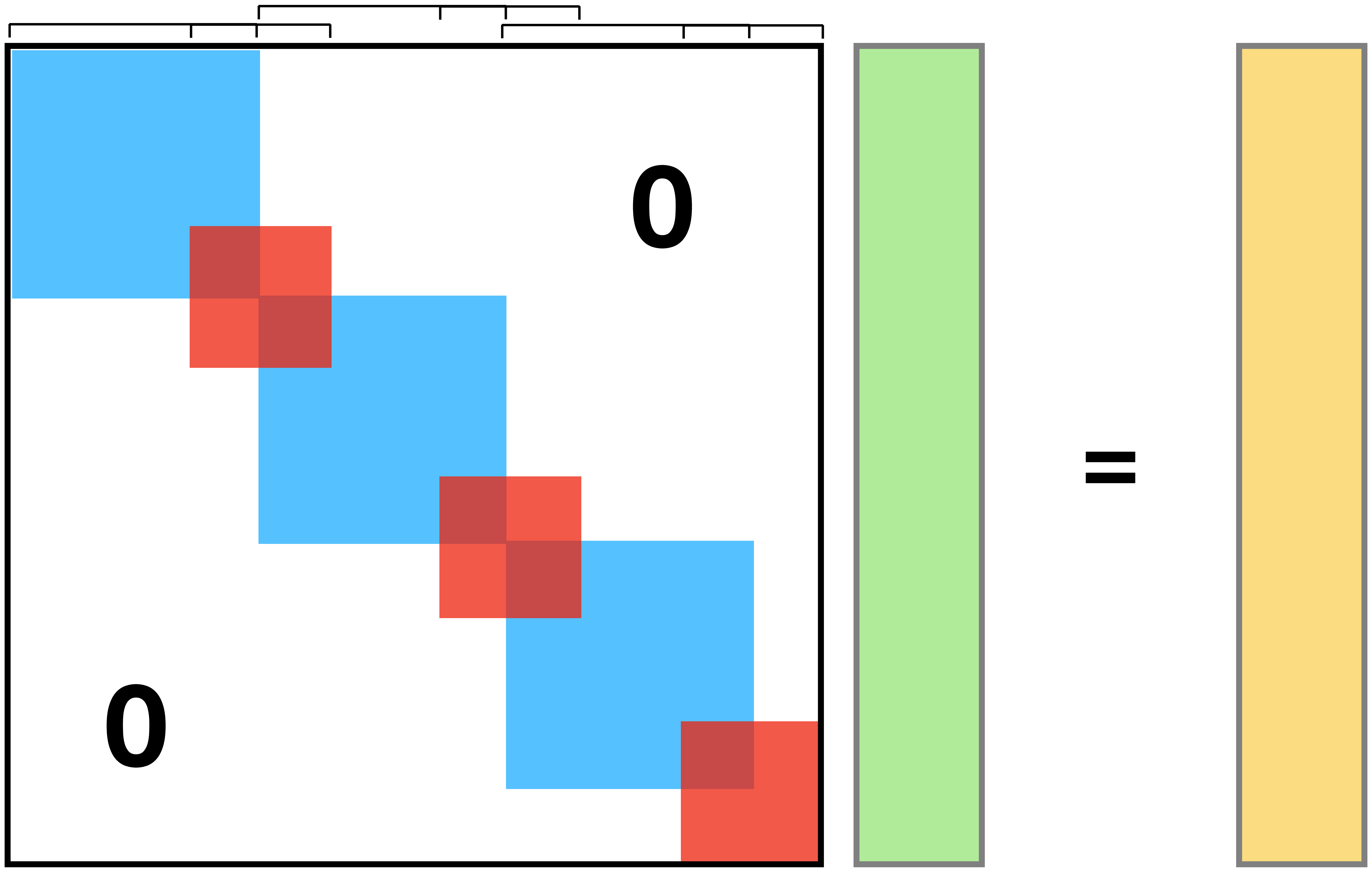
Schultz et al., 2010
Budhiraja et al., 2018

LAAS CNRS

Whole-body Problem Structure

Sparsity of the Optimization Problem

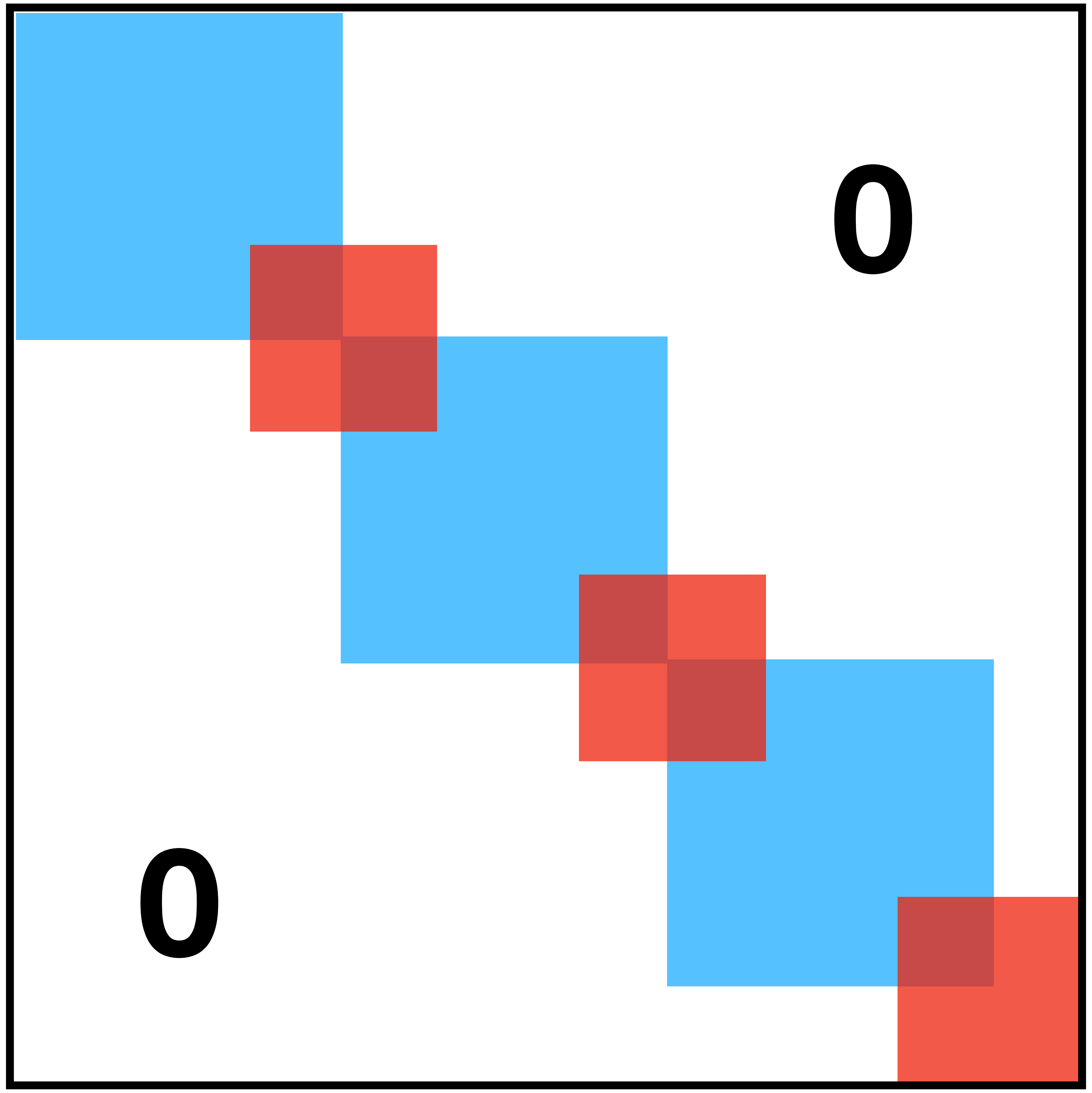
To find the solution,
we need to invert a sparse
matrix like this



Differential Dynamic Programming 101

An efficient way to handle this sparsity

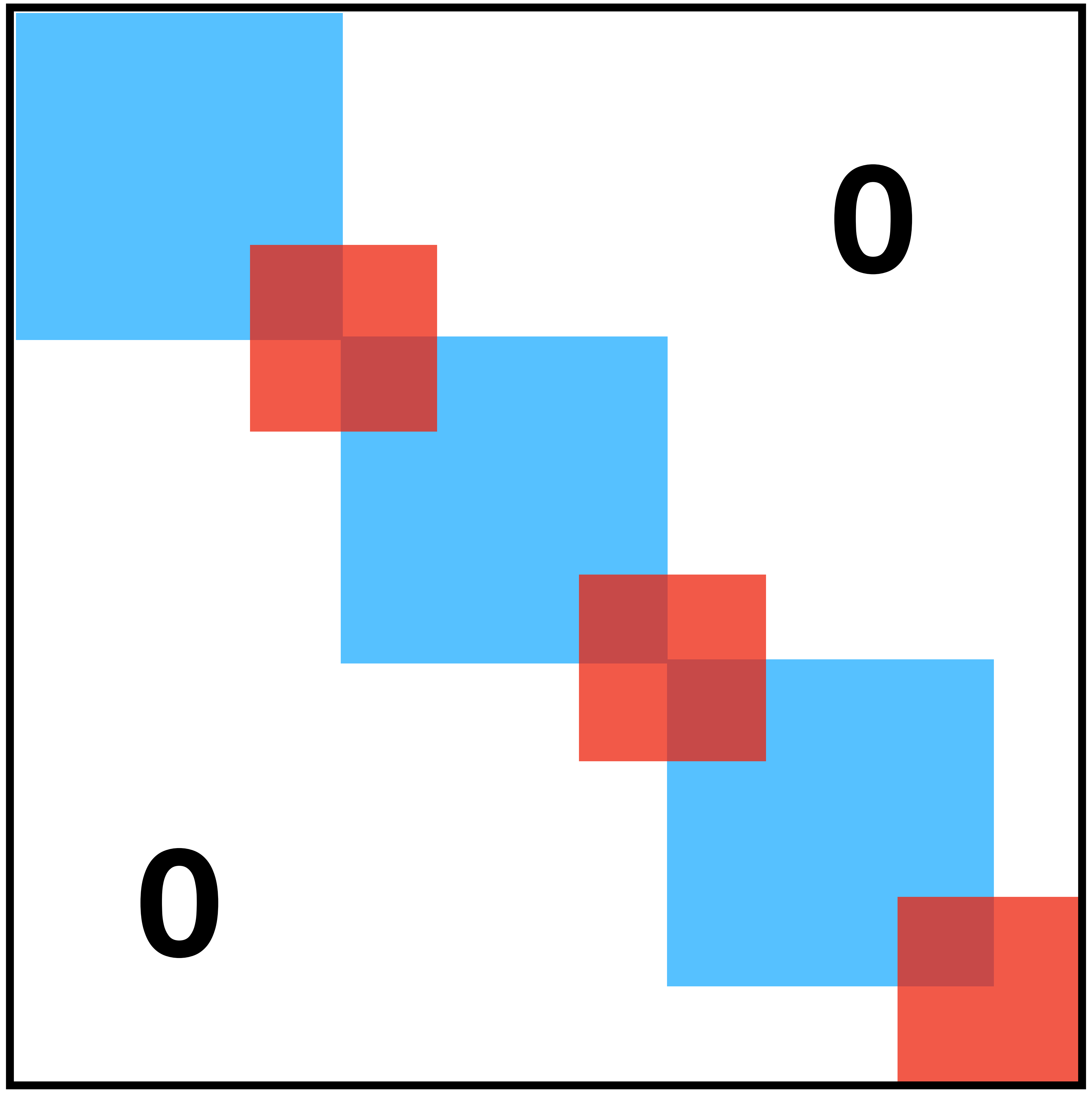
Iteratively invert
one block at a time



Differential Dynamic Programming 101

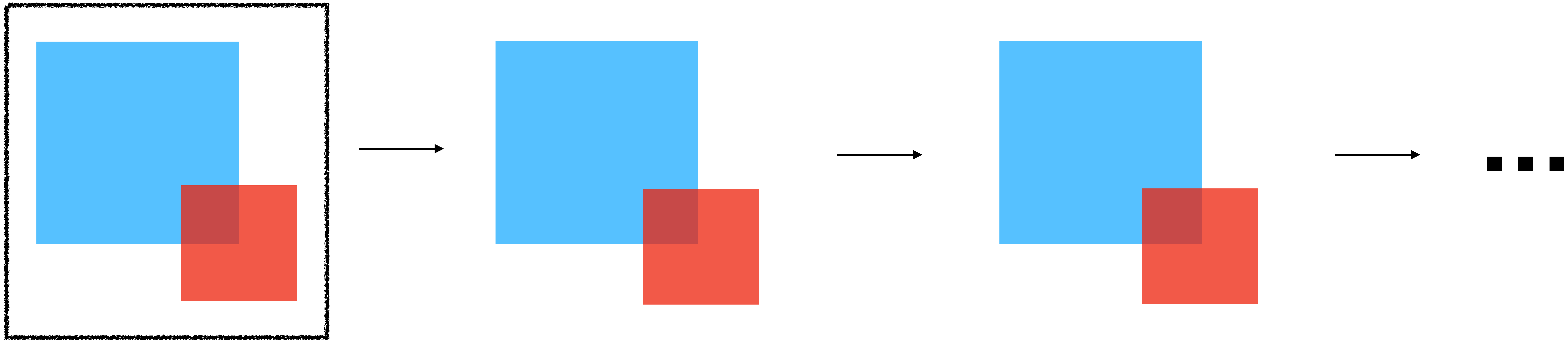
An efficient way to handle this sparsity

Value Function
Feedback and Feedforward Gains



Crocoddyl: Simple API for DDP with Contact

The Shooting Problem



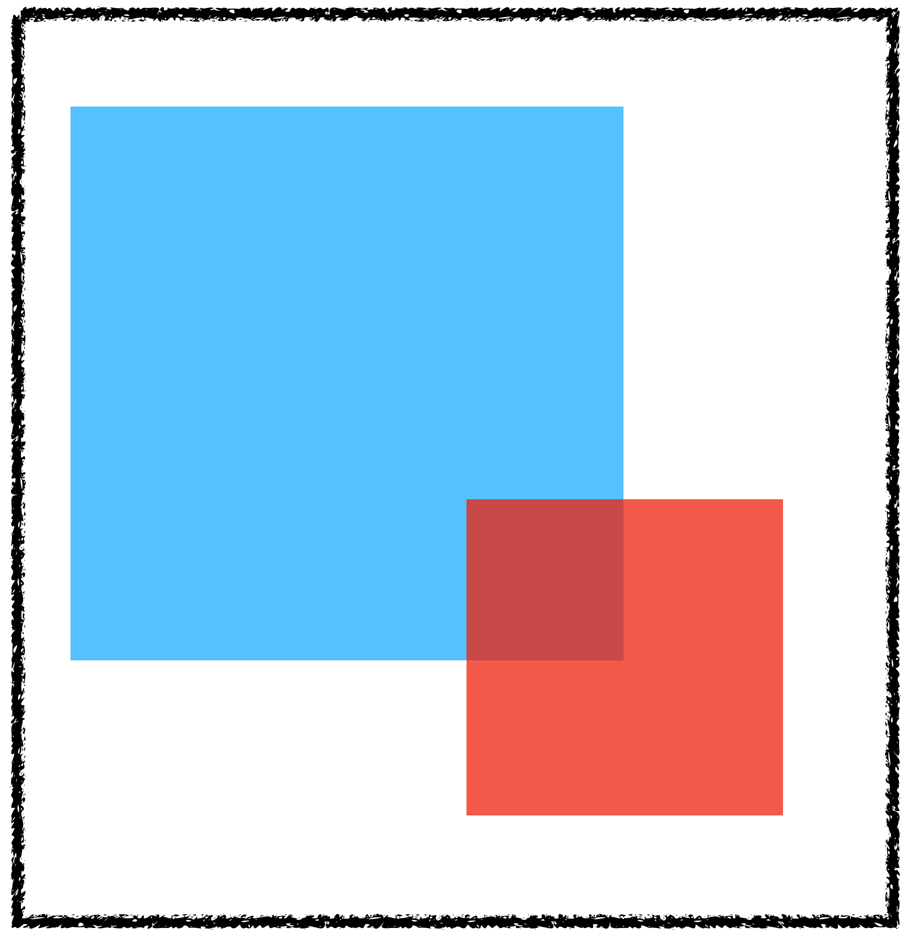
Action Model

$$l^k(x^k, u^k)$$

$$x^{k+1} = \mathcal{I}_s(x^k, u^k)$$

Crocoddyl: Simple API for DDP with Contact

The Shooting Problem



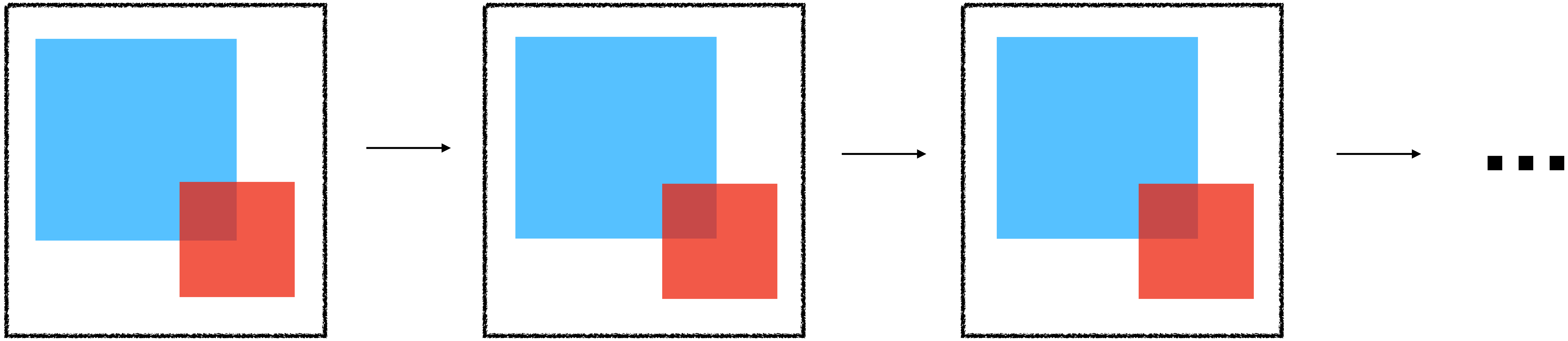
calc $\longleftarrow$ Action Model $\longrightarrow$ calcDiff

$$l^k(x^k, u^k)$$
$$x^{k+1} = \mathcal{I}_s(x^k, u^k)$$

$$\frac{\partial}{\partial x}, \frac{\partial}{\partial u}\begin{pmatrix} l^k(x^k, u^k) \\ x^{k+1} = \mathcal{I}_s(x^k, u^k) \end{pmatrix}$$

Crocoddyl: Fast Resolution of DDP

Code Generation Support

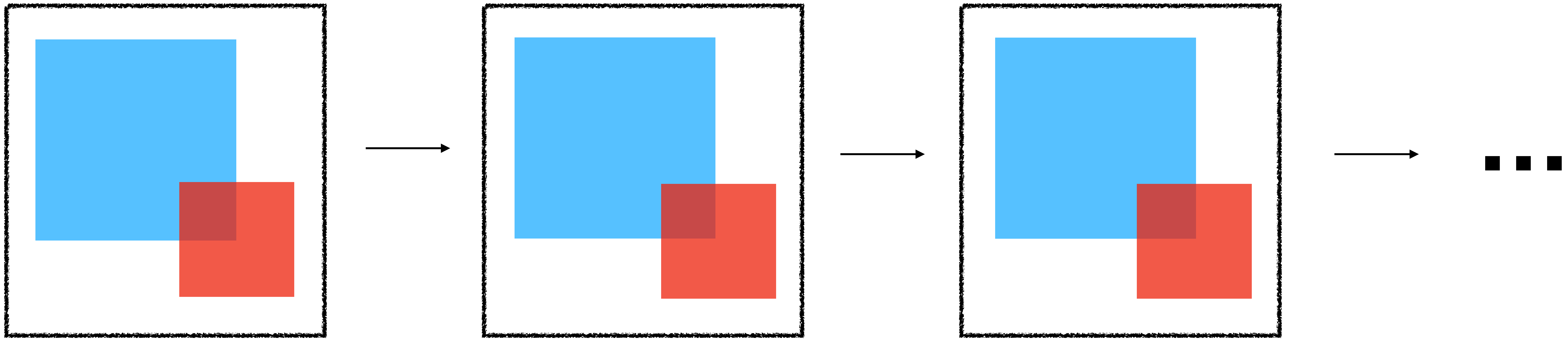


```
template <typename _Scalar>
class ActionModelAbstractTpl {
```

- Fully Templatized Action Models on Scalar type.
- Highly efficient C source-code generation for calc and calcDiff computations

Crocoddyl: Fast Resolution of DDP

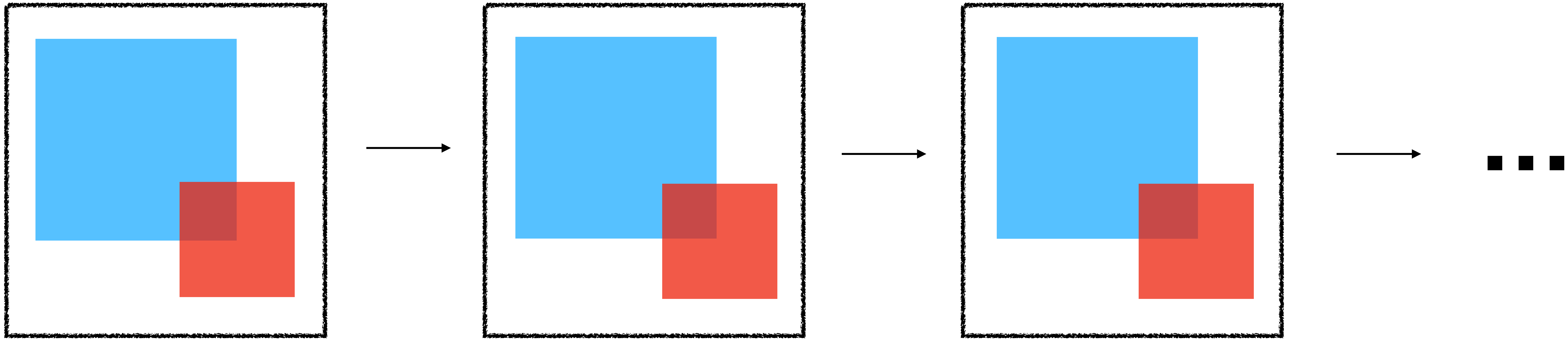
Multi-threading Support



- Parallel Computation of the Derivatives
- 4 threads mean almost 4 times faster computation of derivatives!

Crocoddyl: Fast Resolution of DDP

Box Control Constraints



$$u_l^i \leq u^i \leq u_u^i$$

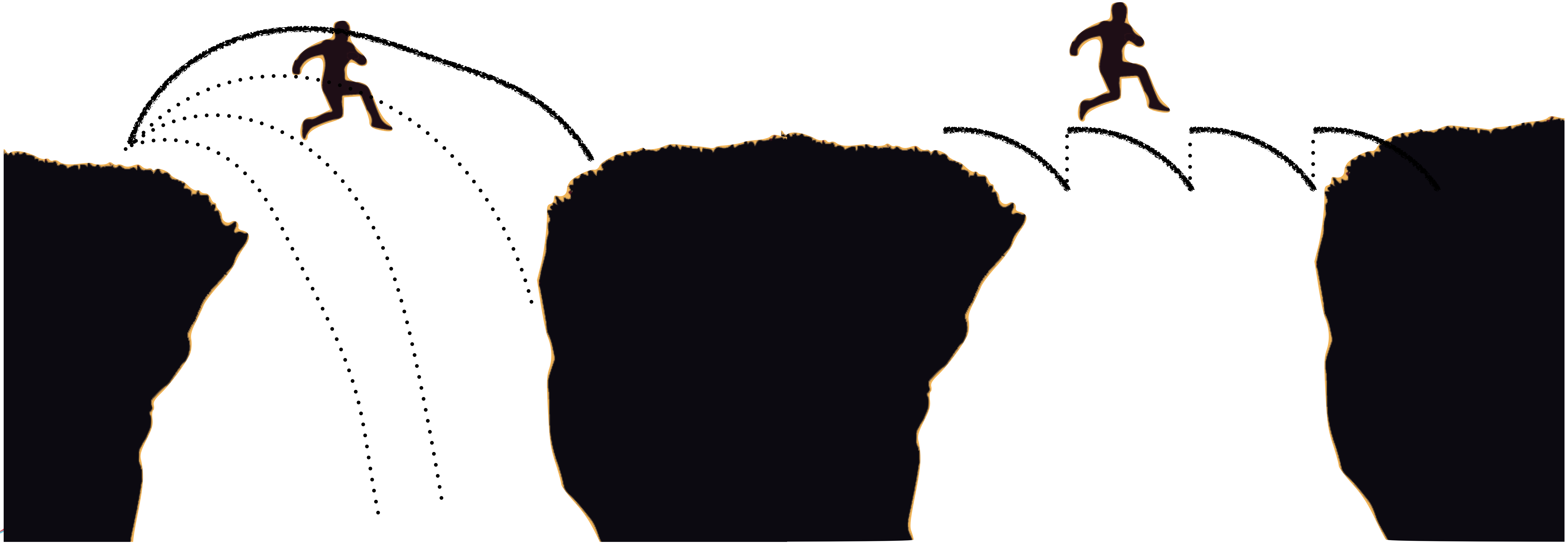
Control inside bounds

- Box-DDP:
  Variation of DDP that handles Box constraints on control variables.

Tassa et.al, 2014

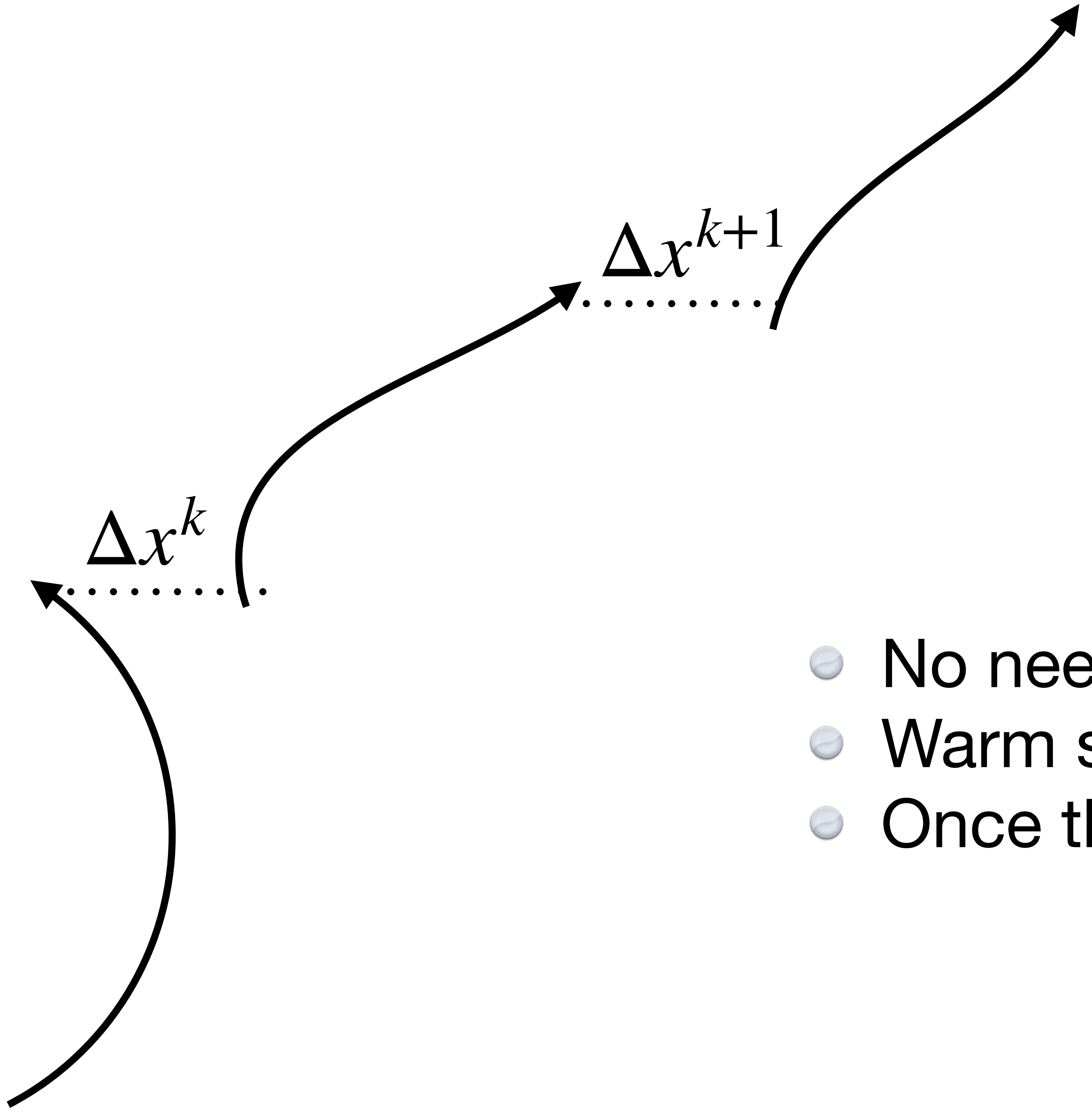
Crocoddyl: Feasibility-Driven DDP

Multiple-shooting with DDP



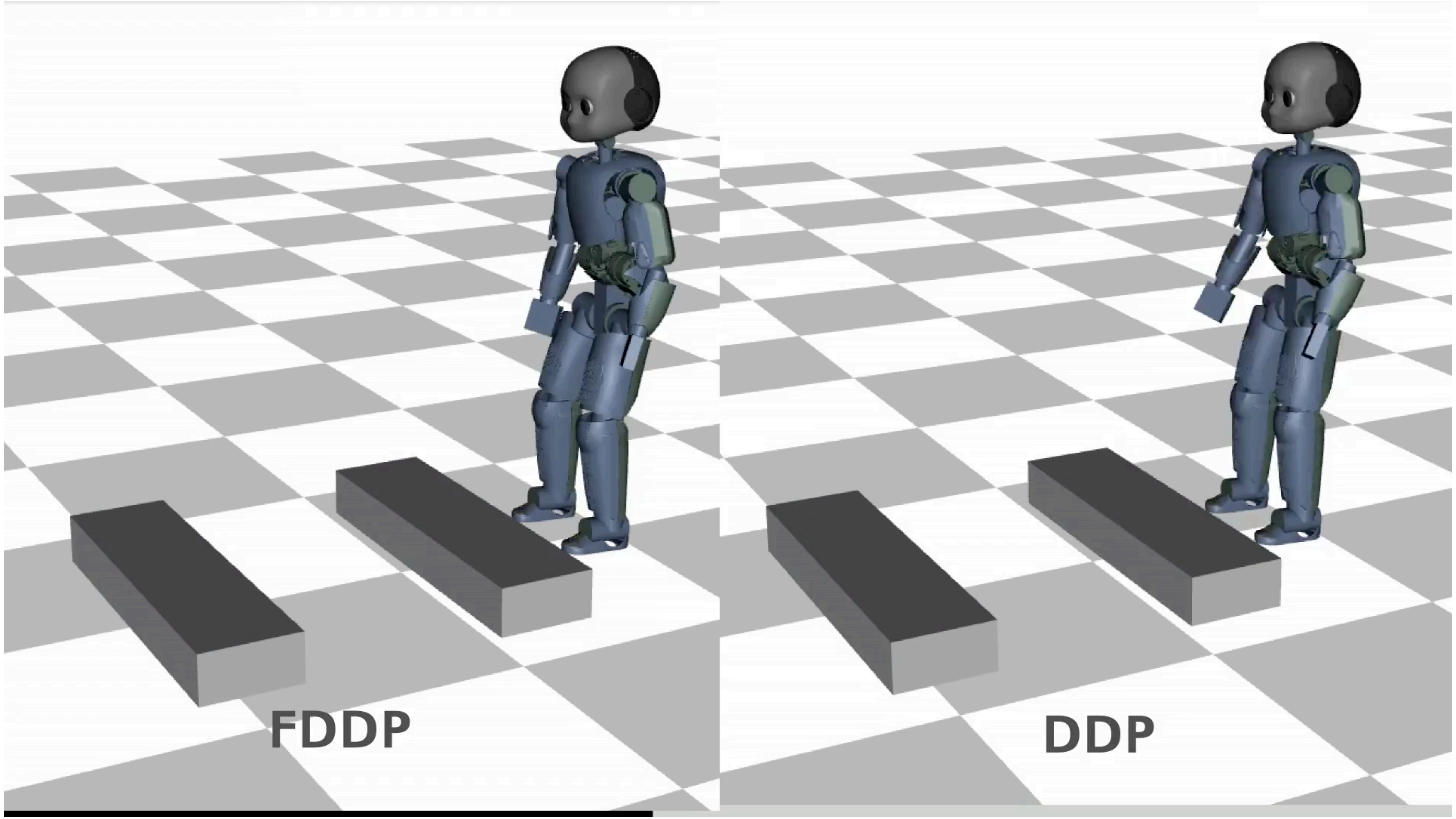
Crocoddyl: Feasibility-Driven DDP

Multiple-Shooting: Gaps vs Objectives?



- No need for merit function between gaps and objectives.
- Warm start with an infeasible trajectory and optimize.
- Once the solution becomes feasible, exactly the same as DDP

Multiple Shooting


FDDP
DDP

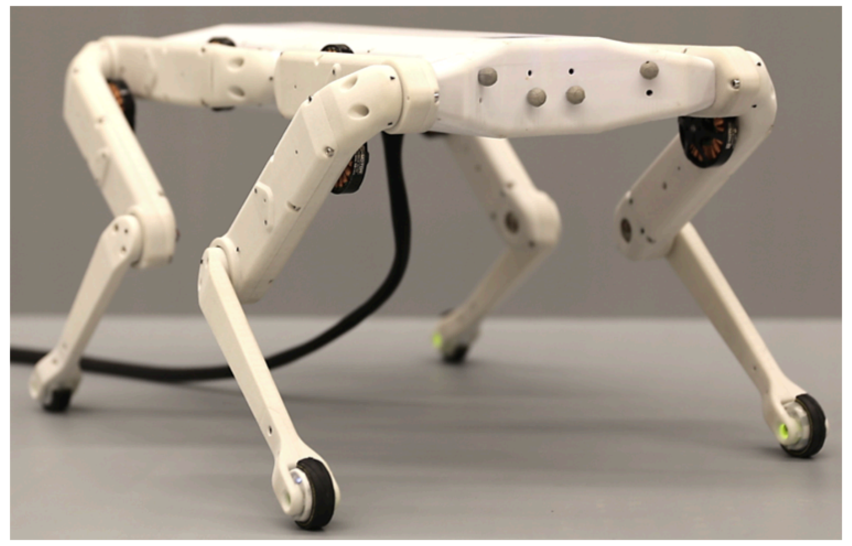
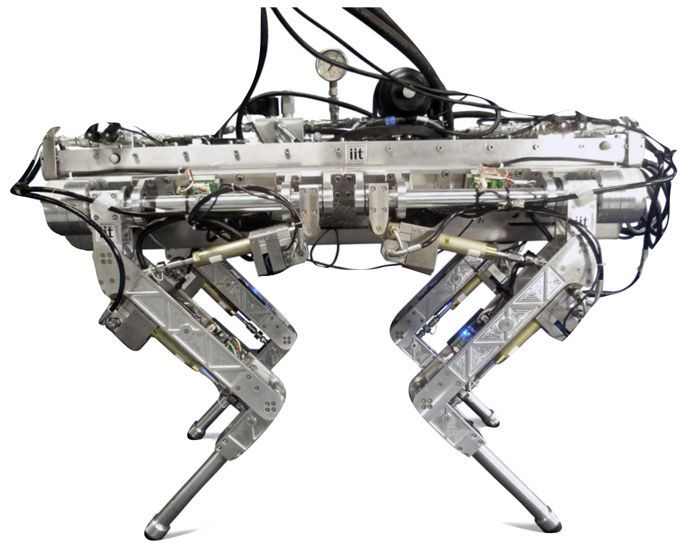
Benchmarks

for one iteration of the solver

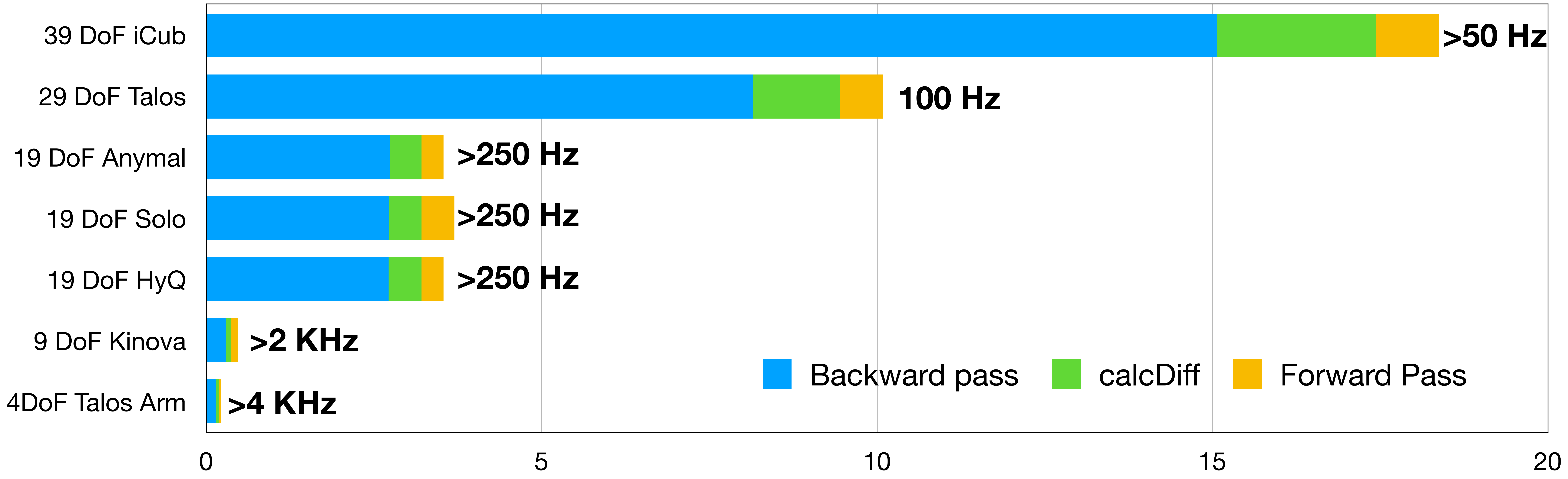
Horizon	**100 Nodes**
CPU Info	**Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz**
Code-Generated	**Yes**
Multithreading	**6 threads**





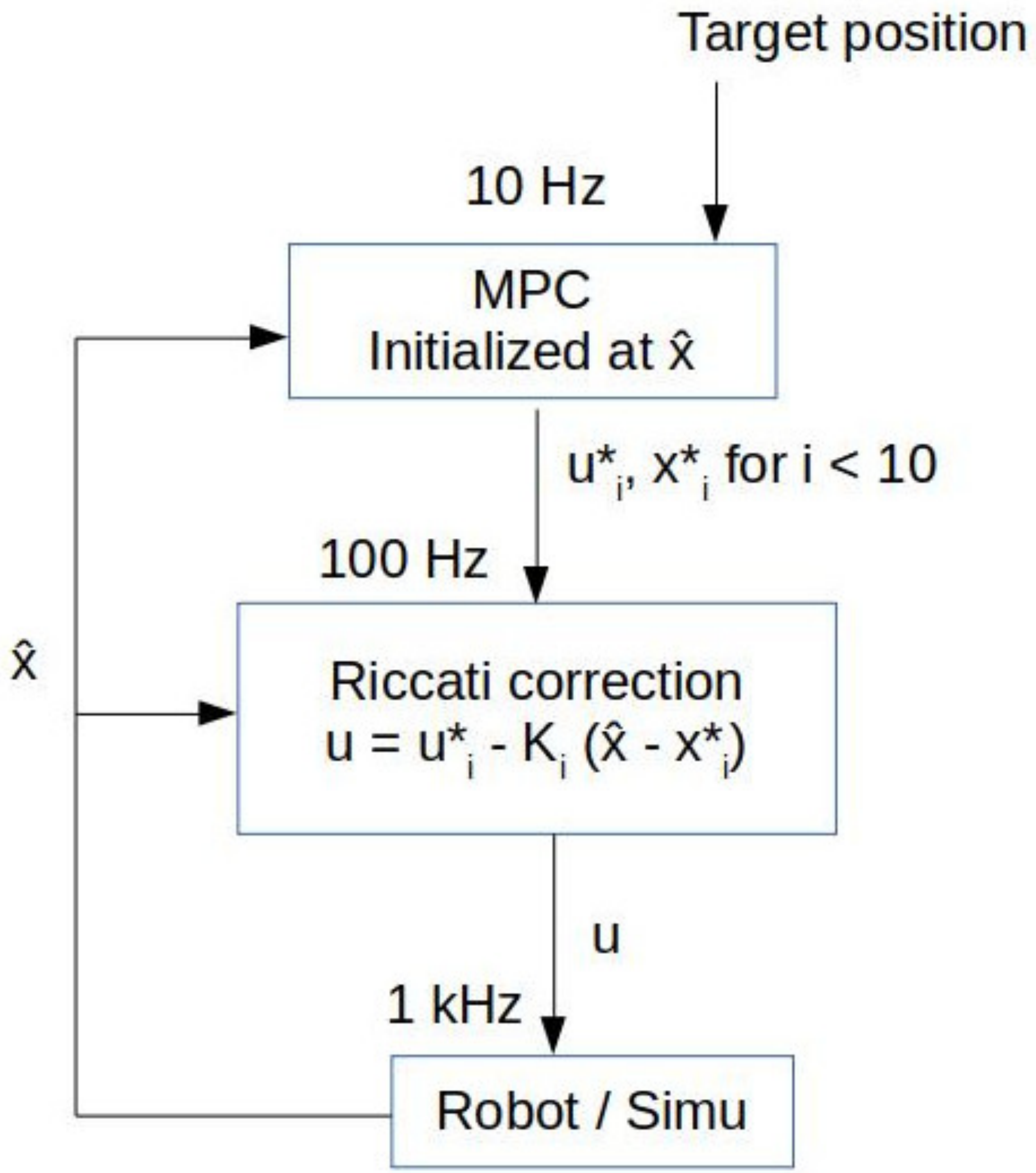
Benchmarks



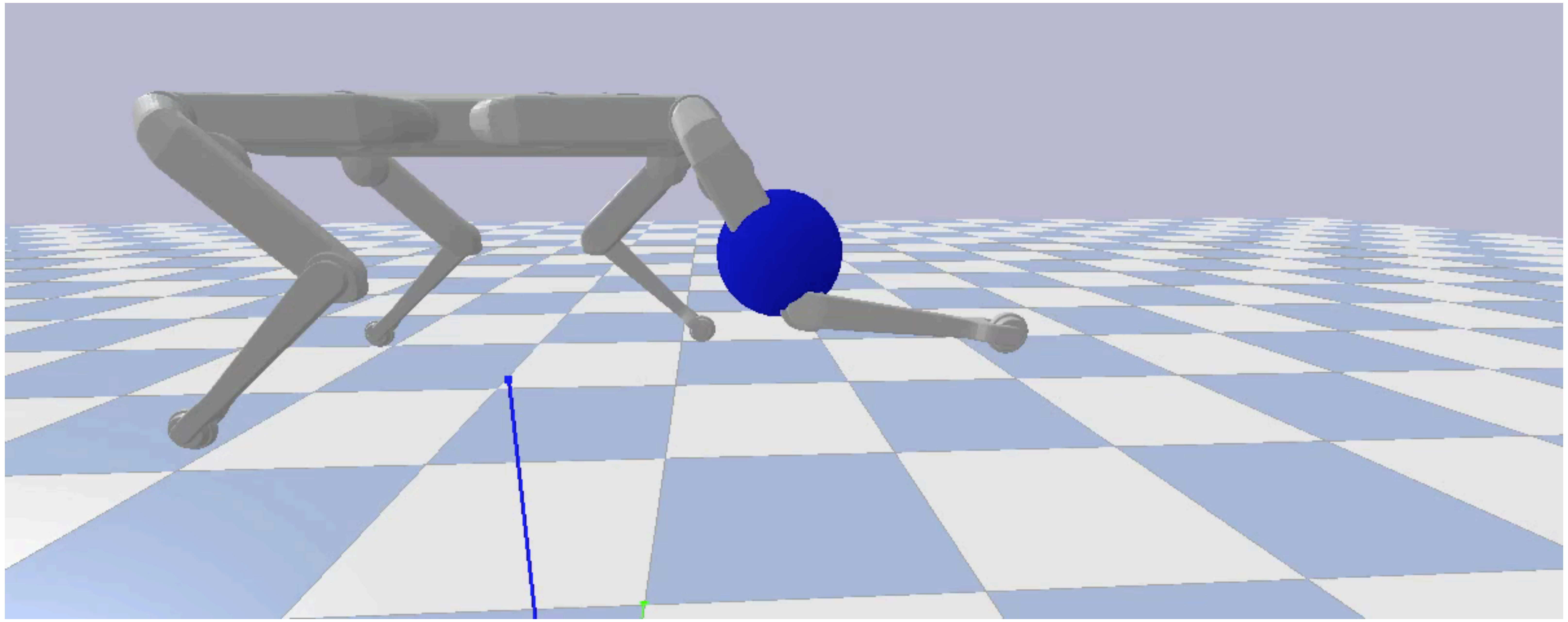
LAAS CNRS

Application: Model Predictive Control

Action Model

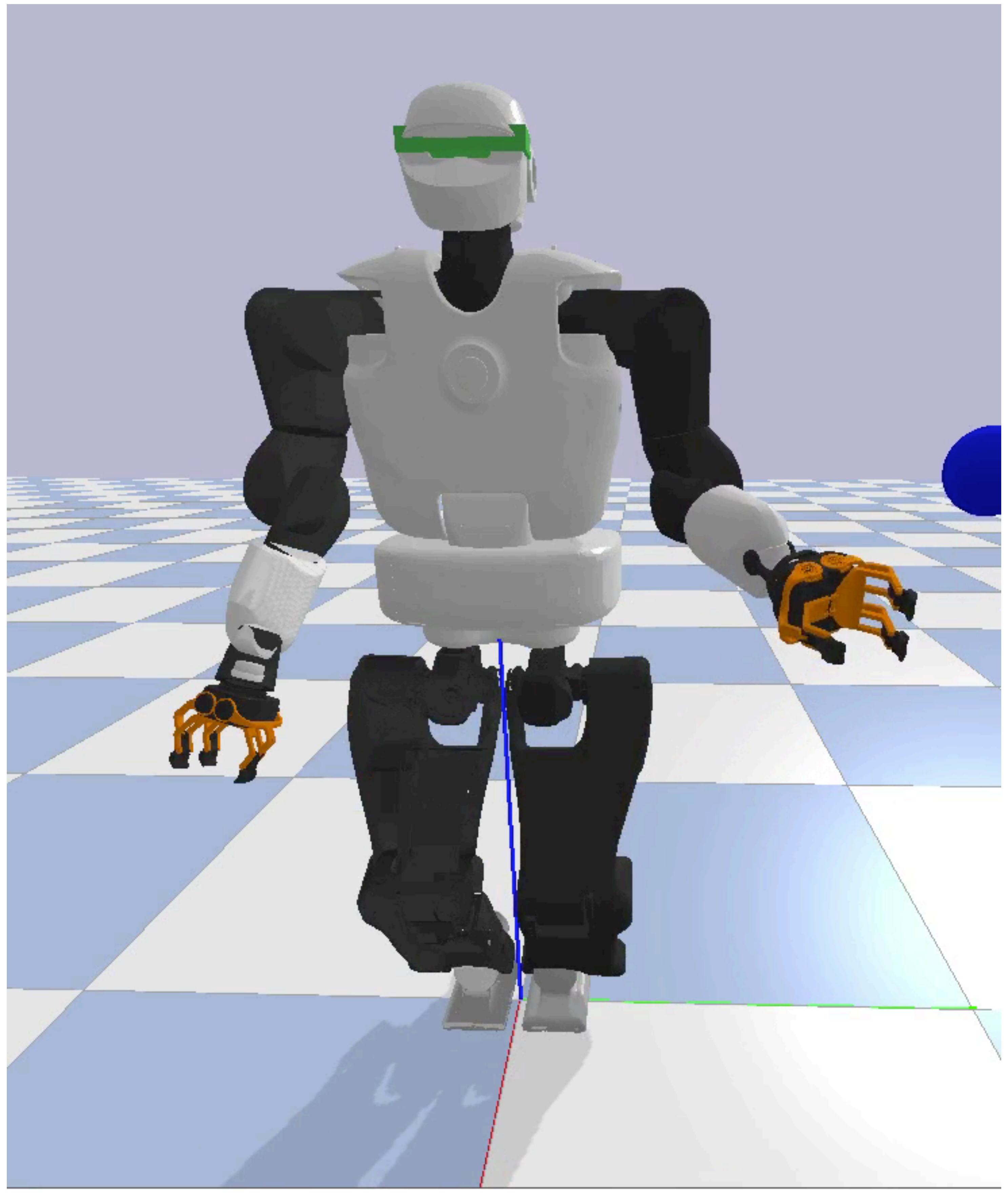


Application: Model Predictive Control

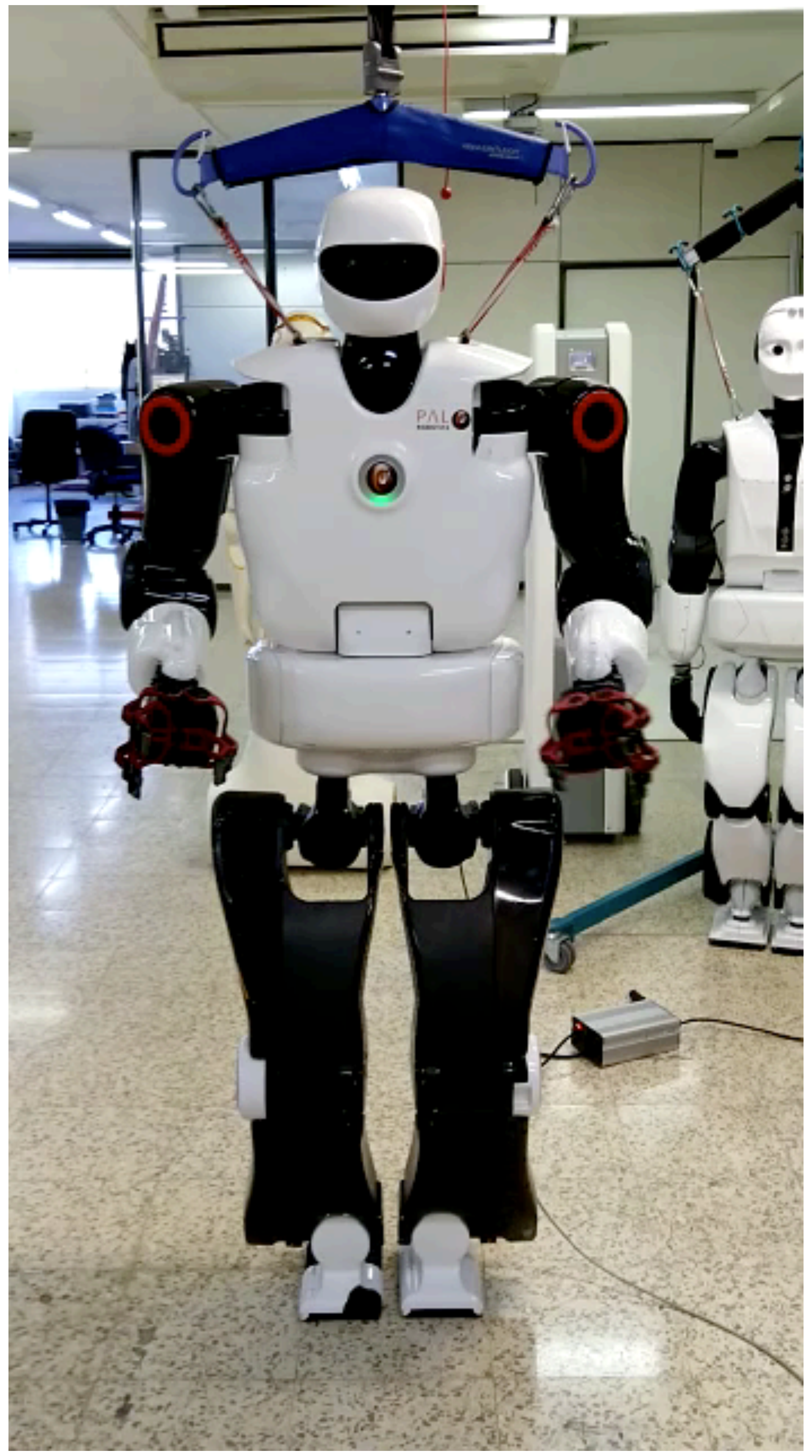


Tracking a Ball
by MPC

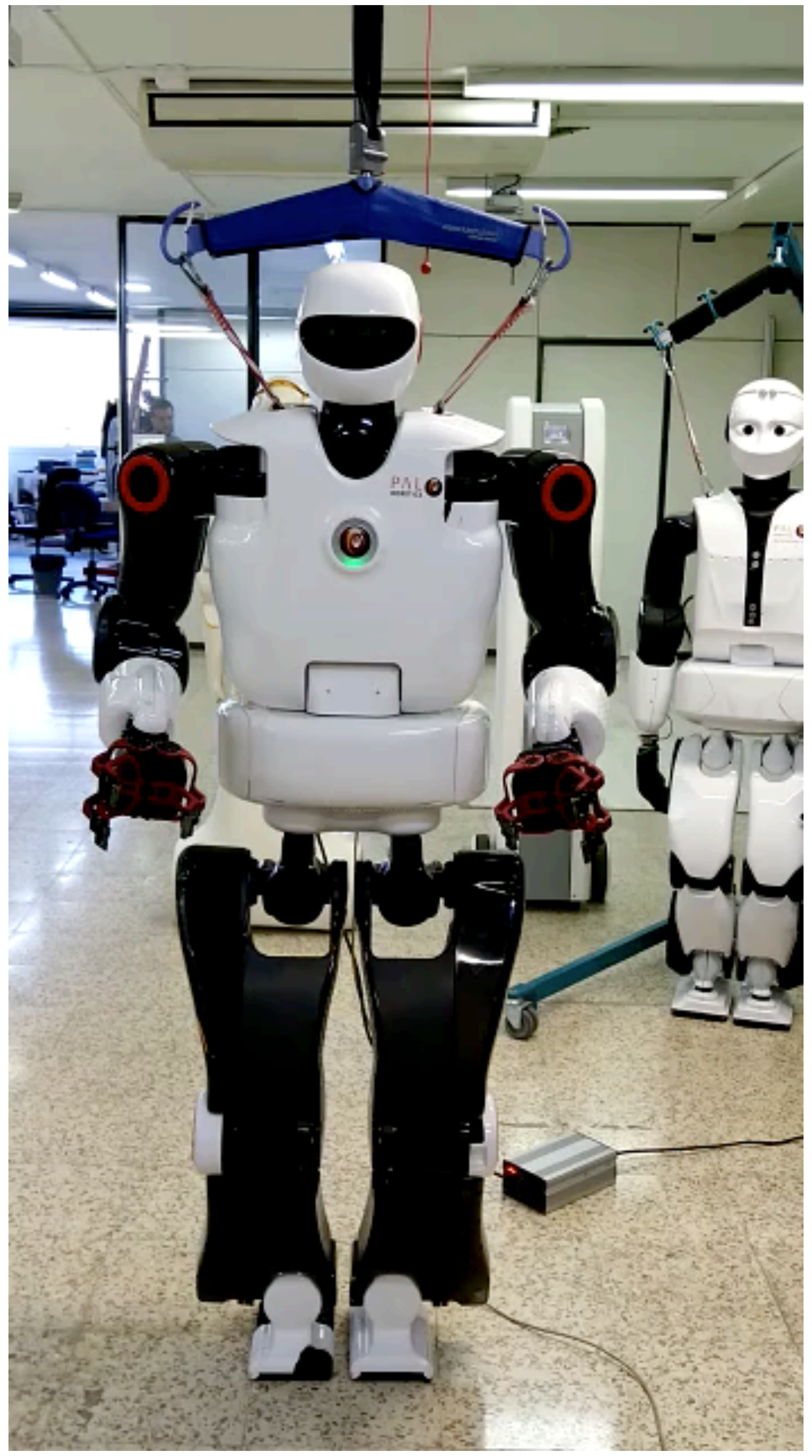
Application: Model Predictive Control



Tracking a Circle by MPC



Tracking a Circle by MPC

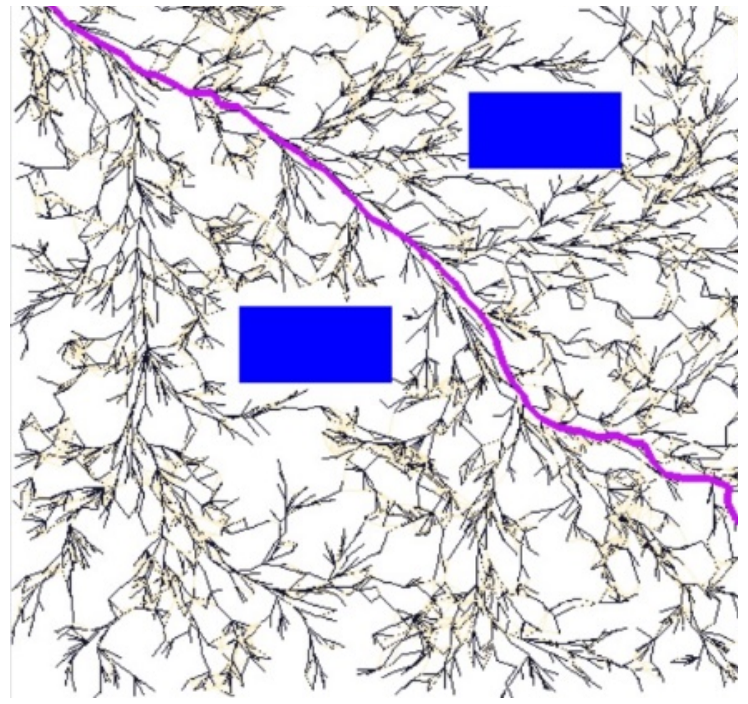


Tracking a Circle with external disturbances

PAL ROBOTICS

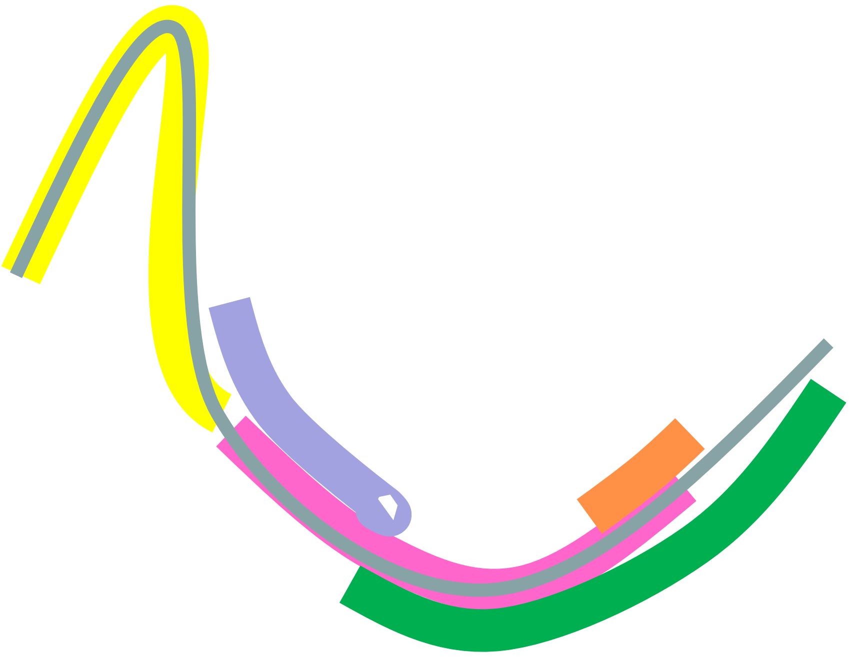
LAAS CNRS

Iterative RoadMap Extension and Policy Approximation



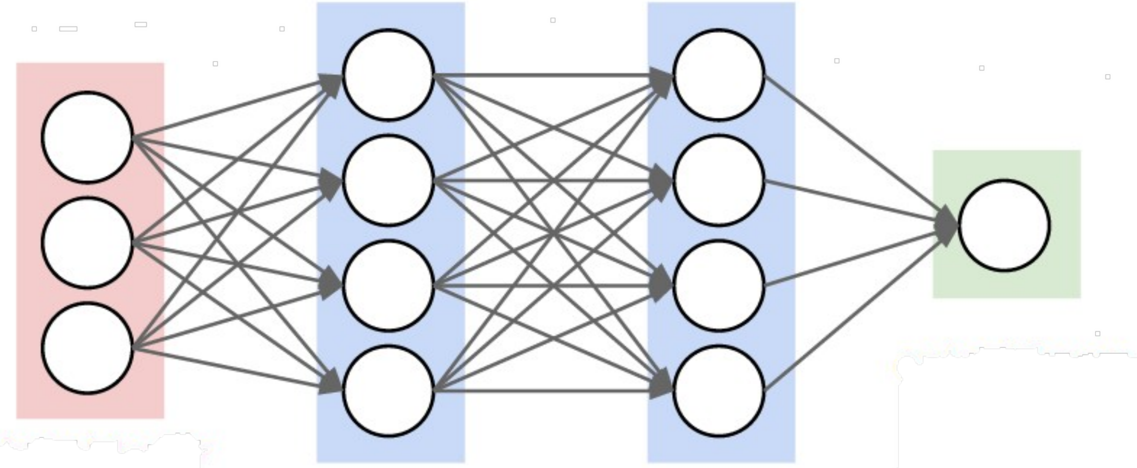
Kino-dynamic
Probabilistic Roadmap
30-50 states, dense connect

Sampling



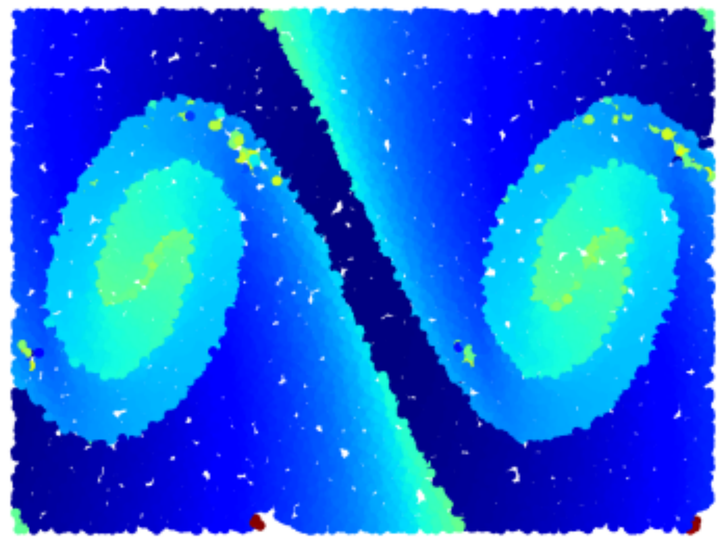
Dataset of
subtrajectories
10-100k items

Regression
(stoch.grad.)



Neural network
2x512 hidden units

Query



HJB approximation
Value function as metric
Policy function as warm-start

Roadmap
extension

LAAS
CNRS

Mansard et.al, 2018

Modified IREPA: Iteratively Learning and Extending Horizon

$V_\infty : x \rightarrow 0$

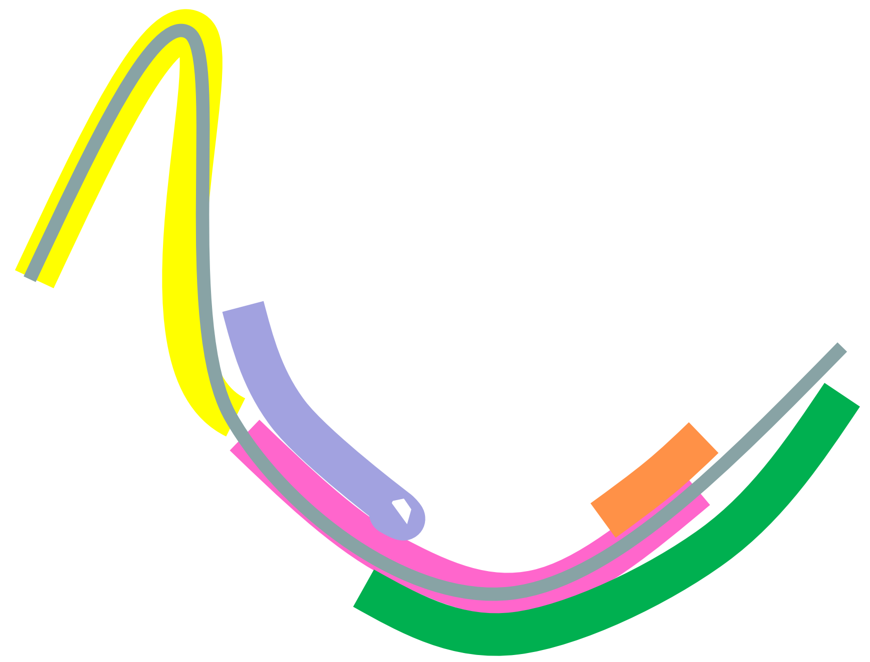
Crocoddyl

V_∞: x→ V(x, T, V_∞(x))

Optimal Control Solution

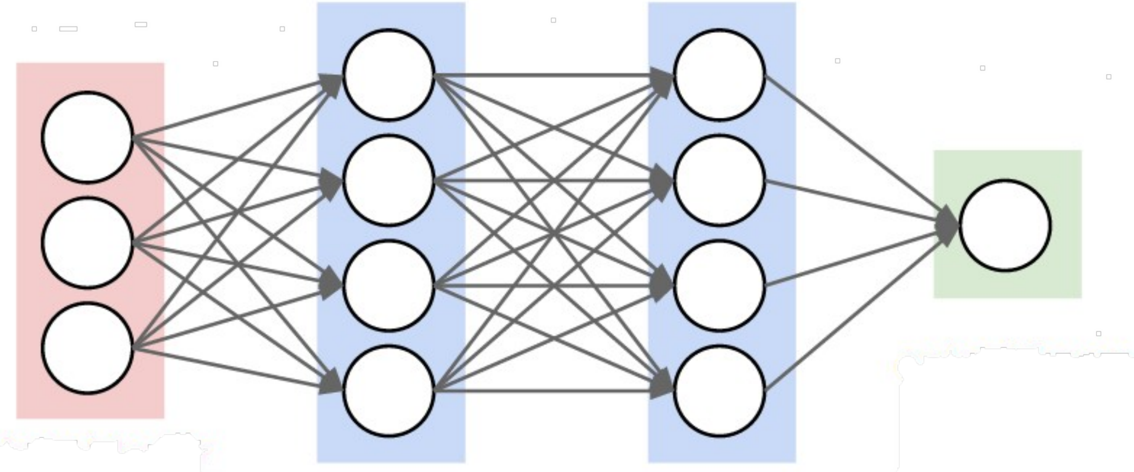
Unicycle, state dim 3

Sampling
Value function
at state x for
horizon T



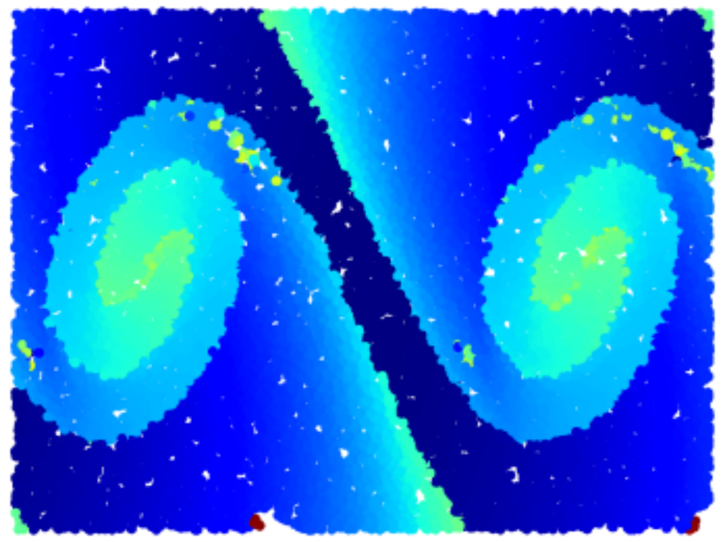
Dataset of Values

Regression
(stoch.grad.)



Neural network

Query

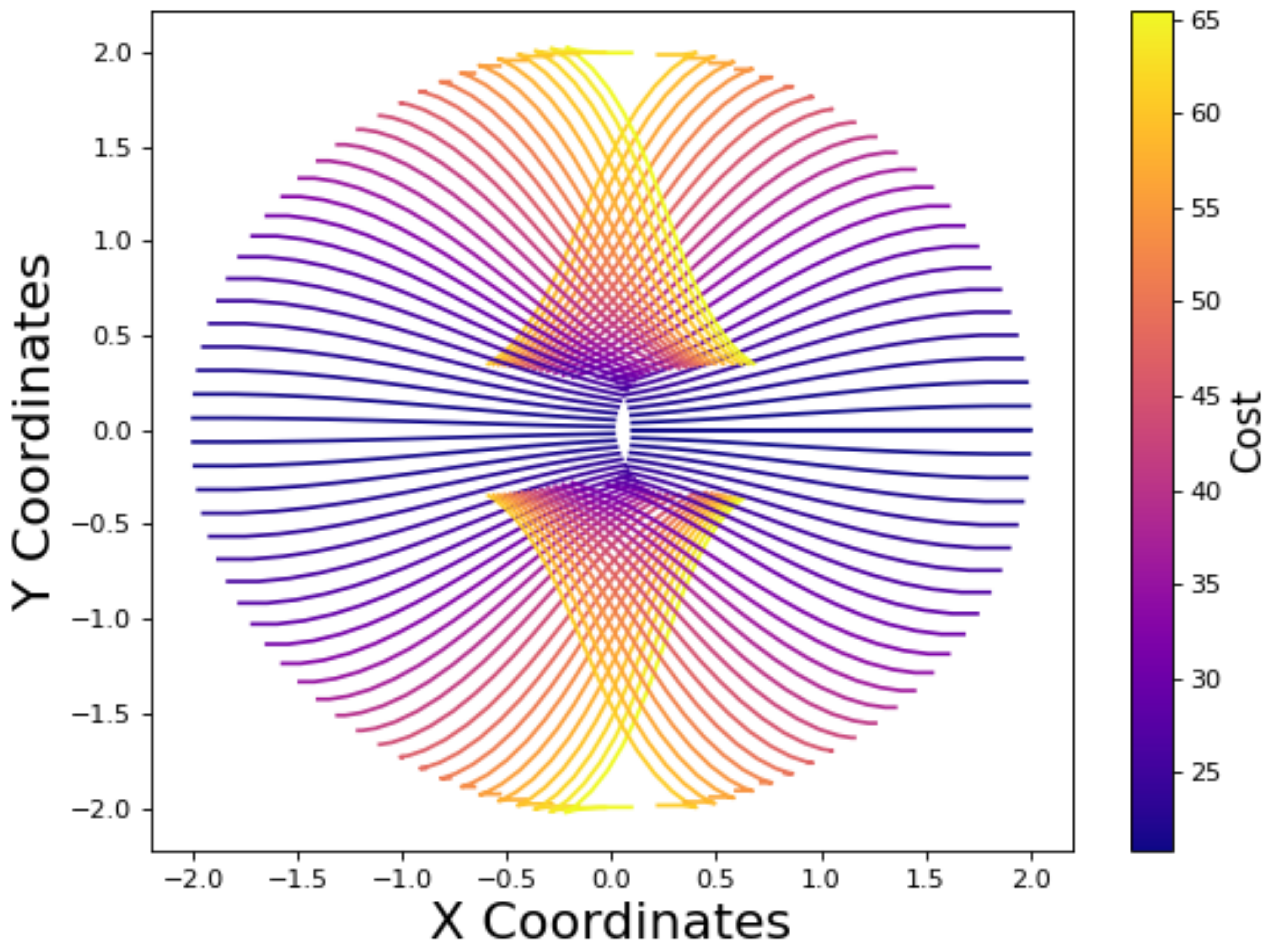


Prediction
V_∞

HJB approximation

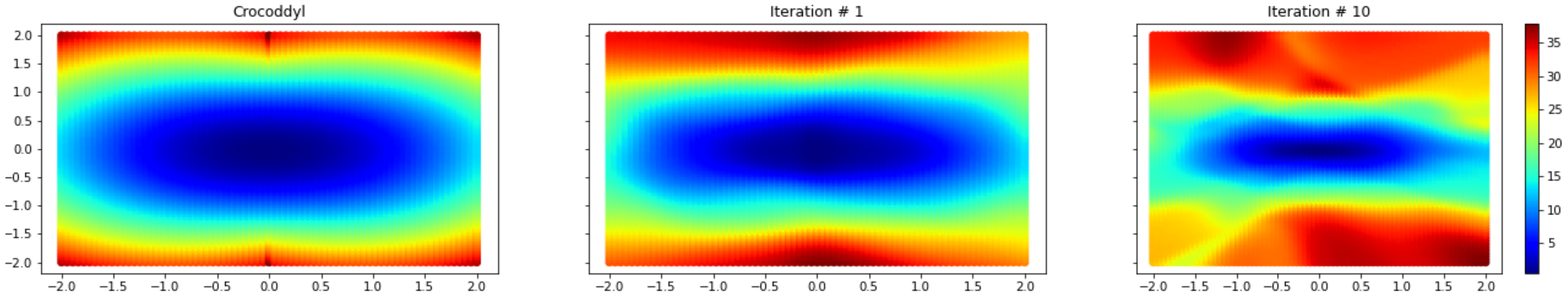
Value function as metric

Modified IREPA: Iteratively Learning and Extending Horizon



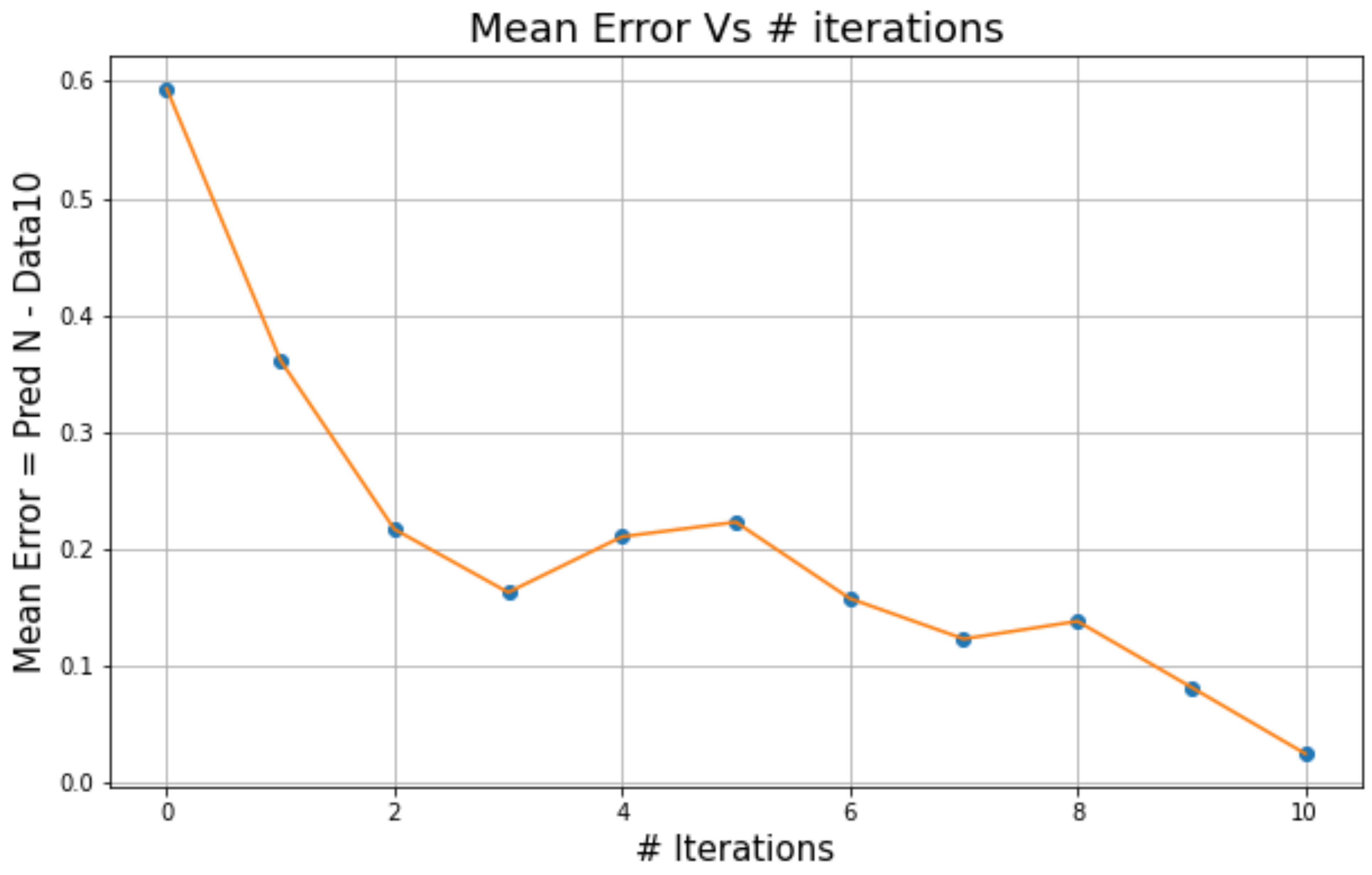
Trajectory and costs
for unicycle starting
at circle
and going to centre

Modified IREPA: Iteratively Learning and Extending Horizon

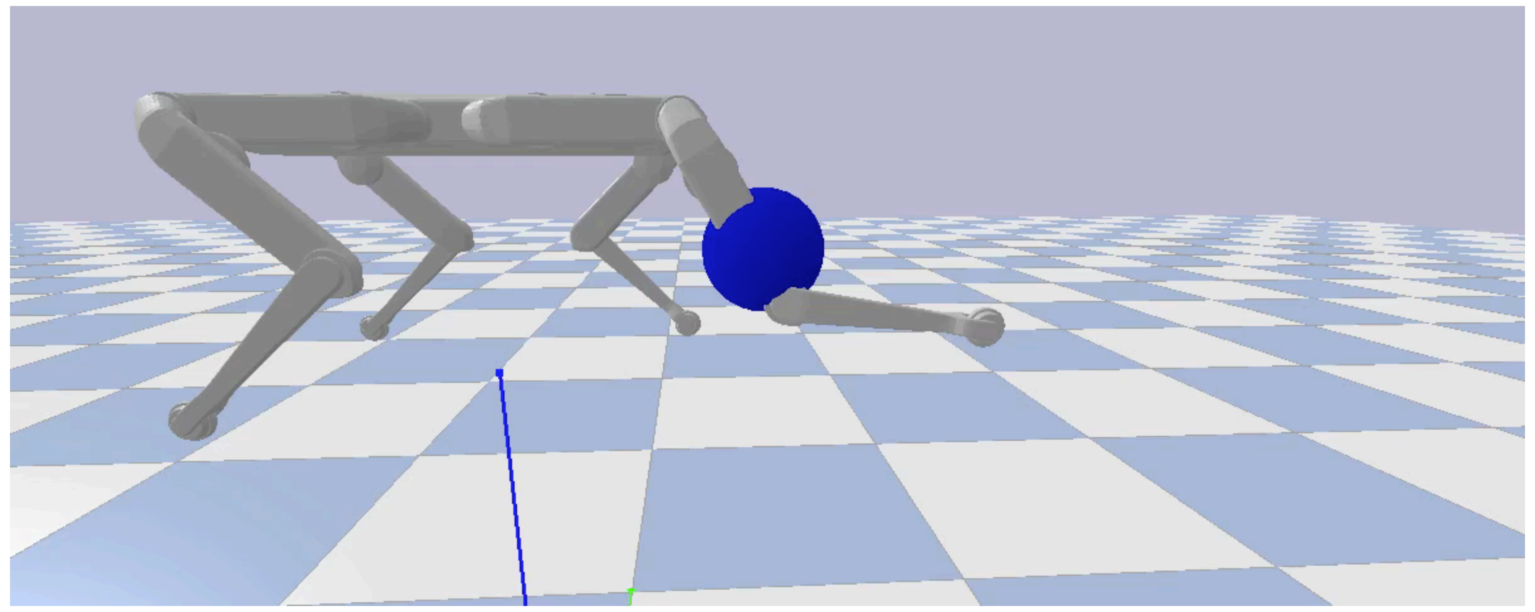


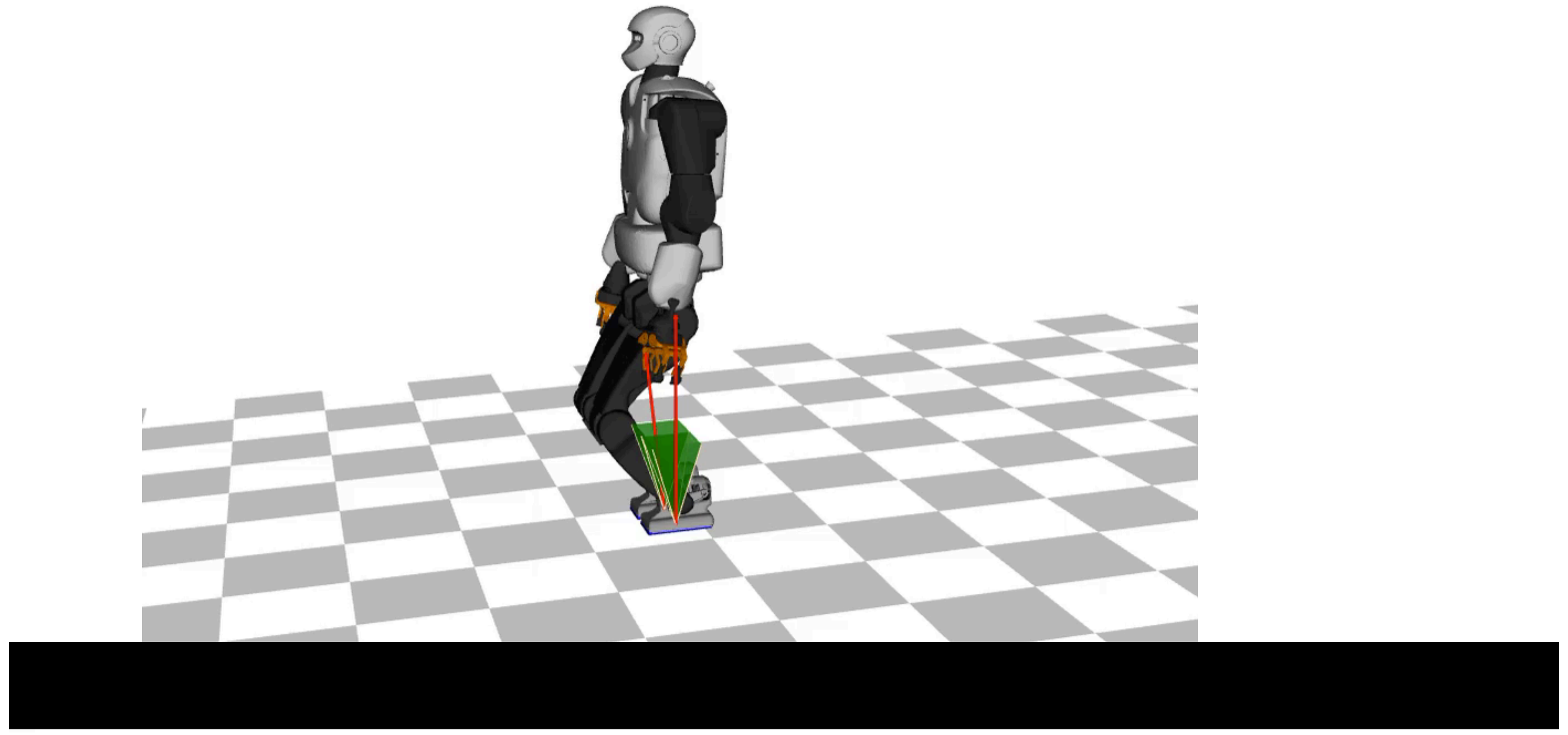
Value Function Scatter Plot for Crocoddyl (IREPA0), IREPA1, and IREPA10 iterations

Convergence of the Scheme



LAAS CNRS





Thank you!