https://github.com/nmansard/wan24

Optimal control for walking robots

Theory and practice with Crocoddyl

Nicolas Mansard

Gepetto, LAAS-CNRS & ANITI

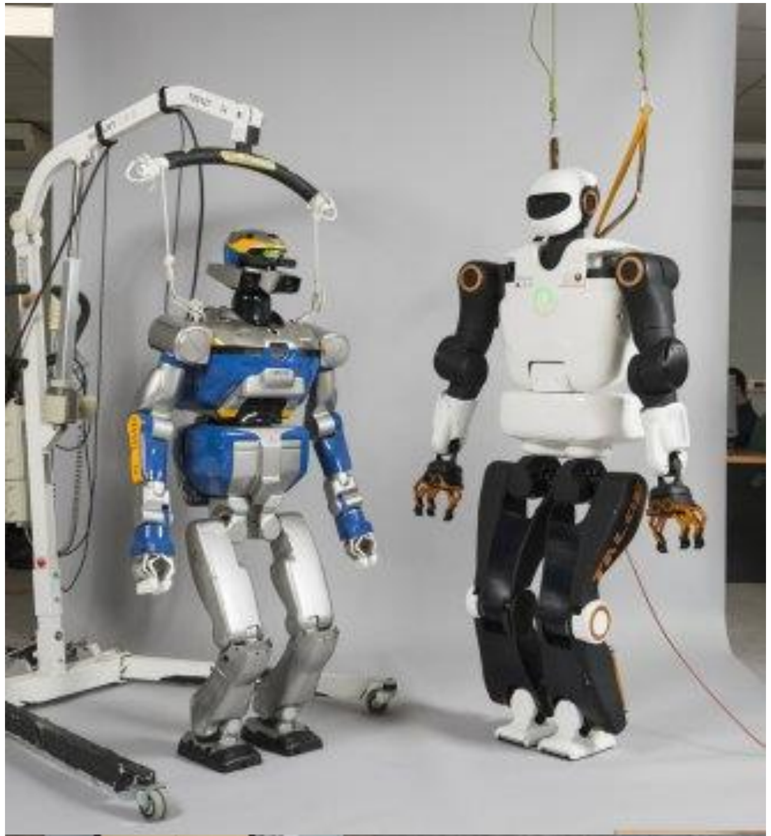
#00: Introduction

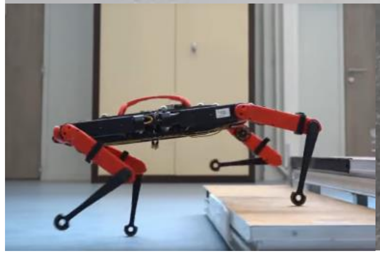
About Myself

- Directeur de Recherche LAAS-CNRS
- Chaire ANITI *« Artificial and Natural Movements »*
- Coordinator of collaborative projects
  - Entracte (Best ANR Numeric)
  - Memory of Motion (ERF Sucess story and *Étoile de l'Europe*)
  - Ongoing: Agimus and Joint-lab Dynamograde

- PhD from Univ Rennes (Francois Chaumette)
- Former researcher at
  - Stanford (O.Khatib), JRL-Japan (A.Kheddar), UW (E.Todorov)
- CNRS Bronze Medal

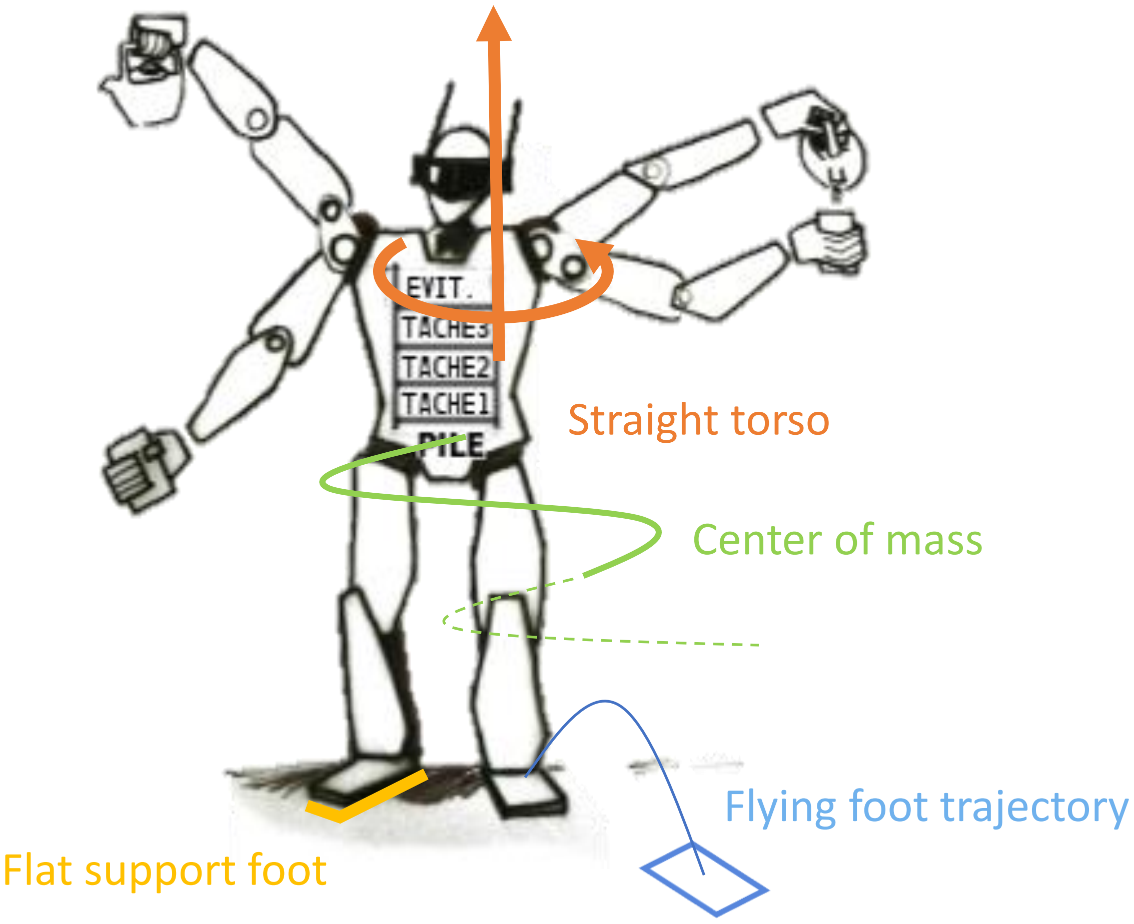
- Initiator, former lead dev or scientific coordinator
  - Pinocchio (and Eigenpy)
  - Crocoddyl (and Aligator)
  - HappyPose
  - Open Robot Dynamic Initiative
  - TSID, StackOfTasks

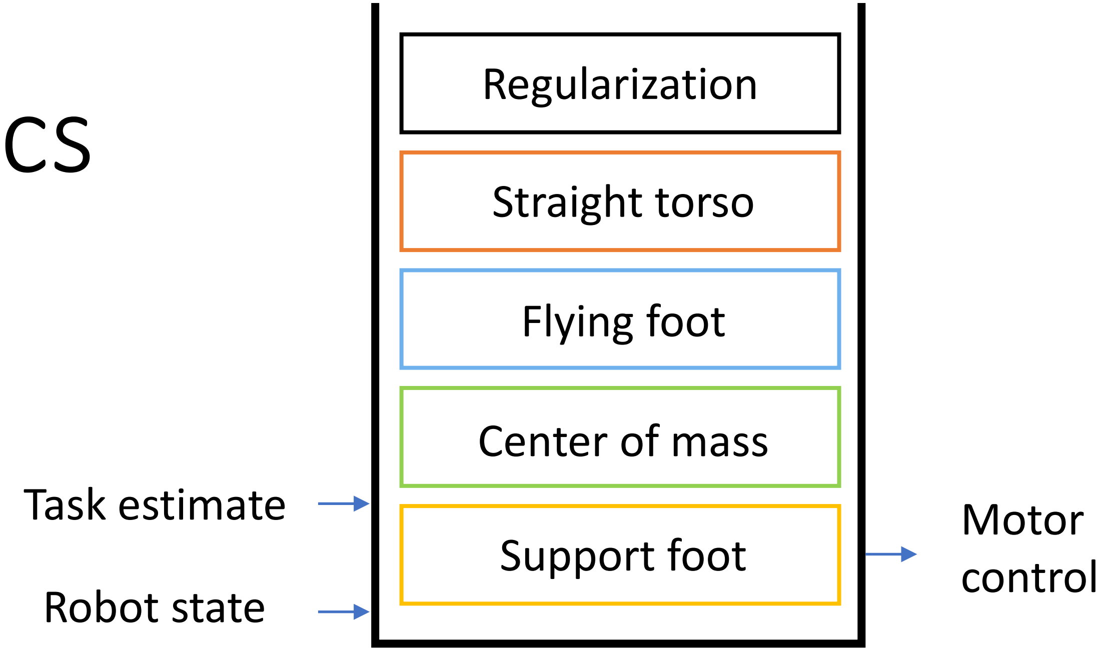




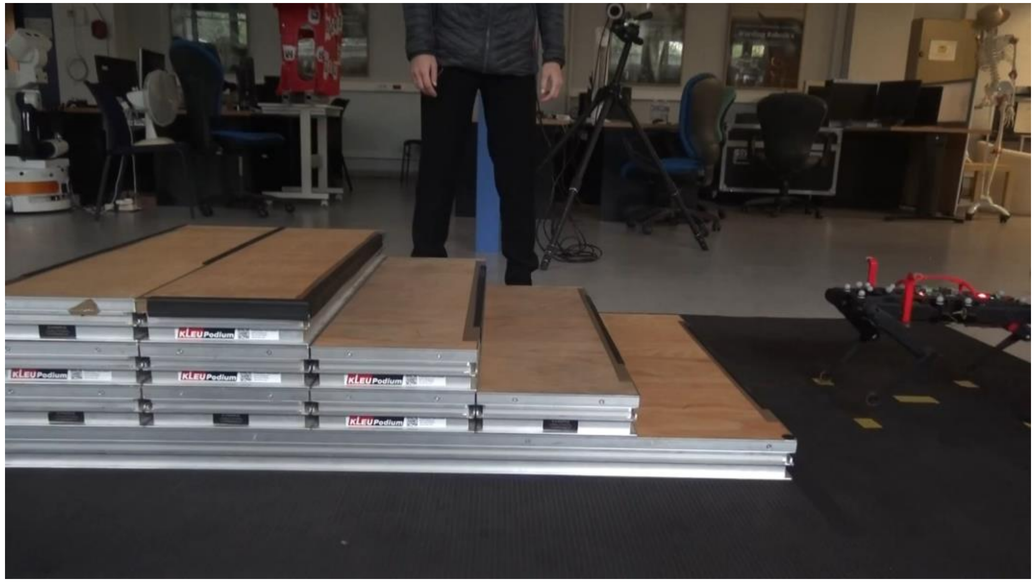
Good old control heuristics

… motion programed *by tasks*





Stack of tasks



Risbourg et al. (IROS’22) – https://hal.science/hal-03594629/

Whole-body control decision

High-level function objective

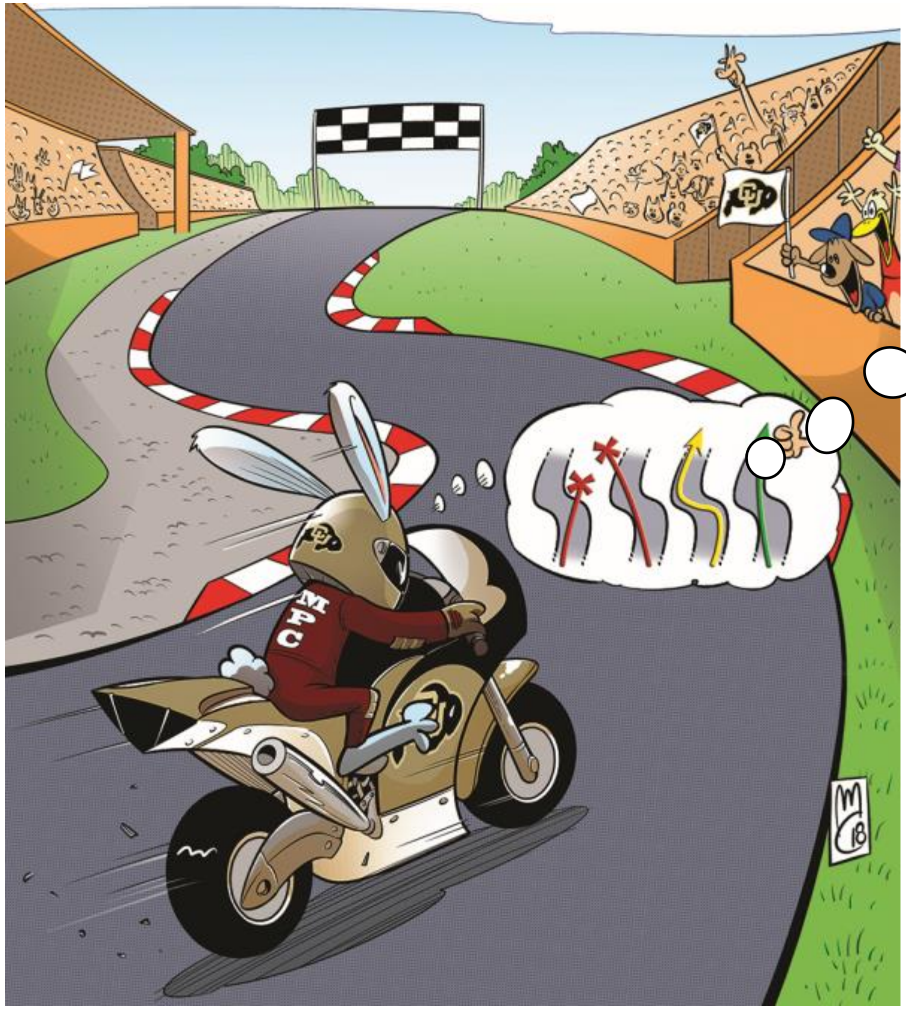
Go that way, bring me that object, open that door

Automatic
motion
intelligence

Model of the motion capabilities

From a known state, this control will bring the robot in that new state

Predictive control



Original artwork by Michele Carminati, commissioned by Marco M. Nicotra (U. Colorado Boulder)

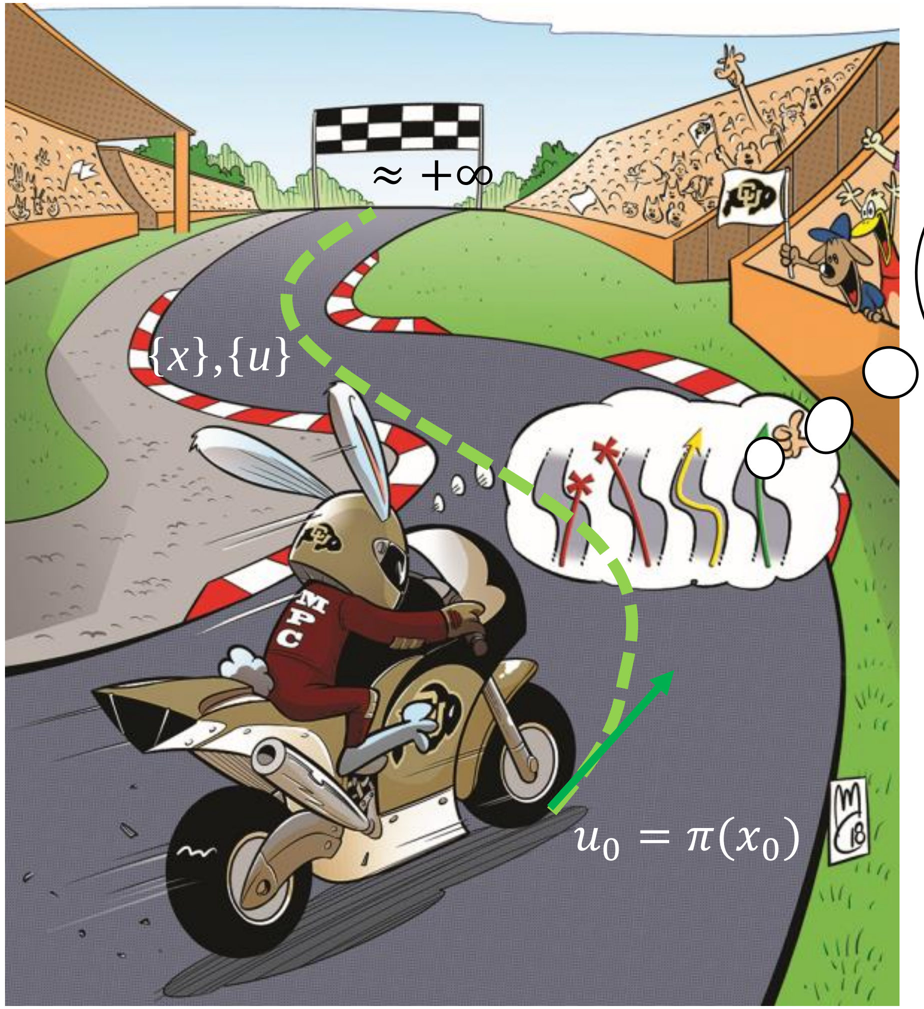
Decide: future robot trajectory

By optimizing an objective function (eg minimum energy)

Imposing:

- Known initial state
- Known evolution model (simulator)
- … and other constraints

Predictive control



Original artwork by Michele Carminati, commissioned by Marco M. Nicotra (U. Colorado Boulder)

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l(x_t, u_t)\, dt$$

so that

$$x(0) = \hat{x}$$

$$\dot{x}(t) = \underbrace{f(x(t), u(t))}_{\textit{Simulator}},\ \forall t=0..T$$

Efficient solvers …

Carlos Mastalli
Univ. Watt @ Edinbourgh

- Features expected from a good optimal control solver
  - Stable prediction: **multiple shooting**
  - Sparsity: **differential dynamic programming**
  - Strict constraints: **augmented Lagrangian**
  - Our solver incorporates all three !

- Performance on real case studies
  - 4 trot cycles for a quadruped: 8K vars, 12 iterations, **9ms / iter**
  - 2 steps for a humanoid: 12K vars, 18 iterations, **13ms / iter**

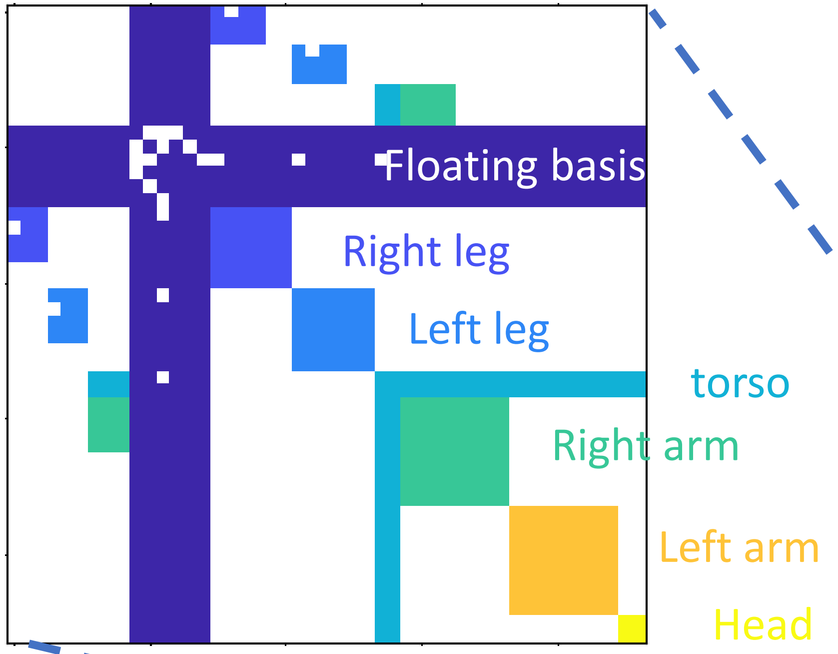
Crocoddyl
https://github.com/loco-3d/crocoddyl

BSD
BSD-2 License

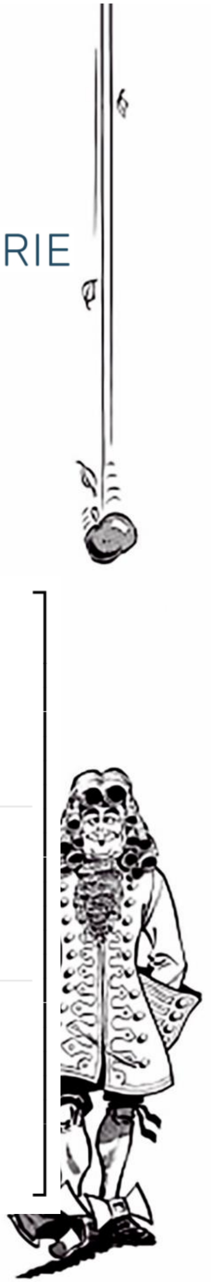
Mastalli et al. (ICRA’20) – https://hal.science/hal-02294059/

… for efficient problems

Justin Carpentier
Inria Paris / PR[AI]RIE



$$
\left[\begin{array}{ccc|cc|ccc}
L_{xx} & & & L_{xu} & & -I & F_x^T & \\
& \ddots & & & \ddots & \ddots & \ddots & \\
& L_{xx} & & & L_{xu} & & -I & F_x^T \\
& & L_{xx} & & & & & -I \\
\hline
L_{ux} & & & L_{uu} & & & F_u^T & \\
& \ddots & & & \ddots & & \ddots & \\
& L_{ux} & & & L_{uu} & & & F_u^T \\
\hline
-I & & & & & & & \\
\boxed{F_x} & -I & & F_u & & & & \\
\ddots & \ddots & & \ddots & & & & \\
& F_x & -I & & F_u & & &
\end{array}\right]
$$



Pinocchio
https://github.com/stack-of-tasks/pinocchio

BSD
BSD-2 License

Carpentier et al. (2019) – https://hal.science/hal-01866228v2

Progress in numerical optimization



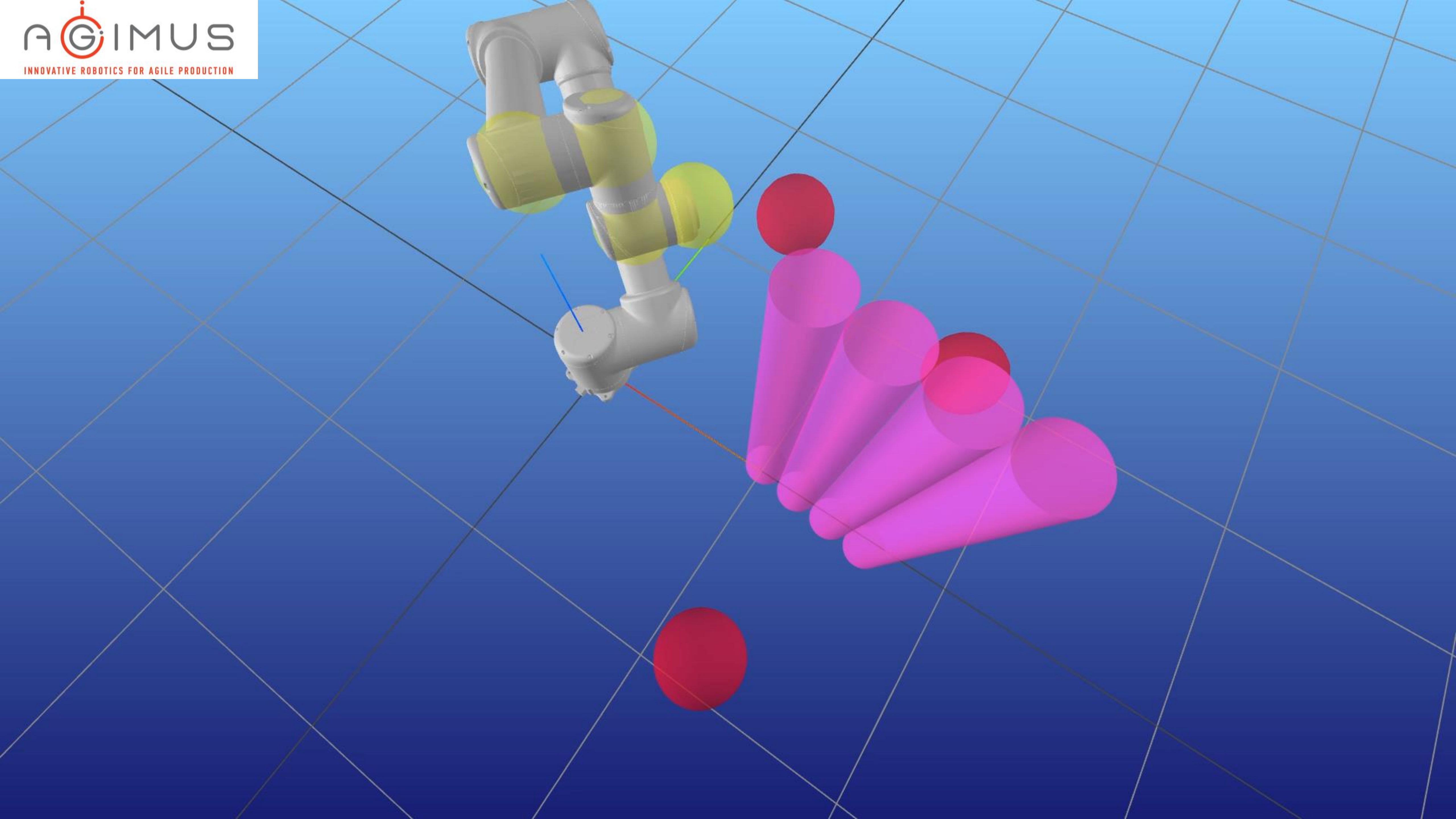
- With a SQP approach (with MiM@NYU)
  - Using the operator-splitting method (OSQP) with Ricatti linear solver
  - Following the main-stream literature
  - Implemented as an ad-on to Crocoddyl (MiM-solver package)

Jordana+Kleff+Meduri et al (subm TRO 2023) – https://ens.hal.science/hal-04330251v1

- With a augmented Lagrangian approach (with Willow@Inria)
  - Using proximal formulation to handle the conditioning
  - Following the main-stream literature
  - Implemented in a proposition of renew of Crocoddyl (Aligator package)

Jallet et al. (subm TRO 2023) – https://hal.science/hal-04332348v1

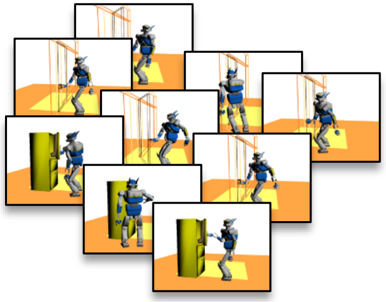
AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION



Memory of motion

memmo

Off-line
Optimization and encoding



Off-line
Exploration by optimization
Globalization

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l(x_t, u_t)dt$$

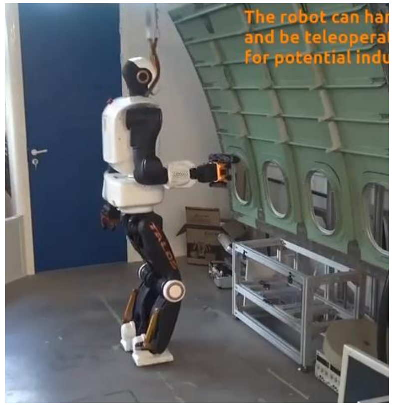
$$\text{s.t.} \quad x(0) = \hat{x},$$

$$\dot{x}(t) = f\big(x(t), u(t)\big), \forall \mathrm{t{=}0..T}$$

On-line
Whole-body predictive control
High-freq by local convergence

https://github.com/MeMory-of-MOtion/docker-loco3d

H2020 Memmo



Case-study #1
Humanoid in factory
of the future

Case-study #2
Exoskeleton
for disabled people

Case-study #3
Quadruped for
inspection

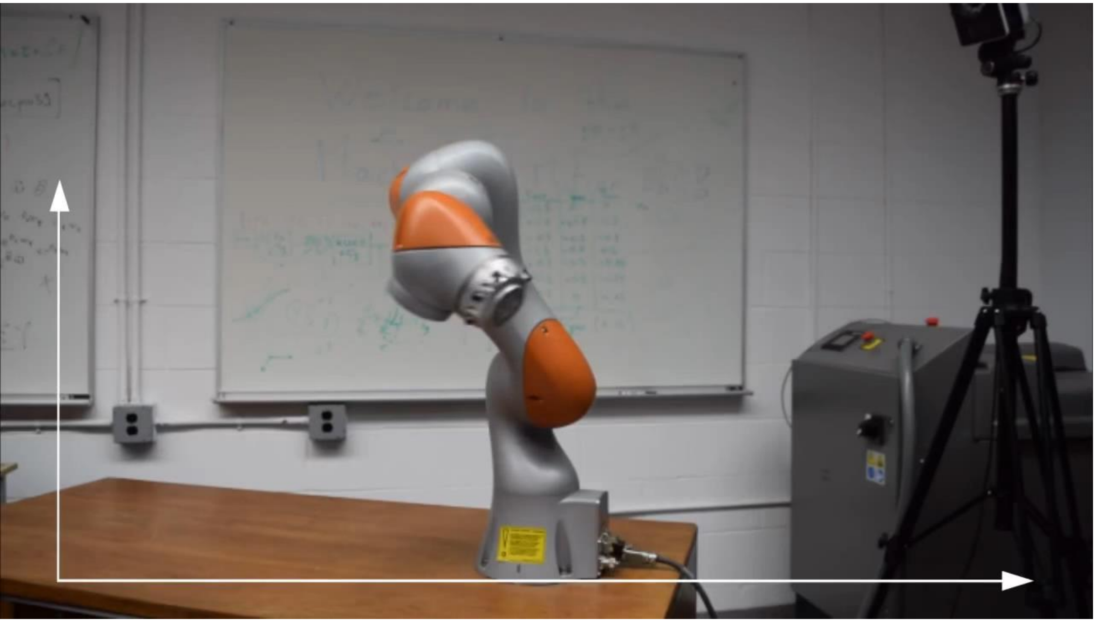
@CNRS

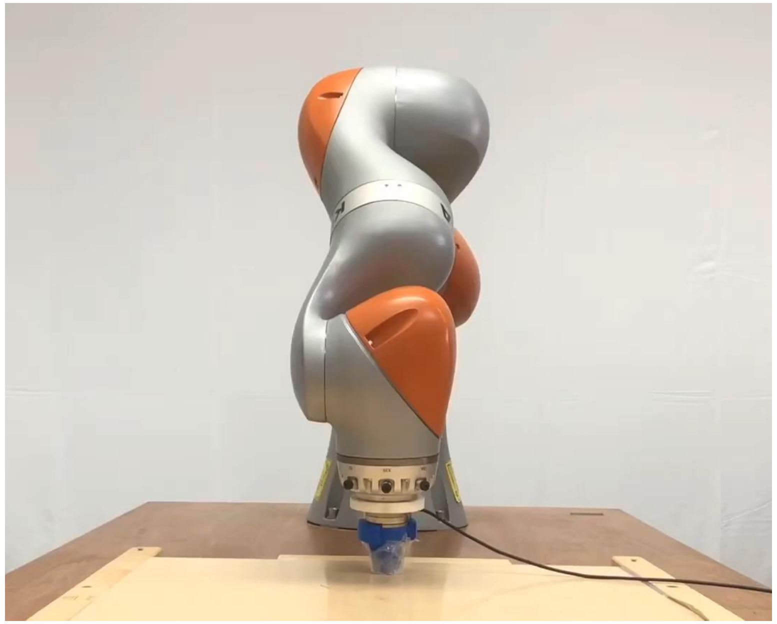
@MPI

@UEDIN

Development of a generic methodology to generate complex movements for robots with legs and arms in real-time

Efficient solver … for efficient problems





Optimize
1 sec of preview
every 1 ms
(2000 variables)

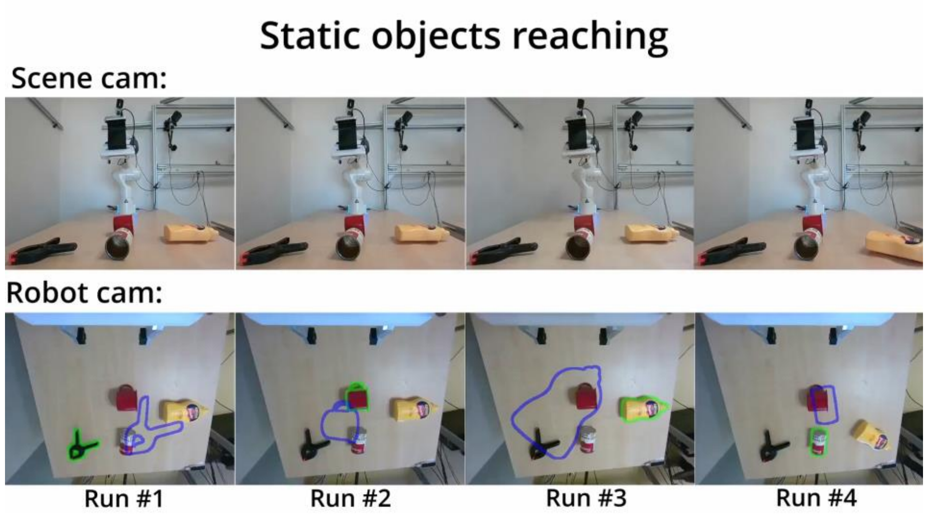
Ludovic Righetti

Sébastien Kleff

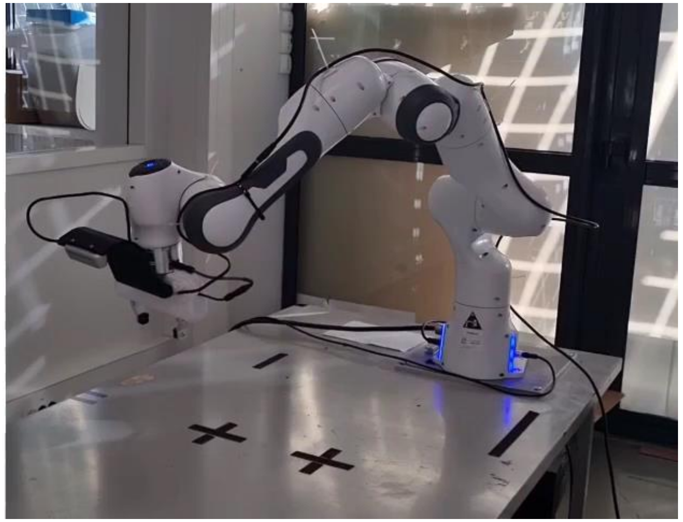
NYU

Kleff et al. (IROS’22) – https://hal.science/hal-03594295/

Efficient solver … for efficient problems



- Model Predictive Control Torque controlled robot.
- Pick and place like task while staying compliant.
- 4 different poses alternatively and randomly reached every 3s.



Mederic Fourmy **Vladimir Petrik**

With visual feedback

https://arxiv.org/pdf/2311.05344

Arthur Haffemayer

With collision avoidance

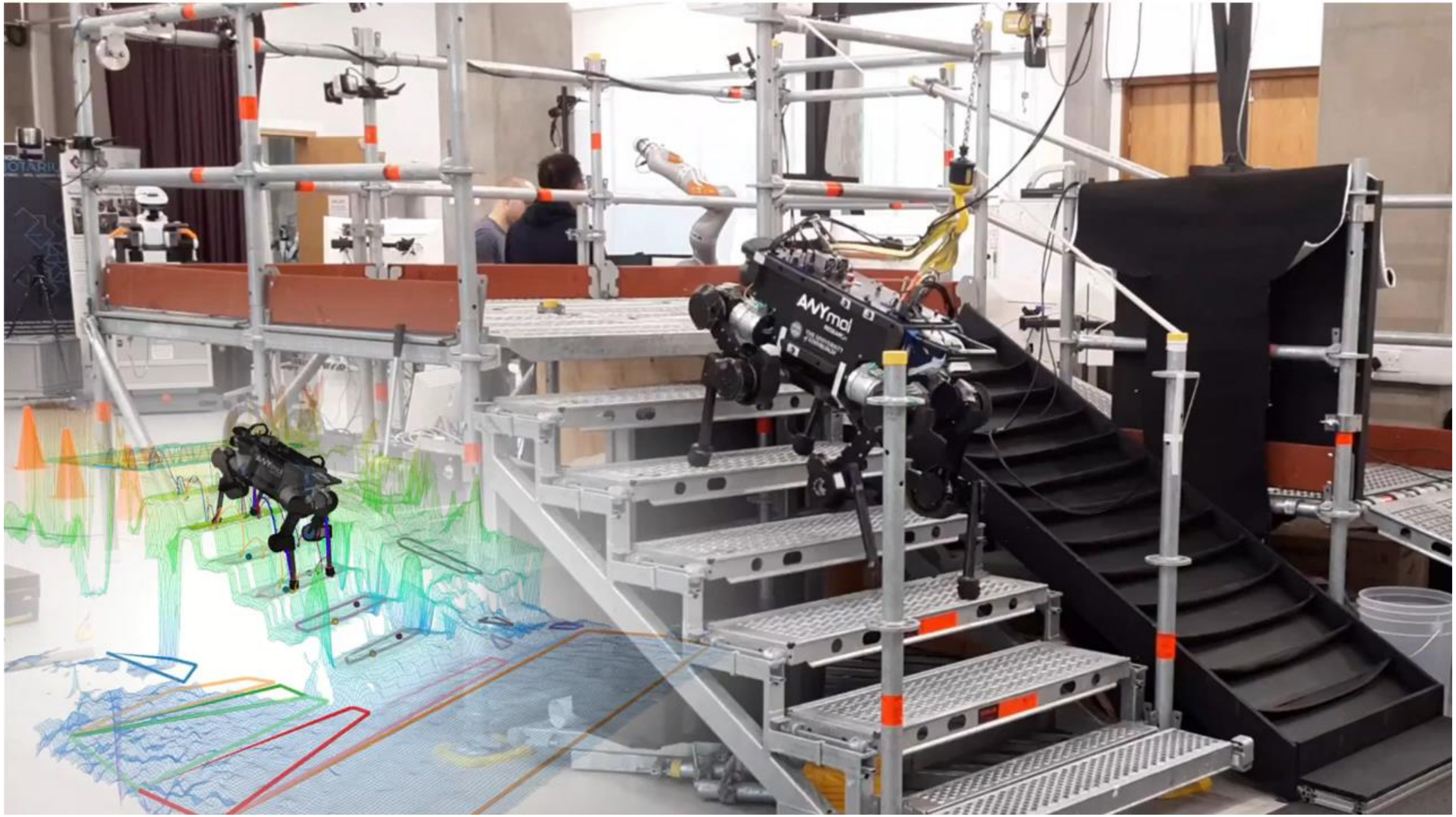
https://hal.science/hal-04425002

Efficient solver … for efficient problems

Ewen Dantec

Dantec et al. (RAL’22) – https://cnrs.hal.science/hal- 03419712v2

Efficient solver … for efficient problems



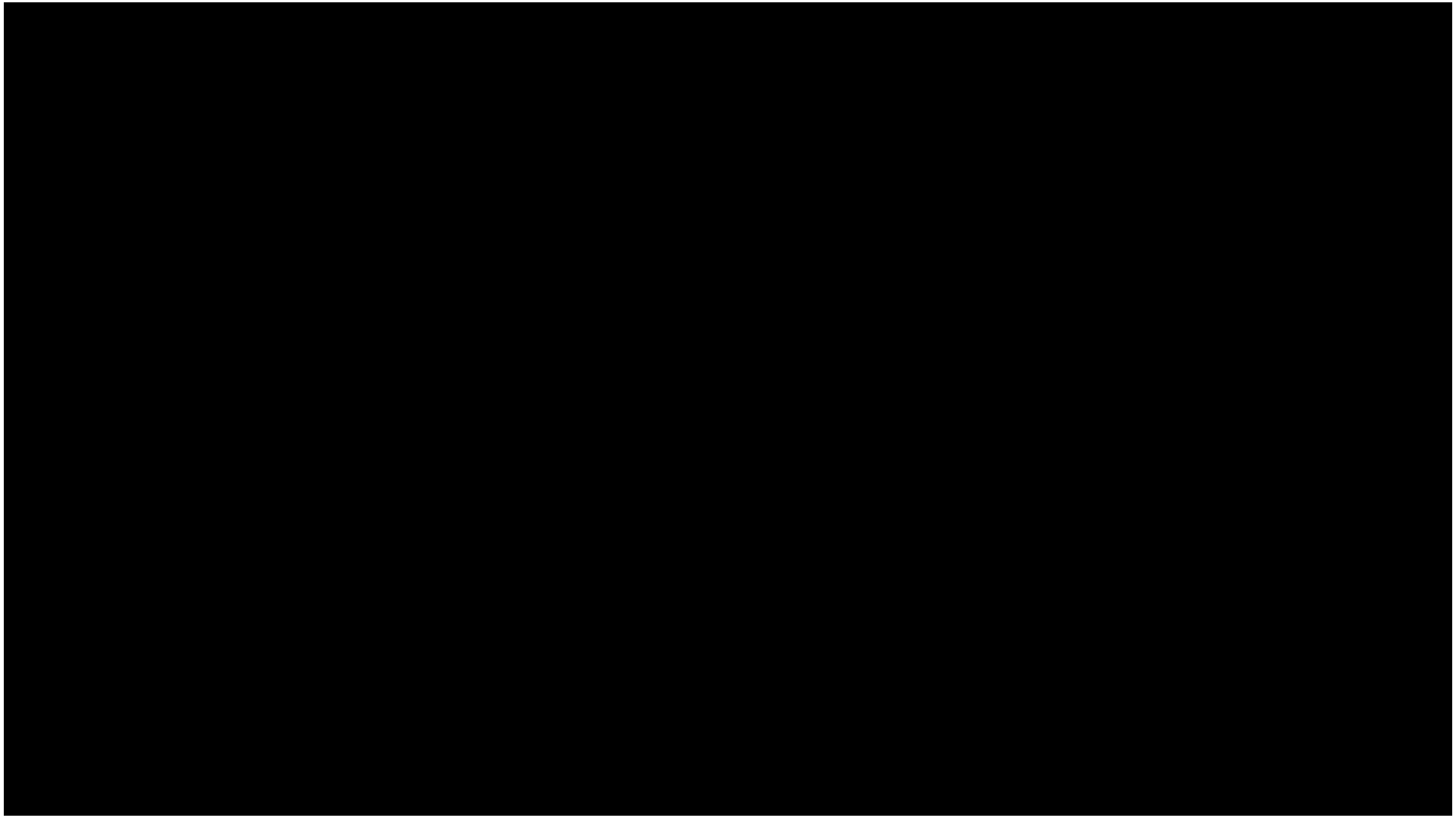
Thomas Corberes **Steve Tonneau**

ORI OXFORD ROBOTICS INSTITUTE

THE UNIVERSITY OF EDINBURGH

Corberes et al. (subm IJRR'23) – https://arxiv.org/abs/2305.08926

Efficient solver … for efficient problems



Fanny Risbourg

Alessandro Assirelli

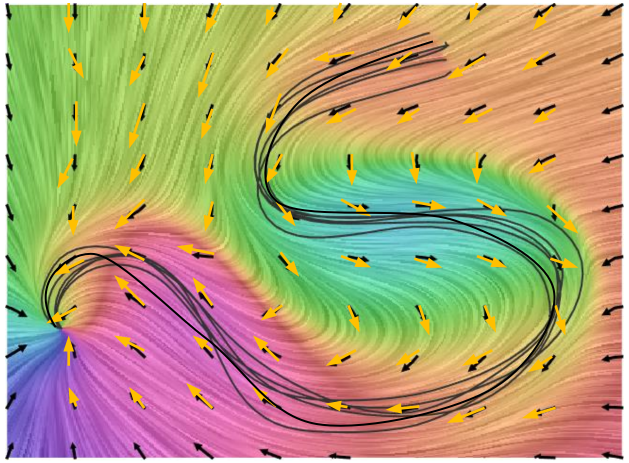
Wilson Jallet

Assirelli et al. (ArXiv 2022) – https://hal.science/hal-03778738

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l(x_t, u_t)dt$$

$$\text{s.t.} \quad x(0) = \hat{x},$$

$$\dot{x}(t) = f\big(x(t), u(t)\big), \forall t=0..T$$



Trajectory optimization

U: t ⟶ u(t)

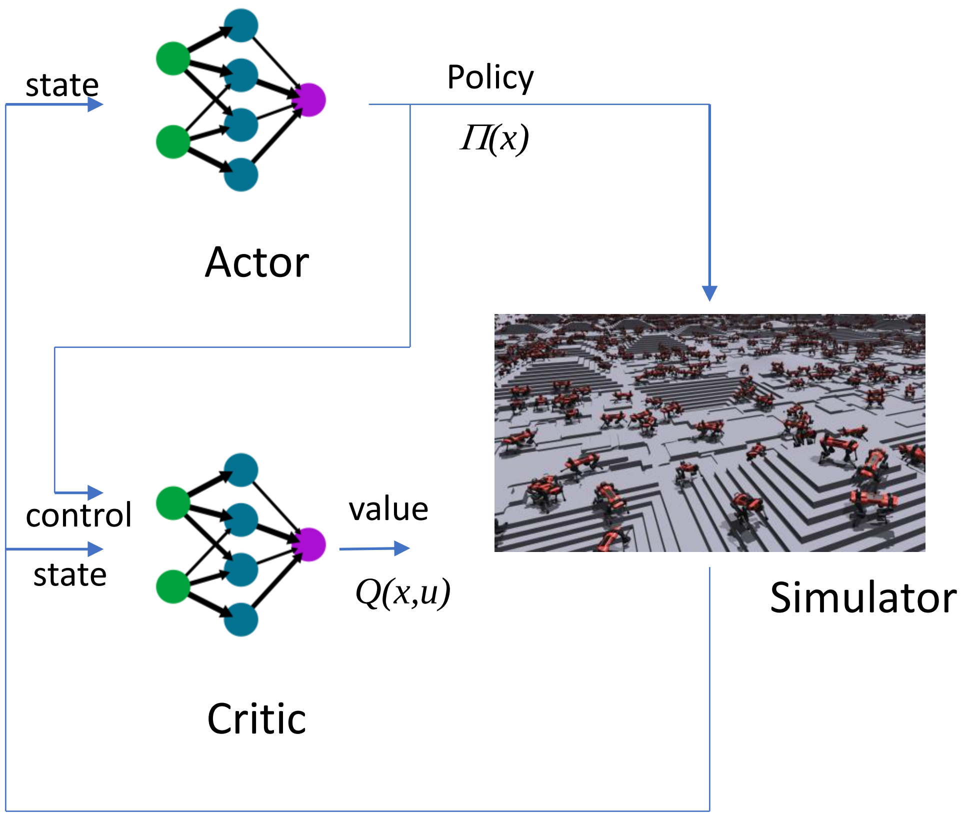
Motion planning

Policy optimization

Π: x ⟶ u = Π(x)

Reinforcement learning

Reinforcement learning

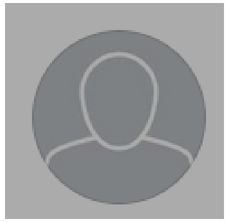


$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l(x_t, u_t)dt$$

$$\text{s.t.} \quad x(0) = \hat{x},$$
$$\dot{x}(t) = f\big(x(t), u(t)\big), \forall t=0..T$$

Belman principle

$$\Pi(x) = \underset{u}{\operatorname{argmin}}\, Q(x,u)$$
$$Q(x,u) = l(x,u) + Q(x', \Pi(x))$$



Michel Aractingi

Philippe Souères

Thomas Flayols

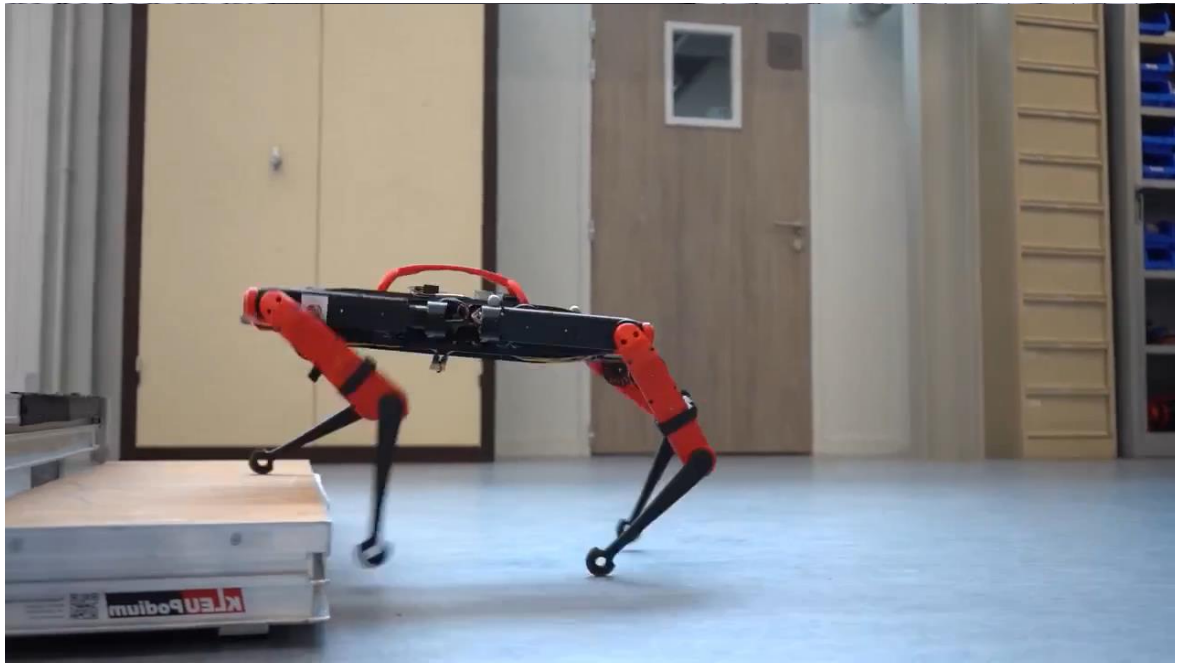
Aractingi et al. (ScRep 2023) – https://laas.hal.science/hal-03761331/

Constraint as termination

- RL with chance constraint

$$\min_{\pi} \; E\left[\sum_{t=0}^{T} \gamma^t \, l(x_t, u_t)\right]$$

s.t.
$$x_0 = \hat{x}$$
$$x_{t+1} \sim P(x_{t+1} | x_t, u_t)$$
$$\mathrm{P}[c_i(x_t, u_t) \geq 0] \leq \epsilon, \forall i$$

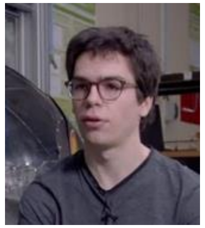


- Reformulation as termination

$$\min_{\pi} \; E\left[\sum_{t=0}^{T} \left(\prod_{k=0}^{t} \gamma(1 - \delta(x_k, u_k)\right) l(x_t, u_t)\right]$$

s.t.
$$x_0 = \hat{x}$$
$$x_{t+1} \sim P(x_{t+1} | x_t, u_t)$$

Elliot Chane-Sane



Pierre-Alexandre Leziart

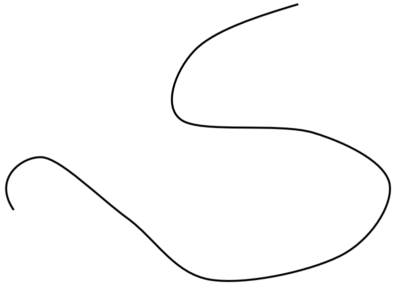
Thomas Flayols

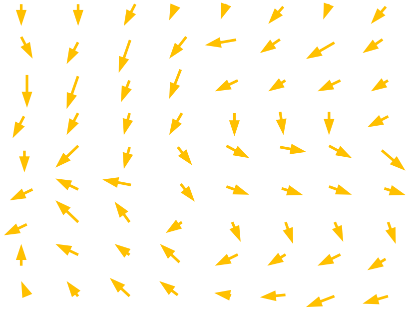
Chane-Sane+Leziart (Subm IROS'24) – https://laas.hal.science/hal-04523167v1

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l(x_t,u_t)dt$$

$$\text{s.t.} \quad x(0) = \hat{x},$$

$$\dot{x}(t) = f\big(x(t), u(t)\big), \forall t=0..T$$





✓ trajectory optimization

✓ super-linear convergence

✓ real-time computation

✓ constraint satisfaction

– local minima (no global policy)

– difficulty with discontinuous dynamics

– no inclusion of multi-modal sensing

✓ (global) policy optimization

✓ handles discontinuities

✓ multi-modal sensing inclusion

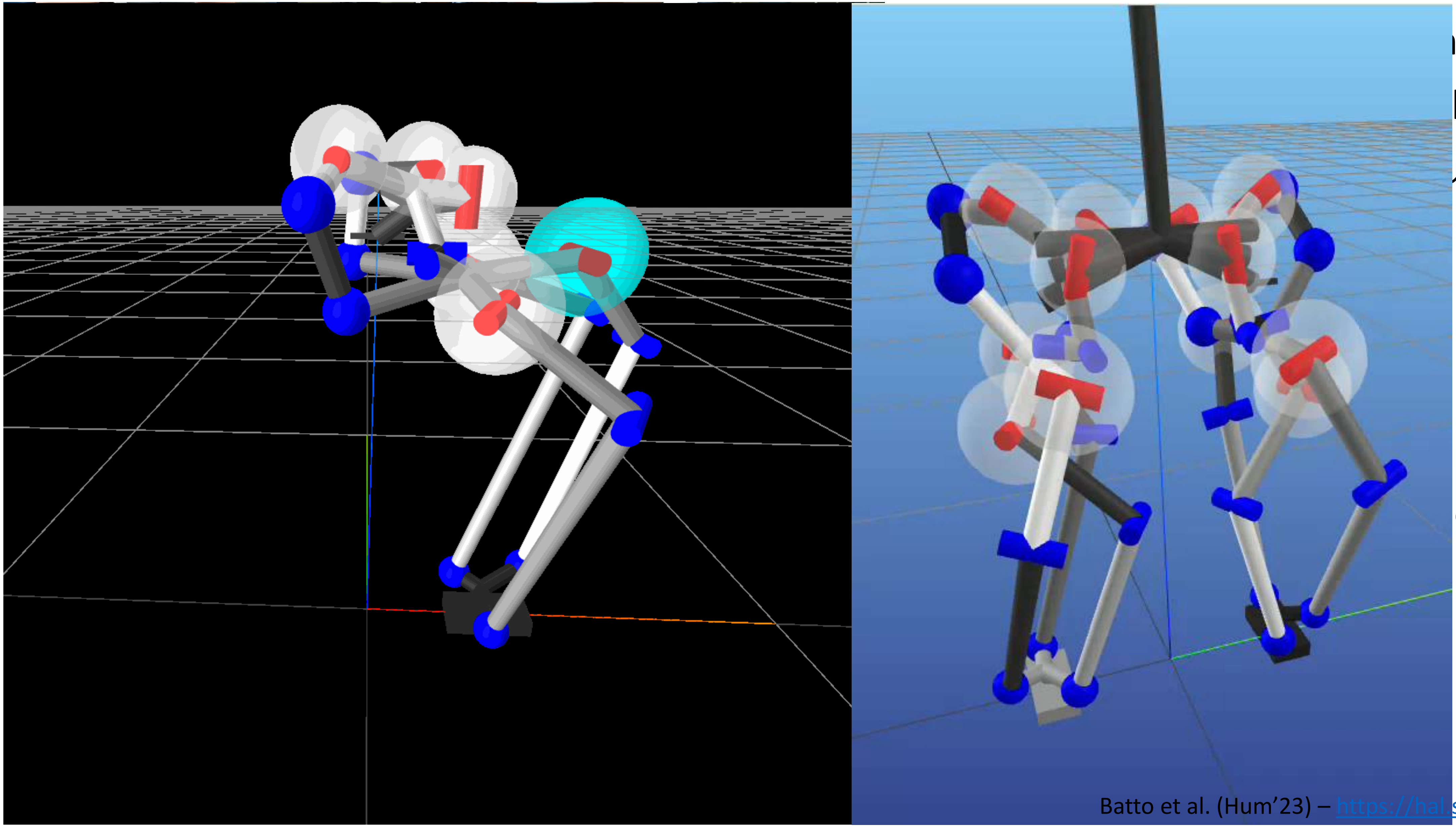
– no guaranteed convergence

– little use of model information

– difficult transfer to robots

– no constraint satisfaction

Software can’t anything without (good)



ıeetah

ı platform

ırce

Thomas Flayols

esign …

Virgile Batto

Batto et al. (Hum’23) – https://hal.science/hal-04191553v2

Main messages

ANITI
Artificial &
Natural Movements

We should both
OPTIMISE and LEARN
the movements of a robot !

- **Trajectory optimization** is necessary
  - 10,000 variables in 10 ms
  - Accurate convergence, constraints satisfaction, generalization

- **Policy learning** is necessary
  - Globalization using a memory of motion
  - Toward super-linear reinforcement algorithms

Main software

- Pinocchio: models and derivatives

  Current v2.7 … secret v3.0 in binary with future public release

  https://github.com/stack-of-tasks/pinocchio

- Crocoddyl: fast optimal control for whole-body models

  Current v2.1 … Aligator released with Pinocchio v3.0 dependency

  https://github.com/loco-3d/crocoddyl/

- HappyPose: accurate object pose estimation

  Current v0.0 … v0.1 soon (10 days?)

  https://github.com/agimus-project/happypose

- MPC on Panda (and future TiagoSEA / TiagoPRO)

  WIP, sources are public

Agenda

- Day 1 — 9h30 → 18h30
  - Intro
  - Basis of nonlinear constrained optimization
  - Optimal control and dynamic programing
  - Transcription: from OCP to NLP
  - First practical case study (with Casadi)
- Day 2 — 9h30 → 12h30, 14h → 18h30
  - Differential dynamic programming (DDP)
  - Optimal control with hard constraints
  - Second practical case study (with Crocoddyl)
  - Model predictive control
  - Q&A, a look at the C++ code
- Day 3 — 9h → 18h
  - Let’s make an exoskeleton climb the stairs
  - Hackaton style?
- Options
  - Connections between trajectory optimization and reinforcement learning
  - Derivatives