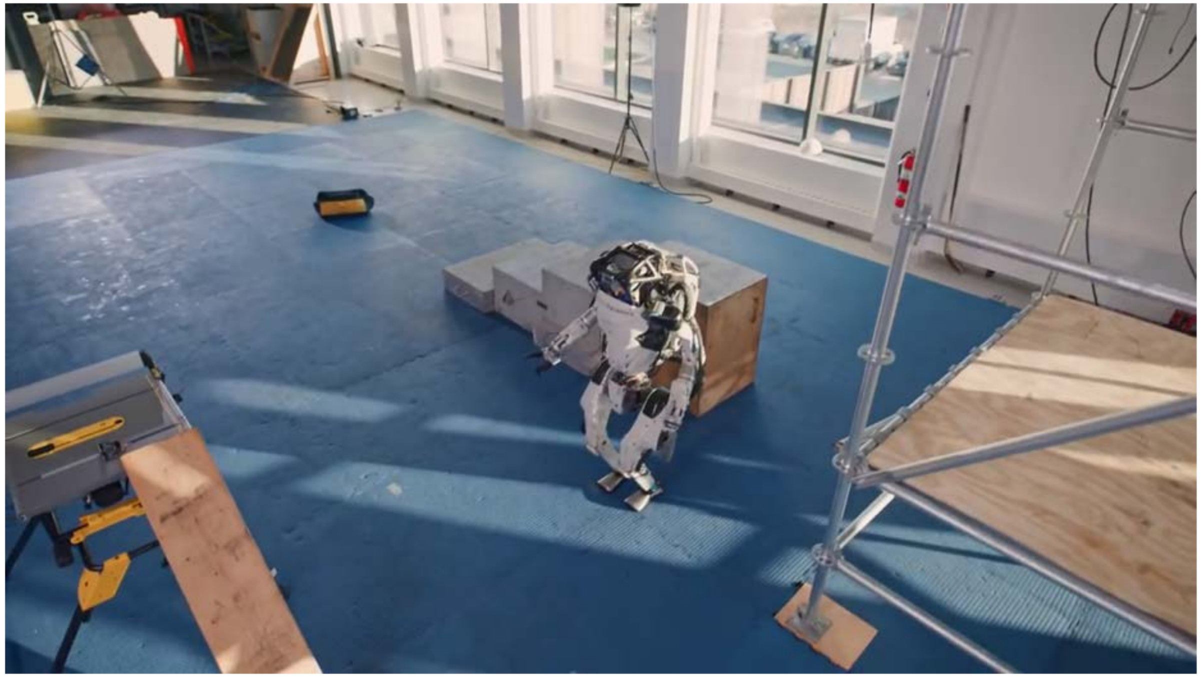
Artificial & Natural Movement

Nicolas Mansard – LAAS-CNRS

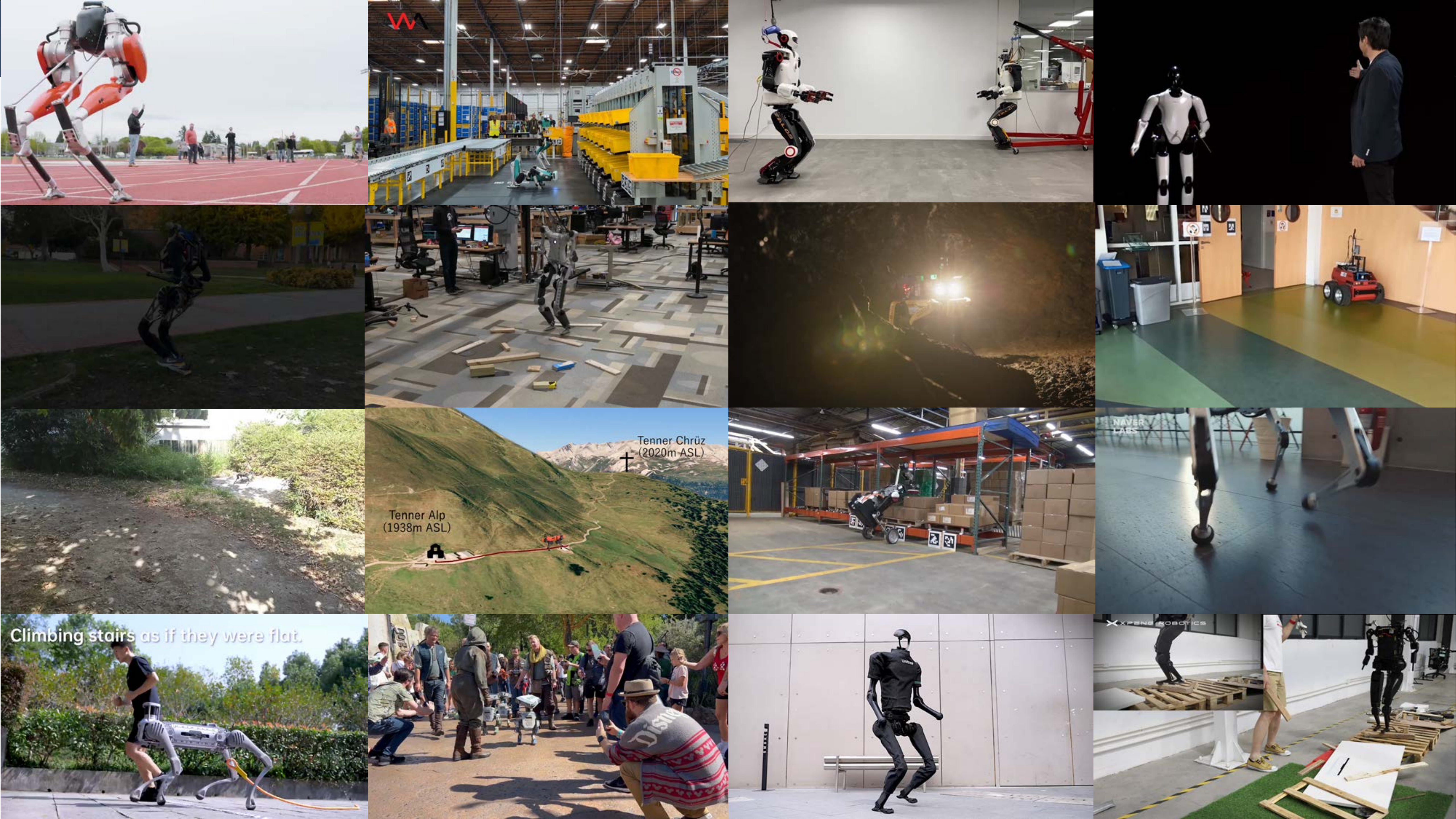
Immense progress in 5 years

Unitree quadruped

BostonDynamics Atlas

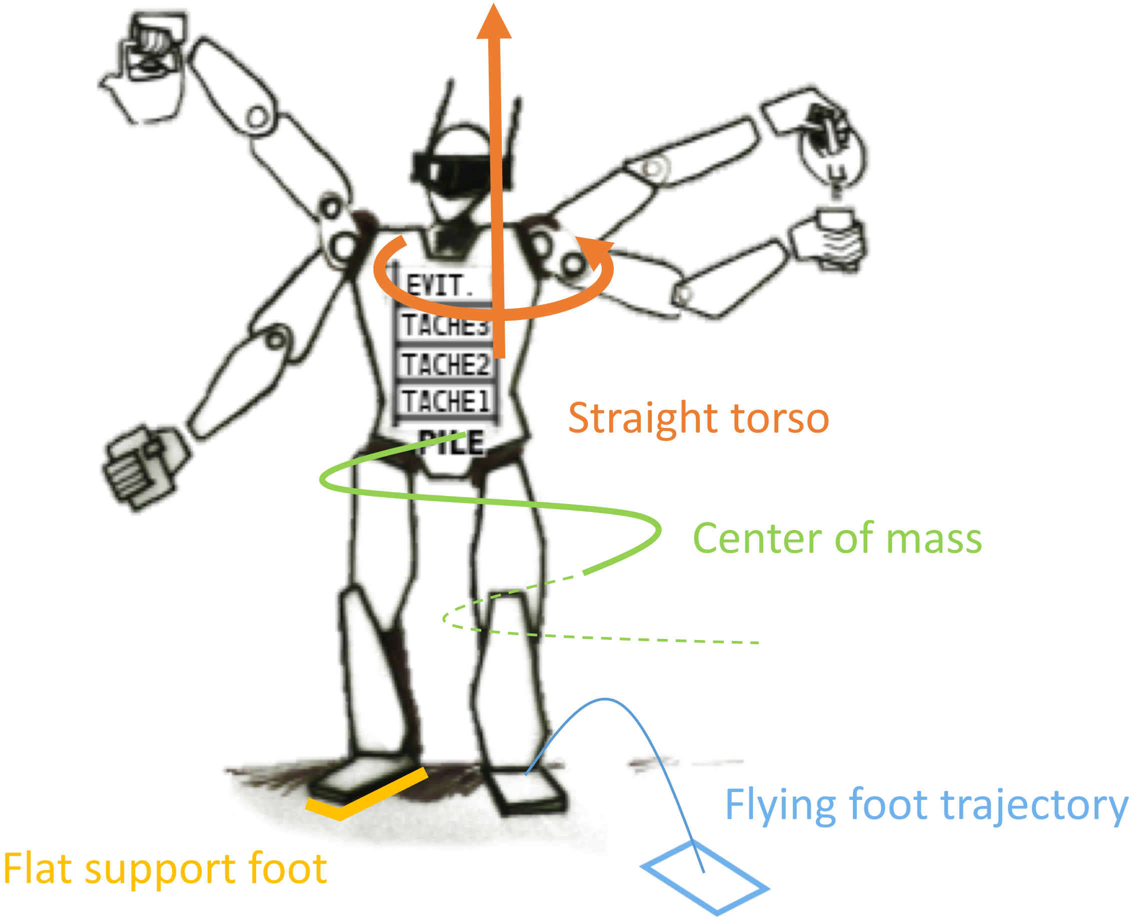


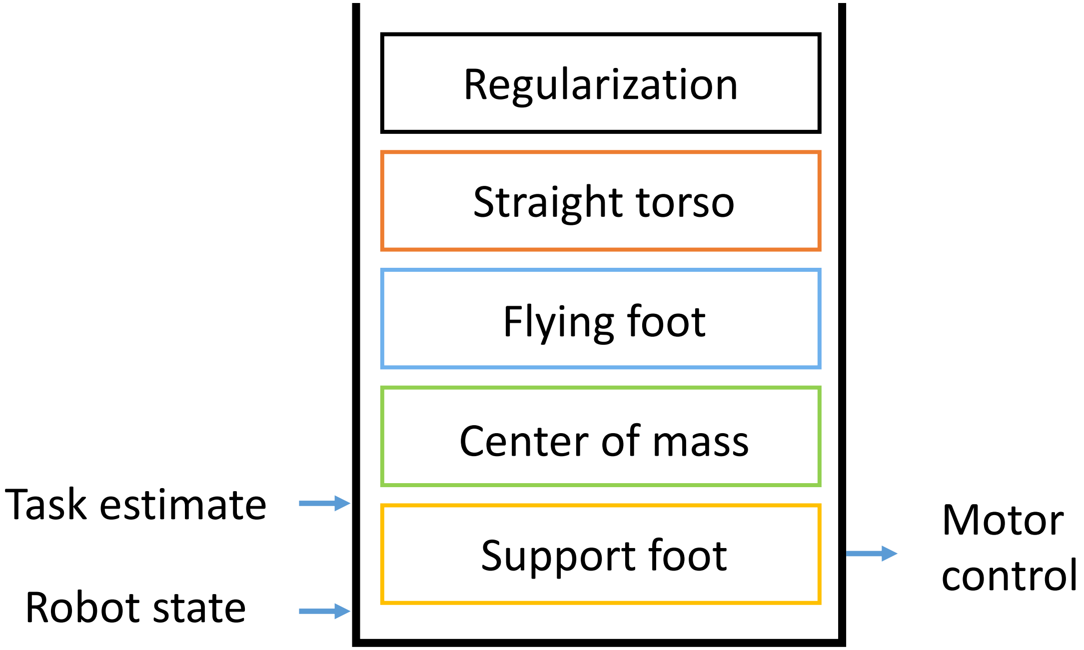
Cf Le dessous des images "Atlas, le robot star de Boston Dynamics"

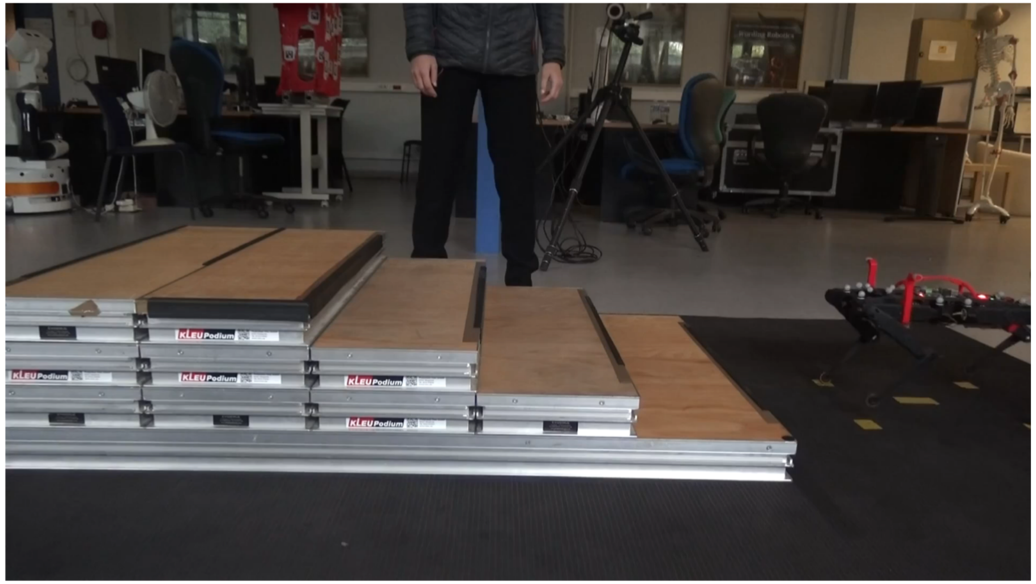

Tenner Chrüz
(2020m ASL)
Tenner Alp
(1938m ASL)
Climbing stairs as if they were flat.
NAVER LABS
XPENG ROBOTICS

Good old control heuristics

… motion programed *by tasks*







Whole-body control decision

High-level function objective

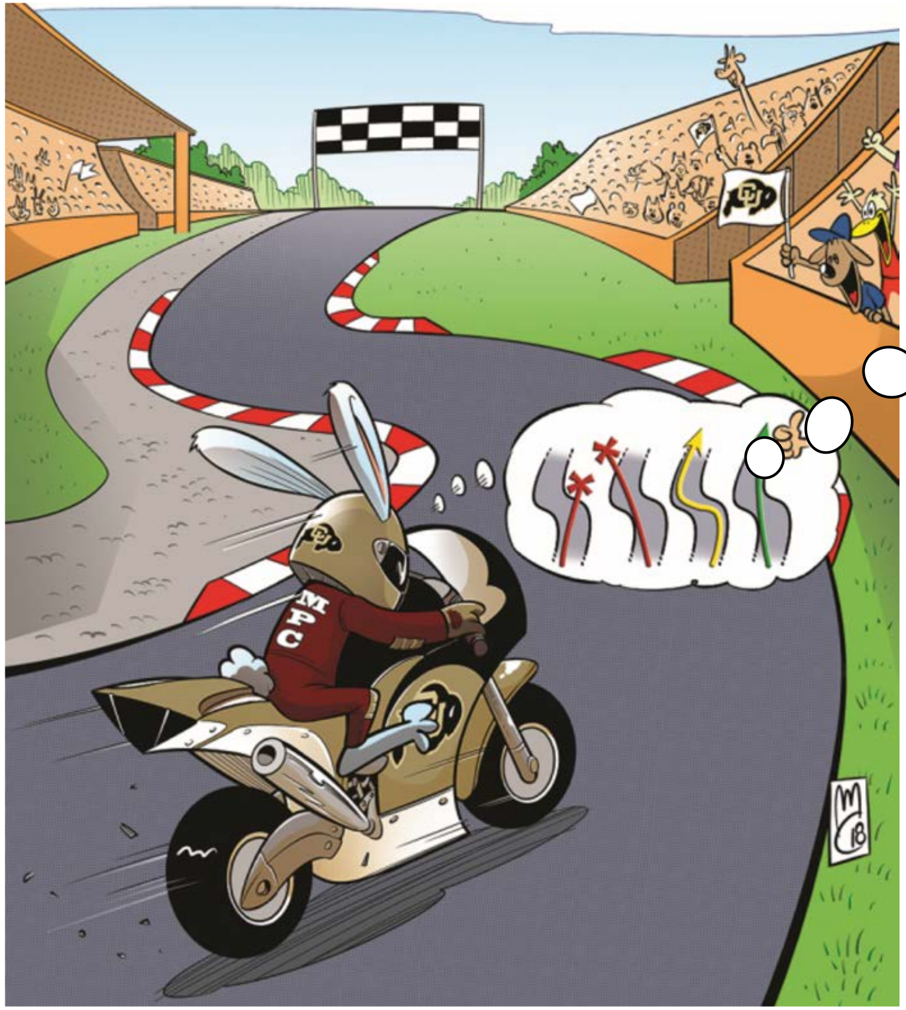
Go that way, bring me that object, open that door

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)

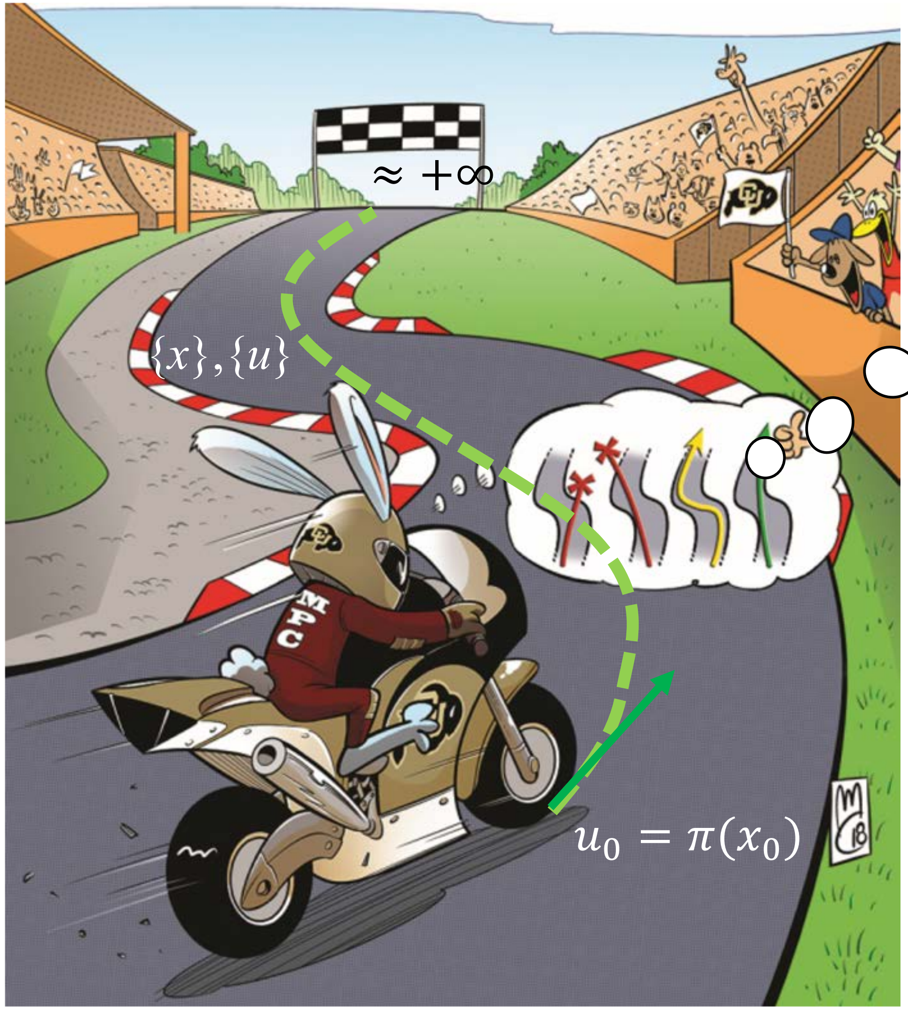
Decide: future robot trajectory

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 …

- 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**

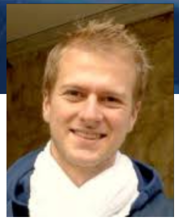
Crocoddyl
https://github.com/loco-3d/crocoddyl

BSD
BSD-2 License

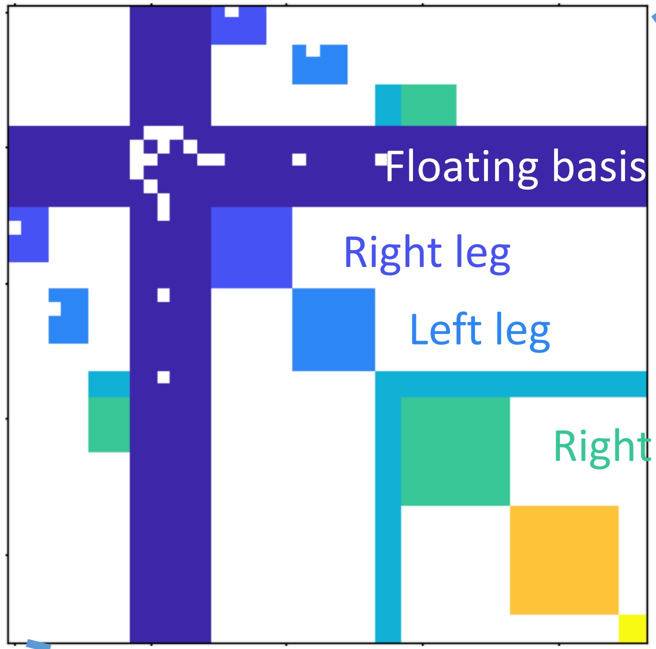
Artificial & Natural Movements

ANITI
Université de Toulouse

… for efficient problems



Justin Carpentier
Inria Paris / PR[AI]RIE

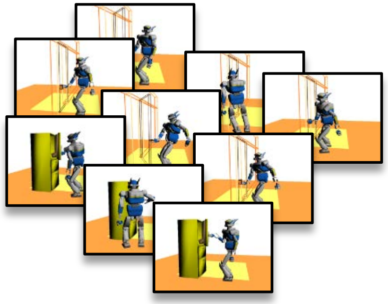


$$\begin{bmatrix}
L_{xx} & & & & L_{xu} & & -I & F_x^T & \\
& \ddots & & & \ddots & & \ddots & \ddots & \\
& & L_{xx} & & & L_{xu} & & -I & F_x^T \\
& & & L_{xx} & & & & & -I \\
L_{ux} & & & & L_{uu} & & F_u^T & & \\
& \ddots & & & \ddots & & & \ddots & \\
& & L_{ux} & & & L_{uu} & & & F_u^T \\
-I & & & & & & & & \\
\boxed{F_x} & -I & & & F_u & & & & \\
& \ddots & \ddots & & \ddots & & & & \\
& & F_x & -I & & F_u & & &
\end{bmatrix}$$

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

Memory of motion

Off-line
Optimization and encoding

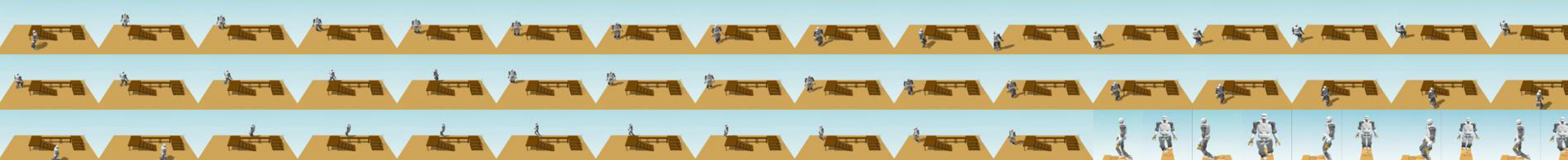


Off-line
Exploration by optimization
Globalization

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l(x_t, u_t)dt$$
$$\text{s.t.} \quad x(0) = \hat{x},$$
$$\dot{x}(t) = f\big(x(t), u(t)\big), \forall \mathrm{t=0..T}$$

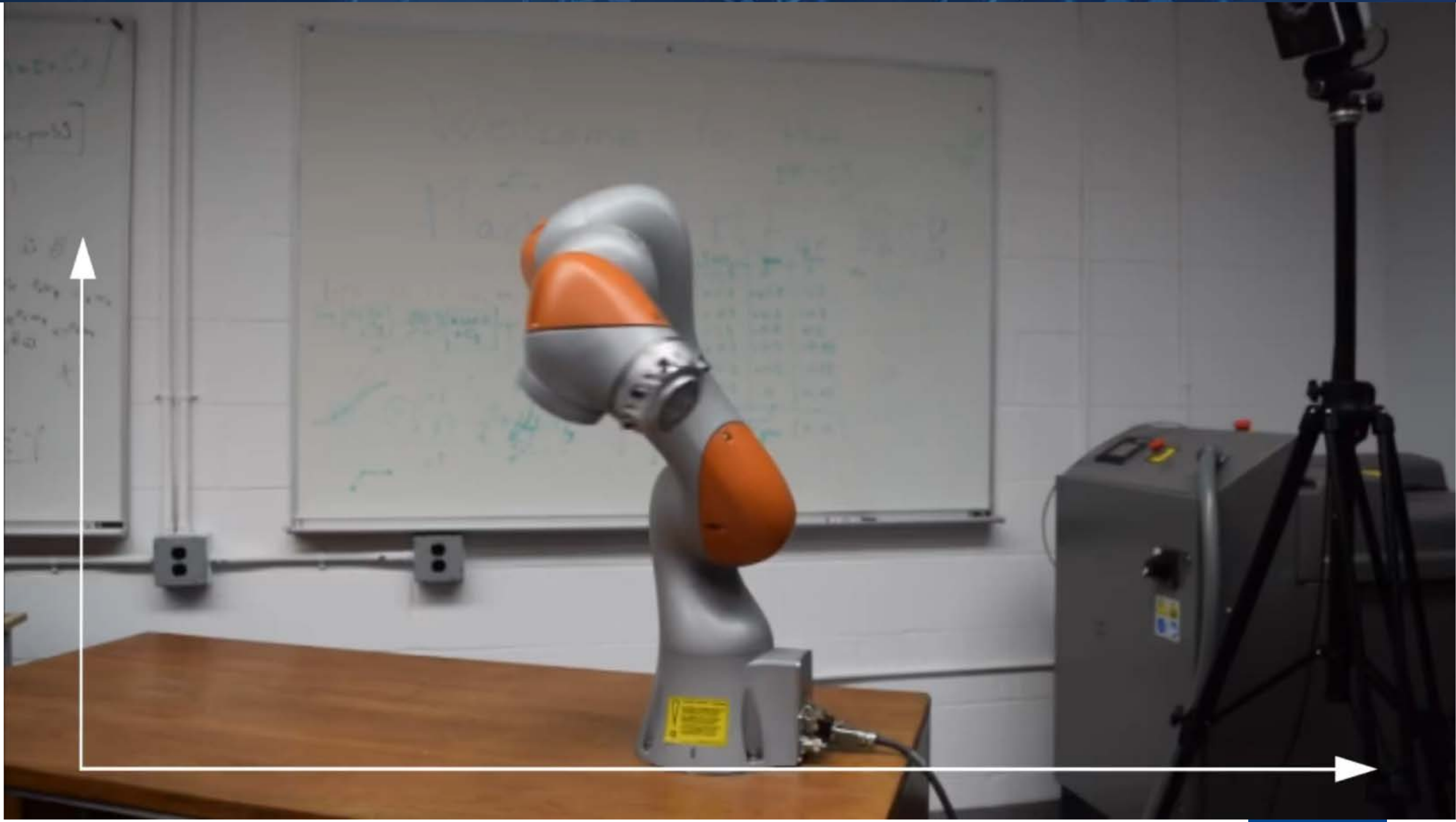
On-line
Whole-body predictive control
High-freq by local convergence

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



Efficient solver … for efficient problems

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



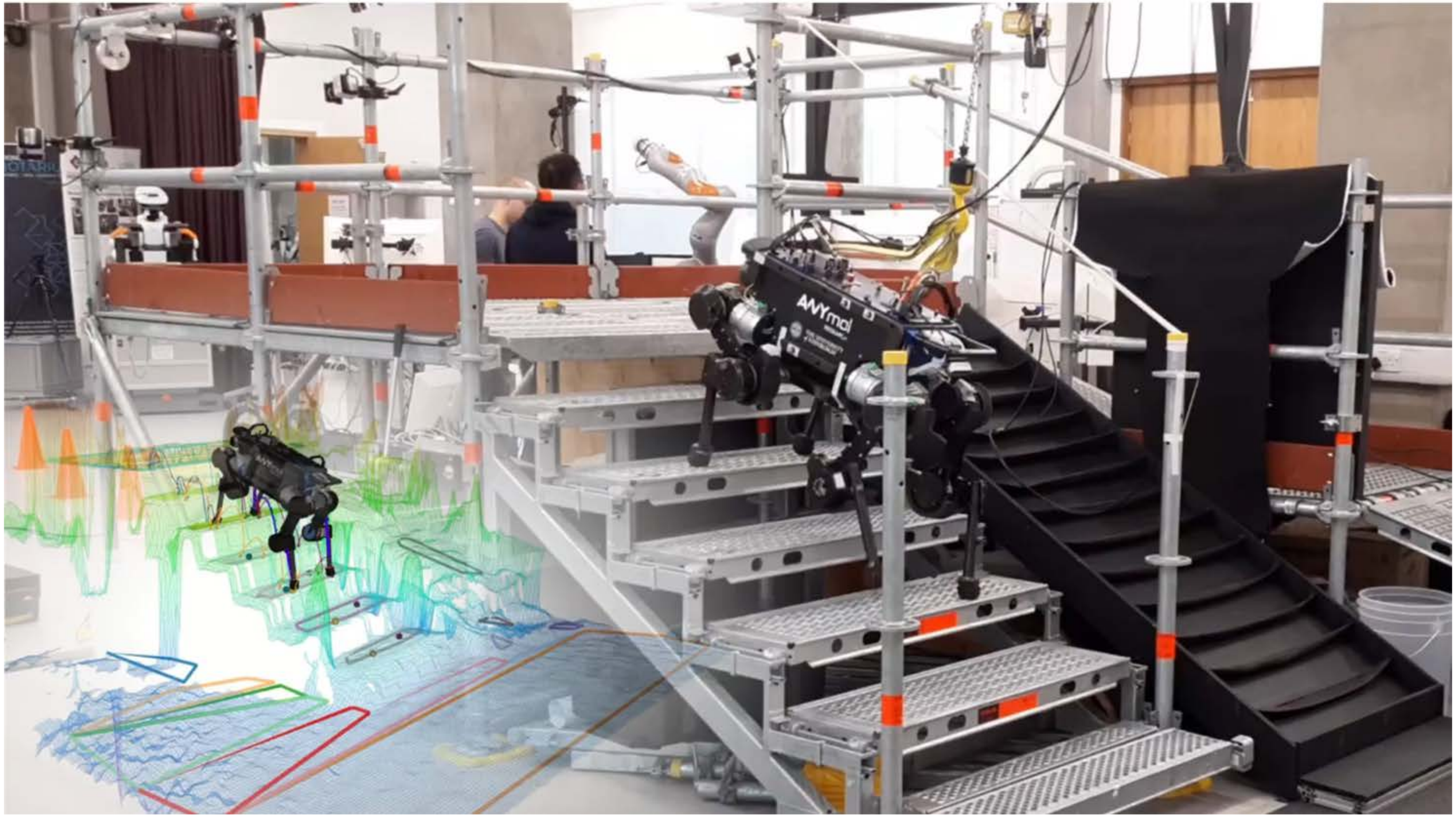
Ludovic Righetti

NYU

Efficient solver … for efficient problems

Ewen Dantec

Efficient solver … for efficient problems




Thomas Corberes


Steve Tonneau

ORI
OXFORD ROBOTICS INSTITUTE

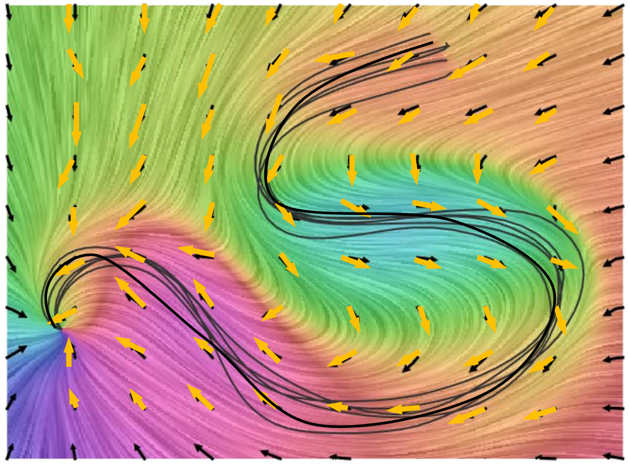
THE UNIVERSITY OF EDINBURGH

ANITI

Université de Toulouse

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

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



Trajectory optimization

$U: t \longrightarrow u(t)$

Motion planning

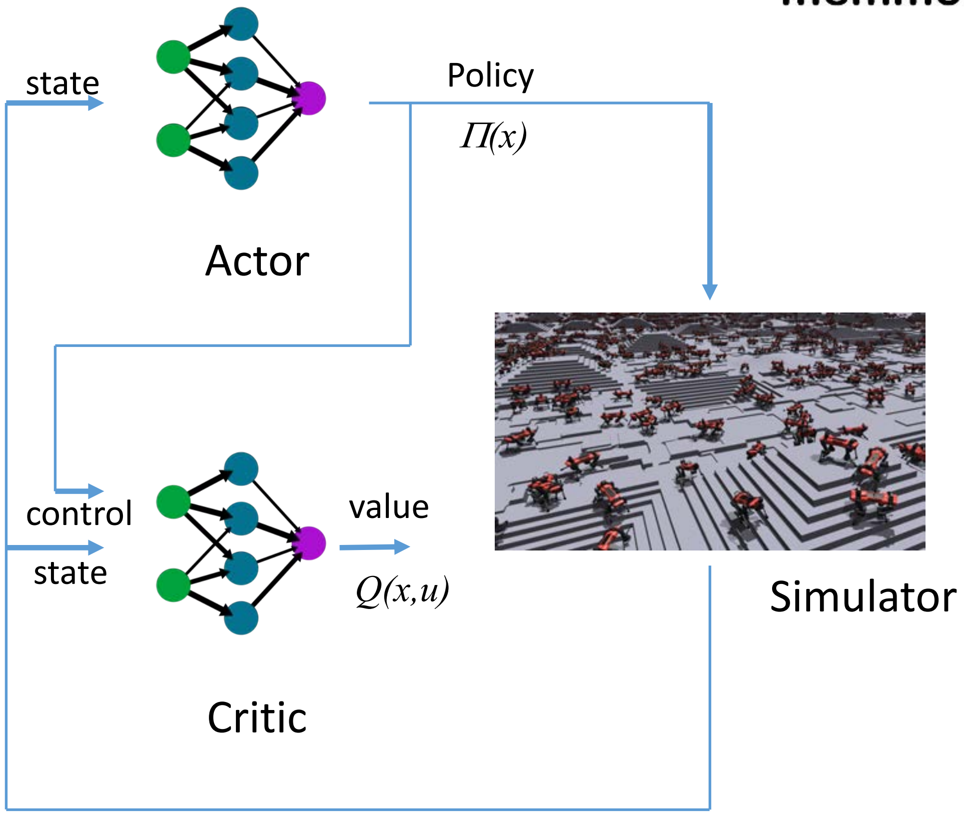
Policy optimization

$\Pi: x \longrightarrow u = \Pi(x)$

Reinforcement learning

Memory of motion

memmo



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

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

Belman principle

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

$$Q(x,u) = l(x,u) + Q(x', \Pi(x))$$

Pierre-Alexandre Leziart **Thomas Flayols**

Artificial & Natural Movements

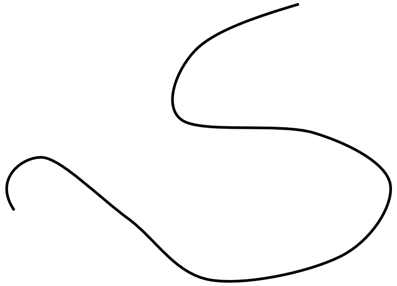
ANITI

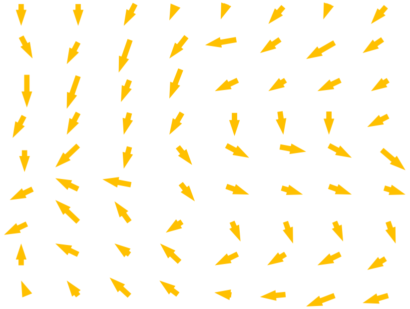
Université de Toulouse

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

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

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





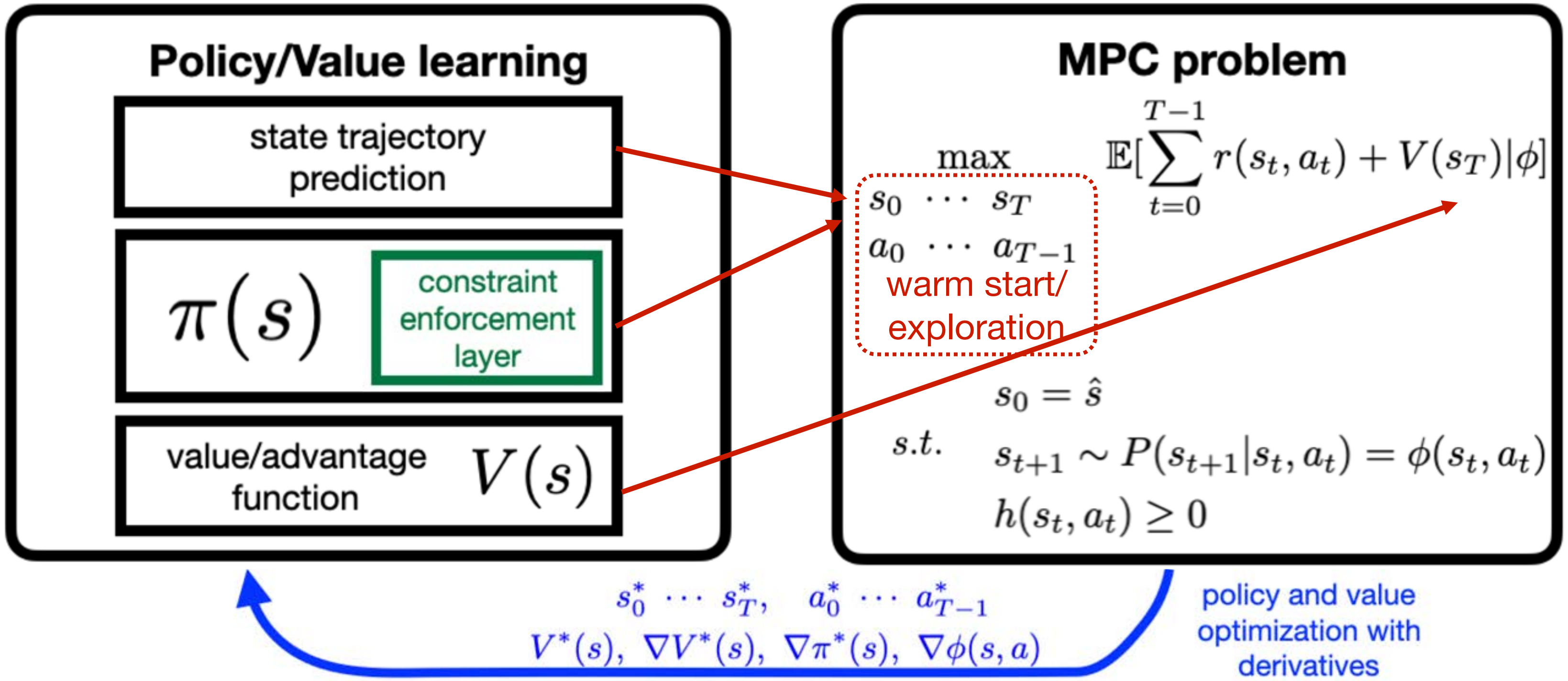
- ✓ trajectory optimization
- ✓ super-linear convergence
- ✓ real-time computation
- ✓ constraint satisfaction
- – local minima (no global policy)
- – difficulty with discontinuous dynamics
- – no inclusion of multi-modal sensing

- ✓ (global) policy optimization
- ✓ handles discontinuities
- ✓ multi-modal sensing inclusion
- – no guaranteed convergence
- – little use of model information
- – difficult transfer to robots
- – no constraint satisfaction

Toward RL solvers with mathematical guarantees

Ludovic Righetti

NYU



Objective 1
Policy optimization with derivatives

1. policy learning with derivatives
2. value function learning with derivatives
3. adding constraints in RL

Objective 2
Differentiable reinforcement learning with mixed discrete-continuous modes

1. local optimization with differentiable simulators
2. globalization using MCTS

Main messages

We should both

OPTIMISE and LEARN

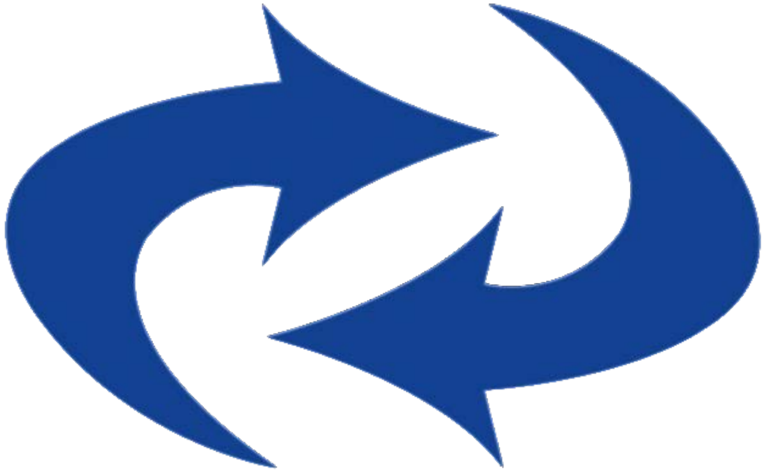
the movements of a robot !

- **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

ANITI : building hybrid intelligence

Optimization



Learning

Joint laboratories

ROB4FAM

AIRBUS

DYNAMOGRAD

TOWARD

PAL ROBOTICS

Artificial & Natural Movements

ANITI

Université de Toulouse