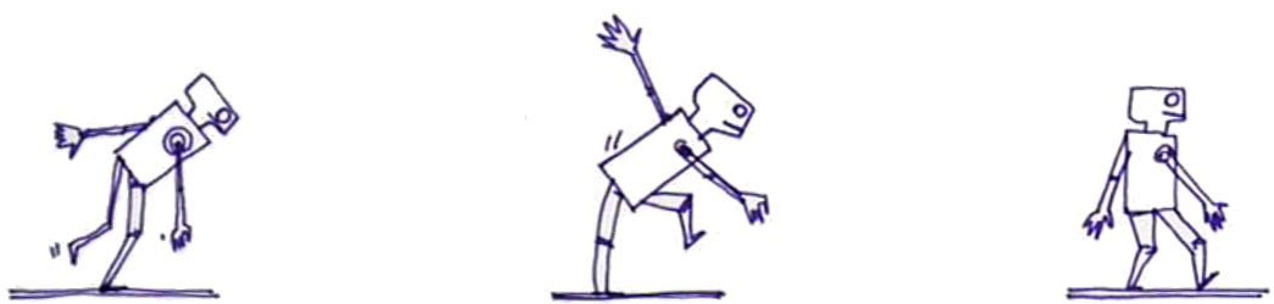
Mathematical Programing in Robotics

Nicolas Mansard



Complex movement for complex robots

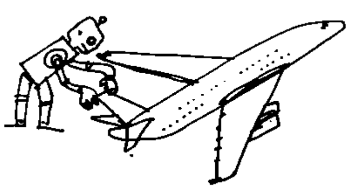
Complexity of the problem **VS** Needed computational efficiency

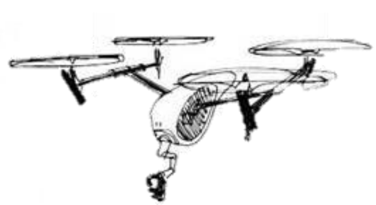
1 hour of computation

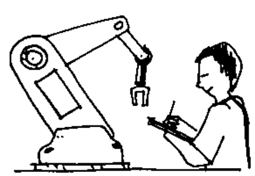
1000 Hz decisions / seconds

Number of degrees of freedom
Uncertainty
Instability of dynamics
Linearity/convexity of problems

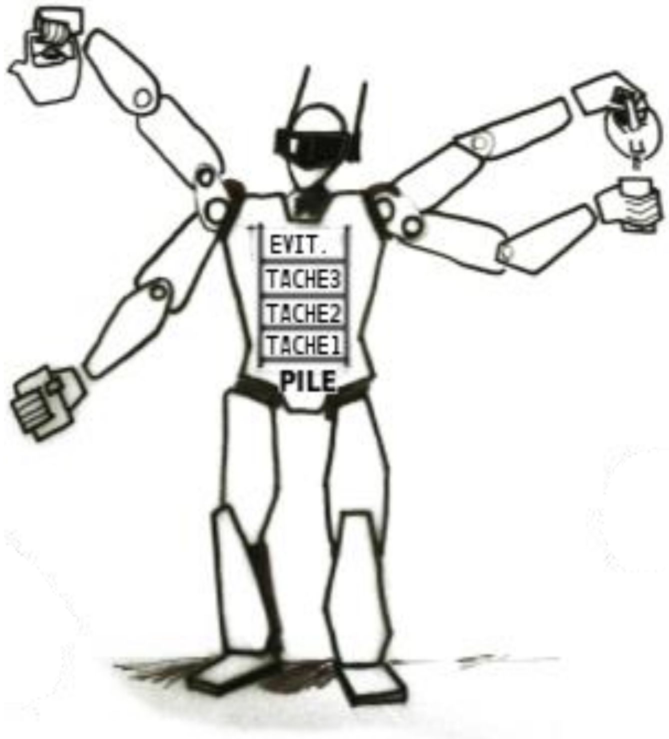
Environment dynamics
Robot speed
Bounded viability
Control of a contact interface

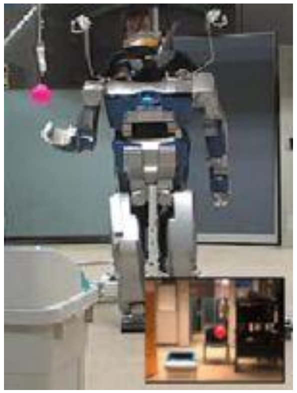




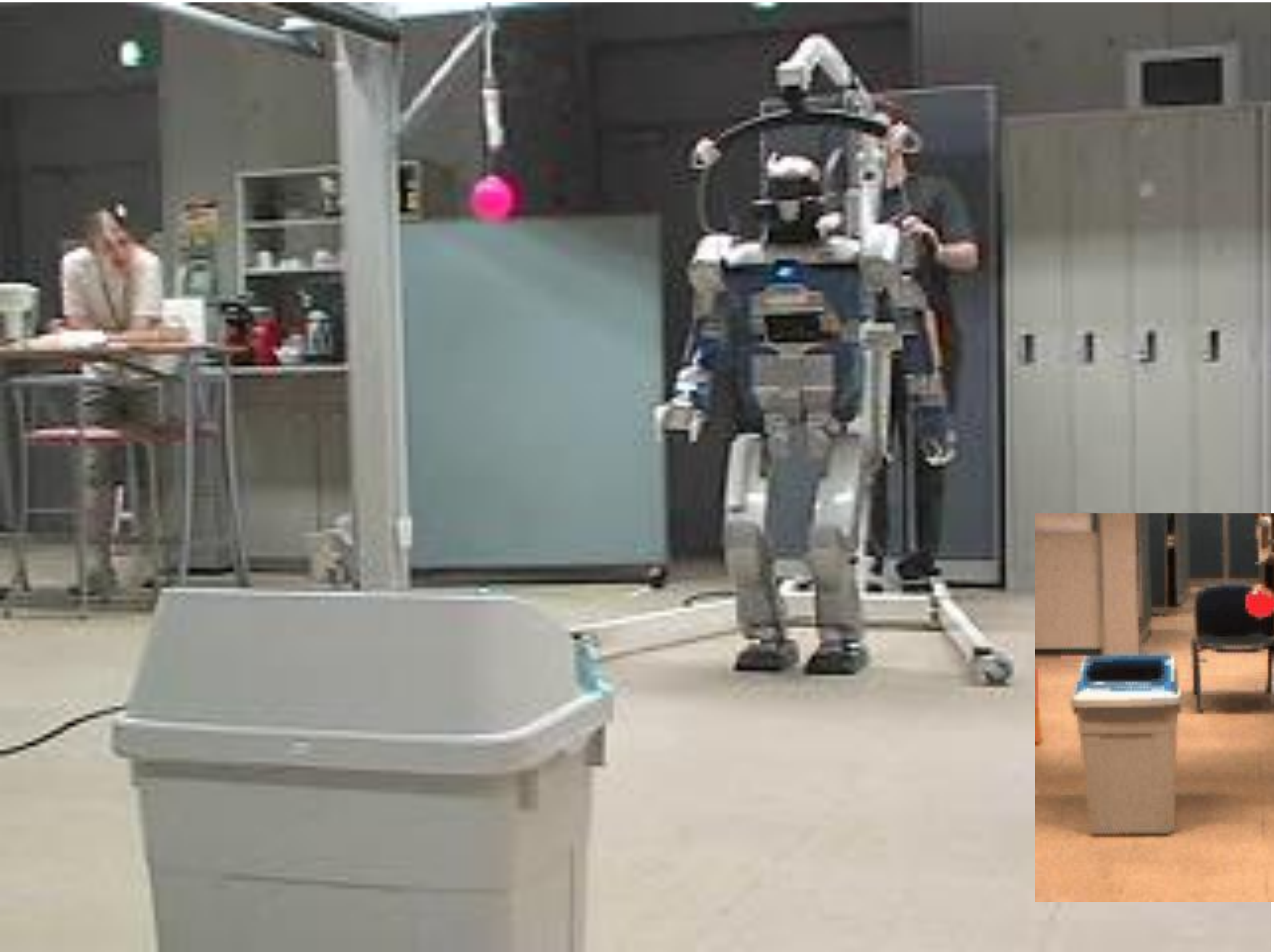


Stacking tasks



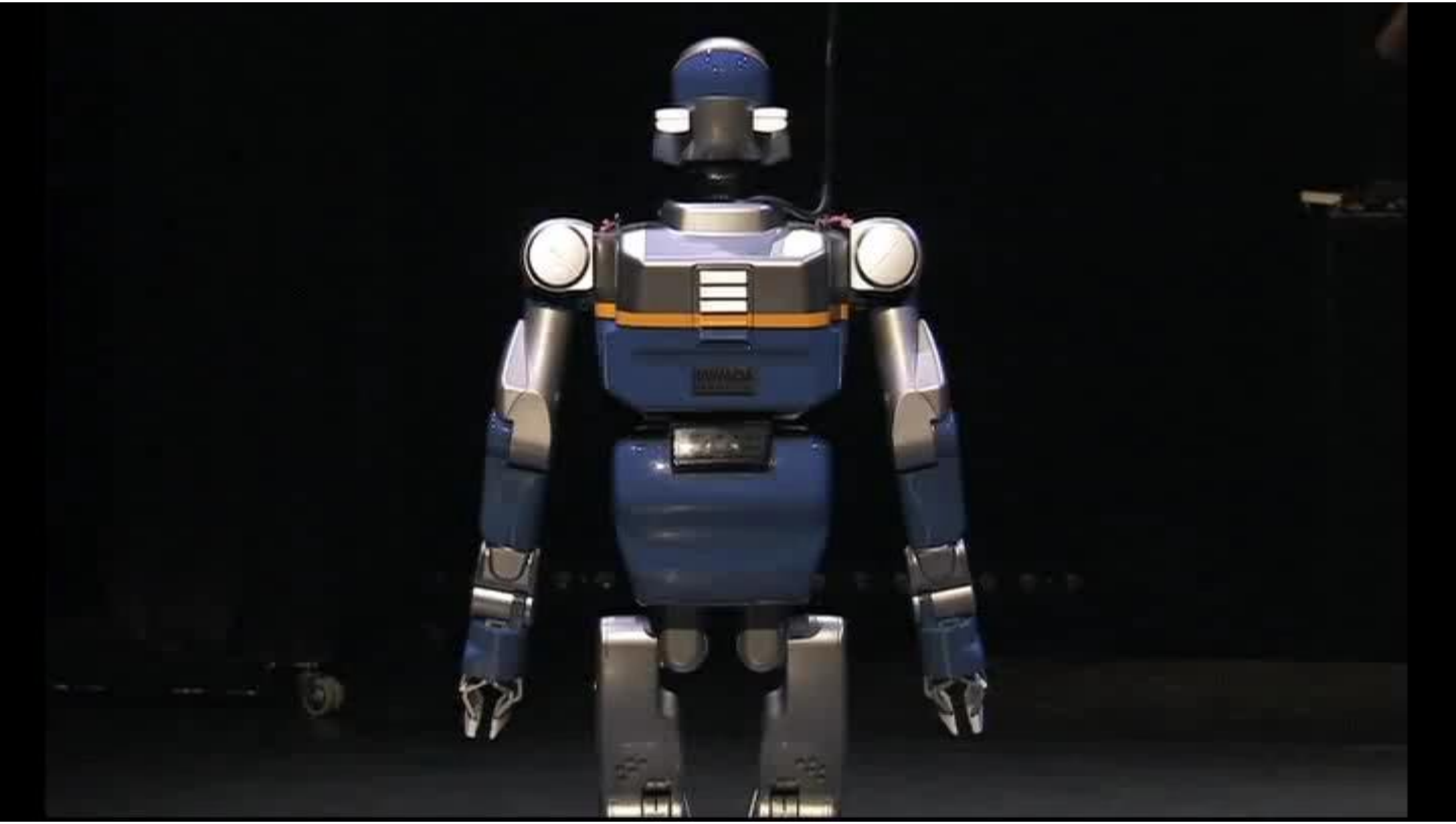


Vision control

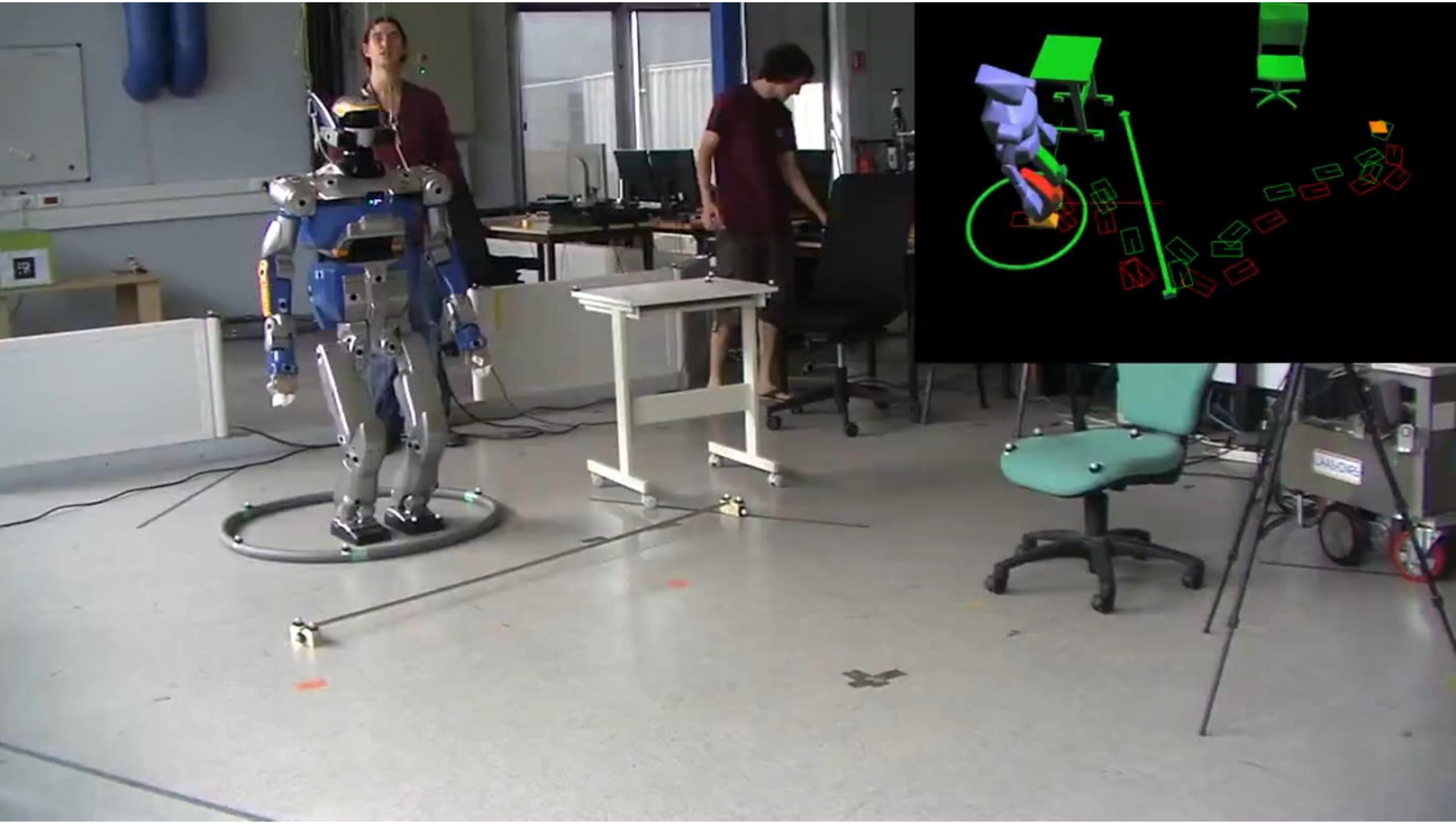


Force control (@JRL+TUM, with LAAS)

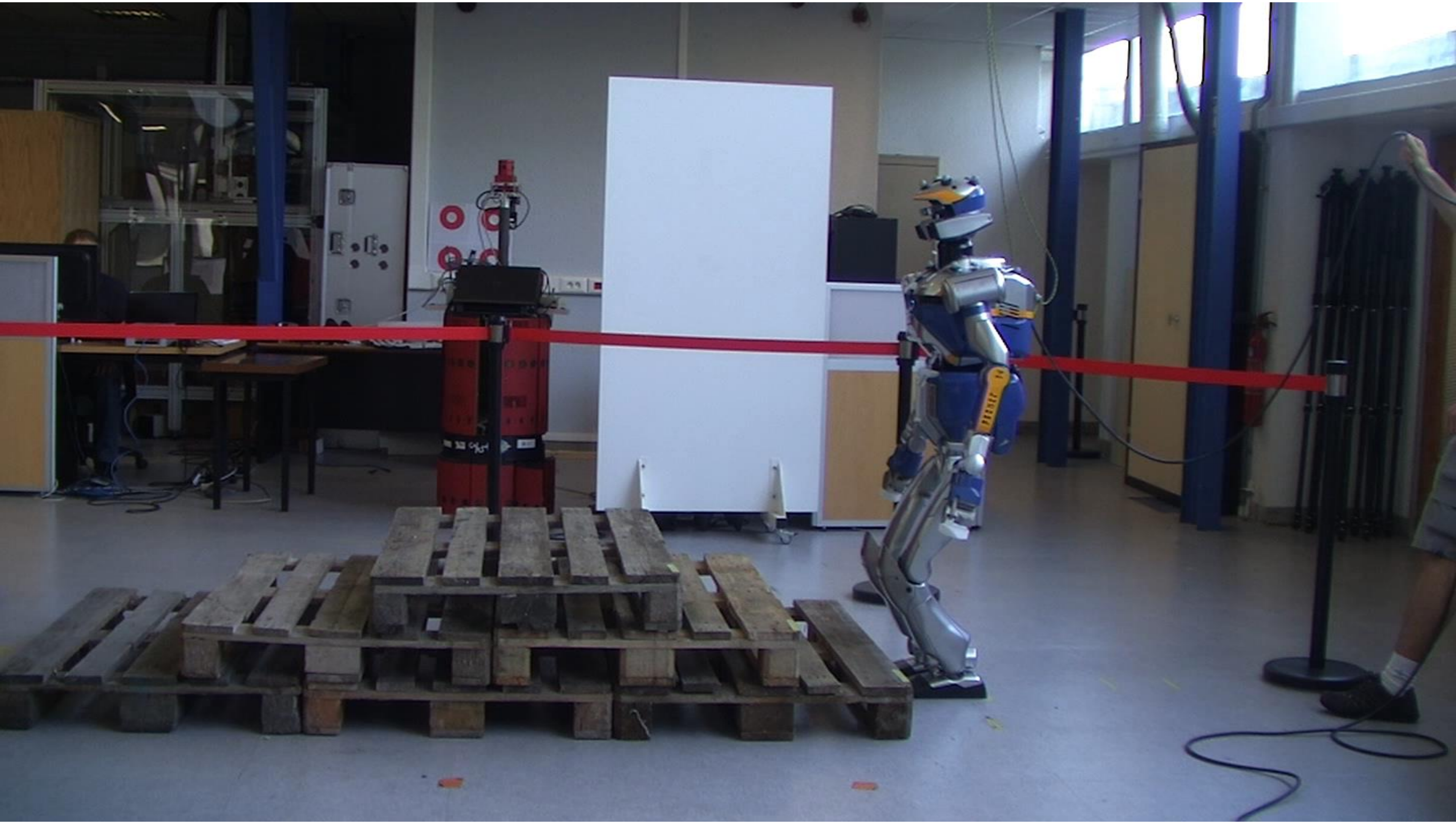
Dynamics



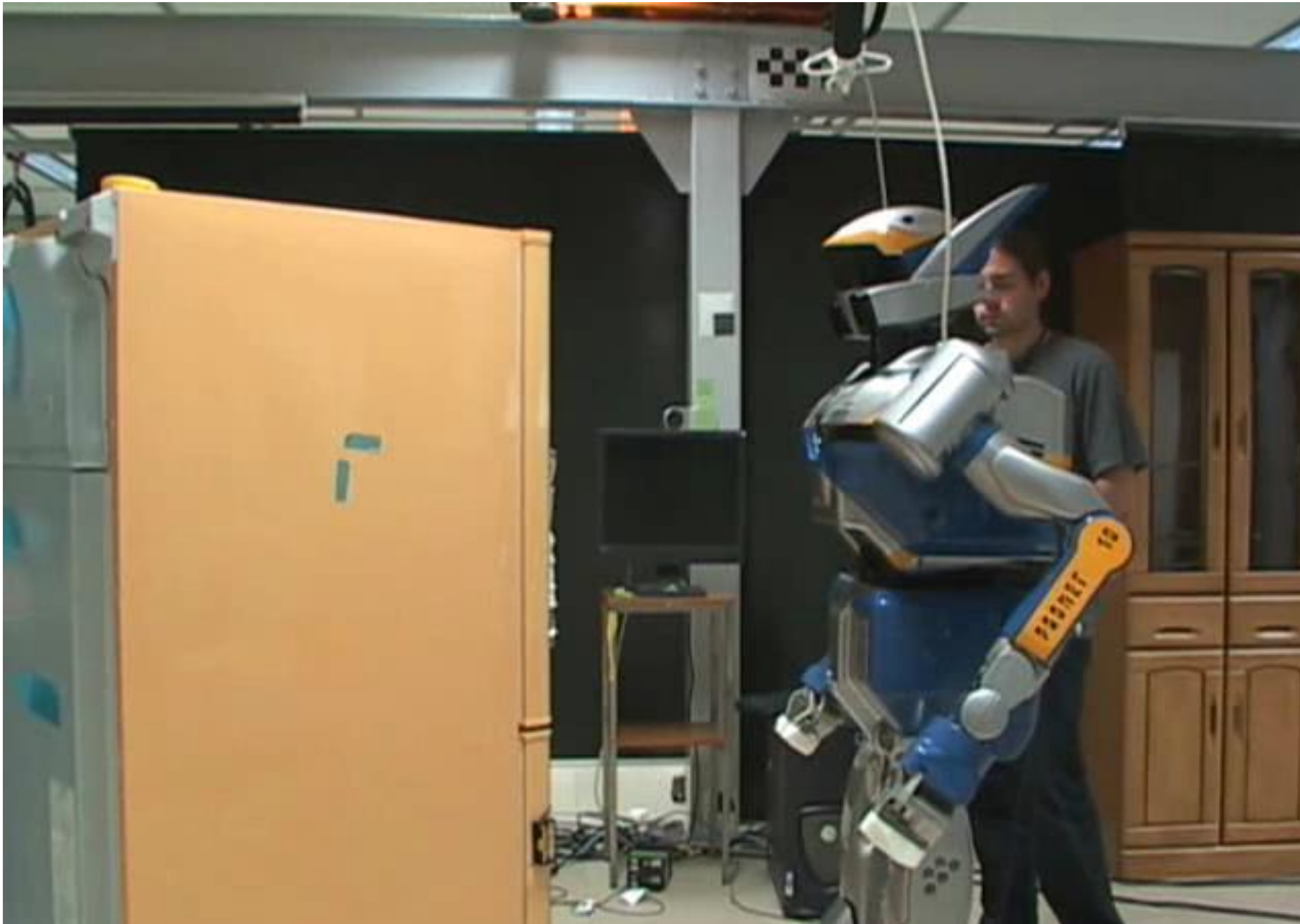
Reactive walk (@LAAS, with JRL)



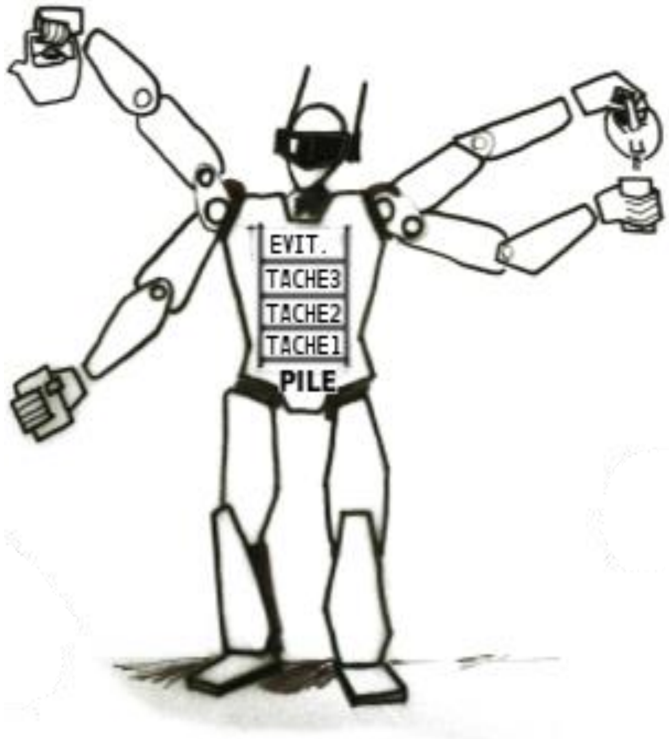
Quasi-flat walk

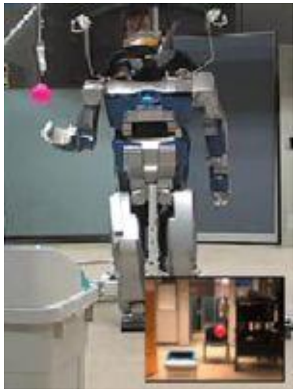


Hierarchy and sequencing



Stacking tasks





Visual control

Hiearchy

Force control

Reactive walk

Dynamics

Quasi-flat

Planning versus control

Control:
Instantaneous versus receding

Feedback
linearization

Limit behavior

planning

Work on the same object

Receding
control

Stake in optimal control

Non-linear / non-convex

Cost and constraints

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l_1(x_t, u_t)dt$$

so that $\forall t,\ \dot{x}(t) = f(x(t), u(t))$

Dynamics

Thousands of (sparse) variables

Current status of our DDP

Unconstraint sparse (ad-hoc) solver

Dedicated to locomotion

- Fixed contact sequence and timings
- Alternance with centroidal optimization

Analytic derivatives

Foot trajectory around obstacles

Target: 1 second horizon / 10 ms computation

3 seconds horizon / 7 ms computation

Current status of constrained optimization

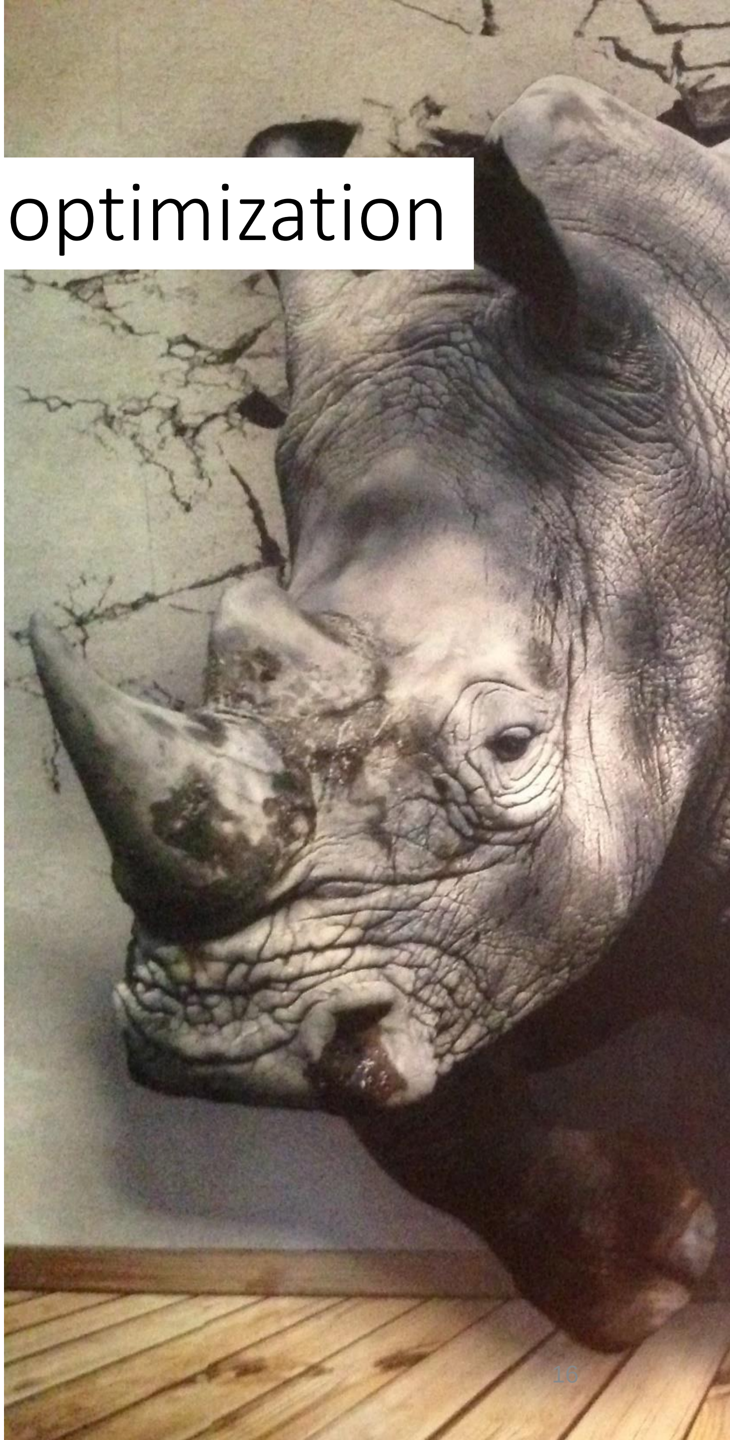
Augmented lagrangian

- Stronger than SQP
- Easier to initialize than Interior-points

Benchmark versus IPOPT and SciPy::SQP

Target: home-made sparse NLP

- with optimization on manifolds



~~Stake~~ Accomplishment in optimal control

Dimension: Thousand of variables

Dedicated DDP, release in Feb 2019

Dynamics: Fast forward simulation

Pinocchio 2.0 with derivative, released in beta

Complexity: Non-linearity and non-convexity

Constraints: physical limits and obstacles

On-going works, preliminary results


robustness
dynamics
force
contact
hierarchy
locomotion
torque
microseconds
algorithm
compute
algorithms
differentiation
derivatives
inverse
body
show
matrix
explicitly
used
using
RNEA
ABA
new
Benchmarks
consuming
algebraic
implemented
spatial
purpose
challenge
robotic
inertia
two
jointspace
efficiently
learning
firstly
major
called
kinematic
articulated
work
chain
finite
recursive
rule
applications
adequate
reinforcement
without
arm
robotics
computing
approaches
varying
additional
wellestablished
forward
17
3
namely
form
proposed
important
algebra
functions
7dof
closedform
Evaluating
outperforming
model
far
art
also
costs
state
thanks
C++
differences
formulations
whether
Rigid
humanoid
framework
introduce
Pinocchio
properties
NewtonEuler
function
computational
automatic
estimation
time
dynamic
paper
derivative
propose
frame
opensource
various
robot
control
composition
marginal
However
optimal
codesign
alternative
differentiate
36dof
cost
analytic
remains
expose

What is missing?

$$\min_{X,U} \int_0^T l(x_t, u_t)\, dt$$

$$s.t.\ \dot{x}_t = f(x_t, u_t)$$

optimal control

=

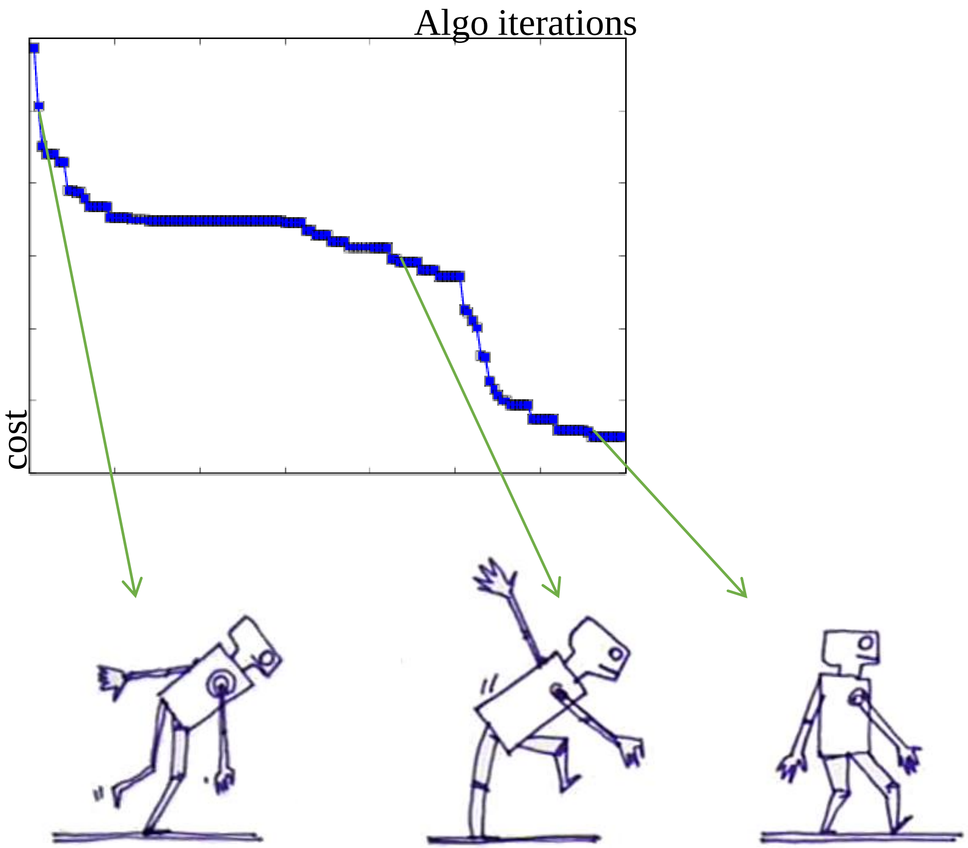
planning

+

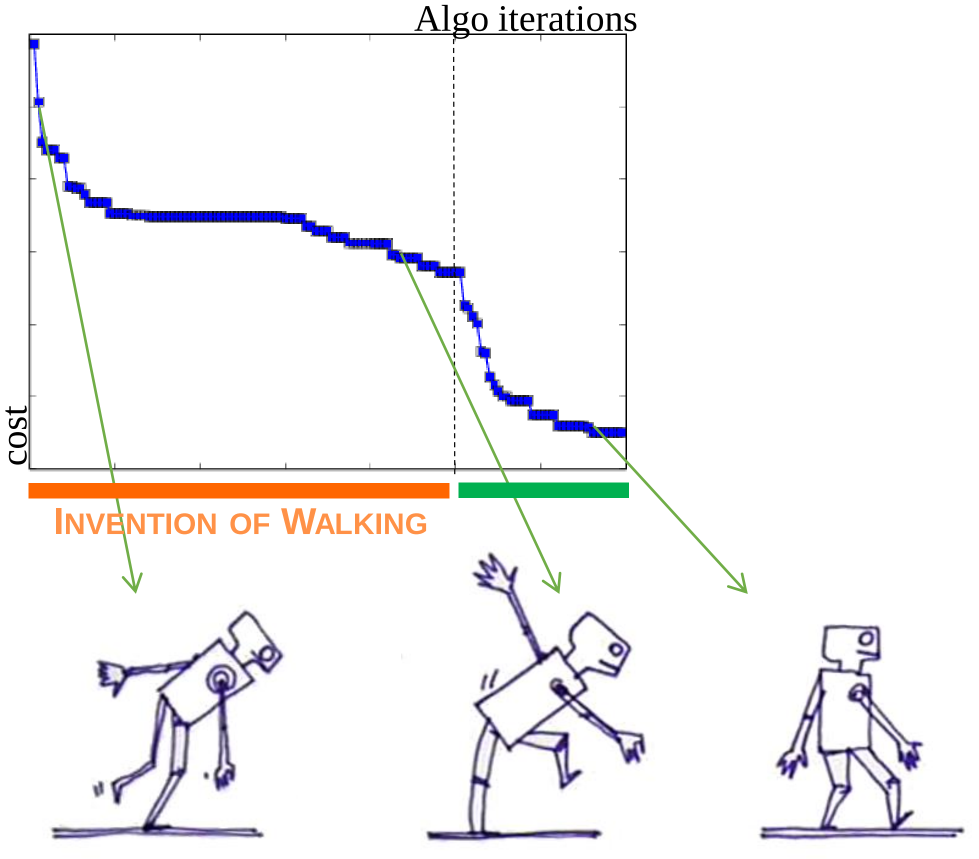
control

… but it needs guidance

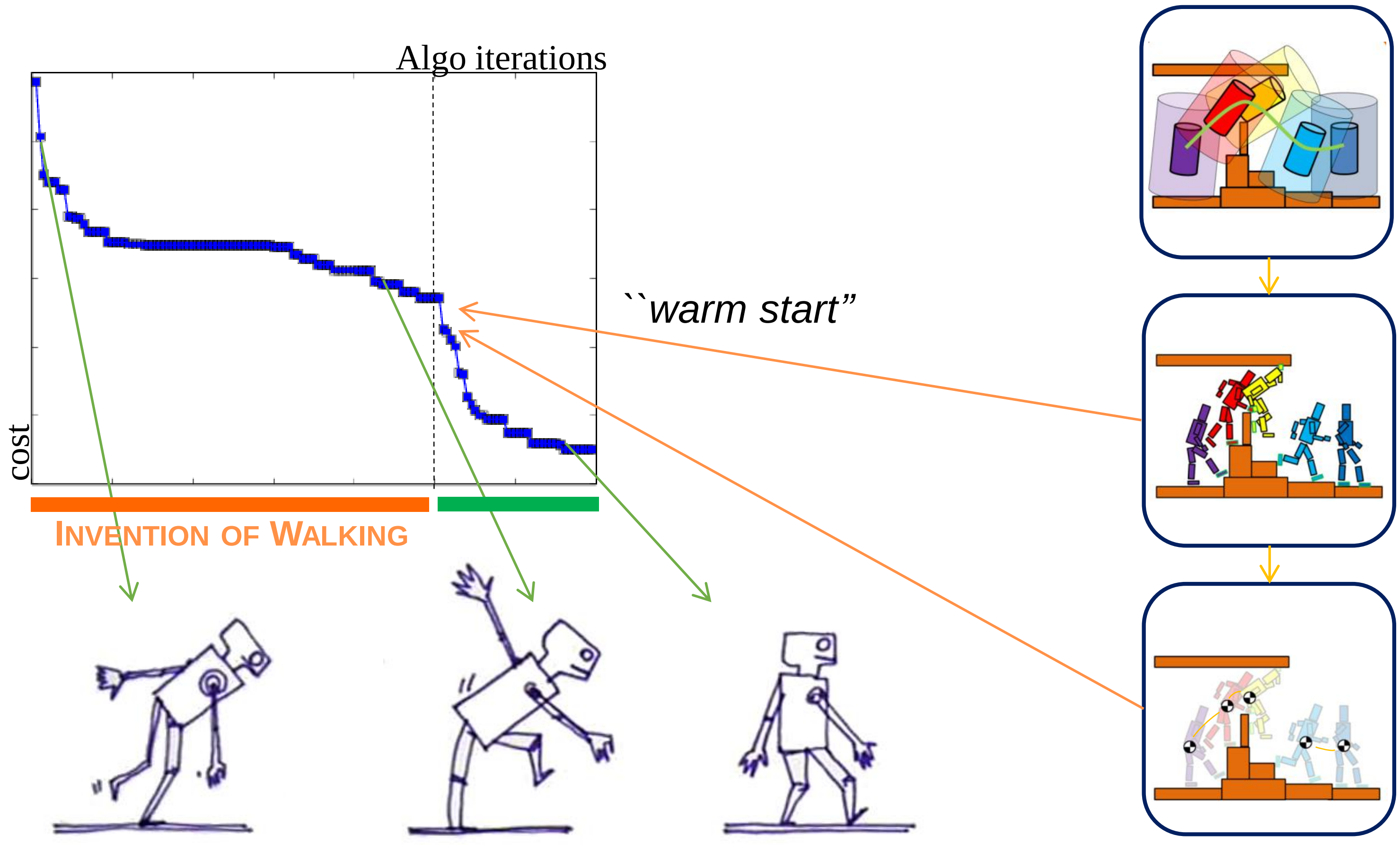
Example of locomotion



Example of locomotion

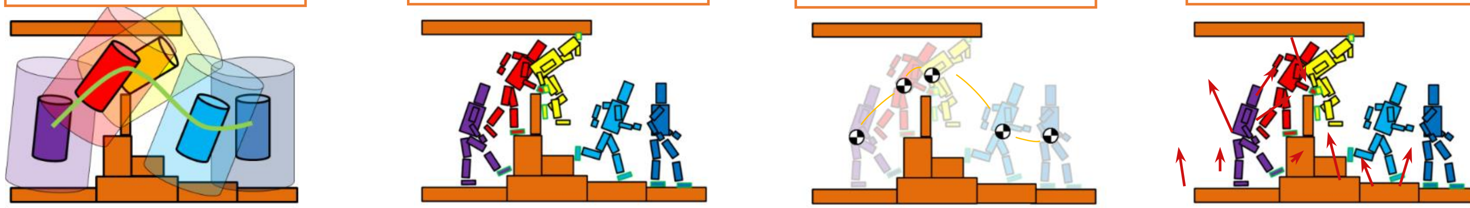


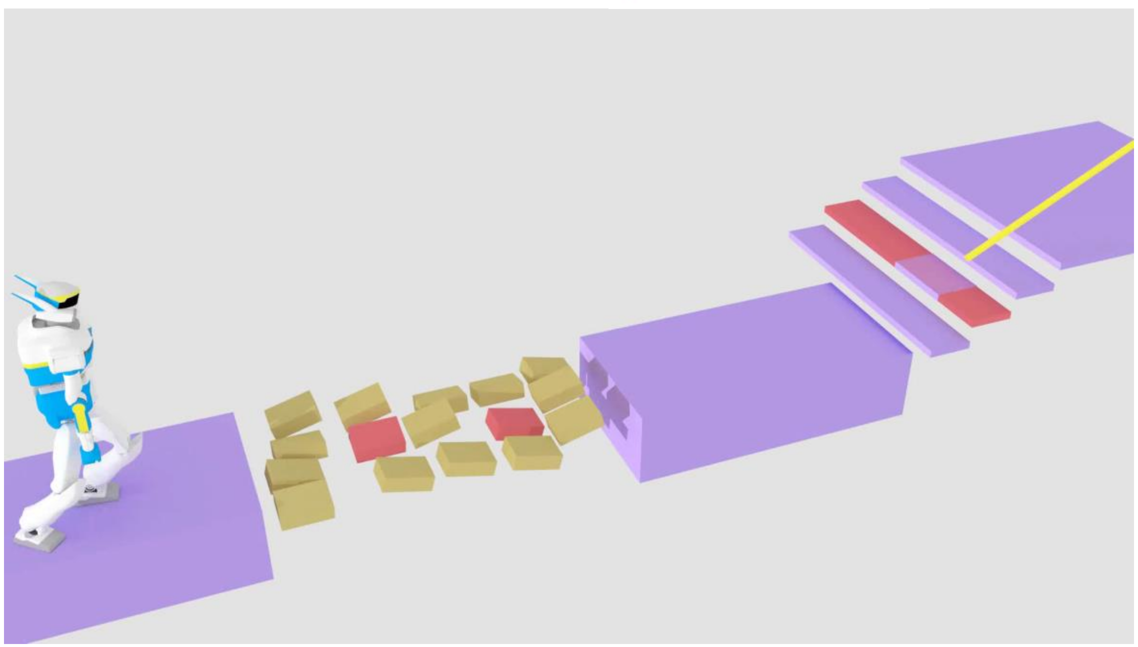
Example of locomotion



Model-based warm start

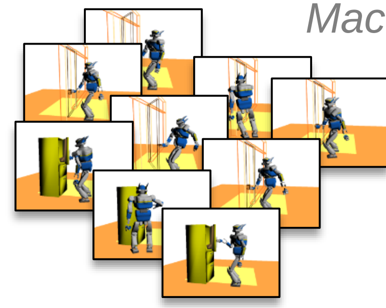
Reachability planner	Sequence of contacts	3D Pattern generator	Whole-body force control





Memory of Motion

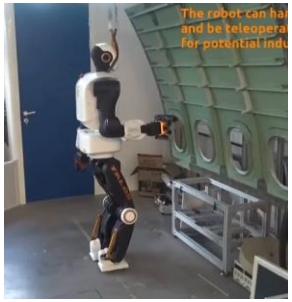
Objective #2. Memory encoding
Machine learning



Objective #3.
Exploration
Offline resolution

$$\min_{X,U} \int_0^T l\big(x(t),u(t)\big)dt$$
$$s.t.\ \ \forall t\ \dot{x}(t) = f\big(x(t),u(t)\big)$$

Objective #1. Data-driven
model predictive control
Targeting real platforms



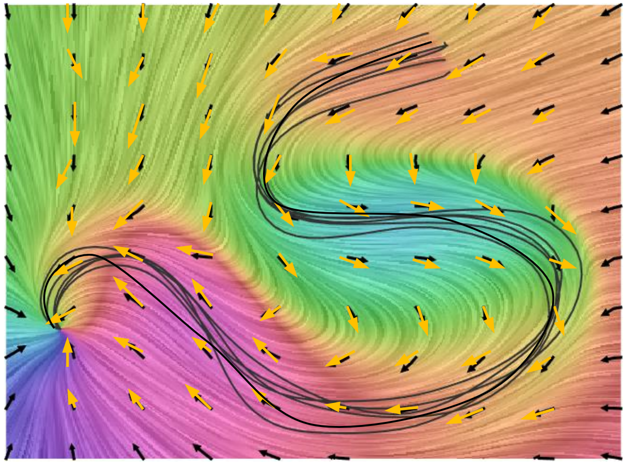
CS #1. Humanoid in factory of the future

CS #2. Exoskeleton for disabled people

CS #3. Quadruped for inspection

$$\min_{\substack{X=(Q,\dot{Q}),\\ U=\tau}} \int_0^T l_1(x_t, u_t)\,dt$$

so that $\quad \forall t,\ \dot{x}(t) = f(x(t), u(t))$



Optimizing a trajectory

U: t $\longrightarrow$ u(t)

Motion planning

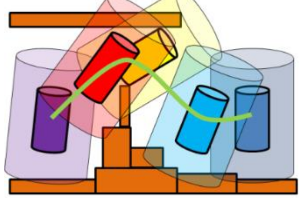
Optimizing a policy

Π: x $\longrightarrow$ u = Π(x)

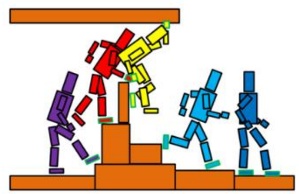
Reinforcement learning

First track: learn HPP-Loco

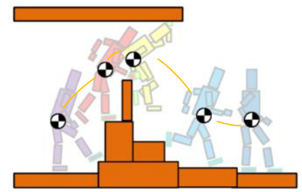
Reachability planner



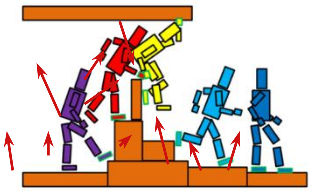
Sequence of contacts

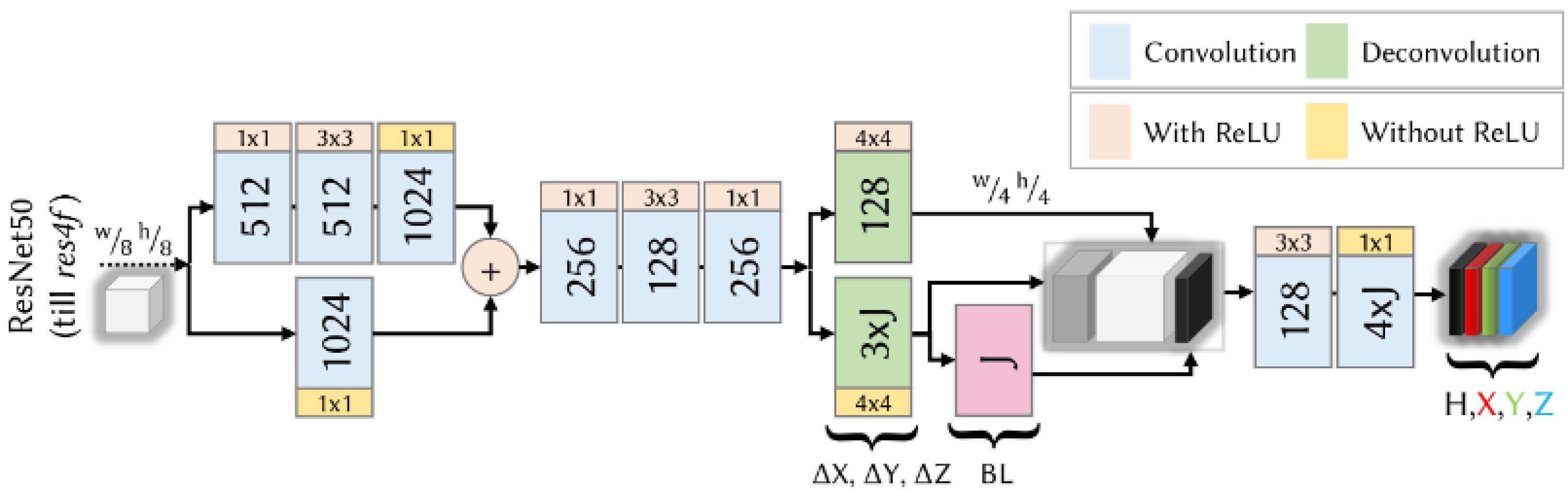


3D Pattern generator

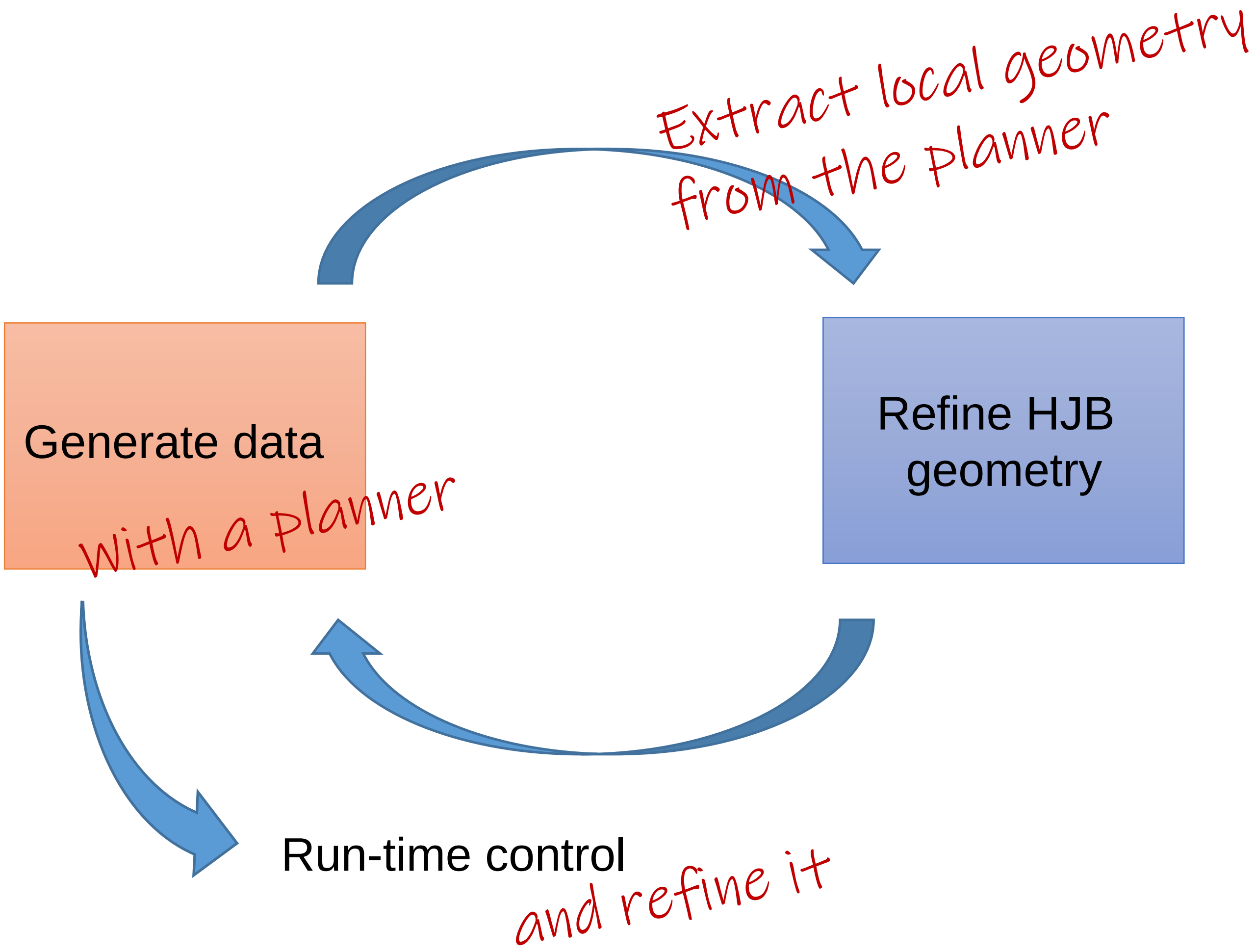


Whole-body force control

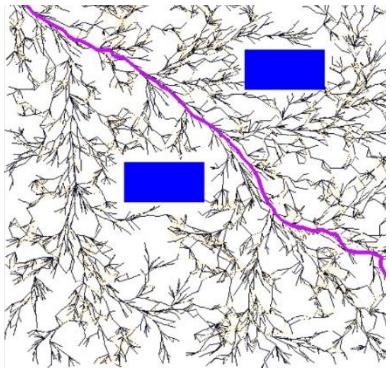




Second track: reinforcement learning

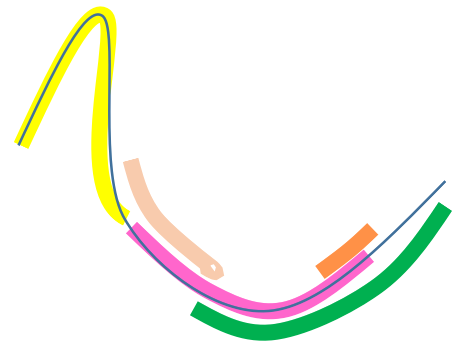


Roadmap/approximation co-training



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

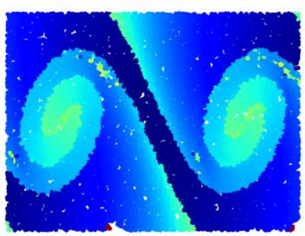
Sampling



Dataset of subtrajectories
10-100k items

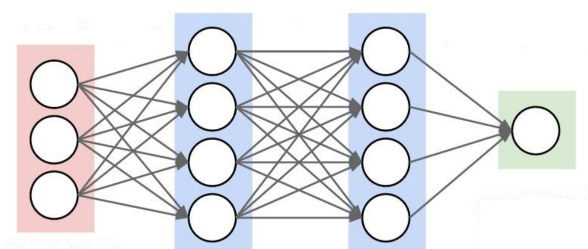
Roadmap
extension

Regression
(stoch.grad.)



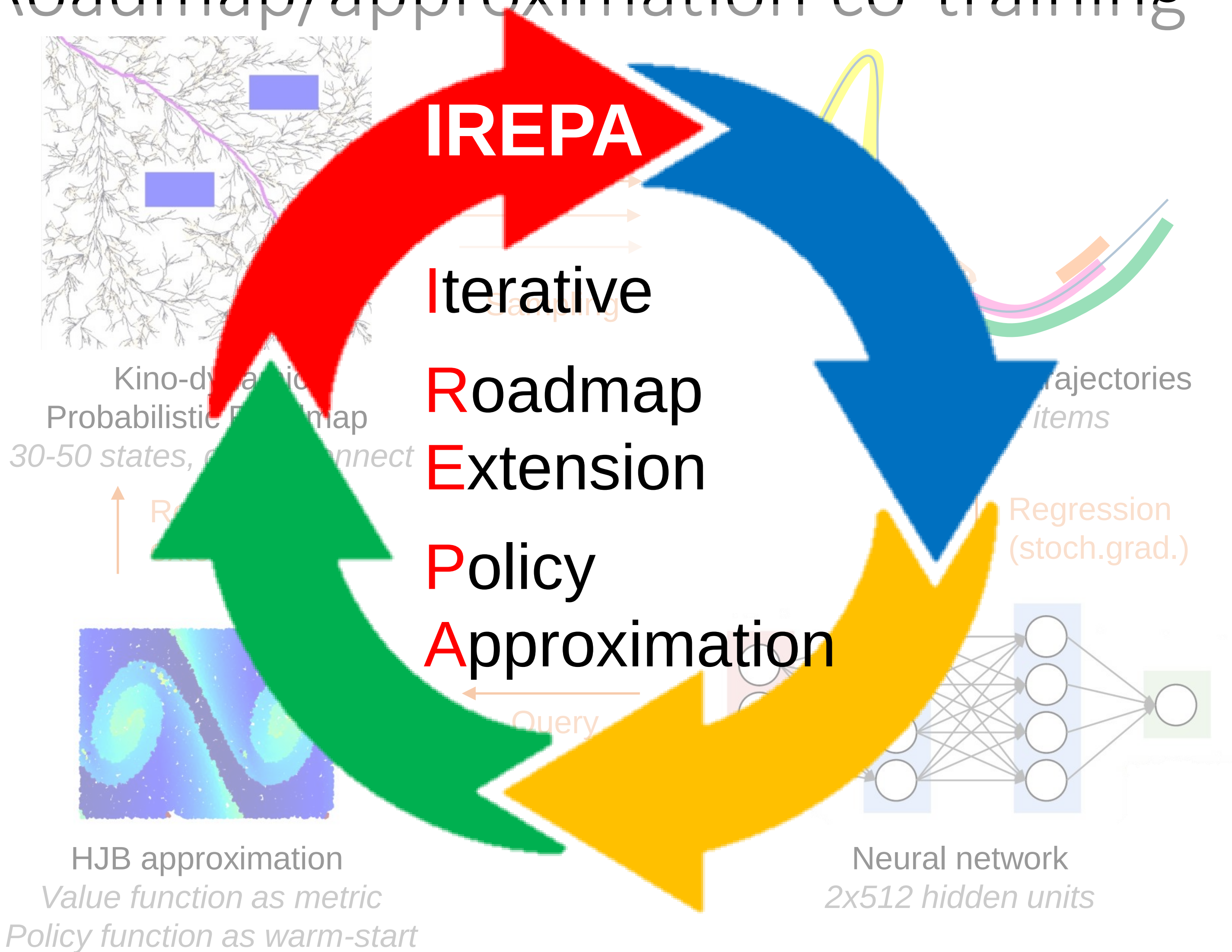
HJB approximation
Value function as metric
Policy function as warm-start

Query

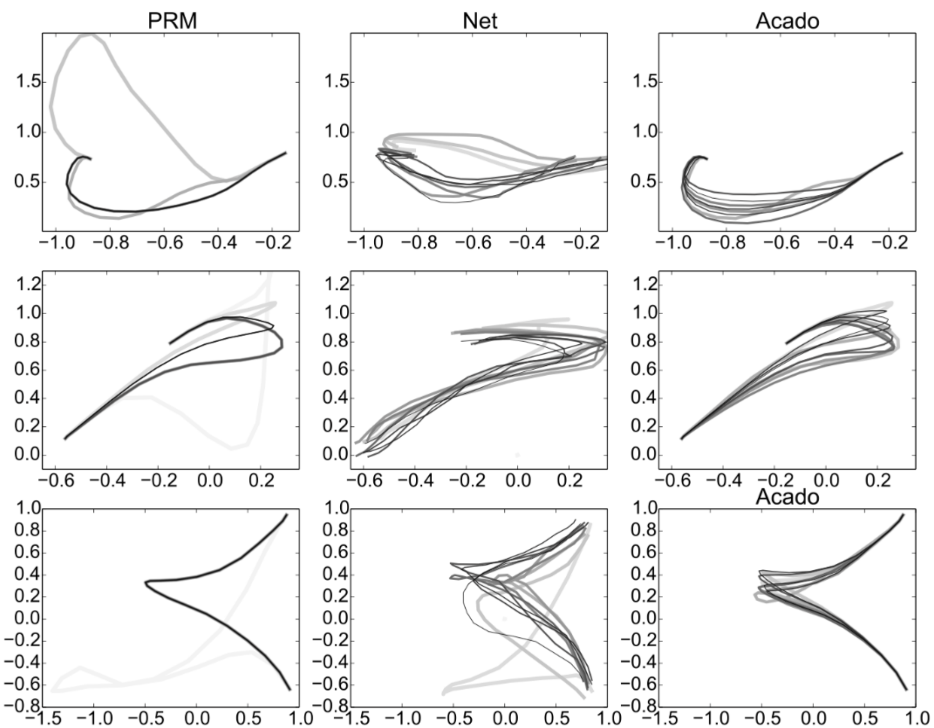


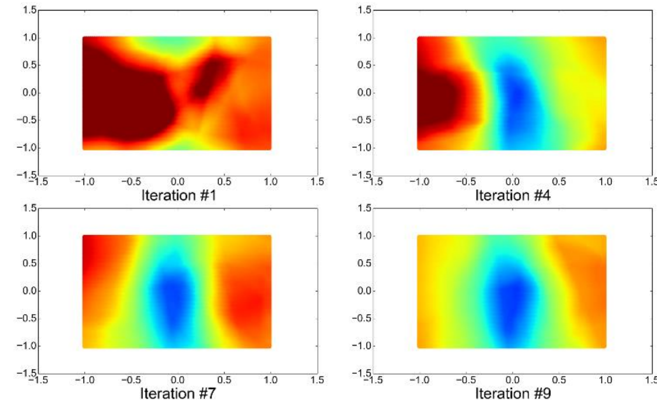
Neural network
2x512 hidden units

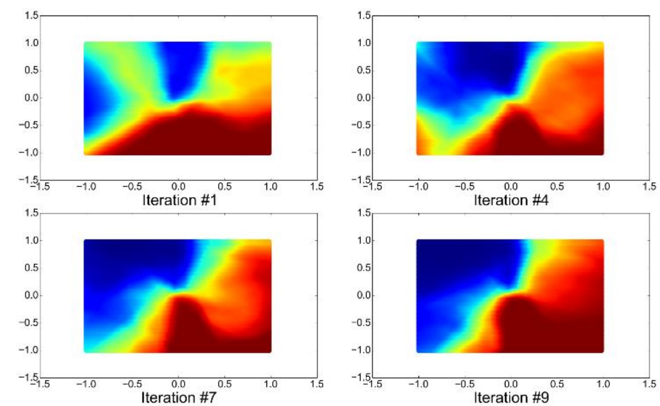
Roadmap/approximation co-training



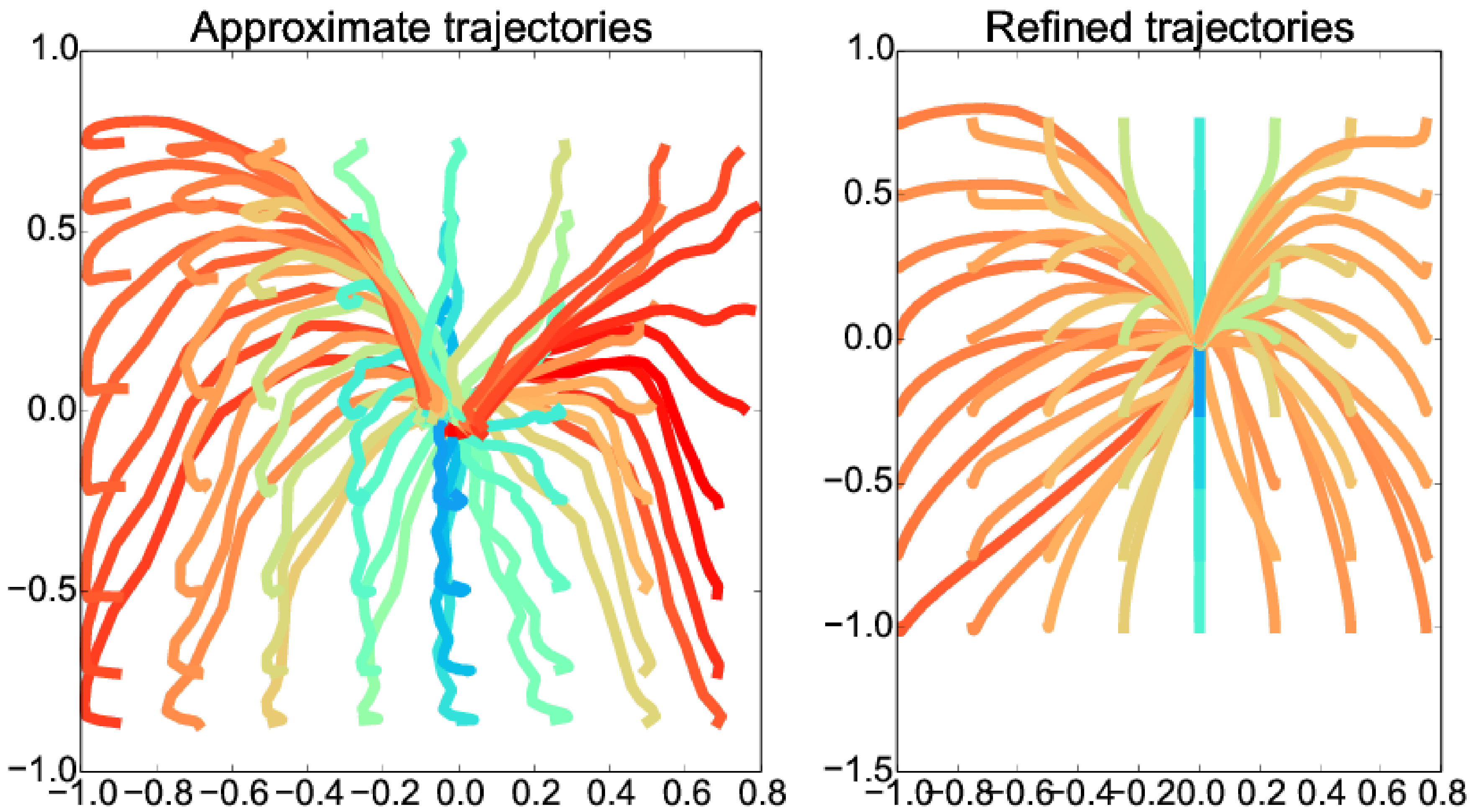
IREPA: iterations





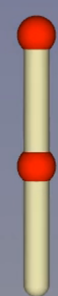


IREPA with MPC



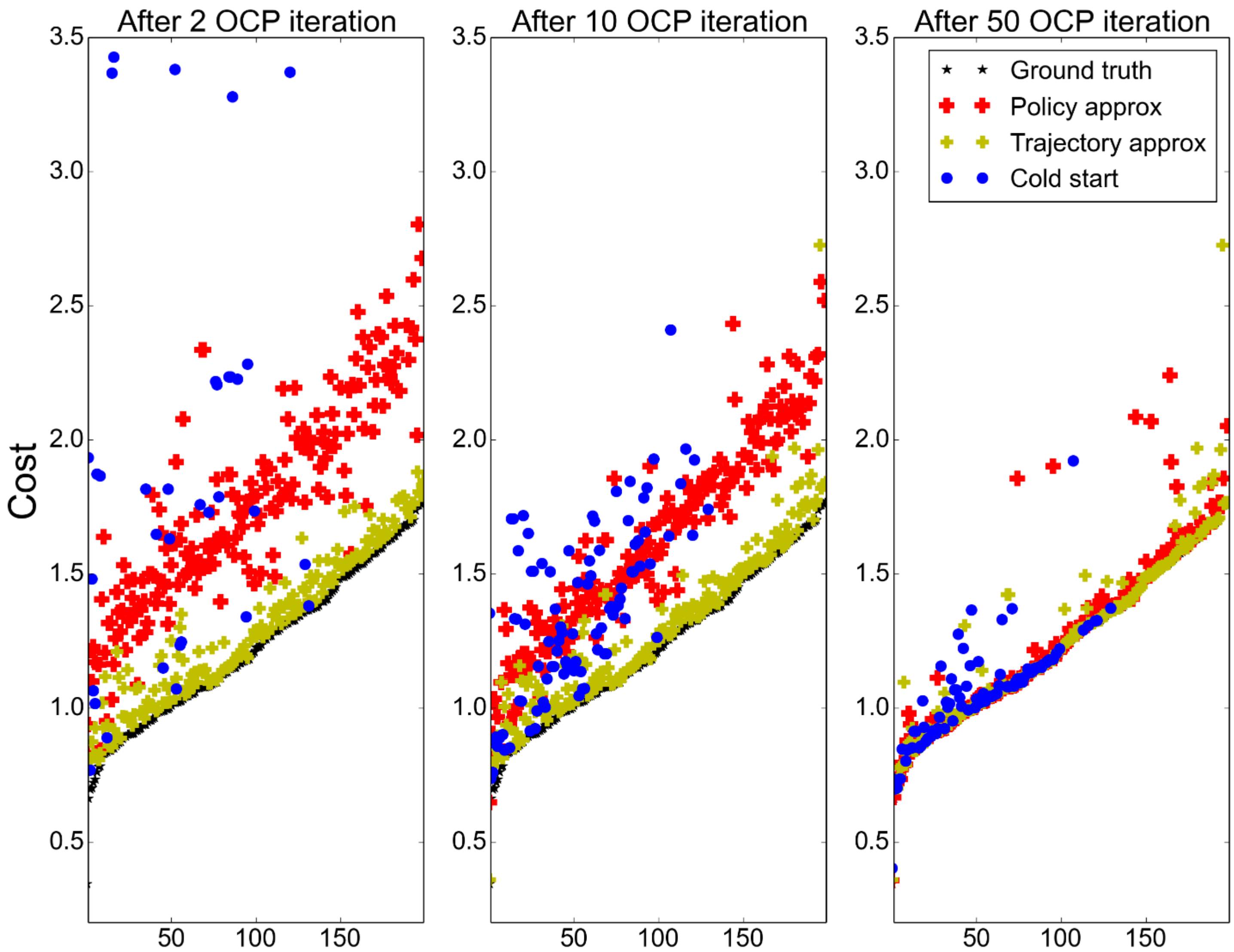
Double Pendulum

Number of states: 4
Number of controls: 2
Torque limites:
- joint 1: 5 Nm
- joint 2: 10 Nm
Mass: 6 kg

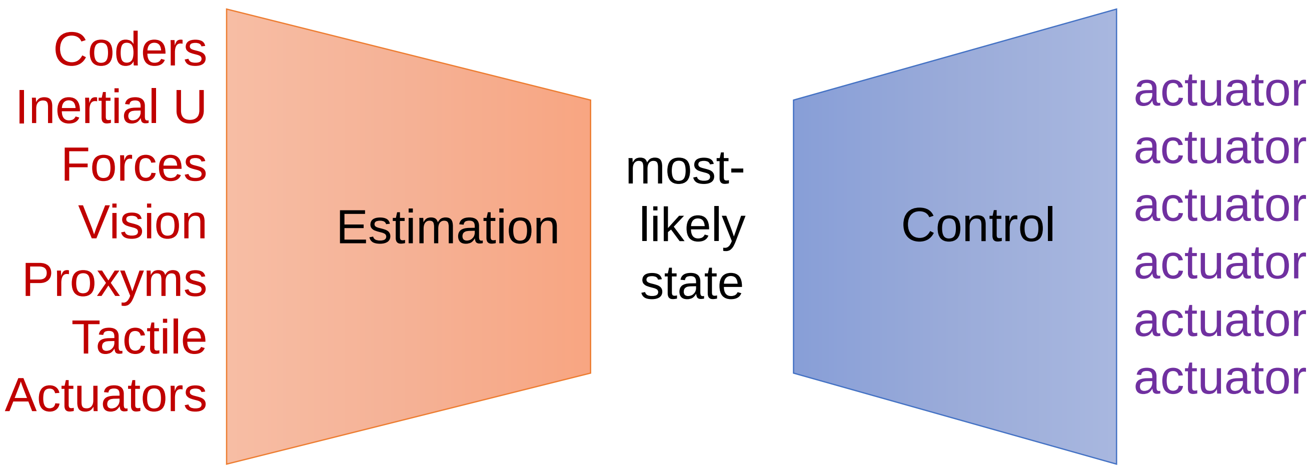


PRM: 30 nodes
IREPA: 6 iterations
Training time: 55 mins

IREPA with MPC



End-to-end control?



1. State-dependent force model
2. Learning local sensori-motor models …

… and use them for control-oriented predictions

Force-based control of balance

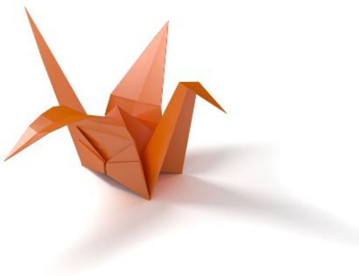
Take-home messages

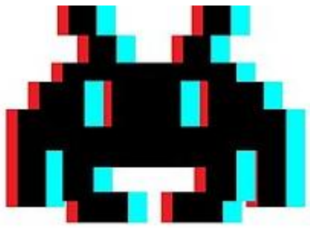
Numerical problems (few/none discrete constraints)

- nonconvex … warm start needed
- very constrained … mostly feasibility problems

The formulation/transcription is our central problem

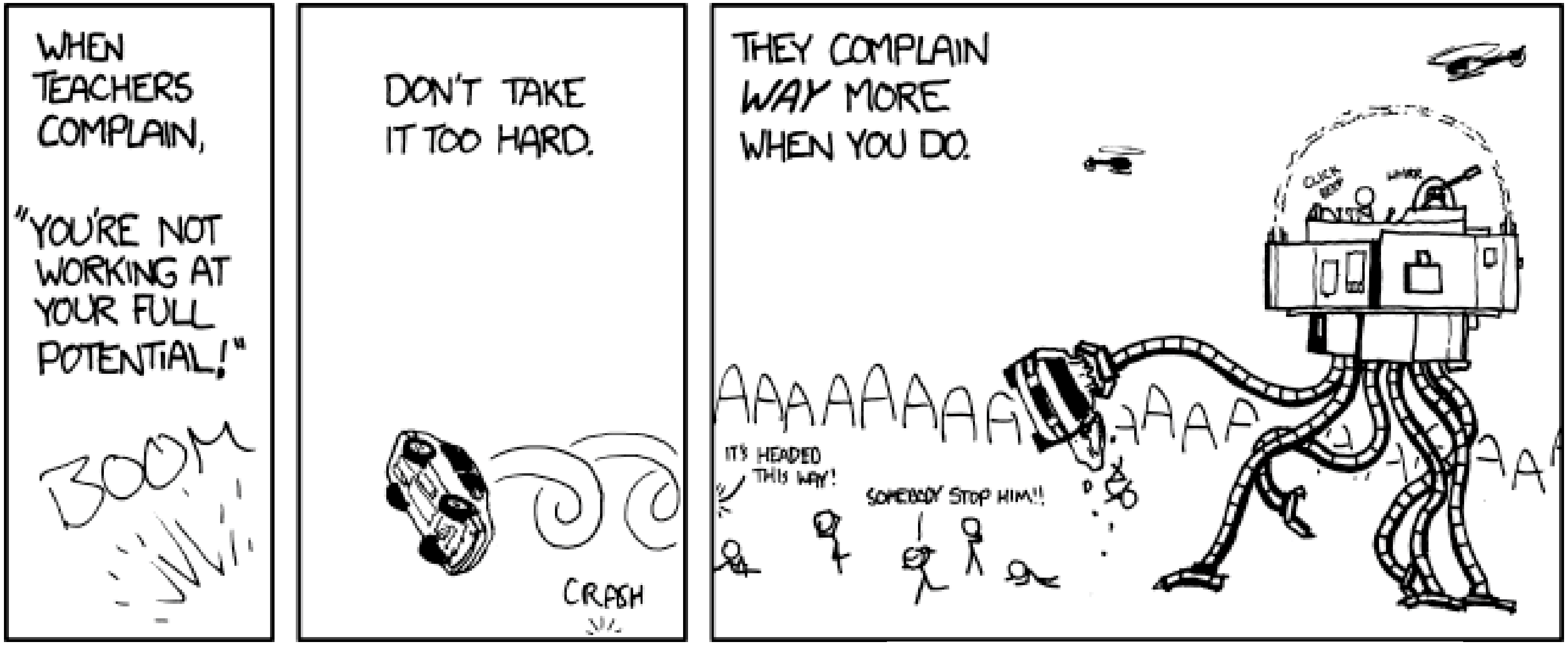
- expert+math knowledge
- keep generalization

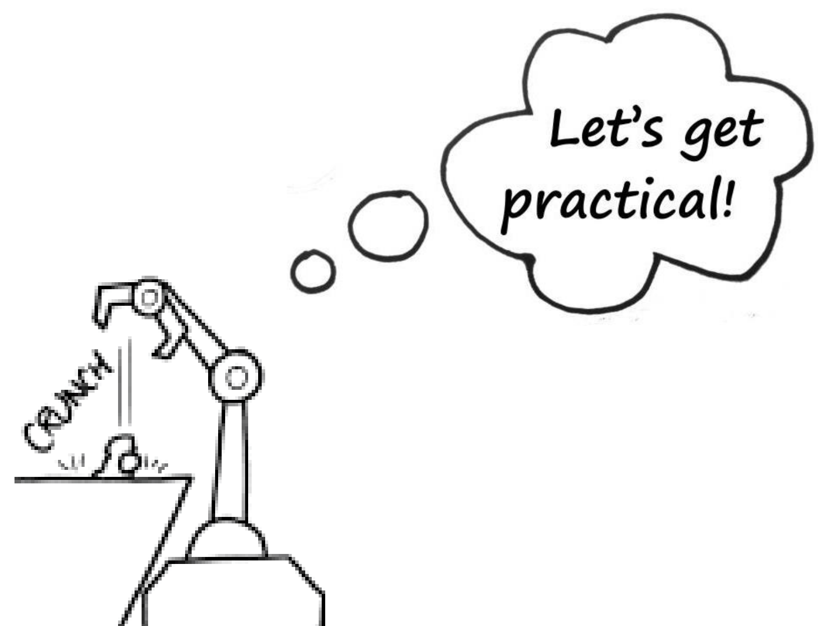




Optimal control = reinforcement learning

- train offline
- generalize online


WHEN TEACHERS COMPLAIN,
"YOU'RE NOT WORKING AT YOUR FULL POTENTIAL!"
BOOM
DON'T TAKE IT TOO HARD.
CRASH
THEY COMPLAIN *WAY* MORE WHEN YOU DO.
IT'S HEADED THIS WAY!
SOMEBODY STOP HIM!!


Let's get practical!
CRUNCH

Agenda 2019-2020

1. Geometry and inverse geometry

2. Kinematics and inverse kinematics

3. Dynamics and control

4. Optimal control and reinforcement learning

7/11	Introduction
8/11	Inverse geometry (1) -- pract.work #0
~~14/11~~	~~Inverse geometry (2) – pract.work #1~~
~~21/11~~	~~Inverse kinematics -- pract.work #2~~
28/11	Dynamics: simulation and control (1)
5/12	Experimental work: with Tiago -- by Olivier Stasse @ AIP
13/12	Industrial conference – Wandercraft, Airbus, Kineo Siemens
20/12	Experimental work: with Tiago – by Olivier Stasse @AIP
10/1	Dynamics: simulation and control (2) – pract.work #3
17/1	Dynamics: actuation and control – by Thomas Flayols
24/1	Experimental work: with Open Dynamic Robot Initiative – by Thomas Flayols @ LAAS
31/1	Optimal control – pract.work #4
6/2	Reinforcement learning – pract.work #5

Practical and experimental work

- In simulation

  Geometry: parallel robotics

  Kinematics: mobile manipulation

  Dynamics: dexterous manipulation

  Optimal control: flying machines

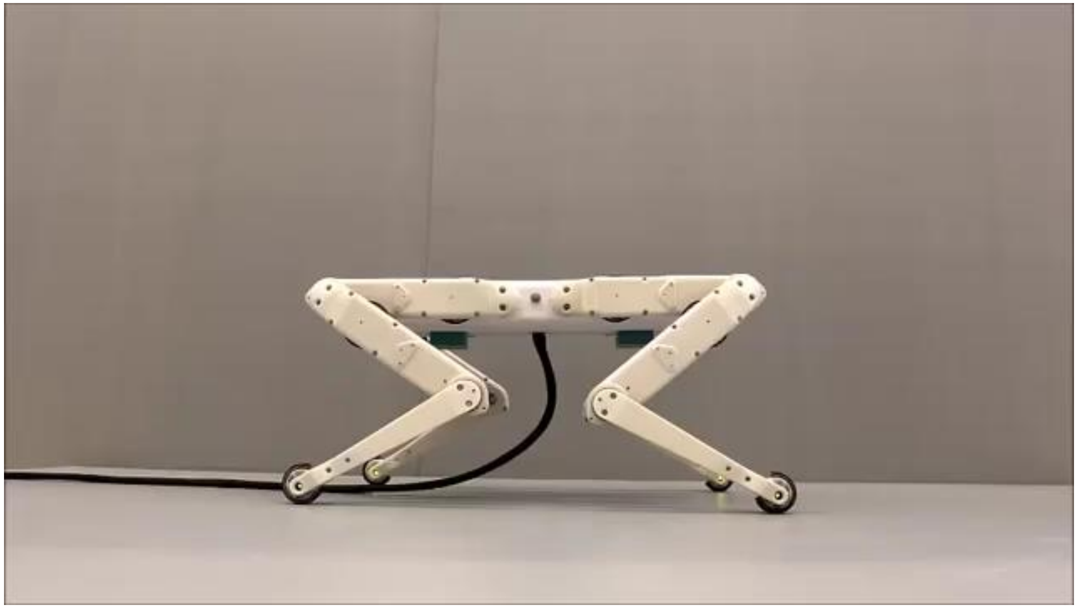
  Reinforcement learning: autonomous cars

- On hardware

  With Tiago: ROS, navigation and inverse kinematics

  With ODRI: actuation, low level control and force feedback


PAL
ROBOTICS
TIAGo
you?



Scoring

- Presentation of a paper
  - 10 minutes presentation
  - During each class, as a recap of previous lesson

- Practical work
  - 1h homework at each lesson (6 in total)
  - Each homework is scored between 0 and 5
  - Send the code – by mail? … maybe not

- Group evaluation VS individual evaluation
  - Likely: presentation in group, practical work individual
  - Collaboration in the class is encouraged

Paper presentation

- Geometry (14/11) ---Hanoune Hervier Bernard
  Learning the problem-optimum map: Analysis and application to global optimization in robotics, by Kris Hauser (TRO 2016)
- Kinematics (28/11) – Corderes Bhada Vachon
  Visual servo control, Part I: Basic approaches, by François Chaumette, S. Hutchinson (RAM 2016)
- Dynamics (10/1) – Debeunne Noel
  Feature-Based Locomotion Controllers, by Martin de Lasa
  Igor Mordatch, Aaron Hertzmann (TOG 2010)
- Simulation (17/1) – Niu Sun
  Staggered Projections for Frictional Contact in Multibody Systems, by D. Kaufman et al (TOG 2008)
  Interactive Simulation of Rigid Body Dynamics in Computer Graphics, by Jan Bender, Kenny Erleben , Jeff Trinkle and Erwin Coumans (STAR 2011)
- Actuation (24/1) – Creuse Arlaud Valette
  MIT Cheetah Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots, by Patrick Wensing et al (TRO 2016)
- Trajectory optimization (31/1) – Zerah Herlmer
  A tutorial on Newton methods for constrained trajectory optimization and relations to SLAM, Gaussian Process smoothing, optimal control, and probabilistic inference, by Marac Toussaint (Book 2017)
  Multi-contact Locomotion of Legged Robots by Justin Carpentier, Nicolas Mansard (TRO 2018)
  Control-Limited Differential Dynamic Programming, by Yuval Tassa, Nicolas Mansard and Emo Todorov (ICRA 2014)
- Reinforcement learning (6/2) – Vidal Consiglieri Templier
  End-to-End Training of Deep Visuomotor Policies, by Levine, Finn, Abeel (JMLR 2017)
  Using a Memory of Motion to Efficiently Warm-Start a Nonlinear Predictive Controller by Nicolas Mansard, Andrea del Prete, Mathieu Geisert, Steve Tonneau, Olivier Stasse (ICRA 2016)
  Interactive Control of Diverse Complex Characters with Neural Networks by Igor Mordatch, Kendall Lowrey, Galen Andrew, Zoran Popovic, Emanuel Todorov (NeurIPS 2016)

Web page of the class

- https://gepettoweb.laas.fr/index.php/Teach/Supaero2020

- Alias: https://frama.link/supaero2020

- Chat room for the class

  https://frama.link/supaero2020chat