Interplay between Optimal Control and Reinforcement Learning for Agile Locomotion and Dexterous Manipulation in Robotics

F. Schramm[1,2,3] N. Perrin-Gilbert[3] J. Carpentier[1,2]

[1]Inria Paris, France [2]Département d'informatique de l'ENS, PSL Research University, France [3]Sorbonne Université, France

Context and scientific objectives

The capability of modern robots to achieve dexterous manipulation and agile locomotion remains limited.

Goal: Learn new skills **efficiently** and develop **robust and versatile** controllers for robotics.
Intuition: Humans solve complex tasks **without thinking** about the movement of each of their muscles separately.
↪ exploits **motor synergies** to control several degrees of freedom simultaneously like central-nervous system
↪ serial and parallel **composition** of skills in a **hierarchic** fashion

Optimal control (OC): impressive results (acrobatic motions by Boston Dynamics [1]), but online re-planning and robustness to uncertainties or external perturbations remain very challenging.
Reinforcement learning (RL): flexible and robust against uncertainties, enables discovering of complex and rich solutions in the face of contact interactions [2], but methods remain data-intensive.

Should we learn or optimize? Leverage the advantages of both worlds:
- founded on the same mathematical principles (Bellman and Hamilton-Jacobi-Bellman equations) [3]
- how to combine RL policies and optimal controllers? High-level vs. low-level decisions and controls
- learn from limited data on a **physical robot** and apply policy efficiently to complex robotic tasks

This thesis is an interdisciplinary project with three main scientific axes: **control**, **perception**, and **experimentation** on simulated and real physical robots.

Advanced robotic platforms

- Experimental validation in simulation → need to overcome sim-to-real gap
- Conduct experiments on state-of-the-art robotic platforms for both locomotion and manipulation
- Lay a new computational framework for robot control on **real hardware**

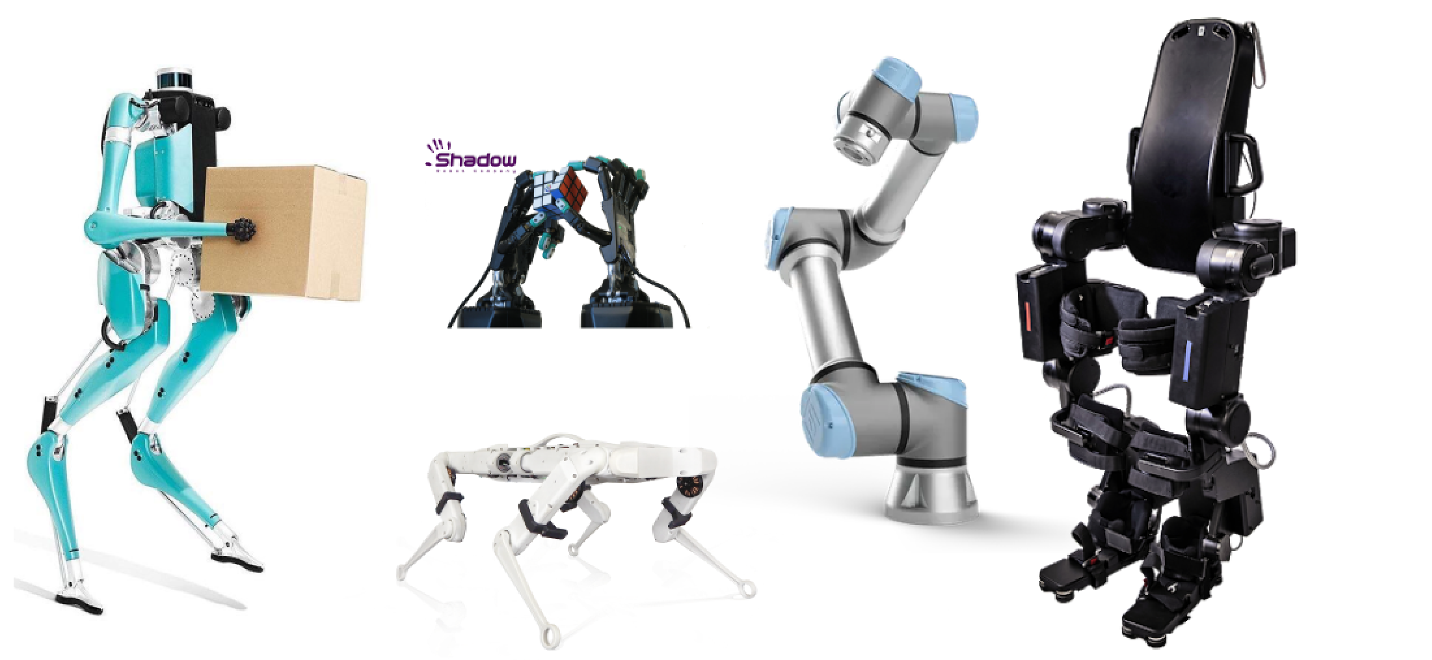

Figure: Biped digit, dexterous hands, quadruped solo, UR5 arm and exoskeleton atalante.

First-order trajectory optimization

Goal: **fast** trajectory optimization for systems with **non-smooth dynamics**.

- Build **differentiable simulator** based on PINOCCHIO [4]
- Use randomized smoothing [5] with automatic noise scheduling
- Compare against state-of-the-art RL algorithms using XPAG [6]

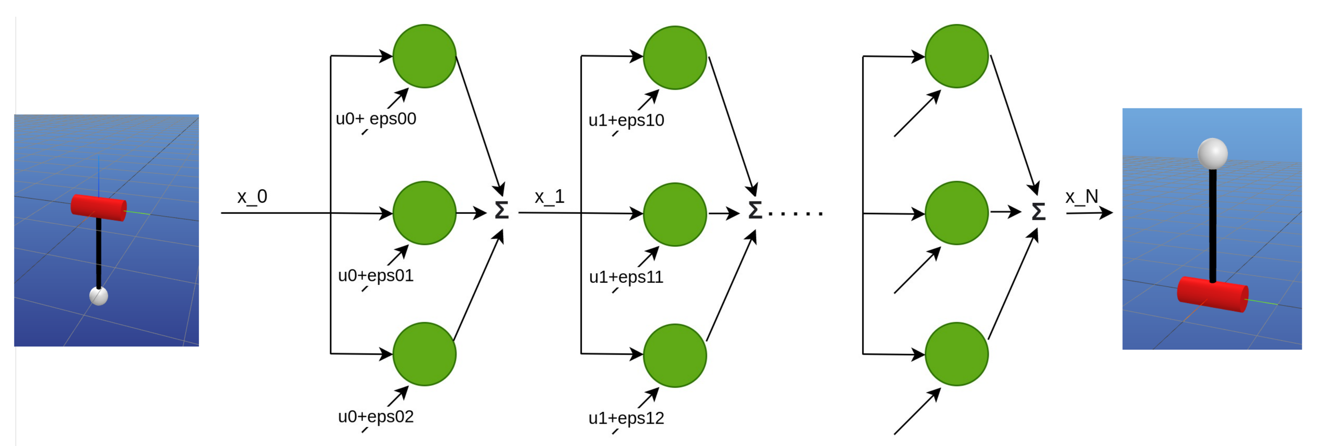

Figure: Randomized smoothing applied to cart-pole system with dry friction.

Randomized smoothing

published in *Nonlinear Analysis: Hybrid Systems, International Federation of Automatic Control (IFAC) journal*, 2024 [5]
- Optimal control (OC) algorithms take advantage of the derivatives of the dynamics to control physical systems efficiently
- Robotic problems can have non-smooth dynamics
- Discontinuities in the derivatives or the presence of non-informative gradients
  ↪ introduce randomization in the optimization
  ↪ more exploratory behavior by collecting samples in the neighborhood

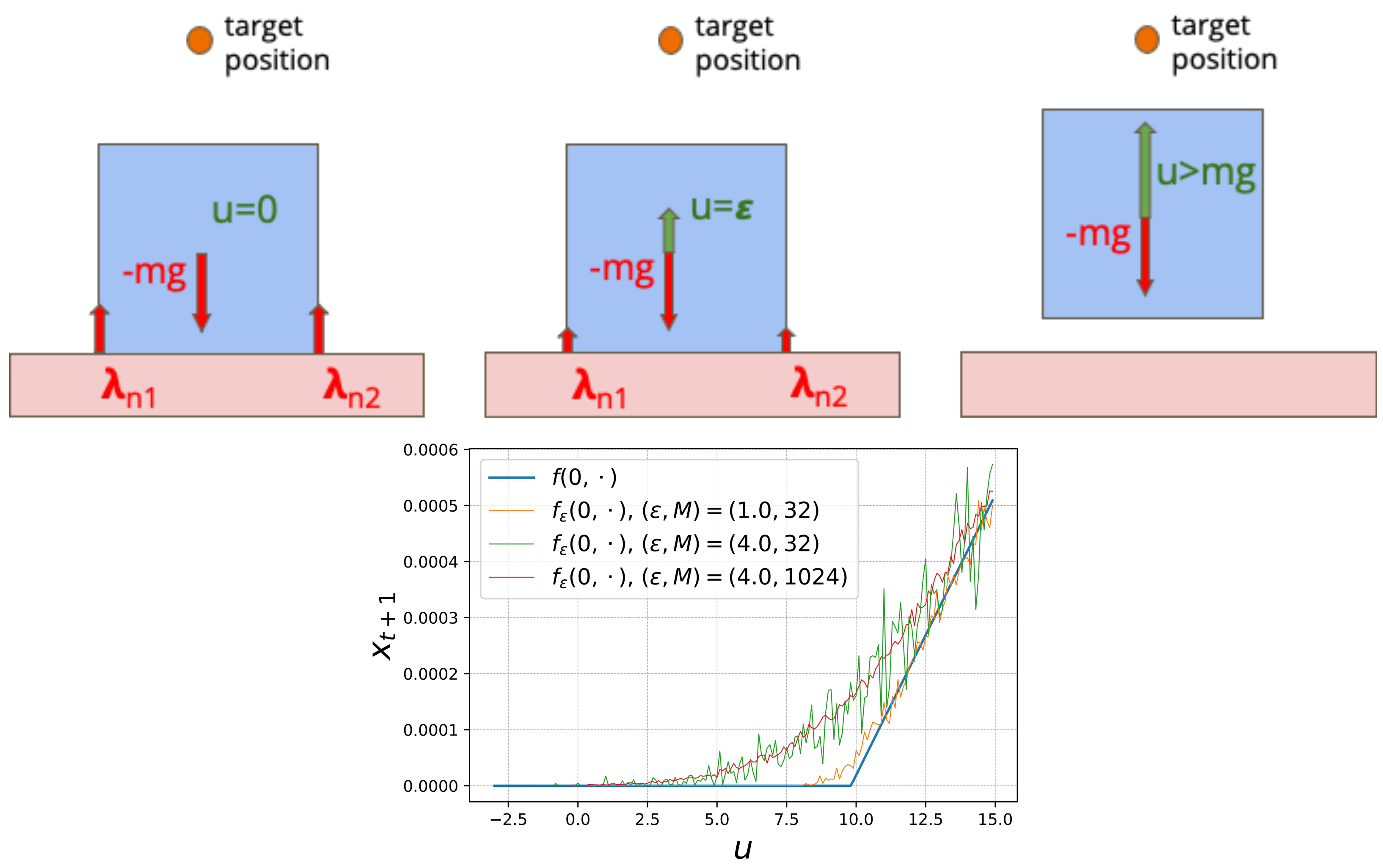

Figure: Lifting a cube exhibits non-smooth behavior and zero-gradient issues.

Gaussian formulation

The optimization problem can be written as:

$$\vec{u}^* = \underset{\vec{u}\in\{U_0,\dots,U_{T-1}\}}{\arg\min} \; \mathcal{L}_T(\vec{u}), \tag{1}$$

with loss function

$$\mathcal{L}_T(\vec{u}) = \|q_T(\vec{u}) - q_{\text{target}}\|^2 + \alpha\|\vec{u}\|^2, \quad \text{where} \|\vec{u}\|^2 = \sum_{i=0}^{T-1}\|u_i\|^2 \tag{2}$$

and the system dynamics are defined recursively $x_{t+1} = f(x_t, u_t)$, where $x = [q, \dot{q}]$.

With the energy of the system $\mathcal{L}_T$, we obtain an (unnormalized) probabilistic Gibbs distribution

$$g(\vec{u}) \propto \exp\left(-\frac{\mathcal{L}_T(\vec{u})}{\tau}\right), \tag{3}$$

where τ is the temperature and function $g(\vec{u})$ should be maximized.
In general, the KL divergence for two distributions is defined as

$$D_{KL}(P\|G) = \int_{-\infty}^{\infty} p(u) \log\left(\frac{p(u)}{g(u)}\right) du = \mathbb{E}_{u\sim U_\theta}\left[\log p(u) - \log g(u)\right].$$

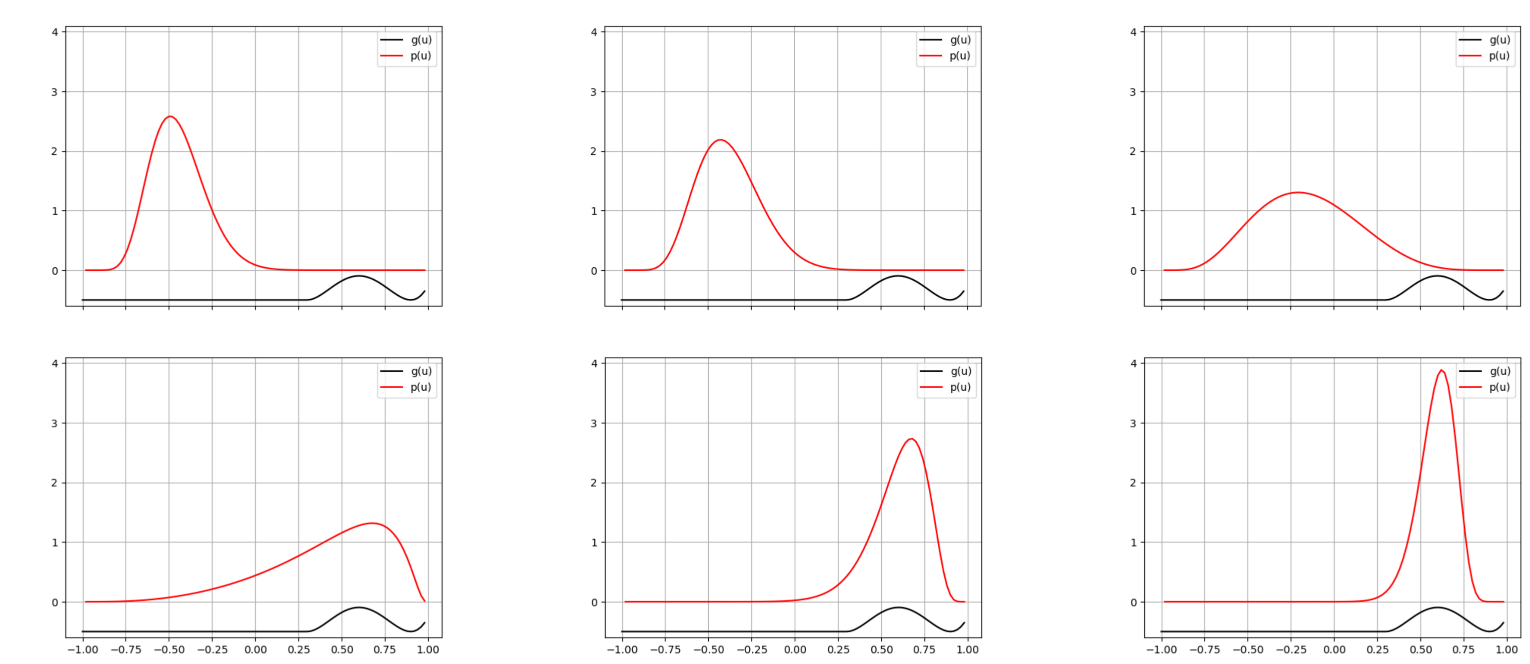

Figure: Randomized smoothing with Gaussians for 1D toy example. Showing iterations 0, 10, 50, 100, 200, and 300.

References

1. Kuindersma, S. *Recent Progress on Atlas, the World's Most Dynamic Humanoid Robot.* https://www.youtube.com/watch?v=EGABAx52GKI9. 2020. (2022).
2. Hwangbo, J. *et al.* Learning agile and dynamic motor skills for legged robots. *Science Robotics* (2019).
3. Bertsekas, D. *Reinforcement learning and optimal control.* (Athena Scientific, 2019).
4. Carpentier, J. & Mansard, N. *Analytical Derivatives of Rigid Body Dynamics Algorithms.* in *Robotics: Science and Systems (RSS 2018)* (Pittsburgh, United States, June 2018). https://hal.laas.fr/hal-01790971.
5. Le Lidec, Q. *et al.* Leveraging randomized smoothing for optimal control of nonsmooth dynamical systems. *Nonlinear Analysis: Hybrid Systems* **52,** 101468. ISSN: 1751-570X. https://www.sciencedirect.com/science/article/pii/S1751570X24000050 (2024).
6. Perrin-Gilbert, N. *xpag: a modular reinforcement learning library with JAX agents.* 2022. https://github.com/perrin-isir/xpag.