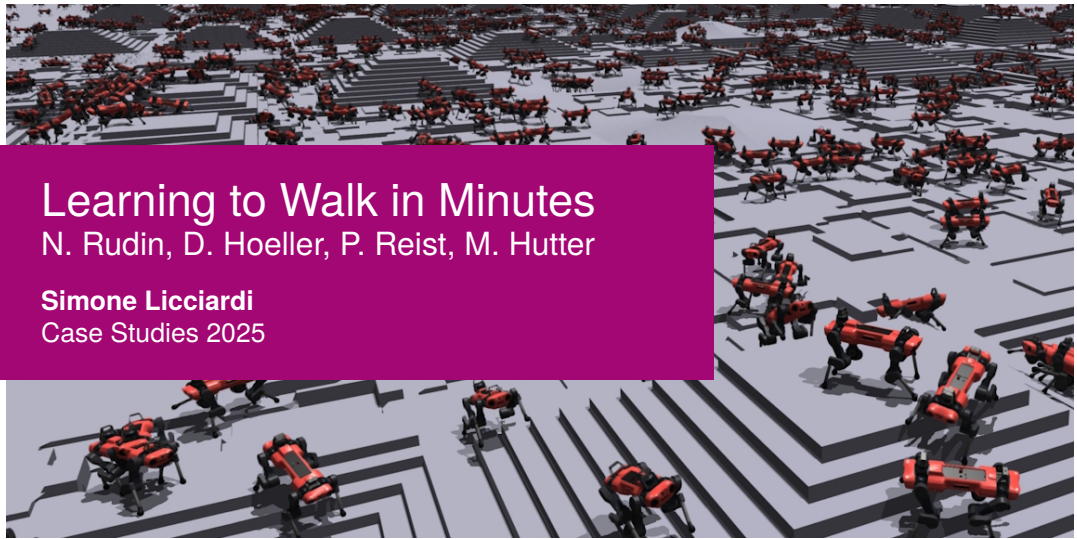
ETH zürich



Learning to Walk in Minutes

N. Rudin, D. Hoeller, P. Reist, M. Hutter

Simone Licciardi
Case Studies 2025

Introduction

What is your deepest desire?

Any respectable CSE person will answer
"simulate **anything**".

This talk is about the **"And then?"**: what does simulation enable in **Robotics**.

Robotics: Yesterday

Planning is out of question. We cannot model the environment well-enough

But we can **improve control** in the 1kHz-10Hz bandwidth.

The framework used to be, as anything in engineering, an optimization problem:

$$\begin{aligned} &\min_{(\dot{\mathbf{u}},\lambda)} . && \dots \\ &\text{s.t.} && \mathbf{F} = m\dot{\mathbf{u}}, \dots \end{aligned}$$

Robotics: Today

Robotics: Today

The **big jump** in performance and robustness we saw in the last few years is due to

- **No longer using sophisticated control** methods to solve a bad model,
- Approximating the control policy by a **neural net**; this is the so-called *Deep*[1] Reinforcement Learning.

[1] Here, *Deep* is slightly misused: normally, we are talking 3 layers, with about 250k-1M wheights. For comparison, ChatGPT's network has 1000B wheights.

Deep Reinforcement Learning

Algorithm 1 Deep RL Training Loop

Require: Initialized policy
  while Policy is not converged **do**
    Evaluate Policy
    Update Policy
  end while

Pros	Cons
High performance (90% of Hardware Limit), Robust to uncertainty	Difficult training (e.g. batch size), data-inefficient, cannot be parallelize trivially

Robotics DRL structure[2]

Algorithm 2 Robotics DRL Training Loop

Require: Initialized policy

while Policy is not converged **do**

Curriculum Choice

Policy Inference

Simulation

Cost and Observation Evaluation

Update Policy

end while

[2] Red you cannot parallelize, Petrol you can, Orange you conditionally can.

Adapting Update Policy

Is parallelization even worth it?

Increasing Batch Size B doesn't necessarily help training, because the information gain plateaus.

The authors kept B fixed, and **parallelized training over** n_{robot}. Consequently, they decreased n_{steps}, according to $B = n_{steps} n_{robots}$.

Adapting Simulation & Automatic Curriculum

We have very accurate engines that can parallelize computations, like **NVIDIA's Isaac**.

But, (through Isaac) we cannot

- increase the task difficulty over time,
- simulate multiple environments simultaneously.

To this end, the authors introduced the **game-inspired curriculum** where

- the environment is fixed,
- and is unique.

Adapting Simulation & Automatic Curriculum

Idea: parallelizing environments

Task diversity handled by spawning distribution rather than environment choice. Link.

Adapting Cost and Observation, and Other notes

The PPO critic (Value Iteration) must accoutn for the fact robots can simultaneously

- fail,
- **time-out**,
- succeed.

In serial implementation, the second state is ignored; but here we cannot. The authors decide to bootstap the Value Function.

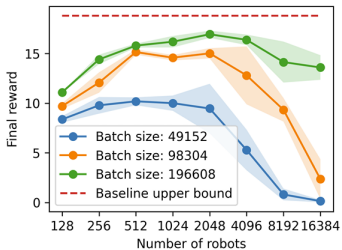
Robustness Results

- Not ANYmal-specific: works on different platforms (Unitree,...), and different types of platforms (Quadrupeds, Bipeds,...).
- The authors also implemented measures to limit the sim-to-real gap: physics and process noise, nonlinear actuation torque response modelling by an LSTM network.

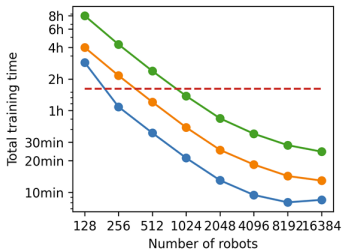
Here ↗ is the out-of-the-box performances.

Robustness Results

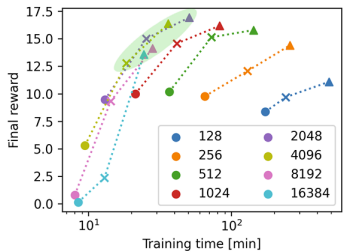
Training Results



(a)



(b)



(c)

Free lunch

We parallelized RL for Robotics reliably and in a flexible manner.

But, this performance we could achieve already without parallelization. So, **WSIC**?

Now training is **orders of magnitude faster**.

Free lunch

Specialization Bypasses Performance

Simulation-enabled online **specialization**

- to the current environment,
- on a single
- on-platform GPU

can drastically help reach the hardware-limit, **without** better methods.

Extensions

In successive works they were able to make this work **online**, **on-platform**, and achieved impressive out-of-the-box **robustness**.

ETH zürich

Thank you.

Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning

Nikita Rudin
ETH Zurich and NVIDIA
rudinn@ethz.ch

David Hoeller
ETH Zurich and NVIDIA
dhoeller@ethz.ch

Philipp Reist
NVIDIA
preist@nvidia.com

Marco Hutter
ETH Zurich
mahutter@ethz.com

Talk of Licciardi, S.

Outline

1. Appendix & Discussion

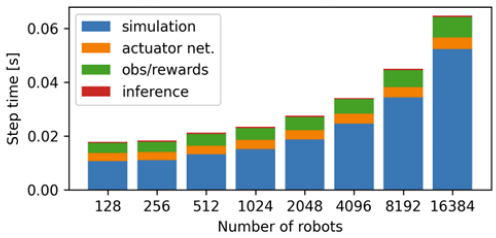
Appendix A: RL Details

In DRL the Neural Network parametrizes a stochastic policy by its moments, namely we model by parameters θ the mapping

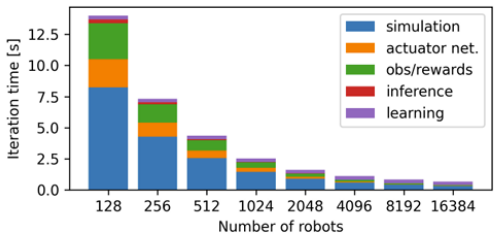
$$x \in \mathcal{X} \mapsto \pi_\theta(\cdot|x)$$

over $\mathcal{A}$ – namely the state and action space. Notably, π_θ is a conditional distribution. The parallelization step hinges on sampling n_{robots} times per timestep this distribution.

Appendix B: Computational Time Distribution



(a)



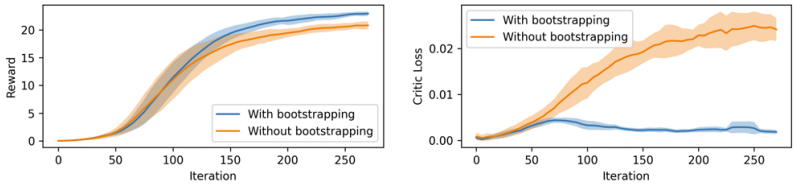
(b)

Appendix C: Simulation vs Policy Step

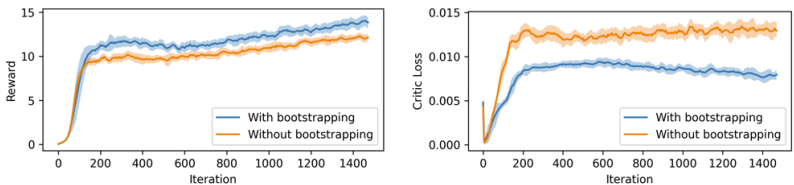
We find that we can not use a time step smaller than $0,005$*s (*200*Hz) which corresponds to four simulation steps per policy step.*

- Maximising simulation time in the mix is an easy way to improve throughput.
- The limit is imposed by the actuator network.

Appendix D: Bootstrapping Helps



(a) Flat Terrain



Appendix D: Bootstrapping Helps

- Boostrapping is not needed: without it, time-outs and failures are considered as the same termination state.
- Keeping the cost fixed for comparability, it helps achieving better performance (15%) per time unit.

Appendix E: Reward function

	definition	weight
Linear velocity tracking	$\phi(\mathbf{v}^*_{b,xy} - \mathbf{v}_{b,xy})$	$1dt$
Angular velocity tracking	$\phi(\boldsymbol{\omega}^*_{b,z} - \boldsymbol{\omega}_{b,z})$	$0.5dt$
Linear velocity penalty	$-\mathbf{v}^2_{b,z}$	$4dt$
Angular velocity penalty	$-\|\|\boldsymbol{\omega}_{b,xy}\|\|^2$	$0.05dt$
Joint motion	$-\|\|\ddot{\mathbf{q}}_j\|\|^2 - \|\|\dot{\mathbf{q}}_j\|\|^2$	$0.001dt$
Joint torques	$-\|\|\boldsymbol{\tau}_j\|\|^2$	$0.00002dt$
Action rate	$-\|\|\mathbf{q}^*_j\|\|^2$	$0.25dt$
Collisions	$-n_{collision}$	$0.001dt$
Feet air time	$\sum_{f=0}^{4}(\mathbf{t}_{air,f} - 0.5)$	$2dt$

Table 2: Definition of reward terms, with $\phi(x) := \exp(-\frac{||x||^2}{0.25})$. The z axis is aligned with gravity.

Appendix F: Physical and Process Noise

A.5 Noise Level in Observations

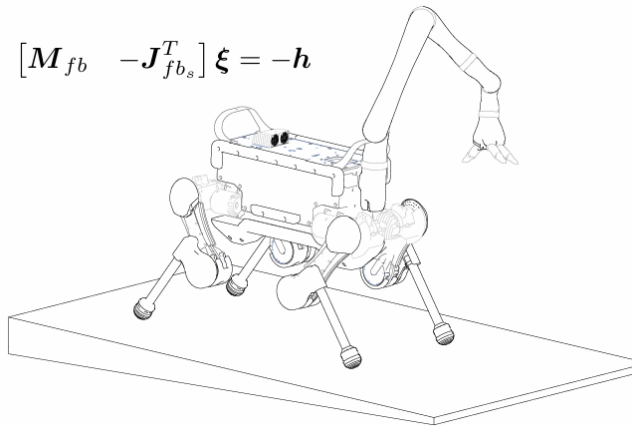
Joint positions	$\pm 0.01\,\mathrm{rad}$
Joint velocities	$\pm 1.5\,\mathrm{rad/s}$
Base linear velocity	$\pm 0.01\,\mathrm{m/s}$
Base angular velocity	$\pm 0.2\,\mathrm{rad/s}$
Projected gravity	$\pm 0.05\,\mathrm{rad/s^2}$
Commanded base linear velocity	$0\,\mathrm{m/s}$
Commanded base angular velocity	$0\,\mathrm{rad/s}$
Measured terrain heights	$\pm 0.1\,\mathrm{m}$

Table 4: Noise scale for the different components of the observations. For each element, the noise value is sampled from a uniform distribution with the given scale and added to the observations.

Appendix G: Classical Way

Priority	Task
1	Equations of Motion
1	Force/torque limits
2	No motion at contact points
3	CoM motion tracking
3	Torso angular motion
3	Foot motion tracking
3	End-effector motion tracking
3	End-effector force tracking
3	Torso orientation adaptation
4	Contact force minimisation

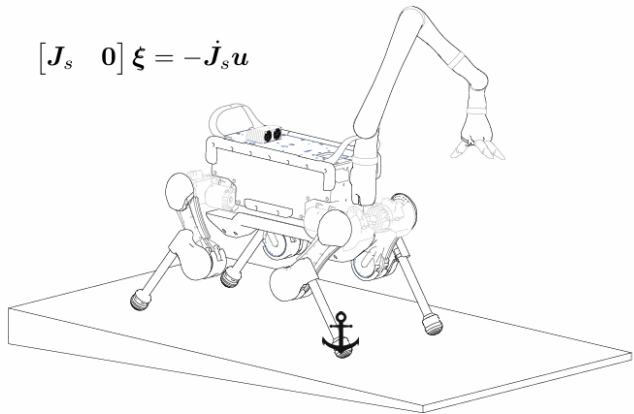
$$\begin{bmatrix} \boldsymbol{M}_{fb} & -\boldsymbol{J}_{fb_s}^T \end{bmatrix} \boldsymbol{\xi} = -\boldsymbol{h}$$



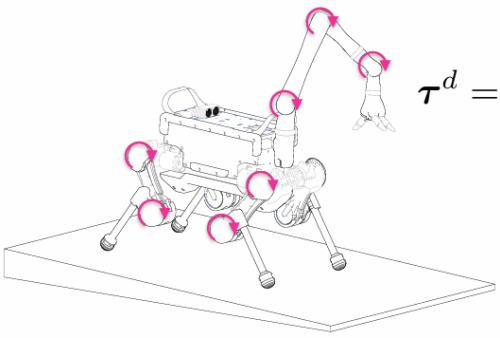
Appendix G: Classical Way

Priority	Task
1	Equations of Motion
1	Force/torque limits
2	No motion at contact points
3	CoM motion tracking
3	Torso angular motion
3	Foot motion tracking
3	End-effector motion tracking
3	End-effector force tracking
3	Torso orientation adaptation
4	Contact force minimisation

$$\begin{bmatrix} \boldsymbol{J}_s & \mathbf{0} \end{bmatrix} \boldsymbol{\xi} = -\dot{\boldsymbol{J}}_s \boldsymbol{u}$$



Appendix G: Classical Way



$$\boldsymbol{\tau}^d = \boldsymbol{M}_j(\boldsymbol{q})\dot{\boldsymbol{u}}^* + \boldsymbol{h}_j(\boldsymbol{q}, \boldsymbol{u}) - \boldsymbol{J}_{s,j}(\boldsymbol{q})\boldsymbol{\lambda}^*$$

$$\boldsymbol{\tau}^{ref} = \boldsymbol{\tau}^d + \boldsymbol{k}_P\tilde{\boldsymbol{q}} + \boldsymbol{k}_D\dot{\tilde{\boldsymbol{q}}}$$