Learning Quadruped Locomotion Using Differentiable Simulation

Alexandre Jaspard
Florian Amann
Martin Mayer

11 Nov 2025

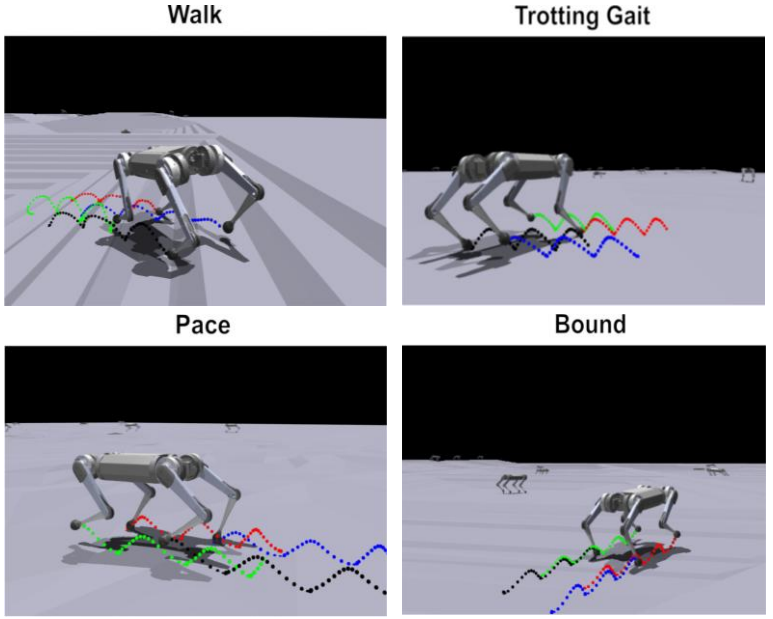
https://www.bostonglobe.com/metro/2019/11/13/watch-mit-new-mini-cheetah-robots-action/LIkmPlpaJ6Yj6B8VzP6ceP/story.html

Introduction

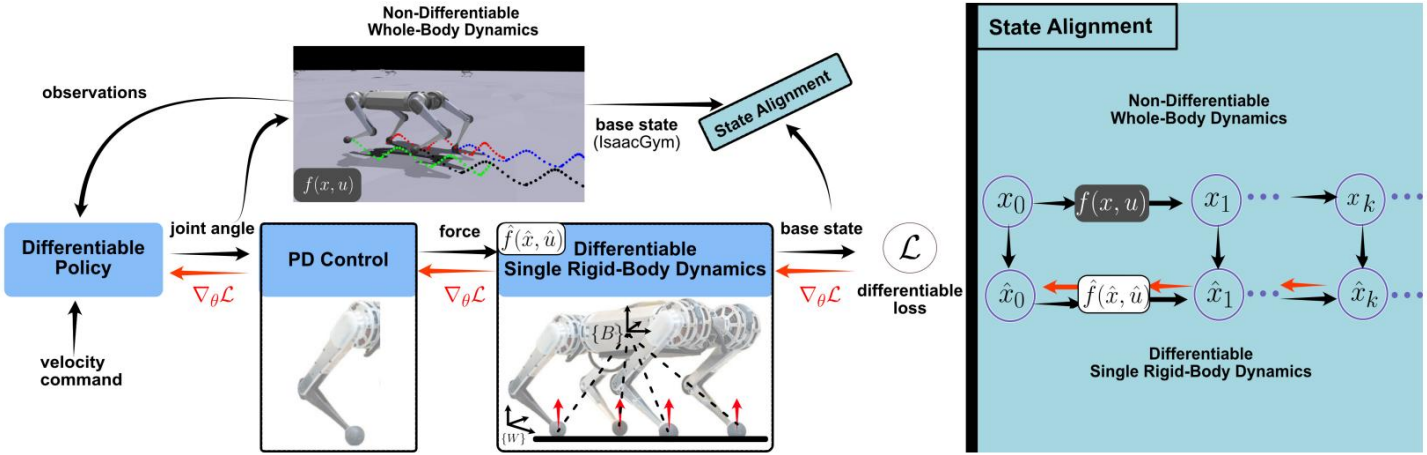
- Differentiable simulation for computing smooth, low variance gradients
  - Faster
  - More stable
- Combination of differentiable & non-differentiable simulators
- Outperforms RL
  - Same accuracy in much less time
  - Transfer to real-world applications without fine-tuning

Key Aspects

- Quadruped robots, with or without parallelization
- With parallelization, multiple gait patterns on variable terrains
  - Trot
  - Pace
  - Bound
  - Gallop
- Control via PD controller on joint position and velocities
- Single rigid body for backpropgation
- Realistic simulator to align states



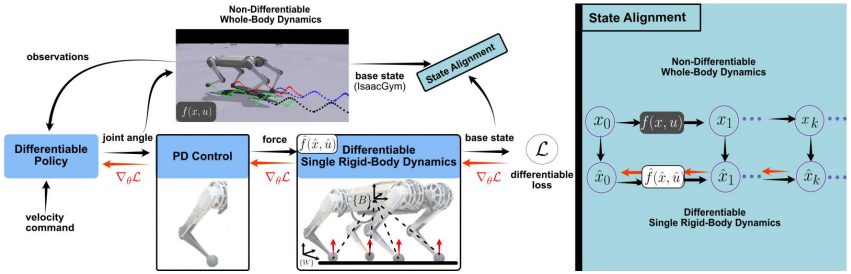
Control policy



- **Simplified** version of the robot – Single rigid-body dynamics for **differentiability**
- **State adjustment** with complex non-differentiable Whole-body dynamics
- **Backpropagation** through the differentiable simulator
- **Optimization** of the control policy to minimize :

$$\min_\theta \mathcal{L}_\theta = \sum_{k=0}^{N-1} l(x_k, u_k) = \sum_{k=0}^{N-1} l(x_k, \pi_\theta(o_k))$$
$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_\theta,$$

Forward propagation



- Simplified version of the robot – Single rigid-body dynamics

$$\dot{\mathbf{p}}_{WB} = \mathbf{v}_{WB} \qquad \dot{\mathbf{v}}_{WB} = \frac{1}{m}\sum_i \mathbf{f}_i + \mathbf{g}$$

$$\dot{\mathbf{q}}_{WB} = \frac{1}{2}\Lambda(\boldsymbol{\omega}_B)\cdot\mathbf{q}_{WB} \qquad \dot{\boldsymbol{\omega}}_B = \mathbf{I}^{-1}\left(\boldsymbol{\eta} - \boldsymbol{\omega}_B \times (\mathbf{I}\boldsymbol{\omega}_B)\right).$$

Input : Reaction forces

Output : Rigid-body's state

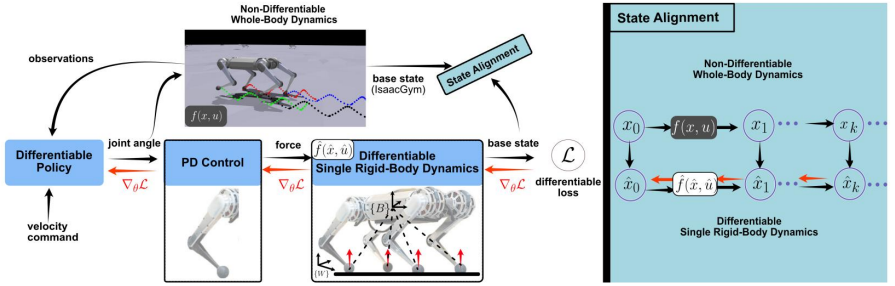
- Control policy $\pi_\theta(o_k)$ output target joins positions
- PD controller to convert into torques maintaining continuity and differentiability

$$\boldsymbol{\tau} = \mathbf{k_p}(\mathbf{q}^{\text{ref}} - \mathbf{q}) + \mathbf{k_d}(\dot{\mathbf{q}}^{\text{ref}} - \dot{\mathbf{q}})$$

- Use of Jacobian to convert toques into reaction force to input in the Single rigid-body dynamics
- This method provide smoothed approximations of reaction forces providing consistency and differentiability

Martin Mayer

Backward propagation Trough Time



Differential model allow :

Back propagation Through Time, every step in the pass is influenced by the result

$$\nabla_\theta \mathcal{L}_\theta = \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{i=1}^{k} \frac{\partial l_k}{\partial x_k} \prod_{j=i}^{k} \underbrace{\left(\frac{\partial x_j}{\partial x_{j-1}} \right)}_{\text{differentiable dynamics}} \frac{\partial x_i}{\partial \theta} + \frac{\partial l_k}{\partial u_k} \frac{\partial u_k}{\partial \theta} \right)$$

Problem : vanishing or exploding gradients → need **short horizons training**.

Solution : short term **training put in series**.

State alignment

- Use the state from a non-differential simulator (IsaacGym) to align the simplified differentiable simulation

- Alignment equation : $$\hat{\mathbf{x}}_{t+1}^{\text{diff}} = \mathbf{x}_{t+1}^{\text{non-diff}} + \alpha * (\mathbf{x}_{t+1}^{\text{diff}} - \mathbf{x}_{t+1}^{\text{diff, detach}})$$

- $\mathbf{x}_{t+1}^{\text{diff}}$ And $\mathbf{x}_{t+1}^{\text{diff, detach}}$ represent the same robot state through differentiable simulator. The *detach* is here to cancel de differential term in the forward but make it appear in the backward for training.

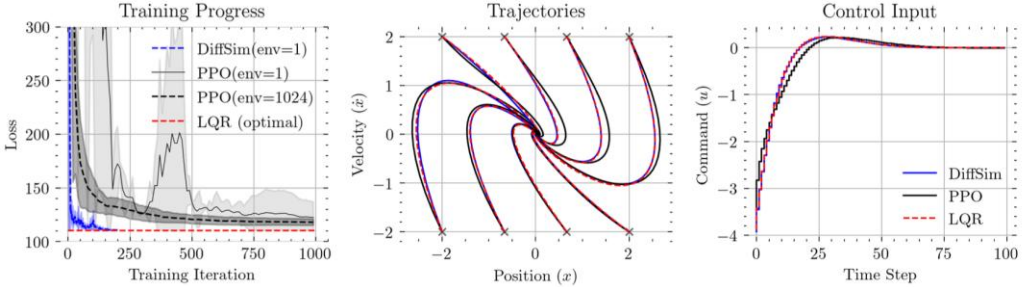
- Backprop. Gradient calculation : $$\partial \hat{\mathbf{x}}_{t+1}^{\text{diff}} / \partial \mathbf{x}_{t}^{\text{diff}} = \mathbf{0} + \alpha * \partial \mathbf{x}_{t+1}^{\text{diff}} / \partial \mathbf{x}_{t}^{\text{diff}}$$

Martin Mayer

Result of the differential method

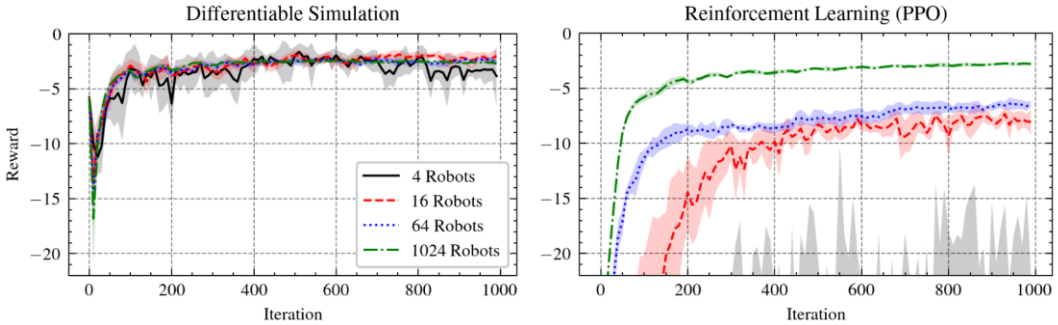
Toy Validation : double integrator

- Differentiable simulation achieves near-optimal LQR policy within few iterations.
- PPO fails to converge to the same level even with 1024 parallel environments.



Quadruped Robot Locomotion

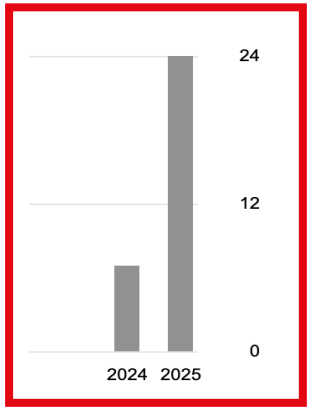
- Single robot, 24 timesteps per iteration.
  Learns to walk forward (vx = 0.2 m/s) within minutes.
- PPO fails under such low-sample regime.
  → Massive sample efficiency improvement.



Result of the differential method

- Learns multiple gaits (trot, pace, bound, gallop) across 1–4 Hz.
- Robust to perturbations and terrain irregularities.
- Sim2real : transfer on Mini Cheetah (100 Hz control) with **no fine-tuning**.

Citation and Influence



- 31 Citations across 2024 and 2025
- Most works relate to locomotion
- No real development of the limitations
- Critics
  - Relies on a few key signals to aid learning, such as the reference foot position and the gait phases
  - Lack of sim-to-real comparison with PPO
  - No other baseline than PPO

OpenReview.net *(3 reviews)*

- **Potential Impact :** The work contains interesting new ideas that will potentially have major impact in robotics or machine learning.
- **Potential Impact :** The work will potentially have some impact in robotics or machine learning.
- **Potential Impact :** The work contains interesting new ideas that will potentially have major impact in robotics or machine learning.

Analysis of Pros & Cons

Advantages

- Overcomes discontinuous dynamics
- Fast convergence
- Very few data needed

Disadvantages

Difficult to generalize

No non-differentiable (binary) rewards

Relies on key ‘signals’ to aid learning

Thank you for your attention

Questions ?