Rapid Motor Adaptation

Hansol Kang

Introduction

- Walking robots in recent days

Conventional Control via
Physical dynamics and several control theories

Require considerable expertise on the part of the human designer.

Reinforcement Learning
Imitation Learning

Sim-to-Real problem is quite challenging

Paper Review

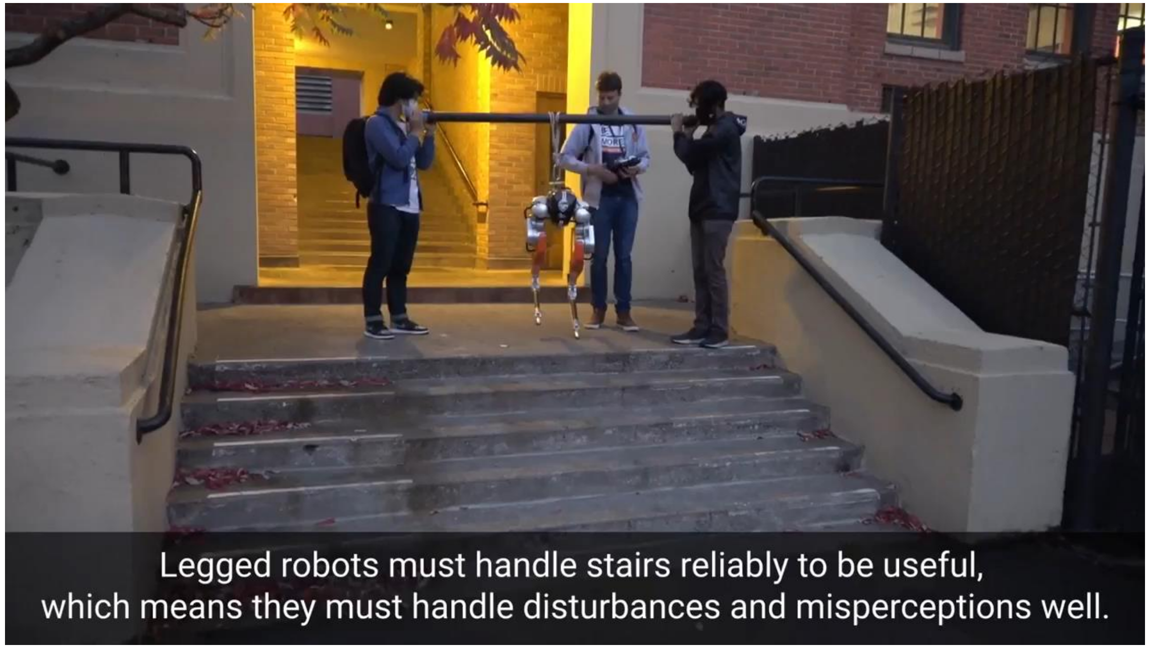
Human Walking

Even if the ground environment changes while walking,
a person can adapt to the environment and walk.

Paper Review

Robot walking

But, Robots are not.



Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning, Jonah Siekmann et al., Robot.: Sci. Syts., 2021

Boston Dynamics' Atlas Falls Over After Demo at the Congress of Future Scientists and Technologists, https://www.youtube.com/watch?v=TxobtWAFh8o

Paper Review

- RMA: Rapid Motor Adaptation for Legged Robots

arXiv:2107.04034v1 [cs.LG] 8 Jul 2021

RMA: Rapid Motor Adaptation for Legged Robots

Ashish Kumar, UC Berkeley
Zipeng Fu, Carnegie Mellon University
Deepak Pathak, Carnegie Mellon University
Jitendra Malik, UC Berkeley, Facebook

Fig. 1: We demonstrate the performance of RMA on several challenging environments. The robot is successfully able to walk on sand, mud, hiking trails, tall grass and dirt pile without a single failure in all our trials. The robot was successful in 70% of the trials when walking down stairs along a hiking trail, and succeeded in 80% of the trials when walking across a cement pile and a pile of pebbles. The robot achieves this high success rate despite never having seen unstable or sinking ground, obstructive vegetation or stairs during training. All deployment results are with the same policy without any simulation calibration, or real-world fine-tuning. Videos at https://ashish-kmr.github.io/rma-legged-robots/

***Abstract*—Successful real-world deployment of legged robots would require them to *adapt in real-time* to unseen scenarios like changing terrains, changing payloads, wear and tear. This paper presents Rapid Motor Adaptation (RMA) algorithm to solve this problem of real-time online adaptation in quadruped robots. RMA consists of two components: a base policy and an adaptation module. The combination of these components enables the robot to adapt to novel situations in fractions of a second. RMA is trained completely in simulation without using any domain knowledge like reference trajectories or predefined foot trajectory generators and is deployed on the A1 robot without any fine-tuning. We train RMA on a varied terrain generator using bioenergetics-inspired rewards and deploy it on a variety of difficult terrains including rocky, slippery, deformable surfaces in environments with grass, long vegetation, concrete, pebbles, stairs, sand, etc. RMA shows state-of-the-art performance across diverse real-world as well as simulation experiments. Video results at https://ashish-kmr.github.io/rma-legged-robots/.**

I. Introduction

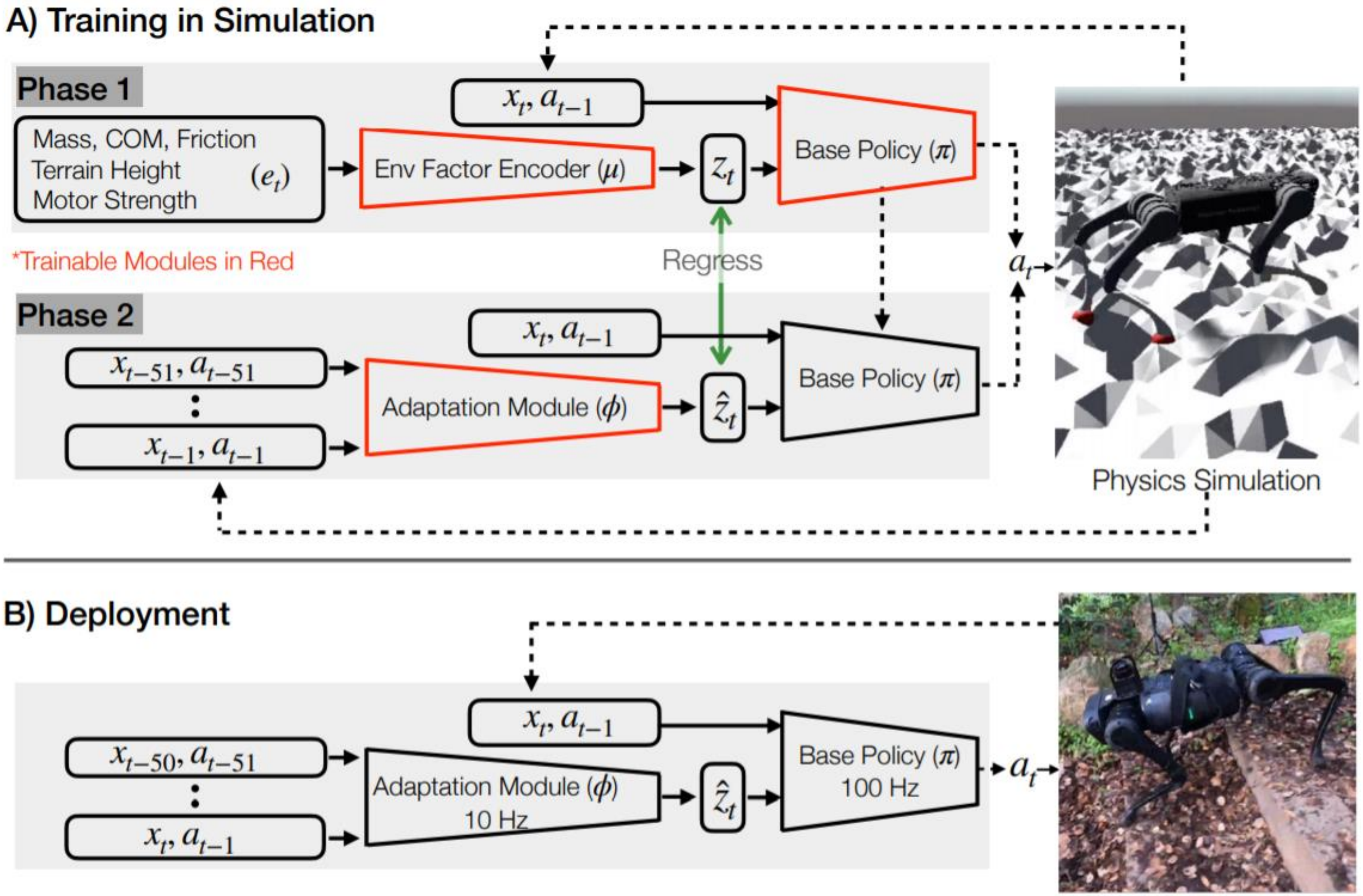
Great progress has been made in legged robotics over the last forty years through the modeling of physical dynamics and the tools of control theory [36, 43, 46, 16, 56, 63, 50, 26, 28, 2, 24]. These methods require considerable expertise on the part of the human designer, and in recent years there has been much interest in replicating this success using reinforcement learning and imitation learning techniques [23, 18, 41, 55, 32] which could lower this burden, and perhaps also improve performance. The standard paradigm is to train an RL-based controller in a physics simulation environment and then transfer to the real world using various sim-to-real techniques [52, 40, 23]. This transfer has proven quite challenging, because the sim-to-real gap itself is the result of multiple factors: (a) the physical robot and its model in the simulator differ significantly; (b) real-world terrains vary considerably (Figure 1) from our models of these in the simulator; (c) the physics simulator fails to accurately capture the physics of the real world – we are dealing here with contact forces, deformable surfaces and the like – a considerably harder problem than modeling rigid bodies moving in free space.

In this paper, we report on our progress on solving this

- 2021.07.08 arXiv
- Robotics: Science and Systems 2021(RSS 2021)
  July, 12 – 16, 2021
- Ashish Kumar[1], Zipeng Fu[2], Deepak Pathak[2], Jitendra Malik[1,3]
- [1]UC Berkeley, [2]Carnegie Mellon University, [3]Facebook
- This paper presents Rapid Motor Adaptation(RMA) algorithm to solve a problem of real-time online adaptation in quadruped robots.

Paper Review

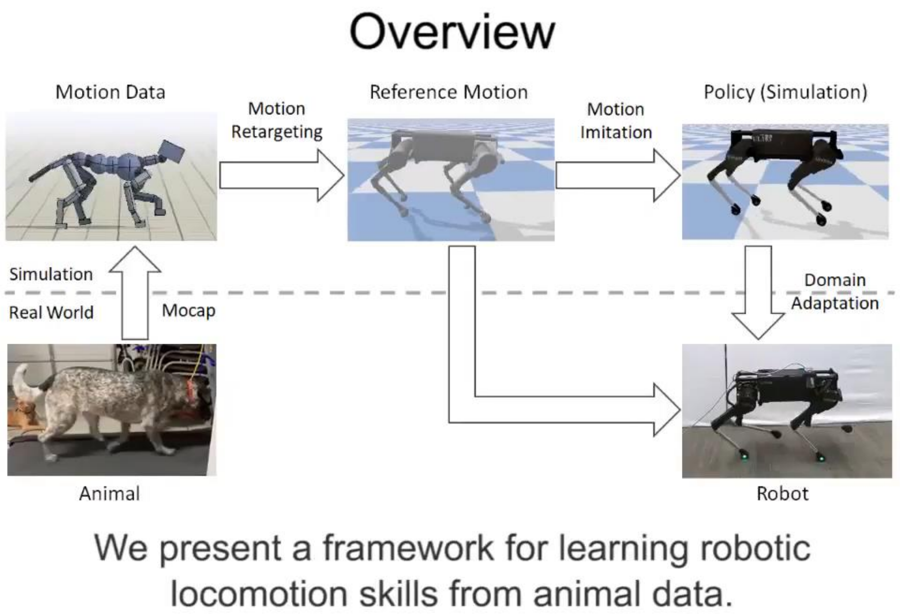
RMA Architecture

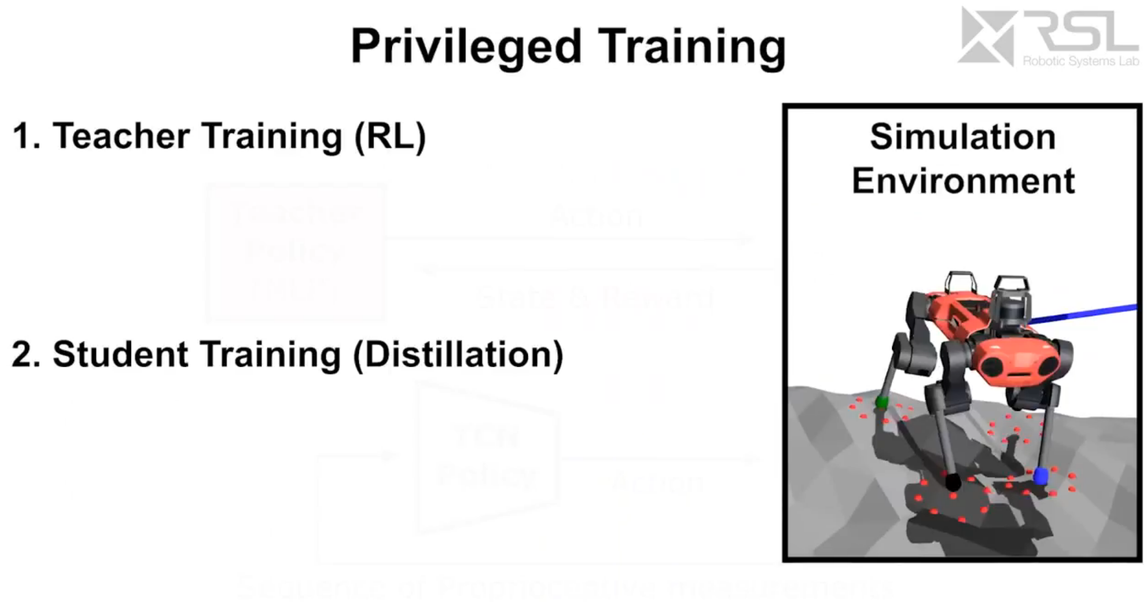


Paper Review

Papers that use encoder

There are some researches that include encoder in their architecture.





Xue Bin Peng et al., Robot.: Sci. Syst., 2020

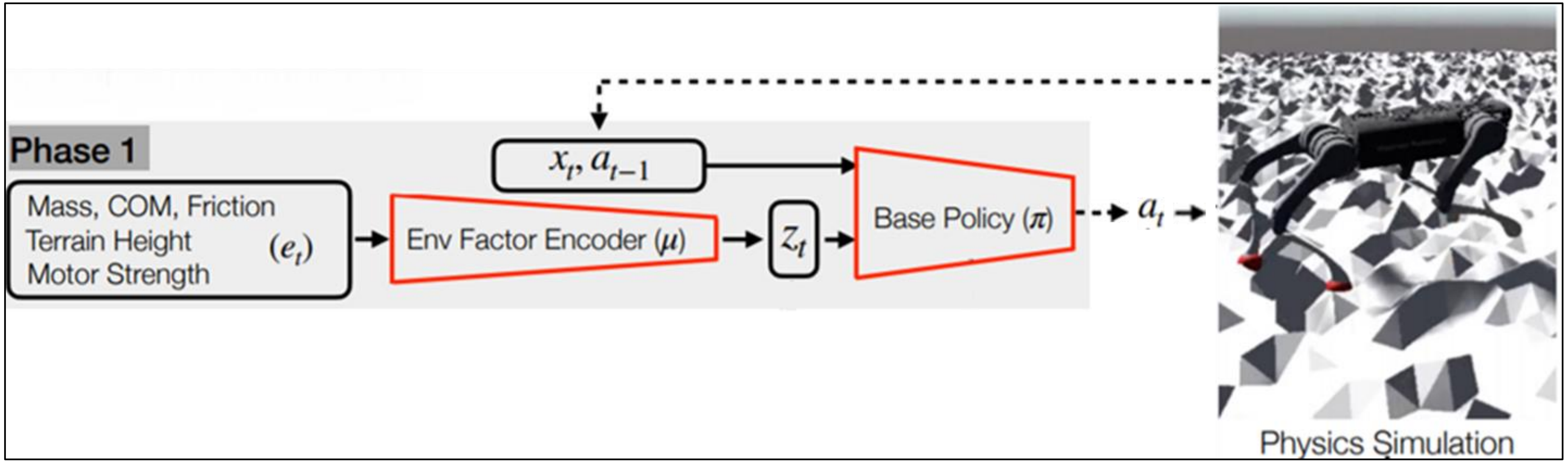
- Reference demonstrations are needed.
- Collecting 4 – 8 mins(50 episodes of 5 – 10s) real world data is needed.

Lee et al., Sci. Robot., 2020

- Predefined trajectory generator was used.
- Motor model was used.

Paper Review

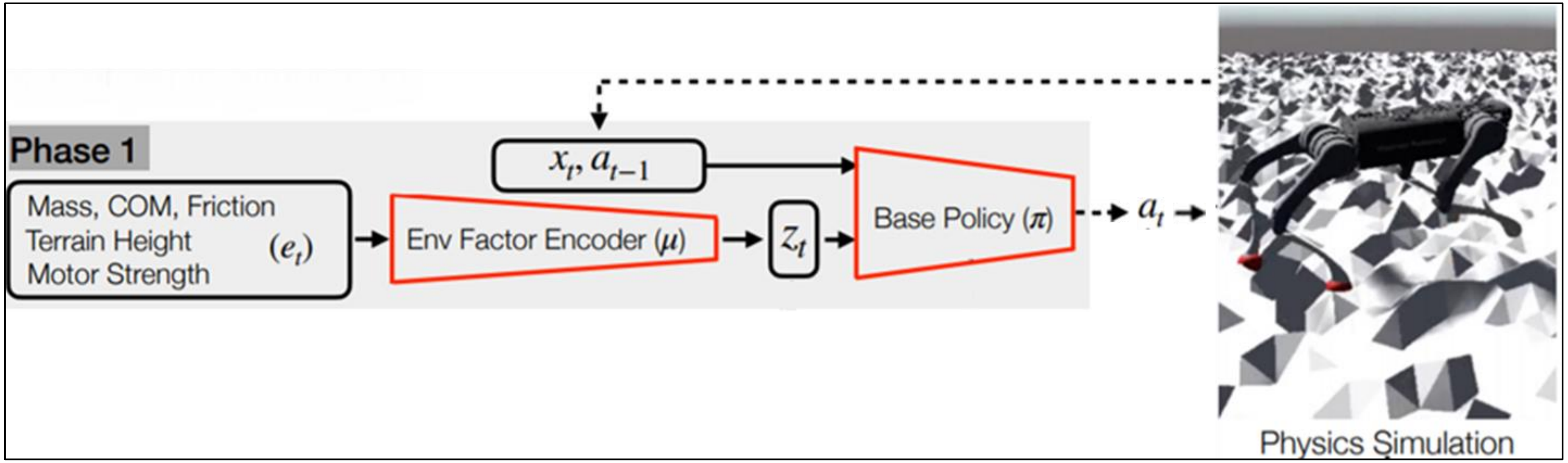
Base Policy



- Current state $x_t \in \mathbb{R}^{30}$ includes joint positions(12), joint velocities(12), roll and pitch of the torso(2) and binary foot contact indicators(4).
- Previous action $a_{t-1} \in \mathbb{R}^{12}$ includes the desired joint angles.
- Extrinsic vector $z_t \in \mathbb{R}^{8}$ is a low dimensional encoding of the environment vector $e_t \in \mathbb{R}^{17}$ generated by encoder μ.
- Environment vector $e_t \in \mathbb{R}^{17}$ includes mass(1), position of CoM(2), motor strength(12), friction(1), local terrain height(1).

Paper Review

Base Policy



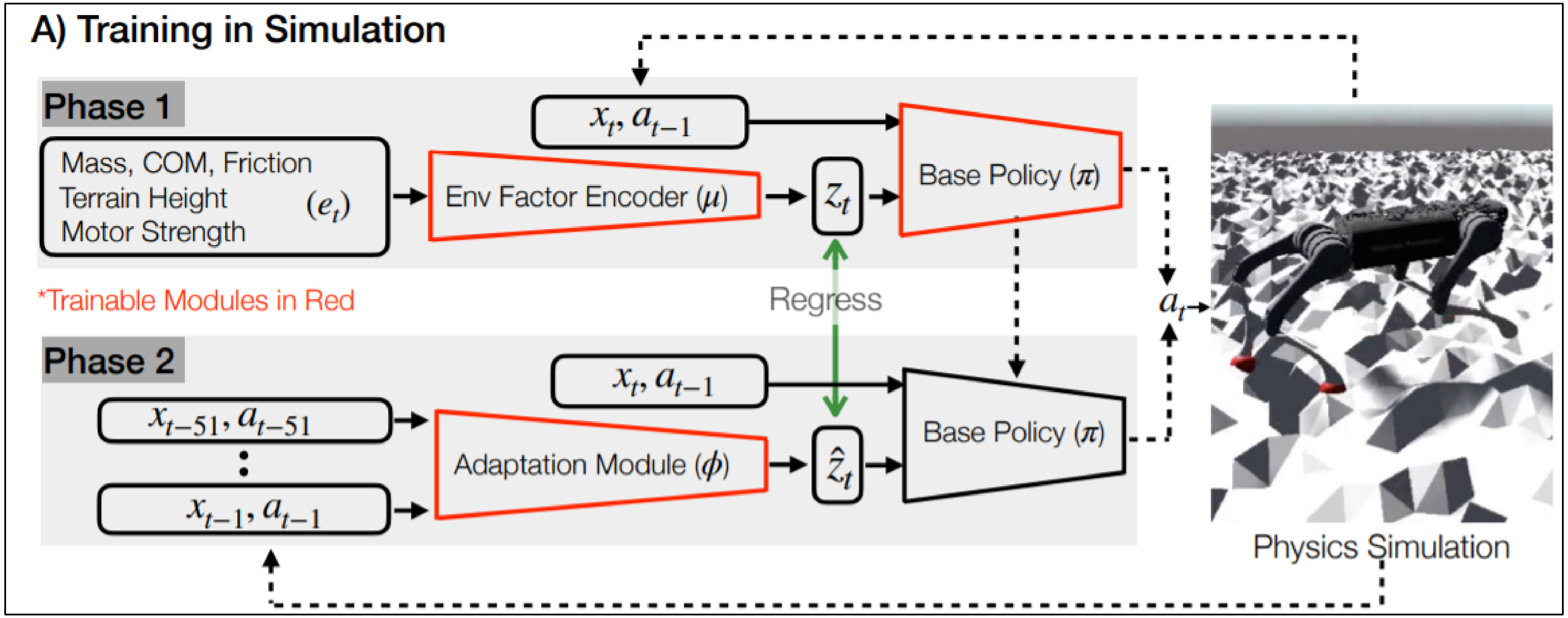
- Use reward function that encourages the agent to move forward with a maximum speed of 0.35m/s, and penalizes it for jerky and inefficient motions.

- Fixed curriculum learning was used on penalty coefficients and difficulty of perturbations such as mass, friction and motor strength.

1) Forward: $\min(v_x^t, 0.35)$
2) Lateral Movement and Rotation: $-\|v_y^t\|^2 - \|\omega_{\text{yaw}}^t\|^2$
3) Work: $-|\boldsymbol{\tau}^T \cdot (\mathbf{q}^t - \mathbf{q}^{t-1})|$
4) Ground Impact: $-\|\mathbf{f}^t - \mathbf{f}^{t-1}\|^2$
5) Smoothness: $-\|\boldsymbol{\tau}^t - \boldsymbol{\tau}^{t-1}\|^2$
6) Action Magnitude: $-\|\mathbf{a}^t\|^2$
7) Joint Speed: $-\|\dot{\mathbf{q}}^t\|^2$
8) Orientation: $-\|\boldsymbol{\theta}^t_{\texttt{roll, pitch}}\|^2$
9) Z Acceleration: $-\|v_z^t\|^2$
10) Foot Slip: $-\|\text{diag}(\mathbf{g}^t) \cdot \mathbf{v_f}^t\|^2$

Paper Review

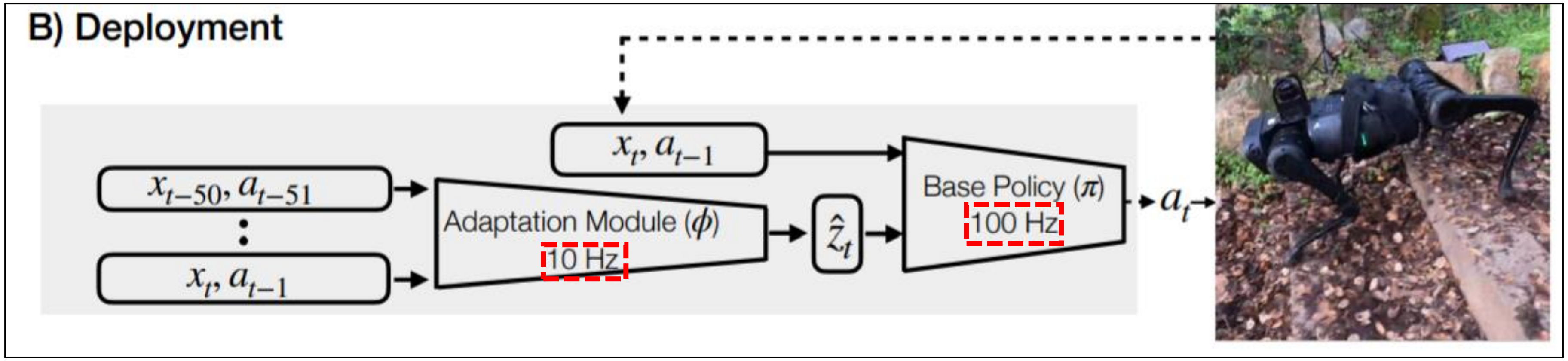
Adaptation module



- To train adaptation module, use the recent history of robot’s state and actions.
- Instead of predicting environment vector e_t, directly estimate the extrinsics z_t.
- Train adaptation module ϕ with on-policy data.

Paper Review

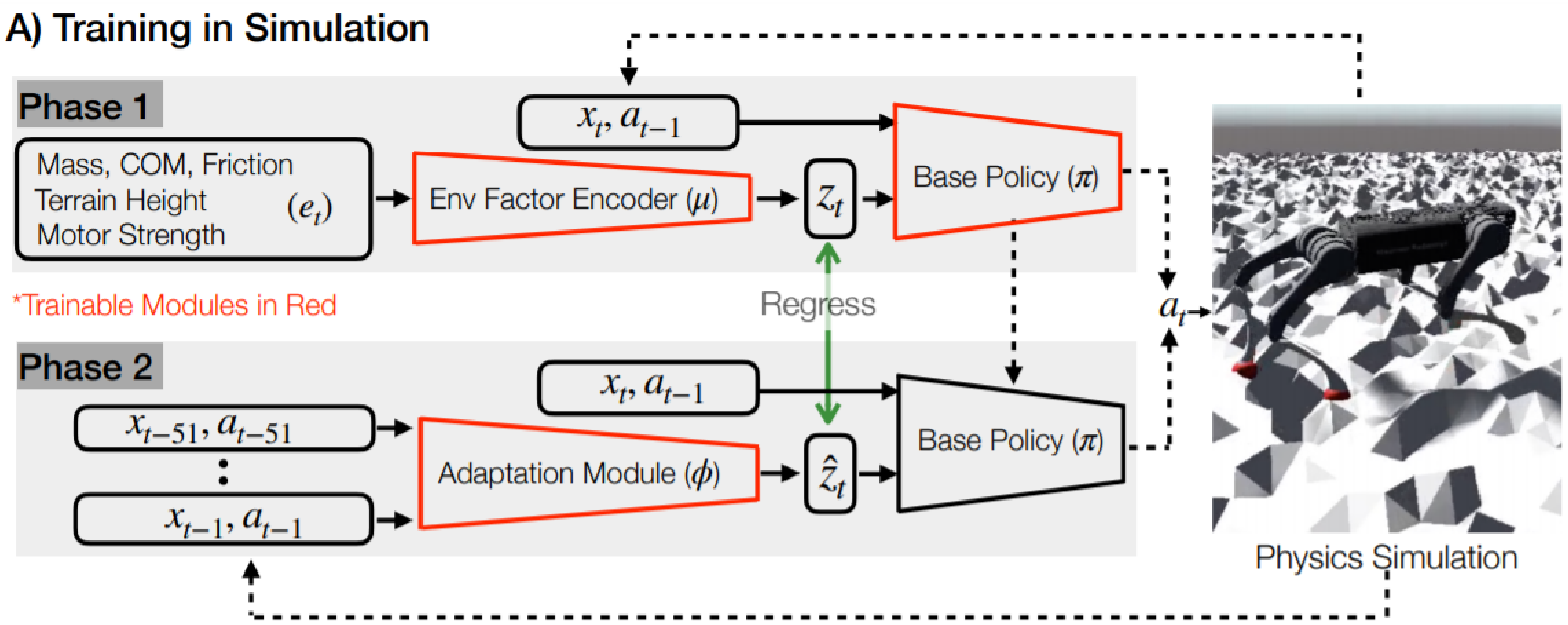
Asynchronous Deployment



- A base policy which directly takes the state and action history as input without decoupling them into the two modules
  (a) leads to unnatural gaits and poor performance in simulation, (b) can only run at 10Hz on the on-board compute, and (c) lacks the asynchronous design which is critical for a seamless deployment of RMA on the real robot without the need for any synchronization or calibration of the two subsystems.

Paper Review

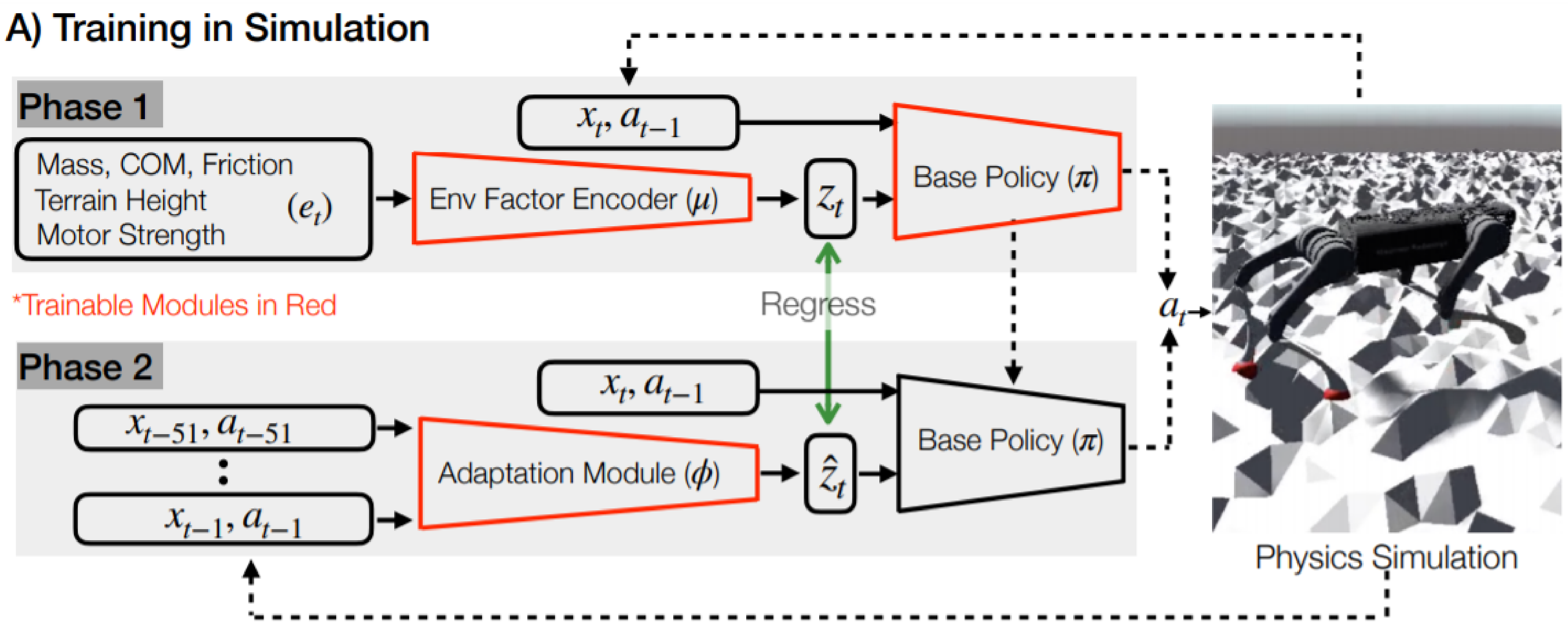
Network details



- Base Policy(π) : 3-layer multi-layer perceptron(MLP), dimension of hidden layer is 128.
- Env Factor Encoder(μ) : 3-layer MLP, hidden layer sizes : 256, 128
- Adaptation Module(ϕ) : 2-layer MLP + 3-layer 1-D Convolutional Neural Network(CNN)
  (Use 1-d CNN to capture temporal correlations in the input)

Paper Review

Training details



- They use PPO to train Base Policy and Env Factor Encoder jointly for 15,000 iterations each of which uses batch size 80,000 split into 4 mini-batches.
- This takes roughly 24 hours(1.2 billion steps) on an ordinary desktop machine with 1 GPU.
- Supervised learning was used to train adaptation module for 1,000 iterations each of which uses a batch size of 80,000 split up into 4 mini-batches.
- This takes roughly 3 hours(80 million steps).
- They use RaiSim for simulation.

Paper Review

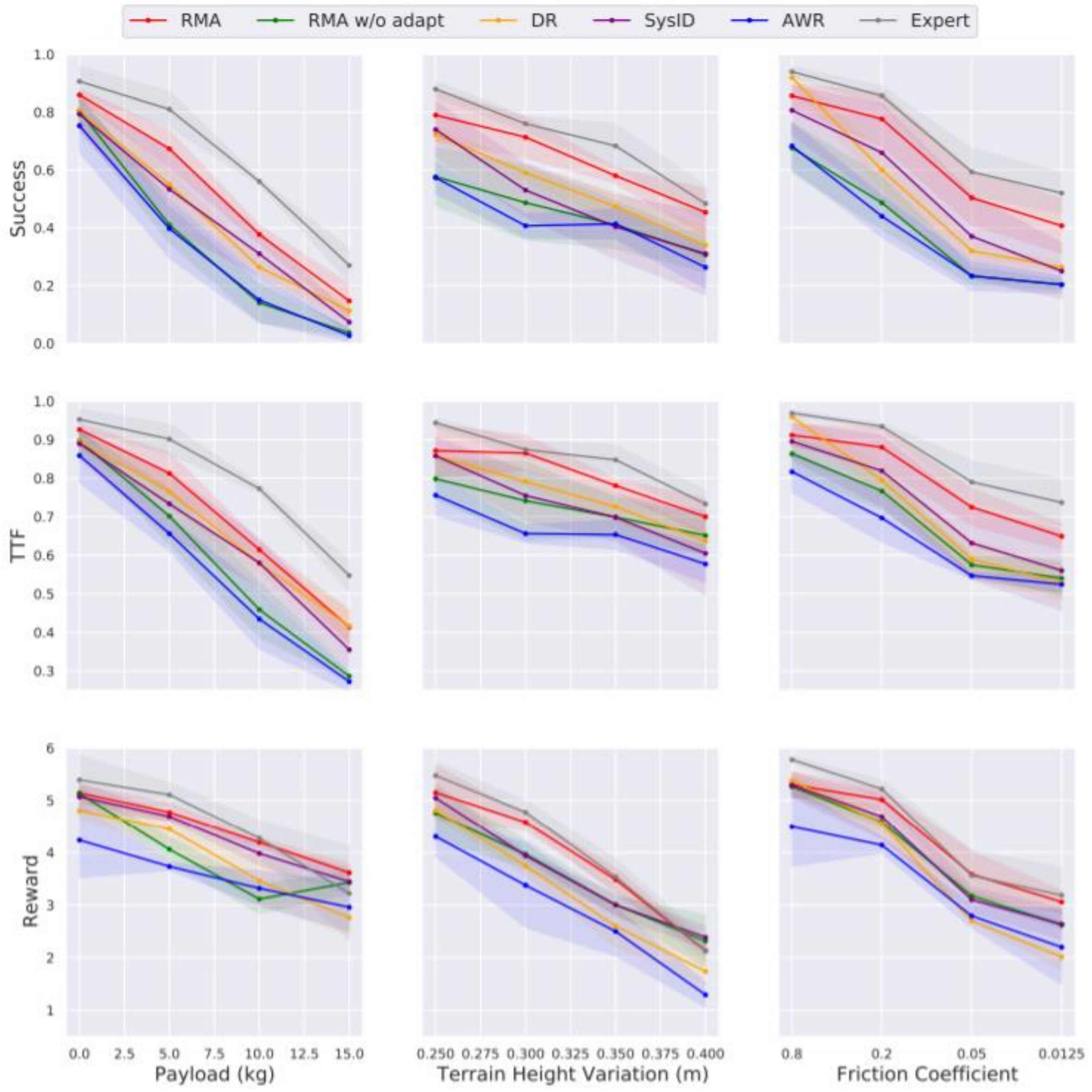
Result & Analysis

- The performance of RMA was compared to baseline controllers in simulation.
  - Baselines
    1. A1 Controller
    2. Robustness through Domain Randomization(Robust)
    3. Expert Adaptation Policy(Expert)
    4. RMA w/o Adaptation
    5. System Identification(SysID, directly estimates environment vector)
    6. Advantage Weighted Regression for Domain Adaptation(AWR)
  - Environment
    - Resample environment parameter with a resampling probability of 0.01 per step

	Success (%)	TTF	Reward	Distance (m)	Samples	Torque	Smoothness	Ground Impact
Robust [52, 40]	62.4	0.80	4.62	1.13	0	527.59	122.50	4.20
SysID [57]	56.5	0.74	4.82	1.17	0	565.85	149.75	4.03
AWR [41]	41.7	0.65	4.17	0.95	40k	599.71	162.60	4.02
RMA w/o Adapt	52.1	0.75	4.72	1.15	0	524.18	106.25	4.55
RMA	73.5	0.85	5.22	1.34	0	500.00	92.85	4.27
Expert	76.2	0.86	5.23	1.35	0	485.07	85.56	3.90

Paper Review

Result & Analysis



Paper Review

- Comparison to A1 Controller in Indoor Condition

Comparison to A1's built in controller

Paper Review

- Comparison to RMA w/o Adaptation in Indoor Condition

Comparison to RMA w/o Adaptation

Paper Review

- RMA in the wild

RMA in the wild

Same policy was deployed in all experiments
with no real world fine-tuning

Q & A

Thank you for your attention