Sim-to-Real Learning of All Common Bipedal Gaits via Periodic Reward Composition

Hansol Kang, Combined Course, 9th semester

Robotics Innovatory

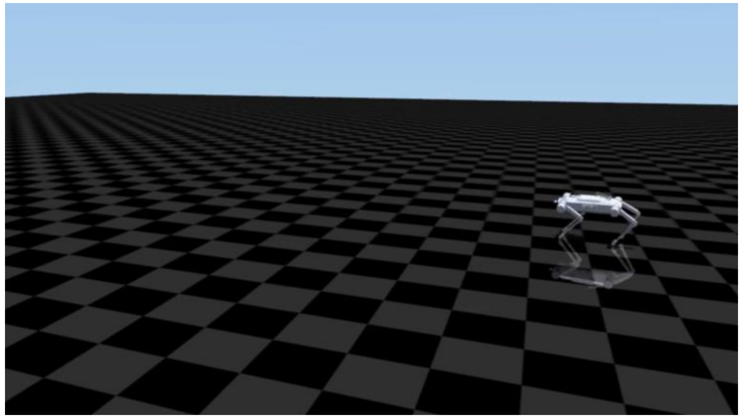
http://mecha.skku.ac.kr

Contents

- Paper Review
- Current Work

Introduction

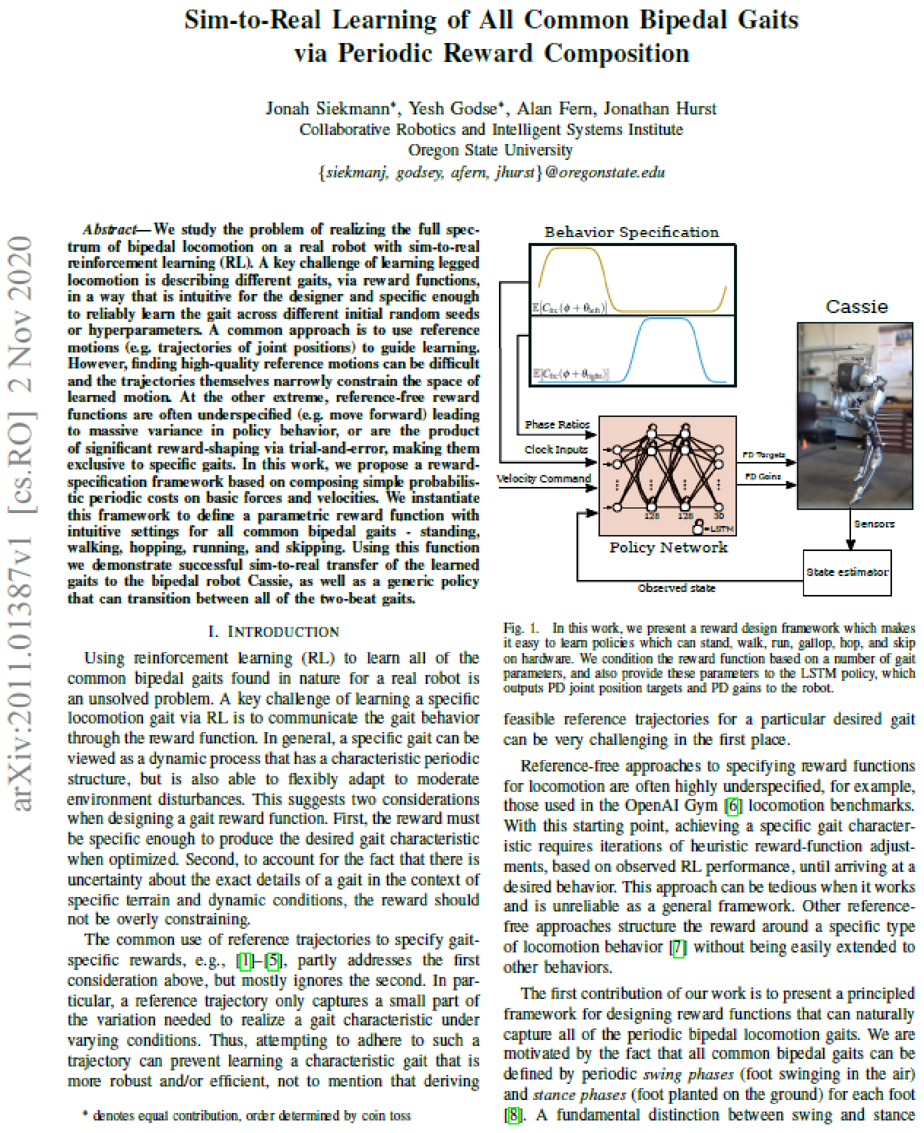
- Quadruped robot control via RL



Learning to Walk via Deep Reinforcement Learning. Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, Sergey Levine. Robotics: Science and Systems (RSS). 2019.
Learning agile and dynamic motor skills for legged robots, Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, Marco Hutter, Science Robotics, 2019

Paper Review

- Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition



Sim-to-Real Learning of All Common Bipedal Gaits via Periodic Reward Composition

Jonah Siekmann*, Yesh Godse*, Alan Fern, Jonathan Hurst
Collaborative Robotics and Intelligent Systems Institute
Oregon State University
{*siekmanj, godsey, afern, jhurst*}@*oregonstate.edu*

***Abstract*—We study the problem of realizing the full spectrum of bipedal locomotion on a real robot with sim-to-real reinforcement learning (RL). A key challenge of learning legged locomotion is describing different gaits, via reward functions, in a way that is intuitive for the designer and specific enough to reliably learn the gait across different initial random seeds or hyperparameters. A common approach is to use reference motions (e.g. trajectories of joint positions) to guide learning. However, finding high-quality reference motions can be difficult and the trajectories themselves narrowly constrain the space of learned motion. At the other extreme, reference-free reward functions are often underspecified (e.g. move forward) leading to massive variance in policy behavior, or are the product of significant reward-shaping via trial-and-error, making them exclusive to specific gaits. In this work, we propose a reward-specification framework based on composing simple probabilistic periodic costs on basic forces and velocities. We instantiate this framework to define a parametric reward function with intuitive settings for all common bipedal gaits - standing, walking, hopping, running, and skipping. Using this function we demonstrate successful sim-to-real transfer of the learned gaits to the bipedal robot Cassie, as well as a generic policy that can transition between all of the two-beat gaits.**

I. Introduction

Using reinforcement learning (RL) to learn all of the common bipedal gaits found in nature for a real robot is an unsolved problem. A key challenge of learning a specific locomotion gait via RL is to communicate the gait behavior through the reward function. In general, a specific gait can be viewed as a dynamic process that has a characteristic periodic structure, but is also able to flexibly adapt to moderate environment disturbances. This suggests two considerations when designing a gait reward function. First, the reward must be specific enough to produce the desired gait characteristic when optimized. Second, to account for the fact that there is uncertainty about the exact details of a gait in the context of specific terrain and dynamic conditions, the reward should not be overly constraining.

The common use of reference trajectories to specify gait-specific rewards, e.g., [1]–[5], partly addresses the first consideration above, but mostly ignores the second. In particular, a reference trajectory only captures a small part of the variation needed to realize a gait characteristic under varying conditions. Thus, attempting to adhere to such a trajectory can prevent learning a characteristic gait that is more robust and/or efficient, not to mention that deriving feasible reference trajectories for a particular desired gait can be very challenging in the first place.

Fig. 1. In this work, we present a reward design framework which makes it easy to learn policies which can stand, walk, run, gallop, hop, and skip on hardware. We condition the reward function based on a number of gait parameters, and also provide these parameters to the LSTM policy, which outputs PD joint position targets and PD gains to the robot.

Reference-free approaches to specifying reward functions for locomotion are often highly underspecified, for example, those used in the OpenAI Gym [6] locomotion benchmarks. With this starting point, achieving a specific gait characteristic requires iterations of heuristic reward-function adjustments, based on observed RL performance, until arriving at a desired behavior. This approach can be tedious when it works and is unreliable as a general framework. Other reference-free approaches structure the reward around a specific type of locomotion behavior [7] without being easily extended to other behaviors.

The first contribution of our work is to present a principled framework for designing reward functions that can naturally capture all of the periodic bipedal locomotion gaits. We are motivated by the fact that all common bipedal gaits can be defined by periodic *swing phases* (foot swinging in the air) and *stance phases* (foot planted on the ground) for each foot [8]. A fundamental distinction between swing and stance

[*] denotes equal contribution, order determined by coin toss

- 2020.11. arXiv

- 2021 ICRA Submitted paper

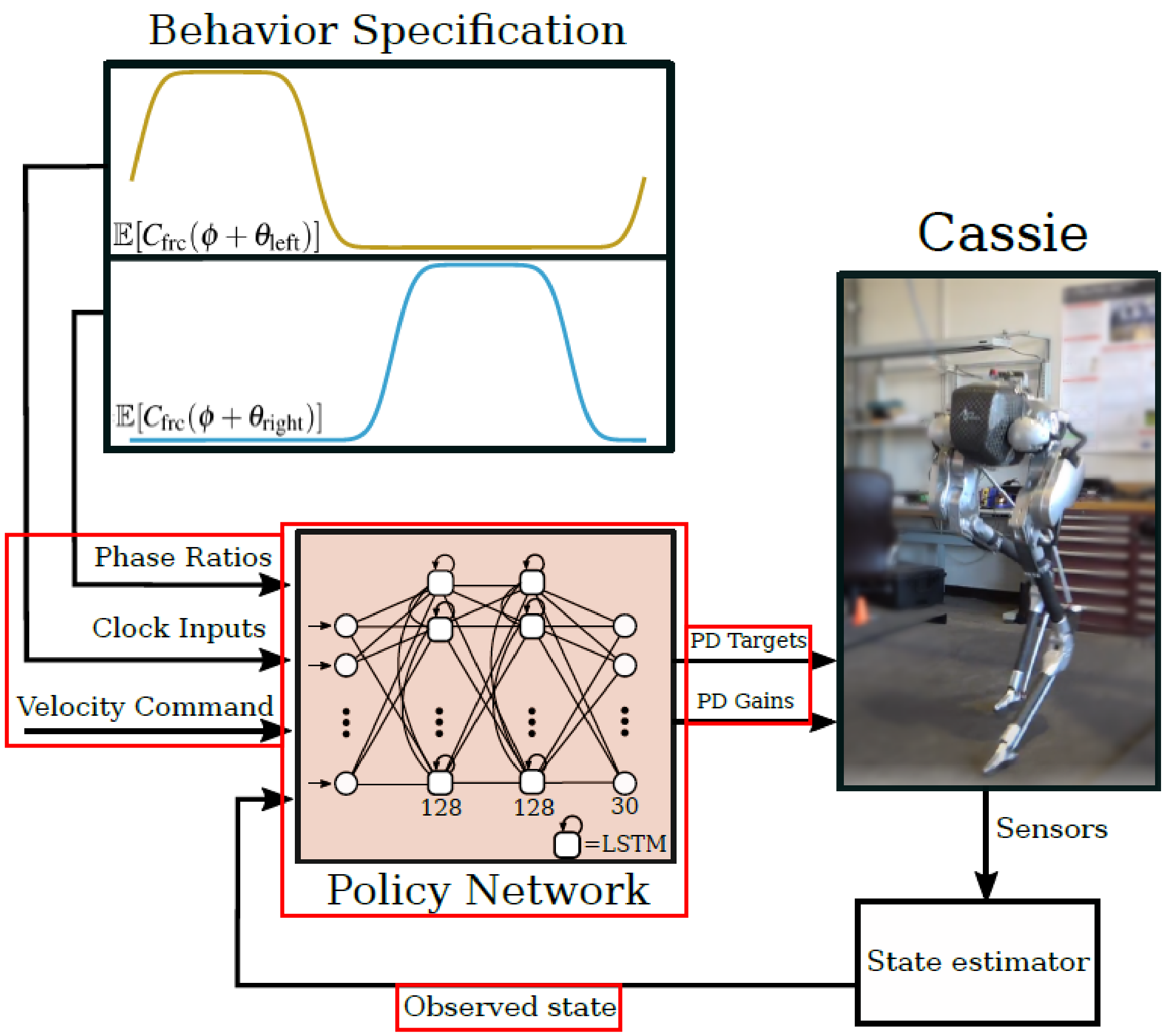
- Jonah Siekmann*, Yesh Godse*, Alan Fern, Jonathan Hurst

- Oregon State University

- Propose a reward function dependent on a periodic signal, and through this, the agent is possible to learn to walk in various gait according to the user's command, and natural gait transition is possible.

Paper Review

- Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition



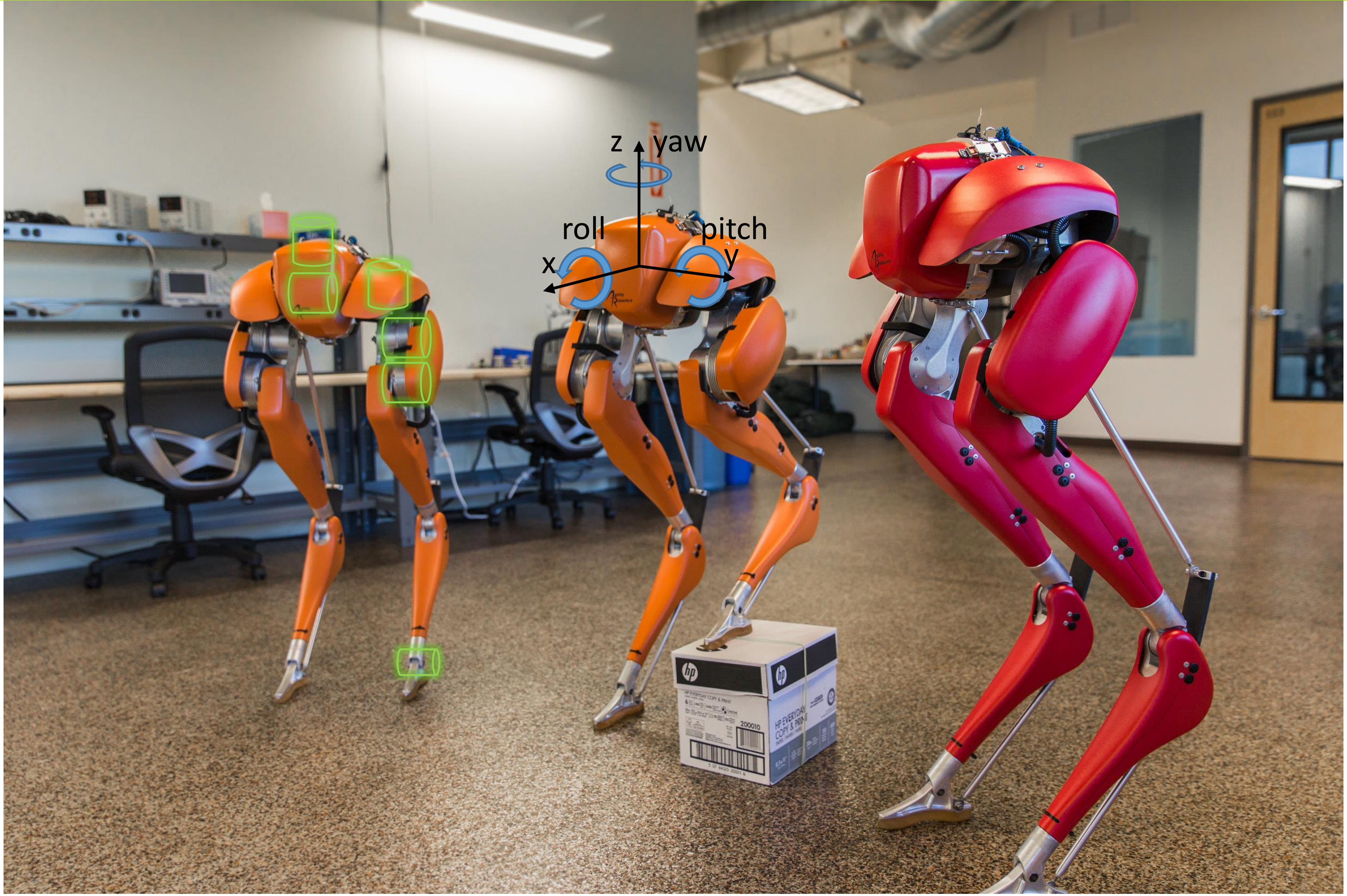
Input : Phase Ratio($\boldsymbol{r}$), Clock Inputs($Clock$),
Velocity Command($\boldsymbol{C}$), Observation($\boldsymbol{O}$)

Long Short-Term Memory
2 Hidden Layers(128 Hidden nodes each)

Output : Target Joint Position(10)
P, D Gain(10 × 2)

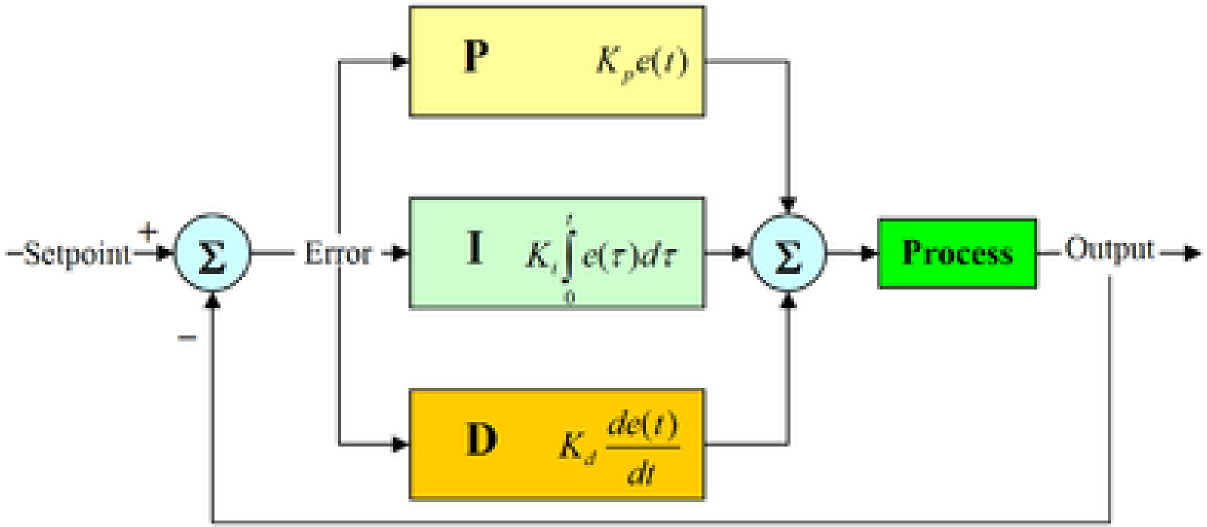
Algorithm : PPO

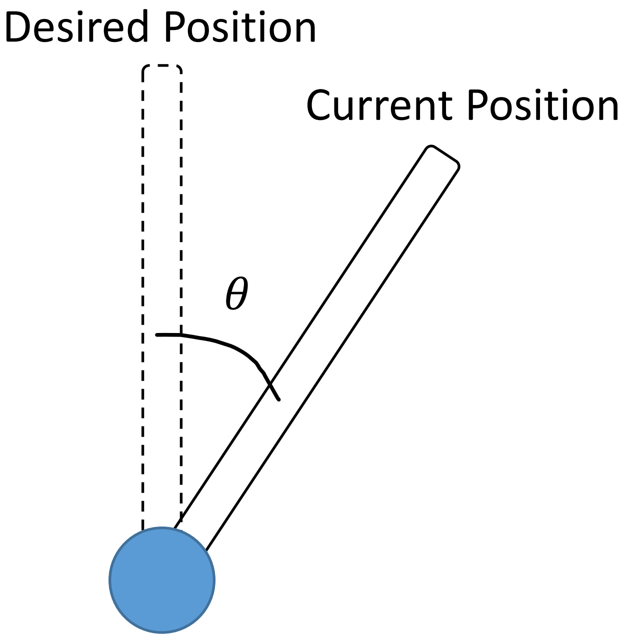
Preliminaries for robot control



https://www.agilityrobotics.com/robots

Preliminaries for robot control





$$Manipulated\ Variable = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de}{dt}$$

Paper Review

Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition

$$R(\boldsymbol{s},\phi) = \beta + \sum_i R_i(\boldsymbol{s},\phi)$$

Normalized time period ϕ

$$R_i(\boldsymbol{s},\phi) = c_i \cdot I_i(\phi) \cdot q_i(\boldsymbol{s})$$

Phase coefficient c_i
Phase indicator $I_i(\phi)$
Phase reward measurement $q_i(\boldsymbol{s})$

$$\mathbb{E}[R(\boldsymbol{s},\phi)] = \sum_i^n c_i \cdot \mathbb{E}[I_i(\phi)] \cdot q_i(\boldsymbol{s}) + \beta$$

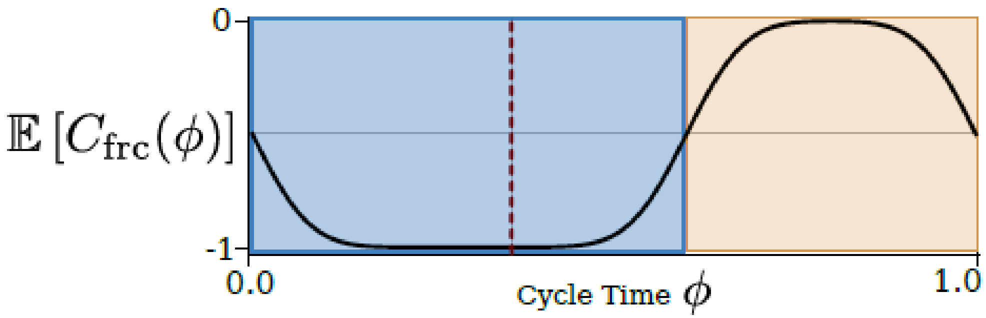
$$I_i(\phi) = \begin{cases} 0, & Stance \\ 1, & Swing \end{cases}$$

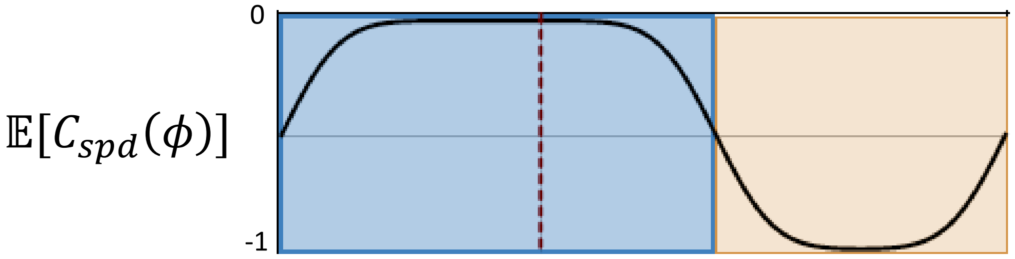
$$\mathbb{E}\left[C_{frc}(\phi)\right] = c_{swing\ frc} \cdot \mathbb{E}\left[I_{swing\ frc}(\phi)\right] + c_{stance\ frc} \cdot \mathbb{E}[I_{stance\ frc}(\phi)]$$

$$c_{stance\ frc} = 0,\ c_{swing\ frc} = -1$$

Paper Review

Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition





- Unipedal

$$\mathbb{E}[R_{unipedal}(\mathbf{s},\phi)] = \mathbb{E}\left[C_{frc}(\phi)\right] \cdot q_{frc}(\mathbf{s}) + \mathbb{E}\left[C_{spd}(\phi)\right] \cdot q_{spd}(\mathbf{s})$$

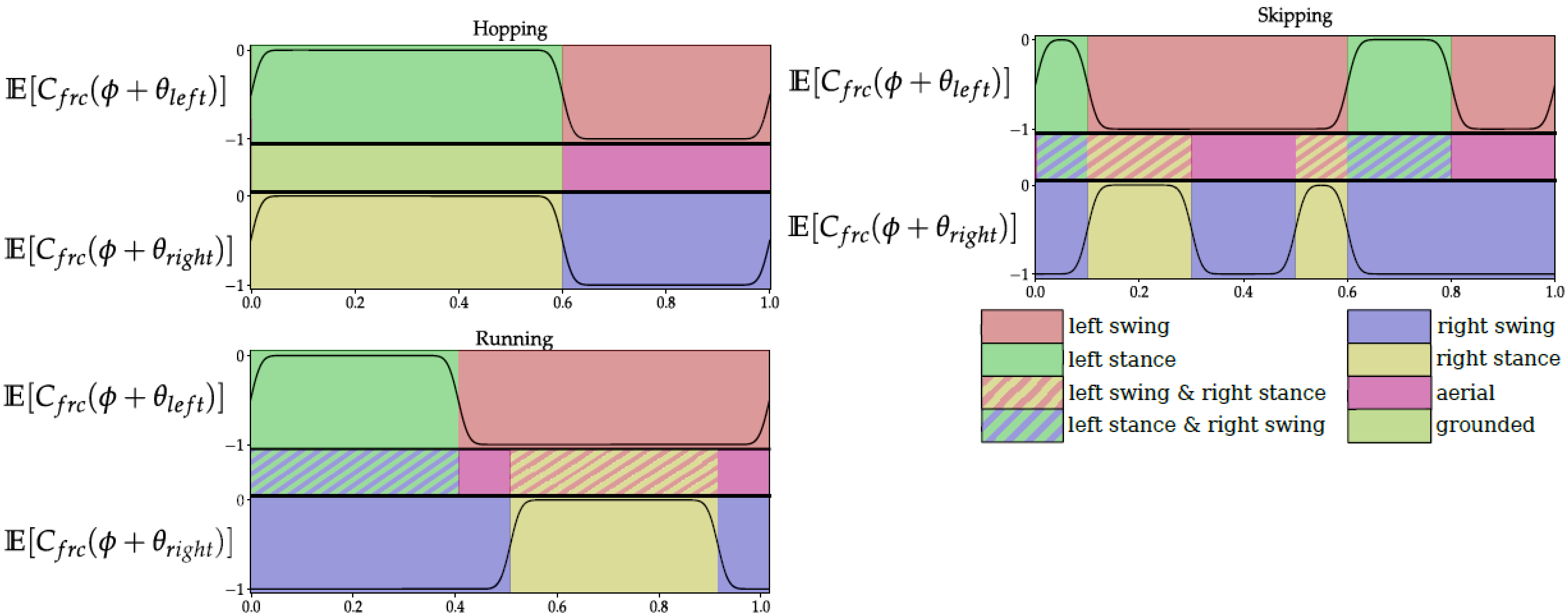
- Biped

$$\begin{aligned}\mathbb{E}\left[R_{biped}(\mathbf{s},\phi)\right] &= \mathbb{E}\left[C_{frc}\left(\phi+\theta_{left}\right)\right] \cdot q_{left\ frc}(\mathbf{s}) + \mathbb{E}\left[C_{spd}\left(\phi+\theta_{left}\right)\right] \cdot q_{left\ spd}(\mathbf{s}) \\ &\quad + \mathbb{E}\left[C_{frc}\left(\phi+\theta_{right}\right)\right] \cdot q_{right\ frc}(\mathbf{s}) + \mathbb{E}\left[C_{spd}\left(\phi+\theta_{right}\right)\right] \cdot q_{right\ spd}(\mathbf{s})\end{aligned}$$

⋮

Paper Review

- Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition



Hopping

Running

Skipping

Paper Review

Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition

- Total Reward Function

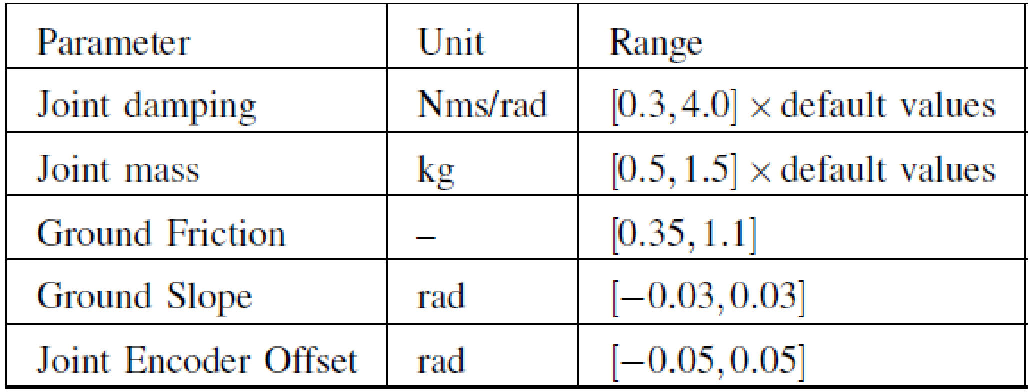
$$R(\boldsymbol{s},\phi) = \beta + \sum_i R_i(\boldsymbol{s},\phi)$$

$$\mathbb{E}[R(\boldsymbol{s},\phi)] = \mathbb{E}\big[R_{biped}(\boldsymbol{s},\phi)\big] + R_{cmd}(\boldsymbol{s}) + R_{smooth}(\boldsymbol{s}) + \beta$$

$$R_{cmd}(\boldsymbol{s}) = (-1)\cdot q_{\dot{x}}(\boldsymbol{s}) + (-1)\cdot q_{\dot{y}}(\boldsymbol{s}) + (-1)\cdot q_{orientation}(\boldsymbol{s})$$

$$\begin{aligned} R_{smooth}(\boldsymbol{s}) = (-1)\cdot q_{action\_diff}(\boldsymbol{s}) + (-1)\cdot q_{torque}(\boldsymbol{s}) \\ +(-1)\cdot q_{pelvis\_acc}(\boldsymbol{s}) \end{aligned}$$

- Dynamic Randomization



Parameter	Unit	Range
Joint damping	Nms/rad	$[0.3, 4.0] \times$ default values
Joint mass	kg	$[0.5, 1.5] \times$ default values
Ground Friction	–	$[0.35, 1.1]$
Ground Slope	rad	$[-0.03, 0.03]$
Joint Encoder Offset	rad	$[-0.05, 0.05]$

TABLE I. The ranges for randomization of several dynamics parameters during training. We use a uniform distribution over the given ranges for all listed parameters.

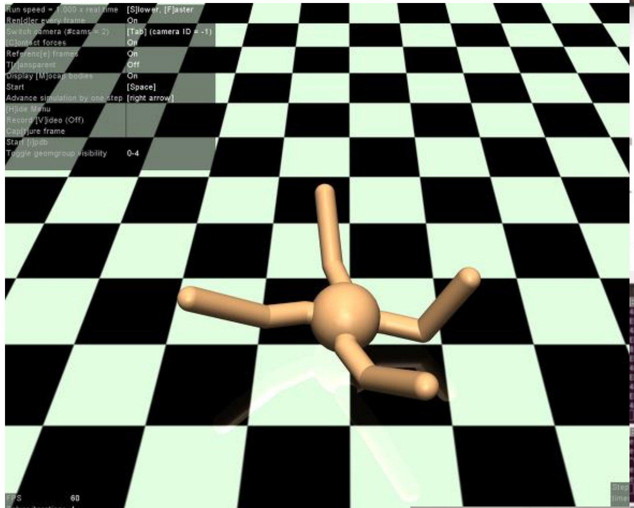
Paper Review

- Sim-to-Real Leaning of All Common Bipedal Gaits via Periodic Reward Composition

In this work, we introduce a framework which makes it possible to construct reward functions for learning locomotion behaviors without the use of reference trajectories.

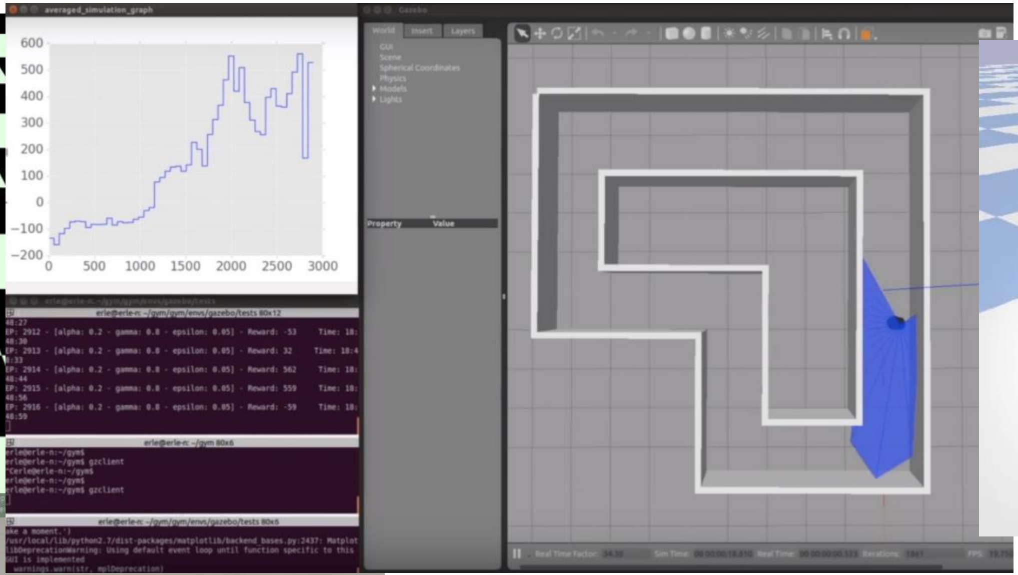
Simulator

- MuJoCo? Gazebo? Pybullet? Unity? RaiSim?



```
Run speed = 1.000 x real time      [S]lower, [F]aster
Ren[d]er every frame               On
Switch camera (#cams = 2)          [Tab] (camera ID = -1)
[C]ontact forces                   On
Referenc[e] frames                 On
T[r]ansparent                      Off
Display [M]ocap bodies             On
Start                              [Space]
Advance simulation by one step     [right arrow]
[H]ide Menu
Record [V]ideo (Off)
Cap[t]ure frame
Start [i]pdb
Toggle geomgroup visibility        0-4
```

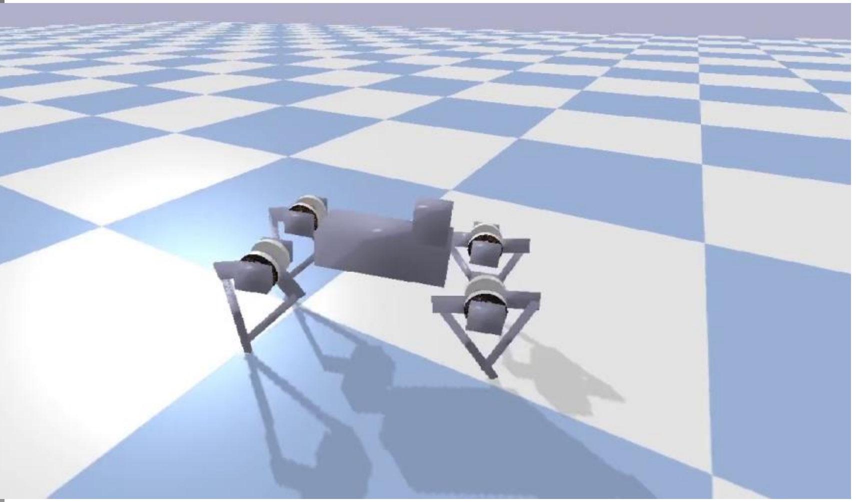
```
FPS             60
Solver iterations  4
```

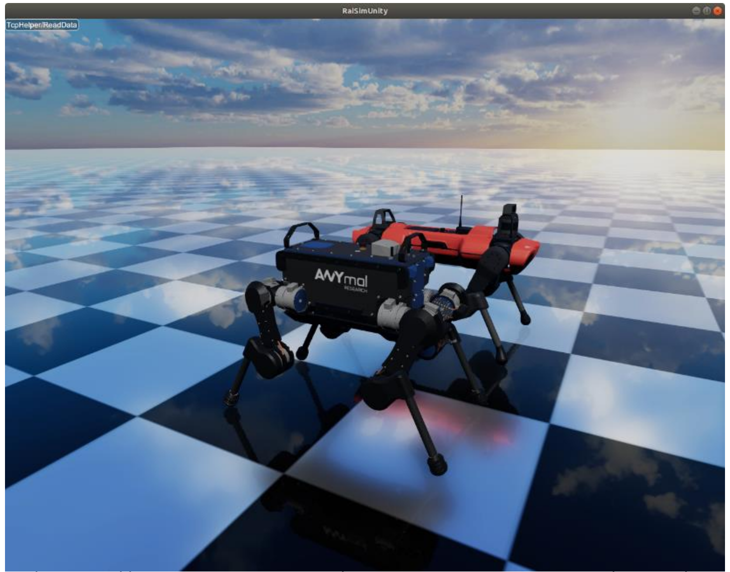


```
erle@erle-n: ~/gym/gym/envs/gazebo/tests 80x12
48:27
EP: 2912 - [alpha: 0.2 - gamma: 0.8 - epsilon: 0.05] - Reward: -53     Time: 10:
48:30
EP: 2913 - [alpha: 0.2 - gamma: 0.8 - epsilon: 0.05] - Reward: 32     Time: 10:4
8:33
EP: 2914 - [alpha: 0.2 - gamma: 0.8 - epsilon: 0.05] - Reward: 562     Time: 10:
48:44
EP: 2915 - [alpha: 0.2 - gamma: 0.8 - epsilon: 0.05] - Reward: 559     Time: 10:
48:56
EP: 2916 - [alpha: 0.2 - gamma: 0.8 - epsilon: 0.05] - Reward: -59     Time: 10:
48:59
```

```
erle@erle-n: ~/gym 80x6
erle@erle-n:~/gym$
erle@erle-n:~/gym$ gzclient
^Cerle@erle-n:~/gym$
erle@erle-n:~/gym$
erle@erle-n:~/gym$ gzclient
```

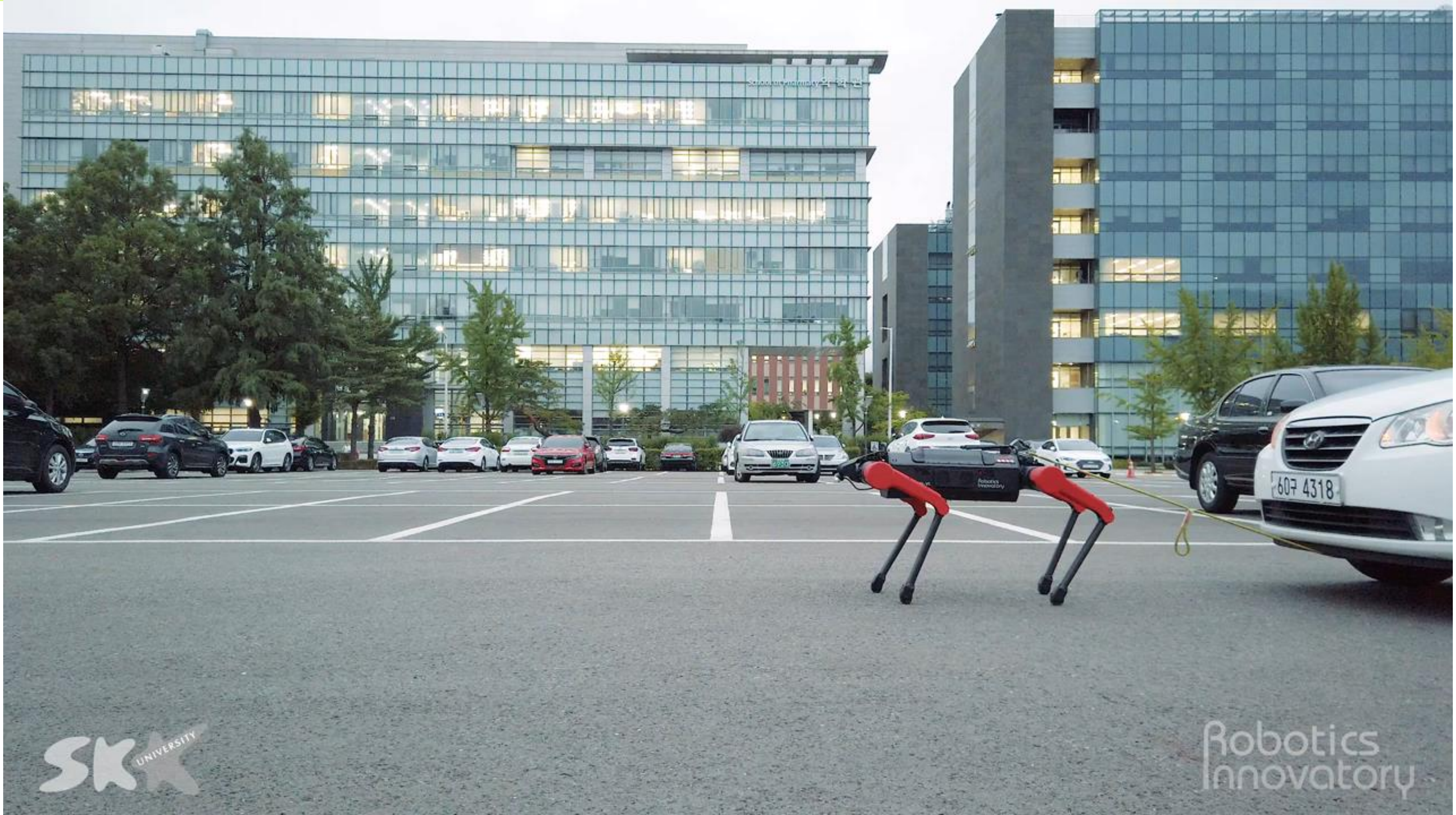
```
erle@erle-n: ~/gym/gym/envs/gazebo/tests 80x6
ake a moment.')
/usr/local/lib/python2.7/dist-packages/matplotlib/backend_bases.py:2437: Matplot
libDeprecationWarning: Using default event loop until function specific to this
GUI is implemented
  warnings.warn(str, mplDeprecation)
```





http://www.mujoco.org/ , https://pybullet.org/wordpress/, https://reinforcement-learning-kr.github.io/2018/08/23/8_implement/ , https://ai-mrkogao.github.io/reinforcement%20learning/ROSRL/, https://raisim.com/

Quadruped Robot, AiDIN-VI

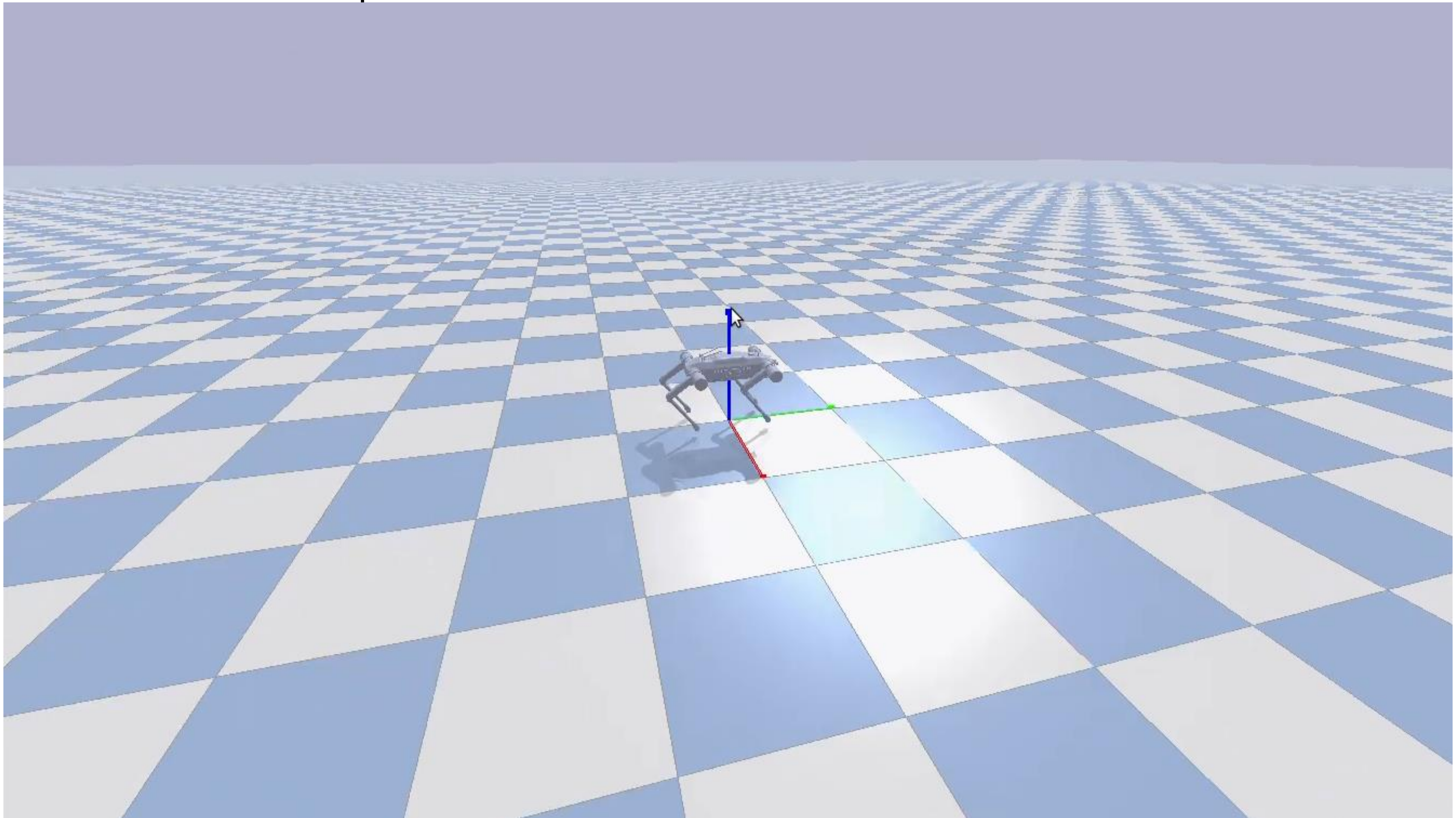


Various Gaits of Quadruped

https://www.youtube.com/watch?v=8qV5pdO_X8U

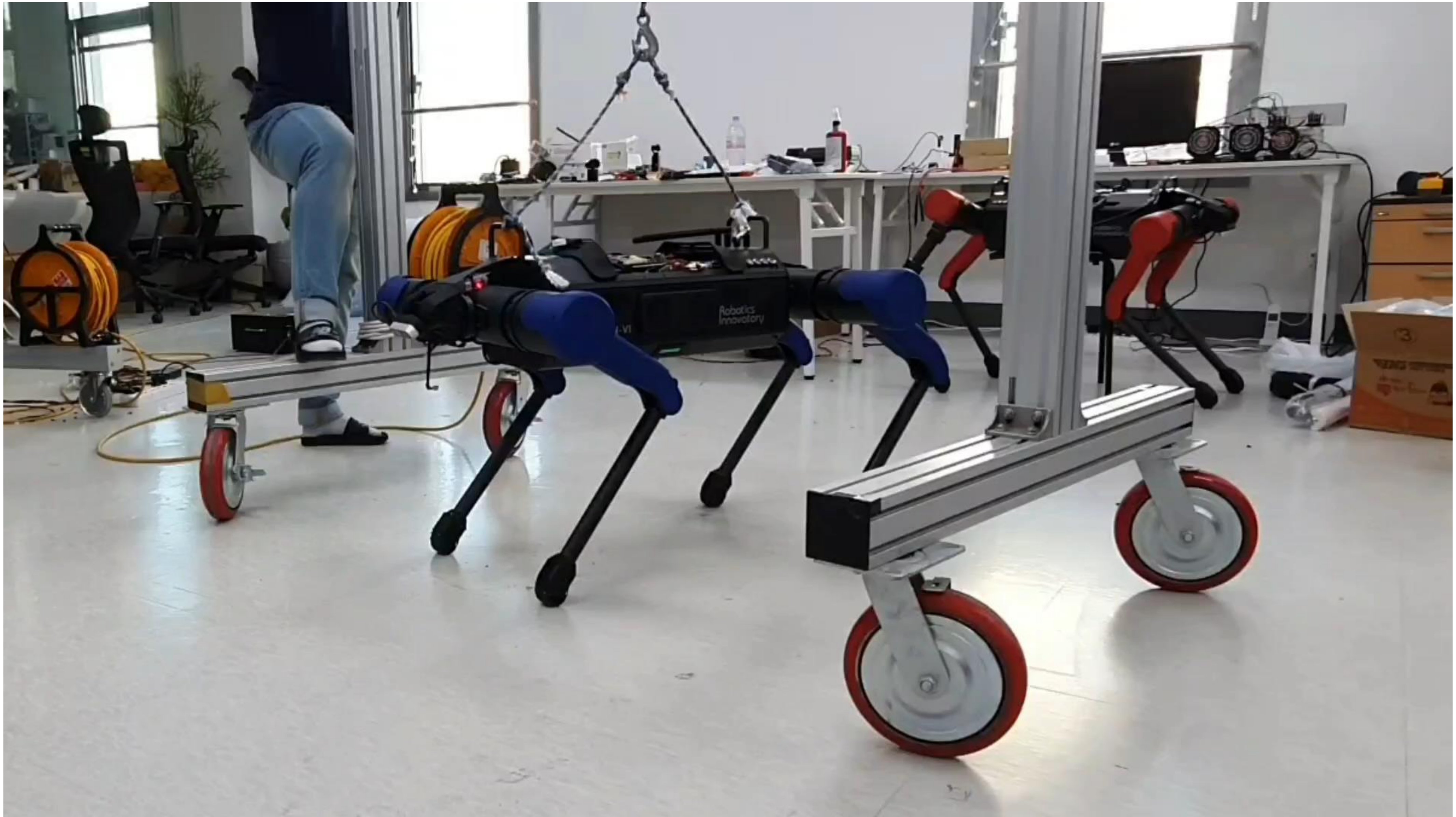
Current Work

- 12-DOF AiDIN-VI + Task Space + Periodic Reward



Current Work

- Sim-to-Real Test



Q & A

Thank you for your attention