The Ohio State University

ME7385 Final Project

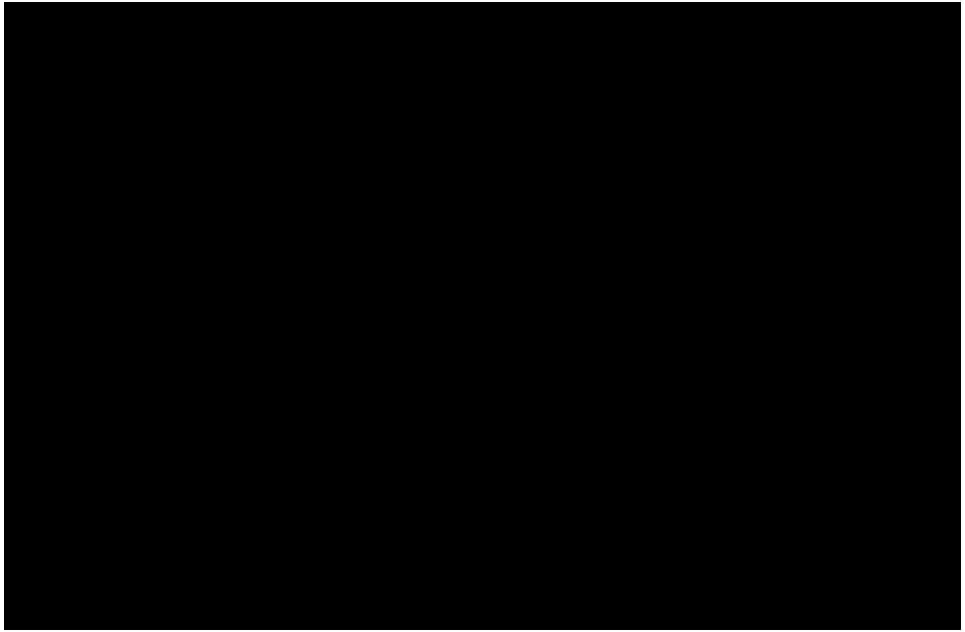
Autumn ‘21

Periodic Motion Analysis of ‘HalfCheetah’ Reinforcement Learned Running Gait

Presentation Outline

Topics

- Motivation
- ‘HalfCheetah’ Model
- MuJoCo & OpenAI Gym
- Reinforcement Learning
- Periodic Motion & Analysis
- Future Work



Motivation

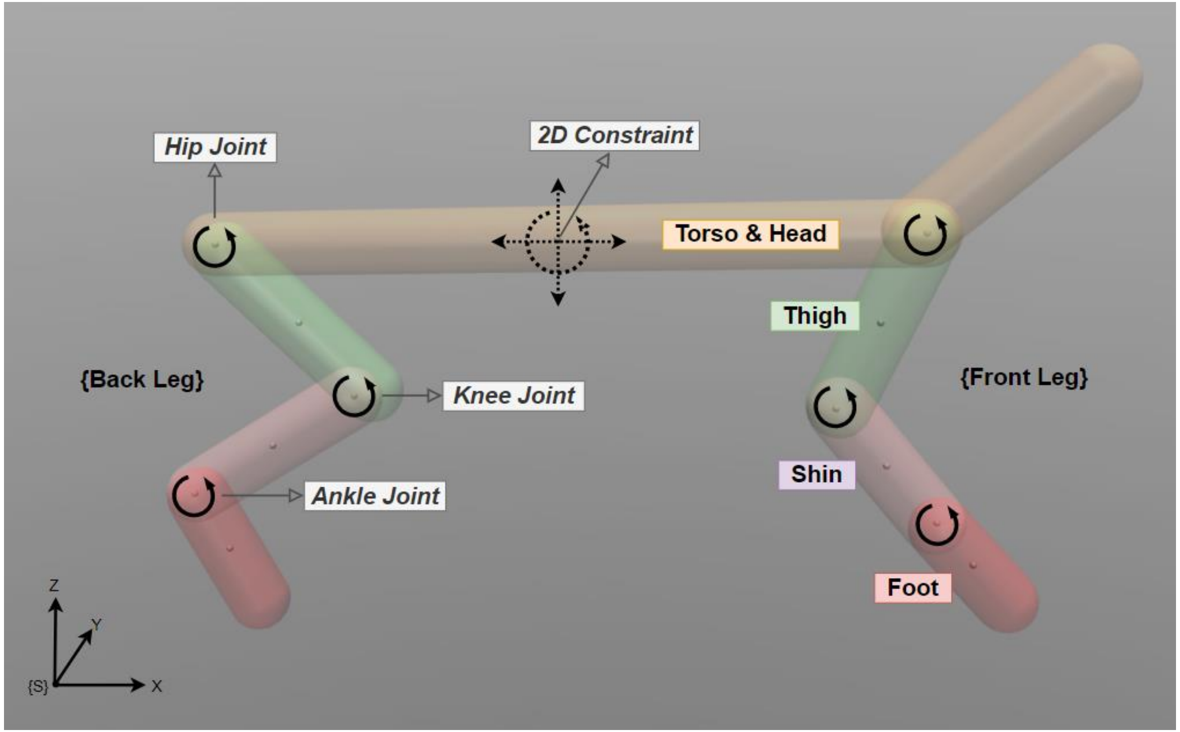
Learning Objectives

- Get first experience with MuJoCo & OpenAI Gym (in Linux)
- Review policy-based RL algorithm
- Perform analysis of “black box” RL controller
- Combine domain knowledge to improve ‘Learning methods’

HalfCheetah

Overview

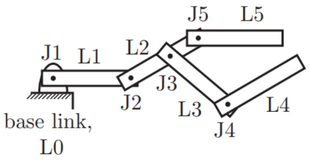
- 6 controlled leg joints
- 2D constraints
- Simple bodies
- Baseline model

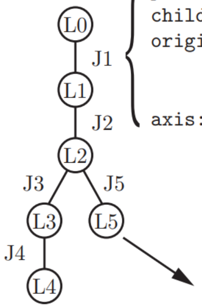


HalfCheetah Modeling

XML Robot Model

- URDF (Unified Robot Description Format), **MJCF**
- Kinematic tree
- Other model & environment objects





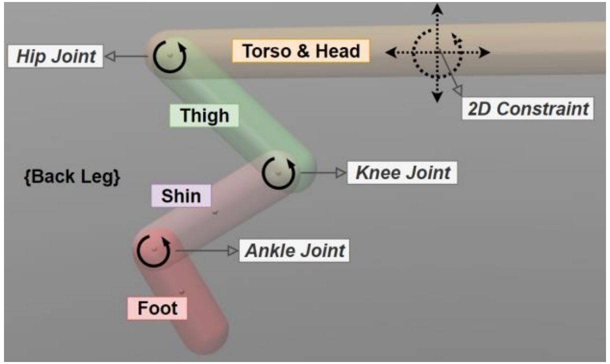
`parent:` J1's parent link, L0
`child:` J1's child link, L1
`origin:` the x–y–z and roll–pitch–yaw coords of the L1 frame relative to the L0 frame when J1 is zero
`axis:` the x–y–z unit vector along the rotation axis in the L1 frame

Kinematic Definition
- Joint relations
- Joint frames

`mass:` L5's mass
`origin:` the x–y–z and roll–pitch–yaw coords of a frame at the center of mass of L5, relative to the L5 frame
`inertia:` six unique entries of inertia matrix in the `origin` frame

[???]

Dynamics Info
- COM frame
- Mass & Inertia




```xml
<!-- Cheetah Model Construction -->
<worldbody>
  <!-- GROUND, etc. -->
  <light cutoff="100" diffuse="1 1 1" dir="-0 0 -1.3" directional="true" exponent="1" pos="0 0 1.3" specular=".1 .1 .1"/>
  <geom conaffinity="1" condim="3" material="MatPlane" name="floor" pos="0 0 0" rgba="0.8 0.9 0.8 1" size="40 40 40"/>
 <!-- TORSE / HEAD -->
 <body name="torso" pos="0 0 .7">
   <camera name="track" mode="trackcom" pos="0 -3 0.3" xyaxes="1 0 0 0 0 1"/>
   <joint armature="0" axis="1 0 0" damping="0" limited="false" name="rootx" pos="0 0 0" stiffness="0" type="slide"/>
   <joint armature="0" axis="0 0 1" damping="0" limited="false" name="rootz" pos="0 0 0" stiffness="0" type="slide"/>
   <joint armature="0" axis="0 1 0" damping="0" limited="false" name="rooty" pos="0 0 0" stiffness="0" type="hinge"/>
   <geom fromto="-.5 0 0 .5 0 0" name="torso" size="0.046" type="capsule"/>
   <geom axisangle="0 1 0 .87" name="head" pos=".6 0 .1" size="0.046 .15" type="capsule"/>
   <!-- BACK LEG -->
   <!-- Thigh (Hip Joint) -->
   <body name="bthigh" pos="-.5 0 0">
     <joint axis="0 1 0" damping="6" name="bthigh" pos="0 0 0" range="-.52 1.05" stiffness="240" type="hinge"/>
     <geom axisangle="0 1 0 -3.8" name="bthigh" pos=".1 0 -.13" size="0.046 .145" type="capsule"/>
     <!-- Shin (Knee Joint) -->
     <body name="bshin" pos=".16 0 -.25">
       <joint axis="0 1 0" damping="4.5" name="bshin" pos="0 0 0" range="-.785 .785" stiffness="180" type="hinge"/>
       <geom axisangle="0 1 0 -2.03" name="bshin" pos="-.14 0 -.07" rgba="0.9 0.6 0.6 1" size="0.046 .15" type="capsule"/>
       <!-- Foot (Ankle Joint) -->
      <body name="bfoot" pos="-.28 0 -.14">
        <joint axis="0 1 0" damping="3" name="bfoot" pos="0 0 0" range="-.4 .785" stiffness="120" type="hinge"/>
        <geom axisangle="0 1 0 -.27" name="bfoot" pos=".03 0 -.097" rgba="0.9 0.6 0.6 1" size="0.046 .094" type="capsule"/>
       </body>
     </body>
  </body>
```

Dynamics

Dynamics Model

- Open chain, no closed loops
- Lagrangian, Recursive Newton-Euler

$$\underbrace{M(\theta)\ddot{\theta}}_{\text{Mass/Inertia Matrix}} + \underbrace{C(\theta,\dot{\theta})\dot{\theta}}_{\text{Coriolis Matrix}} + \underbrace{g(\theta)}_{\text{Gravity}} = \underbrace{\tau \ldots}_{\text{Torques \& Others}}$$

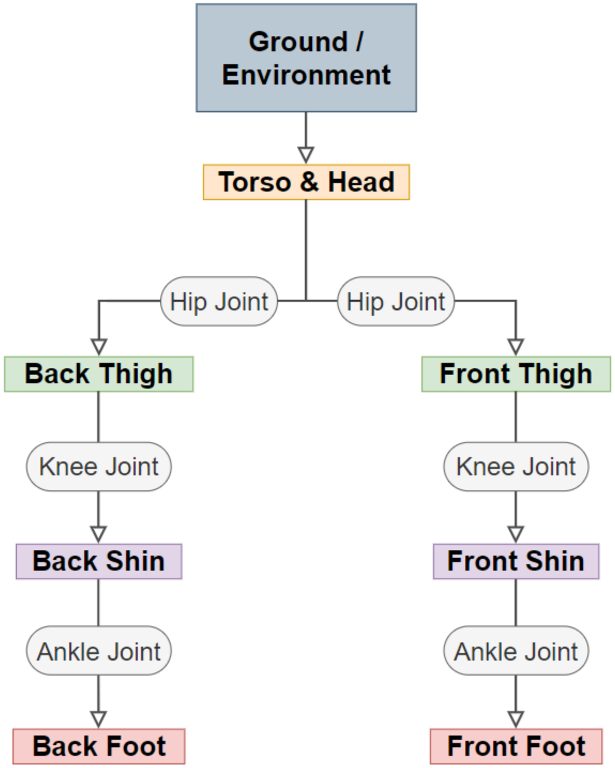
Mass/Inertia Matrix
- From total KE
- Positive-definite

Coriolis Matrix
- Centripetal/Coriolis
- Christoffel Symbols
- Quadratic & trigonometric terms

Gravity
- From total gravitational PE

Torques & Others
- Joint torques
- Friction/damping
- Force control

Kinematic Tree



MuJoCo [1]

Multi-Joint Dynamics with Contact

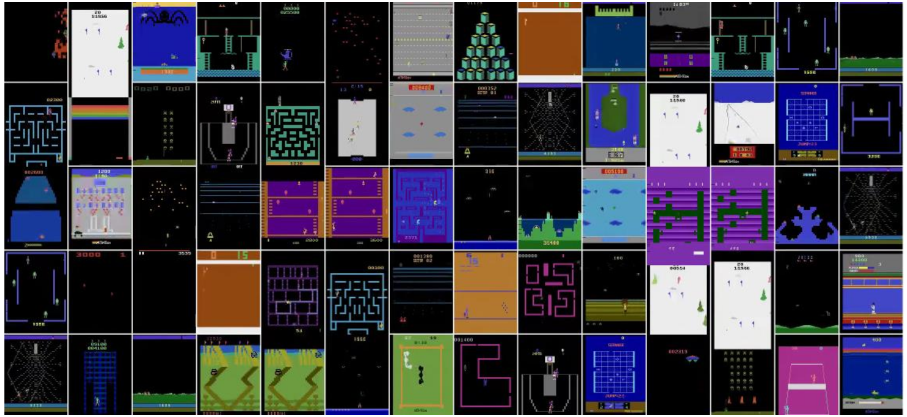
- Physics engine for RL, robotics, biomechanics, graphics research
- High performance with contacts
- **Free** now, purchased by DeepMind Oct 2021
- C/C++ with Python bindings

[https://www.roboti.us/index.html]

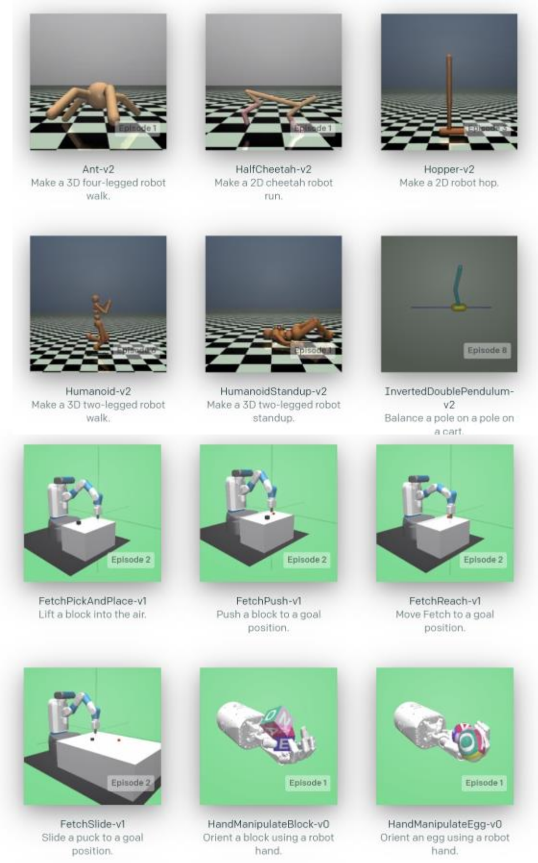
OpenAI Gym [2]

RL Research Platform

- Premade environments and interaction-API
- Classical control, Atari games, Robotics
- Standardization & performance baselines



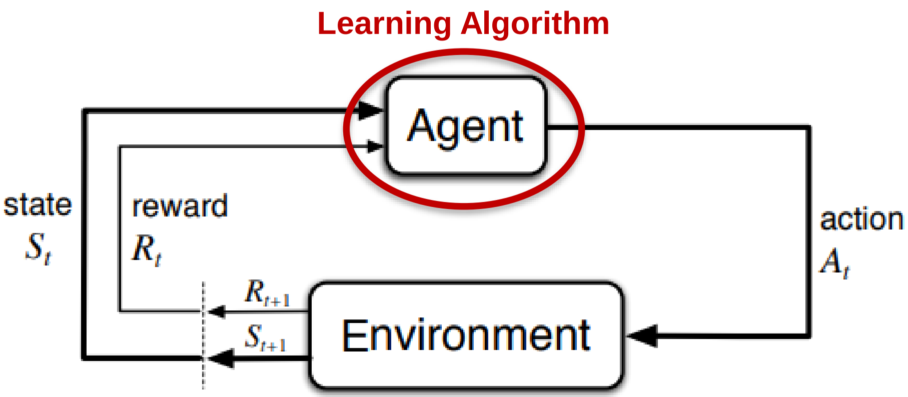
[https://gym.openai.com/envs]



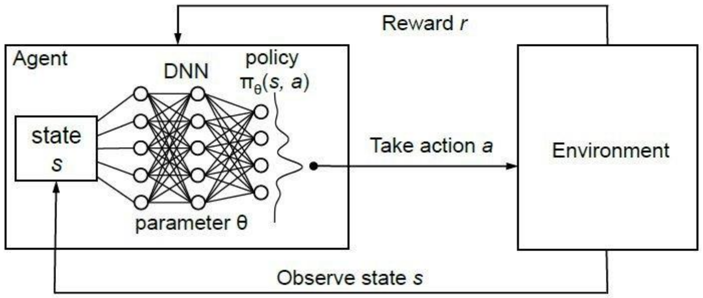
Reinforcement Learning

Overview

- Learning via interaction & feedback
- Model-free, data-driven
- Classical & Deep RL
- Value-based vs. policy-gradient based
- Discrete vs. Continuous action space
- On-policy vs. Off-policy training



[3, Sutton & Barto]



[4, Koo et al.]

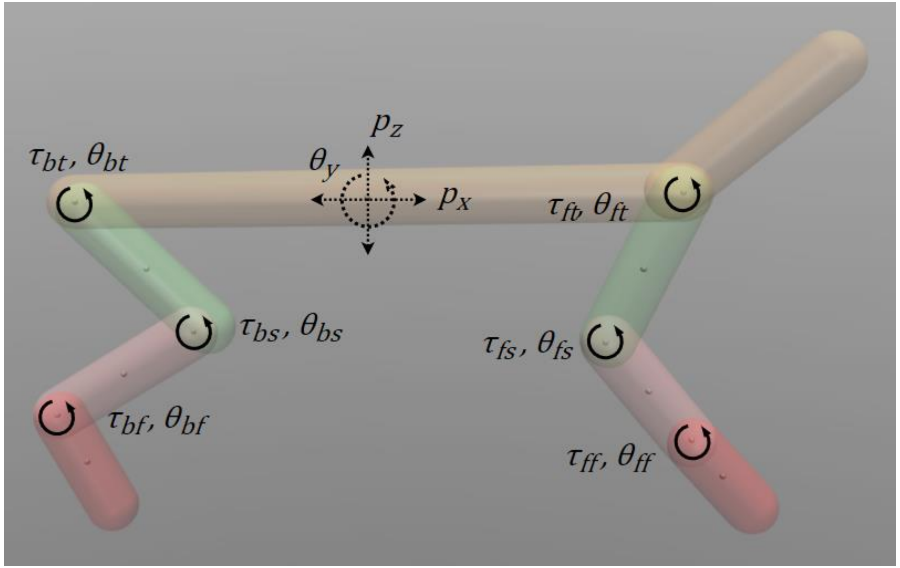
RL with HalfCheetah

- **State Space** (Observation)

$$[\theta_y, p_z, \theta_{bt}, \theta_{bs}, \theta_{bf}, \theta_{ft}, \theta_{fs}, \theta_{ff}, \dot{p}_x, \dot{\theta}_y, \dot{p}_z, \dot{\theta}_{bt}, \dot{\theta}_{bs}, \dot{\theta}_{bf}, \dot{\theta}_{ft}, \dot{\theta}_{fs}, \dot{\theta}_{ff}]$$

- **Action Space**

$$[\tau_{bt}, \tau_{bs}, \tau_{bf}, \tau_{ft}, \tau_{fs}, \tau_{ff}]$$



- **Reward Function (*V3*)**

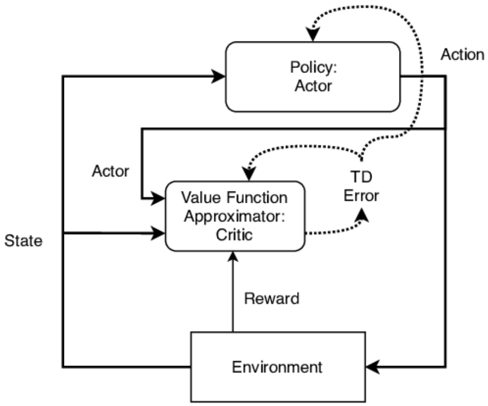
$$R(s,a) = \left[\omega_1 \dot{p}_x - \omega_2 \sum {\tau_{j,k}}^2\right]$$

TD3 [5]

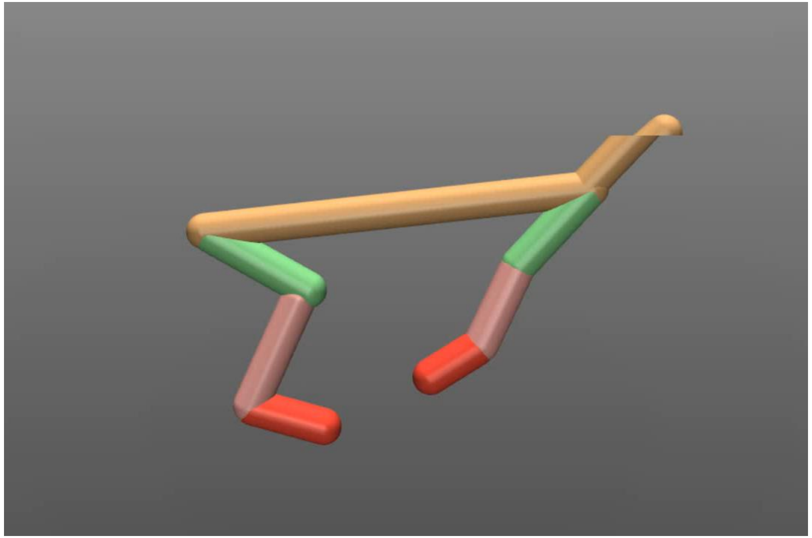
Twin Delayed **Deep Deterministic Policy Gradients [TD3]**

- Pretrained from RL Baselines3 Zoo & Stable Baselines3 [6,7]
- Actor-Critic algorithm
- Continuous control

Actor-Critic Architecture for Vanilla DDPG



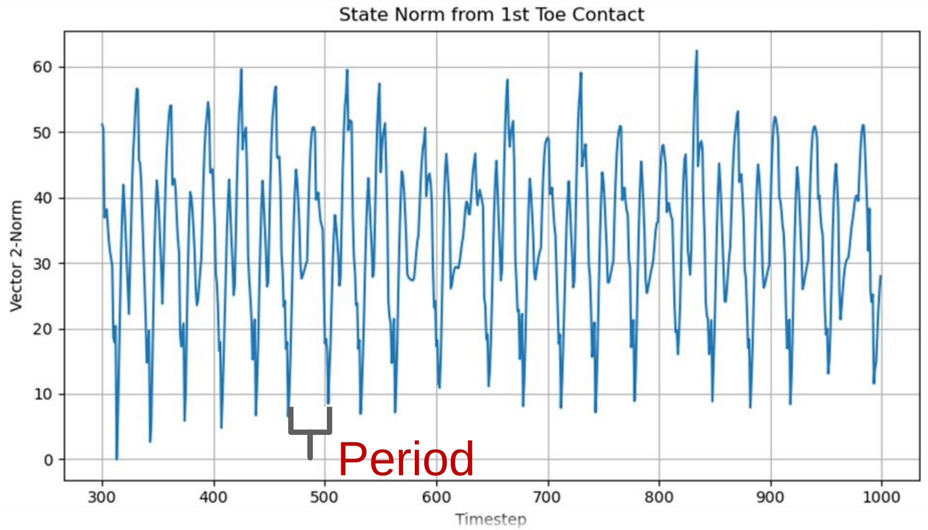
[8, Garrido et al.]

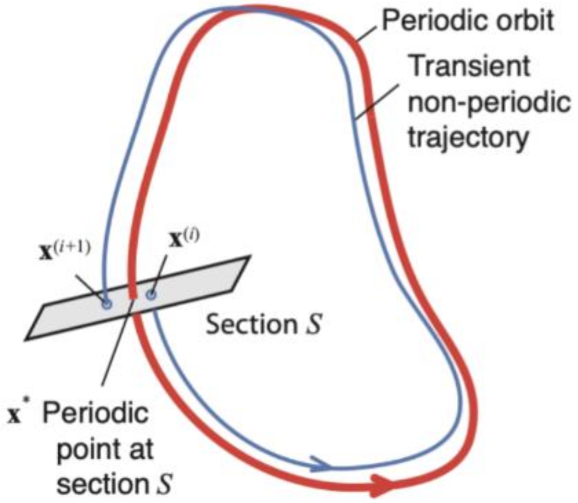


Periodic Stability Analysis

- Convert continuous dynamics into “discrete dynamics”
- Determine Poincare Map method
- Linearize Poincare Map → analyze eigenvalues

$$(x_{i+1} - x^*) = J\ (x_i - x^*)$$





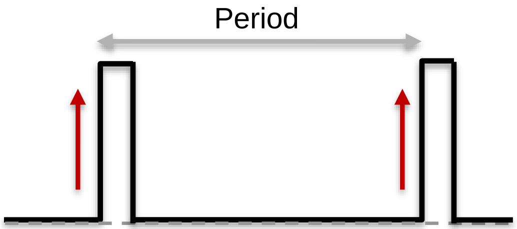
$$x_{i+1} = P(x_i)$$

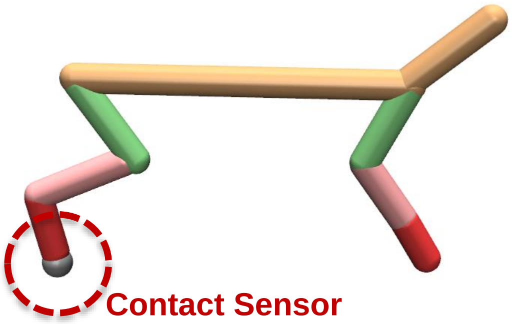
[9]

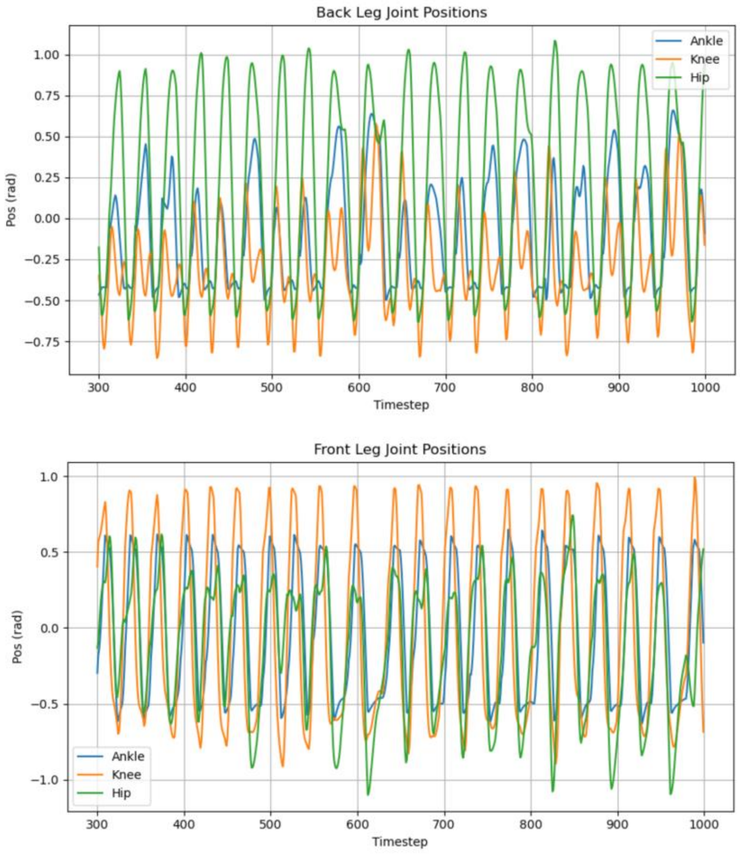
Motion Collection

Event-Based Poincare Map

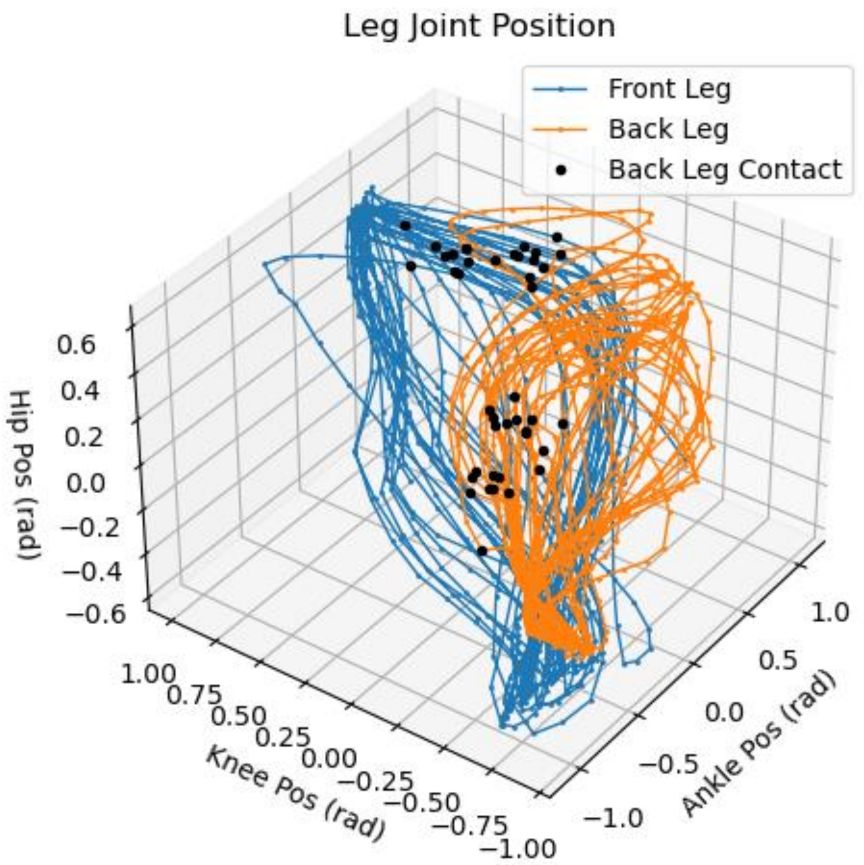
- Foot contact sensor, rising edge
- Show plots of raw data with foot contact

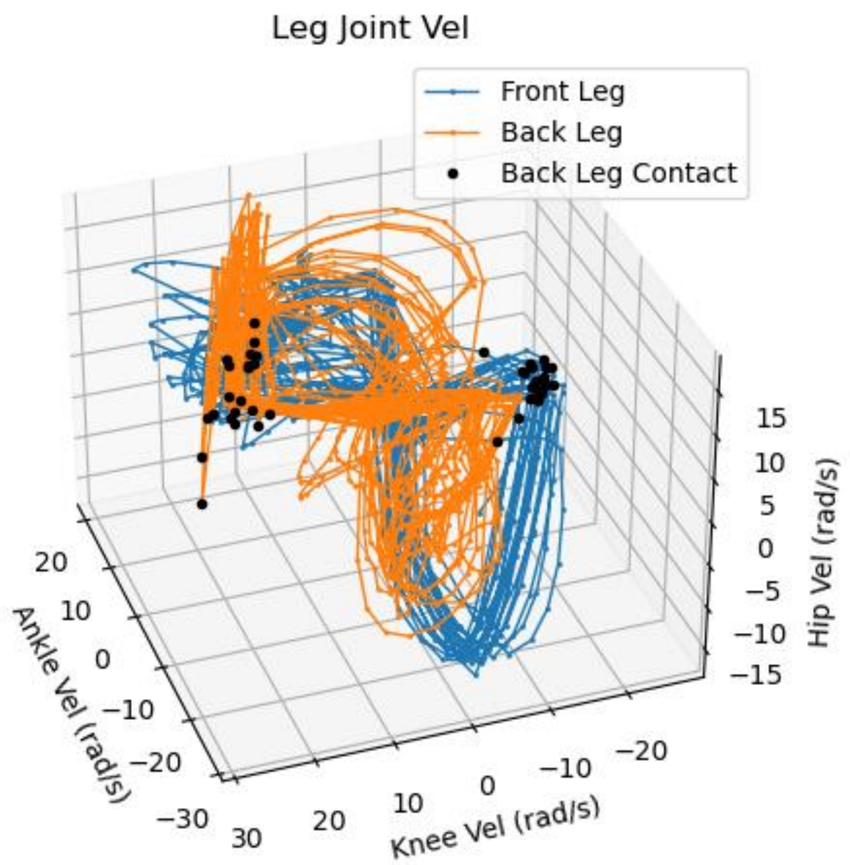






Motion Collection

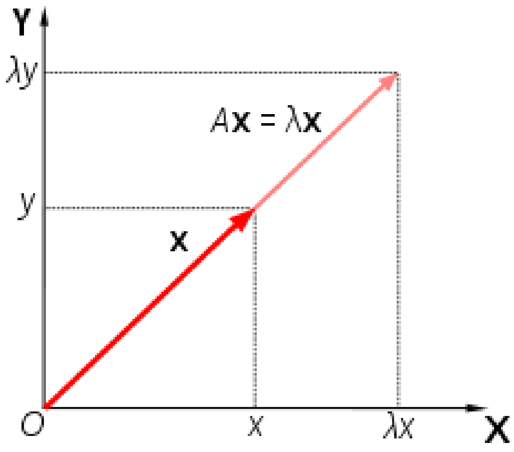




Stability Analysis

Data-Driven Poincare Linearization

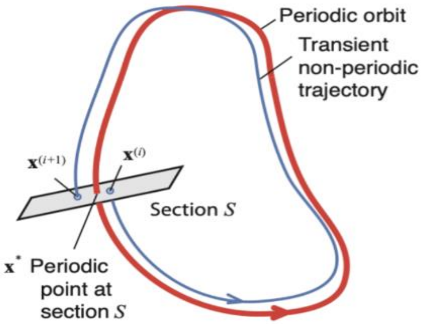
- Sampled from continuous simulation, no perturbations
- "Floquet Multipliers" of Jacobian, J
- Eigenvalue interpretation of stability



[https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors]

Linearize the function $P(x)$ around x^*

$$(x_{i+1} - x^*) = J\ (x_i - x^*)$$



$$x_{i+1} = P(x_i)$$

$$\left[\{s_1(T^{e1}), s_3(T^{e3}), s_5(T^{e5}), \dots\} - s_{avg}\right] = \boldsymbol{J}\left[\{s_1(0), s_3(0), s_5(0), \dots\} - s_{avg}\right]$$

Results

Eigenvalues of 17x17 Jacobian

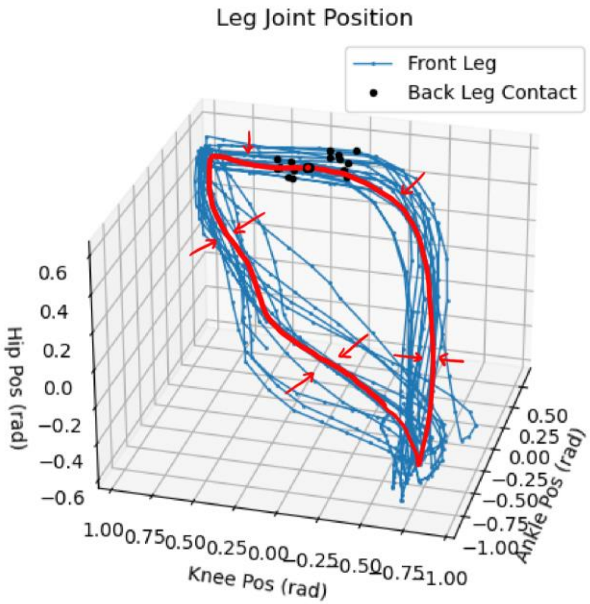
- Unstable
- Interpretation?

```
[-1.09715304+0.j          0.82956509+0.j          0.79674724+0.j
  0.10219659+0.695062j    0.10219659-0.695062j   -0.33976416+0.33917728j
 -0.33976416-0.33917728j  0.39890679+0.j         -0.35084479+0.16897418j
 -0.35084479-0.16897418j -0.37606791+0.j         -0.04917381+0.30565782j
 -0.04917381-0.30565782j  0.25404888+0.j          0.135096  +0.j
 -0.18004989+0.j         -0.09438665+0.j        ]
```

Future Work

Future Fun Research

- Motivation: Often agent learns “weird” policy that works well for objective, but how can we tune it to exhibit the subtle behaviors we want/expect

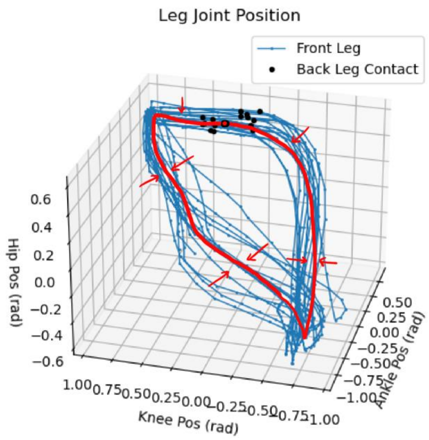


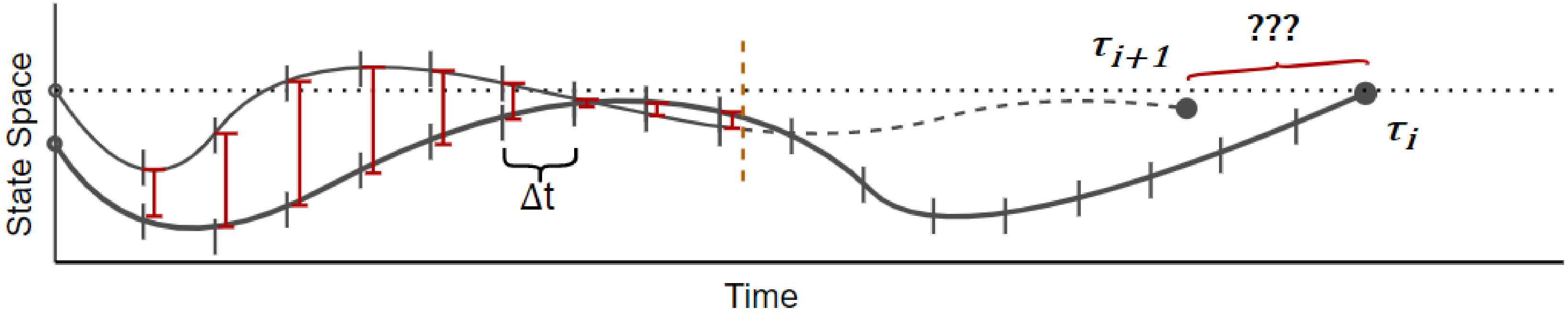
[https://www.businessinsider.com/sai]

Future Work

Current Idea – Reward Engineering

- Fine tuning of pretrained agent
- Real time or batch periodic analysis
- Dense vs. Sparse rewards

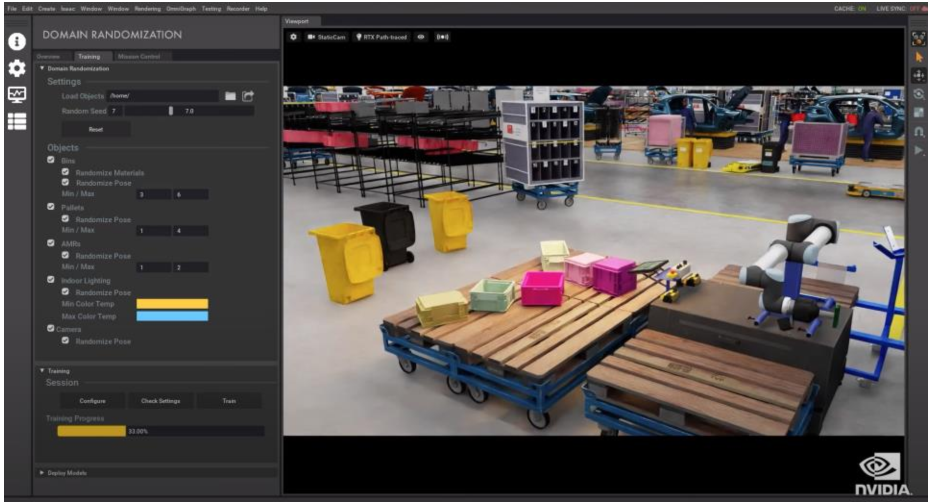




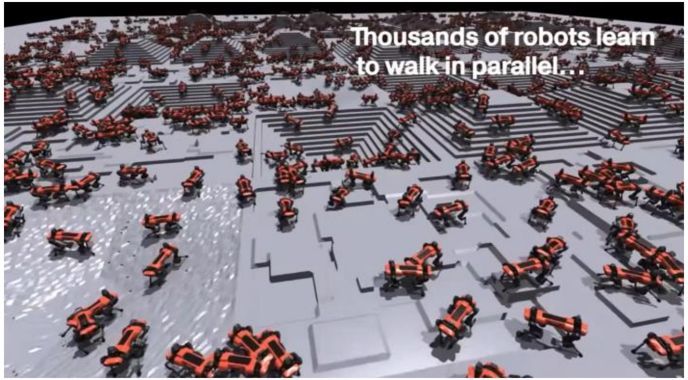
Sim2Real

Future Innovation

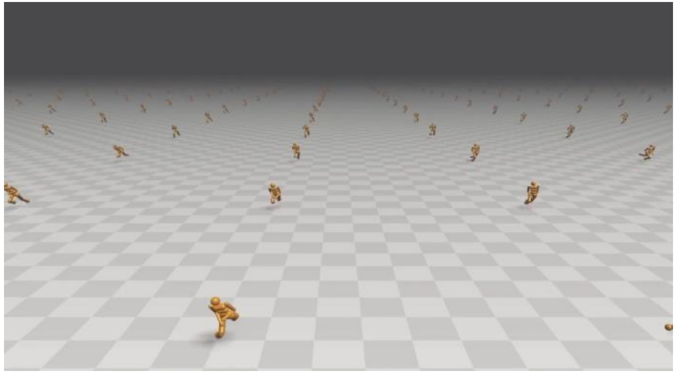
- Learning control, Sim2Real
- Transfer learning
- Issac Sim from NVIDIA, etc.



[https://developer.nvidia.com/isaac-sim]



[https://leggedrobotics.github.io/legged_gym/]



[10]

References:

[**My Project Repo**] https://github.com/mechyai/periodic_motion_rl
[**1**] MuJoCo https://mujoco.org/, https://github.com/deepmind/mujoco
[**2**] OpenAI Gym https://gym.openai.com/, https://github.com/openai/gym
[**3**] Sutton, Richard S, and Andrew G Barto. "Reinforcement Learning: An Introduction," n.d., 352.
[**4**] Koo, Jaehoon, Veena Mendiratta, Muntasir Raihan Rahman, and Anwar Walid. *Deep Reinforcement Learning for Network Slicing with Heterogeneous Resource Requirements and Time Varying Traffic Dynamics*, 2019.
[**5**] Fujimoto, Scott, Herke van Hoof, and David Meger. "Addressing Function Approximation Error in Actor-Critic Methods." *ArXiv:1802.09477 [Cs, Stat]*, October 22, 2018. http://arxiv.org/abs/1802.09477.
[**6**] RL Baselines3 Zoo https://github.com/DLR-RM/rl-baselines3-zoo
[**7**] Stable Baselines3 https://github.com/DLR-RM/stable-baselines3, https://stable-baselines3.readthedocs.io/en/master/, Raffin, Antonin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. "Stable-Baselines3: Reliable Reinforcement Learning Implementations," n.d., 8.
[**8**] Garrido, Luis, Rajiv Nishtala, and Paul Carpenter. "Continuous-Action Reinforcement Learning for Memory Allocation in Virtualized Servers," 2019.
[**9**] Dr. Manoj Srinivasan's ME7385 Periodic Stability Lecture Notes
[**10**] NVIDIA Developer Blog. "Reinforcement Learning Algorithm Helps Train Thousands of Robots Simultaneously," October 30, 2018. https://developer.nvidia.com/blog/nvidia-researchers-develop-reinforcement-learning-algorithm-to-train-thousands-of-robots-simultaneously/.

Thank you!

Questions?