Feedback Control For Cassie With Deep Reinforcement Learning

Zhaoming Xie[1] Glen Berseth[1] Patrick Clary[2] Jonathan Hurst[2] Michiel van de Panne[1]

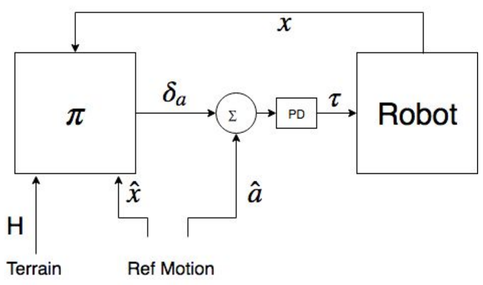
1.Department of Computer Science University of British Columbia
2.School of Mechanical, Industrial and Manufacturing Engineering Oregon State University

Goal and Questions

Design robust walking controllers for the bipedal robot Cassie using deep reinforcement learning.

- Can deep RL learn control policies for **realistic** underactuated bipedal robots?
- Can we improve learning efficiency with a reference motion? Prior work often generates unrealistic motions and requires a large number of training samples.
- How robust is the learned controller?

System

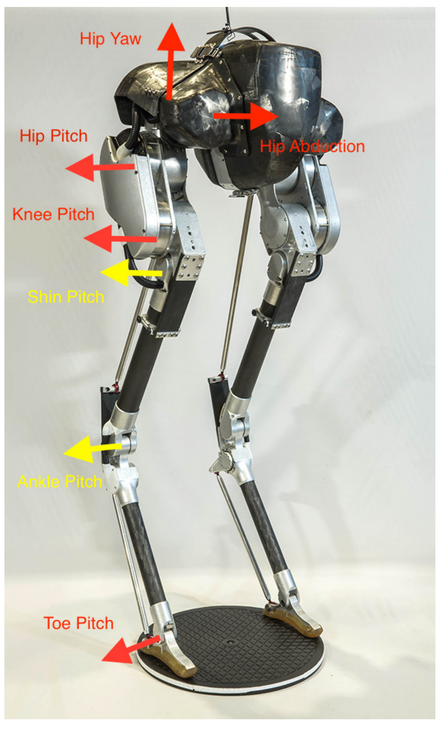


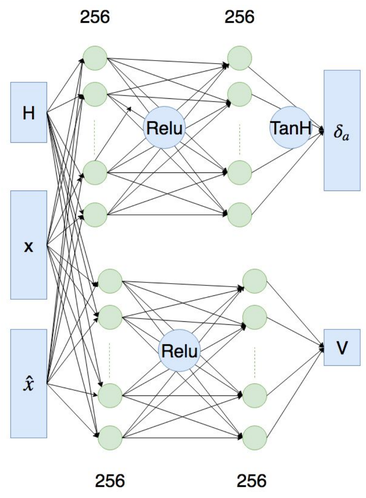
- $x = [q, \dot{q}],\ a = q_d,\ s = [H, x, \hat{x}]$.
- States: height map of upcoming terrains, robot pose and reference pose.
- Actions: PD-control targets.
- Simulation: Mujoco with realistic model of Cassie.
- Learning Algorithm: actor-critic reinforcement learning with Proximal Policy Optimization.

Method

- Objective: tracking a reference motion.
- Input choices:
  (a) full state estimate
  (b) full state estimate + terrains
  (c) sensory input like readings from IMU and encoders for end-to-end learning.
- Curriculum Learning: train on easy terrains before moving to harder ones.
- Interpolation: transition between different policies.

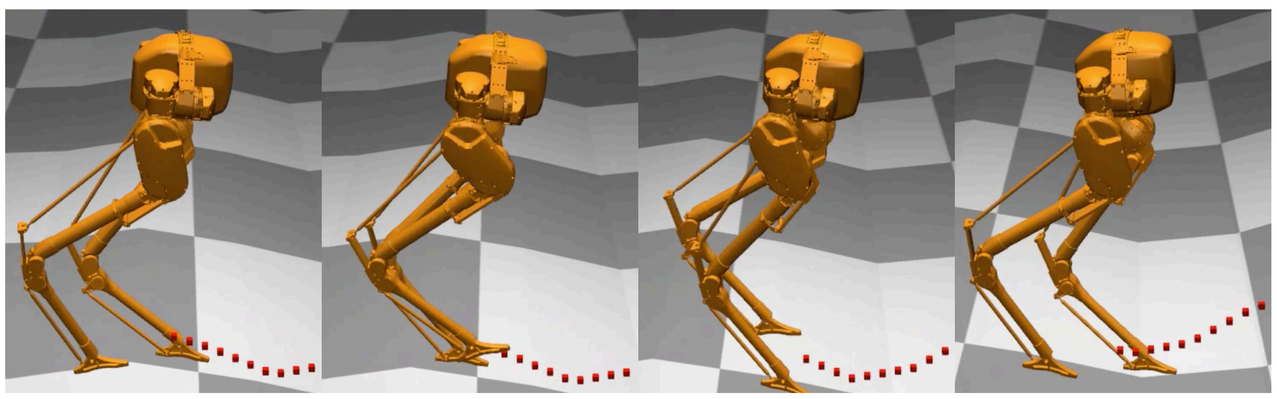
Training Details





Network structure

Example terrain used for training



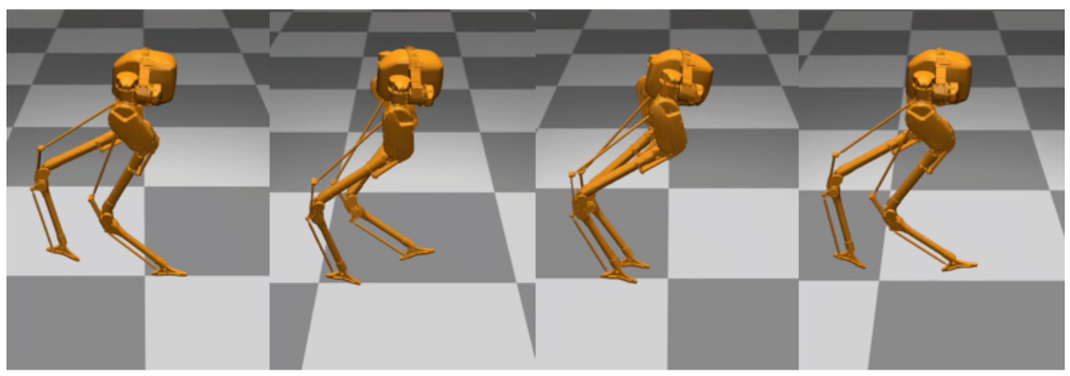
- Random Terrains: during training, slopes are sampled from [-20, 20] degrees, and changed every 40cm.
- Terrain Inputs: height of the upcoming 50cm terrains are sampled every 5cm and fed into the policy.

Efficiency

- OpenAI gym Humanoid: millions of samples + days are needed.
- With reference motions, only 0.9 million samples and 1-2 hours are needed for blind walking.

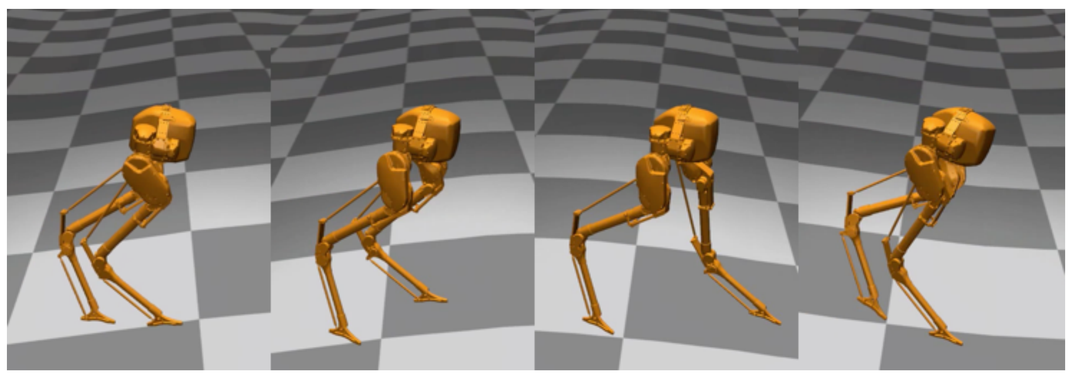
Results

- Sensory Delay: The controllers perform well with a 5ms delay, but fails at 10ms delay (PD controller receive delay as well).
- Force Perturbation: Controller can recover from pushes that lasts for 0.2s with a magnitude of 140N in the forward direction, 90N in the backward direction and 50N from either left or right side.
- Faster walking gait can be created by scaling the translation of the pelvis and retraining. Transition between different speeds can be achieved via interpolation between policies.



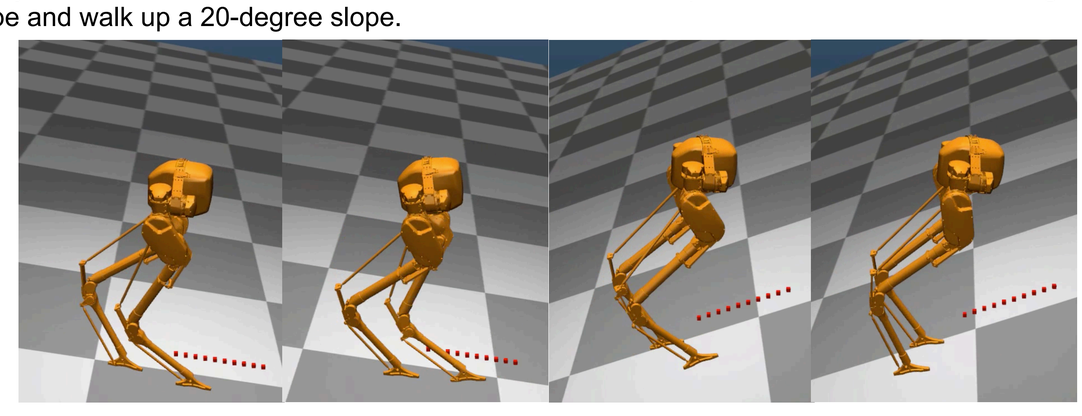
Variable speeds using interpolated policy

- Blind Walking: Controller trained blindly on flat terrain can walk across sinusoidal terrain with maximum slope of 8 degrees.
- Interpolation: Interpolated policies based on the current speed of the robot can walk blindly across sinusoidal terrain with maximum slope of 12 degrees.



Blind walking on sinusoidal terrain

- Policy trained on random terrains is evaluated on constant slopes. It can walk down a 10-degree slope and walk up a 20-degree slope.



Terrain aware walking

Ongoing Work

- More terrain types: stairs, gaps, etc.
- More Skills: unified policy that can track various motions.
- Sim-To-Real: Can the policies be reliably transferred to the real robot?

Related Work

- Peng, XB., Abbeel, P., Levine, S., and van de Panne, M. (2018). DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills.
- Schulman, J., Wolski F., Dhariwal P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms.
- Da, X., Harib, O., Hartley, R., Griffin, B., and Grizzle, J. (2016). From 2D Design of Underactuated Bipedal Gaits to 3D Implementation: Walking With Speed Tracking
- Berseth, G., Xie, C., Cernek, P., and van de Panne, M. (2018). Progressive Reinforcement Learning With Distillation for Multi-Skilled Motion Control.