Learn to Move with Deep Reinforcement Learning

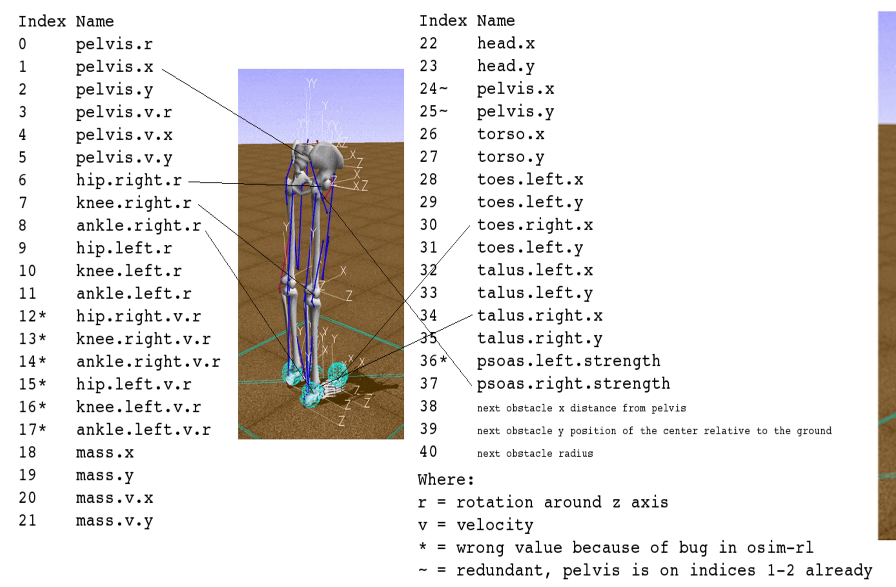
Samet Demir Kemal Bektaş
Computer Eng. Mechanical Eng.
Boğaziçi University

Introduction

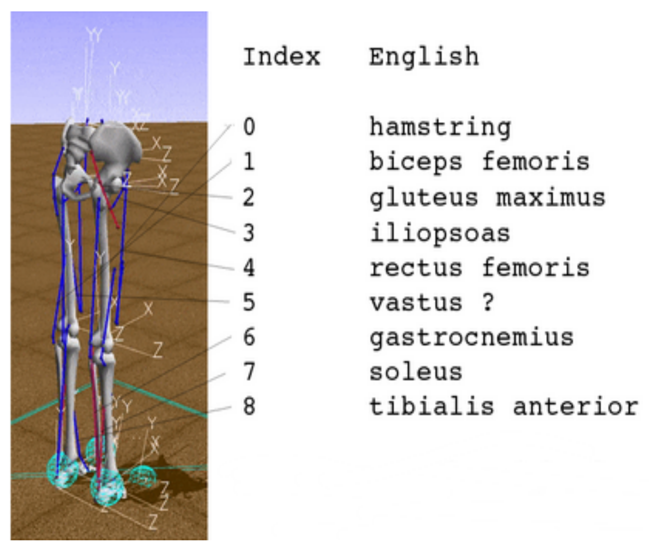
This project is based on NIPS (Neural Information Processing Systems) 2017 challenge named Learning to Run by Stanford Neuromuscular Biomechanics Laboratory. The task is developing a controller to move a physiologically-based human model. We decided to use deep reinforcement learning method because the task seemed suitable for it.

- State: R^{41} vector with coordinates and velocities of various body parts and obstacle locations.
- Action: R^{18} vector with muscles activations, 9 per leg, each in [0, 1] range.
- Reward: Change in x location of Pelvis minus small penalty for using ligament forces

Observation Parameters[5]



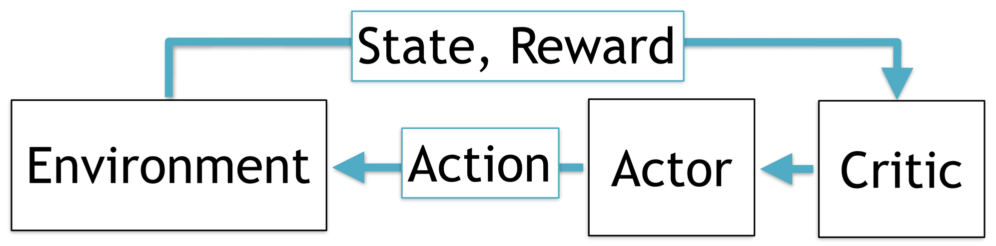
Action Parameters[5]



Methods

Deep Reinforcement Learning

Actor Critic Methods



1) Deep Deterministic Policy Gradient (DDPG)

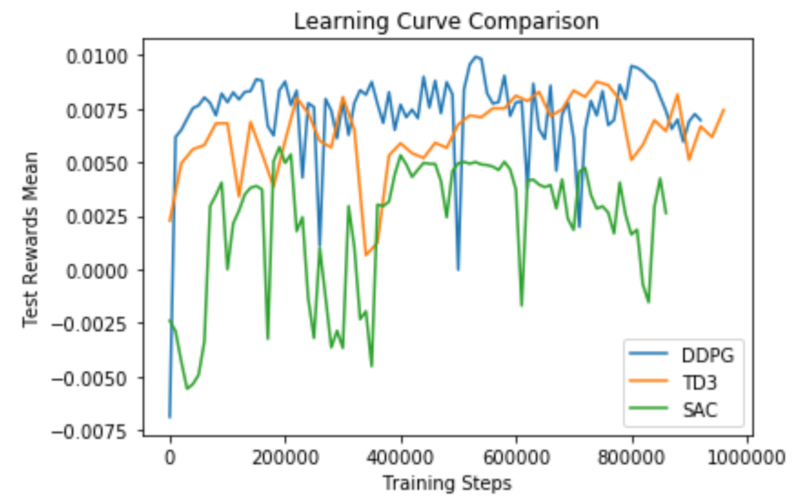
2) Soft Actor Critic (SAC)

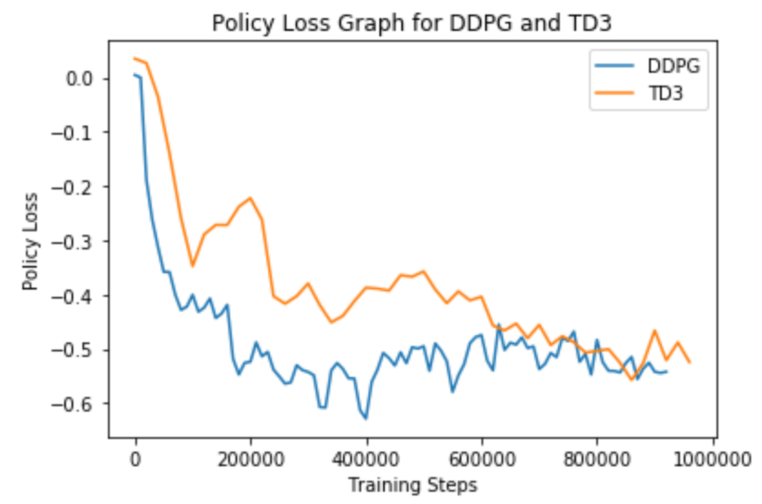
3)Twin Dueling Deep Deterministic Policy Gradient (TD3)

Results

We ran all models for ~1.000.000 steps and DDPG agent produced the most promising results by learning to take a step.

Max Rewards	
DDPG	0.020
TD3	0.025
SAC	0.021

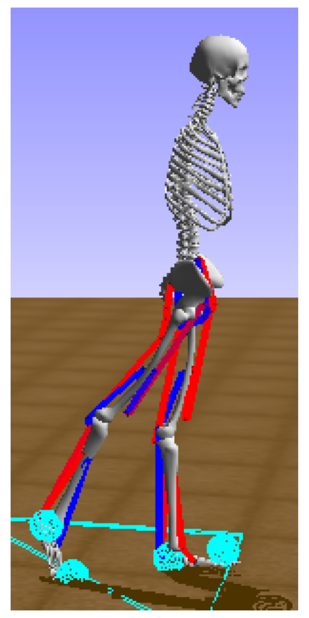




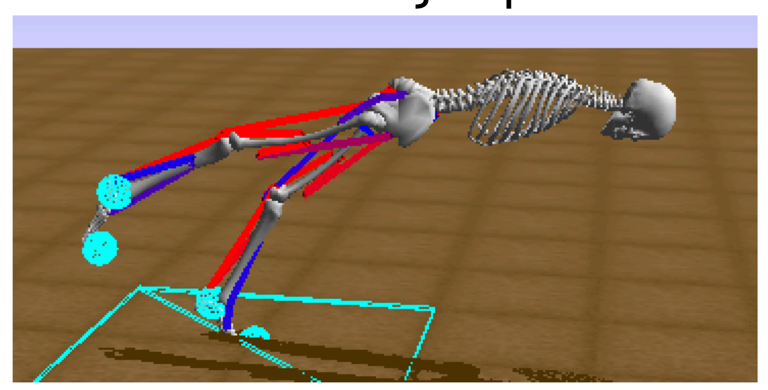
Experiments

(Pictures from the Simulation)

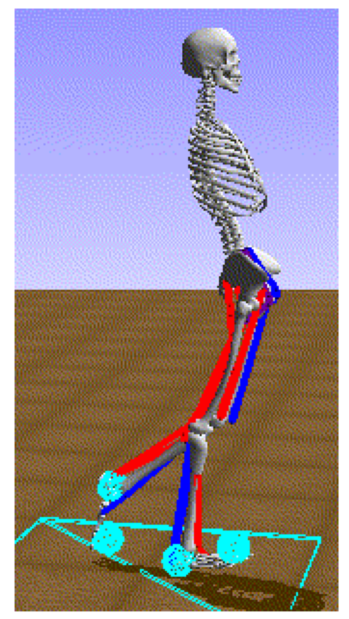
a) learned left step



c) learned forward jump



b) learned right step

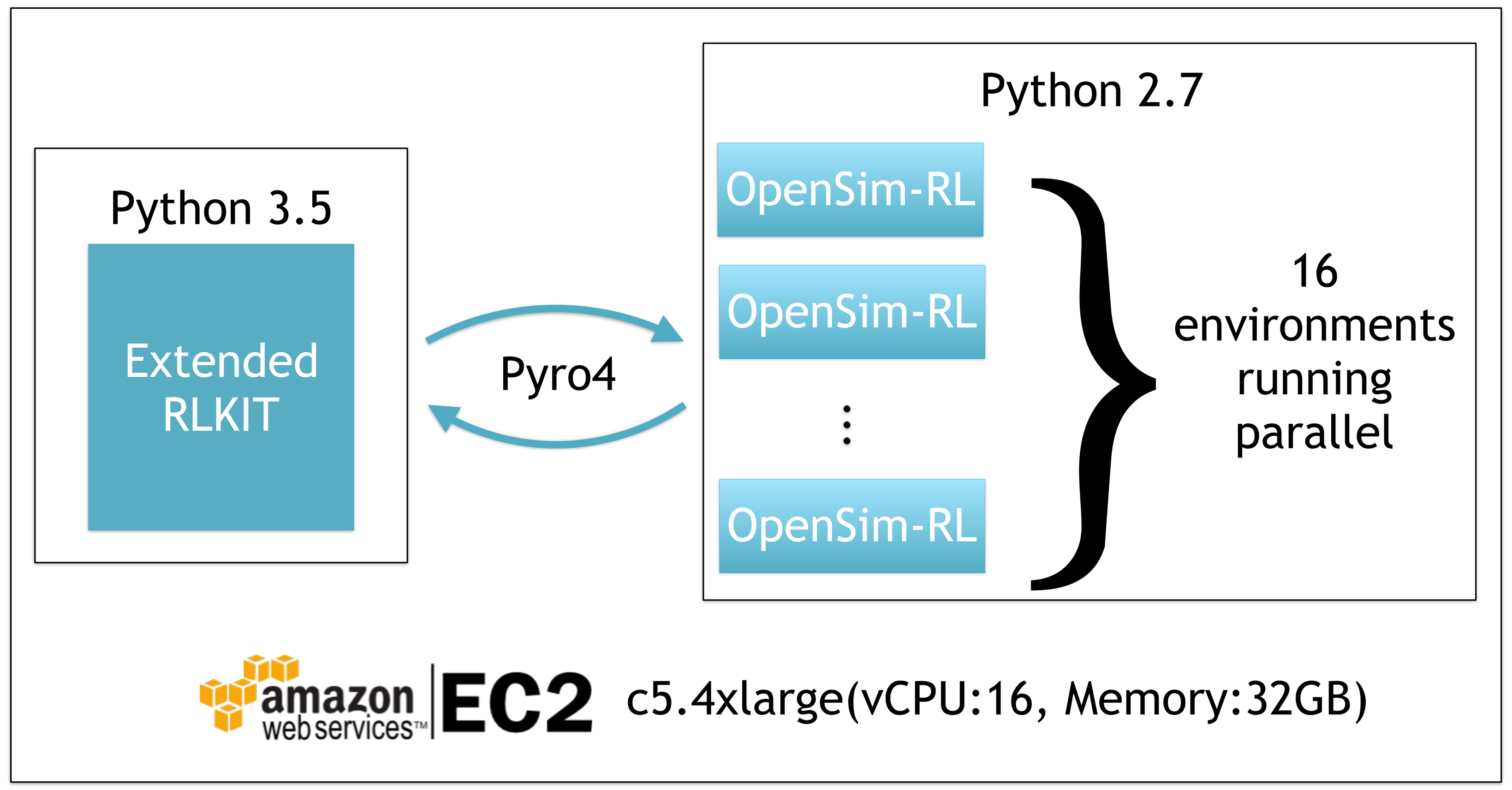


Hyperparameters

	DDPG	SAC	TD3
reward discount	0.99-0.995	0.99	0.99-0.995
qf learning rate	1E-03	3E-03	1E-03
pol. learning rate	1E-04	3E-03	1E-03
batch size	128	128	128
size of hidden layers	64-256	64	256
number of steps	~1M	~1M	~1M

Training System

The environment is computationally expensive. Therefore, a system with multiple environment running in parallel is necessary.



RLKIT : Reinforcement learning framework
OpenSim-RL : Simulation environment, extension of OpenSim to RL
Pyro4 : Communication library

Feature Engineering

- Velocities of body parts are added to observations
- Noise is added to actions

Discussion & Future Work

There is a big space to improve our results since the parallel data collecting system is still slow and we couldn't try long training.

References

1. Kidzinski, Lukasz and Mohanty, Sharada P and Ong, Carmichael and Hicks, Jennifer and Francis, Sean and Levine, Sergey and Salath'e, Marcel and Delp, Scott. (2018) "Learning to Run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning". NIPS 2017 Competition Book. Springer.
2. Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. (2015). Continuous control with deep reinforcement learning. arXiv:1509.02971
3. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv:1801.01290
4. Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. arXiv:1802.09477.
5. Stelmaszczyk, Adam. "Our NIPS 2017: Learning to Run Approach". Medium-MLReview. 19 Nov. 2017. https://medium.com/mlreview/our-nips-2017-learning-to-run-approach-b80a295d3bb5