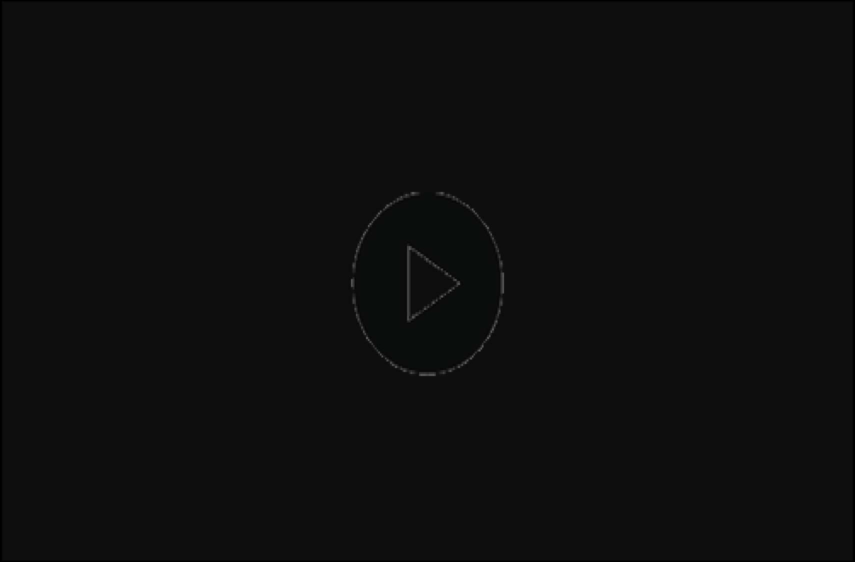
Robust control strategies for musculoskeletal models using deep reinforcement learning

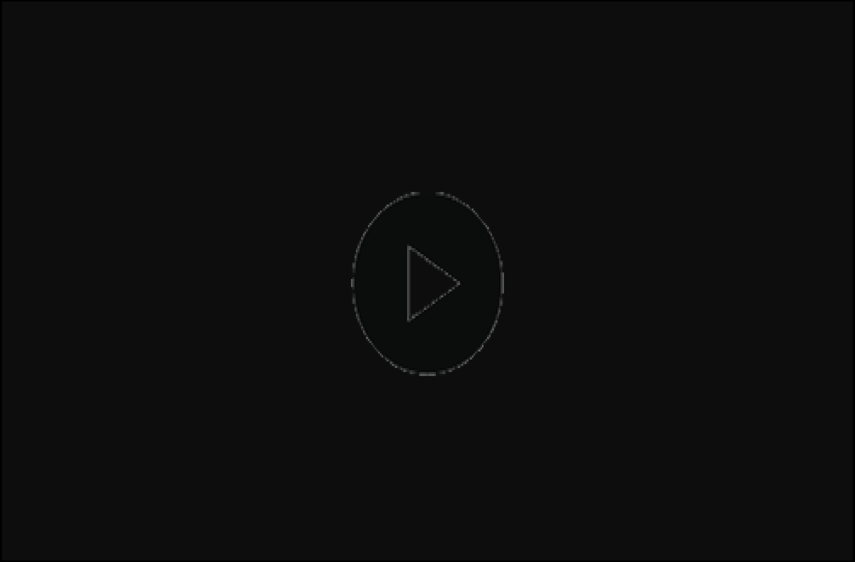
Łukasz Kidziński & Carmichael Ong

Stanford University

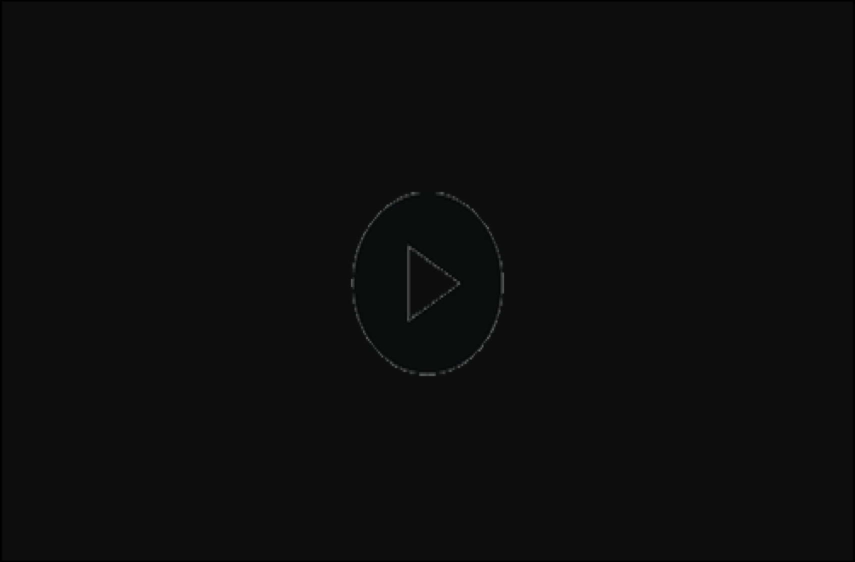
August 7, 2018



Ref: Gillette Children's Specialty Healthcare



Ref: OpenSim (https://www.youtube.com/watch?v=HLFQM1e-vJw), Kat Steele



Ref: Emergence of Locomotion Behaviours in Rich Environments, Google DeepMind

How to combine modern control frameworks with knowledge and expertise embeded in musculoskeletal models?

Contents

What can we do with Reinforcement Learning?

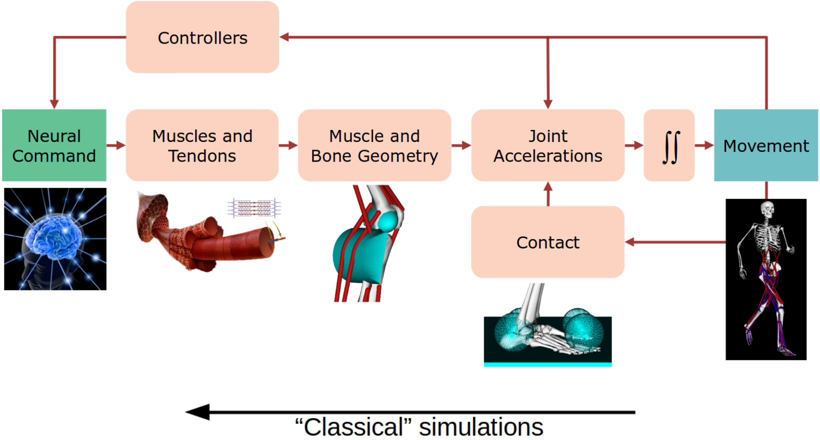
Stanford University

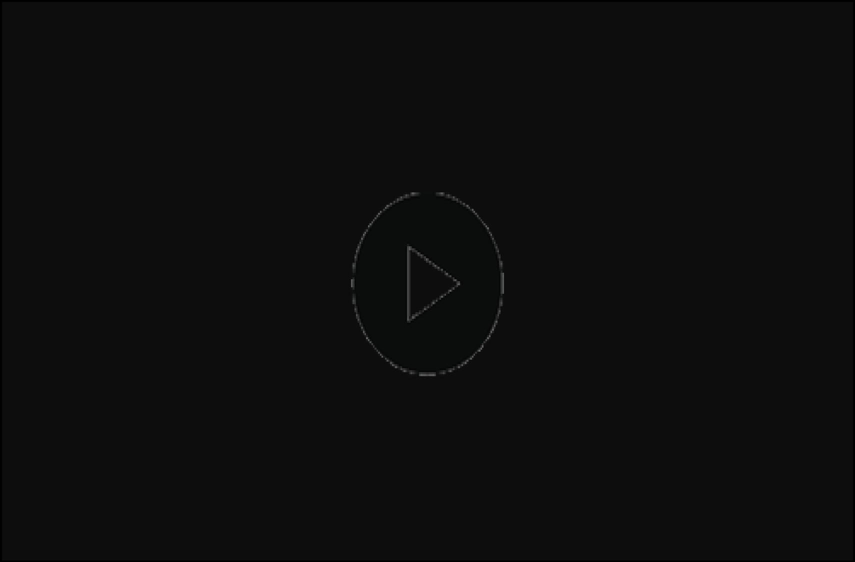
1. Model adaptation of kinematics
2. Generate data for statistical models
3. Study:
   - motor control
   - muscle synergies
4. Synthesize physiologically accurate motion

OpenSim

Stanford University

OpenSim is a freely available software that allows you to build, exchange, and analyze musuloskeletal models and dynamic simulations of movement. (http://opensim.stanford.edu/)



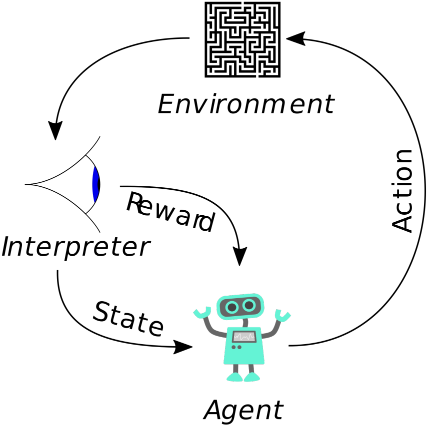


Ref: NNAISENSE

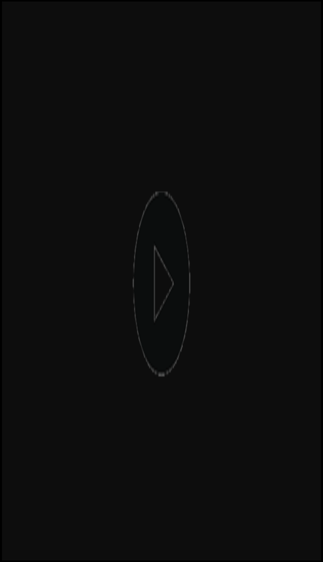
Reinforcement learning framework

Stanford University

Reinforcement learning models **a policy** (a decision process) of **an agent** interacting with **an environment** by taking **actions** optimizing **rewards**.



Ref: Wikipedia



Ref: Very good friends and their child

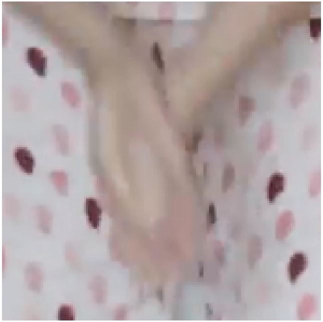
Agent

Stanford University

- Tries to achieve the *objective* (long-term)
- Tries to maximize the *reward* (short-term)
- Interacts with the environment

Reward / Penalty

Stanford University



Clapping

Falling

Observation and action

Stanford University

Observation:

- State of the world
- Sensory input
- State of muscles, joints and bones
- etc.

Action:

- Muscle excitations

Policy:

- Links observations with actions

Mathematical definition of a policy

Stanford University

Policy maps states to actions

Let S be a space of states, A be a space of actions.
Policy is a function $P_\theta : S \to A$.

Any statistical model can be a policy

Normally, a policy is parametrized with some parameters $\theta \in \mathbb{R}^p$.

We want to find the policy maximizing rewards

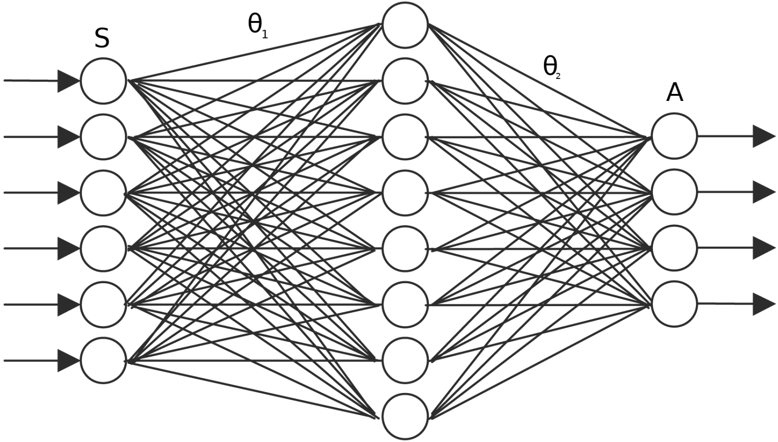
Let $R : S \to \mathbb{R}$ be the reward function. We optimize

$$\arg\max_\theta \sum_{t=1}^{T} R(s_{t,\theta}),$$

where $\{s_{t,\theta}\}_{1 \leq t \leq T}$ is a trajectory dependent on the policy.

The most popular model is a neural network

Stanford University



Stacked linear regressions with nonlinear transformations in outputs.

Training a policy

Stanford University

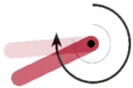
We want to find the policy maximizng rewards.

Generic optimization approach:

- Start with an empty (random) policy
- Repeat until convergence
    - Use the policy with noise to explore environment
    - Improve the policy given the experience

Toy problem

Stanford University

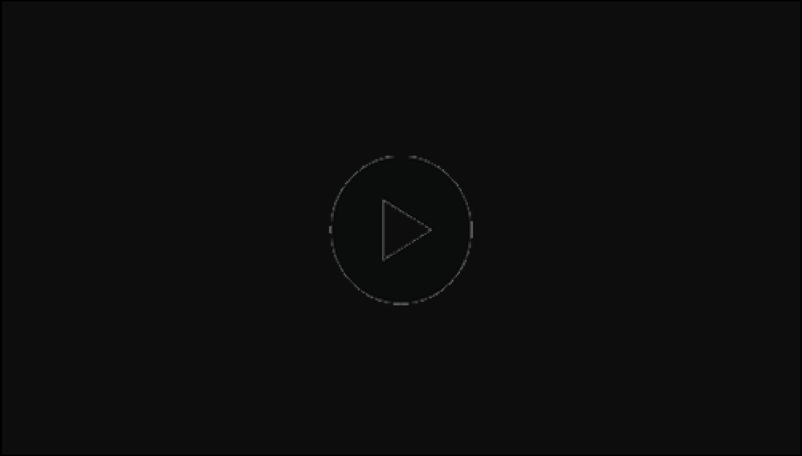


Pendulum Swing-up

- **Environment:** Pendulum (one joint, one degree of freedom)
- **Obejective:** Swing it up
- **Observation:** Angular position & velocity (a vector in $\mathbb{R}^2$)
- **Action:** Apply torque
- **Reward:** Negative distance from 0 velocity, 0 angle

Exploration of the environment

Stanford University

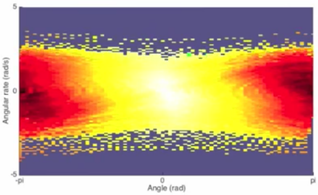


Ref: Matthew Sheen, Reinforcement Learning - Pendulum Swing-up

https://www.youtube.com/watch?v=YLAWnYAsai8

Solve with policy gradient (DDPG)

Stanford University



Let $Q : S \times A \to \mathbb{R}$ be an approximation of a state-action value.
Let $P_\theta : S \to A$ be the best policy so far.

Now iteratively:

- Update approximation of Q from history of trials
- Optimize $J(\theta) = Q(s, P_\theta(s))$
- Compute the gradient of J
- Update θ

Contents

Stanford University

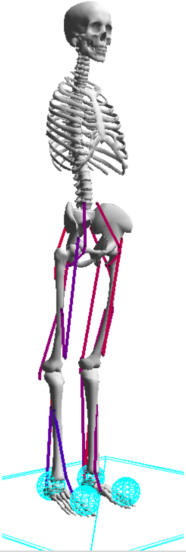
NIPS 2017: Learning to Run challenge

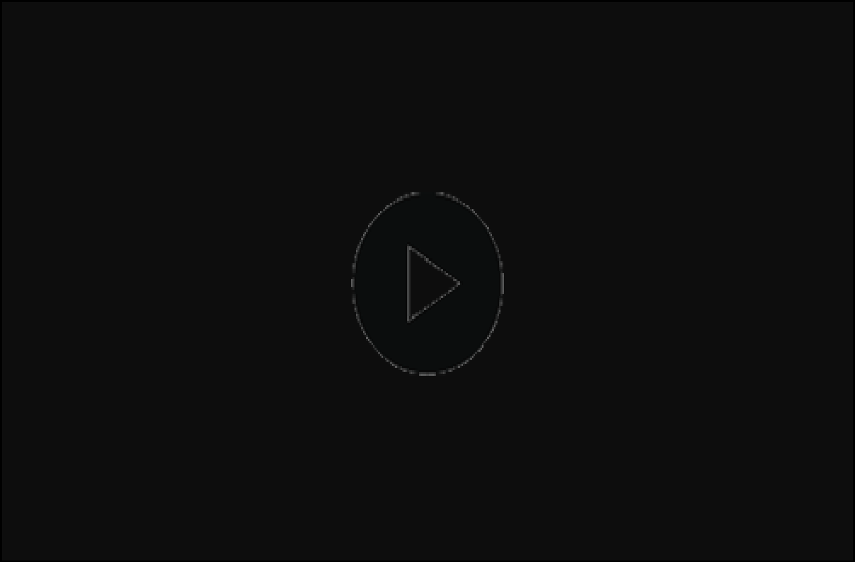
Task:

- **Environment:** A model with 18 muscles and 9 DOF
- **Objective:** Go as far as possible in 10 seconds
- **Observation:** State of the model
- **Action:** Muscle excitations
- **Reward:** Speed

Some details:

- Obstacles in the ground
- 2D model (the model doesn't fall sideways)
- Don't use experimental data

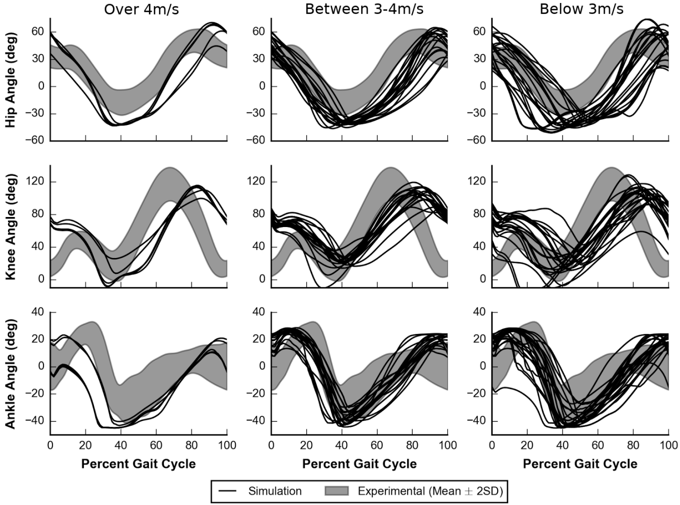




Ref: NNAISENSE

Experimental vs synthesized data

Stanford University

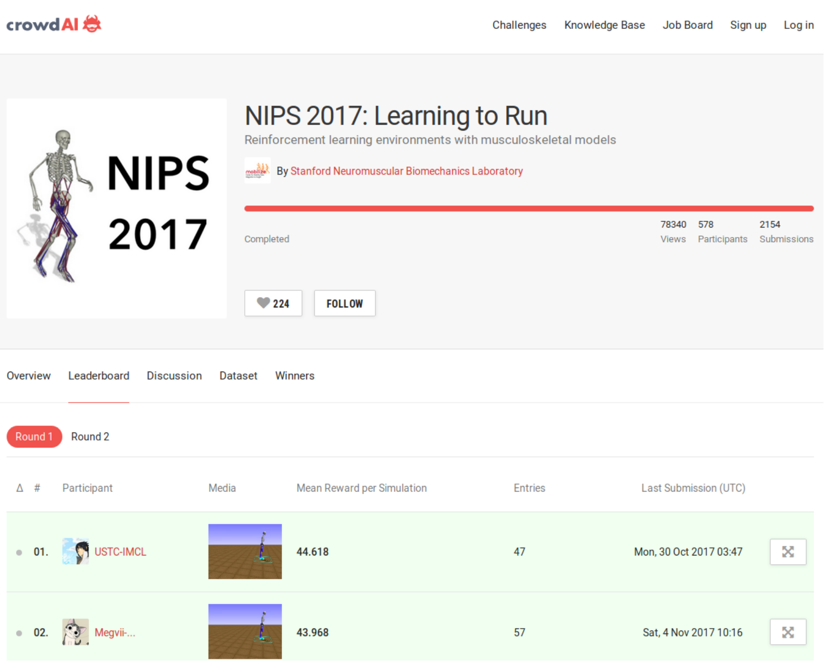


Speeding up exploration

Stanford University

- Frameskip: instead of sending signals every 1/100 of a second, keep the same control for, for example, 5 frames.
- Binary actions: excitations 0 or 1 instead of values in the interval [0,1].
- Reward shaping: modifying the reward for training in such a way that it still makes the model train faster for the actual initial reward.

Ref: Kidziński, Łukasz, et al. "Learning to Run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments." arXiv preprint arXiv:1804.00361 (2018).


crowdAI

Challenges Knowledge Base Job Board Sign up Log in

NIPS 2017

NIPS 2017: Learning to Run

Reinforcement learning environments with musculoskeletal models

By Stanford Neuromuscular Biomechanics Laboratory

Completed

78340 Views 578 Participants 2154 Submissions

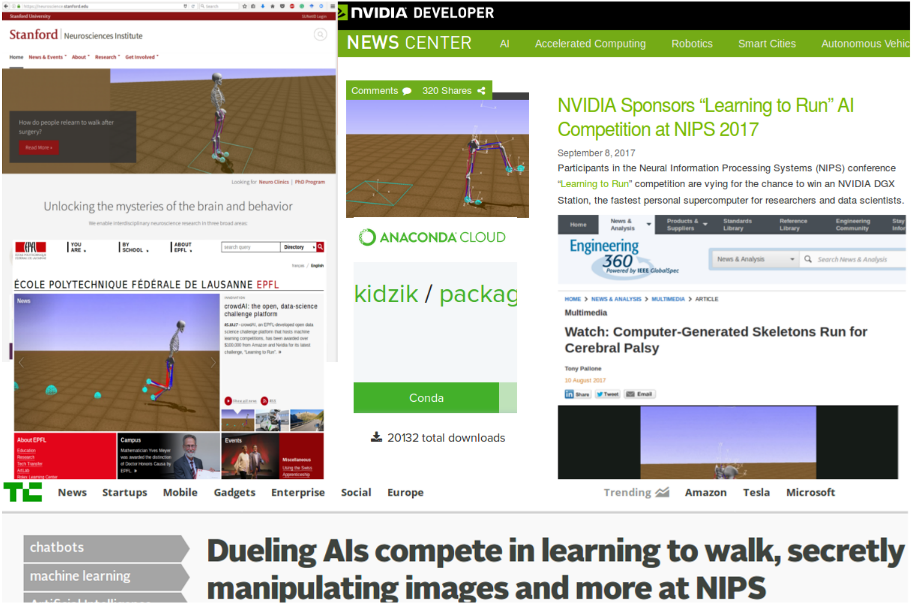
224 FOLLOW

Overview Leaderboard Discussion Dataset Winners

Round 1 Round 2

Δ	#	Participant	Media	Mean Reward per Simulation	Entries	Last Submission (UTC)
•	01.	USTC-IMCL		44.618	47	Mon, 30 Oct 2017 03:47
•	02.	Megvii-...		43.968	57	Sat, 4 Nov 2017 10:16

https://www.crowdai.org/challenges/nips-2017-learning-to-run


Stanford Neurosciences Institute
How do people relearn to walk after surgery?
Unlocking the mysteries of the brain and behavior
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE EPFL
crowdAI: the open, data-science challenge platform
NVIDIA DEVELOPER
NEWS CENTER
AI
Accelerated Computing
Robotics
Smart Cities
Comments
320 Shares
NVIDIA Sponsors "Learning to Run" AI Competition at NIPS 2017
September 8, 2017
Participants in the Neural Information Processing Systems (NIPS) conference "Learning to Run" competition are vying for the chance to win an NVIDIA DGX Station, the fastest personal supercomputer for researchers and data scientists.
ANACONDA CLOUD
kidzik / packag
Conda
20132 total downloads
Engineering 360
Multimedia
Watch: Computer-Generated Skeletons Run for Cerebral Palsy
Tony Pallone
10 August 2017
News
Startups
Mobile
Gadgets
Enterprise
Social
Europe
Trending
Amazon
Tesla
Microsoft
chatbots
machine learning
Dueling AIs compete in learning to walk, secretly manipulating images and more at NIPS

NIPS 2018: AI for prosthetics

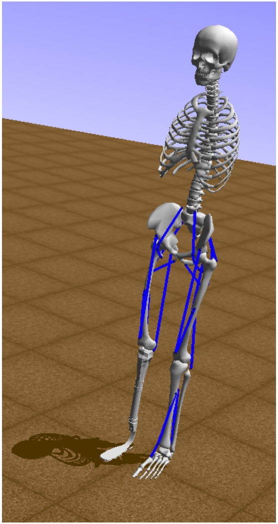
Stanford University

Rules:

- 19 muscles and 14 DOF
- Match requested velocity vector
- 3D model
- Experimental data is allowed

Rewards:

- Solving a real medical problem!
- Travel grants to Stanford & EPFL
- $250 Google Cloud Credits for start
- 4 NVIDIA GPUS to win (∼$3,000 each)

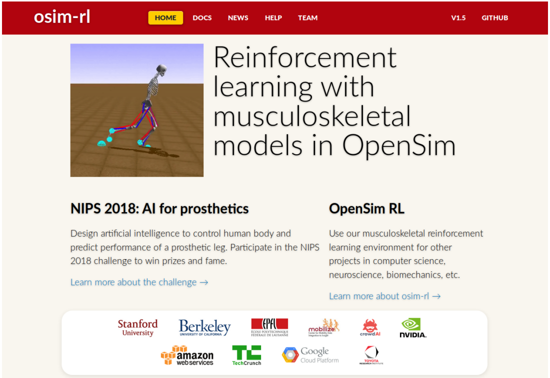


Contents

Stanford University

Linking OpenSim with RL (osim-rl)

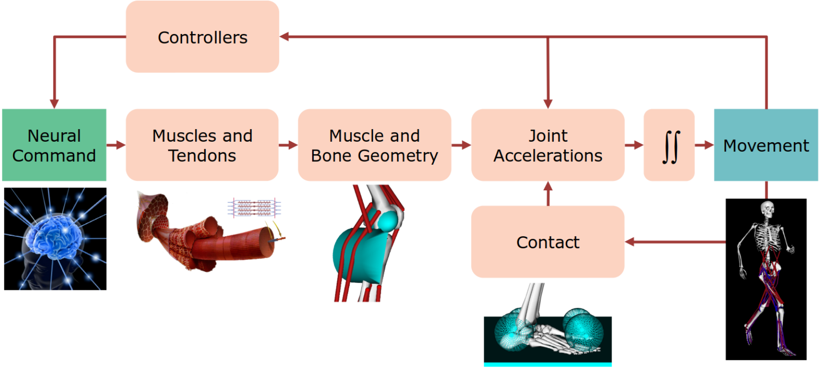
Stanford University



`osim-rl` is an **interface** for building **robust controllers** for your **OpenSim models** using open **deep reinforcement learning algorithms**.

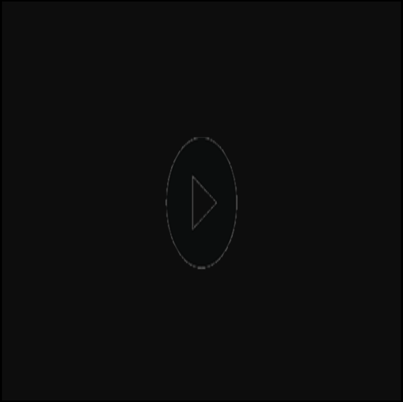
http://osim-rl.stanford.edu/

```
> # Assume that the brain variable is a policy function
> env = ProstheticsEnv(visualize=True)
> observation = env.reset()
> for i in range(200):
>     observation = env.step(brain.action(observation))
```



Arm model

Stanford University



https://github.com/stanfordnmbl/osim-rl/blob/master/examples/legacy/train.arm.ipynb

Summary

Stanford University

- Reinforcement learning (RL) allows you to build a robust controller by specifying a high-level objective
- `osim-rl` package allows you to use RL frameworks to control your OpenSim models