[10.1.2021]

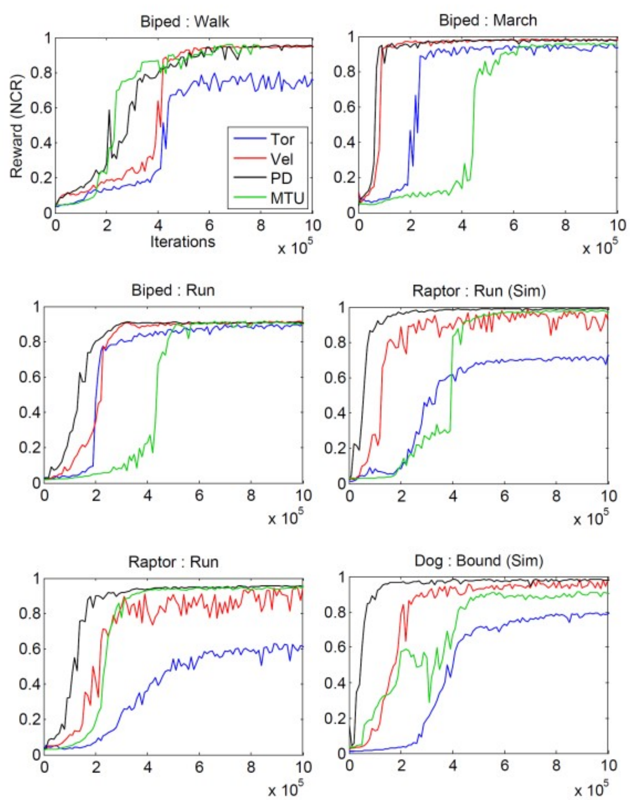
[Learning Locomotion Skills Using DeepRL: Does the Choice of Action Space Matter?]

Summary

This paper researches about the effect of using different action space in reinforcement learning.

Some key points of this paper:

➢ It is normally nice to keep some simple local feedback structure → maybe neural network is unable/inefficient to learn that feedback?



➢ <u>Prefer to choose joint target position or velocity target instead of using torques directly</u>. → probably that is also why ANYmal learning motor papers introduces a actuator network that maps joint position to torque?→ In some sense, it is simply learning the kd and kp parameters...

Major Analysis and Comparison

1) Elements of RL:
    - Four kinds of action space:
        - target joint angles, and the torque can be computed: \tau = kp(q*-q)+kd(\dot q*-\dot q) → we still have some parameters to tune, and we set the target joint velocity to zero;
        - target joint velocity, torque: \tau = kd(\dot q*-\dot q)
        - use torque directly as action space, no extra parameter to tune
        - Muscle activation → I think it is not relevant to robotics yet.
    - State space: (height of body, position of foots, center of mass velocity of each link, etc, for each state, also include the target reference's corresponding state, and a motion phase)
    - reference motion: desired joint position → desired joint velocity is approximated by interpolation.
    - Reward → to get as close to reference motion as possible
2) Initial state is sampled along the desired trajectory → to explore state space uniformly around the reference trajectory

Thoughts

1) FEM approach → not related to real robotics for now
    1. model based approach (using inverse kinematics, etc is more promising)
2) Need to check a bit about basic reinforcement learning framework → step based policy update or episode based policy update.
3) Can simple neural network learn feedback?