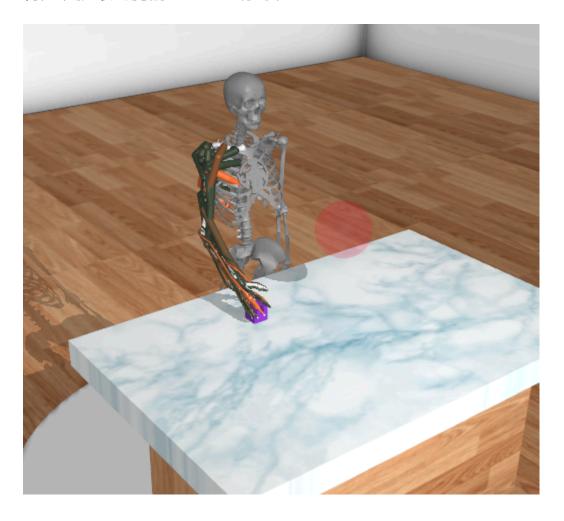
- MyoArm relocate_v1 env
 - 环境配置
 - 环境测试
 - register 参数说明
 - 文件
 - tensorboard使用

MyoArm relocate_v1 env

将方块抓取到胸前红色的目标球内



环境配置

(因为myosuite中是没有这个任务的,所以要把该任务的仿真文件等设置一下)

1、把arm文件夹复制到C:\Users\user\anaconda3\envs\myo\Lib\site-packages\myosuite\envs\myo\assets 下,即conda环境的myosuite包中的\envs\myo\assets位置

- 2、把assets文件夹复制到 C:\Users\user\anaconda3\envs\myo\Lib\site-packages\myosuite\simhive\myo_sim\arm 下,即conda环境的myosuite包中的\simhive\myo_sim\arm位置
- 3、在register中将model_path设置为1中 C:\Users\user\anaconda3\envs\myo\Lib\sitepackages\myosuite\envs\myo\assets\arm\myoarm_relocate_v1.xml的xml的路径

环境测试

运行envtest.py,看是否打开了Relocate-v1环境,arm以随机的肌肉激活乱动

register 参数说明

max_episode_steps=500 一个eposide的最大步数

'normalize_act': True, #是否将肌肉激活设置为-1~1

'pos_th': 0.1, # 目标点距离阈值范围,物体中心距离目标点多远算成功

'rot_th': np.inf, # 目标点角度旋转阈值范围,物体姿态离目标姿态差多少算成功,这里设置为无穷大,即不关心旋转,咋样都算成功

'target_xyz_range': {'high':[0.1, -.35, 1.2], 'low':[0.1, -.35, 1.2]}, #目标点的位置在每个eposide中是随机的生成的,high low设置的相同,那么就是固定点,否则就在这个范围内随机生成

'target_rxryrz_range': {'high':[0.0, 0.0, 0.0], 'low':[0.0, 0.0, 0.0]} #目标姿态的随机生 成范围

文件

relocate_v1.py 环境代码

PPO_train.py 循环训练PPO 每200000个timestep之后,测试一次,然后继续训练,共训练200000*50 = 10M 步

PPO_train_v2.py 直接训练PPO 10M步,进行测试

test.py 测试已经训练好的模型

plot.py 画肌肉激活曲线的

logs tensorboard的存储位置

models: 存储的训练模型

train_test: 训练过程中测试,肌肉激活曲线和视频 有一个文件夹的肌肉激活曲线都画到

一个图上了,啥也看不出来,另一个的分开画的

tensorboard使用

可以在训练过程中利用tensorboard查看实时效果

安装: pip install tensorboard==2.10.0

打开tensorboard:

tensorboard --logdir=(logs的存储位置)

如 tensorboard --logdir=logs

