控制任务计划

1.固定目标单步控制（目前）

2.固定目标多步控制

↓

3.移动目标多步控制

状态空间

state:{自由度[8]+末端位置[3]+目标位置[3]}

之前使用的是{末端位置[3]+目标位置[3]}
计划之后增加多个采样点不知道效果能不能更好

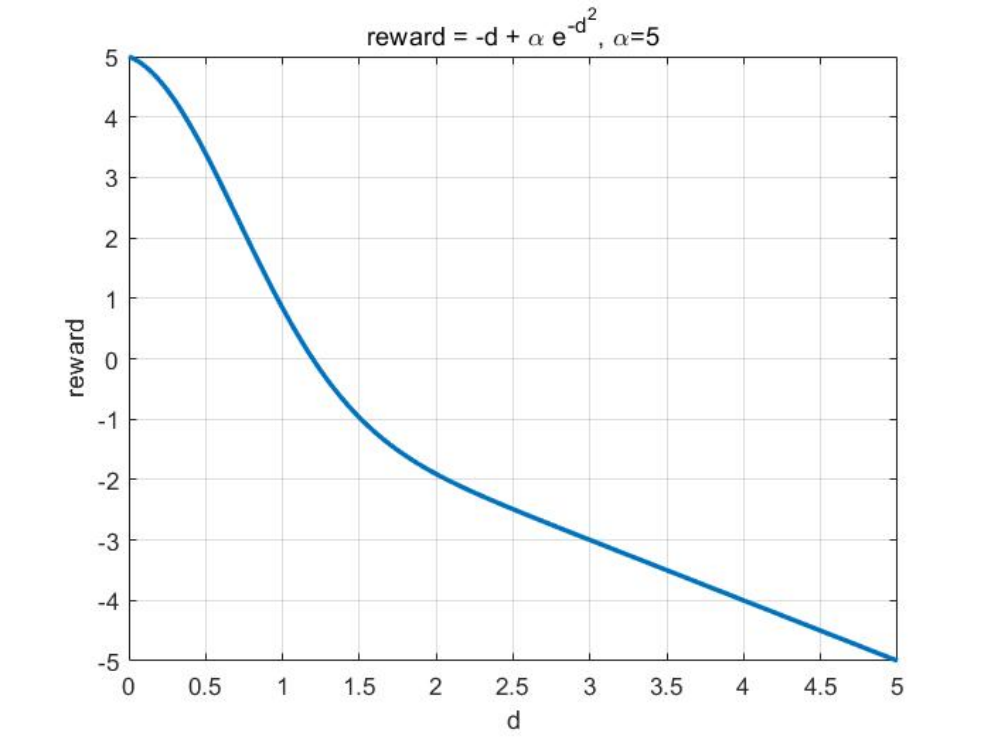
Elastica

The state S of the system is defined by the 44 element array: $S = [x_a, v_a, x_t, v_t]$, where x_a, x_t describe the three-dimensional position of the arm and the target, and v_a, v_t describe the magnitude and the direction of the velocity of the tip of the arm and the target. Note that x_a is constructed by taking multiple (in our case, 11) equidistant points along the arm to keep the dimensionality of the state low while maintaining enough information.

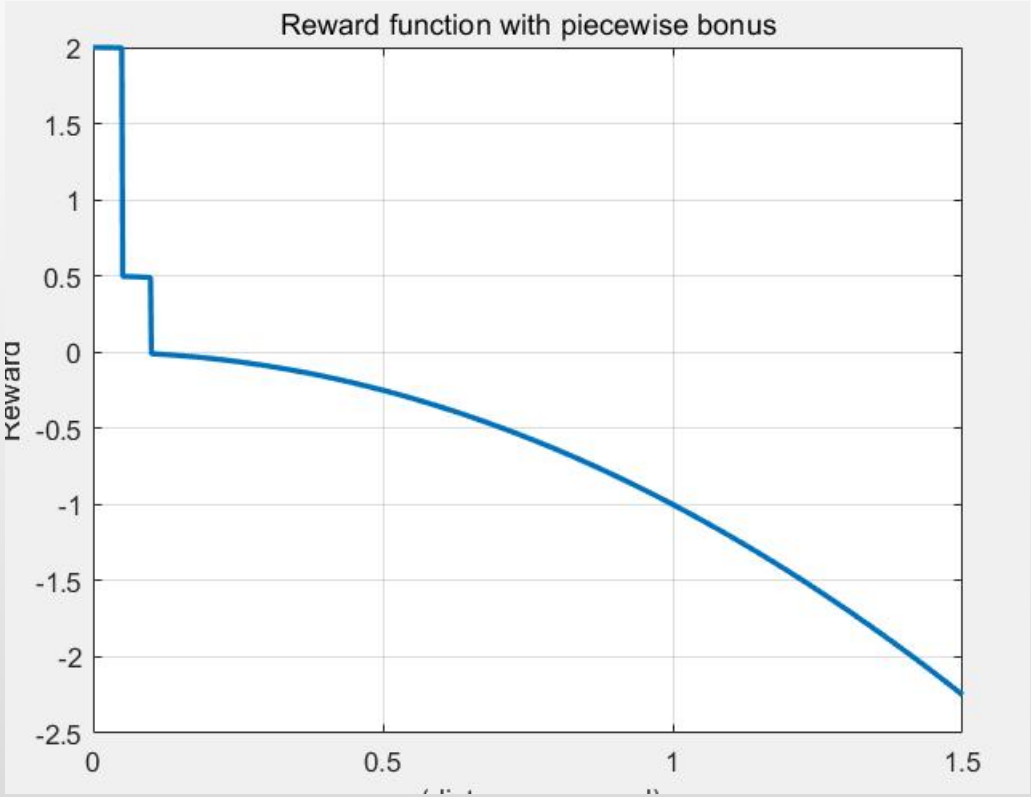
奖励函数

$$reward = -d + \alpha e^{-d^2}$$

其中d表示机器人末端到目标点的直线距离，α位常数，目前取5



Elastica

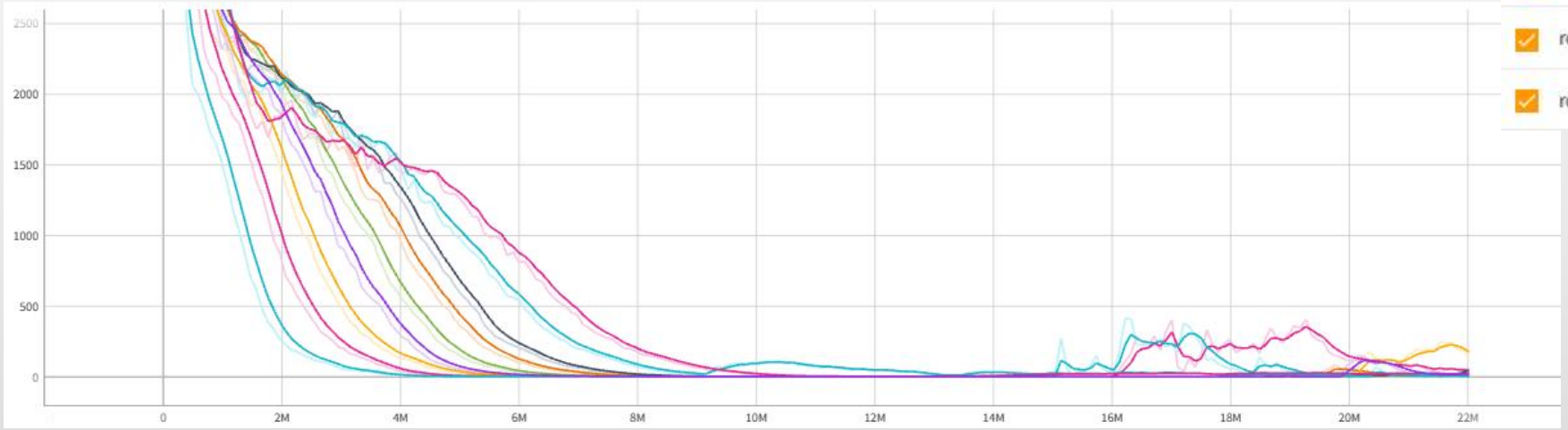


规格严格 功夫到家

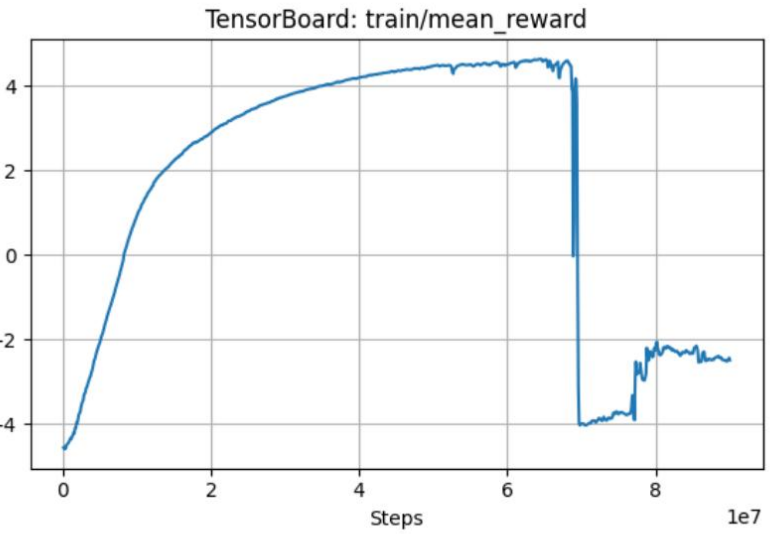
done函数

done=(reward>=α)

随着α的变大收敛变慢，在α在-0.5左右效果最好



现有最优效果



Model Name	./model/test_judge_3
Time	2025-04-23 23:52
Model_Type	PPO
seed	1
Timesteps	90000000
Control Mode	1
Device	cuda
Network Arch	[1024, 1024, 512]
Average Error	0.5524067835328776
batch	50000
buffer_size	100000
train_freq	4
learning_starts	20000
n_steps	1024
n_epochs	10
learning_rate	0.0003
n_env	96
best_reward	4.634823785664095

Reward Function Source

```
def my_custom_reward(distance):
    return -distance + np.exp(-distance**2) * 5
```

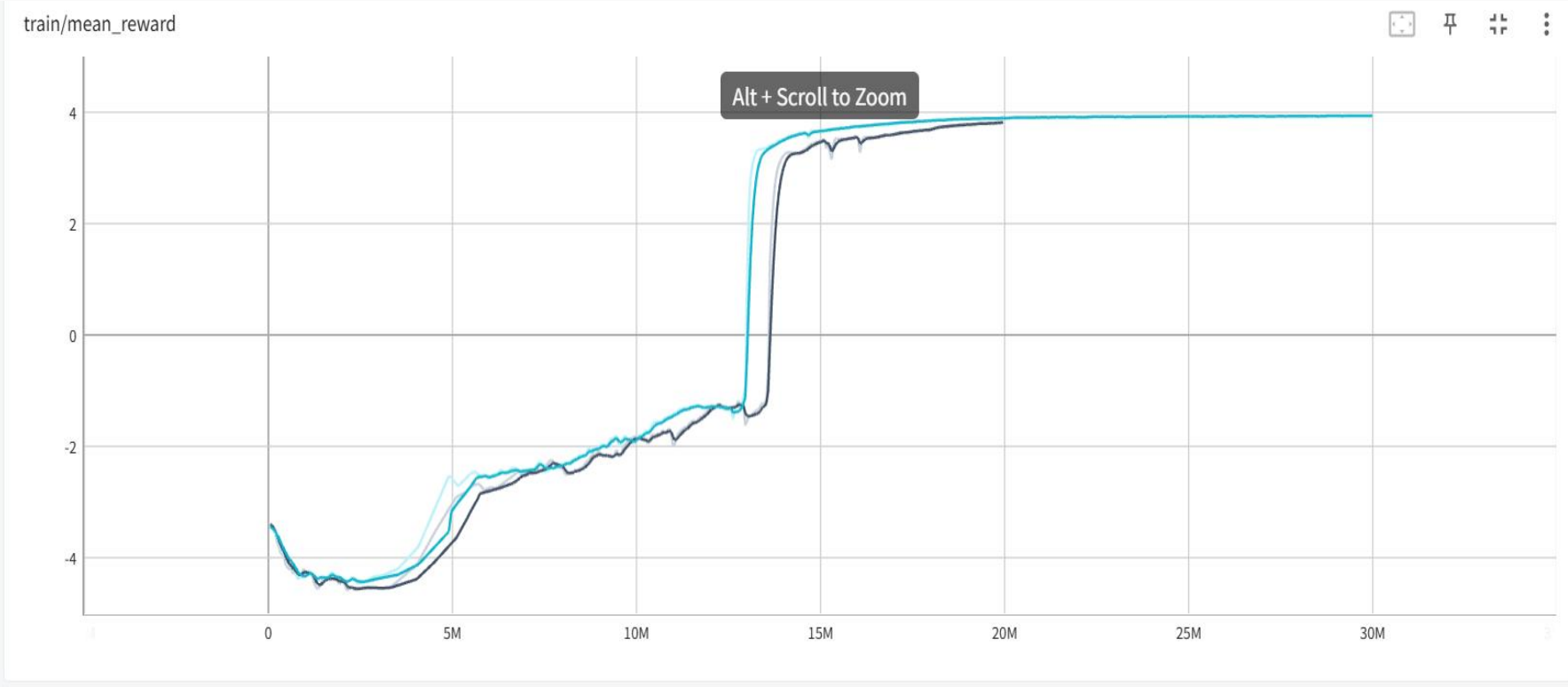
Done Function Source

```
def my_custom_done(reward, step,distance,in_step):
    return reward>=-0.5
```

其他模型（还在探索）

哈爾濱工業大學 HARBIN INSTITUTE OF TECHNOLOGY

SAC：



问题：
1.buffer回放机制
2.运算速度问题
在learning_start前速度和PPO持平，但是进入经验回放区后开始降速，最终fps只有2000上下（4090D+18vcpu)
且对gpu要求较高，在PPO中由于几乎对gpu没有要求，追求cpu常用vgpu32G+25vcpu，在训练sac进入经验回放区后fps下降飞快，最终只有1300上下

主要瓶颈/后续计划

1.始终没有办法做到更高精度
2.控制方式问题
3.运算速度/算力
4.其他模型：
on-pollicy中 TRPO好像已经不主流，在stable-baselines3中已经不被支持，且本身就是PPO的前身
off-pollicy中SAC, TD3, DDPG.之后尝试一下，
5.状态空间和奖励函数