**实验记录**

**2023-5-29**

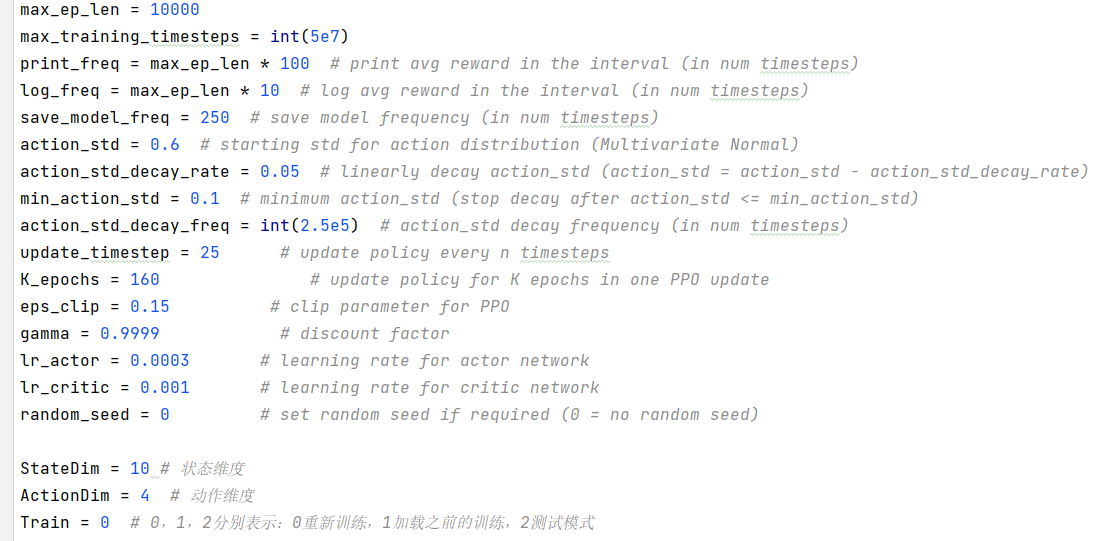
两机近距离追击任务

5-27训练了多想定，无法收敛，想定包含度方位，多航向差等等。

5-29重新训练了单想定，解决了距离公式的BUG，原来的distance通过经纬度计算，没有包含高度，精准度有问题，飞机容易向上飞行越界。修改了攻击角和逃逸角的计算公式。

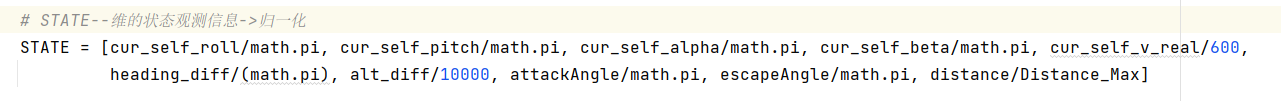
随机想定数量：1，方位0，相对朝向30度，可以收敛。在这个基础上又训练了相对朝向-30度的，发现问题：接近目标飞机时候，飞机有一定概率向下飞行坠落越界。初步怀疑是奖励函数攻击角的问题。攻击角不是越小越好，因为飞机会学习0度的攻击角，可能会导致这种概率问题。

**1.超参数设置**：



**2.状态空间**（共10维）：

自身俯仰角、自身滚转角、alpha、bata、自身速度、两机航向差、两机高度差、攻击角、逃逸角、两机距离



**3.奖励设计**

1. 对局结束奖励：
2. 高度越界（小于1000米或大于10000米）

TotalReward = -50 + math.tanh((1000 - distance) / 10000)

1. 超时（300s）

TotalReward = -50 + 0.8\*math.tanh((1000 - distance) / 10000) + 0.2\*math.tanh(-math.fabs(attackAngle) / math.pi)

1. 达到追击距离（1000米）以及攻击角小于30度

TotalReward = 50 + 0.6\*math.exp(-CurTime / 300) + 0.2\*math.exp(-(math.fabs(attackAngle))/math.pi) + 0.2\*math.exp(-(math.fabs(escapeAngle))/math.pi)

1. 单步奖励：

dis\_reward = math.tanh((pre\_distance-distance)/10)

alt\_reward = math.tanh((pre\_alt\_diff-alt\_diff)/10)

attack\_reward = math.tanh((pre\_attackAngle-attackAngle)\*100)

spd\_reward = math.tanh((distance/10000)\*(cur\_self\_v\_real-pre\_self\_v\_real))

if cur\_self\_alpha < -math.pi / 18 or cur\_self\_alpha > math.pi / 6:  
 alpha\_reward = -1

TotalReward = (0.5\*dis\_reward + 0.1\*alt\_reward + 0.3\*attack\_reward + 0.001\*spd\_reward + 0.1\*alpha\_reward)\*0.01

1. **实验结果**