日期：2025.3.5

Q1：reward未与上一步step形成关联，需调整reward方式。

A1：仅考虑当前参数与目标的绝对误差，缺乏对参数变化方向的引导。

方案1：

| **奖励成分** | **计算公式** | **作用** |
| --- | --- | --- |
| 绝对误差奖励 | 100 \* exp(-MSE) | 保证最终收敛到目标参数 |
| 方向奖励 | sign(Δerror)\*50 | 增加方向引导 |  |  |
| 复合奖励 | 0.7绝对+0.3方向 | 平衡长期目标与短期优化 |  |  |

效果：曲线仍不收敛

问题分析：

1.改进动作空间映射：将Actor输出直接映射到参数有效范围，避免后续clip操作；替换原有Actor类，初始化时传入param\_ranges ×参数仍然趋于边界值

2. 增强奖励函数：引入滞回曲线形状相似性奖励（占比30%）；使用指数衰减形式强化形状匹配的重要性 ×参数仍然趋于边界值

3. 动态边界惩罚：当参数进入边界10%区域时施加二次函数增长的惩罚；惩罚最大值限制为50，防止奖励值域失控 ×参数仍然趋于边界值后在边界值附近跳跃并再次趋于边界值

可能是 train() 函数中对 Actor 或 Critic 的反向传播更新存在错误？并未采样 buffer 来更新网络？没有编写损失函数来对网络进行反向传播？

Q2：建立word，及时保存并更新问题与改进方案。

Q3：可以尝试先固定部分参数，观察reward收敛情况。

A3：固定第三个参数，其余保持不变。

Q4：滞回曲线绘制太粗糙，增加数据点。

A4：使用插值（np.interp）生成2000个点。

日期：2025.3.12

Q1：利用插值形成的2000个点设计一个新的reward。

A1：初步设想reward = exp(1/(MSE+0.01)

问题假设：动作是对参数调整，奖励是根据数据点求得，奖励无法直接作用于参数，两个不同曲线求得奖励相同但是参数不同，拟合效率低。如何根据奖励引导参数方向变化需要考虑，需设计合理的reward让收敛更显著

Q2：检查代码是否存在问题。

A2：1.两个参数之间相互影响，再固定一个参数

2.重新设计一个代码，观察是否出现同样的情况

##### 1. 智能体（Actor-Critic结构）

- 演员网络（Actor）：

- 输入：当前状态（参数误差、系统状态等）。

- 输出：连续动作（待优化参数的调整量，如刚度、阻尼等）。

- 评论家网络（Critic）：

- 输入：状态和动作。

- 输出：Q值（评估当前动作对参数优化的贡献）。

##### 2. 环境设计

输入

- 目标滞回曲线：预先给定的实验数据或参考曲线（openseespy计算所得的滞回曲线代码）。

- 初始参数：待优化的模型参数（代码中的参数1和参数2）。

输出

- 模型预测曲线：根据当前参数生成的模拟曲线。

- 误差反馈：模型预测与目标曲线的差异。

##### 3. 动作空间（连续）

- 定义：直接输出参数的调整步长（如Δ参数1, Δ参数2）。

- 约束：

- 参数需在物理合理范围内（如参数1取（100，400），参数2取（50，200））。

- 使用tanh激活函数将动作映射到[-1,1]，再通过缩放因子（如乘以0.1）调整实际步长。

##### 4. 状态空间

- 当前参数值：所有待优化参数的当前值。

- 拟合误差：模型预测与目标曲线的误差（如逐点误差）。

- 系统状态：若模型具有动态特性（如当前加载步数、位移值等）。

- 时间步：用于判断训练进度（可选）。

##### 5. 奖励函数（使用参数误差奖励与曲线误差奖励）

- 核心目标：最小化模型预测与目标曲线的误差。

- 设计建议：

- reward = exp（1/（MSE+0.01））（MSE为相同数据点y与y`之间的均方误差）。

- 稳定性奖励：对参数调整的平滑性进行约束（如惩罚动作突变，暂时不考虑）。

##### 6. 训练流程

1. 初始化参数：随机初始化或基于先验知识设定初始值。

2. 环境交互：

- 智能体根据当前状态输出参数调整动作。

- 环境更新参数，生成新的模型预测曲线，计算误差和奖励。

3. 经验存储：将(状态,动作,奖励,下一状态) 存入经验回放池。

4. 模型更新：

- 定期从经验回放中采样批量数据，更新Actor和Critic网络。

- 使用目标网络和梯度裁剪提高训练稳定性。

5. 终止条件：误差收敛或达到最大训练步数（训练100回合，每一回合最大1000步）。

##### 7. 实现注意事项

- 数据预处理：对状态和奖励进行归一化（如标准化到[-1,1]）。

- 探索噪声：在动作中加入高斯噪声（如Ornstein-Uhlenbeck过程），平衡探索与利用。

- 离线训练：若目标曲线固定，可提前生成所有数据点，在训练中循环使用。

- 模型验证：训练完成后，在独立测试数据上验证参数泛化能力。

##### 试验1表现效果很好

##### 试验2表现效果一般

结论：两个参数直接相互影响导致学习达不到效果

解决方法：1.先后顺序，当参数1达到效果后再让参数2拟合（复用神经网络、迁移学习）

2.先将openseespy生成的曲线拆分成2000个数据点保存为csv文件，再导入这个文件生成目标曲线，再根据数据点形成新的奖励（2000个点相互影响可能很难达到效果）

新的问题：预训练模型与如何实现参数识别

日期：2025.3.19

Q1：数据点

A1：使用opensees生成一份excel文件，从文件导入数据并生成曲线，使用force二范数（error\_norm）作奖励

Q2：误差梯度

A2：调整超参数、网络结构、奖励函数、噪声探索 表现效果均一般

Q3：终止条件

A3：达到最大step即停止，后续可增加一个停止条件，当error\_norm小于某一值时停止

日期：2025.4.2

Q1：减少step

Q2：到达边界参数即终止episode，开始下一episode

Q3：不设边界参数