基于DDPG的位置控制强化学习实验

1. 实验目的

- 理解并实践 DDPG 在无人机位置控制中的应用。
- 熟悉 RflySim 工具链与 MATLAB/Simulink 软件在环 (SIL) 流程。
- 掌握通过 RL Agent 生成 X 向加速度指令的控制逻辑与评估方法。

2. 实验要求

- 软件要求：Windows 10 及以上；RflySim 工具链；MATLAB 2022b 及以上；QGC、CopterSim、RflySim3D。
- 硬件要求：笔记本/台式机 1 台（推荐配置参见 安装指南）。

3. 实验地址

例程目录：`[安装目录]\RflySimExtCtrl\2.AdvExps\e5.RLExps\1.Simulink`

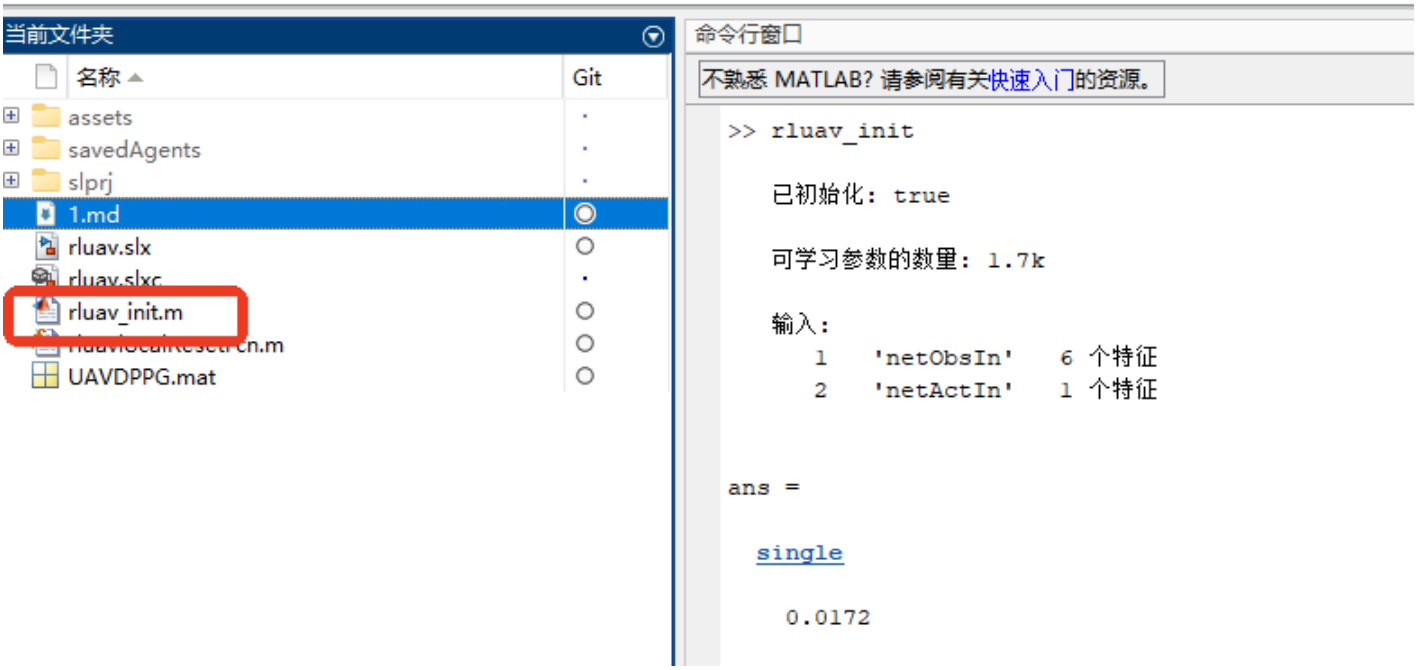
主要文件：

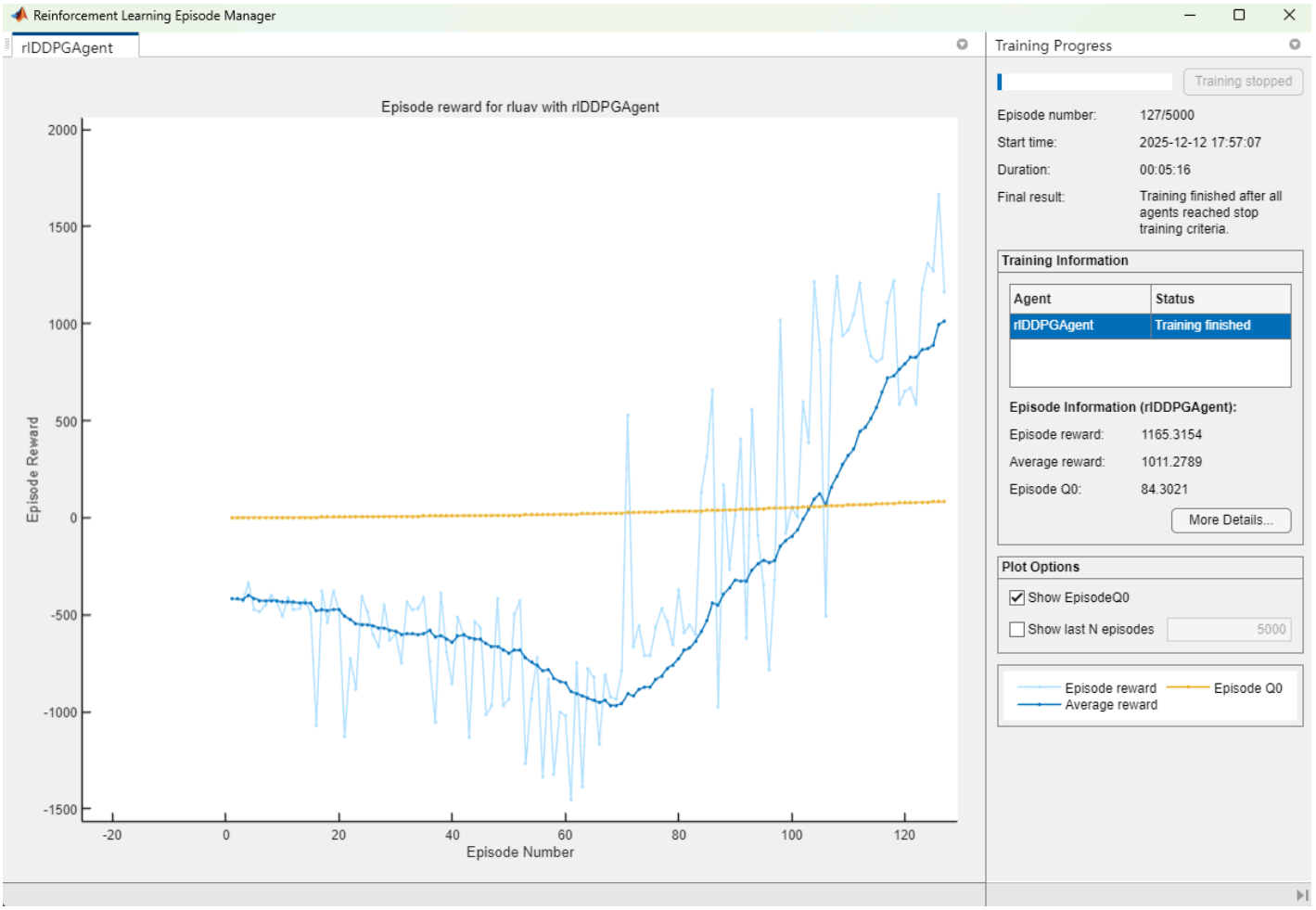
- `rlhighDynamicAcc.slx`：SIL 模型，包含 RL Agent 位置控制。
- `rlhighDynamicAcc_init.m`：模型初始化脚本，加载预训练 UAVDPPG.mat。
- `train/rluav_init.m`：训练脚本示例，可复现实验训练流程。
- `train/rluav.slx`：训练用环境模型，rlSimulinkEnv 绑定的 Simulink 场景。
- `train/rluavlocalResetFcn.m`：训练复位函数，重置初始状态与积分量。
- `train/savedAgents/`：训练过程中按阈值自动保存的 Agent 目录。
- `RflyUdpMavlinkRealSim.bat`：一键启动仿真环境。
- `UAVDPPG.mat`：预训练 DDPG Agent 权重。

4. 实验内容或步骤

4.1 启动训练并得到权重

在MATLAB 中运行 `train/rluav_init.m` 脚本，等待训练完成。





如图所示，训练提前在第 127/5000 回合结束，满足停止条件（reward/Q0 达到阈值）。
Episode reward（蓝色）前期约 -400~-1200，约在 80 回合后快速提升，最终单回合奖励 ~1165。
平均回报（深蓝）在 80 回合后持续上升，收敛到 ~1011。
Episode Q0（黄线）缓慢抬升到 ~84，说明 Critic 对初始状态价值估计在变好。
总体趋势：策略在中后期显著改善，已达到可用的收敛水平。

将训练得到的模型权重 `UAVDPPG.mat` 拷贝到train目录外面用于仿真验证

4.2 闭环仿真验证

SIL 标准流程速览：

1. 双击运行 `RflyUdpMavlinkRealSim.bat`，等待 `GPS 3D fixed & EKF initialization finished`。
2. 打开并运行 `rlhighDynamicAcc.slx`，按上述模式切换完成起飞与操作。
3. 结束时依次 Land → Disarm → 停止模型 → 退出仿真程序。

4.2.1 启动仿真环境

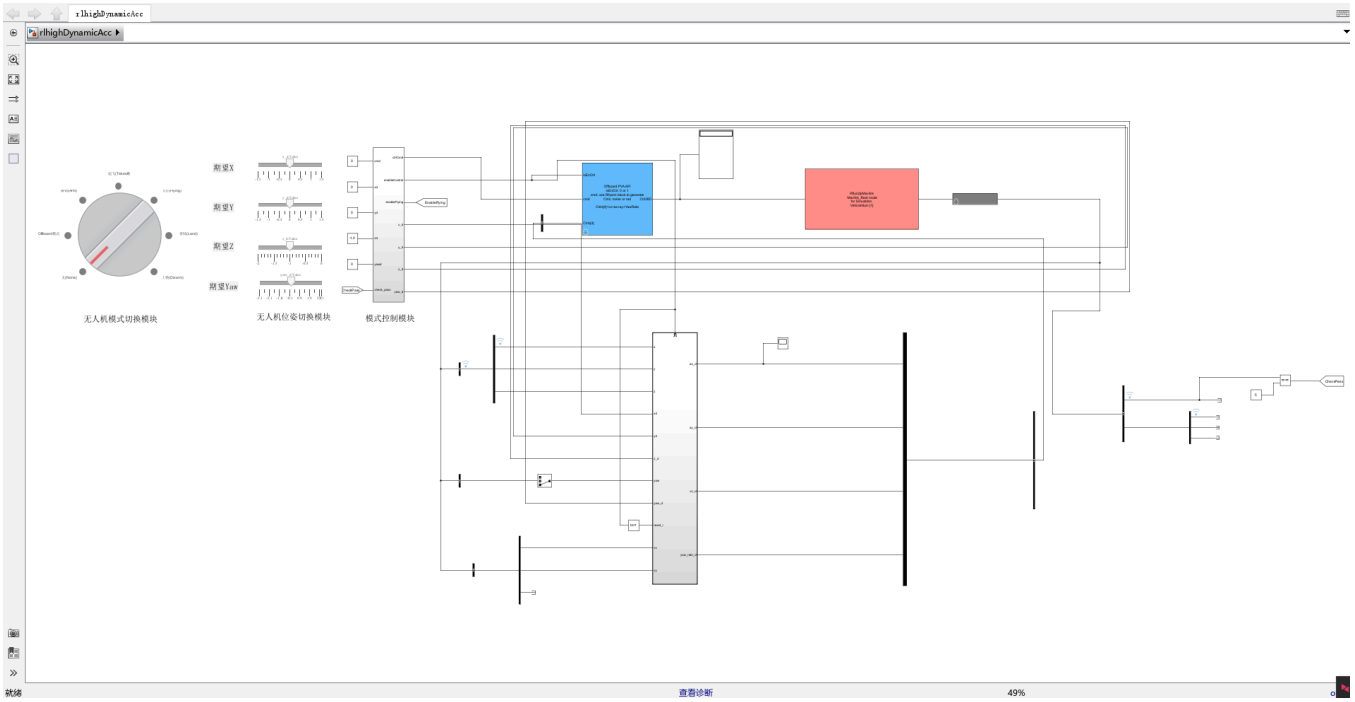
1. 双击 `RflyUdpMavlinkRealSim.bat`，等待 CopterSim 日志出现 `GPS 3D fixed & EKF initialization finished`。
2. 确认 QGC、CopterSim、RflySim3D 均已启动且未报错。

4.2.2 打开并运行模型

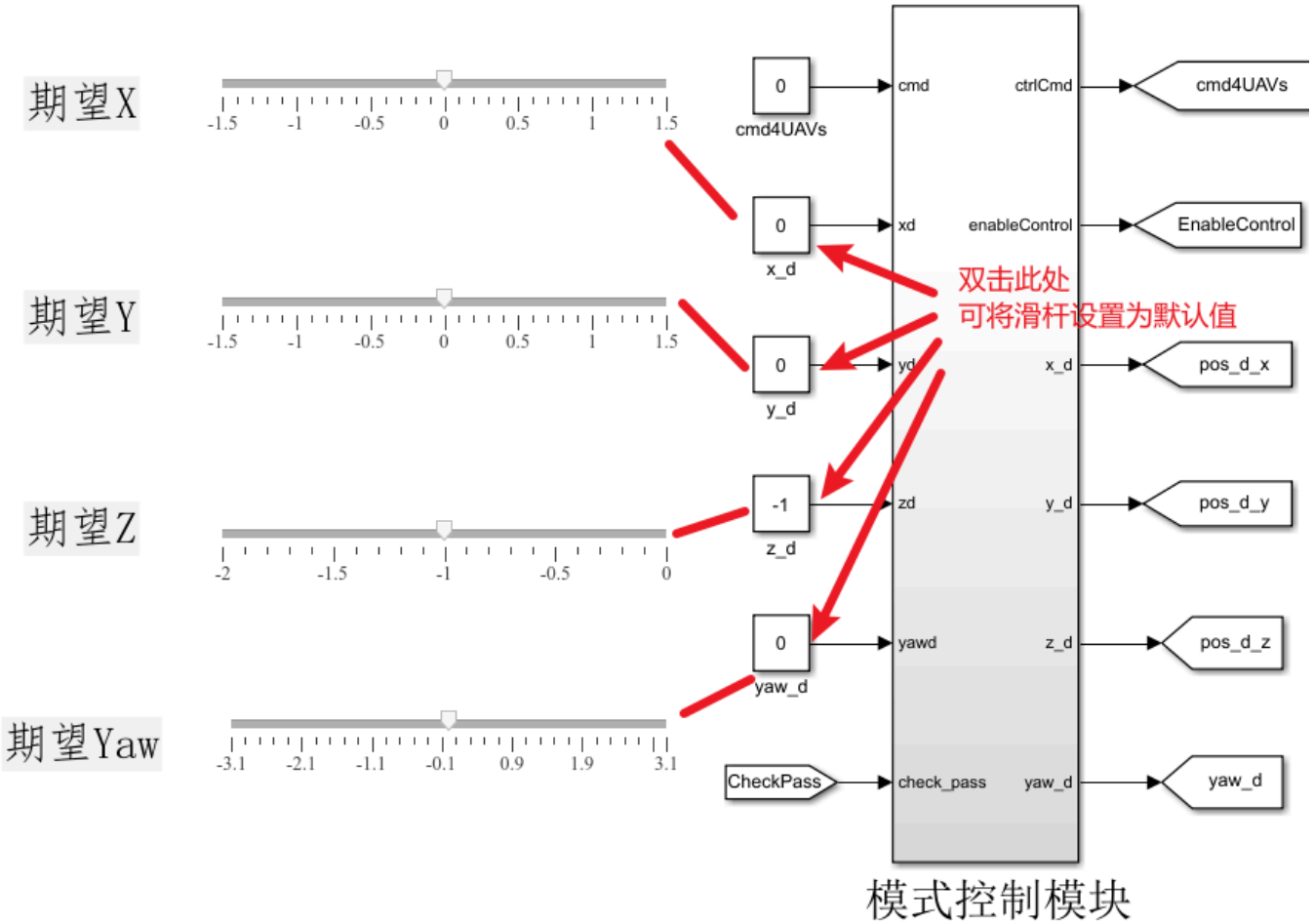
1. 双击 `rlhighDynamicAcc.slx` 打开模型，等待加载完成。
2. 运行前确保滑杆默认值：期望 X/Y/Yaw 为 0，期望 Z 为 -1（可在常量块中确认）。
3. 点击 Simulink “运行”按钮，等待所有状态灯转为绿色。

4.2 打开并运行模型

1. 双击 `rlhighDynamicAcc.slx` 打开模型，等待加载完成。



2. 运行前确保滑杆默认值：期望 X/Y/Yaw 为 0，期望 Z 为 -1（可在常量块中确认）。

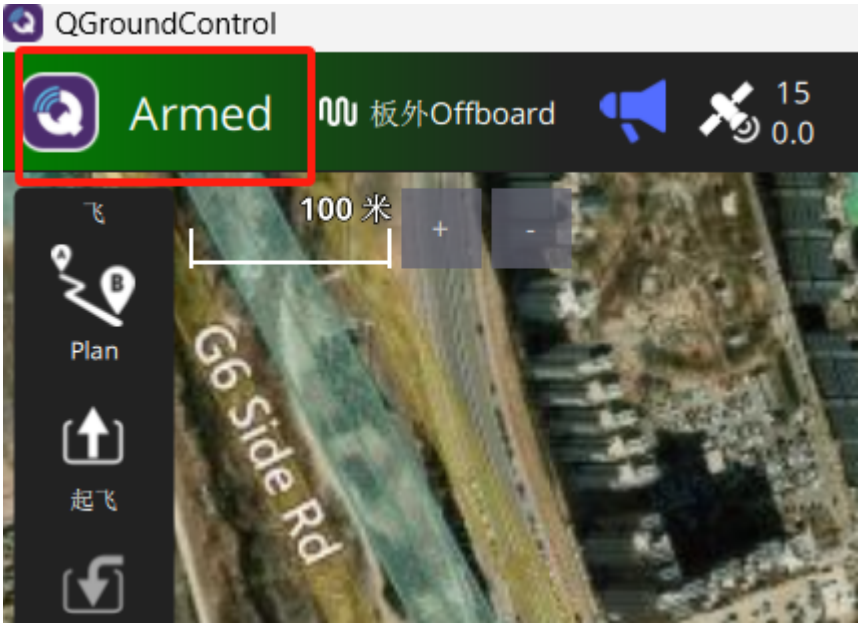


3. 点击 Simulink “运行”按钮。

4.2.3 模式切换与起飞

（1）把模式切换旋钮切至Offboard档，QGC中飞机进入Ready To Fly状态，且进入板外Offboard模式。

（2）模式切换旋钮切至解锁(Arm)档，飞机进入Armed状态（解锁状态）。

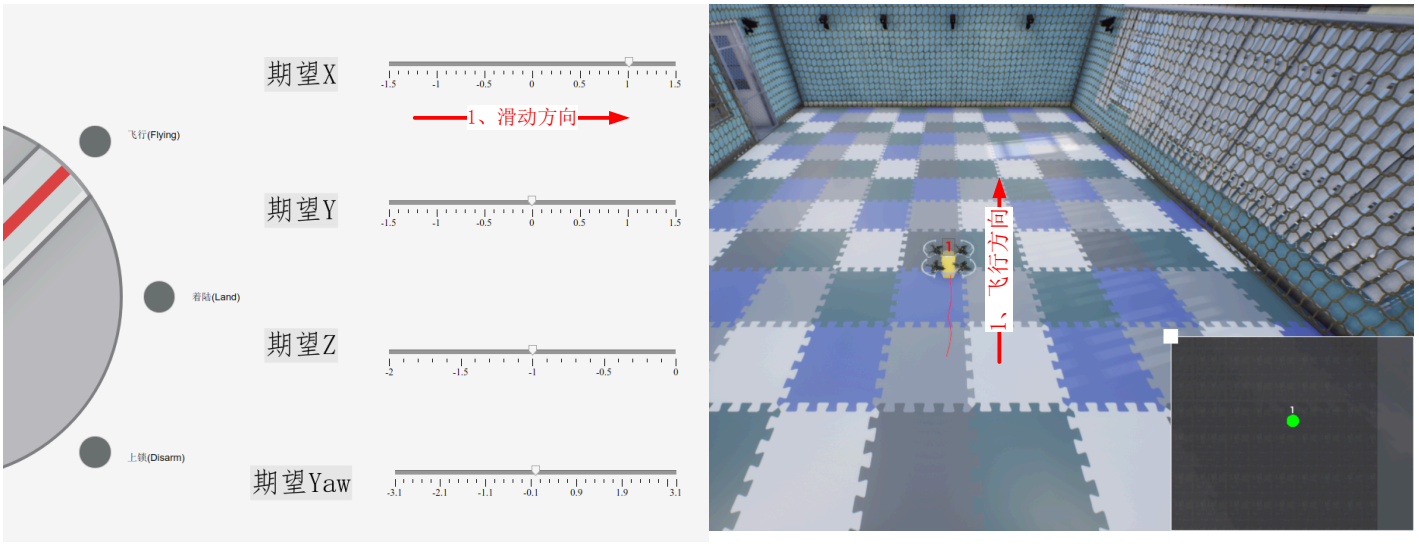


（3）模式切换旋钮切至起飞(Takeoff)，飞机将起飞并悬停，QGC显示进入Flying状态。

4.2.4 滑杆控制与 RL 作用

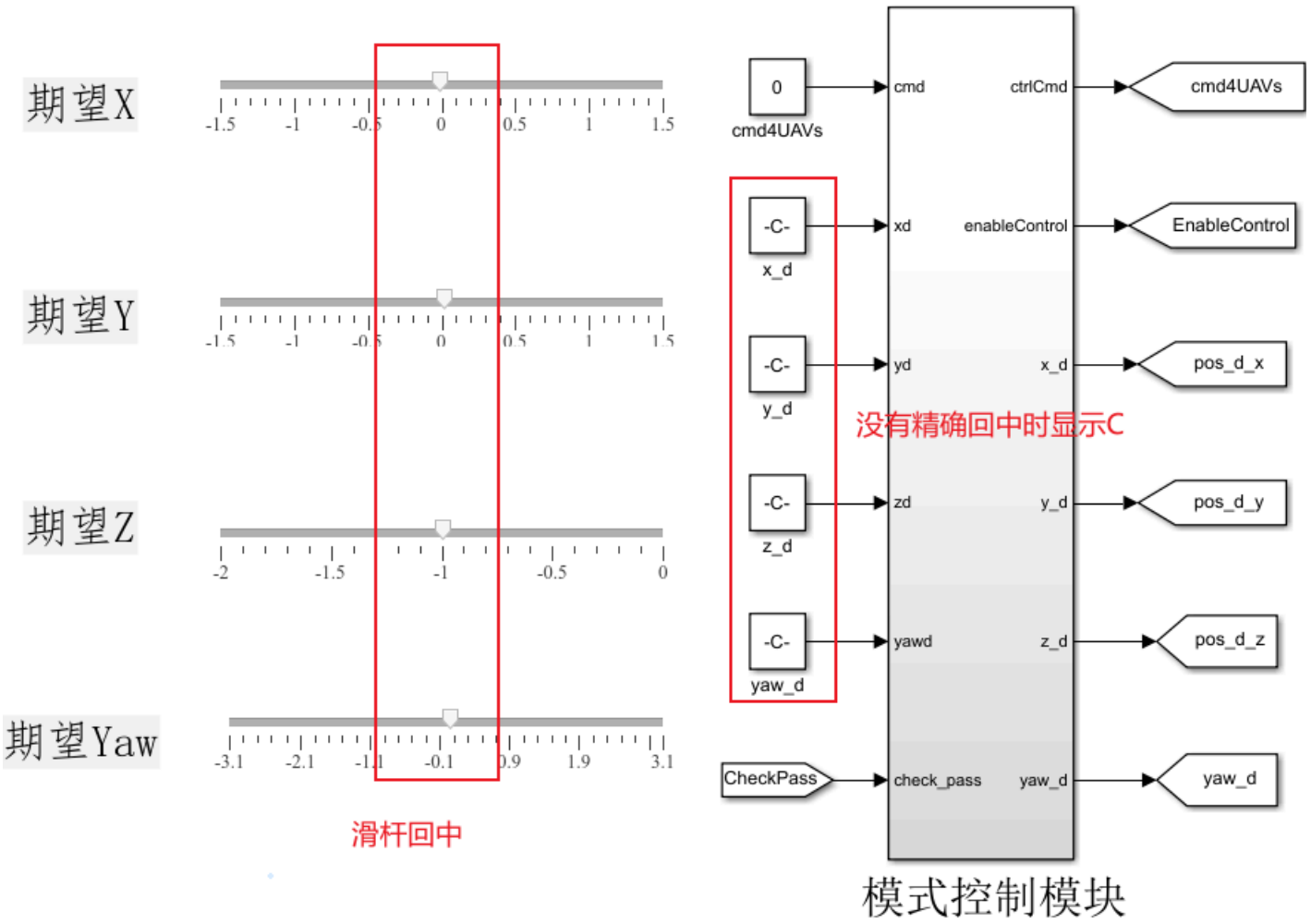
（1）模式切换旋钮切至飞行(Flying)档，QGC显示飞机继续保持Flying状态，此时用户修改的位置才会生效。

（2）滑动设置期望X的滑杆至大约1m处。可通过鼠标左键在滑杆1m处单击实现滑动，可在RflySim3D中观察到飞机沿着X方向运动。反复移动滑杆，可看到飞机在X轴上始终较平滑地到达期望位置。



（3）同理设置期望Y、期望Z、期望Yaw的滑杆。也可以观察到飞机在相应方向的移动。

（4）如图 15所示，将期望X、期望Y、期望Z和期望Yaw回中，此时期望X、期望Y和期望Yaw的值接近0，期望Z的值接近-1。**只有回中后飞机才能安全降落**。



旋钮切至着陆(Land)。着落之前请确保滑杆已经回中，允许回中略有偏差。降落后，飞机的螺旋桨仍

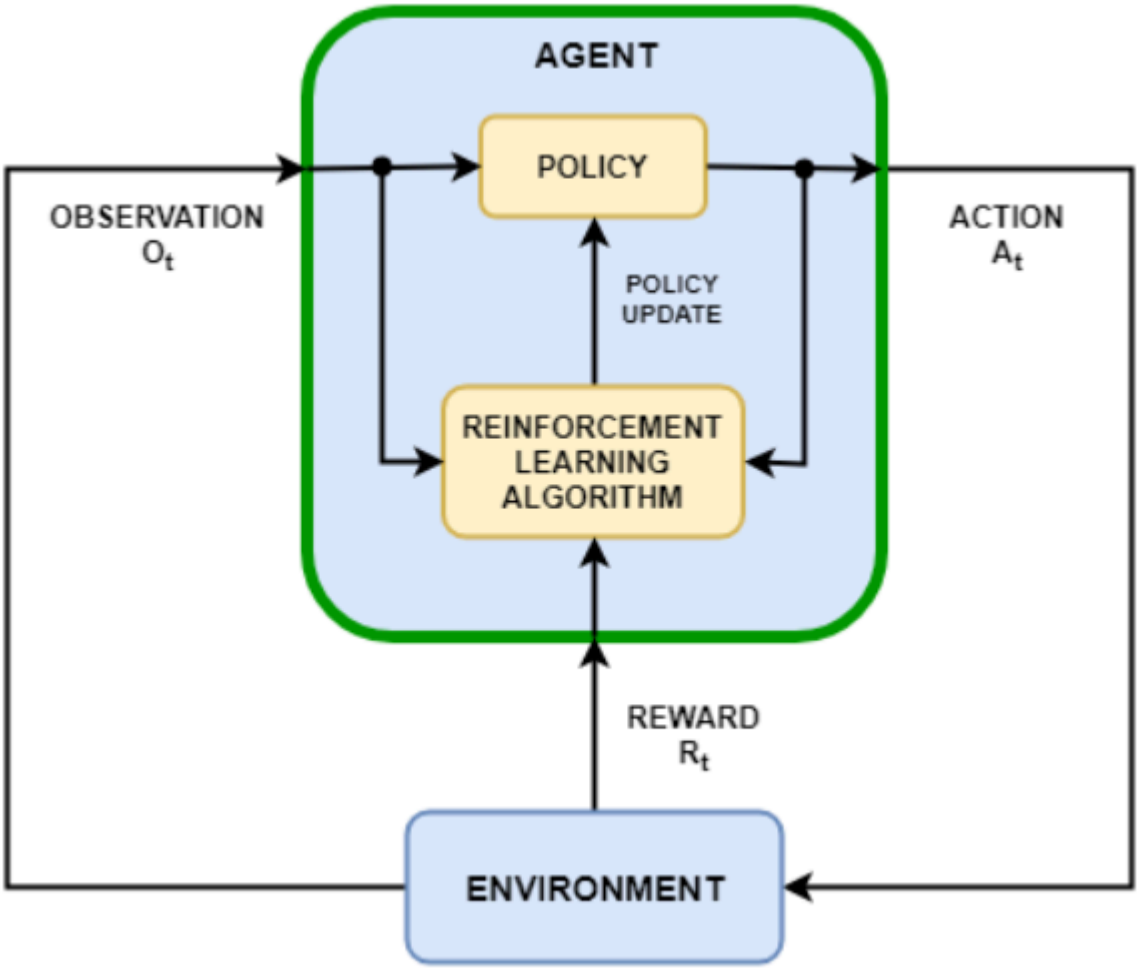
然在旋转。
，切换旋钮切至上锁(Disarm)。当无人机接近地面时上锁，模式切换旋钮切至上锁(Disarm)，飞机将上锁并螺旋桨停止旋转。结束MATLAB控制程序并退出在环仿真。

5. 关键知识点

强化学习中的基于值的强化学习（Value - Based）、基于策略的强化学习（Policy -Based）以及Actor -Critic方法是三种重要的类别，下面从原理、算法特点、优缺点和应用场景等方面进行对比介绍：

1. 基于值的强化学习。核心思想是通过估计每个状态或状态-动作对的值函数来间接得到最优策略。值函数表示从某个状态或执行某个动作后，遵循特定策略所能获得的累积奖励的期望。常见的值函数有状态值函数$V(s)$和动作值函数$Q(s, a)$。智能体通过不断更新值函数，最终根据值函数来选择具有最大价值的动作，即遵循贪心策略。Q-Learning是一种基于值的强化学习方法。
2. 基于策略的强化学习。直接对策略进行参数化表示，通过优化策略的参数来最大化累积奖励。策略函数可以表示为一个概率分布，例如策略梯度算法(PG)。
3. Actor–Critic是值方法与策略方法的结合。Actor负责根据当前状态选择动作，其策略同样可以表示为一个概率分布。Critic则负责评估Actor选择的动作的价值，通过估计值函数来给出反馈。Actor根据
   Critic的反馈来更新自己的策略，从而实现更高效的学习。DDPG、TD3、SAC、PPO、TRPO和MBPO等Agent都是基于Actor
   – Critic框架设计的。

强化学习的目标是训练智能体在不确定的环境中完成任务。在每个时间间隔，智能体接收来自环境的观察和奖励，并向环境发送一个动作。奖励是衡量先前的行动（从先前的状态中采取）在完成任务目标方面有多成功。智能体和强化学习算法两部分：策略是从当前环境观察到要采取的行动的概率分布的映射在代理内部策略由参数可调的函数逼近器和特定的逼近模型如深度神经网络来实现。学习算法根据行动、观察和奖励不断更新策略参数。学习算法的目标是找到一个最优策略，以最大化在任务期间获得的预期累积长期奖励。



DDPG

深度确定性策略梯度（DDPG）算法是一种无模型、在线、非策略的强化学习方法。DDPG代理是一种Actor–Critic强化学习代理，它搜索最大化预期累积长期回报的确定性的最优动作 A。

Actor 用 $\pi(S;\theta)$ 表示，其中 S 表示观测量，θ 表示参数，将返回使得长期奖励最优化的动作。为了改善优化过程中的稳定性，定义目标 Actor $\pi_t(S,\theta_t)$，目标网络是周期性从 θ 中更新最新参数。目标网络的参数是主网络参数的缓慢更新版本，用于计算目标值，减少训练过程中的波动。

Critic 用 $Q(S,A;\varnothing)$ 表示，其中 S 表示观测量，A 表示动作，$\varnothing$ 表示参数。在给定观测量和动作的前提下返回期望的长期奖励。类似地，为了减小训练过程中的参数波动，定义目标值函数 $Q_t(S,A;\varnothing_t)$。

强化学习中，首先需要设置一些初始化条件。例如强化学习一个无人机高度控制器，无人机的初始高度和期望高度等在学习的过程中都需要进行调整，这样才能保证学习到的高度控制器具有一定的泛化能力。

在强化学习训练过程中添加噪声是一项至关重要且具有多方面意义的操作。首先，它能极大地助力智能体探索策略空间。在训练初期，智能体对环境认知有限，若缺乏噪声，容易过早收敛至局部最优策略。而噪声的加入使智能体有机会尝试更多原本不会选择的动作，在探索与利用之间达成平衡，增加策略的多样性，进而有更大可能发现全局最优策略。其次，添加噪声有助于防止过拟合。它能增加训

练数据的多样性，让智能体学习到更具通用性的特征和策略，避免过度依赖特定环境细节，提高模型的泛化能力。再者，可增强模型的鲁棒性。真实环境充满不确定性与干扰，训练时添加噪声能模拟这种情况，使智能体学会在不同噪声条件下有效执行任务，提升应对干扰的能力。此外，噪声还能加速训练收敛。它能为陷入局部最优的智能体提供
“扰动”，引导其朝着更有潜力的方向探索，改变策略梯度估计，使智能体更快地找到接近全局最优的策略，提高训练效率。

MATLAB强化学习训练

MATLAB强化学习模块需要以指令的方式运行，通过一系列指令创建强化学习agent，然后再使用Simulink中的RL
Agent模块将通过命令所创建的agent与Simulink仿真关联起来。

本实验训练流程示例（对应 rluav_init.m）：

- 环境与规格：步长 Ts=0.02 s，单回合时长 Tf=20 s；观测 6 维（误差积分、误差、测量位置、测量速度、速度积分、速度微分，带上下界），动作 1 维 accx∈[-4,4]。通过 rlSimulinkEnv 绑定模型并指定 rluavlocalResetFcn 作为复位函数。
- 网络结构：Critic 采用双支路（状态 6→50→25，动作 1→25，add 后 25→1）；Actor 为 6→3→1，tanh 输出动作。均由 layerGraph 搭建并转换为 dlnetwork。
- Agent 与噪声：使用 rlDDPGAgent，OU 噪声方差 0.1、衰减 1e-5；折扣因子 1.0，目标平滑 1e-3，经验回放 1e6，MiniBatch 64；学习率 Critic 5e-4、Actor 5e-5，梯度裁剪 1。
- 训练参数：rlTrainingOptions 设定最大 5000 回合、每回合步数 ceil(Tf/Ts)，平均奖励窗口 20，奖励阈值 1000 触发提前停止；当单回合奖励≥1000 自动保存 agent（SaveAgentCriteria=EpisodeReward）。
- 执行流程：若 doTraining=true 则 train(agent,env,trainOpts) 开始训练，完成后保存 UAVDPPG.mat；否则直接加载预训练权重。训练后可用 sim(env,agent,simOpts) 回放单回合验证。
- 代码位置：见 train/rluav_init.m，可调节噪声、回放长度、学习率、回合数后运行脚本。

简化运行示例（核心片段）：

```
run("rluav_init.m");          % 构建环境、Agent，并按 doTraining 开始训练或直接加载
% 可选：调整超参后再运行
agent.AgentOptions.NoiseOptions.Variance = 0.1;
agent.AgentOptions.CriticOptimizerOptions.LearnRate = 5e-4;
save("UAVDPPG.mat","agent"); % 训练完成保存权重
experiences = sim(env,agent,rlSimulationOptions(MaxSteps=ceil(Tf/Ts)));
```

Actor-Critic在MATLAB中的表示方法：

- $$Q(S, A; \varnothing)$$

  → rlQValueFunction（单输出 Critic）

- $Q_i(S, A_i; \varnothing_i)$

  → rlVectorQValueFunction（多输出 Critic）

- $\pi(S; \theta)$

  → rlContinuousDeterministicActor（连续动作 Actor）

- 离散动作 → rlDiscreteCategoricalActor

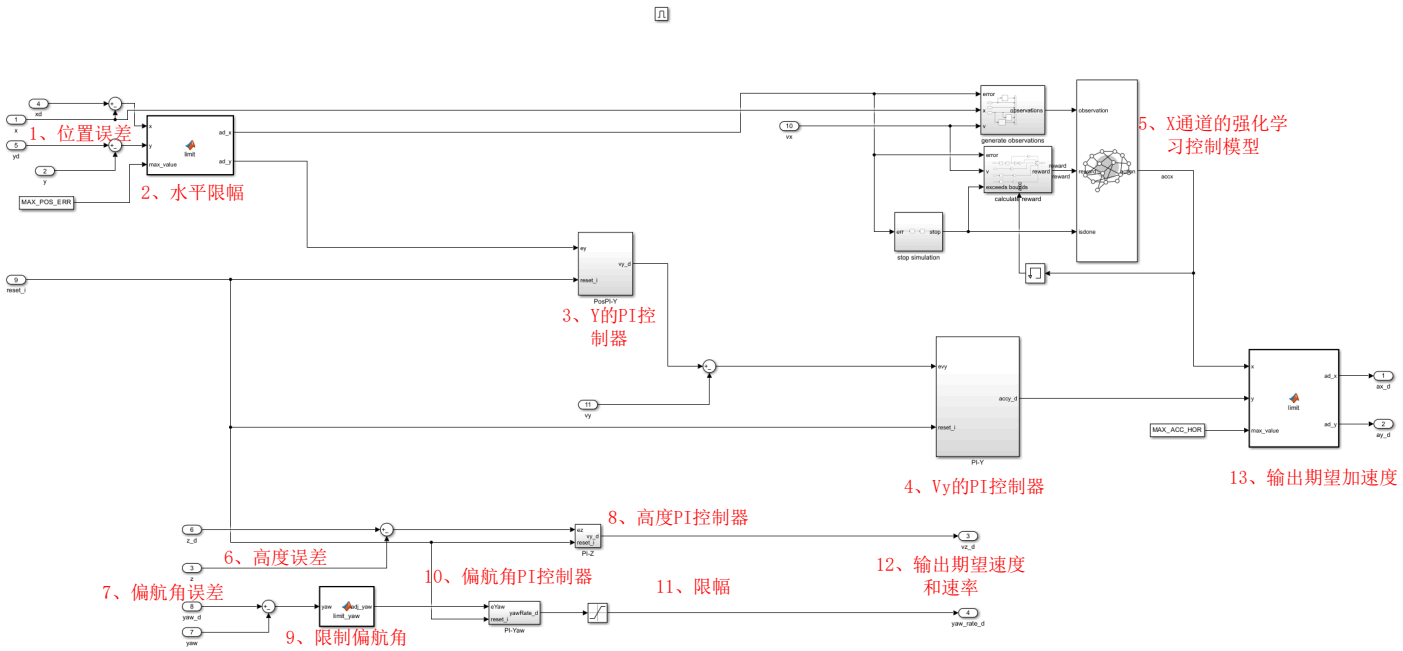
rlSimulinkEnv 创建环境示例：

```
% 四参形式：显式绑定观测/动作规格
env = rlSimulinkEnv(mdl, agentBlocks, obsInfo, actInfo);

% 二参形式：规格已在工作区创建时使用
env = rlSimulinkEnv(mdl, agentBlocks);

% 复位函数示例
env.ResetFcn = @(in)localResetFcn(in);
```

layerGraph 使用要点：

- layerGraph() 创建空图；layerGraph(layers) 从层数组创建并顺序连接；layerGraph(net) 从已有网络创建。
- addLayers 添加层数组，跨数组的层需用 connectLayers(src,dst) 显式连接。
- 完成拓扑后用 dlnetwork 转为可训练网络对象。

闭环验证



本实验的位置通道在 X 轴使用 RL Agent Block 生成水平加速度指令，替代传统前馈或 PID，整体机制如下：

- 状态观测（6 维）：位置误差积分、当前位置误差、当前测得位置、当前测得速度、速度积分、速度微分，均在生成观测模块内归一化/限幅后送入智能体。
- 动作输出（1 维）：水平加速度 accx，范围 [-4, 4]，经限幅模块与 MAX_ACC_HOR 约束后再送往后级动力学/控制。
- 奖励与终止：calculate reward 依据误差、越界情况生成 reward；stop simulation 与 exceeds bounds 信号用于回合终止，保证训练与推理安全。
- 智能体结构：DDPG（确定性策略），Actor 为两层全连接+tanh 输出 1 维动作，Critic 将状态与动作分别编码后相加回归 Q 值；采样时间 0.02 s。
- 训练与权重：初始化脚本会加载预训练模型 UAVDPPG.mat，并在 rlhighDynamicAcc_init.m 中设置观测/动作规格、网络结构与超参数。

图示位置如上蓝框所示，RL Agent Block 位于生成观测、奖励、终止信号与限幅模块之间，直接为 X 轴位控生成加速度指令；Y、Z、Yaw 通道仍由常规 PI 控制完成。

进一步细节：RL Agent 采样时间 0.02 s，基于 DDPG（确定性 Actor + Q Critic），Actor 网络为 6→3→1 的两层全连接并用 tanh 输出动作，Critic 将 6 维状态与 1 维动作分别编码后相加回归 Q 值。启动仿真时脚本加载预训练权重 UAVDPPG.mat（工作区变量名 agent），并在 rlhighDynamicAcc_init.m 中设定噪声方差/衰减、折扣因子、经验回放长度、MiniBatch、优化器学习率与梯度裁剪等超参数。

关键配置代码（节选）：

```
Ts = 1/50;                % 0.02 s 采样
obsInfo = rlNumericSpec([6 1], ...);
actInfo = rlNumericSpec([1 1], 'LowerLimit', -4, 'UpperLimit', 4);
% Critic: 状态支路 6->50->25，动作支路 1->25，add 后 25->1
% Actor: 6->3->1，tanh 输出
agent = rlDDPGAgent(actor, critic);
agent.SampleTime = Ts;
agent.AgentOptions.TargetSmoothFactor = 1e-3;
agent.AgentOptions.DiscountFactor = 1.0;
agent.AgentOptions.MiniBatchSize = 64;
agent.AgentOptions.ExperienceBufferLength = 1e6;
agent.AgentOptions.NoiseOptions.Variance = 0.3;
agent.AgentOptions.NoiseOptions.VarianceDecayRate = 1e-5;
agent.AgentOptions.CriticOptimizerOptions.LearnRate = 1e-03;
agent.AgentOptions.CriticOptimizerOptions.GradientThreshold = 1;
agent.AgentOptions.ActorOptimizerOptions.LearnRate = 1e-04;
agent.AgentOptions.ActorOptimizerOptions.GradientThreshold = 1;
load("UAVDPPG.mat", "agent"); % 载入预训练权重
```

UAVDPPG.mat 含义：

- 保存了上述 DDPG 智能体的预训练权重（Actor 与 Critic 网络参数）。
- 仍沿用 rlhighDynamicAcc_init.m 的观测/动作规格与超参数，mat 文件主要替换网络权重而非重新定义超参。
- 加载后直接推理，无需再训练；若需复训，可在此初始化基础上继续采样和更新。

6. 参考资料

1. RflySim 官方文档
2. MATLAB Reinforcement Learning Toolbox 文档
3. DDPG 原始论文 Lillicrap et al., 2015

7. 常见问题

Q1：仿真无响应或未起飞

A1：按标准流程重启仿真；确认杀毒软件关闭；确保 CopterSim 日志出现 GPS 固定提示。

Q2：MATLAB 版本提示或 slpj 缓存报错

A2：在模型目录删除 slprj/slpj 缓存；MATLAB 2022b 及以上可正常运行。

Q3：视角看不到飞机

A3：RflySim3D 视角滚轮调节即可恢复机体视图。