

基于IsaacSim训练无人机跟随圆形案板实验

1. 实验目的

训练完成后，本例程生成一个可由键盘控制的圆形案板，通过视觉接口获取 RflySim3D 图像，进行图像处理并计算控制指令，使无人机实时跟随案板移动；适用于视觉控制与仿真验证教学与调试。

2. 实验要求

- 软件要求：Windows 10 及以上、RflySim 工具链、推荐安装 WSL2-GPU（用于训练）、Python 3.8 环境。
- 硬件要求：台式机/笔记本 1 台，推荐带 GPU（用于训练）。
- 开发工具：VS Code（可选）。

推荐参考：<https://rflysim.com>

3. 实验地址

例程路径：`[安装目录]\RflySimAPIs\6.RflySimExtCtrl\2.AdvExps\6.IsaacSimExps`

`RL_attctrl/`: 包含圆形跟随控制相关的脚本和配置文件

`RL_attctrl/circle_follow.bat`: 一键启动仿真相关程序（CopterSim、QGC、RflySim3D）

`RL_attctrl/circle_follow.py`: 示例控制程序，实现视觉检测与跟随逻辑

`RL_attctrl/Python38Run.bat`: 用于启动 Python 环境

`RL_attctrl/Config.json`: 视觉传感器配置

`RL_attctrl/rl.py`: 强化学习策略推理与控制逻辑（加载训练得到的权重 `model_*.pt`，输出跟随控制量；需按实际权重文件名在脚本内修改加载路径/文件名）

`RL_attctrl/cont.py`: 控制算法相关代码

`RL_attctrl/model_*.pt`: 训练好的神经网络模型权重文件

[index.html](#): 主页入口文件

[WinWSL2.bat](#): 启动WSL2环境的批处理文件

[Readme.md](#): 项目中文说明文档

[Readme_en.md](#): 项目英文说明文档

4. 实验内容与步骤

4.1 步骤1:准备训练环境并完成跟踪训练

创建训练环境

下载isaacsim基本训练环境和资源包以及IsaacLab 训练工程。

链接: https://pan.baidu.com/s/1yX7elTV_Ber-cdiVxB6Zqg?pwd=dt48 提取码: dt48

下载完成后, 进入下载得到的isaacsim目录下, 双击 `Install_IsaacLab_Env_Assets.bat` 安装训练环境并生成 `Python311Run.bat` 脚本。

```
Command Prompt
[INFO] ENV_ARCHIVE       : "D:\isaacsim\env_isaacsim.tar.gz"
[INFO] ENV_DEST          : "C:\PX4PSP\IsaacSim\env_isaacsim"
[INFO] ENVIRON_ARCHIVE   : "D:\isaacsim\Environments.zip"
[INFO] ROBOTS_ARCHIVE    : "D:\isaacsim\Robots.zip"
[INFO] ASSETS_DEST       : "C:\PX4PSP\IsaacSim"
[INFO] PSP_PATH          : "C:\PX4PSP"
[INFO] INSTALL_ROOT      : "C:\PX4PSP\IsaacSim"
[INFO] Python311Run.bat  : "D:\isaacsim\Python311Run.bat"

[INFO] Unpacking environment...
[INFO] Env root detected: "C:\PX4PSP\IsaacSim\env_isaacsim"

[INFO] Running conda-unpack (log: "C:\PX4PSP\IsaacSim\env_isaacsim\conda-unp
[INFO] conda-unpack OK.

Access is denied.

Access is denied.

[INFO] Generating "D:\isaacsim\Python311Run.bat"

[INFO] All done.
[INFO] Next:
  1) cd /d "D:\isaacsim\"
  2) Python311Run.bat
     - or run: Python311Run.bat your_script.py

Press any key to continue . . .
```

将得到的 `Python311Run.bat` 脚本拷贝到 `IsaacLab/` 目录下

四旋翼跟踪物体强化学习训练

在 `IsaacLab/` 目录下，双击 `Python311Run.bat` 启动训练环境。

在终端中执行如下命令：

启动训练示例 (headless)，默认训练200轮：

```
1 | isaacsim.bat -p scripts/reinforcement_learning/rsl_rl/train.py --task Isaac-Quadcopter-Direct-v0 --headless
```

临时指定训练2000轮 (推荐)：在命令后加 `--max_iterations 2000`

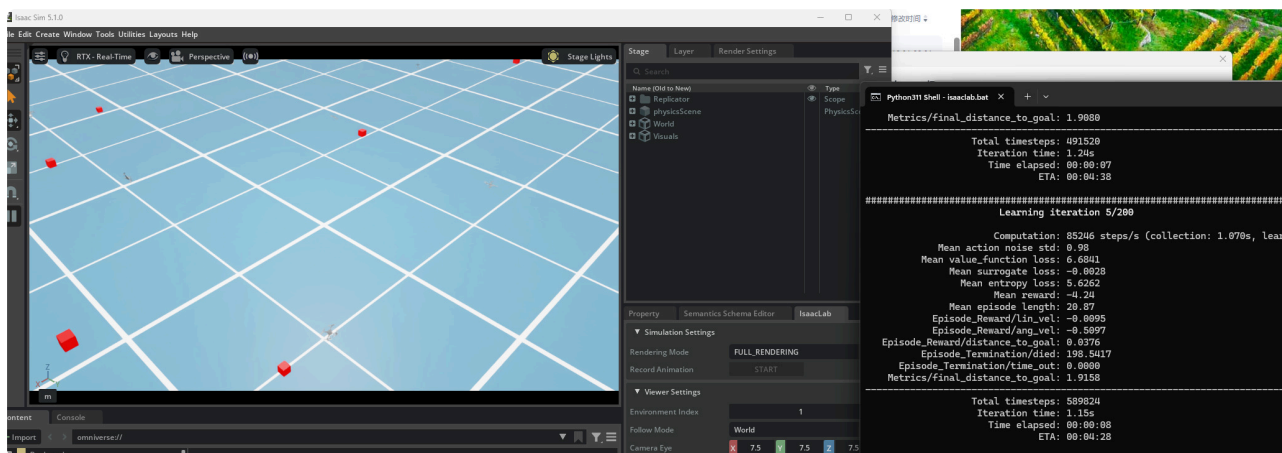
```
1 | isaacsim.bat -p scripts/reinforcement_learning/rsl_rl/train.py --task Isaac-Quadcopter-Direct-v0 --headless --max_iterations 2000
```

接着上次（已训 200 轮）继续训到 2000：加 --resume（必要时再加 --load_run <日志子目录名> / --checkpoint <模型文件名>）

```
1 | isaacsim.bat -p scripts/reinforcement_learning/rsl_rl/train.py --task Isaac-Quadcopter-Direct-v0 --headless --resume --max_iterations 2000
```

也可以去掉 --headless 选项以开启可视化（推荐使用3080Ti以上的显卡，若显存不足 (<16gb) 可以将并行环境数调小，如后缀 --num_envs 512；默认并行环境数是4096）

```
1 | isaacsim.bat -p scripts\reinforcement_learning\rsl_rl\train.py --task Isaac-Quadcopter-Direct-v0
```

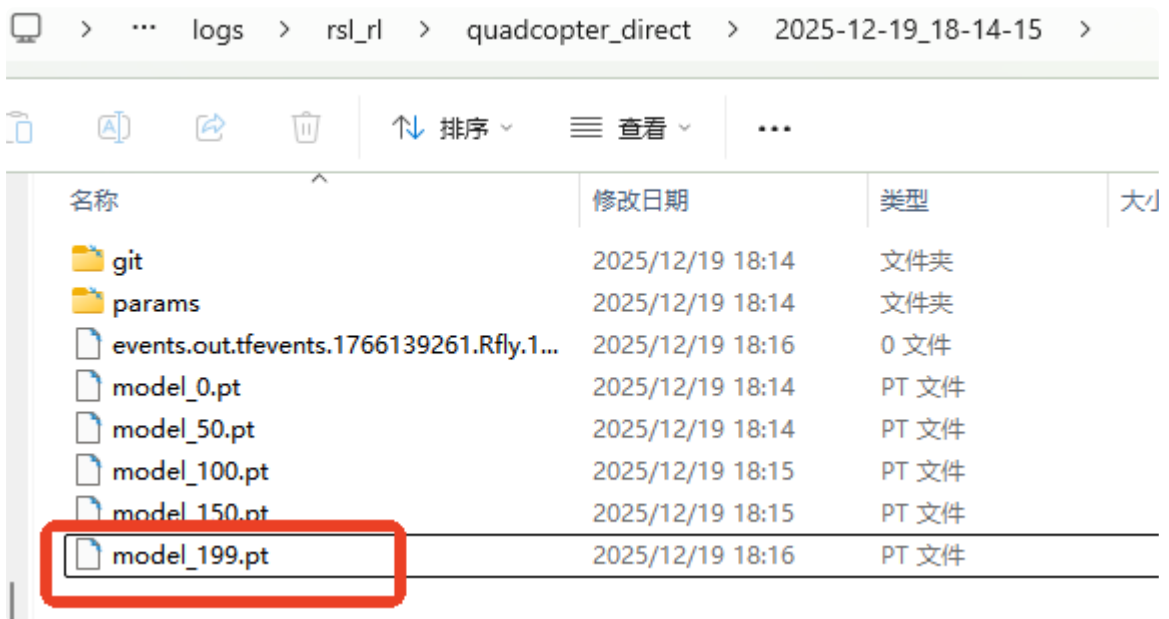


等待训练完成并拷贝训练得到的模型权重

训练完成后

```
root@Rfly: /mnt/d/issac/Isaac x + v
Episode_Termination/time_out: 27.0000
Metrics/final_distance_to_goal: 0.0305
-----
Total timesteps: 19562496
Iteration time: 0.40s
Time elapsed: 00:01:19
ETA: 00:00:00
#####
Learning iteration 199/200
Computation: 254494 steps/s (collection: 0.309s, learning 0.077s)
Mean action noise std: 0.10
Mean value_function loss: 0.0222
Mean surrogate loss: 0.0059
Mean entropy loss: -4.4969
Mean reward: 131.96
Mean episode length: 499.00
Episode_Reward/lin_vel: -0.0120
Episode_Reward/ang_vel: -0.0221
Episode_Reward/distance_to_goal: 13.2501
Episode_Termination/died: 0.0000
Episode_Termination/time_out: 20.9583
Metrics/final_distance_to_goal: 0.0322
-----
Total timesteps: 19660800
Iteration time: 0.39s
Time elapsed: 00:01:20
ETA: 00:00:00
```

在IsaacLab的如下目录找到训练得到的模型权重，比如这里是
D:\issac\IsaacLab\logs\rsl_rl\quadcopter_direct\2025-12-19_18-14-15



将这个权重 `model_199.pt` 文件(根据具体的训练轮次确定)拷贝到本实验目录RL_attctrl/
下，修改RL_attctrl/rl.py文件，加载的权重文件名改为model_199.pt。

6.RflySimExtCtrl > 2.AdvExps > e6.IsaacSimExps > RL_attctrl

名称	修改日期	类型	大小
circle_follow	2025/10/1 23:16	Windows 批处理...	
circle_follow	2024/11/7 16:12	Python 源文件	
Config	2025/5/18 0:07	JSON 源文件	
cont	2025/10/17 12:34	Python 源文件	
model_199.pt	2025/12/15 10:44	PT 文件	
model_1999.pt	2025/10/12 0:38	PT 文件	
Python38Run	2025/10/1 23:16	Windows 批处理...	
rl	2025/12/15 11:08	Python 源文件	
WinWSL	2025/10/1 23:16	Windows 批处理...	
WslGUI	2025/10/1 23:16	Windows 批处理...	

```

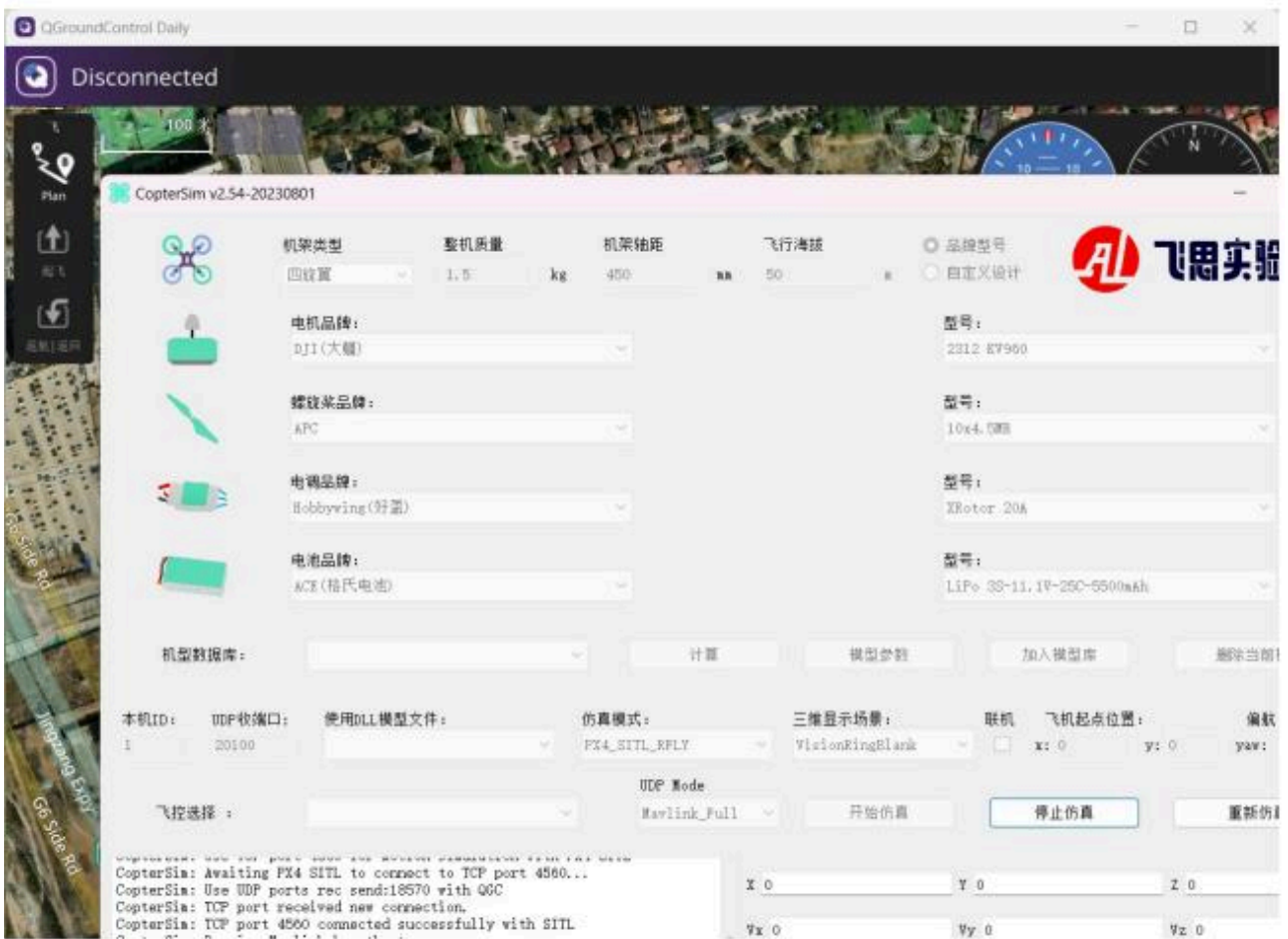
RL_attctrl > rl.py > run_rl_controller
71 def get_rl_observation(mav, target_pos):
72     # 目标相对位置在机体坐标系下的投影
73     pos_err_w = np.array(target_pos) - pos_w
74     desired_pos_b = R @ pos_err_w
75
76     obs = np.concatenate([vel_b, ang_vel_b, g_b, desired_pos_b], axis=0)
77     return torch.tensor(obs, dtype=torch.float32).unsqueeze(0)
78
79 # =====
80 # 5. 主控制逻辑
81 # =====
82 def run_rl_controller(model_path="model_199.pt"):
83     device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
84     model = load_model(model_path, device)
85
86     # 初始化 MAVLink 通信
87     mav = PX4MavCtrl.PX4MavCtrl(1)
88     mav.InitMavLoop()
89     time.sleep(0.5)
90     mav.initOffboard()
91     time.sleep(1)
92
93     # 起飞

```

4.2 步骤2：仿真验证实验结果

开启仿真

双击运行circle_FOLLOW.bat启动软件在环仿真并等待初始化完成。



运行控制程序

双击Python38Run.bat启动python环境，在终端执行 `python circle_FOLLOW.py`

```
C:\windows\system32\cmd.exe x + v
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with Python
F:\d3\6.RflySimExtCtrl\2.AdvExps\6.IsaacSimExps\RL_attctrl>python circle_follow.py
```

使用键盘上的 Up(↑)：表示案板向上移动；

Down (↓)：表示案板向下移动；

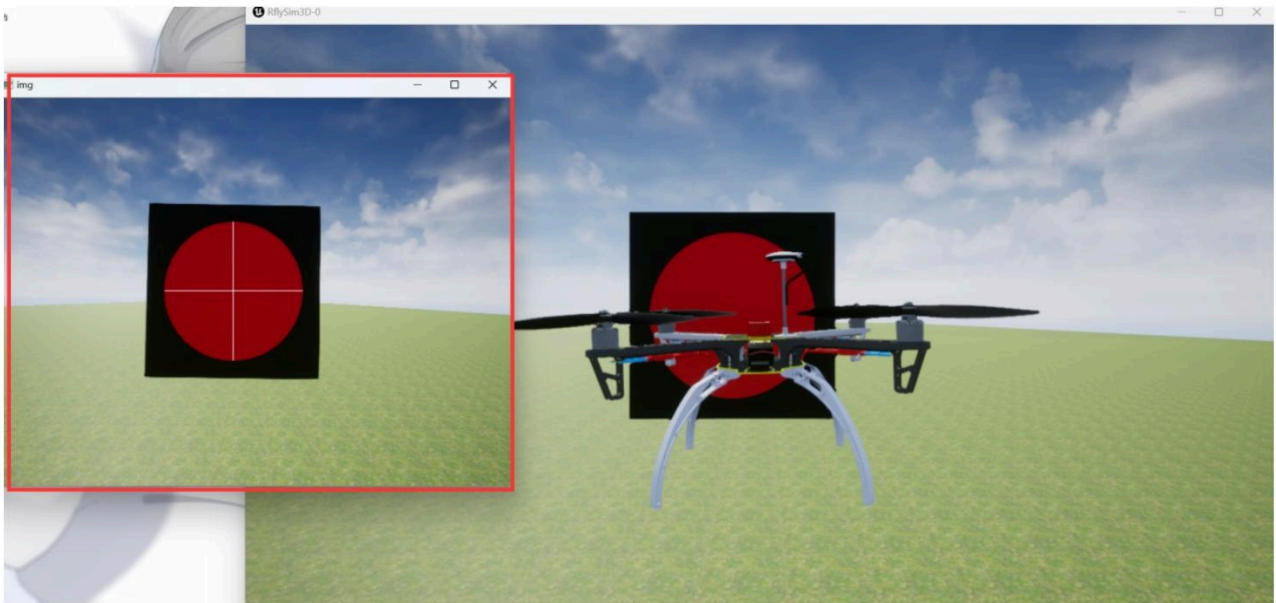
Left (←)：表示案板向左右移动；

Right (→)：表示案板向右移动；

Ctrl+Up: 表示案板向飞机前方移动，(远离飞机)；

Ctrl+Down表示案板向飞机后移动，(靠近飞机)；

无人机跟随案板移动。



结束仿真

在“circle_FOLLOW.bat”脚本开启的命令提示符CMD窗口中，按下回车键（任意键）就能快速关闭CopterSim、QGC、RflySim3D等所有程序。

5. 关键知识点

5.1 关键知识点1：视觉采集

使用 `VisionCaptureApi.VisionCaptureApi()` 创建视觉传感器实例并按 `Config.json` 配置取图，主要流程：

```
1 vis = VisionCaptureApi.VisionCaptureApi()
2 vis.jsonLoad()
3 isSuccess = vis.sendReqToUE4()
4 vis.startImgCap()
5 if vis.hasData[i]:
6     img = vis.Img[i]
7     cv2.imshow('Img'+str(i), img)
```

注意：`Config.json` 中 `SendProtocol = 0` 使用共享内存传输，仅在 Windows 下可用。

常见字段示例：

```
1 "SeqID": 0,
2 "TypeID": 1,
3 "TargetCopter": 1,
4 "SendProtocol": [0,0,0,0,0,0,0,0],
5 "SensorPosXYZ": [0.3, 0, 0]
```

5.2 关键知识点2：飞控与 UE4 接口

- 使用 `PX4MavCtrl` 向飞控发送命令（示例）：

```
1 mav = PX4MavCtrl.PX4MavCtrl(1)
2 mav.InitMavLoop()
3 mav.initOffboard()
4 mav.SendMavArm(True)
5 nav.SendPosNED(0,0,-10,0)
```

- 使用 `UE4CtrlAPI` 控制 UE4 显示与对象创建：

```
1 ue = UE4CtrlAPI.UE4CtrlAPI()
2 ue.sendUE4Cmd('r.setres 1280x720w', 0)
3 ue.sendUE4Cmd('t.MaxFPS 30', 0)
4 ue.sendUE4PosNew(3, vehicleType=809, PosE=[2,0,-2])
```

5.3 关键知识点3：控制逻辑（关键函数）

- `MoveCircle(val)`：响应键盘，移动案板目标。
- `CalCircle(img)`：检测图像中的红色区域，返回边界与中心坐标。
- `CtrlDrone()`：根据视觉反馈计算并下发飞控指令，通常在独立线程中运行。
- 键盘热键示例：

```
1 | keyboard.add_hotkey('ctrl+up', MoveCircle, args=('front',))
```

6. 参考资料

1. RflySim 官方文档：<https://rflysim.com/doc/zh/>
2. IsaacLab 官方文档：
https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/index.html

7. 常见问题

Q1：如何在 Windows 下使用图像共享？

A1：请确保 `Config.json` 中 `SendProtocol` 设置为 `0`（共享内存），并在 Windows 环境运行 `circle_FOLLOW.py`。

Q2：如何加载自己训练的权重？

A2：将训练得到的模型权重拷贝到 `circle_FOLLOW.py` 指定的模型路径，或在脚本中修改模型加载路径。