

MAVROS1控制无人机起飞实验

1. 实验目的

本实验旨在通过MAVROS接口实现对PX4无人机的离板(offboard)控制。通过该实验，用户将学习如何使用ROS节点与PX4飞控系统进行通信，掌握无人机的离板控制模式设置、无人机自主起飞的基本流程，并理解MAVROS在无人机自动化控制中的作用。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链 [\[1\]](#)。
- 硬件要求：笔记本/台式电脑1台 [\[2\]](#)。

3. 实验地址

例程目录：

[\[安装目录\]\RflySimAPIs\2.RflySimUsage\0.ApiExps\e10_Mavros\2.Mavros1Offboard](#)

文件目录结构说明(使用代码块格式)：

```
1 | └─ RosSwitch.bat           # ROS版本切换脚本
2 | └─ SITLRunROS.bat        # SITL仿真环境启动脚本
3 | └─ WinWSL.bat           # 进入Ubuntu环境脚本
4 | └─ offboard.py          # 无人机离板控制Python程序
```

4. 实验内容或步骤

4.1 ROS 版本切换

双击 [RosSwitch.bat](#)，确认当前处于ROS1模式。

```
Current version is ROS2
New ROS Version 1 or 2:1
Switch to ROS1
Switch to ROS1
Restart terminal to take effect.
Successful.
Press any key to exit.
请按任意键继续...
```

4.2 仿真环境初始化

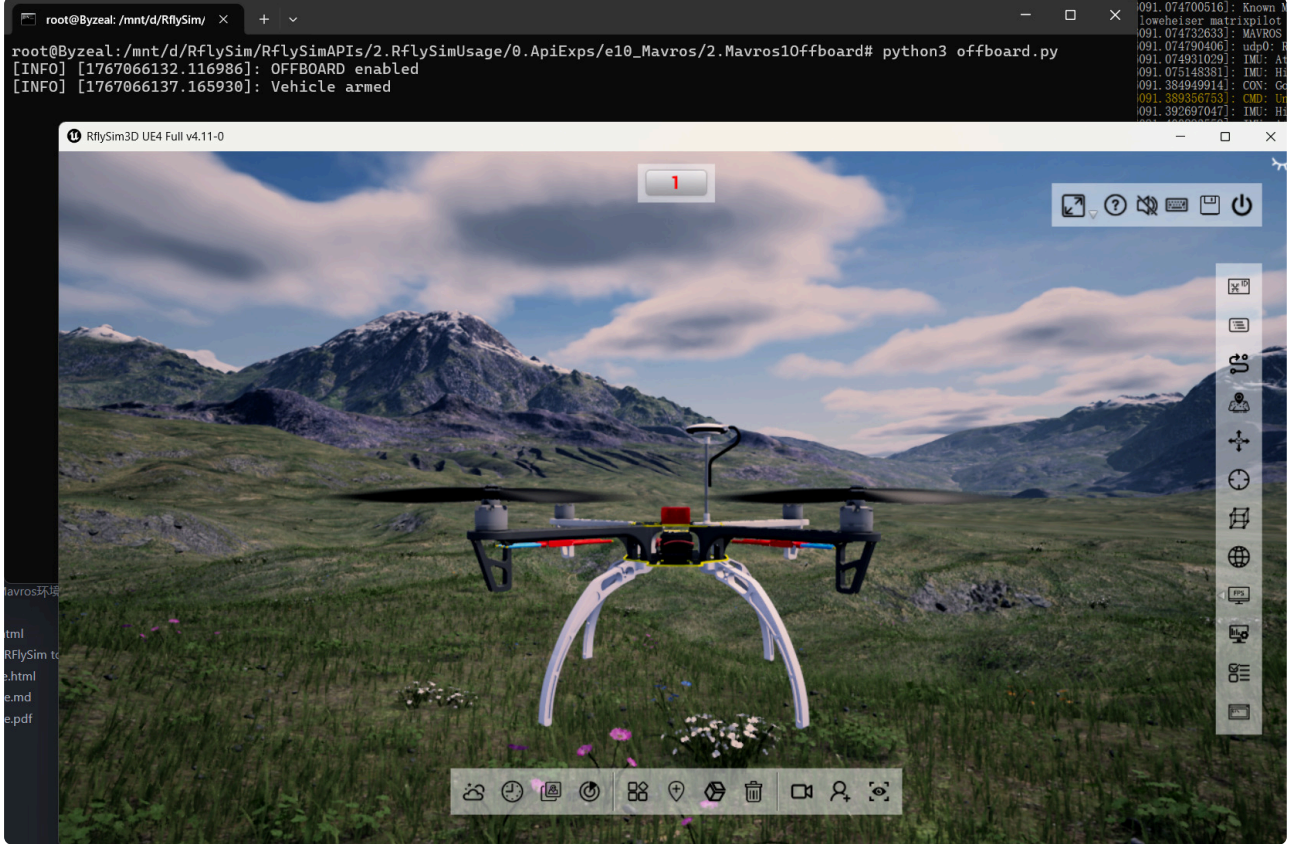
双击运行 `SITLRunROS.bat` 文件，等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件并启动 mavros，等待 CopterSim 软件下侧日志栏必须打印出 `GPS 3D fixed & EKF initialization finished` 字样代表初始化完成。如下图所示：



```
[INFO] [1767066091.070974202]: Plugin vision_speed_estimate initialized
[INFO] [1767066091.071077540]: Plugin waypoint loaded
[INFO] [1767066091.074038211]: Plugin waypoint initialized
[INFO] [1767066091.074095881]: Plugin wheel_odometry blacklisted
[INFO] [1767066091.074160213]: Plugin wind_estimation loaded
[INFO] [1767066091.074494015]: Plugin wind_estimation initialized
[INFO] [1767066091.074637375]: Built-in SIMD instructions: SSE, SSE2
[INFO] [1767066091.074673996]: Built-in MAVLink package version: 2025.5.5
[INFO] [1767066091.074700516]: Known MAVLink dialects: common ardupilotmega ASLUAV AVSSUAS all csAirLink cubepilot devel
icarus loweheiser matrixpilot paparazzi standard storm32 uAvionix ualberta
[INFO] [1767066091.074732633]: MAVROS started. MY ID 1.240, TARGET ID 1.1
[INFO] [1767066091.074790406]: udp0: Remote address: 127.0.0.1:20100
[INFO] [1767066091.074931029]: IMU: Attitude quaternion IMU detected!
[INFO] [1767066091.075148381]: IMU: High resolution IMU detected!
[INFO] [1767066091.384949914]: CON: Got HEARTBEAT, connected. FCU: PX4 Autopilot
[WARN] [1767066091.389356753]: CMD: Unexpected command 512, result 2
[INFO] [1767066091.392697047]: IMU: High resolution IMU detected!
[INFO] [1767066091.400893559]: IMU: Attitude quaternion IMU detected!
Press any key to exit
[INFO] [1767066092.398922283]: GF: Using MISSION_ITEM_INT
[INFO] [1767066092.398979313]: RP: Using MISSION_ITEM_INT
[INFO] [1767066092.399010856]: WP: Using MISSION_ITEM_INT
[INFO] [1767066092.399028290]: VER: 1.1: Capabilities 0x000000000000e8ff
[INFO] [1767066092.399053158]: VER: 1.1: Flight software: 010f0400 (99c40407ff000000)
[INFO] [1767066092.399090566]: VER: 1.1: Middleware software: 010f0400 (99c40407ff000000)
[INFO] [1767066092.399112816]: VER: 1.1: OS software: 060657ff (5d74bc138955e6f0)
[INFO] [1767066092.399148574]: VER: 1.1: Board hardware: 00000001
[INFO] [1767066092.399184839]: VER: 1.1: VID/PID: 0000:0000
[INFO] [1767066092.399215681]: VER: 1.1: UID: 4954414c44494e4f
```

4.3 运行控制程序

双击 WinWSL.bat 进入 Ubuntu 环境，输入 python3 offboard.py，可以看到飞机进入 offboard 并起飞。



5. 关键知识点

本实验的整体思路是通过MAVROS包实现ROS与PX4飞控系统之间的通信，使用ROS节点控制无人机进入离板模式并执行预设位置的飞行任务。整个系统框架包括PX4飞控、MAVROS接口、ROS节点和控制算法。

主要涉及以下几个关键部分：

1. 状态订阅：订阅 `mavros/state` 话题获取无人机当前飞行状态
2. 位置发布：向 `mavros/setpoint_position/local` 话题发布目标位置信息
3. 模式切换：通过服务调用将无人机切换至OFFBOARD模式
4. 自主导航：控制无人机自动起飞到指定高度

对 `offboard.py` 程序的关键代码解析：

1. 初始化ROS节点与订阅状态信息：

```
1 | rospy.init_node("offb_node_py")
2 | state_sub = rospy.Subscriber("mavros/state", State, callback = state_cb)
3 | local_pos_pub = rospy.Publisher("mavros/setpoint_position/local", PoseStamped,
  | queue_size=10)
```

2. 获取服务代理，用于模式切换和无人机解锁：

```
1 | rospy.wait_for_service("/mavros/cmd/arming")
2 | arming_client = rospy.ServiceProxy("mavros/cmd/arming", CommandBool)
3 |
4 | rospy.wait_for_service("/mavros/set_mode")
5 | set_mode_client = rospy.ServiceProxy("mavros/set_mode", SetMode)
```

3. 设置目标位置，本例中为相对当前位置的坐标：

```
1 | pose = PoseStamped()
2 | pose.pose.position.x = 0
3 | pose.pose.position.y = 0.2
4 | pose.pose.position.z = 0.8
```

4. 在主循环中，首先切换到OFFBOARD模式，然后解锁无人机，最后持续发布目标位置以维持无人机在目标位置：

```

1 | if(current_state.mode != "OFFBOARD" and (rospy.Time.now() - last_req) >
2 |   rospy.Duration(5.0)):
3 |     if(set_mode_client.call(offb_set_mode).mode_sent == True):
4 |         rospy.loginfo("OFFBOARD enabled")
5 |         last_req = rospy.Time.now()
6 |     else:
7 |         if(not current_state.armed and (rospy.Time.now() - last_req) >
8 |           rospy.Duration(5.0)):
9 |             if(arming_client.call(arm_cmd).success == True):
10 |                 rospy.loginfo("Vehicle armed")
11 |                 last_req = rospy.Time.now()

local_pos_pub.publish(pose)

```

通过该程序，实现了无人机从手动模式到离板模式的自动切换，以及无人机的自动解锁和飞向目标位置的功能。

6. 参考资料

1. RflySim官方文档
2. MAVROS官方文档: <http://wiki.ros.org/mavros>
3. PX4开发指南: <https://docs.px4.io/>, https://docs.px4.io/main/zh/ros/mavros_offboard_cpp.html.
4. ROS机器人操作系统: <http://wiki.ros.org/>

7. 常见问题

Q1: 运行 `offboard.py` 时，无人机无法进入 OFFBOARD 模式?

A1: 检查仿真环境是否已完全初始化，等待CopterSim软件下侧日志栏打印出 `GPS 3D fixed & EKF initialization finished` 字样后再运行程序。另外，确保在QGC地面站中已设置允许MAVLink消息控制。

Q2: 无人机起飞后不稳定或无法到达目标位置?

A2: 检查 `offboard.py` 中的目标位置设置是否合理，以及发布频率是否满足要求（需要大于2Hz）。代码中使用了30Hz的发布频率，确保 `rate = rospy.Rate(30)` 设置正确。

Q3: 运行Python脚本时出现"ModuleNotFoundError: No module named rospy"错误?

A3: 确保已正确安装ROS环境并配置了ROS环境变量。在运行脚本前需要先启动ROS核心服务，并确保Python脚本在正确的ROS环境中执行。

Q4: 如何修改无人机的目标飞行高度?

A4: 在 `offboard.py` 文件中修改 `pose.pose.position.z` 的值，该值表示无人机相对于起飞点的高度。例如，设置为1.0表示无人机飞到距离起飞点1米的高度。

-
1. <https://rflysim.com/> ↩
 2. 推荐配置请见: <https://rflysim.com/doc/zh/HowToInstall.pdf> ↩