

MAVROS2控制无人机起飞实验

1. 实验目的

本实验旨在通过MAVROS2框架实现无人机的离板(offboard)控制，验证ROS2与PX4飞控系统之间的通信机制。通过本实验，学习者将掌握MAVROS2的基本使用方法，理解无人机离板控制的原理和实现方式，以及如何通过ROS2节点对无人机进行姿态控制和飞行操作。

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\3.Mavros2Offboard](#)

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        # 仿真环境启动脚本
3 | └─ WinWSL.bat           # WSL环境启动脚本
4 | └─ offboard.py          # ROS2离板控制程序
```

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。

```
C:\WINDOWS\system32\cmd.exe
Current version is ROS1
New ROS Version 1 or 2:2
Switch to ROS2
Switch to ROS2
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
[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

本实验的核心思路是通过MAVROS2包实现ROS2与PX4飞控系统之间的通信，从而实现对无人机的离板控制。整体框架包括状态订阅、模式切换、无人机解锁和位置控制等模块。

`offboard.py` 程序解析：

```
1  #!/usr/bin/env python
2  import rclpy
3  from rclpy.node import Node
4  from rclpy.qos import QoSProfile, ReliabilityPolicy, HistoryPolicy,
5  DurabilityPolicy
6  from geometry_msgs.msg import PoseStamped
7  from mavros_msgs.msg import State
8  from mavros_msgs.srv import CommandBool, SetMode
9
10 current_state = State()
11
12 def state_cb(msg):
13     global current_state
14     current_state = msg
```

首先导入必要的ROS2和MAVROS2库，定义全局变量 `current_state` 存储无人机当前状态，并创建回调函数 `state_cb` 实时更新无人机状态。

```
1  if __name__ == "__main__":
2      rclpy.init()
3      ros_node = Node("RflyRos")
4
5      t1 = threading.Thread(target=rclpy.spin, args=(ros_node,))
6      t1.start()
```

初始化ROS2节点，并在独立线程中运行事件循环，确保节点能够持续接收和处理消息。

```

1 | qos_profile = QoSProfile(
2 |     reliability=ReliabilityPolicy.BEST_EFFORT,
3 |     durability=DurabilityPolicy.TRANSIENT_LOCAL,
4 |     history=HistoryPolicy.KEEP_LAST,
5 |     depth=1
6 | )
7 |
8 | state_sub=ros_node.create_subscription(State, "/mavros/state", state_cb,
9 | qos_profile)
10 | local_pos_pub = ros_node.create_publisher(PoseStamped,
11 | "mavros/setpoint_position/local", 10)
12 |
    arming_client = ros_node.create_client(CommandBool, "mavros/cmd/arming")
    set_mode_client = ros_node.create_client(SetMode, "mavros/set_mode")

```

创建QoS配置，设置订阅者监听无人机状态，创建发布者发布位置设定值，创建客户端用于解锁无人机和切换飞行模式。

```

1 | # Wait for Flight Controller connection
2 | while(rclpy.ok() and not current_state.connected):
3 |     rate.sleep()
4 |
5 | pose = PoseStamped()
6 | pose.pose.position.x = float(0)
7 | pose.pose.position.y = float(0.2)
8 | pose.pose.position.z = float(0.8)

```

等待飞控连接，然后设置目标位置为x=0, y=0.2, z=0.8米。

```

1 | # Send a few setpoints before starting
2 | for i in range(100):
3 |     if(not rclpy.ok()):
4 |         break
5 |     local_pos_pub.publish(pose)
6 |     rate.sleep()

```

在切换到offboard模式前，先发送100个设定值，确保飞控系统已接收到目标位置。

```

1 |     offb_set_mode = SetMode.Request()
2 |     offb_set_mode.custom_mode = 'OFFBOARD'
3 |
4 |     arm_cmd = CommandBool.Request()
5 |     arm_cmd.value = True
6 |
7 |     while(rclpy.ok()):
8 |         if(current_state.mode != "OFFBOARD" and (ros_node.get_clock().now() -
9 | last_req) > Duration(seconds=5)):
10 |             if(set_mode_client.call(offb_set_mode).mode_sent == True):
11 |                 ros_node.get_logger().info("OFFBOARD enabled")
12 |                 last_req = ros_node.get_clock().now()
13 |             else:
14 |                 if(not current_state.armed and (ros_node.get_clock().now() - last_req)
15 | > Duration(seconds=5)):
16 |                     if(arming_client.call(arm_cmd).success == True):
17 |                         ros_node.get_logger().info("Vehicle armed")
18 |                         last_req = ros_node.get_clock().now()
19 |                 local_pos_pub.publish(pose)
20 |                 rate.sleep()

```

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，使无人机飞向目标点并保持悬停。

6. 参考资料

1. RflySim官方文档: <https://rflysim.com/doc/zh/>
2. MAVROS官方文档: <https://mavros.readthedocs.io/>
3. PX4开发指南: <https://docs.px4.io/>

7. 常见问题

Q1: 运行python3 `offboard.py`时提示找不到rclpy模块。

A1: 这是由于缺少ROS2环境配置导致的。需要确保已正确安装ROS2并配置环境变量，在使用前需先运行 `WinWSL.bat` 进入预配置的Ubuntu环境。

Q2: 无人机无法进入OFFBOARD模式或无法解锁。

A2: 首先确保仿真环境已完全初始化（显示GPS 3D fixed & EKF initialization finished），然后需要在发送OFFBOARD指令前持续发送设定点以满足安全要求。

Q3: 无人机起飞后不稳定或无法到达目标位置。

A3: 检查 `offboard.py` 中的目标位置参数设置是否合理，同时确保PX4飞控的参数配置正确，特别是与位置控制相关的参数。

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