

Simulink/DLL 综合模型 ROS 转发 (Python 版)

1. 实验目的

- 使用 Python ROS 转发程序替代 mavros，实现 RflySim 综合模型与 ROS 话题的双向桥接。
- 支持 ROS1/ROS2 下多机控制，便于将 PX4 软硬件在环算法迁移到综合模型。
- 通过示例脚本验证多机起飞与状态获取。

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\6.RflySimExtCtrl\0.ApiExps\21.SimulinkAndDllRosTrans\2.PythonDemo](#)

[3.CopterSimSILNoPX4](#) 说明：综合模型仿真启动脚本

[SimulinkAndDllMavrosStart.py](#) 说明：多机桥接启动脚本

[start_all.sh](#) : ROS1/ROS2 一键启动脚本

[Ros12MultiUav.py](#) 说明：ROS 控制示例

[WinWSL.bat](#) : WSL 环境启动脚本

4. 实验内容或步骤

4.1 步骤1 启动综合模型仿真

4.* 双击运行 [3.CopterSimSILNoPX4/MulticopterNOpx4.bat](#)，启动 3 架综合模型仿真。
观察仿真是否成功启动。

4.2 步骤2 启动 Python 转发节点

4.* 运行 [WinWSL.bat](#) 进入 WSL，执行：

```
./start_all.sh
```

脚本会提示输入 Copter ID (如 1,2,3) 和 target_ip (默认 127.0.0.1)。完成后自动启动 ROS1/ROS2 与桥接程序。

```
root@yyx: /mnt/e/4WorkReco x + v
root@yyx: /mnt/e/4WorkRecord/1-1RflySim研发组工作/1-研发需求/25emo# ./start_all.sh
请输入 Copter ID (多个用逗号分隔, 默认 1, 按回车使用默认) :
1,2,3
请输入 target_ip (默认 127.0.0.1, 按回车使用默认) :
```

输入完成后可看到实例启动:

```
root@yyx: /mnt/e/4WorkReco x + v
清理旧进程...
进入 ROS1 模式...
启动 roscore (后台运行) ...
启动 Python 程序...
ROS1: True, ROS2: False
启动统一测试例程...
启动飞机实例: [1, 2, 3]
目标 IP: 127.0.0.1
检测到 ROS1 (Noetic) 环境
ROS1 节点初始化完成
[Copter1] 初始化 _init_socket...
[Copter1] 初始化 _init_ros...
[Copter1] 启动 UdpRosBridge...
[Copter1] 四元数 norm 过小 (0.000000), 使用默认单位四元数
[Copter1] ROS 事件处理线程已启动
Bridge for Copter 1 已启动
[Copter2] 初始化 _init_socket...
[Copter2] 初始化 _init_ros...
[Copter2] 启动 UdpRosBridge...
[Copter2] 四元数 norm 过小 (0.000000), 使用默认单位四元数
[Copter2] ROS 事件处理线程已启动
Bridge for Copter 2 已启动
[Copter3] 初始化 _init_socket...
[Copter3] 初始化 _init_ros...
[Copter3] 启动 UdpRosBridge...
[Copter3] 四元数 norm 过小 (0.000000), 使用默认单位四元数
[Copter3] ROS 事件处理线程已启动
Bridge for Copter 3 已启动
所有 Bridge 启动完成, 按 Ctrl+C 停止
```

可选: 直接运行脚本并指定参数:

```
python3 SimuLinkAndDllMavrosStart.py --copter_ids "1,2,3" --target_ip 127.0.0.1
```

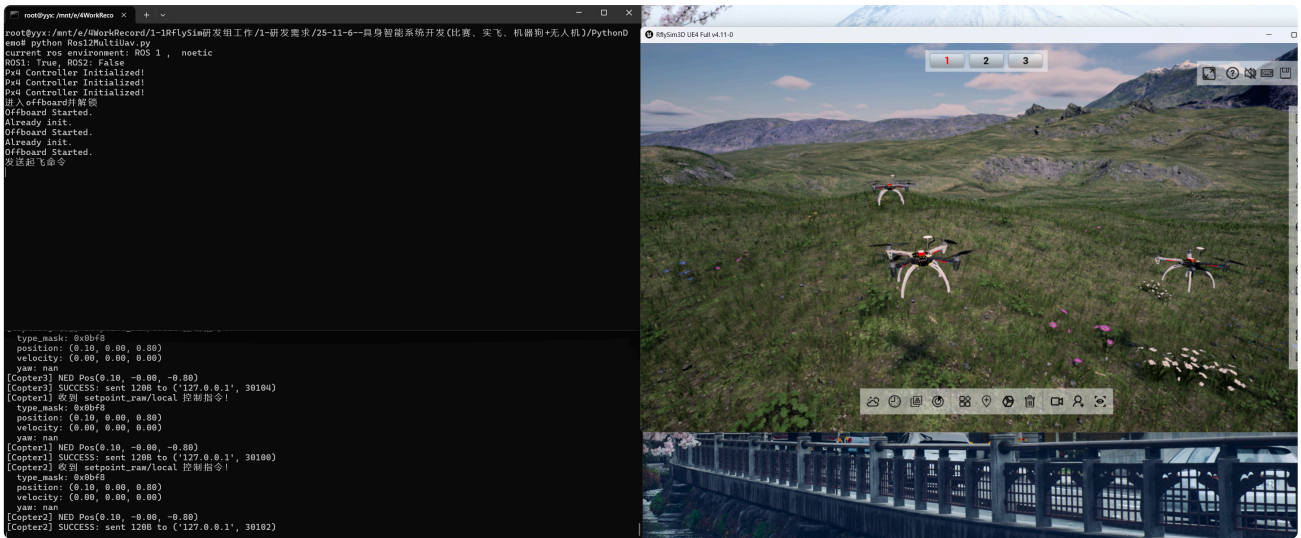
4.3 步骤3 运行控制脚本验证

4.* 再次运行 WinWSL.bat, 执行 python Ros12MultiUav.py 默认启动Copterid=1的节点

如果使用非默认的 Copter ID, 请指定参数:

```
python Ros12MultiUav.py --copter_ids 1,2,3
```

观察飞机起飞及状态打印, 说明 ROS 转发链路生效。



验证控制脚本是否成功运行。

5. 关键知识点

关键知识点1：转发程序原理

本例程中的 Python 转发程序（基于 `DllSimCtrlAPIROS.py`）旨在为 **RflySim 综合模型（非 PX4 模式）** 提供与 `mavros` 兼容的通信接口。

- **状态反馈**：监听 CopterSim 从 `30101` 等端口发出的 UDP 原始状态数据，解析后发布至 `/mavros/local_position/pose` 等标准 ROS 话题。
- **控制指令**：订阅 `/mavros/setpoint_raw/local` 等控制话题，将其封装为 CopterSim 可识别的控制指令并转发，从而实现算法在 PX4 模式与综合模型模式间的无缝迁移。

关键知识点2：核心文件说明

- `DllSimCtrlAPIROS.py`：核心类库，封装了 UDP 通信逻辑与 ROS 消息转换逻辑。
- `SimulinkAndDllMavrosStart.py`：多机启动管理脚本，负责为每架飞机实例化转发节点，并处理命名空间映射（如 `uav1/mavros/...`）。
- `Ros12MultiUav.py`：通用的多机控制示例，展示了如何通过简单的 Python 代码同时兼容 ROS1 与 ROS2 下的无人机控制。
- `start_all.sh`：自动化环境配置脚本，能够自动检测当前的 `ROS_VERSION` 并启动相应版本的桥接程序。

关键知识点3：话题映射关系

转发程序模拟了部分 `mavros` 核心功能，支持的关键话题包括：

- 类别：**发布 (Pub)** 话题名称：`mavros/state` 功能说明：无人机连接状态、解锁状态及模式
- 类别：**发布 (Pub)** 话题名称：`mavros/local_position/pose` 功能说明：无人机本地笛卡尔坐标位姿
- 类别：**发布 (Pub)** 话题名称：`mavros/local_position/velocity_local` 功能说明：无人机本地速度
- 类别：**订阅 (Sub)** 话题名称：`mavros/setpoint_raw/local` 功能说明：位置/速度/加速度期望值控制
- 类别：**订阅 (Sub)** 话题名称：`mavros/cmd/arming` 功能说明：解锁与上锁指令
- 类别：**订阅 (Sub)** 话题名称：`mavros/set_mode` 功能说明：切换飞行模式（如 OFFBOARD）

6. 参考资料

1. [RflySim官方文档](#)
2. PX4 对比实

验：

```
[RflySim安装路径]\RflySimAPIs\6.RflySimExtCtrl\0.ApiExps\e18_MavrosExps\2.RflyRosAPIMultiUav
```

7. 常见问题

Q1: `start_all.sh` 无法检测 ROS 版本?

A1: 确认已安装 ROS；或在终端里设置 `export ROS_VERSION=1` (ROS1) 或 `export ROS_VERSION=2` (ROS2) 后再运行。

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