

# PX4 Minimum Snap 轨迹跟踪 (LADRC 版)

本项目是一个 ROS 2 软件包，用于在 PX4-Gazebo 仿真环境中实现高精度的轨迹跟踪。

它结合了 `minisnap` 轨迹规划器 (`trajectory_planner_py`) 和**线性自抗扰控制器** (`ladrc_controller`)，以取代原版的 PID/几何控制器。

## 项目核心

本项目由两个核心 ROS 2 包组成：

### 1. `trajectory_planner_py` (Python 规划器)

- 使用 Minimum Snap 算法生成平滑的、动态的 "8" 字形轨迹。
- 以 100Hz 的频率发布完整的轨迹状态，包括**位置** (`/target_pose`)、**速度** (`/target_twist`) 和**加速度** (`/target_accel`)。
- 轨迹被设置为**循环播放**，以允许控制器有足够的时间进行初始化。

### 2. `ladrc_controller` (C++ 控制器)

- 实现了一个基于 (LESO + LSEF) 的**线性自抗扰控制器**。
- 此节点负责**自动起飞**：在启动 10 秒后自动执行解锁 (Arming) 和切换 Offboard 模式。
- 订阅规划器的全部状态 (位置、速度、加速度)，实现**PD + 前馈 + 扰动补偿**控制律，以高精度跟踪动态轨迹。
- 处理 ENU (控制器) 和 NED (规划器Z轴) 之间的坐标系反转。

## 环境依赖

- ROS 2 (Humble 或更高版本)
- PX4 自动驾驶仪 (用于 SITL 仿真，工作空间位于 `~/px4_ros2_ws` 目录)
- Gazebo 仿真环境
- `px4_msgs` (ROS 2 消息包)

## 编译

1. 将 `ladrc_controller` 和 `trajectory_planner_py` 文件夹放置在您的 ROS 2 工作空间 (例如 `~/px4_ros2_ws/src`)。
2. 编译工作空间：

```
cd ~/px4_ros2_ws
rm -rf build install log
colcon build
```

## 如何运行

要运行完整的仿真，请打开**三个**独立的终端。

## 终端 1 — 启动 PX4 SITL + Gazebo

```
cd ~/px4_ros2_ws/src/PX4-Autopilot/  
make px4_sitl gz_x500
```

## 终端 2 — 启动 PX4-ROS2 通信桥

```
MicroXRCEAgent udp4 -p 8888
```

## 终端 3 — Sourcing 工作空间并启动规划器与控制器

```
cd ~/px4_ros2_ws/  
source install/setup.bash  
ros2 launch trajectory_planner_py planner.launch.py
```

## 预期行为

启动 `planner.launch.py` 后：

1. 规划器 (`planner_node`) 和控制器 (`ladrc_position_controller_node`) 将同时启动。
2. 规划器立即开始循环播放 "8" 字轨迹。
3. 控制器将**等待 10 秒**以稳定系统。
4. 等待 10 秒后，控制器将自动发送**解锁和切换 Offboard 模式**指令。
5. 进入 Offboard 模式后，LADRC 控制器将立即开始跟踪规划器发布的动态轨迹。

## 参考文献

[1] T. Lee, M. Leok, N.H. McClamroch, "Geometric tracking control of quadrotor UAV on SE(3)," 49th IEEE Conference on Decision and Control, 2010

[2] D. Mellinger, V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors". In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9–13, 2011