

Lab1 四旋翼无人机集群系统时变编队控制

北京航空航天大学人工智能学院

1 实验目标

设计并实现四旋翼无人机集群的编队控制算法，在仿真平台上进行验证。具体要求如下：

- 掌握四旋翼无人机集群系统的建模，理解无人机内外环控制框架与无人机模型的简化方法。
- 掌握集群系统的时变编队控制问题的数学描述方法。
- 示例提供了一种分布式时变编队控制算法的设计与代码实现，在仿真平台中可以实现 3 架无人机的固定编队，展示了无人机位置与速度话题的订阅以及加速度控制指令的发布方法，同时展示了控制误差的计算、发布与分析。同学们的主要任务是设计编队控制算法并代码实现，在仿真平台上完成要求的时变编队任务。控制方法不限，可以使用示例中的控制算法，或进一步优化，提升鲁棒性与适应性。不限编程语言，推荐 Python 或 C++。
- 掌握控制误差分析方法，可以借助 Plotjuggler 进行实时数据可视化；也可自行记录和分析 rosbag 数据，并绘制误差曲线以评估算法性能。

2 实验内容

建立四旋翼无人机的模型，给出无人机集群系统的时变编队控制问题的数学描述，设计编队控制算法并代码实现，在仿真平台上验证算法的有效性，分析控制误差。

2.1 四旋翼无人机集群系统建模

考虑由 N 个四旋翼无人机组成的集群系统，每个四旋翼无人机建模可以参考 [1]。定义惯性系与机体系两个坐标系如图 1 所示。 $E = \{e_1, e_2, e_3\}$ 表示惯性坐标系，有 $e_1 = [1, 0, 0]^T$ ， $e_2 = [0, 1, 0]^T$ ， $e_3 = [0, 0, 1]^T$ 。 $B_j = \{b_{j1}, b_{j2}, b_{j3}\}$ 表示第 j 个无人机的机体系，有 $b_{j1} = [1, 0, 0]^T$ ， $b_{j2} = [0, 1, 0]^T$ 与 $b_{j3} = [0, 0, 1]^T$ 。

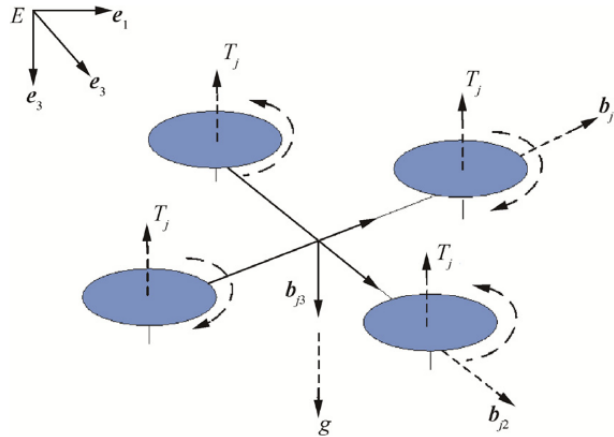


图 1: 坐标系定义

将无人机视为刚体，使用牛顿-欧拉方程描述刚体的平动与转动，第 j 个无人机的动态可以被描述为：

$$\begin{cases} \Sigma_j^1 : \begin{cases} \dot{\mathbf{p}}_j = \mathbf{v}_j \\ m_j \dot{\mathbf{v}}_j = m_j g \mathbf{e}_3 - T_j \mathbf{R}(\mathbf{Q}_j)^T \mathbf{e}_3 + \mathbf{d}_{Fj} \end{cases} \\ \Sigma_j^2 : \begin{cases} \dot{\mathbf{q}}_j = \frac{1}{2} (\eta_j \mathbf{I}_3 + \mathbf{S}(\mathbf{q}_j)) \boldsymbol{\omega}_j \\ \dot{\eta}_j = -\frac{1}{2} \mathbf{q}_j^T \boldsymbol{\omega}_j \\ \mathbf{I}_{fj} \dot{\boldsymbol{\omega}}_j = \boldsymbol{\Gamma}_j - \mathbf{S}(\boldsymbol{\omega}_j) \mathbf{I}_{fj} \boldsymbol{\omega}_j + \mathbf{d}_{\Gamma j} \end{cases} \end{cases}$$

其中， $\mathbf{p}_j \in \mathbf{R}^3$ 与 $\mathbf{v}_j \in \mathbf{R}^3$ 分别是无人机在惯性系下的位置与速度， $\boldsymbol{\omega}_j \in \mathbf{R}^3$ 是角速度， $\mathbf{Q}_j = [\mathbf{q}_j^T, \eta_j]^T$ 是由单位四元数描述的无人机姿态，满足 $\mathbf{q}_j = [q_{j1}, q_{j2}, q_{j3}]^T \in \mathbf{R}^3$ 与标量 η_j ， m_j 是无人机的质量， g 是重力加速度， \mathbf{I}_{fj} 是惯性矩阵。 T_j 表示四个桨叶产生的沿机体 Z 轴 b_{j3} 的总推力的幅值， $\boldsymbol{\Gamma}_j$ 是控制力矩， \mathbf{d}_{Fj} 与 $\mathbf{d}_{\Gamma j} \in \mathbf{R}^3$ 是外部扰动。 $\mathbf{R}(\mathbf{Q}_j) = (\eta_j^2 - \|\mathbf{q}_j\|^2) \mathbf{I}_3 + 2\mathbf{q}_j \mathbf{q}_j^T - 2\eta_j \mathbf{S}(\mathbf{q}_j)$ 是惯性系到机体系的旋转矩阵，反对称矩阵 $\mathbf{S}(\bullet)$ 定义如下

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

上述微分方程完整描述了无人机的动态，其中四元数表示姿态可避免万向锁问题，提升姿态仿真与控制的数值计算稳定性。

实践中，通常采用内外环控制架构：内环控制器用于姿态稳定，外环控制器用于位置与速度控制。一个典型的内外环控制框架可参考 PX4 开源飞控 [2]，其模块化设计集成飞行控制、传感器融合与路径规划，广泛应用于学术研究与工业无人机系统。PX4 飞控的控制框架如图 2 所示，外环控制器包括了位置控制器与速度控制器，内环控制器包括角度控制器（基于四元数）与角速度控制器，其中位置与角度控制器使用比例控制方法，速度与角速度控制器使用抗饱和的 PID 方法。

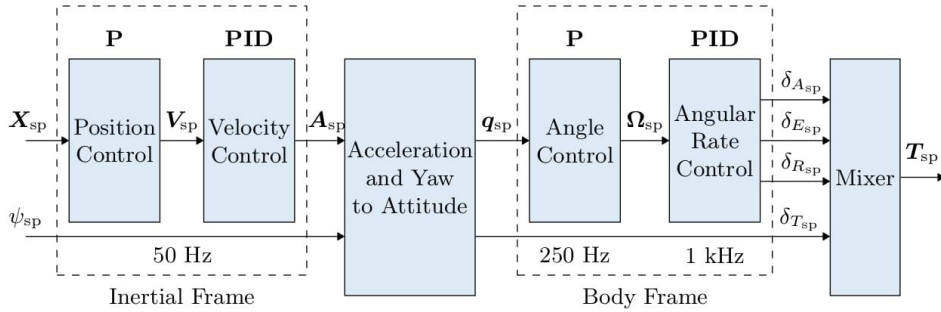


图 2: 使用 PX4 飞控的多旋翼的内外环控制架构

注意，仿真平台支持使用 PX4 飞控的四旋翼无人机的仿真。PX4 飞控中内环控制器已经完成了姿态控制，可以快速响应输入的期望加速度。但是，外环控制器不能满足编队控制的需要，示例中给出了不使用外环控制器将自己算法计算的加速度指令通过 ROS1 话题通信发送给仿真平台的方法。同学们的主要任务是设计控制算法，根据无人机（与其他无人机）当前位置与速度，计算期望的加速度指令，最终驱动整个集群系统实现编队。以下简化后的无人机运动学模型可以用来设计控制算法：

$$\begin{cases} \dot{\mathbf{p}}_j = \mathbf{v}_j \\ \dot{\mathbf{v}}_j = \mathbf{u}_j \end{cases}$$

其中, \mathbf{u}_j 是加速度控制输入。可以将无人机在惯性系下的位置 \mathbf{p}_j 与速度 \mathbf{v}_j 视为无人机 j 的状态, 记为 $x_j = [\mathbf{p}_j, \mathbf{v}_j]^T$ 。

2.2 控制目标

常用的描述集群系统的编队控制问题的方法有三种 [3], 基于位置 (Position-based)、基于距离 (Distance-based) 与基于位移 (Displacement-based)。这里以基于位移为例, 对时变编队控制问题进行数学描述。

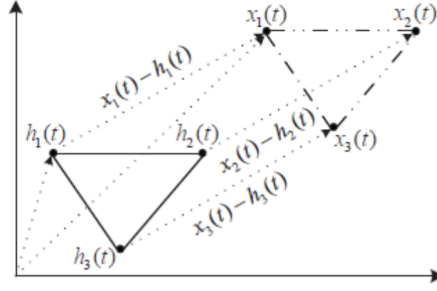


图 3: 无人机集群时变编队定义

假设无人机在 XY 水平面上保持编队, 在 Z 轴上定高飞行。如图 3 所示, 使用使用分段连续可微的向量 $h(t) = [h_1^T(t), h_2^T(t), \dots, h_N^T(t)]^T \in \mathbf{R}^{4N}$ 对期望的时变编队进行描述。如果

$$\lim_{t \rightarrow \infty} \left(x_i(t) - h_i(t) - \frac{1}{N} \sum_{j=1}^N (x_j(t) - h_j(t)) \right) = 0, i = 1, 2, \dots, N,$$

则称无人机集群系统实现了时变编队 $h(t)$ 。其中, $\frac{1}{N} \sum_{j=1}^N (x_j(t) - h_j(t))$ 可以视为编队中心, $h_i(t) = [h_{ip}(t), h_{iv}(t)]^T$ 是无人机 i 与编队中心保持的状态偏移向量, 包含沿 XY 两个轴的位置偏移向量 $h_{ip}(t) \in \mathbf{R}^2$ 和速度偏移向量 $h_{iv}(t) \in \mathbf{R}^2$ 。定义 $\xi_i = x_i(t) - h_i(t) - \frac{1}{N} \sum_{j=1}^N (x_j(t) - h_j(t))$ 是无人机 i 的编队控制误差, 可以用来评估控制算法的有效性。

2.3 控制算法设计与代码实现

为了使集群系统中 N 架无人机的编队控制误差 $\xi_i(t)$ 都可以快速收敛, 假设每架无人机除了可以使用自身状态 (包括位置与速度), 同时可以使用其他部分无人机的状态, 从而设计分布式的控制方法。

考虑这 N 架四旋翼无人机构成的集群系统的交互拓扑可以用有向图 $G = \{Q, \varepsilon, W\}$ 描述, 其中 $Q = \{q_1, q_2, \dots, q_N\}$ 是顶点集合, $\varepsilon \subseteq \{(q_i, q_j) : q_i, q_j \in Q\}$ 是边集合, $W = [w_{ij}] \in \mathbf{R}^{N \times N}$ 是具有非负元素 w_{ij} 的有权邻接矩阵。无人机 i 可以视为 G 中的顶点 q_i , 而从无人机 i 到无人机 j 的交互通道可表示为边 $e_{ij} = (q_i, q_j)$ 。当且仅当 $e_{ij} \in \varepsilon$, 另外, $w_{ii} = 0$ 对于所有 $i \in \{1, 2, \dots, N\}$ 。节点 q_i 的邻居的集合表示为 $N_i = \{q_j \in Q : (q_j, q_i) \in \varepsilon\}$ 。节点 q_i 的入度表示为 $\deg_{\text{in}}(q_i) = \sum_{j=1}^N w_{ij}$ 。则 $L = D - W$ 表示有向图 G 对应的拉普拉斯矩阵。假设图 G 具有有向生成树, 即存在至少一个节点能够通过有向路径到达所有其他节点。

参考 [4] 中的分布式编队控制算法, 可以设计每一架无人机的加速度控制输入为

$$u_i(t) = K_1 (x_i(t) - h_i(t)) + K_2 \sum_{j \in N_i} w_{ij} ((x_j(t) - h_j(t)) - (x_i(t) - h_i(t))) + \dot{h}_{iv}(t),$$

其中, $i = 1, 2, \dots, N$, K_1 与 K_2 是增益矩阵, $\dot{h}_{iv}(t)$ 是时变编队补偿控制输入。代码实现见 `xgc_ws/src/lab1/scripts/formation_controller_simple.py`。

3 实验步骤

在仿真平台中运行示例代码, 通过其中分布式编队控制算法, 可以实现 3 架无人机的固定编队控制。参照示例中状态话题的订阅与控制指令的发布方式, 设计控制算法并实现, 完成 5 架无人机的时变编队控制, 并分析编队控制误差。

3.1 运行示例代码

如图4所示, 在仿真平台中导入 `/home/xgc_documents/XGC/Config/lab1-simple.xconfig` 文件 (默认路径下应该能看到该文件)。

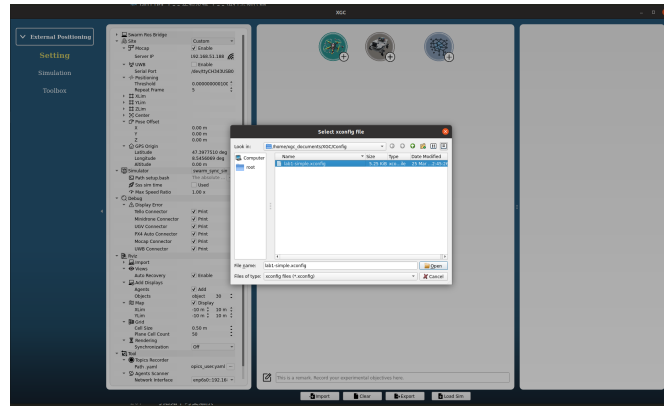


图 4: 导入示例配置

成功导入后, 可以看到三架无人机的配置, 同时在右侧可以看到名为 `lab1-simple` 的示例算法。该算法启动前会由仿真平台自动执行 `source /home/xgc_ws/devel/setup.bash` (该路径是容器下的路径, 不需要修改, 所有算法都相同), 确保执行 `roslaunch` 时可以找到功能包 `lab1`。之后自己编写的算法推荐存放在 `lab1` 功能包中, 也可以参照这个算法进行配置。除了 `py` 脚本名需要修改, 算法的其余配置都相同, 算法的命名可以随意更改。

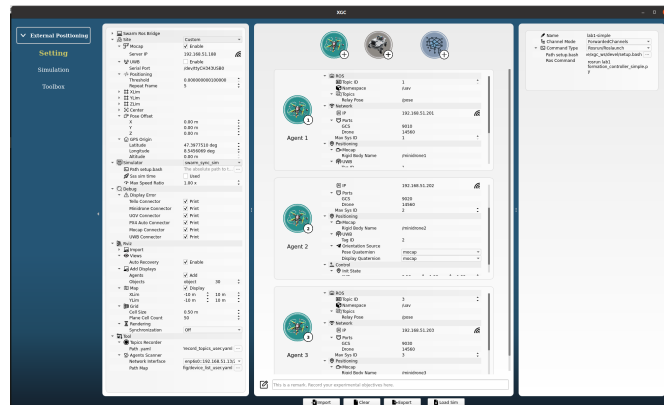


图 5: 成功导入效果

点击 Load Sim 加载仿真按钮，会自动跳转到主界面，并启动仿真器（启动成功，左侧仪表会变蓝色背景），如下图所示。

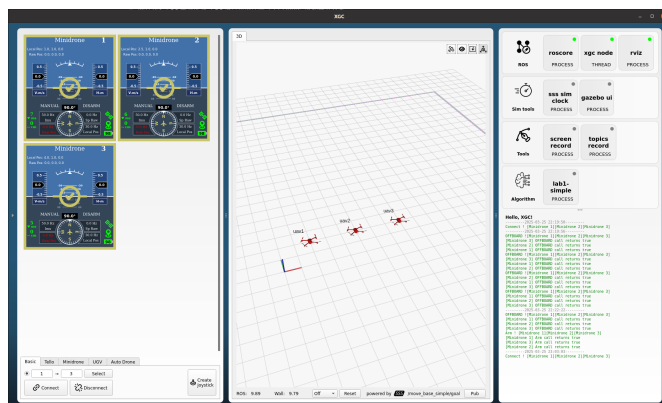


图 6: 仿真加载完成效果

注意，上一步添加了三架无人机和一个算法，3D 可视化区域可以看到三架无人机，在右侧可以看到多了一个 lab1-simple 算法按钮，点击该按钮，等同于依次执行以下命令

```
# 进入容器，执行 source 刷新环境变量，roslaunch 启动示例算法
docker exec -it xgc "/bin/bash"
source /home/xgc_ws/devel/setup.bash
roslaunch lab1 formation_controller_simple.py
```

点击右侧 lab1-simple 算法按钮，运行三架无人机编队算法（右上角指示灯常绿表示算法正在运行，点击按钮可以结束或者启动算法）。之后参照指南中起飞的步骤，切换无人机状态机，三架无人机将起飞并形成固定三角形编队。

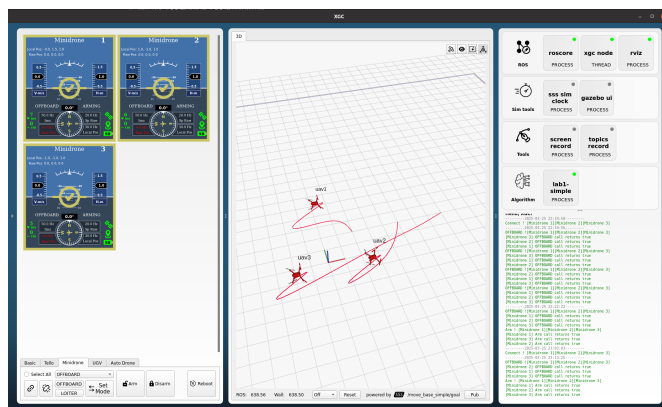


图 7: 固定三角形编队效果

示例代码 `xgc_ws/src/lab1/scripts/formation_controller_simple.py` 中已经将 XY 轴的位置与速度误差发布到 `/uavi/alg/error` 话题中，`i` 是无人机编号。按照指南，启动 PlotJuggler，订阅相关 ROS 话题，将字段拖入右侧显示区域可以查看实时编队控制误差。

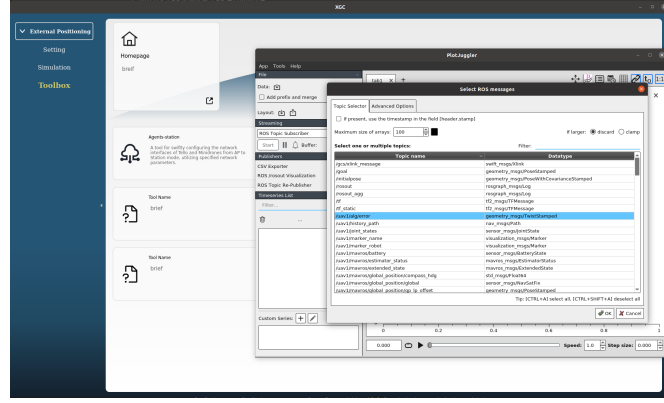


图 8: 选择三架无人机的控制误差的 ROS 话题

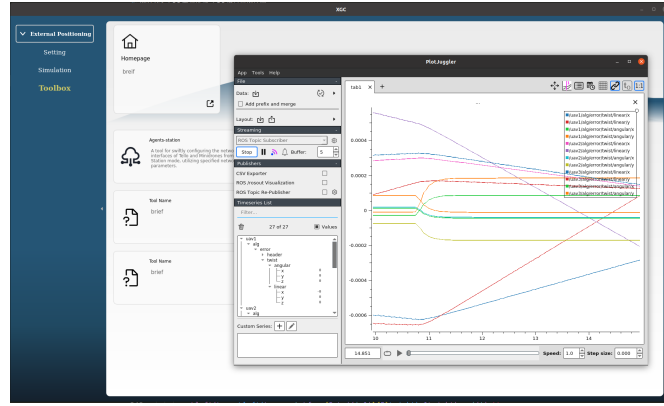


图 9: 查看误差曲线

3.2 实现时变编队控制目标

参考示例代码，设计控制算法实现 5 架四旋翼无人机的时变编队，编队在 XY 平面内进行。5 架四旋翼无人机之间的通信拓扑不限，可以是全联通，也可以选择与下图类似的交互关系。

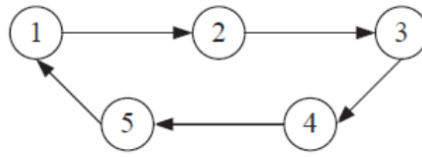


图 10: 通信拓扑

考虑如下期望时变编队

$$h_i(t) = \begin{bmatrix} r \left(\cos \left(\omega t + \frac{2\pi(i-1)}{5} \right) - 1 \right) g_i(t) \\ r \sin \left(\omega t + \frac{2\pi(i-1)}{5} \right) \\ -\omega r \sin \left(\omega t + \frac{2\pi(i-1)}{5} \right) g_i(t) \\ \omega r \cos \left(\omega t + \frac{2\pi(i-1)}{5} \right) \end{bmatrix} \quad (i = 1, 2, \dots, 5)$$

其中, $r = 7 \text{ m}$, $\omega = 0.214 \text{ rad/s}$, $g_i(t) = \text{sign}(\sin(\omega t/2 + \pi(i-1)/5))$ 。如果四旋翼无人机集群系统实现了期望的时变编队, 那么 5 架四旋翼无人机将会在保持相位差的同时按照数字 8 的编队飞行, 最终效果参考 [4], 5 架无人机的位置轨迹应该类似图 11。

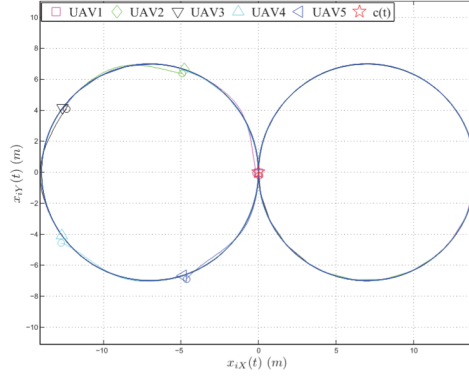


图 11: 期望实现的 5 架无人机的编队效果 (位置轨迹在 XY 平面上的投影)

3.3 数据记录与分析

参考示例代码, 实现 5 架无人机编队控制误差的发布。并根据指南, 在 PlotJuggler 工具中对 5 架无人机的误差进行实时可视化并分析。也可以在容器中使用 rosbag 录制话题数据, 使用绘图工具进行分析。

4 作业要求

在仿真平台上, 添加 5 架四旋翼无人机, 实现 3.2 中队形, 并分析控制误差, 具体控制要求如下:

- 每架无人机编队控制误差在 15 s 内收敛, XY 轴位置的稳态误差小于 3 cm。

作业提交要求如下:

- 以小组为单位撰写实验报告, 内容包括但不限于四旋翼无人机集群建模、控制问题描述、控制算法设计与实现代码介绍、编队过程仿真平台截图、编队控制误差时间曲线图与分析等。报告文件命名, 例如: 实验一报告-组员姓名.pdf(docx)。
- 将 lab1 的 ROS 功能包与实验报告打包成压缩文件, 需要包含算法源码所在 src 或 scripts 文件夹。压缩文件命名格式, 例如实验一-组员姓名.zip(rar)。

参考文献

- [1] Rui Wang and Jinkun Liu. Adaptive formation control of quadrotor unmanned aerial vehicles with bounded control thrust. *Chinese Journal of Aeronautics*, 30(2):807–817, 2017.
- [2] PX4 Autopilot. Px4 自动驾驶用户指南-控制器图解, 2024. URL https://docs.px4.io/v1.14/zh/flight_stack/controller_diagrams.html.
- [3] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

-
- [4] Xiwang Dong, Bocheng Yu, Zongying Shi, and Yisheng Zhong. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Transactions on Control Systems Technology*, 23(1):340–348, 2014.