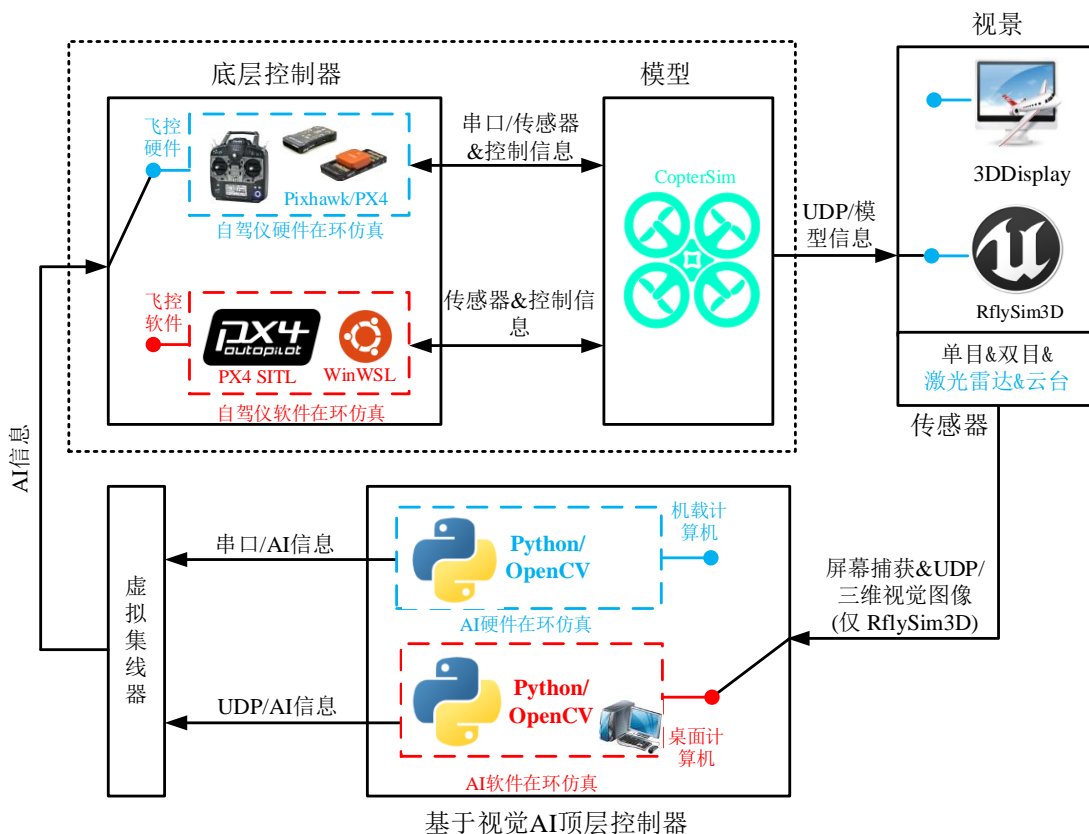


RflySim 第 10 章 集群控制算法开发

本讲专注于无人机集群的通信体系结构与协同控制技术, 内容被系统性地划分为基础实验、进阶实验两个层次, 旨在实现无人机在局域网中的高效组网通信以及精密编队控制。实验平台集成了 RflyUdpFast 通信模块, 支持 MATLAB Simulink, Python 等环境, 支持本地单机模拟及多机联网下的集群控制实验。



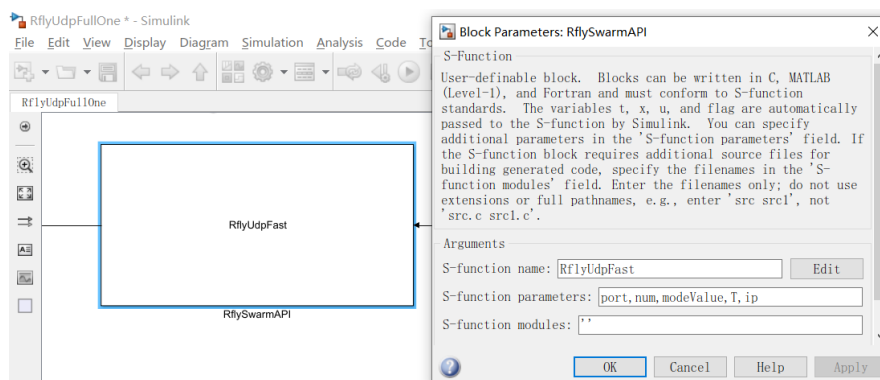
集群概念最早源于生物学研究。法国动物学家 Grasse 基于白蚁筑巢行为, 首次提出了共识自主性(Stigmergy)的概念: 一种个体间间歇协调的机制, 即无需任何集中规划以及直接通信完成复杂智能活动。这是自主集群概念开始走入人类视野并逐步发展的开端。从生物系统延展到多智能体系统 (Multi-Agent System, MAS), 集群的概念亦在不断演化和丰富。多智能体系统是指一个系统内同时存在多个需要控制的智能体, 每个智能体在和环境交互的同时还需和其他的智能体展开竞争或者合作, 当多智能体系统具备很大的规模用于执行非常复杂的任务时, 它被称为集群[1]。集群控制主要研究的是一致性问题, 即利用分布式协调控制算法, 使“网络”中各自主体的状态或输出达到某种一致性。所谓分布式协调控制算法, 即通过简单的局部规则, 形成协调的全局行为。

Simulink 集群接口

打开 “ RflySimAPIs\SimulinkSwarmAPI\RflyUdpFullOne.slx ” 文件, 如左下图可看到

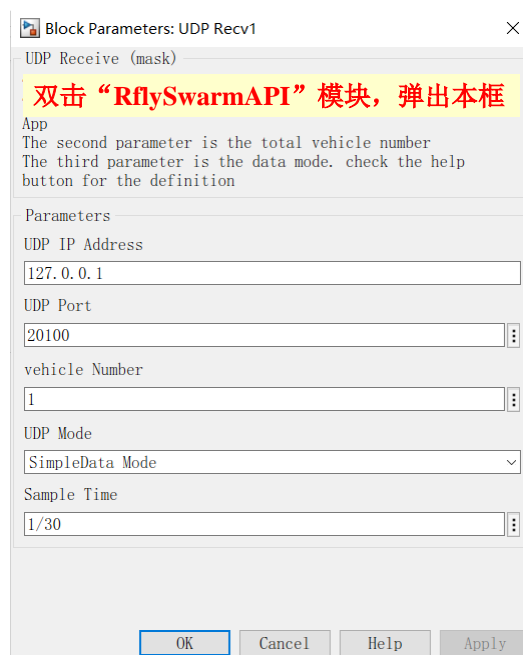
“RflySwarmAPI”模块即为集群通信模块。

该模块有 Simulink S 函数通过 C++混编实现，源文件见“RflyUdpFast.cpp”。



“RflySwarmAPI”模块需要和“RflyUdpFast.mexw64”放在同一个文件夹才能调用。因此新建 slx 项目时，拷贝“RflySwarmAPI”模块的同时还需要拷贝“RflyUdpFast.mexw64”文件到新建 slx 文件所在目录。“RflyUdpFast.mexw64”是源文件“RflyUdpFast.cpp”在 MATLAB 中编译出来的。

在 Simulink 中双击“RflySwarmAPI”模块查看详细信息。



(a) 第一项“UDP IP Address”是目标电脑的 IP 地址，输入“127.0.0.1”则只能接受本机的 CopterSim 转发的 Pixhawk 自驾仪状态并进行控制；“255.255.255.255”则能接收并控制局域网内所有电脑中运行的 CopterSim 程序（其他电脑的 CopterSim 需要勾选“联机”按钮）；“192.168.1.12”之类的指定 IP 则只会向该 IP 地址的主机发送控制指令。一般而言，在小范围集群的时候“255.255.255.255”广播已经能够满足需求，当飞机数量继续增多，则需要启用指定 IP 来减小网络负载，提高通信速度和可靠性。

(b) 第二项“UDP Port”是第一个飞机的初始端口号，默认起始端口是 20100。每个 CopterSim 的收发消息需要各占用一个端口。例如，本模块需要仿真飞机 ID 为 10~15 的飞机，这一项则需要填 $20100+10*2=20120$ ，后面一项“Vehicle number”飞机数量需要填 5。

(c) 第三项“Vehicle number”飞机数量表示需要连接的 CopterSim 数量, 同时控制该模块的输入输出端口数量, 如果输入 10, 则模块会自动生成 10 对输入输出接口。

(d) 第四项“UDP mode”是输入输出接口的数据模式协议, 主要包含 FullData 完整模式 (数据最全, 但传输数据量较大); SimpleData 精简数据模式 (较多飞机>8, 避免数据过大网络阻塞) 和 UltraSimple 超精简模式 (单电脑飞机数>20), 延迟更小。

(e) 第五项是“Sample Time”采样时间, 该时间应该与 Simulink 仿真时间对应。

RflySwarmAPI 模块共有三种 UDP 模式, 分别为 FullData 模式、SimpleData 模式和 UltraSimple 模式。

Python 集群接口

注意 Python 代码阅读和运行需要安装 VS Code 并关联平台 Python38 环境, 具体安装流程请参考高级课程第 6 讲。

集群控制的 Python 通信接口文件和视觉 Python 控制接口完全相同, 都在“PX4MavCtrlV4.py”接口文件中, 其基本使用方法与高级课程第 6 讲相同。

Python 集群控制使用的接口文件为“PX4MavCtrlV4.py”和 PythonVision 视觉控制使用的 API 文件完全相同。

根据第 6 讲的教程可知, Python 与 CopterSim/PX4 通信包含了两种通信模式 (UDP 结构体和 MAVLink 数据流) 和两种优化模式 (Full 和 Simple)。

MAVLink_Full 和 MAVLink_Simple 通信模式 (对应 InitMavLoop(2)和 InitMavLoop(3)) 直接使用 MAVLink 数据流进行 Python 与 PX4 的通信 (经过 CopterSim 中转), 数据量大、占用带宽高, 但是和真机更贴近且功能完善, 在大规模集群时易阻塞宽带;

UDP_Full 和 UDP_Simple 模式 (对应 InitMavLoop(0)和 InitMavLoop(1))下, Python 发送精简 UDP 结构体消息与 CopterSim 通信, 后者压缩或解压后再通过 MAVLink 与 PX4 通信, 这种方法能够有效的减小局域网内的数据量, 适合集群仿真控制。

Python 脚本与 CopterSim 通信使用了 20100+2i 系列接口来接收飞控数据

(InitMavLoop 函数启用监听, uav***变量来读取数据), 30100+2i 系列接口来接收真值数据 (InitTrueDataLoop 函数来启用数据监听, true***变量来读取数据)。

除了与 PX4 与 CopterSim 联合仿真 (6DOF 高粒度模型); 集群平台还支持 Python 与 RflySim3D 精简仿真 (质点低粒度模型), 使用 Python 内部精简飞机模型, 计算量小, 支持大规模集群。

控制模型

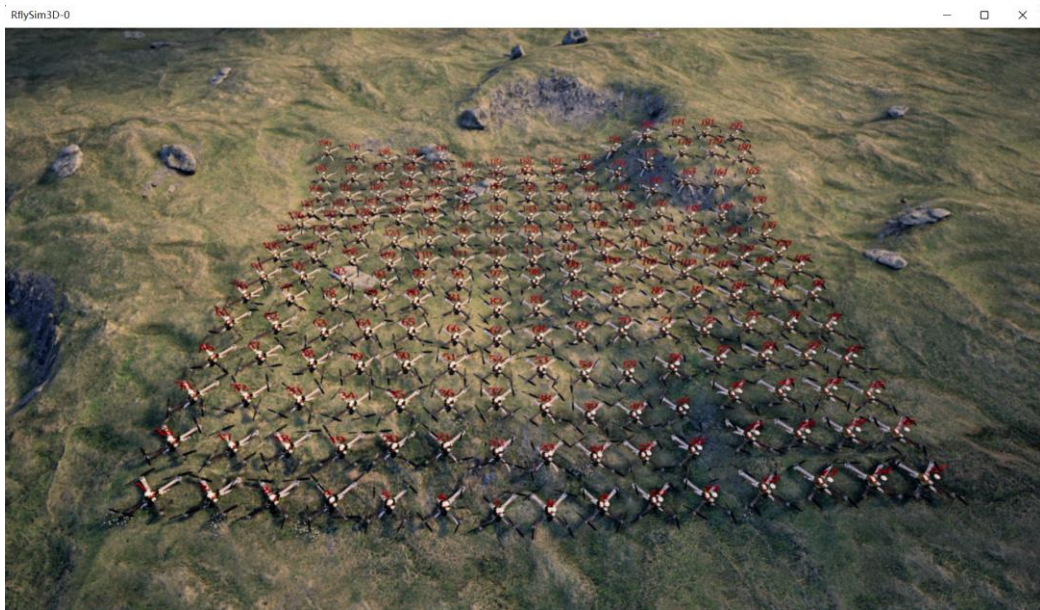
在控制模型层面, 实验平台囊括了多种适用于集群控制的模型, 包括但不限于基于高精度动态模型的 PX4 飞行控制器集成方案, 以及 Simulink 环境下的自定义控制器设计。同时, 还支持质点模型、旋翼机型和固定翼机型等多种物理模型的仿真与控制。

从模型精度的角度, 使用高精度 6DOF 模型 (CopterSim) + 真实飞控系统 (PX4) 的软/硬件在环仿真闭环的方式, 能够有效提高模型可信度, 从而减小仿真与真机实验的差距。

但是上述构架的代码运算量较为复杂, 导致一台电脑上运行的无人机数量受到限制 (通常 SITL 软件在环<20 架, HITL 硬件在环<30 架)。

为了提高单台电脑仿真集群飞机的数量，就需要降低模型精度并使用简化飞控模型。

因此本平台在 Python 下开发出了质点多旋翼模型，只需 Python 和 RflySim3D 两个软件即可在单台电脑上实现百驾级别的无人机集群仿真（免费版只支持最多 12 驾的例程），且每个飞机的输入输出接口与 SITL 和 HITL 仿真保持一致，确保向真机实验顺利过渡。



软硬件在环

为确保实验的多样性和深度，平台支持软件在环（SIL）和硬件在环（HIL）两种仿真模式，以充分验证和调试集群控制算法在虚拟与实际硬件条件下的性能表现。而在控制策略方面，涵盖了位置控制、速度控制和姿态控制等多种控制模式，从而全方位探究无人机集群在复杂任务执行中的通信协调与控制策略。

```
SITLRun
-----
Please input UAV swarm number:4
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../../logs
Using Airframe File: 10016_iris
starting instance 1 in /mnt/e/PX4PSP/Firmware/build/px4_sitl_default/instance_1
starting instance 2 in /mnt/e/PX4PSP/Firmware/build/px4_sitl_default/instance_2
starting instance 3 in /mnt/e/PX4PSP/Firmware/build/px4_sitl_default/instance_3
starting instance 4 in /mnt/e/PX4PSP/Firmware/build/px4_sitl_default/instance_4
Copying rcS files
PX4 instances start finished
Press any key to exit
```



可执行文件生成

MATLAB 本身会占用大量的 CPU 和内存资源，在运行复杂的 Simulink 控制程序时，一方面计算量太大导致算法运行缓慢，无法达到实时要求（Simulink 中运行 1s 中大于现实时钟 1s），这样就无法实时控制仿真系统（或真实系统）的集群飞机；第二方面，在仿真时 Simulink 如果占用大量的计算资源，会导致 RflySim3D 和 CopterSim 的计算资源分配较少，导致飞机仿真变差，飞机剧烈抖动甚至坠机。

将 Simulink 控制器编译生成 exe 之后，算法可以脱离 MATLAB 运行，而且本身是二进制可执行文件，运行效率非常高，即使大型的控制算法，也能保证实时控制。

```
C:\Users\dream\Desktop\DroneyeeSwarm0505\SimulinkDemo\RflyUdp
*** Using a default buffer of size 1024 for logging var
** starting the model **
**May run forever. Model stop time set to infinity.**
** Sample Time: 0.033333 **
```

▼	RflyUdpUltraSimpleEight.exe	0%	0.6 MB
	C:\PX4PSPFull\RflySimAPIs\...	占用几乎为0	