

# 基于RflySim的底层控制算法开发工作流

## 1. 概述与开发环境搭建

### 1.0 背景概述

无人系统底层控制算法作为连接上层决策与底层执行的“脊髓”中枢，承担着将期望指令转化为精确机动行为的核心任务，直接决定了载具的稳定性与抗扰性能。随着技术演进，该领域正逐步从经典的PID控制向自抗扰（ADRC）、模型预测（MPC）及智能控制过渡，以解决欠驱动、强耦合、模型不确定性及复杂环境干扰等严峻挑战。当前，基于模型设计（MBD）的开发范式已成为行业标准，通过RflySim等平台实现从软件在环（SIL）到硬件在环（HIL）的“虚实结合”迭代，有效突破了传统“手写试飞”模式的效率瓶颈与安全风险，为高性能控制算法的工程化落地提供了从理论设计到实飞验收的完整闭环支撑。

### 1.1 核心组件与原理

RflySim 平台的底层控制系统采用开源自驾仪（飞行控制器，简称飞控）"PX4 软件+Pixhawk硬件"体系。用户可以在Simulink 中开发自己的位置控制与滤波估计算法，并以模块（独立APP）的方式运行在PX4 构架的飞控软件系统中。PX4 以其高度模块化、可扩展性强、支持多种飞行模式等特点，在无人机领域备受欢迎。它不仅适用于多旋翼、固定翼、VTOL 等多种飞行器，还可以适配无人车、无人艇、水下潜航器等其他无人系统。

#### 1.1.1 PX4自驾仪介绍

PX4是一款开源的自动驾驶仪软件，专为无人机和其他自主车辆设计。它具有高度的模块化架构，支持多种飞行器类型，包括多旋翼、固定翼、垂直起降飞机（VTOL）等。PX4提供了丰富的功能，如姿态估计、飞行控制、导航、任务规划等，被广泛应用于学术研究、商业产品开发和工业应用中。PX4软件系统主要文件夹结构如下：

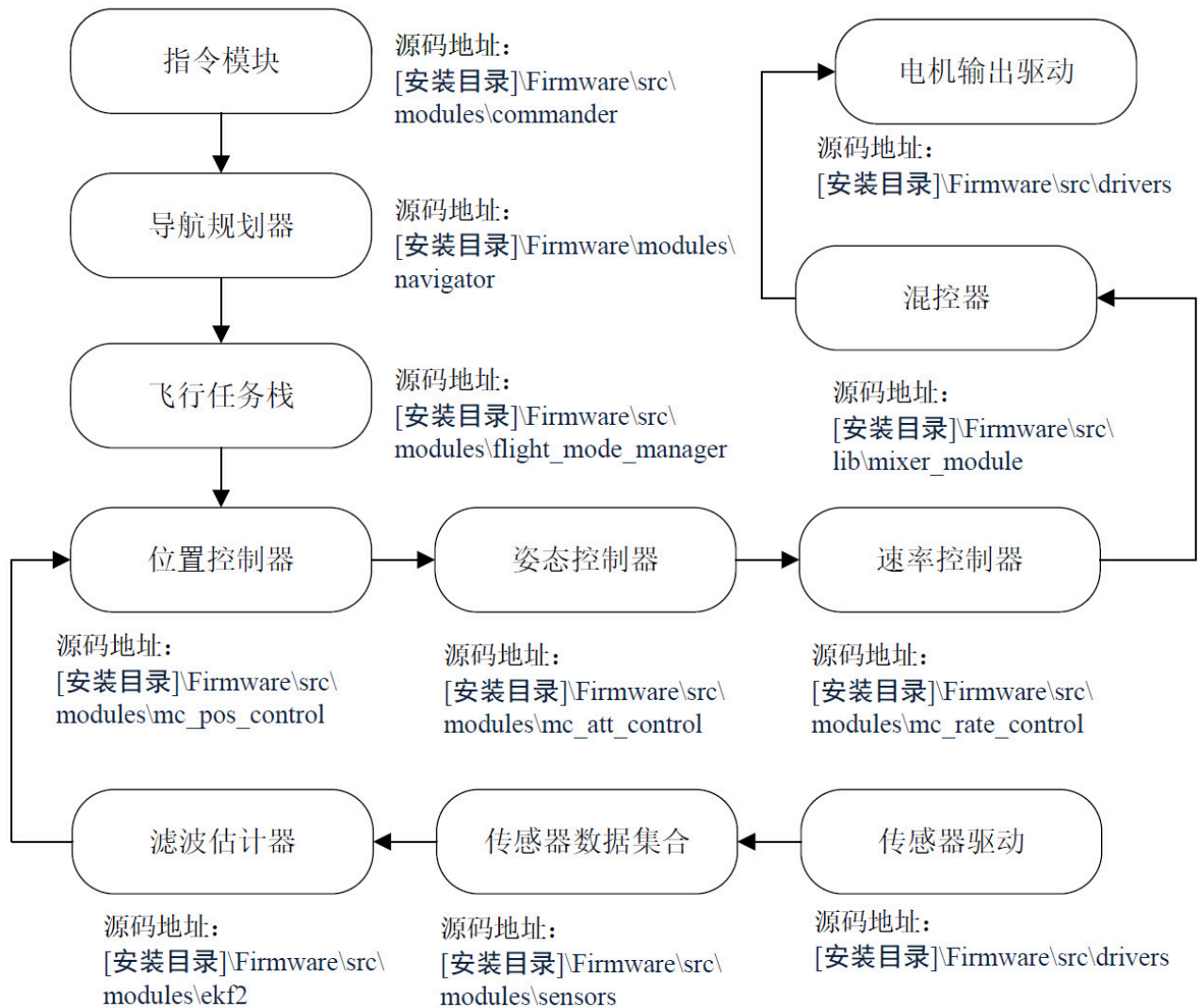
```

1 | [RflySim安装目录]\Firmware/
2 | └─ boards/ # [硬件配置] 各大厂商飞控板的定义 (引脚、时钟、驱动配置)
3 | └─ src/ # [核心代码] 飞控系统的核心源代码
4 |   └─ modules/ # 核心功能模块 (EKF2状态估计, 导航, 飞行控制, MAVLink等)
5 |   └─ drivers/ # 传感器与外设驱动 (IMU, GPS, 磁力计, 电调, 距离传感器等)
6 |   └─ lib/ # 共享算法库 (Math, Matrix, PID, 混控器, 磁场模型等)
7 |   └─ systemcmds/ # 系统级终端命令 (top, reboot, listener, param等)
8 |   └─ examples/ # 开发者示例代码 (Hello Sky 等)
9 |   └─ include/ # 公共头文件 API
10 | └─ msg/ # [通信协议] uORB 消息定义文件 (.msg)
11 | └─ ROMFS/ # [启动系统] 嵌入式只读文件系统 (系统启动与配置)
12 | └─ platforms/ # [平台适配] 操作系统与硬件抽象层 (HAL)
13 | └─ Tools/ # [工具集] 开发、编译、仿真与调试辅助工具
14 | └─ cmake/ # [构建系统] CMake 编译脚本与宏定义
15 | └─ build/ # [编译产物] 编译生成的二进制文件存放处 (通常不纳入版本控制)
16 | └─ test/ # [测试] 单元测试集 (Unit Tests)
17 | └─ integrationtests/ # [测试] 集成测试集 (Integration Tests)
18 | └─ Documentation/ # [文档] 官方开发文档资料

```

注：更多详细介绍可见 [7.1 PX4 文件目录详细说明](#) 或 [PX4官方文档](#)，源码仓库为：  
<https://github.com/PX4/PX4-Autopilot.git>。

PX4的模块化设计使得开发者可以方便地添加新的功能模块，同时保持系统的稳定性和兼容性。在RflySim环境中，用户可以在不修改原有核心代码的情况下，通过添加新的模块来实现自己的控制算法。

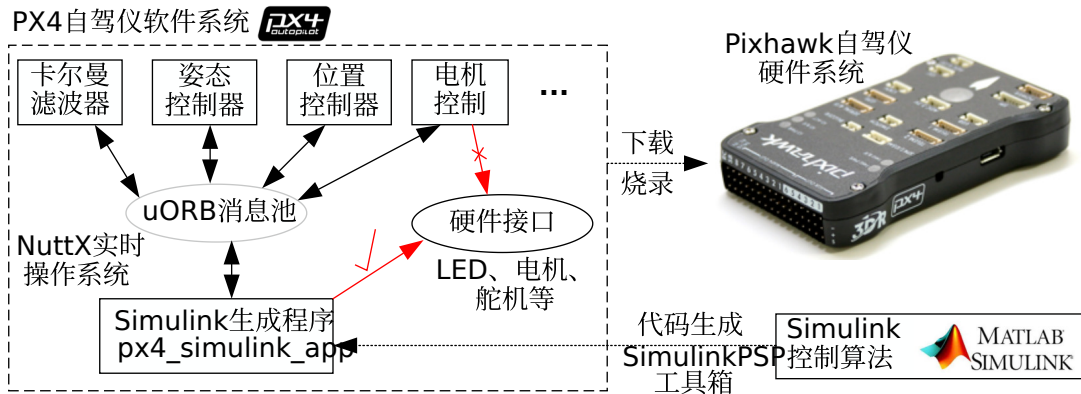


### 1.1.2 基于Simulink/PX4 的自动代码生成工具线

RflySim 底层控制的核心基于PX4软件系统的模块化特性，结合MATLAB/SimulinkPSP工具箱，使用户能够快速搭建自定义控制算法，并与PX4uORB 消息机制进行高效交互。借助MATLAB/Simulink 自动代码生成功能和RflySim 自定义代码模板（Target Language Compiler, TLC），用户可以快速编译生成PX4 兼容的飞控固件，并直接烧录到Pixhawk 硬件上，实现高效开发与验证。如下图所示，PX4 软件系统由多个小模块（例如姿态控制、位置控制、电机控制等）组成，每个模块独立运行（支持多进程并行），各个模块通过uORB 消息机制的订阅与发布功能实现数据传输和交互。当Simulink 生成的代码部署到PX4 自驾仪软件后，它不会干扰原生PX4 自驾仪软件的正常运行，而是新增一个名为 `px4_simulink_app` 的独立模块（独立线程）的形式并行运行。这一机制具有以下特点：

1. 独立线程运行：`px4_simulink_app` 作为新模块，不会影响原生PX4 的飞行控制逻辑，而是通过uORB 进行数据交互。
2. 与PX4 现有功能兼容：可读取PX4 传感器数据（如IMU、GPS、气压计），并发布控制命令给执行器。

3. 高效的数据交互：由于Simulink 生成的代码采用PX4 进程间通信机制（uORB），数据传输延迟极低，能够满足飞行控制的实时性需求。

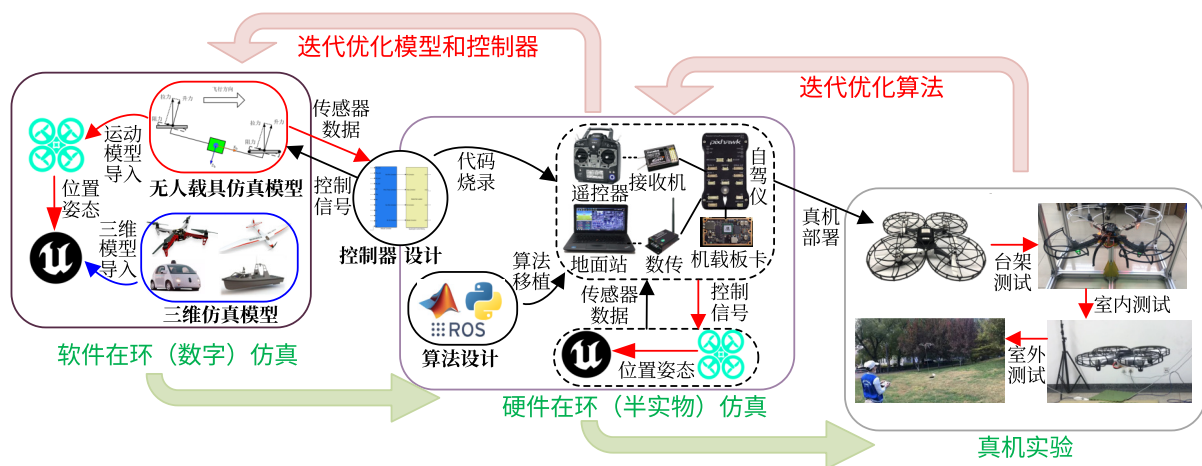


由于原生PX4 控制算法可能需要访问和 `px4_simulink_app` 相同的硬件输出资源，这可能会导致读写冲突。例如，当如上图所示的中Simulink生成的 `px4_simulink_app` 模块同时控制电机硬件接口时，会互相抢夺电机控制权，产生一顿一顿的冲突现象，导致无人机飞行异常。因此，平台的一键部署脚本提供了自动屏蔽PX4 电机控制代码的选项，确保只有 `px4_simulink_app` 模块能够控制电机输出。

**注意：** 关于RflySim 工具链的PSP工具箱关于底层控制算法开发部分的详细介绍可见：[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\API.pdf](#)

## 1.2 底层控制框架理解

基于 RflySim 平台的底层控制开发遵循“模型驱动、虚实迭代”的V型开发范式。该 workflow 以 MATLAB/Simulink 为算法设计核心，首先通过高保真软件在环仿真（MIL/SIL）验证控制律的理论可行性与逻辑完备性；进而利用自动代码生成技术将算法无缝部署至真实飞控，在硬件在环仿真（HIL）中以“零试错成本”完成对算力边界、通信时序及故障容错的半实物验证；最终通过台架与实飞实验完成物理验收，并利用飞行数据反哺修正仿真模型。这一“设计-仿真-实现-验证”的闭环流程，有效突破了理论算法向工程代码转化的技术鸿沟，极大提升了底层控制算法开发的安全性、可靠性与研发效率。



## 1.3 软硬件准备

### 1.3.1 软件准备

1. **MATLAB/Simulink**: RflySim 底层控制算法开发是基于MATLAB/Simulink来进行控制器设计，因此需要安装MATLAB/Simulink 2022a 或更高版本。

### 1.3.2 硬件准备

1. **PX4 支持飞控(自驾仪)**: 若需要进行HIL仿真实验，需要支持运行PX4软件系统的飞控，推荐飞控有：Holybro Pixhawk 6X & 6X mini、Holybro Pixhawk 6C、CUAV Pixhawk V6X、Holybro Pixhawk 4。其余支持飞控可参见：

[https://docs.px4.io/main/en/flight\\_controller/](https://docs.px4.io/main/en/flight_controller/)。相关配置过程可见：

[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e2.FCIntro](#)

2. **遥控器及接收机**: 遥控器及接收机主要与飞控进行连接，用于手动控制仿真或实飞无人机进行飞行，以此来验证底层控制算法的效果。推荐的遥控器有：WFLY ET10和RF209S等。需要注意的是：购买前需要确认与飞控的兼容性，建议提前询问商家。相关配置过程可见：

[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e1.RCIntro](#)

3. **实飞整机**: 若需要将测试的仿真，可在电商平台购买相关组件进行组装。具体组装流程可参见：

[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\1.BasicExps\e10.DroneAssTutorial](#)

。

## 1.4 开发环境搭建

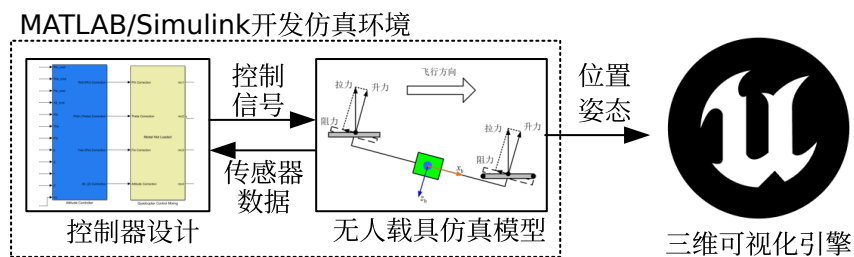
整个开发环境均是基于RflySim工具链来实现的，工具链的下载链接可见：

<https://rflysim.com/download.html>。具体安装教程可见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

下面将以四旋翼无人机为例详细进行讲解如何基于RflySim实现从前期开发、仿真验证到真机实飞全流程。

## 2. 第一阶段：建模与模型在环仿真（MIL/SIL）

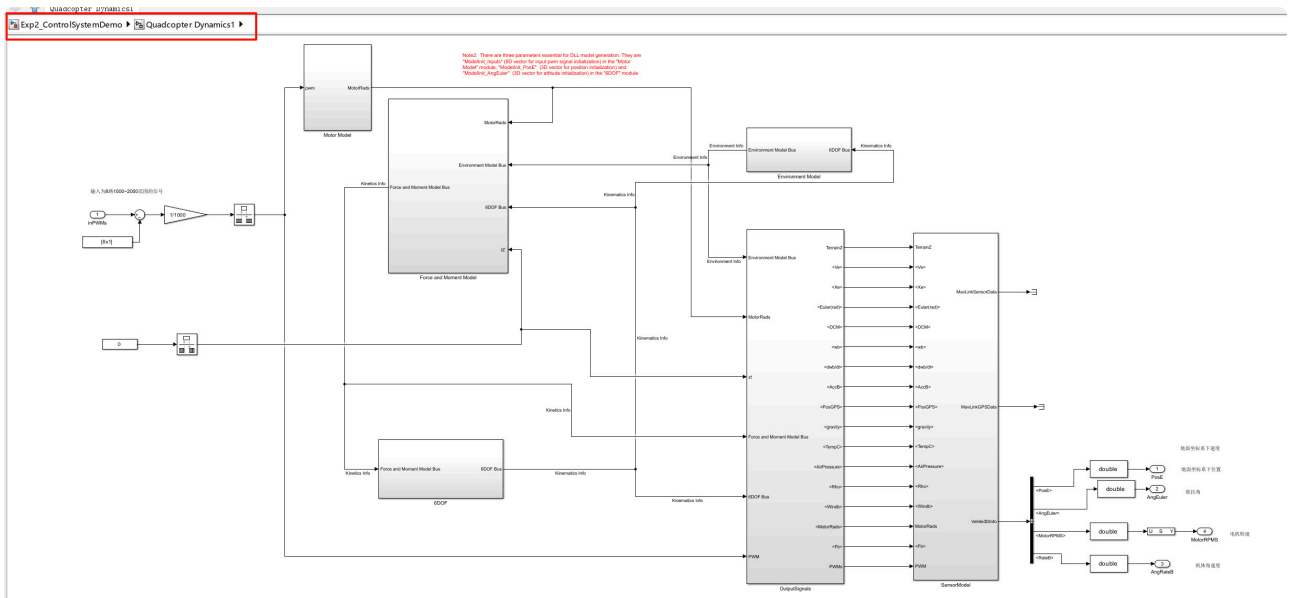
本阶段的主要任务是在零硬件成本、零炸机风险的前提下，通过数学手段验证控制逻辑的完备性。如下图所示，整个过程包括控制器设计、无人载具仿真模型开发以及三维可视化引擎观察效果。



### 2.1 模型定义与分析

四旋翼无人机是一个典型的 6 自由度（6-DOF）刚体。我们需要算出力（Force）和力矩（Moment）如何影响位置和姿态，旨在 Simulink 环境中高保真地复现真实飞行器的物理属性与运动规律。该过程基于 **牛顿-欧拉方程** 建立六自由度（6-DOF）刚体模型，精确解算推力与力矩如何驱动机体产生位置位移与姿态旋转，同时引入 **一阶惯性环节** 模拟电机从指令下达到实际转速响应的物理迟滞特性，并结合 **混控矩阵** 完成从控制量到执行器输出的数学映射。这一“数字孪生”模型作为控制算法的直接被控对象，通过严谨的数学描述替代了理想化假设，为后续验证控制律的鲁棒性提供了符合物理法则的仿真基座。如例程文件：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp4\\_AtitudeSystemCodeGen.slx](#)



如图所示，即为四旋翼无人机的运动模型，在该框架中，运动模型系统整体细分为多个子系统模块，如：电机模块、力和力矩模块、环境模块、6自由度模块等等。关于无人系统的模型搭建及学习可见：[\[RflySim安装目录\]\RflySimAPIs\4.RflySimModel\API.pdf](#)

## 2.2 控制器需求定义与分析

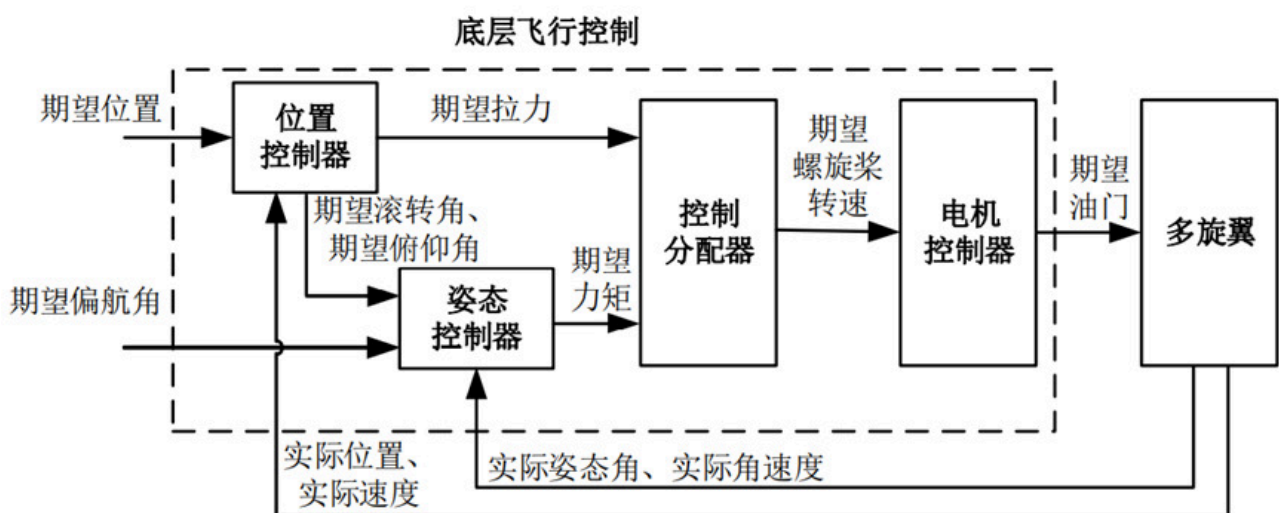
对于四旋翼无人机的控制需求，可定义如下：

- 输入数据：
  - i. 遥控器 Ch1~Ch5 通道信号，数据范围大约为 1100-1900，在处理遥控器数据时需要校准或考虑死区；
  - ii. 角速度反馈量 AngRateB，三个分量用 p,q,r 表示（单位：rad/s），分别代表滚转角速度（沿机体x轴转动）、俯仰角速度（沿机体y轴转动）和偏航角速度（沿机体z轴转动）；
  - iii. 多旋翼欧拉角（单位为 rad）。这里主要考虑滚转角和俯仰角，暂不考虑偏航控制。
- 输出数据：
  - i. 四个电机的PWM控制信号，数据范围 1000~2000；
  - ii. 是否解锁标识符，数据类型 bool 型
- 实现效果：
  - i. CH1向左推滚转摇杆（即CH1<1500）控制多旋翼向左飞；反则则向右飞。
  - ii. CH2向前推俯仰摇杆（即CH2<1500）控制多旋翼向前飞；反则则向后飞。
  - iii. CH3油门通道控制飞机升降；
  - iv. CH5向下拉拨杆开关（即CH5>1500）解锁控制器。

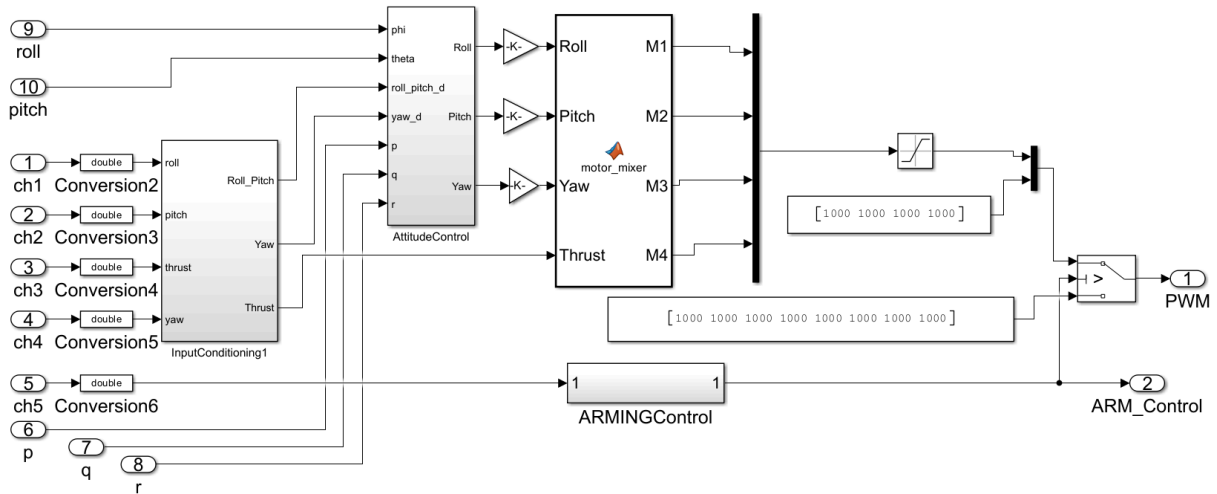


## 2.3 底层控制器模型搭建与仿真

基于上一步的需求分析，可知四旋翼是一个欠驱动系统，即独立控制量个数小于系统自由度个数的系统，它有六个自由度（三自由度位置、三自由度姿态），但是只有四个独立输入（四个电机转速）。整体设计思路如下图所示，采用内外环控制。外环控制器为位置控制器，内环控制器为姿态控制器，即先控制好四旋翼的姿态，再控制四旋翼的位置。



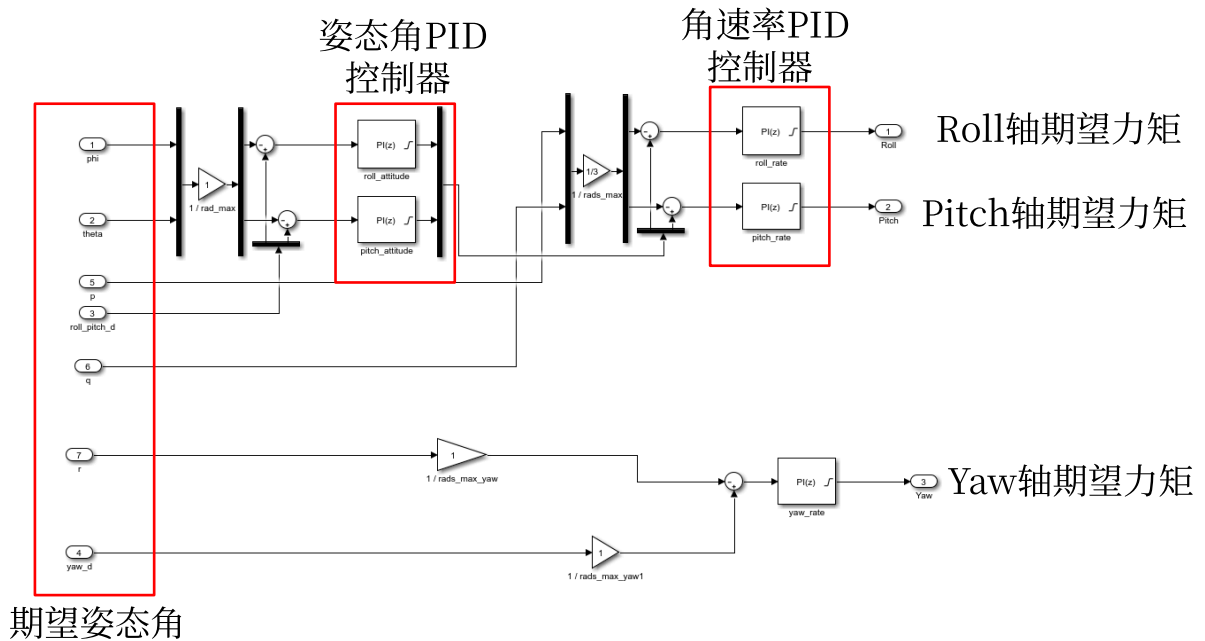
下面我们先对姿态环进行设计，可直接在Simulink中搭建一个四旋翼无人机的控制器模型，如下图所示：



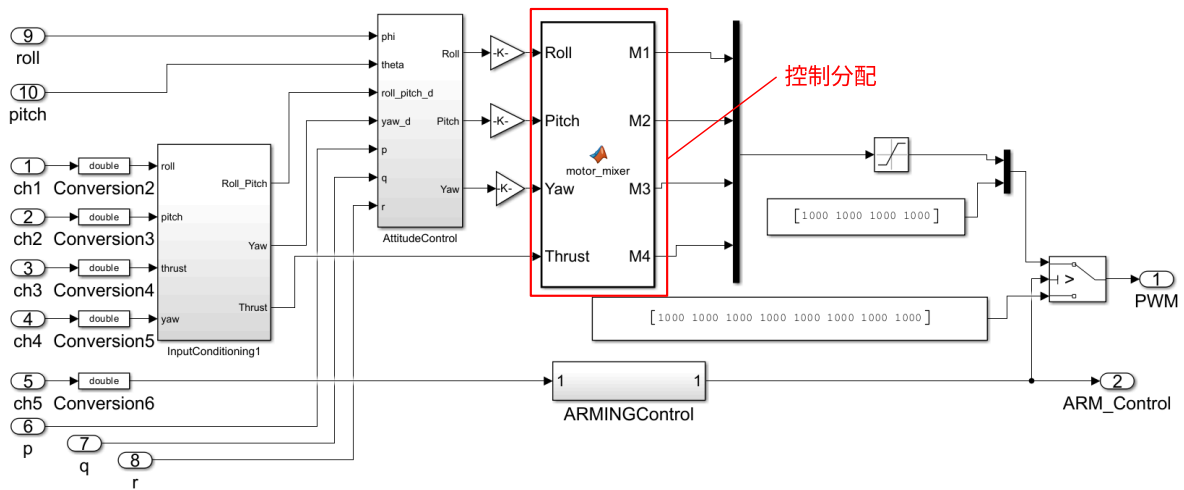
详细文件可见：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp1\\_AttitudeController.slx](#)

打开该文件可仔细阅读其中的子模块的实现方法，为了对功能进行完善，已经将偏航通道的控制加入。



姿态控制的主要目标是将 **期望姿态角** -> **解算期望力矩**



其中，控制分配器主要是将控制器输出的角度转换为电机的转速控制指令。如多旋翼是四旋翼，那么力和力矩四个独立输入将换算成四个螺旋桨的控制。我们期望这些控制作用到真实的四旋翼也能产生同样的力和力矩。如多旋翼是十八旋翼，那么也只有力和力矩四个独立输入，并将其换算成十八个螺旋桨的控制。

## 2.4 模型在环仿真 (MIL/SIL)

前文（2.1 模型定义与分析、2.3 底层控制器模型搭建与仿真）已经完成了，控制器、运动模型的搭建，及RflySim显示模块整合

通过前文对控制器的设计，我们需要设置控制器的输入和输出，输入为链接模拟遥控器的CH1~CH5和反馈量，输出为PWM，PWM作为模型的输入，模型输出飞机的状态量分别作为控制器的反馈量和RflySim3D显示模块的输入量，如下图所示。具体例程文件可见：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp2\\_AttitudeSystemCodeGen.slx](#)



**注意：** 图中的RflySim3D模块为Simulink在仿真时，与RflySim3D进行数据交互的模块，点击打开后，可查看具体的定义。详细协议说明可参考：

点击图中左侧CH5的Slide Switch模块到on，即代表无人机解锁，解锁之后CH1~CH4的Slide模块即可实现对无人机的横滚、俯仰、油门、偏航的调整。同时，在RflySim3D中也可看到飞机的飞行情况，完成软件仿真。



## 3. 第二阶段：硬件在环仿真（HIL）

上述2. 第一阶段：建模与模型在环仿真（MIL/SIL）完全基于Simulink进行了数字样机级别的软件在环仿真，初步验证了搭建的姿态控制器的准确性，并且实现了整个仿真的闭环。那么，接下来就需要进行硬件在环仿真。

### 3.1 RflySim环境配置确认

RflySim工具链支持PX4的多个版本，因此需要在进行HIL仿真时，根据自己已有的硬件设备对RflySim工具链进行重新配置，以保证能够正常进行HIL仿真。例如，我们目前的手上得硬件为Pixhawk 6X mini飞控（如下图所示）。那么，我们可以在其[官网](#)上查询到它的编译命令为：`px4_fmu-v6x_default`，可参考例程：

[\[RflySim安装目录\]\RflySimAPIs\2.RflySimUsage\1.BasicExps\e18\\_PX4VersionSwitch](#)。  
将RflySim工具链的当前配置进行修改



**注意：** 若已经配置过，则无需重复配置，直接进行下一步。

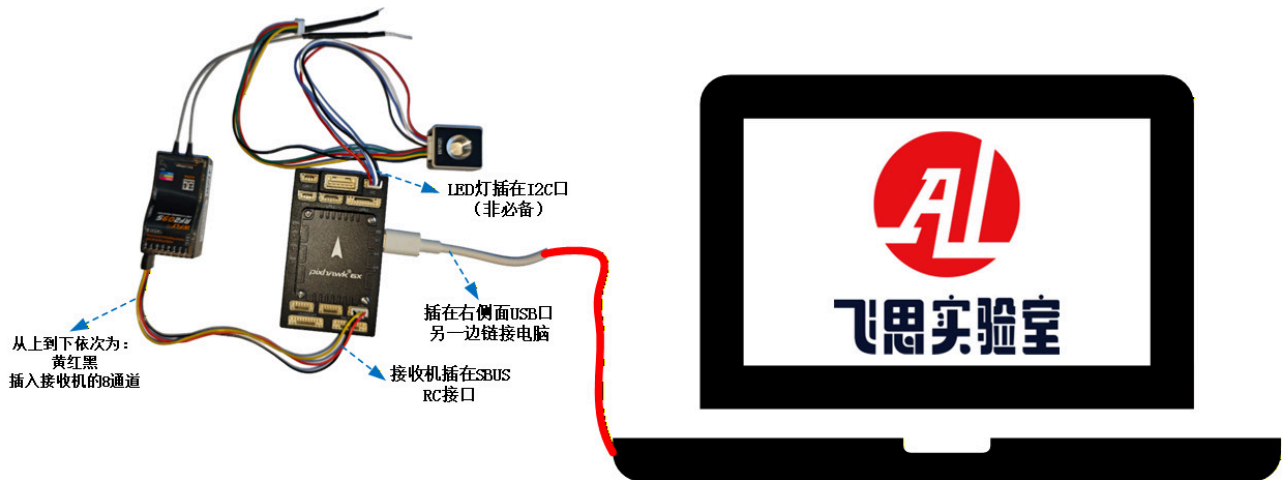
### 3.2 硬件环境搭建与配置

硬件环境的搭建与配置主要是指具备硬件在环仿真的最基本的设备，飞思实验室官方有一套专为HIL仿真打造的套装，可点击链接购买：

<https://item.taobao.com/item.htm?id=847685763887&spm=a1z10.5-c-s.w4002-25696003294.10.4a024dc8WOzjmF>



实现HIL仿真（底层控制算法开发）的核心部件有：飞控、遥控器、遥控器接收机、数据线。具体的组装方式可见下图。



**注意：** 其中，LED灯非必要，可忽略。

硬件链接完成后，可按照实验：

[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\2.FCUIntro](#)，来将硬件环境配置完成并实现初步的HIL仿真。

**注意：** 遥控器配置可见实验：

[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\2.RCIntro](#)。或直接按照自己遥控器的使用说明书实现与飞控的连接。

## 3.3 Simulink SIL模型修改

由于HIL仿真中需要将Simulink中的模型文件通过代码自动生成的方式形成固件，从而，上传到飞控中，因此，需要将Simulink进行修改使其满足上述需求，具体修改内容有如下几个方面。

### 3.3.1 模型属性设置

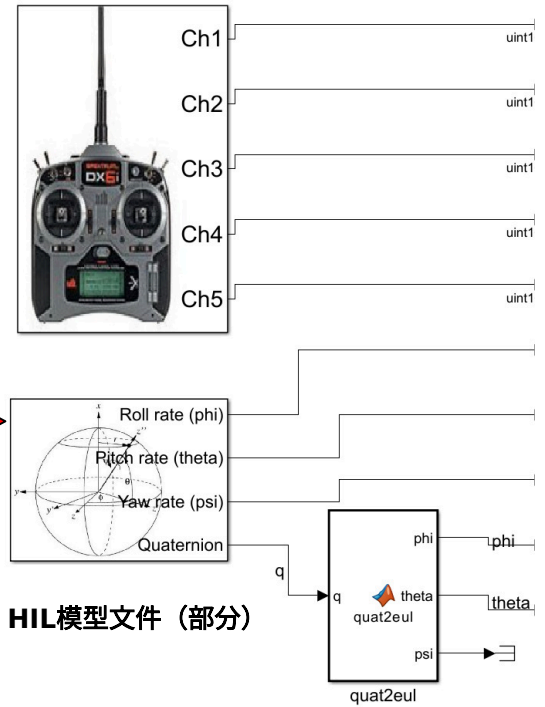
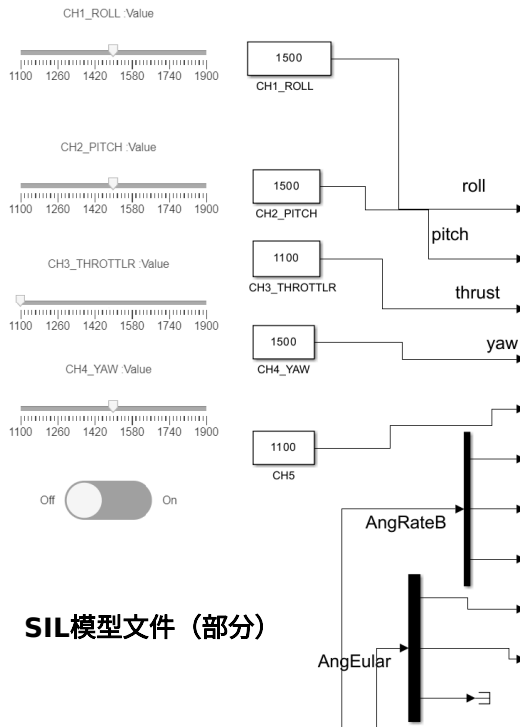
设置Simulink模型文件的属性，使其满足自动代码生成的要求。这里可查看例程中的文件：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp3\\_BlankTemp.slx](#)

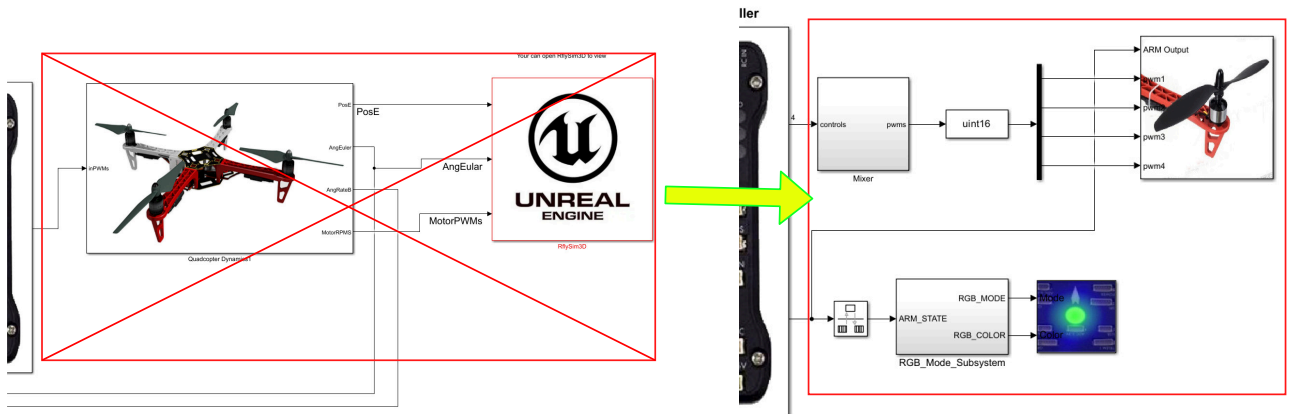
。该文件中已经配置完成了模型属性，可将SIL模型完全复制到该文件中。

### 3.3.2 输入输出设置

首先，修改输入部分，修改遥控器、反馈信息的输入，如下图所示。图中新增的几个模块为RflySim工具链安装完成之后在Simulink中内置的PSP工具箱中的模块，



然后，修改控制器的输出部分，这里我们是删除了在 2.1 模型定义与分析 中搭建的运动模型，这个运动模型并不是说在HIL的时候就不存在了，而是我们直接使用CopterSim中的模型作为我们控制器的使用反馈到PX4中。如下图所示。



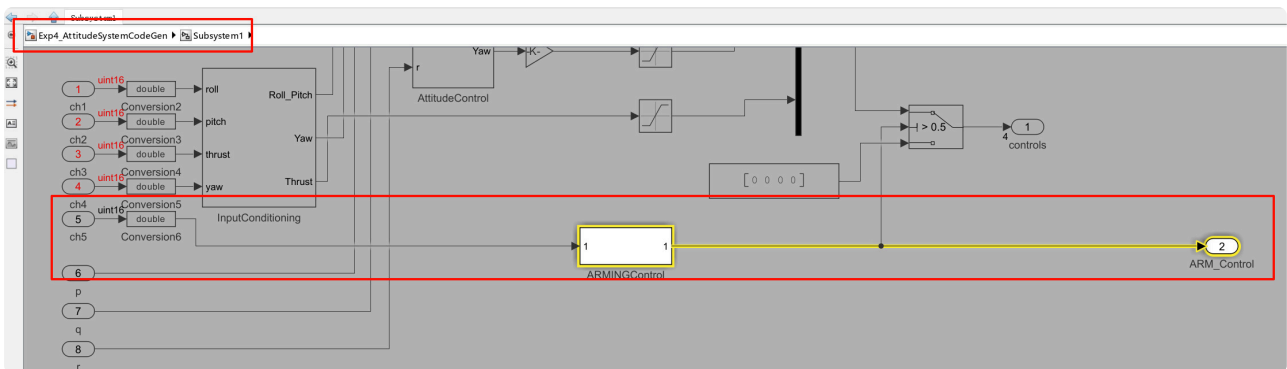
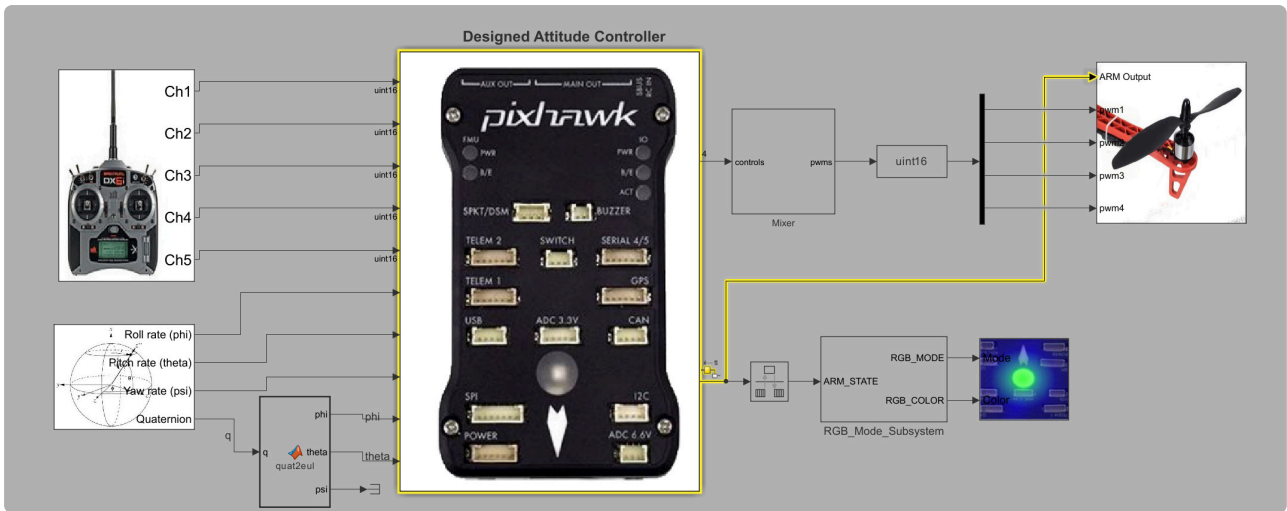
。修改之后的模块是飞控I/O口的PWM接收模块以及LED灯的输出模块。（图中的Mixer还是之前搭建控制器中的控制分配模块）。修改之后的模型文件可见：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp4\\_AttitudeSystemCodeGen.slx](#)

，模型中使用的一些模块的详细介绍可见：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\API.pdf](#)

至此，对于HIL模型的搭建，已经完成。但是，我们需要确认一下的是飞机的解锁方式，可以看到左侧遥控器输入时从CH1~CH5，这个其实符合我们在前期的需求分析部分的定义，CH5定义为解锁。如下图所示，查看电机模块的解锁指令输入可看出最终是链接到了CH5。



### 3.3.3 模型代码生成并烧录飞控

接下来需要将模型进行代码自动生成，并上传到飞控中，完成HIL仿真。具体实验操作步骤可见：

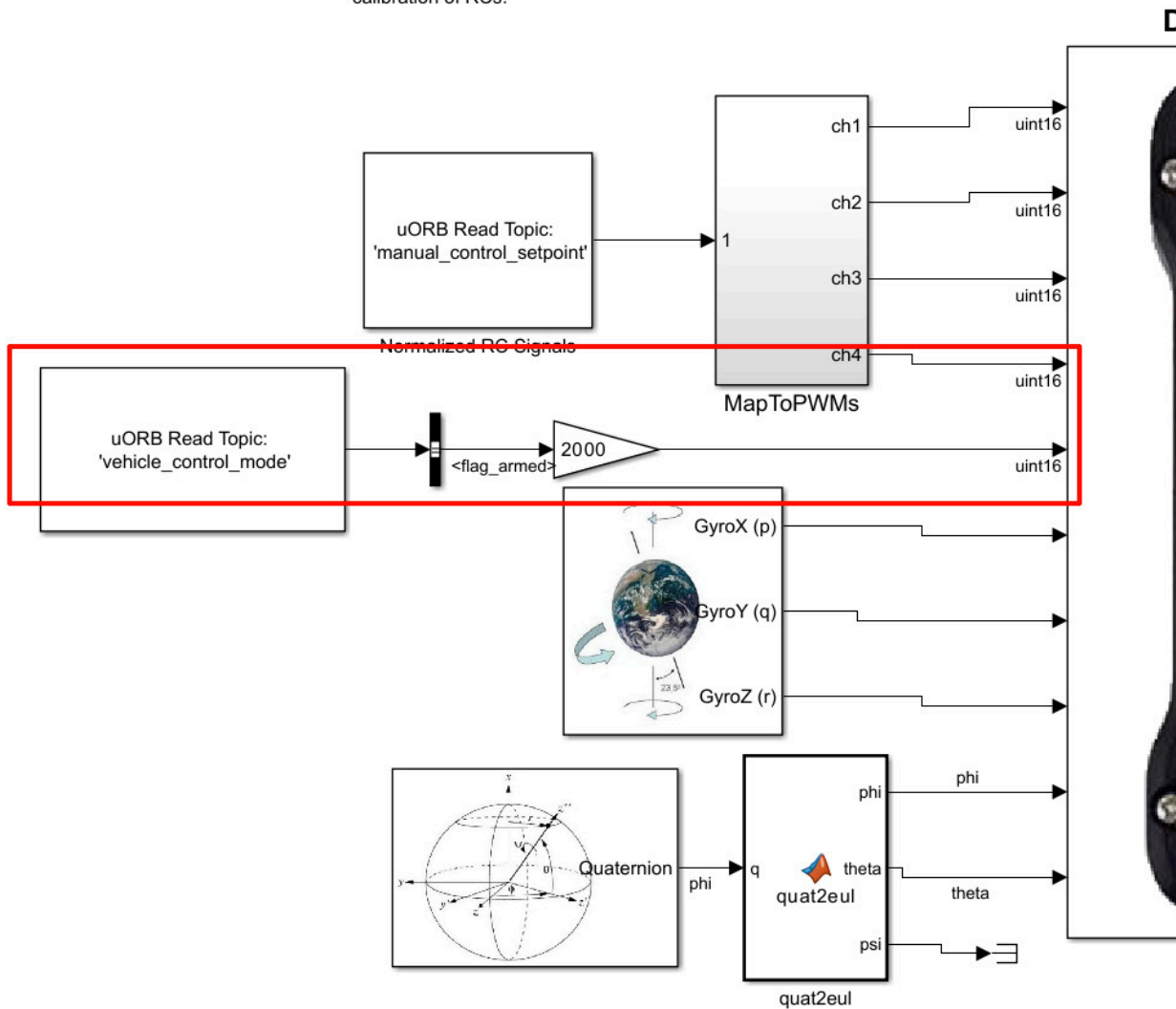
[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Readme.pdf](#)

。另外，对于部分需求，可能需要通过遥控器的“内八”进行解锁，那么可以参考例程：

[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\FLY\\_X450\Exp5\\_AttitudeSystemCodeGenRealFlight\\_all.slx](#)

。通过定义载具的 `vehicle_control_mode` (PX4的不同版本名称可能不同)来使能解锁。

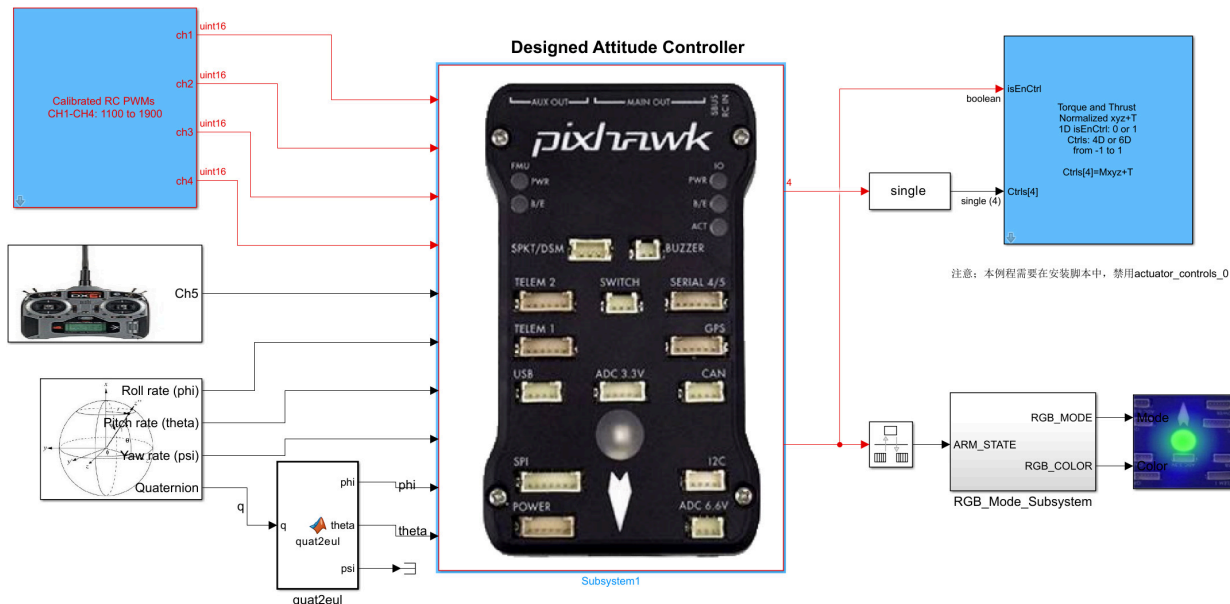
this is more reliable and convenient because you don't need to worry about calibration of RCs.



除此之外，可看下列程文件：

[RflySim安装目录]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp6\_AttitudeSystemCodeGen3.slx

o



注意：本例程需要在安装脚本中，禁用actuator\_controls\_0

该程序中，控制器的最终输出到了 `TorqueThrustCtrls` 模块，该模块是直接发送力和力矩（归一化后的）控制量 `actuator_controls_0` 消息（或PX4 v1.14版本中的 `vehicle_torque/thrust_setpoint`），并经过PX4自带的Mixer混控器计算电机转速后，驱动给CopterSim的inPWMs输入，控制无人机的运动。真机实飞时，则直接驱动飞机进行飞行。相较于前面的I/O口的PWM接收模块为较上层的指令。该模块的优势就是模型文件中不再需要有Mixer（控制分配器），而是直接使用PX4中的控制分配器。

**注意：** 使用时，需要在安装脚本中，禁用`actuator_controls_0`。可参考例程：  
[\[RflySim安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\16.CtrlsSingalsAPI](#)。

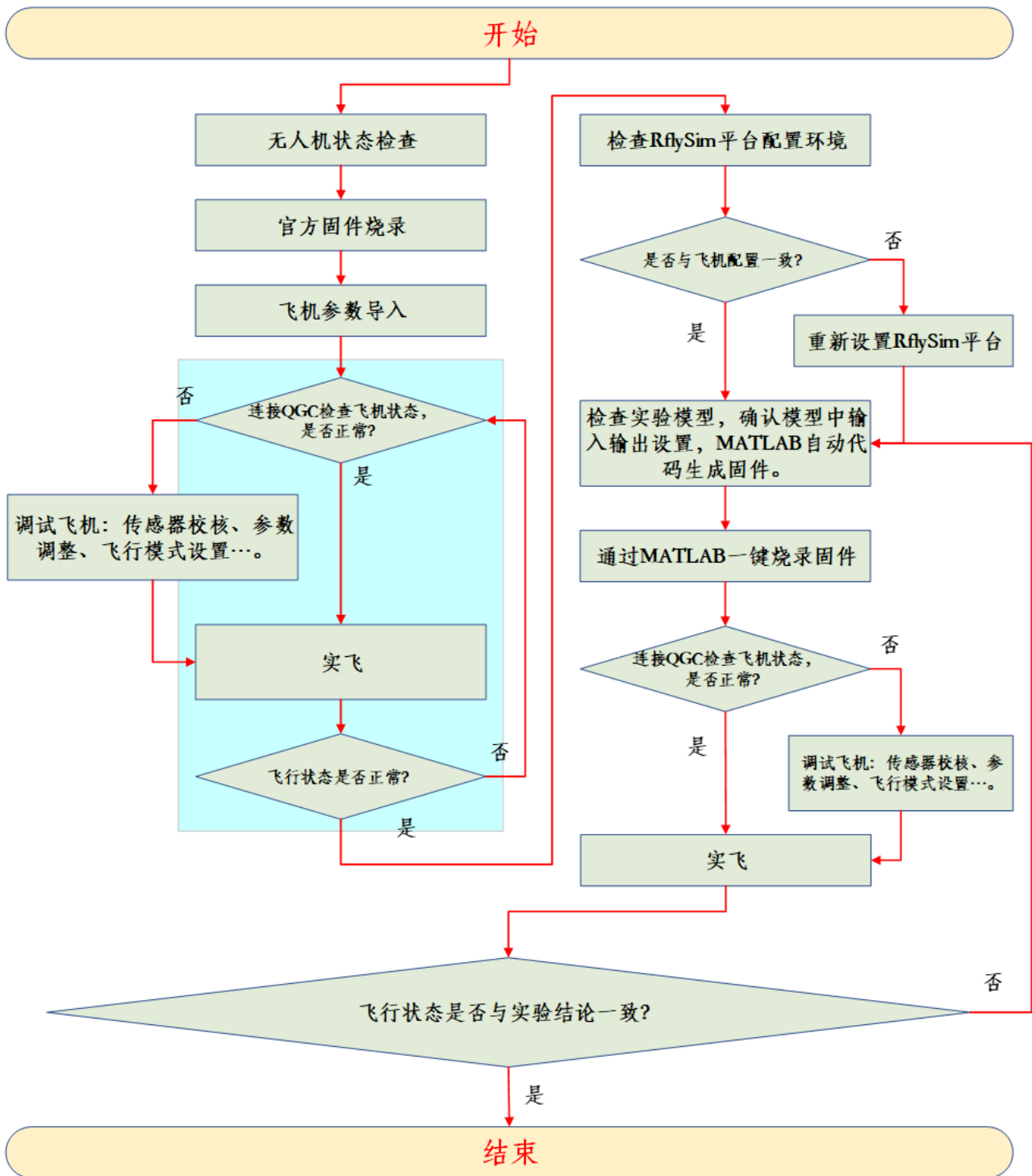
## 4. 第三阶段：台架与实飞实验

实飞实验是控制算法开发的最终验证环节，通过在真实飞行环境中测试控制器性能，确保算法能够满足实际应用需求。关于四旋翼（或其他）无人机的详细组装流程可见：

[\[RflySim安装目录\]\RflySimAPIs\1.RflySimIntro\1.BasicExps\e10.DroneAssTutorial](#)

### 4.1 无人机的设置

关于实飞的流程上，我们推荐按照如下图所示进行，RflySim中在HIL和实飞过程中，烧录的飞控中的固件是可以完成相同的，



首先，我们要确保组装的无人机在硬件和基础官方固件是完好且能正常飞行的，为后续的算法验证打下基础。首先检查无人机的物理状态（电机、桨叶、接线等）是否完好。给无人机飞控板刷入官方的基础PX4固件，并导入该机型对应的基础参数。连接 QGC（QGroundControl 地面站软件）检查飞机状态。进行传感器校准（如指南针、加速度计）、参数调整、飞行模式设置等，然后再次连接 QGC 检查，直到状态正常。实飞后，判断飞行状态是否正常？若不正常，说明硬件或参数还有问题，返回 QGC 检查环节继续调试。若正常，说明无人机的“肉体”和“基础灵魂”没问题了，可以进入下一阶段。

然后，当基础硬件验证通过后，开始引入自己设计的控制算法，即使用在

**3. 第二阶段：硬件在环仿真（HIL）**中生成的固件。检查 RflySim 的配置与自己无人机中分飞控硬件是否完全匹配（与 **3.1 RflySim环境配置确认** 相同）。检查 MATLAB 中的实验模型（即你自己设计的控制算法），确认输入输出接口设置无误后，利用 MATLAB 的代码生成工具，直接将 Simulink 模型生成为可以直接在飞控上运行的固件。将刚刚生成的包含了你自定义算法的固件刷入真实的无人机飞控中（与 **3.3 模型代码生成并烧录飞控** 步骤相同）。

最后，刷入自定义算法后，需要再次进行安全检查和实飞验证。刷入新固件后，再次连接 QGC 检查状态。同样需要重新调试飞机（校准传感器等），直到系统正常。在使用自己的算法进行第二次实飞。

最终结论验证，观察实飞情况，判断飞行状态是否与实验结论（仿真预期）一致？若不一致，说明你的算法在真实物理世界中表现不佳（可能是模型有偏差，或者参数没调好）。此时流程会打回原形，顺着最右侧的红线回到 **检查实验模型...** 这一步，你需要修改 MATLAB 里的算法模型，重新生成代码，重新烧录，再次试飞。若一致，说明你设计的算法在真实飞机上运行完美，达到了预期效果，流程结束。

## 4.2 台架实验

本步骤非开发过程中的必选项，若有条件支持，可进行本步骤。若无，可直接跳到下一步进行实飞实验。

实飞实验前，我们可以将无人机安装在实验台架上进行受约束飞行测试，这种方式能够在保证安全的前提下验证控制器在真实物理环境中的性能表现。

### 4.2.1 台架实验的目的与优势

台架实验作为从仿真到实飞的重要过渡环节，具有以下重要意义：

**核心价值：**

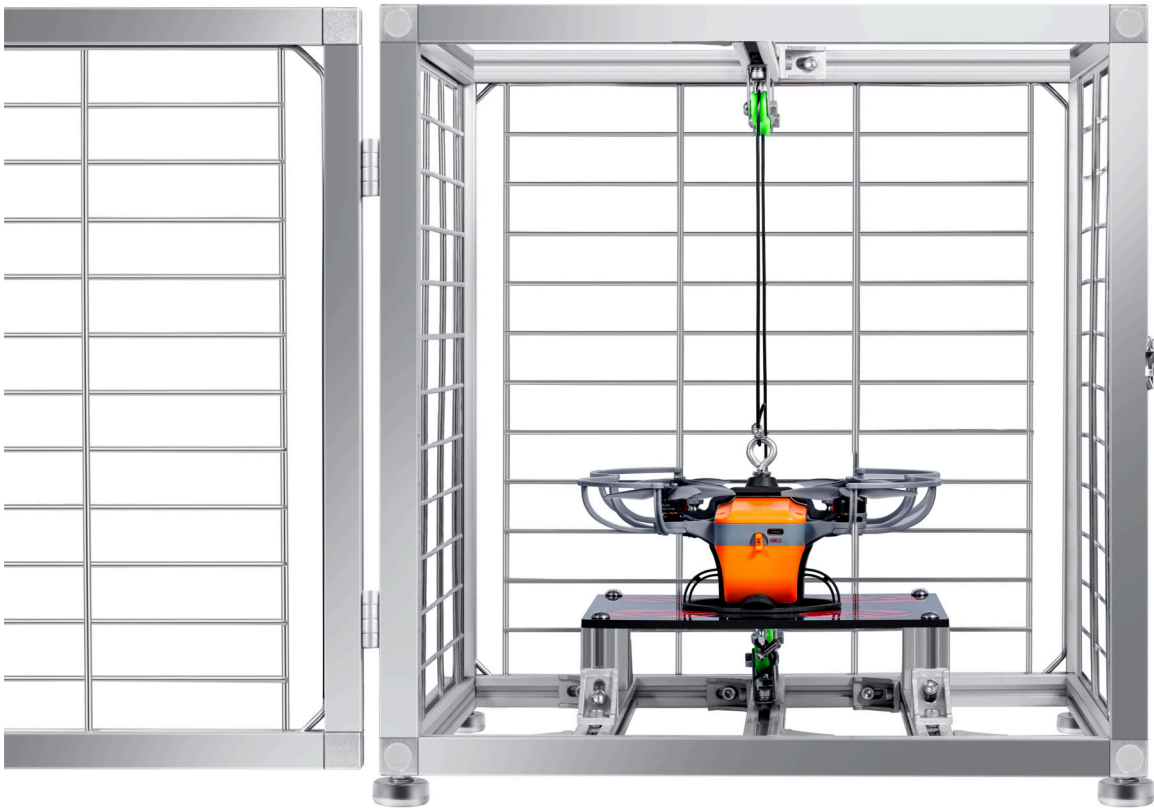
- **安全性保障：** 无人机被物理约束，避免失控坠毁风险
- **性能验证：** 在真实重力场和空气动力学环境下测试控制效果
- **参数调优：** 基于实际响应特性精细调节控制器参数
- **故障排查：** 提前发现硬件兼容性问题 and 潜在故障点

### 4.2.2 实验台架搭建

设备	用途	注意事项
六自由度台架	提供XYZ三轴平移和RPY三轴旋转约束	选择承重能力大于无人机重量2倍的台架

设备	用途	注意事项
安全防护网	防止螺旋桨意外伤人	使用航空级尼龙网，孔径 $\leq 5\text{mm}$
数据采集系统	记录飞行数据用于后续分析	推荐使用PX4内置日志系统配合QGroundControl
应急断电装置	紧急情况下快速切断动力	设置物理开关，响应时间 $< 0.1\text{s}$
防护眼镜	操作人员安全防护	必须佩戴，防止碎片飞溅

如下图所示，为飞思实验室针对自研的飞思J150四旋翼无人机的一款桌面级的台架，安装简单，体积小，可快速对控制算法进行验证。



另外，也可以直接通过主流电商平台进行购买相关台架或自行制作。更多台架测试方式的详细说明可见 [7. 附录->7.2 台架实验测试流程及指标评估](#)

## 4.3 实飞实验

### 4.3.1 实飞前准备

1. 飞行前准备与场地评估 (“人-机-环” 确认), 实飞前的核心是排除外部干扰, 确保飞机处于最佳待命状态。场地与环境阈值:
  - 室外: 最小 30m×30m 无遮挡硬质地面, 风速 < 6m/s。确保 GPS 水平精度 < 1.5m, 无强高压线等电磁干扰。
  - 室内: 依赖视觉 SLAM 或 UWB 定位, 需保证照明均匀且无强反射地面。必须配置柔性安全防护网。
2. 起飞前系统自检 (Pre-flight Check):
  - 硬件与动力: 机架无松动, 螺旋桨无裂纹, 电池满电且压差正常。
  - 航电与通信: 完成加速度计、指南针校准。遥控器与地面站数传链路稳定 (信噪比 > 20dB)。
  - 安全与合规: 确认空域合法性, 设定主备降落区, 明确紧急切断动力 (Kill-switch) 的触发条件与责任人。
3. 渐进式飞行测试 (从地到空的安全爬升): 严禁首次测试直接切入全自主模式, 必须按阶段验证算法的稳定性。
  - 阶段一: 地面怠速与响应验证, 不上桨或切断电机输出, 推拉摇杆, 在地面站观察控制面或电机 PWM 指令响应是否正确, 验证遥控器映射无误。
  - 阶段二: 低空悬停与紧急介入 (1-3米), 起飞至脱离地效区 (1-3米) 悬停。重点评估姿态稳定性 (无明显抖动或低频振荡)。测试紧急降落和手动接管功能, 确保在算法跑飞时能随时救机。
  - 阶段三: 基础机动与控制律验证, 执行方形/圆形航线, 测试横滚、俯仰的跟踪精度。急推急拉油门, 验证高度环 (Altitude Hold) 的定高能力和抗扰动表现。
  - 阶段四: 高级功能与边界测试, 测试自动航点飞行、自动返航 (RTL)、防撞避障等高级任务逻辑。验证极限工况下的表现, 如抗风测试、最大起飞重量测试。

### 4.3.2 核心飞行数据分析与调优 (闭环迭代)

实飞不是目的, 获取数据并优化 Simulink/C++ 控制模型才是核心。

- \*.ulg 日志解析与指标提取, 重点比对 期望姿态 (Setpoint) 与 实际估计姿态 (Estimated) 的跟随延迟与超调量。检查机体原始振动数据 (Raw Acceleration), 评估是否需要调整物理减震或软件低通滤波器 (Low-pass Filter) 截止频率。
- PID 参数实测调优, 基于日志表现, 调节内外环 PID 参数。若高频抖动则降低 D 值或调整滤波器; 若响应迟缓则适当提高 P 值。系统健康度监控: 分析 CPU 负荷、内存占用及各传感器数据的丢失率, 确保自研代码未导致飞控资源过载。

### 4.3.3 实飞总结与成熟度评估报告

将测试结果量化，为下一轮研发提供明确方向。

- 数据汇总：统计有效飞行架次、任务成功率、定点悬停精度（如：误差  $< 0.2\text{m}$ ）等关键性能指标 (KPI)。
- 算法成熟度 (TRL) 评估：客观评价自动生成的代码在真实物理世界中的表现，是否达到商业化部署或演示标准。
- 迭代建议清单：明确下一步是需要修改上层控制模型、优化状态估计算法，还是直接进行硬件升级。

## 6. 进阶开发技巧

### 6.1 控制器日志记录及分析

RflySim工具链提供多种日志记录的功能，并且具有多个进行日志分析的实验。

- SD卡日志记录，实验：

```
[RflySim安装目录]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\5.Log-Write-Read
```

，可将控制器中的部分数据写入飞控的SD卡中，记录飞行过程中的个别数据。

- uORB消息日志记录，实验：

```
[RflySim安装目录]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\7.uORB-Create
```

，可快速创建PX4中的uORB消息，自定义的消息可以用于数据传递，也可以用于飞行过程中的部分日志的记录。

- 日志分析，对于日志分析，可参考实验：

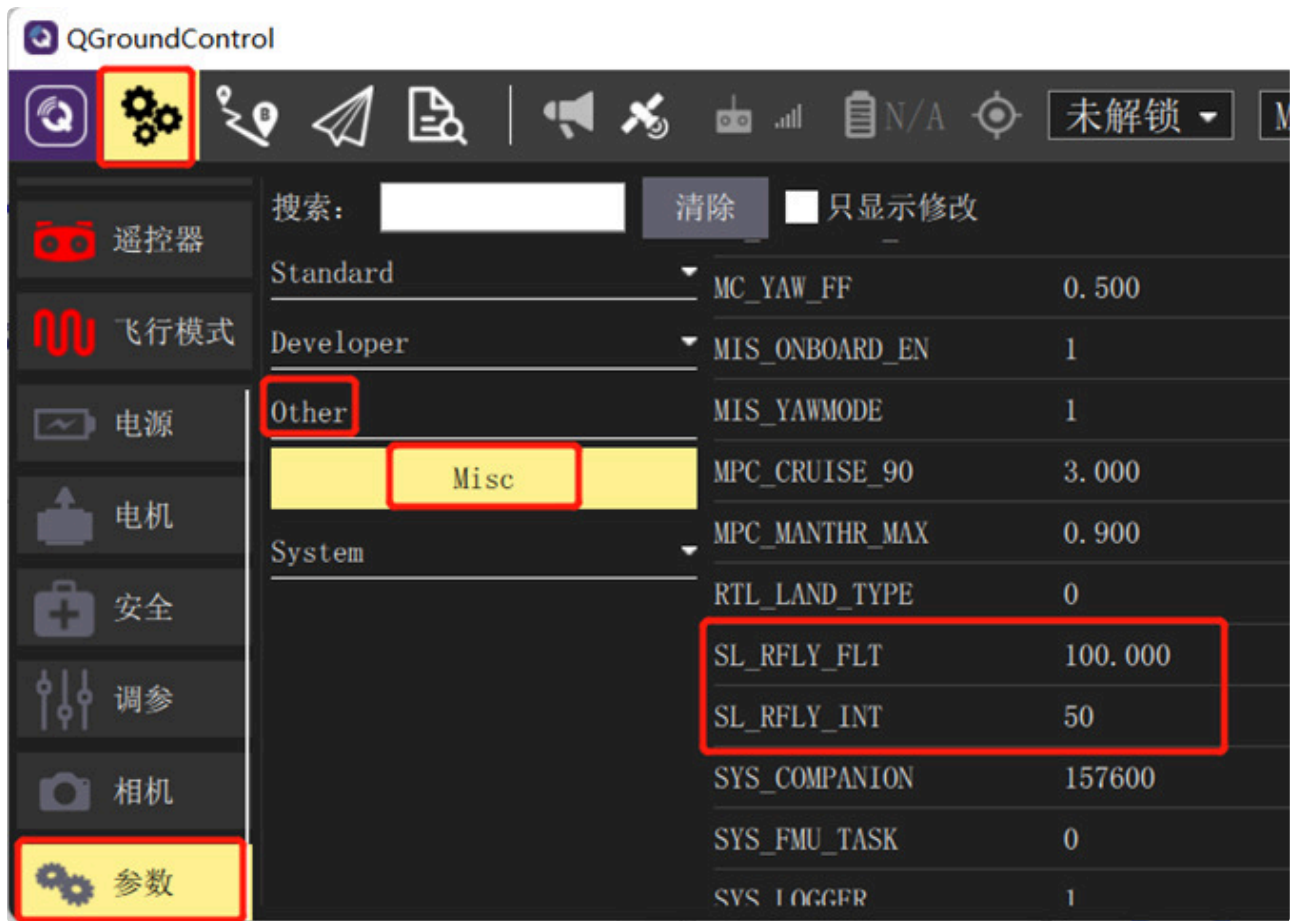
```
[RflySim安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e10.Log-GetAnalysis
```

。该系列文件夹中几乎包含了RflySim工具链用于日志分析的所有小实验，包含有通过PX4官网、Python、MATLAB、PlotJuggler等工具的日志分析，各有优缺点，可针对性选择。

### 6.2 自定义控制器参数的导入QGC参数界面

在进行硬件在环仿真和真机实验时，常常需要在QGC地面站中观察飞行状态，并对自定义控制器的参数进行实时调整，以使得飞机达到最佳的控制效果。常规的调试方式为在MATLAB/Simulink中进行相关参数的修改，完成后，在进行自动代码生成导入的飞控中进行验证，整个过程较为繁琐，而实

验 [RflySim安装目录]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\10.QGC-Param-Tune 提供了一种便捷的方式，可以将自定义的控制器参数，导入的QGC地面站的参数界面中，并且可以进行调参。



## 6.3 PX4官方控制器与自定义控制器自由切换

自定义控制器在进行飞行的时候，往往是存在较大的飞行风险，极有可能发生意外，因此，RflySim工具链针对性的开发了可快速进行对PX4官方控制器与自定义控制器的切换的功能。若在飞行过程中，出现炸机或者马上炸机，可快速切换到PX4官方控制器上，以此来避免因炸机造成的额外损失，具体实验可见：

[RflySim安装目录]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\21.SwitchPX4Simulink

## 7. 附录

### 7.1 PX4 文件目录详细说明（以PX4 1.15.4为例）

#### 1. `src/` - 核心源代码目录

这是项目的核心目录，包含所有飞控相关的源代码。

子目录	描述
<code>modules/</code>	飞控核心模块，包括控制器、导航器、状态估计等主要功能模块
<code>drivers/</code>	硬件驱动程序，支持各种传感器（IMU、气压计、GPS、磁力计等）
<code>lib/</code>	公共库，包含数学库、控制库、滤波器等通用功能
<code>systemcmds/</code>	系统命令行工具，如参数设置、性能监测、诊断工具等
<code>examples/</code>	示例程序，帮助开发者学习如何编写 PX4 模块
<code>include/</code>	公共头文件
<code>templates/</code>	模块模板

#### `src/modules/` 中的关键模块

模块名称	功能描述
<code>commander</code>	<b>指挥官模块</b> - 飞行状态管理、解锁/上锁控制、故障处理
<code>ekf2</code>	<b>扩展卡尔曼滤波器</b> - 姿态和位置状态估计
<code>navigator</code>	<b>导航器</b> - 任务执行、航点导航、RTL(返航)
<code>mc_pos_control</code>	<b>多旋翼位置控制</b>
<code>mc_att_control</code>	<b>多旋翼姿态控制</b>
<code>mc_rate_control</code>	<b>多旋翼角速度控制</b>

模块名称	功能描述
fw_pos_control	固定翼位置控制
fw_att_control	固定翼姿态控制
vtol_att_control	垂直起降姿态控制
flight_mode_manager	飞行模式管理
land_detector	着陆检测
sensors	传感器数据处理和融合
mavlink	MAVLink 通信协议 - 与地面站通信
logger	日志记录器
control_allocator	控制分配 - 将期望的力和力矩分配到各个执行器
simulation	仿真支持模块
airspeed_selector	空速选择器
battery_status	电池状态管理
gimbal	云台控制
gyro_calibration	陀螺仪校准
manual_control	手动控制输入处理
rc_update	遥控器数据更新
temperature_compensation	温度补偿
px4_simulink_app	Simulink 集成应用 (RflySim 专用)

#### src/drivers/ 中的关键驱动

驱动类别	描述
imu/	惯性测量单元 (加速度计、陀螺仪) - 包含多个型号驱动
barometer/	气压计 - MS5611, BMP280, LPS22HB 等

驱动类别	描述
magnetometer/	磁力计 - HMC5883, IST8310, QMC5883 等
gps/	GPS 模块 - u-blox, MTK 等
distance_sensor/	测距传感器 - 超声波、激光雷达等
optical_flow/	光流传感器 - PX4FLOW 等
pwm_out/	PWM 输出 - 电机/舵机控制
dshot/	DShot 协议输出
rc_input/	遥控器输入
rc/	遥控器协议解析 (SBUS, PPM, DSM 等)
uavcan/	UAVCAN 总线驱动
cyphal/	Cyphal 通信协议
telemetry/	数传模块
lights/	LED 灯光控制
power_monitor/	电源监测
transponder/	应答机 (ADS-B 等)

### src/lib/ 中的关键库

库名称	功能描述
mathlib/	数学库 - 向量、矩阵、四元数运算
matrix/	矩阵库 - 来自 PX4 ECL
controllib/	控制库 - PID、滤波器等
geo/	地理计算库 - 坐标转换、距离计算
pid/	PID 控制器
rate_control/	角速度控制器
motion_planning/	运动规划
collision_prevention/	避障

库名称	功能描述
wind_estimator/	风速估计
world_magnetic_model/	世界磁场模型
tecs/	总能量控制系统 (固定翼)
npfg/	非线性路径跟踪制导
battery/	电池管理
parameters/	参数系统
events/	事件系统
perf/	性能计数器

### src/systemcmds/ 系统命令

命令	功能描述
param	参数读写命令
reboot	重启命令
top	系统资源监控
perf	性能统计
dmesg	系统日志
actuator_test	执行器测试
led_control	LED 控制
tune_control	蜂鸣器控制
tests	系统测试集
ver	版本信息

### src/examples/ 示例程序

示例	描述
px4_simple_app	最简单的 PX4 应用示例
hello	Hello World 示例

示例	描述
<code>work_item</code>	工作队列示例
<code>px4_mavlink_debug</code>	MAVLink 调试示例
<code>fake_gps</code>	虚拟 GPS 模块
<code>fake_imu</code>	虚拟 IMU 模块

## 2. boards/ - 飞控板配置

包含各种飞控硬件板的配置文件，支持超过 27 个厂商/平台：

厂商	描述
<code>px4/</code>	原生 PX4 飞控板 (Pixhawk 系列)
<code>holystone/</code>	Holystone 飞控 (Pixhawk 4/6, Durandal, K akute 等)
<code>cubepilot/</code>	CubePilot 飞控 (Cube Orange/Yellow 等)
<code>cuav/</code>	CUAV 飞控板 (V5+, V5 Nano, Pixhawk V6 X)
<code>matek/</code>	Matek 飞控板
<code>ark/</code>	ARK Electronics 飞控
<code>nxp/</code>	NXP 飞控板
<code>bitcraze/</code>	Crazyflie 微型无人机
<code>raspberrypi/</code>	树莓派飞控
<code>mro/</code>	mRobotics 飞控
<code>modalai/</code>	ModalAI 飞控
<code>freefly/</code>	Freefly 飞控
<code>hkust/</code>	香港科技大学定制飞控
<code>beaglebone/</code>	BeagleBone 飞控

厂商	描述
更多...	

每个飞控板目录包含：

- 引脚定义
- 传感器配置
- 默认参数
- 启动脚本
- CMake 配置

### 3. `platforms/` - 平台抽象层

提供不同操作系统/平台的抽象支持：

平台	描述
<code>nuttx/</code>	<b>NuttX RTOS</b> - 嵌入式实时操作系统，用于真正的飞控硬件
<code>posix/</code>	<b>POSIX 兼容系统</b> - 用于 Linux/macOS 上的 SITL 仿真
<code>qurt/</code>	<b>Qualcomm 平台</b> - 用于高通骁龙飞控
<code>ros2/</code>	<b>ROS 2 集成</b> - 与 ROS 2 系统的接口
<code>common/</code>	<b>公共平台代码</b> - 跨平台共享的代码

### 4. `msg/` - uORB 消息定义

定义了所有 **uORB (Micro Object Request Broker)** 消息类型。uORB 是 PX4 的发布-订阅消息系统，用于模块间高效通信。

包含 **218 个消息定义文件**，主要类别包括：

## 飞行器状态消息

消息	描述
VehicleAttitude.msg	飞行器姿态 (四元数)
VehicleLocalPosition.msg	本地位置 (NED 坐标)
VehicleGlobalPosition.msg	全球位置 (经纬度)
VehicleStatus.msg	飞行器状态
VehicleControlMode.msg	飞行控制模式
VehicleLandDetected.msg	着陆检测状态

## 传感器消息

消息	描述
SensorCombined.msg	传感器融合数据
SensorAccel.msg	加速度计数据
SensorGyro.msg	陀螺仪数据
SensorMag.msg	磁力计数据
SensorBaro.msg	气压计数据
SensorGps.msg	GPS 数据

## 控制消息

消息	描述
VehicleAttitudeSetpoint.msg	姿态设定值
VehicleRatesSetpoint.msg	角速度设定值
TrajectorySetpoint.msg	轨迹设定值
ActuatorMotors.msg	电机输出
ActuatorServos.msg	舵机输出

## ● RflySim 专用消息

消息	描述
RflyCtrl.msg	RflySim 控制消息
RflyPX4.msg	RflySim PX4 通信消息
RflyExt.msg	RflySim 扩展消息
RflyInsils.msg	RflySim INS/ILS 消息

## ● 5. ROMFS/ - 只读文件系统

包含飞控启动脚本和默认参数配置：

```
1 | ROMFS/  
2 | └─ px4fmu_common/           # 通用 FMU 配置  
3 |   └─ init.d/                # 机架初始化脚本  
4 |     └─ rc.autostart         # 自动启动脚本  
5 |     └─ 4001_quad_x         # 四旋翼 X 型配置  
6 |     └─ 4010_dji_f330      # DJI F330 配置  
7 |     └─ ...                  # 更多机架配置  
8 |   └─ init.d-posix/         # POSIX 仿真环境脚本  
9 | └─ px4fmu_test/           # 测试配置  
10| └─ cannoder/              # CAN 节点配置
```

机架配置脚本定义了：

- 电机布局和混控器
- 默认参数值
- PWM 输出配置
- 传感器配置

## ● 6. Tools/ - 开发工具

包含各种开发、测试和调试工具：

工具/目录	描述
<code>simulation/</code>	<b>仿真工具</b> - Gazebo、jMAVSim、AirSim 等仿真器配置和模型
<code>HIL/</code>	<b>硬件在环(HIL)</b> 测试工具
<code>ecl_ekf/</code>	<b>EKF 分析工具</b> - 状态估计分析和调试
<code>px_uploader.py</code>	<b>固件上传工具</b> - 通过 USB 上传固件
<code>mavlink_shell.py</code>	<b>MAVLink Shell</b> - 远程命令行调试
<code>mavlink_uflog_streaming.py</code>	<b>日志流传输</b>
<code>setup/</code>	<b>开发环境配置脚本</b>
<code>serial/</code>	<b>串口配置工具</b>
<code>Matlab/</code>	<b>MATLAB 相关工具</b>
<code>px4airframes/</code>	<b>机架处理工具</b>
<code>px4events/</code>	<b>事件处理工具</b>
<code>px4moduledoc/</code>	<b>模块文档生成</b>
<code>uorb_graph/</code>	<b>uORB 关系图生成</b>
<code>kconfig/</code>	<b>Kconfig 配置工具</b>
<code>sitl_multiple_run_rfly.sh</code>	<b>RflySim 多机仿真脚本</b>

## 7. `cmake/` - 构建系统配置

包含 CMake 构建系统的配置文件:

文件	描述
<code>px4_add_module.cmake</code>	添加 PX4 模块的宏定义
<code>px4_add_library.cmake</code>	添加库的宏定义
<code>px4_config.cmake</code>	平台配置
<code>kconfig.cmake</code>	Kconfig 配置系统集成

文件	描述
<code>px4_git.cmake</code>	Git 版本信息获取
<code>coverage.cmake</code>	代码覆盖率配置
<code>sanitizers.cmake</code>	代码检查器配置

## 8. `test/` 和 `integrationtests/` - 测试

- `test/` - 单元测试
  - 数学库测试
  - 控制器测试
  - 参数系统测试
- `integrationtests/` - 集成测试
  - SITL 仿真测试
  - 飞行模式测试
  - 任务执行测试

## 9. `launch/` - 启动配置

ROS/ROS2 启动文件，用于与 ROS 生态系统集成。

## 10. 其他重要文件和目录

文件/目录	描述
<code>CMakeLists.txt</code>	主 CMake 配置文件
<code>Makefile</code>	主构建入口，定义构建目标
<code>Kconfig</code>	内核配置文件 (类似 Linux)
<code>README.md</code>	项目说明文档
<code>.gitmodules</code>	Git 子模块配置
<code>.github/</code>	GitHub 工作流和配置

文件/目录	描述
<code>.vscode/</code>	VS Code 配置
<code>Documentation/</code>	文档目录
<code>posix-configs/</code>	POSIX 仿真配置
<code>validation/</code>	验证脚本
<code>BkFile/</code>	<b>备份文件目录</b> (RflySim 专用)
<code>PX4VerRfly.txt</code>	<b>RflySim 版本标识</b>
<code>PX4VerRfly.mat</code>	<b>RflySim 版本 MAT 文件</b>

## 11. RflySim 相关修改

从项目文件中可以看出，这个版本已经针对 **RflySim** 仿真平台进行了定制修改：

### 1. 新增消息定义

- `msg/RflyCtrl.msg` - RflySim 控制消息
- `msg/RflyPX4.msg` - RflySim PX4 通信消息
- `msg/RflyExt.msg` - RflySim 扩展消息
- `msg/RflyInsils.msg` - RflySim INS/ILS 消息

### 2. 新增驱动和模块

- `src/drivers/drv_rfly.h` - RflySim 驱动头文件
- `src/modules/p4_simulink_app/` - Simulink 集成模块

### 3. 新增工具

- `Tools/sitl_multiple_run_rfly.sh` - RflySim 多机仿真脚本

### 4. 版本标识

- `BkFile/` - 备份文件目录
- `PX4VerRfly.txt` - 版本号
- `PX4VerRfly.mat` - MATLAB 版本文件

## 7.2 台架实验测试流程及指标评估

### 7.2.1 台架实验测试流程

#### 1. 预实验检查清单

- 1 □ 无人机硬件完整性检查
- 2 □ 螺旋桨安装牢固，无裂纹
- 3 □ 电池电量充足 (>80%)
- 4 □ 飞控固件版本确认
- 5 □ 传感器校准完成
- 6
- 7 □ 台架系统安全检查
- 8 □ 约束系统紧固可靠
- 9 □ 防护设施完好
- 10 □ 应急装置功能正常
- 11 □ 操作区域清场完毕
- 12
- 13 □ 通信链路测试
- 14 □ 遥控器信号强度 > -70dBm
- 15 □ 数传链路延迟 < 50ms
- 16 □ QGroundControl连接稳定
- 17 ```
- 18

## 19 2. 分阶段测试方案

### 20 \*\*阶段一：静态特性测试\*\*

- 21 - 测试项目：电机响应特性、PWM输出精度
- 22 - 操作方法：固定姿态下分别测试各通道阶跃响应
- 23 - 预期结果：响应时间<50ms，超调量<10%
- 24
- 25

### 26 \*\*阶段二：单自由度动态测试\*\*

- 27 - 测试项目：滚转/俯仰通道独立控制性能
- 28 - 操作方法：释放对应方向约束，其他方向保持锁定
- 29 - 评价指标：
  - 30 - 上升时间： $\leq 1.5s$
  - 31 - 稳态误差： $\leq 2^\circ$
  - 32 - 超调量： $\leq 15\%$
- 33

### 34 \*\*阶段三：耦合特性测试\*\*

- 35 - 测试项目：多通道协同控制能力
- 36 - 操作方法：执行预设的复合机动指令
- 37 - 关键观测：通道间耦合效应是否在可接受范围内
- 38

## 39 3. 数据采集与分析

### 40 \*\*核心数据项：\*\*

41 ```

#### 42 传感器数据：

- 43 - IMU原始数据 (1kHz采样)
- 44 - 姿态估计输出 (400Hz)
- 45 - 电机转速反馈
- 46

#### 47 控制数据：

- 48 - 遥控器输入指令
- 49 - 控制器输出信号
- 50 - 执行器实际响应
- 51
- 52

53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81

性能指标:

- 姿态跟踪误差
- 响应时间常数
- 系统带宽特性
- ...

**\*\*典型测试序列:\*\***

1. **\*\*阶跃响应测试\*\***: 输入 $\pm 20^\circ$ 阶跃信号, 记录系统响应曲线
2. **\*\*频率响应测试\*\***: 扫频信号(0.1-20Hz), 获取系统频率特性
3. **\*\*抗干扰测试\*\***: 施加外部扰动, 测试系统恢复能力
4. **\*\*极限性能测试\*\***: 接近物理极限的操作指令, 验证安全边界

#### 7.2.2 关键性能指标评估

1. 控制品质评价标准

指标	优秀	良好	合格	不合格
<b>**上升时间**</b> (s)	<1.0	1.0-1.5	1.5-2.0	> 2.0
<b>**超调量**</b> (%)	<10	10-15	15-25	> 25
<b>**稳态误差**</b> ( $^\circ$ )	<1	1-2	2-5	> 5
<b>**振荡次数**</b>	0-1	1-2	2-3	> 3
<b>**带宽**</b> (Hz)	>8	5-8	3-5	< 3

2. 稳定性判据

- **\*\*相位裕度\*\***:  $\geq 45^\circ$
- **\*\*幅值裕度\*\***:  $\geq 6\text{dB}$
- **\*\*Nyquist图\*\***: 不包围(-1, j0)点
- **\*\*根轨迹\*\***: 所有极点位于左半平面

---