

姿态控制器SIL_HIL_实飞实验

1. 实验目的

本实验旨在通过完整的姿态控制器开发流程，让学习者掌握从控制器设计、软件在环仿真、硬件在环仿真到实际飞行验证的全过程。具体实验目的包括：

- 掌握姿态控制器设计原理：**学习PID控制器在多旋翼无人机姿态控制中的应用，理解滚转、俯仰、偏航三个通道的控制逻辑和参数整定方法。
- 熟悉软件在环仿真流程：**通过Simulink搭建控制器模型，与RflySim3D仿真环境进行软件在环仿真，验证控制器设计的正确性。
- 掌握硬件在环仿真技术：**学习将Simulink控制器模型编译为PX4飞控可执行代码，通过硬件在环仿真验证控制器在真实硬件环境下的性能。
- 学习转台实验方法：**掌握通过数传和WiFi两种方式连接飞控进行转台实验的技术，在安全环境下验证控制器的实际控制效果。
- 完成实飞验证：**将设计的控制器部署到真实无人机上进行实际飞行测试，验证控制器在真实飞行环境中的性能。
- 了解DShot协议应用：**学习DShot数字电调协议在PX4飞控中的应用，掌握基于DShot模式的控制器开发方法。

通过本实验，学习者将全面掌握无人机姿态控制器的设计、仿真、测试和验证全流程，为后续更复杂的飞行控制算法开发奠定基础。

2. 实验要求

- 软件要求：Windows 10及以上版本；RflySim工具链^[1]，MATLAB2022B以上版本，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.14.3。
- 硬件要求：笔记本/台式电脑1台^[2]，遥控器和遥控器接收机；数据线和杜邦线等^[3]。

3. 实验地址

例程目录：[\[安装目录\]\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps](#)

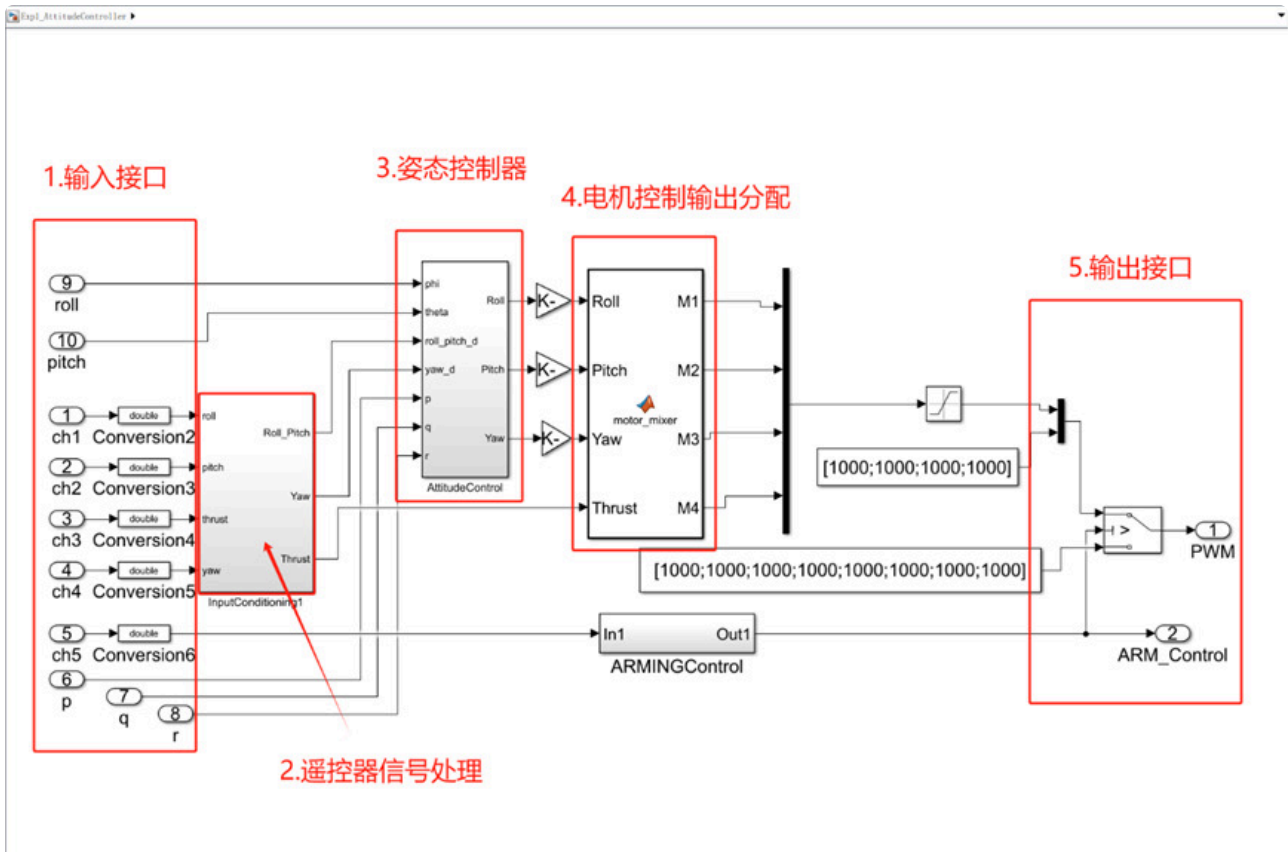
```
1 | FLY_X450 : 飞思老款F 450飞机 HIL及实飞程序
2 | FS-J150 : 飞思J150飞机实飞程序
3 | Exp4_5_Rotarytable : 数传链接下得转台实验程序
4 | control-PID.py : WIFI链接下得转台实验程序
5 | Init_control.m : 控制器初始化参数文件。
6 | Exp1_AttitudeController.slx : 控制模型文件。
7 | Exp2_ControlSystemDemo.slx : 软件在环仿真模型文件。
8 | Exp3_BlankTemp.slx : 代码生成模板文件。
9 | Exp4_AttitudeSystemCodeGen.slx : 硬件在环仿真模型文件(遥控器输入归一化HIL+实飞)。
10 | Exp5_AttitudeSystemCodeGen2.slx : 硬件在环仿真模型文件(遥控器输入归一化, 仅HIL仿真)
11 | Exp6_AttitudeSystemCodeGen3.slx : 硬件在环仿真模型文件(控制信号直出, HIL+实飞)
12 | Init_control.m : 控制器初始化参数文件。
```

4. 实验内容或步骤

4.1 控制器搭建

`Exp1_AttitudeController.slx` 为一个进行搭建完成的控制器模型文件，如下图所示，第 1~5 号输入接口对应了遥控器的五个通道的输入(“ch1” ~ “ch5”);第 6~8号输入接口对应了来自陀螺仪传感器的滚转、俯仰和偏航方向的角速度(“p” “q” 和 “r”);第 9~10 号接口对应了滚转角和俯仰角(“phi” 和 “theta”)。四旋翼飞行器的整个控制器模块的计算过程大体分为五个步骤。

1. 输入接口模块:接收遥控器信号和飞机状态观测信号。
2. 遥控信号处理模块:将遥控器的5个通道信号映射为期望的滚转和俯仰角度。
3. 姿态控制器模块:计算期望输出力和力矩大小，来控制多旋翼飞行器姿态到期望角度。
4. 电机控制输出分配:模块将力和力矩的控制量映射为4个电机的油门控制量(一般是1000~2000之间)。
5. 输出接口模块:将剩余的四维控制量补齐，并映射出PWM 调制信号(一般是1000~2000微秒)，构成八维(Pixhawk自驾仪硬件上有8个PWM 的输出口)的PWM控制信号，



4.2 数字仿真（软件在环仿真）

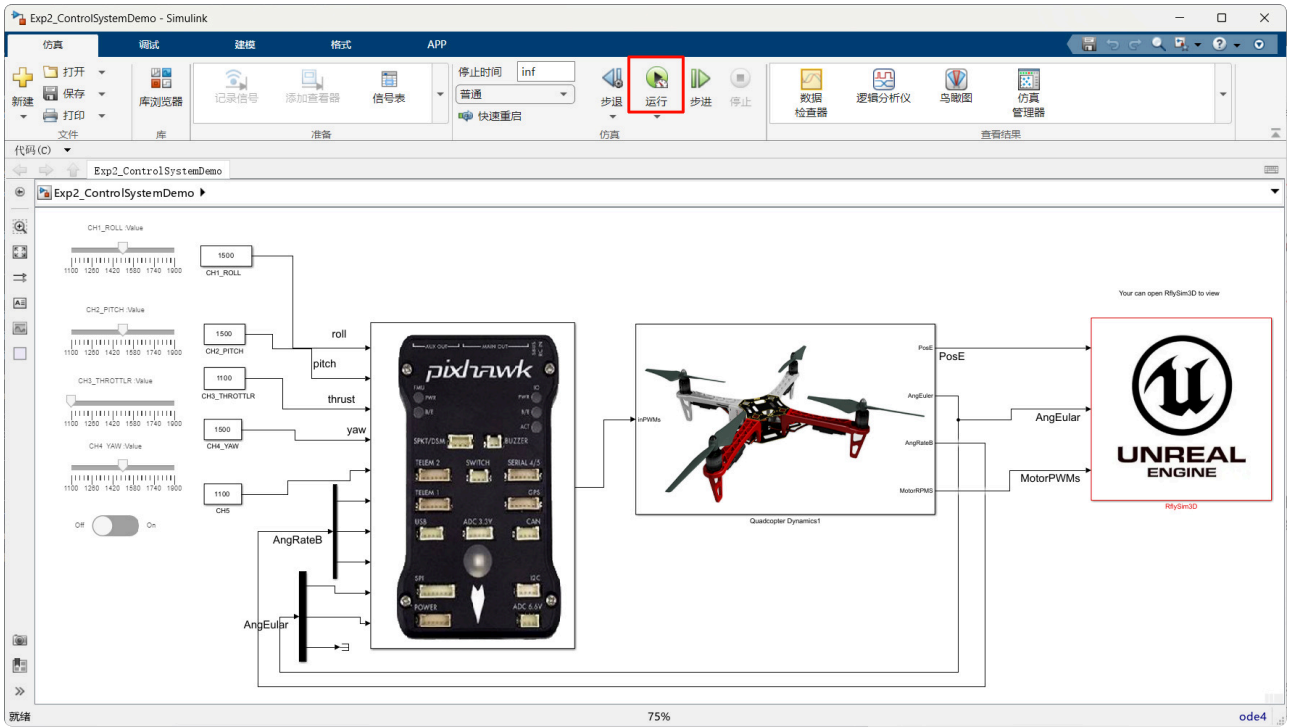
基于 4.1 控制器搭建 中的控制器模型搭建，在 `Exp2_ControlSystemDemo.slx`，如下图所示，该仿真系统包含三个子系统模块：



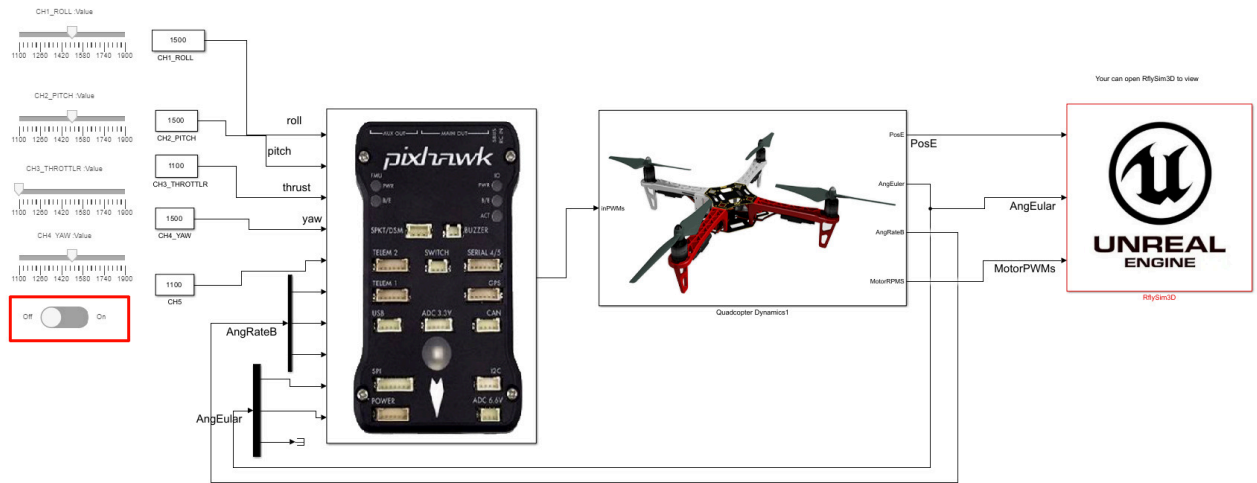
1. Subsystem: “控制器” 子系统。
2. Quadcopter Dynamics: “多旋翼模型” 子系统。
3. RflySim3D: “RflySim3D接口” 子系统。

开始仿真前，先双击 `*\Desktop\RflyTools\RflySim3D.lnk`，打开RflySim3D软件。然后点击 `Exp2_ControlSystemDemo.slx` 中的 运行 按钮，即可看到RflySim3D中的地图自动发生

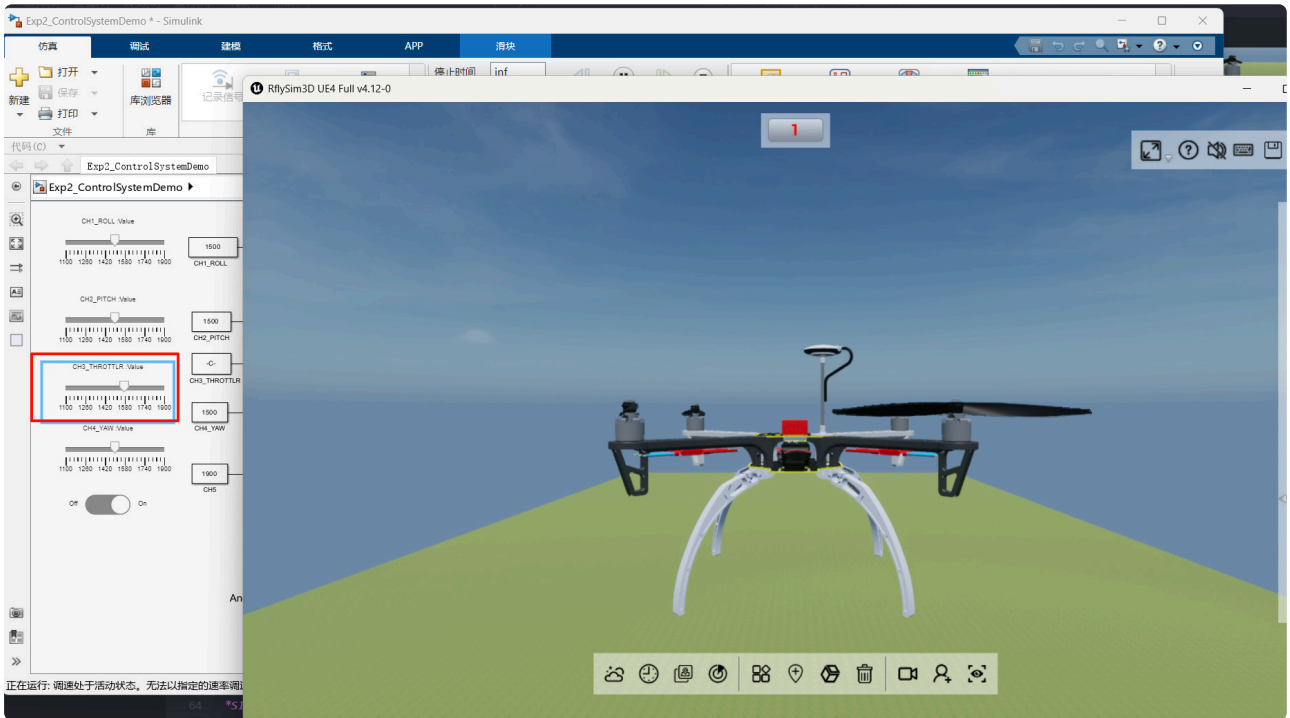
切换，并自动初始化了了飞机。



然后，在 `Exp2_ControlSystemDemo.slx` 程序运行的情况下，点击 `解锁` 按钮到 `No` 即可解锁无人机。



最后，滑动 Exp2_ControlSystemDemo.slx 中右侧的 CH3 油门通道即可起飞无人机，另外，再同样滑动其余通道可控制无人机的姿态，可在RflySim3D中观察无人机的飞行状态。



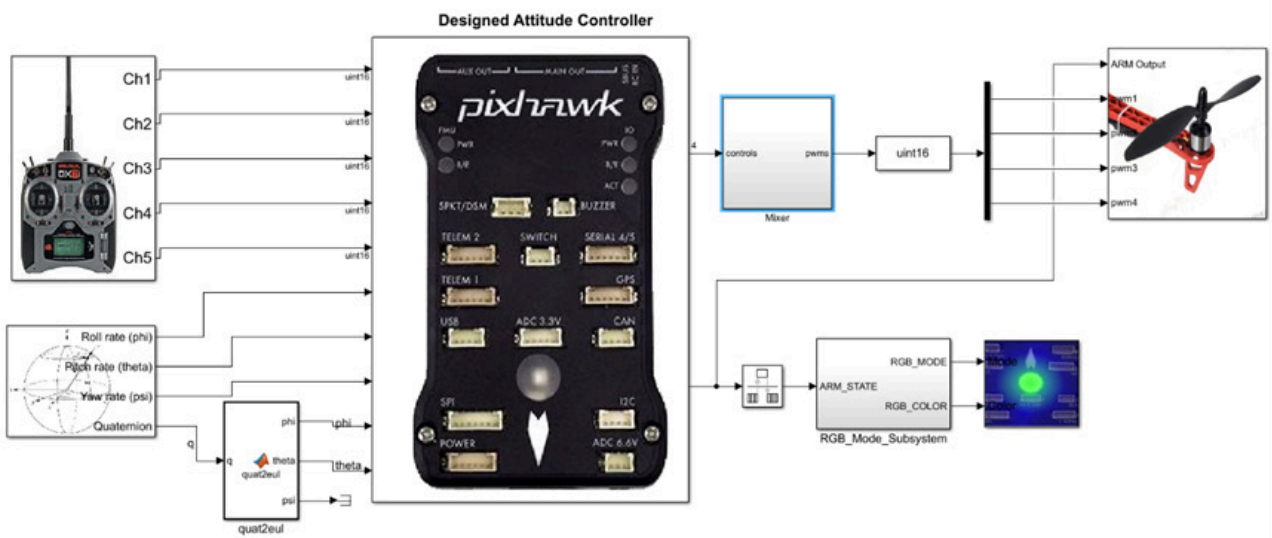


4.3 硬件在环仿真

在进行正式硬件在环之前，请务必将飞控状态进行相关调整，可优先完成实

验：`[RflySim安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e2.FCUIntro`

`Exp4_AttitudeSystemCodeGen.slx`，如下图所示，该系统的输入为遥控器的Ch1-Ch5通道、姿态角速率和俯仰和滚转角。

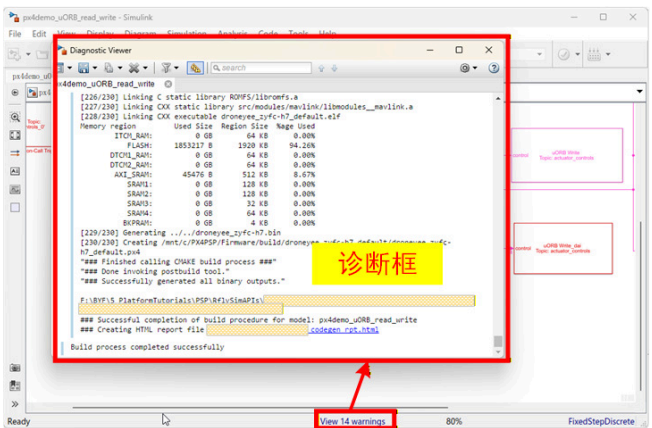
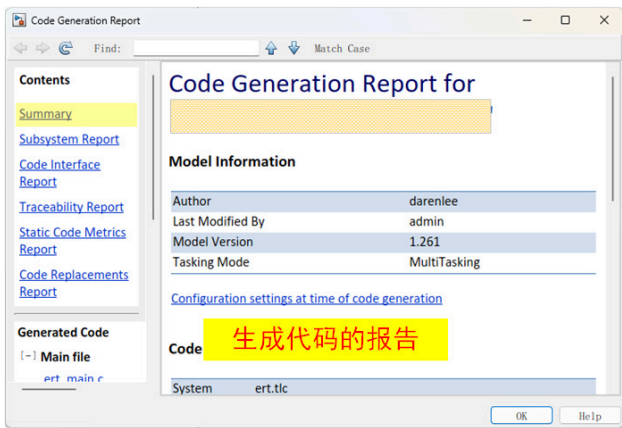
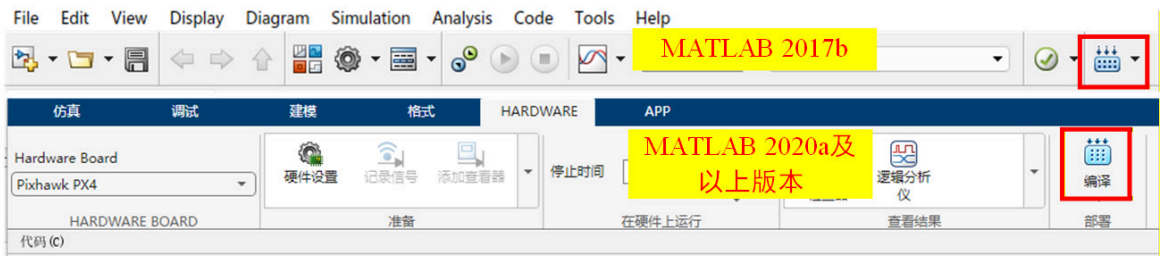


相较于 `Exp2_ControlSystemDemo.slx` 模型，整体上做了如下改动：

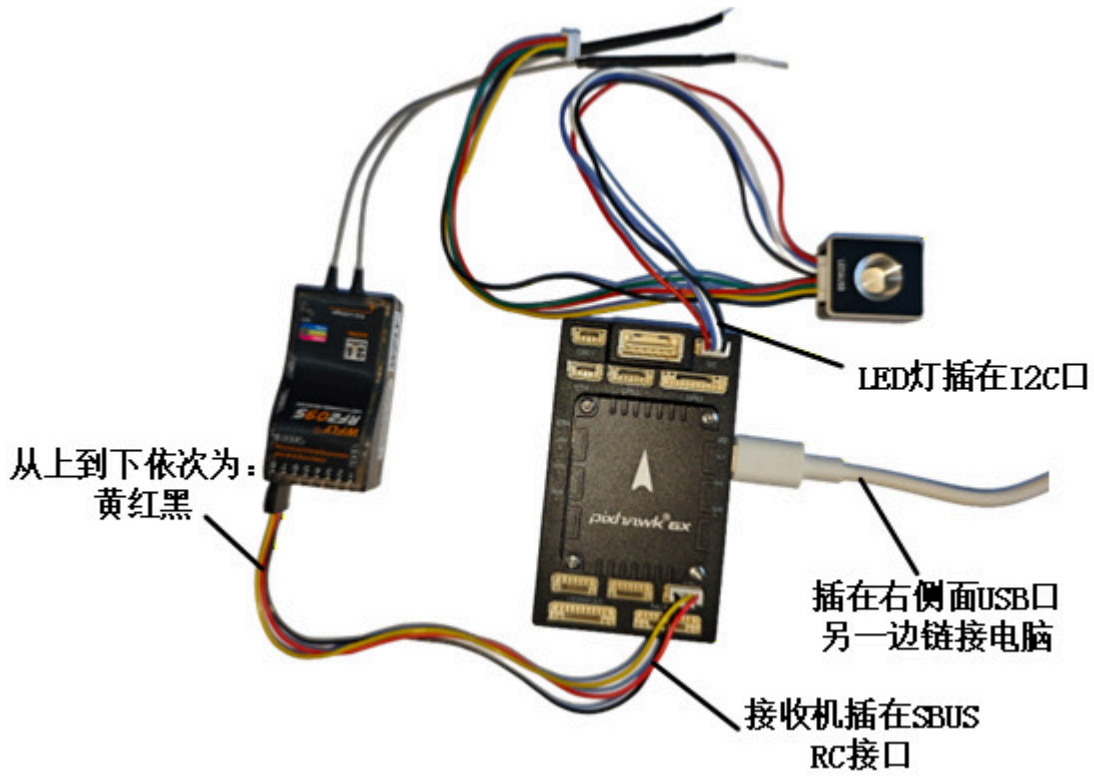
1. 控制器模型得左侧输入部分，修改为遥控器输出和PX4中输出得IMU输出，作为整个控制器得输入。
2. 删除无人机动态模型模块，该模型会直接使用CopterSim存储得默认模型或者DLL模型用于仿真。

3. 控制器输出直接链接到无人机电机模块，该模块即可用于HIL仿真，也可以用于直接进行实飞。另外，也可以将控制器得输出直接给到具体得uORB消息（如： `actuator_outputs` ，具体可参见文件 `Exp5_AttitudeSystemCodeGen2.slx` ）。

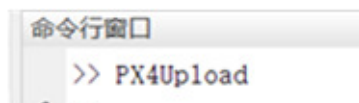
在 MATLAB 中打开 `Init_control.m` 文件，点击运行。按下图所示单击编译按钮，对文件 `Exp4_AttitudeSystemCodeGen.slx` 进行编译，编译完成后会生成代码报告，在 Simulink 的下方点击 View diagnostics 指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出 Build process completed successfully，即可表示编译成功，下图为生成的编译报告。



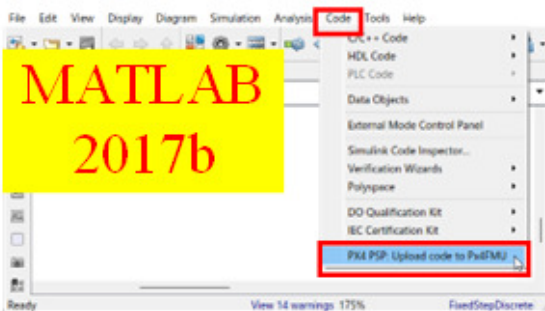
编译成功后，按下图所示用 USB 数据线连接飞控与电脑。



在 MATLAB 命令行窗口输入：PX4Upload 并运行或点击 PX4 PSP: Upload code to Px4FMU。



或



弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传至100%即表示成功。

```

C:\WINDOWS\SYSTEM32\cmd X + v
Loaded firmware for board id: 1010,0 size: 1770080 bytes (96.46%), waiting for the bootloader...

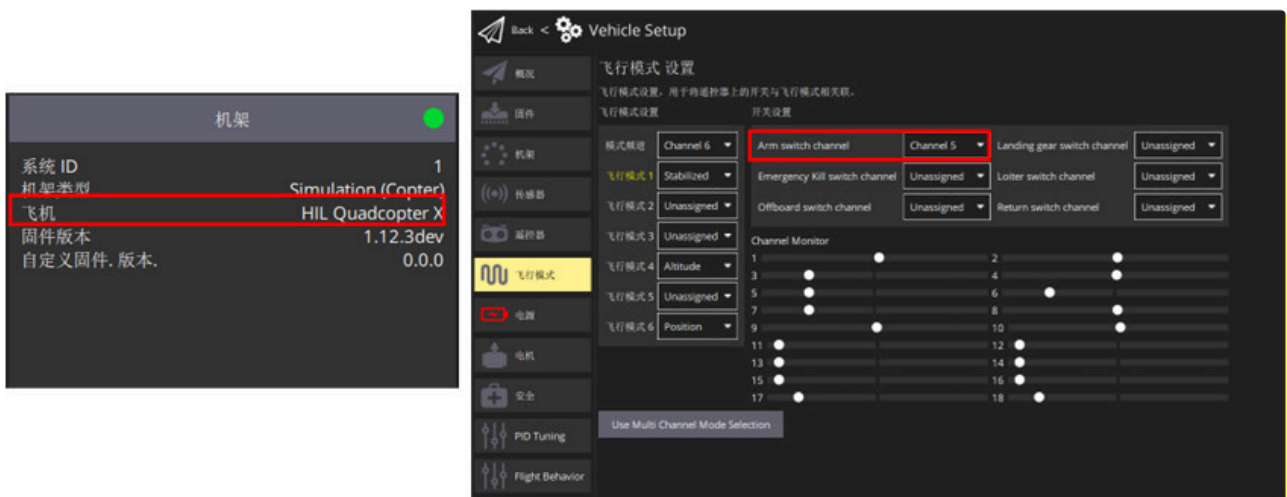
Attempting reboot on COM13 with baudrate=57600...
If the board does not respond, unplug and re-plug the USB connector.

Found board id: 1010,0 bootloader version: 5 on COM13
sn: 001b00333133510836303930
chip: 20036450
family: b'STM32H7[4|5]x'
revision: b'V'
flash: 1835008 bytes
Windowed mode: False

Erase : [=====] 100.0%
Program: [=====] 100.0%
Verify : [=====] 100.0%
Rebooting. Elapsed Time 72.073

```

确认QGC中机架与飞行模式设置如下图所示。打开QGC地面站，连接上自驾仪，(1) 进入“Airframe” 标签，确保模型处于“HIL Quadcopter X” 机架模式 (2) 进入“Flight Modes” 标签页，确认“Emergency Kill switch channel” 切换开关为CH5。即可在QGroundControl 中设置飞机起飞等操作。



上传成功后，双击打开 *桌面\RflyTools\HITLRun.lnk 或 *PX4PSP\RflySimAPIs\HITL Run.bat 文件，在弹出得对话框中输入飞控得串口号。

```
HITLRun
-----
Please input the Pixhawk COM port list for HIL
Use ',' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks

Available COM ports on this computer are:
COM3: USB ???

Recommended COM list input is: 3

-----
My COM list for HITL simulation is:3
```

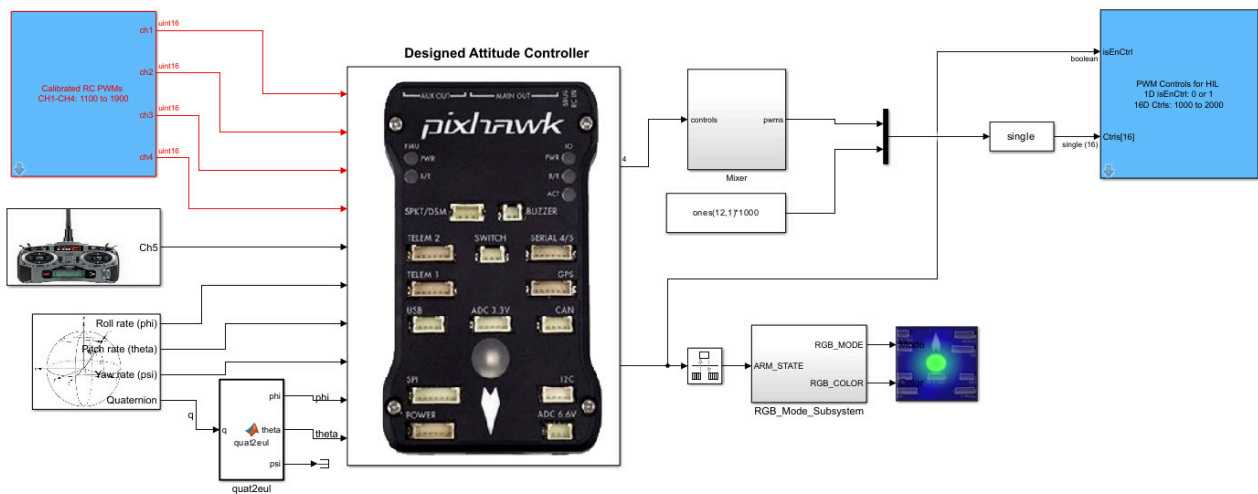
等待仿真环境初始化完成。脚本将会启动 1 个 QGC 地面站，1 个 CopterSim、1 个 RflySim3D 软件，等待CopterSim软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成。如下图所示：



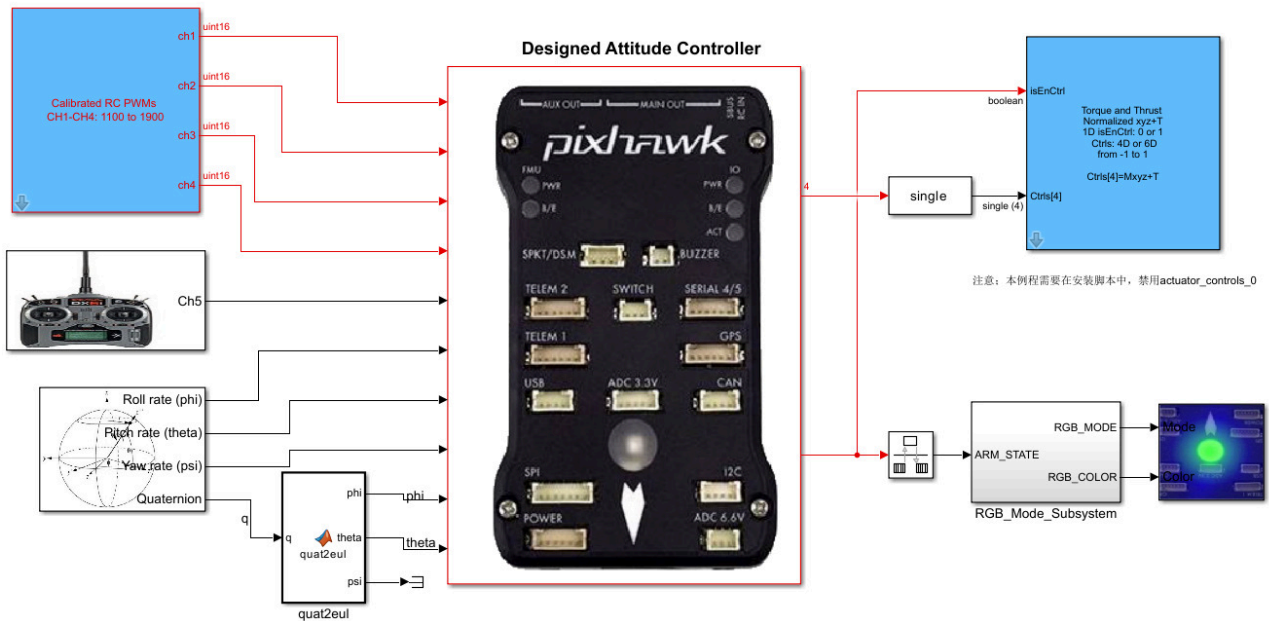
通过 CH5 解锁之后，在 RflySim3D 中即可看到飞机正常起飞，通过CH1~Ch4通道调整飞机姿态和高度。



另外， `Exp5_AttitudeSystemCodeGenRealFlight.slx` ，如下图所示，该模型与 `Exp4_AttitudeSystemCodeGen.slx` 对比，遥控器输入改为了归一化处理后得模块输入。控制器最终输出链接到了HIL专用得归一化模块。



`Exp6_AttitudeSystemCodeGen3.slx` ，如下图所示，相较于 `Exp4_AttitudeSystemCodeGen.slx` 模型。控制器输出链接直接给到了控制信号接收模块，该模型中直接删除了原模型得混控器模块，使用了PX4中得混控器。

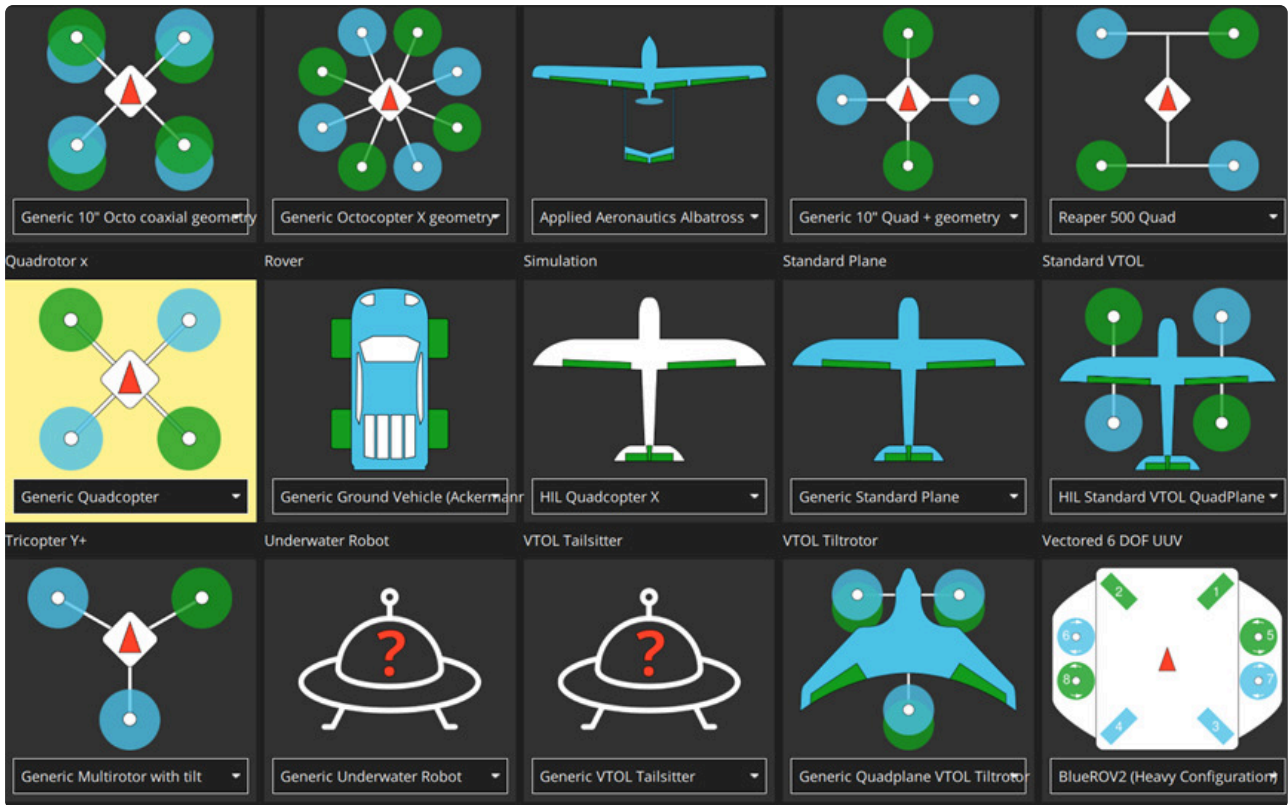


4.4 转台实验

转台实验是实飞前可选择性进行得实验，可以快速得验证自己搭建得模型在真机上得效果如何。

4.4.1 实飞前准备

在QGC中，通常需要在“载具设置”中选择无人机的机架类型。对于多旋翼无人机，如四轴、六轴等，应选择与你的无人机实际机架相匹配的选项。如果QGC中没有直接列出你的机架类型，可能需要手动设置或选择一个相近的类型，并调整相关参数以匹配你的无人机。



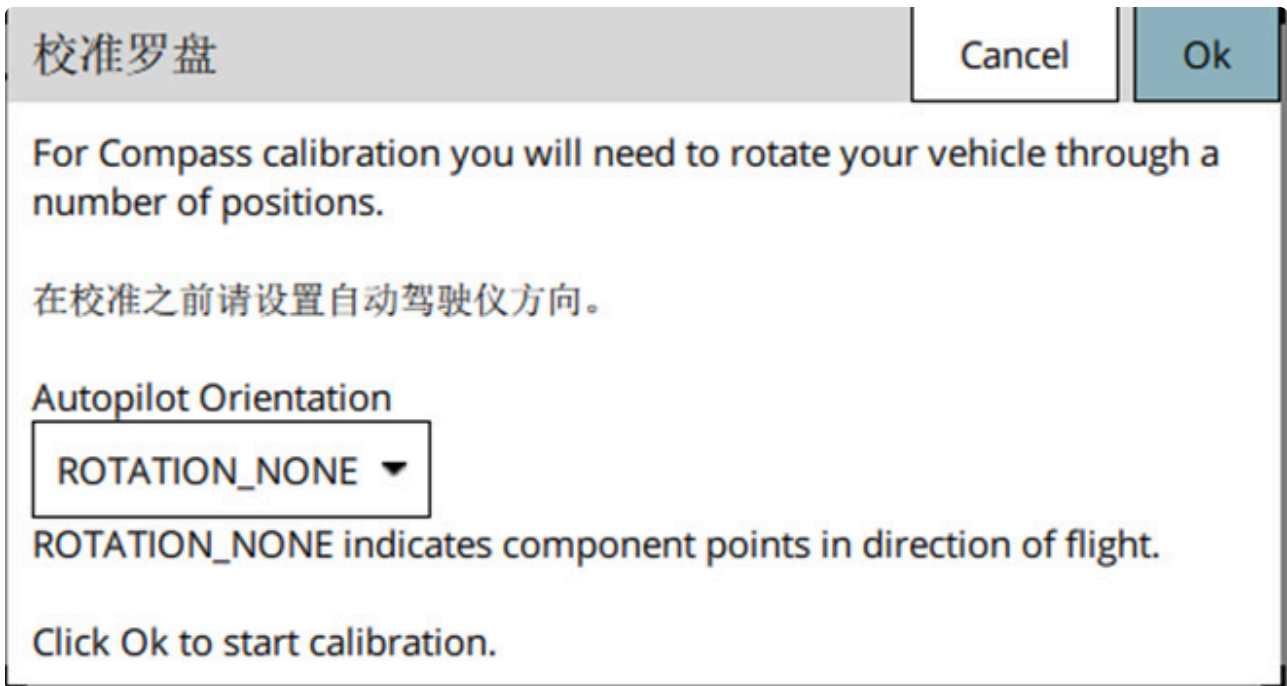
机架 设置

机架设置可用于选择与您的设备相匹配的机架构型。这将设置并可能改变一些飞行参数的数值。
你已经连接了 HIL Quadcopter X。要更改此配置，请在下方选择所需的机型，然后点击“应用并重启”。

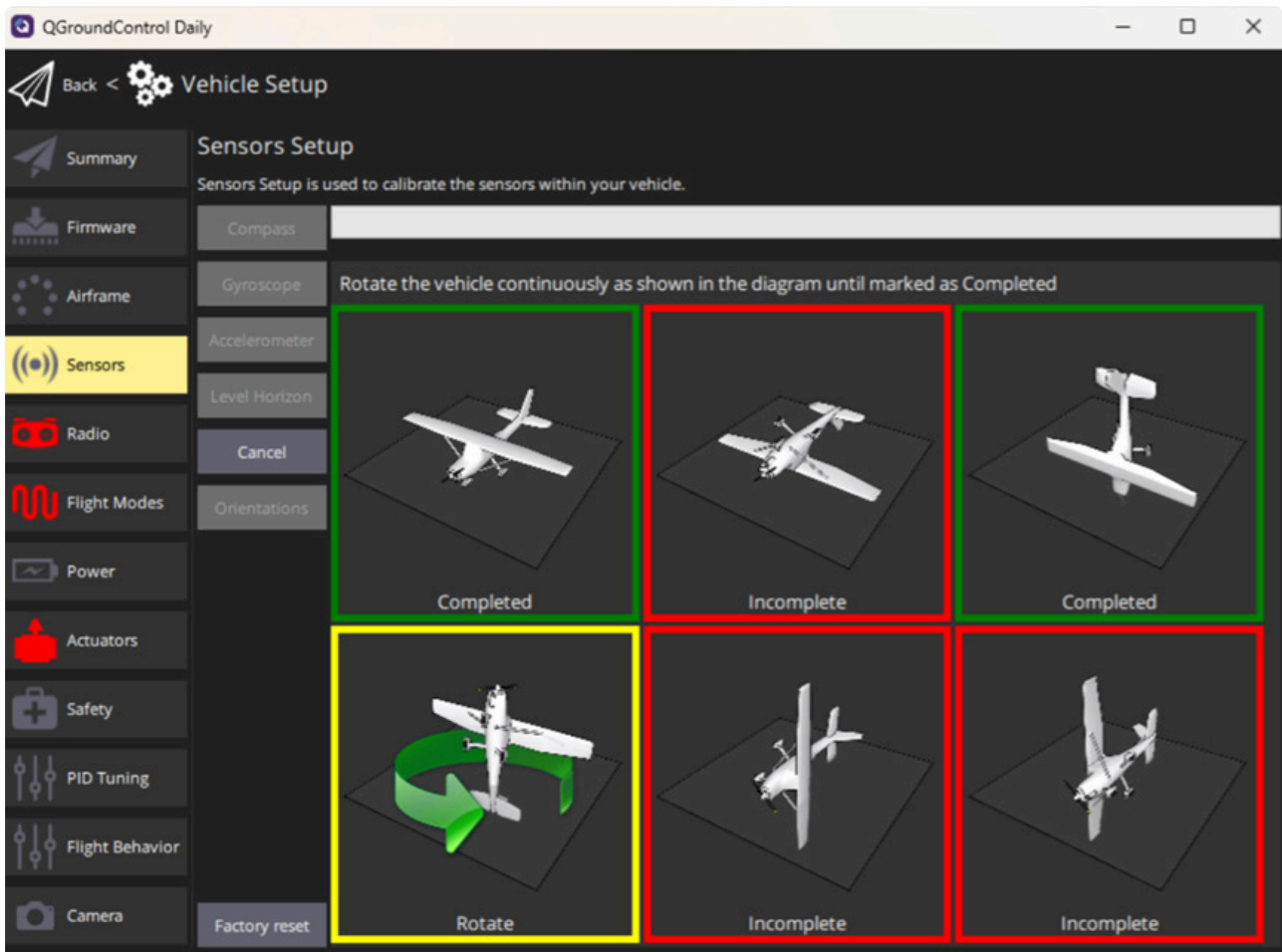
应用并重启

4.4.2 传感器校核

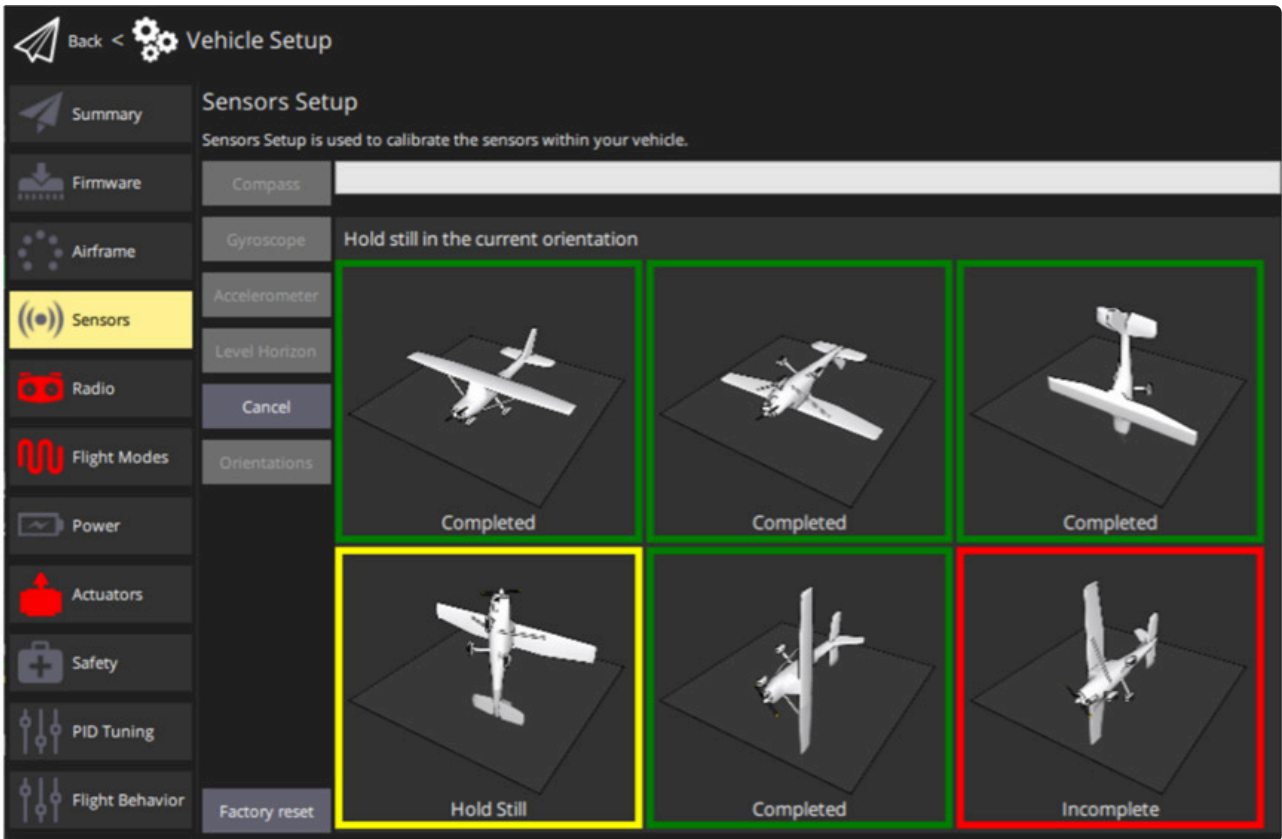
启动QGroundControl并连接飞控。在选择左上角“Q”图标>选择“Vehicle Setup”，进入到车辆设置当中>选择“传感器”（侧边栏），进入到传感器校准中。单击“Compass”或“罗盘”按钮。



单击“Ok”开始校准。把你的飞机放置在下面显示的某一个方向，并保持静止。随后根据提示（方向图像变为黄色）指定方向旋转飞行器。该位置标定完成后，屏幕上的相应图示将变成绿色。



依次类推，完成传感器界面得所有校核，完成后，断电重启无人机。

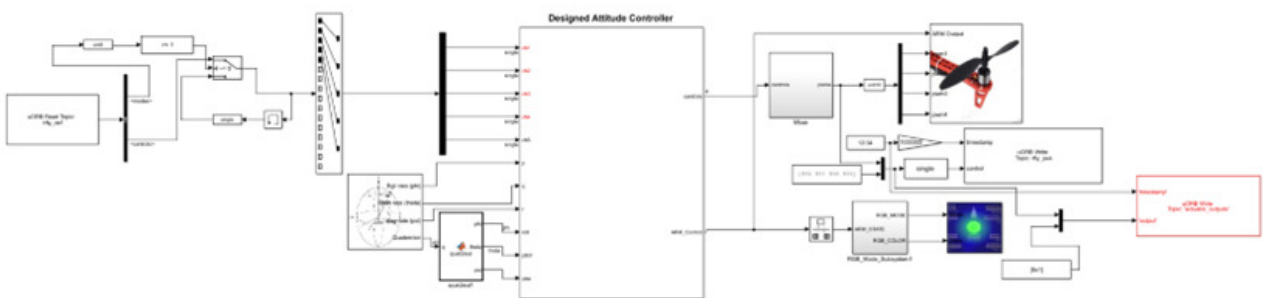


4.4.3 转台实验1——数传链接

转台实验需要将无人机通过无线得形式链接到电脑，链接方式又多种可根据自己飞机上搭载得设备选择性链接。

在MATLAB软件中打开文件：`\Exp4.5_Rotarytable\Init_control.m`。

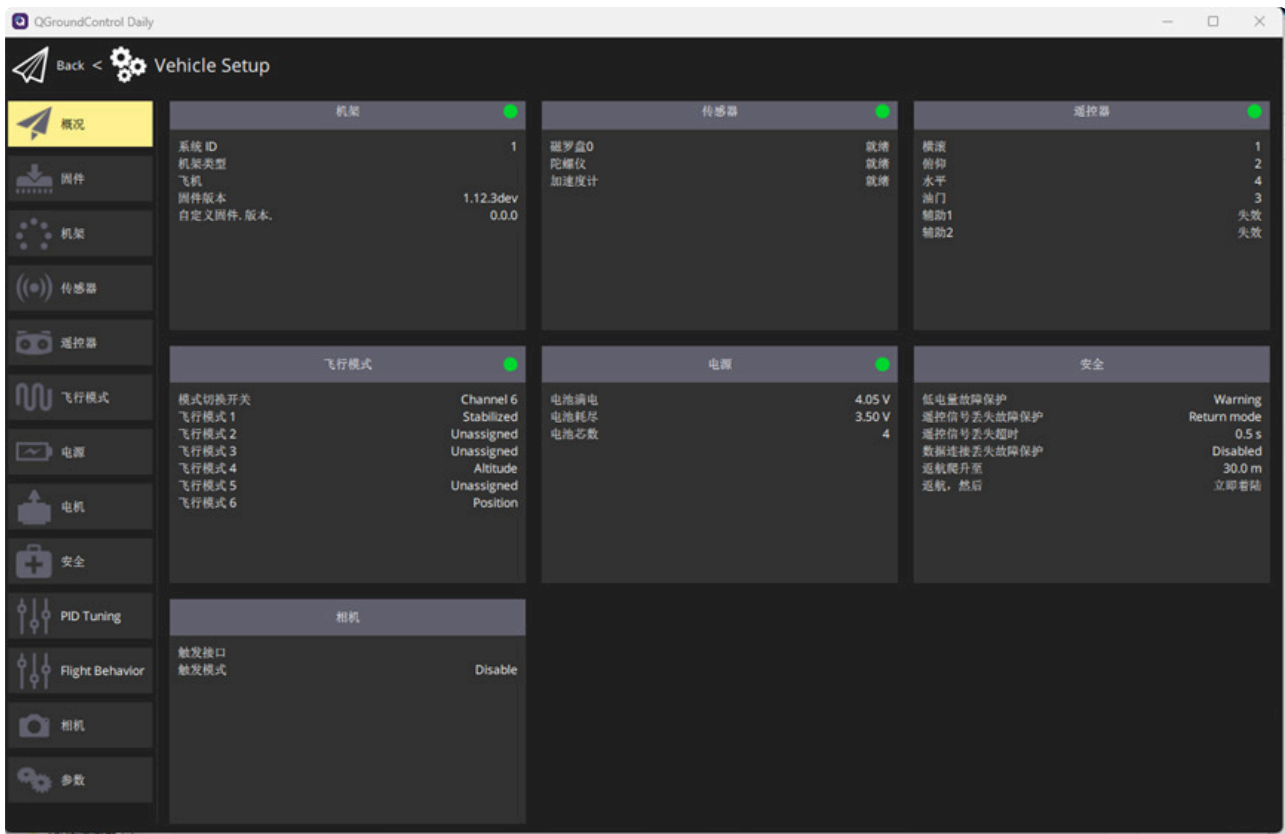
`Init_control.m` 运行后会自动打开 `Rotarytable_Quadrotor_Control.slx`。



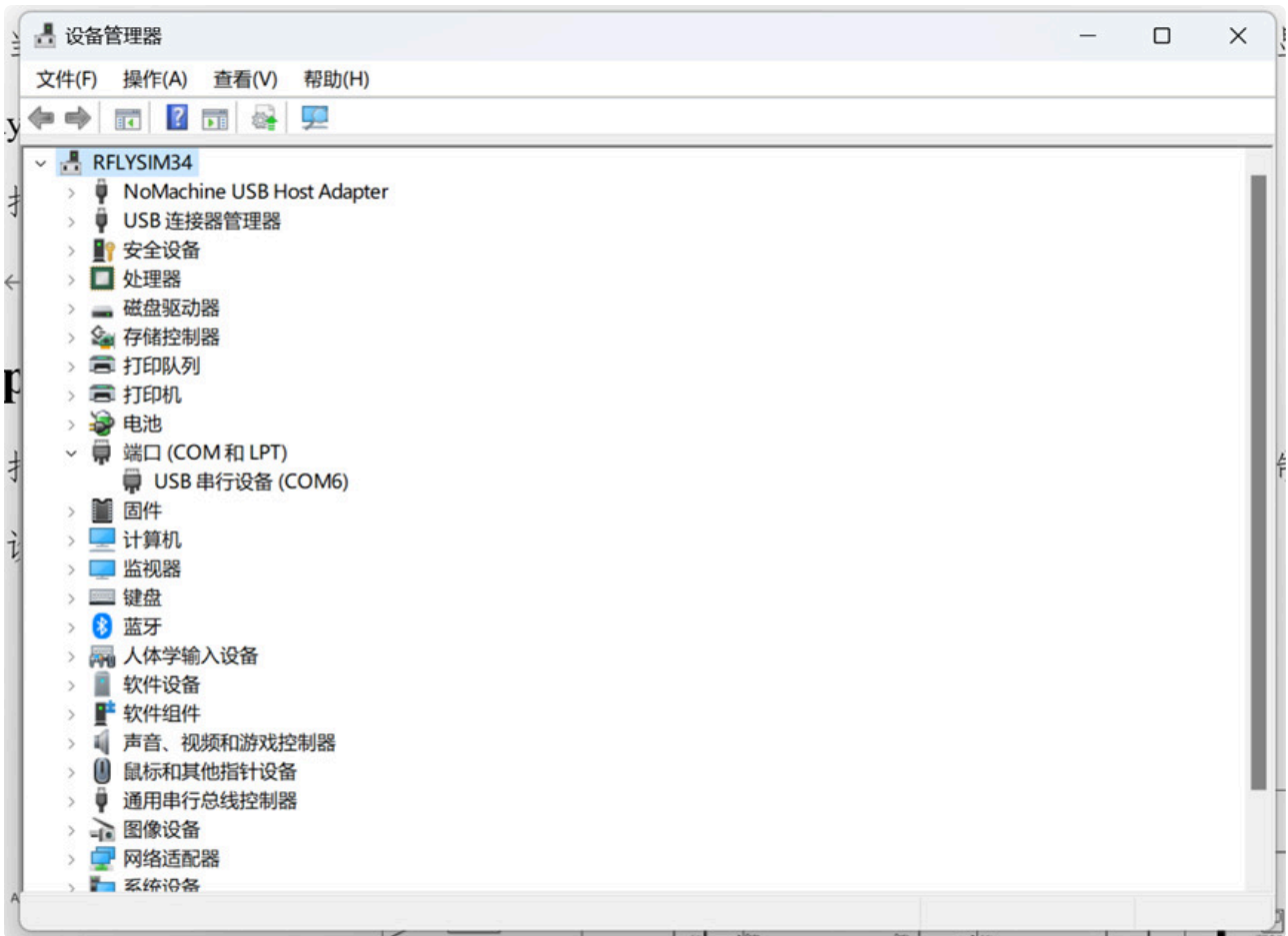
在打开 `Init_control.m` 文件，查看参数设置。如下参数适用于立式转台的X200飞机。可对P参数和I参数进行微调，优化控制效果。

```
Init_control.m x +
1
2 % 俯仰
3 - Kp_PITCH_ANGLE = 5;
4 - Kp_PITCH_AngleRate = 0.1;
5 - Ki_PITCH_AngleRate = 0.05;
6 - Kd_PITCH_AngleRate = 0.0001;
7
8 % 滚转
9 - Kp_ROLL_ANGLE = 5;
10 - Kp_ROLL_AngleRate = 0.1;
11 - Ki_ROLL_AngleRate = 0.05;
12 - Kd_ROLL_AngleRate = 0.0001;
13
14 % 偏航
15 - Kp_YAW_ANGLE = 2;
16 - Kp_YAW_AngleRate = 0.1;
17 - Ki_YAW_AngleRate = 0.01;
18 - Kd_YAW_AngleRate = 0.00;
19
20 % 积分饱和
21 - Saturation_I_RP_Max = 0.2;
22 - Saturation_I_RP_Min = -0.2;
23 - Saturation_I_Y_Max = 0.1;
24 - Saturation_I_Y_Min = -0.1;
25
26 % 最大角速率限制, rad/s
27 - MAX_CONTROL_ANGLE_RATE_PITCH = 5;
28 - MAX_CONTROL_ANGLE_RATE_ROLL = 5;
29 - MAX_CONTROL_ANGLE_RATE_YAW = 3;
30
31 % 启动模型
32 - Rotarytable_Quadrotor_Control
33
```

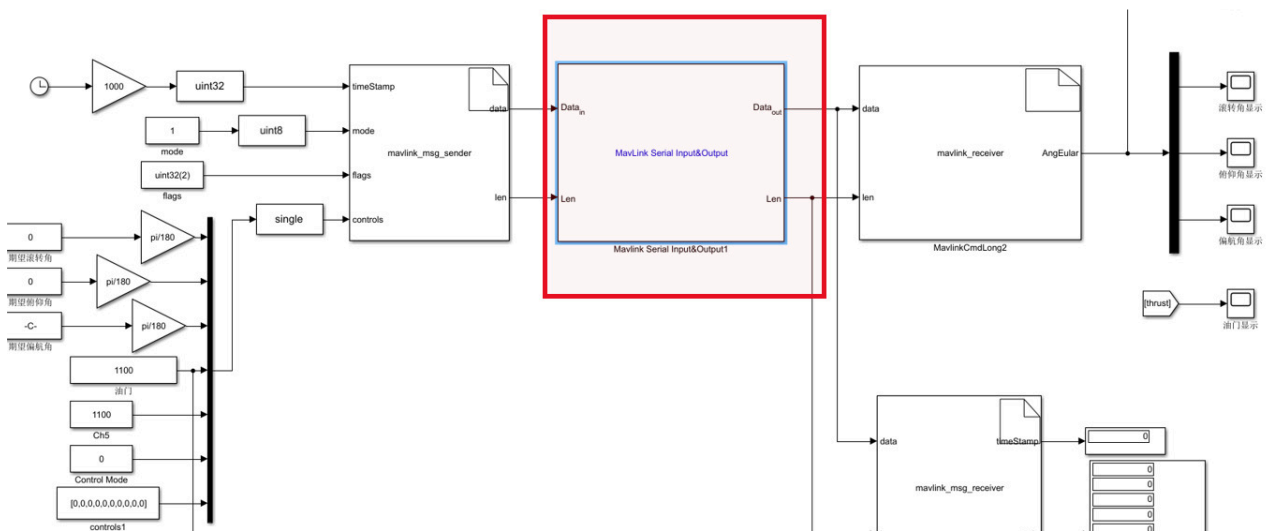
编译模型，连接自己得飞机中得飞控，烧录代码（与[硬件在环仿真](#)中得步骤相同）。完成传感器校准、机架、遥控器等配置工作。

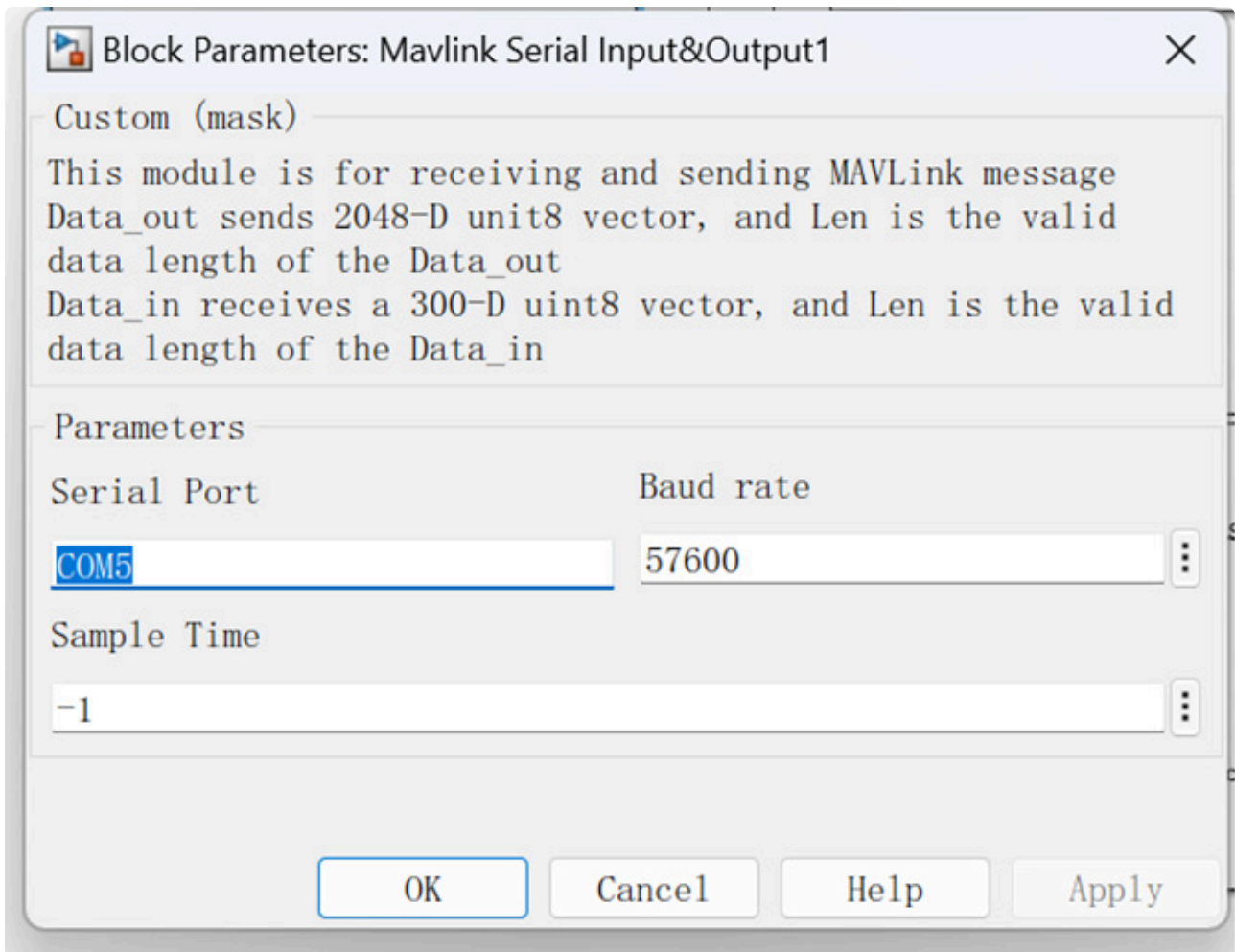


飞机固定在台架上，并连接数传。当连接数传之后，我们可以在QGC上看到数传与无人机是否建立连接，当QGC显示Ready to fly,需要关闭QGC，否则会抢占端口，使后续工作无法完成。打开设备管理器，找到端口。之后，需要在Simulink中相应的模块改为对应的端口。

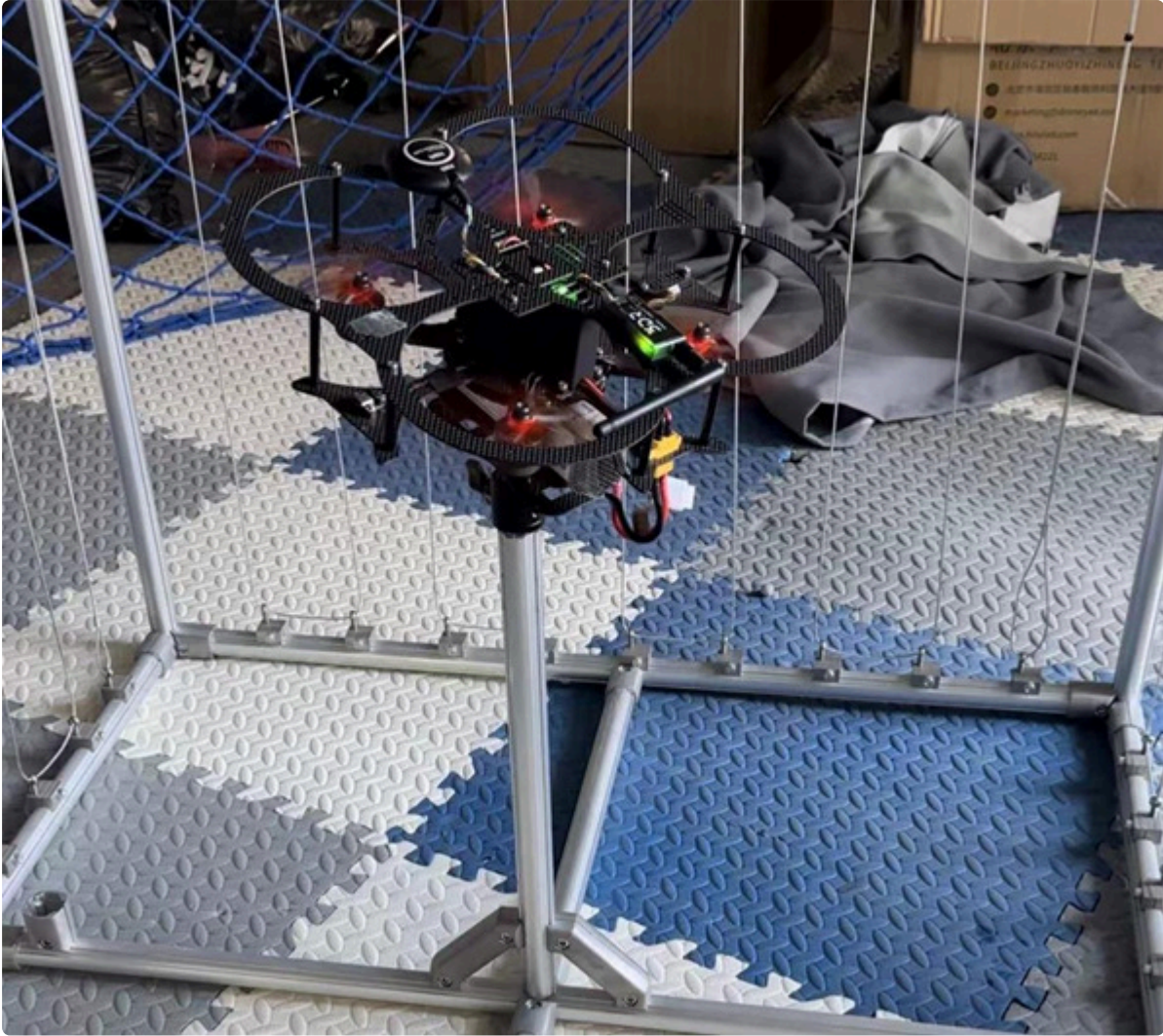


注：若数传链接电脑之后，未显示端口号，请根据数传型号下载相关驱动文件，本平台提供3DR X6平台数传驱动地址为：`"*\RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\0.ResourcesFile"`。

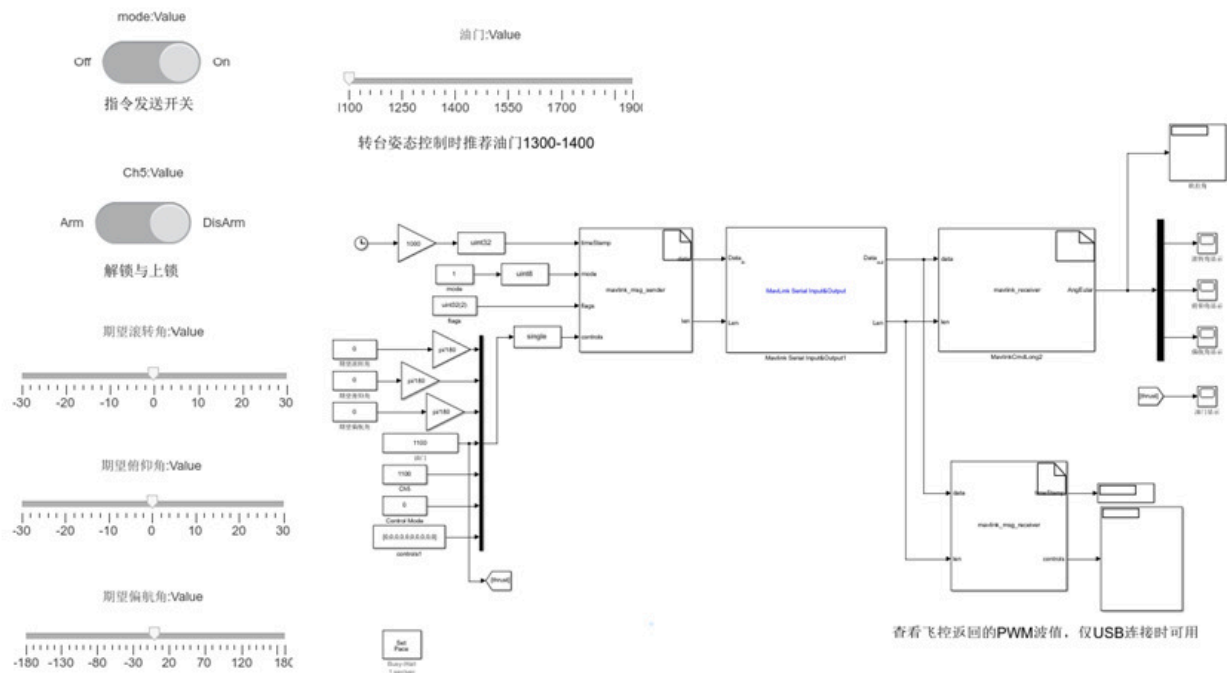




如下图所示。

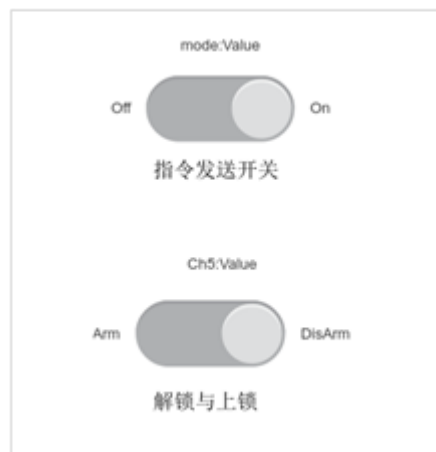


打开 `.\Exp4.5_Rotarytable\Matlab Sender\Mav_Control_Quadrotor.slx` 发送消息控制无人机，整体形式如下图所示。控制指令的设置集中在左侧和上部，返回结果的显示集中在右侧。

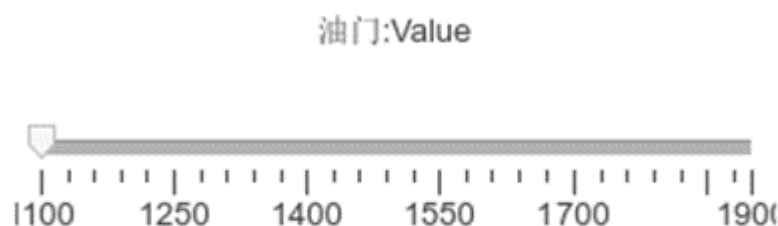


下面详细介绍每一个部分的功能。如下图所示，是发送端的两个关键控制开关，排在上面的是“指令发送开关”，排在下面的是“解锁与上锁”开关。

- 当“指令发送开关”设置为Off时，Mav_Control_Quadrotor.slx上的任何修改都不起作用，即不会发送任何有效指令。当“指令发送开关”设置为On时，“解锁与上锁”开关将变得有效，同时滚转、俯仰、偏航及油门的设置也将生效。
- “解锁与上锁”开关用于控制电机解锁。在首次解锁时，需要三步才能将电机进行解锁：第一步将“解锁与上锁”开关打到Arm，等待2s，电机发出滴答声；第二步将“解锁与上锁”开关打到DisArm，等待2s，电机发出滴答声；第三步，将“解锁与上锁”开关打到Arm，将完成电机的解锁。
- 警告⚠：电机解锁前，应仔细检查油门、期望滚转角、期望俯仰角、期望偏航角的设置，否则可能因转速过大损坏飞机或者造成人员受伤。
- 建议：首次实验时，油门值设定为最低，期望滚转角、期望俯仰角、期望偏航角设置为0。待熟悉了平台的基本使用后，再调整油门值、期望滚转角、期望俯仰角、期望偏航角。

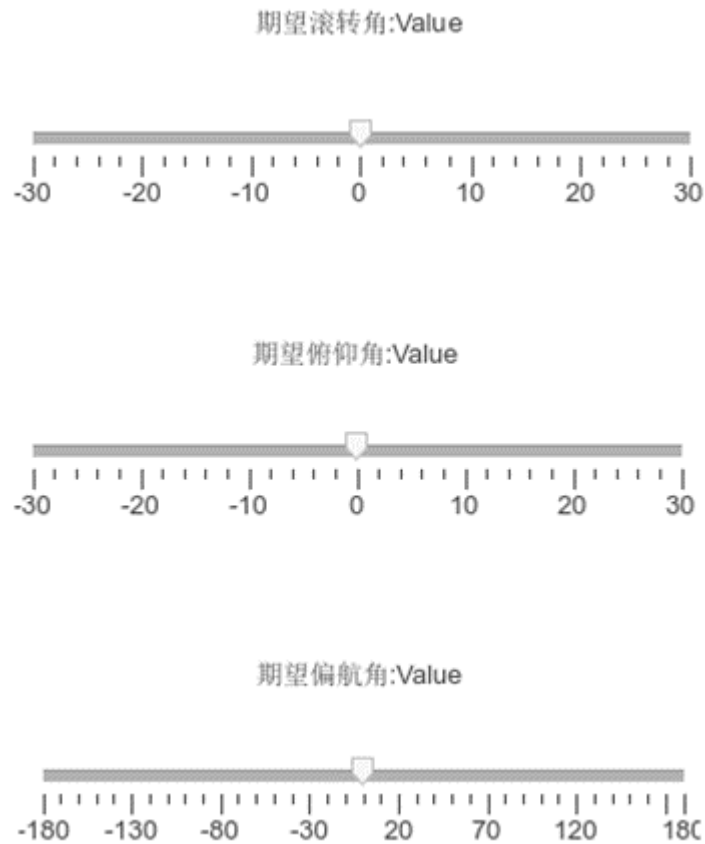


油门值设置。如下图滑杆，可以设置油门值。油门值可以理解为PWM脉宽，通常无刷电机支持的脉宽是1000-2000，但为了安全起见仅支持设置1100-1900。当滑杆打到最左侧时，将设置为最低值1100，当打到最右侧时将设置为最大值1900。在做转台上V200飞机姿态控制时，建议将油门设置为1300-1400。

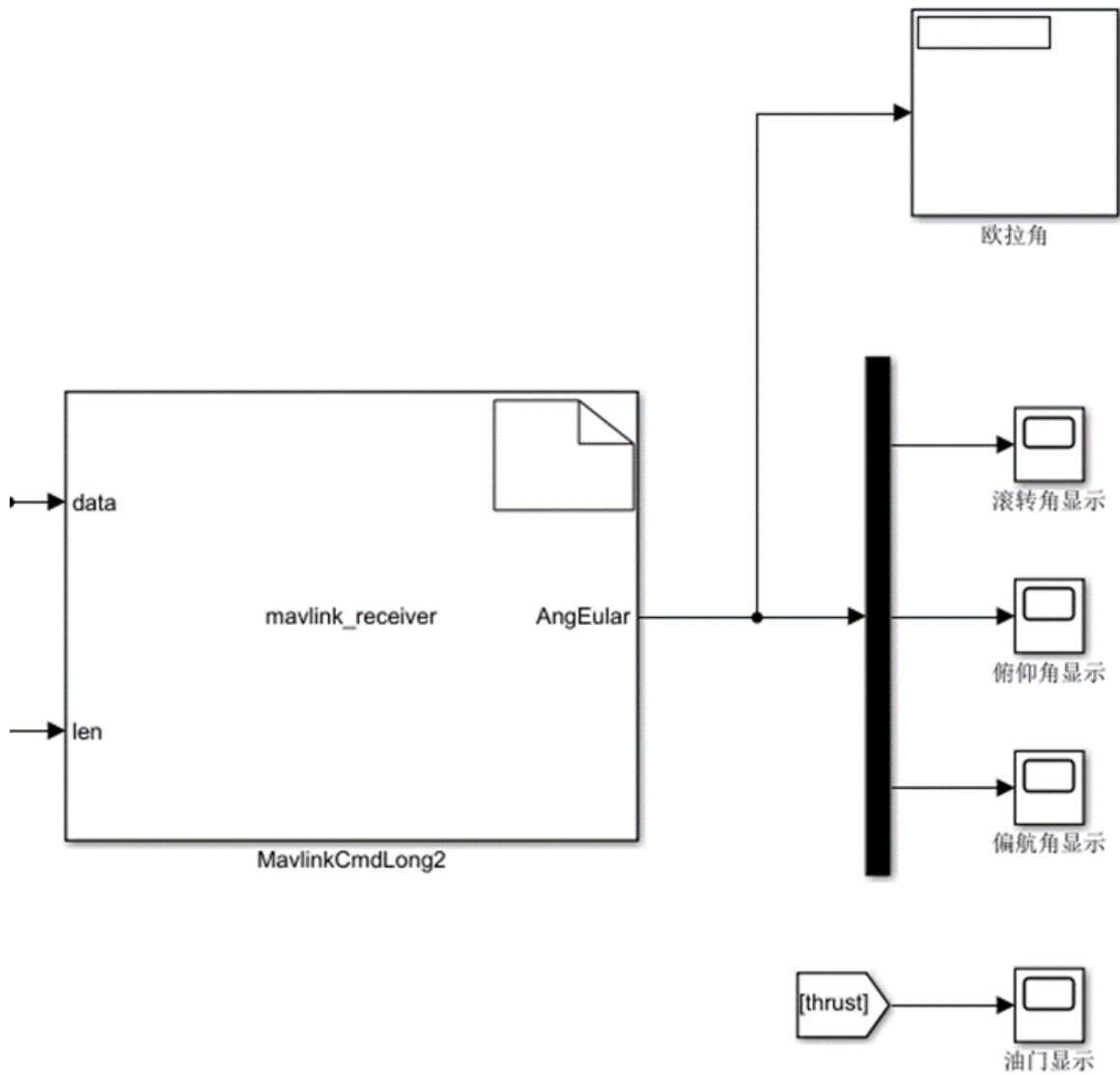


转台姿态控制时推荐油门1300-1400

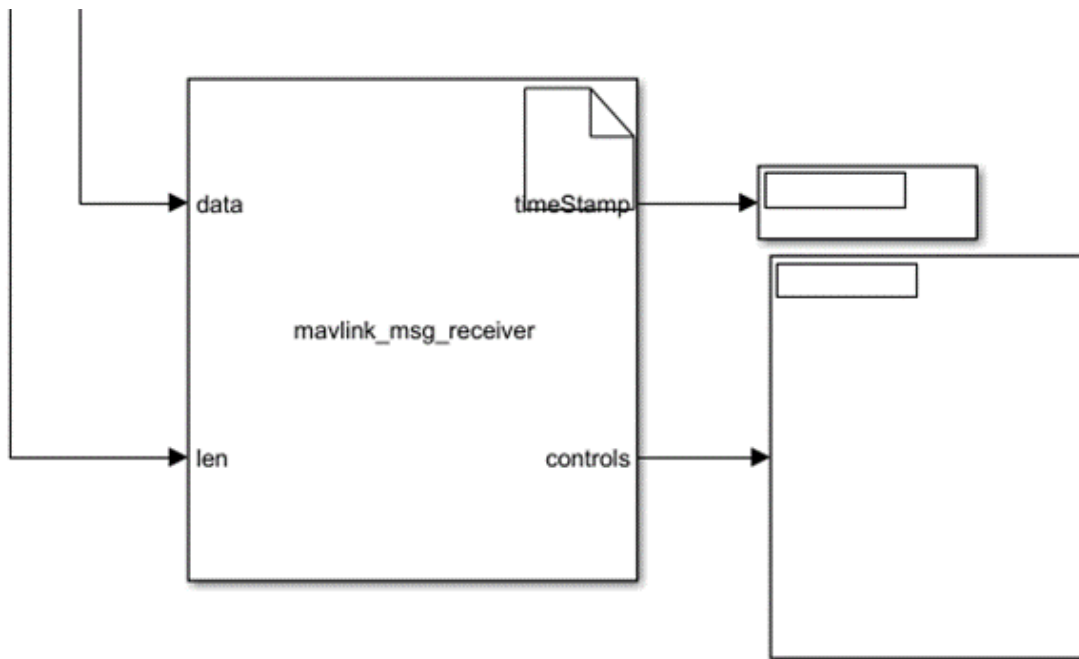
期望姿态设置。如下可设置期望滚转角、期望俯仰角、期望偏航角。期望滚转角、期望俯仰角可设置 -30° ~ 30° ，在开始调试时建议最大倾角不超过 20° 。偏航角支持 -180° ~ 180° ，偏航角可随意设置，对系统稳定性影响较小。



在Mav_Control_Quadrotor.slx的右侧，可以显示实时欧拉角的数值，也可以通过示波器查看欧拉角的曲线。在最下面的一个示波器，可以显示油门值。



在设计控制器时，往往需要验证控制逻辑是否符合预期，这时需要查看控制器的原始输出。如下图的模块可以显示控制器的原始输出，即每一个电机对应的PWM波的脉宽。该功能仅使用USB连接飞控时可用，因为使用数传连接时速率太低，相应的消息不会发送。



查看飞控返回的PWM波值，仅USB连接时可用

实验结果表明所设计的控制器能够支持MATLAB发送指令控制姿态。

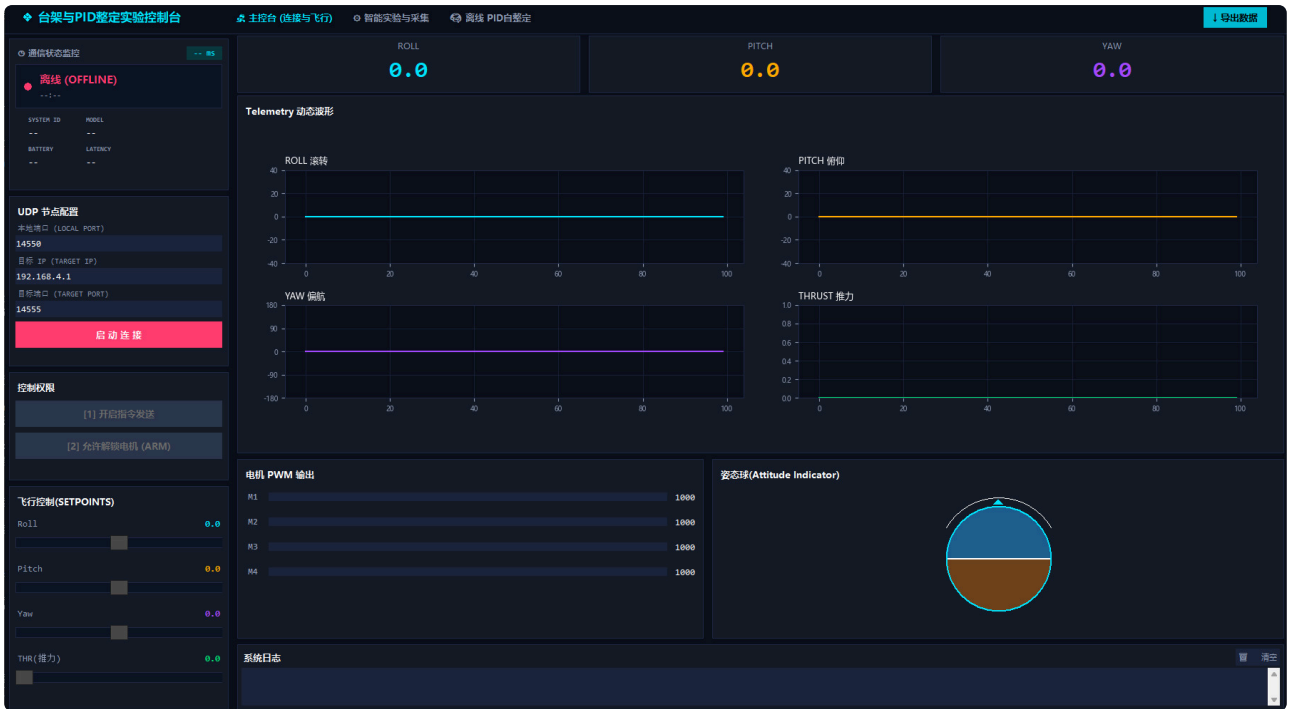
- (1) MAVLink消息可用于上位机控制无人机，MATLAB有相应的模块支持从MAVLink收发消息。
- (2) PID控制器需要限幅，这样能提升控制器的鲁棒性。

4.4.4 转台实验2——WiFi链接

WiFi链接方式需要飞机本身基本将飞控输出得MAVLink消息进行转发得功能，也就是需要具备在无线传得情况下，地面站QGC可链接到飞机得功能。下面介绍WiFi链接下的实验步骤：

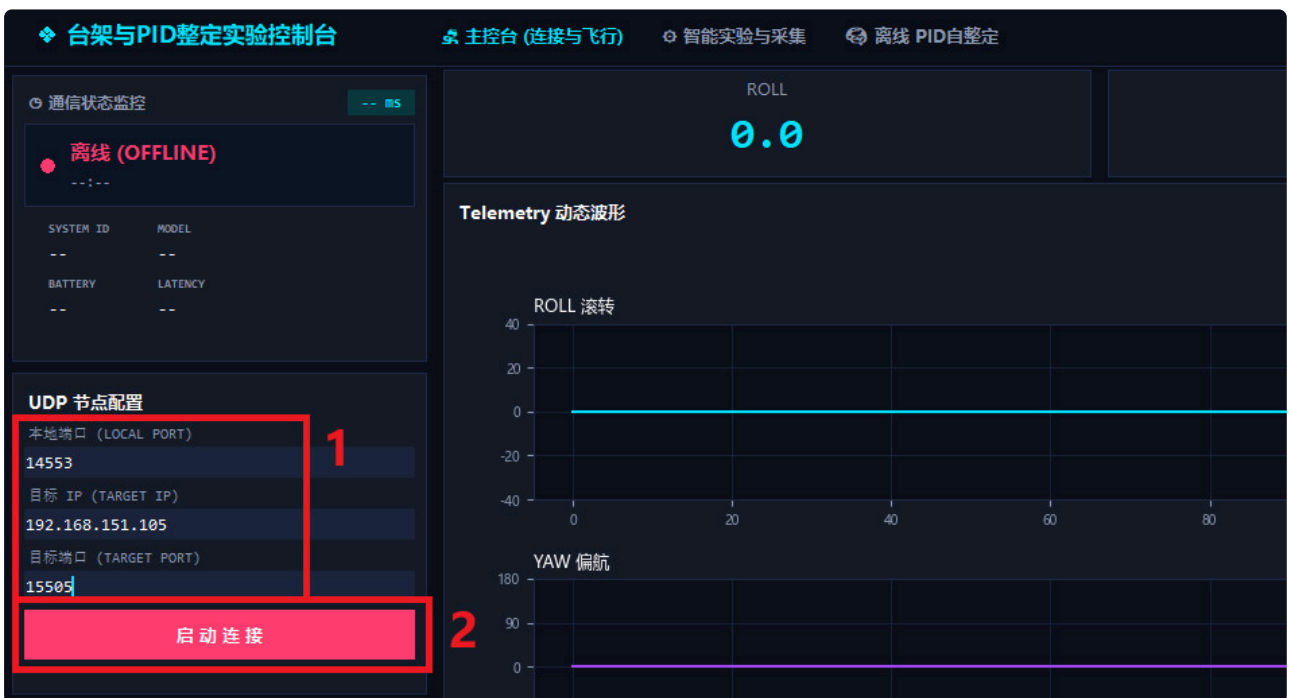
注：本步骤得实验中，飞控中得固件可以是官方固件也可以是本实验得 `Exp4_AttitudeSystemCodeGen.slx` 生成得固件。但需要注意得是：若为本实验中得模型文件生成得固件，那么就控制器中得参数是 `Init_control.m` 中得参数，请注意调参得修改方式。

双击 `Python38Run.bat` 脚本，在弹出的对话框中输入：`python control-PID.py`，启动“台架与PID整定实验控制台”软件。整体界面如下图所示。控制指令与参数设置集中在左侧，遥测数据与动态波形的显示集中在右侧。



配置通信节点：在界面左侧的“UDP 节点配置”区域，根据您的飞机中得连接方式填写目标 IP 与端口（若使用软件在环仿真通常填 127.0.0.1，若使用 WiFi 数传请填飞控实际 IP）。

建立连接：点击红色的“启动连接”按钮。



确认状态：观察右上角“通信状态监控”区域，当指示灯变为绿色，且显示“连接正常 (LINK OK)”时，说明上位机已成功接管飞控，此时右侧姿态球应随无人机晃动而同步转动。



下面详细介绍控制台左侧控制权限与期望设定的功能。如下图所示，是主控端的两个关键控制开关，排在上面的是“[1] 开启指令发送”按钮，排在下面的是“[2] 允许解锁电机 (ARM)”按钮。



当“[1] 开启指令发送”处于未开启（灰色）状态时，主控台上的任何姿态修改都不起作用，即不会向飞控发送任何控制指令流。点击开启后，该按钮将变为深色，系统会瞬间记录当前物理坐标作为保护原点，“[2] 允许解锁电机 (ARM)”按钮将随之激活变亮，同时下方的滚转、俯仰、偏航及油门的滑杆设置也将正式生效。

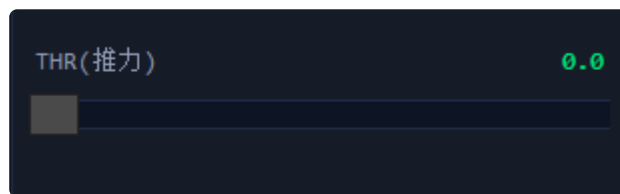
“[2] 允许解锁电机 (ARM)”按钮用于控制飞控进入 OFFBOARD 模式并解锁电机。点击后，飞控会发出滴答声，电机进入怠速状态。如果实验中出现异常，随时再次点击该按钮（此时显示为红色的“紧急上锁”），即可瞬间切断动力。

警告⚠：电机解锁前，应仔细检查下方 THR (油门/推力)、Roll (滚转)、Pitch (俯仰)、Yaw (偏航) 滑杆的设置，否则可能因起步指令过大导致飞机剧烈撞击台架或造成人员受伤。

建议：首次实验时，THR 滑块设定为最左侧最低值（1100），其余姿态期望均保持在 0 位。待电机解锁并平稳怠速后，再缓慢推起油门。



油门值设置。如上图的 THR(推力) 滑杆，可以设置油门值。油门值可以理解为发送给飞控底层的推力百分比（映射为 PWM 脉宽）。滑杆范围被安全限制在 1100-1900。当滑杆打到最左侧时为最低值 1100（怠速），当打到最右侧时为最大值 1900（全油门）。在做转台上 V200 飞机姿态控制实验时，建议将油门缓慢推至 1400-1550 之间，使飞机绳索恰好拉直。



期望姿态设置。在滑杆区域可以设置期望的 Roll (滚转角)、Pitch (俯仰角) 和 Yaw (偏航角)。为保障台架安全，滚转和俯仰角的范围被硬性限制在 -30° ~ 30° ，在开始手动调试时建议操作幅度不要超过 15° 。偏航角支持 -180° ~ 180° ，偏航角可随意设置，对台架系统稳定性影响较小。



在主控台的右侧区域，系统提供了丰富的实时反馈。“Telemetry 动态波形”模块会以折线图的形式，高频刷新并显示当前飞机四轴的实际物理响应状态。而在右下方的“姿态球 (Attitude Indicator)”中，可以直观地看到飞机当前的倾斜姿态。最下方的“电机 PWM 输出”柱状图则展示了四个电机的实时转速脉宽，方便我们验证控制逻辑是否正常下发到了动力层。

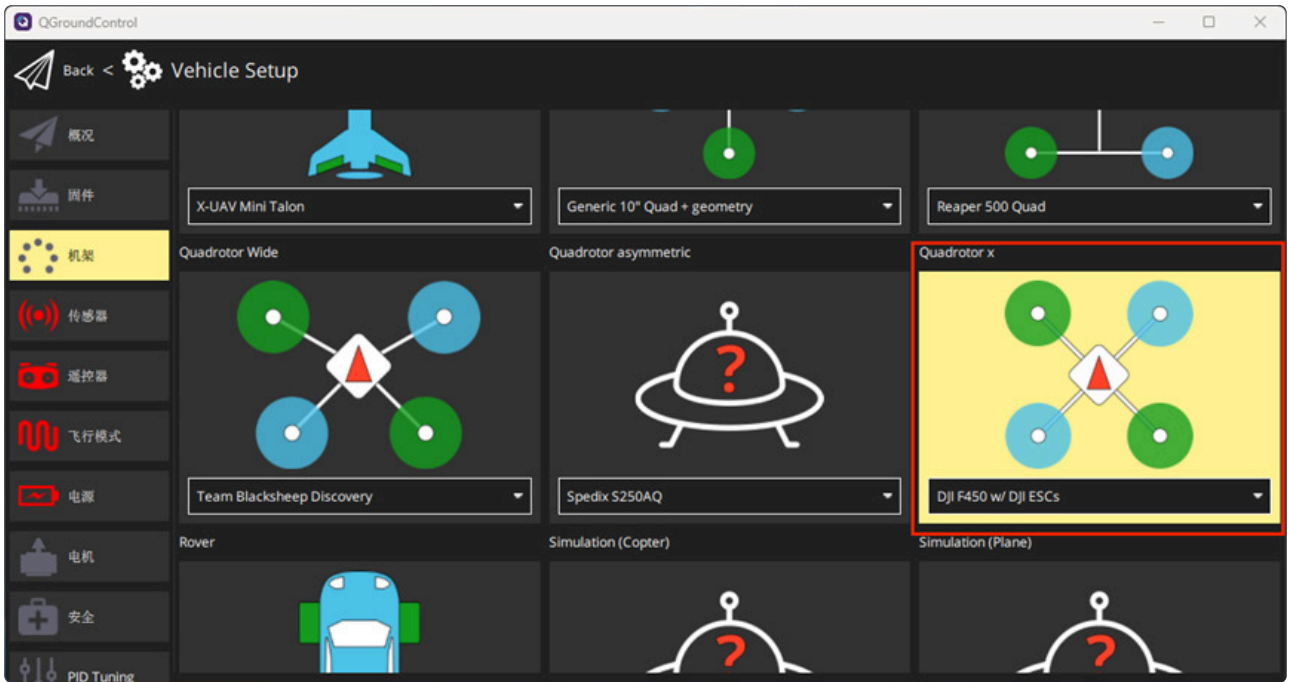


4.5 实飞实验

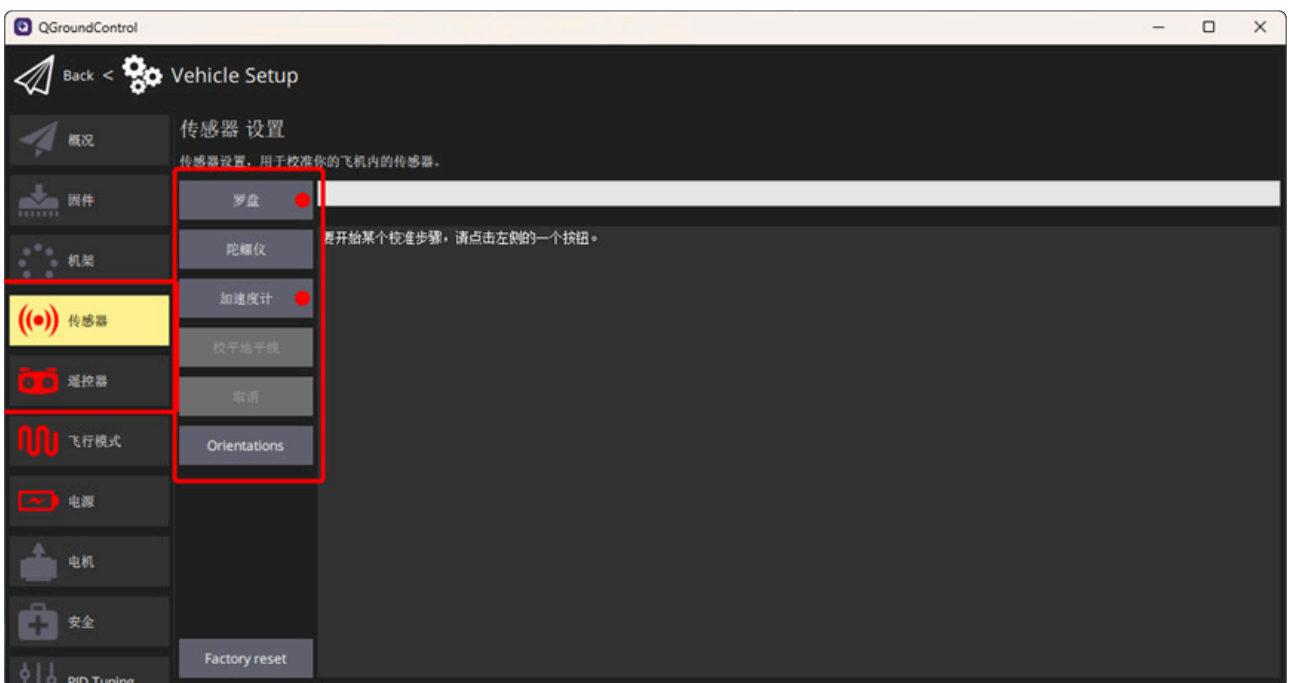
通过QGC烧录给飞控的官方固件。具体操作如下：



将飞机通过USB与电脑进行连接，打开QGC软件，设置机架为：`DJI F450 w/ DJI ESCs`或其他实飞机架；

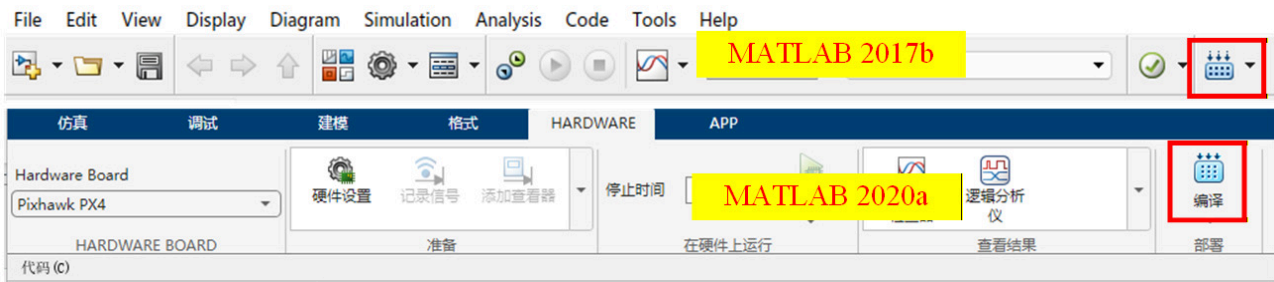


依次校核传感器、遥控器和电池。

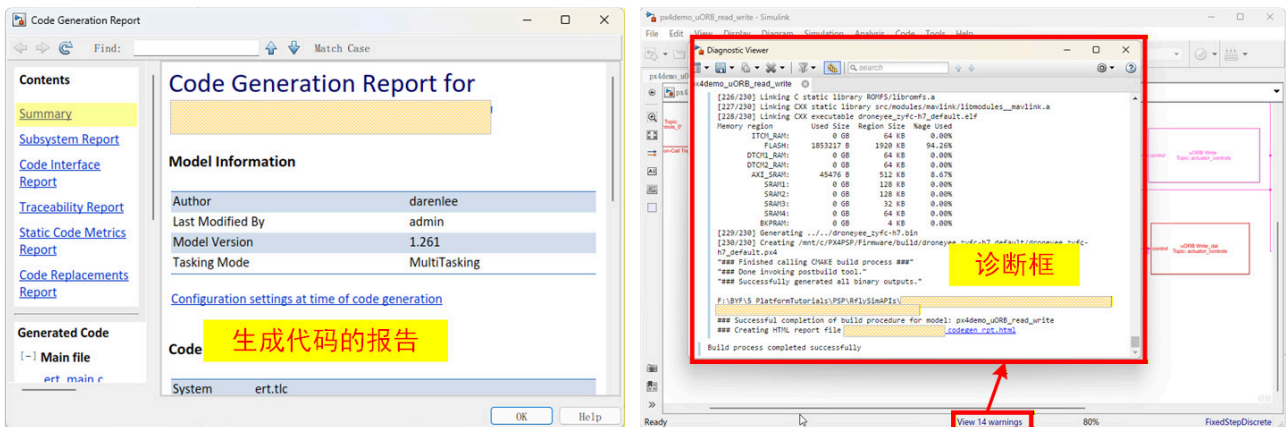


最后，请在指定飞场进行无人机实飞，若正常起飞，说明无人机状态良好；若未正常起飞，请检查传感器校准、参数设置等，具体请联系飞机生产厂家进行解决。请务必保证飞机状态良好的情况下，再进行下一步操作。

打开 `Exp4_AttitudeSystemCodeGen.slx` 文件，在Simulink中，点击编译命令。



在Simulink的下方点击View diagnostics指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出Build process completed successfully，即可表示编译成功，左侧为生成的编译报告。



用USB数据线链接飞控(或飞机)与电脑。在MATLAB命令行窗口输入：PX4Upload并运行，弹出CMD对话框，显示正在上传固件至飞机中，等待上传成功。



```

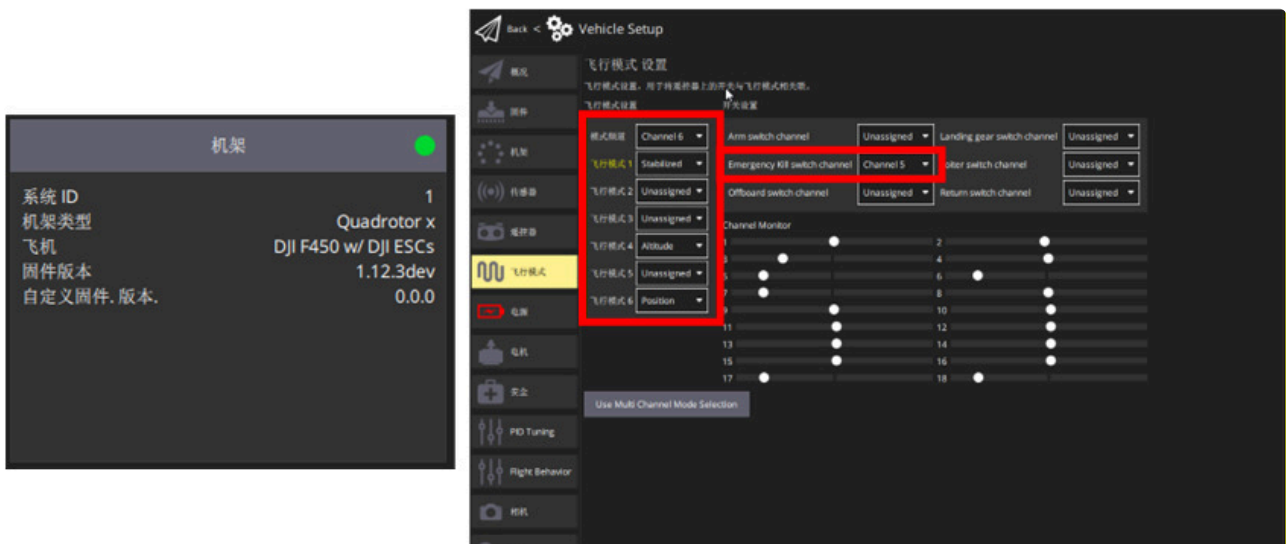
C:\WINDOWS\SYSTEM32\cmd.exe
Loaded firmware for board id: [redacted] size: 1903433 bytes (92.20%), waiting for the bootloader...

Found board id: [redacted] bootloader version: 5 on COM5
sn: 001e00354256500c20323441
chip: 10016451
family: b'STM32F7[6|7]x'
revision: b'Z'
flash: 2064384 bytes
Windowed mode: False

Erase : [=====] 100.0%
Program: [ ] 3.4%

```

打开QGroundControl软件，等待飞机连接成功。确认无人机机架类型选择如下图，并设置遥控器通道如下，其中CH5为解锁，CH6为模式切换。



遥控器的设置如下图。注：遥控器设置中，CH5通道需设置为二段式开关，CH6通道设置为三段式开关。具体设置请见实

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

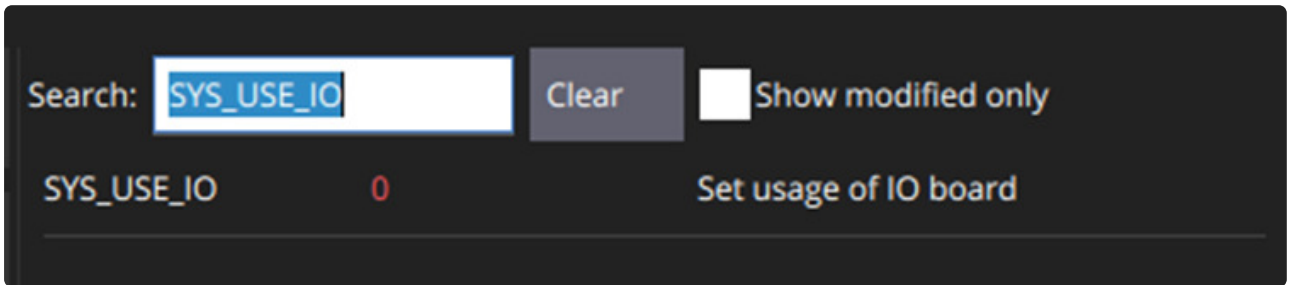


为确保安全，可在飞机上系上安全绳，并将安全绳的另一端固定在重物上。飞行时人在安全半径以外，在姿态模式下，高度可能比较难控，注意不要急推油门，让油门在中位附近，缓慢推油门。

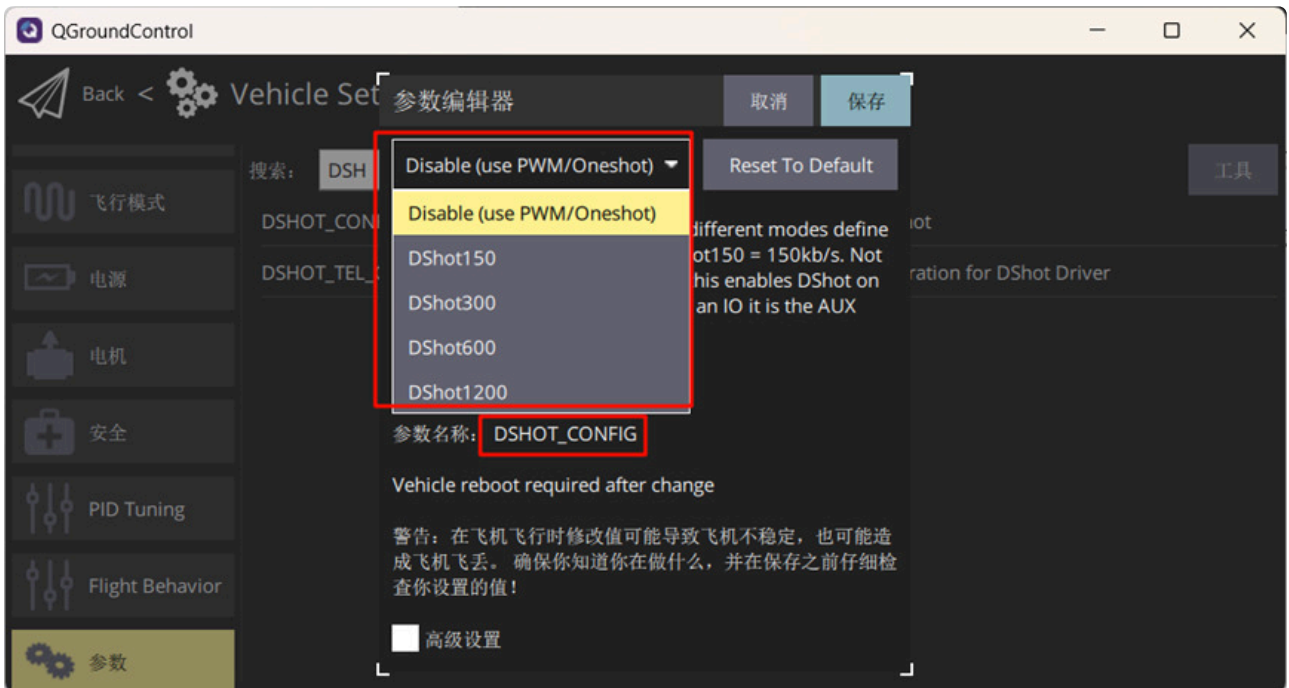


4.6 DShot模式下实飞

DSHOT是一种数字电调协议，广泛用于无人机飞控系统（如PX4、Betaflight、iNav等），相比传统的PWM协议具有低延迟、抗干扰、免校准等优势。对于只有MAIN端口的Pixhawk飞行控制器：将DSHOT ESC连接到MAIN端口，参考你的机架（airframe）配置进行接线。对于同时具有AUX和MAIN端口的Pixhawk：DSHOT只能在FMU端口（通常标为AUX）上使用。需要设置SYS_USE_IO=0，使AUX端口作为MAIN端口使用，并将ESC连接到AUX端口。



修改DSHOT_CONFIG来启用DSHOT，默认情况下该参数是使用PWM，我们只需要选择对应电调的协议就等于启动了DSHOT。设置为ESC支持的最高速度选项（例如：DSHOT150、DSHOT300、DSHOT600或DSHOT1200，单位为千比特/秒）。



若是需要基于DSHOT模式下来开发飞行控制算法时，PWM_output模块、AUX_output模块则无法使用，可直接将控制器的输出输出到力和力矩模型（如：PSP工具箱中的TorqueThrustCtrls模块）来控制电机。可见本实验中的 `Exp6_AttitudeSystemCodeGen3.slx` 程序。该模型中直接将控制器的输出信号直接发布，在PX4软件中驱动电机的转动。具体实验步骤与实飞实验中的步骤相同。

5. 关键知识点

关键知识点1：姿态控制器设计仿真-实验原理

控制器和多旋翼模型的参数都存储在一个初始化脚本 “5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Init_control.m” 和 “5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\icon\Init.m” 中。Init_control.m 这个脚本在开始 Simulink 仿真时需要初始化运行,以将所有参数导入工作空间,保证仿真正常运行。下面是对 Init_control.m 脚本的解析:

清除 MATLAB 工作区和命令窗口中的内容,以确保开始时是一个干净的状态。

```
1 | clear
2 | clc
```

将当前目录下的 ./icon/ 文件夹添加到 MATLAB 的搜索路径中。

```
1 | path(path, './icon/');
2 | Init;
```

比例因子。

```
1 | kFactor = 1.2;
```

滚转角度、俯仰角度、滚转角速率、俯仰角速率和偏航角速率控制器的比例增益和积分增益。

```
1 | KpRollAttitude = 0.630138625117681 / kFactor;
2 | KiRollAttitude = 0.0552303199591522 / kFactor;
3 | KpPitchAttitude = 0.630138625117681 / kFactor;
4 | KiPitchAttitude = 0.0552303199591522 / kFactor;
5 | KpRollRate = 1.31533836563383 / kFactor;
6 | KiRollRate = 0.460338699849681 / kFactor;
7 | KpPitchRate = 1.31533836563383 / kFactor;
8 | KiPitchRate = 0.460338699849681 / kFactor;
9 | KpYawRate = 1.31533836563383 / kFactor * 2;
10 | KiYawRate = 0.460338699849681 / kFactor * 2;
11 | KpThrust=0.6;
```

Init.m代码如下:

PID参数。

```

1 | Kp_RP_ANGLE =6.5;
2 | Kp_RP_AnggleRate = 0.10;
3 | Ki_RP_AnggleRate = 0.02;
4 | Kd_RP_AnggleRate = 0.001;
5 | Kp_YAW_AngleRate = 0.3;
6 | Ki_YAW_AngleRate = 0.1;
7 | Kd_YAW_AngleRate = 0.00;

```

滚转、俯仰和偏航角速率控制器的积分项上下限。

```

1 | Saturation_I_RP_Max=0.3;
2 | Saturation_I_RP_Min=-0.3;
3 | Saturation_I_Y_Max=0.2;
4 | Saturation_I_Y_Min=-0.2;

```

四旋翼悬停的油门值。

```

1 | THR_HOVER=0.609;

```

最大控制角度和角速度。

```

1 | MAX_CONTROL_ANGLE_ROLL = 35;
2 | MAX_CONTROL_ANGLE_PITCH = 35;
3 | %max control angle rate
4 | MAX_CONTROL_ANGLE_RATE_PITCH = 220;
5 | MAX_CONTROL_ANGLE_RATE_ROLL = 220;
6 | MAX_CONTROL_ANGLE_RATE_Y = 200;

```

多旋翼飞行器初始状态。

```

1 | ModelInit_PosE=[0,0,0];
2 | ModelInit_VelB=[0,0,0];
3 | ModelInit_AngEuler=[0,0,0];
4 | ModelInit_RateB=[0,0,0];
5 | ModelInit_RPM = 0; %Initial motor speed (rad/s)
6 | ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0];

```

多旋翼构型，3表示X型四旋翼；电机数量

```

1 | ModelParam_uavType = int8(3);
2 | ModelParam_uavMotNumbs = int8(4);
3 | %ModelParam_ControlMode = int8(1);
4 | ModelParam_motorMinThr=0.05;
5 | ModelParam_motorCr=842.1;
6 | ModelParam_motorWb=22.83;
7 | ModelParam_motorT= 0.0214;%0.0261;
8 | ModelParam_motorJm =0.0001287;
9 | ModelParam_rotorCm=2.783e-07;
10 | ModelParam_rotorCt=1.681e-05;
11 | %ModelParam_uavCtrlEn = int8(0);
12 | ModelParam_uavCd = 0.055;
13 | ModelParam_uavCCm = [0.0035 0.0039 0.0034];
14 | ModelParam_uavMass=1.515;%%1.5;
15 | ModelParam_uavDearo = 0.12;%%unit m
16 | ModelParam_uavR=0.225;
17 | ModelParam_uavJxx = 0.0211;%%1.491E-2;
18 | ModelParam_uavJyy = 0.0219;%%1.491E-2;
19 | ModelParam_uavJzz = 0.0366;%%2.712E-2;
20 | ModelParam_uavJ=
21 | [ModelParam_uavJxx,0,0;0,ModelParam_uavJyy,0;0,0,ModelParam_uavJzz];
22 | ModelParam_GlobalNoiseGainSwitch =0.4;
23 | %Environment Parameter
24 | ModelParam_envLongitude = 116.259368300000;
25 | ModelParam_envLatitude = 40.1540302;
26 | ModelParam_GPSLatLong = [ModelParam_envLatitude ModelParam_envLongitude];
   | ModelParam_envAltitude = -50;

```

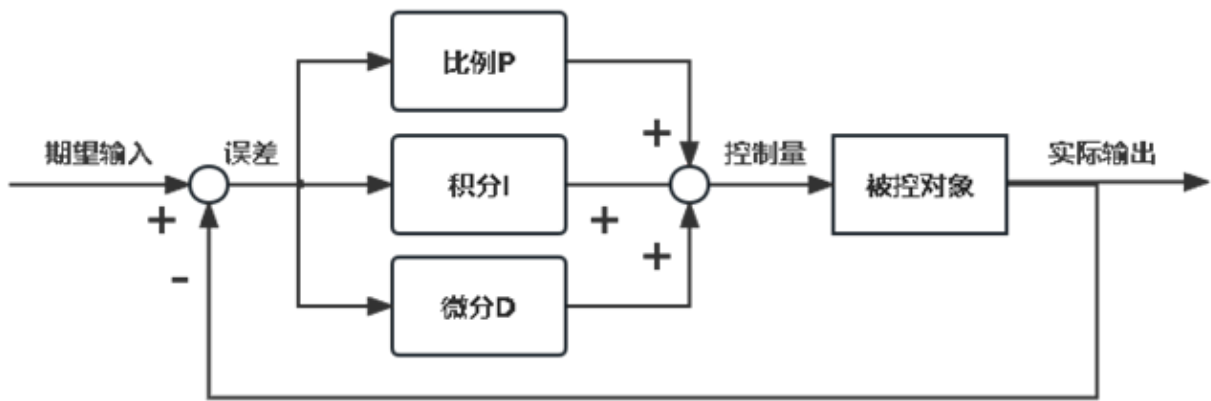
关键知识点2：姿态控制器设计转台-实验原理

本实验的关键内容分为两个部分，一部分是MAVLINK通信，一部分是控制器设计。本实验中MAVLINK协议用于飞控与上位机之间进行通信，一方面MATLAB发送的控制指令通过MAVLINK协议传递给飞控。另一方面飞控将关键状态通过MAVLINK协议传递给MATLAB进行显示，便于调试和分析。控制器则采用经典的PID控制器，为了让控制器更加安全，有必要对各个通道的计算结果进行限幅。

MAVLink（Micro Air Vehicle Link）是一种用于小型无人载具的通信协议，于2009年首次发布。该协议广泛应用于地面站（Ground Control Station, GCS）与无人载具

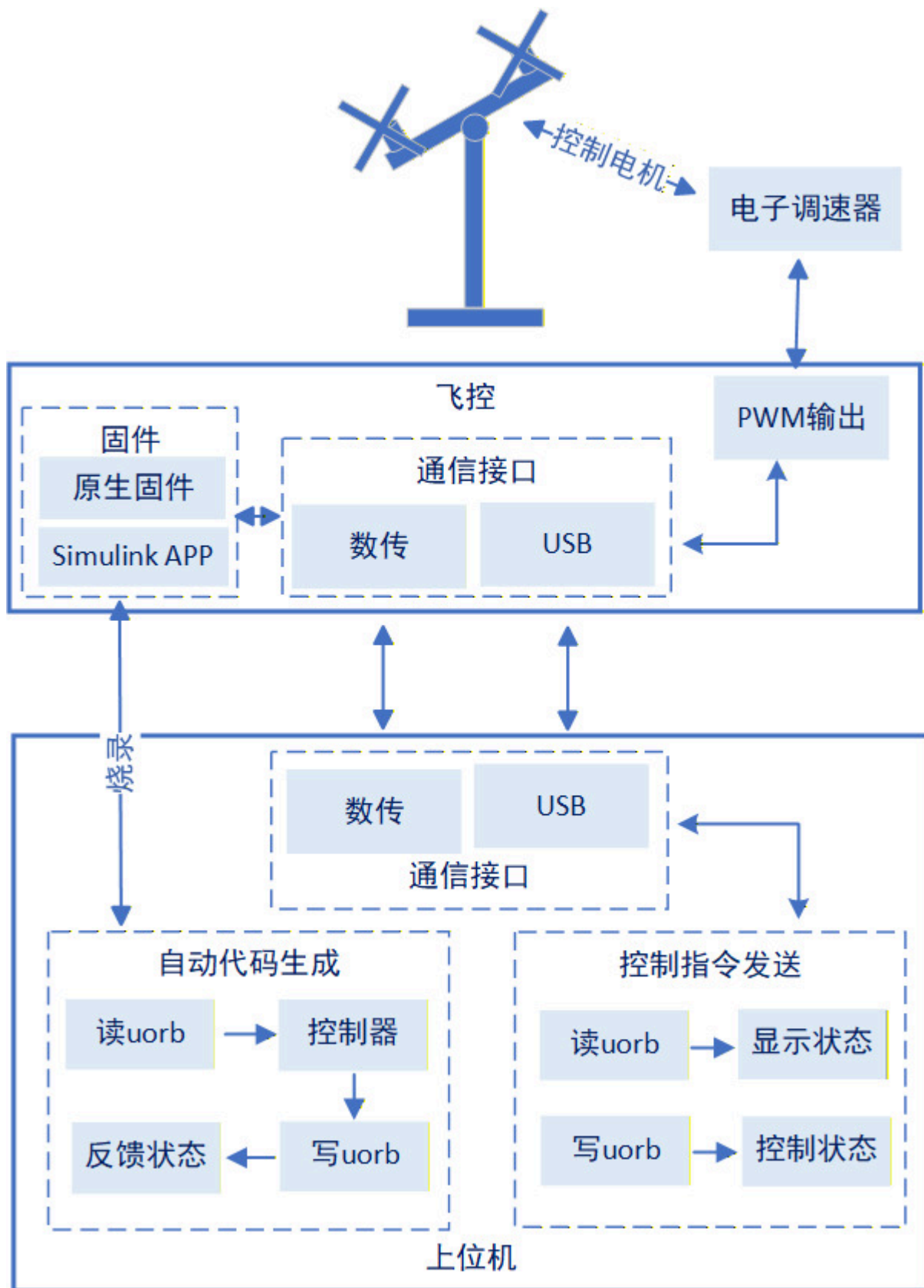
（Unmanned vehicles）之间的通信，同时也应用在载具上机载计算机与Pixhawk之间的内部通信中，协议以消息库的形式定义了参数传输的规则。MAVLink协议支持无人固定翼飞行器、无人旋翼飞行器、无人车辆等多种载具。

控制器则采用应用最为广泛的PID控制器。PID控制算法是由比例 P（Proportion）、积分 I（Integration）、微分 D（Differentiation）组成的一种控制算法。PID控制就是将变量偏差的比例、积分和微分之和作为控制输入，计算输出值的一种控制方法。传统 PID 控制系统的结构框图如图所示。



图中的比例“P”参数起到矫正偏差的作用，P参数越大对误差的灵敏度越高，误差的矫正速度越快，从而系统的动作也越快，但是过大的P参数会使系统出现过大的超调。积分“I”参数可以起到提高精度的作用，I越大跟踪误差越小，系统跟踪越精确。微分“D”参数可以起到抑制超调的作用，可以反应误差信号的变化率，能在跟踪快要达到期望目标时提前减速，一般D参数不会太大，过大的D参数会引起系统震荡。

PID控制算法被广泛应用，因为其结构简单、稳定性好、可靠性高且调节方便而成为工业控制的主要技术之一。其最大的优势在于不需要对系统建模或者系统辨识，只需设计完成后在实验中反复调节即可。对于一般的无人系统其建模复杂，所以PID控制算法使用频率很高。



本实验的总体框架如上图所示，在上位机上使用PSP工具箱设计MATLAB控制器。MATLAB控制器可以直接编译成可在飞控上使用的Simulink APP。将Simulink APP烧录到飞控当中，控制器即可计算控制律并对转台上的无人机进行控制。在上位机上，可以通过Simulink读取飞控的状态或者发送控制指令给飞控。上位机与飞控的物理连接可以通过数传或者USB，数传连接时通信速率较慢部分消息不可用，而USB连接时通信速率高，可以显示所有信息。

控制器和多旋翼模型的参数都存储在一个初始化脚本

本"5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Init_control.m"和"5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\icon\Init.m"中。 [Init_control.m](#)这个脚本在开始 Simulink 仿真时需要初始化运行,以将所有参数导入工作空间,保证仿真正常运行。下面是对 [Init_control.m](#) 脚本的解析:

首先,清除 MATLAB 工作区和命令窗口中的内容,以确保开始时是一个干净的状态。

```
1 | clear
2 | clc
```

将当前目录下的 ./icon/ 文件夹添加到 MATLAB 的搜索路径中。

```
1 | path(path, './icon/');
2 | Init;
```

比例因子,用于统一调整控制器参数。

```
1 | kFactor = 1.2;
```

滚转角度、俯仰角度、滚转角速率、俯仰角速率和偏航角速率控制器的比例增益和积分增益。这些参数决定了控制器的响应速度和稳定性,是整个姿态控制系统的核心。

```
1 | KpRollAttitude = 0.630138625117681 / kFactor; % 滚转角度比例系数
2 | KiRollAttitude = 0.0552303199591522 / kFactor; % 滚转角度积分系数
3 | KpPitchAttitude = 0.630138625117681 / kFactor; % 俯仰角度比例系数
4 | KiPitchAttitude = 0.0552303199591522 / kFactor; % 俯仰角度积分系数
5 | KpRollRate = 1.31533836563383 / kFactor; % 滚转角速率比例系数
6 | KiRollRate = 0.460338699849681 / kFactor; % 滚转角速率积分系数
7 | KpPitchRate = 1.31533836563383 / kFactor; % 俯仰角速率比例系数
8 | KiPitchRate = 0.460338699849681 / kFactor; % 俯仰角速率积分系数
9 | KpYawRate = 1.31533836563383 / kFactor * 2; % 偏航角速率比例系数
10 | KiYawRate = 0.460338699849681 / kFactor * 2; % 偏航角速率积分系数
11 | KpThrust=0.6; % 推力控制比例系数
```

[Init.m](#)代码如下,这是另一个重要的参数文件:

PID参数,这些是控制算法的核心参数,用于调整控制精度和稳定性。

```

1 | Kp_RP_ANGLE =6.5;           % 滚转和俯仰角度比例系数
2 | Kp_RP_AngleRate = 0.10;    % 滚转和俯仰角速率比例系数
3 | Ki_RP_AngleRate = 0.02;    % 滚转和俯仰角速率积分系数
4 | Kd_RP_AngleRate = 0.001;   % 滚转和俯仰角速率微分系数
5 | Kp_YAW_AngleRate = 0.3;    % 偏航角速率比例系数
6 | Ki_YAW_AngleRate = 0.1;    % 偏航角速率积分系数
7 | Kd_YAW_AngleRate = 0.00;   % 偏航角速率微分系数

```

滚转、俯仰和偏航角速率控制器的积分项上下限，用于防止积分饱和。

```

1 | Saturation_I_RP_Max=0.3;    % 滚转和俯仰积分项上限
2 | Saturation_I_RP_Min=-0.3;  % 滚转和俯仰积分项下限
3 | Saturation_I_Y_Max=0.2;    % 偏航积分项上限
4 | Saturation_I_Y_Min=-0.2;   % 偏航积分项下限

```

四旋翼悬停的油门值，这是维持无人机悬停所需的推力值。

```

1 | THR_HOVER=0.609;

```

最大控制角度和角速度，用于限制无人机的姿态变化范围，保证飞行安全。

```

1 | MAX_CONTROL_ANGLE_ROLL = 35;      % 最大滚转角度(度)
2 | MAX_CONTROL_ANGLE_PITCH = 35;    % 最大俯仰角度(度)
3 | %max control angle rate
4 | MAX_CONTROL_ANGLE_RATE_PITCH = 220; % 最大俯仰角速率(度/秒)
5 | MAX_CONTROL_ANGLE_RATE_ROLL = 220; % 最大滚转角速率(度/秒)
6 | MAX_CONTROL_ANGLE_RATE_Y = 200;  % 最大偏航角速率(度/秒)

```

多旋翼飞行器初始状态，定义了仿真开始时无人机的位置、速度和姿态。

```

1 | ModelInit_PosE=[0,0,0];          % 初始位置(北东地坐标系)
2 | ModelInit_VelB=[0,0,0];          % 初始速度(机体坐标系)
3 | ModelInit_AngEuler=[0,0,0];     % 初始姿态角(滚转、俯仰、偏航)
4 | ModelInit_RateB=[0,0,0];         % 初始角速率(机体坐标系)
5 | ModelInit_RPM = 0;               % 初始电机转速(rad/s)
6 | ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]; % 初始控制输入

```

多旋翼构型参数，定义了无人机的物理特性和动力学模型。

```

1 ModelParam_uavType = int8(3);           % 无人机类型, 3表示X型四旋翼
2 ModelParam_uavMotNumbs = int8(4);      % 电机数量
3 %ModelParam_ControlMode = int8(1);
4 ModelParam_motorMinThr=0.05;           % 最小油门值
5 ModelParam_motorCr=842.1;              % 电机阻力系数
6 ModelParam_motorWb=22.83;              % 电机带宽参数
7 ModelParam_motorT= 0.0214;             % 电机时间常数
8 ModelParam_motorJm =0.0001287;         % 电机转动惯量
9 ModelParam_rotorCm=2.783e-07;          % 反扭矩系数
10 ModelParam_rotorCt=1.681e-05;         % 推力系数
11 %ModelParam_uavCtrlEn = int8(0);
12 ModelParam_uavCd = 0.055;              % 阻尼系数
13 ModelParam_uavCCm = [0.0035 0.0039 0.0034]; % 阻力矩系数
14 ModelParam_uavMass=1.515;              % 无人机质量(kg)
15 ModelParam_uavDearo = 0.12;            % 机臂长度(m)
16 ModelParam_uavR=0.225;                % 旋翼半径(m)
17 ModelParam_uavJxx = 0.0211;           % 绕X轴转动惯量
18 ModelParam_uavJyy = 0.0219;           % 绕Y轴转动惯量
19 ModelParam_uavJzz = 0.0366;           % 绕Z轴转动惯量
20 ModelParam_uavJ=
21 [ModelParam_uavJxx,0,0;0,ModelParam_uavJyy,0;0,0,ModelParam_uavJzz]; % 惯量矩阵
22 ModelParam_GlobalNoiseGainSwitch =0.4; % 噪声增益开关
23 %Environment Parameter
24 ModelParam_envLongitude = 116.259368300000; % 经度
25 ModelParam_envLatitude = 40.1540302;   % 纬度
26 ModelParam_GPSLatLong = [ModelParam_envLatitude ModelParam_envLongitude]; % GPS经纬度
    ModelParam_envAltitude = -50;          % 海拔高度(m)

```

5.2 姿态控制器设计转台-实验原理

无人机姿态控制的基础概念

本实验的关键内容分为两个部分，一部分是MAVLINK通信，一部分是控制器设计。本实验中MAVLINK协议用于飞控与上位机之间进行通信，一方面MATLAB发送的控制指令通过MAVLINK协议传递给飞控。另一方面飞控将关键状态通过MAVLINK协议传递给MATLAB进行显示，便于调试和分析。

对于零基础的学习者，我们需要先了解几个基本概念：

姿态角度：

- **滚转角 (Roll)：**无人机绕机身纵轴的旋转，就像翻滚动作
- **俯仰角 (Pitch)：**无人机绕机身横轴的旋转，就像点头动作
- **偏航角 (Yaw)：**无人机绕机身竖轴的旋转，就像摇头动作

控制原理：姿态控制器的目标是使无人机的实际姿态尽可能接近期望姿态，通过不断比较期望值和实际值，调整电机转速来实现姿态控制。

MAVLink通信协议详解

MAVLink (Micro Air Vehicle Link) 是一种用于小型无人载具的通信协议，于2009年首次发布。该协议广泛应用于地面站 (Ground Control Station, GCS) 与无人载具 (Unmanned vehicles) 之间的通信，同时也应用在载具上机载计算机与Pixhawk之间的内部通信中，协议以消息库的形式定义了参数传输的规则。MAVLink协议支持无人固定翼飞行器、无人旋翼飞行器、无人车辆等多种载具。

MAVLink协议的重要性在于：

1. **标准化：**提供了一套统一的消息格式，不同厂家的设备可以相互通信
2. **高效性：**压缩的消息格式，减少传输时间
3. **可靠性：**具有错误检测机制，确保数据传输的准确性

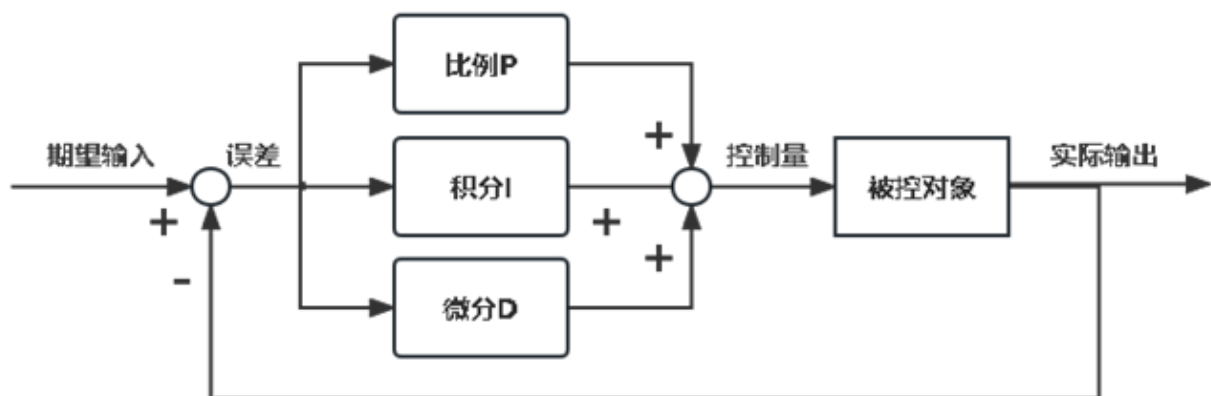
PID控制器原理

控制器则采用应用最为广泛的PID控制器。PID 控制算法是由比例 P (Proportion)、积分 I (Integration)、微分 D (Differentiation) 组成的一种控制算法。PID控制就是将变量偏差的比例、积分和微分之和作为控制输入，计算输出值的一种控制方法。

零基础理解PID：

- **比例 (P) 项：**像一位"反应迅速的司机"，误差越大，纠正力度越大
- **积分 (I) 项：**像一位"有记忆的助手"，持续累积过去的误差，消除长期偏差
- **微分 (D) 项：**像一位"预见未来的专家"，根据变化趋势提前调整，防止过度反应

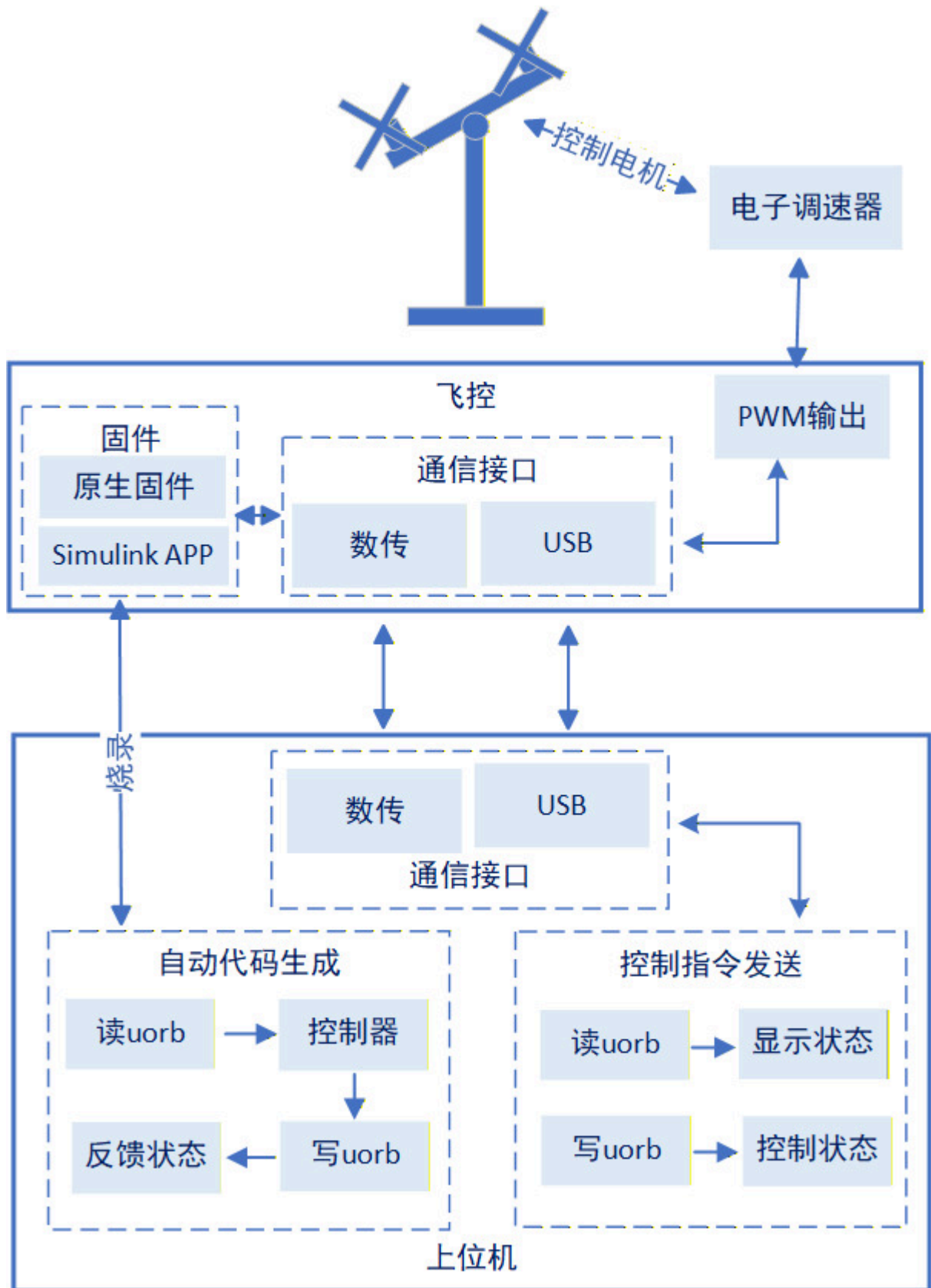
传统 PID 控制系统的结构框图如图所示。



PID参数的含义：

- **比例"P"参数**：起到矫正偏差的作用，P 参数越大对误差的灵敏度越高，误差的矫正速度越快，从而系统的动作也越快，但是过大的 P 参数会使系统出现过大的超调
- **积分"I"参数**：可以起到提高精度的作用，I 越大跟踪误差越小，系统跟踪越精确
- **微分"D"参数**：可以起到抑制超调的作用，可以反应误差信号的变化率，能在跟踪快要达到期望目标时提前减速，一般D参数不会太大，过大的D参数会引起系统震荡

PID 控制算法被广泛应用，因为其结构简单、稳定性好、可靠性高且调节方便而成为工业控制的主要技术之一。其最大的优势在于不需要对系统建模或者系统辨识，只需设计完成后在实验中反复调节即可。对于一般的无人系统其建模复杂，所以 PID控制算法使用频率很高。



本实验的总体框架如上图所示，在上位机上使用PSP工具箱设计MATLAB控制器。MATLAB控制器可以直接编译成可在飞控上使用的Simulink APP。将Simulink APP烧录到飞控当中，控制器即可计算控制律并对转台上的无人机进行控制。在上位机上，可以通过Simulink读取飞控的状态或者发送控制指令给飞控。上位机与飞控的物理连接可以通过数传或者USB，数传连接时通信速率较慢部分消息不可用，而USB连接时通信速率高，可以显示所有信息。

转台实验的参数设置

控制器和多旋翼模型的参数都存储在一个初始化脚本

本"5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp4.5_Rotarytable\Init_control.m"和"5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\Exp4.5_Rotarytable\icon\Init.m"中。

[Init_control.m](#) 这个脚本在开始 Simulink 仿真时需要初始化运行,以将所有参数导入工作空间,保证仿真正常运行。

以下是 [Init_control.m](#) 中重要参数的说明:

俯仰角的比例参数和俯仰角速率的比例、积分和微分参数:

```
1 | Kp_PITCH_ANGLE = 5;           % 俯仰角度比例系数
2 | Kp_PITCH_AngleRate = 0.1;   % 俯仰角速率比例系数
3 | Ki_PITCH_AngleRate = 0.05;  % 俯仰角速率积分系数
4 | Kd_PITCH_AngleRate = 0.0001; % 俯仰角速率微分系数
```

滚转角的比例参数和滚转角速率的比例、积分和微分参数:

```
1 | Kp_ROLL_ANGLE = 5;           % 滚转角度比例系数
2 | Kp_ROLL_AngleRate = 0.1;    % 滚转角速率比例系数
3 | Ki_ROLL_AngleRate = 0.05;   % 滚转角速率积分系数
4 | Kd_ROLL_AngleRate = 0.0001; % 滚转角速率微分系数
```

偏航角的比例参数和偏航角速率的比例、积分和微分参数:

```
1 | Kp_YAW_ANGLE = 2;           % 偏航角度比例系数
2 | Kp_YAW_AngleRate = 0.1;    % 偏航角速率比例系数
3 | Ki_YAW_AngleRate = 0.01;   % 偏航角速率积分系数
4 | Kd_YAW_AngleRate = 0.00;   % 偏航角速率微分系数
```

滚转、俯仰和偏航角速率控制器的积分项上下限:

```
1 | Saturation_I_RP_Max = 0.2;  % 滚转、俯仰积分项上限
2 | Saturation_I_RP_Min = -0.2; % 滚转、俯仰积分项下限
3 | Saturation_I_Y_Max = 0.1;  % 偏航积分项上限
4 | Saturation_I_Y_Min = -0.1; % 偏航积分项下限
```

最大角速率限制, 单位rad/s:

```
1 | MAX_CONTROL_ANGLE_RATE_PITCH = 5; % 最大俯仰角速率
2 | MAX_CONTROL_ANGLE_RATE_ROLL = 5;  % 最大滚转角速率
3 | MAX_CONTROL_ANGLE_RATE_YAW = 3;  % 最大偏航角速率
```

启动Rotarytable_Quadrotor_Control.slx模型：

```
1 | Rotarytable_Quadrotor_Control
```

Exp4.5_Rotarytable\icon\Init.m代码如下：

PID参数：

```
1 | Kp_RP_ANGLE =6.5;           % 滚转和俯仰角度比例系数
2 | Kp_RP_AngleRate = 0.10;    % 滚转和俯仰角速率比例系数
3 | Ki_RP_AngleRate = 0.02;    % 滚转和俯仰角速率积分系数
4 | Kd_RP_AngleRate = 0.001;   % 滚转和俯仰角速率微分系数
5 | Kp_YAW_AngleRate = 0.3;    % 偏航角速率比例系数
6 | Ki_YAW_AngleRate = 0.1;    % 偏航角速率积分系数
7 | Kd_YAW_AngleRate = 0.00;   % 偏航角速率微分系数
```

滚转、俯仰和偏航角速率控制器的积分项上下限：

```
1 | Saturation_I_RP_Max=0.3;    % 滚转、俯仰积分项上限
2 | Saturation_I_RP_Min=-0.3;  % 滚转、俯仰积分项下限
3 | Saturation_I_Y_Max=0.2;    % 偏航积分项上限
4 | Saturation_I_Y_Min=-0.2;   % 偏航积分项下限
```

四旋翼悬停的油门值：

```
1 | THR_HOVER=0.609;
```

最大控制角度和角速度：

```
1 | MAX_CONTROL_ANGLE_ROLL = 35;      % 最大滚转角度
2 | MAX_CONTROL_ANGLE_PITCH = 35;    % 最大俯仰角度
3 | %max control angle rate
4 | MAX_CONTROL_ANGLE_RATE_PITCH = 220; % 最大俯仰角速率
5 | MAX_CONTROL_ANGLE_RATE_ROLL = 220; % 最大滚转角速率
6 | MAX_CONTROL_ANGLE_RATE_Y = 200;  % 最大偏航角速率
```

多旋翼飞行器初始状态：

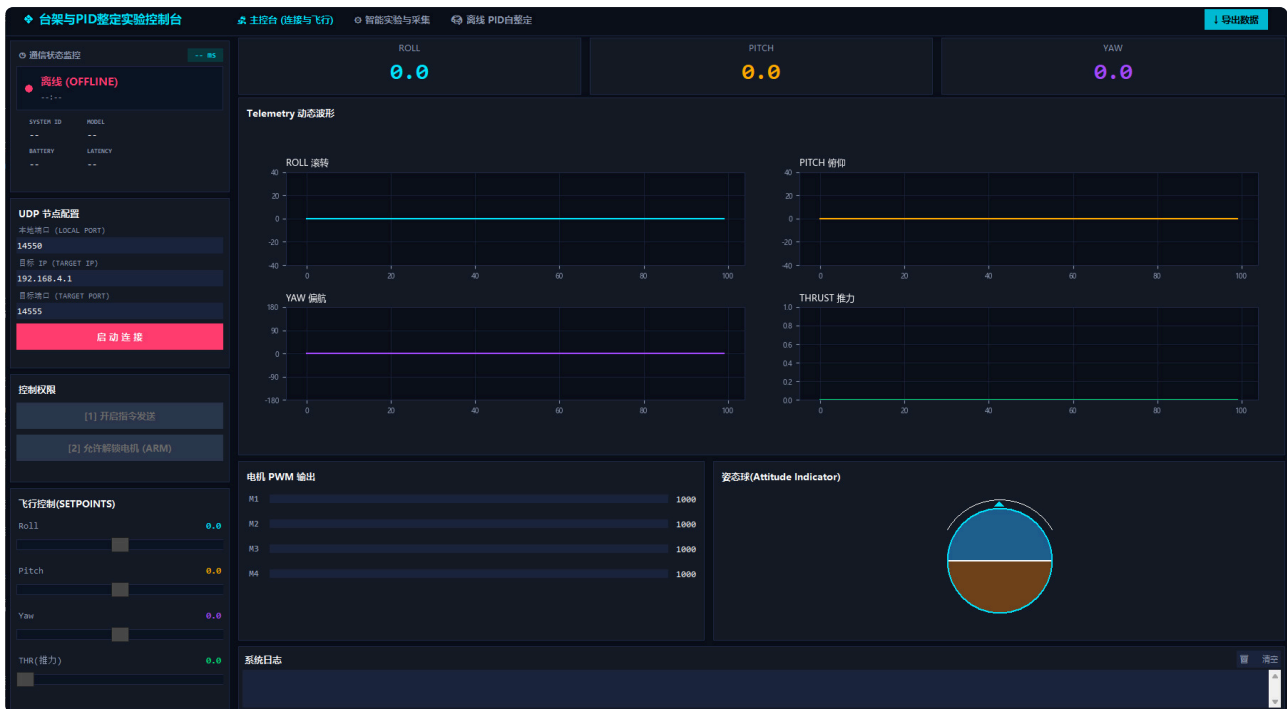
```
1 | ModelInit_PosE=[0,0,0];          % 初始位置
2 | ModelInit_VelB=[0,0,0];          % 初始速度
3 | ModelInit_AngEuler=[0,0,0];      % 初始姿态角
4 | ModelInit_RateB=[0,0,0];         % 初始角速率
5 | ModelInit_RPM = 0;                % 初始电机转速
6 | ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]; % 初始控制输入
```

多旋翼构型参数：

```
1 ModelParam_uavType = int8(3);           % 3表示X型四旋翼
2 ModelParam_uavMotNumbs = int8(4);       % 电机数量
3 %ModelParam_ControlMode = int8(1);
4 ModelParam_motorMinThr=0.05;           % 最小油门
5 ModelParam_motorCr=842.1;              % 电机阻力系数
6 ModelParam_motorWb=22.83;              % 电机带宽
7 ModelParam_motorT= 0.0214;             % 电机时间常数
8 ModelParam_motorJm =0.0001287;         % 电机转动惯量
9 ModelParam_rotorCm=2.783e-07;          % 反扭矩系数
10 ModelParam_rotorCt=1.681e-05;         % 推力系数
11 %ModelParam_uavCtrlEn = int8(0);
12 ModelParam_uavCd = 0.055;              % 阻尼系数
13 ModelParam_uavCCm = [0.0035 0.0039 0.0034]; % 阻力矩系数
14 ModelParam_uavMass=1.515;              % 无人机质量
15 ModelParam_uavDearo = 0.12;           % 机臂长度
16 ModelParam_uavR=0.225;                 % 旋翼半径
17 ModelParam_uavJxx = 0.0211;           % 绕X轴转动惯量
18 ModelParam_uavJyy = 0.0219;          % 绕Y轴转动惯量
19 ModelParam_uavJzz = 0.0366;           % 绕Z轴转动惯量
20 ModelParam_uavJ=
21 [ModelParam_uavJxx,0,0;0,ModelParam_uavJyy,0;0,0,ModelParam_uavJzz]; % 惯量矩阵
22 ModelParam_GlobalNoiseGainSwitch =0.4; % 全局噪声增益开关
23 %Environment Parameter
24 ModelParam_envLongitude = 116.259368300000; % 经度
25 ModelParam_envLatitude = 40.1540302;   % 纬度
26 ModelParam_GPSLatLong = [ModelParam_envLatitude ModelParam_envLongitude]; % GPS经纬度
    ModelParam_envAltitude = -50;         % 海拔高度
```

Python控制台界面操作说明

双击3.DesignExps\Python38Run.bat脚本，在弹出的对话框中输入：`python control-PID.py`，启动"台架与PID整定实验控制台"软件。整体界面如下图所示。控制指令与参数设置集中在左侧，遥测数据与动态波形的显示集中在右侧。



下面详细介绍控制台左侧控制权限与期望设定的功能。如下图所示，是主控端的两个关键控制开关，排在上面的是"[1] 开启指令发送"按钮，排在下面的是"[2] 允许解锁电机 (ARM)"按钮。



当"[1] 开启指令发送"处于未开启（灰色）状态时，主控台上的任何姿态修改都不起作用，即不会向飞控发送任何控制指令流。点击开启后，该按钮将变为深色，系统会瞬间记录当前物理坐标作为保护原点，"[2] 允许解锁电机 (ARM)"按钮将随之激活变亮，同时下方的滚转、俯仰、偏航及油门的滑杆设置也将正式生效。

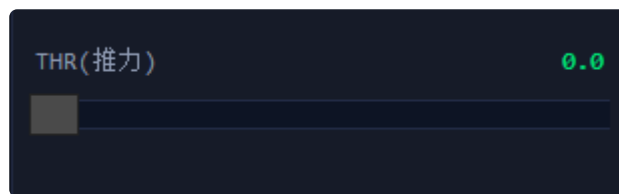
"[2] 允许解锁电机 (ARM)"按钮用于控制飞控进入 OFFBOARD 模式并解锁电机。点击后，飞控会发出滴答声，电机进入怠速状态。如果实验中出现异常，随时再次点击该按钮（此时显示为红色的"紧急上锁"），即可瞬间切断动力。

警告：电机解锁前，应仔细检查下方 THR (油门/推力)、Roll (滚转)、Pitch (俯仰)、Yaw (偏航) 滑杆的设置，否则可能因起步指令过大导致飞机剧烈撞击台架或造成人员受伤。

建议：首次实验时，THR 滑块设定为最左侧最低值（1100），其余姿态期望均保持在 0 位。待电机解锁并平稳怠速后，再缓慢推起油门。



油门值设置。如上图的 THR(推力) 滑杆，可以设置油门值。油门值可以理解为发送给飞控底层的推力百分比（映射为 PWM 脉宽）。滑杆范围被安全限制在 1100-1900。当滑杆打到最左侧时为最低值 1100（怠速），当打到最右侧时为最大值 1900（全油门）。在做转台上 V200 飞机姿态控制实验时，建议将油门缓慢推至 1400-1550 之间，使飞机绳索恰好拉直。



期望姿态设置。在滑杆区域可以设置期望的 Roll (滚转角)、Pitch (俯仰角) 和 Yaw (偏航角)。为保障台架安全，滚转和俯仰角的范围被硬性限制在 $-30^{\circ}\sim 30^{\circ}$ ，在开始手动调试时建议操作幅度不要超过 15° 。偏航角支持 $-180^{\circ}\sim 180^{\circ}$ ，偏航角可随意设置，对台架系统稳定性影响较小。



在主控台的右侧区域，系统提供了丰富的实时反馈。"Telemetry 动态波形"模块会以折线图的形式，高频刷新并显示当前飞机四轴的实际物理响应状态。而在右下方的"姿态球 (Attitude Indicator)"中，可以直观地看到飞机当前的倾斜姿态。最下方的"电机 PWM 输出"柱状图则展示了四个电机的实时转速脉宽，方便我们验证控制逻辑是否正常下发到了动力层。



5.3 姿态控制器设计实飞-实验原理

实飞参数配置

控制器和多旋翼模型的参数都存储在一个初始化脚本"[安装目录]5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\FLY_X450\Init_control.m"和"[安装目录]\5.RflySimFlyCtrl\0.ApiExps\3.DesignExps\FLY_X450\icon\Init.m"中。

[Init_control.m](#) 这个脚本在开始 Simulink 仿真时需要初始化运行,以将所有参数导入工作空间,保证仿真正常运行。

实飞实验的参数设置与仿真和转台实验类似,但考虑到实际飞行的安全性,某些参数会有所调整。这些参数经过了大量的仿真和测试,以确保飞行的稳定性和安全性。

清除 MATLAB 工作区和命令窗口中的内容,以确保开始时是一个干净的状态:

```
1 | clear
2 | clc
```

将当前目录下的 ./icon/ 文件夹添加到 MATLAB 的搜索路径中:

```
1 | path(path, './icon/');
2 | Init;
```

比例因子:

```
1 | kFactor = 1.2;
```

滚转角度、俯仰角度、滚转角速率、俯仰角速率和偏航角速率控制器的比例增益和积分增益：

```
1 | KpRollAttitude = 0.630138625117681 / kFactor; % 滚转角度比例系数
2 | KiRollAttitude = 0.0552303199591522 / kFactor; % 滚转角度积分系数
3 | KpPitchAttitude = 0.630138625117681 / kFactor; % 俯仰角度比例系数
4 | KiPitchAttitude = 0.0552303199591522 / kFactor; % 俯仰角度积分系数
5 | KpRollRate = 1.31533836563383 / kFactor; % 滚转角速率比例系数
6 | KiRollRate = 0.460338699849681 / kFactor; % 滚转角速率积分系数
7 | KpPitchRate = 1.31533836563383 / kFactor; % 俯仰角速率比例系数
8 | KiPitchRate = 0.460338699849681 / kFactor; % 俯仰角速率积分系数
9 | KpYawRate = 1.31533836563383 / kFactor * 2; % 偏航角速率比例系数
10 | KiYawRate = 0.460338699849681 / kFactor * 2; % 偏航角速率积分系数
11 | KpThrust=0.6; % 推力控制比例系数
```

FLY_X450\icon\Init.m代码如下：

PID参数：

```
1 | Kp_RP_ANGLE =6.5; % 滚转和俯仰角度比例系数
2 | Kp_RP_AngleRate = 0.10; % 滚转和俯仰角速率比例系数
3 | Ki_RP_AngleRate = 0.02; % 滚转和俯仰角速率积分系数
4 | Kd_RP_AngleRate = 0.001; % 滚转和俯仰角速率微分系数
5 | Kp_YAW_AngleRate = 0.3; % 偏航角速率比例系数
6 | Ki_YAW_AngleRate = 0.1; % 偏航角速率积分系数
7 | Kd_YAW_AngleRate = 0.00; % 偏航角速率微分系数
```

滚转、俯仰和偏航角速率控制器的积分项上下限：

```
1 | Saturation_I_RP_Max=0.3; % 滚转、俯仰积分项上限
2 | Saturation_I_RP_Min=-0.3; % 滚转、俯仰积分项下限
3 | Saturation_I_Y_Max=0.2; % 偏航积分项上限
4 | Saturation_I_Y_Min=-0.2; % 偏航积分项下限
```

四旋翼悬停的油门值：

```
1 | THR_HOVER=0.609;
```

最大控制角度和角速度：

```

1 | MAX_CONTROL_ANGLE_ROLL = 35;      % 最大滚转角度
2 | MAX_CONTROL_ANGLE_PITCH = 35;    % 最大俯仰角度
3 | %max control angle rate
4 | MAX_CONTROL_ANGLE_RATE_PITCH = 220; % 最大俯仰角速率
5 | MAX_CONTROL_ANGLE_RATE_ROLL = 220; % 最大滚转角速率
6 | MAX_CONTROL_ANGLE_RATE_Y = 200;  % 最大偏航角速率

```

多旋翼飞行器初始状态:

```

1 | ModelInit_PosE=[0,0,0];          % 初始位置
2 | ModelInit_VelB=[0,0,0];        % 初始速度
3 | ModelInit_AngEuler=[0,0,0];    % 初始姿态角
4 | ModelInit_RateB=[0,0,0];       % 初始角速率
5 | ModelInit_RPM = 0;             % 初始电机转速
6 | ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0]; % 初始控制输入

```

多旋翼构型, 3表示X型四旋翼; 电机数量:

```

1 | ModelParam_uavType = int8(3);
2 | ModelParam_uavMotNumbs = int8(4);
3 | %ModelParam_ControlMode = int8(1);
4 | ModelParam_motorMinThr=0.05;
5 | ModelParam_motorCr=842.1;
6 | ModelParam_motorWb=22.83;
7 | ModelParam_motorT= 0.0214;%0.0261;
8 | ModelParam_motorJm =0.0001287;
9 | ModelParam_rotorCm=2.783e-07;
10 | ModelParam_rotorCt=1.681e-05;
11 | %ModelParam_uavCtrlEn = int8(0);
12 | ModelParam_uavCd = 0.055;
13 | ModelParam_uavCCm = [0.0035 0.0039 0.0034];
14 | ModelParam_uavMass=1.515;%%1.5;
15 | ModelParam_uavDearo = 0.12;%%unit m
16 | ModelParam_uavR=0.225;
17 | ModelParam_uavJxx = 0.0211;%%1.491E-2;
18 | ModelParam_uavJyy = 0.0219;%%1.491E-2;
19 | ModelParam_uavJzz = 0.0366;%%2.712E-2;
20 | ModelParam_uavJ=
21 | [ModelParam_uavJxx,0,0;0,ModelParam_uavJyy,0;0,0,ModelParam_uavJzz];
22 | ModelParam_GlobalNoiseGainSwitch =0.4;
23 | %Environment Parameter
24 | ModelParam_envLongitude = 116.259368300000;
25 | ModelParam_envLatitude = 40.1540302;
26 | ModelParam_GPSLatLong = [ModelParam_envLatitude ModelParam_envLongitude];
    ModelParam_envAltitude = -50;

```

I 6.参考资料

1. [RflySim官方文档](#) - RflySim无人机仿真平台官方文档，包含安装指南、使用教程和API参考。
2. [PX4官方用户指南](#) - PX4自动驾驶仪官方文档，涵盖飞控配置、参数调校和开发指南。
3. [MATLAB Simulink文档](#) - MATLAB Simulink建模与仿真工具官方文档。
4. [MAVLink协议文档](#) - MAVLink通信协议官方文档，包含消息定义和通信规范。
5. [QGroundControl用户指南](#) - QGroundControl地面站软件官方使用指南。
6. [PID控制器原理与应用](#) - PID控制器基本原理、参数整定方法和应用案例。
7. [DShot协议规范](#) - DShot数字电调协议技术规范和使用指南。

I 7.常见问题

I Q1: Simulink模型编译失败，提示"Build process completed unsuccessfully"

现象：在MATLAB中点击编译按钮后，编译过程报错，诊断对话框显示编译失败，无法生成可执行代码。

原因：

1. MATLAB版本不兼容，需要MATLAB2022B以上版本。
2. Simulink模型存在语法错误或配置问题。
3. 缺少必要的工具箱，如Embedded Coder、Simulink Coder等。
4. PX4工具链未正确安装或配置。

解决方案：

1. 确认MATLAB版本符合要求（2022B以上），检查许可证是否包含必要的工具箱。
2. 检查模型配置参数：打开Model Configuration Parameters，确认"System target file"设置为"ert.tlc"，"Hardware board"设置为"PX4 Pixhawk Series"。
3. 运行 `Init_control.m` 脚本初始化工作空间参数，确保所有变量已正确加载。
4. 查看详细错误信息：在诊断对话框中查看具体错误，根据提示修复模型或配置问题。

I Q2: 飞控无法通过USB连接电脑，设备管理器未识别到COM端口

现象：使用USB数据线连接飞控与电脑后，设备管理器中未显示对应的COM端口，QGroundControl无法连接飞控。

原因：

1. USB数据线故障或仅支持充电不支持数据传输。
2. 飞控USB接口损坏或接触不良。
3. 电脑缺少PX4飞控驱动程序。
4. 飞控未上电或电源故障。

解决方案：

1. 更换高质量的数据传输USB线，确保线缆支持数据传输。
2. 尝试不同的USB端口，避免使用USB集线器，直接连接电脑主板USB接口。
3. 安装PX4驱动程序：访问PX4官网下载并安装最新驱动程序，或使用Zadig工具安装WinUSB驱动。
4. 检查飞控电源：确保飞控通过USB或外部电源正常供电，LED指示灯应正常闪烁。
5. 在设备管理器中查看"通用串行总线控制器"下是否有未知设备，尝试手动更新驱动程序。

I Q3: 硬件在环仿真时，CopterSim日志未显示"GPS 3D fixed & EKF initialization finished"

现象：启动HITL仿真后，CopterSim软件下侧日志栏长时间未显示初始化完成信息，RflySim3D中的飞机无法正常响应控制指令。

原因：

1. 飞控与CopterSim通信失败，串口号配置错误。
2. 飞控固件版本与CopterSim不兼容。
3. 仿真环境初始化超时，传感器数据未正常注入。
4. QGroundControl未正确关闭，占用了通信端口。

解决方案：

1. 确认串口号：在HITLRun脚本中输入正确的飞控COM端口号，可在设备管理器中查看。
2. 检查固件版本：确保飞控中烧录的固件与CopterSim版本兼容，推荐使用PX4 1.14.3版本。
3. 重启仿真环境：关闭所有相关软件（QGC、CopterSim、RflySim3D），重新运行HITLRun脚本。
4. 关闭QGroundControl：确保QGC已完全退出，不会抢占通信端口。
5. 检查网络连接：确保CopterSim能正常访问仿真服务器，防火墙未阻止相关端口。

Q4：转台实验时，MATLAB无法通过数传连接飞控

现象：使用数传连接飞控后，MATLAB的Mav_Control_Quadrotor.slx模型无法发送控制指令，飞控状态无法读取。

原因：

1. 数传端口号配置错误，Simulink中串口模块设置的端口与实际不符。
2. 数传波特率不匹配，默认应为57600或115200。
3. QGroundControl未关闭，占用了数传端口。
4. 数传驱动未安装或硬件故障。

解决方案：

1. 确认端口号：在设备管理器中查看数传对应的COM端口，在Simulink的Serial Configuration模块中设置为相同端口。
2. 检查波特率：确保数传模块与Simulink中设置的波特率一致，通常为57600。
3. 关闭QGroundControl：完全退出QGC软件，释放串口资源。
4. 安装数传驱动：根据数传型号安装相应驱动程序，3DR X6数传驱动可在"RflySimAPIs\5.RflySimFlyCtrl\0.ApiExps\0.ResourcesFile"中找到。
5. 测试数传连接：使用串口调试工具测试数传是否能正常收发数据。

Q5：实飞时无人机解锁后电机不转或转速异常

现象：通过遥控器CH5通道解锁无人机后，电机不转动或转速不稳定，无法正常起飞。

原因：

1. 安全开关未解除，飞控处于安全锁定状态。
2. 油门位置过低，未达到解锁阈值。

3. 电机接线错误或电调校准问题。
4. 飞控参数配置错误，如电机混控器设置不正确。

解决方案：

1. 检查安全开关：确认遥控器CH5通道已正确映射为安全开关，开关处于解锁位置。
2. 调整油门位置：将油门摇杆置于最低位置后再解锁，解锁后缓慢推高油门。
3. 检查电机接线：确认电机顺序与飞控输出端口对应关系正确，参考机架配置图。
4. 校准电调：通过QGroundControl的"电机"标签页进行电调校准。
5. 检查飞控参数：确认 `MOT_SPIN_ARMED` 参数设置为1，允许解锁后电机怠速转动。

Q6: DShot模式下电机控制异常，电机不响应或响应延迟

现象：启用DShot协议后，电机无法正常控制，出现不响应、延迟或转速异常现象。

原因：

1. DShot协议配置错误，协议版本不匹配。
2. 电调不支持DShot协议或固件版本过旧。
3. 飞控AUX端口配置错误，DShot只能在FMU端口使用。
4. 接线错误或信号干扰。

解决方案：

1. 确认DShot配置：在QGroundControl中设置 `DSHOT_CONFIG` 参数，选择与电调匹配的协议版本（DShot150/300/600/1200）。
2. 更新电调固件：确保电调固件支持DShot协议，必要时刷写最新BLHeli或Bluejay固件。
3. 检查端口配置：对于具有AUX端口的Pixhawk，设置 `SYS_USE_I0=0` 使AUX端口作为FMU端口使用，将DShot ESC连接到AUX端口。
4. 检查接线：使用高质量信号线，避免过长线缆，确保信号地线连接良好。
5. 测试PWM模式：暂时切换回PWM模式，确认硬件连接正常，再切换回DShot模式。

Q7: PID控制器参数调校困难，无人机姿态不稳定

现象：无人机在飞行中出现振荡、超调或响应迟缓，调整PID参数后改善不明显。

原因：

1. PID参数初始值不合适，比例、积分、微分增益不匹配。
2. 传感器噪声影响控制性能。
3. 控制频率设置不当。
4. 未进行参数限幅，积分饱和导致控制异常。

解决方案：

1. 采用逐步调参法：先调P（比例）增益，使系统有快速响应但不振荡；再调D（微分）增益抑制超调；最后调I（积分）增益消除稳态误差。
2. 添加低通滤波器：在角速度反馈通道添加低通滤波器，减少传感器噪声影响。
3. 调整控制频率：确保控制器运行频率与传感器更新频率匹配，通常设置为250-500Hz。
4. 设置积分限幅：在PID控制器中添加积分限幅，防止积分饱和，参考 `Init_control.m` 中的 `Saturation_I_RP_Max/Min` 参数。
5. 使用调参工具：利用MATLAB的PID Tuner工具或RflySim提供的自动调参功能优化参数。

1. <https://rflysim.com/> ←

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

[^3]本实验演示所使用的遥控器为：天地飞ET10、配套接收器为：WFLY RF209S。遥控器相关配置见：<https://rflysim.com/doc/zh/B/3.1ET10.html> ←