



智能无人集群系统开发与实践

基于RflySim平台的全栈开发案例

第5讲 位姿控制与滤波估计



大纲

1. 实验平台配置

6. 扩展案例（完整版）

2. 关键接口介绍

7. 小结

3. 基础实验案例（免费版）

4. 进阶接口实验（个人版）

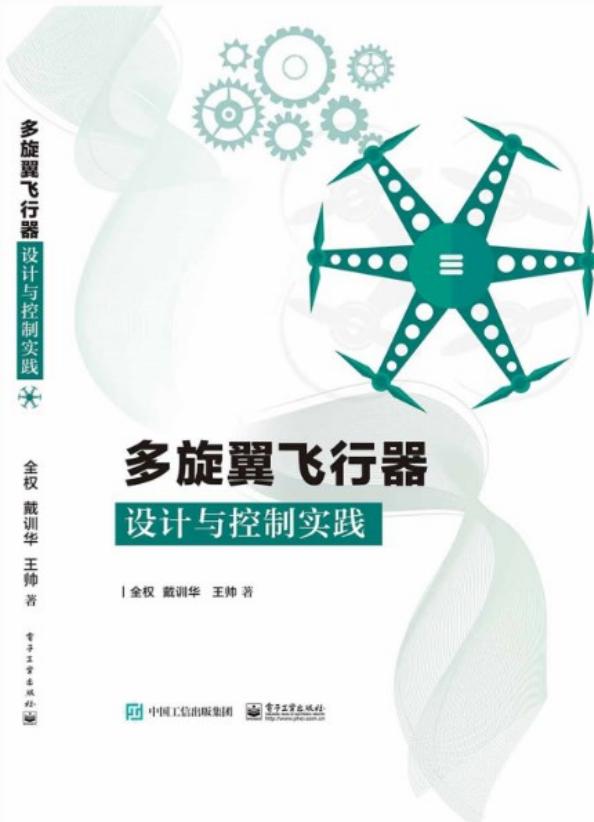
5. 进阶案例实验（集合版）



1.实验平台配置

1.1 参考教材

- 全权, 戴训华, 王帅著. 多旋翼飞行器设计与控制实践[M]. 北京:电子工业出版社. 2020



注:

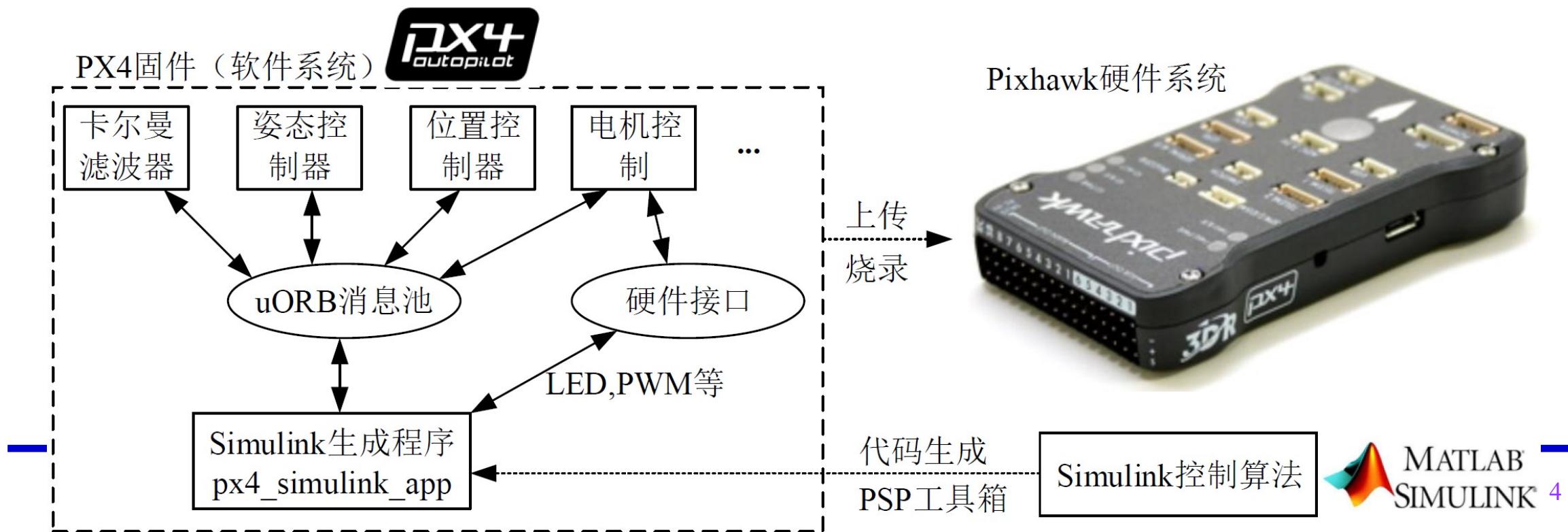
- 左侧的《多旋翼飞行器设计与控制实践》为2020年推出的针对飞行控制算法开发的实践课程，里面包含了部分理论知识与一系列实验，使得读者可以很快地将自己算法在Simulink中编程，并下载到Pixhawk真机中做飞行实验。
- 右侧的《多旋翼飞行器设计与控制》为2017年推出的教程，主要针对多旋翼控制理论。



1.实验平台配置

1.2 Pixhawk/PX4/Simulink代码生成平台构架

- Pixhawk是硬件（相当于电脑主机），PX4是飞控软件（相当于Windows操作系统），Simulink控制器生成代码后编译成固件（相当于系统iso镜像），上传到Pixhawk硬件（相当于重装系统），Simulink控制器以新线程（相当于电脑上第三方APP）独立于PX4官方控制器（相当于系统预装软件），并行运行。

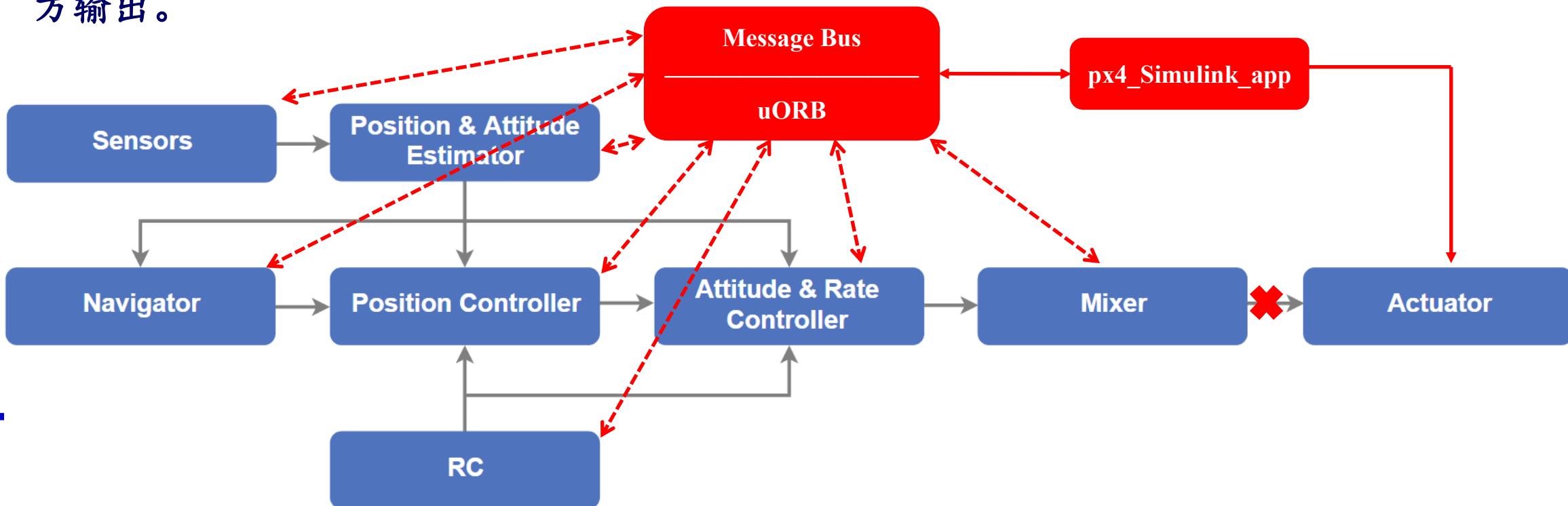




1.实验平台配置

1.3 为何屏蔽PX4输出

- PX4采用uORB发布与订阅消息机制，任何APP都可以从uORB消息池中获取和发布数据。
- Simulink代码生成到Pixhawk后生成名为px4_Simulink_app的一个APP，可以通过uORB消息池机制与其他APP通信。
- Simulink_App不能和PX4控制器同时访问电机，不然会产生冲突，因此需要屏蔽PX4官方输出。





1.实验平台配置

1.4 Simulink自动代码生成配置

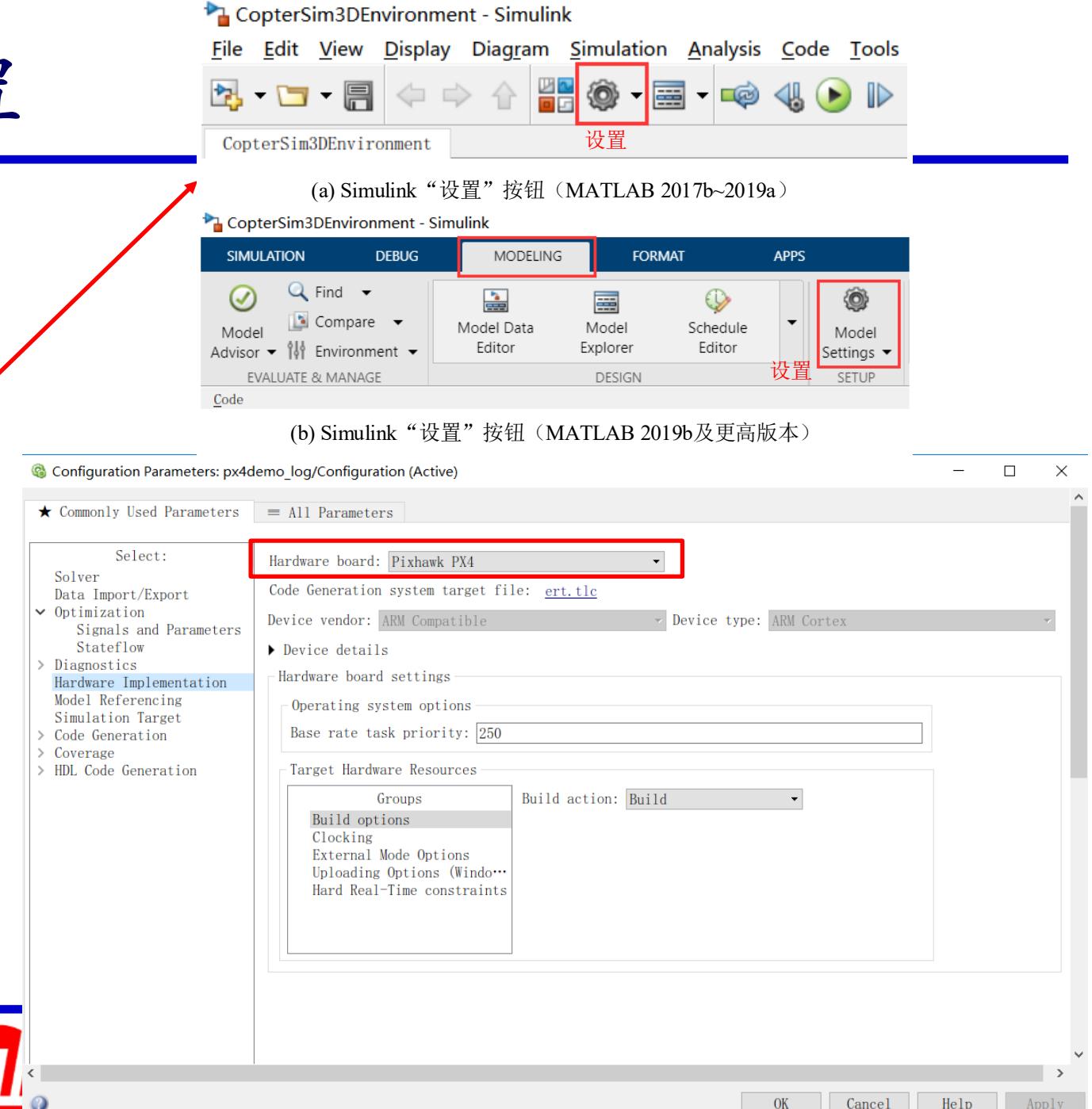
新建任意空白slx例程文件

1、进入Simulink设置页面（MATLAB 2019b及以上需要在MODELING标签页点击设置按钮）。

2、选择Hardware board设置为Pixhawk PX4后，会自动完成本平台需要的所有代码生成设置。

3、可以定制任务优先级。

4、配置编译选项。

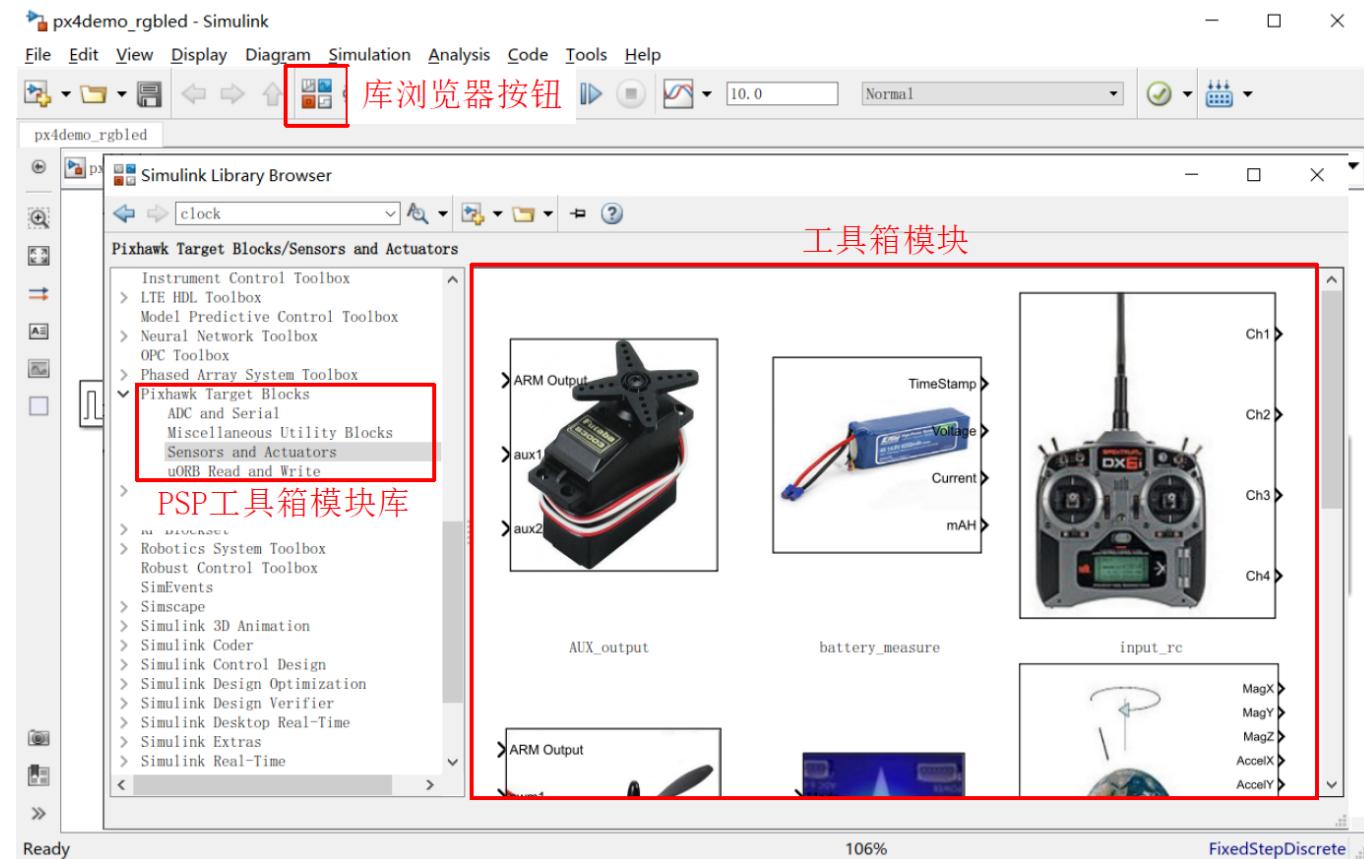




1.实验平台配置

1.5 PX4PSP工具箱位置

- 任意打开一个Simulink文件，点击“库浏览器（Simulink Library Brower）”按钮，可以在其中找到PSP工具箱的“Pixhawk Target Blocks”模块库。
- 其中的模块可以看做是从PX4的uORB池中订阅或发布数据的上层封装的接口，包含了传感器、遥控器、电机、串口等功能。
- 注意：这些模块内部没有模型，只有在生成代码并编译成PX4固件时才能获取数据，在Simulink中直接运行的话，接收到的数据为全0。
- 各个模块更详细的使用方法，可以参考基础版课程或教程的第三讲。

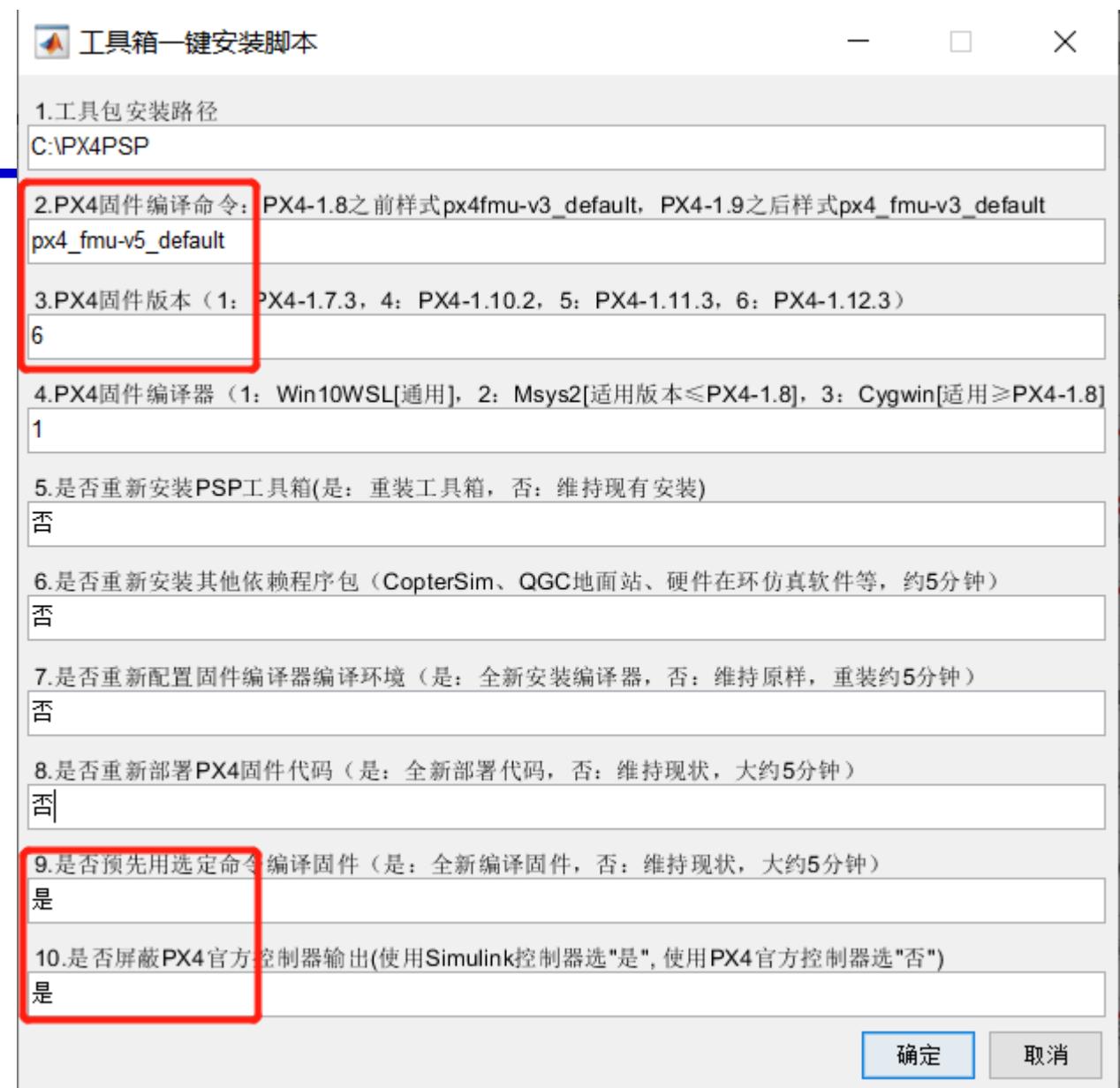
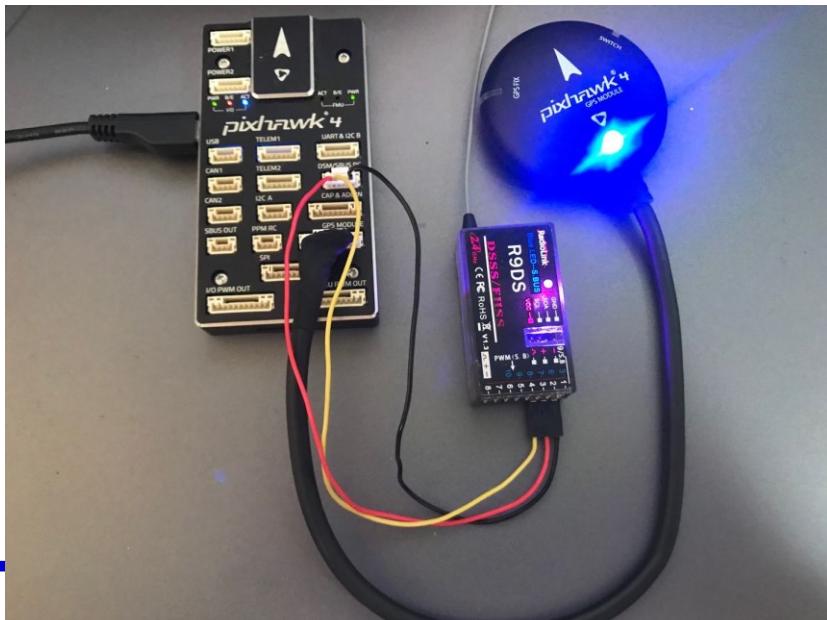




1.实验平台配置

1.6 实验配置—推荐安装配置选项

- 推荐使用最新的Pixhawk 4（外观见下图）自驾仪，其编译命令为px4_fmu-v5_default。
- 重新运行安装包内“OnekeyScript.p”脚本
- 使用最新的PX4固件版本“6”—PX4-1.12.3
- 其他配置如右图所示。



注：或者在MATLAB中输入“PX4CMD
px4_fmu-v5_default”命令来快速切换。



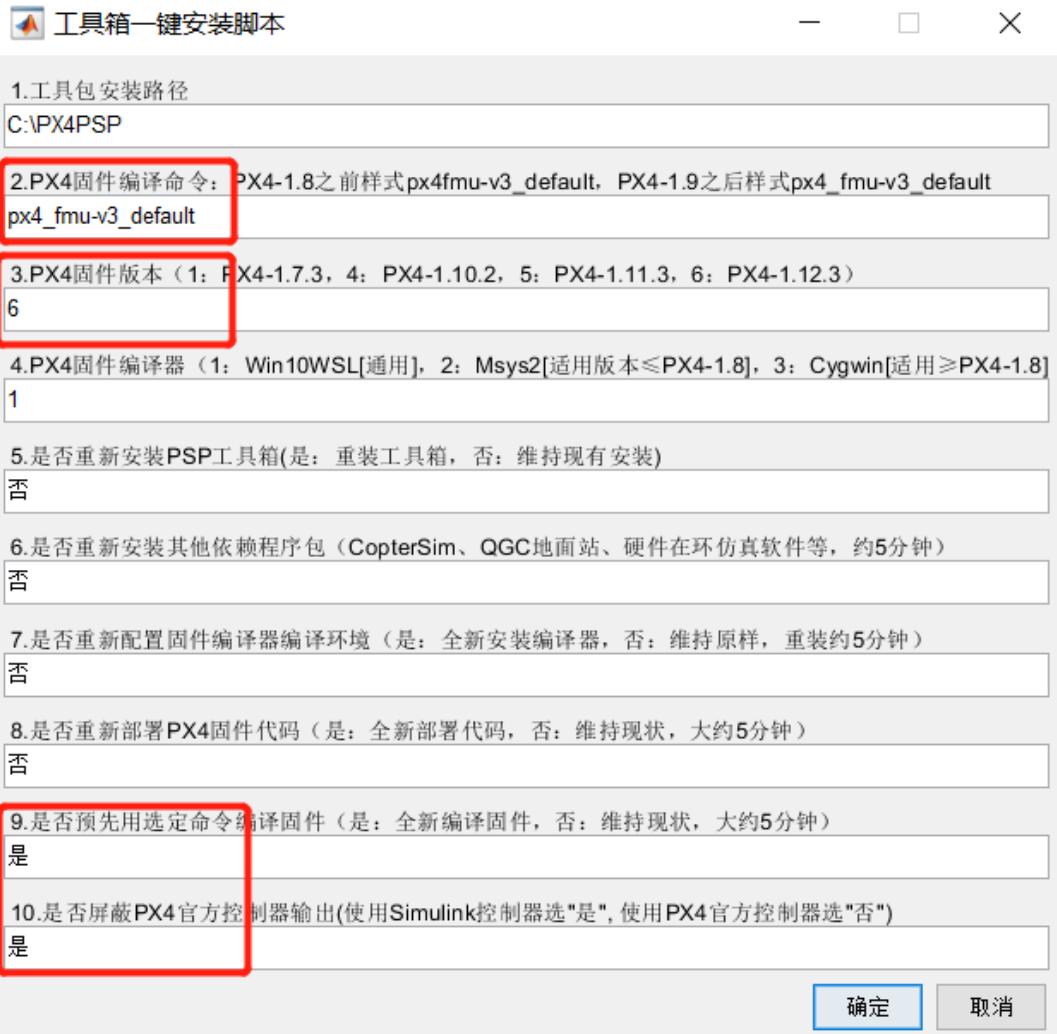
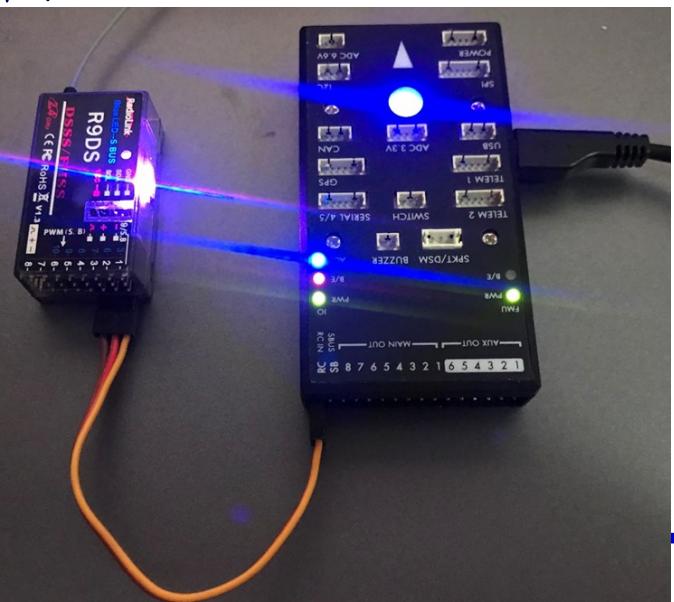
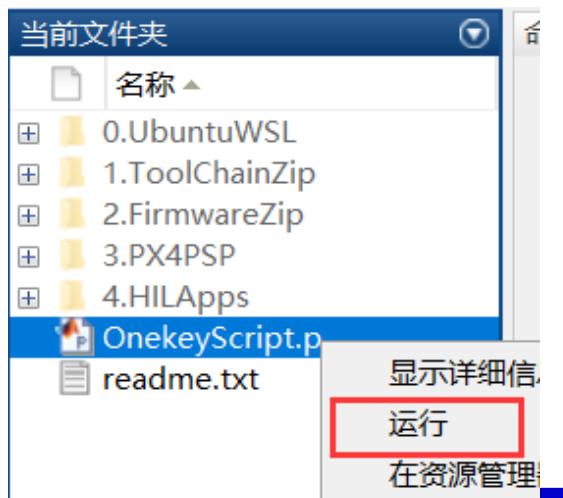


1.实验平台配置

注：如果使用教程推荐的Pixhawk 1自驾仪，推荐按本页配置，也可按照书上的配置方法，选择PX4 1.7.3版本+Msys2编译器（不推荐）。

1.7 实验配置—教材旧版安装配置选项

- 如果使用教材推荐的Pixhawk 1（外观见下图）的编译命令px4_fmu-v3_default。
- 重新运行安装包内“OnekeyScript.p”脚本。
- 使用最新PX4固件版本“6”—PX4-1.12.3，和编译器“1”—Win10WSL。
- 其他配置如右图所示。



注：或者在MATLAB中输入“PX4CMD px4_fmu-v3_default”命令来快速切换。



1.实验平台配置

1.8 实验配置—硬件配置与校验

1、硬件配置方法，请参考教程链接：

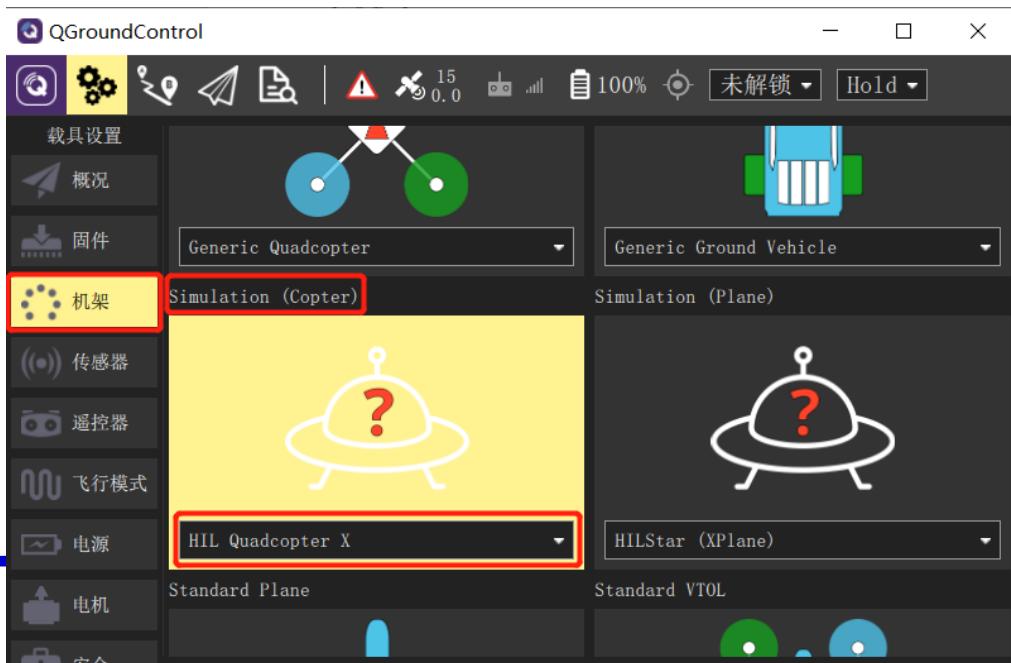
https://doc.rflysim.com/hardware/3RC/AT9s_Pro.html

2、确认完成以下配置：

- 确认Pixhawk已经在QGC中烧入最新的1.12版官方固件，且LED正常闪烁
- 正确连接Pixhawk与接收机，正确连接遥控器与接收机，打开QGC地面站，确定能观察到遥控器摇杆的动作信号。
- 正确对遥控器进行配置并在QGC中校准，最低位置和最高位置满足教程链接中定义。
- 确认Pixhawk飞控已经在QGC中设置选择HIL Quadcopter X机架。
- 确认QGC中飞行模式按教程配置。



注：箭头表示PWM值增大的拨动方向

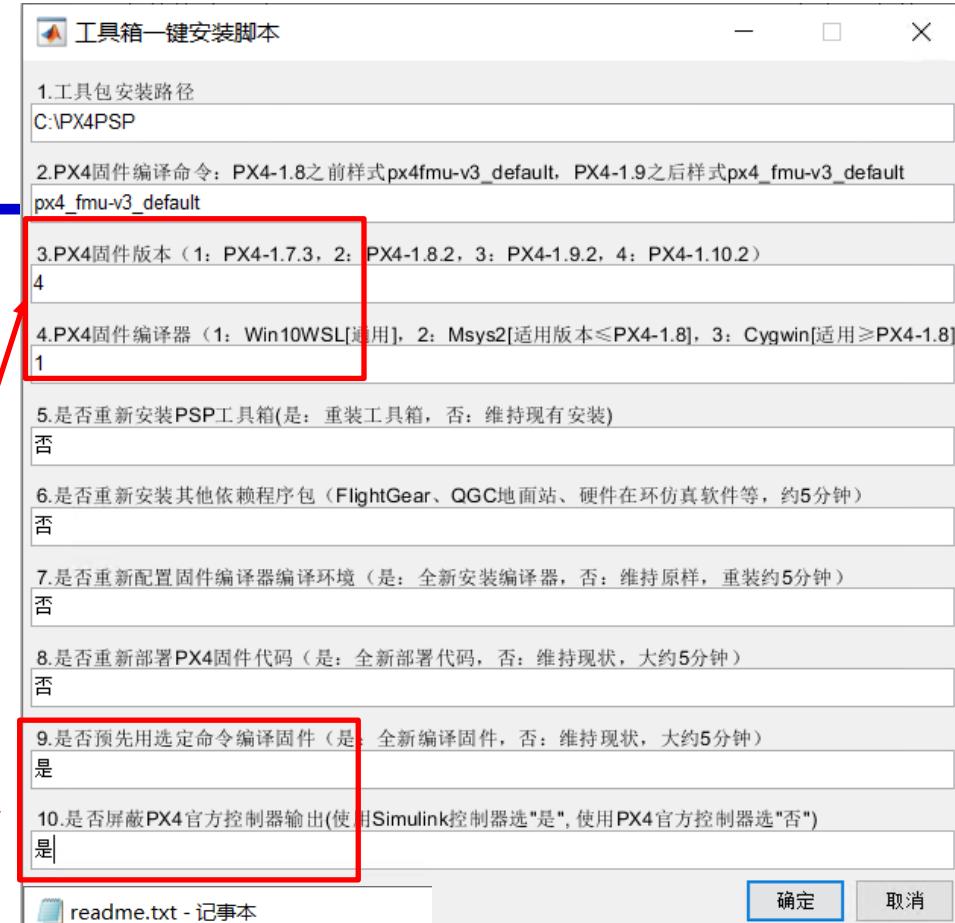
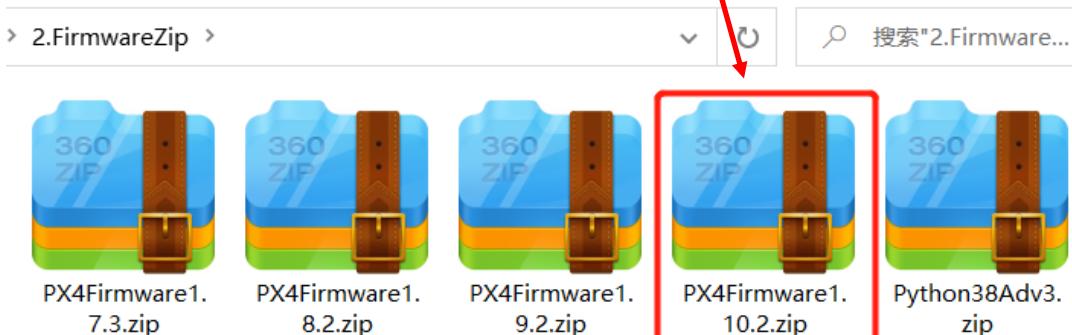




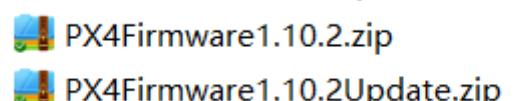
1.实验平台配置

1.9 选用其他固件版本（或自己的固件）

- 如果您需要使用自己的PX4固件代码，请将您的代码文件夹改名为**Firmware**，并压缩为**Firmware.zip**文件，然后根据**2.FirmwareZip\readme.txt**规则重命名，并选择需要固件版本。
- 例如，自己基于PX4 1.10开发的代码，命名为“**PX4Firmware1.10.2.zip**”，并替换原来掉“**2.FirmwareZip**”文件夹下的同名文件，在右图安装选项中的固件版本处选择“4”即可。
- 是否屏蔽PX4输出项目选择“是”，脚本会自动完成所有需要的固件修改以适配本平台。



注：也可以将PX4Firmware1.10.2.zip官方固件的增量文件打包并命名为“**PX4Firmware1.10.2Update.zip**”的格式放在**2.FirmwareZip**目录，安装时会自动拷贝到固件文件夹。





大纲

1. 实验平台配置

2. 关键接口介绍

3. 基础实验案例（免费版）

4. 进阶接口实验（个人版）

5. 进阶案例实验（集合版）

6. 扩展案例（完整版）

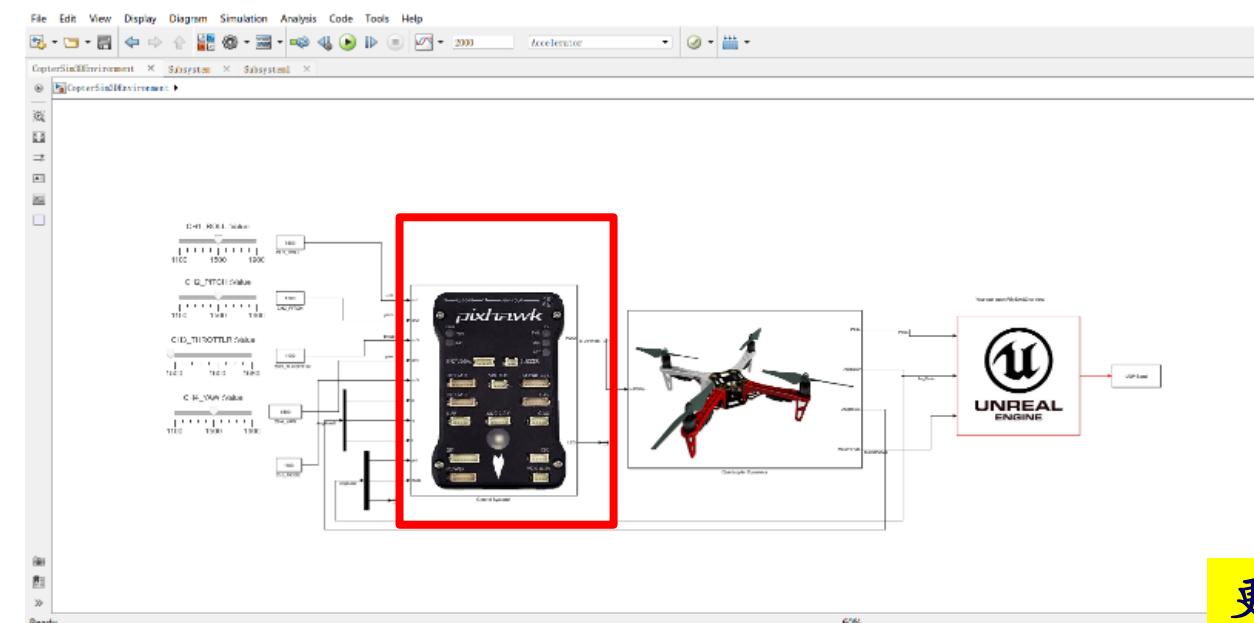
7. 小结



2.关键接口介绍

2.1 软件在环仿真实验

如右图所示，将多旋翼控制到指定的俯仰滚转角度，并将姿态的保持与控制，控制器响应遥控器的控制输入，可拖动左侧Slider模块模拟遥控器的输入。具体实验操作见文件Readme.pdf，实验效果如右图所示。



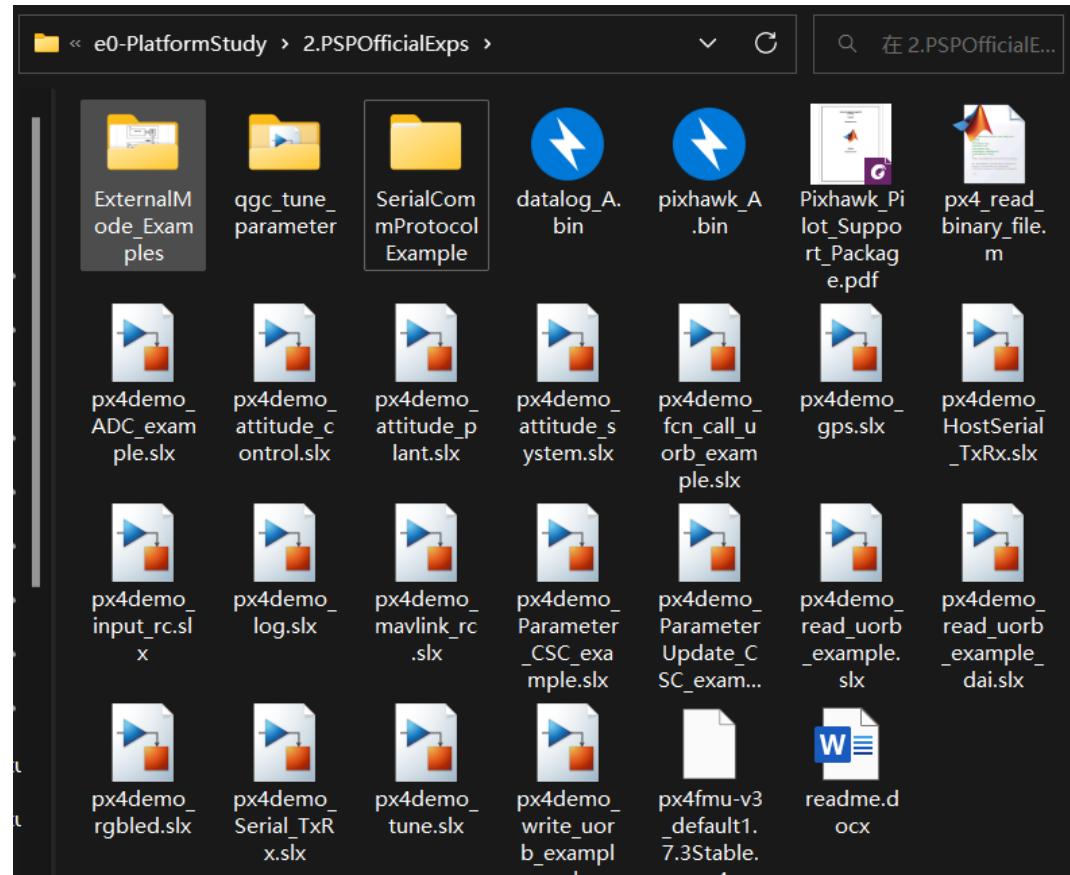
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.2 自动代码生成工具箱官方例程实验

MATLAB官方提供了Pixhawk Pilot Support Package (PSP)的相关例程和帮助文档(Pixhawk_Pilot_Support_Package.pdf)，用户可以通过学习实现在Simulink中对飞行控制模型的建模、模拟和验证，并通过自动代码生成的功能部署到控制系统集成的PX4硬件上。具体实验操作见文件[readme.pdf](#)。



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.3 姿态控制器设计实验

本实验设计了一个以遥控器Ch1~Ch5通道信号、角速度反馈量AngRateB以及多旋翼欧拉角（单位为rad）为输入的控制器，并在Simulink中搭建模型实现软件在环仿真→自动代码生成→硬件在环仿真→实飞的整个流程。

实验的具体操作见文件[readme.pdf](#)。

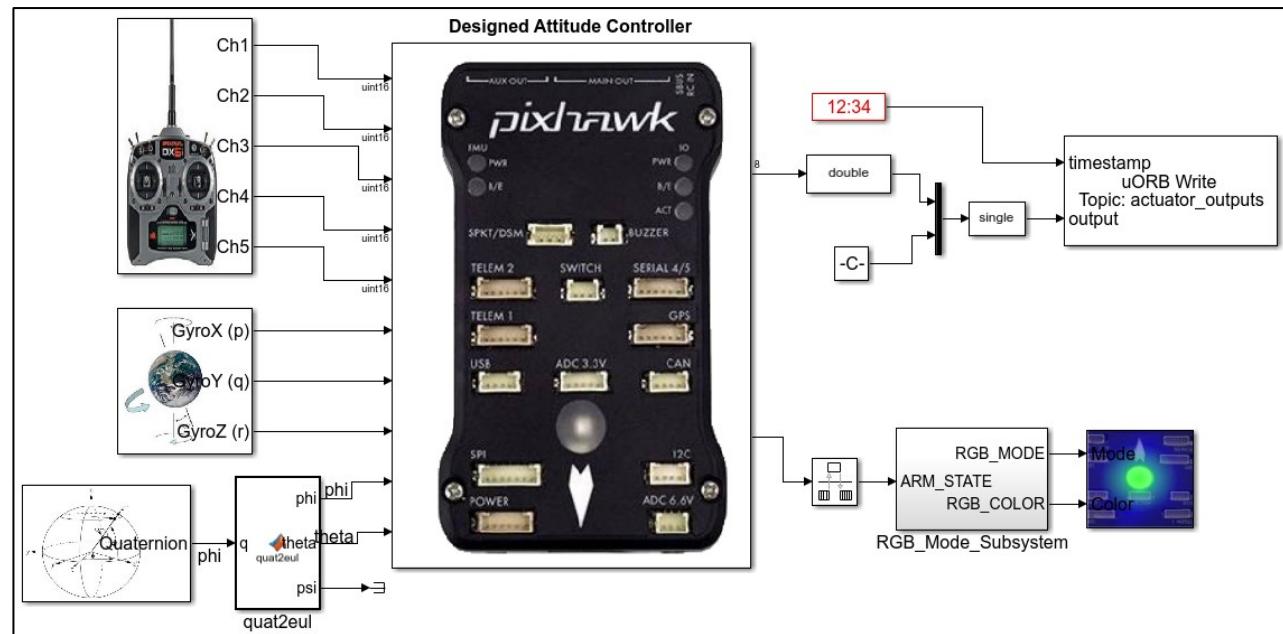


更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.基础实验

2.3 姿态控制器设计实验



硬件在环Simulink模型

```

Diagnostic Viewer
Exp4_AttitudeSystemCodeGen x
f.c.obj
[7/11] Building C object
src/modules/px4_simulink_app/CMakeFiles/modules_px4_simulink_app.dir/nuttxini
tialize.c.obj
[8/11] Linking C static library
src/modules/px4_simulink_app/libmodules_px4_simulink_app.a
[9/11] Linking CXX executable nuttx_px4fmu-v3_default.elf
[10/11] Generating px4fmu-v2.bin
[11/11] Creating /mnt/c/PX4PSP/Firmware/build/px4fmu-v3_default/px4fmu-
v3_default.px4
### Finished calling CMAKE build process #####
### Done invoking postbuild tool.
### Successfully generated all binary outputs.

C:\Users\dream\Desktop\e0\3.DesignExps\Exp4_AttitudeSystemCodeGen_ert_rtw>exit /B 0
### Successful completion of build procedure for model:
Exp4_AttitudeSystemCodeGen
### Creating HTML report file Exp4\_AttitudeSystemCodeGen\_codegen\_rpt.html

Build process completed successfully

```

```

C:\WINDOWS\SYSTEM32\cmd.exe
Loaded firmware for 9.0, size: 879196 bytes, waiting for the bootloader...
If the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.
PX4.SIMULINK = None
attempting reboot on COM3...
if the board does not respond, unplug and re-plug the USB connector.
attempting reboot on COM3...
if the board does not respond, unplug and re-plug the USB connector.
attempting reboot on COM3...
if the board does not respond, unplug and re-plug the USB connector.
Found board 9.0 bootloader rev 4 on COM3
50583400 00ac2600 00100000 00ffffff ffffffff ffffffff ffffffff 66ed47ff ff73cc15 c8ad940c dbc59f39 d6c20e06 f95
3d3ef f3073019 d035ab0d 3f60334e 10da9f8 cdb0cb0d 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8f
5c 5e07dec9 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e
fffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4
idtype: =0
vid: 000026ac
pid: 00000010
coa: Zu1H//9zzBXIrZQM2SWF0dbCDgb5U9Pv8wcwGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2mI7x.jQIMhvmjt01bJHjuPXF4H3syca+WjRYo
VE0u7vChto1zlqLhApe8BnKXIm7GDuwDwwAbgGiZzdf9XHKQdlCSqZi4=
sn: 0038001f3432470d31323533
Erase : [=====] 100.0%

```

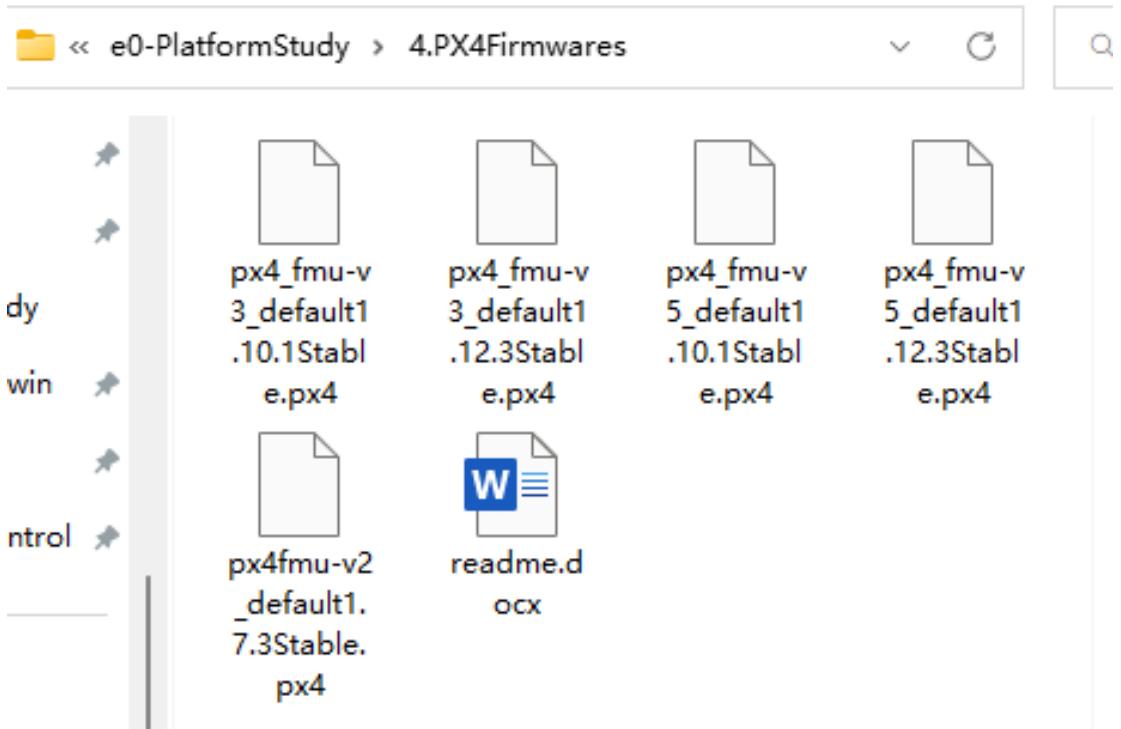
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.4 飞控固件烧录实验

本实验提供了部分飞控固件，可通过**QGroundControl**进行飞控固件的烧录实验，具体操作步骤见文件[readme.pdf](#)。



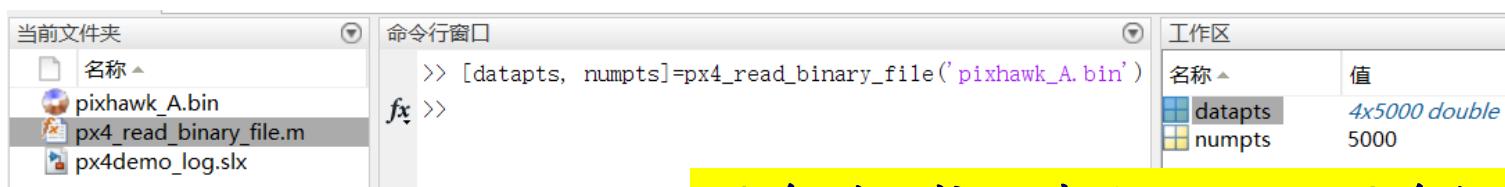
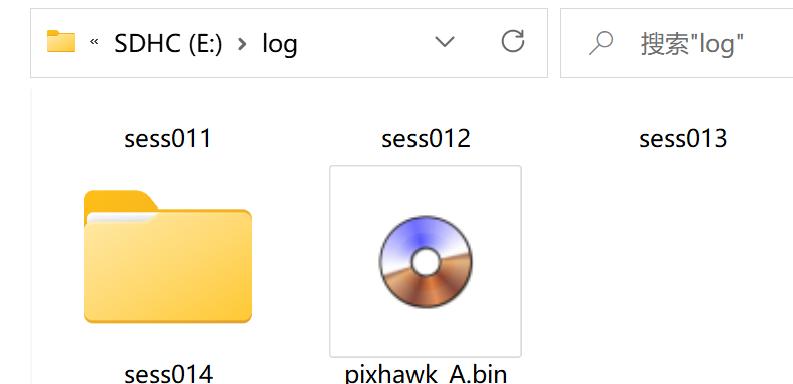
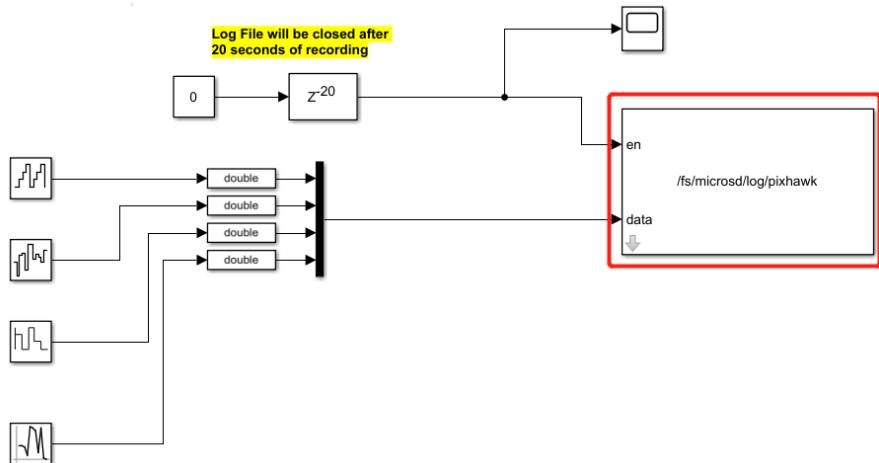
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.5 log数据记录

使用二进制日志记录模块：binary_logger，完成飞行数据写入与读取。具体操作步骤见文件[readme.pdf](#)，实验效果如下：



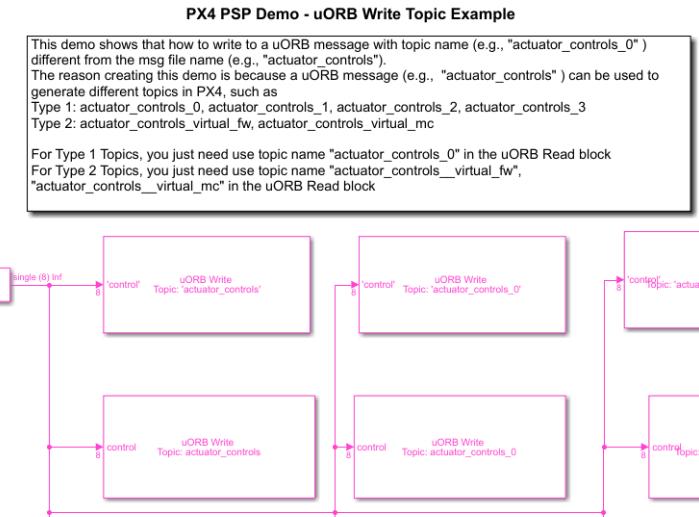
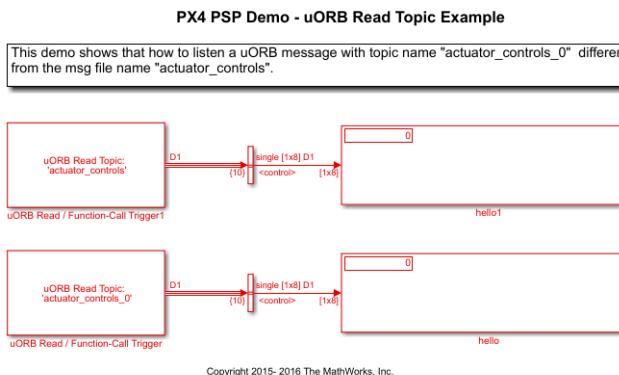
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



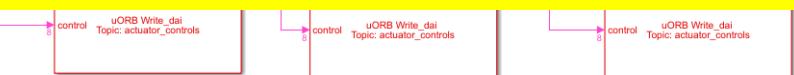
2.关键接口介绍

2.6 uORB读写通信

PX4 的 uORB 消息系统是提供了非常强大且方便的内部模块间数据交互能力，所有模块都可以将数据放在消息池中，其他模块可以从消息池订阅到所需数据，具体操作步骤见文件[readme.pdf](#)，实验效果如下：



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。





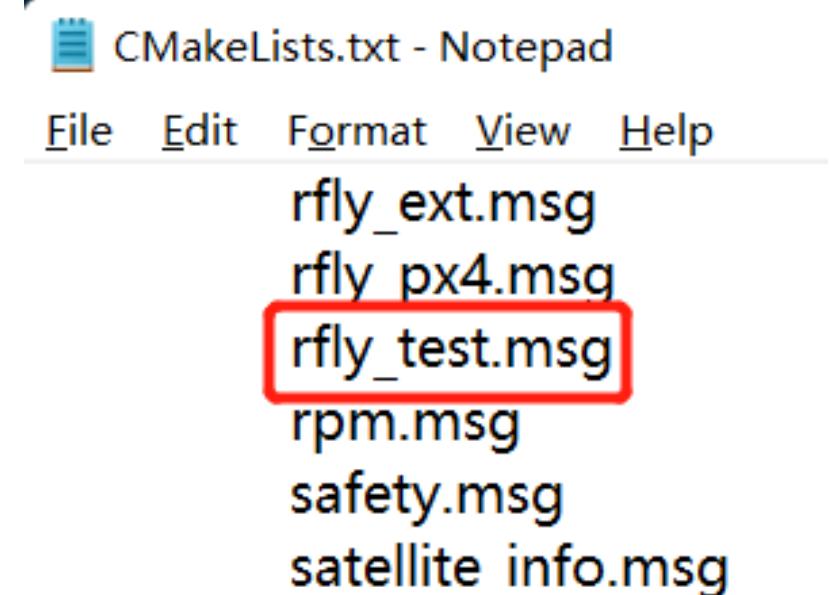
2.关键接口介绍

2.7 自定义uORB消息

通过创建一个自定义的 uORB 消息实现读写功能，以此熟悉并掌握 PX4 的 uORB 消息系统。

具体操作步骤见文件[readme.pdf](#)，实验效果如下：

```
rfly_test.msg ×  
D: > OneDrive > RflySimSource > RflySim/  
1 uint64 timestamp  
2 float32[8] control  
3 |
```



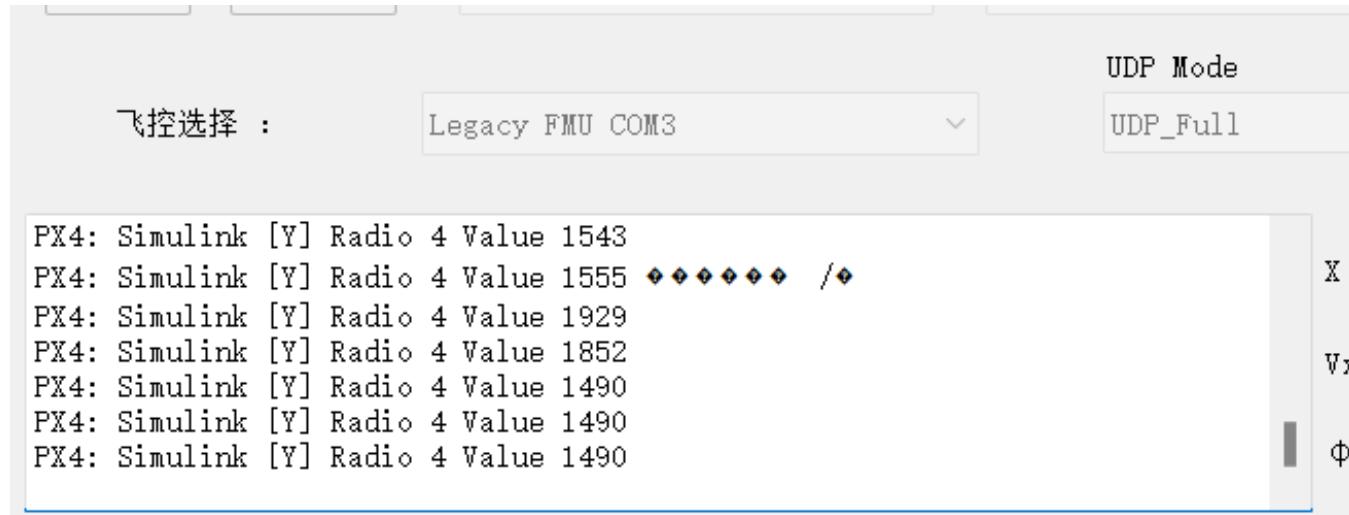
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.8 回传提示消息

在飞控中，我们时常需要向外发布一些文字消息，来反映系统当前的运行状态，这个功能可以通过发送“mavlink_log”的uORB 消息来实现。具体操作步骤见文[readme.pdf](#)，实验效果如下：



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.9 PX4控制器的外部通信

本例程以外部发送的 rfly_ctrl 数据来作为遥控器输入，同时会将收到的数据向 rfly_px4 发送出去，回传给外部程序。具体操作步骤见文件[readme.pdf](#)，实验效果如下：

The screenshot shows the QGroundControl interface with the 'MAVLink Inspector' tab selected. The main window displays real-time MAVLink messages. A specific message, 'ACTUATOR_CONTROL_TARGET', is highlighted with a red box. The message details are shown in a table:

1	ACTUATOR_CONTROL_TARGET	30.0Hz
1	ALTITUDE	10.0Hz
1	ATTITUDE	50.7Hz
1	ATTITUDE_QUATERNION	50.7Hz
1	ATTITUDE_TARGET	8.0Hz

On the right, detailed information for the selected message is provided:

信息: ACTUATOR_CONTROL_TARGET (140) 30.0Hz
组件: 1
计数: 3242

Name	Value	Type
time_usec	136000000	uint64_t
group_mlx	123	uint8_t
controls	1500, 1500, 1100, ...	float

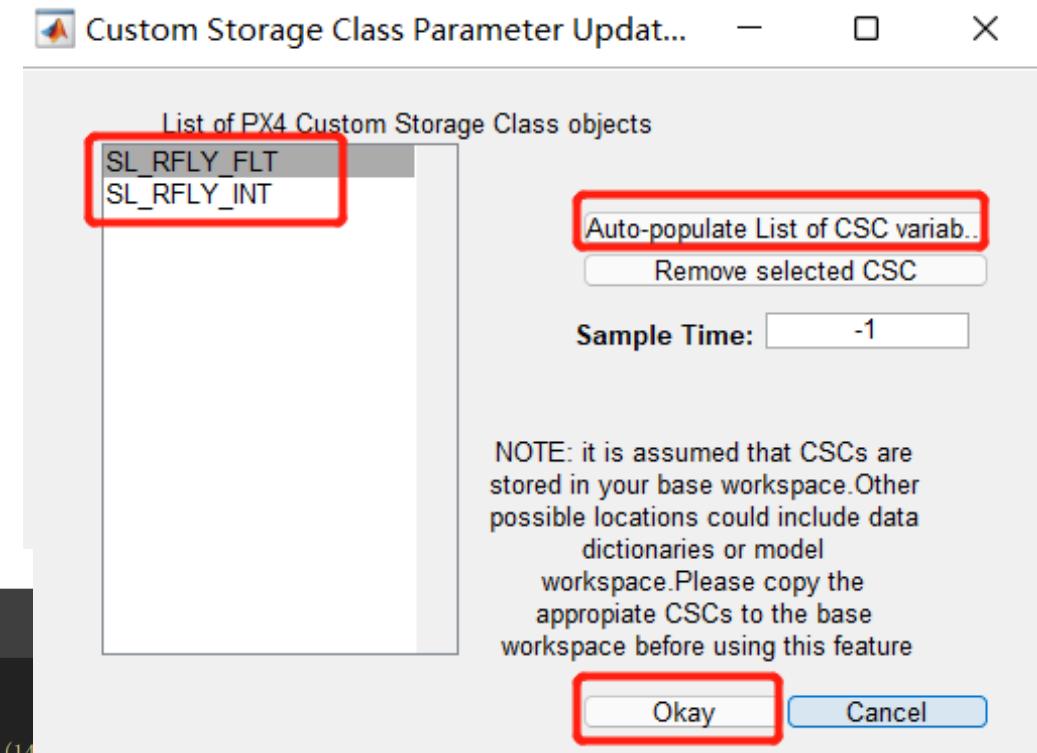
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.10 QGC实时调整控制器参数

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，并对控制器参数进行实时调整，以使得飞机达到最佳的控制效果。具体操作步骤见文件[readme.pdf](#)，
(部分)实验效果如下：



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



2.关键接口介绍

2.11 传感器数据读取实验

通过RflySim 的底层开发接口，可获取的传感器数据包含磁力计、加速度计、陀螺仪、气压计和时间戳以及GPS 数据等信息。本实验将进行上述传感器部分数据的获取，以此思路可订阅更加多样的传感器数据。具体操作步骤见文件[readme.pdf](#), (部分) 实验效果如下：

The screenshot shows the RflySim Analyze Tools interface. On the left, there is a sidebar with several tabs: '日志下载' (Log Download), '地理标记图像' (Geotag Image), 'Mavlink 控制台' (Mavlink Console) which is highlighted in yellow, 'MAVLink 检测' (MAVLink Detection), and '振动' (Vibration). The main area is titled 'Analyze Tools' and contains the message 'Provides a connection to the vehicle's system shell.' Below this, a terminal window displays NuttShell (NSH) NuttX-10.1.0. The user has run the command 'nsh> listener rfly_px4'. The output shows a message on the 'rfly_px4_s' topic: 'timestamp: 1691301514776 (18446742382788.000000 seconds ago)', 'control: [0.4676, -9.6510, 0.0045, 0.0034, 58.0359, 401540288.0000, 1162593664.0000, 58060.0000]'. A red box highlights this terminal output.

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。

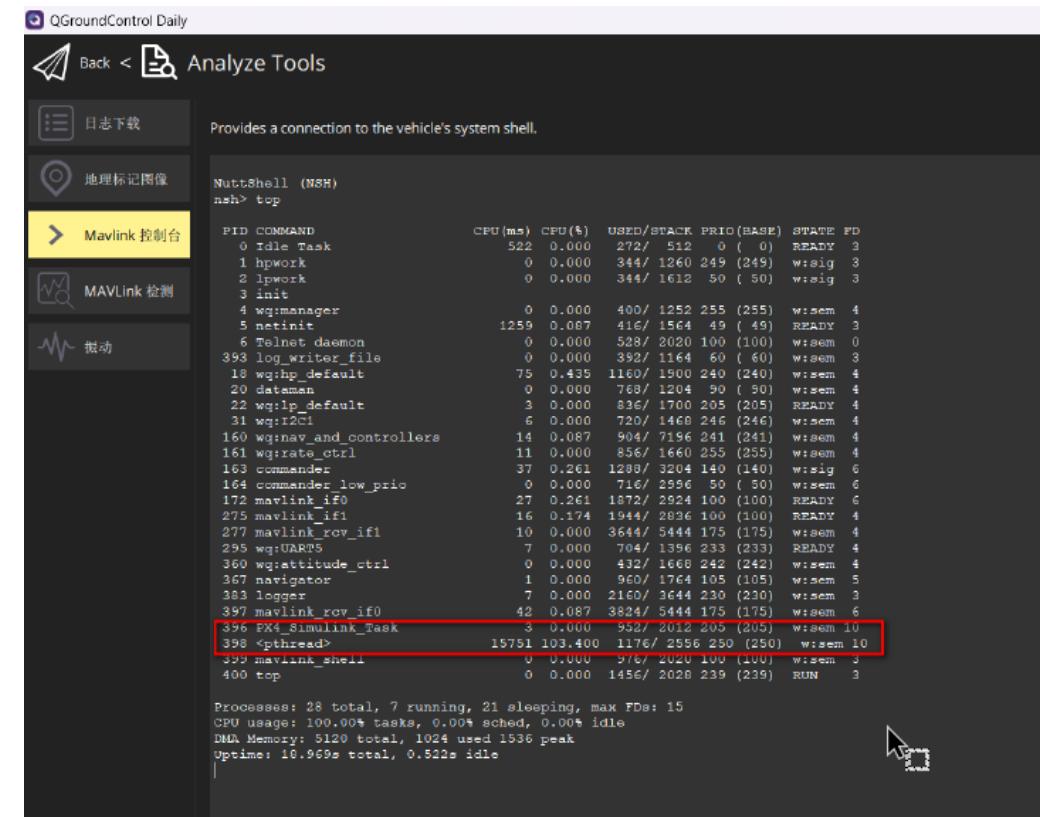


2.关键接口介绍

2.12 自驾仪CPU 使用率查看实验

在使用 RflySim 平台进行底层开发的时，通常是需要在自驾仪硬件上验证我们自己开发的算法，但在 Simulink 中搭建的算法模型在自动代码生成自驾仪固件时，可能会因算法的复杂度和模型搭建的合理性，造成自驾仪系统的 CPU 占用率超负载，从而导致实验失败，如下图。本实验将演示如何查看自己自驾仪系统的 CPU 占用率情况。具体操作步骤见文件 [readme.pdf](#)，(部分)实验效果如下：

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。





2.关键接口介绍

2.13 Simulink中M Function 和S Function 对自驾仪系统资源占用对比实验

PX4 固件的飞控系统都是基于Nuttx 这个操作系统的。Nuttx 是一个实时嵌入式操作系统（Embedded RTOSTOS），它很小巧，在微控制器环境中使用。本实验将分别烧录由M Function 和S Function 搭建的Simulink 模型，通过分析自驾仪系统的资源占用情况，可得S Function 搭建的Simulink 模型占用自驾仪资源更少。具体操作步骤见文件[readme.pdf](#), (部分) 实验效果如下：

更多详细接口请见：[API](#), 更多例程请见：[Readme](#).

```

CPU(msec) CPU(s) USED/PRACK PRI/IO/IRQ SCHED ID
----- -----
1 idle Task 35610 0.5 625 292/ 312 0 ( 0 ) READY 5
1 hwclock 0 0.000 340/ 1268 249 (249) w:seq 3
2 iwork 0 0.000 340/ 1620 50 ( 50 ) w:seq 3
3 init 0 0.000 2204/ 2332 100 (100) w:seqn 3
4 wpm:manager 0 0.000 420/ 1260 255 (255) w:seqn 3
108 ekf2 0 0.013 764/ 2028 100 (100) w:seq 3
22 wqip_default 0 0.001 1132/ 1900 237 (237) w:seqn 3
26 dataman 0 0.000 783/ 2028 100 (100) w:seqn 3
28 wrlp:default 10 0.051 3004/ 1820 236 (236) w:seqn 3
189 uavcan_fw_rcv 152 0.855 1796/ 3628 236 (236) w:seqn 3
197 uavcan_fw_tx 111 0.624 1572/ 6004 120 (120) w:seqn 3
206 wqinav_and_controllers 36 0.202 1096/ 2244 242 (242) w:seqn 3
207 wqrates_ctrl 0 0.000 404/ 1956 255 (255) w:seqn 3
209 commander 69 0.379 1440/ 3212 140 (140) w:seq 5
217 navlink_if0 58 3.288 1748/ 2828 100 (100) READY 6
361 navlink_if1 123 0.639 1563/ 2740 100 (100) w:seq 4
362 navlink_rcv_if1 22 0.031 1804/ 4716 175 (175) w:seqn 4
391 pxio 34 0.151 944/ 1484 237 (237) w:seqn 4
486 log_writer_file 0 0.000 364/ 1172 60 ( 60 ) w:seqn 3
492 navigator 0 0.003 1092/ 1772 109 (109) w:seqn 6
475 logger 29 0.162 2396/ 3644 230 (230) w:seqn 3
493 qphread 513 2.871 476/ 2564 250 (250) w:seqn 7
492 px4_simulink_Task 0 0.000 700/ 2020 205 (205) w:seqn 6
198 navlink_rx_if0 24 0.132 1652/ 4176 175 (175) w:seqn 6
495 navlink_shell 0 0.000 849/ 2028 100 (100) w:seqn 3
500 top 70 0.451 1452/ 4084 237 (237) RUN 3

Processes: 26 total, 3 running, 23 sleeping, max FDs: 16
CPU usage: 10.00% tasks, 0.30% sched, 85.69% idle
DMA Memory: 5120 total, 1024 used 1536 peak
Uptime: 37.433s total, 33.611s idle

```

```

CPU(msec) CPU(s) USED/PRACK PRI/IO/IRQ SCHED ID
----- -----
1 idle Task 114550 50.033 2957/ 312 0 ( 0 ) READY 5
1 hwclock 0 0.000 340/ 1268 249 (249) w:seq 3
2 iwork 0 0.000 340/ 1620 50 ( 50 ) w:seq 3
3 init 0 0.000 2130/ 2332 100 (100) w:seqn 3
4 wpm:manager 1 0.012 764/ 2028 100 (100) w:seqn 3
108 ekf2 22 0.060 784/ 1924 205 (205) w:seqn 3
22 wqip_default 5 0.060 784/ 1924 205 (205) w:seqn 3
189 uavcan_fw_rcv 78 0.855 1796/ 3628 236 (236) w:seqn 3
197 uavcan_fw_tx 98 0.639 1572/ 6004 120 (120) w:seqn 3
206 wqinav_and_controllers 18 0.025 1096/ 2244 242 (242) w:seqn 3
207 wqrates_ctrl 0 0.000 404/ 1956 255 (255) w:seqn 3
209 commander 35 0.379 1440/ 3212 140 (140) w:seq 5
217 navlink_if0 297 3.288 1748/ 2828 100 (100) READY 6
361 navlink_if1 1 0.493 1563/ 2740 100 (100) w:seq 6
362 navlink_rcv_if1 11 0.121 1804/ 4716 175 (175) w:seqn 4
391 pxio 37 0.151 944/ 1484 237 (237) w:seqn 4
486 log_writer_file 0 0.003 1052/ 1772 105 (105) w:seqn 6
475 logger 13 0.149 2396/ 3644 230 (230) w:seqn 3
493 qphread 46 0.511 500/ 2564 250 (250) w:seqn 7
492 px4_simulink_Task 0 0.000 700/ 2020 205 (205) w:seqn 6
198 navlink_rx_if0 12 0.136 1652/ 4176 175 (175) w:seqn 6
495 navlink_shell 0 0.000 849/ 2028 100 (100) w:seqn 3
500 top 95 0.397 1452/ 4084 237 (237) RUN 3

Processes: 26 total, 3 running, 23 sleeping, max FDs: 16
CPU usage: 7.67% tasks, 0.30% sched, 92.02% idle
DMA Memory: 5120 total, 1024 used 1536 peak
Uptime: 120.018s total, 117.593s idle

```



2.关键接口介绍

2.14 传感器数据读取实验

通过RflySim 的底层开发接口，可获取的传感器数据包含磁力计、加速度计、陀螺仪、气压计和时间戳以及GPS 数据等信息。本实验将进行上述传感器部分数据的获取，以此思路可订阅更加多样的传感器数据。具体操作步骤见文件[readme.pdf](#), (部分) 实验效果如下：

The screenshot shows the 'Analyze Tools' interface in RflySim. On the left is a sidebar with icons for '日志下载' (Log Download), '地理标记图像' (Geotag Image), 'Mavlink 控制台' (Mavlink Console, highlighted in yellow), 'MAVLink 检测' (MAVLink Detection), and '振动' (Vibration). The main area is a terminal window titled 'Back < Analyze Tools'. It displays NuttShell (NSH) NuttX-10.1.0 logs. A red box highlights the following text:

```
NuttShell (NSH) NuttX-10.1.0
nsh> listener rfly_px4
TOPIC: rfly_px4
rfly_px4_s
    timestamp: 1691301514776 (18446742382788.000000 seconds ago)
    control: [0.4676, -9.6510, 0.0045, 0.0034, 58.0359, 401540288.0000, 1162593664.0000, 58060.0000]
nsh>
```

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



大纲

1. 实验平台配置

6. 扩展案例（完整版）

2. 关键接口介绍

7. 小结

3. 基础实验案例（免费版）

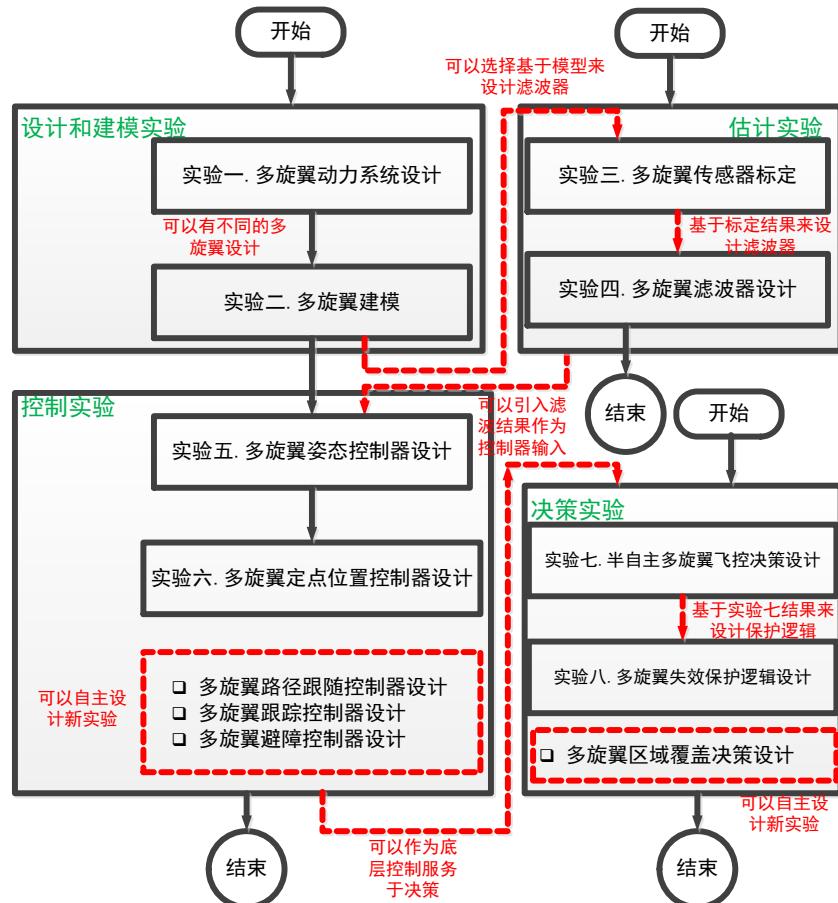
4. 进阶接口实验（个人版）

5. 进阶案例实验（集合版）



3.进阶实验

3.0 进阶实验总览



- 本平台提供的例程可以保证每个实验或者每个版块的实验可以被单独完成
- 为了使任务目标有差异化，我们课程可以按照一种递进的结构完成。递进路线可以分为：
 - (1) 设计和建模实验 → 控制实验
 - (2) 设计和建模实验 → 控制实验 → 决策实验
 - (3) 设计和建模实验 → 估计实验 → 控制实验 → 决策实验
- 需要设计不同的飞行器，这样将会使模型各不相同，而且建模方法也可以各不相同，控制实验的设计也各不相同。
- 教师还可以自行增加附加实验



3. 基础实验案例（免费版）

3.0 进阶实验总览

打开例程，阅读并运行程序代码，然后观察并记录分析数据。

指导读者修改例程，运行修改后的程序并收集和分析数据。

在上述两个实验的基础上，针对给定的任务，进行独立的设计。



3. 基础实验案例（免费版）

3.0 进阶实验总览

表. 实验类型、目标和内容

目标	基础实验	分析实验	设计实验
熟悉开发平台	✓	✓	✓
熟悉分析过程	✗	✓	✓
熟悉设计方法	✗	✗	✓
进行软件在环仿真	✓	✓	✓
进行硬件在环仿真	✓	✓	✓
实际实验测试	✗	✗	✓



3. 基础实验案例（免费版）

3.1 动力系统设计实验

本实验的目标如下：

- 1、利用多旋翼飞行评估网站，设计多旋翼飞行器动力系统；
- 2、根据已知信息，设计出多旋翼飞行器的动力系统并与多旋翼飞行评估网站生成的参数进行对比，分析在不同城市、温度、螺旋桨大小和数量对多旋翼飞行器的悬停事件影响。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.1 动力系统设计实验

实验效果如下(部分):

详细信息					
悬停性能:		最大油门性能:		整体性能:	
悬停时间	: 22.5 min.	飞行时间	: 7 min.	正常使用	: 17.8 min.
油门百分比	: 63.6 %	总升力	: 94.3 N	整机重量	: 4.56 kg
电调电流	: 6.69 A	电机电流	: 21.8 A	剩余载重	: 2.8 kg
电机转速	: 4623.5 rpm	电机转速	: 6716.3 rpm	最大起飞海拔	: 3.85 km
电机输出功率	: 132.2 W	电机输出功率	: 417.8 W	最大倾斜角度	: 51.7 °
电池输出电压	: 23.7 V	电池输出电压	: 22.9 V	最大平飞速度	: 12.4 m/s
电池输出电流	: 27.2 A	电池输出电流	: 87.3 A	单程飞行距离	: 8.5 km
能量效率	: 80.9 %	能量效率	: 79.8 %	抗风等级	: 4 级

螺旋桨尺寸/英寸	悬停时间/min
10	17
9.4	16.5
9	15.9
8	14.5

地点	海拔/m	悬停时间/min
上海	4	16.5
北京	43.5	16.5
长沙	500	16.1
拉萨	3658	13.5

温度/°C	悬停时间/min
0	17.1
10	16.8
20	16.6

更多详细接口请见: [API](#), 更多例程请见: [Readme](#).



3. 基础实验案例（免费版）

3.2 动态建模实验

本实验的目标如下：

- 1、分析多旋翼总质量、转动惯量矩阵、螺旋桨推力系数、螺旋桨拉力系数对整个多旋翼飞行性能产生的影响。
- 2、建立完整的多旋翼飞行器模型，并在RflySim3D中添加四旋翼的三维模型。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

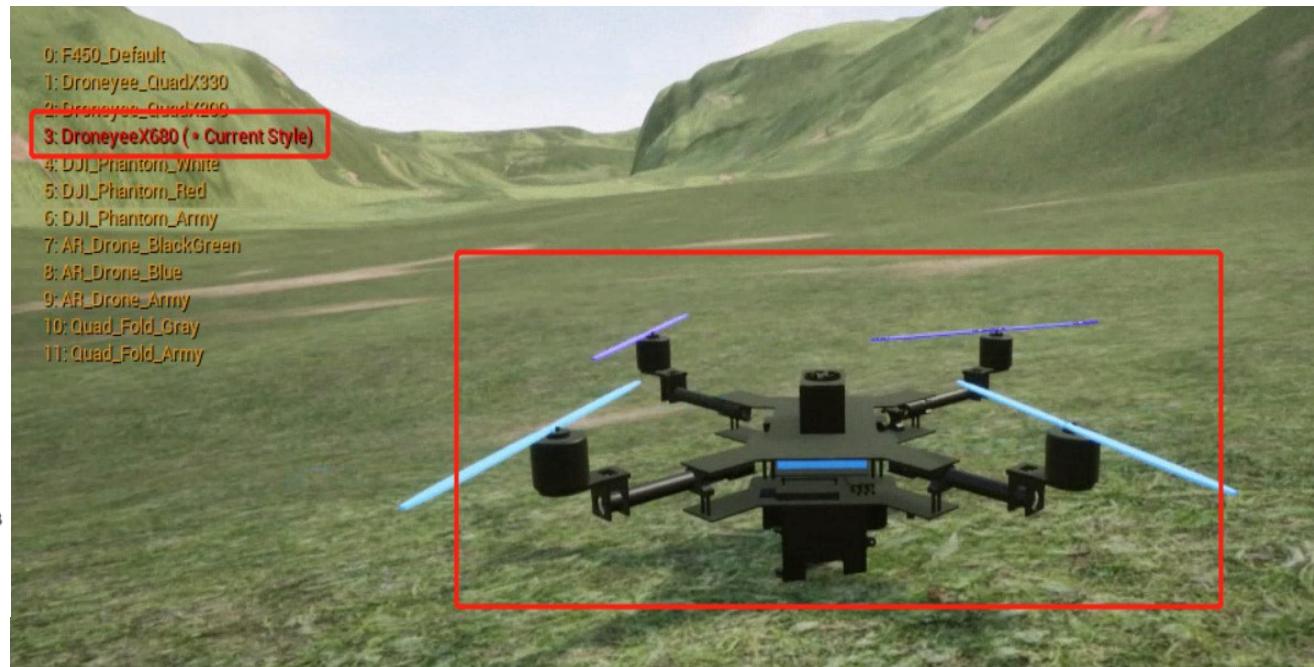
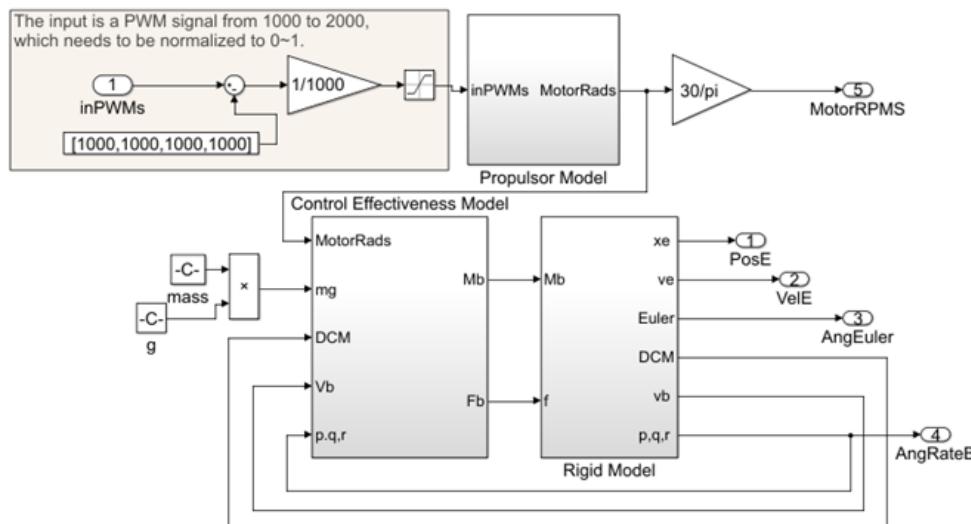
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.2 动态建模实验

实验效果如下(部分):



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.3 传感器标定实验

本实验的目标如下：

- 1、根据实验步骤完成加速度的标定。
- 2、根据给出的磁力计误差模型，设计磁力计数据采集模型，用测得的数据和LM 算法函数求出模型参数的最优解，完成磁力计的标定。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.3 传感器标定实验

实验效果如下(部分):

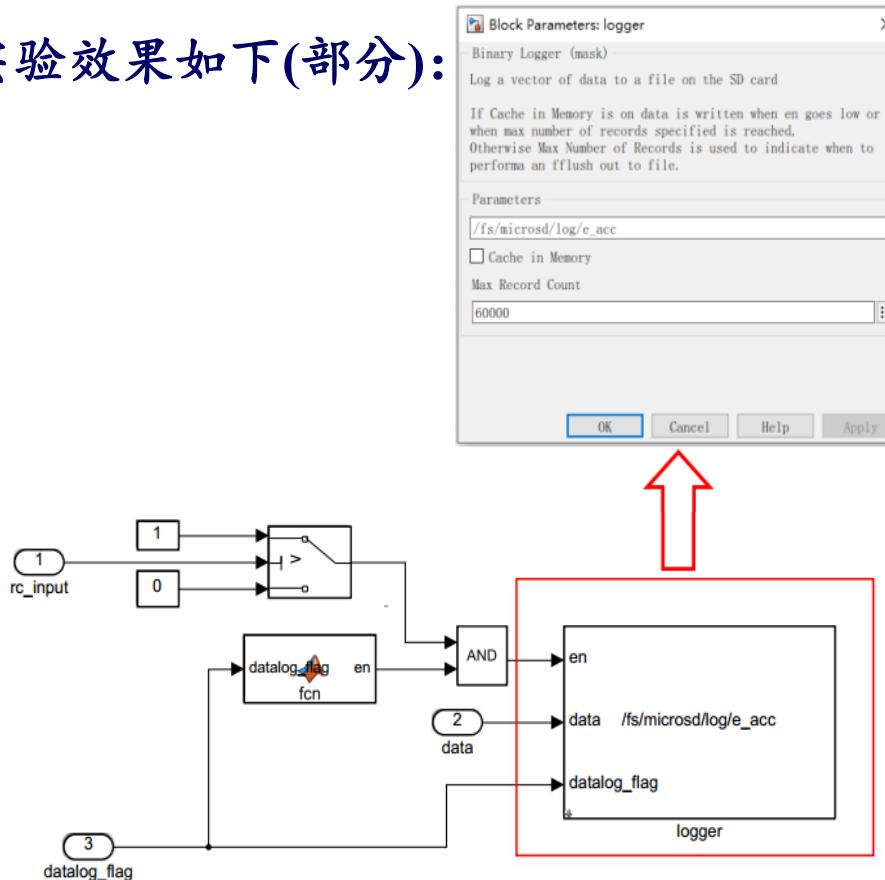


图. 加速度数据采集“binary_logger”模块

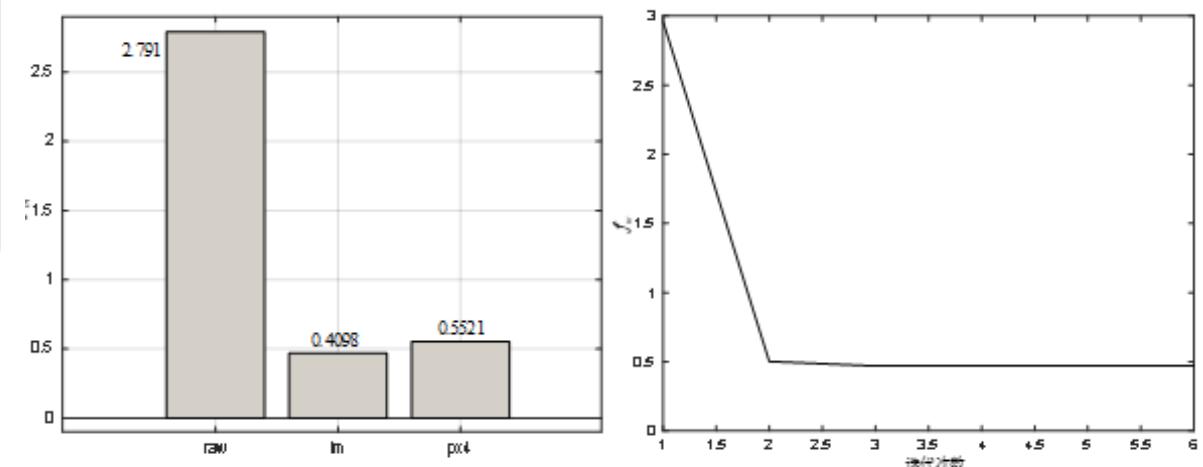


图. 磁力计校正对比值

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.4 传感器标定实验

本实验的目标如下：

- 1、根据实验中所提供的数据，完成互补滤波，与原数据和Pixhawk自带的滤波器算出的数据进行比较，以理解互补滤波器的优点。
- 2、改进互补滤波器中的参数，分析互补滤波器参数对滤波效果的影响。
- 3、理解卡尔曼滤波原理，设计卡尔曼滤波器实现滤波器，处理加速度和角速度数据，并绘制出相关姿态角得数据图。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

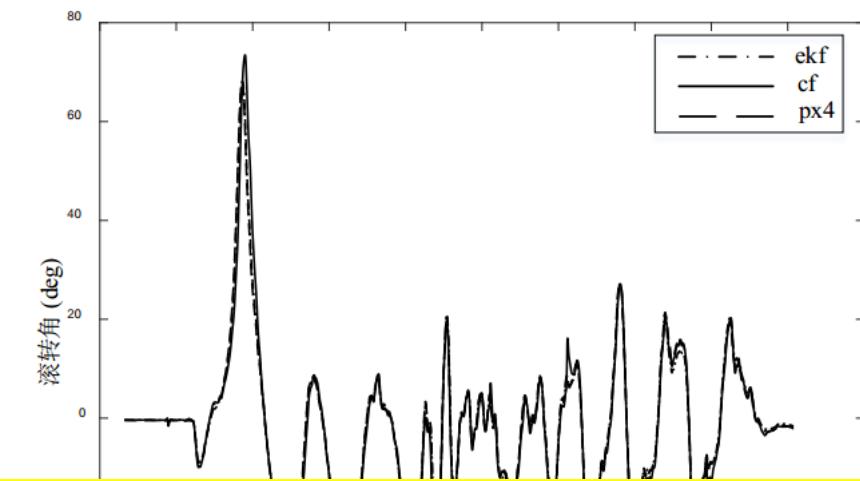
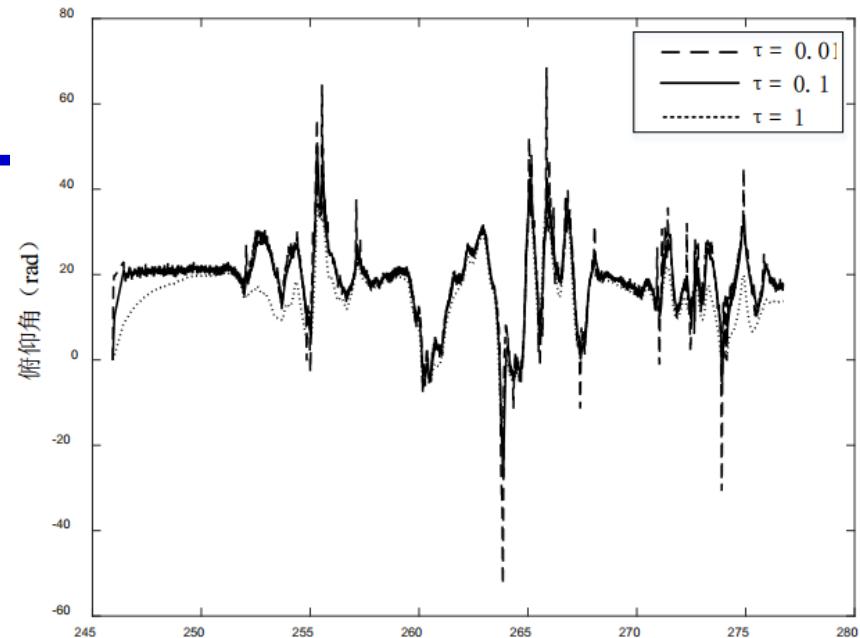
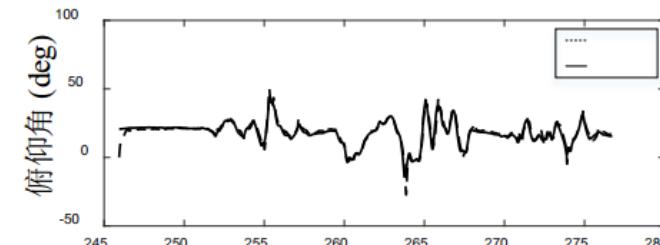
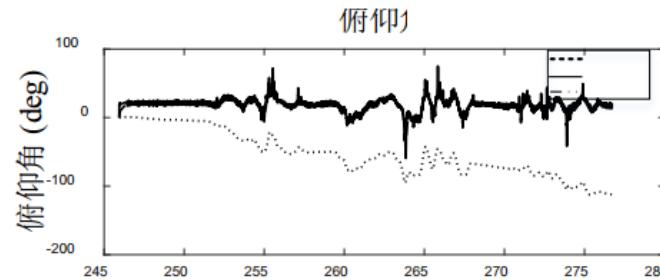
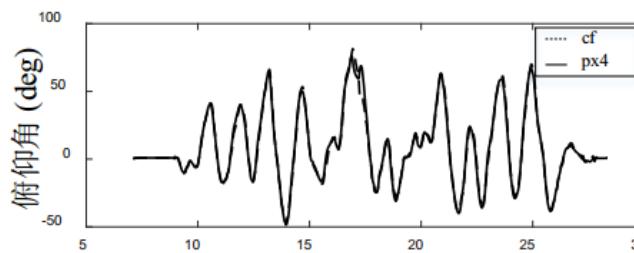
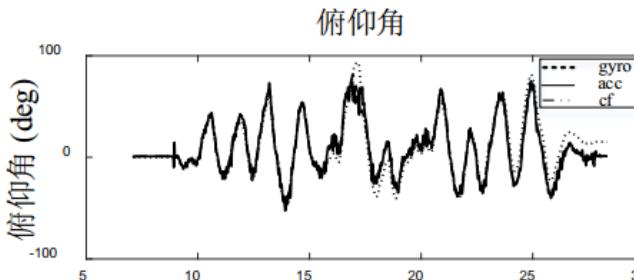
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.4 传感器标定实验

实验效果如下(部分):



更多详细接口请见：[API](#), 更多例程请见：[Readme](#)。



飞思实验室



3. 基础实验案例（免费版）

3.5 姿态控制器设计实验

本实验的目标如下：

- 1、复现四旋翼飞行器的Simulink仿真，分析控制分配器的作用；记录姿态的阶跃响应，并对开环姿态控制系统进行扫频以绘制Bode图，分析闭环姿态控制系统的稳定裕度；完成四旋翼硬件在环仿真。
- 2、调节PID控制器相关参数以改善控制性能并记录超调量和调节时间，得到一组恰当参数；使用调试后的参数，对系统进行扫频以绘制Bode图，观察系统幅频响应,相频响应曲线，分析其稳定裕度。
- 3、建立姿态控制通道的传递函数模型，设计校正控制器；使用自己设计得控制器进行软、硬件仿真实验及实飞实验。

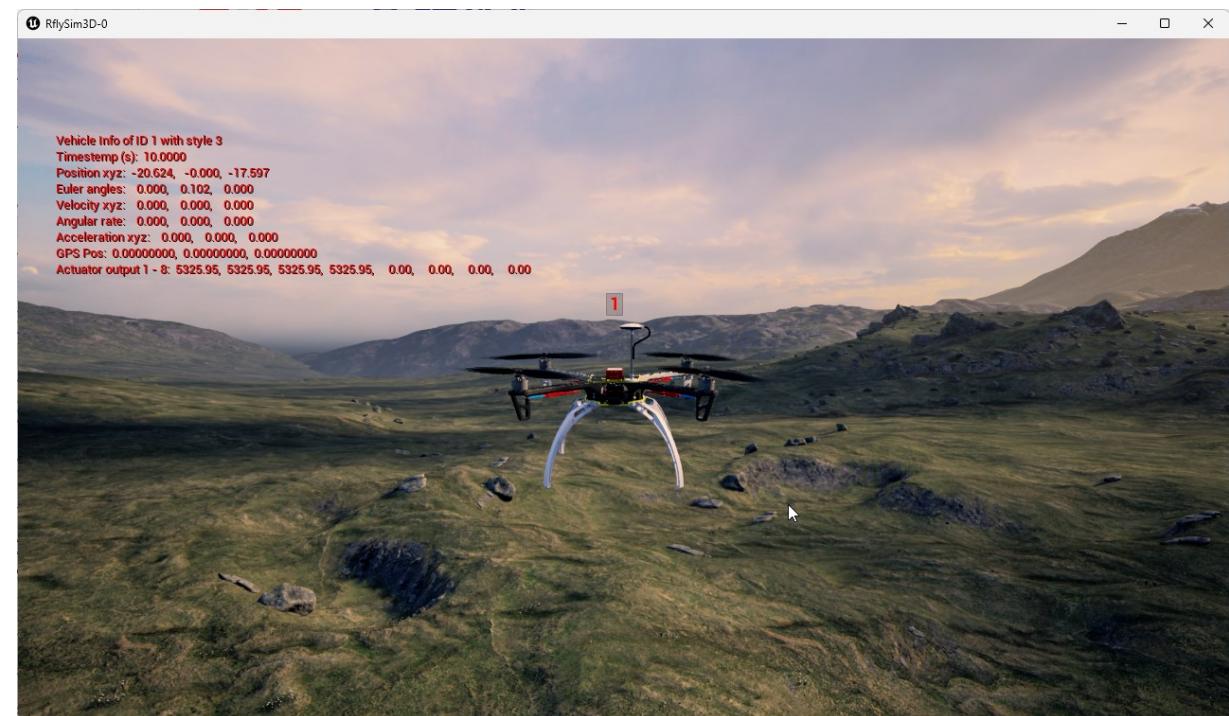
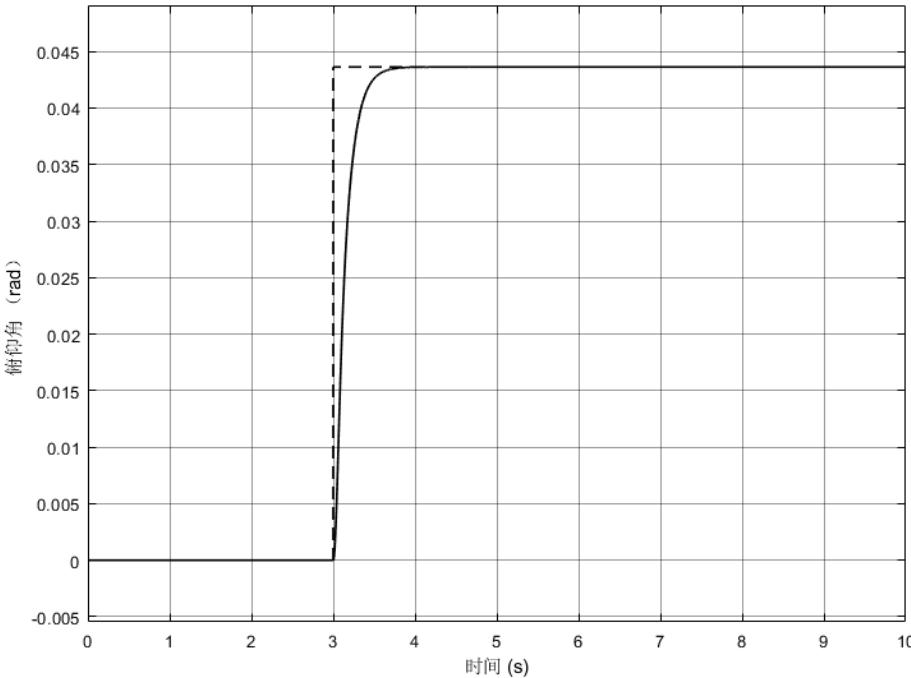
具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。



3. 基础实验案例（免费版）

3.5 姿态控制器设计实验

实验效果如下(部分):



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.6 定点位置控制器设计实验

本实验的目标如下：

- 1、复现四旋翼Simulink仿真，分析控制作用在 o_bx_b 轴和 o_by_b 轴的解耦；对系统进行扫频以绘制bode图，分析闭环位置控制系统稳定裕度。
- 2、调节PID控制器的相关参数改善系统控制性能；在得到满意参数后，对系统进行扫频以绘制Bode图。
- 3、建立位置控制通道的传递函数模型，使用MATLAB“ControlSystemDesigner”设计校正控制器，并调节系统误差、相对裕度等参数。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.6 定点位置控制器设计实验

实验效果如下(部分):

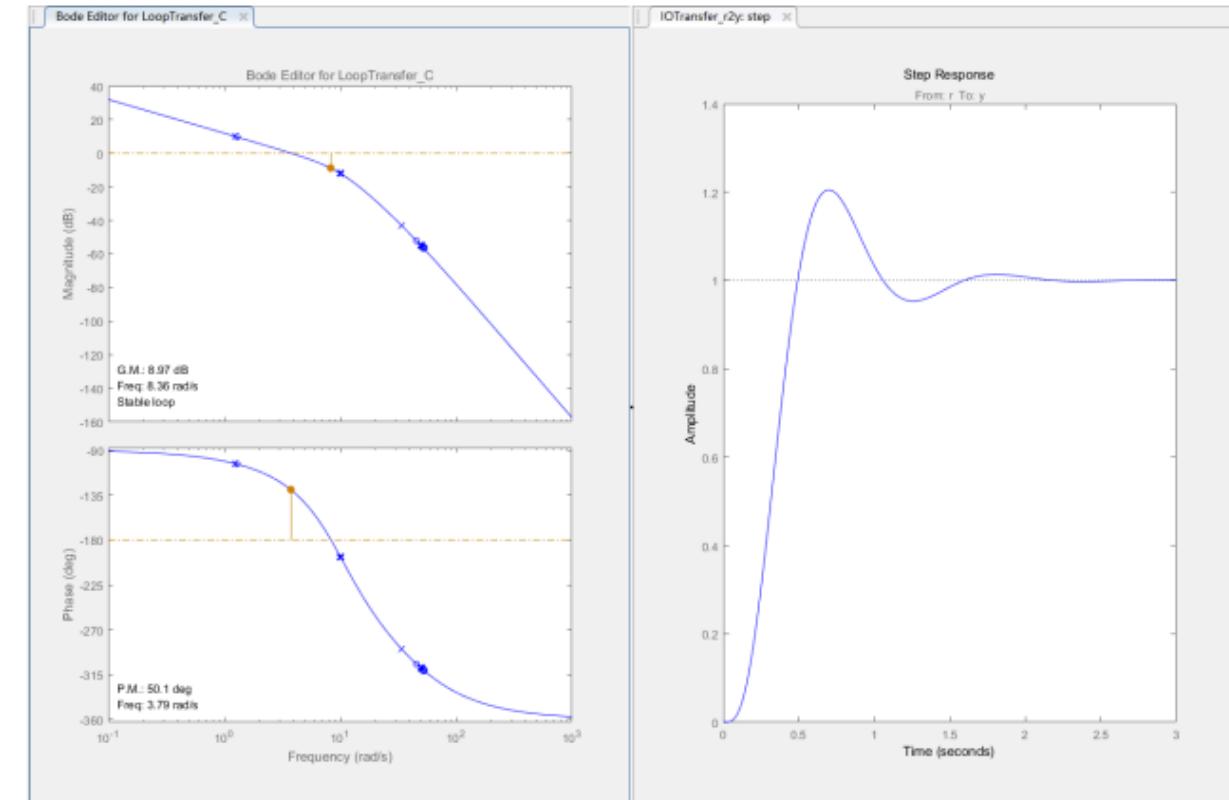
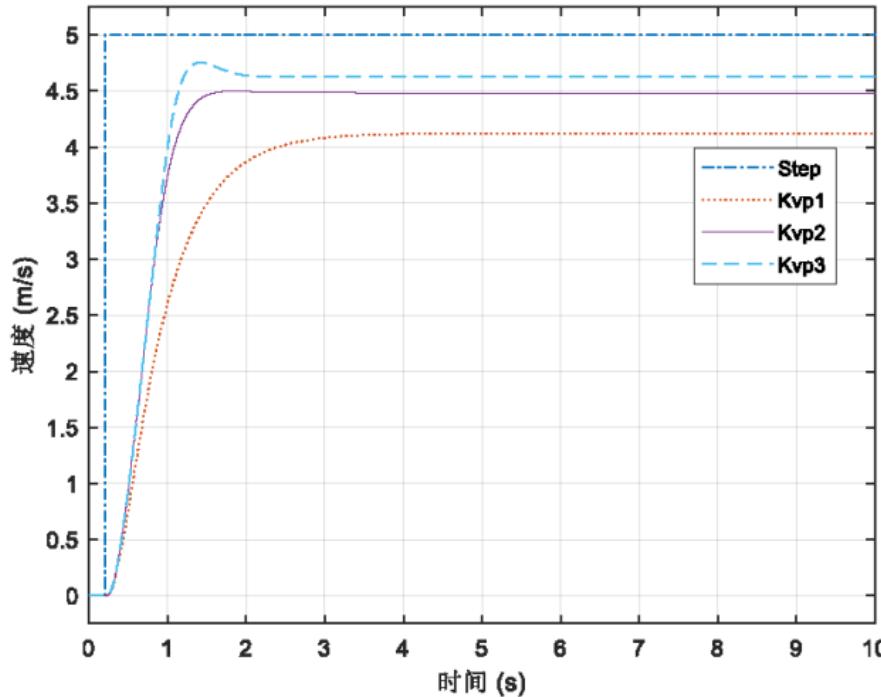


图. 不同比例项系数下的阶跃响应

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.7 半自主控制模式设计实验

本实验的目标如下：

- 1、在基于Simulink的控制器设计与仿真平台上，复现仿真实验分析四旋翼姿态和位置响应的特点；
- 2、在自稳模式的基础上改成定高模式。根据实验分析，与自稳模式相比，多旋翼在定高模式下姿态和位置输出值的变化。
- 3、在自稳模式的基础上改成定点模式。根据实验分析，与自稳模式相比，多旋翼在定点模式下姿态和位置输出值的变化；并利用三段拨码开关实现三种模式的自由切换。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

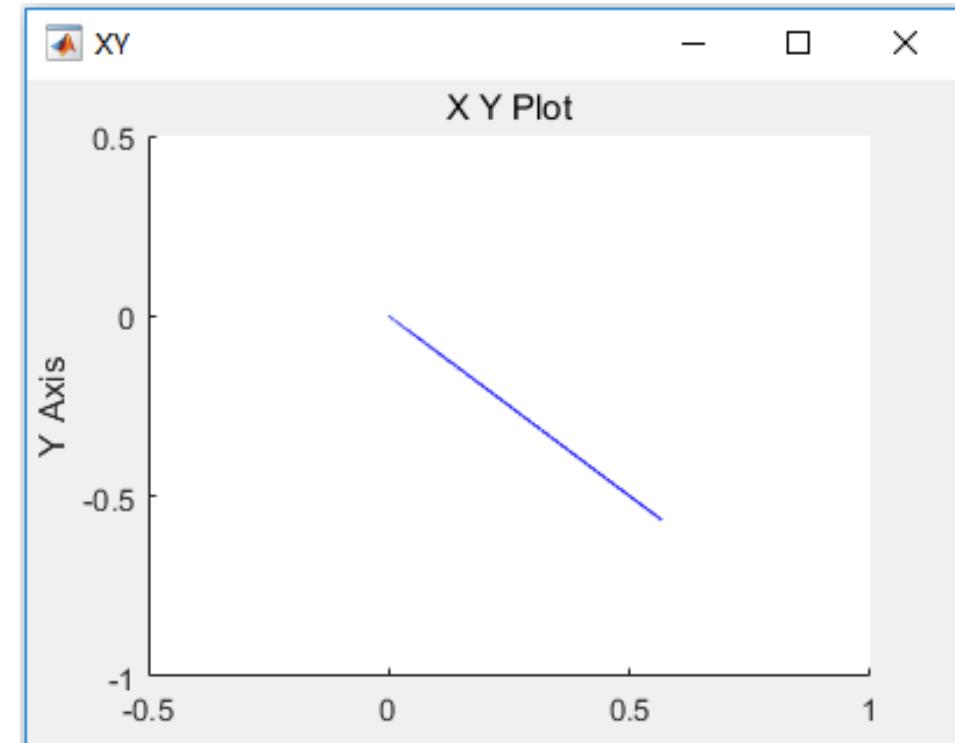
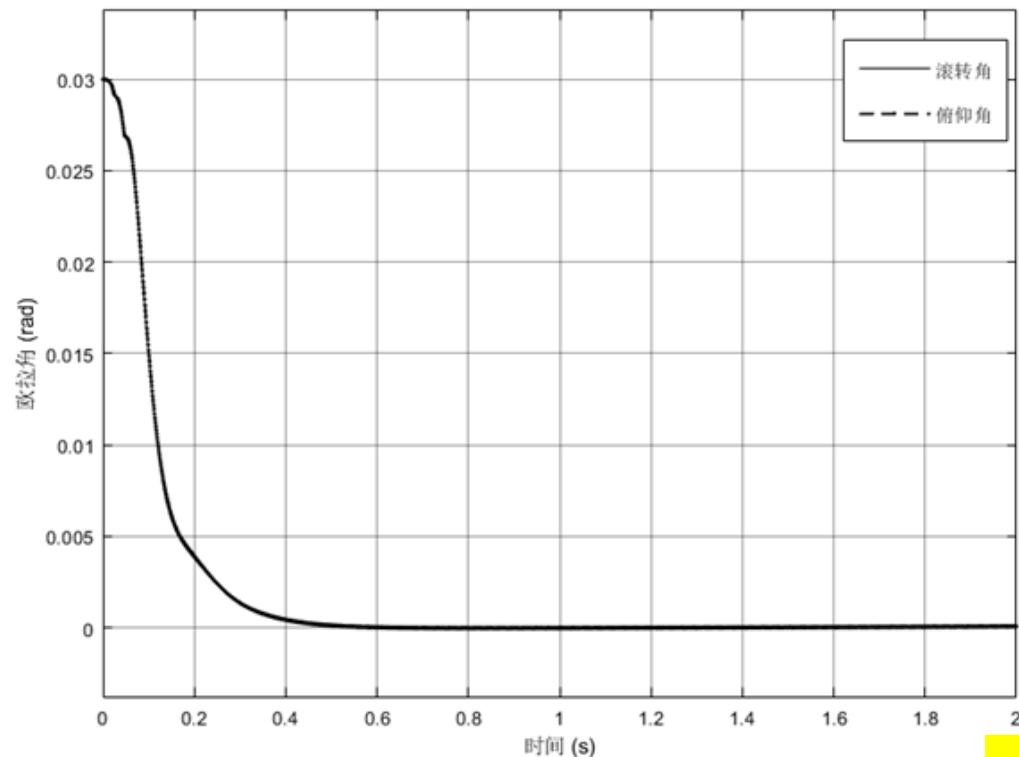
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.7 半自主控制模式设计实验

实验效果如下(部分):



更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.8 失效保护逻辑设计实验

本实验的目标如下：

- 1、在Simulink仿真环境中，在手动模式下，实现飞行器的返航与着陆，并记录和分析仿真结果。
- 2、在基础实验的基础上，添加相应状态转移，在手动模式下，实现飞行器的返航和着陆，并且返航和着陆之间可以相互切换。
- 3、在前面实验的基础上，增加遥控器断电失联事件，完成新的模态和切换设计，即加入失效返航和失效着陆两个状态，完成状态机的设计。

具体实验原理请学习进阶版课程，实验操作步骤见文件[readme.pdf](#)。

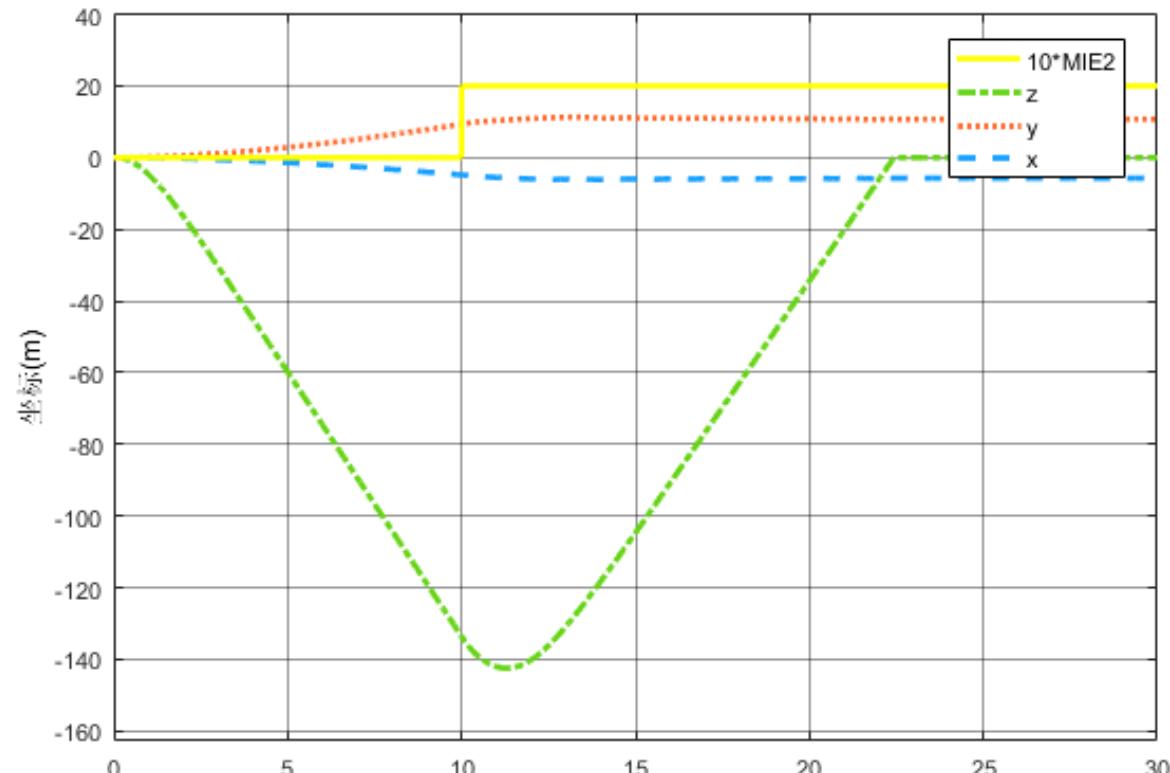
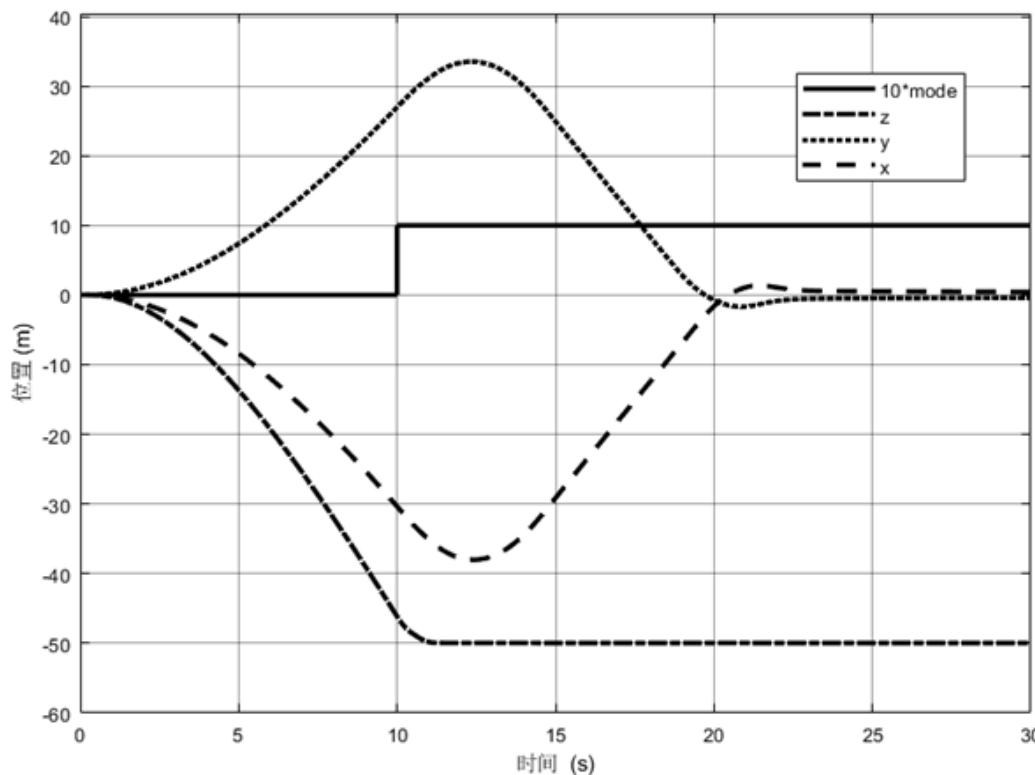
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.8 失效保护逻辑设计实验

实验效果如下(部分):



更多详细接口请见：[API](#), 更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.9 PX4模块替换实验

本实验的目标如下：

实现利用生成的Simulink代码快速替换PX4控制软件的某些原生模块(传感器、滤波器、姿态控制器等)，该实验提供两种方法来实现，分别是：

- 1、打开“Firmware\src\modules\ekf2\ekf2_main.cpp”文件，手动注释掉需屏蔽的模块代码；
- 2、修改PX4模块启动脚本文件“Firmware\ROMFS\px4fmu_common\init.d\rcS”，并注释掉需屏蔽的模块。

具体实验原理请学习进阶版课程。实验操作步骤见文件[readme.pdf](#)。

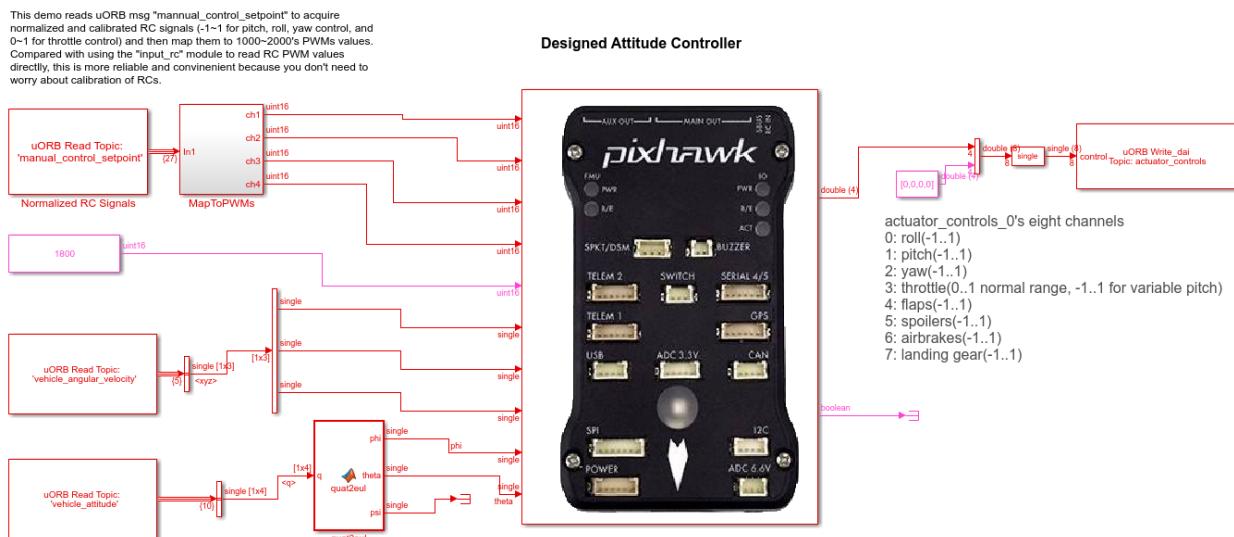
更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



3. 基础实验案例（免费版）

3.9 PX4模块替换实验

本实验对“Position & Attitude Estimator”滤波器模块进行了屏蔽，将2.3小节(姿态控制器设计实验)进行改造，搭建了“Exp6_ReplacePX4AttitudeCtrler.slx”模型，如下：



注：在本实验开发完成后，请务必归将修改的代码归回原位，以免影响其他功能的开发。

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



大纲

1. 实验平台配置

6. 扩展案例（完整版）

2. 关键接口介绍

7. 小结

3. 基础实验案例（免费版）

4. 进阶接口实验（个人版）

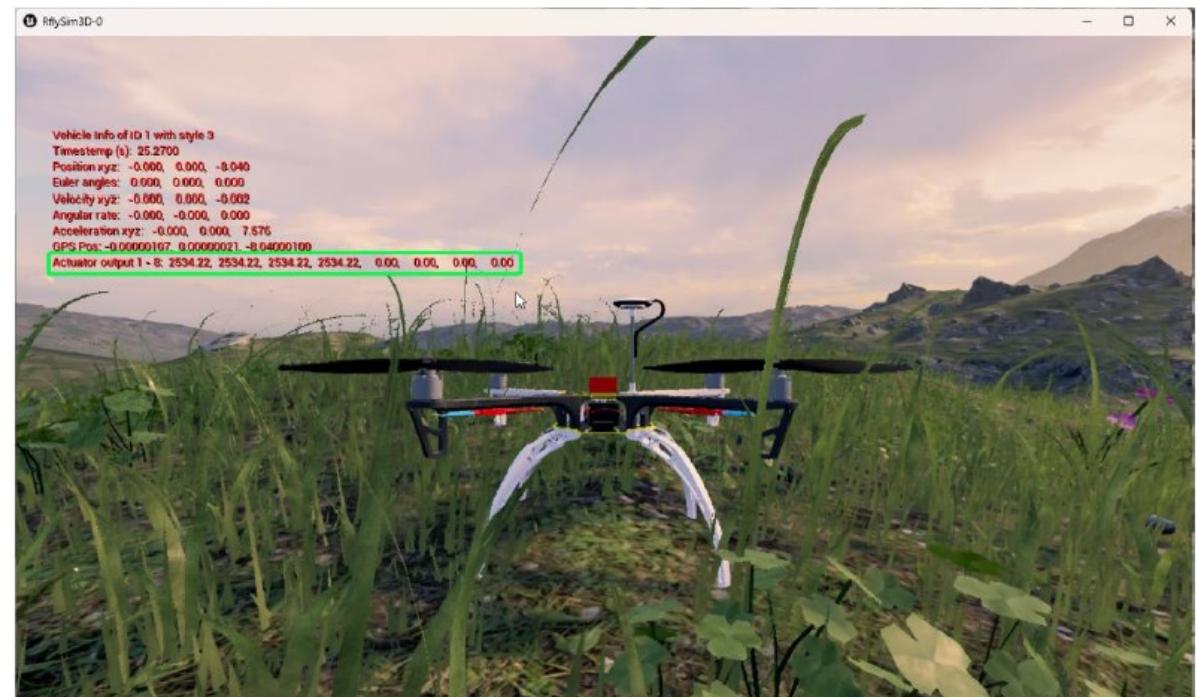
5. 进阶案例实验（集合版）



4. 进阶接口实验（个人版）

4.1 自定义屏蔽PX 4中任意模块输出

本实验 通过替换 PX4 中的姿态角速率环的 uORB 消息“`actuator_control_0`”语句 和替换修改好的 CPP 文件替换的方式 屏蔽 PX4 相关功能 为例详细阐述实验步骤。具体操作步骤见文件 [readme.pdf](#), (部分)实验效果如下：



更多详细接口请见：[API](#), 更多例程请见：[Readme](#)。



4. 进阶接口实验（个人版）

4.2 重命名PX 4 应用名称实验：

基于 PX 4 软件系统中的多进程运行状态， MATLAB 自动代码生成的 PX 4 应用名称为： px 4_simulink_a pp，本实验可将其进行重命名并新增创建一个全新的应用在 PX 4 软件系统中并编译。具体操作步骤见文件 [readme.pdf](#)，(部分)实验效果如下：

```
>> PX4AppName 'rfly_simulink_app'  
Firmware目录中已存在rfly_simulink_app目录。  
当前的编译命令为: px4_fmu-v5_default  
成功找到px4_fmu-v5_default的cmake文件  
重命名完成。  
开始重新添加px4_simulink_app模版...
```

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



4. 进阶接口实验（个人版）

4.3 加载PX 4应用 实验：

RflySim 平台支持 加载自定义开发完成的 PX 4 应用，根据本实验所提供的 PX 4 应用程序可直接加载到 PX 4 软件系统中进行固件编译。具体操作步骤见文件 [readme.pdf](#)，(部分)实验效果如下：

```
>> PX4AppLoad('C:\PX4PSP\rfly_simulink_app')
当前的编译命令为: px4_fmu-v5_default
Firmware目录中已存在rfly_simulink_app目录。
当前的编译命令为: px4_fmu-v5_default
成功找到px4_fmu-v5_default的cmake文件
重命名完成。
开始重新添加px4_simulink_app模版...
```

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



4. 进阶接口实验（个人版）

4.4 创建多个PX 4 应用实验：

基于 PX 4 软件系统中的多 进程运行状态， MATLAB 自动代码生成的 PX 4 应用 名称为： px4_simulink_app ， 本实验可将其 进行重命名后， 再通过 MATLAB 自动代码生成新的 PX 4 应用， 以此即可实现同时创建多个 PX 4 应用。 具体操作步骤见文件 [readme.pdf](#)， (部分)实验效果如下：

```
>> PX4AppName 'rfly_simulink_app'  
Firmware目录中已存在rfly_simulink_app目录。  
当前的编译命令为: px4_fmu-v5_default  
成功找到px4_fmu-v5_default的cmake文件  
重命名完成。  
开始重新添加px4_simulink_app模版...
```

更多详细接口请见：[API](#)， 更多例程请见：[Readme](#)。



大纲

1. 实验平台配置

6. 扩展案例（完整版）

2. 关键接口介绍

7. 小结

3. 基础实验案例（免费版）

4. 进阶接口实验（个人版）

5. 进阶案例实验（集合版）



5. 进阶案例实验（集合版）

5.1 ADRC姿态控制器设计实验：

把系统的模型作用当做系统的内扰，那么它连同系统的外扰一起，均可作为对系统的扰动。这个补偿分量并不区分内扰和外扰，直接检测并补偿他们的总和作用—对系统的总扰动。由于这个分量的补偿作用，被控对象实际上被化成积分器串联型而易于构造出理想的控制器，这个补偿分量的补偿作用实质上是一种抗扰作用。因此我们将此控制器称为“自抗扰控制器”（ADRC）。具体操作步骤见文件[readme.pdf](#)。

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



5. 进阶案例实验（集合版）

5.2 ADRC位置控制器设计实验：

把系统的模型作用当做系统的内扰，那么它连同系统的外扰一起，均可作为对系统的扰动。这个补偿分量并不区分内扰和外扰，直接检测并补偿他们的总和作用——对系统的总扰动。由于这个分量的补偿作用，被控对象实际上被化成积分器串联型而易于构造出理想的控制器，这个补偿分量的补偿作用实质上是一种抗扰作用。因此我们将此控制器称为“自抗扰控制器”（ADRC）。具体操作步骤见文件[readme.pdf](#)。

更多详细接口请见：[API](#)，更多例程请见：[Readme](#)。



大纲

1. 实验平台配置

6. 扩展案例（完整版）

2. 关键接口介绍

7. 小结

3. 基础实验案例（免费版）

4. 进阶接口实验（个人版）

5. 进阶案例实验（集合版）



6. 扩展案例（完整版）

开发中。 . .



大纲

1. 实验平台配置

6. 扩展案例（完整版）

2. 关键接口介绍

7. 小结

3. 基础实验案例（免费版）

4. 进阶接口实验（个人版）

5. 进阶案例实验（集合版）



7. 小结

- 本讲主要对飞行控制算法的开发课程进行讲解，分为基础实验和进阶实验两部分，使各位学员能够尽快熟悉多旋翼的理论设计、RflySim平台仿真、物理真机控制等开发流程。
- 基础实验是基于RflySim平台软件在环和硬件在环仿真流程学习为主，进阶实验是从多旋翼理论设计和建模实验→估计实验→控制实验→决策实验的学习路线进行教学。

如有疑问，请到<https://doc.rflysim.com/>查询更多信息。



RflySim更多教程



扫码咨询与交流



飞思RflySim技术交流群



谢谢！