

---

# 1. 实验名称及目的

## 1.1. 实验名称

基于扩张状态观测器的飞控设计实验（仅限完整版及以上版本）

## 1.2. 实验目的

该方案适用于在扰动和非确定性条件下的飞行器控制问题。通过将飞行器解耦成多个级联 SISO 系统，系统包括一个名义模型和一个不确定项，该不确定项包括模型/识别的不确定性，控制失配，以及外部扰动。为了估计系统状态和总的不确定性，设计了一个扩张状态观测器 (ESO)，利用 ESO 的输出，控制器在线补偿总的不确定性。

## 1.3. 关键知识点

一般的单输入单输出(SISO)不确定非线性系统可以通过合适的微分同胚变换转化为以下标准形式：

$$\begin{cases} \dot{z} = g(x, z, \omega) \\ \dot{x} = Ax + B(f(x, t, \omega) + b(x, t, \omega)u) \\ y = Cx \end{cases}$$

其中， $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$  代表系统的状态， $y \in \mathbb{R}$  是输出， $u \in \mathbb{R}$  是控制输入， $\omega$  表示外部干扰。矩阵  $A \in \mathbb{R}^{n \times n}$ ， $B \in \mathbb{R}^{n \times 1}$ ，和  $C \in \mathbb{R}^{1 \times n}$  用来表示链式积分器。函数  $f(\cdot): \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$  和  $b(\cdot): \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$  都是局部利普西斯连续的，并且可以分解为已知和未知部分，如下：

$$\begin{aligned} f(x, \omega) &= f_0(x) + \Delta f(x, \omega), \\ b(x, \omega) &= b_0(x) + \Delta b(x, \omega), \end{aligned}$$

其中， $f_0(\cdot), b_0(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}$  是已知的标准名义模型，对于  $x \in \mathbb{R}^n$ ， $b_0(x) \neq 0$ ，而  $\Delta f(\cdot), \Delta b(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}$  分别表示潜在的建模误差和控制不匹配。注意到，我们只能实时获取系统的确切状态  $x_1$ 。根据自抗扰控制(ADRC)的基本理念，通过定义扩张状态  $x_{n+1} \triangleq \Delta f(x, \omega) + \Delta b(x, \omega)u$ ，系统的总不确定性可以由扩张状态观测器(ESO)估计。

基于  $x$  和  $x_{n+1}$  的值，可以设计各类控制器。

考虑如下在三维平面的飞行器，

$$\begin{cases} \dot{x} = V_x \\ \dot{V}_x = f_x \cdot u_x + g_x \\ \dot{y} = V_y \\ \dot{V}_y = f_y \cdot u_y + g_y \end{cases}$$

其 x 方向与 y 方向可以解耦为两个非线性 SISO 系统，通过任意一种控制器令无人机跟踪椭圆形轨迹：

$$\begin{cases} \dot{x}_r = -\frac{\pi}{20} \sin\left(\frac{\pi}{20}t\right) \\ \ddot{x}_r = -\left(\frac{\pi}{20}\right)^2 \cos\left(\frac{\pi}{20}t\right) \\ \dot{y}_r = \frac{\pi}{10} \cos\left(\frac{\pi}{20}t\right) \\ \ddot{y}_r = -2 \times \left(\frac{\pi}{20}\right)^2 \sin\left(\frac{\pi}{20}t\right) \end{cases}$$

通过在外环设计扩张状态观测器，可以观测出无人机在 x 和 y 方向上的速度和总不确定性。

## 2. 实验效果

分别设计了 SITL，HITL，实飞实验验证了扩张状态观测器的效果。

## 3. 文件目录

文件夹/文件名称	说明
SITL	包含 SITL 所需的 slx 文件和参数文件
HITL&FLY	包含 HITL 和实飞所需的 slx 文件和参数文件

## 4. 运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 <sup>①</sup>	1
2	RflySim 工具链	Pixhawk 6X 或 Pixhawk 6X mini <sup>②</sup>	1
3	MATLAB 2022b 及以上	飞思 X200 飞机 <sup>②</sup>	1
4		遥控器 <sup>③</sup>	1
5		遥控器接收器	1
6		数据线、杜邦线等	若干

①：推荐配置请见：<https://doc.rflysim.com>

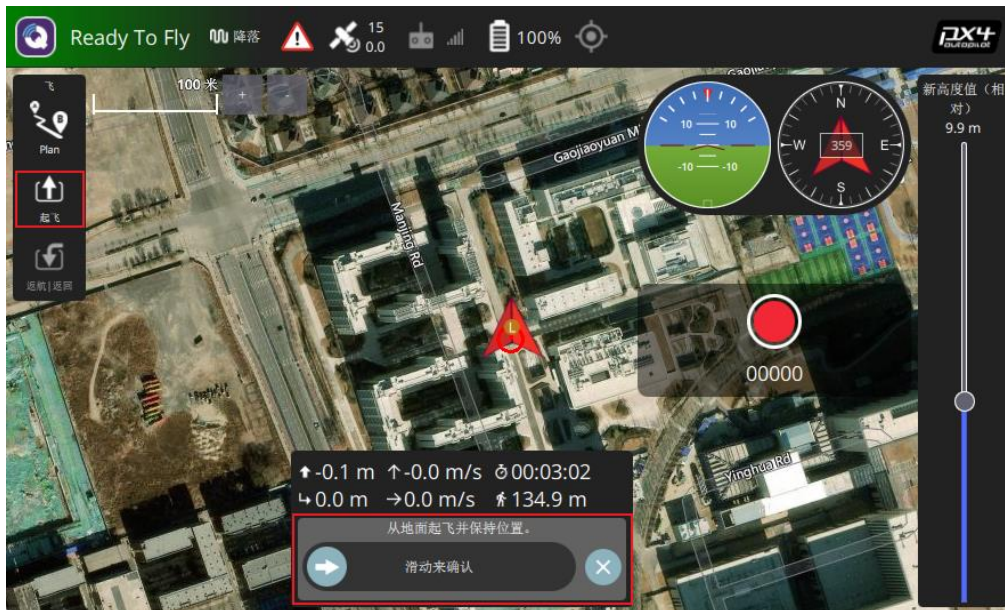
②：须保证平台安装时的编译命令为：px4\_fmu-v6x\_default，固件版本为：1.12.3。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

③：本实验演示所使用的遥控器为：天地飞 WFLY-ET10、配套接收器为：WFLY-RF2 09S。遥控器相关配置见：<https://rflsim.com/doc/zh/B/3.1ET10.html>

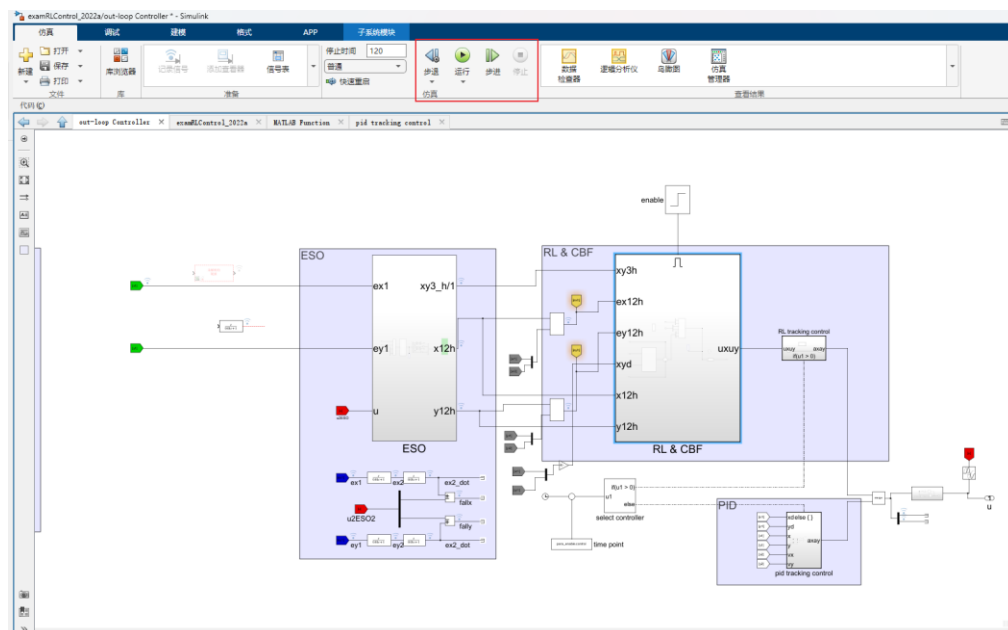
## 5. 实验步骤

### 5.1. SITL 实验

管理员运行"\*\桌面\RflyTools\SITLRun.lnk"或"\*\PX4PSP\RflySimAPIs\SITLRun.bat"文件，在弹出的 CMD 对话框中输入仿真数量：1，即可自动启动 RflySim3D、CopterSim、Q GroundControl 软件，等待 CopterSim 的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。在 QGC 中点击起飞，滑动解锁。



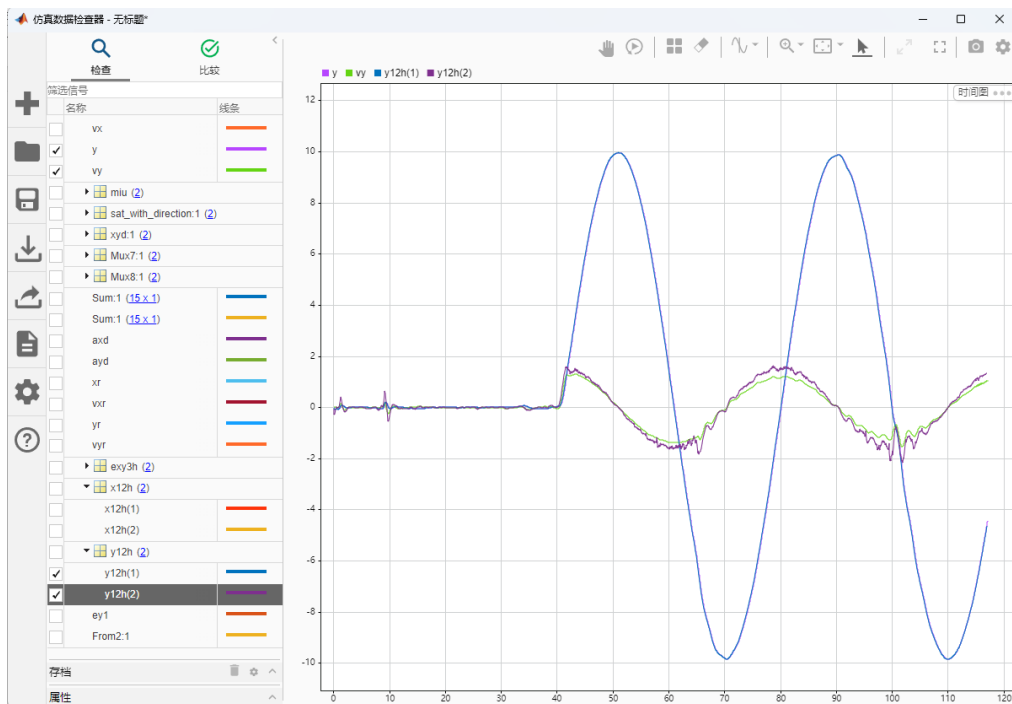
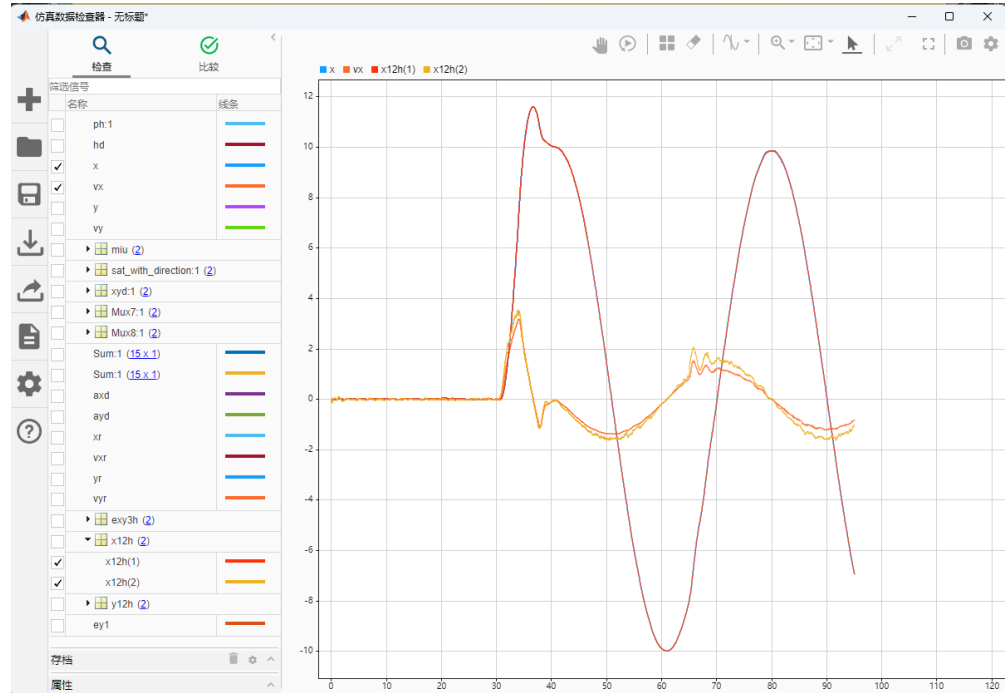
等待飞机起飞至指定高度，用 MATLAB 运行 [SITL/main\\_param.m](#)，然后运行刚 [SITL/RL\\_SITL.slx](#) 打开的文件。



观察无人机跟踪圆形轨迹一段时间后，打开数据检查器。



如下图所示，可以看到 ESO 可以很好的估计无人机的 x 方向和 y 方向上的各个状态，控制器基于 ESO 的观测值，控制无人机跟踪既定的轨迹圆。





## 5.2. HITL 实验

在进行硬件在环仿真前，请保证平台安装设置如下，第 10 项中选择为：actuator\_controls\_0，即屏蔽 PX4 软件中的 actuator\_controls\_0 消息输出。

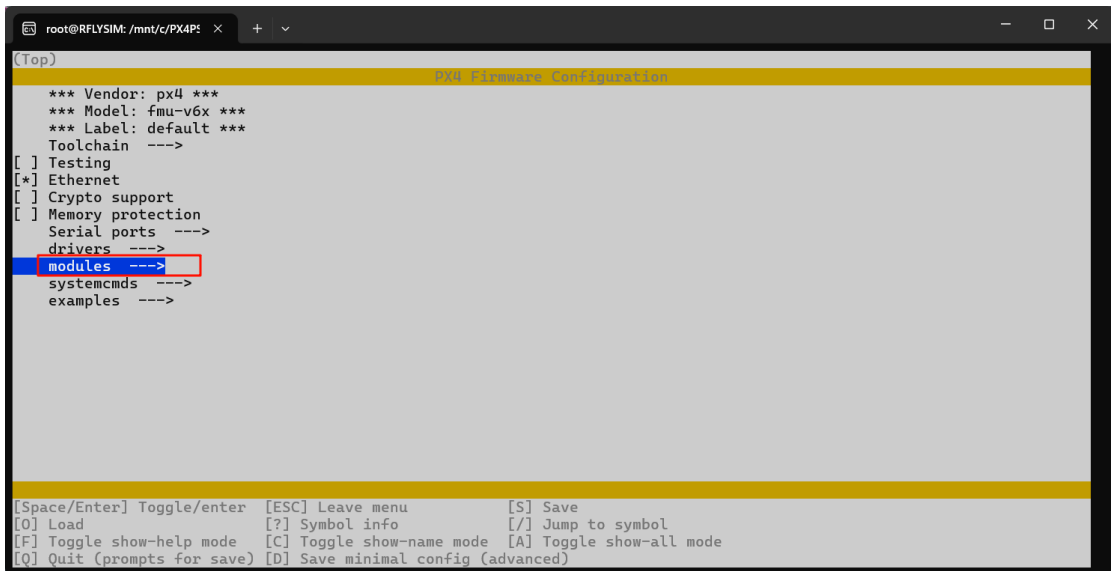
工具箱—一键安装脚本 RflySimAdv3Full-V3.02-20240315	
1.工具包安装路径	C:\PX4PSP
2.PX4固件编译命令：见Firmware\boards目录，例如px4_fmu-v5_default、droneyee_zyfc-h7_default等	px4_fmu-v6x_default
3.PX4固件版本（1: PX4-1.7.3, ..., 5: PX4-1.11.3, 6: PX4-1.12.3, 7: PX4-1.13.2, 8: PX4-1.14.*）	7
4.PX4固件编译器（1: WinWSL[通用], 2: Msys2[适用版本≤PX4-1.8], 3: Cygwin[适用≥PX4-1.8]	1
5.是否重新安装PSP工具箱(是：重装工具箱，否：维持现有安装)	否
6.是否重新安装其他依赖程序包（CopterSim、QGC地面站、硬件在环仿真软件等，约5分钟）	否
7.是否重新配置固件编译器编译环境（是：全新安装编译器，否：维持原样，重装约5分钟）	否
8.是否重新部署PX4固件代码（是：全部署代码，否：维持现状，大约5分钟）	是
9.是否预先选定命令编译固件（是：全新编译固件，否：维持现状，大约5分钟）	是
10.是否屏蔽PX4官方控制器输出(使用Simulink控制器选"是", 使用PX4官方控制器选"否")	actuator_controls_0

确定 取消

注：本实验的 HIL 和实飞模型程序文件较大，在 MATLAB/Simulink 中编译前需要将 P

X4 软件中部分不用的模块禁用掉步骤如下：

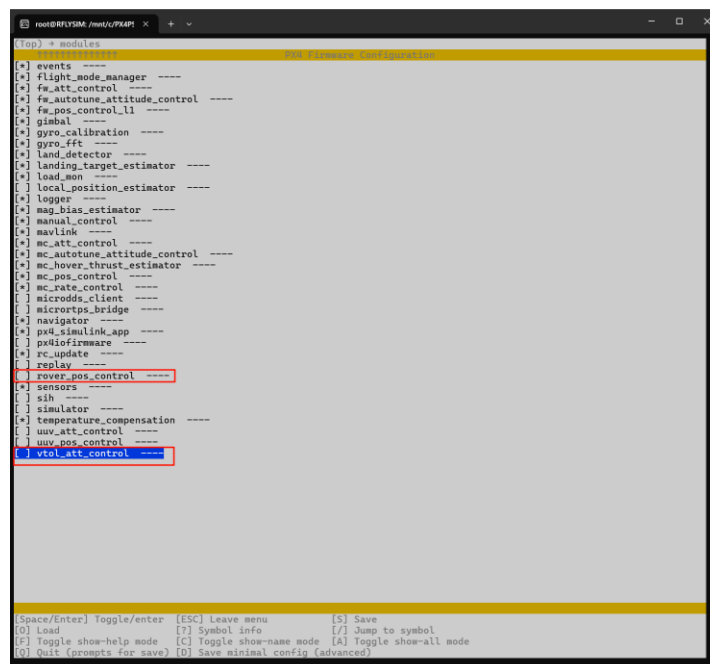
打开本平台的"E:\桌面\RflyTools\WinWSL.lnk"子系统，在 cmd 窗口输入：`make px4_fmuv6x_default boardconfig`(编译命令请以自己飞控为准)，在弹出的界面中选择：`modules`。



```
root@RFLYSIM: /mnt/c/PX4P1 x + v
[Top]
PX4 Firmware Configuration
*** Vendor: px4 ***
*** Model: fmu-v6x ***
*** Label: default ***
Toolchain --->
[ ] Testing
[*] Ethernet
[ ] Crypto support
[ ] Memory protection
Serial ports --->
drivers --->
modules --->
systemcmds --->
examples --->

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                    [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

利用上下方向键，移动光标到需要禁用的模块上，按下 Enter 后即可，如下图所示为禁用了 `rover_pos_control` 和 `vtol_att_control` 模块。之后按下“S”键保存，按下“Enter”键退出。



```
root@RFLYSIM: /mnt/c/PX4P1 x + v
[Top] > modules
PX4 Firmware Configuration
[*] events ---
[*] flight_mode_manager ---
[*] fw_att_control ---
[*] fw_autotune_attitude_control ---
[*] fw_pos_control_ll ---
[*] gimbal ---
[*] gyro_calibration ---
[*] gyro_fft ---
[*] land_detector ---
[*] landing_target_estimator ---
[*] load_mon ---
[*] local_position_estimator ---
[*] logger ---
[*] mag_bias_estimator ---
[*] manual_control ---
[*] mavlink ---
[*] mc_att_control ---
[*] mc_autotune_attitude_control ---
[*] mc_hover_thrust_estimator ---
[*] mc_pos_control ---
[*] mc_rate_control ---
[*] microdds_client ---
[*] micrortps_bridge ---
[*] navigator ---
[*] px4_simulink_app ---
[*] px4iofirmware ---
[*] rc_update ---
[*] replay ---
[*] rover_pos_control ---
[*] sensors ---
[*] shh ---
[*] simulator ---
[*] temperature_compensation ---
[*] uuv_att_control ---
[*] uuv_pos_control ---
[*] vtol_att_control ---

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                    [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

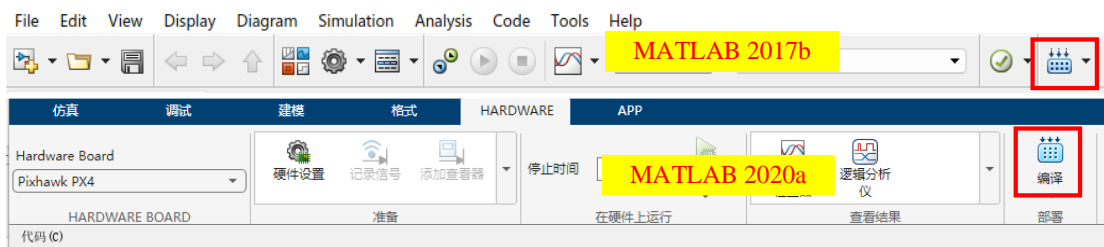
PX4 版本选择 1.12.3 及以下的版本，需要修改对应飞控的“default.cmake”文件，对于 Pixhawk 6X&6X mini 则为：`【安装目录】\Firmware\boards\px4\fmuv6x\default.cmake` 文件，其余飞控可在 `【安装目录】\Firmware\boards\px4` 文件夹中自行选择。

```

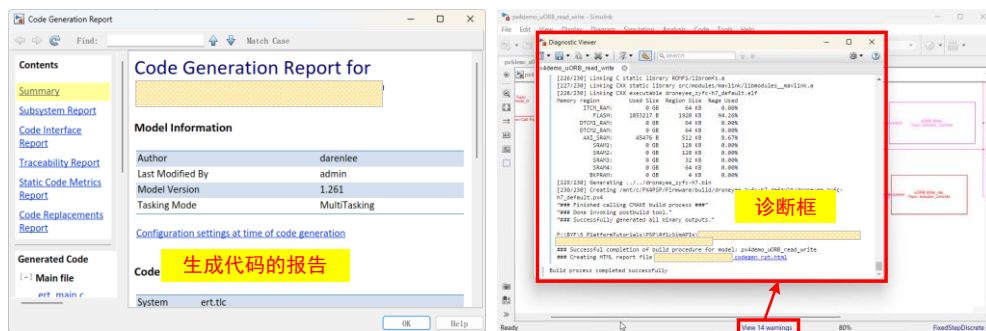
MODULES
airspeed_selector
attitude_estimator_q
camera_feedback
commander
dataman
ekf2
esc_battery
events
flight_mode_manager
fx_att_control
fx_pos_control_ll
gyro_calibration
gyro_fft
land_detector
landing_target_estimator
load_mon
#local_position_estimator
logger
mavlink
mc_att_control
mc_hover_thrust_estimator
mc_pos_control
mc_rate_control
#microtips_bridge
navigator
rc_update
rover_pos_control
sensors
sih
temperature_compensation
#uuv_att_control
#uuv_pos_control
vmount
vtol_att_control

```

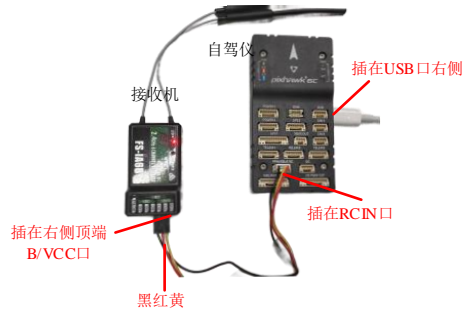
打开 MATLAB 运行 [HITL&FLY\main\\_param](#) 文件后，打开 [HITL&FLY\ESO\\_HITL\\_FL](#) [Y.slx](#) 文件，在 Simulink 中，点击编译命令。



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，左图侧为生成的编译报告。

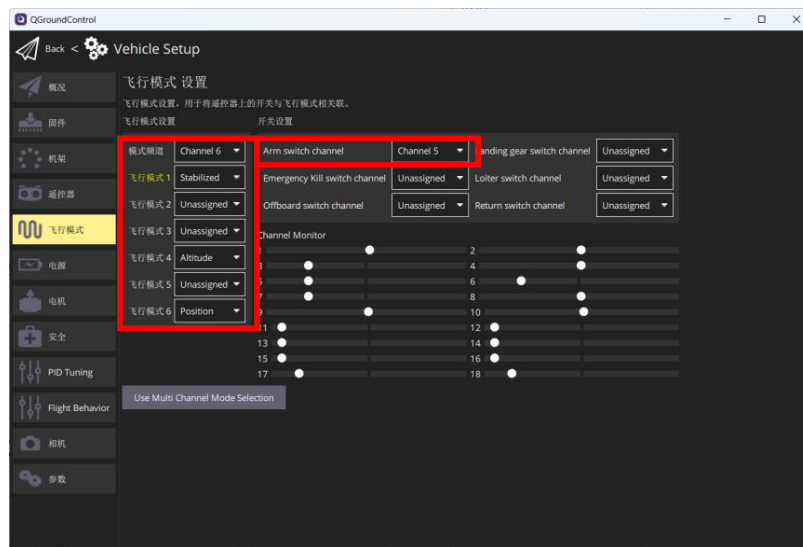
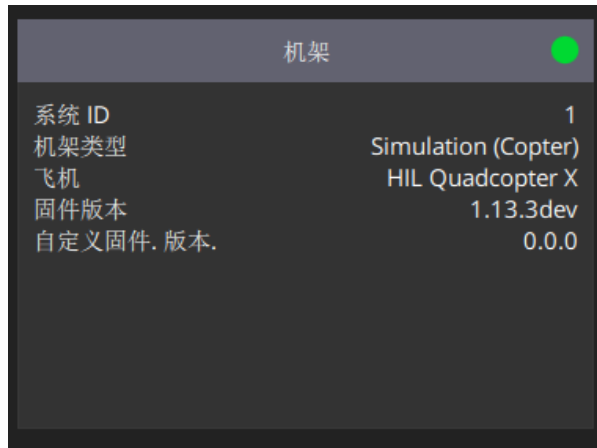


用 USB 数据线链接飞控与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行，弹出 CMD 对话框，显示正在上传固件至飞控中，等待上传成功。

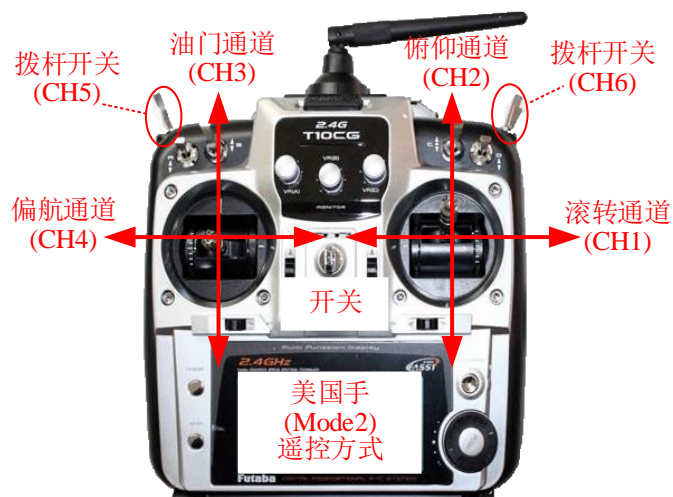


命令行窗口  
 >> PX4Upload  
 或

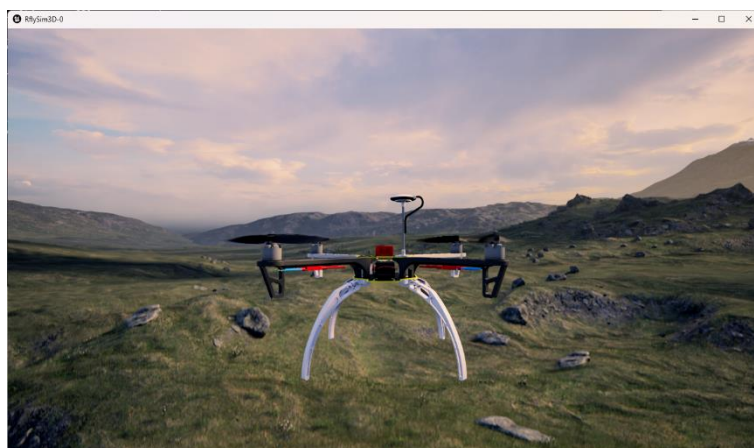
上传成功后，打开 QGroundControl 软件，确认为如下设置：



管理员运行"\*\桌面\RflyTools\SITLRun.lnk"或"\*\PX4PSP\RflySimAPIs\HITLRun.bat"文件，在弹出的CMD对话框中输入插入的飞控Com端口号，即可自动启动 RflySim3D、CopterSim、QGroundControl 软件，等待 CopterSim 的状态框中显示：PX4: GPS 3D fixed & EKF initialization finished。即可在 QGroundControl 中设置飞机起飞等操作。遥控器的设置如下图，通过控制不同的通道即可在 RflySim3D 中观察到无人机的飞行姿态，完成硬件在环仿真。**注：具体设置请见本平台的[遥控器配置手册](#)。**



拨动 CH5，解锁无人机，先用遥控器控制无人机以自稳模式或定高/定点模式起飞，然后拨动 CH7 切换至强化学习控制模式，查看控制结果，发现扩张状态观测器可以较好的观测无人机 x 方向和 y 方向上的位置和速度。



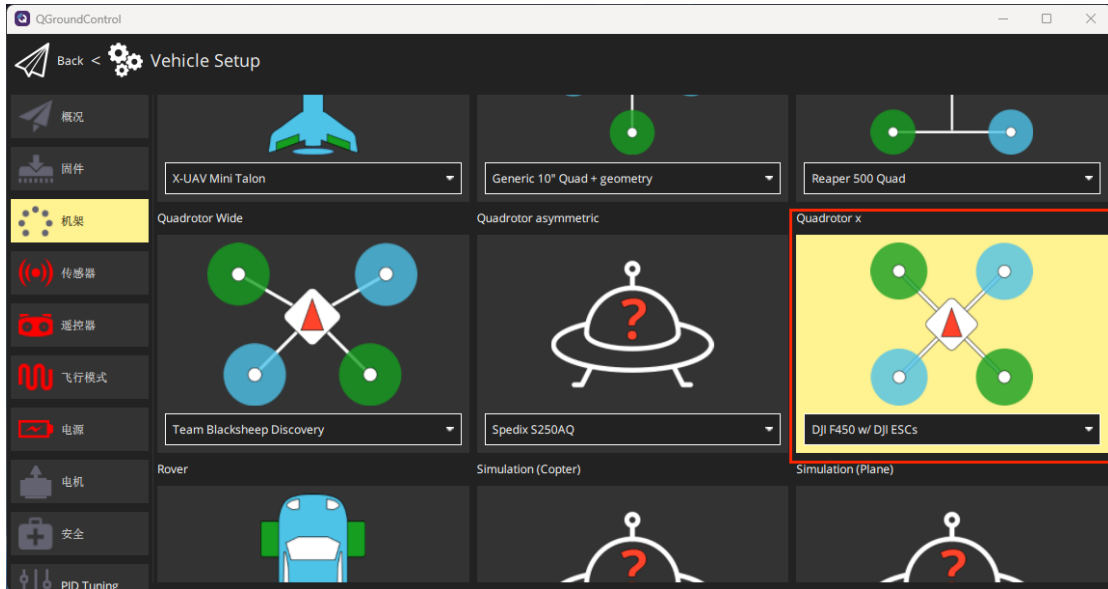
### 5.3. 官方固件实飞实验

请扫码或点击下方二维码，将本例程文件夹下：[HIL&FLY\px4\\_fmU-v6c\\_default1133.px4](#) (飞控固件)上传至飞控中。

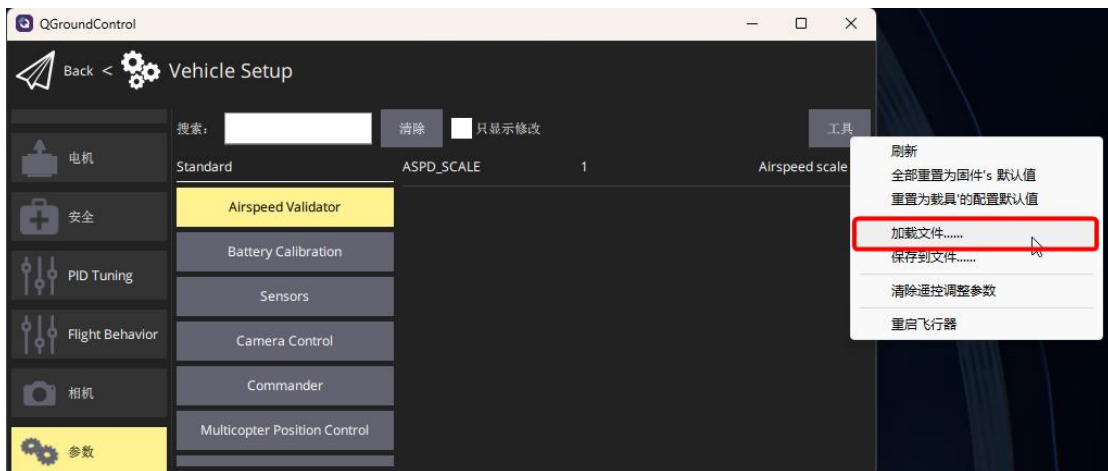


将飞机通过 USB 与电脑进行连接，打开 QGC 软件，设置机架为：DJI F450 w/DJI ESC

s;

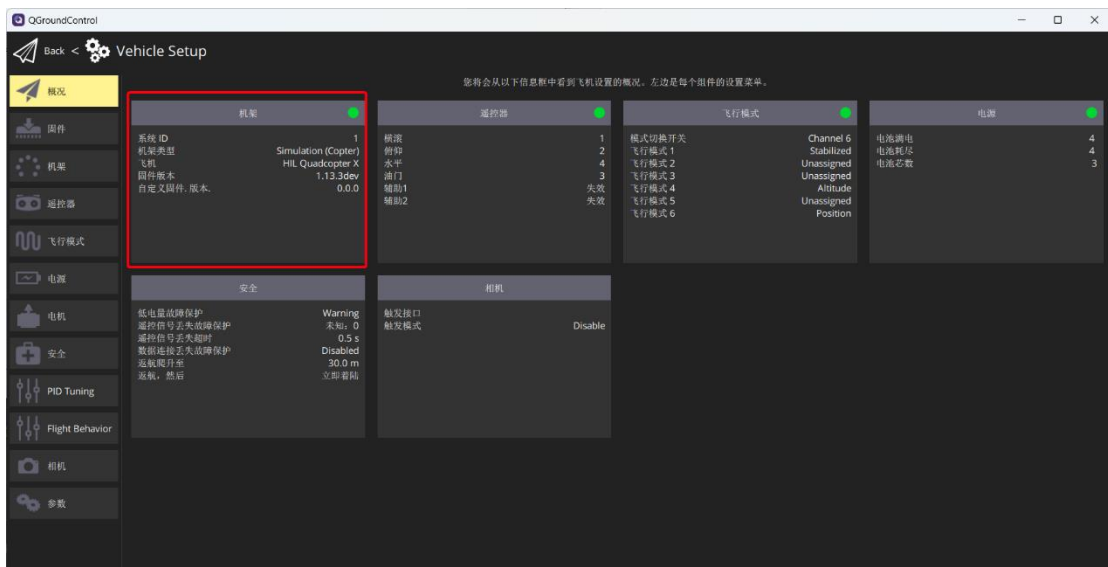


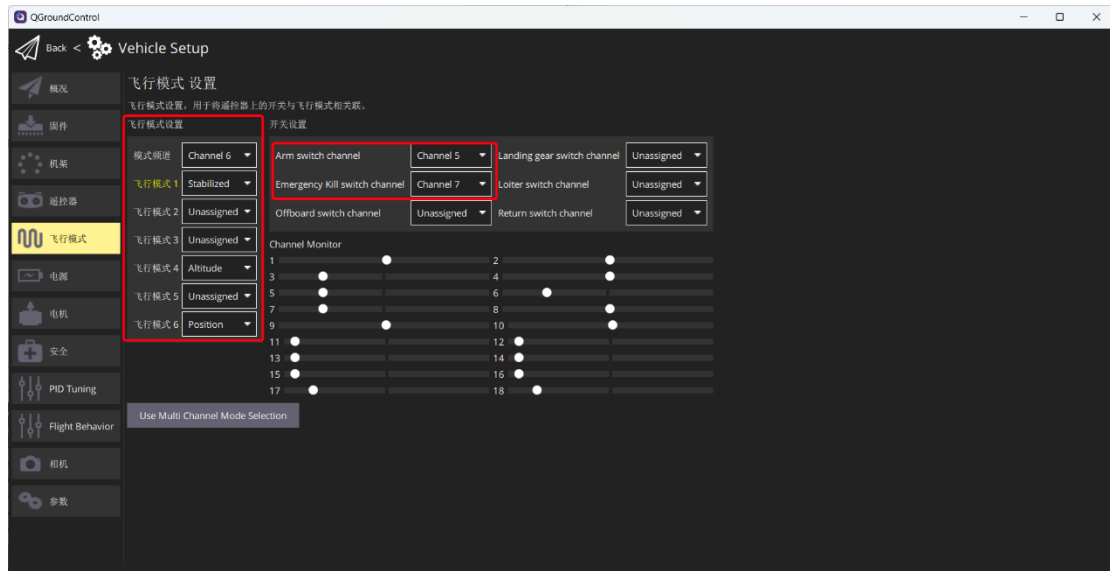
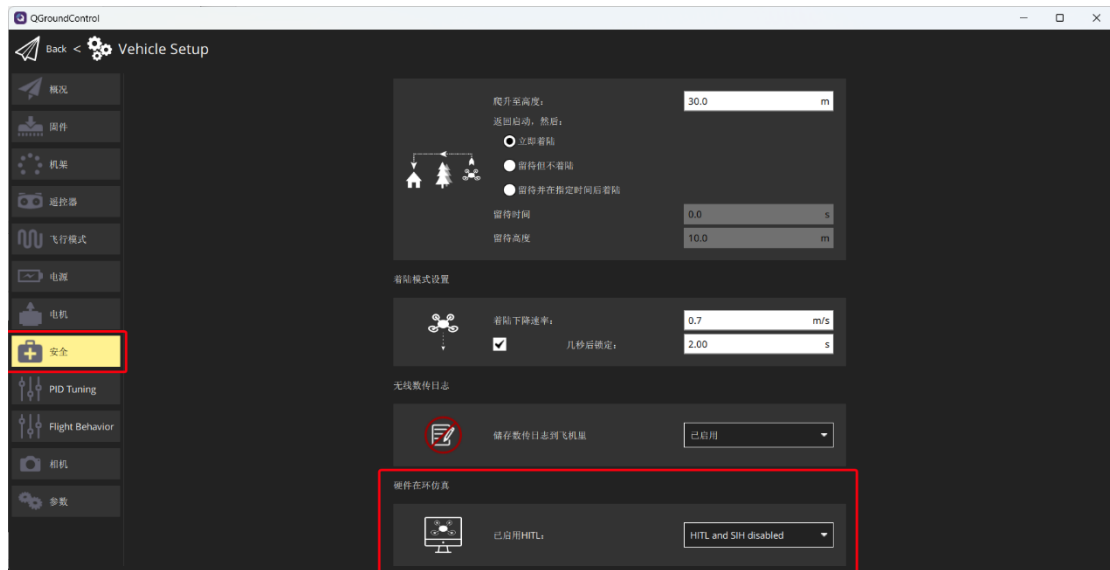
选择加载本例程文件夹下的参数文件：[X450.params](#) 文件。



加载成功后，断开飞机，再次进行连接飞机确保所有设置均已完成。

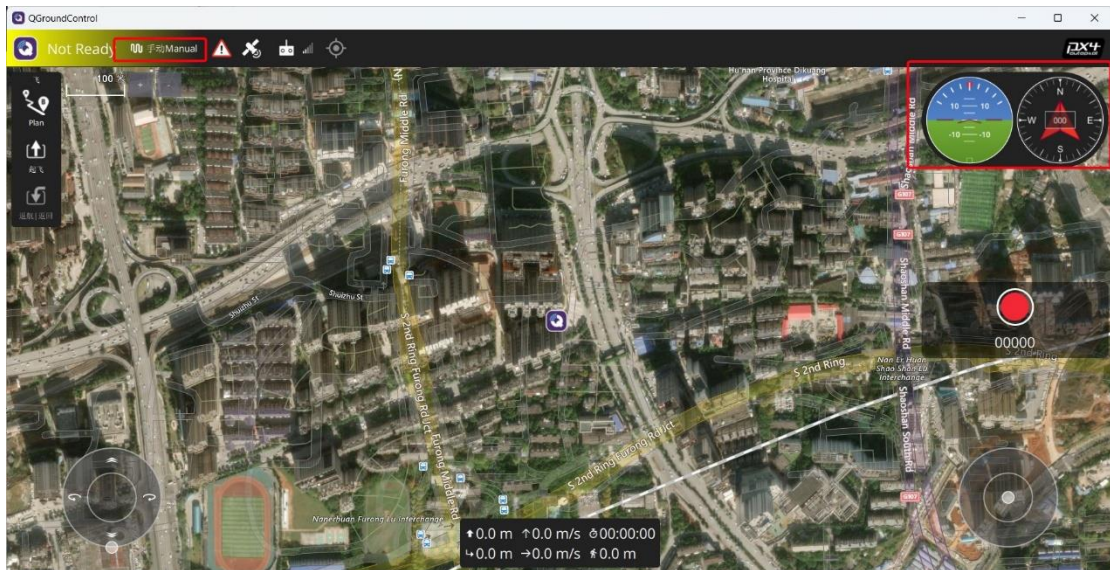
打开 QGC 地面站在其中进行如下设置：





注：该飞行模式中的各通道设置须于遥控器中所设置的通道对映。

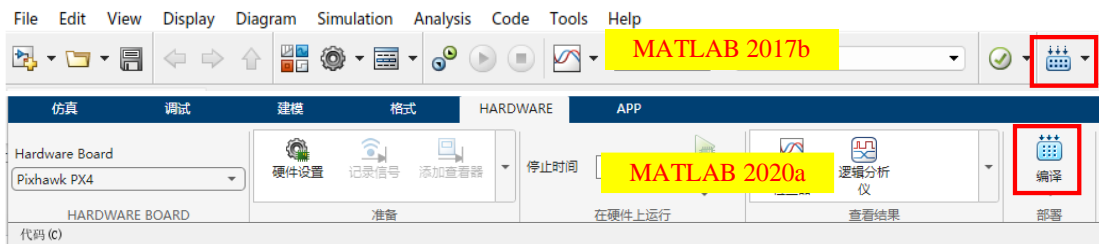
手动摆动飞机，查看 QGC 右上角仪表盘的显示情况，并确认飞机状态切换到手动 Manual 模式下。



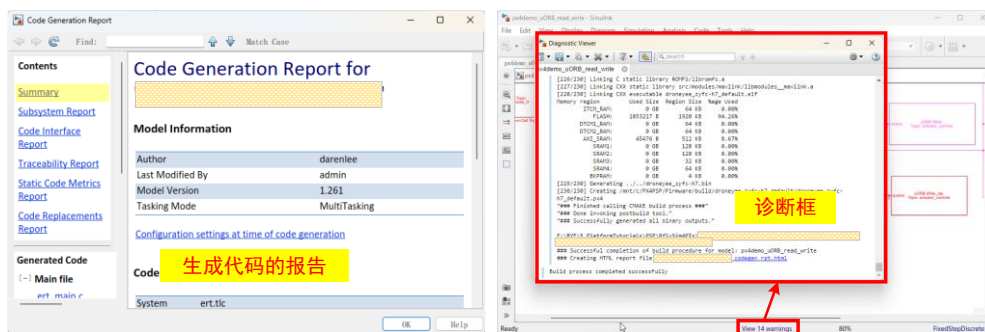
请在指定飞场进行无人机实飞，若正常起飞，说明无人机状态良好；若未正常起飞，请检查传感器校准、参数设置等，具体请联系飞机生产厂家进行解决。**请务必保证飞机状态良好的情况下，再进行下一步操作。**

## 5.4. 官方固件实飞实验

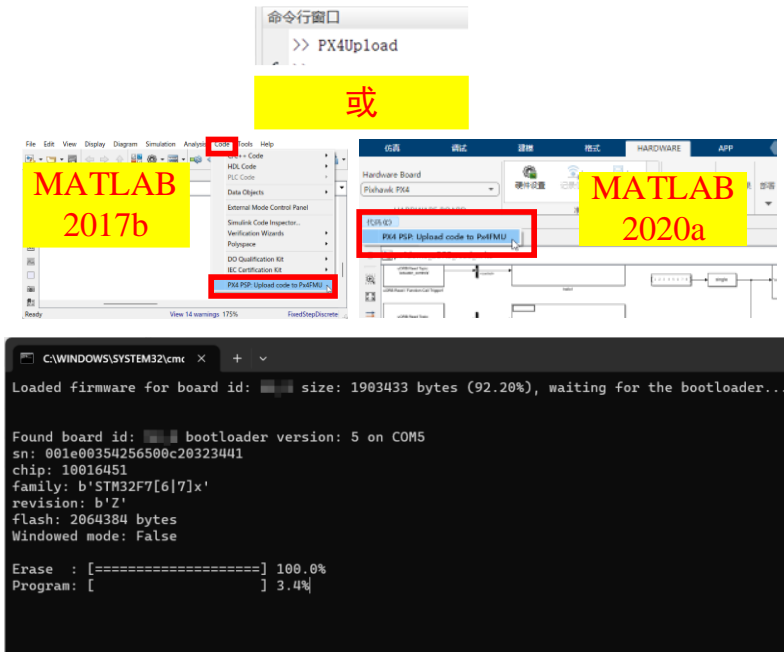
打开 [HITL&FLY\ESO\\_HITL\\_FLY.slx](#) 文件，在 Simulink 中，点击编译命令。



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。

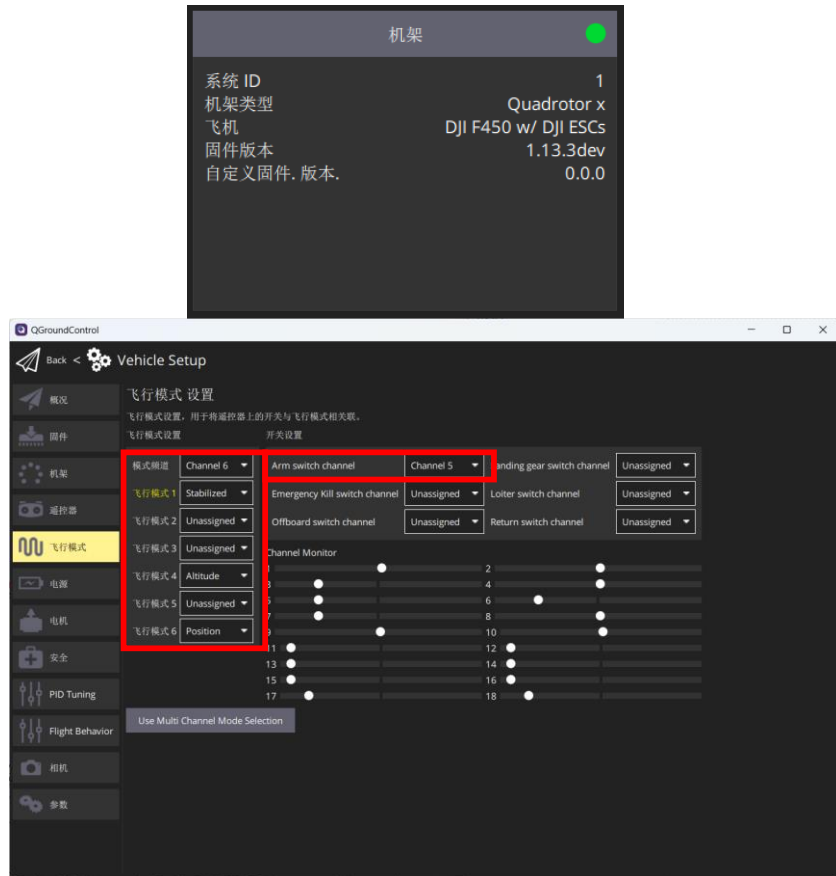


用 USB 数据线链接飞控(或飞机)与电脑。在 MATLAB 命令行窗口输入：PX4Upload 并运行，弹出 CMD 对话框，显示正在上传固件至飞机中，等待上传成功。



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。





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。具体设置请见本平台的[遥控器配置手册](#)。

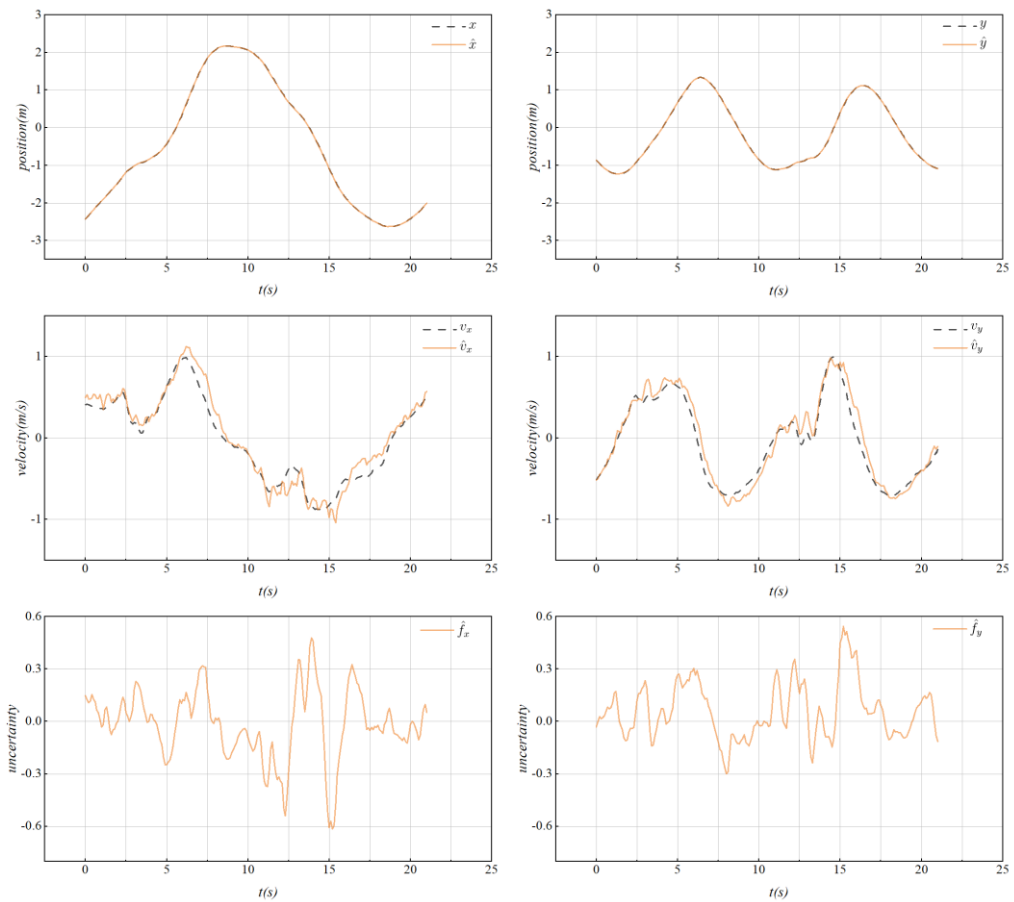


为确保安全，可在飞机上系上安全绳，并将安全绳的另一端固定在重物上。飞行时人在安全半径以外，在定点模式下，让油门在中位附近，即可实现定点。

打开 QGC，对无人机的各项参数进行初始化设置，如果在室内请打开 ekf 滤波器的其他传感器模式。将无人机放在试验场地，注意远离人群，保证安全。

拨动 CH5，解锁无人机，先用遥控器控制无人机以自稳模式或定高/定点模式起飞，然后拨动 CH7，切换至强化学习控制模式，查查看控制结果，发现扩张状态观测器可以较好

的观测无人机 x 方向和 y 方向上的位置和速度。部分控制结果如下图所示：



## 6. 参考资料

[1] 无。

## 7. 常见问题

Q1: 自定义的 uORB 消息在导出的 .ulg 日志文件中无法生成日志数据

A1: 打开 “\*:\PX4PSP\Firmware\src\modules\logger\logged\_topics.c” 文件。将自定义的 uORB 消息添加到该文件中，即在第一个函数中添加代码 “add\_topic("rfly\_test");” 如下：

```
C++ logged_topics.cpp 9+, M ●
src > modules > logger > C++ logged_topics.cpp > add_default_topics()
37 #include <parameters/parameters.h>
38 #include <px4_platform_common/log.h>
39 #include <px4_platform_common/px4_config.h>
40 #include <uORB/topics/uORBTopics.hpp>
41
42 #include <string.h>
43
44 using namespace px4::logger;
45
46 void LoggedTopics::add_default_topics()
47 {
48     add_topic("action_request");
49     add_topic("rfly_test");
50     add_topic("actuator_armed");
51     add_topic("actuator_controls_0", 50);
52     add_topic("actuator_controls_1", 100);
53     add_topic("actuator_controls_2", 100);
54     add_topic("actuator_controls_3", 100);
```

将该文件保存后，双击打开"\*\桌面\RflyTools\Win10WSL.lnk"的 WSL 子系统，进行编译固件编译完成后，重复 Step3 烧录飞控当中，即可在 QGroundControl 导出 .ulg 文件，处理后可得到自定义消息发布的数据。