



---

# 多旋翼飞行器设计与控制 实践

## 失效保护逻辑设计实验

自动化科学与电气工程学院  
北京航空航天大学



# 大纲

---

1. 实验原理
2. 基础实验
3. 分析实验
4. 设计实验
5. 小结



# 实验原理

---

在多旋翼起飞前、飞行中都存在影响多旋翼安全的问题，这些安全问题与多旋翼决策存在着紧密的联系。在这里，主要考虑以下三个方面：

1. 通信故障(Communication Breakdown)
2. 传感器失效(Sensor Failure)
3. 动力系统异常(Power System Anomaly)





# 实验原理

---

## □ 安全问题

### (1) 通信故障

**通信故障**主要是指遥控器 (Remote Controller, RC) 与多旋翼之间、地面站 (Ground Control Station, GCS) 与多旋翼之间无法正常通信。该类故障主要分为三个方面：

#### 1) 遥控器未校准

遥控器未校准是指多旋翼在第一次飞行之前，飞控手未对遥控器进行校准，即没有让多旋翼“**知道**”遥控器每个摇杆和开关的具体作用。若未对遥控器进行校准，则在多旋翼飞行过程中其飞行控制系统无法识别用户的指令，甚至可能识别错误的指令导致飞行事故。





# 实验原理

---

## □ 安全问题

### (1) 通信故障

#### 2) 遥控器失联

遥控器失联是指在多旋翼起飞前或飞行过程中，遥控器和机载信号接收器无法正常通信。若使用遥控器控制多旋翼飞行，遥控器失联将导致多旋翼不受控制，产生安全问题。

#### 3) 地面站失联

地面站失联是指在多旋翼起飞前或飞行过程中，地面站与多旋翼无法正常通信。若多旋翼需要完成设定任务，则地面站失联将导致多旋翼无法获取任务点，导致任务无法完成。



# 实验原理

---

## □ 安全问题

### (2) 传感器失效

传感器失效主要是指多旋翼上机载传感器测量不准确，无法正常工作，等等。这里介绍四种传感器失效。

- 1) 气压计(Barometer)失效
- 2) 电子罗盘(Compass)失效
- 3) 全球定位系统(GPS)失效
- 4) 惯导系统(Inertial Navigation System)失效



# 实验原理

---

## □ 安全问题

### (3) 动力系统异常

动力系统异常主要指**电池**失效，以及**电调**、**电机**和**桨**的硬件故障导致飞行控制系统执行环节的失效。

- 1) 电池失效。通常是由于电池电量不足或者电池寿命减少导致的多旋翼供电不足。
- 2) 电调故障。主要表现在两个方面：①电调无法正确识别自驾仪给出的PWM指令；②电调无法给电机提供正确的输出电压。
- 3) 电机故障。主要表现在一定的输入电压下，输出不正确的转速。
- 4) 螺旋桨失效。桨的故障主要表现在桨叶松动，桨叶磨损和折断等。



# 实验原理

---

## □ 任务决策设计过程

### (1) 定义飞行器的飞行模式

#### 1) 手动飞行模式

手动飞行模式可以让飞控手手动控制多旋翼。

#### 2) 返航模式

这里将返航模式定义为正常返航模式，此时是飞控手通过遥控器手动给出返航指令。在该模式下，多旋翼会从当前位置**返回到飞机起飞位置**，并且在此处悬停。

#### 3) 自动着陆模式

在该模式下，多旋翼通过调整油门指令，在原水平位置实现自动着陆。



北航可靠飞行控制研究组

BUAA Reliable Flight Control Group





# 实验原理

## □ 任务决策设计过程

### (2) 事件定义

多旋翼事件定义是多旋翼状态和飞行模态的切换的根据，是多旋翼决策的重要基础。在这里，我们主要定义两种事件：**人工输入事件**和**飞行器自触发事件**。

**1) 人工输入事件**主要是指飞控手利用遥控器或者地面站发出的指令，并基于此改变多旋翼状态和飞行模态的输入事件。

**a.解锁和锁定动作**定义为MIE1。MIE1=0 时为定锁，MIE=1 时为解锁。**b.人工操作指令**定义为 MIE2。MIE2=1 时切换到手动飞行模式，MIE2=2 时切换到返航模式，MIE2=3 时切换到自动着陆模式。该指令通过遥控器上（如图）的通道 5（三段拨动开关）进行切换操作。

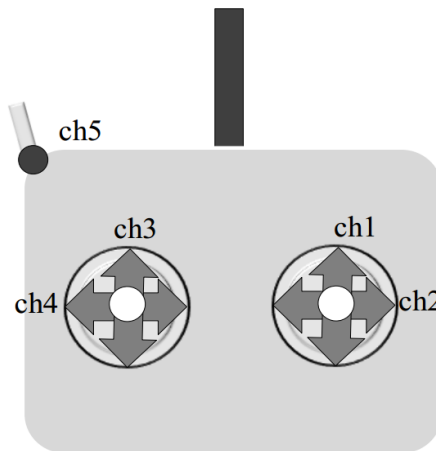


图. 遥控器示意图





# 实验原理

## □ 任务决策设计过程

### (2) 事件定义

多旋翼事件定义是多旋翼状态和飞行模态的切换的根据，是多旋翼决策的重要基础。在这里，我们主要定义两种事件：**人工输入事件**和**飞行器自身事件**。

**2) 飞行器自触发事件 (ATE)**：飞行器自触发事件与飞控手操作无关，主要取决于机上各部件的工作状态及多旋翼状态。

在基础实验中假定遥控器连接正常，即不把飞行器自触发事件作为判断条件，仅仅以人工输入事件作为条件进行考虑。

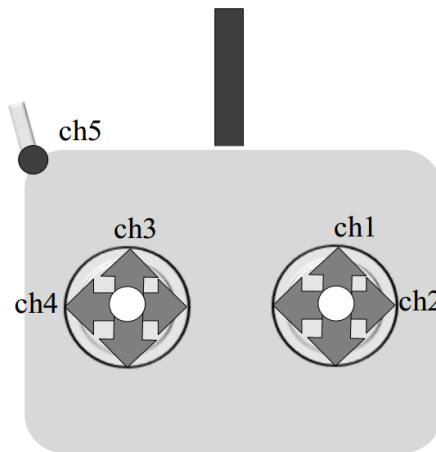


图. 遥控器示意图





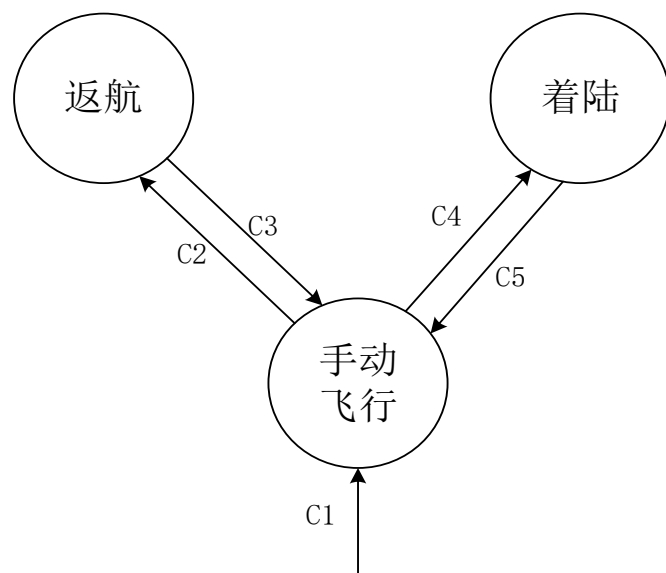
# 实验原理

## □ 任务决策设计过程

### (3) 状态定义

以手动飞行与返航、着陆之间转换

为例，状态转移如下图所示



图中  $C_i$  表示相应的转移条件：

$C1: MIE1=1;$

该条件描述多旋翼的成功解锁条件，多旋翼必须满足：飞控手发出解锁指令 ( $MIE1=1$ )

$C2: MIE2=2;$

该条件描述的是多旋翼从手动飞行模式到返航模式，多旋翼必须满足：飞控手利用遥控器手动将多旋翼切换到返航模式 ( $MIE2=2$ )

$C3、C5: MIE2=1;$

该条件描述的是多旋翼从返航到手动飞行和从着陆到手动飞行的条件，多旋翼必须满足：飞控手利用遥控器手动将多旋翼切换到手动飞行模式 ( $MIE2=1$ )

$C4: MIE2=3;$

该条件描述的是多旋翼从手动飞行到着陆的条件，多旋翼必须满足：飞控手利用遥控器手动将多旋翼切换到着陆模式 ( $MIE2=3$ ) .





# 实验原理

## □ 任务决策设计过程

### (4) 控制器指令

在模型根据输入事件不同进行判断后产生状态转换后，还需要根据不同的状态对控制器施加相应的指令以使多旋翼达到不同状态的要求：

- 1) **手动飞行状态**：飞控手手动控制多旋翼。
- 2) **返航状态**：返航状态要求多旋翼返回起飞点，这属于定点位置控制，故这时姿态控制器输入需要接入位置控制器输出接口。故这时的位置控制指令为  $p_{x_d} = 0, p_{y_d} = 0$ ，即给控制器加入返回原点的指令。**同时高度通道上，若当前高度高于设定的安全高度，则  $p_{z_d} = \hat{p}_z$ ，即控制多旋翼高度不变；若当前高度低于设定安全高度，则多旋翼先上升到设定高度再返回起飞原点。**
- 3) **着陆状态**：着陆状态要求多旋翼保持原来的水平位置，而调整油门指令实现降落。此时姿态控制器输入需要接入位置控制器输出接口，同时，给位置控制器的控制指令为水平期望速度  $v_{x_d} = 0, v_{y_d} = 0$ ，使多旋翼维持原来的水平位置不变；高度通道上  $p_{z_d} = 0$ ，使多旋翼实现降落。





# 实验原理

---

## □ 实现软硬件

### (1) Stateflow

状态机可以在Stateflow 工具箱内实现，Stateflow 是 MathWorks 开发的利用状态机和流程图用于对响应系统进行建模的工具。它是一种图形化的设计工具，能够实现有限状态机，实现各个状态间的转换。它和 Simulink 同时使用时可以接收 Simulink 的输入，同时给 Simulink 输出，实现和 Simulink 中模型的联系。将设计好的状态机在Stateflow 中实现。



# 实验原理

## □ 实现软硬件

### (2) 遥控器

模式切换中的手动飞行模式需要遥控器来发送控制指令，故添加指令时模拟遥控器实际发送的信号值，如右图所示，遥控器的端口中，通道 1 控制遥控器的前后运动，通道 2 控制其左右运动，通道 3 控制油门，通道 4 控制偏航角，通道 5 发送对模态的控制指令，通道 5 是三段拨动开关，刚好可以满足三种模式。

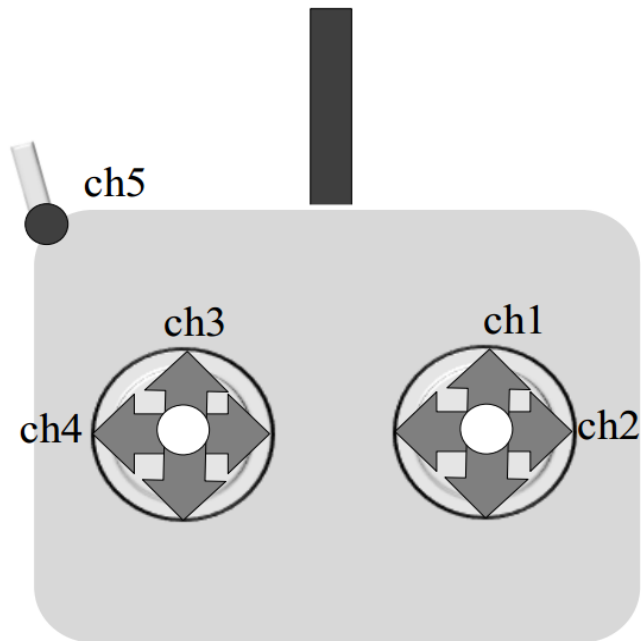


图. 遥控器示意图

在实际操作中，这五个通道的值都在 1000-2000 范围内，这里为了更加接近真实情况，仿真时也输入 1000-2000 的值，同时这些信号在进入控制器前要对其进行归一化处理，使它们变成 0-1 之间的值。对于通道 5，将其设置为它的值在 1000-1400 时，模态切换中的人工输入事件 MIE2 为 1，对应为手动飞行模式；值在 1400-1600 时，模态切换中的人工输入事件 MIE2 为 2，对应返航模式；值在 1600-2000 时，MIE2 为 3，对应着陆模式。





# 基础实验

---

## □ 实验目标

### ■ 已知

- (1) 硬件：Pixhawk自驾仪系统，多旋翼硬件系统；
- (2) 软件：MATLAB 2017b或以上的版本，基于Simulink的控制器设计与仿真平台，硬件在环仿真平台，实验指导包“e8.1”（下载地址：<https://rflysim.com/course>）。

### ■ 目标

- (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) 完成硬件在环仿真。





# 基础实验

## □ 实验步骤

### (1) 步骤一：软件在环仿真

#### 1) 参数初始化

打开文件“e8\e8.1\Init\_control.m”进行参数初始化，“e8\_1\_Sim.slx”将会自动打开如图所示。

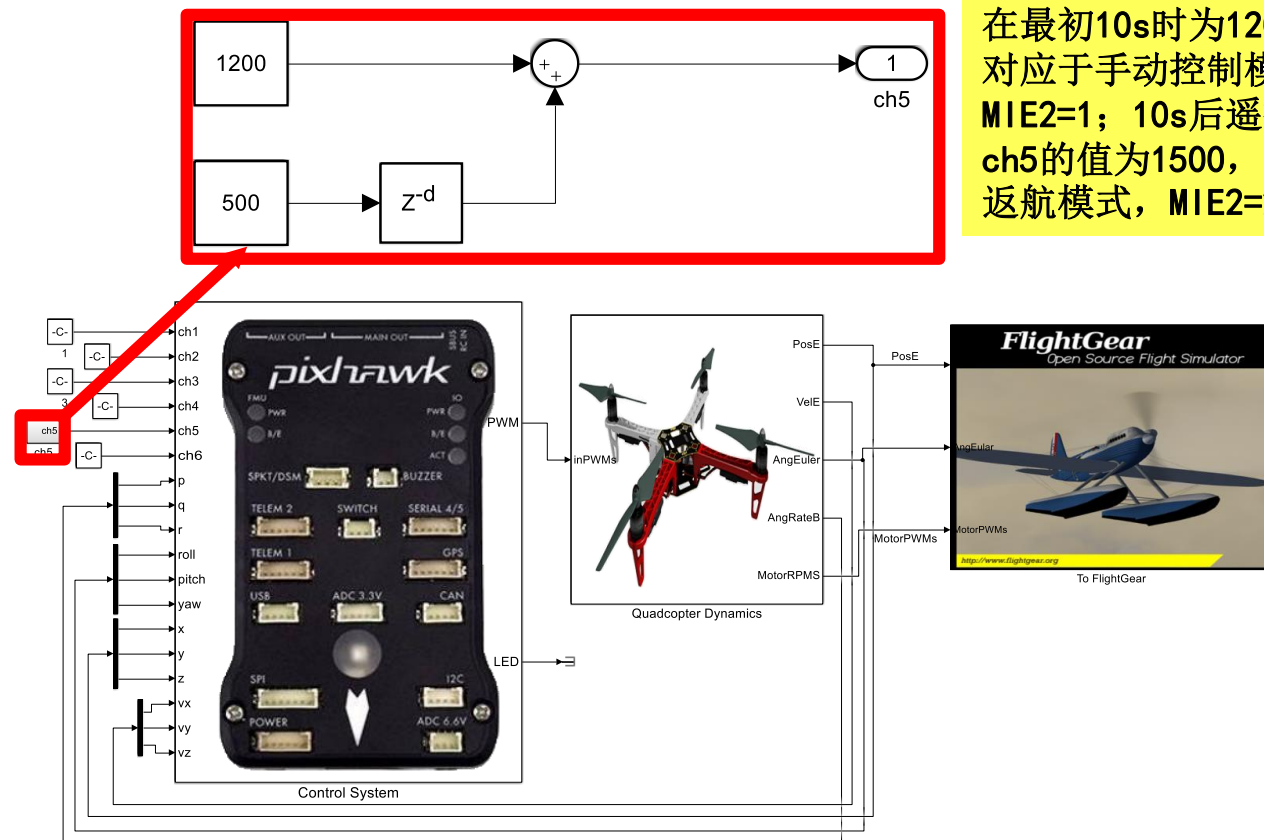


图. 返航、着陆模型





# 基础实验

## □ 实验步骤

### 2) 返航仿真

仿真结果如右图所示。可以看到0~10s时在手动控制模式下，多旋翼在给定指令下在空间中自由飞行。10s后进入返航模式，高度经过一小段时间调整后保持不变，水平位置逐渐趋于0，最后稳定在0位置，完成返航。

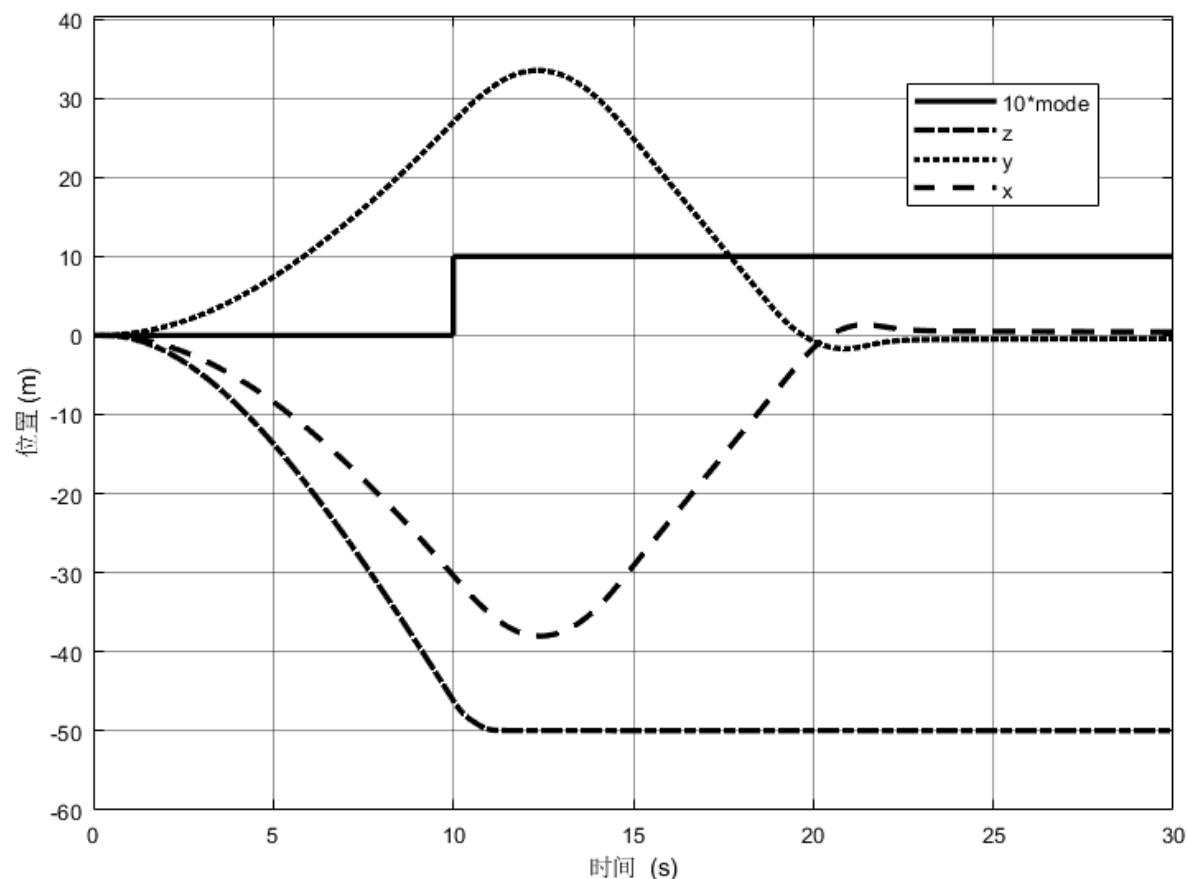


图. 手动模式下返航





# 基础实验

## □ 实验步骤

### 3) 着陆仿真

这里，“ch5”值的修改如下：前10s输入1200，对应人工飞行模式，MIE2=1;10s后输入1700，对应自动着陆模式，MIE3=3。仿真结果如右图所示。在0~10s内，四旋翼在给定的指令下以手动飞行方式起飞爬升。10秒后，条件C4满足，四旋翼进入着陆模式，水平位置保持不变，高度逐渐下降到0。

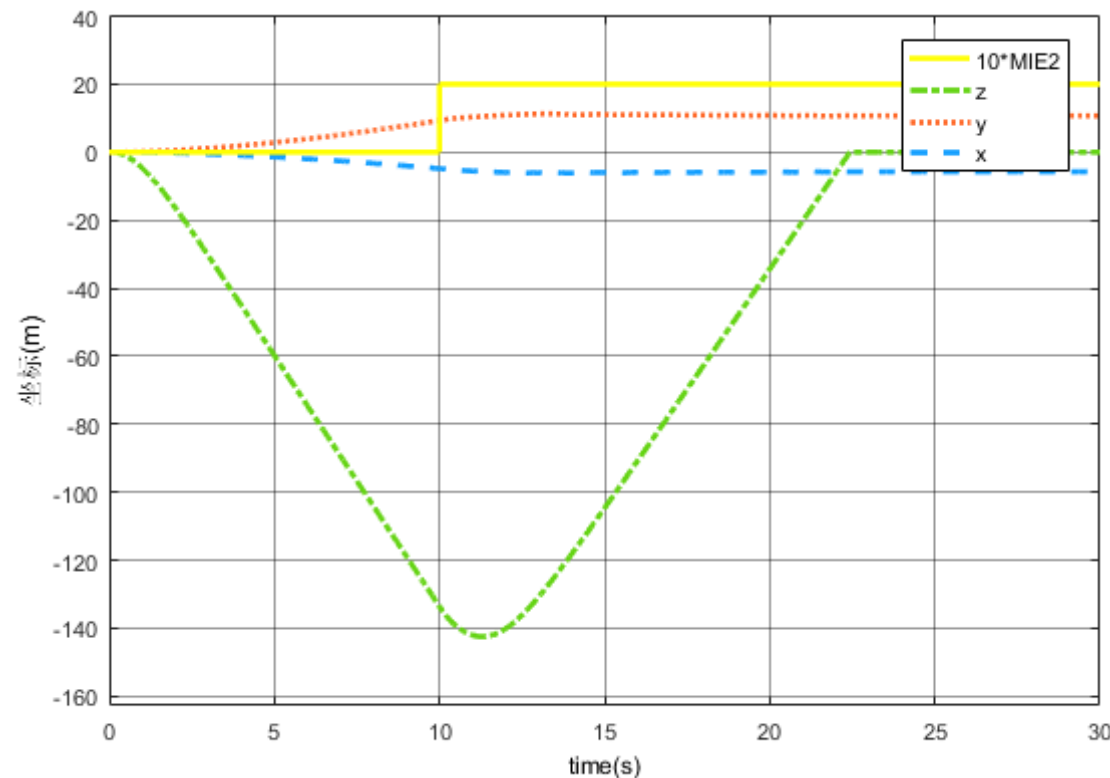


图. 手动模式下返航



# 基础实验

## □ 实验步骤

### (2) 步骤二：硬件在环仿真

#### 1) 参数初始化

运行“e8\e8.1\HIL\Init\_control.m”初始化模型参数，“e8\_1\_HIL.slx”将会自动打开，如右图。注意这里的“Control System”模块与软件在环仿真相同。

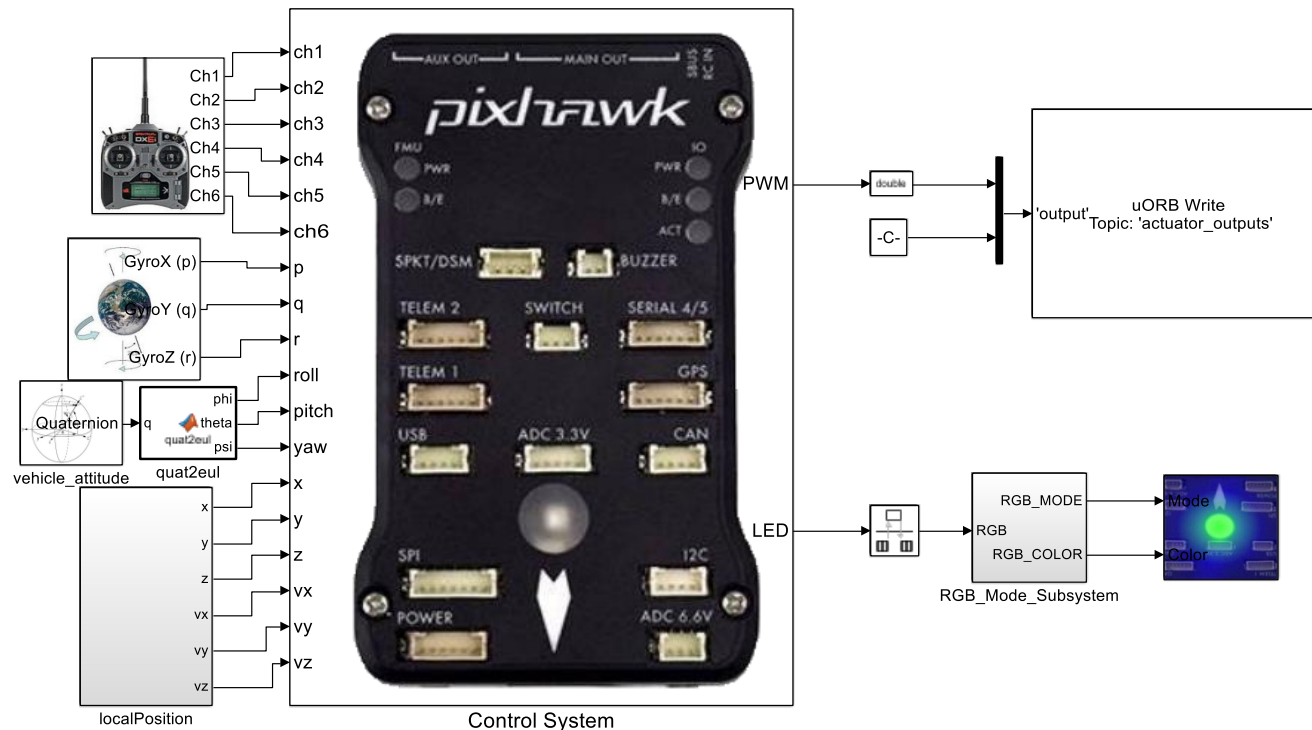


图. e8\_1\_HIL.slx截图



# 基础实验

## □ 实验步骤

### 2) 硬件连接

需要注意的是，在进行硬件在环仿真时**机架类型**应该是“**HIL Quadcopter X**”。

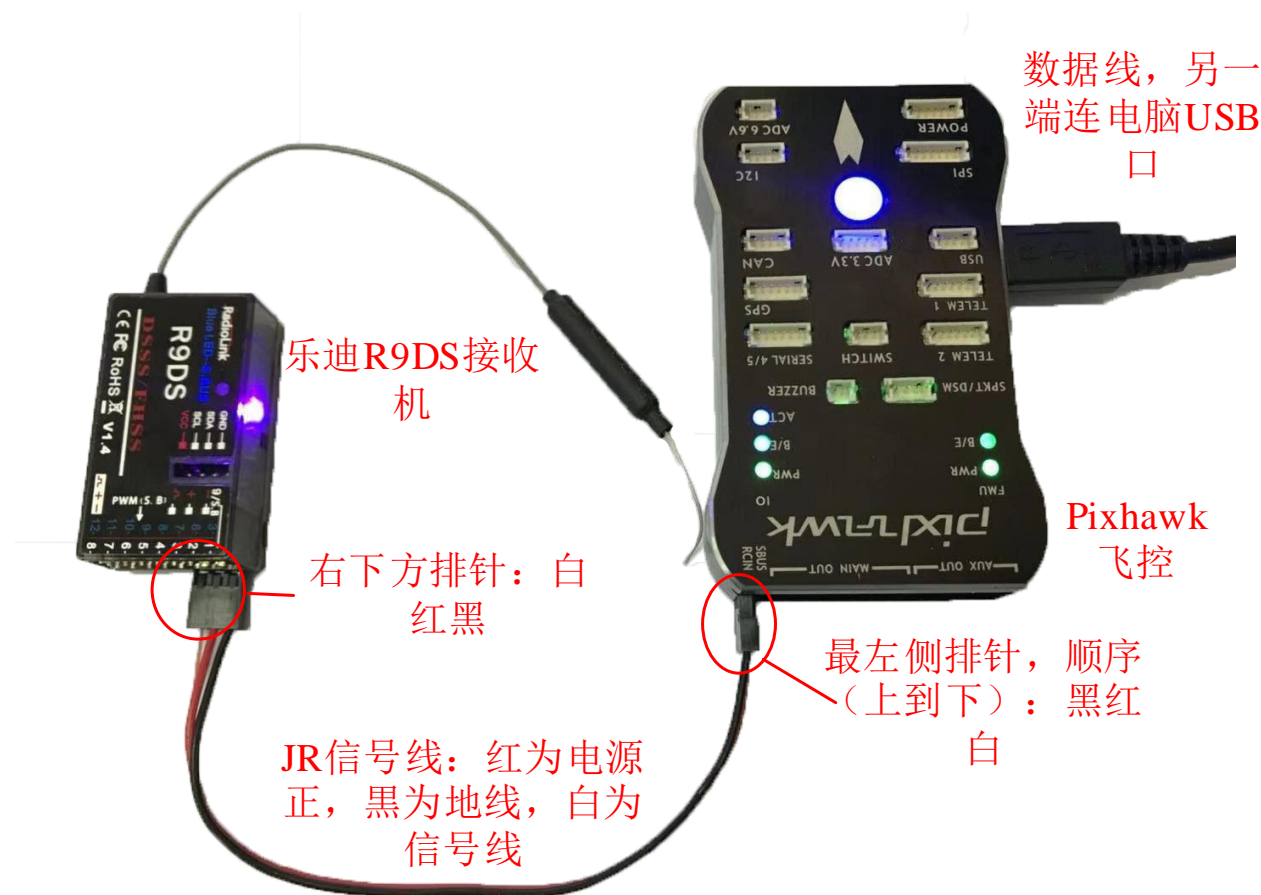


图. 硬件系统连接



# 基础实验

## 实验步骤

### 3) 代码编译及下载

将硬件在环仿真模型编译并下载文件到Pixhawk自动驾驶仪中。这样就可以在Pixhawk自动驾驶仪中运行我们自己设计的失效保护控制程序。

The image shows the process of compiling and downloading code from a Simulink model to a Pixhawk autopilot. The top window is the Simulink interface for 'E1\_rgbled\_system'. The 'Code' menu is open, and 'PX4 PSP: Upload code to Px4FMU' is selected. A red arrow points from the 'Code' menu to the terminal window below. The terminal window shows the output of the compilation and download process. The output includes the following text:

```
### Successfully generated all binary outputs.  
Loaded firmware for 9.0, size: 875004 bytes, waiting for the bootloader...  
if the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.  
PX4_SIMULINK = y  
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 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8f  
5c 5e07decc 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e  
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4  
idtype: =00  
vid: 000026ac  
pid: 00000010  
coa: ZuIH//9zzBXIrZQM28Wf0dbCDgb5U9Pv8wcwGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2m17xjQIMhvmjt01bJHjuPXF4H3syca+W1RYo  
VEOu7vCHto1z1qLhApe8BnKXIm7GDuWdwwAbbGiZzdf9XHKQd1CSqZi4=  
sn: 0038001f3432470d31323533  
Erase : [=====] 100.0%  
Program: [=====] 100.0%  
Verify : [=====] 100.0%  
Rebooting.  
H:
```

Red boxes highlight the 'PX4 SIMULINK = y' line, the 'if the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.' warning, and the progress bars for Erase, Program, and Verify. A red arrow points from the 'PX4 PSP: Upload code to Px4FMU' menu item to the terminal window. The text '点击下载' (Click to download) is written below the menu, and '下载完成' (Download complete) is written below the terminal output.

图. 代码编译及下载流程



# 基础实验

## 实验步骤

### 4) 模型仿真器软件配置。

双击 CopterSim 桌面快捷方式即可打开 CopterSim。读者可以选择不同的动力系统模型，步骤如下：单击“模型参数”自定义参数，然后单击“存储并使用参数”存储并使用参数。软件会自动匹配串口号，单击“开始仿真”按钮就可以进入硬件在环仿真模式。此时可以看到如图所示的界面左下角收到自仪返回的相关消息。

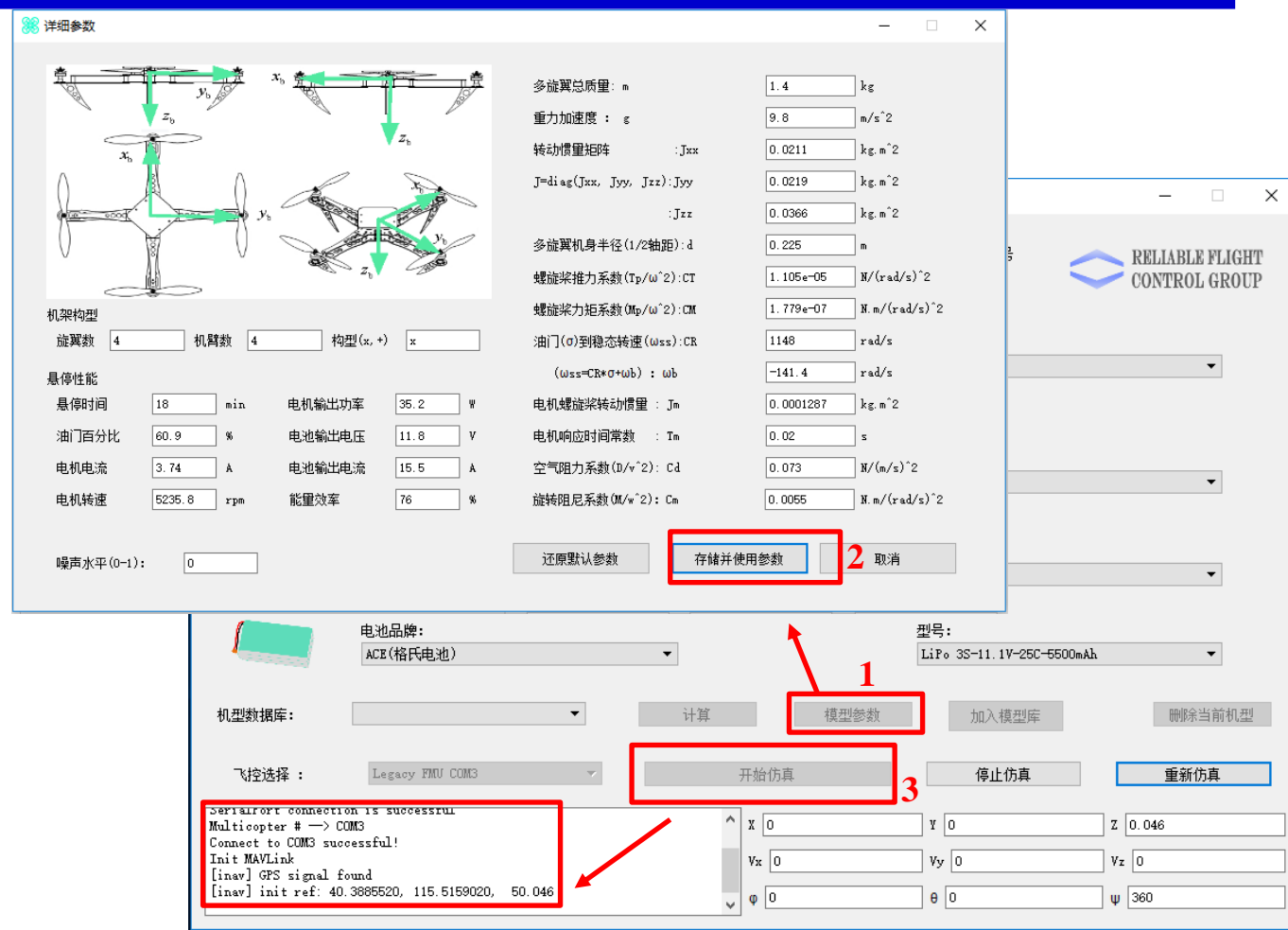


图. 模型仿真器软件配置





# 基础实验

## 实验步骤

### 5) 3DDisplay配置

双击3DDisplay桌面快捷方式打开

3DDisplay。

### 6) 仿真效果

解锁多旋翼，并将四旋翼飞行到一定高度。将CH5拨到中间位置实现返航。如果再往上拨四旋翼仍处于返航状态，这是因为状态转移条件中，没有返航与着陆的转换。随后拨动CH5使四旋翼重新进入手动控制模式，快速拨动CH5到离使用者最远的档位，四旋翼进入着陆模式。



图. 3D Display



# 分析实验

---

## □ 实验目标

### ■ 已知

- (1) 硬件：Pixhawk自驾仪系统，多旋翼硬件系统；
- (2) 软件：MATLAB 2017b或以上的版本，基于Simulink的控制器设计与仿真平台，硬件在环仿真平台，实验指导包“e8.2”（下载地址：<https://rflysim.com/course>）。

### ■ 目标

- (1) 在基础实验的基础上，添加相应的状态转移，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。
- (2) 实现硬件在环仿真。





# 分析实验

## □ 实验设计

本实验在基础实验的基础上增加返航和着陆之间的切换，其对应的状态机如下图所示。

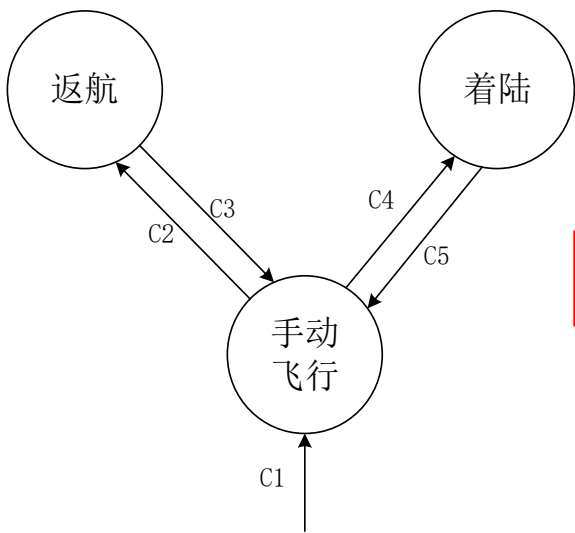


图. 基础实验状态机

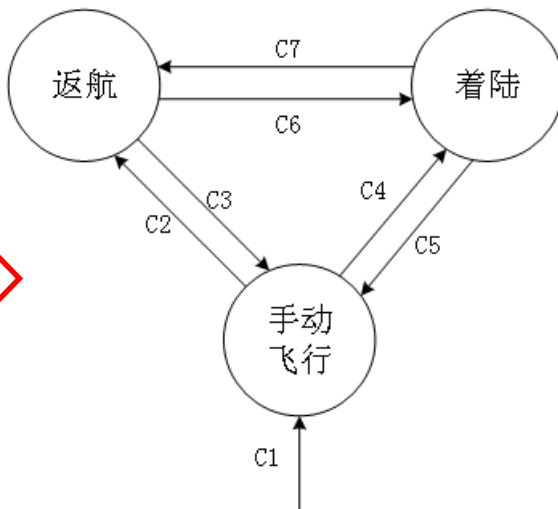


图. 案例分析实验状态机

新增的两个状态的转移条件为：

### ■ C6: MIE2=3

该条件描述的是多旋翼从返航模式到着陆模式，多旋翼必须满足：飞控手利用遥控器手动将多旋翼切换到着陆模式（MIE2=3）

### ■ C7: MIE2=2

该条件描述的是多旋翼从着陆模式到返航模式，多旋翼必须满足：飞控手利用遥控器手动将多旋翼切换到返航模式（MIE2=2）



# 分析实验

## 实验步骤

### (1) 步骤一：软件在环仿真

1) 复制基础实验中的模型，找到模型中的Control System模型（右上图）中名为mode switch的状态机（右下图）。

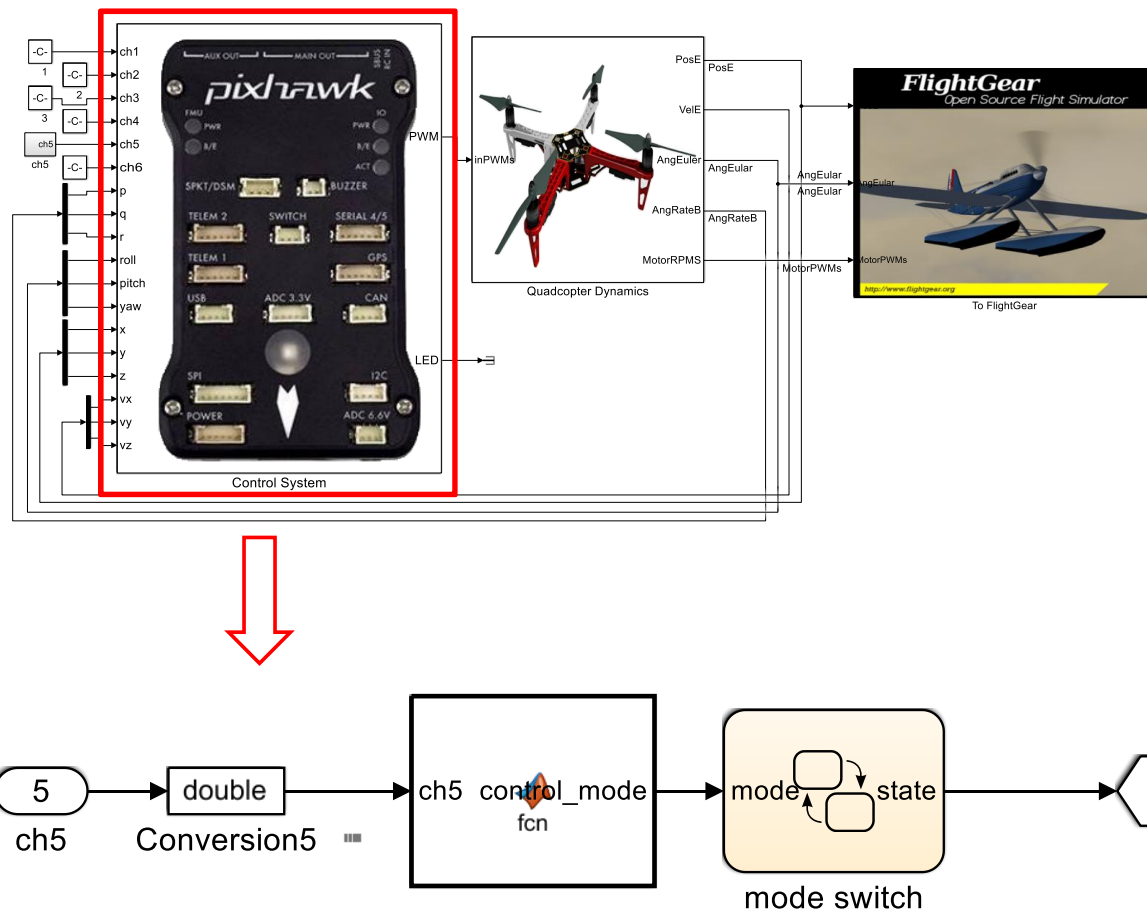


图. “Control system” 中的 “mode switch” 模块





# 分析实验

## □ 实验步骤

### 2) 修改状态机。

在状态机的转换条件中，是以事件为判断条件进行的，比如MIE2，这是输入的条件。在状态机中加入返航和着陆的转换条件，如右图。

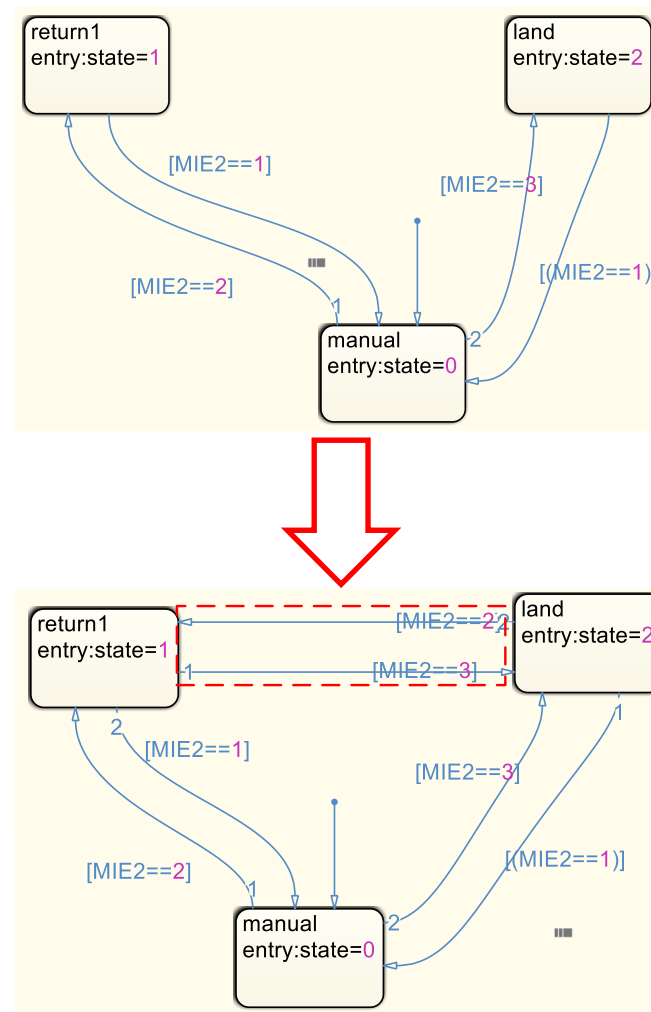


图. 修改状态机



# 分析实验

---

## □ 实验步骤

### 3) 修改模型输入。

将模型中“ch5”通道的输入改为可模拟遥控器三段输入的形式，最终实现在0~10s，输入为1200，即为手动模式，MIE2=1；10~30s，输入为1500，即为返航模式，MIE2=2；30~50s输入为1800，即为着陆模式，MIE2=3。整个过程模拟遥控器由手动切换到返航，再切换到着陆的过程。

### 4) 保存模型，并初始化参数。

将模型保存到文件“e8\e8.2\e8\_2\_sim.slx”。运行“e8\e8.2\Init\_control.m”进行参数初始化。





# 分析实验

## □ 实验步骤

### 5) 仿真效果

直接得到仿真结果如右图所示。可以看到0~10s多旋翼在空间中自由飞行，10~30s为返航状态，高度保持不变，水平位置回到(0, 0)。30s之后变成着陆模式，水平位置依旧保持(0, 0)，高度下降为0。

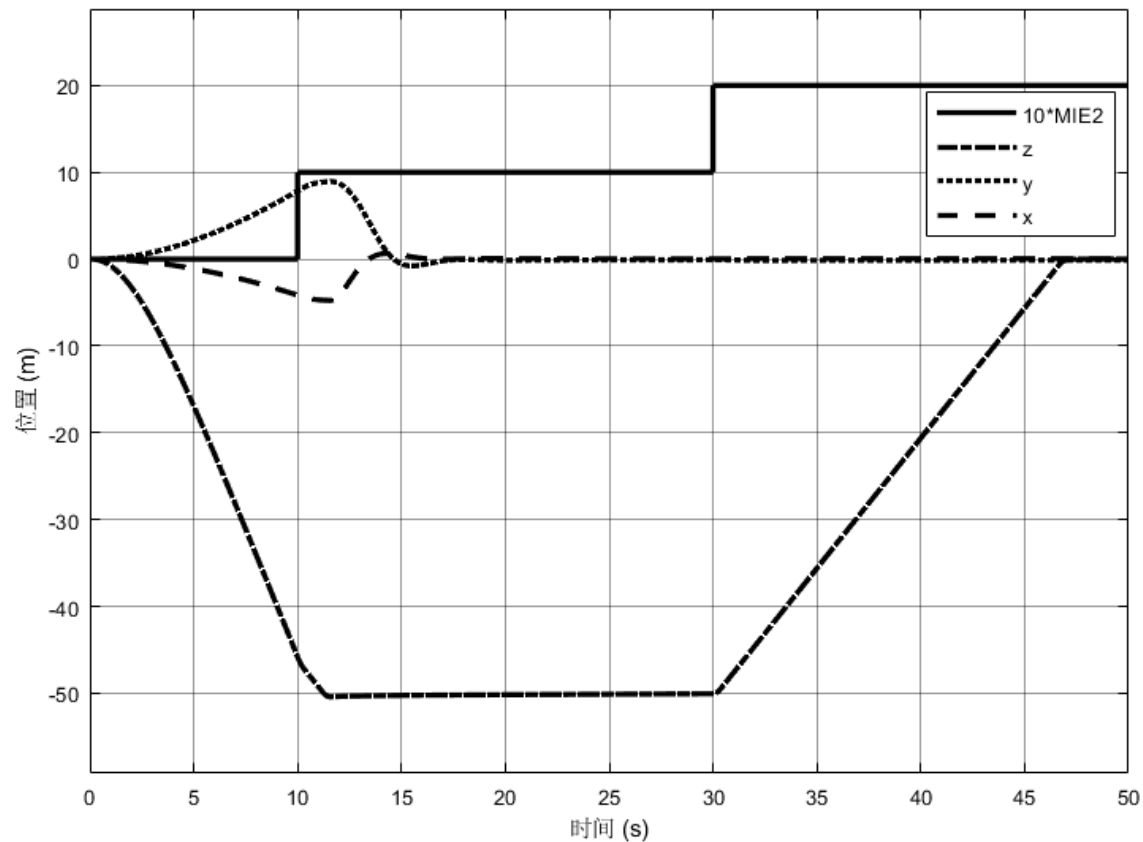


图. 返航与着陆切换仿真结果





# 分析实验

## □ 实验步骤

### (2) 步骤二：硬件在环仿真

#### 1) 参数初始化

打开“e8\e8.2\HIL\Init\_control.m”进行参数初始化，“e8\_2\_HIL.slx”模型会自动打开，如右图。值得注意的是，这里的“Control System”模块与软件在环仿真的相同。

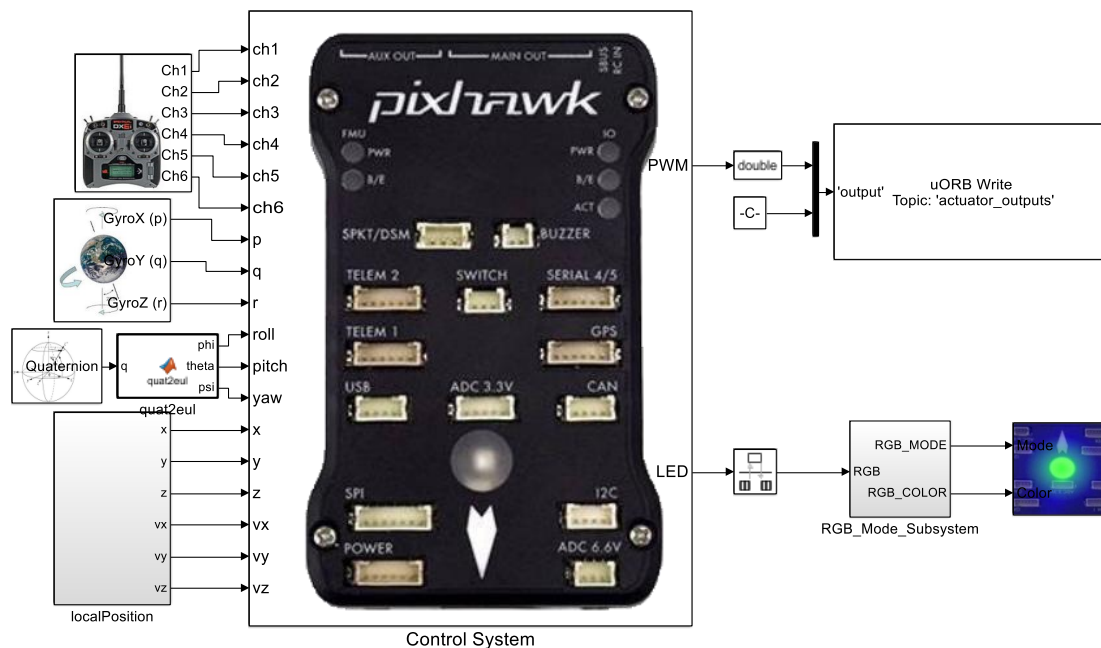


图. e8\_2\_HIL. slx截图





# 分析实验

## □ 实验步骤

### 2) 硬件连接

需要注意的是，在进行硬件在环仿真时**机架类型**应该是“**HIL Quadcopter X**”。

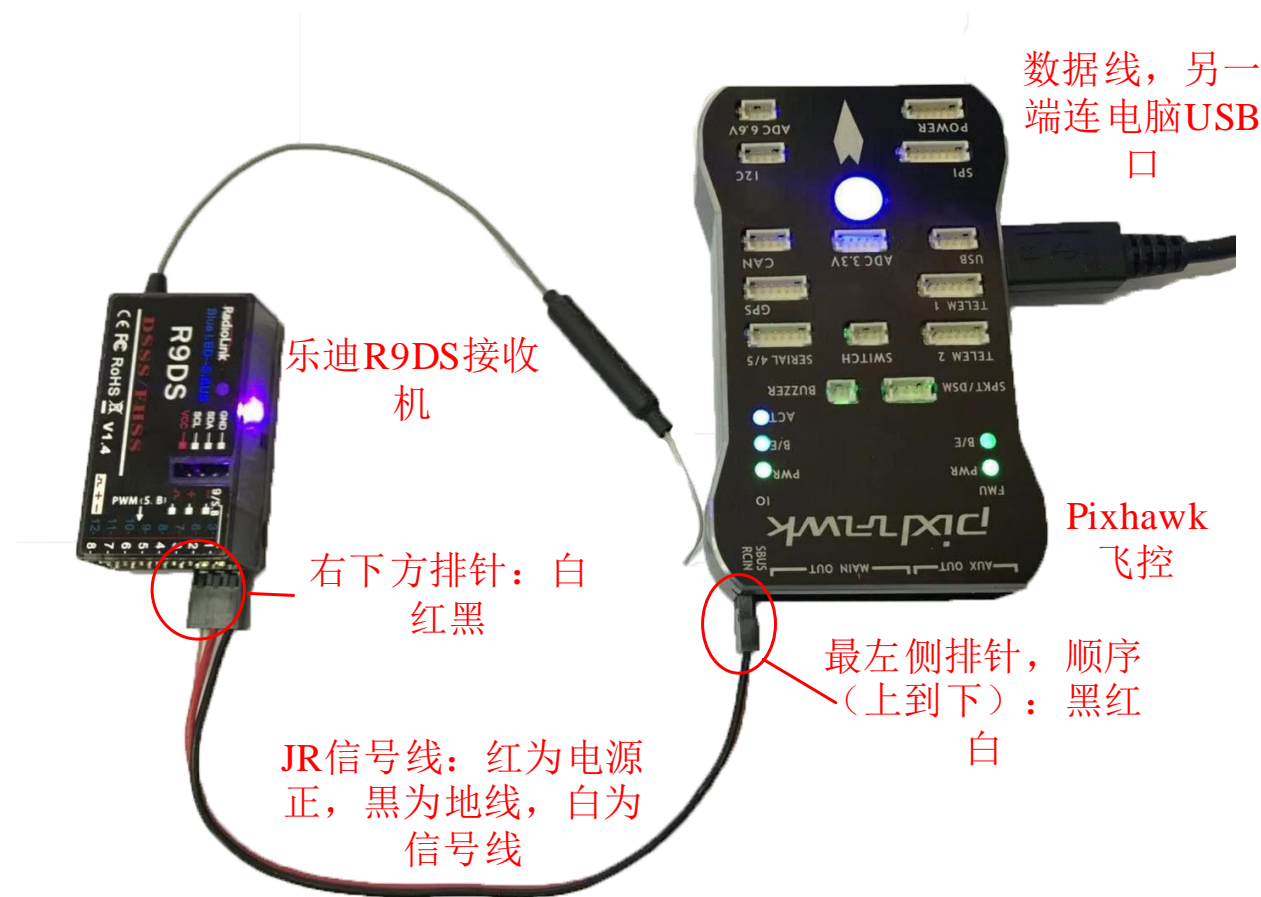


图. 硬件系统连接





# 分析实验

## 实验步骤

### 3) 代码编译及下载

将硬件在环仿真模型编译并下载文件到Pixhawk自动驾驶仪中。这样就可以在Pixhawk自动驾驶仪中运行我们自己设计的失效保护控制程序。

The image shows the process of compiling and downloading code to a Pixhawk. It includes a Simulink window titled 'E1\_rgbled\_system' with a red box around the 'PX4' button in the toolbar, labeled '点击编译' (Click Compile). Below it, a dropdown menu is shown with 'PX4 PSP: Upload code to Px4FMU' selected, labeled '点击下载' (Click Download). To the right, a terminal window shows the output of the compilation and download process, with a red box around the 'Download complete' message, labeled '下载完成' (Download Complete).

```
### Successfully generated all binary outputs.  
Loaded firmware for 9.0, size: 875004 bytes, waiting for the bootloader...  
if the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.  
PX4_SIMULINK = y  
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 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8f  
5c 5e07decc 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e  
ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4  
idtype: =00  
vid: 000026ac  
pid: 00000010  
coa: ZuIH//9zzBXIrZQM28Wf0dbCDgb5U9Pv8wcwGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2mL7xjQIMhvmjt01bJHjuPXF4H3syca+W1RYo  
VEOu7vCHto1z1qLhApe8BnKXIm7GDuWdwwAbbG1Zzdf9XHKQd1CSqZi4=  
sn: 0038001f3432470d31323533  
Erase : [=====] 100.0%  
Program: [=====] 100.0%  
Verify : [=====] 100.0%  
Rebooting.  
H:
```

图. 代码编译及下载流程





# 基础实验

## 实验步骤

### 4) 模型仿真器软件配置。

双击 CopterSim 桌面快捷方式即可打开 CopterSim。读者可以选择不同的动力系统模型，步骤如下：单击“模型参数”自定义参数，然后单击“存储并使用参数”存储并使用参数。软件会自动匹配串口号，单击“开始仿真”按钮就可以进入硬件在环仿真模式。此时可以看到如图所示的界面左下角收到自驾仪返回的相关消息。

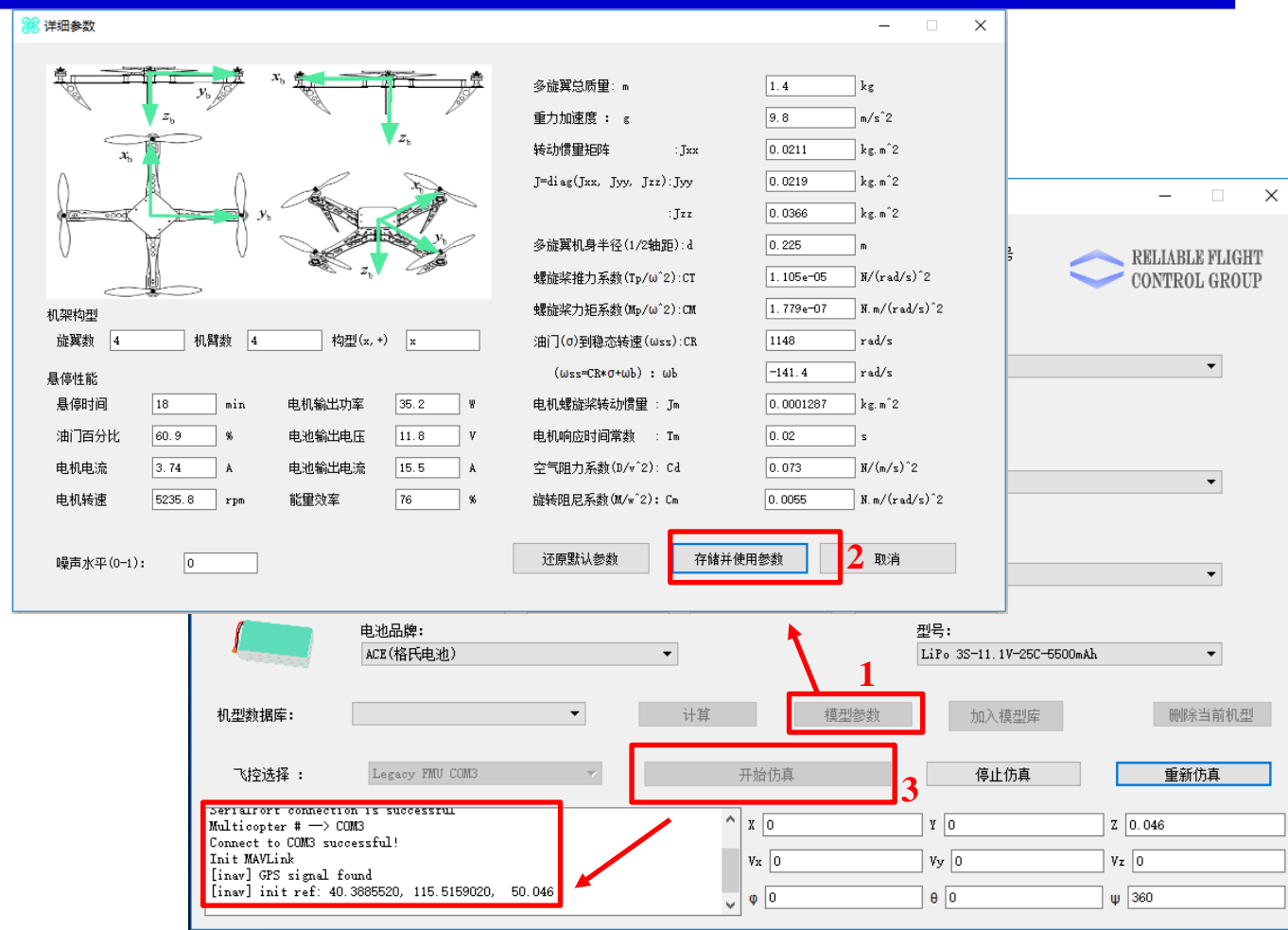


图. 模型仿真器软件配置



# 分析实验

## □ 实验步骤

### 5) 3DDisplay配置

双击3DDisplay桌面快捷方式打开3DDisplay。

### 6) 仿真效果。

解锁多旋翼，并操控多旋翼飞行到一定高度，将CH5向后拨一段实现返航。返航完成后，再将CH5向后拨一段实现着陆。这里与基础实验不同，基础实验中因为没有返航和着陆之间的切换，返航后直接切到着陆是没有响应的。



图. 3D Display



# 分析实验

---

## □ 注意事项

- (1) 多旋翼在返航和着陆的过程中需要一点时间，在仿真过程中应该空出足够的时间让多旋翼保持稳定，以保证出现和实际情况相符的仿真结果，最终达到良好的仿真效果。
- (2) 在更改状态机时，要特别注意状态转移之间的关系和转移实现的条件，按照已有的状态机的设计例程，完成新的转移条件的添加，并且进行相应的仿真。



# 设计实验

---

## □ 实验目标

### ■ 已知

- (1) 硬件：Pixhawk自驾仪系统，多旋翼硬件系统；
- (2) 软件： MATLAB 2017b或以上的版本，基于Simulink的控制器设计与仿真平台，硬件在环仿真平台，实验指导包“e8.3”和“e8.4”（下载地址：<https://rflysim.com/course>）。

### ■ 目标

- (1) 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。此外还需加入如下功能：如果飞行器离基地很近，直接着陆；而如果飞行器离基地有一定距离，先进行返航，再进行着陆；
- (2) 完成硬件在环仿真实验和实飞实验。



# 设计实验

---

## □ 实验设计

### (1) 步骤一：四旋翼的飞行模式

为了方便逻辑设计，四旋翼从飞行到着陆分为五种飞行模式，添加了两  
种失效的模式：（定义同基本原理）

1) 手动飞行模式

2) 返航模式

3) 着陆模式

4) 失效着陆模式：遥控器断电时自动着陆

5) 失效返航模式：遥控器断电时自动返航



# 设计实验

---

## □ 实验设计

### (2) 步骤二：定义事件

1) 人工输入事件 (MIE)：该事件主要是指由飞控手利用遥控器发出指令。

① MIE1：解锁和锁定指令； ② MIE2：人工操作指令。

2) 飞行器自触发事件 (ATE)：飞行器自触发事件与飞控手操作无关，主要取决于机上各部件的工作状态及多旋翼状态。（增添ATE3的判断条件）

① ATE1：遥控器连接状态是否正常。

ATE1=1:正常, ATE1=0: 异常。

② ATE3:旋翼到原点位置的距离与设定阈值的比较。

ATE3=1: 距离低于设定阈值; ATE3=0: 距离高于设定阈值。





# 设计实验

## □ 实验设计

### (3) 步骤三：设计状态机

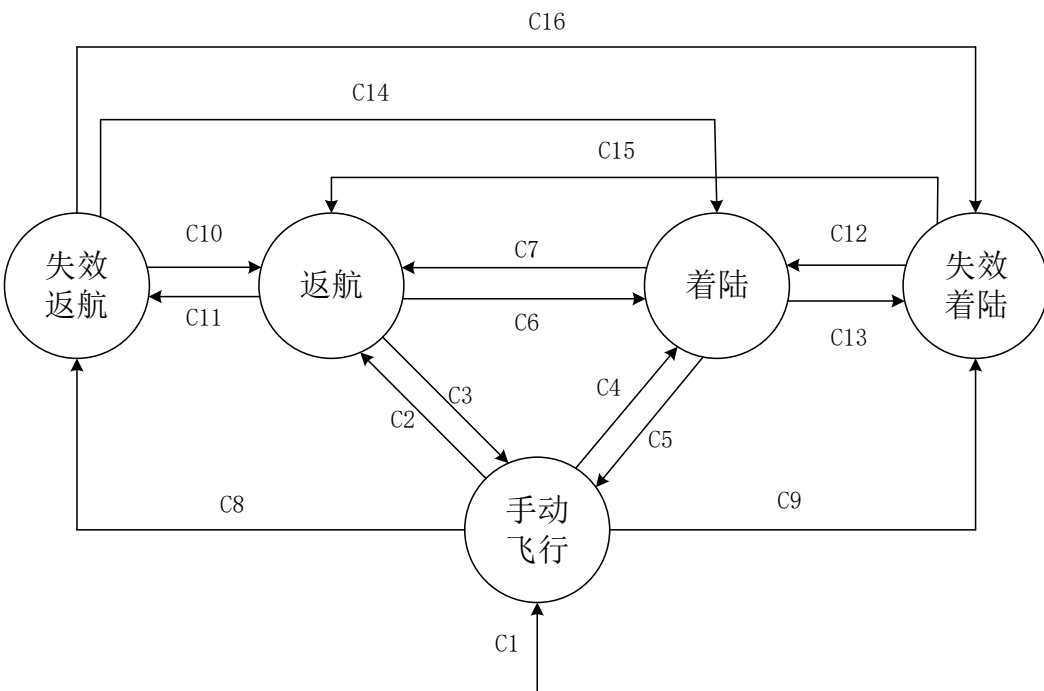


图. 状态机

图中  $C_i$  表示相应的转移条件：

#### ■ $C1: MIE1=1$

该条件描述多旋翼的成功解锁条件，多旋翼必须满足：飞控手发出解锁指令 ( $MIE1=1$ )；

#### ■ $C2、C7、C10、C15: ATE1=1 \& MIE2=2$

该条件描述的是多旋翼从手动飞行模式到返航模式 ( $C2$ )、着陆到返航 ( $C7$ )、失效着陆到返航 ( $C10$ ) 以及失效着陆到返航 ( $C15$ ) 的转移条件，多旋翼必须满足：遥控器连接正常 ( $ATE1=1$ ) 且飞控手利用遥控器手动将多旋翼切换到返航模式 ( $MIE2=2$ )；

#### ■ $C3、C5: ATE1=1 \& MIE2=1$

该条件描述的是多旋翼从返航到手动飞行 ( $C3$ ) 和从着陆到手动飞行 ( $C5$ ) 的条件，多旋翼必须满足：遥控器连接正常 ( $ATE1=1$ ) 且飞控手利用遥控器手动将多旋翼切换到手动飞行模式 ( $MIE2=1$ )；

#### ■ $C4、C6、C12、C14: ATE1=1 \& MIE2=3$

该条件描述的是多旋翼从手动飞行到着陆 ( $C4$ )、返航到着陆 ( $C6$ )、失效着陆到着陆 ( $C12$ ) 以及失效返航到着陆 ( $C14$ ) 的条件，多旋翼必须满足：遥控器连接正常 ( $ATE1=1$ ) 且飞控手利用遥控器手动将多旋翼切换到着陆模式 ( $MIE2=3$ )；





# 设计实验

## □ 实验设计

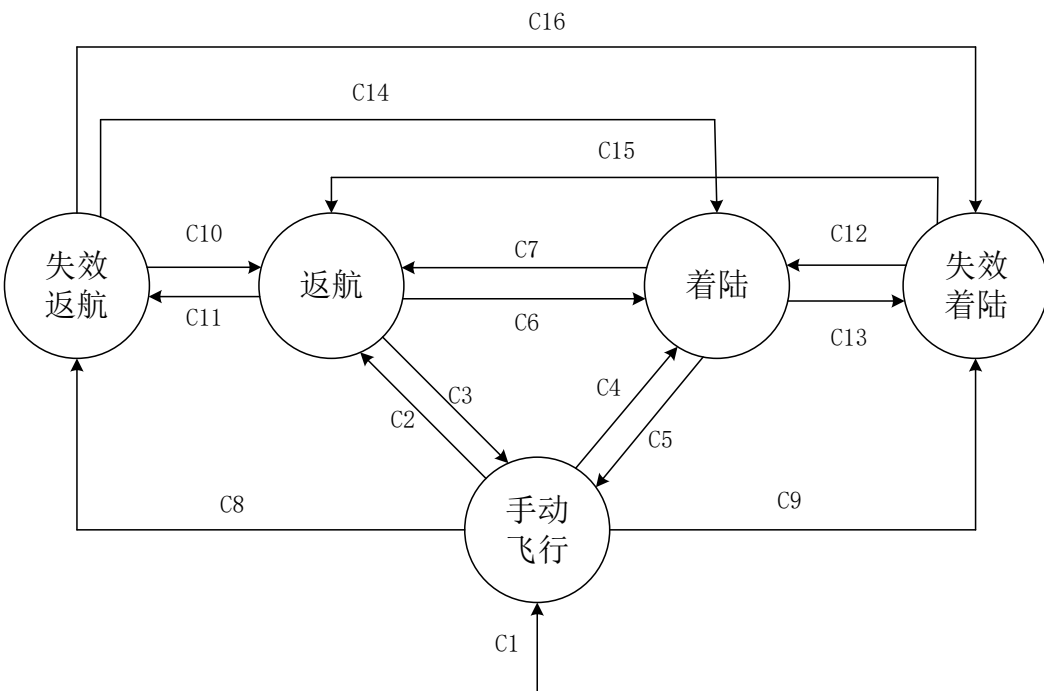


图. 状态机

图中  $C_i$  表示相应的转移条件:

### ■ C8: $ATE1=0 \& ATE3=0$

该条件描述的是多旋翼从手动飞行到失效返航的条件，多旋翼必须满足：遥控器连接异常 ( $ATE1=0$ ) 且旋翼到原点位置的距离高于设定阈值 ( $ATE3=0$ )

### ■ C9: $ATE1=0 \& ATE3=1$

该条件描述的是多旋翼从手动飞行到失效着陆的条件，多旋翼必须满足：遥控器连接异常 ( $ATE1=0$ ) 且旋翼到原点位置的距离低于设定阈值 ( $ATE3=1$ )

### ■ C11、C13: $ATE1=0$

该条件描述的是多旋翼从返航到失效返航(C11)以及着陆到失效着陆(C13)的条件，多旋翼必须满足：遥控器连接异常 ( $ATE1=0$ )

### ■ C16: $ATE1=0 \& ATE3=1$

该条件描述的是多旋翼从失效返航到失效着陆的条件，多旋翼必须满足：遥控器连接异常 ( $ATE1=0$ ) 且旋翼到原点位置的距离低于设定阈值 ( $ATE3=1$ )，此时多旋翼飞行器才可以实现着陆







# 设计实验

## □ 实验设计

### (4) 修改仿真模型

#### 1) 修改状态机模型

根据前三个步骤，在分析实验的基础上，修改了状态机模型，如右图所示。

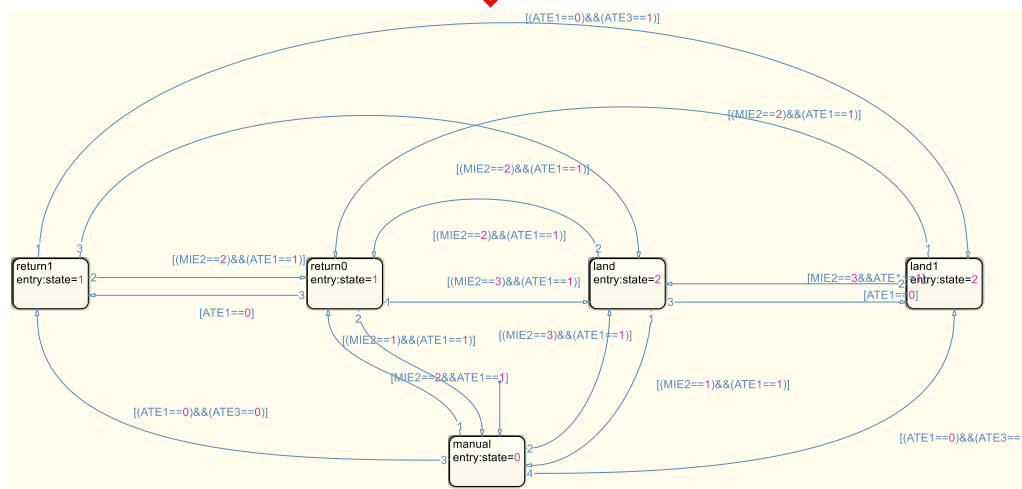
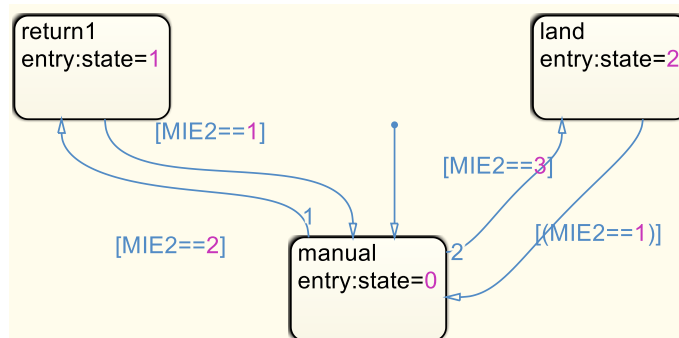


图. 带有失效保护的  
状态机



# 设计实验

## □ 实验设计

### 2) 修改遥控器通道输入

在原有模型上增加一个遥控器通道模拟遥控器失联事件，其输入值前10s为0，然后为1，以模拟遥控器正常连接10s后，连接失效事故。

### 3) 保存模型

将模型保存到“e8\8.3\sim\e8\_3.slx”。

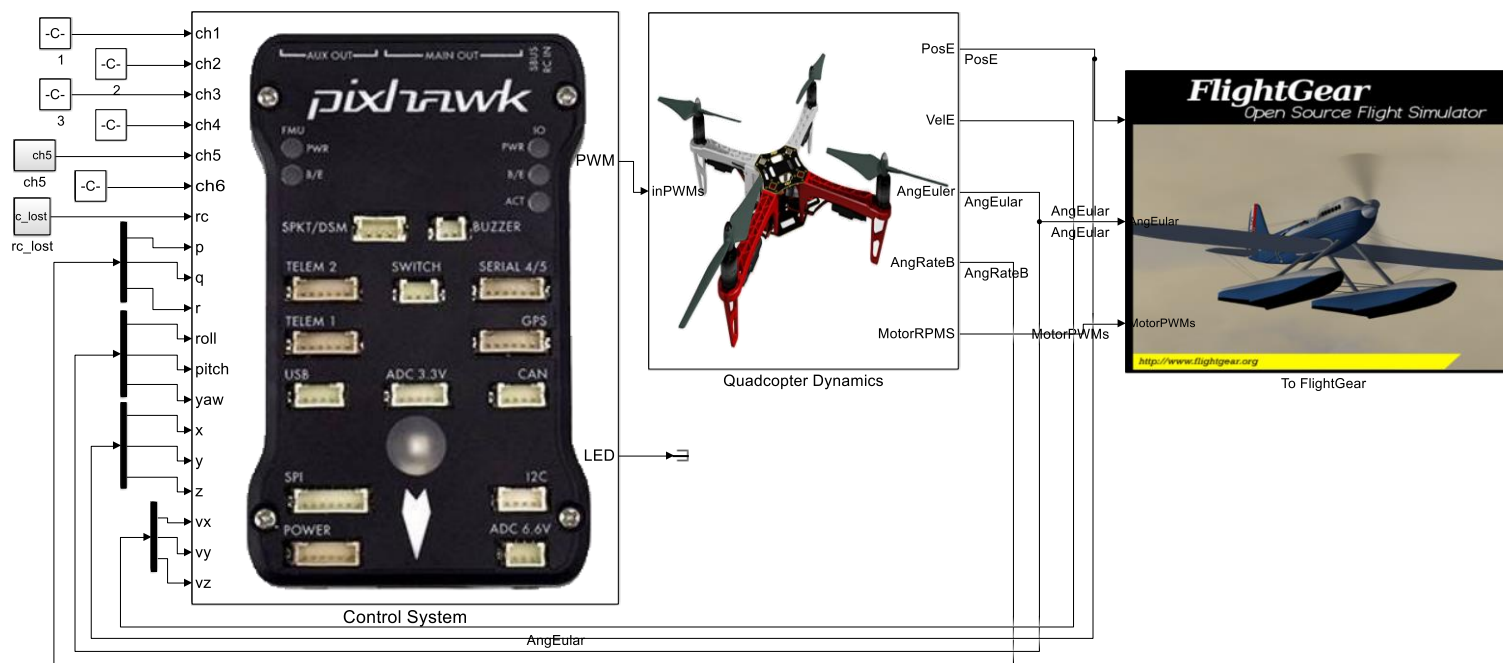


图. e8\_3. slx截图



# 设计实验

## □ 仿真实验步骤

### (1) 步骤一：软件在环仿真

运行“Init\_control.m”文件进行参数初始化。  
点击“Run”按钮，即得到仿真结果，如图所示。  
可以看到在0~10s，多旋翼处于手动模式，飞行器自由飞行。10s后遥控器失联，同时水平位置偏离起飞原点较远，高度小于阈值，多旋翼先上升到安全高度，然后进入返航模式，18s基本完成返航操作，水平位置回到原点，多旋翼进入着陆模式，35s完成着陆。

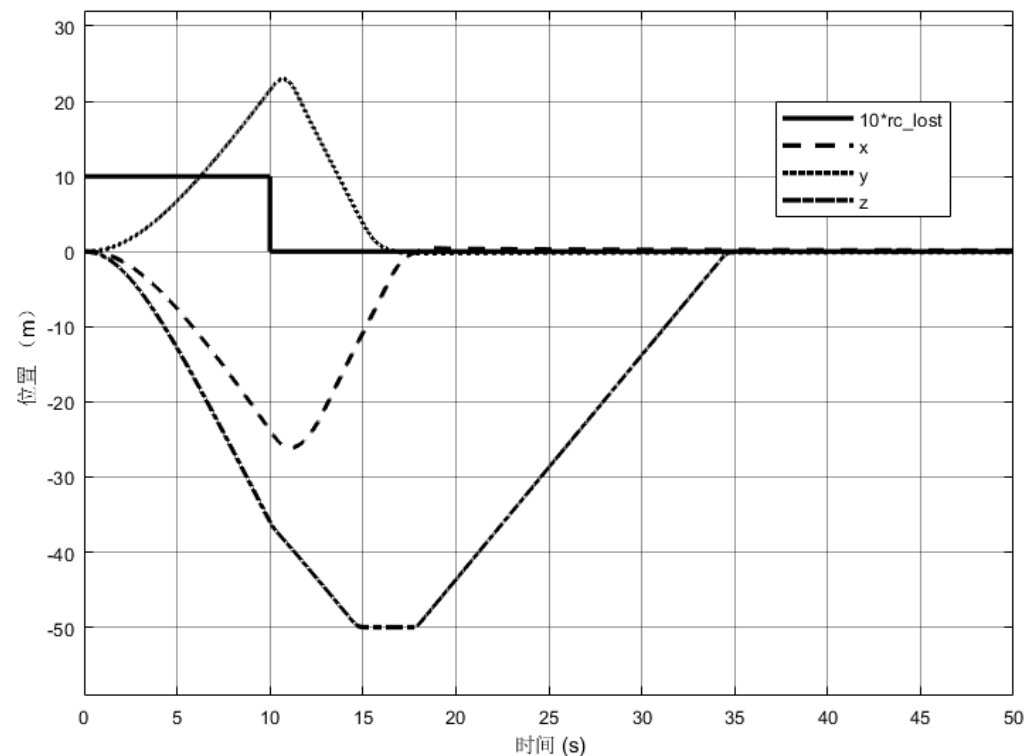


图. 失效保护仿真结果



# 设计实验

## □ 实验步骤

### (2) 步骤二：硬件在环仿真

#### 1) 打开Simulink仿真模型

运行“e8/e8.3/HIL/Init\_control.m”，初始化参数。“e8\_3\_HIL.slx”文件会自动打开，如右图。值得注意的是，这里的“Control System”模块与软件在环仿真中的相同。

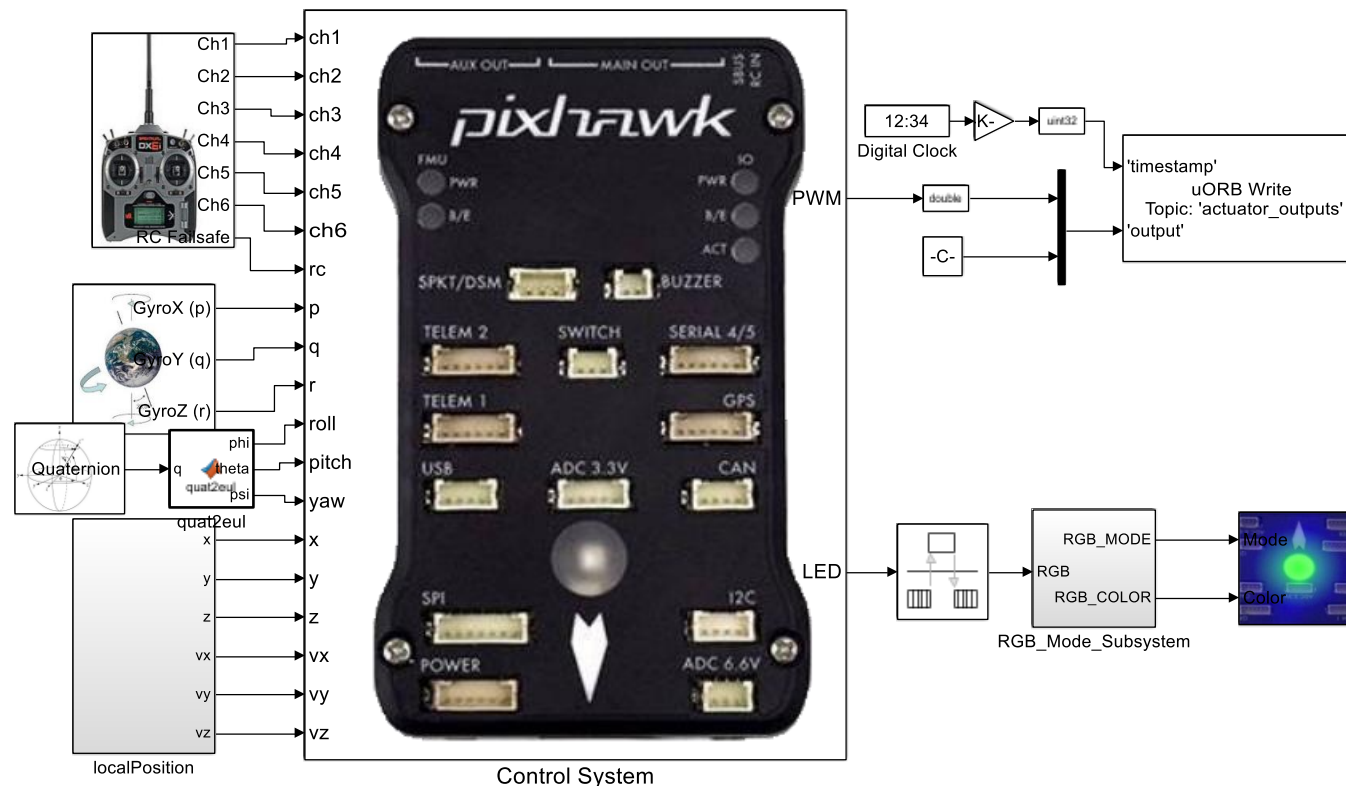


图. e8\_1\_HIL.slx截图



# 设计实验

## □ 实验步骤

### 2) 硬件连接

需要注意的是，在进行硬件在环仿真时**机架类型**应该是“**HIL Quadcopter X**”。

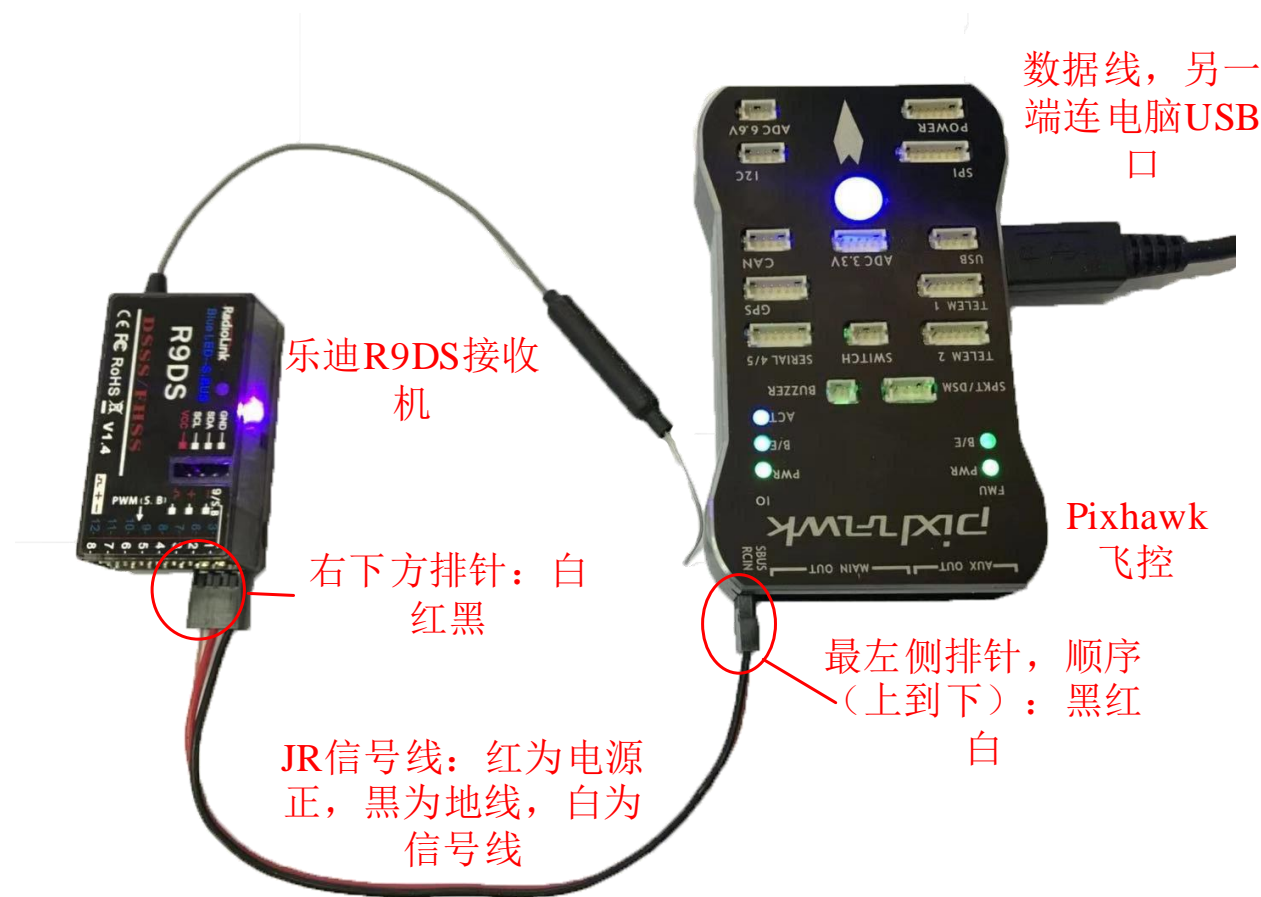


图. 硬件系统连接



# 设计实验

## 实验步骤

### 3) 代码编译及下载

将硬件在环仿真模型编译并下载文件到Pixhawk自动驾驶仪中。这样就可以在Pixhawk自动驾驶仪中运行我们自己设计的失效保护控制程序。

The image shows the process of compiling and downloading code to a Pixhawk. It consists of three main parts:

- Simulink Window:** The top window shows the 'E1\_rgbled\_system' project in Simulink. The 'Code' menu is open, and the 'PX4 PSP: Upload code to Px4FMU' option is highlighted. A red arrow points from this option to the command prompt.
- Command Prompt:** The bottom window shows the output of the compilation and download process. The text includes: '### Successfully generated all binary outputs.', 'Loaded firmware for 9.0, size: 875004 bytes, waiting for the bootloader...', 'if the board does not respond within 1-2 seconds, unplug and re-plug the USB connector.', 'PX4\_SIMULINK = y', '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 10dda9f8 cdb0cbbd 42cdc6b6 3ba305f7 81532581 84ee3da6 23bc6340 8321be68 edd356c9 1e3b8f', '5c 5e07decc 9c6be5a2 458a1513 4bbbbc21 eda35ce5 a8b840a5 ef019ca5 c89bb183 bb00f0c0 06db1a26 7375ff57 1ca41d94 24aa662e', 'ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff type: PX4', 'idtype: =00', 'vid: 000026ac', 'pid: 00000010', 'coa: Zu1H//9zzBXIrZQM28Wf0dbCDgb5U9Pv8wcwGdA1qw0/YDNOEN2p+M2wy71Czca206MF94FTJYGE7j2m17xjQIMhvmjt01bJHjuPXF4H3syca+W1RYo', 'VEOu7vCHto1z1qLhApe8BnKXIm7GDuWdwwAbbG1Zzdf9XHKQd1CSqZi4=', 'sn: 0038001f3432470d31323533', 'Erase : [=====] 100.0%', 'Program: [=====] 100.0%', 'Verify : [=====] 100.0%', 'Rebooting.', 'H:'.
- Annotations:** Red arrows and text labels indicate the flow: '点击下载' (Click Download) points to the 'PX4 PSP: Upload code to Px4FMU' option, and '下载完成' (Download Complete) points to the 'Rebooting.' status in the command prompt.

图. 代码编译及下载流程





# 基础实验

## 实验步骤

### 4) 模型仿真器软件配置。

双击 CopterSim 桌面快捷方式即可打开 CopterSim。读者可以选择不同的动力系统模型，步骤如下：单击“模型参数”自定义参数，然后单击“存储并使用参数”存储并使用参数。软件会自动匹配串口号，单击“开始仿真”按钮就可以进入硬件在环仿真模式。此时可以看到如图所示的界面左下角收到自驾仪返回的相关消息。

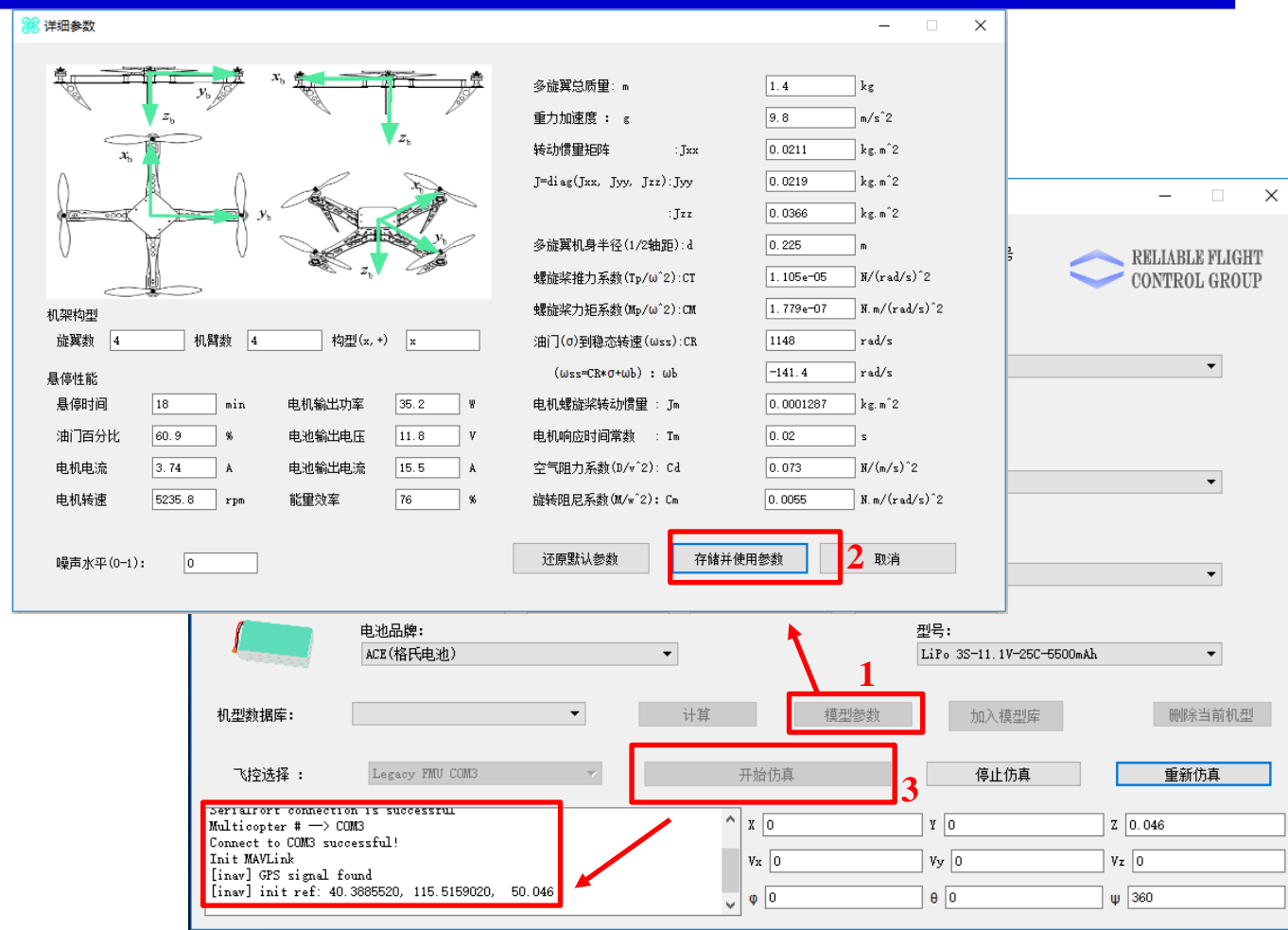


图. 模型仿真器软件配置





# 设计实验

## □ 实验步骤

### 5) 3DDisplay配置

双击3DDisplay桌面快捷方式打开三维显示软件。

### 6) 仿真效果

解锁多旋翼，先在手动模式下飞行一段时间，然后关闭遥控器电源，这意味着遥控器失联了，可以看到飞行器自动返航并着陆。

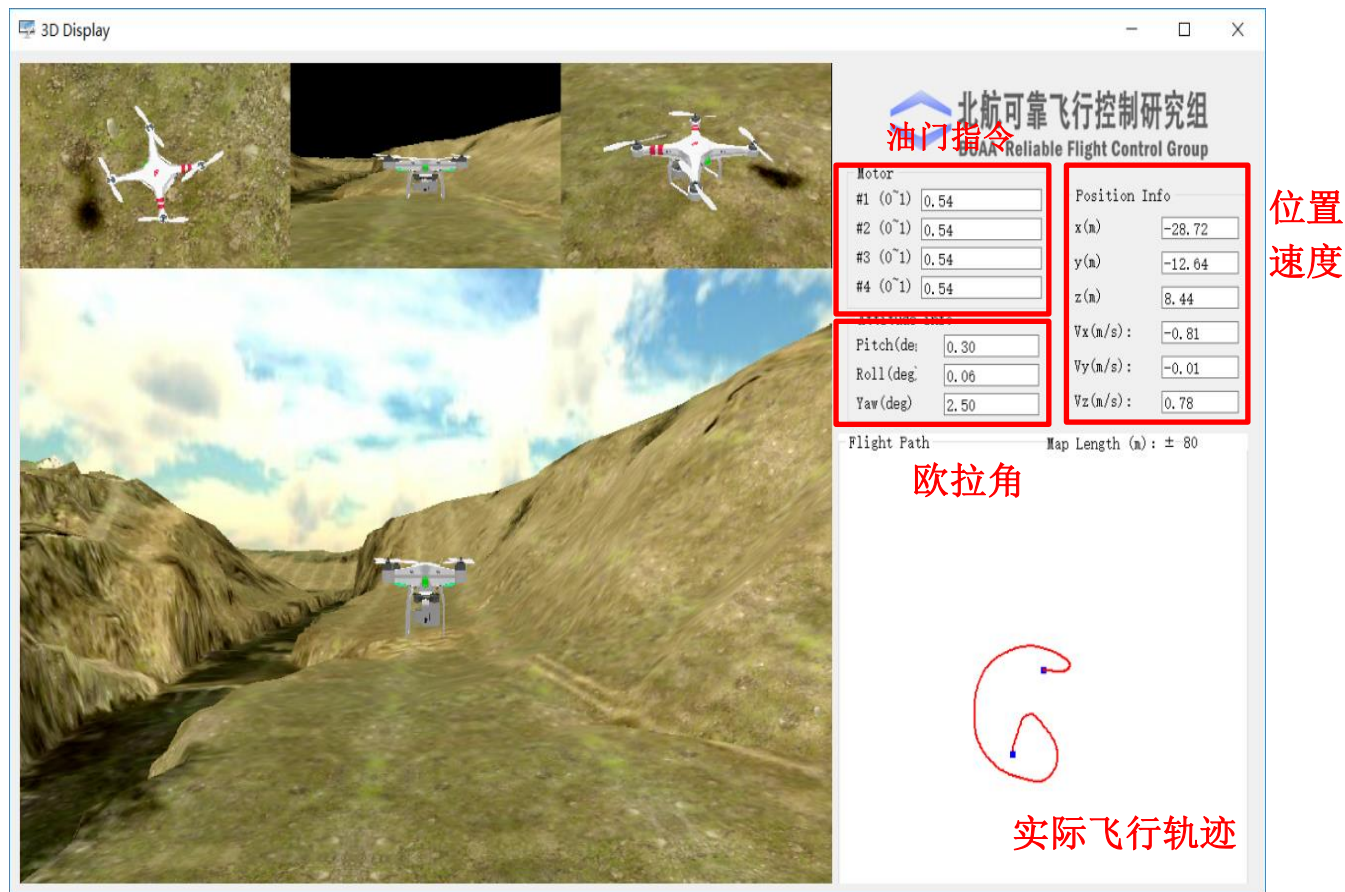


图. 3D Display



# 设计实验

## □ 实飞实验步骤

### (1) 步骤一：四旋翼准备

实际飞行试验所采用的多旋翼为F450 四旋翼，如图所示，在实际飞行时需要在QGC 中将Pixhawk 的机架类型从“HIL Quadcopter X”修改为“DJI Flame Wheel F450”，并完成传感器校准。



图. F450 机架示意图





# 设计实验

## □ 实飞实验步骤

### (2) 步骤二：Simulink实飞模型

相比设计实验中的硬件在环仿真模型，这里只是将其PWM 输出部分替换了。另一个改动部分将仿真中的安全高度设置为5m。为了记录飞行中的数据，我们在模型中加入了一种新的数据记录的方式，在打开模型时会提示“invalid .msg specified”的错误对话框。添加自定义logger数据详细步骤见实验五。

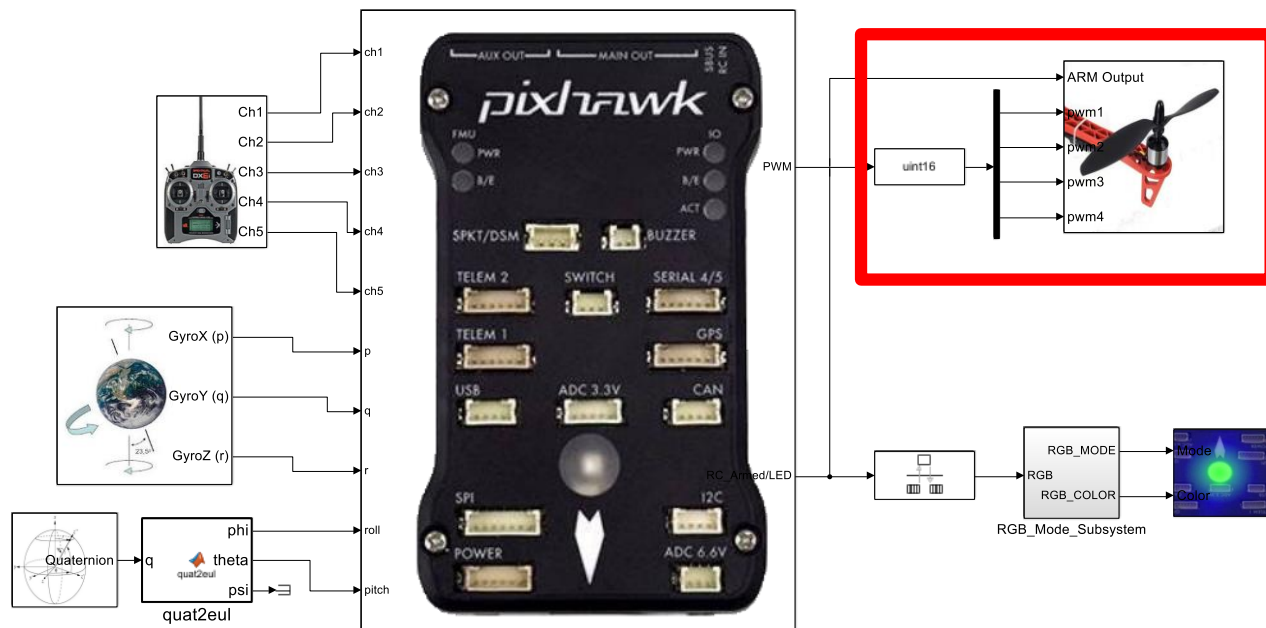


图. 失效保护实飞模型







# 设计实验

## □实验步骤

### (3) 步骤三：下载代码

与硬件在环仿真过程中编译下载代码过程完全一致。

### (4) 步骤四：室外实飞测试

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，如右图所示。另外应在空旷场地上进行实验，以确保GPS信号良好。



图. 室外实飞







# 设计实验

---



北航可靠飞行控制研究组  
BUAA Reliable Flight Control Group



# 设计实验

## □ 实验结果

读取飞控的log数据，如图所示。前335s是自稳模式，如图中“a”阶段所示，在这个阶段四旋翼在飞控手操控下自由飞行。在335s左右遥控器失联，如图中“b”阶段所示，在这个阶段四旋翼高度升至5m，同时水平位置回到起飞点，完成返航。最后四旋翼进行着陆，如图中“c”阶段所示，在这个阶段四旋翼水平位置保持在起飞点，高度降为0. 实际飞行效果和实验数据都表明四旋翼能够实现失效保护。

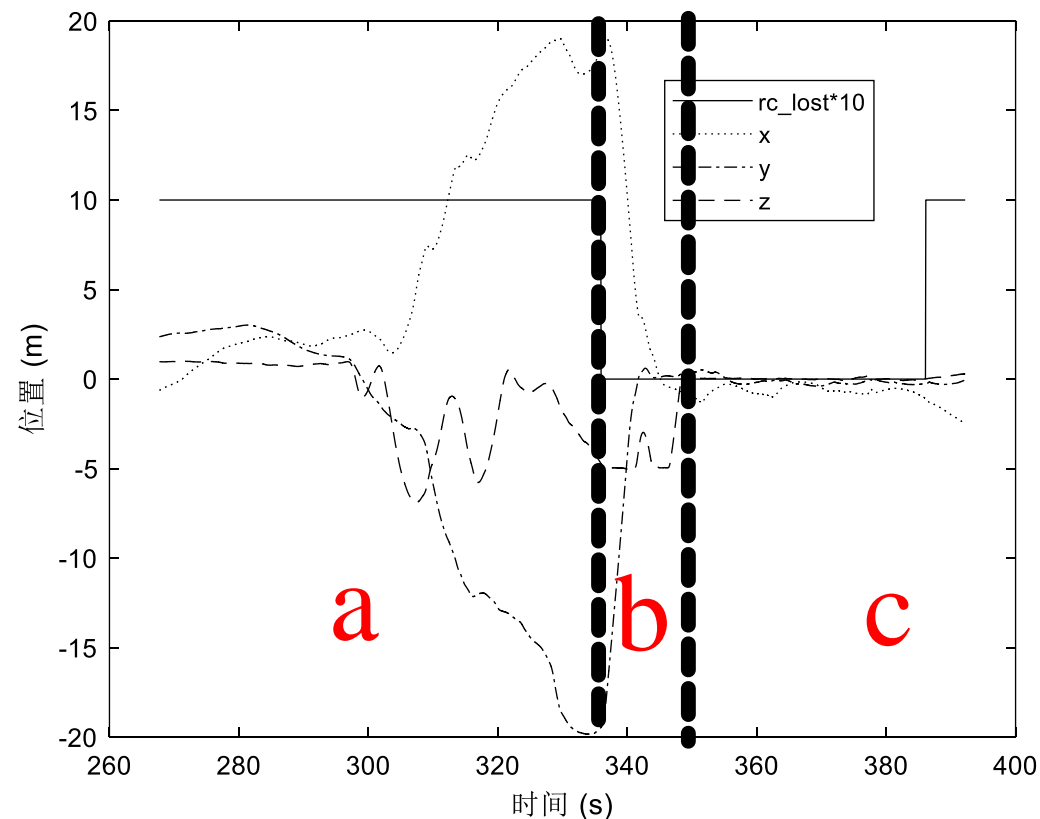


图. 失效保护室外飞行数据





## 小结

---

- (1) 根据实际情况，可以设计状态机的模式和事件来简化对飞行的控制。
  - (2) 在本章节，还要注意的是对三种飞行模态的理解。返航模态中多旋翼返回到出发时的水平位置，并保持在一定的高度；而着陆模态中多旋翼保持水平位置不变，降低高度以回到地面，这两者是不同的。这两种模态是由自驾仪控制，而在手动控制模式，遥控器和自驾仪同时起作用。
  - (3) 在此次实验中，主要考虑的是遥控器的连接情况，以及高度和水平位置的限制条件，在实际情况中，则需要在现在的基础上，添加其他的失效事件以及考虑健康评估（包括起飞前和飞行中），从而完善已有的状态，使得设计满足实际状况下的可靠飞行。
- 如有疑问，请到<https://rflsim.com>查询更多信息。





---

# 谢 谢！