

# 1. 实验名称及目的

## 1.1 实验名称

六旋翼综合模型设计及仿真验证（仅限完整版及以上版本）

## 1.2 实验目的

在Simulink的Dll模型基础上，基于MATLAB/Simulink设计六旋翼控制器，并将控制器和动力学模型放在同一个slx文件中，依据特定的输入输出接口，形成一个六旋翼整体仿真闭环，即综合模型。在得到综合模型后，通过外部控制的方法实现顶层控制。

## 1.3 关键知识点

本实验需要电脑中部署Visual Studio

2022环境，部署方式见：[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\6.VisualStudioInstall](#)

### 1.3.1 背景

在原有动力学模型的基础上实现控制器，构成综合模型。控制器使用MATLAB/Simulink实现基本姿态控制、定点功能。控制器直接拿取模型的真实状态作为输入。

PX4能够通过有线或者wifi被独立的辅助计算机控制，辅助计算机通常通过MAVLink API进行通信。载具执行辅助计算机通过MAVLink设定的位置、速度、姿态指令。Offboard模式主要用于做空中机动，至于起飞、降落、返航则选用相应专有的模式更为合适。

综合模型Offboard控制的消息主要参考pymavlink实现，pymavlink Offboard控制主要分为两类，一类是位置类，常用于编队飞行；一类是姿态类，常用于特技动作。

#

设置期望位置、速度、加速度消息、偏航角、偏航角速率，对于位置使用经纬高时会化为整数

```
set_position_target_local_ned_send(self, time_boot_ms, target_system,
target_component, coordinate_frame, type_mask, x, y, z, vx, vy, vz, afx, afy,
afz, yaw, yaw_rate, force_mavlink1=False)
```

# 设置期望姿态、角速率、油门

```
set_attitude_target_send(self, time_boot_ms, target_system, target_component,
type_mask, q, body_roll_rate, body_pitch_rate, body_yaw_rate, thrust,
force_mavlink1=False)
```

## 1.3.2 综合模型输入接口

### 1) inSILInts协议

inSILInts的第0个数字用于表征和修改状态，相应位为1时表明系统为相应的状态。例如，第1位表示仿真模式，当接收到的inSILInts[0]第1位为1时表明系统进入仿真模式。

只有当0:hasCMD为1时才去设置一次状态，否则综合模型将沿用原有状态。原有状态可以来自设置外部设置值，也可以是内部状态自动转换。例如，接收到起飞命令后，首先切换到起飞模式，起飞完成后自动切换到定点模式。以下为inSILInts[0]协议具体定义，该协议采用的按位选取，比如，如果需要输入解锁指令，则需要将inSILInts[0]置为2的2次方，4；同理可以解析其他标志位。

inSILInts[0] Vehicle Command Bitmap

0: hasCMD	1: SIL	2: Armed	3:	4:	5:	6:	7:
有新命令	仿真	解锁					
8: Takeoff	9: Position	10: Land	11: Return	12: Lotier	13: Height	14: Hor	15:
起飞	定点/waypoint	着陆	返航	盘旋 (固定翼)	定高模式	水平位置控制	
16: OffboardPos	17: OffboardAtt	18:	19:	20:	21:	22:	23:
Offboard 位置控制系列	Offboard 位置姿态系列						
24:	25:	26:	27:	28:	29:	30:	31:

注：当使能位置控制时，水平位置和竖直位置同时使能。

inSILInts[1]的第0-7位为位置类标志位，第8-15位为姿态类标志，解析规则同上。

注：当仅作为位置控制时对应的值为1，当仅作为速度控制时对应的值为2，当仅作为偏航角控制时对应的值为8，当仅作为角速率控制时对应的值为16。如果需多种控制结合，则将单独控制时的值相加。

### 2) inSILFloats协议

inSILFloats用于存放实际的数据，其具体含义可根据inSILInts的设定发生一些变化。如前3维表示位置，但具体是哪个坐标系下的则由inSILInts控制。

inSILFloats[0-2]=pos;

inSILFloats[3-5]=vel; //

速度|遥控器俯仰、滚转、偏航信号|inSILFloats[3]可以用作速率

inSILFloats[6-8]=acc;

inSILFloats[9-11]=att; // 姿态控制使用欧拉角，对于用户来说更加直观。

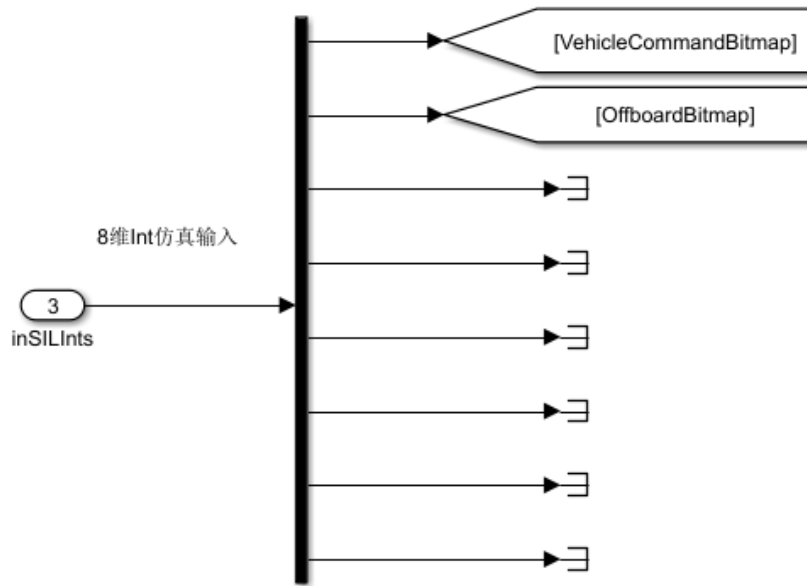
```
inSILFloats[12-14]=attRate;
```

```
inSILFloats[15]=thrust; // 油门
```

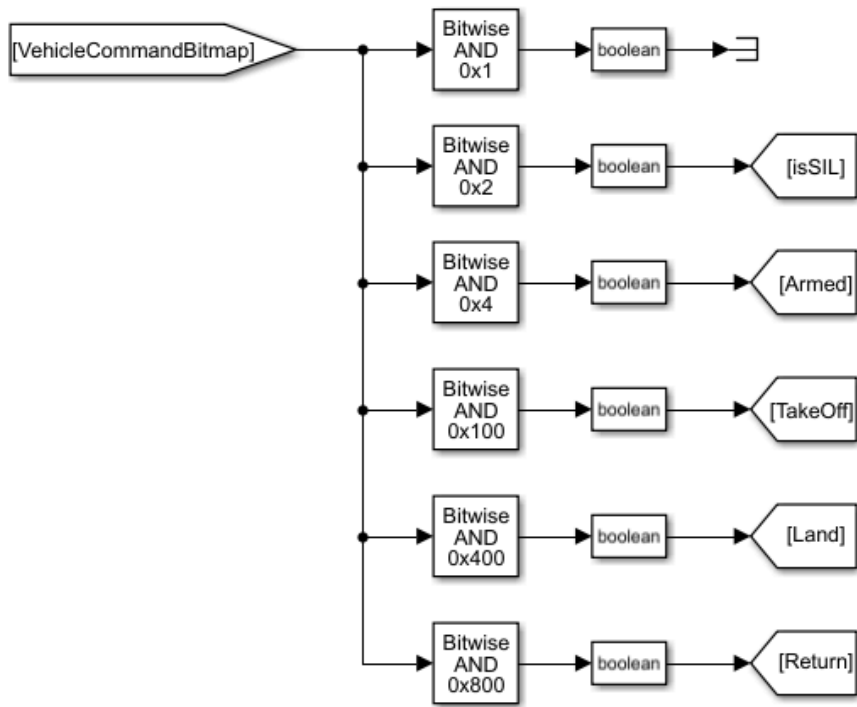
### 1.3.3 旋翼机综合模型实现

#### 1) 协议解析

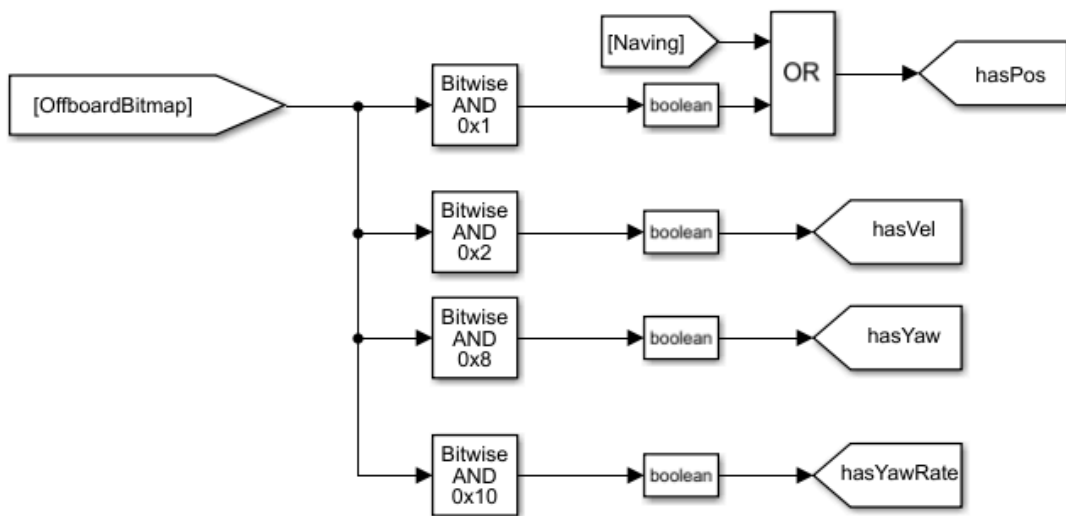
协议解析是将CopterSim收到的网络包解析成对应的指令。inSILInts是一个8维的输入，当前仅使用第0个数指令和第1个数Offboard模式。后续第6个数和第7个数将作为Global坐标系下的纬度和经度的整数表示。下图中，将inSILInts向量，分解为了8个单独的数。



指令的每一位都有相应的含义，所以还需进一步对每一位进行解析。如下图使用Bitwise模块对位进行解析，第0位暂时没有使用。第1位标识是否是仿真模式，当该位为1时表示使用仿真模式，当该位为0时表示硬件在环模式，只有仿真模式时综合模型中的控制器才生效。第2位表示是否解锁，其它位还包括起飞、降落、返航功能，具体含义可查看模型。



如下是标识有哪些offboard控制信息的标志。目前的协议支持完整的PX4 offboard控制，下面支持了当前项目中最为急需的，包括位置、速度、偏航、偏航角速率。

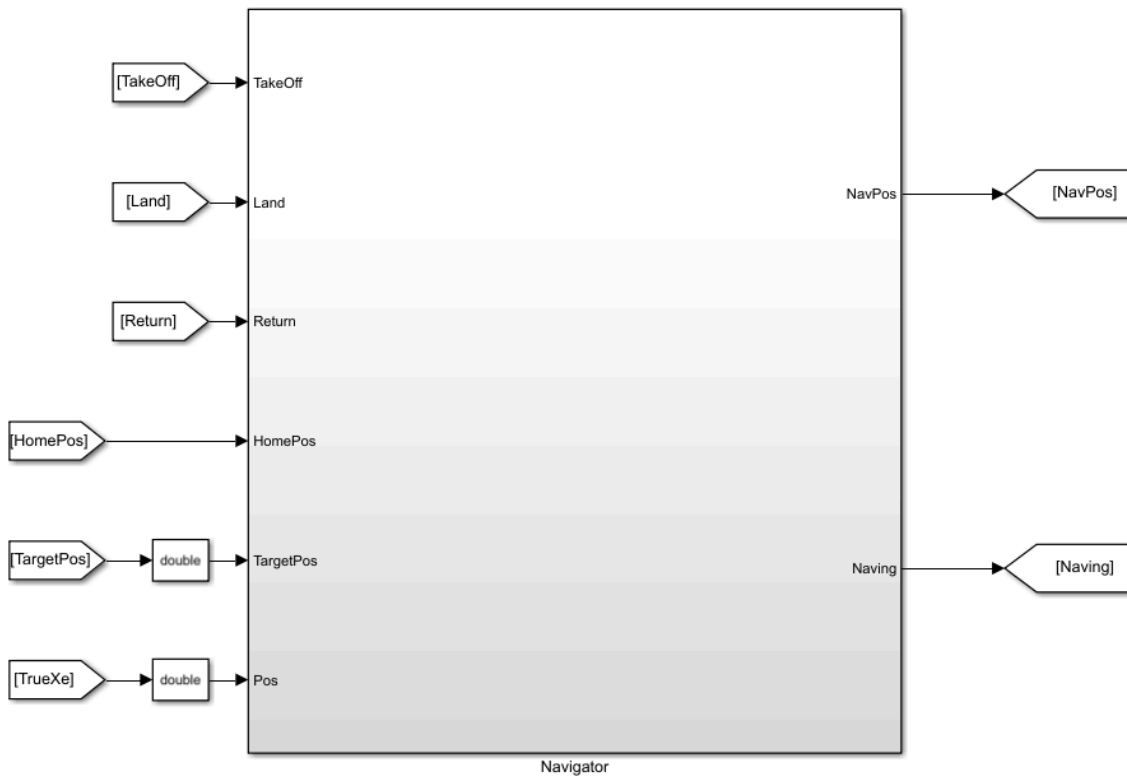


进一步，如下图所示解析offboard发送的实际位置、速度值等。在上图中是为了标识相应的值存不存在，而下图则是具体的数值。当前仅使用位置、速度、偏航、偏航角速率。在下图的解析中，使用了`hasVel`标志和`hasYawRate`标志。当这两个标志为假时，表明没有速度和偏航角速率的输入，这时将切换为遥控器模式。在遥控器模式下，1500代表期望速度或期望偏航角速率为0。

## 2) Navigator

为了让综合模型能够接收上层指令，如起飞、降落、返航等功能，需要将这些上层指令转换为一系列的期望位置。这个工作由Navigator模块完成。Navigator首先需要接收这些指令，然后在得到期望位置时需要依赖当前位置信息和用户设定的期望位置信息。举例而言，用户在起飞、返航、着落时都可以设置高度。此外用

户发送起飞指令从设计上只需发送一次，那么Navigator就需要自己去判断起飞是否完成，这样就需要同时知道期望位置和实际位置。

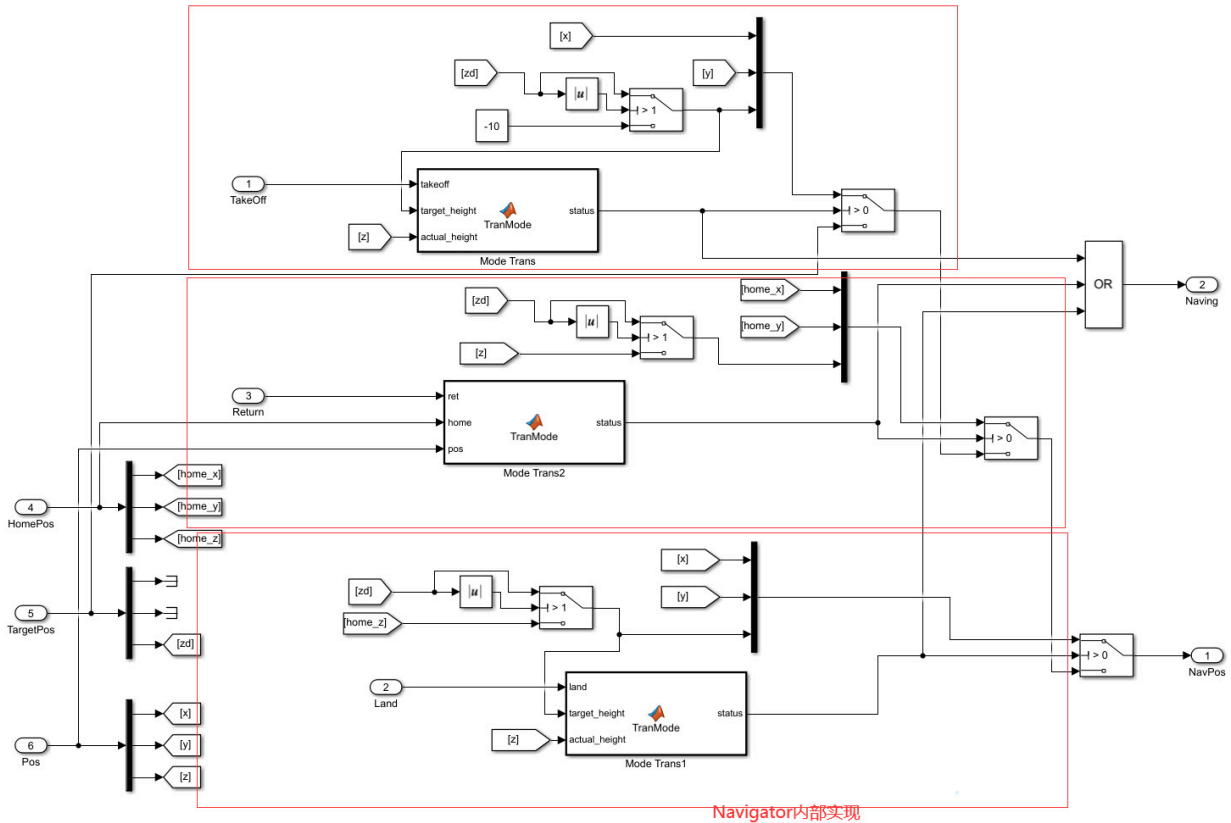


如下图，是Navigator的内部实现。最上面是起飞模块，起飞模块的优先级最低。也就是同时收到起飞、返航、降落的优先级关系为：起飞<返航<降落。当用户没有指定高度时，其默认高度是NED坐标下-10m。考虑到从设计上来讲，只需成功接收到一次起飞指令，后续Navigator就会持续生成期望位置直至起飞到指定高度。所以设计了Mode

Trans模块，该模块的作用是当接收到起飞命令时切换到起飞模式，当飞机到达指定高度时将切换到位置模式。在起飞过程中，飞机不会响应用户设定的位置。

在返航时，飞机将从当前位置保持高度不变返航到home点位置。返航也可设置返航高度，当没有指定高度时将以上一时刻的高度作为期望高度，即做高度保持。因为默认情况下不设置高度时，高度值为0，考虑到GPS定位精度一般不超过1m，所以将1m作为是否设置了高度的判据。home点是模型运行时，第一次运行到位置采集时记录的初始位置。

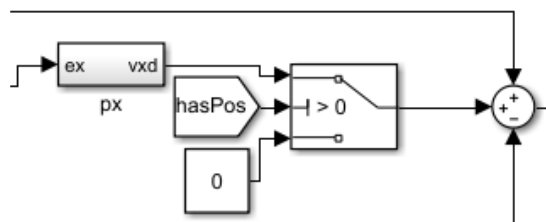
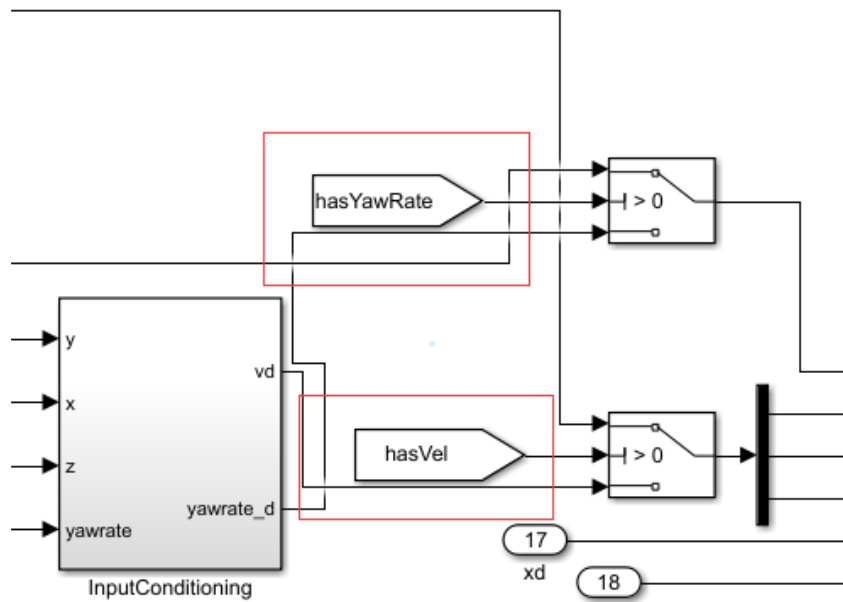
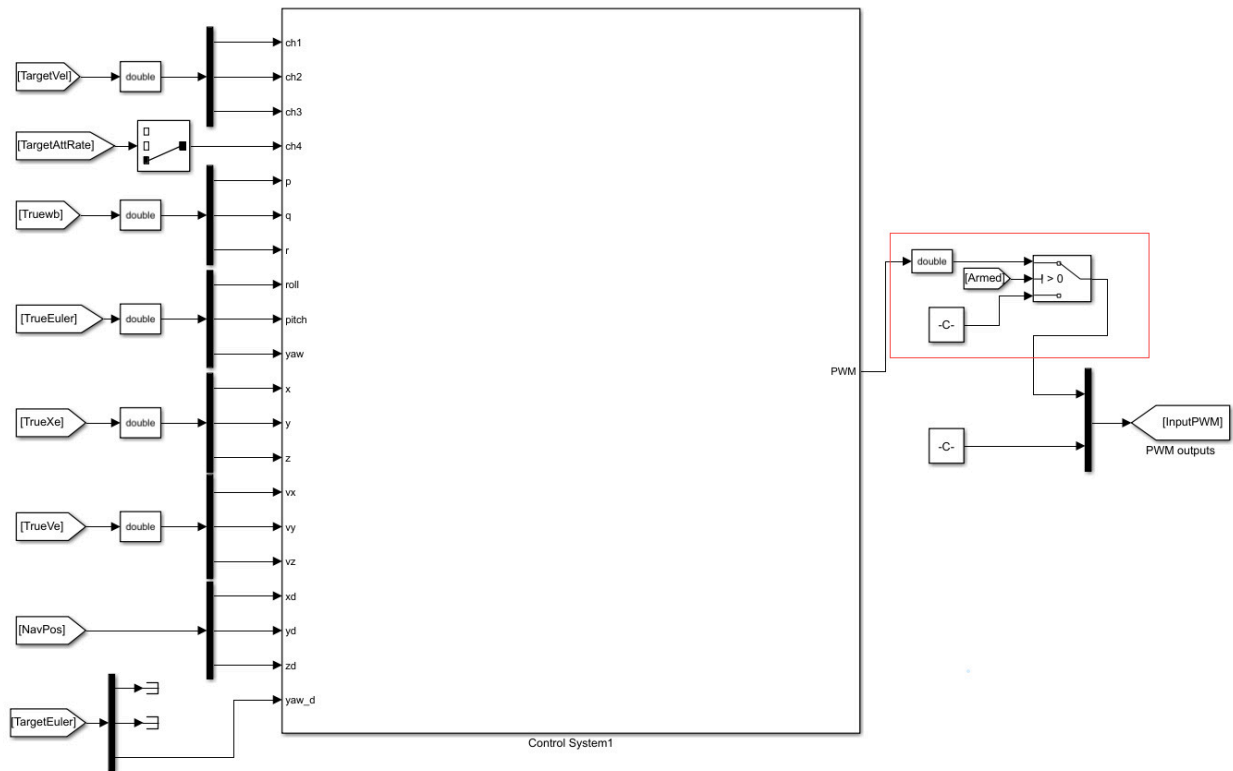
着陆是优先级最高的指令，着陆时同样可以指定高度。这是考虑到，着陆并不一定在home点位置着陆。其它外部模块可以检测当前离地面的距离，这样就可以确定着陆高度，所以着陆时预留了设置高度接口。



Navigator的输出是NavPos，是指期望的位置信息。Naving是指Navigator处于执行指令的过程中，当Naving为True时，会把hasPos设置为真。

### 3) 控制器

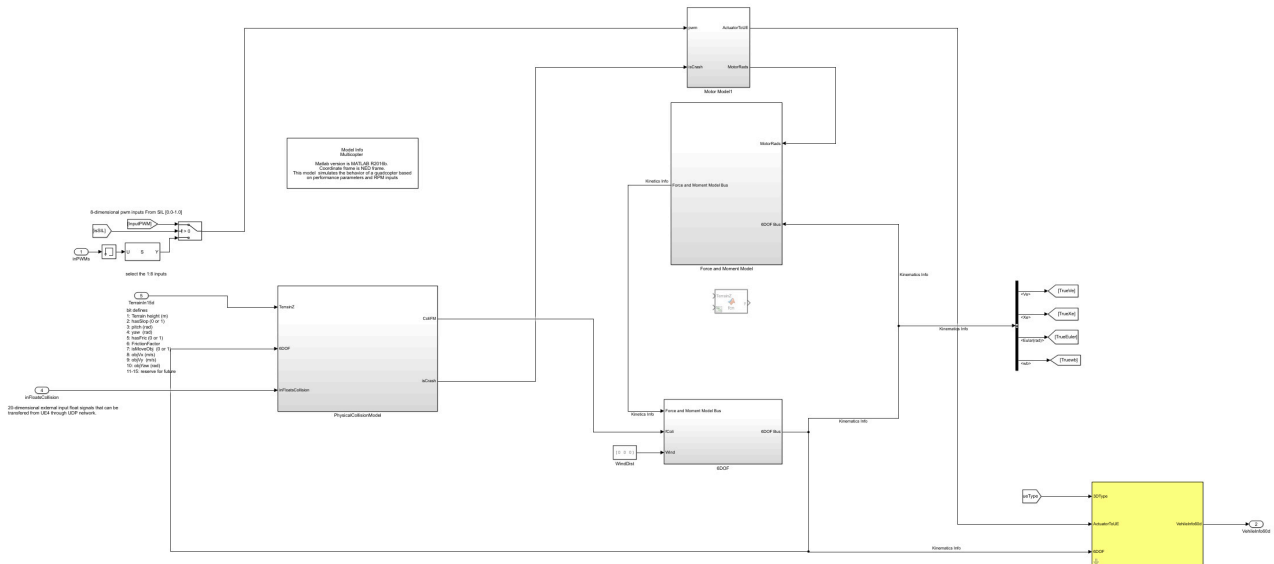
旋翼机综合模型是一个四串级的PID控制器，包含位置环、速度环、姿态环和角速度环。当前支持位置、速度、偏航角、偏航角速率的控制。当不进行位置控制时，hasPos标志为假，位置控制器的输出为0，这可在下图观察到。当没有收到速度信号或者偏航角速度信号时，将进入遥控器控制模式。当前只是预留了遥控器控制接口，遥控器的值设定为1500，没有真正启用遥控器。



## 4) 模型

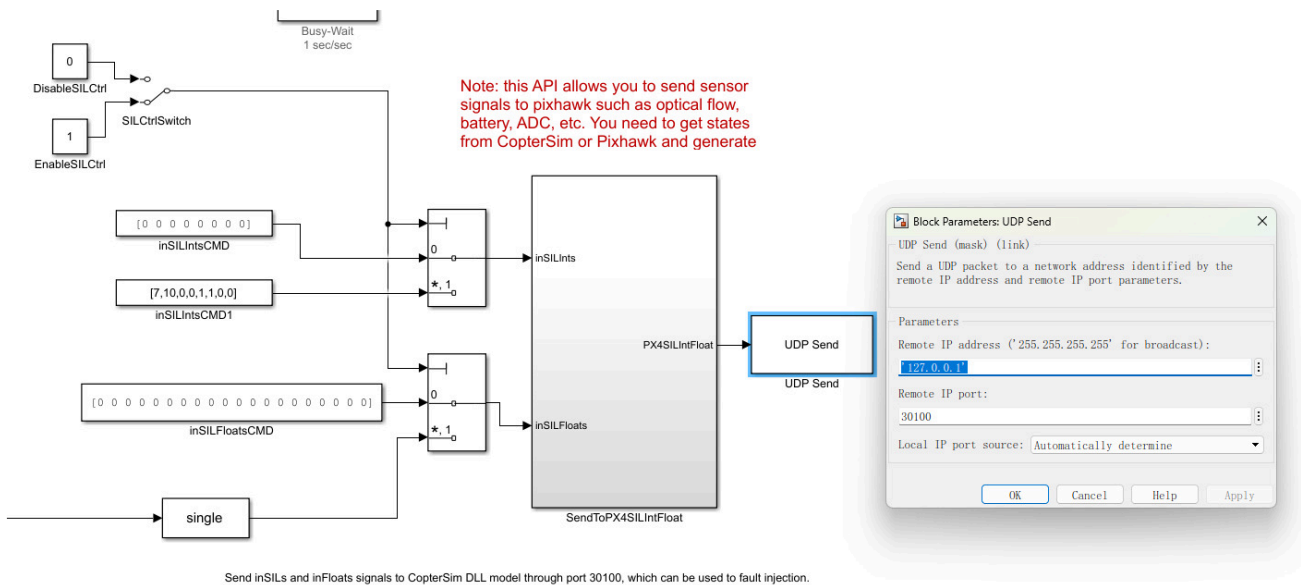
如下图所示是多旋翼的模型。该模型可以接收内部控制器的输入，也可以接收外部控制器的输入。这两种模式通过inSIL进行控制，当该标志为true时，表明是仿真模式，从内部控制器获得PWM波脉宽。当该标志为false时将从外部获得PWM波。

模型中包含的主要模块：电机模型、动力学模型、6DOF运动学模型、碰撞模型、三维显示模块。对于内部控制，由于没有设计滤波器，所以直接使用真值。同时将真实仿真数据发送给RflySim3D，用于仿真显示与测试。

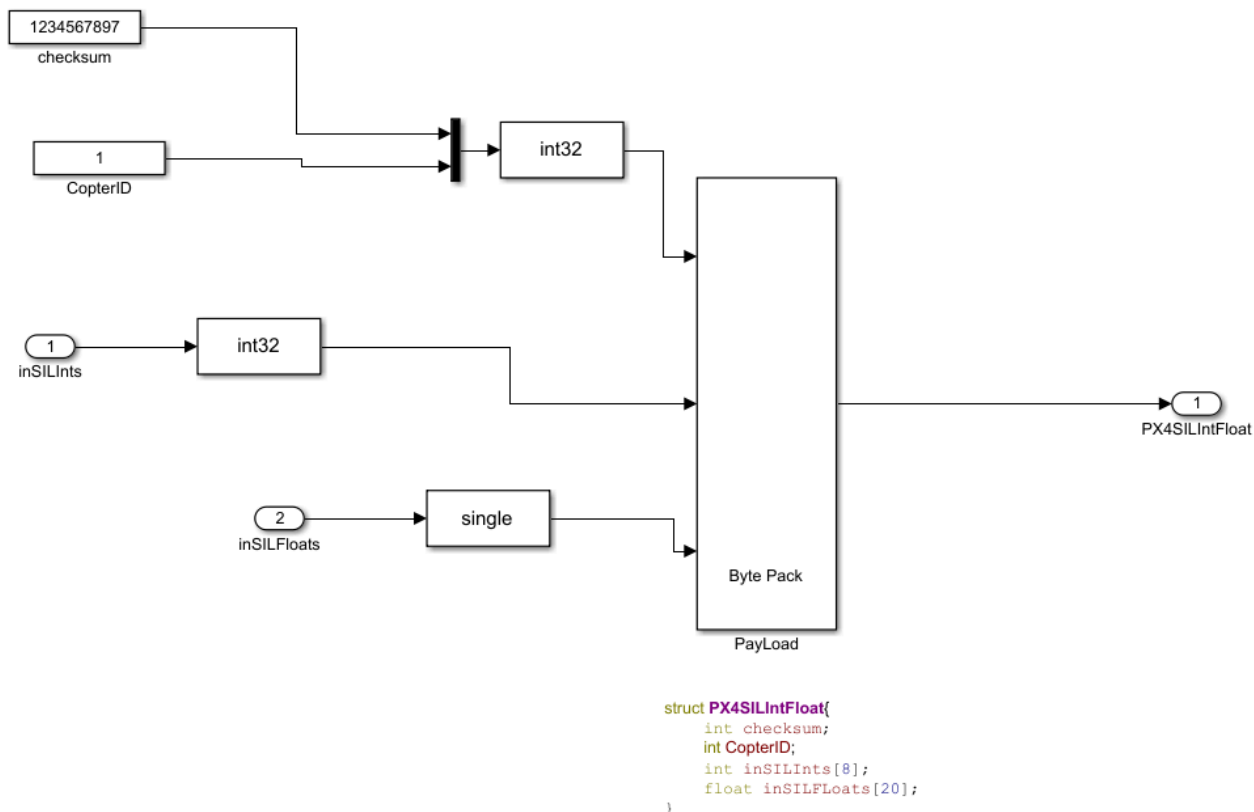


## 1.3.4 外部控制

Matlab通过UDP30100端口向旋翼综合模型发送期望位置、期望速度等控制指令。CopterSender.slx可以向带控制器的综合模型（模型中有inSILInts和inSILFloats接口）发送指令，从而控制多旋翼模型。



Send inSILs and inFloats signals to CopterSim DLL model through port 30100, which can be used to fault injection.



## 2. 实验效果

启动软件在环仿真后，通过外部控制的方法发送期望位置、速度等控制综合模型运动。

## 3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\4.RflySimModel\3.CustExps\e5\_CopterSimSILNoPX4\3.HexarotorNoPX4

文件夹/文件名称	说明
HexarotorNOpx4.slx	六旋翼综合模型，控制器为速度控制
CopterSender.slx	外部控制文件
<a href="#">HexarotorNOpx4.bat</a>	六旋翼综合模型启动脚本
HexarotorNOpx4.dll	六旋翼综合模型动态链接库，由HexarotorNOpx4.slx自动代码生成后打包形成
GenerateModelDLLFile.p	用于将自动代码生成的C++文件封装成动态链接库
<a href="#">Init.m</a>	旋翼模型参数
<a href="#">Init_control.m</a>	控制器参数

## 4. 运行环境

### 4.1 软件要求

Windows 10及以上版本；RflySim工具链；MATLAB 2017B及以上。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4\_fmuv6x\_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：<https://rflsim.com/doc/zh/1/Hardware.html>

### 4.2 硬件要求

笔记本/台式电脑① 1台；\\台；\\台。

①：推荐配置请见：<https://rflsim.com/>

## 5. 实验步骤

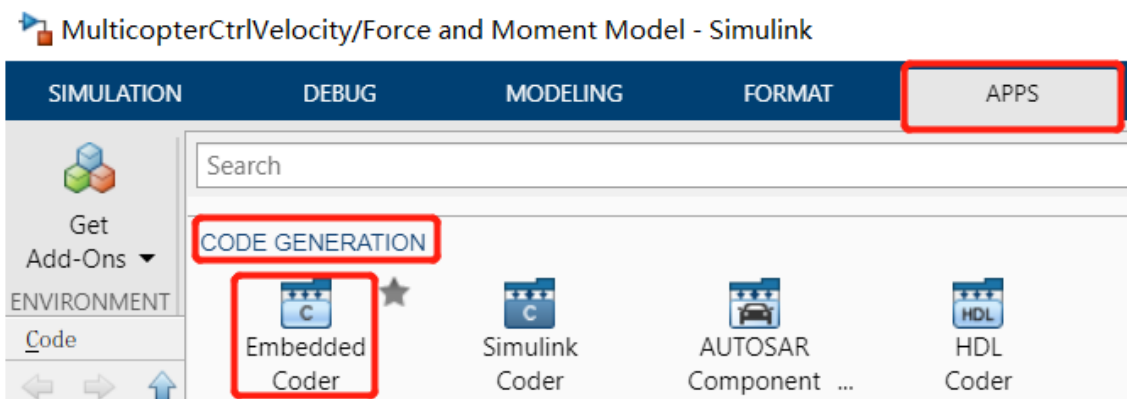
### Step 1: 编译模型

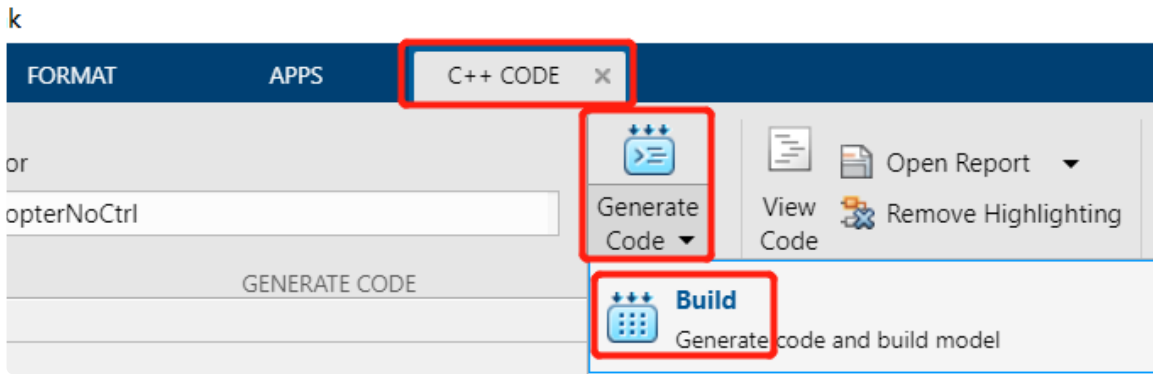
在MATLAB中打开HexarotorNOpx4.slx文件，在Simulink中，点击编译命令。编译配置可参考[4.RflySimModel\0.ApiExps\2.UserDefinedC++\2.GenC++\Readme.pdf](#)

对于MATLAB 2019a及之前版本，工具栏样式见下图，直接点击它的编译按钮“Build”即可。

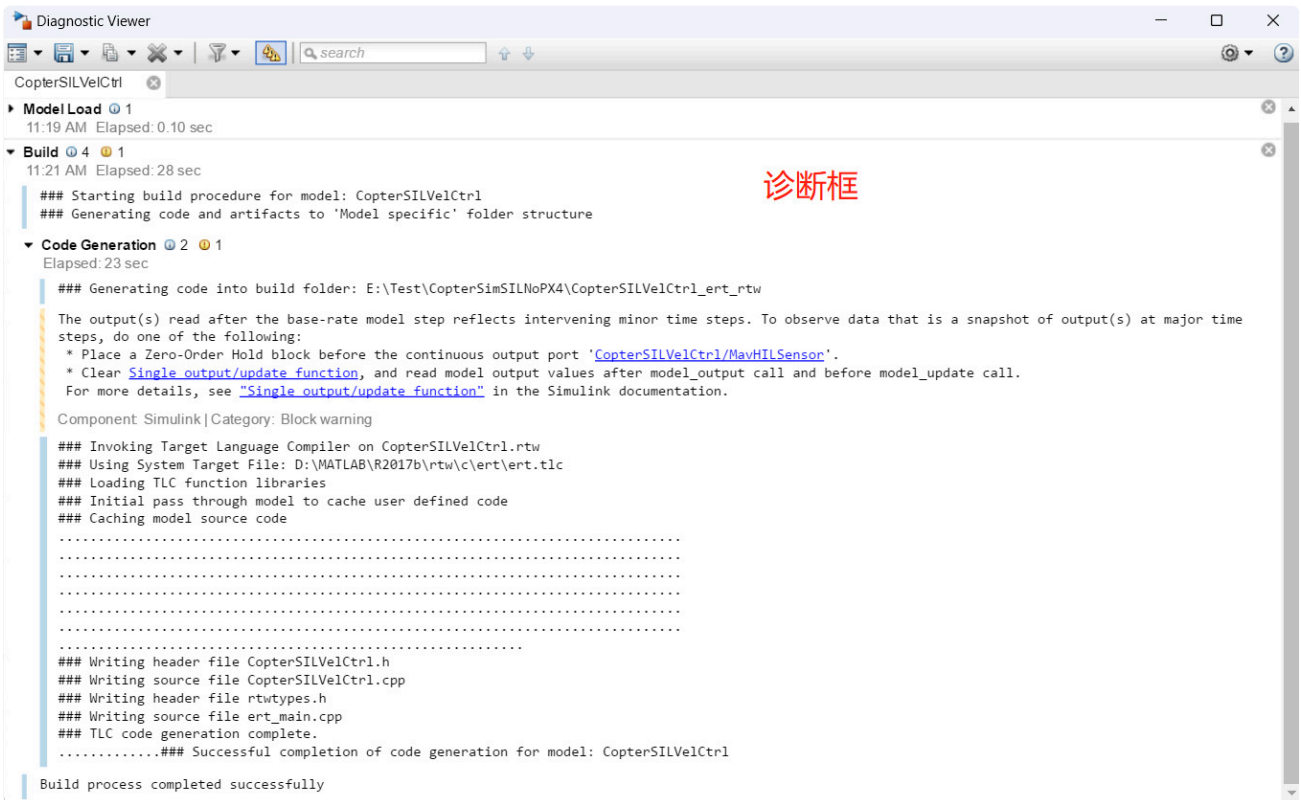


对于2019b及之后版本，点击APPS - CODE GENERATION - Embedded Coder才能弹出代码生成工具栏，在其中如下图所示点击“C++CODE” - “Generate Code” - “Build”按钮就能编译生成代码。



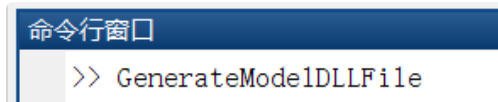


在Simulink的下方点击View diagnostics指令，即可弹出诊断对话框，可查看编译过程。在诊断框中弹出Build process completed successfully，即表示编译成功。

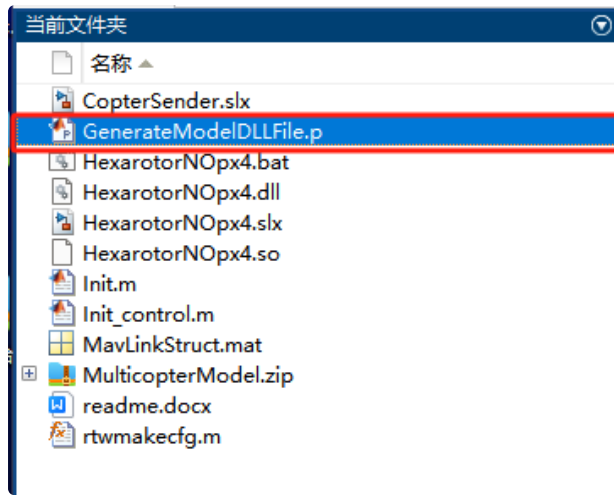


## Step 2: 生成DLL模型

右键运行GenerateModelDLLFile.p文件或在命令行窗口中输入GenerateModelDLLFile后回车，得到综合模型动态链接库HexarotorNOPx4.dll。

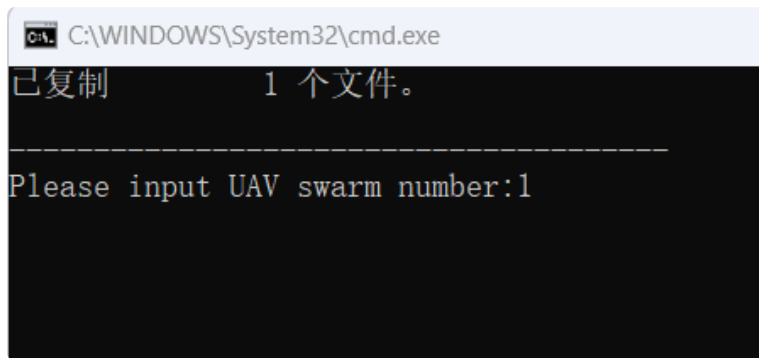


或



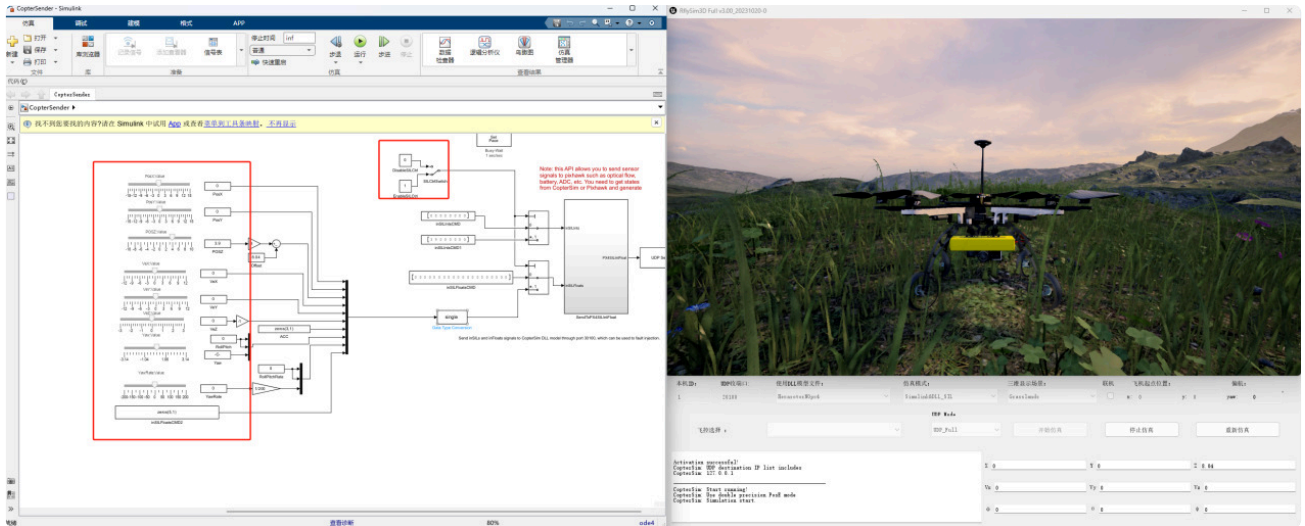
### Step 3: 启动仿真

右键点击 [HexarotorNOpx4.bat](#) 并以管理员身份运行，输入1，启动1架六旋翼综合模型的软件在环仿真。



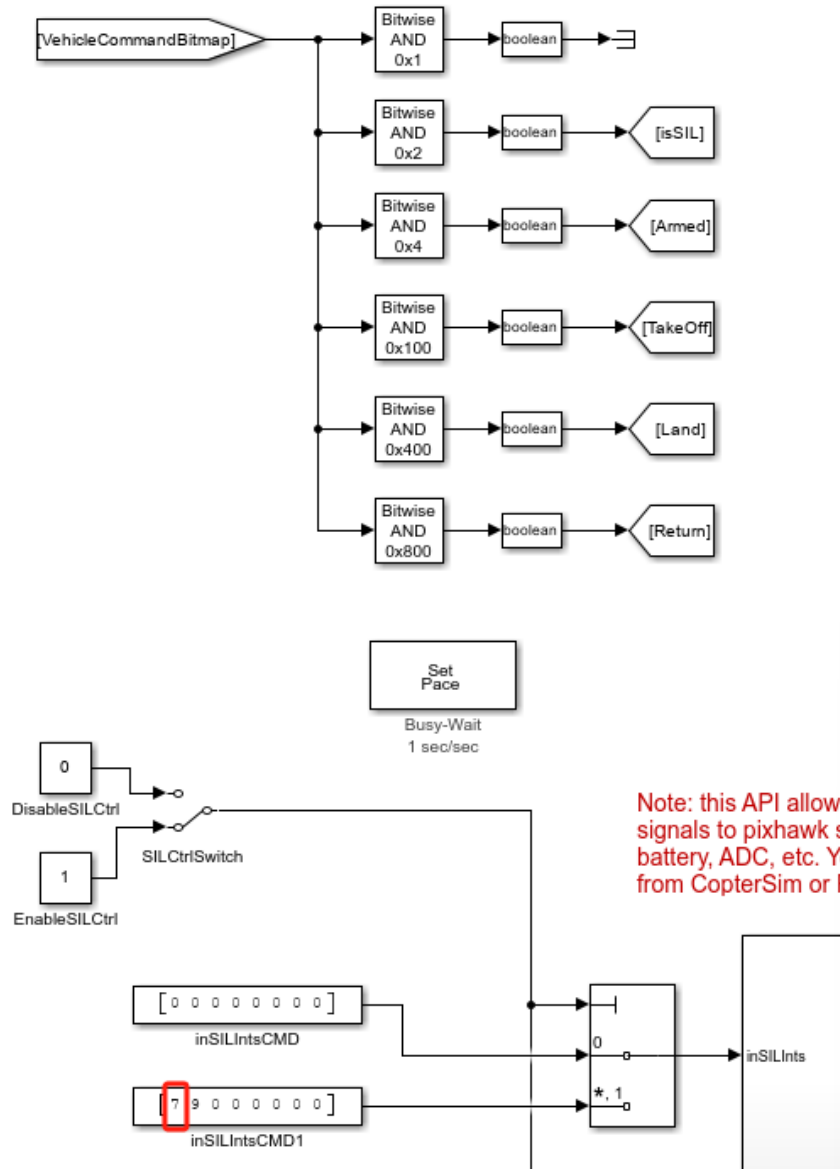
### Step 4: 运行控制模型

右键打开 [CopterSender.slx](#)，运行该文件，点击2处将使能标志切换到EnableSILCtrl。



## Step 5: 解锁并起飞

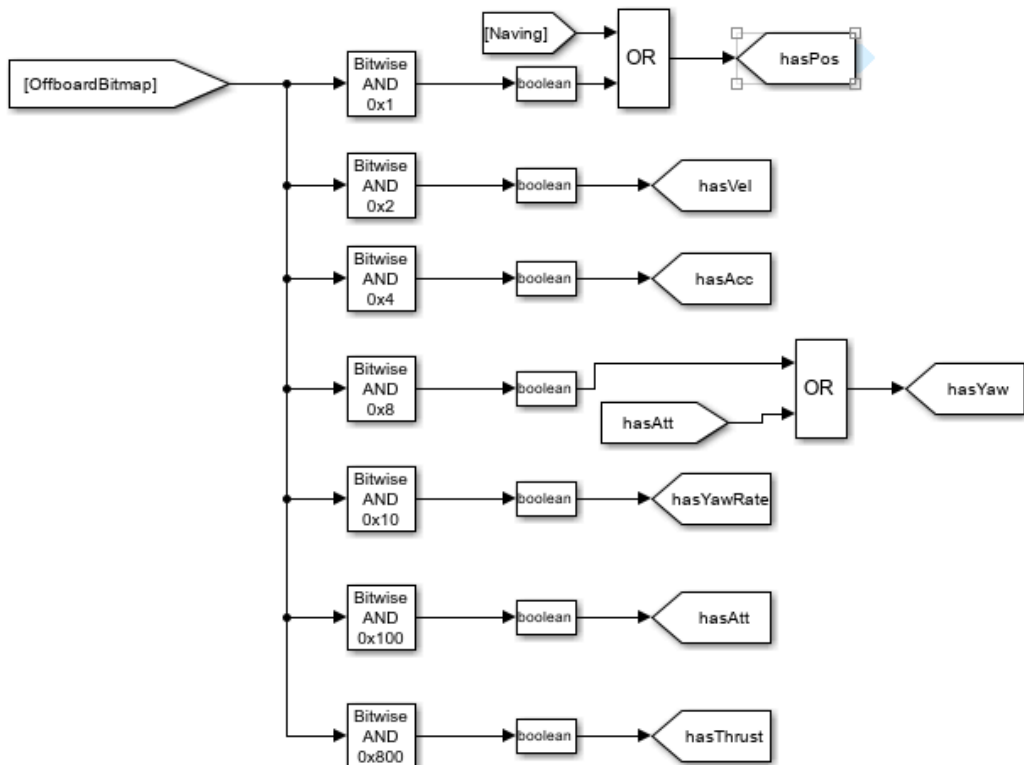
根据模型Bitmap, inSILInts[0]为7时六旋翼解锁并起飞, 接收控制指令。

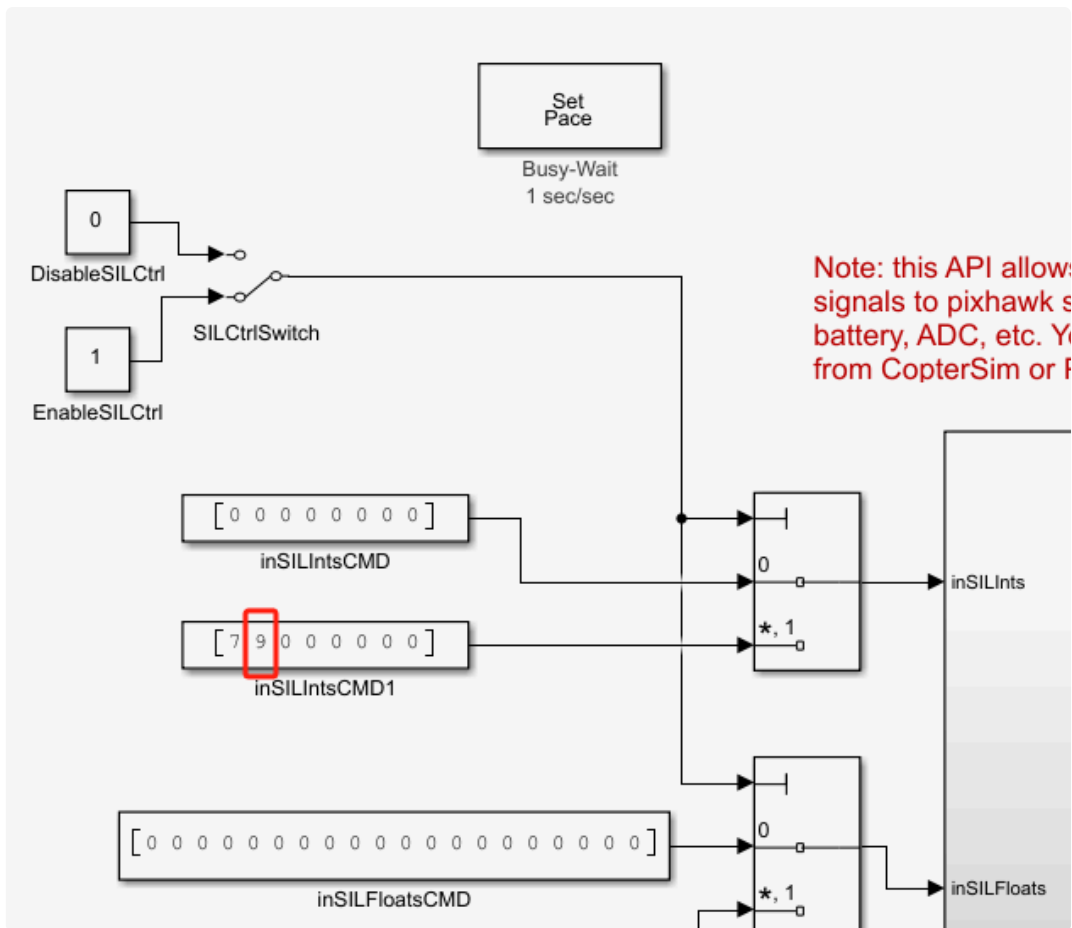




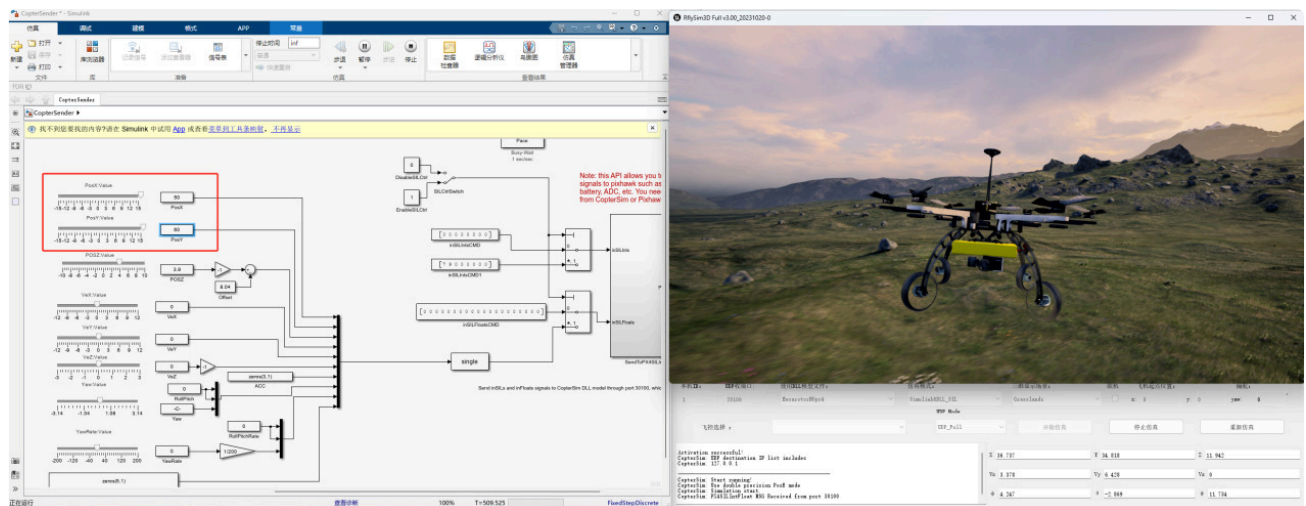
## Step 6: 位置与偏航角控制

根据模型Bitmap，inSILInts[1]为六旋翼支持的控制指令类型。

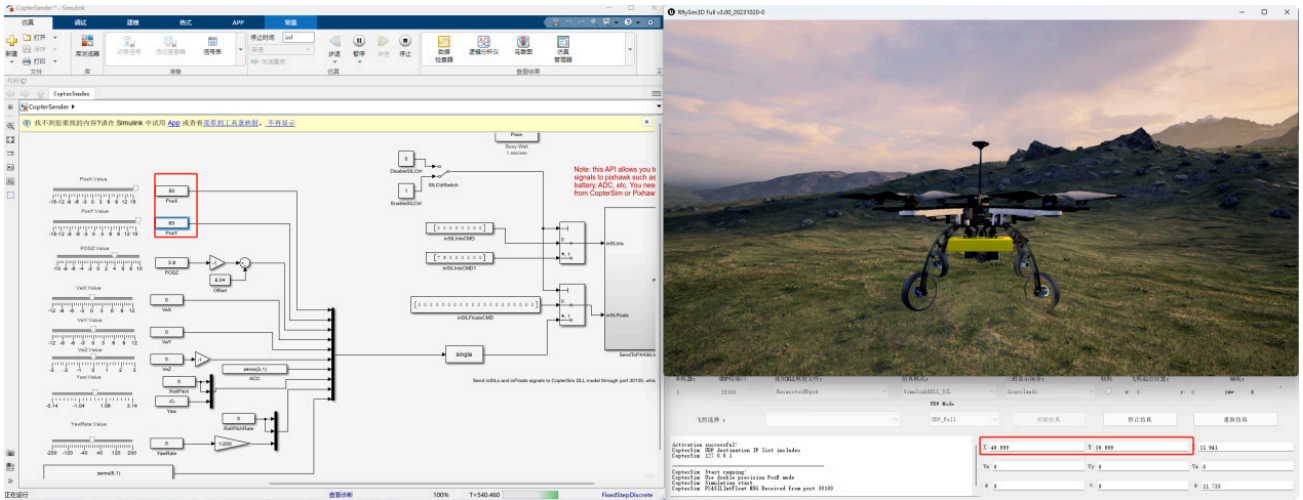




当inSILInts[1]为9时控制指令支持位置与偏航角控制，可在inSILFloats的对应位置滑动滑块控制六旋翼的位置与偏航角。

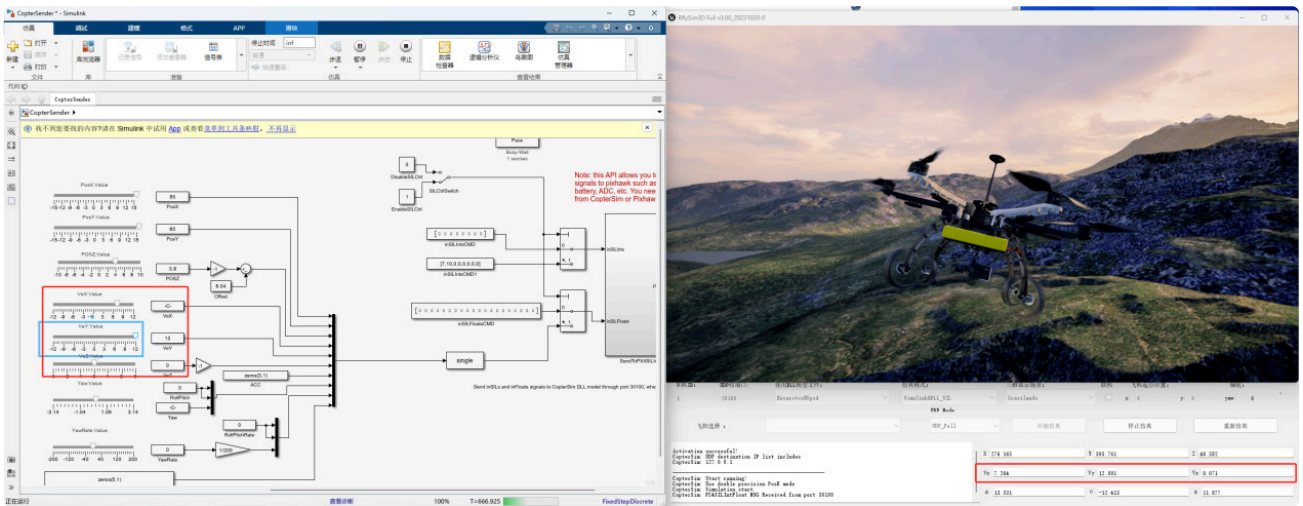


到达指定位置后六旋翼会悬停在期望位置。



## Step 7: 速度与偏航角控制

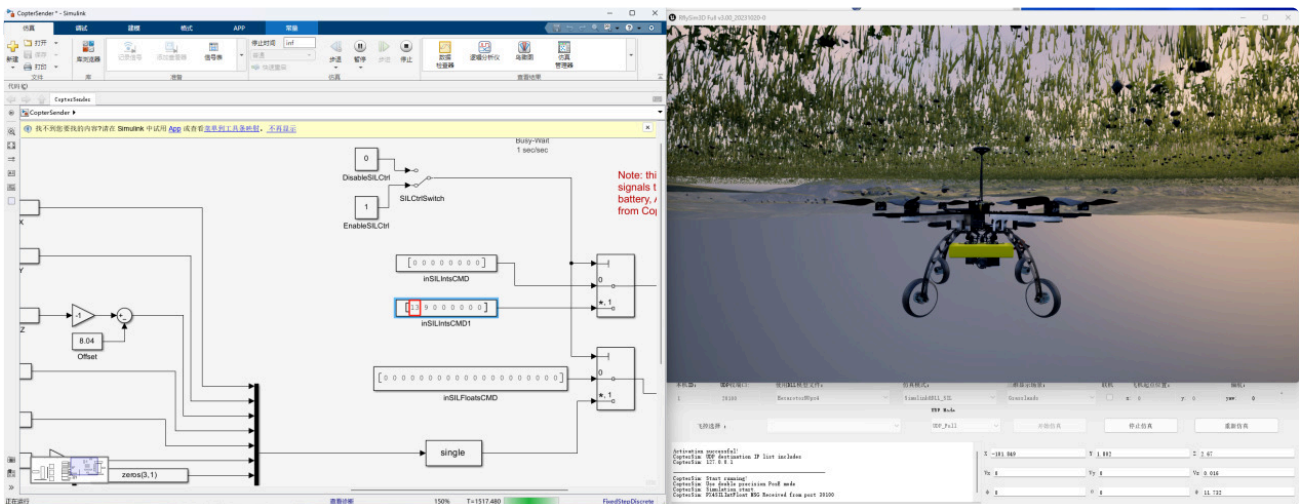
当inSILInts[1]为10时控制指令支持速度与偏航角控制，可在inSILFloats的对应位置滑动滑块控制六旋翼的速度与偏航角。



当各方向速度到达期望速度值时速度保持不变进行运动。

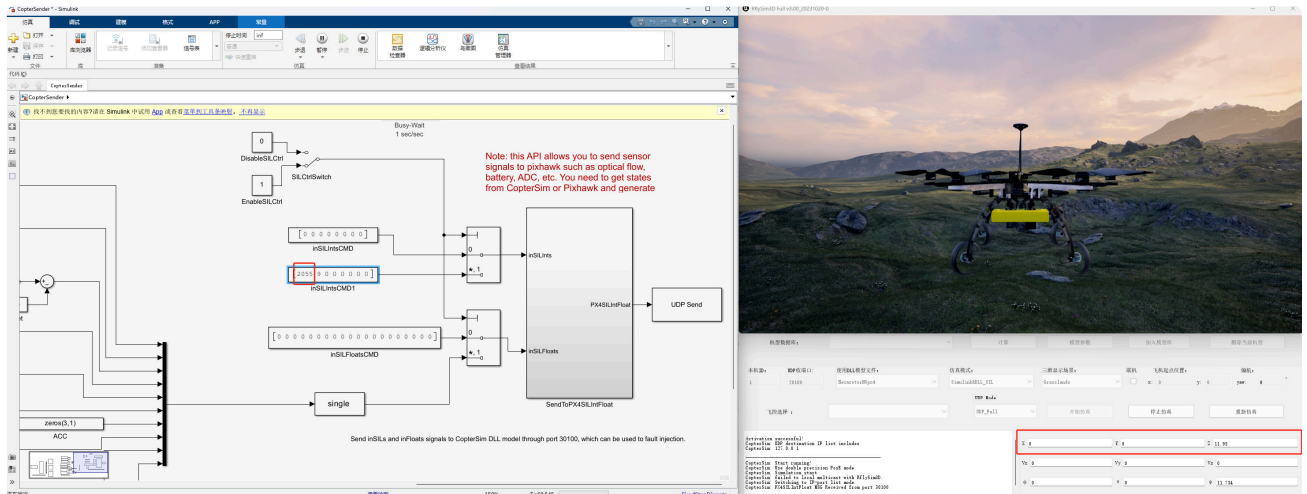
## Step 8: 降落

当inSILInts[0]为13时六旋翼会快速降落。



## Step 9: 返回至起点上方悬停

当inSILInts[0]为2055时四旋翼会返回起始点上方悬停。



## 6.参考资料

1. PX4PSP\RflySimAPIs\4.RflySimModel\API.pdf中DLL/SO模型与通信接口的重要参数部分。
2. [RflySim安装目录]/RflySimAPIs/4.RflySimModel/API.pdf
3. [RflySim安装目录]/RflySimAPIs/4.RflySimModel/API.pdf
4. [RflySim安装目录]/RflySimAPIs/4.RflySimModel/API.pdf
- 5.

## 7.常见问题

Q1:

A1:

Q2: 编译报错，无法加载库文件



A2: 这可能是由于安装平台时PX4PSP工具箱未更新到最新版，更新RflySim安装包后按照如下配置重新安装平台即可

Toolbox one-key installation script: RflySimA...

(1) Software package installation directory  
C:\PX4PSP

(2) PX4 firmware compiling command: firmware versions <= PX4-1.8 use format px4fmu-v3\_default; >= PX4-1.9 use format px4\_fmu-v3\_default  
px4\_fmu-v6c\_default

(3) PX4 firmware version (1: PX4-1.7.3, ... , 6: PX4-1.12.3, 7: PX4-1.13.2, 8: PX4-1.14.4, 9: PX4-1.15.0)  
9

(4) PX4 firmware compiling toolchain (1: WinWSL[suitable for all versions], 2: Msys2[suitable for <= PX4-1.8], 3: Cygwin[for >=PX4-1.8])  
1

(5) Whether to reinstall PSP toolbox (yes to reinstall and no to remain current installation)  
yes

(6) Whether to reinstall the dependent software packages (CopterSim, QGroundControl, CopterSim, etc. About 5 minites)  
no

(7) Whether to reinstall the selected compiling toolchain (yes to reinstall and no to remain unchanged, about 5 minites)  
no

(8) Whether to reinstall the selected PX4 firmware source code (yes to reinstall and no to remain unchanged, about 5 minites)  
no

(9) Whether to pre-compile the selected firmware with the selected command (yes to compile and no to remain unchanged, about 5 minites)  
no

(10) Whether to block the actuator outputs in the PX4 firmware code ("yes" to use Simulink controller, "no" to use PX4 official controller)  
no

OK Cancel