

1. 实验名称及目的

四旋翼综合模型仿真验证实验: 在 Simulink 的 Dll 模型基础上, 基于 MATLAB/Simulink 设计四旋翼控制器, 并将控制器和 Dll 模型放在同一个 slx 文件中, 依据特定的输入输出接口, 形成一个飞机整体仿真闭环, 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。

2. 实验原理

DllSilNoPX4TempDemo.slx 是四旋翼综合模型的实现, DllSilNoPX4TempSender.slx 可以通过 inSIL 接口向带控制器的综合模型发送指令, 并通过监听 30101 端口获取飞机数据。

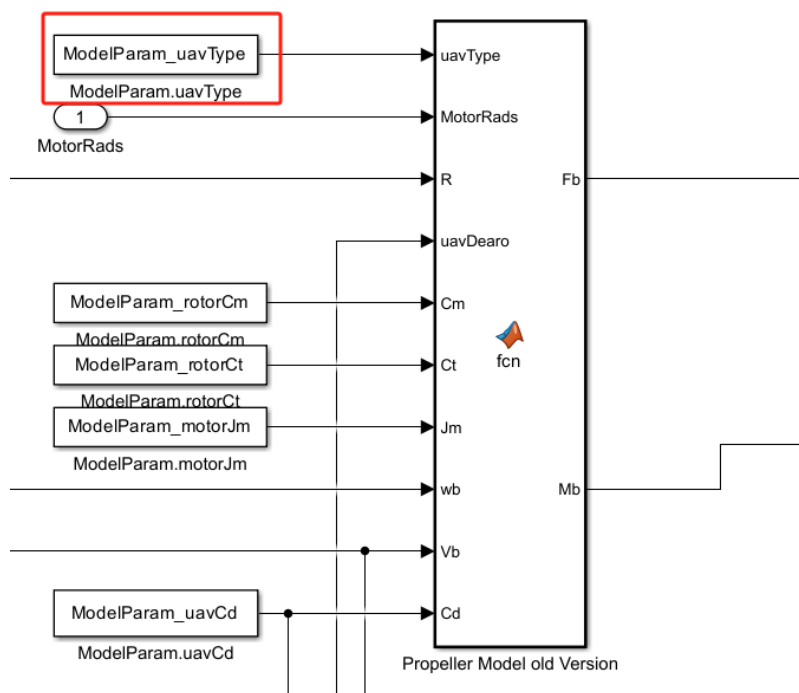
2.1. 模型参数介绍

1) 重要参数

DllSilNoPX4TempDemo_init.m 中定义了四旋翼综合模型的各种参数, 关键数据如下。

飞机的三维显示样式

`ModelParam_uavType = int16(3);` %这个参数决定了飞机的三维显示样式, 需要和 RflySim3D 的 XML 文件中的 `ClassID` 相匹配同时对于多旋翼飞行器的螺旋桨模型 (该模型在力和力矩模块中可见), 不同的机型, 要对应不同的 `ID` 以计算机架配置和力矩分配



飞机的初始位姿参数

`ModelInit_PosE=[0,0,0];` %用于设置飞机的初始位置, 对应了 CopterSim 上的 X 和 Y 初始值。Z 值利用 TerrainZ 实现了从 CopterSim 中读取当前地形高度数据, 使得飞机可以初始化在复杂地形的地表面 (例如 Grassland 地图)。

`ModelInit_AngEuler=[0,0,0];` %用于设定飞机的初始姿态。飞机姿态角的前两位 (俯仰和滚转角) 可以通过 ModelInit_AngEuler 参数来配置, 但是偏航角需要在 CopterSim 中配置。针对导弹等竖直起飞的飞行器, 需要设定合适的俯仰和滚转值。

QGC 中显示的地图坐标和高度原点 (在 RflySim3D 的 Cesium 大场景中能任意指定飞机在地球三维场景中的坐标)

```
ModelParam_GPSLatLong = [40.1540302 116.2593683];%飞机初始的纬度和精度，单位度。  
ModelParam_envAltitude = -50;%原点的海拔高度，竖直向下为正，高于海平面填负值，单位米。
```

执行器的初始参数

```
ModelInit_Inputs = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];% 十六维输入向量，定义电机 PWM 初始值，默认全 0，对固定翼和小车需要修改，因为它们的油门在初始状态处于最小值（-1），见“Motor Model”模块
```

故障接口参数

FaultInParams: 可通过外部消息动态改变的 32 维参数向量，在故障注入或者可变形的异构飞行器上有用，也可动态地调整传感器模型噪声等；与 **inSILInts** 和 **inSILFloats** 形成功能互补。

FaultParamAPI.FaultInParams = zeros(32,1); 定义了一个名为 **FaultInParams** 的 32 维向量，该向量被初始化为所有元素都为零。

FaultParamAPI.FaultInParams(3)=1; 将 **FaultInParams** 向量的第三个元素设置为 1。

2) 参数调用过程

DllSilNoPX4TempDemo.slx 是 DLL 模型生成的模板，模型启动运行（编译）时会调用 DllSilNoPX4TempDemo_init.m



DllSilNoPX4TempDemo_init.m 中包含了模型的参数信息，本脚本会在 DllSilNoPX4TempDemo.slx 编译（编译所需环境配置参考 [API.pdf 中的环境配置](#)）时被调用将参数载入 MATLAB 工作空间。

工作区	
名称 ▲	值
ext	'slx'
FaultParamAPI	1x1 struct
filepath	'E:\d2\4.RflySimMo...
HILGPS	1x1 Bus
InitFileName	'Exp2_MaxModelTe...
MavLinkGPS	1x1 Bus
MavLinkSensor	1x1 Bus
MavLinkStateQuat	1x1 Bus
MavVehileInfo	1x1 Bus
ModelInit_AngE...	[0,0,0]
ModelInit_Inputs	1x16 double
ModelInit_PosE	[0,0,0]
ModelInit_RateB	[0,0,0]
ModelInit_RPM	0
ModelInit_VelB	[0,0,0]
ModelParam_en...	-50
ModelParam_GP...	[40.1 1x1 double]
ModelParam_m...	842.1000
ModelParam_m...	1.2870e-04
ModelParam_m...	0.0500
ModelParam_m...	0.0214
ModelParam_m...	22.8300
ModelParam_rot...	2.7830e-07
ModelParam_rot...	1.6810e-05
ModelParam_ua...	[0.0035,0.0039,0.00...
ModelParam_ua...	0.0550
ModelParam_ua...	0.1200
ModelParam_uavJ	[0.0211,0,0;0,0.0219...
ModelParam_ua...	1.5150
ModelParam_ua...	4
ModelParam_ua...	0.2250
ModelParam_ua...	3
name	'Exp2_MaxModelTe...

GenerateModelDLLFile.p 是将 slx 模型转化为 DLL 模型文件的脚本，使用 RflySim 平台进行载具软硬件在环仿真时，需要将 DLL(windows 下)/SO(Linux 下)模型导入到 CopterSim，形成运动仿真模型，因此，在 Simulink 模型编译完成后，需要将模型对应的 C++ 文件打包成 DLL/SO 模型。

2.2. 输入信号[5]

最大模板的 12 个输入数据包括电机控制量、地形数据、

1) 电机数据 inPwms

输入接口 inPWMs, 16 维执行器控制量输入，已归一化到-1 到 1 尺度(通常电机是 0-1，舵机是 -1~1)，它的数据来自飞控回传的电机控制 MAVLink 消息

mavlink_hil_actuator_controls_t 的 controls，具体定义如下：

```
typedef struct __mavlink_hil_actuator_controls_t {  
    uint64_t time_usec; //时间戳，从开机后的时间，单位 ms  
    uint64_t flags; //标志位，用于显示当前的飞行状态  
    float controls[16]; //控制量，16 维电机的控制量，发送到模型中，驱动飞机飞行  
    uint8_t mode; // 模型，用于显示飞机当前的飞行模式和是否上锁等信息 }  
mavlink_hil_actuator_controls_t;
```

软件在环仿真时，电机控制指令从 PX4 SITL 控制器通过 TCP 4561 系列端口以 MAVLink 协议发送到运动仿真模型的 inPWMs 接口，而硬件在环仿真时，该指令是从飞控通过串口以 MAVLink 协议发送到运动仿真模型的 inPWMs 接口。

2) 地形高度 terrainZ

最大模板利用 TerrainZ 实现了从 CopterSim 中读取当前地形高度数据，使得飞机可以初始化在复杂地形的地表面（例如 RflySim3D 中的 Grassland 地图）。

3) inSILInts

8 维 Int32 型输入，通过 UDP 协议获取，来自 30100++2 系列端口号，软硬件在环仿真时，可通过该端口向模型输入一些量；同时，该接口是实现综合模型的关键接口。

4) inSILFloats

20 维 float 型输入，通过 UDP 协议获取，来自 30100++2 系列端口号，软硬件在环仿真时，可通过该端口向模型输入一些量；同时，该接口是实现综合模型的关键接口。

5) 碰撞数据 inFloatsCollision

利用 inFloatsCollision 实现了一个简单地物理引擎，可以根据 RflySim3D 回传的四周距离数据，实现碰到障碍物的回弹、碰到其他飞机便坠毁等功能

6) 飞控状态量(uORB 数据)输入 inCopterData

32 维，其中后 8 维接收 PX4 消息，数据来自 uORB msg rfly_px4.control[0:7]。

7) inFromUE

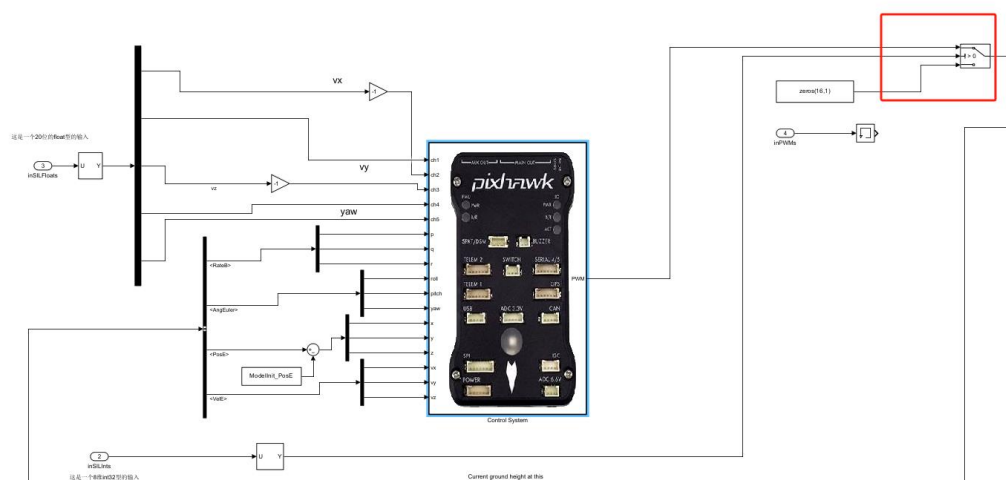
6 维 double 型数据，来自三维引擎（Rflsim3D/RflySimUE5），可用于实现地面交互、碰撞引擎等需要与三维引擎进行数据交互的相关功能。

8) inCtrlExt1~5

和 inSILInts 和 inSILFloats 类似，用于接收局域网内的故障注入消息

2.3. 控制器模块

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 0，没有真正启用遥控器。



表格 1 模型参数对照表

油门到电机稳态转速曲线零点	ω_b	ModelParam.motorWb	22.83(rad/s)
电机螺旋桨转动惯量	J_{RP}	ModelParam.motorJm	0.0001287 (kg/m^2)
电机响应时间常数	T_m	ModelParam.motorT	0.0214(s)
阻力系数	C_d	ModelParam.uavCd	0.055(N/(m/s)^2)
阻尼力矩系数	C_{dm}	ModelParam.uavCCm	[0.0035 0.0039 0.0034] (N/(rad/s)^2)

1) Motor Model 电机模块

在该模块中输入为 PWM 值（通过 inPWMs 接口获取），经过各电机的非线性动力学模型后得到各电机转速，该模块的输出分别为输入给力和力矩模型的电机转速（弧度每秒）；输入给 UE 的电机转速（转每分）

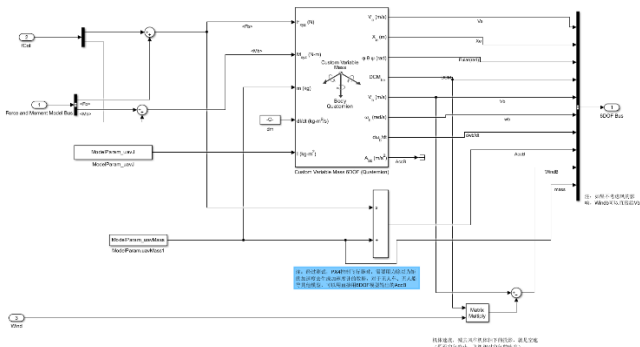
2) Force and Moment Model 力和力矩模块

该模块输入为电机转速 MotorRads、飞机运动学姿态 6DOF 和地形高度输入 TerrainZ，输出为多旋翼合力、合力矩 Force and Moment Model Bus。

3) 6DOF 六自由度刚体运动学模块

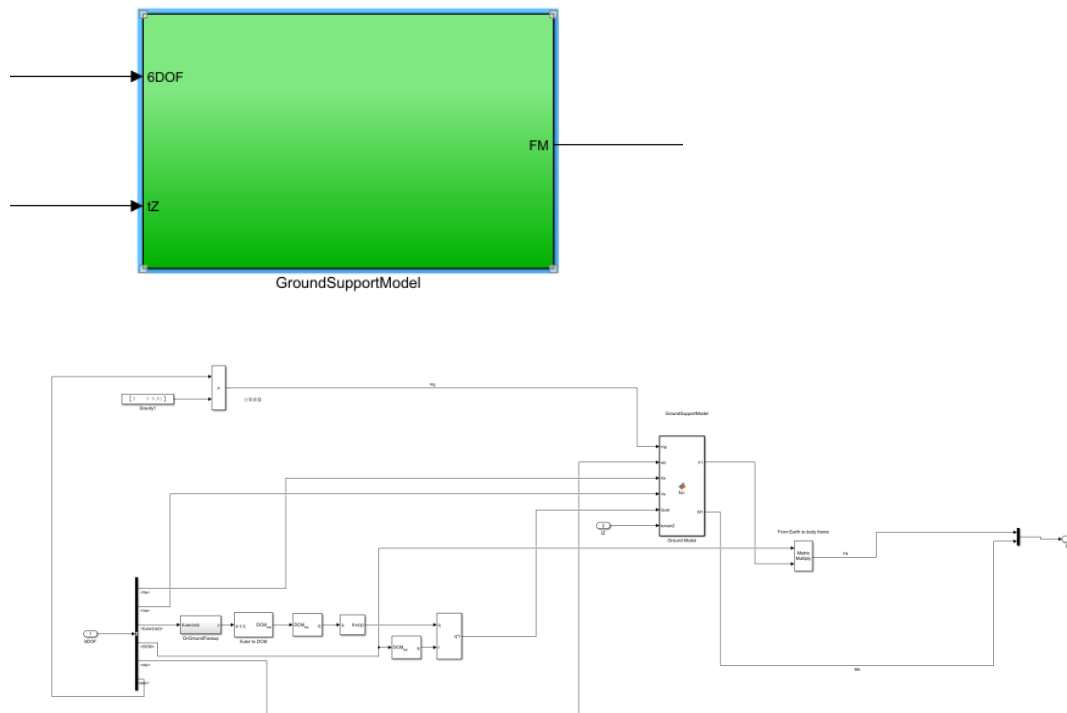
用于描述无人机在空中运动时的姿态和位置变化。考虑了无人机在三个坐标轴上的旋转运动（俯仰、横滚和偏航）以及机体与地球坐标系上的平移运动（前后、左右和上下）。

还可以根据实际需求对模型进行扩展，考虑更多的因素，如飞行器的非线性特性、气动力和惯性矩等。



4) GroundSupportModel 地面支撑模块

GroundSupportModel 地面支撑模块实际上是 [PhysicalCollisionModel 碰撞检测模块](#) 的一个子模块，这里将所有物体简化为较为简单的基本几何体（例如圆柱体或者长方体）来计算其与地面之间的物理接触受力。



5) 3DOutput 三维显示模块

该模块会将 `***_init.m` 中的 `ModelParam_uavType`（三维显示 ID）、来自电机模型的 `ActuatorToUE` 以及来自 6DOF 模型的 6DOF Bus 的位置、速度、姿态和加速度等输出为 `MavVehile3DInfo`，并按协议对输入信息进行数据打包后通过该接口将数据发送至三维引擎

2.5. 输出信号

1) MavVehile3Dinfo (真实仿真数据输出)

模型发送给飞控的各种传感器数据的集合，对应了 MAVLink 的 `mavlink_hil_sensor_t` 消息。输出信号中包括了加速度传感器的加速度值、陀螺仪传感器的角速度值、磁罗盘传感器的磁场值，气压和空速传感器的气压值等。

```
struct SOut2Simulator {
    int copterID; //飞机 ID, 用于区分局域网内不同飞机
    int vehicleType; //飞机样式, 区分同种飞机 (如四旋翼) 下的不同样式 (例如, 大疆、AR.Drone)
    double runnedTime; //时间戳, 当前时刻的时间, 单位毫秒
    float VelE[3]; //速度向量, 地球坐标系的 xyz 速度 (z 向下为正), 单位 m/s
    float PosE[3]; //位置向量, 地球坐标系下的 xyz 方向 (z 向下为正, 单位 m, 以起飞点为坐标原点
    float AngEuler[3]; //姿态角, 飞机的欧拉角, 定义于机体坐标系, 单位弧度
    float AngQuatern[4]; //四元数, 飞机姿态的四元数, 定义于机体坐标系
    float MotorRPMS[8]; //电机转速, 飞机的各个旋翼转速, 单位转每分
```

```
float AccB[3]; //加速度，飞机的运动加速度，单位 m/s^2
float RateB[3]; //角速度，飞机的转动角速度，单位 rad/s
double PosGPS[3]; //GPS 坐标，飞机的经纬高坐标，单位度、度、米
};
```

3. 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速度控制综合模型运动。

4. 文件目录

文件夹/文件名称		说明
icon	MavLinkStruct.mat	MAVLink 结构体数据文件。
	pixhawk.png	Pixhawk 硬件图片。
	readme.pdf	机架类型修改说明文件。
	F450.png	F450 飞机模型图片。
DllSilNoPX4TempDemo.slx		四旋翼综合模型，控制器为速度控制
CopterSILVelCtrlSender.slx		外部控制文件
DllSilNoPX4TempDemo.bat		四旋翼综合模型启动脚本
DllSilNoPX4TempDemo.dll		四旋翼综合模型动态链接库，由 CopterSILVelCtrl.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
Init_control.m		控制器及模型初始化参数文件
DllSilNoPX4TempDemo_init		模型初始化参数文件。

5. 运行环境

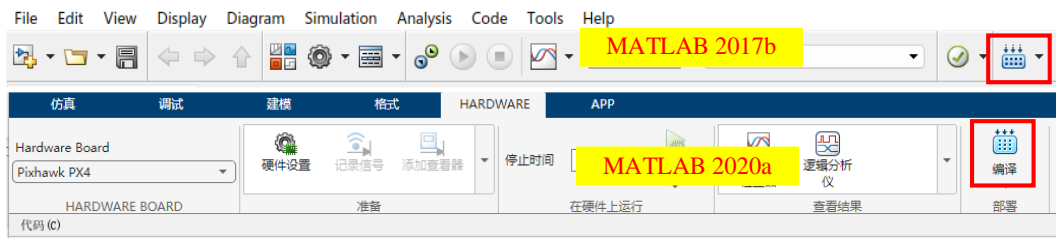
序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台收费版	\	\
3	MATLAB 2017B 及以上	\	\

①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6. 实验步骤

Step 1:

打开 MATLAB 软件，在 MATLAB 中打开 DllSilNoPX4TempDemo.slx 文件，在 Simulink 中，点击编译命令。

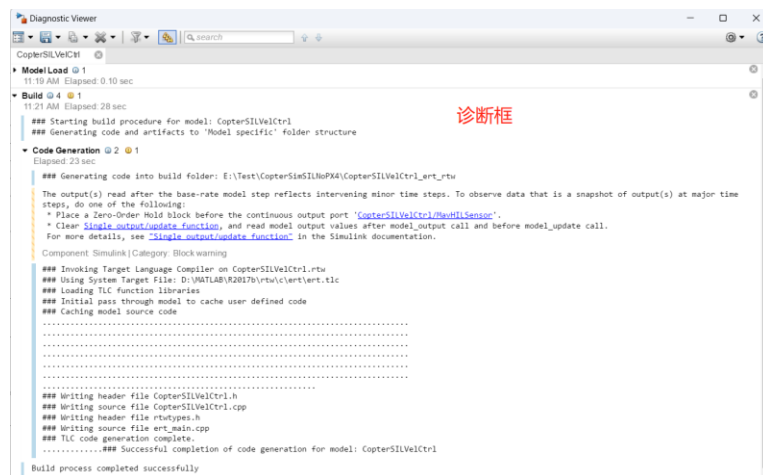


注意事项：与常规动力学模型先比，综合模型特点如下：

综合模型与常规动力学模型仿真方式不同，常规模型仿真方式为软硬件在环仿真方式，其通过 PX4 官方控制器控制动力学模型，由 QGC 发送控制指令；而综合模型中包含了控制器，通过 inSILInts 和 inSILFloats 的端口外部控制从而实现顶层控制，与 QGC 无关只在 RflySim3D 中进行渲染显示。

Step 2:

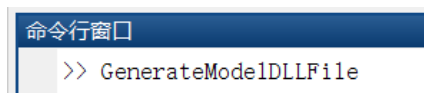
在 Simulink 的下方点击 View diagnostics 指令,即可弹出诊断对话框,可查看编译过程。在诊断框中弹出 Build process completed successfully, 即表示编译成功。



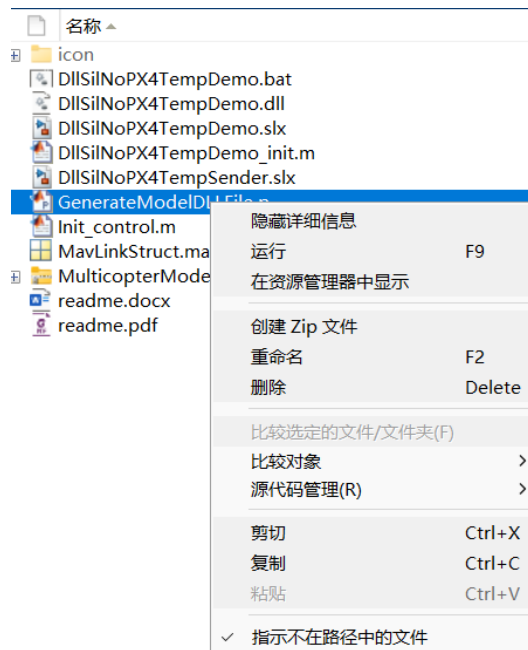
在本文件夹中会生成一个名为: MulticopterModel.zip 的压缩包文件,用于下一步生成.dll 文件。

Step 3:

在 MATLAB 中,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 DllSilNoPX4TempDemo.dll。

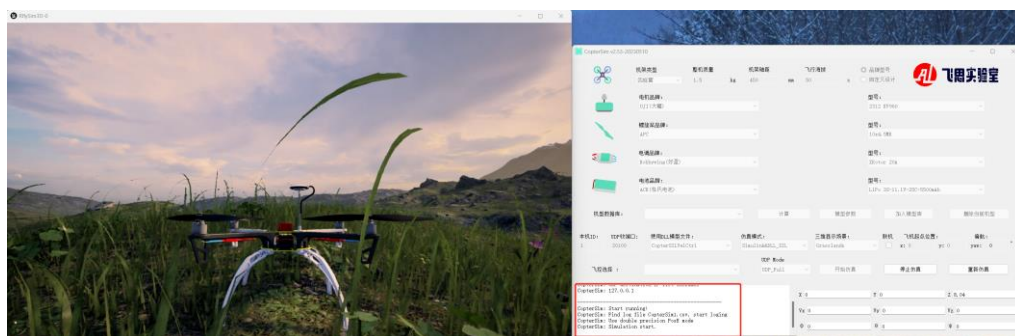
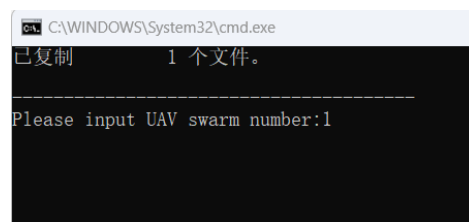


或



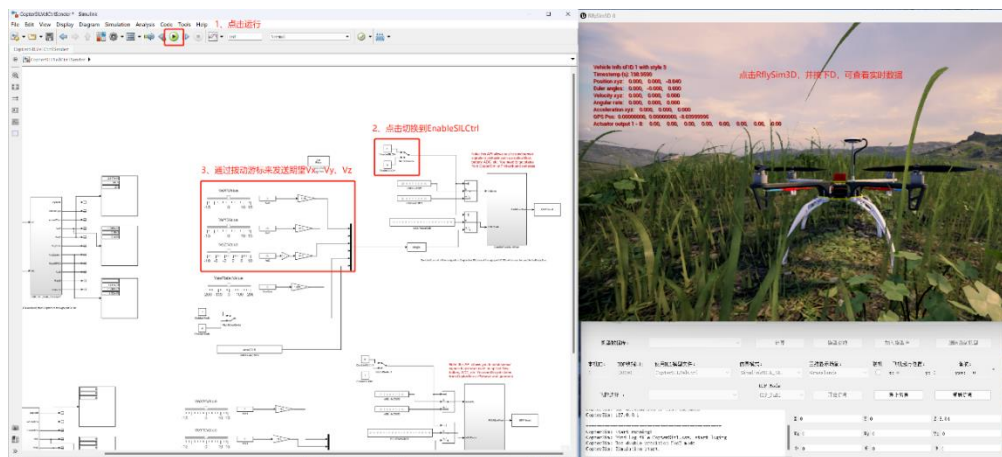
Step 4:

右键点击 DllSilNoPX4TempDemo.bat 并以管理员身份运行，输入 1，启动 1 架四旋翼综合模型的软件在环仿真。



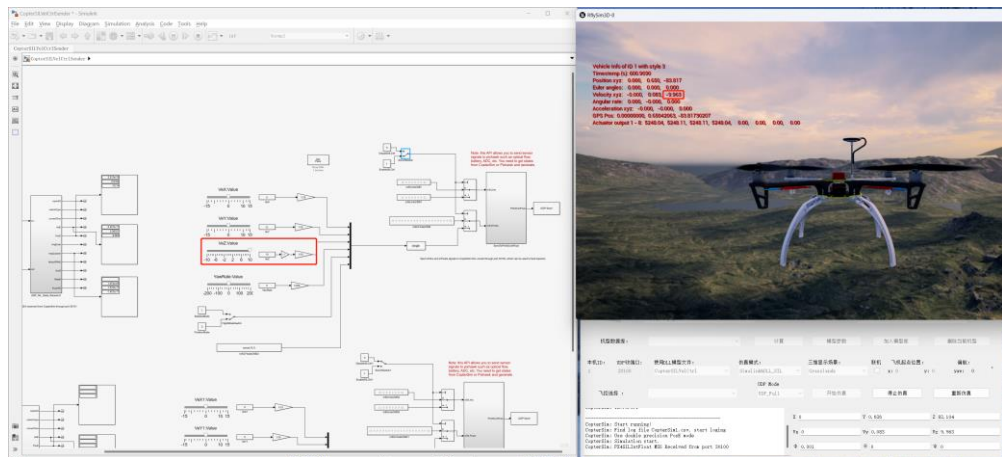
Step 5:

右键打开 DllSilNoPX4TempSender.slx，运行该文件，点击 2 处将使能标志切换到 EnableSILCtrl。



Step 6:

拖动滑杆发送期望的 V_x 、 V_y 、 V_z ，观察 RflySim3D 中四旋翼的实际速度是否与期望速度一致。此处发送期望 V_z 为-10，可看到四旋翼 V_z 为-9.963（坐标系为 NED 坐标系，向下为正）。



7. 参考资料

- [1]. [API.pdf](#) 中 DLL/SO 模型与通信接口的重要参数部分。
- [2]. [API.pdf](#) 中的环境配置
- [3]. [API.pdf](#) 中的 Simulink 建模模板介绍
- [4]. [API.pdf](#) 中 DLL/SO 模型与通信接口的数据协议部分
- [5]. [API.pdf](#) 中综合模型部分
- [6].

8. 常见问题

Q1: 若在进行 Step3 时，出现如下报错。

```
Found output signal MavHILGPS
Found output signal MavVehicle3DInfo
Found input signal inPWMs
Found input parameter ModelInit_Inputs
错误使用 mex
未找到支持的编译器或 SDK。您可以安装免费提供的 MinGW-w64 C/C++ 编译器；请参阅安装 MinGW-w64 编译器。有关更多：
间 http://www.mathworks.com/support/compilers。
出错 GenerateModelDLLFile
```

A1：请自行下载对应版本的 Microsoft Visual 软件(如：MATLAB 为 2017B，则需要安装 Microsoft Visual 2017)。安装完成之后在 MATLAB 的命令行窗口输入：

```
mex -setup
```

```
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2017 (C)' 以进行 C 语言编译。
警告：MATLAB C 和 Fortran API 已更改，现可支持
    包含 2^32-1 个以上元素的 MATLAB 变量。您需要
    更新代码以利用新的 API。
    您可以在以下网址找到更多的相关信息：
    http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html。

要选择不同的语言，请从以下选项中选择一种命令：
mex -setup C++
mex -setup FORTRAN
```

若出现上述显示，即安装完成。