

- 1.实验名称及目的
 - 1.1 实验名称
 - 1.2 实验目的
 - 1.3 关键知识点
 - 1.3.1 绳子动力学建模概述
 - 1.3.2 绳子动力学建模详细思路
 - 物理建模假设
 - 数学模型基础
 - 力学模型推导
 - 边界条件处理
 - 数值积分方法
 - 力和力矩输出
 - 代码实现架构
 - 建模方法的优势与局限
 - 2.实验效果
 - 3.文件目录
 - 4.运行环境
 - 4.1 软件要求
 - 4.2 硬件要求
 - 5.实验步骤
 - 5.1. 必做实验：DLL模型生成
 - Step 0：基于建模模板搭建绳索动力学模型
 - Step 1：编译模型
 - Step 2：生成DLL文件
 - 5.2. 必做实验：软件在环仿真
 - Step 1：启动仿真
 - Step 2：等待初始化完成
 - Step 3：四旋翼起飞
 - 5.3. 选做实验：硬件在环仿真
 - Step 1：连接飞控
 - Step 2：设置硬件在环机架
 - Step 3：配置硬件在环参数
 - Step 4：启动仿真
 - Step 5：仿真过程
-

1. 实验名称及目的

1.1 实验名称

系留无人机建模与仿真

1.2 实验目的

在 RflySim/CopterSim 平台上验证将无人机通过绳索（或系绳）与地面锚点连接后的动力学影响与控制行为；演示如何把带绳索模型的 Simulink 模型生成 DLL 并在软件/硬件在环（SITL/HITL）中运行。

1.3 关键知识点

1.3.1 绳子动力学建模概述

绳子建模：将绳子视为 N 段线性弹簧-阻尼单元，节点间仅承受张力（无压缩），使用有限元离散化思想。

核心功能说明：

- 输入：
 - `drone_state` : 6×1 , $[x;y;z; vx;vy;vz]$ (单位 $m / m/s$)
 - `rope_params` : 8×1 , $[N, rest_length, stiffness, damping, rope_mass, anchor_x, anchor_y, anchor_z]$
- 输出：
 - `force_drone` : 在无人机挂点处的合力 (3×1 , N)
 - `moment_drone` : 作用在机体上由挂点产生的力矩 (3×1 , $N \cdot m$)

实现要点：

- 用 N 段将绳索离散为 $N+1$ 个节点（节点 1 为地面锚点，节点 $N+1$ 为无人机挂点）。
- 每段采用线性弹簧-阻尼模型，仅受张力： $T = \max(kextension + cvel_along, 0)$ 。
- 使用 persistent 状态保存上一步节点位置/速度（适合集成到固定步长仿真循环或 Simulink S-Function）。
- 当前演示代码使用显式 Euler 步进 ($dt = 0.001 s$)，对刚性系统需谨慎。

1.3.2 绳子动力学建模详细思路

物理建模假设

系留绳索建模基于以下物理假设：

1. 材料特性假设

- 绳索材料为线弹性，满足胡克定律
- 绳索只能承受拉力，不能承受压力（绳索松弛时张力为零）
- 绳索质量均匀分布
- 忽略绳索的弯曲刚度和扭转效应

2. 几何假设

- 绳索可发生大变形，但应变较小
- 节点之间用直线单元连接
- 绳索横截面积相对于长度可忽略

3. 动力学假设

- 采用集中质量法，质量集中在节点上
- 阻尼为粘性阻尼，与相对速度成正比
- 忽略空气阻力的影响

数学模型基础

系留绳索动力学建模采用有限元方法，将连续的绳索离散化为多个节点和单元：

1. 空间离散化

```
% N段绳索 => N+1个节点
num_nodes = N + 1;
% 节点1: 地面锚点 (固定边界条件)
% 节点2~N: 绳索中间节点 (自由度3*(N-1))
% 节点N+1: 无人机挂点 (边界条件, 由无人机状态更新)
```

2. 单元特性

- 每个单元的自然长度: $l_0 = \frac{L_{total}}{N}$
- 单元弹性系数: $k_{elem} = k_{total} \times N$ (串联弹簧原理)
- 单元阻尼系数: $c_{elem} = c_{total} \times N$
- 节点质量: $m_{node} = \frac{m_{rope}}{N-1}$ (除去固定端)

力学模型推导

张力计算 (仅拉伸):

```

% 计算单元当前状态
vec = pos(:, node2) - pos(:, node1);           % 单元向量
current_l = norm(vec);                          % 当前长度
if current_l > 0
    unit_vec = vec / current_l;                 % 单位方向向量
else
    unit_vec = zeros(3, 1);                     % 避免除零
end

% 弹性力: 基于胡克定律, 仅当拉伸时产生张力
extension = current_l - l0_elem;                % 伸长量
elastic_force = k_elem * max(extension, 0);     % 仅拉伸

% 阻尼力: 沿单元方向的相对速度产生的阻尼
rel_vel = vel(:, node2) - vel(:, node1);       % 相对速度向量
vel_along = dot(rel_vel, unit_vec);            % 轴向分量
damping_force = c_elem * vel_along;            % 阻尼力

% 总张力 (确保为正, 绳索不能推)
T_mag = max(elastic_force + damping_force, 0);

```

节点动力学方程:

根据牛顿第二定律, 每个节点的运动方程为:

$$m_i \ddot{\mathbf{r}}_i = \sum_{j \in \text{连接单元}} \mathbf{T}_{ij} + m_i \mathbf{g}$$

其中:

- \mathbf{r}_i : 第*i*个节点的位置向量 $[x_i, y_i, z_i]^T$
- \mathbf{T}_{ij} : 连接节点*i*和*j*的单元张力向量
- $\mathbf{g} = [0, 0, -9.81]^T$: 重力加速度向量
- m_i : 第*i*个节点的质量

力的分配原理:

```

% 对于每个单元, 张力作用在两个端节点上
% 节点1受到指向节点2的拉力
force_net(:, node1) = force_net(:, node1) + T_mag * unit_vec;
% 节点2受到指向节点1的拉力 (牛顿第三定律)
force_net(:, node2) = force_net(:, node2) - T_mag * unit_vec;

```

边界条件处理

1. 地面锚点 (节点1):

```

% Dirichlet边界条件: 固定位置
pos(:, 1) = anchor_pos;           % 位置约束
vel(:, 1) = 0;                     % 速度约束
accel(:, 1) = 0;                   % 加速度约束

```

2. 无人机挂点 (节点N+1):

```

% 由无人机动力学提供边界条件
drone_pos = drone_state(1:3); % 无人机位置
drone_vel = drone_state(4:6); % 无人机速度

% 更新绳索末端节点状态
pos(:, end) = drone_pos;
vel(:, end) = drone_vel;

```

数值积分方法

采用显式欧拉法进行时间积分：

```

% 状态向量定义
% y = [pos(:); vel(:)] 其中pos为3x(N+1), vel为3x(N+1)
% 状态向量长度: 6x(N+1)

% 导数计算
dydt = ode_fun(y, N, l0_elem, k_elem, c_elem, mass_vec, anchor_pos);

% 前向欧拉积分
dt = 0.001; % 积分步长 (需要足够小保证稳定性)
y = y + dydt * dt;

```

时间步长选择原则：

为保证数值稳定性，时间步长需满足：

$$\Delta t < \frac{2}{\omega_{max}}$$

其中 $\omega_{max} = \sqrt{\frac{k_{elem}}{m_{node}}}$ 是系统的最高固有频率。

注意事项：

- 显式欧拉法对刚性系统不稳定，需要足够小的时间步长
- 对于高刚度绳索，建议使用隐式积分方法（如后向欧拉法）
- 可考虑使用Runge-Kutta等阶方法提高精度

力和力矩输出

作用于无人机的力：

```

% 绳索对无人机的作用力 (最后一个节点的合力)
force_drone = force_net(:, end);

```

作用于无人机的力矩：

```
% 挂点相对于无人机质心的位置（机体坐标系）
attach_point = [0.1; 0; -0.05]; % [前向; 右向; 下向] (m)

% 力矩计算:  $M = r \times F$ 
moment_drone = cross(attach_point, force_drone);
```

力矩的物理意义:

- 绳索张力会在无人机质心处产生附加力矩
- 该力矩会影响无人机的姿态控制
- 挂点位置设计需要考虑力矩对飞行稳定性的影响

代码实现架构

```
rope_model()
├── 参数解析和安全限制
│   ├── 限制单元数量避免计算爆炸
│   └── 参数有效性检查
├── 状态初始化 (persistent 变量)
│   ├── 节点位置初始化 (沿锚点到无人机方向分布)
│   ├── 节点速度初始化 (零初值)
│   └── 状态向量组装
├── 实时状态更新
│   ├── 从无人机状态更新边界条件
│   └── 数值积分更新内部节点状态
├── 力计算模块 compute_forces()
│   ├── 遍历所有单元
│   ├── 计算各单元张力
│   ├── 累加各节点受力
│   └── 添加重力作用
└── 输出计算
    ├── 提取无人机节点受力
    ├── 计算挂点力矩
    └── 格式化输出向量
```

建模方法的优势与局限

优势:

- **物理准确性:** 基于连续介质力学和有限元理论
- **数值稳定性:** 适当的离散化和张力限制保证计算稳定
- **计算效率:** 线性时间复杂度 $O(N)$, 适合实时仿真
- **易于扩展:** 可方便添加非线性材料、空气阻力、多段绳索等特性
- **参数可调:** 通过调整 N 值平衡精度与计算成本

局限性:

- **线性假设:** 未考虑绳索的几何非线性和材料非线性
- **忽略弯曲:** 假设绳索无弯曲刚度
- **显式积分:** 对于刚性系统可能不稳定
- **简化阻尼:** 采用简单的粘性阻尼模型

这种建模方法在无人机系留仿真中能够提供足够的精度，同时保持良好的计算效率，适合工程应用。

2. 实验效果



3. 文件目录

例程目录：[\[安装目](#)

[录\]\RflySimAPIs\4.RflySimModel\3.CustExps\le1_MultiModelCtrl\1.TetheredModelCtrl](#)

文件/目录	说明
TetheredModelCtrl.slx	Simulink 原始模型（系留模型整合在 ground model2 模块）
TetheredModelCtrl.dll	已生成的 DLL，可直接由 CopterSim 加载（位于仓库根）
TetheredModelCtrl_init.m	模型参数初始化脚本，Simulink 编译/运行时会调用
GenerateModelDLLFile.p	生成 DLL 的脚本

文件/目录	说明
TetheredModelCtrlSITL.bat	启动软件在环（SITL）仿真的批处理脚本
TetheredModelCtrlHITL.bat	启动硬件在环（HITL）仿真的批处理脚本

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③。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmu-v6x_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://rflysim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑① 1台；Pixhawk 6X或其它飞控② 1台；数据线 1台。

①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

5.1. 必做实验：DLL模型生成

Step 0：基于建模模板搭建绳索动力学模型

相对于例程 4.RflySimModel\1.BasicExps\e0_MinModelTemp 中提供的最小建模模板

Exp1_MinModelTemp.slx，本例的TetheredModelCtrl.slx主要在地面模块GroundSupportModel中增加了绳索牵引的模型（Ground Model2）



Step 1: 编译模型

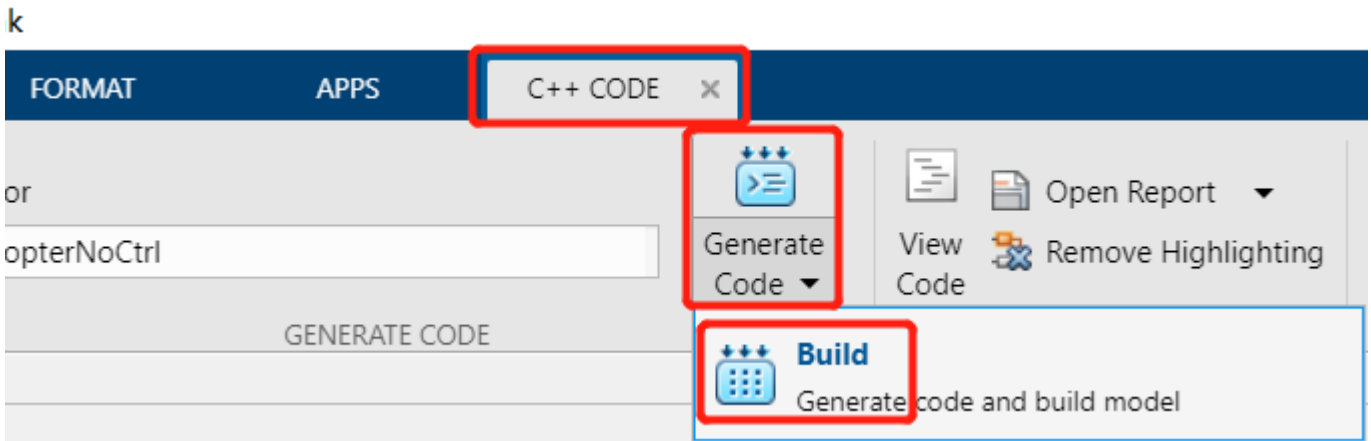
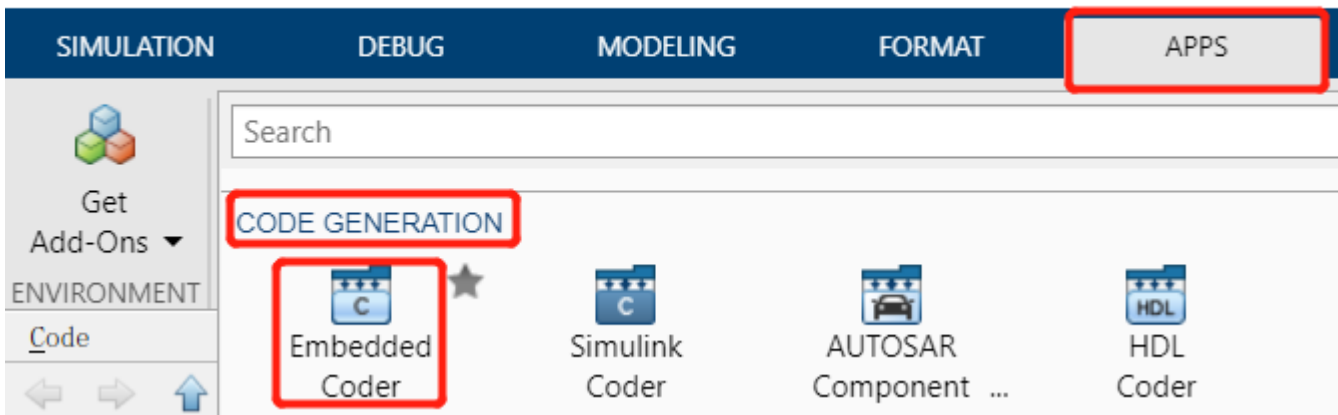
在Matlab中打开“TetheredModelCtrl.slx”Simulink 文件，点击Build Model按钮编译生成代码。

名称	大小	修改日期
readmemd		10/11/2025 6:51 PM
GenerateModelDLLFile.p	7 KB	9/28/2025 10:52 AM
MulticopterModel.zip	140 KB	9/28/2025 10:52 AM
RflySimExpsRes.md	6 KB	9/24/2025 1:49 AM
TetheredModelCtrl.dll	223 KB	9/28/2025 10:52 AM
TetheredModelCtrlHITL.bat	6 KB	9/28/2025 10:52 AM
TetheredModelCtrl_init.m	3 KB	9/28/2025 10:52 AM
TetheredModelCtrlSITL.bat	6 KB	9/28/2025 10:52 AM
TetheredModelCtrl.slx	140 KB	9/28/2025 10:52 AM

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 –EmbeddedCoder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。



Step 2: 生成DLL文件

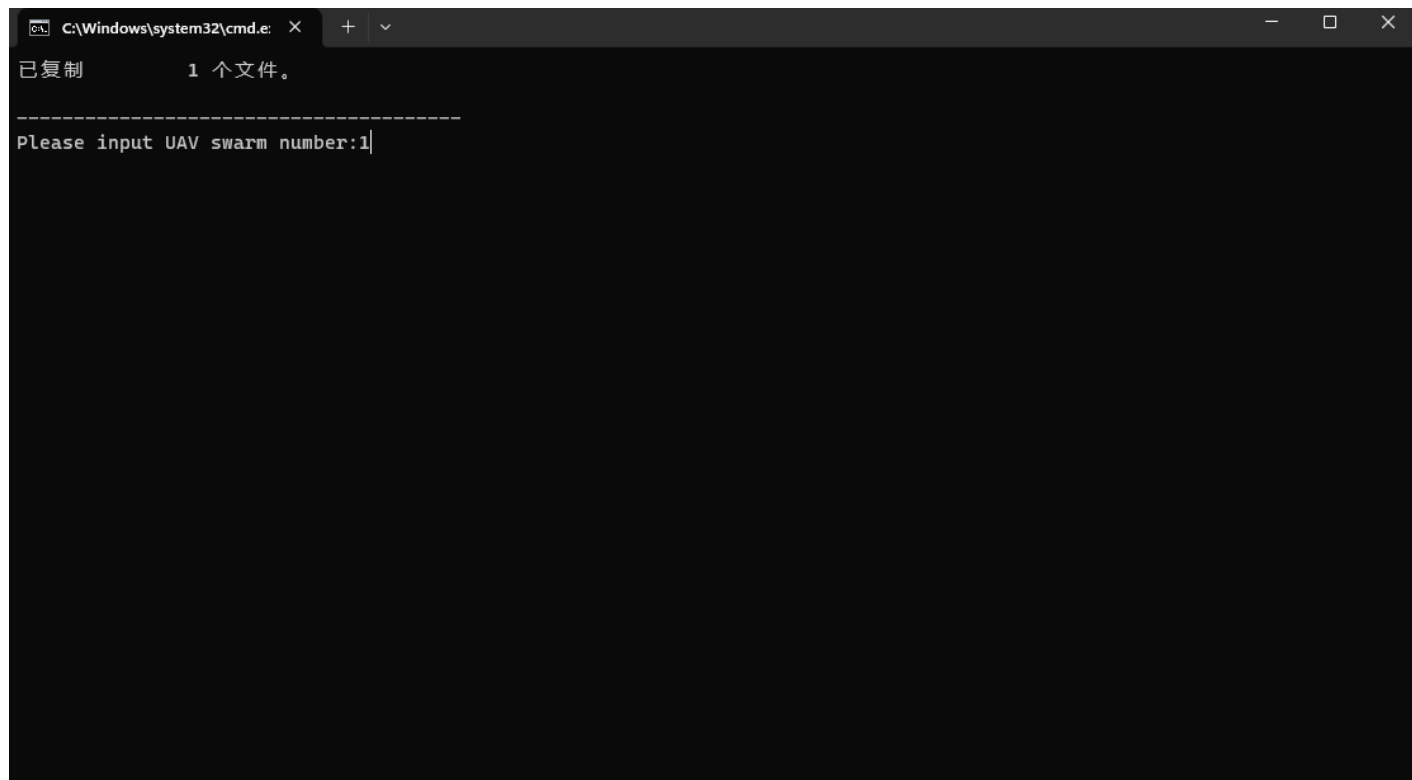
编译完成后，在 matlab 中右键“GenerateModelDLLFile.p”文件，点击运行，生成 DLL 文件。



5.2. 必做实验：软件在环仿真

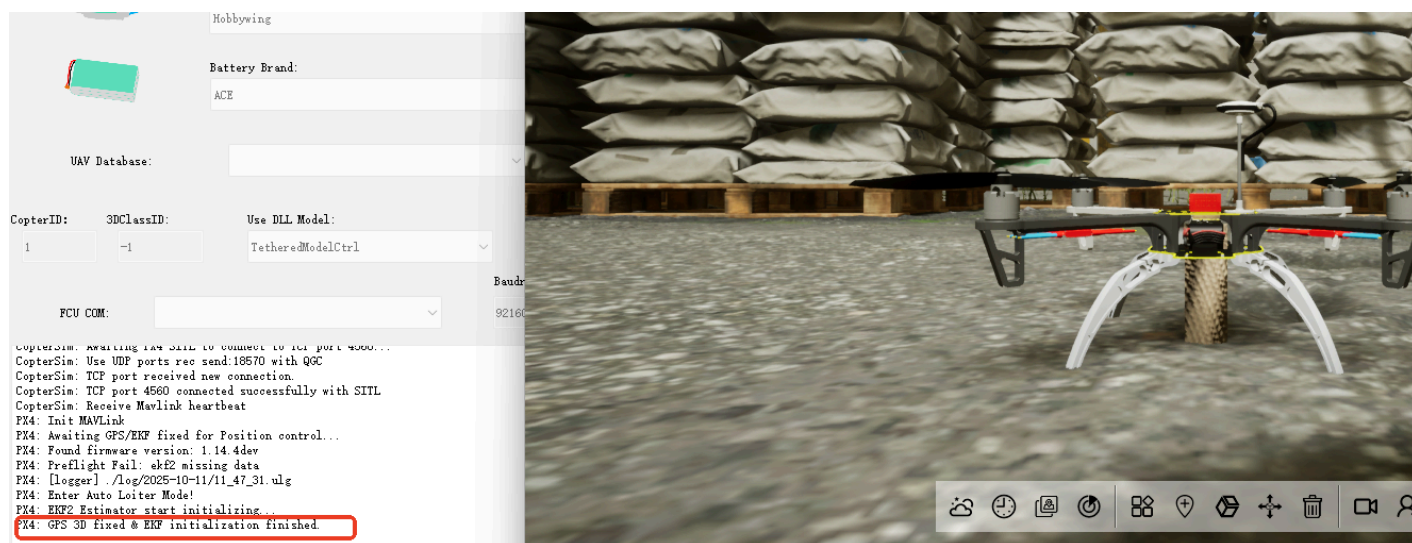
Step 1：启动仿真

双击运行“TetheredModelCtrlSITL.bat”批处理文件，在弹出的终端窗口中输入1，启动一架飞机的软件在环仿真。



Step 2：等待初始化完成

等待 CopterSim 中显示连接上 RflySim3D。



Step 3：四旋翼起飞

通过QGC使四旋翼飞机起飞到空中：点击“起飞”，确认起飞高度后，滑动下方滑块解锁起飞。



5.3. 选做实验：硬件在环仿真












Step 1: 连接飞控

硬件在环需要准备一个飞控，如下图所示，将飞控通过USB线连接电脑，并确保完成硬件在环仿真配置。注意，本图使用Pixhawk6x飞控，其他飞控配置方法类似（推荐使用Pixhawk飞控）。



Step 2: 设置硬件在环机架

在 Rflytools 文件夹中打开 QGC 地面站。

 3DDisplay	2023/7/27 15:02	快捷方式	1 KB
 CopterSim	2023/7/27 15:02	快捷方式	1 KB
 FlightGear-F450	2023/7/27 15:02	快捷方式	2 KB
 HITLRun	2023/7/27 15:02	快捷方式	2 KB
 Python38Env	2023/7/27 15:02	快捷方式	2 KB
 QGroundControl	2023/7/27 15:02	快捷方式	1 KB
 RflySim3D	2023/7/27 15:02	快捷方式	1 KB
 RflySimAPIs	2023/7/27 15:02	快捷方式	1 KB
 RflySimUE5	2023/7/27 15:02	快捷方式	1 KB
 SITLRun	2023/7/27 15:02	快捷方式	2 KB
 Win10WSL	2023/7/27 15:02	快捷方式	2 KB

打开QGC地面站，机架设置为“Generic Quadcopter”，点击QGC右上角的“应用并重启”。

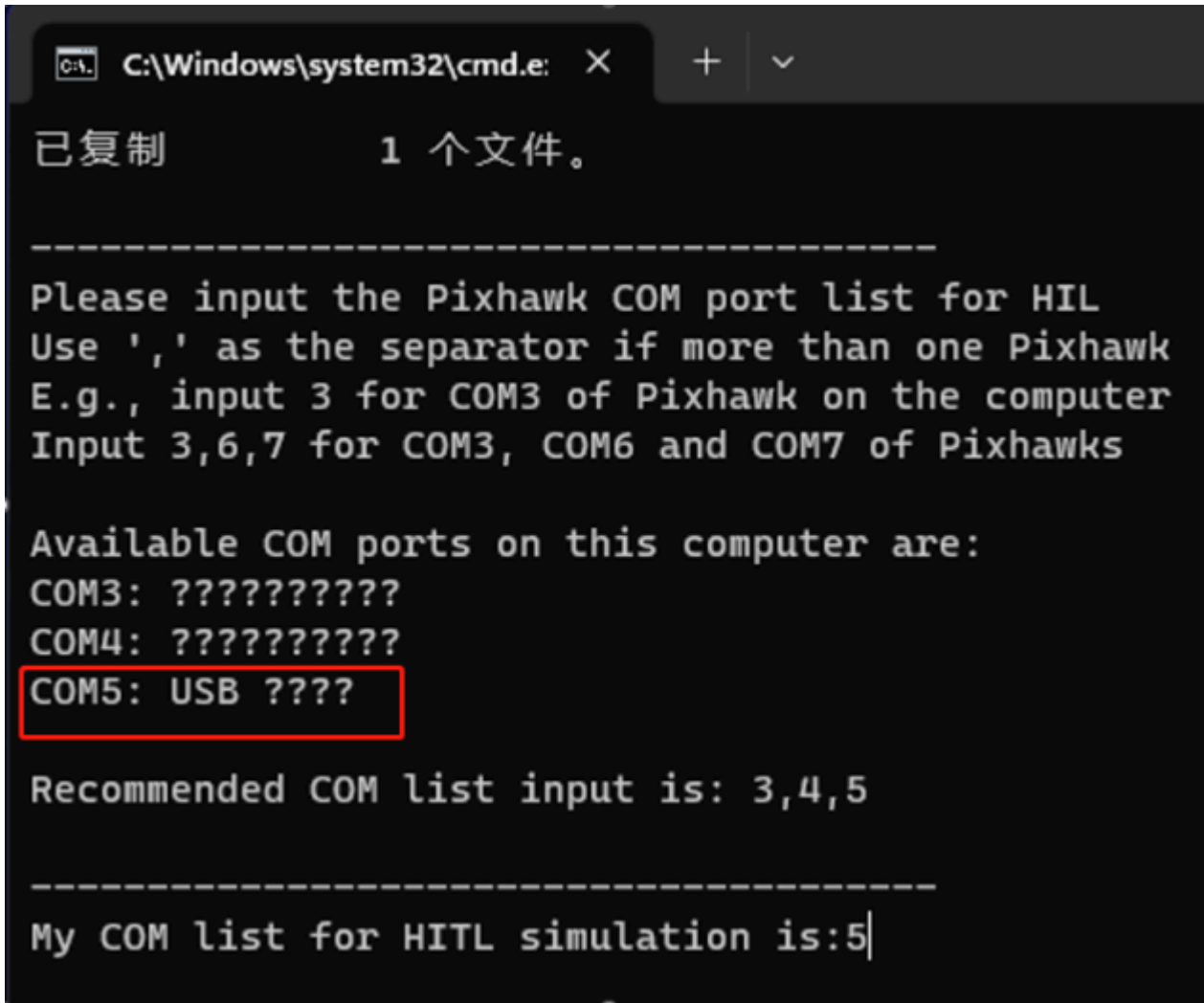


Step 3: 配置硬件在环参数

点击“安全”，设置硬件在环仿真为“HITL enabled”。

Step 4: 启动仿真

右键以管理员身份运行“TetheredModelCtrlHITL.bat”批处理文件，自动完成所有的配置，输入端口号启动一架飞机的硬件在环仿真。



```
C:\Windows\system32\cmd.e: X + v
已复制      1 个文件。

-----
Please input the Pixhawk COM port list for HIL
Use ', ' as the separator if more than one Pixhawk
E.g., input 3 for COM3 of Pixhawk on the computer
Input 3,6,7 for COM3, COM6 and COM7 of Pixhawks

Available COM ports on this computer are:
COM3: ??????????
COM4: ??????????
COM5: USB ????'
Recommended COM list input is: 3,4,5

-----
My COM list for HITL simulation is:5|
```

Step 5: 仿真过程

之后测试步骤与软件在环仿真的Step 2到Step3相同，运行之后观察四旋翼能否按照指令飞行