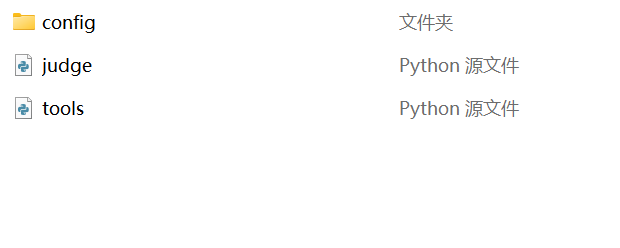
2024年无人机仿真任务验收规则

# 四、裁判机介绍

提供了一个judgev3.zip包,其中包含一个config文件夹和两个python文件：



同学们需要将judge.py 和 tools.py 文件一起放到hector\_quadrotor\_demo/ scripts文件夹下，config文件夹整个放到hector\_quadrotor\_demo/文件夹下

以配置初赛使用的裁判机。裁判机共有以下三个功能：

·发出比赛开始信号，并开始比赛计时；  
 ·检测无人机降落，并停止计时，计算比赛所用时间；  
 ·接收无人机发布的目标检测结果（相关格式及topic见附录A）。

以课程中提供的简单仿真例程为例。可以在

roslaunch hector\_quadrotor\_demo start\_nics\_world.launch

roslaunch hector\_quadrotor\_demo put\_robot\_in\_world.launch

后，打开一个新终端，运行命令：

rosrun hector\_quadrotor\_demo judge.py

即可打开一个裁判机窗口。可以按照该窗口的指示，尝试使用裁判机。

目标检测结果的ground truth存储在hector\_quadrotor\_demo/config/target.yaml中。更换赛场环境时需要修改该文件。

# 四、裁判机使用方法

1. **比赛开始判别条件：**

无人机起飞后高度大于0.3m。

1. **目标检测判别条件：**

参赛者通过/tello/target\_result 发送五位字符串，裁判机接受后自动判别识别成功数量。

/tello/target\_result接收的类型为std\_msgs/String。无人机向此topic发布目标识别、检测的结果。发布信息应为长度为5的字符串，字符串的每一位代表对应位置的目标检测结果：红球为‘r’，黄球为‘y’，蓝球为‘b’，没有目标为‘e’。例如，检测到位置1为红球，位置2为黄球，位置3为蓝球，则正确结果应为“rybee”。

1. **比赛结束判别条件：**

参赛无人机飞行到目标区域内，并降落（高度小于0.2m）

**附录A：仿真环境中可能用到的ROS topic**

**A.1 由仿真环境发布**

A.1.1 /front\_cam/camera/image

类型为sensor\_msgs/Image。包含无人机前置摄像头的图像信息。

A.1.2 rostopic echo /ground\_truth/state

类型为nav\_msgs/Odometry。包含了无人机当前6D位姿信息。

**A.1 由参赛者环境发布**

A.2.1 /tello/target\_result

类型为std\_msgs/String。无人机向此topic发布目标识别、检测的结果。发布信息应为长度为5的字符串，字符串的每一位代表对应位置的目标检测结果：红球为‘r’，黄球为‘y’，蓝球为‘b’，没有目标为‘e’。例如，检测到位置1为红球，位置2为黄球，位置3为蓝球，则正确结果应为“rybee”。