**“无人机双机协同自主搜救任务”参赛选手手册**

组织单位、参赛对象、比赛日程等相关信息，详见第十九届“挑战杯”全国大学生课外学术科技作品竞赛官方网站。

1. **答题要求**
2. 挑战赛（仿真赛）
3. 作品形式：

**材料文档：**作品的录屏视频、总结报告（包含设计思路、使用说明等内容）、score1.bag日志文件。

**软件模块：**满足作品在指定仿真软件上的部署和运行的方案作品的源代码。

**作品考核：**

综合考察信息融合、信息决策、无人机控制领域知识，要求参赛选手具备信息获取、信息处理、信息融合与决策的能力。选手需要通过自主编写智能算法程序，实现在一定区域内，利用智能算法操控虚拟无人机自主完成任务。

**考核背景：**

无人机凭借其灵活性、低成本和高效率在侦察与监视、目标打击、反恐与排爆以及搜救任务等方面多个领域广泛应用。

在复杂地形的搜索任务中，由于结构复杂、视野受限以及存在坍塌风险，地面搜救人员面临较大的安全隐患和行动障碍。相比之下，无人机能够灵活穿梭于建筑之间，从空中快速获取全局视角，提升搜索效率并降低风险。

在双机协同的模式下，垂起固定翼无人机快速进行大范围扫描，发现可疑目标后将其坐标位置通过机间数据链传送给多旋翼无人机，多旋翼无人机可以抵近该目标进行识别，确认为搜救对象，并在识别后空投应急物品。整个搜救过程由人工智能算法自主控制，双机自主协同完成，不需要人为干预。

1. 参赛选手在挑战赛阶段需在指定的仿真软件上展开算法调试并完成方案设计；
2. 参赛选手需要提供具体的算法描述，并自行组织对软件设计进行合理性评估；
3. 参赛选手必须确保其作品的独立原创性，严禁抄袭或剽窃他人成果，并且必须遵守国家相关的知识产权保护规定。参赛选手不得侵犯任何第三方的知识产权或其他权益。如若发生知识产权纠纷，参赛者承担相应责任；
4. 参赛选手提交的材料原则上不予退还，请参赛选手自行保存底稿；

**评分规则：**

详见附件1

补充：作品已获得国际竞赛、国家级奖励和其他全国性竞赛获奖作品的，不在申报作品范围之列。

1. **任务概要**
2. 赛题：

挑战赛（仿真赛）构建了该区域的仿真环境，选手利用双机协同完成搜救任务。垂起无人机从起飞平台飞至居民区附近的GPS引导点，在该区域搜索发现并区分三位模拟人员，将相关信息发送给四旋翼无人机并返航。四旋翼无人机前往目标区域，对已知位置信息的2个目标(健康、危重)进行任务分配与路径规划，最后返回旋翼区降落完成整个任务。

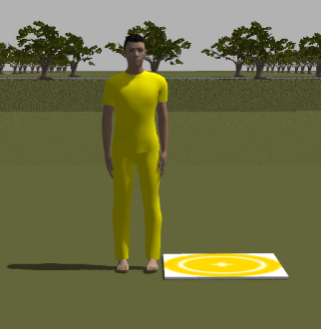
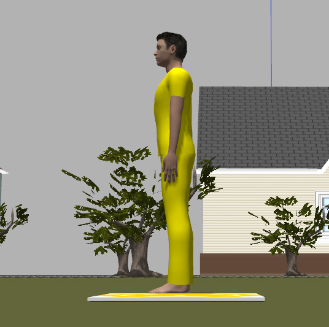
开始解题后，建议垂起无人机起飞至20米高度后，将垂起无人机切换为固定翼模式飞行。飞行过程中，垂起无人机需要避开两个限制范围（固定），根据GPS提供的位置信息飞行至GPS引导点后，开始搜索三个模拟人员。

GPS引导点处于两个居民区中心，在居民区中间区域内，有3个模拟人员：1个危重人员（穿红色衣服），1个轻伤人员（穿黄色衣服），1个健康人员（穿白色迷彩）。由于一些不可控因素，轻伤人员和健康人员的位置会不断发生变化，垂起无人机需要通过视觉识别并分辨目标，最终将3个模拟人员对应的降落平台中心坐标发送给四旋翼无人机后返航。

垂起无人机返回旋翼模式区域降落锁桨后，四旋翼无人机起飞，首先需飞往健康人员位置，到达健康人员位置后，发送该目标对应的降落平台中心坐标。随后飞往危重人员位置，到达危重人员位置后，发送该目标对应的降落平台中心坐标，最后返回旋翼模式区域降落。



危重人员示例图

轻伤人员示例图

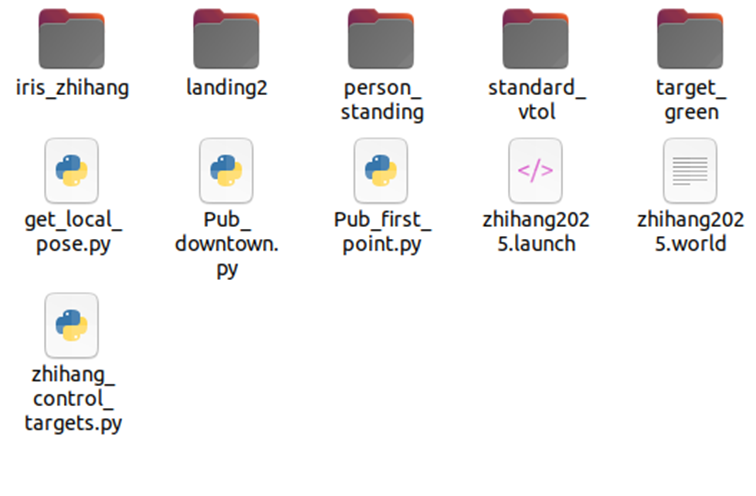
  

健康人员示例图

1. 解题

**选手可通过发送邮件（邮箱：18210263093@163.com），获得仿真环境安装教学视频和比赛赛题相关文件。**

比赛赛题相关文件内容如下：



请选手按照比赛官方教程配置好环境后将赛题文件夹中的文件放入相应位置:

（1）模型文件：standard\_vtol，iris\_zhihang，target\_green, landing2放置位置为：

~/PX4\_Firmware/Tools/sitl\_gazebo/models/ 若放置时提示文件夹内已经有同名文件，直接将同名文件替换即可。

（2）模拟人员模型文件：person\_standing 放置位置为：

~/.gazebo/models/ 若放置时提示文件夹内已经有同名文件，直接将同名文件替换即可。

（3）Launch文件：zhihang2025.launch放置位置为：

~/PX4\_Firmware/launch/ 若放置时提示文件夹内已经有同名文件，直接将同名文件替换即可。

（4）World文件：zhihang2025.world放置位置为:

~/PX4\_Firmware/Tools/sitl\_gazebo/worlds/ 若放置时提示文件夹内已经有同名文件，直接将同名文件替换即可。

（5）Gazebo位姿真值文件：get\_local\_pose.py放置位置为：

~/XTDrone/sensing/pose\_ground\_truth/ 若放置时提示文件夹内已经有同名文件，直接将同名文件替换即可。

（6）其他文件：统一保存在zhihang2025文件夹中，将zhihang2025文件夹放置为：

~/XTDrone/

* 注意：比赛官方提供的所有文件都不得随意更改，若更改则判定选手的成绩无效。

1. 操作步骤:
2. 启动仿真程序:

roslaunch px4 zhihang2025.launch



1. 运行通信脚本：

垂起无人机：

cd ~/XTDrone/communication/

python3 vtol\_communication.py standard\_vtol 0

旋翼无人机：

cd ~/XTDrone/communication/

python3 multirotor\_communication.py iris 0

1. 开启gazebo位姿真值：

垂起无人机：

cd ~/XTDrone/sensing/pose\_ground\_truth/

python3 get\_local\_pose.py standard\_vtol 1

旋翼无人机：

cd ~/XTDrone/sensing/pose\_ground\_truth/

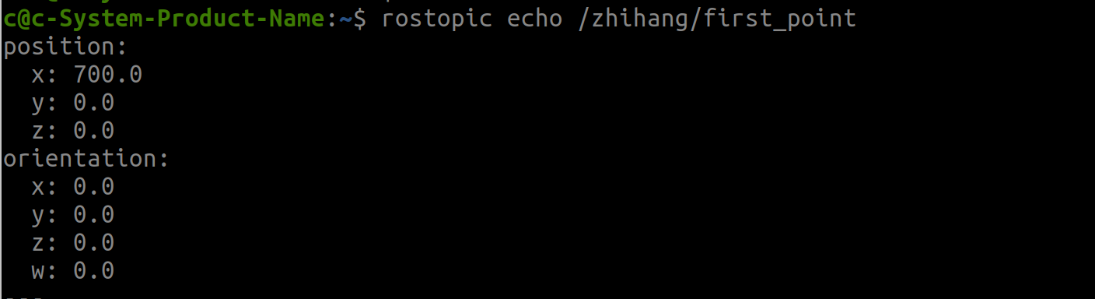
python3 get\_local\_pose.py iris 1

1. 发布两居民区间GPS引导点

cd ~/XTDrone/zhihang2025

python3 Pub\_first\_point.py

同时，选手可以通过订阅/zhihang/first\_point话题来获取GPS点信息，其中position下的x，y为GPS引导点的中心坐标。

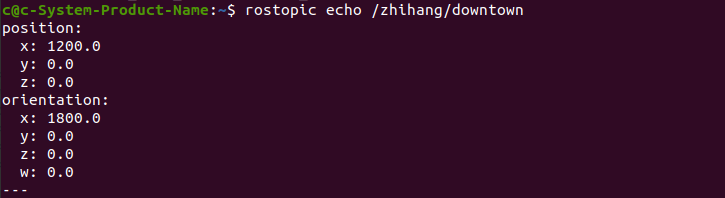


1. 发布两个居民区中心坐标

cd ~/XTDrone/zhihang2025

python3 Pub\_downtown.py

同时，选手可以通过订阅/zhihang/downtown话题来获取两个居民区中心点信息，其中position下的x，y为居民区1的中心坐标，orientation下的x，y为居民区2的中心坐标。



1. 开启待救援目标移动代码

cd ~/XTDrone/zhihang2025

python3 zhihang\_control\_targets.py

1. 记录数据，要求选手在无人机起飞前启动数据记录：

cd ~/XTDrone/zhihang2025

rosbag record -O score1 /standard\_vtol\_0/mavros/state /iris\_0/mavros/state /gazebo/model\_states /xtdrone/standard\_vtol\_0/cmd /xtdrone/iris\_0/cmd /zhihang/first\_point /zhihang2025/first\_man/pose /zhihang2025/second\_man/pose /zhihang2025/third\_man/pose /zhihang2025/iris\_healthy\_man/pose /zhihang2025/iris\_bad\_man/pose /zhihang/downtown

* 注意：选手需严格按照比赛方要求步骤启动程序，比赛过程中不能随意关闭或重新开启以上程序，否则可能会导致得分无效。

**选手需要严格按照第七条指令生成score1.bag文件，否则可能导致成绩无效。**

选手需要将score1.bag文件压缩并提交主办方。

1. **作品评判标准**

评委主要按照作品的符合性、作品的完整性、系统得分进行综合评价，分值分配情况如下:

1.作品符合性：20分

研究思路合理性(分值:10分)

技术路线可行性(分值5分)

作品创新性(分值5分)

2.作品完整性：10分

作品的源代码可以在赛方提供的仿真环境进行部署和正确运行(分值:10分)

3.系统得分：70分

阶段一任务：虚拟无人机A（垂起固定翼）以旋翼模式从起飞平台起飞，建议飞至20m高度切换成固定翼模式完成后续除降落外的所有任务。虚拟无人机A需要利用居民区间GPS引导点信息以及两居民区中心坐标，确定两居民区中间区域并在该范围内搜索并区分3个模拟人员：1个危重人员（穿红色衣服），1个轻伤人员（穿黄色衣服），1个健康人员（穿白色迷彩）。虚拟无人机A将3个模拟人员对应降落平台的中心坐标发送给虚拟无人机B（四旋翼）后返航。

危重人员信息话题：/zhihang2025/first\_man/pose，该话题的消息类型为Pose，要求position.x，position.y分别表示危重人员的x，y坐标；

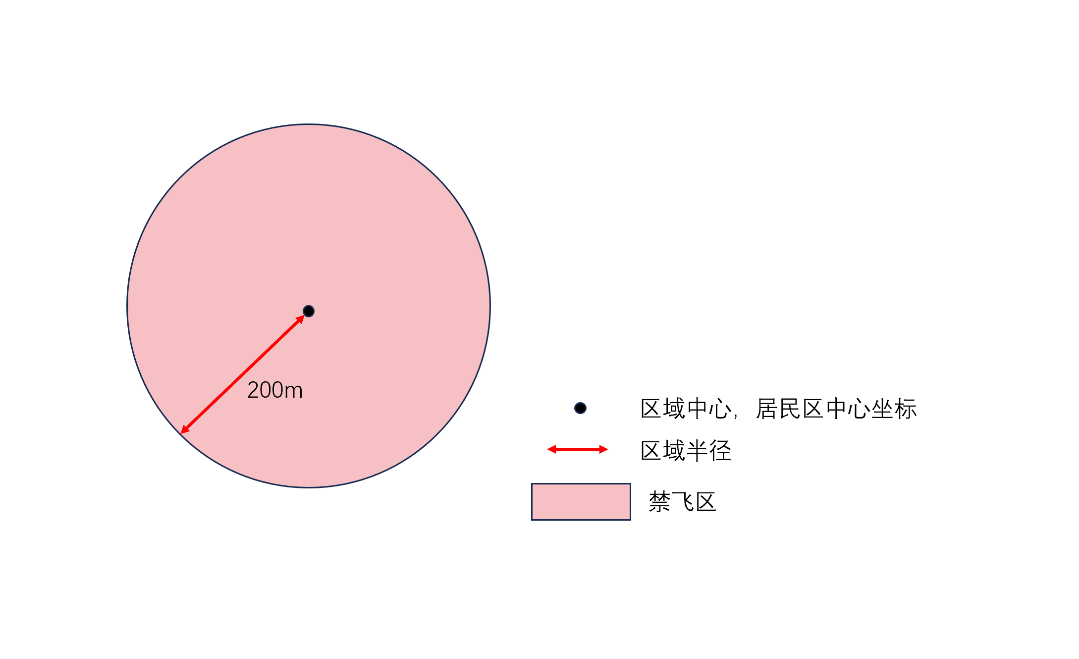
轻伤人员信息话题：/zhihang2025/second\_man/pose，该话题的消息类型为Pose，要求position.x，position.y分别表示轻伤人员的x，y坐标；

健康人员信息话题：/zhihang2025/third\_man/pose，该话题的消息类型为Pose，要求position.x，position.y分别表示健康人员的x，y坐标。

阶段一评分规则（30分）：

1、躲避居民区考察。在地图区域范围内存在两个禁飞区，禁飞区范围为：居民区中心为圆心，半径为200m的圆形区域。

若虚拟无人机A飞至禁飞区范围内，虚拟无人机A视作坠毁，总分SCORE为坠毁前阶段一所有分数之和，后续任务不再计分。



禁飞区其一示例

2、识别目标准确度考察。以危重人员搜索任务为例：危重人员坐标以虚拟无人机A首次发布危重人员信息话题为准，按照虚拟无人机发布的危重人员对应降落平台坐标与该时刻危重人员对应降落平台坐标的水平距离S1计算分数。计算公式score1\_1=(100-10xD)x0.1。

当S1> 10米时，score1\_1=0分。

危重人员最终得分为score1\_1，轻伤人员与健康人员都按照该标准评分，二者最终得分为score1\_2和score1\_3。阶段一总分SCORE1=score1\_1+score1\_2+score1\_3。

3、固定翼模式完成任务考察。要求虚拟无人机A在执行搜索任务时一直保持固定翼飞行状态，针对虚拟无人机A设置了旋翼模式区，要求在该范围外必须以固定翼模式飞行，若违反该规则总分SCORE=0分。

4、垂起无人机降落考察。要求虚拟无人机A完成阶段一所有任务后在旋翼模式区域锁桨降落，若违反该规则总分SCORE=0分。

旋翼模式区范围为以固定翼起始点为圆心，半径为100m的圆形区域。

阶段二任务：虚拟无人机B（四旋翼）接收到虚拟无人机A发布的模拟人员位置信息，待虚拟无人机A在旋翼模式区域锁桨降落后，虚拟无人机B解锁对已知位置信息的2个目标（健康、危重）进行任务分配与路径规划。虚拟无人机B首先抵达健康人员目标位置，对健康人员对应降落平台进行追踪，当无人机与地面垂直相对距离小于等于0.7m时，发布健康人员对应降落平台的坐标。之后飞抵危重人员对应降落平台目标位置，当无人机与地面垂直相对距离小于等于0.5m时，发布危重人员对应降落平台的坐标。

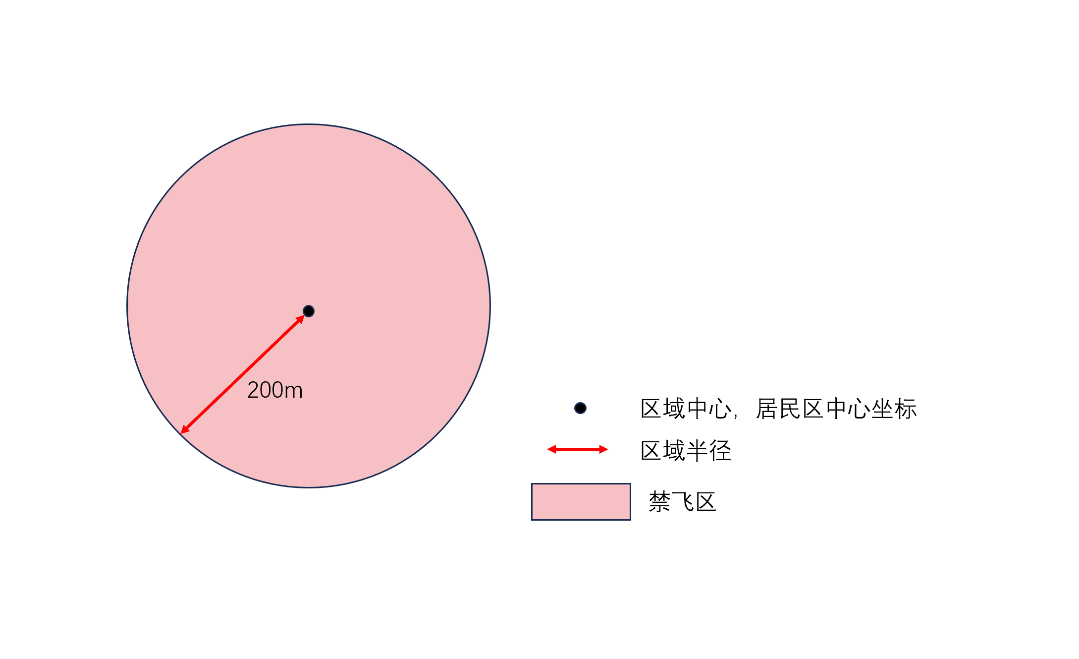
阶段二健康人员信息话题：/zhihang2025/iris\_healthy\_man/pose，该话题的消息类型为Pose，要求position.x，position.y分别表示健康人员对应降落平台中心的x，y坐标。

阶段二危重人员话题信息：/zhihang2025/iris\_bad\_man/pose，该话题的消息类型为Pose，要求position.x，position.y分别表示危重人员对应降落平台中心的x，y坐标。

阶段二评分规则（40分）：

1、躲避居民区考察。在地图区域范围内存在两个禁飞区，禁飞区范围为：居民区中心为圆心，半径为200m的圆形区域。

若虚拟无人机B飞至禁飞区范围内，虚拟无人机B视作坠毁，总分SCORE为坠毁前阶段一所有分数之和，后续任务不再计分。



禁飞区其一示例

2、识别健康人员准确度考察。健康人员坐标以虚拟无人机B首次发布阶段二健康人员信息话题为准，按照虚拟无人机B发布的健康人员坐标与该时刻健康人员对应降落平台中心真实坐标的水平距离S2计算分数，要求此时无人机相对地面高度小于等于0.7m。计算公式score2\_1=(100-33xD)x0.1。

当S2> 3米 时，score2\_1=0分。

3、识别危重人员准确度考察。危重人员坐标以虚拟无人机B首次发布阶段二危重人员信息话题为准，按照虚拟无人机B发布的危重人员坐标与该时刻危重人员对应降落平台中心真实坐标的水平距离S2计算分数，要求此时无人机相对地面高度小于等于0.5m。计算公式score2\_2=(100-33xD)x0.3。

当S3> 3米 时，score2\_2=0分。

阶段二分数SCORE2=score2\_1+score2\_2。

4、旋翼无人机降落考察。要求虚拟无人机B完成阶段二所有任务后在旋翼模式区域锁桨降落，若违反该规则阶段二总分为0分。

5、旋翼无人机解锁起飞考察。要求虚拟无人机B在虚拟无人机A锁桨降落后起飞，若违反该规则阶段二总分为0分。

总分：SCORE = SCORE1+SCORE2。

**选手得分相同时，根据执行任务总时长T1进行排名，T1从虚拟**

**无人机A解锁开始计时，至虚拟无人机B锁桨计时结束。（T1时长取小数点后四位）**

1. **环境配置及安装教程**

仿真平台及参数

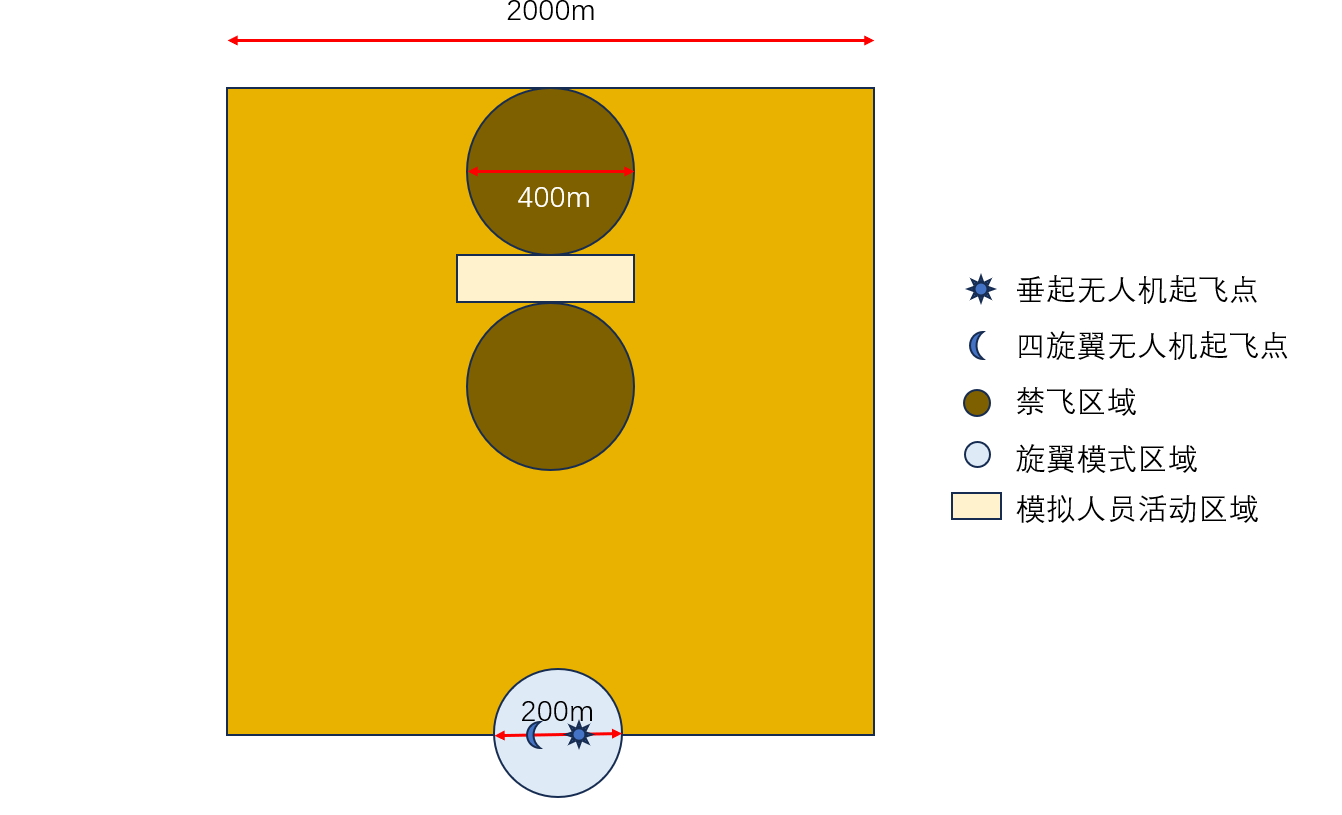
1. 操作系统

Ubuntu20.04

1. 仿真平台

Gazebo11，搭配QGC使用

1. 仿真地图



仿真环境示例图

面积：2千米×2千米=4平方千米

虚拟无人机参数：

Standard\_vtol无人机在仿真环境中：长：0.55米，翼展：2.144米，高：0.05米，重量：5千克。

四旋翼模式推荐速度：3m/s

固定翼模式推荐速度：10-17m/s，转弯半径：FW\_L1\_PERIOD（L1控制器时间常数）推荐范围10-15s。

地面站参数详见：

http://docs.px4.io/v1.11/zh/advanced\_config/parameter\_reference.html

* 附录：px4开发环境搭建教程

px4开发环境搭建教程

注意：本教程系统使用的是ubuntu20.04

1.Ubuntu系统换源

由于Ubuntu20.04系统安装默认源为国外源（服务器位于国外），导致国内用户在开发的过程中会遇到一系列问题，为避免该系列问题，需要我们对安装好的Ubuntu系统进行换源操作。

打开终端依次输入：

备份---sudo cp /etc/apt/sources.list /etc/apt/[sources.list.back](http://sources.list.back/" \o "sources.list.back)

修改源---sudo vim /etc/apt/sources.list （选择添加的镜像源和这部分操作可看博客）https://blog.csdn.net/tangling1/article/details/132150597或者https://blog.csdn.net/qq\_45878098/article/details/126037838）

更新---sudo apt-get update

2.一键安装ros （此部分推荐使用该指令一键安装，不要看ros官网的安装步骤）

（1）所需要的指令只有一个，在终端输入：

wget http://fishros.com/install -O fishros && . fishros

安装过程中需要在[ ]输入数字，依次是1、2、1、1 就能安装成功（注意看是noetic版本）

（2）验证，开三个不同的终端分别输入指令：

roscore

rosrun turtlesim turtlesim\_node

rosrun turtlesim turtle\_teleop\_key

键盘运行小乌龟检测是否安装成功

3.依赖安装（只有三条指令，注意第二条指令是pip3不是pip2，三条指令复制完整）

sudo apt install ninja-build exiftool ninja-build protobuf-compiler libeigen3-dev genromfs xmlstarlet libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev python3-pip gawk

pip3 install pandas jinja2 pyserial cerberus pyulog==0.7.0 numpy toml pyquaternion empy pyyaml

pip3 install packaging numpy empy toml pyyaml jinja2 pyargparse kconfiglib jsonschema future

这部分如果出现报错，参考语雀教程，可先更新 setuptools 和 pip

4.Gazebo安装

（1）先新建工作站，在终端依次输入以下指令

mkdir -p ~/catkin\_ws/src

mkdir -p ~/catkin\_ws/scripts

cd catkin\_ws && catkin init

catkin build

（2）卸载之前的Gazebo，可在上一个终端往下操作

cd..

sudo apt-get remove gazebo\*

sudo apt-get remove libgazebo\*

sudo apt-get remove ros-noetic-gazebo\*

（3）安装gazebo

sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb\_relea se -cs` main"> /etc/apt/sources.list.d/gazebo-stable.list' （粘贴时注意这是一条完整的指令，中间没有换行）

cat /etc/apt/sources.list.d/gazebo-stable.list

wget https://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -

sudo apt-get update（确定这条指令执行完成后再操作下面）

sudo apt-get install gazebo11

sudo apt-get install libgazebo11-dev

最后在终端输入指令：gazebo，若出现界面则成功安装

（4）Gazebo的ros插件

先安装依赖 （下面是一条完整的指令，没有回车）：

sudo apt-get install ros-noetic-moveit-msgs ros-noetic-object-recognition-msgs ros-noetic-octomap-msgs ros-noetic-camera-info-manager ros-noetic-control-toolbox ros-noetic-polled-camera ros-noetic-controller-manager ros-noetic-transmission-interface ros-noetic-joint-limits-interface

安装完成后依次输入：

cd ~/catkin\_ws

catkin build （这里会报错，出现failed表示不成功，不成功后可以输入指令cakin clean 再继续catkin build，多重复几次；）

编译成功后执行如下两条指令，判断gazebo\_ros是否安装成功：

打开第一个终端，输入roscore

打开第二个终端，依次输入：

source ~/catkin\_ws/devel/setup.bash

rosrun gazebo\_ros gazebo

出现gazebo界面表示成功；

最后还需要下载语雀文档里的models.zip，将该附件解压缩后放在~/.gazebo中（注意.gazebo是隐藏文件夹，需要ctrl+h显示出来）解压完成后在~/.gazebo/models/路径下可以看到很多模型

5.MAVROS安装

在新的终端依次输入以下四条指令

sudo apt install ros-noetic-mavros ros-noetic-mavros-extras

wget [https://gitee.com/robin\_shaun/XTDrone/raw/master/sitl\_config](https://gitee.com/robin_shaun/XTDrone/raw/master/sitl_config/mavros/install_geographiclib_datasets.sh)

[/mavros/install\_geographiclib\_datasets.sh](https://gitee.com/robin_shaun/XTDrone/raw/master/sitl_config/mavros/install_geographiclib_datasets.sh)

sudo chmod a+x ./install\_geographiclib\_datasets.sh

sudo ./install\_geographiclib\_datasets.sh #这步需要装一段时间

6.PX4配置

这里采用的方法是直接下载语雀文档里的压缩包再编译的办法。

语雀文档链接：

https://www.yuque.com/xtdrone/manual\_cn/basic\_config\_13

（1）先下载PX4\_Firmware.zip，解压后在终端依次输入：

cd PX4\_Firmware (确定进入到该文件夹下)

make px4\_sitl\_default gazebo（会报错ninja，解决办法先执行：make clean再执行该指令，编译完成后出现gazebo界面关闭即可）

（2）修改路径（在home中点开隐藏的~/.bashrc文件，在最下面添加如下四条路径，注意不要更改下面四个路径的顺序）：

source ~/catkin\_ws/devel/setup.bash

source ~/PX4\_Firmware/Tools/setup\_gazebo.bash ~/PX4\_Firmware/ ~/PX4\_Firmware/build/px4\_sitl\_default

export ROS\_PACKAGE\_PATH=$ROS\_PACKAGE\_PATH:~/PX4\_Firmware

export ROS\_PACKAGE\_PATH=$ROS\_PACKAGE\_PATH:~/PX4\_Firmware/Tools/sitl\_gazebo

添加完毕并保存后运行：source ~/.bashrc

再开个终端依次运行：

cd ~/PX4\_Firmware

roslaunch px4 mavros\_posix\_sitl.launch

出现gazebo界面里面有个无人机表示成功

（3）通信验证

再开一个终端运行命令：rostopic echo /mavros/state

若出现connected: True,则说明MAVROS与SITL通信成功。

如果是false，先确保.bashrc里的路径正确，如果确定正确还是false，在launch文件夹下找到 mavros\_posix\_sitl.launch文件点开，修改第25行的：nane="fcu ur1" default= ”udp: I/:145400locaLhost:14557"，把udp修改成“udp:“:245400LocaLhost: 34580”保存，并重新运行roslaunch px4 mavros\_posix\_sitl.launch和rostopic echo /mavros/state

7.QGC地面站安装

（1）新开一个终端依次输入以下5条指令：

sudo usermod -a -G dialout $USER

sudo apt-get remove modemmanager -y

sudo apt install gstreamer1.0-plugins-bad gstreamer1.0-libav gstreamer1.0-gl -y

sudo apt install libqt5gui5 -y

sudo apt install libfuse2 -y

（2）点开QGC官网：<https://docs.qgroundcontrol.com/master/en/qgc-user-guide/getting_started/download_and_install.html>

找到“Ubuntu Linux”处点击下载



下载完成后先在终端里cd到下载的文件夹里，依次输入下面两个指令：

chmod +x ./QGroundControl.AppImage

./QGroundControl.AppImage

安装完成后会跳出QGC的界面表示成功

8.XTDrone

（1）新开一个终端或者在上一个终端输入cd..，然后依次输入以下指令（下面的指令只要不报红色的错误都可以不管，继续往下进行）：

git clone https://gitee.com/robin\_shaun/XTDrone.git

cd XTDrone

git checkout 1\_13\_2

git submodule update --init --recursive

# 修改启动脚本文件

cp sitl\_config/init.d-posix/\* ~/PX4\_Firmware/ROMFS/px4fmu\_common/init.d-posix/

# 添加launch文件

cp -r sitl\_config/launch/\* ~/PX4\_Firmware/launch/

# 添加世界文件

cp sitl\_config/worlds/\* ~/PX4\_Firmware/Tools/sitl\_gazebo/worlds/

# 修改部分插件（这部分可能会显示缺东西什么的，可以不用管，尤其是后两条指令，报错可以继续往下进行）

cp sitl\_config/gazebo\_plugin/gimbal\_controller/gazebo\_gimbal\_controller\_plugin.cpp ~/PX4\_Firmware/Tools/sitl\_gazebo/src

cp sitl\_config/gazebo\_plugin/gimbal\_controller/gazebo\_gimbal\_controller\_plugin.hh ~/PX4\_Firmware/Tools/sitl\_gazebo/include

cp sitl\_config/gazebo\_plugin/wind\_plugin/gazebo\_ros\_wind\_plugin\_xtdrone.cpp ~/PX4\_Firmware/Tools/sitl\_gazebo/src

cp sitl\_config/gazebo\_plugin/wind\_plugin/gazebo\_ros\_wind\_plugin\_xtdrone.h ~/PX4\_Firmware/Tools/sitl\_gazebo/include

# 修改CMakeLists.txt

cp sitl\_config/CMakeLists.txt ~/PX4\_Firmware/Tools/sitl\_gazebo

# 修改部分模型文件

cp -r sitl\_config/models/\* ~/PX4\_Firmware/Tools/sitl\_gazebo/models/

# 替换同名文件

cd ~/.gazebo/models/

rm -r stereo\_camera/ 3d\_lidar/ 3d\_gpu\_lidar/ hokuyo\_lidar/

（2）重新编译，依次输入下面三条指令

cd ~/PX4\_Firmware

rm -r build/

make px4\_sitl\_default gazebo（编译时报错可以make clean再继续执行编译，出现报错可以多编译几次）

新开一个终端，输入指令：source ~/ . bashrd

9.验证环境是否可用

使用键盘控制无人机

在一个终端中运行以下指令：

cd ~/PX4\_Firmware

roslaunch px4 indoor1.launch

再开一个终端打开通信程序：

cd ~/XTDrone/communication/

python multirotor\_communication.py iris 0

再开一个终端运行：

cd ~/XTDrone/control/keyboard

python multirotor\_keyboard\_control.py iris 1 vel

如果室内场景能够成功加载、四旋翼可以通过键盘控制起飞，证明环境已经成功配置完成。

附件1

评分规则

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评分项目 | 评分标准 | 评分细节 | 分值 | 得分 |
| 作品符合性（20分） | 研究思路合理性  （10分） | 作品中的各个部分应当形成一个完整的逻辑链条，相互之间有联系、有依存，构成一个统一的整体 | 5 |  |
| 内容有深度和广度，能够深入探讨问题的本质、涵盖相关方向和观点 | 3 |  |
| 逻辑严谨、层次分明、能够清晰地传达作品内容 | 2 |  |
| 技术路线可行性  （5分） | 技术路线设计合理，步骤清晰，每一步都有明确的目标和实现方法 | 2 |  |
| 技术路线中的关键技术点有详细的解释和论证 | 2 |  |
| 技术路线能够有效解决研究问题，具有实际操作的可行性 | 1 |  |
| 作品创新性  （5分） | 技术路线融入了创新的理念和方法，采用了前沿的技术手段和独特的技术方案，展现出对技术的独到见解和创造性思维 | 2 |  |
| 研究思路合理且新颖，具有一定的独特性和前瞻性 | 2 |  |
| 技术具备一定的实际应用价值，能够解决技术问题或者满足实际需求，具有适用性和可操作性 | 1 |  |
| 作品完整性（10分） | 代码结构完整性 （5分） | 代码模块化设计，各模块功能明确，易于维护和扩展 | 2 |  |
| 代码注释清晰，便于理解和后续开发 | 2 |  |
| 代码风格一致，符合行业标准 | 1 |  |
| 仿真环境正确部署 （2分） | 选手的代码可以在仿真环境上成功部署 | 2 |  |
| 代码成功运行 （3分） | 代码运行结果正确，能够实现预期功能 | 2 |  |
| 代码无错误，能够顺利编译和运行 | 1 |  |
| 系统得分（70分） | 阶段一（30分） | 无人机A发布的降落平台坐标与危重人员对应降落平台坐标的水平距离S1计算分数。计算公式score1\_1=(100-10xD)x0.1。  当S1> 10米时，score1\_1=0分。 | 10 |  |
| 无人机A发布的降落平台坐标与轻伤人员对应降落平台坐标的水平距离S1计算分数。计算公式score1\_2=(100-10xD)x0.1。  当S1> 10米时，score1\_2=0分。 | 10 |  |
| 无人机A发布的降落平台坐标与健康人员对应降落平台坐标的水平距离S1计算分数。计算公式score1\_3=(100-10xD)x0.1。  当S1> 10米时，score1\_3=0分。 | 10 |  |
| 阶段二（40分） | 无人机B发布的健康人员坐标与健康人员对应降落平台中心真实坐标的水平距离S2计算分数，要求此时无人机相对地面高度≤0.7m。计算公式score2\_1=(100-33xD)x0.1。  当S2> 3米 时，score2\_1=0分。 | 10 |  |
| 无人机B发布的危重人员坐标与该时刻危重人员对应降落平台中心真实坐标的水平距离S2计算分数，要求此时无人机相对地面高度≤0.5m。计算公式score2\_2=(100-33xD)x0.3。  当S3> 3米 时，score2\_2=0分。 | 30 |  |
| 补充： | 选手得分相同时，根据执行任务总时长T1进行排名，T1从虚拟  无人机A解锁开始计时，至虚拟无人机B锁桨计时结束。（T1时长取小数点后四位） |  |  |