空中智能博弈指导手册

链接：https://pan.baidu.com/s/13tZamW7AHXB6ySkRwj5D2A

提取码：isz9

2021年9月

**目录**

[1 任务概述 1](#_Toc83404613)

[1.1 任务内容 1](#_Toc83404614)

[1.2 任务规则 1](#_Toc83404615)

[2 软件安装 2](#_Toc83404616)

[2.1 概述 2](#_Toc83404617)

[2.2 虚拟机环境的使用 2](#_Toc83404618)

[2.3 Docker方式的环境配置 3](#_Toc83404619)

[2.4 AI开发环境配置 4](#_Toc83404620)

[2.5 态势显示工具安装 5](#_Toc83404621)

[3 工具使用说明 6](#_Toc83404622)

[3.1 虚拟机的使用 6](#_Toc83404623)

[3.1.1 打开虚拟机 6](#_Toc83404624)

[3.1.2 分布式配置 6](#_Toc83404625)

[3.1.3 启动仿真引擎 7](#_Toc83404626)

[3.1.4 python代码 8](#_Toc83404627)

[3.2 态势显示工具的使用 9](#_Toc83404628)

[4 AI开发指南 11](#_Toc83404629)

[4.1 训练环境介绍 11](#_Toc83404630)

[4.2 AI开发框架 11](#_Toc83404631)

[4.2.1 基类智能体 11](#_Toc83404632)

[4.2.2 仿真环境构建和运行 12](#_Toc83404633)

[4.2.3 连接客户端 12](#_Toc83404634)

[4.3 AI开发流程 13](#_Toc83404635)

[4.4 态势接口说明 13](#_Toc83404636)

[4.5 指令说明 15](#_Toc83404637)

[附录：补充说明 18](#_Toc83404638)

# 任务概述

## 任务内容

空中智能博弈以异构、协同全透明态势空战为想定，为AI提供一个虚拟空战场景，考验AI在编队行动、协同探测、目标打击等方面的决策能力。作战想定为红蓝双方均为1架有人机携带4架无人机进行确定信息透明态势的自由空战，并模拟雷达探测功能及空空导弹攻击过程，进攻方向为红蓝迎头进入，初始高度9000～10000米，速度0.9马赫，作战空域为300km×300km，战斗时长为20分钟。

## 任务规则

**基本原则：**有人机为双方的指挥所级别设置，一旦被摧毁，即判定编队丧失作战能力，根据当前的双方积分进行胜负判定。

**比赛得分：**击落得分和剩余战斗力得分。

**裁判规则：**时间达到总时长T时或者存在有人机被击落时，单场比赛停止，进行得分计算和胜负判定。

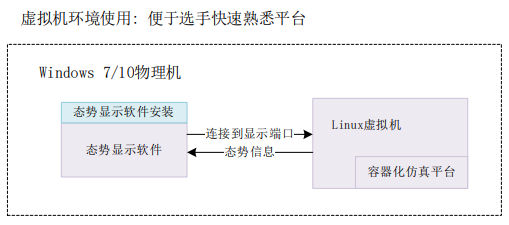
**胜负判定：**优先根据红蓝编队双方的击落得分进行胜负判定，积分高者获胜；当击落得分双方相等时，根据剩余战斗力得分进行胜负判定，剩余战斗力高者获胜；当前两项等分都相等且有人机被击落时，有人机先被击落一方判定为负，有人机后被击落一方判定为胜；当前两项等分都相等且有人机未被击落且存在战机被击落时，哪一方先击落敌方第一架战机，哪一方获胜；当上述条件都无法判定胜负时，判定平局。

**积分原则：**每局初始分为0分；若有人机战损，减50分，且当局结束判负；若无人机战损，每1架减10分；每发射1枚导弹减3分，且一方无弹时当局结束判负。除前述情形外，每局依据小分评判胜负，若双方得分相同，则依据AI的作战表现，判定积极对抗且能够更多控制对方半场的一方胜利

# 软件安装

## 概述

提供虚拟机与Docker镜像两种方式供选择。对于虚拟机方式，下载“601vm.rar”虚拟机环境后，直接启动虚拟机，运行对战框架里的调度程序，然后连接物理机上的态势显示软件即可直观地观看对战过程和效果。对于Docker镜像方式，需自行在Linux物理机上配置容器化仿真环境镜像和对战框架程序，且暂不支持查看对战过程和效果。



**图1.虚拟机方式**



**图2.Docker镜像方式**

## 虚拟机环境的使用

* 下载并安装VMware Workstation 14.1.0 Pro；
* 启动VMware Workstation，“文件”->“打开”，选择从官网下载的虚拟机镜像文件(后缀名为.vmx的文件)，点击“我已复制该虚拟机”，即可加载，加载完成后启动虚拟机；
* 虚拟机启动后，打开/home/ds/soft/pycharm-community-2021.1.1/bin目录， 打开终端，运行./pycharm.sh；
* 打开pycharm后按照后文要求修改配置文件，然后运行 run.py 文件即可；
* 启动态势显示软件，打开\config\XPlayer.config 文件，修改“态势显示域 ID”，设置为16://1，同时将“使用网络通信”的值改为“true”，运行“MapStudioPlayer.bat”启动态势回放工具，然后运行仿真引擎，即可与仿真平台建立连接，实时显示战场态势。

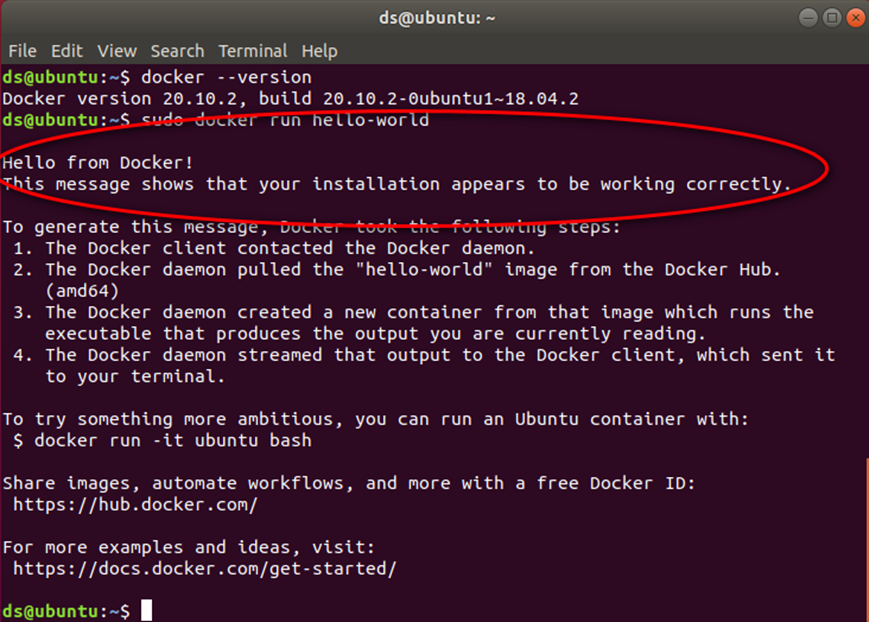
## Docker方式的环境配置

以下是在Ubuntu 16.04上在线安装Docker 20.10.2版本（Docker最新版）的操作流程：

* 移除已经安装过的docker（如果曾经安装过的话）：

sudo apt-get remove docker docker-engine docker.io

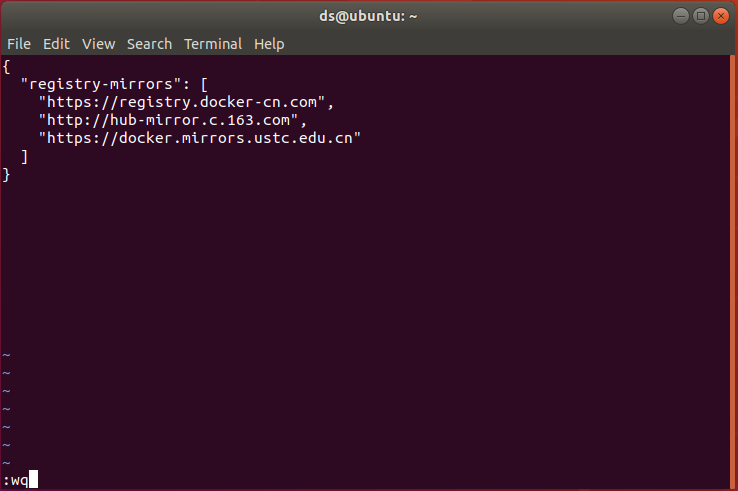
* 更新包索引：sudo apt-get update
* 安装最新的docker版本：sudo apt install docker.io
* 检查自己安装的版本，并且运行hello-word实例：docker version
* 运行实例：sudo docker run hello-world



**图3.成功运行结果**

* 更换国内源，配置镜像加速，创建或修改/etc/docker/daemon.json文件：

sudo vi /etc/docker/daemon.json



**图4.配置文件内容**

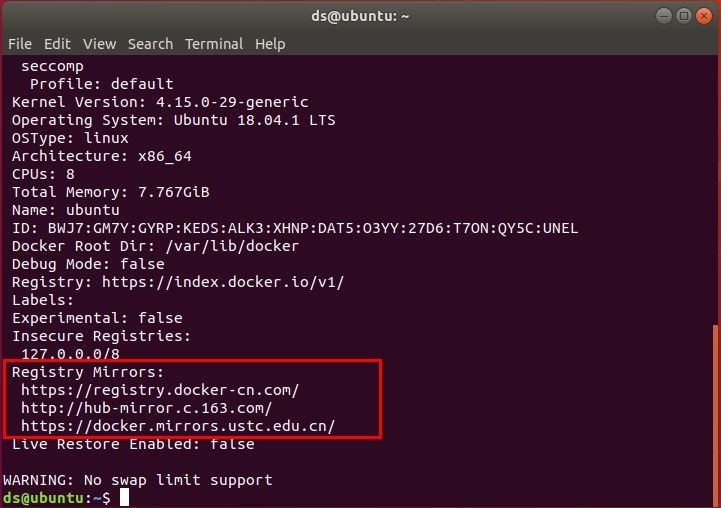
按ESC键后输入wq保存文件并退出。

* 加载并重启 docker：

sudo systemctl daemon-reload

sudo systemctl restart docker

* 查看是否成功：sudo docker info



**图5.成功安装结果**

**Docker镜像加载：**Docker安装完成后，使用如下命令加载docker镜像：docker load -i xsim.ta，加载完的镜像可被所提供开发框架里面的环境管理类EnvManager直接调用和运行。

## AI开发环境配置

基于python(3.7.7)进行AI的开发，运行给定的AI对战框架代码，建议采用tensorflow或pytorch进行AI开发。为避免不必要的环境冲突，要求统一使用tensorflow1.14.0版本或者pytorch1.5.1版本进行开发。暂不支持使用其他深度学习框架进行开发。

表1.python依赖包列表

|  |  |
| --- | --- |
| **Package** | **Version** |
| six | 1.15.0 |
| wheel | 0.36.2 |
| setuptools | 44.1.1 |
| protobuf | 3.17.0 |
| numpy | 1.20.3 |
| grpcio | 1.37.1 |
| rpyc | 4.1.4 |
| grpcio-tools | 1.28.1 |

## 态势显示工具安装

解压MapStudio-601.rar并运行MapStudioPlayer.bat即可启动态势显示工具。

# 工具使用说明

## 虚拟机的使用

### 打开虚拟机

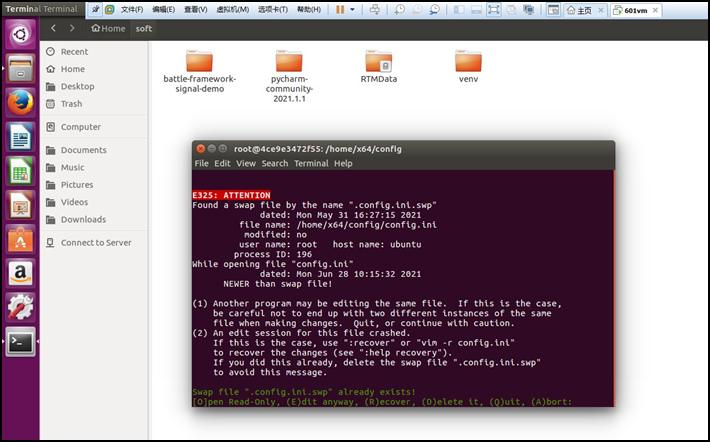
利用“VMware Workstation Pro”虚拟机软件打开“601vm.vmx”，输入虚拟机密码：1，在“/home/soft/battle-framework-signal-demo”目录下，找到“config.py”文件，并打开此文件，修改“ADDRES”对应的IP地址为本机Linux地址。

### 分布式配置

以下步骤是在虚拟机中修改态势显示工具的 IP 地址：

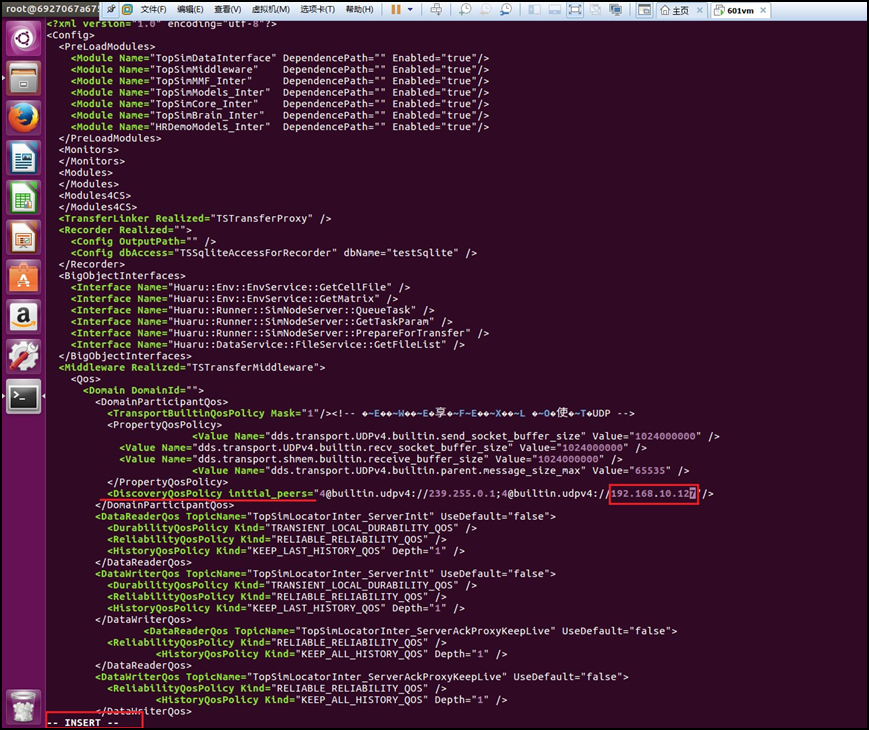
* 新建终端：右键点击“Open in terminal”；
* 第一行输入命令行：docker run -it xsim:v5.0，用于运行 docker；
* 第二行输入命令行：cd home/x64/config/，用于更改当前目录 config；
* 第三行输入命令行：vi config.ini ，用于打开文件 config.ini。

按下回车键后，出现以下画面：



**图6.准备打开界面**

在上图中，输入：E，按下回车键后，出现如图 1-6 所示界面，在图中找到“DiscoveryQosPolicy initial\_peers”对应的 IP 地址：192.168.10.127 (图中已用红框表示)，点击键盘的字母‘I’键，接下来会出现“-- INSERT --”，然后才可以修改 IP 地址。



**图7.打开后的修改界面**

修改下图中红框圈出的 IP 地址为态势显示工具所在物理机的 IP 地址。修改完成后，按Esc键进行保存，然后直接输入“:wq”，退出当前的界面。

新建终端：右键点击“Open in terminal”；第一行输入命令行：docker ps，查看当前运行的docker的ID，图中docker的CONTAINER ID为4ce9e3472f55；第二行输入命令行：docker commit -m container 4ce9e3472f55 xsim:v6.0，用于创建新的镜像版本xsim:v6.0；



**图8.创建新的镜像**

第三行输入命令行：docker images，确认是否创建新的镜像版本xsim:v6.0。

在/home/soft/battle-framework-signal-demo目录下，找到config.py文件并打开此文件，修改“IMAGE”对应的为 xsim:v6.0

### 启动仿真引擎

不需要手动启动docker和仿真引擎，已经在代码中实现程序运行自动启动。单引擎运行时在“battle-framework-signal-demo”工程下的“config.py”文件中，当“ISHOST = True”时，只能启动单个引擎，可以通过“态势显示工具”进行显示。多引擎运行时在“battle-framework-signal-demo”工程下的“config.py”文件中，当‘ISHOST = False’时，可以通过控制 XSIM\_NUM 控制启动引擎数量(取值范围：[1,10])；不能通过态势回放工具实时显示。

### python代码

在“/home/soft/pycharm-community-2021.1.1/bin”目录下，新建终端，输入命令行：./pycharm.sh，即可打开python代码。在单引擎下（即 host 模式下）分为两种运行模式：启动态势显示工具模式、不启动态势显示工具模式。

单引擎启动态势显示工具运行模式时：

* 在 Python 代码的 config.py 文件中修改“HostID”（取值范围：[1,1000]）。如果当前网段只有一台电脑启动态势显示工具，则不需要修改“HostID”。
* 在“MapStudio-601/config”文件夹下的“XPlayer.config”中修改“态势显示域 ID”为“16://HostID”。
* 打开“态势显示工具”。
* 在“battle-framework-signal-demo”工程下打开“run.py”文件，在打开的“run.py”文件窗口中点击鼠标右键，在弹出的对话框中点击【Run‘run’】按钮，即可运行python代码。
* 在“是否已启动态势显示工具? Y or N”后面直接输入“Y”，然后回车。
* 在“是否开始运行? Y or N”后面直接输入“Y”，然后回车，在态势显示工具中可以看到态势实时显示。
* 对战一局结束后，按照提示在态势显示工具中点击【实时态势】按钮（或者按快捷键 F2），清除上一局数据。
* 数据清理完成后（即“态势显示工具”恢复到初始界面），在Python 中的“是否重置运行? Y or N”后面直接输入“Y”，然后回车，代码会再次运行，态势回放工具会实时显示。
* 每一局结束，重复前两步，直到训练次数结束。

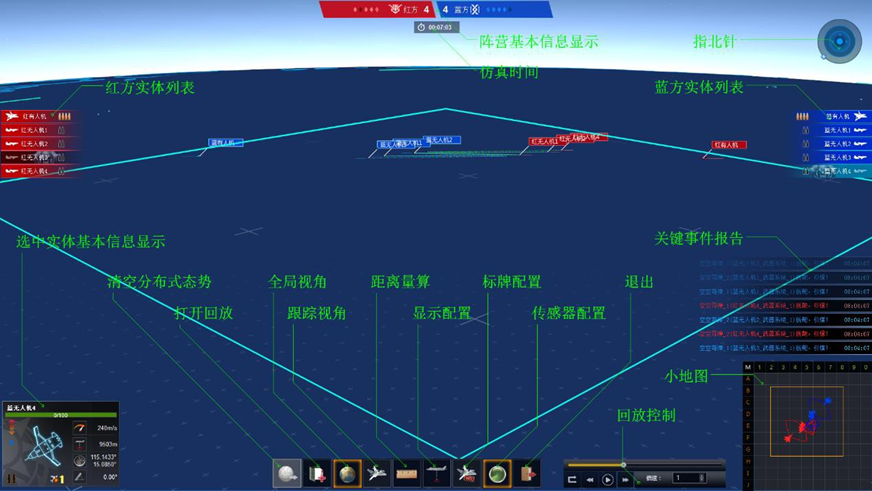
多引擎运行模式时，在“battle-framework-signal-demo”工程下打开“run.py”文件，在打开的“run.py”文件窗口中点击鼠标右键，在弹出的对话框中点击【Run‘run’】按钮，即可运行python代码。

注意：程序可以自动运行到停止；在程序运行过程中，也可以通过点击【停止运行】按钮，停止程序运行，在出现“PRocess finished with exit code 1”之前，不能再点击【停止运行】按钮。当程序停止运行后，要等待出现“PRocess finished with exit code 1”或“PRocess finished with exit code 0”才可以再次运行程序。

## 态势显示工具的使用

态势显示工具对战场态势进行形象直观的三维展示，提供态势控制功能，使军事分析人员、仿真实验人员可以根据关注时间或地点，随时控制态势的进展或切换地图视点，工具同时提供地图元素的显隐控制，更好地展现仿真过程。

态势显示工具功能组成主要包括离线回放、实时态势显示、实体列表显示、视角切换、标牌显示、传感器显示、显示控制、仿真时间和兵力显示、距离测量以及小地图展示。



**图9.态势显示工具**

**若要显示实时态势，按以下步骤操作：**

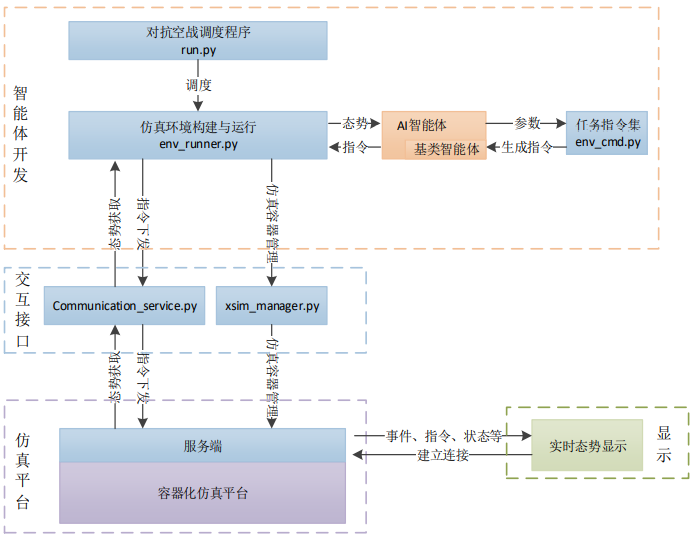
* 在 Python 代码的 config.py 文件中修改“HostID”，如果当前网段只有一台电脑启动态势显示工具，则不需要修改“HostID”。
* 打开态势显示工具文件夹下的配置文件（名称为：“MapStudio- 601/config/XPlayer.config”），修改“态势显示域 ID”与“HostID”相同。
* 运行“MapStudio-601/MapStudioPlayer.bat”启动态势显示工具。
* 成功启动态势显示工具后，选择【实时态势】，选择实时态势模式后，【打开回放】按钮置灰。
* 在“battle-framework-signal-demo”工程下打开“run.py”文件， 在打开的“run.py”文件窗口中点击鼠标右键，在弹出的对话框中选【Run‘run’】，即可运行python代码。
* 在“是否已启动态势显示工具? Y or N”后面直接输入“Y”， 然后回车。引擎成功运行后，态势显示工具刷新实时态势信息。
* 在“是否开始运行? Y or N”后面直接输入“Y”，然后回车，在态势显示工具中可以看到态势实时显示。
* 对战一局结束后，如下图 2-12 所示。在“态势显示工具”中点击【实时态势】按钮，清除上一局数据。
* 数据清理完成后（即“态势显示工具”恢复到初始界面），在Python 中的“是否重置运行? Y or N”后面直接输入“Y” ，然后回车，代码会再次运行，态势回放工具会实时显示。

**若要离线态势回放，按以下步骤操作：**

* 运行“MapStudio-601/MapStudioPlayer.bat”启动态势回放工具，点击导航栏【打开回放】按钮，弹出“回放文件列表”窗口，选择回放文件，点击【确定】按钮，态势显示工具自动加载回放文件，并展现初始态势。注：回放文件存放在“RTMData”文件夹下。
* 在播放控制条中，点击【开始】按钮（点击后，【开始】按钮将变为【暂停】按钮），对想定作战过程进行回放。
* 通过【重置】、【暂停】和【继续】（开始运行后，点击【暂 停】按钮，【暂停】按钮变为【继续】按钮），对态势回放进行控制；在“倍速”中输入需要的数值来改变态势回放的速度。

# AI开发指南

## 训练环境介绍



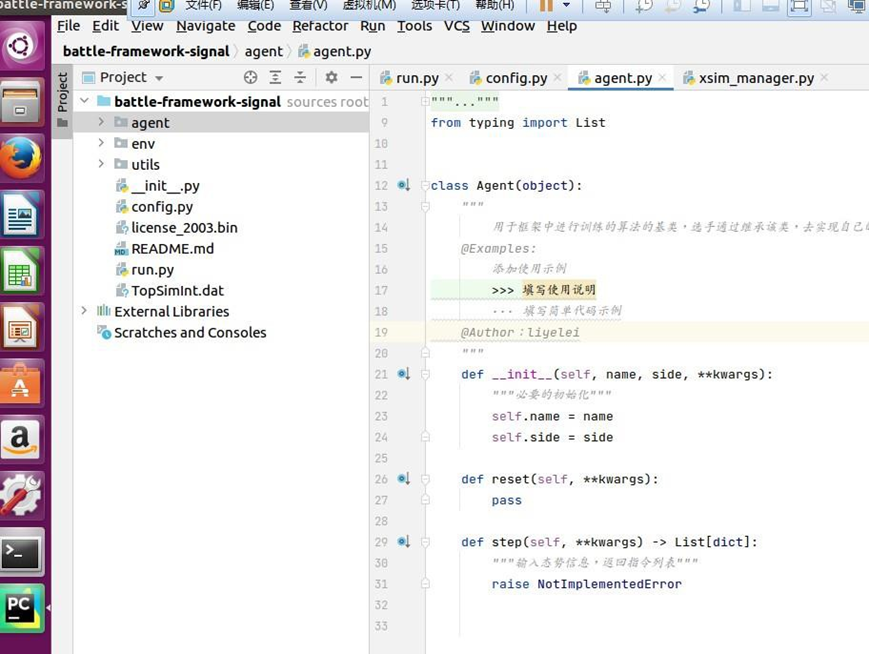
**图10.训练环境整体架构**

训练环境的整体架构如上图所示，从下至上分别是仿真平台层、交互接口层和智能体开发应用层，开发工作主要关注接口层与智能体开发层。

## AI开发框架

### 基类智能体

在模块agent.py中定义用于构建自定义智能体的基类Agent，限定其必须包含如下成员，方法和固定格式返回值，其中step方法返回值必须是可执行的指令列表。具体可参见内置规则智能体DemoAgent的构建方法。



**图11.基类智能体**

### 仿真环境构建和运行

模块env\_runner.py下的类EnvRunner为对战环境构建和运行的基类，主要方法包括对战环境初始化、启动仿真环境、根据配置信息构建红蓝双方智能体、态势获取、发送指令等，各个主要方法的功能如下。

表2.类 EnvRunner 主要方法功能说明

|  |  |
| --- | --- |
| **名称** | **功能说明** |
| init | 对战环境初始化 |
| \_init\_agents | 根据配置信息构建红蓝双方智能体 |
| \_reset | 智能体重置 |
| get\_done | 推演是否结束 |
| get\_actions | 向服务端发送指令 |
| \_agent\_step | 调用智能体 step 方法生成指令，再调用 get\_action 方法发送指令 |

在仿真环境构建和运行过程中，需要用到如下三个模块提供的功能：grpc连接客户端、任务指令、本地仿真平台容器管理。

### 连接客户端

通过 xsim\_env.py 模块下的 XsimEnv 类来建立 grpc 连接、获取态势和下发指令。

表3.类XsimEnv及其方法

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 功能 | 参数说明 | |
| init | 用于建立grpc 连接的  初始化方法 | address | 宿主机 IP 地址和端口 |
| time\_ratio | 运行的时间比 |
| reset | 重置引擎 | / | |
| close | 关闭引擎 | / | |
| step | 下发指令 | action | 智能体生成的指令列表 |

## AI开发流程

可按下述顺序设计智能体：确定状态空间->对态势信息进行预处理->确定输出动作空间->设计并搭建神经网络结构->根据所采用学习算法不同编写智能体训练脚本，对智能体进行训练。

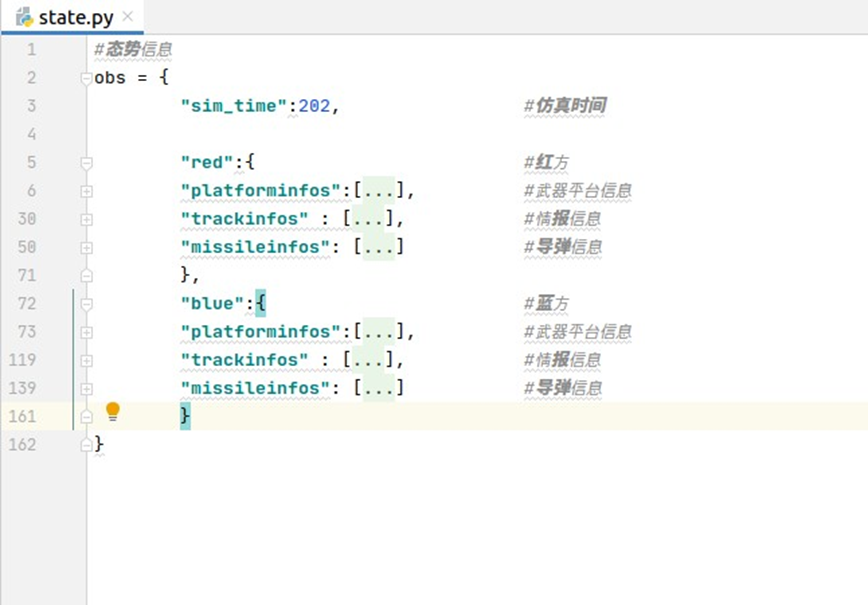
参考开发目录规范如下。

* training目录是专门供开发的目录，首先开发目录下根据学号新建子文件夹，所有训练相关的代码均放在此目录下。
* 在“training/学号”子目录下，编写基于规则或者神经网络的决策智能体，并编写相应辅助模块代码，其中构建的智能体需继承自Agent基类。然后根据自己需要改写基类智能体的代码。
* init （用于初始化），reset（用于变量重置）和 step（用于根据态势信息生成指令列表）方法，具体用法参见内置规则智能体相关代码。
* 根据个人训练环境、所采用训练框架和学习算法的不同，编写相应训练脚本，加载所开发模型与规则智能体进行对抗和训练。

**Hint：**在使用强化学习方法训练智能体时，通常需要将所提供的仿真环境封装成为一个标准的强化学习环境再进行训练。由于平台的稳定性限制，若要进行的空战模拟双方飞机数量不足5架，请向不参战的飞机持续发送原地盘旋指令而不是放弃初始化。此外，仿真平台仅提供了六类最基本的指令，如有需要请自行在其基础上封装更复杂的指令。

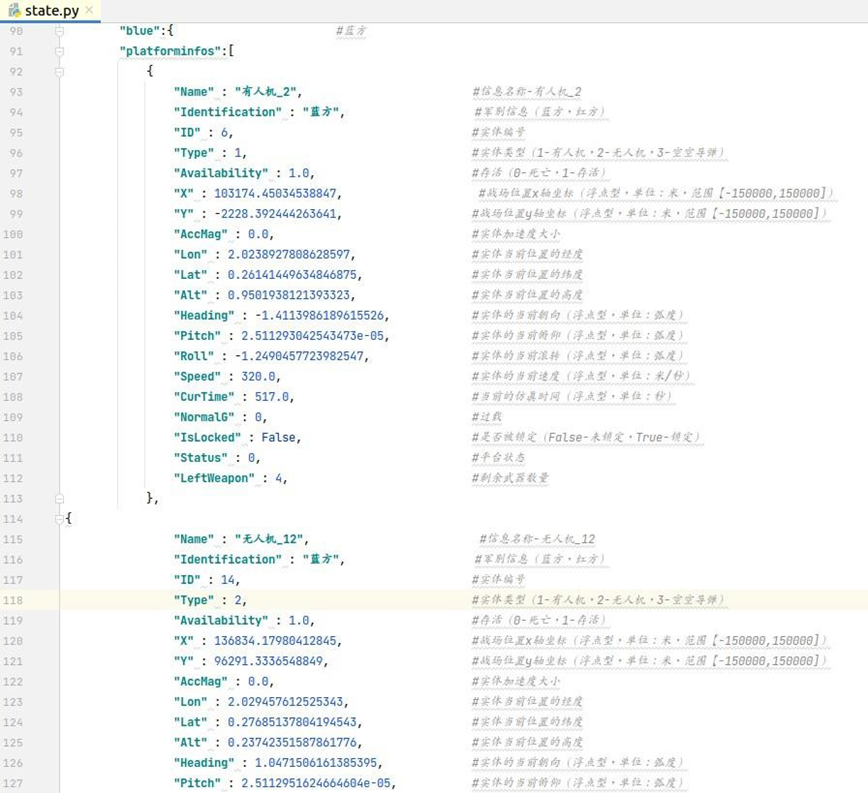
## 态势接口说明

态势信息obs的内容包括仿真时间、红方态势和蓝方态势三部分， “platforminfos”为武器平台信息、“trackinfos”为情报信息、“missileinfos ”为敌方发射的导弹信息。需要说明的是不需要探测即可获取敌方所有情报信息。需要说明的是若某个武器平台已经损毁，则态势信息不会再报有关该武器平台的任何信息。



**图12.态势信息**

**武器平台信息：**访问方式示例：obs[‘blue’][‘platforminfos’]，得到蓝方武器平台信息，列表每个元素为字典结构，用于描述该平台属性。武器平台信息里的Type为该平台的类型，三种枚举类型分别为1-有人机、2-无人机以及3-空空导弹；Status为该平台的状态信息，三种枚举类型分别为1-飞行、2-被命中以及3-被锁定。

****

**图13.武器平台信息**

**情报信息：**指当前可获取的敌方单位信息。访问方式示例：obs[‘red’][‘trackinfos’]，得到红方可见的情报信息，列表每个元素为字典结构，用于描述当前可见单位的信息。需要注意的是，红蓝双方的武器平台信息会直接报给对方，而不需要通过雷达探测。每个键的意义与武器平台相同。

**导弹信息：**访问方式示例：obs[‘blue’][‘missileinfos’]，得到红方和蓝方已经发射但尚未 爆炸的导弹信息。每个键的意义与武器平台相同。

## 指令说明

以下对可以下达给飞机的指令类型及参数含义作具体说明，每个指令对应EnvCmd类的一个方法。

表4.初始化实体指令

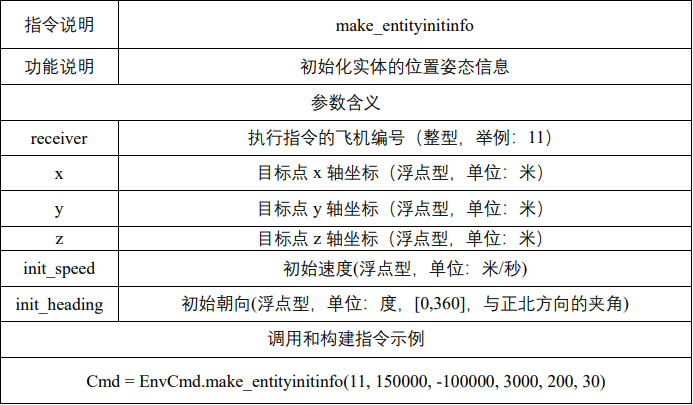


表5.打击目标指令



表6.航线巡逻指令

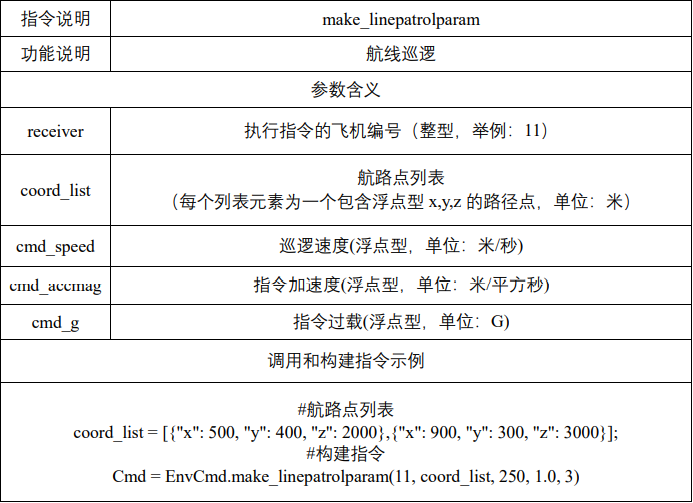


表7.区域巡逻指令

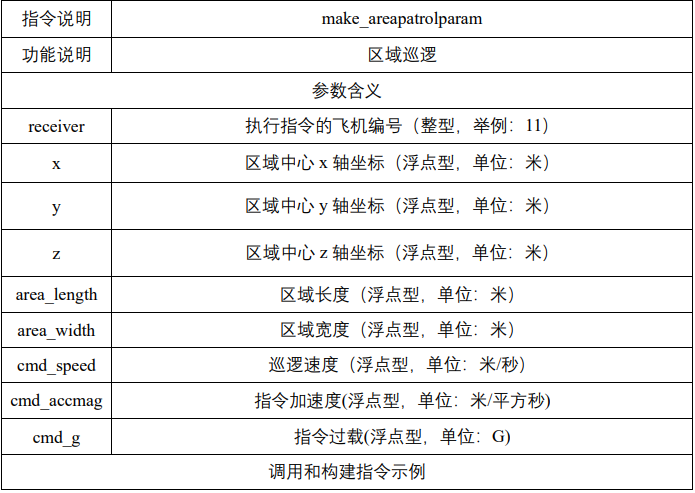




表8.机动参数调整指令

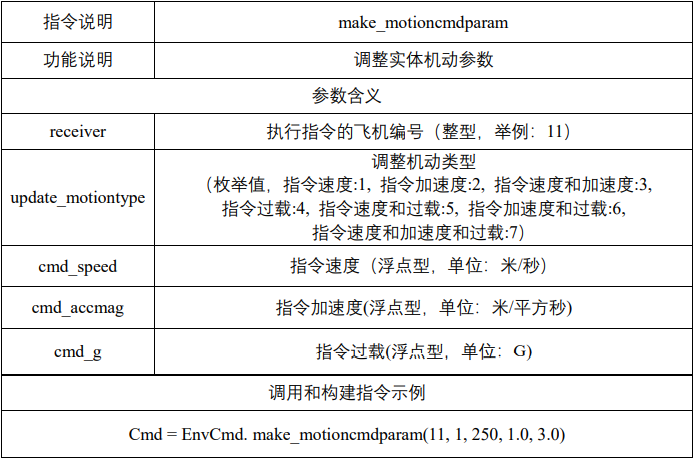
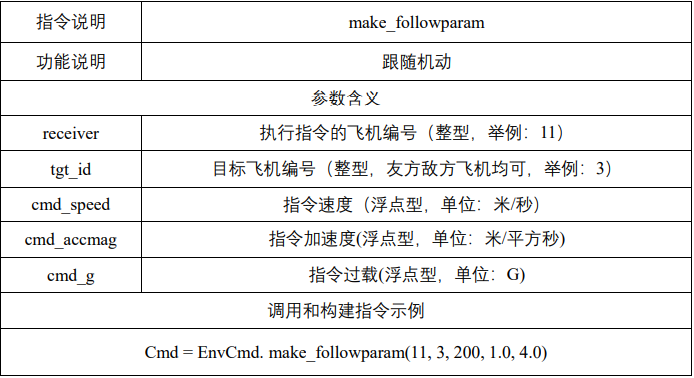


表9.跟随机动指令



# 附录：补充说明

表10.装备参数限制



**飞机巡逻规则：**

当飞机没有执行任何指令时，飞机会自行按“双180”航线巡逻，巡逻速度200米/秒，“双180”长边20公里，有人机“双180”短边8公里，无人机“双180”短边5公里，长边平行于作战区域的x轴方向。

**对导弹的制导规则：**

对导弹的制导分为中制导和末制导两个阶段，导弹被发射出来后就进入中制导阶段，当导弹距离目标20公里时，进入末制导阶段。在中制导阶段，必须有飞机的火控雷达为导弹制导，可以是发射导弹的飞机，也可以是己方的其他飞机，但是目标必须一直在提供制导的飞机火控雷达的视场范围和距离范围内，否则中制导失败，导弹丢失目标，后续也不可能进入末制导阶段。只有在中制导成功的情况下，当导弹距离目标20公里时，才能成功转入末制导阶段。