

1. 实验名称及目的

1.1 实验名称

自动测试平台使用多机型单实例自动化测试。

1.2 实验目的

掌握自动测试平台多机型单实例自动化测试的基本结构和使用流程。

1.3 关键知识点

本实验可以对python脚本进行修改，实现平台的自动测试。

程序流程：

1. 导入模块：导入Python标准库模块sys和os，以及自定义的无人机控制模块PX4MavCtrlV4和AutoMavCtrl。
2. 添加当前目录到系统路径：通过sys.path.append(os.getcwd())确保脚本能够导入当前目录下的模块。
3. 定义无人机配置：conf列表定义了无人机的机型（四旋翼或固定翼）、仿真类型（软件在环SITL或硬件在环HITL）和模拟无人机的数量。
4. 创建无人机控制器实例：mav列表创建了两个PX4MavCtrl实例，每个实例代表一个不同机型的无人机。
5. 设置地图和初始位置信息：map变量包含了地图名称和无人机的初始位置、偏航角和间隔。
6. 初始化自动环境：调用AutoMavCtrl.InitMavAutoEnv函数，使用无人机控制器实例、配置和地图信息来初始化自动控制环境。
7. 启动无人机监控线程：调用AutoMavCtrl.MavMonitor启动无人机监控线程，用于监控无人机的状态。
8. 创建自动控制对象并启动循环：为每个无人机创建AutoMavCtrl.AutoMavCtrl对象，并调用AutoMavLoopStart方法启动自动控制循环。
9. 启动仿真监控：调用AutoMavCtrl.SimMonitor启动仿真环境监控。

10. (可选) 监控无人机状态: 被注释的代码段展示了如何实时监控和打印无人机的状态信息。

实现功能的函数和类:

- `sys.path.append(os.getcwd())`: 添加当前工作目录到系统路径。
- `PX4MavCtrl.PX4MavCtrlr(port)`: 创建PX4飞行控制器的实例, `port`是通信端口号。
- `AutoMavCtrl.InitMavAutoEnv(mav, conf, map)`: 初始化自动控制环境。
- `AutoMavCtrl.MavMonitor()`: 启动无人机监控线程。
- `AutoMavCtrl.AutoMavCtrlr(mav_instance, conf)`: 为指定的无人机实例和配置创建自动控制对象。
- `AutoMavCtrl.AutoMavLoopStart()`: 启动无人机的自动控制循环。
- `AutoMavCtrl.SimMonitor()`: 启动仿真监控。

多机型单实例配置示例:

代码中通过`conf`列表配置了两种机型(四旋翼和固定翼)各一架无人机在软件在环(SITL)模式下的仿真。

监控无人机状态的示例代码:

被注释的循环代码展示了如何监控无人机的速度(`trueVelNED`)、位置(`truePosNED`)和加速度(`trueAccB`)

2. 实验效果

通过本实验可以对多机型单实例进行自动化测试。

3. 文件目录

例程目录:

[\[安装目录\]\RflySimAPIs\7.RflySimPHM\2.AdvExps\e6_MultiFrameSingleInsExp](#)

文件夹/文件名称	说明
AutoTest.py	自动测试脚本
data文件夹	仿真数据存放处
Model文件夹	仿真用到的模型和bat脚本

文件夹/文件名称	说明
Python38Run.bat	Python程序执行脚本。

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Visual Studio Code。

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。

①：推荐配置请见：<https://rflysim.com/>

5. 实验步骤

5.1. 必做实验

Step 1：了解文件性质

由于RflySimSaT的高度集成性，其只包含了一个主测试入口文件AutoTest.py文件和db.json，其余的文件均放置在RflySim平台的公共库RflySimSDK中。

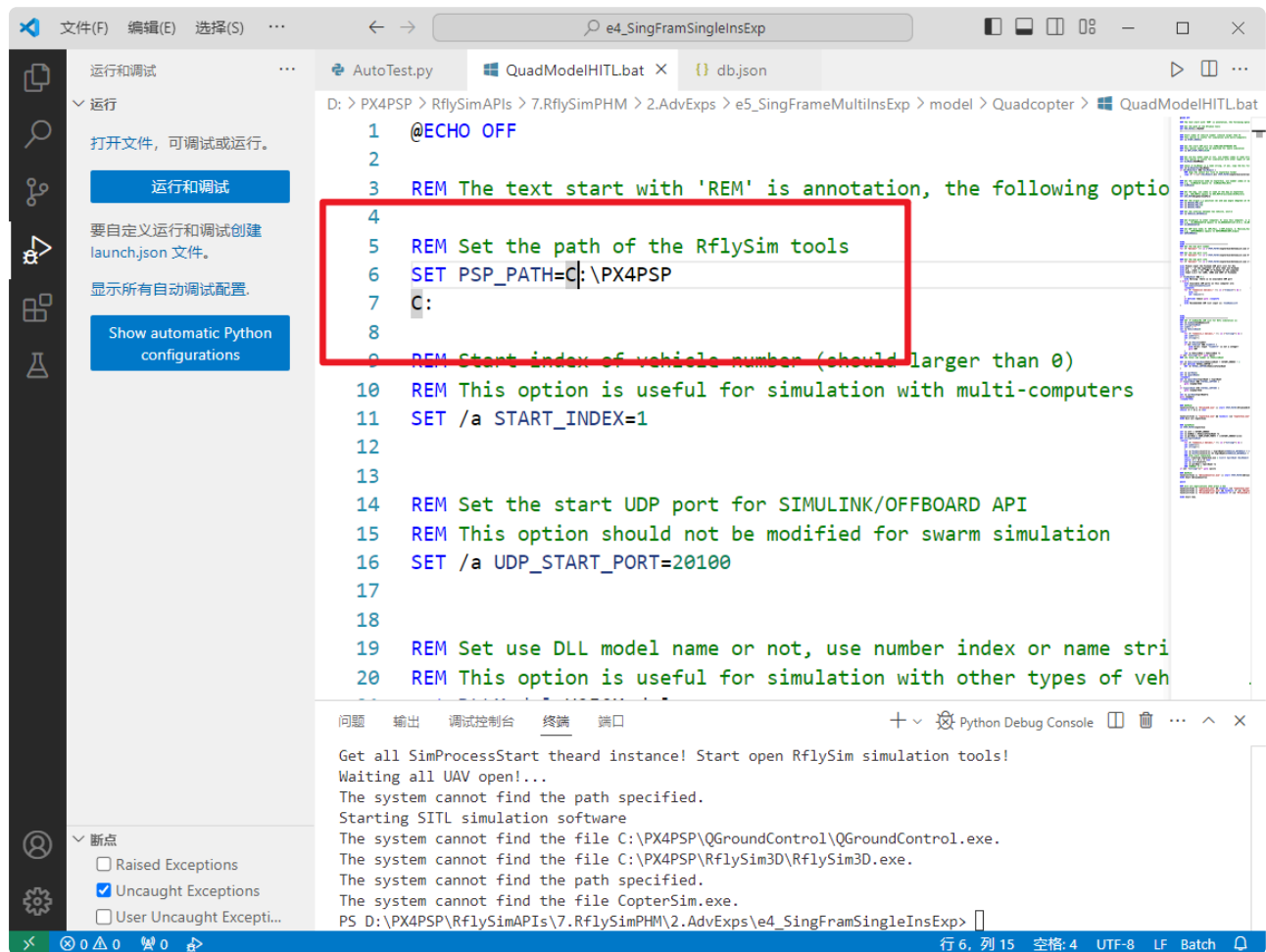
打开AutoTest.py文件，可以看到关键配置参数conf和mav,这两个参数通过列表维护不同的机型和实例数量。在本文件中设置为多机单实例模式；其中conf的机架数量为2，实例数量为2。

打开db_Quadcopter.json和db_Fixedwing.json，可以看到在“TEST_CASE”关键字后面的测试用例设置均为“1,2”，由于为多机型单实例模式，因此不同机型的实例数均为1，故

json文件应设置为单机模式，用逗号即可。此意为分别测试固定翼和四旋翼的1号和2号测试用例。注：当测试多机型时，不同机型的测试用例个数应相同。

Step 2: 配置仿真参数

根据安装平台路径修改"e6_MultiFrameSingleInsExp\model******.bat"以及"e6_MultiFrameSingleInsExp\model******.bat"。(***是指目录下的所有文件夹都要修改，***.bat是指目录下的所有.bat文件都要修改)



```
1 @ECHO OFF
2
3 REM The text start with 'REM' is annotation, the following optio
4
5 REM Set the path of the RflySim tools
6 SET PSP_PATH=C:\PX4PSP
7 C:
8
9 REM Start index of vehicle number (should larger than 0)
10 REM This option is useful for simulation with multi-computers
11 SET /a START_INDEX=1
12
13
14 REM Set the start UDP port for SIMULINK/OFFBOARD API
15 REM This option should not be modified for swarm simulation
16 SET /a UDP_START_PORT=20100
17
18
19 REM Set use DLL model name or not, use number index or name stri
20 REM This option is useful for simulation with other types of veh
```

```
问题 输出 调试控制台 终端 端口 Python Debug Console
Get all SimProcessStart theard instance! Start open RflySim simulation tools!
Waiting all UAV open!...
The system cannot find the path specified.
Starting SITL simulation software
The system cannot find the file C:\PX4PSP\QGroundControl\QGroundControl.exe.
The system cannot find the path specified.
The system cannot find the file C:\PX4PSP\RflySim3D\RflySim3D.exe.
The system cannot find the path specified.
The system cannot find the file CopterSim.exe.
PS D:\PX4PSP\RflySimAPIs\7.RflySimPHM\2.AdvExps\e4_SingFramSingleInsExp>
```

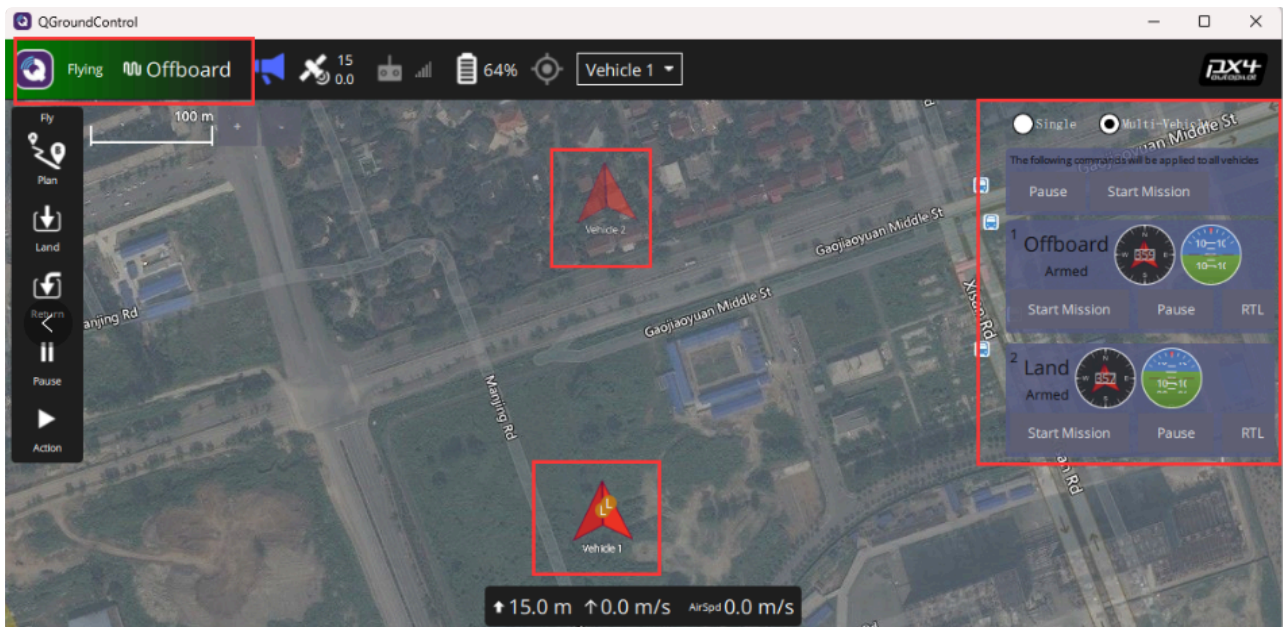
Step 3: 运行py文件

在文件夹下，双击Python38Run.bat，打开集成好的python环境，在该环境下运行AutoTest.py文件，输入python AutoTest.py，运行AutoTest.py。

```
C:\windows\system32\cmd.exe
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put Python38Run.bat into your code folder
Use the command: 'python XXX.py' to run the script with python
D:\备用\7.RflySimPHM\2.AdvExps\e6_MultiFrameSingleIn_Exp>python AutoTest.py_
```

可以看到本实验初始了一架四旋翼、一架固定翼，并自动化测试了1号四旋翼的测试用例1和2号固定翼的测试用例2，并自动化记录了测试数据，在 data\multi\Instance\TestCase_1_1\Quadcopter文件夹和 data\multi\Instance\TestCase_1_1\Fixedwing下。





Step 4: 观察数据

自动化记录了该测试用例的测试状态、测试结果等信息，并呈现了可视化结果，可以在beta\data文件夹下的TestInfo.html和TestResult.html看到。（如果运行过程中出现ulog2csv报错则需要手动将\data***\sInstance***\TestCase_**\log文件夹下的.ulg文件转换为csv）

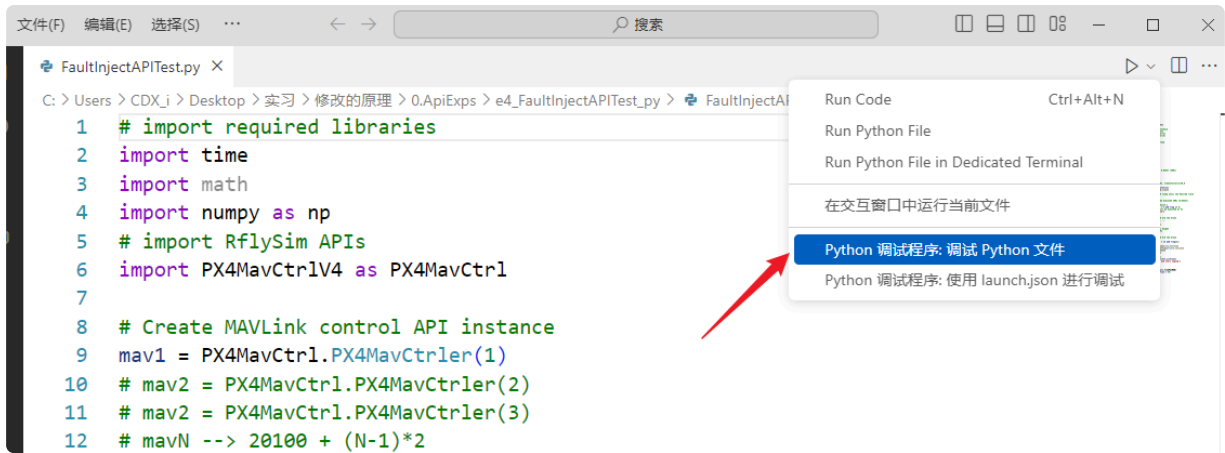
5.2. 选作实验（VS Code调试运行）

准备工作：

- 先确保已经按 [RflySimAPIs/1.RflySimIntro/2.AdvExps/e3.PythonConfig/Readme.pdf](#) 步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，在运行AutoTest.py时，可使用VS Code（或Pycharm等工具）来打开AutoTest.py文件，并阅读代码，修改代码，调试执行等。

扩展实验：

- 请自行使用VS Code阅读AutoTest.py源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。



```
1 # import required libraries
2 import time
3 import math
4 import numpy as np
5 # import RflySim APIs
6 import PX4MavCtrlV4 as PX4MavCtrl
7
8 # Create MAVLink control API instance
9 mav1 = PX4MavCtrl.PX4MavCtrl(1)
10 # mav2 = PX4MavCtrl.PX4MavCtrl(2)
11 # mav2 = PX4MavCtrl.PX4MavCtrl(3)
12 # mavN --> 20100 + (N-1)*2
```

Run Code Ctrl+Alt+N

Run Python File

Run Python File in Dedicated Terminal

在交互窗口中运行当前文件

Python 调试程序: 调试 Python 文件

Python 调试程序: 使用 launchjson 进行调试

6. 参考资料

1. 无

7. 常见问题

Q1: 无

A1: 无