

实验名称及目的

实验名称

自动测试平台使用单机多实例自动化测试。

实验目的

掌握自动测试平台单机多实例自动化测试的基本结构和使用流程。

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标准库模块和自定义的无人机控制模块。
2. 添加当前目录到系统路径：通过`sys.path.append(os.getcwd())`确保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类型、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`列表创建了多个PX4MavCtrl实例，每个实例代表一个无人机。
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：`AutoMavCtrl.InitMavAutoEnv`函数用于初始化无人机的自动控制环境。
7. 启动无人机监控线程：调用`AutoMavCtrl.MavMonitor`启动无人机监控线程。
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.PX4MavCtrler(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，包括无人机列表、配置和地图信息。
- `AutoMavCtrl.MavMonitor()`: 启动无人机监控线程，监控无人机的状态。
- `AutoMavCtrl.AutoMavCtrler(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，监控仿真环境的状态。

多无人机配置示例：

代码中通过`conf`和`mav`列表配置了两架四旋翼无人机（Quadcopter）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`）。

实验效果

通过本实验可以对单机多实例进行自动化测试。

文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\7.RflySimPHM\2.AdvExps\e5_SingFrameMultiInsExp](#)

文件夹/文件名称	说明
AutoTest.py	自动测试脚本
db.json	数据库文件
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
Python38Run.bat	Python程序执行脚本。

运行环境

序号	软件要求	硬件要求	
		名称	数量 (个)
1	Windows 10及以上版本	笔记本/ 台式电脑①	1
2	RflySim工具链		
3	Visual Studio Code		
**： **推荐配置请见： HowToInstall.pdf (rflsim.com)			

实验步骤

5.1. 必做实验

Step 1： 配置仿真参数

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。在本文件中设置为单机多实例模式；其中conf的实例数量设置为2。

打开db.json，可以看到在“TEST_CASE”关键字后面的测试用例设置为“1； 2”，其中分号为多机测试模式，此意为1号飞机测试测试用例1,2号飞机测试测试用例2。

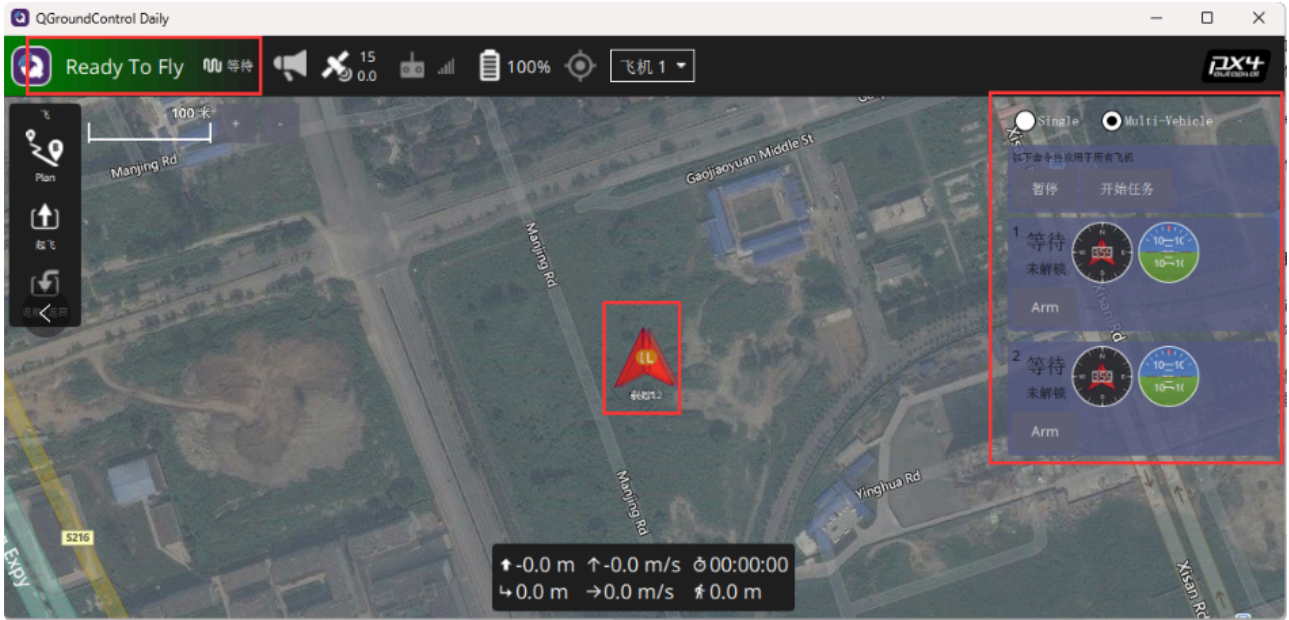
Step 2： 运行py文件开始仿真

在文件夹下，双击Python38AdminRun.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\e5_SingFrameMultiInsExp>python AutoTest.py
```

可以看到本实验初始了两架四旋翼，并自动化测试了1号四旋翼的测试用例1和2号四旋翼的测试用例2，并自动化记录了测试数据，在data\single\mInstance\Quadcopter文件夹下的TestCase_1_2。





Step 3: 仿真结束后检查数据

等待测试结束，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）

Date	Frame	CaseID	S/HITL	TestInfo	DataPath	Normal?
2024-08-08 17:54:35	Quadcopter	1	SITL	0s: Armed! 5s: Send fixed-point flight command, fly target position [0.0, 0.15] 15s: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0, 1.0] 25s: Exit test!	data\single\mInstance\Quadcopter\TestCase_2_1\mav2	Yes
2024-08-08 17:54:40	Quadcopter	2	SITL	0s: Armed! 5s: Send fixed-point flight command, fly target position [0.0, 0.15] 15s: Send fault injection command Fault injection type:123544 Fault injection parameters:[5.0] 25s: Exit test!	data\single\mInstance\Quadcopter\TestCase_2_1\mav1	Yes

CaseID	FaultID	CaseDescription	FaultMode	ControlSequence	TestResult
1	123450	Frame: Quadcopter S/HITL: SITL Injection Motor Fault During Normal Flight	Decreased efficiency of motor execution	0s: Armed! 5s: Send fixed-point flight command, fly target position [0.0, 0.15] 15s: Send fault injection command Fault injection type:123450 Fault injection parameters:[1.0, 1.0, 1.0, 1.0] 25s: Exit test!	Is Fall: No Fall Time: No Fall Vel: No Fall Energy: No Fallsafe Trigger: No Flight Status After Fault Injection: Violent rolling Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [1.27949273 3.59681607 18.57317039] Failure Safety Score: 0.14 Failure Safety Level: Catastrophic
2	123544	Frame: Quadcopter S/HITL: SITL Injection Magnetometer noise During Normal Flight	Magnetometer noise interference	0s: Armed! 5s: Send fixed-point flight command, fly target position [0.0, 0.15] 15s: Send fault injection command Fault injection type:123544 Fault injection parameters:[5.0] 25s: Exit test!	Is Fall: No Fall Time: No Fall Vel: No Fall Energy: No Fallsafe Trigger: No Flight Status After Fault Injection: Smooth flight Deviation From Expected Speed After Fault Injection: None Deviation From Expected Position After Fault Injection: [0.05646704 5.0108593 4.28103081] Failure Safety Score: 0.312 Failure Safety Level: Catastrophic

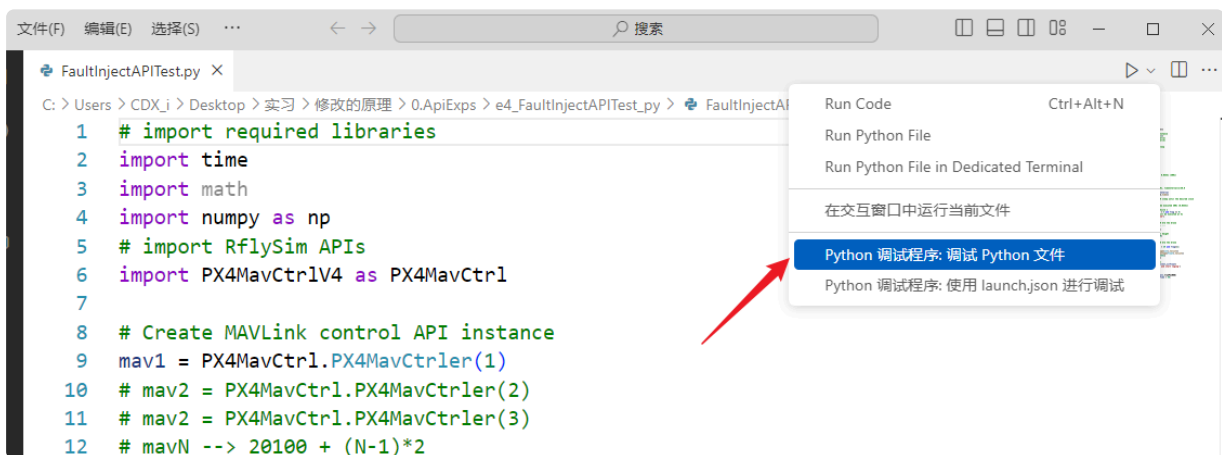
5.2. 选作实验（VS Code调试运行）

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。



参考资料

1. 无

常见问题

Q1: 无

A1: 无