

# 1. 实验名称及目的

## 1.1 实验名称

声音视觉传感器融合实验

## 1.2 实验目的

掌握声音与视觉传感器融合识别

## 1.3 关键知识点

由于默认的音频引擎（如 Unity/UE 自带的空间音频或操作系统层面的立体声 API）在当前仿真平台上不被支持，因此必须自己获取原始麦克风信号，这里改用了基于 ITD

（Interaural Time Difference，双耳时差）原理的声音传感器。该传感器只输出水平面上的声源偏向量（左右声像），等价于一个“2D 音频声像摆位器”，即通过左右声道的时间差或强度差来反映声源的水平角度，而不涉及上下或距离的空间化。

## 1. ITD（Interaural Time Difference）原理

### 1. 基本概念

- 人耳根据声波到达左右耳的时间差来判断声源在水平面的方位。
- 时间差  $\Delta t$  与声源与听者构成的几何角度  $\theta$  存在对应关系：

$$\Delta t = \frac{d \sin \theta}{c}$$

其中  $d$  是双耳间距， $c$  是声速， $\theta$  是水平偏航角。

### 2. 在传感器中的实现

- 两个麦克风采样同步音频流，计算短时帧内的互相关（cross-correlation）峰值对应的延迟。
- 将检测到的  $\Delta t$  转换成角度值：

$$\theta = \arcsin\left(\frac{c \Delta t}{d}\right)$$

- 输出一个标量  $\theta$  (例如  $-90^\circ \dots +90^\circ$  范围内)。
- ### 3. “2D 音频声像摆位器” 的含义
- **二维平面**
    - 仅在水平 (左右) 维度给出声源位置, 不涉及上下 (俯仰) 或前后距离。
  - **声像 (Panning)**
    - 将上一步算出的  $\theta$  直接映射到左右声道的增益比例:
      - $\theta = 0^\circ$  时, 左右声道增益相等 (二者声像在中间)。
      - $\theta > 0^\circ$  时, 声像偏向右声道;  $\theta < 0^\circ$  时偏向左声道。
  - **数据接口**
    - 输出往往是一个归一化的值  $p = \theta/90^\circ$ , 范围  $[-1,1]$ , 可以直接驱动音频引擎或可视化仪表 (如左右声道电平条、雷达扇形指示等)。

## 2. 与视觉和飞控结合的流程

### 1. 持续采样

- 后台线程不断从麦克风对读入音频帧, 计算  $ITD \rightarrow \theta \rightarrow p$ 。

### 2. 生成控制信号

- 将归一化的  $p$  或角度  $\theta$  传给飞控模块, 替代视觉丢失时的跟踪信息。

### 3. 可视化展示

- 用水平扇形模拟声源方向, 直观反馈声像位置。

### 4. 数据记录

- 同步把每次算出的  $\theta$  存入全局 `angle_list`, 后续通过 `save_list()` 一并落盘, 以便和视觉角度、真实角度做误差分析。

## 3. 主程序解析

本工程将视觉检测、音频定位与 MAVLink 飞行控制有机结合, 通过多线程架构并行执行

### 初始化

- 加载配置 `vis.jsonLoad()`, 启动共享内存图像捕获
- 实例化并初始化两架飞机的 MAVLink 通信 (`MavList`)
- 设置 UE4 窗口分辨率和帧率

### 检测与跟踪 (`detect_quadrotor`)

- 以固定 30 Hz 循环:
  - 获取最新渲染帧

- 调用 `DetectQaudrotor()` 获取图像检测结果
- 若视觉不可用，则切换到音频控制 `Controller_audio()`
- 否则根据偏移量调用 `Controller()` 生成 yaw 控制命令
- 最终使用 `mav.SendVelFRD(...)` 发送速度指令

### 目标飞机巡航 ( `CtrlTargetDrone` )

- 可选线程：按预设路径点队列飞行，用于作为“被跟随”目标

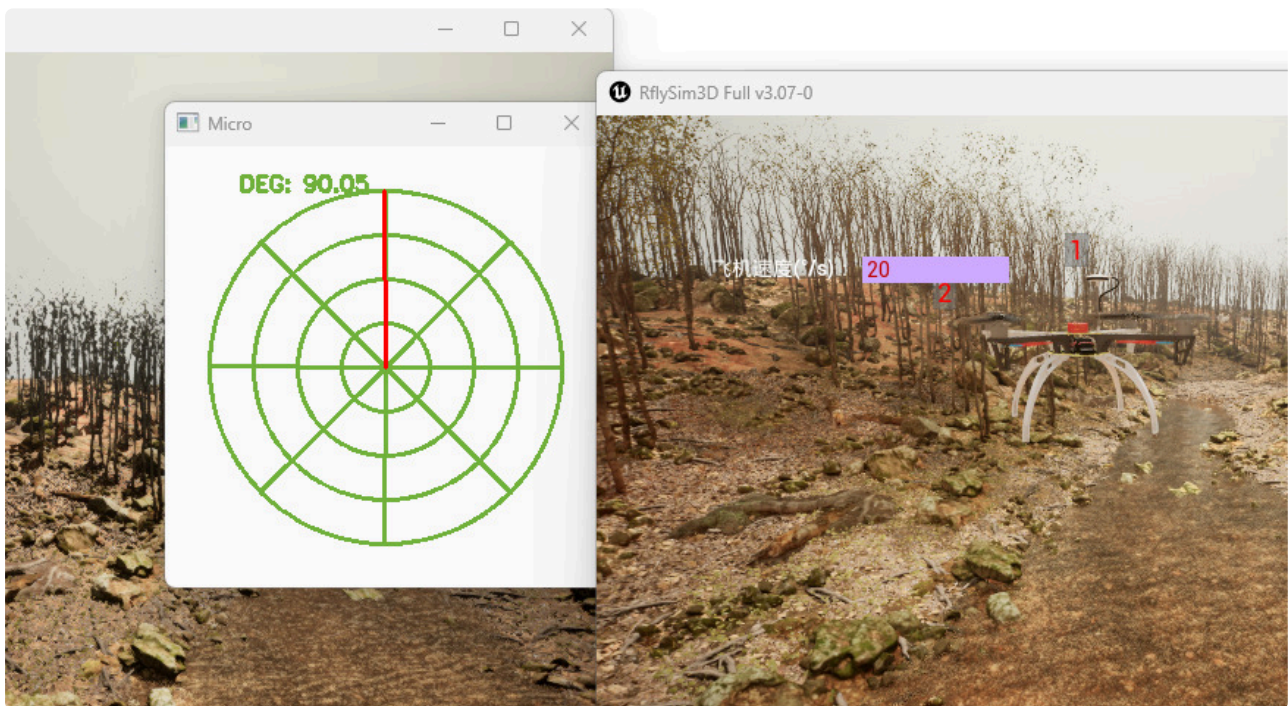
### 时间戳记录 ( `detected_change` & `time_record` )

- 监控 `detected` 标志位变化，记录检测到/丢失的时间
- 最终通过 `save_list()` 响应按键“s”保存到文件

### 音频雷达可视化 ( `audio_detect_show` )

- 在独立窗口中绘制仿雷达背景和动态扫描线
- 实时展示 `uerecord.angle` 作为声源方向

## 2. 实验效果



## 3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\8.RflySimVision\3.CustExps\e12\_ObjDetectaudio-visual

文件夹/文件名称	说明

## 4.1 软件要求

Windows 10及以上版本；RflySim工具链；yolov5。

①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4\_fmuv6x\_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台。

①：推荐配置请见：<https://rflysim.com/>

## 5. 实验步骤

### 4.1 声视觉传感器融合实验

#### Step 0 下载额外文件完成实验配置

##### 1. 实验主程序

通过如下链接下载主程序文件quadrotor\_follow.zip并解压到本实验目录，文件结构如下  
链接：<https://pan.baidu.com/s/1GBJHCIDDSGwrij8YR6Fb4A?pwd=qa6x> 提取码: qa6x

文件夹/文件名称	说明
img/	存放示例或调试用的图片资源

文件夹/文件名称	说明
<code>models/</code>	存放模型配置文件
<code>runs/</code>	YOLOv5 推理结果输出目录（如 <code>runs/detect/...</code> ）
<code>utils/</code>	通用工具脚本，如绘图、数学工具、辅助函数
<code>yolo/</code>	本地 YOLOv5 源代码
<code>best.pt</code>	训练后的检测模型权重
<code>yolov5s.pt</code>	YOLOv5 官方预训练模型权重
<code>Config.json</code>	配置文件，包含相机参数、UE4 地址、PX4MavCtrl 参数等
<code>detect_ball.py</code>	封装了 YOLOv5 推理接口，用于目标检测
<code>getImage.py</code>	从 RflySim 共享内存或 Socket 中获取渲染帧的脚本
<code>uerecord.py</code>	音频采集与声源定位模块，实现 ITD 声像估计和卡尔曼滤波平滑
<code>quadrotor_follow.py</code>	主脚本，集成视觉检测、音频定位和 PX4 飞控控制逻辑
<code>quadrotor_follow.bat</code>	软件在环一键启动脚本
<code>Python38Run.bat</code>	兼容 Python 3.8 的批处理启动脚本，用于初始化环境并运行主脚本

## 2. 实验环境

通过如下链接下载启用了 ITD 的完整 RflySim3D 工程（限 UE5）RflySimUE5Audio.zip，包括搭载声音传感器的三维场景和飞机模型，将该工程解压到 RflySim 安装目录下，最终目录例如：`C:\PX4PSP\RflySimUE5Audio`

链接: <https://pan.baidu.com/s/1oqJ0nRJsejUdffwyl8y6NA?pwd=ke9j> 提取码: ke9j

同时将本实验目录下的 ForestMap.png 和 ForestMap.txt 拷贝到平台安装目录的对应位置 `C:\PX4PSP\CopterSim\external\map`

## 3. 额外 python 依赖

双击运行 `Python38Run.bat`（确保本目录下有 requirements.txt 文件），通过如下命令安装额外的 python 库

```
1 | pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

若需安装到无网络环境，则通过如下命令预先下载

```
1 | pip download -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple -d ./whe
```

然后将得到的wheelhouse文件夹通过如下命令安装

```
1 | pip install --no-index --find-links=./wheelhouse -r requirements.txt
```

## Step 1 启动软件在环

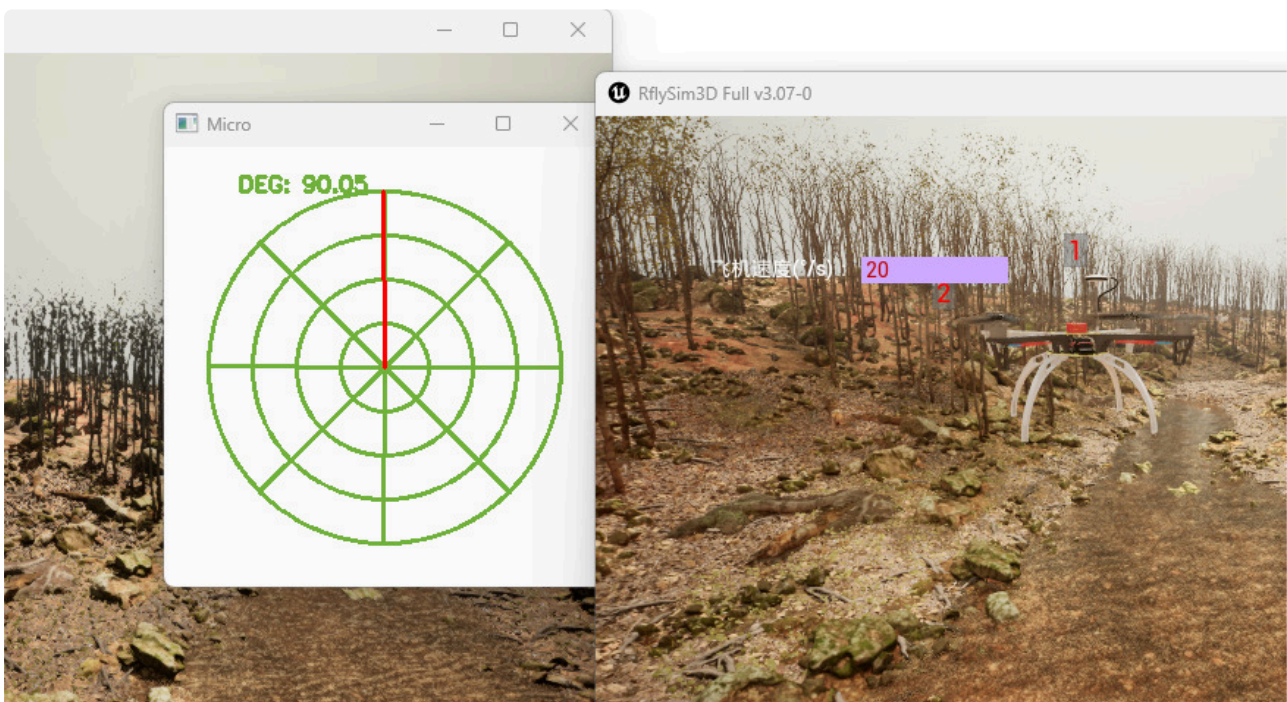
双击运行quadrotor\_follow实验目录下的quadrotor\_follow.bat文件，在特定的声音场景中启动2架飞机的软件在环仿真；包括目标飞机和监察飞机。

## Step 2 运行实验主程序

在本实验目录下，双击 [Python38Run.bat](#)，打开默认集成好的python环境，在该环境下运行quadrotor\_follow.py文件，输入 `python quadrotor_follow.py`

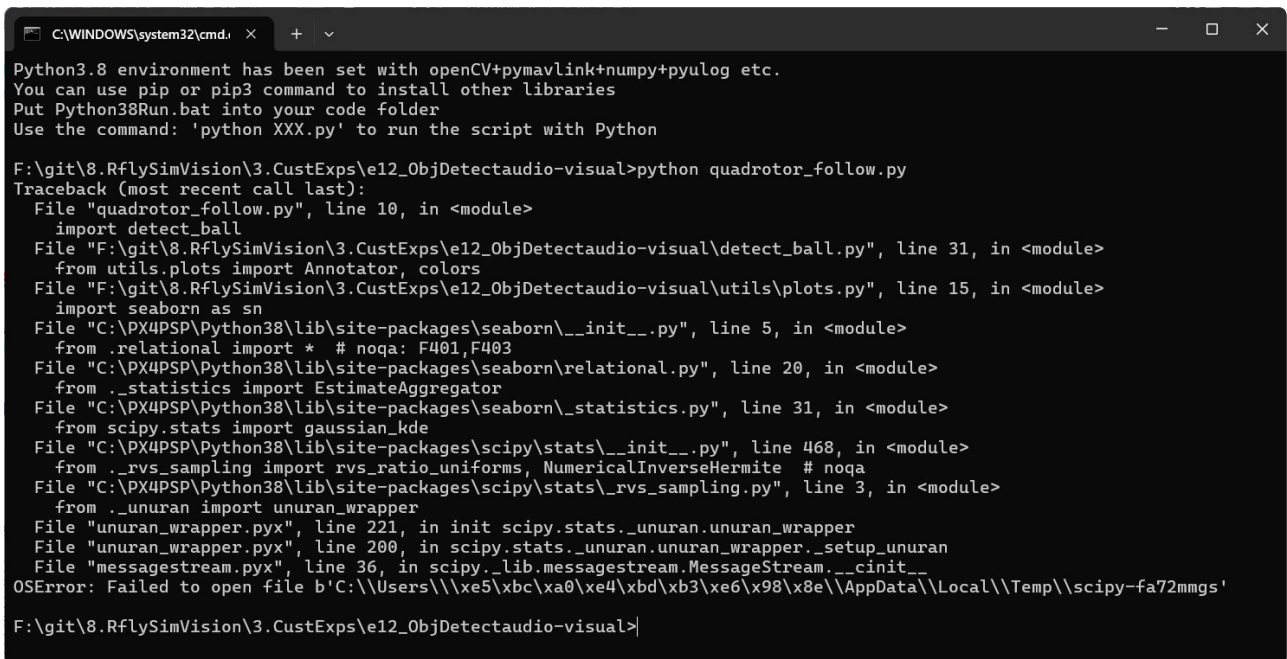
## Step 3 观察结果

两架飞机起飞后，监察飞机会根据声音雷达调整偏航转向目标飞机（注意要保持RflySim3D窗口在最前面才能收到声音）



# 7. 常见问题

Q1:如果运行时出现如下报错



```
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

F:\git\8.RflySimVision\3.CustExps\e12_ObjDetectaudio-visual>python quadrotor_follow.py
Traceback (most recent call last):
  File "quadrotor_follow.py", line 10, in <module>
    import detect_ball
  File "F:\git\8.RflySimVision\3.CustExps\e12_ObjDetectaudio-visual\detect_ball.py", line 31, in <module>
    from utils.plots import Annotator, colors
  File "F:\git\8.RflySimVision\3.CustExps\e12_ObjDetectaudio-visual\utils\plots.py", line 15, in <module>
    import seaborn as sn
  File "C:\PX4PSP\Python38\lib\site-packages\seaborn\__init__.py", line 5, in <module>
    from .relational import * # noqa: F401,F403
  File "C:\PX4PSP\Python38\lib\site-packages\seaborn\relational.py", line 20, in <module>
    from .statistics import EstimateAggregator
  File "C:\PX4PSP\Python38\lib\site-packages\seaborn\statistics.py", line 31, in <module>
    from scipy.stats import gaussian_kde
  File "C:\PX4PSP\Python38\lib\site-packages\scipy\stats\__init__.py", line 468, in <module>
    from ._rvs_sampling import rvs_ratio_uniforms, NumericalInverseHermite # noqa
  File "C:\PX4PSP\Python38\lib\site-packages\scipy\stats\_rvs_sampling.py", line 3, in <module>
    from ._unuran import unuran_wrapper
  File "unuran_wrapper.pyx", line 221, in init scipy.stats._unuran.unuran_wrapper
  File "unuran_wrapper.pyx", line 200, in scipy.stats._unuran.unuran_wrapper._setup_unuran
  File "messagestream.pyx", line 36, in scipy.lib.messagestream.MessageStream.__cinit__
OSError: Failed to open file b'C:\\Users\\...\\AppData\\Local\\Temp\\scipy-fa72mngs'

F:\git\8.RflySimVision\3.CustExps\e12_ObjDetectaudio-visual>
```

A1:将系统临时目录改为纯英文路径

## 原因

Windows 默认临时目录位于 `%USERPROFILE%\AppData\Local\Temp`，如果用户文件夹包含中文或其他非 ASCII 字符，SciPy 在临时目录中解包 C 扩展文件时可能失败 [github.com](https://github.com) [blog.csdn.net](https://blog.csdn.net)。

# 5. 实验步骤

1. 新建一个只含英文路径的文件夹，例如 `D:\TempSciPy`。
2. 打开 **系统属性** → **高级** → **环境变量**。
3. 在 **用户变量** 和 **系统变量** 中，将 `TEMP` 和 `TMP` 的值都改为 `D:\TempSciPy` [knowledge.civilgeo.com](https://knowledge.civilgeo.com) [99rdp.com](https://99rdp.com)。
4. 重启电脑或重新登录，使环境变量生效。

## 结果

SciPy 将在新的英文路径下解包和加载 C 扩展文件，避免因路径编码导致的 “Failed to open file” 错误 [superuser.com](https://superuser.com)。