1.实验名称及目的

1.1实验名称

外部传感器之声音定位实验（仅限完整版及以上版本）

1.2实验目的

1.3关键知识点

例程功能API

UE控制接口：python库文件 UE4CtrlAPI.py： RflySimSDK/html/UE4CtrlAPI_8py.html

UE控制命令发送接口sendUE4Cmd：
classRflySimSDK_1_1ue_1_1UE4CtrlAPI_1_1UE4CtrlAPI.html#sendUE4Cmd

载具控制接口：python库文件 PX4MavCtrlV4.py：
RflySimSDK/html/PX4MavCtrlV4_8py.html

MAVLink消息侦听启用接口InitMavLoop：
RflySimSDK/html/PX4MavCtrlV4_8py.html

Offboard模式启用接口initOffboard：
RflySimSDK/html/PX4MavCtrlV4_8py.html#initOffboard

定点位置控制接口SendPosNED：
RflySimSDK/html/PX4MavCtrlV4_8py.html#SendPosNED

降落控制接口sendMavLand：
RflySimSDK/html/PX4MavCtrlV4_8py.html#sendMavLand

关键知识点1：声音传感器的实现原理

这个声音传感器是基于UE引擎的内置功能实现的，类似于游戏中的听声辩位功能，它的输出是音频采集设备（耳机）直接能听到的声音数据，传感器与场景中的声源离得近就声音大，离得远就声音小，并且左右耳有一些细微差别（时间差和强度差）来构建立体声（与左

右眼视差构成3D图像的原理类似），具体的计算模型可以参考头部相关传递函数（Head-Related
Transfer Function, HRTF）。

关键知识点2：VoiceCapAPIDemo.py关键代码解析

1. 初始化部分

ue = UE4CtrlAPI.UE4CtrlAPI()

mav = PX4MavCtrl.PX4MavCtrler(1)

vis = VisionCaptureApi.VisionCaptureApi()

这些行创建了控制UE4（Unreal Engine
4）、MAVLink通信和视觉捕捉API的实例。UE4CtrlAPI 和 VisionCaptureApi
类用于与虚拟环境交互，而 PX4MavCtrler
用于建立MAVLink通信，可能与无人机控制有关。

ue.sendUE4Cmd('r.setres 1280x720w',0)

ue.sendUE4Cmd('t.MaxFPS 30',0)

这些命令向UE4发送指令，设置窗口分辨率为1280x720并将最大帧率限制为30
FPS，以节省资源。这不会影响数据捕捉的分辨率，因为分辨率是通过配置文件在
VisionCaptureApi中设置的。

vis.jsonLoad()

isSuss = vis.sendReqToUE4()

这两行代码通过jsonLoad函数加载传感器的配置文件（Config.json），并通过
sendReqToUE4函数向RflySim3D发送数据捕获请求。

2. UDP数据接收配置

UDP_IP = "127.0.0.1"

UDP_PORT = 28003

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

sock.bind((UDP_IP, UDP_PORT))

这一部分配置了UDP通信的IP地址和端口，用于接收UE4发来的音频数据。127.0.0.1表示本地计算机，28003为预定义的端口，需与UE4中设置的端口一致。sock.bind将套接字绑定到指定地址和端口，使程序能够监听从该端口发来的数据。

3. 数据接收与缓冲处理

buffer_size = 204800

sample_rate = 48000

channels = 2

data_buffer = b""

elapsed_time = 0

- buffer_size：定义了接收数据的缓冲区大小。值为204800字节，根据音频数据的大小进行调整。
- sample_rate：假设的音频采样率为48kHz，即每秒采样48000次。
- channels：音频的声道数，假设为2，即立体声。
- data_buffer：一个空字节串，用于存储接收的音频数据。
- elapsed_time：记录累积接收到的数据时间，以便判断是否达到3秒。

4. 主接收循环

while True:

data, addr = sock.recvfrom(buffer_size)

data_buffer += data

elapsed_time += len(data) / (sample_rate * channels * 2)

- sock.recvfrom(buffer_size)：从UDP套接字接收数据，最大接收buffer_size字节的数据，并将其存储到data中。
- data_buffer += data：将接收到的数据追加到data_buffer中。
- elapsed_time：计算并累积接收的时间，公式为len(data) / (sample_rate * channels *
  2)，这里的2是因为每个样本占用2字节（16位），这确保程序正确计算累积的时间长度。

5. 保存和绘制音频数据

if elapsed_time >= 3:

with wave.open("output_new.wav", "wb") as wave_file:

wave_file.setnchannels(channels)

wave_file.setsampwidth(2)

wave_file.setframerate(sample_rate)

wave_file.writeframes(data_buffer)

print(f"lendata:{len(data_buffer)}")

plot_stereo_wav_waveform2("output_new.wav")

break

当累计接收时间达到3秒时，程序将数据保存为WAV文件并绘制波形图。

- wave.open("output_new.wav", "wb")：打开一个WAV文件以写入模式（wb）创建文件。
- wave_file.setnchannels(channels)：设置声道数为2（立体声）。
- wave_file.setsampwidth(2)：设置采样宽度为2字节（16位）。
- wave_file.setframerate(sample_rate)：设置采样率为48kHz。
- wave_file.writeframes(data_buffer)：将接收到的音频数据写入文件。

plot_stereo_wav_waveform2("output_new.wav")调用了前面定义的plot_stereo_wav_waveform2函数，加载保存的WAV文件并绘制左右声道波形图。

6. 重置数据和时间

data_buffer = b""

elapsed_time = 0

在3秒音频数据保存和绘制后，重置data_buffer和elapsed_time，为下一个3秒数据的接收做准备。

2.实验效果

可以监听到蓝图飞机的音频数据并输出左右声道的声音序列


RflySim3D Full v3.03-0
303.55462
飞机速度(°/s)：
20

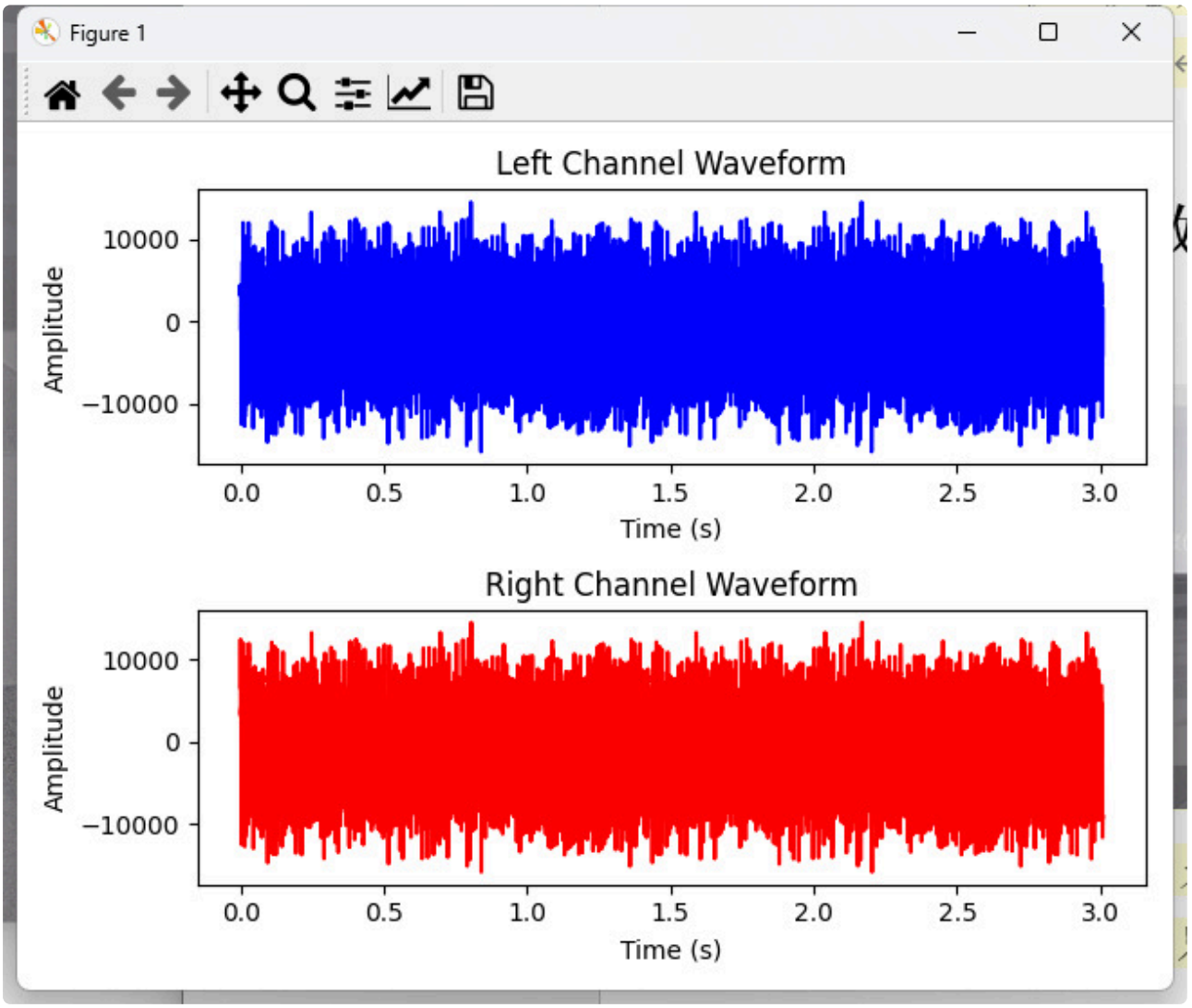


Figure 1
Left Channel Waveform
Amplitude
10000
0
−10000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Time (s)
Right Channel Waveform
Amplitude
10000
0
−10000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Time (s)

注意，因为UE只能固定输出每一帧的声音直接到音频设备，这里的传感器没办法动态装载，暂时只能预先装载到场景中的特定位置

3.文件目录

例程目录：
[安装目录]\RflySimAPIs\4.RflySimModel\3.CustExps\e7_ExternalSensors\e2_Voice_PyRec

文件夹/文件名称	说明
Python38Run.bat	环境启动脚本
ue4.bat	RflySim3D启动脚本
VoiceCapAPIDemo.py	例程功能主程序

4.运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；VS 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

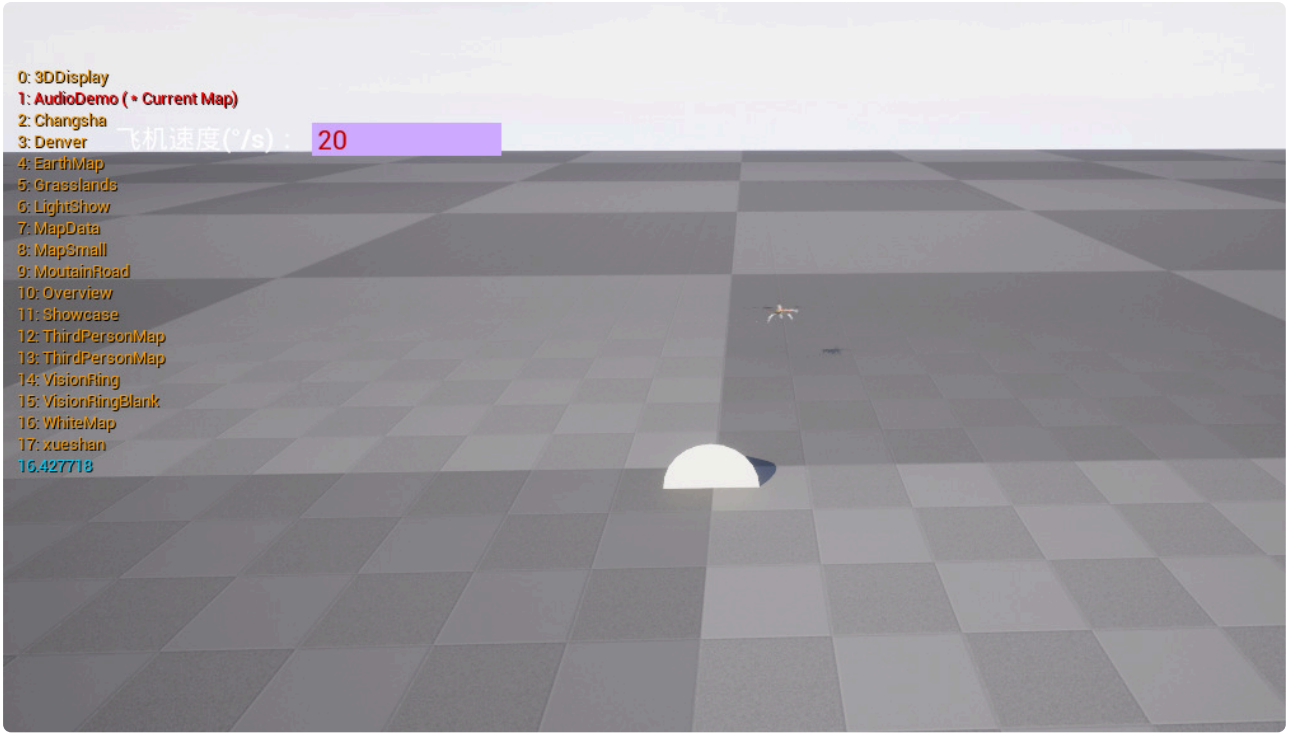
声音传感器实验（选做）

Step 1：启动RflySim3D

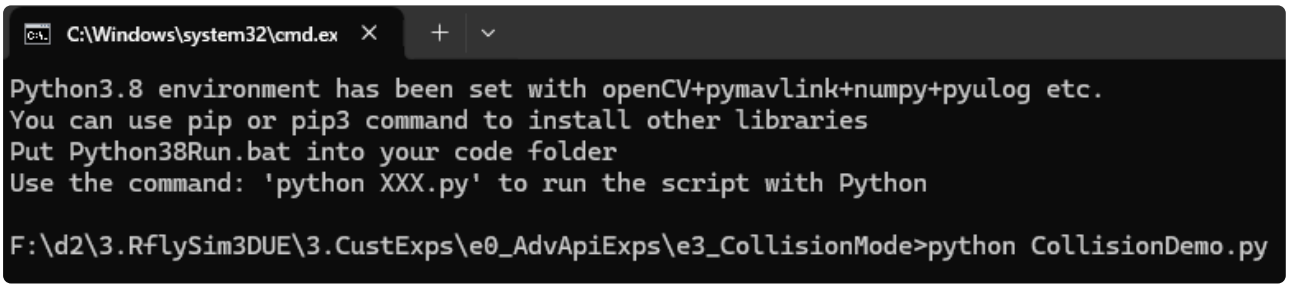
双击运行ue4.bat打开一个RflySim3D

Step 2：切换到带有声源的场景

在RflySim3D中按M键切换到带有声源的场景

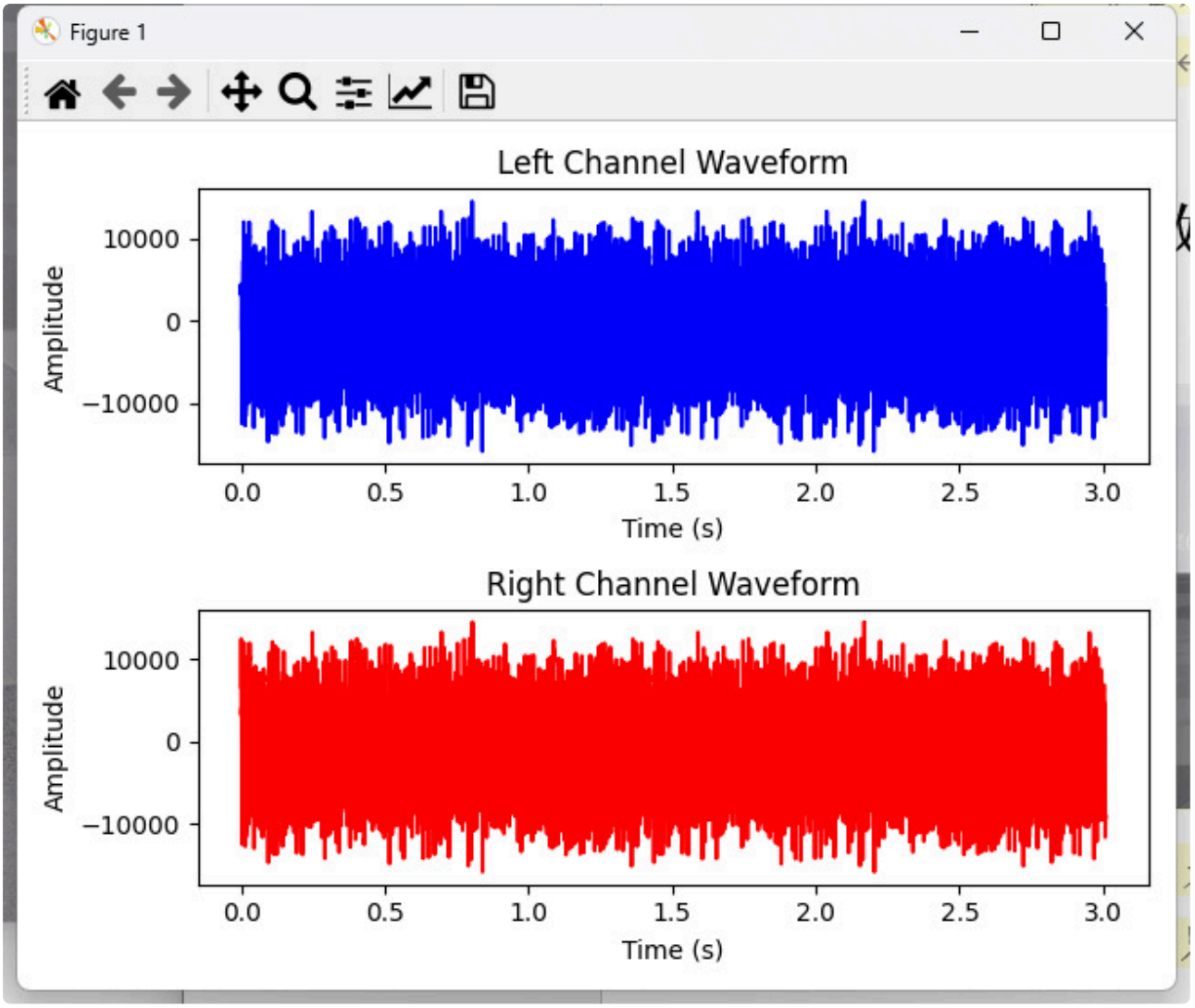


Step 3：运行python脚本加载声音传感器



在文件夹下，双击38Run.bat，打开集成好的环境，输入 VoiceCapAPIDemo.py，回车运行。

可以监听到蓝图飞机的音频数据并输出左右声道的声音序列



1. **Vscode调试运行实验（选做）**

准备工作：

- 先确保已经按RflySimAPIs\1.RflySimIntro\2.AdvExps\e3_Config\Readme.pdf步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义环境。
- 其他步骤与上文相同，在运行文件时，可使用VS Code（或Pycharm等工具）来打开文件文件，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阅读例程中的源码，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。
- 请尝试替换更逼真的声音资源，以全面利用声音传感器的功能

6.参考资料

[1]. [安装目录]\RflySimAPIs\3.RflySim3DUE\API.pdf

[2]. UE4性能调试分析常用方法 - 知乎 : https://zhuanlan.zhihu.com/p/273608458

[3].

7.常见问题

Q1：***

A1：***