



西北工业大学

本科毕业设计（论文）

题 目 四旋翼无人机编队飞行数字孪生系统的设计与实现

专业名称 电子商务

学生姓名 LAI HUI SHAN

指导教师 钟冬

毕业时间 2022 年 7 月 31 日

摘 要

随着人们对四旋翼无人机的深入研究和发​​展,无人机最早从主要作为军事用途为主,到现在演变成在商业发展和消费娱乐都能被广泛应用的重要工具,并且能够支持诸多领域的解决方案,如建筑、能源、农业、救灾等领域。其中,四旋翼无人机由于成本低廉、操作简单和高效安全的特点,更宜应用于多种场合。

数字孪生则是一个以数据化形式通过能准确模拟和映射物理对象行为、属性和规则等的虚拟模型。数字孪生的研究对象一般配备了相关的传感器,从而达到在测试中产生和物理对象相关的性能数据(如风向、天气状况、能量输出等),使其能在虚拟模型中实现更为准确模拟,最后再通过虚拟模型来进行研究和改进产品的性能,进而提供更加高效和智能的服务。

为了达到四旋翼无人机在编队飞行过程中的虚实交互、实时响应和精确控制等来实现如编队无人机灯光秀的模拟展示,以通过虚幻四引擎开发工具开发一款四旋翼无人机编队飞行数字孪生的模拟应用系统作为任务需求,实现基于数字孪生技术的四旋翼无人机编队飞行全局整体动态演进的模拟。同时,搭建四旋翼无人机编队飞行数字孪生系统架构,使其系统有队形编辑界面,可以编辑不同队形,具有统一数据库和信息交互的接口模块,并对四旋翼无人机编队飞行数字孪生系统的功能和意义进行了介绍。最后,从几何、物理、行为和规则等四个方面融合构建四旋翼无人机编队飞行的数字孪生模型和系统。

关键词: 数字孪生, 四旋翼无人机, 编队飞行

ABSTRACT

With the in-depth research and development of UAV, UAV has evolved from mainly used for military purposes at the earliest to an important tool that can be widely used in commercial development and consumer entertainment and can support solutions in many fields, such as construction, energy, agriculture, disaster relief and so on. Among them, quadrotor UAV due to low cost, simple operation and high efficiency and safety characteristics, more suitable for application in a variety of occasions.

Digital twin is a virtual model that can accurately simulate and map the behavior, properties and rules of physical objects in digital form. Digital twin study general equipped with relevant sensors, so as to achieve in the test performance data related to physical objects (such as wind and weather conditions, power output, etc.), make its can achieve more accurate simulation in virtual model, and then through the virtual model to study and improve the performance of the products, thus provides a more efficient and intelligent service.

In order to achieve virtual/real interaction, real-time response and precise control of quadrotor UAV during formation flight to achieve simulation demonstration such as cluster UAV light show, to develop a simulation application platform for digital twin of quadrotor UAV formation flight through Unreal 4 engine development tool as mission requirements, Realize the simulation of the global dynamic evolution of quadrotor UAV formation flight based on digital twin technology. At the same time, the digital twinning system architecture of formation flight of quadrotor UAV is built, so that the system has formation editing interface, can edit different formation, has unified database and information interaction interface module, and the function and significance of the digital twinning system of formation flight of quadrotor UAV are introduced. Finally, the digital twin model and system of quadrotor UAV formation flight are constructed from geometry, physics, behavior and rules.

KEY WORDS: digital twin, UAV, formation flight

目 录

第一章 绪论	6
1.1 课题研究背景和意义	6
1.1.1 数字孪生的概念.....	6
1.1.2 四旋翼无人机编队概念.....	7
1.1.3 研究意义.....	8
1.2 国内外研究现况	8
1.2.1 国内外四旋翼无人机研究现状.....	8
1.2.2 国内外数字孪生系统研究现状.....	11
1.3 研究内容以及章节安排	13
第二章 四旋翼无人机编队数字孪生系统的总体方案	15
2.1 概述	15
2.2 设计结构	15
2.3 四旋翼无人机的设计	16
2.4 虚拟环境	17
2.4.1 重力.....	17
2.4.2 地球磁场.....	17
2.4.3 空气密度和气压.....	18
2.5 物理引擎和传感器	18
2.7 视觉呈现	19
2.8 无人机编队实现	20
2.9 小结	22
第三章 数字孪生系统的设计与实现	24
3.1 概述	24
3.2 计算机软硬件配置	24
3.3 Airsim 安装与环境配置.....	25
3.3.1 虚幻四引擎的安装.....	25
3.3.2 AirSim 的安装.....	26
3.4 Airsim 环境测试.....	27
3.5 模拟环境的搭建	28
3.6 操控无人机飞行	30
3.7 API 接口程序与 python 的环境搭建	31

西北工业大学本科毕业设计（论文）

3.7.1 Anaconda 环境搭建.....	31
3.7.2 PyCharm 环境搭建	32
3.7.3 代码运行测试.....	33
3.8 setting.json 文件	34
3.8.1 多无人机生成代码实现.....	34
3.8.2 远程通信代码的实现.....	35
3.9 远程通信接口模块	36
3.10 用户界面和环境调整	37
3.11 数据采集和数据库.....	38
3.12 小结	42
第四章 效果测试与评估	43
4.1 系统测试效果	43
4.2 目标完成度评估	48
第五章 总结和展望	50
参考文献	51
致 谢	53
毕业设计小结	54
附 录	55

第一章 绪论

1.1 课题研究背景和意义

1.1.1 数字孪生的概念

数字孪生是一个以数据化形式通过能准确模拟和映射物理对象行为、属性和规则等的虚拟模型。数字孪生的研究对象一般配备了相关的传感器，从而达到在测试中产生和物理对象相关的性能数据（如风向、天气状况、能量输出等），使其能在虚拟模型中实现更为准确模拟，最后再通过虚拟模型来进行研究和改进产品的性能，进而提供更加高效和智能的服务^{错误!未找到引用源。错误!未找到引用源。}。

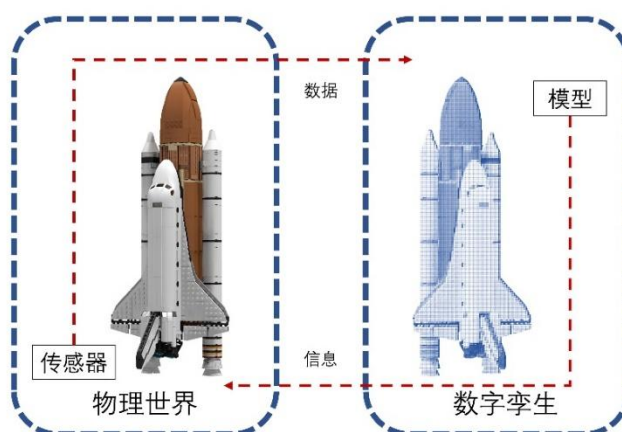


图1-1 数字孪生

数字孪生最早是由密歇根大学教授Michael Grieves所提出的一个概念，他之后扩展了关于数字孪生的概念所跨越的产品的整个生命周期，包括构建、运营、维护和处理，因此数字孪生还包含三种不同的类型。这些类型包括数字孪生原型 (Digital twin prototype)、数字孪生实例 (Digital twin instance) 和数字孪生聚合 (Digital twin aggregate)。数字孪生原型 (DTP) 是用于数字孪生的基础，在理想状态下，这样的一个原型通常是由高保真且数据精确模型组成。其次，在产品的生命周期中，数字孪生实例 (DTI) 能与物理实体保持连接，收集包含产品过程的参数、控制参数、传感器数据和历史数据等。数字孪生聚合 (DTA) 是多个数字孪生实例 (DTI) 的集合，其数据和信息可用于物理产品的故障预测和机器学习。因此若和相关领域中能够确保模拟数据足够精确并且能够持续更新，在加上具备足够的算

例，数字孪生技术将能够更有效率的测试研究和改进产品的缺陷^{错误!未找到引用源。}。

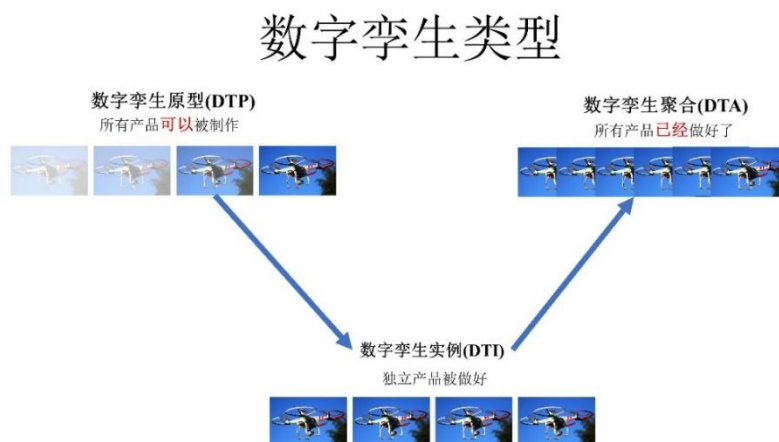


图1-2 数字孪生类型

1.1.2 四旋翼无人机编队概念

在过去的几十年里，电子元器件和传感器往越来越廉价和微型的方向发展，让四旋翼无人机的应用得到快速的推广和发展。四旋翼无人机是一个具有四个呈十字交叉型螺旋桨的小型飞行器，因为这种小型飞行器具备低惯性和结构简单的特征，且被允许使用一个极其简单的飞行控制系统，意味着这种无人机的建造成本低廉，导致四旋翼无人机在许多领域都具有广泛的实际用途和前景，也因此成为了许多科技企业和科研单位的热门研究项目^{[3][4]}。

随着科技的数字化和智能化使得编队项目能够得到有效的技术支持，因此针对四旋翼无人机编队系统的研究也越来越被重视，无人机编队飞行是指多架无人机在3D空间中以一定的队列飞行。编队在飞行过程中能够维持或改变一定范围内的相对位置，并可以根据外部条件和任务要求进行动态调整，以保持编队的协调性和一致性。编队特征通常具有更强大的数据搜集能力，并且也更能实现更为复杂的任务处理。在军用领域，协同控制多架无人机将能有效提升各种任务的成功率（如军事侦察、电子对抗、干扰等），有效降低人力损耗，也是各军事大国之间争相研究的热门项目；在商用领域，四旋翼无人机编队能在大型庆典中实现编队表演等能力，让所有在场的观众或游客都产生难以忘怀的印象，从而促进旅游业的发展^{错误!未找到引用源。} [5]。

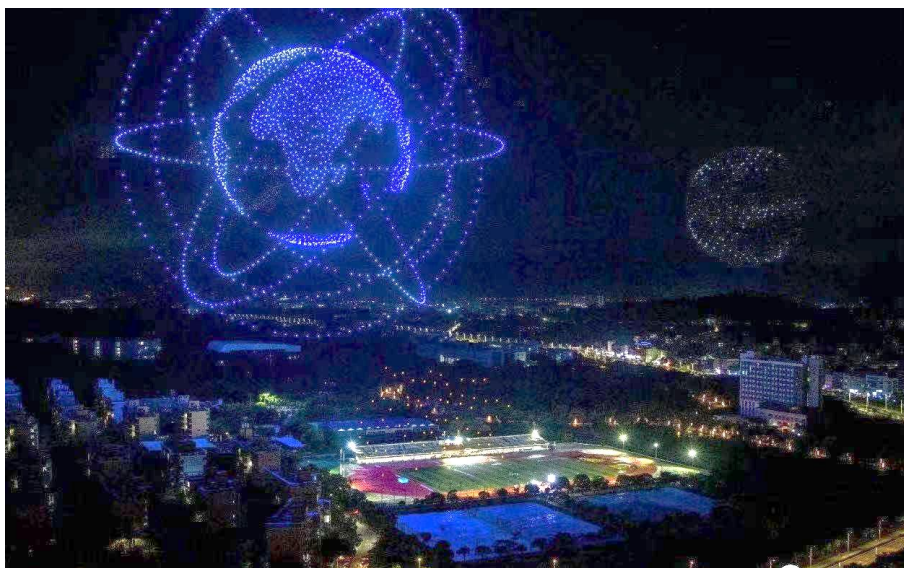


图1-3 珠海无人机灯光秀

1.1.3 研究意义

综上所述，如果我们能够队四旋翼无人机编队飞机进行一个以数字孪生技术来进行研究，将能够以更低的经济效益作为研究基础，研究四旋翼无人机编队如何在数字孪生系统中达到协同控制、自动规避、人机交互、网络通信等功能，用以为物理实体的四旋翼无人机的编队飞行测试时提供完整的思路并达到更搞得精确度^[6]。

1.2 国内外研究现况

1.2.1 国内外四旋翼无人机研究现状

世界上第一个四旋翼飞机是由Breguet兄弟研制的GrroPlane No.1，当时因为基于技术发展原因，四旋翼飞机对比其他飞机具有相当大的缺陷，导致之后很长一段时间四旋翼飞机的发展都停止不前。但是后来在几十年后由于IMU和MEMS系统的成功研发，四旋翼无人机取得了迅速的发展和技术突破。

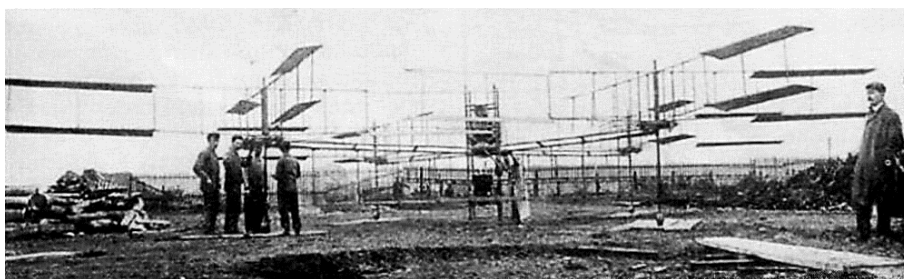


图1-4 GrroPlane No.1

目前在国外具有代表性的四旋翼无人机的研究分别是：

1) 宾夕法尼亚大学的四旋翼无人机

宾夕法尼亚大学的实验室GRASP在蜂鸟四旋翼无人机的基础上加上了室内红外光成像与IMU进行定位和姿态测量，使无人机能在室内是实现更为稳定的飞行，同时使用SLAM技术进行研究，以减少无人机对外部的依赖性。另外，他们还在此基础上完善了多无人机协调控制、目标识别、自动避障等功能。



图1-5 宾夕法尼亚大学的四旋翼无人机

2) 斯坦福大学的四旋翼无人机

为了达到降低碰撞时对四旋翼无人机的螺旋桨的破坏，斯坦福大学在设计无人机系统时采用PC104作为主控器，系统由GPS、超声波、姿态检测元件、视频传输等模块组成，并且能够利用手机Wifi和PPM遥控器对无人机进行控制。他们还设计了高效的控制器，建立了准确的数学模型，用以实现无人机翻转等飞行动作，并在这些基础上利用无人机的设计实现了对地面目标的可靠追踪。

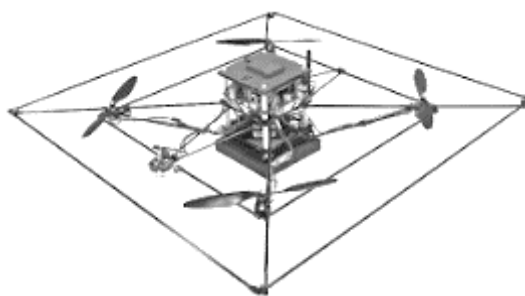


图1-6 斯坦福大学的四旋翼无人机

3) 纽约市立大学城市学院的四旋翼无人机

纽约市立大学城市学院的学者在总结近几年无人机技术的研究基础上设计了被称为CityFlyer，一个能够自主飞行和定位的四旋翼无人机。CityFlyer搭载了2个ARM7微处理器和一个1.6GHz的Atom处理器，使其能够进行IMU融合数据功

能并且对飞行控制算法的实现。



图1-7 纽约市立大学城市学院的四旋翼无人机

而在中国，最早的无人机系统是由西北工业大学与1958年研发的，并且在上个世纪80年代研制出了被命名为“D-4民用无人机系统”，用以实现军用领域如航空拍摄、物理探矿和灾情监视等功能。但是直到2006年大疆公司创办以前民用无人机的发展都非常缓慢，目前在中国主导无人机研究的大学主要有清华大学、西北工业大学、哈尔滨工业大学、国防科技大学、上海交通大学等高校，这些高校都各自研制出了自己的四旋翼无人机样机，技术上也基本达到了国外的先进水平，如今大疆无人机甚至在2021年占据了全世界76%的民用无人机市场销售份额

[6]错误:未找到引用源。



图1-8 大疆无人机

Global market share of consumer and commercial drone manufacturers in March 2021, based on sales volume

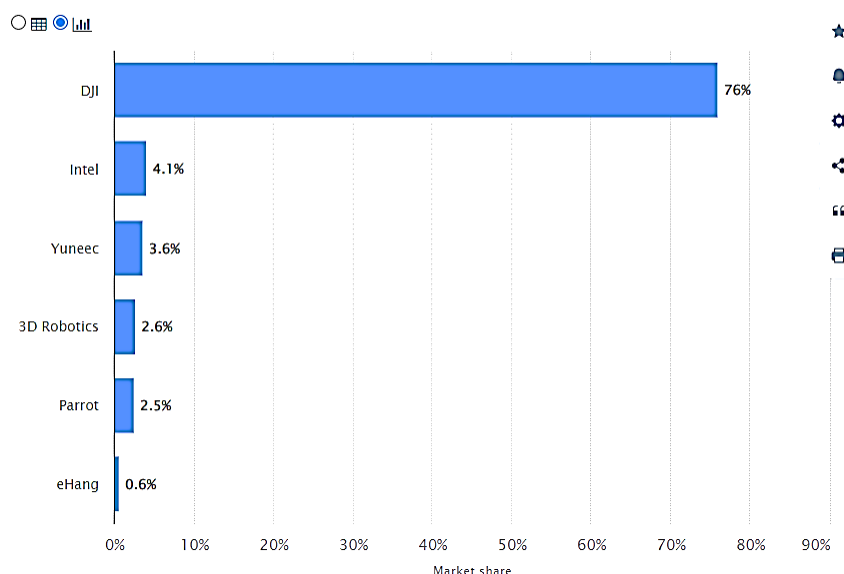


图1-9 2021年各大品牌在全球无人机消费市场的市占率

1.2.2 国内外数字孪生系统研究现状

基于实现无人机飞行的数字孪生技术，在国外，一开始由美国NASA在阿波罗计划中为了实现在轨装配技术的模拟与测试而完善“数字孪生”的概念，并借此在其项目中构建两个参数一样的飞行器，而在地球上运行的飞行器则被称为孪生体。

2003年，美国空军推出的 OCP-HITL 项目，即通过开放式的控制系统来实现硬件在环，用于专门测试和开发无人机飞行控制的算法，但是最早只是用于单架无人机的测试，后来，How等人也基于此项目开发了能实现 8 架飞机的测试系统。

2005年，美国的Steven Rasmussen基于MATLAB、Simulink 和 C++开发了MultiUAV2系统，使其能够实现多无人机协同作战模拟任务和测试。该系统展现了如何实现多架无人机在协同作战时进行任务，并且主要实现了关于飞行器的动力学、任务分配、打击目标、通信、传感、算法等几个模块。

2007年，韩国的建国大学基于TMO的模型架构，自主开发了面向多无人机协同控制的实时硬件在环模拟系统（HILS）。该系统是采用快速原型技术，在个人电脑上能运行无人机动力学模型，并在嵌入式控制器上进行飞控算法的运行，最后还开发了地面站与视景显示的模块。

2010年，美国陆军研究所的Richard Garcia也基于商用飞行模拟软件 X-Plane 搭配MATLAB搭建了无人机导航和编队的模块，开发了能实现多无人机的模拟系统。研究人员可以在该系统上进行 4 台无人机的控制与编队算法测试。该系

统通过了专业级的 X-Plane 模拟软件来实现比较贴近真实的无人机飞行模拟效果。

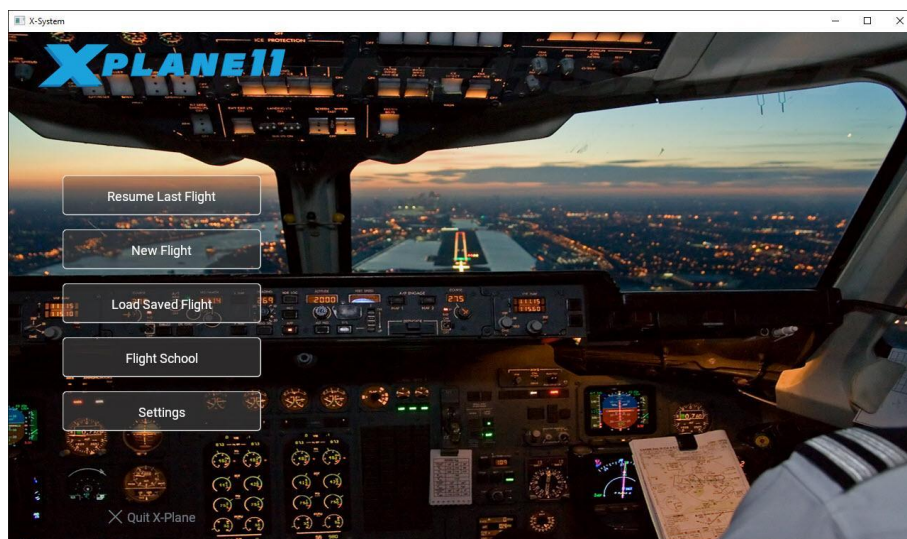


图1-10 X-Plane11用户界面

2017年，微软公司在由Chris Lovett 等开发人员主导并开源了AirSim项目，该项目是基于PX4和Unreal游戏开发引擎提供了一种面向自动驾驶的硬件在环高保真度的模拟系统插件。项目是基于C++语言开发，并实现了实时性达1KHz的四旋翼无人机虚拟模型，且在用户层提供了一套与模拟模型交互的API，以便于进行人工智能和深度学习的相关研究。

中国也在1996年提出的“863 计划”将“分布式虚拟环境”确定为重点研究项目，并以中国北京航空航天大学为主的研究机构，参考了美国的SIMNET施行的DVNET计划，研制了中国第一代分布式战场模拟环境。中国国防科技大学在基于SimulinkRTW技术上搭建的分布式模拟系统方面具有丰富的研究成果。而中国的西北工业大学、南京航空航天大学也基于MATLAB和VxWorks实时操作系统搭建了飞行器快速原型模拟系统。清华大学的学者们则基于多Agent技术开发了一个名为UAVsim的无人机模拟系统，该系统采用了分层的体系结构，并在单系统内将内核、算法和显示进行完全的分离，从而实现多无人机在模拟系统的测试。

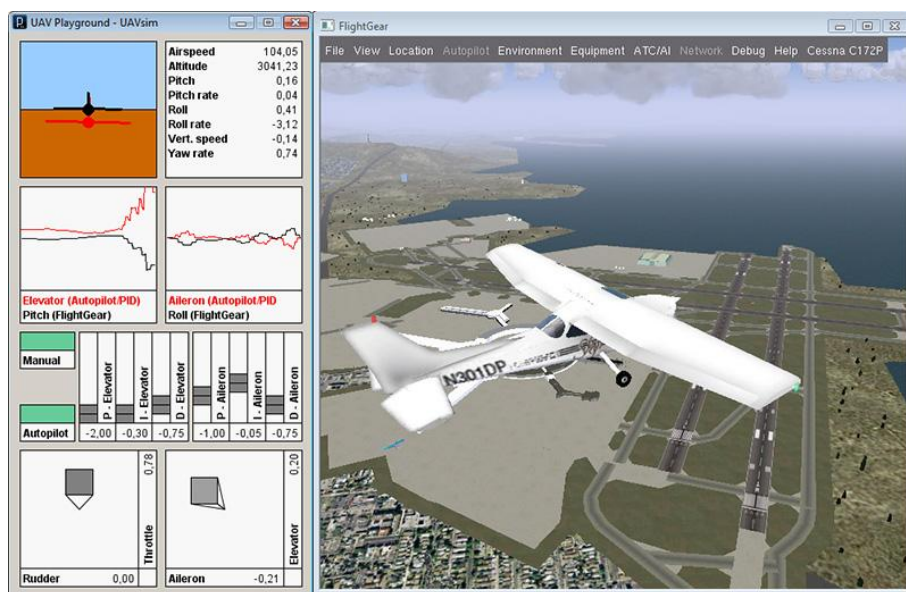


图1-11 UAVsim运行界面

1.3 研究内容以及章节安排

本论文将基于实现四旋翼无人机编队飞行而搭建一个数字孪生系统，并在其基础上构建多个基础模块，最后实现对无人机编队飞行的测试和模拟。在开发过程中，我们将会在 Windows11 系统采用虚幻四引擎并搭配 AirSim 插件作为数字孪生系统的开发框架，同时结合 Ubuntu、Python、硬件遥控器、数据库等系统和软件来实现 API 接口模块、数据存储、远程通信、人机交互等功能。

要构建一个四旋翼无人机编队飞行数字孪生系统，具体内容包括：

- 1) 构建可视化模拟场景一，模拟一个四旋翼无人机编队飞行全局的动态视图场景，该场景视图四周有数据显示区域和操作区域可以显示四旋翼无人机编队飞行全局整体动态演进的相关的状态信息，可进行相应的功能操作。
- 2) 构建可视化模拟场景二，模拟具体某一架正在运行的飞机，该场景视图四周有数据显示区域和操作区域可以显示该架飞机运行时的相关状态信息，可进行相应的功能操作，如果需要可进一步构建更加细化的局部模拟场景。
- 3) 构建统一的数据库，所有数据可统一的访问和存储。
- 4) 构建信息交互的接口模块，可与外部真实系统进行数据的交互。
- 5) 构建对外共享的元数据目录，并基于该目录构建一个信息发布模块，可对外发布开放的共享信息。

最后，根据论文的主要内容研究，章节安排如下所示：

第一章主要针对四旋翼无人机和数字孪生的研究现状做出研究分析和总结，以此确定研究内容和研究意义。

第二章为论述实现四旋翼无人机编队数字孪生系统的总体方案，以分析需求来确定开发环境的选择，概述相关设计的总体架构，介绍的相关开发环境，并概述无人机编队实现算法。

第三章为详细论述四旋翼无人机数字孪生系统开发的设计与实现过程，并且在最后进行测试以验证可行性。

第四章对

第五章对四旋翼无人机编队的数字孪生系统进行总结和展望，并且指出开发过程的困难点和解决方案，分析其中的利弊关系，并在最后分析类似该系统未来的扩展和应用趋势。

第二章 四旋翼无人机编队数字孪生系统的总体方案

2.1 概述

本文将会基于 AirSim 设计系统架构。AirSim 是一个能通过丰富的基于物理的模型来创建真实的动态模型以及渲染场景的开源平台,使这种模拟能够向现实世界转移。这个项目的动机和灵感来自于目前已有的几个模拟器,其中一个关键目标是建立一个社区来推动最先进的技术。

该平台能够高保真地模拟大量物理现象,如重力、磁场、大气条件,并提供试图模拟现实生活的传感器模型。此外,该平台利用图形学的最新进展来构建逼真的环境,这反过来在训练感知模块方面显示出了希望。该框架的主要优势之一是,它提供了一种统一的方式来对各种真实世界的无人机进行闭环控制和规划。目前,该平台包含了飞行器的实现——更具体地说,是四旋翼无人机的实现。

2.2 设计结构

模拟现实世界是一项艰巨的任务。不仅需要对多种现象进行建模,还需要逼真地模拟它们复杂的交互作用。在图 2-1 中展示了不同的组件及其交互。该体系结构的核心构成了模拟环境。在最高级别的模拟器包含无人机模型,环境和物理引擎来计算产生的运动。

此外, AirSim 的框架还支持记录模拟真实世界行为的传感器观察(如传感器漂移、位置误差)。最后,一个大的组件是通过虚幻引擎的真实感渲染,使计算机视觉分析可以转移到现实世界。该框架还支持硬件在环(HIL)模拟,其中像 Pixhawk 这样的飞行控制器可以直接与模拟环境交互。此外,该飞行控制器可以直接与 RC 控制器连接,或与由绘图、定位和计划模块组成的决策引擎通信。在编写 AirSim 当前的实现时,既包含异步命令行接口,也包含 PixHawk 和 DJI A3 等流行的自动驾驶仪以及反应性避障模块的易用 API。最后,通过简单的通信通道切换,实现了与物理无人机的无缝接口。

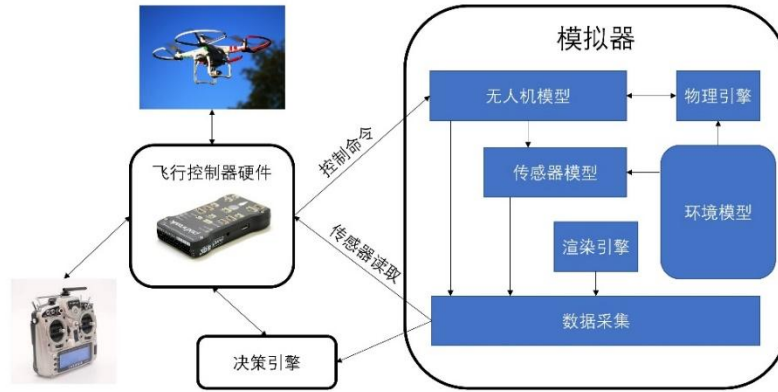


图2-1 系统设计架构

2.3 四旋翼无人机的设计

四旋翼无人机的定义主要包括其总质量、形状和可应用控制输入的一组点的信息。无人机定义还包含每个控制输入如何映射为每个顶点的合力和扭矩。

形式上，质量为 m 的无人机定义为 K 个顶点 $\{x_1, \dots, x_k\}$ ，每一个都有一个控制输入 $\{u_1, \dots, u_2\}$ 。其中 x_i 表示顶点相对于质心的位置， $u_i \in [0, 1]$ 是对应的控制命令。形状信息包含 3×3 转动惯量矩阵 I ，三维向量 A 描述截面面积 $[A_x, A_y, A_z]^T$ 。最后的组成部分是将控制输入 u_i 映射到顶点 i 处的力($F_i = f(u_i)$)和扭矩($\tau_i = g(u_i)$)的方程。

图2-2显示了如何将四旋翼飞行器描述为四个顶点的集合。控制输入 u_i 是一个映射到位于四个顶点的每个螺旋桨的角速度。我们可以用标准方法来计算转动惯量和截面面积。此外，四个顶点上的局部力和力矩可以用以下公式表示：

$$F_i = C_T \sigma \omega_{max}^2 D^4 u_i \quad \text{和} \quad \tau_i = \frac{1}{2\pi} C_{pow} \sigma \omega_{max}^2 D^5 u_i \quad (2-1)$$

C_T 和 C_{pow} 分别是推力和功率系数，是根据螺旋桨的物理特性计算的， σ 是空气密度， D 是螺旋桨的直径， ω_{max} 是每分钟转的最大角速度。还要注意，平台也允许这些顶点的移动。这种移动点的能力使我们能够模拟推力方向的变化，例如垂直起降(VTOL)飞行器和最近的四旋翼飞行器在飞行中改变其配置。

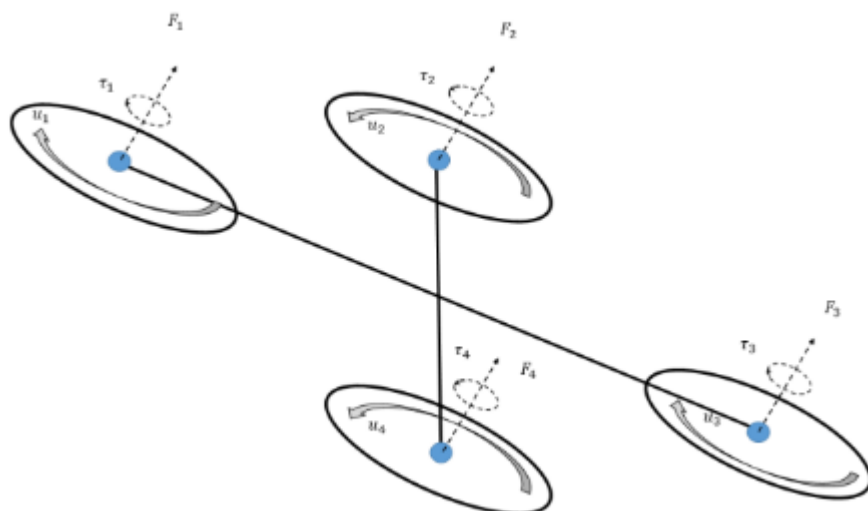


图2-2 四旋翼无人机的四个顶点

2.4 虚拟环境

四旋翼无人机会与周围环境相互作用。该系统将模拟各种物理现象，如重力、空气密度、地球磁场以及气压的影响。这里的目标是在不使用大量计算资源的情况下尽可能准确地提供这些自然现象的模拟。本文将在下面描述这些组件。

2.4.1 重力

为了精确模拟重力的影响，系统将模拟重力加速度 g 随高度的变化。虽然与地球半径相比，高度引起的变化可以忽略不计，但当地面平原本身处于海拔高度或在高空飞行时，考虑它的影响是很重要的。

在形式上，我们可以很容易地看出在高度 h 处得到的重力加速度 g 与地球表面恒定的重力加速度 g_{const} 的关系如下：

$$g = g_{const} \cdot \frac{R_e^2}{(R_e + h)^2} \approx g_{const} \cdot \left(1 - 2 \frac{h}{R_e}\right) \quad (2-2)$$

这里， R_e 是地球的半径，第一个等式是由牛顿万有引力定律的简单代数运算产生的。这个近似使我们避开代价高昂的平方运算，它是应用二项式定理和忽略高次幂的结果。

2.4.2 地球磁场

模拟地球磁场是一项非常重要而又复杂的任务。磁力计是大多数自动驾驶汽车的关键传感器之一，但考虑到其中的复杂性，主流模拟器没有一个试图准确模拟这种效应也就不足为奇了。美国国家海洋和大气管理局(NOAA)制作的世界磁场模型(WMM)模型是目前最好的公开磁场计算代码之一。但是，这个模型需要

昂贵的计算资源和大内存来加载大的查找表。因此，在高频仿真中使用WMM模型是非常重要的。该模拟器的核心贡献之一是一个有效的组件，可以准确地模拟地球磁场。

我们从地球的倾斜偶极模型开始，我们假设地球是完美的偶极球，忽略除了磁场的一阶项以外的所有项。我们可以用这样的近似，因为我们只对磁场在空间中移动时的变化感兴趣，而不是绝对值。给定地理纬度 θ ，经度 φ 和高度 h (从地球表面)，我们首先用以下方法计算磁共纬 θ_m ：

$$\cos \theta_m = \cos \theta \cos \theta^0 + \sin \theta \sin \theta^0 \cos (\varphi - \varphi^0) \quad (2-3)$$

当 θ^0 和 φ^0 分别表示真磁北极的经纬度。则总磁场强度 $|B|$ 计算为：

$$|B| = B_0 \left(\frac{R_e}{R_e + h} \right)^3 \sqrt{1 + 3 \cos^2 \theta_m} \quad (2-4)$$

这里 B_0 是地球表面磁赤道处磁场的平均值， θ_m 是磁共纬， R_e 是地球的平均半径。接下来，我们通过以下关系确定倾角 α 和赤纬 β 角：

$$\begin{aligned} \tan \alpha &= 2 \cot \theta_m, \\ \sin \beta &= \begin{cases} (\sin(\varphi - \varphi^0) \frac{\cos \theta^0}{\cos \theta_m}, & \cos \theta_m > \sin \theta^0 \sin \theta \\ \cos(\varphi - \varphi^0) \frac{\cos \theta^0}{\cos \theta_m}, & \text{其他} \end{cases} \end{aligned}$$

最后，我们可以计算磁场矢量的水平分量(H)、垂直分量(Z)、纬度分量(X)和纵向分量(Y): $H = |B| \cos \alpha$, $Z = |B| \sin \alpha$, $X = H \cos \beta$, $Y = H \sin \beta$ 。

2.4.3 空气密度和气压

空气密度和气压会对飞行器的推力，升力和阻力造成影响。尤其是要模拟气压和密度随高度变化的变化。在系统模拟实现中，我们可以使用标准大气模型来表示51公里以下的高度，然后计算压力 P 和温度 T 下的空气密度 $\sigma = \frac{P}{R \cdot T}$ ，其中 R 是特定气体常数。

2.5 物理引擎和传感器

物理引擎的目标是考虑四旋翼无人机及其运行环境，并确定在施加控制时，由于力和扭矩而产生的运动。虽然有许多现成的物理引擎可以使用，但平台实现了自己的模块来模拟物理。这样做主要是为了提高计算效率。

对于实时无人机应用来说，物理引擎在非常高的频率下运行是非常重要的。根据Airsim测试的经验评估显示，我们需要在大约1000赫兹的频率下运行这样的引擎。现有的大多数发动机都不能在如此高的频率下工作。其中一个主要原因是在这些发动机碰撞检查过程最终需要大量的计算。我们可以通过将碰撞检查任务

交给视觉渲染引擎来缓解这个问题，从而只关注物理计算的核心方面。

传感器的主要目标之一则是通过收集训练数据，使现实系统的机器学习成为可能。因此，也必须对收集数据的传感器进行尽可能逼真的建模。因为模拟器根据定义知道地面的真实情况，传感器可以模拟无人机如何感知环境。

2.7 视觉呈现

由于真实感渲染是AirSim的关键需求，我们可以选择虚幻引擎4 (UE4)作为渲染管道。UE4带来了几个关键特性，使它成为一个有吸引力的选择：

1. 真实感渲染

UE4带来了一些最尖端的图形进步，包括实时全球照明。下图显示了来自AirSim系统的几个屏幕截图，突出显示了接近真实世界的渲染质量。



图2-3 Airsim屏幕截图

2. 开源、跨平台

UE4完全开源，免费使用模拟，并无缝地在Linux， Windows、iOS、Android和OS x也非常可靠和附带了一组丰富的开发人员工具从而使空中信息学的发展在开发平台上更加容易。

3. 素材商店

在素材商店中，人们可以购买预先制作的精致环境，使从各种环境中收集数据变得容易。

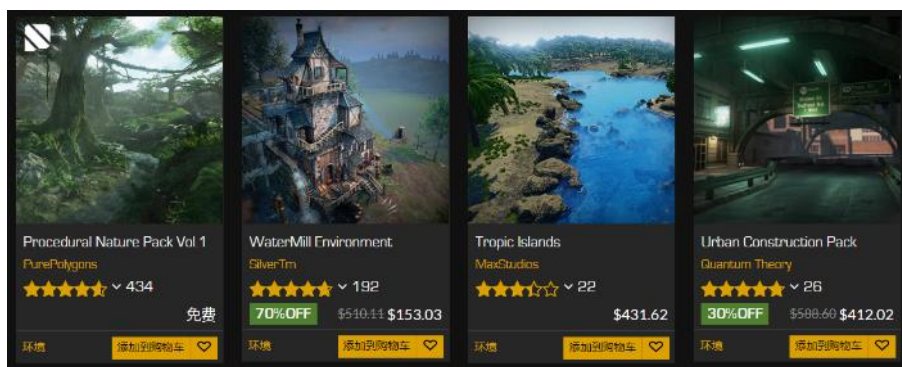


图2-4 虚幻四引擎素材商店

在AirSim中，我们可以增加了在任何无人机上添加虚拟单眼和立体摄像机的功能，并在选择的框架(例如ROS)中接收RGB和深度图像流。如图2-5所示为立体图像对示例和相应的深度图像。这一功能对于在现实环境中收集大量感知数据至关重要。



图2-5 Airsim中的深度摄像头

2.8 无人机编队实现

本节在开始前的要讨论在于解决了分布式编队控制文献中经常没有考虑到的两个具有挑战性的主要问题：

- 1) 无人机如何在不依赖集中运动规划方案避免碰撞的情况下实现理想的编队飞行？
- 2) 无人机如何在不使用全局位置测量的情况下定位它们的邻接飞行器？

为了解决第一个问题，我们可以运用一种控制设计，它具有几个可用于避免碰撞的属性。针对第二个问题，我们可以运用一种新的相机姿态估计方法，利用机载无人机相机捕获的图像来估计编队控制所需的相对位置和方向信息。

在接下来的内容中，我们将介绍拟议管道的编队控制和摄像机姿态估计组件，并展示如何将它们合并为一个完整的系统。我们假设无人机希望在一个恒定的高度上操作，所有希望的编队都是平面的，无人机配备了低空飞行控制器，可以将其速度调节到所需的值。

1. 分布式编队控制

为使一组 n 架无人机达到所需的平面编队，为每架无人机定义线速度控制向量 $u_i \in R^2$ 为：

$$u_i := \sum_{j \in N_i} A_{ij} q_j^i \quad (2-2)$$

其中， $i \in \{1, 2, \dots, n\}$ 表示每个无人机的指数， $N_i \subset N$ 是该指数的相邻无人机的第 i 个无人机， $q_j^i \in R^2$ 表示无人机的位置在 i 的本地坐标系无人机， $A_{ij} \in R^{2 \times 2}$ 是常数控制增益矩阵。在任意时刻，每架无人机的低空飞行控制器将无人机的线速度调节到公式(2-2)中给定的期望控制矢量 u_i 。

如果选择合适的 A_{ij} 矩阵，则所有无人机的都会产生合适的相互作用。为了设计这些矩阵，搜索了一种基于半定问题(SDP)公式的方法，该方法不仅确保达到预期的生成，而且进一步具有以下性质，这些性质在实际工作中得到了证明：

- a) 该控制对输入中的扰动、噪声和扰动具有鲁棒性。
- b) 该控制对未建模动态和输入饱和具有鲁棒性。
- c) 控制向量的任何正缩放都不会影响无人机收敛到所需编队。
- d) 如果无人机在理想的方向移动由 $\pm 90^\circ$ 旋转扰动，收敛到理想的编队保持。

我们可以利用 d) 来设计一个完全分布式的避碰策略。在该策略中，当飞行器沿控制矢量方向运动导致碰撞时，通过旋转控制矢量来避免碰撞。如果所需的旋转超过 90° ，无人机将会停止飞行，直到找到一个可行的方向。尽管由于这种策略的分布式的特性，还是有可能发生不可避免的碰撞，但在广泛的模拟中可以观察到，如果无人机一开始相距很远，它们通常也可以克服这些情况，并收敛到所需要的编队。

如果无人机之间的传感和通信图是时变的，通过向SDP问题添加额外的约束，可以确保无人机实现所需的编队，无论传感拓扑中是否开关。

2. 摄像机相对姿态估计

图像特征点是图像中的边缘、角等，可以在从场景的两个不同位置拍摄的图像中识别和匹配。如图2-6所示，从两架无人机的机载图像中提取带有 + 符号的SURF特征点，并与虚线进行匹配。匹配的特征点对 $m, n \in R^3$ 的齐次坐标由图像可知，必须满足刚体运动约束：

$$uRm + t = vn \quad (2-3)$$

其中 $R \in SO(3)$ ， $t \in R^3$ 分别为每个视图下摄像机坐标帧之间未知的相对旋转矩阵和平移向量(即位姿)， $u, v \in R$ 为对应坐标帧内特征点的未知深度。通过

使用旋转矩阵 R 的四元数表示，新姿态估计算法QuEst可以从低至5个通用特征点对恢复相对姿态。平均而言，与最先进的算法相比，QuEst的估计精度提高了50%。此外，还可以加入RANSAC (random sample consensus)来消除任何潜在的不匹配特征点

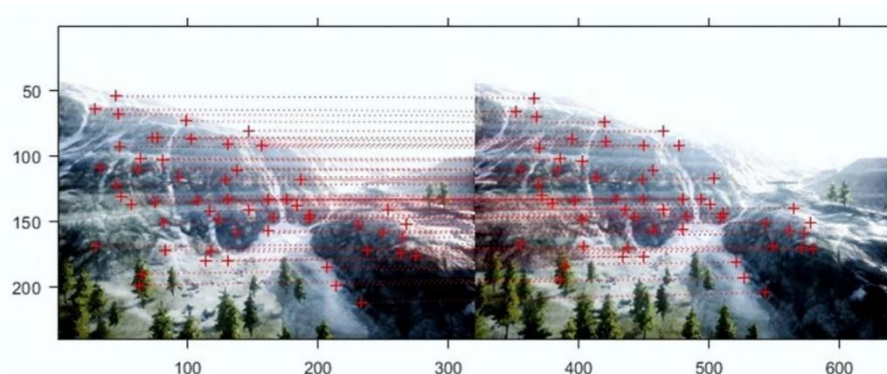


图2-6 两幅由无人机前置摄像头拍摄的 320×240 的图像

3. 基于视觉的分布式编队控制

在无人机飞行编队控制管道中，计算控制矢量所需的相对位置测量是根据机载无人机相机捕获的图像估计的。无人机从机载图像中提取特征点像素坐标及其描述符，并将其传递给相邻图像。如果摄像机具有重叠的视场，则可以根据其描述符匹配对应的特征点，并在QuEst中用于估计相对位姿。一般而言，仅使用图像估计的相对位置只能恢复到一个共同的正未知比例因子。然而，由于这在之前提出的编队控制策略中，因此并不影响无人机收敛到所需编队。

为了研究管道所需的通信带宽，我们可以考虑SURF图像特征，并假设在50 ms内从 320×240 像素图像中提取500个特征点。每个特征点由2个单精度 $x - y$ 坐标和一个64位元素的单精度描述子向量组成，其中每个单精度数用32位表示。因此，每架相邻无人机所需的通信带宽约为5.2 KB/s，完全能在当前无线通信技术的实现范围内。

2.9 小结

Airsim是能够实现数据驱动无人机系统的快速培训和开发的平台。值得一提的是，该平台能够实现高保真模拟，从而可以用于收集训练数据，以构建机器学习模型。核心组件包括一个具有物理现象详细模型的快速物理引擎，以及一个能够训练和测试计算机视觉模块的真实感知引擎。通过利用这样的模拟器，我们将有效地利用强化学习和模仿学习等方法，使模拟器能够向现实世界转移并学习技术。

第三章 数字孪生系统的设计与实现

3.1 概述

本文的四旋翼无人机编队的数字孪生系统将会通过 AirSim 和虚幻 4 引擎来进行开发。本章节将叙述系统的各种的功能实现和开发过程，并达到在不同场景资源下模拟出不同的场景逻辑对四旋翼无人机编队的影响。

3.2 计算机软硬件配置

Airsim 是基于虚幻四引擎的插件，因此要流畅运行虚幻四引擎对计算机系统和硬件有一定性能需求，同时因为在开发过程中需要进行大量的 3D 渲染，因此也需要一个性能达标的显卡，下表为官方推荐最低配置和本文在开发过程中所使用的计算机配置。

表3-1 环境搭建所需硬件配置

	最低要求	本文所使用的计算机
系统	Windows 7	Windows 11 64-bit
处理器	Intel 或 AMD 四核处理器， 2.5GHz 或更快	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
内存	8GB	16GB
显卡	DirectX11 或 12 兼容显卡	NVIDIA GeForce GTX 1650
硬盘		500GB SSD

本文将会使用的软件版本配置如下：

- AirSim 版本：v1.5.0
- 虚幻引擎版本：4.27
- Visual Studio 版本：Visual Studio Community 2019
- Python 版本：Python 3.10
- Ubuntu 版本：20.04

3.3 Airsim 安装与环境配置

3.3.1 虚幻四引擎的安装

在安装 AirSim 之前,我们需要先安装虚幻四引擎来运行 AirSim 插件,首先,我们需要到 EpicGames 官网安装其客户端。

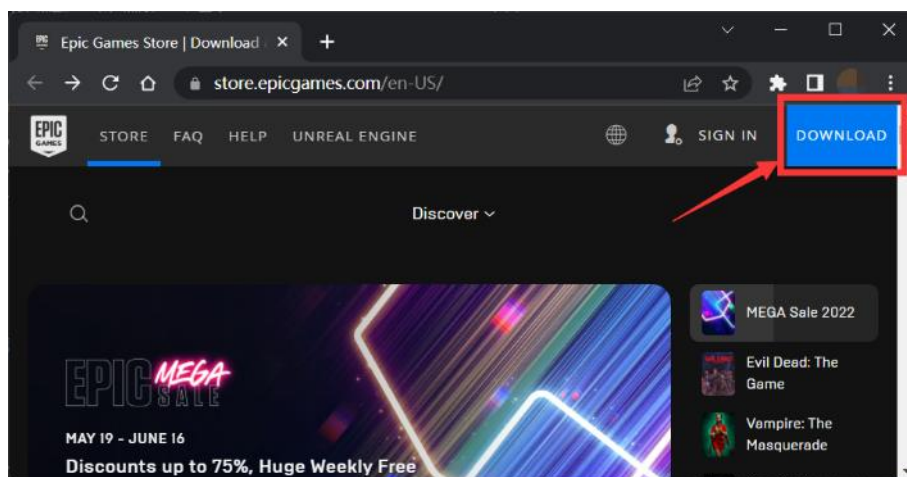


图 3-1 EpicGames 客户端下载

安装完成后打开客户端界面的左上角菜单栏就有虚幻引擎页面的选择,点开虚幻引擎页面后选择“库”就能选择相应版本来安装虚幻引擎。

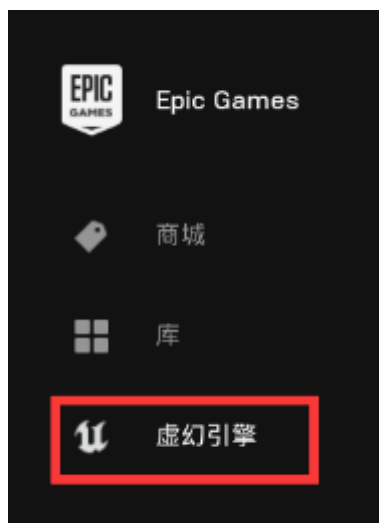


图 3-2 虚幻引擎的下载

目前虚幻引擎系列其实已经出到第五代了,第五代虚幻引擎具有更强大的保真度和灵活性,但是本文基于开发稳定的考虑,我们这里选择依然是 4.27.2 的驱

动版本。安装虚幻引擎需要至少 25GB 的硬盘空间，加上后续开发系统和各种软件的安装，推荐至少预留 100GB 左右的硬盘空间。



图 3-3 虚幻引擎的版本

3.3.2 AirSim 的安装

虚幻引擎安装完成后，我们可以开始 Airsim 的安装，首先我们需要安装 Visual Studio 2019 并搭配“使用 C++ 的桌面开发”的组件，安装完成后通过 Windows 的开始菜单搜索“Developer Command Prompt for VS 2019”，这里建议以管理员身份运行，因为本文设计时只有 C 盘，因此安装 Airsim 到 C 盘需要管理员权限，但是如果打算安装在 D 盘的话可以直接打开。

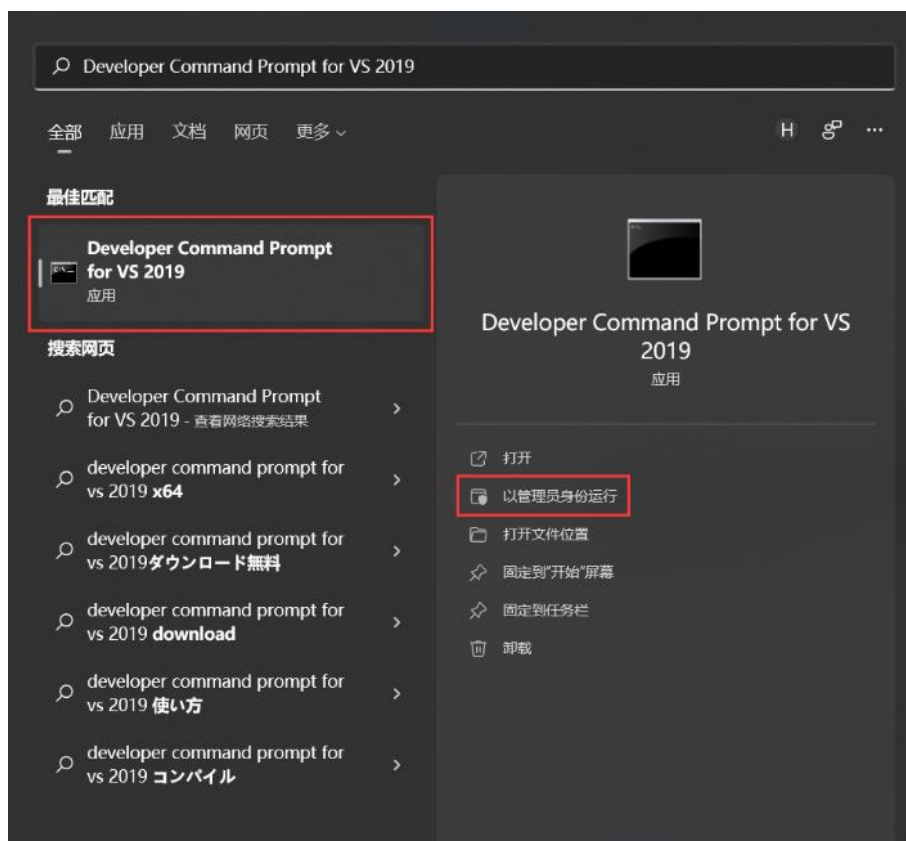


图 3-4 Developer Command Prompt for VS 2019

打开后输入 `git clone https://github.com/Microsoft/AirSim.git` 进行安装，安装好后 `cd AirSim` 进入文件输入 `build.cmd` 开始编译，编译好会在 AirSim 的目录生成一个 Plugins 的文件，把这个文件复制到虚幻引擎的工程文件就能在虚幻引擎工程里面使用 Airsim 插件。

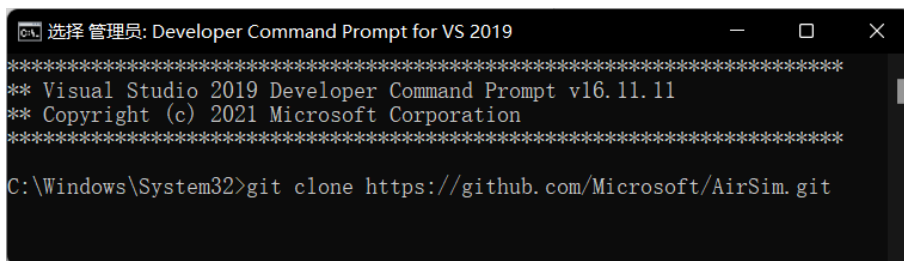


图 3-5 安装 AirSim

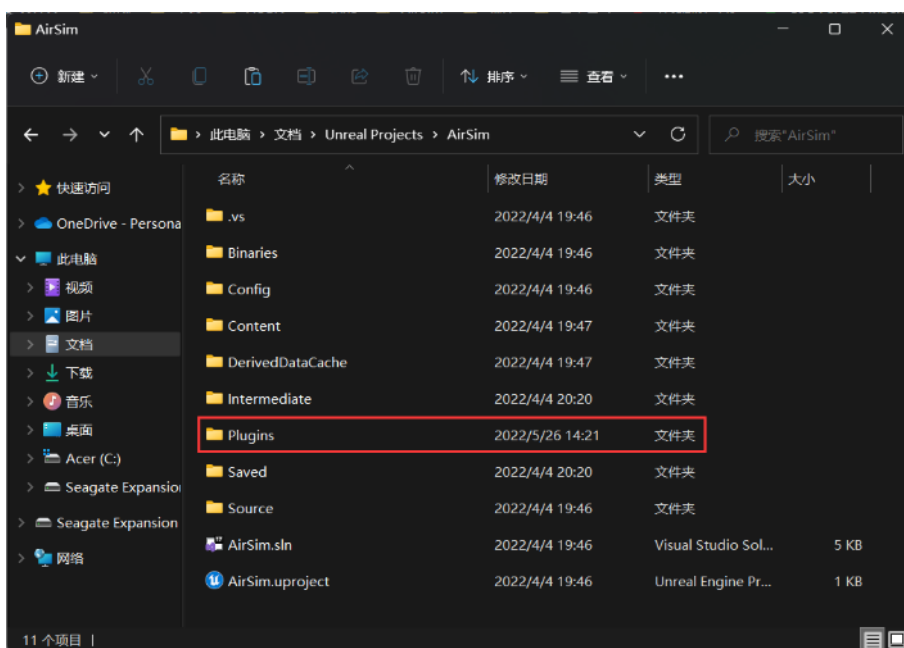


图 3-6 Plugins 文件生成后挪动至所需工程文件

3.4 Airsim 环境测试

安装好 Airsim 后，在目录文件中找到 AirSim>Unreal>Environment，会出现一个名叫 Block 的工程文件，我们可以直接基于这个工程文件进行开发，打开 Blocks.uproject 后如图 3-7 所示就代表 AirSim 成功安装并完美运行。

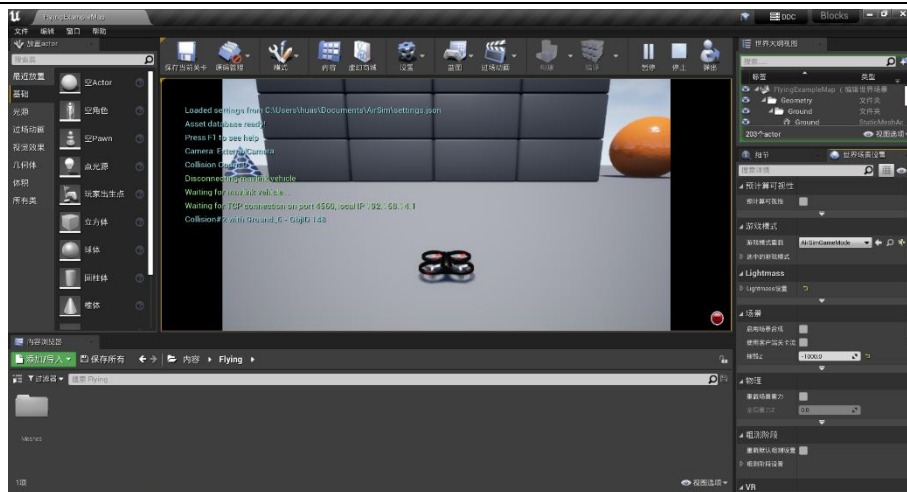


图 3-7 Blocks 工程文件成功运行的画面

3.5 模拟环境的搭建

只基于 Block 工程文件是很难测试无人机在各种环境的实际情况的，毕竟 Block 文件是官方基于虚幻四本就有的环境模板上的一个基础测试，因此我们需要进一步对模拟世界进行搭建，在这里^[7]我们能看到关于 Landscape Mountains 环境搭配 AirSim 的安装教程，Landscape Mountains 环境是一个由 Epic 官方制作的免费且优良的官方教程环境，但在这里我们不多介绍如何使用这个环境进行搭建，我们可以直接运用 AirSim 官方提供的多个预先编译好的环境，如图 3-9 所示，而本文设计开发基本基于 MSBuild2018 环境，因为这个环境资源运用的相对较小，在进行一些简单的模拟测试时能节省计算机资源。

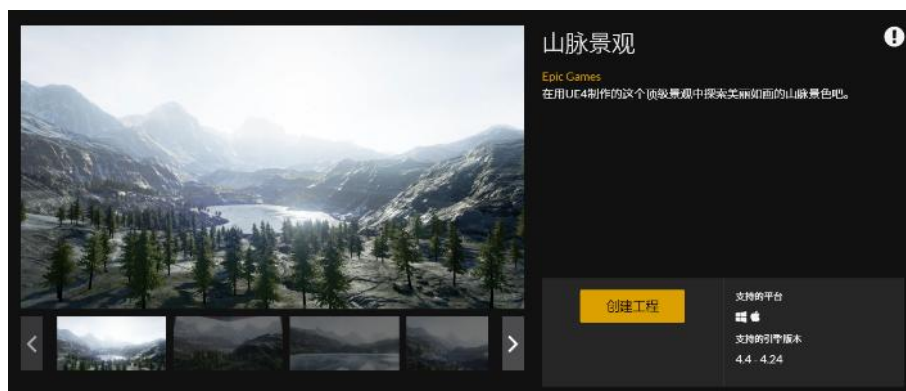


图 3-8 Landscape Mountains 环境

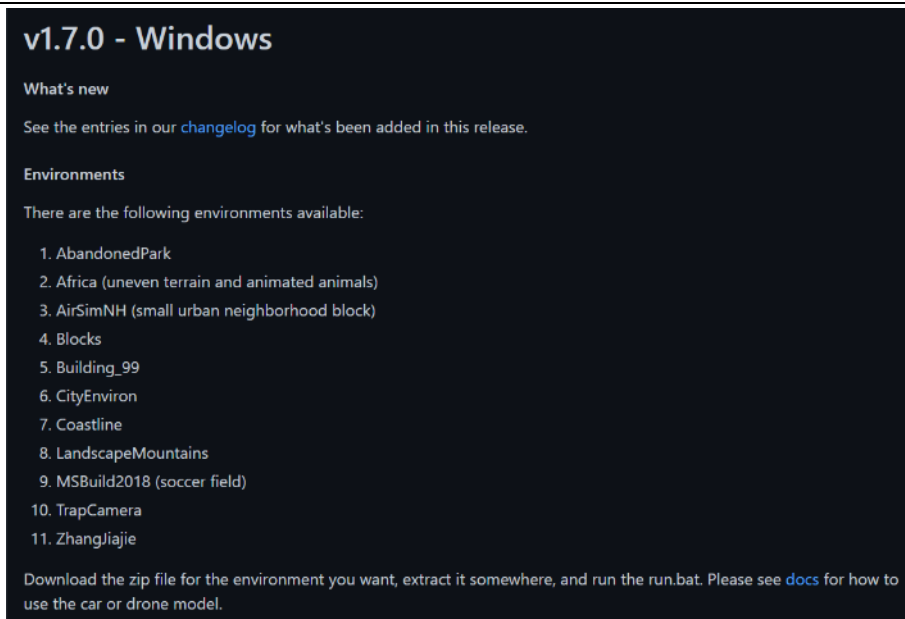


图 3-9 官方提供的环境

下载完成后解压，直接打开 MSBuild2018.exe 会出现如图 3-10 所示，选择“否”就能在环境中直接生成一个无人机，成功运行的软件如图 3-11 所示。但是目前我们还无法操控无人机飞行，操控无人机需要接入遥控器硬件，之后本文还会介绍如何通过更改 setting.json 文件来实现只默认生成无人机，并实现多无人机生成和各种功能搭配。



图 3-10 打开模拟器时的交通工具选择询问



图 3-11 进入模拟器的画面

3.6 操控无人机飞行

一般要接入飞控系统，我们需要将 PX4 加入模拟器之中，但是因为 AirSim 自带了一个叫 `simple_flight` 的四旋翼无人机飞控系统，因此我们可以直接接上遥控器硬件来直接操控 AirSim 环境里的无人机。官方文件推荐了一款 FrSky Taranis X9D Plus 的遥控器型号，这款遥控器大约要价 220 美元，最后基于经济条件，本文选择了现有的 Xbox 游戏手柄作为替代遥控器。



图 3-12 FrSky Taranis X9D Plus



图 3-13 Xbox 游戏手柄

3.7 API 接口程序与 python 的环境搭建

AirSim 的 API 采用了一个叫 msgpack-rpc 的网络通信协议，在 AirSim 模拟器运行期间会打开一个 41451 的端口，因此我们可以使用 Python 基于这个端口发送 RPC 包和 AirSim 进行通信。

3.7.1 Anaconda 环境搭建

在搭配安装 Python 时，我们需要安装 Anaconda 来对 Python 进行环境管理，而安装好 Anaconda 后，点击 Environments 并点击 Create 来搭建一个 airsim 环境，如图 3-14 所示：

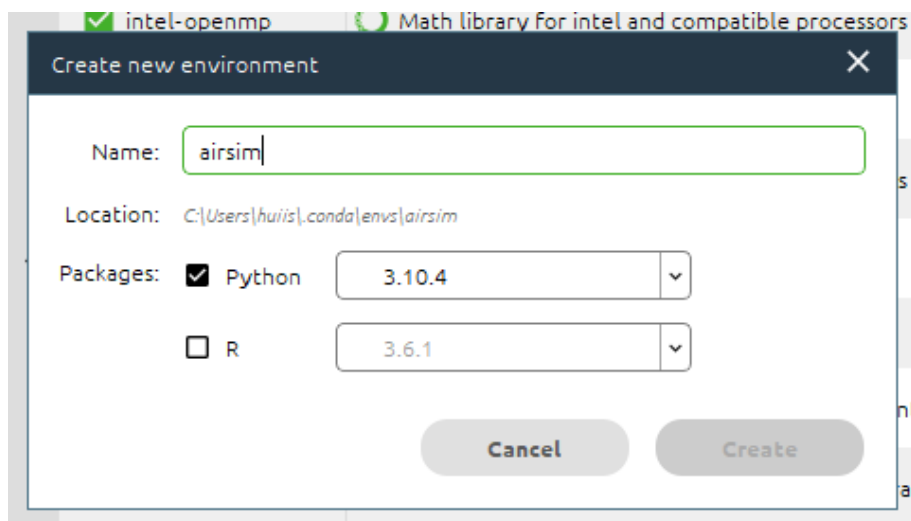


图 3-14 在 Anaconda 创建一个名叫 airsim 的环境

搭建好后在右边搜索 numpy 的库并进行安装，numpy 是一个矩阵运算库，在做无人机编队算法的时候需要用到 numpy。

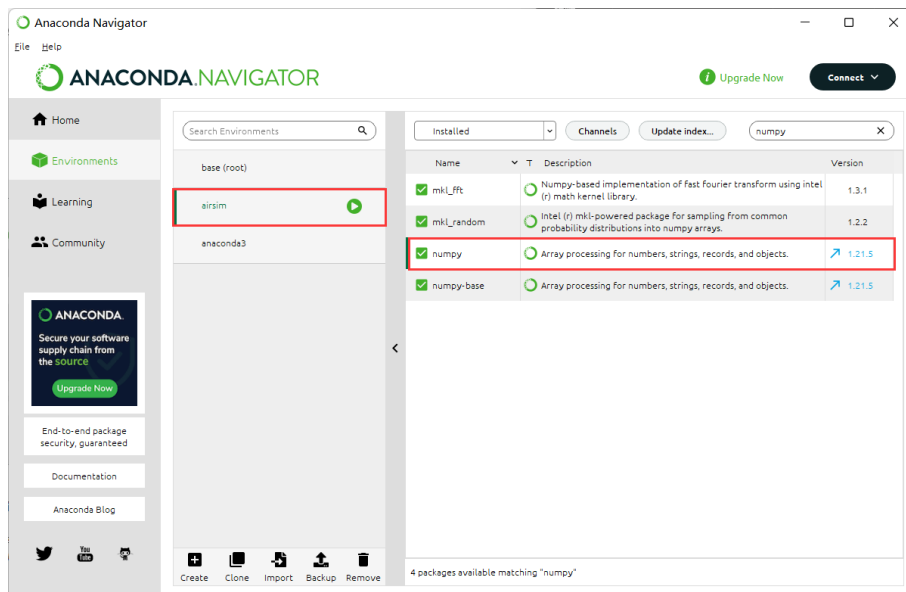


图 3-15 安装 numpy

之后打开 terminal 并在命令行输入 `pip install msgpack-rpc-python` 和 `pip install aircsim`，至此，AirSim 仿真测试所需要的包都安装好了。

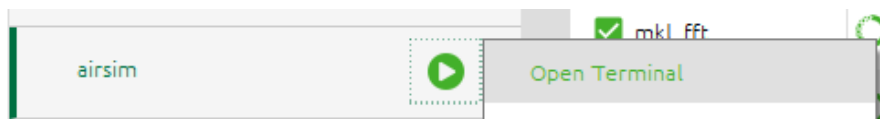


图 3-16 在 Anaconda 打开 Terminal

3.7.2 PyCharm 环境搭建

PyCharm 是一个主要用于 Python 的集成开发环境，我们只需要下载社区版就能免费使用许多功能，本文用的是 PyCharm Community 2021.3.3 的版本，安装好 PyCharm 后，打开 New Project 并跟着图 3-17 设置就能开始运用 Python 的 API 进行 AirSim 模拟测试了。

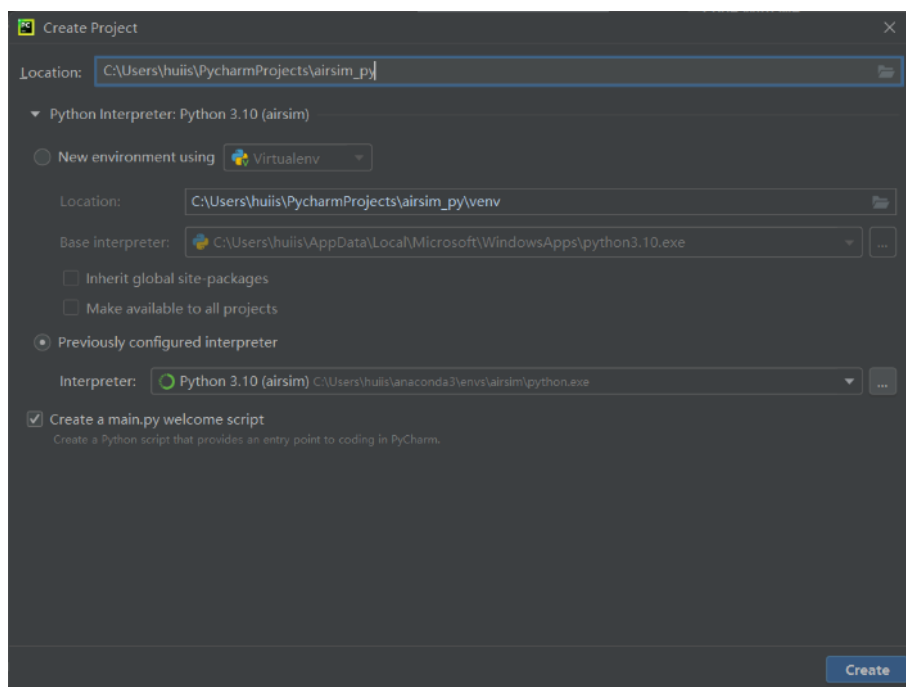


图 3-17 PyCharm 环境搭建

3.7.3 代码运行测试

打开 MSBuild2018.exe 进入模拟器环境，这时候打开上节在 PyCharm 搭建好的工程文件，输入一些相关代码就能对无人机进行各种操作，图 3-18 代码为简单让无人机自动起飞和降落。

```
1  """
2  test python environment
3  """
4  import airsims
5
6  # connect to the AirSim simulator
7  client = airsims.MultirotorClient()
8  client.confirmConnection()
9
10 # get control
11 client.enableApiControl(True)
12
13 # unlock
14 client.armDisarm(True)
15
16 # Async methods returns Future. Call join() to wait for task to complete.
17 client.takeoffAsync().join()
18 client.landAsync().join()
19
20 # lock
21 client.armDisarm(False)
22
23 # release control
24 client.enableApiControl(False)
```

图 3-18 无人机起飞和降落代码

3.8 setting.json 文件

setting.json 是运行 Airsim 模拟器的一个核心文件，在第一次打开 MSBuild2018.exe 的，AirSim 就会自动创建该文件，setting.json 文件一般存储在 C:\Users\用户名\Documents\AirSim 的路径之下，通过修改 setting.json 的代码我们能够在模拟器中实现许多功能，而关于 setting.json 的格式官方文件^[8]有说明，因此不详细叙述。这里本文只说明一些相关代码实现。

3.8.1 多无人机生成代码实现

在 setting.json 中，我们只需要通过简单修改几段代码就能实现多个无人机生成，生成的无人机能够实现跟随主飞机飞行，也能实现简单的编队飞行，下图为创建三个无人机的简单代码示范：

```
{
  "ClockSpeed": 1,
  "SeeDocsAt": "https://github.com/Microsoft/AirSim/blob/master/docs/settings.md",
  "SettingsVersion": 1.2,
  "SimMode": "Multirotor",
  "Vehicles": {
    "Drone0": {
      "Pitch": 0,
      "Roll": 0,
      "VehicleType": "SimpleFlight",
      "X": 0.0,
      "Y": 0.0,
      "Yaw": 0,
      "Z": 0
    },
    "Drone1": {
      "Pitch": 0,
      "Roll": 0,
      "VehicleType": "SimpleFlight",
      "X": 0.0,
      "Y": 2.0,
      "Yaw": 0,
      "Z": 0
    },
    "Drone2": {
      "Pitch": 0,
      "Roll": 0,
      "VehicleType": "SimpleFlight",
      "X": 0.0,
      "Y": 4.0,
      "Yaw": 0,
      "Z": 0
    }
  }
}
```

图 3-19 生成三个无人机的代码实现

之后，我们甚至能给每个无人机搭配各种功能，比如传感器，下图为无人机搭配传感器和摄像头的代码实现范例。

```

    "Dronel": {
      "VehicleType": "SimpleFlight",
      "Sensors": {
        "imu_2": {
          "SensorType": 2,
          "Enabled": true
        },
        "lidar1": {
          "SensorType": 6,
          "Range": 100,
          "Enabled": true,
          "NumberOfChannels": 64,
          "PointsPerSecond": 100000,
          "VerticalFOVUpper": 90,
          "VerticalFOVLower": 0,
          "HorizontalFOVStart": -90,
          "HorizontalFOVEnd": 90,
          "X": 0, "Y": 0, "Z": -0.1,
          "DrawDebugPoints": true
        }
      },
      "Cameras": {
        "camera1": {
          "CaptureSettings": [
            {
              "ImageType": 0,
              "Width": 672,
              "Height": 376,
              "FOV_Degrees": 90,
              "TargetGamma": 1.5
            },
            {
              "ImageType": 1,
              "Width": 672,
              "Height": 376,
              "FOV_Degrees": 90,
              "TargetGamma": 1.5
            }
          ],
          "X": 0.1, "Y": -0.06, "Z": 0.10,
          "Pitch": 0.0, "Roll": 0.0, "Yaw": 0.0
        },
        "X": 4.0, "Y": 0.0, "Z": 0,
        "Pitch": 0, "Roll": 0, "Yaw": 0
      }
    }
  }
}
```

图 3-20 给每个无人机搭配摄像头和传感器代码实现

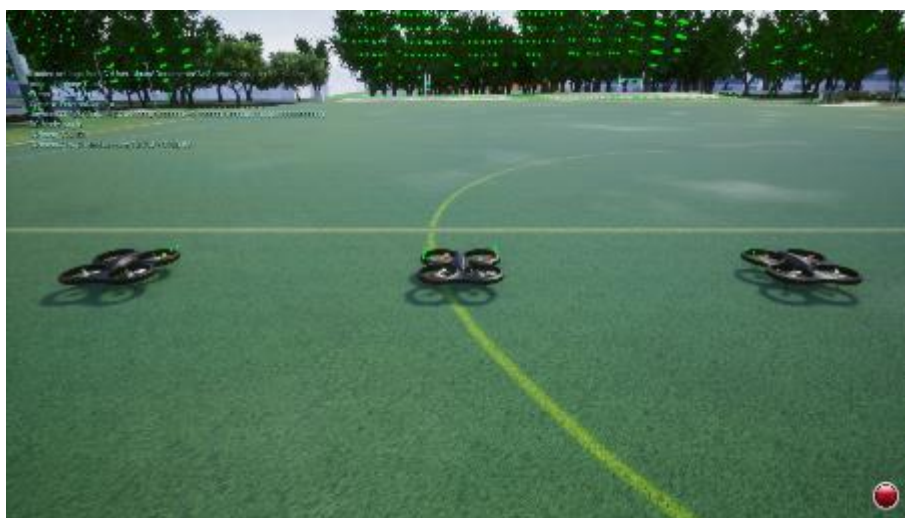


图 3-21 多无人机代码生成结果

3.8.2 远程通信代码的实现

以下代码是通过 PX4 实现客户端远程通信和操控，关于 PX4 的安装则会在下一节介绍。

```
{
  "SettingsVersion": 1.2,
  "SimMode": "Multirotor",
  "ClockType": "SteppableClock",
  "Vehicles": {
    "PX4": {
      "VehicleType": "PX4Multirotor",
      "UseSerial": false,
      "LockStep": true,
      "UseTcp": true,
      "TcpPort": 4560,
      "ControlIp": "remote",
      "ControlPortLocal": 14540,
      "ControlPortRemote": 14580,
      "LocalHostIp": "192.168.14.1",
      "Sensors": {
        "Barometer": {
          "SensorType": 1,
          "Enabled": true,
          "PressureFactorSigma": 0.0001825
        }
      }
    },
    "Parameters": {
      "NAV_RCL_ACT": 0,
      "NAV_DLL_ACT": 0,
      "COM_OBL_ACT": 1,
      "LPE_LAT": 47.641468,
      "LPE_LON": -122.140165
    }
  }
}
```

图 3-22 通过 PX4 进行客户端连接代码实现

3.9 远程通信接口模块

为了达到远程通信的测试来实现如远程操控模拟器甚至连接物理世界的功能，本文会描述在虚拟机中基于 Ubuntu20.04 操作系统的版本实现这个功能。

在 Ubuntu 操作系统中我们需要通过 terminal 并跟着 PX4 官网的教程^[9]对 PX4 进行安装，安装好之后，我们还需要参考这里的网站^[10]对 QrGround 软件进行安装，QrGround 是一个基于 PX4 的飞行配置软件，如果不安装 QrGround 的话，会在调用 PX4 命令时产生问题。

在 Ubuntu 把所有软件都安装好之后，在 terminal 依次输入：

```
cd PX4
```

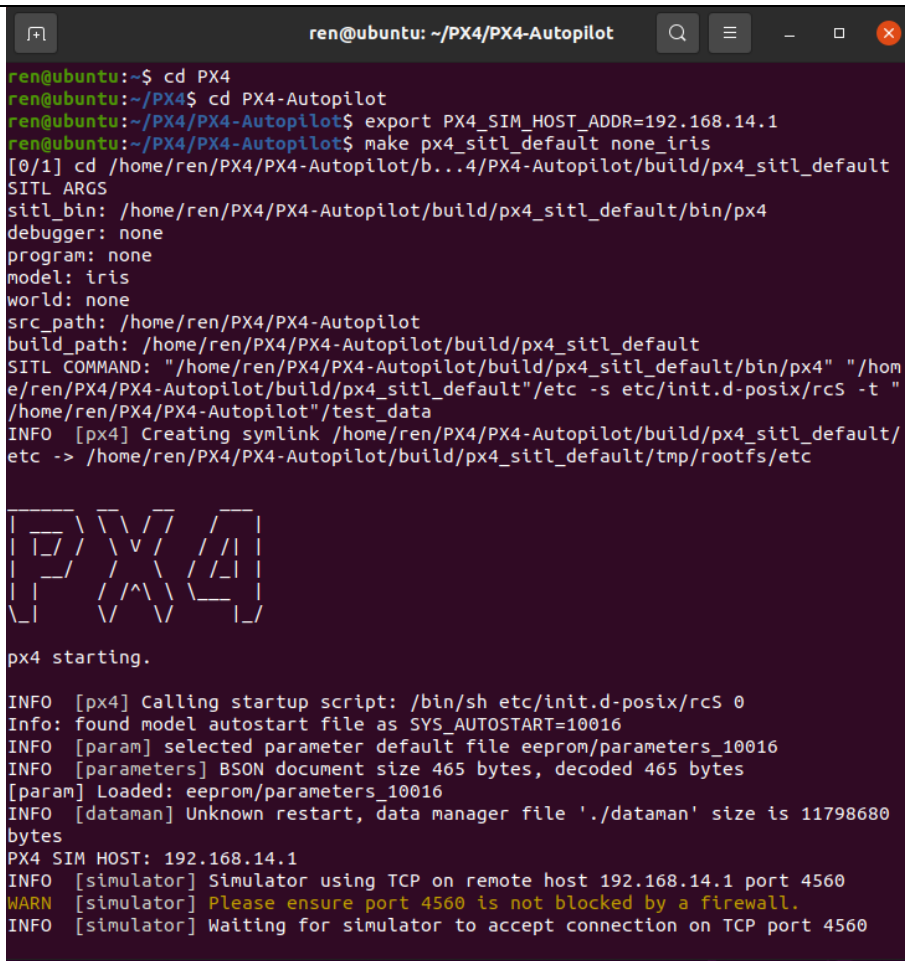
```
cd PX4-Autopilot
```

```
export PX4_SIM_HOST_ADDR=192.168.14.1
```

（这里需要参考主机 IP 地址，本文在进行测试时主机 IP 是 192.168.14.1）

```
make px4_sitl_default none_iris
```

之后如果出现以下画面就代表可以正常通信。



```

ren@ubuntu: ~/PX4/PX4-Autopilot
ren@ubuntu:~/PX4$ cd PX4-Autopilot
ren@ubuntu:~/PX4/PX4-Autopilot$ export PX4_SIM_HOST_ADDR=192.168.14.1
ren@ubuntu:~/PX4/PX4-Autopilot$ make px4_sitl_default none_iris
[0/1] cd /home/ren/PX4/PX4-Autopilot/b...4/PX4-Autopilot/build/px4_sitl_default
SITL ARGS
sitl_bin: /home/ren/PX4/PX4-Autopilot/build/px4_sitl_default/bin/px4
debugger: none
program: none
model: iris
world: none
src_path: /home/ren/PX4/PX4-Autopilot
build_path: /home/ren/PX4/PX4-Autopilot/build/px4_sitl_default
SITL COMMAND: "/home/ren/PX4/PX4-Autopilot/build/px4_sitl_default/bin/px4" "/home/ren/PX4/PX4-Autopilot/build/px4_sitl_default/etc -s etc/init.d-posix/rcS -t "/home/ren/PX4/PX4-Autopilot"/test_data
INFO [px4] Creating symlink /home/ren/PX4/PX4-Autopilot/build/px4_sitl_default/etc -> /home/ren/PX4/PX4-Autopilot/build/px4_sitl_default/tmp/rootfs/etc

PX4

px4 starting.

INFO [px4] Calling startup script: /bin/sh etc/init.d-posix/rcS 0
Info: found model autostart file as SYS_AUTOSTART=10016
INFO [param] selected parameter default file eeprom/parameters_10016
INFO [parameters] BSON document size 465 bytes, decoded 465 bytes
[param] Loaded: eeprom/parameters_10016
INFO [dataman] Unknown restart, data manager file './dataman' size is 11798680 bytes
PX4 SIM HOST: 192.168.14.1
INFO [simulator] Simulator using TCP on remote host 192.168.14.1 port 4560
WARN [simulator] Please ensure port 4560 is not blocked by a firewall.
INFO [simulator] Waiting for simulator to accept connection on TCP port 4560
    
```

图 3-23 两幅由无人机前

这时候打开在 Windows 主机的 AirSim 模拟器，并在 Ubuntu 的 terminal 输入 **commander takeoff** 就能实现远程操控飞行了，具体更多命令需要参考 PX4 官网，这里不一一叙述。

3.10 用户界面和环境调整

在模拟器中，我们可以通过按 F1 来得知用户可以通过什么按键来对模拟器进行调整，如图 3-24 所示我们可以通过几个按键打开无人机深度摄像头、改变无人机视角等等，按 F10 则可以打开图 3-25 的界面来调整天气状况和风向，这将有助于在仿真环境下测试无人机在各种天气状况时的表现。



图 3-24 帮助界面

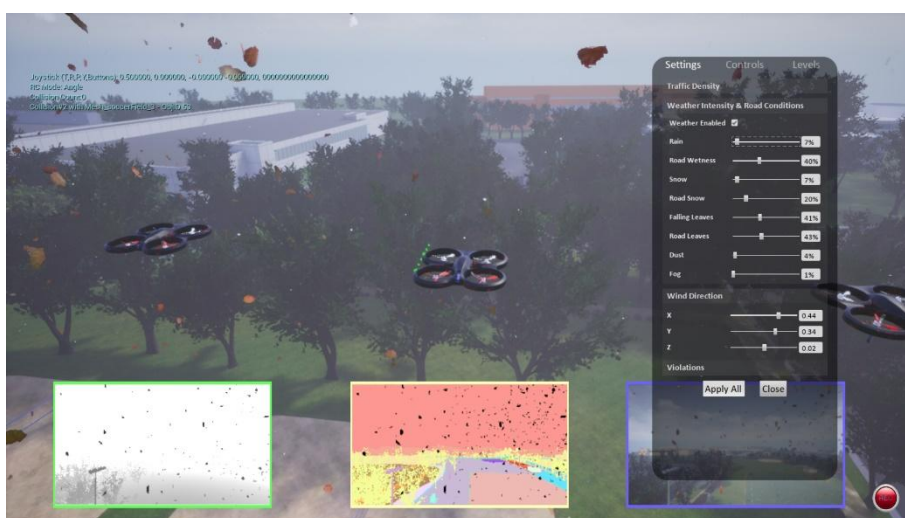


图 3-25 天气状况调控界面

3.11 数据采集和数据库

在 AirSim 模拟环境中，我们可以用系统自带的功能进行数据采集，按 R 键能开始录制并采集无人机在仿真环境时的数据，录制好的数据一般会存储在和 setting.json 一样的路径，录制的图像会以一堆 256×144 分辨率大小的图像搭配一个真值信息的文件呈现，要调整录制的图像大小只需要在 setting.json 稍作修改，但是调整的图像越大也会消耗更多计算机性能，这里已经够用因此不作调整。

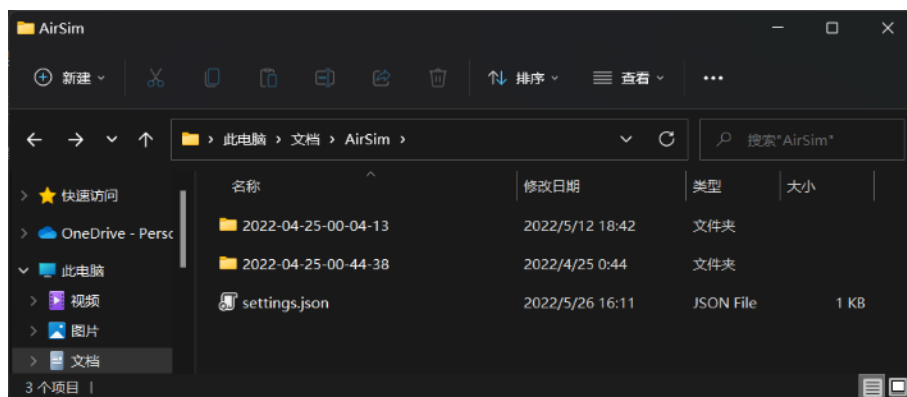


图 3-26 数据采集的存储路径

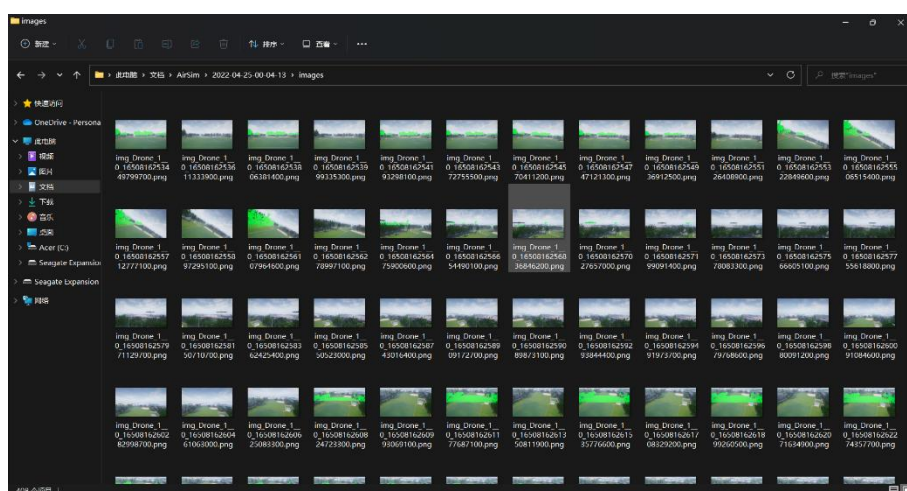


图 3-27 数据采集的图像

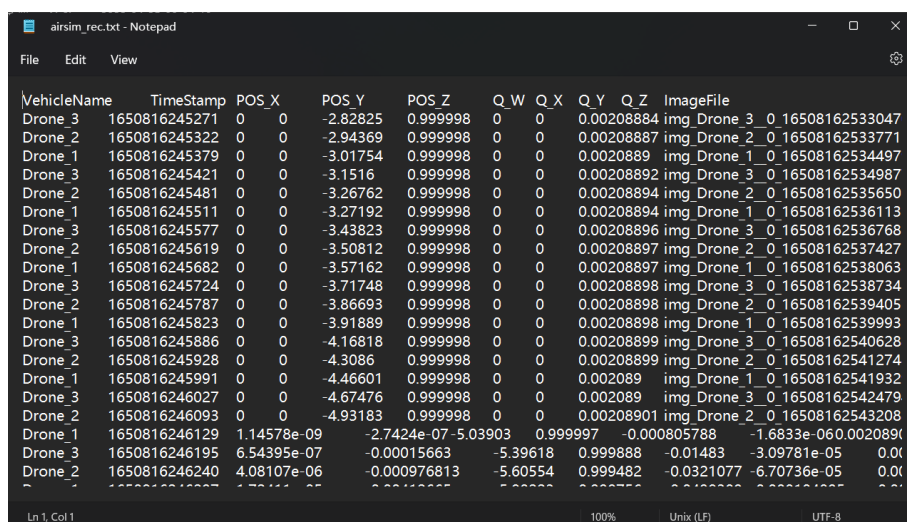


图 3-28 数据采集的真值信息

我们可以用录制的真值信息文件通过 Access 和内置的 Visual Basic 代码实现为其自动导入录制的数据库并构建一个数据库，每次导入数据时 Access 会根据真

值文件自动帮我们创建数据库的相关字段。因为每次录制信息时 AirSim 都会自动生成一个 airsim_rec.txt 文件，如果录制次数太多会导致数据不方便进行整理，而且每次导入数据进 Access 时也不便捷。为了解决这个问题，如图 3-29 的代码所示，我们可以通过 Visual Basic 自动遍历 Airsim 录制文件中所有的 airsim_rec.txt 文件以进行自动导入数据文件。数据库字段将包含“VehicleName”、“TimeStamp”、“POS_X”、“POS_Y”、“POS_Z”、“Q_W”、“Q_X”、“Q_Y”、“Q_Z”、“ImageFile”，各自字段的数据类型和意义如表 3-2 所示。数据库建好后可以直接通过点击在 Access 功能区的“外部数据”，选择“其他”后选择“XML 文件”来导出 XML 文件，通过 Access 导出的.xml 文件还包括.xsd 架构文件，这也将有助于他人更容易得知和还原数据库实施规则的架构标记，导出的 XML 文件是基于 Access 所构建的数据库架构解释，构造如图 3-31 所示。

```
Public Sub LOAD()
    Dim filepath As String, files As String, filename As String

    filepath = CurrentProject.Path & "\AirSim\*.txt"
    files = Dir(filepath)
    'Debug.Print filepath
    Do Until files = ""
        filename = CurrentProject.Path & "\AirSim\" & files
        DoCmd.TransferText acImportDelim, "airsim_rec", filename, True
        Debug.Print filename
        files = Dir '下一个文件
    Loop
```

图 3-29 通过 Visual Basic 代码实现自动导入在 AirSim 录制真值信息

表3-2 数据库的字段、数据类型和意义

字段	数据类型	意义
VehicleName	短文本	无人机的名称，在多无人机飞行时能迅速区分主机和从机的飞行信息
TimeStamp	长整型	时间戳，记录格式通常为：年-月-日-小时-分钟-秒
POS_X	双精度型	无人机所在帧的 X 坐标位置
POS_Y	双精度型	无人机所在帧的 Y 坐标位置
POS_Z	双精度型	无人机所在帧的 Z 坐标位置
Q_W	双精度型	四元素 W，表示无人机旋转时的角度
Q_X	双精度型	四元素 X，表示无人机旋转时位于 X 坐标的向量
Q_Y	双精度型	四元素 Y，表示无人机旋转时位于 Y 坐标的向量
Q_Z	双精度型	四元素 Z，表示无人机旋转时位于 Z 坐标的向量
ImageFile	短文本	所在帧录制的图像名字，在问题数据出现时能够帮助测试者找到相应图像以作更好的情况分析。



图 3-30 在 Access 数据库中导出.xml 文件

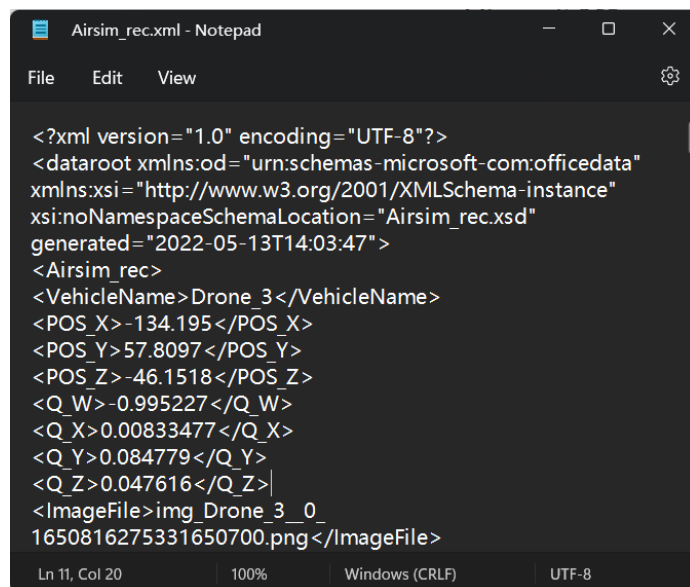


图 3-31 通过 XML 文件的标记了解数据结构和数据类型

视图

剪贴板

排序和筛选

记录

查找

文本格式

VehicleRec

VehicleName

TimeStamp

POS_X

POS_Y

POS_Z

Q_W

Q_X

Q_Y

Q_Z

ImageFile

Drone_3

-134.195

57.8097

-46.1518

-0.995227

0.00833477

0.084779

0.047616

img_Drone_3

Drone_2

-134.341

58.6392

-46.3004

-0.998471

0.0070635

0.0511397

0.0197736

img_Drone_2

Drone_1

-134.528

59.3469

-46.4229

-0.999374

0.00601652

0.0328131

-0.0117802

img_Drone_1

Drone_3

-134.787

60.1235

-46.7584

-0.998479

0.00494226

0.020578

-0.0509031

img_Drone_3

Drone_2

-134.919

60.9377

-46.9404

-0.994962

0.00355276

0.012323

-0.0994193

img_Drone_2

Drone_1

-135.089

61.5983

-47.086

-0.989607

0.00305357

0.00800394

-0.143546

img_Drone_1

Drone_3

-135.333

62.3278

-47.4483

-0.981311

0.00233898

0.00506898

-0.19235

img_Drone_3

Drone_2

-135.383

62.8275

-47.5599

-0.974418

0.00189159

0.0036627

-0.224707

img_Drone_2

Drone_1

-135.541

63.4456

-47.7007

-0.966084

0.00140484

0.00243189

-0.258213

img_Drone_1

Drone_3

-135.734

63.9667

-47.9842

-0.960124

0.0010929

0.00176407

-0.279568

img_Drone_3

Drone_2

-135.828

64.6604

-48.1336

-0.95283

0.000756275

0.00112945

-0.303501

img_Drone_2

Drone_1

-136.013

65.4034

-48.2818

-0.946246

0.000497496

0.000696171

-0.323448

img_Drone_1

Drone_3

-136.236

66.0501

-48.5765

-0.941812

0.000348985

0.000468237

-0.33614

img_Drone_3

Drone_2

-136.34

66.8014

-48.7108

-0.937483

0.000225536

0.000290635

-0.34803

img_Drone_2

Drone_1

-136.49

67.405

-48.8045

-0.934636

0.000156578

0.000196438

-0.355606

img_Drone_1

Drone_3

-136.71

68.0447

-49.1031

-0.932322

0.000108224

0.000132757

-0.361629

img_Drone_3

Drone_2

-136.787

68.6911

-49.2395

-0.930333

-0.000953846

0.0026959

-0.366704

img_Drone_2

Drone_1

-136.918

69.2239

-49.3498

-0.928838

-0.00564008

0.014472

-0.37016

img_Drone_1

Drone_3

-137.157

69.9557

-49.7203

-0.926632

-0.0158588

0.0399887

-0.373499

img_Drone_3

Drone_2

-137.218

70.5676

-49.8821

-0.924497

-0.0242107

0.0606651

-0.375552

img_Drone_2

Drone_1

-137.361

71.1988

-50.057

-0.921183

-0.0364268

0.0905843

-0.376881

img_Drone_1

Drone_3

-137.564

71.839

-50.421

-0.917871

-0.0462254

0.114369

-0.377221

img_Drone_3

Drone_2

-137.592

72.4381

-50.5903

-0.916287

-0.0497111

0.122604

-0.378042

img_Drone_2

Drone_1

-137.731

73.1384

-50.8044

-0.92024

-0.0338689

0.0826027

-0.38108

img_Drone_1

记录: 14

第 1 项/共 408 项

数据视图

搜索

数字

国

图 3-32 通过 Access 构建的数据库

3.12 小结

在本章节可以看到，我们通过 AirSim 就能基本达到系统的基础功能设计，但是这个系统依然需要进行许多优化，比如 AirSim 自带的飞行控制系统是个非常简单的控制系统，在现实世界我们当然是比理想状态之下还要不可预测的，因此如果需要进一步优化还需把 PX4 加入 AirSim 来进一步升级它的飞行控制系统，另外，目前本文的数字孪生系统的无人机编队算法的代码实现实际上也还需要进行调整，目前仅能实现简单的从机跟随飞行，和基础的避障功能。

第四章 效果测试与评估

4.1 系统测试效果

通过对 `setting.json` 的修改，我们可以基于该文本在系统中轻松生成多个无人机，但是每多出一个无人机都会更消耗计算机资源，若太多无人机将会让计算机在渲染时造成极大的负担从而造成系统画面严重掉帧，并进一步影响飞行数据结果，因此多无人机的情况我们会尽量节省计算机资源，除非测试用的计算机性能足够强大，因此如若要搭配摄像头和 Lidar 扫描等功能实现，应只给个别无人机搭配而不是全部无人机。图 4-1 为 25 个无人机生成的画面。本文在测试多无人机编队飞行时发现在多个无人机的状态下，通过 API 实现的编队飞行的代码甚至会出现个别无人机失去联系的情况而使系统进入安全模式，如图 4-2 所示，在通过 Python API 进行简单编队飞行时，第一个无人机会无视代码往后相撞。因此本文在进行基础测试时一般只用 1 到 3 个无人机实现。

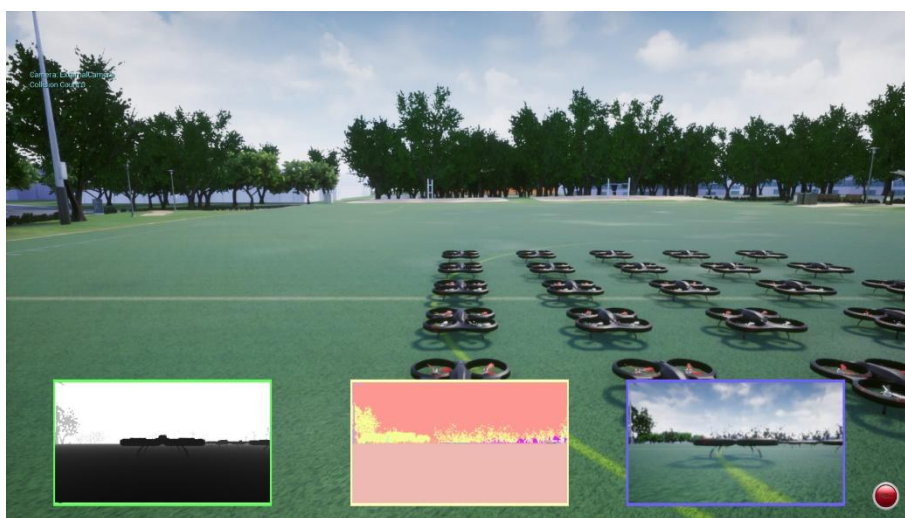


图 4-1 25 个无人机生成实现



图 4-2 前方无人机失去联系而与中间无人机相撞

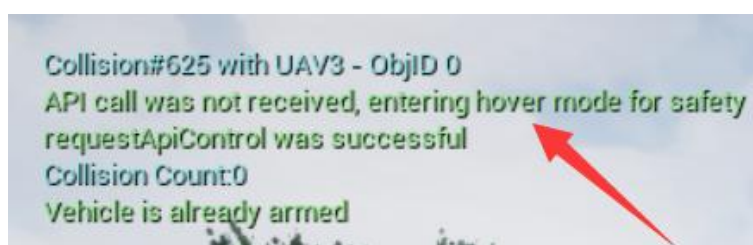


图 4-3 API 失去联系

除了通过 API 实现操控无人机飞行，在系统中，我们无法通过计算机键盘和鼠标来操控无人机。而 API 因为要基于代码实现飞行，若要进行更复杂的飞行情况是很不便捷的，当然也有 API 代码能实现直接让计算机键盘来操控无人机飞行，但这样的话反而会失去数字孪生的意义，因为现实无人机操控是不使用键盘的。因此在外接了 3-6 节所提到的 Xbox 游戏手柄之后，我们就能如图 4-4 所示的让无人机飞起来。Xbox 游戏手柄在经过 AirSim 官方的优化后操作手感并不比真实飞控遥控器差太多，手柄中左摇杆用作起飞降落，右摇杆用作飞行方向和旋转功能，当然如果要想实现更专业的飞控功能比起专业飞控遥控器来说依然还欠缺不少，但在本文进行数字孪生系统的简单测试来说已经足够使用。

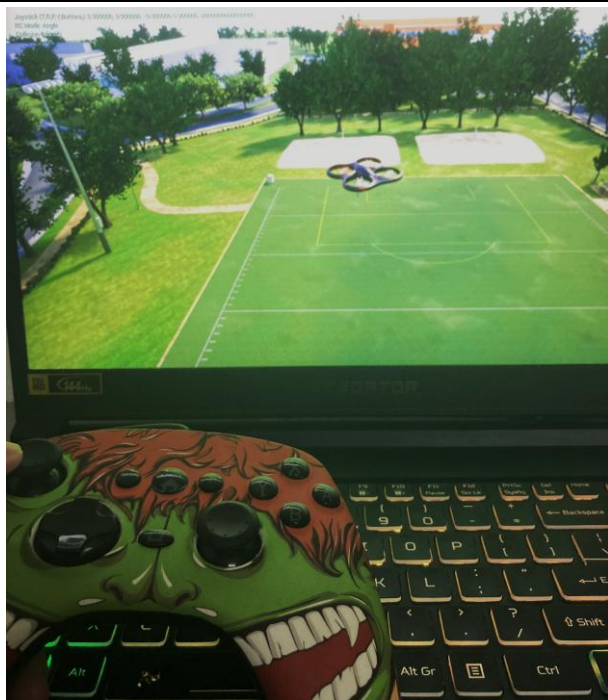


图 4-4 实际遥控画面

根据 3-10 节中图 3-24 所提到的 F1 帮助界面，我们可以对无人机视角进行测试，如下图所示，我们可以选择分别“FPV”（第一人称视角，见图 4-4），“fly with me”（跟随飞行视角，见图 4-5），“ground observer”（地面观察视角，见图 4-6），“chase with spring arm mode”（第三人称视角，见图 4-7）等视角。在“FPV”视角下我们没办法看见无人机，因此可以用作测试无人机收集数据时的机器视角。“ground observer”则顾名思义时操控者只能数字孪生系统的在地面上观察无人机飞行而无法跟随无人机的视角。“fly with me”和“chase with spring arm mode”分别在于“fly with me”不会锁定无人机的视角，因此操控者面向正面时无人机可能会面向反面，而“chase with spring arm mode”则如同“FPV”一样会锁定前方视角，区别在是否能看见无人机。通过这些视角，我们能够模拟出并了解在各种情景下对无人机收集数据的情况。



图 4-4 FPV 视角



图 4-5 fly with me 视角



图 4-6 ground observer 视角

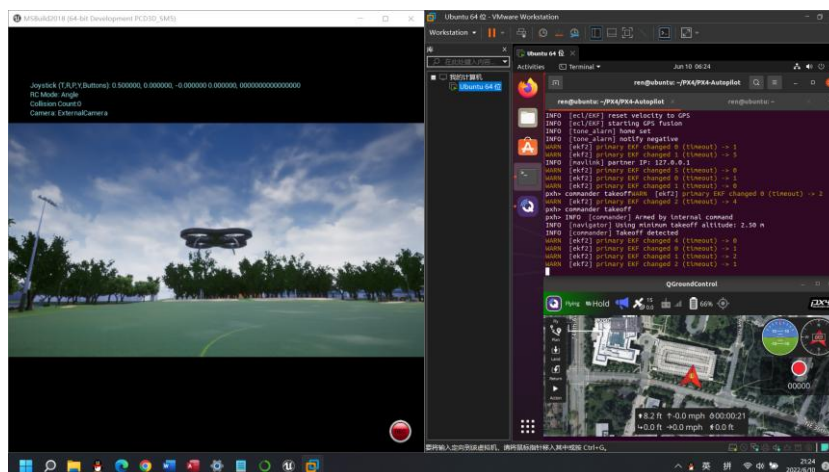


图 4-9 远程通信测试

4.2 目标完成度评估

最后，我们先对在 1-3 节中所提出的目标完成情况进行评估，并分别针对 Windows 主机端和 Linux 客户端的操作进行对比，具体内容如表 4-1 和表 4-2 所示。

表4-1 目标完成情况

目标	完成情况
构建可视化模拟场景一。	已完成，系统中能模拟一个四旋翼无人机编队飞行全局的动态视图场景，该场景视图四周有数据显示区域和操作区域可以显示四旋翼无人机编队飞行全局整体动态演进的相关的状态信息，可进行相应的功能操作。
构建可视化模拟场景二。	已完成，系统中能模拟具体某一架正在运行的飞机，该场景视图四周有数据显示区域和操作区域可以显示该架飞机运行时的相关状态信息，可进行相应的功能操作。
构建统一的数据库。	已完成，所有数据都可统一的访问和存储。
构建信息交互的接口模块。	已完成，客户端可与主机端进行交互。
构建对外共享的元数据目录。	已完成，基于 XML 文件创建的元数据目录。

表4-2 主机端和客户端对比

功能	Windows 主机端	Linux 客户端
----	-------------	-----------

西北工业大学本科毕业设计（论文）

无人机能否编队飞行？	能，只要操控一个主无人机，所有从无人机会自动跟随。	能，只要操控一个主无人机，所有从无人机会自动跟随。
能否切换无人机？	能，只要通过简单的 API 就能实现随意切换任何一个从无人机作为主无人机，但是若要 进行数据录制则需要提前通过代码实现给每个无人机搭载摄像头。	否，在 Linux 应用 Px4 操控无人机编队飞行时绑定了主飞机，因此也仅能操控主飞机，若是强行切换则可能会丢失连接。
是否能对对场景进行操作？	能，主机客户端能够方便对场景进行任何操作，我们可以在主机的平台上进行任意本文前面章节提过的任何环境操作。	有限操作，在 Linux 客户端中没办法直接对虚拟场景中的环境（如风向、天气等）进行任何调整，但能通过 API 操作主机端进行调整。
是否能进行数据录制？	能。	有限操作，需要通过 API 对主机端进行录制。
数据能否共享？	能。	能。

第五章 总结和展望

在本文，我们基本是基于 AirSim 完成数字孪生系统开发的大多数功能，AirSim 是微软一个开源、跨平台的自动系统仿真平台，开发的交通工具对象包括自动驾驶汽车、轮式机器人、无人机，甚至静态物联网设备。它是一款作为 Epic Games 的虚幻引擎的插件。甚至还有一个针对 Unity 游戏引擎的实验版本。

如本文前面所显示的，我们能通过 AirSim 解决开发者在开发自主系统时面临的两个主要问题：训练和测试系统所需要大量数据、在模拟器中实现各种调试功能。通过 AirSim 能为开发人员提供一个拥有各种培训经验的平台，以便自主系统在部署到现实世界之前可以实现不同的场景中的测试。

在本文，我们还了解到 AirSim 通过支持 PX4 飞行控制器来实现的半实物仿真，提供了物理和视觉上的逼真模拟。PX4 是一个可以容易扩展，以适应各种新型自动驾驶汽车、硬件平台和软件协议。其可扩展的架构也允许我们快速添加自定义系统模型和新的传感器到模拟器之中。

未来，相信数字孪生系统也会更进一步的发展并实现更加复杂的功能，目前不止在无人机领域，许多大企业也在深入研究数字孪生，如特斯拉的自动驾驶。想象一下如果没数字孪生技术要如何训练一架汽车而不会造成任何损失呢？如果没有数字孪生技术，我们要如何用更低的成本去测试航空飞行器呢？甚至各种各样的仪器开发都可见数字孪生的身影，可见数字孪生技术的未来对于人类来说至关重要，它将帮助我们在更低成本的成本达到更快的技术突破。

参考文献

- [1] 陶飞,刘蔚然,刘检华,等. 数字孪生及其应用探索[J]. 计算机集成制造系统,2018,24(01):1-18.
- [2] IBM. 什么是数字孪生? [EB/OL]. [2022-5-20]. <https://www.ibm.com/cn-zh/topics/what-is-a-digital-twin>.
- [3] Michael Grieves. Digital Twins[EB/OL]. 2019-12-2[2022-5-20]. <https://portal.manufacturingleadershipcouncil.com/2019/12/02/digital-twins/>.
- [4] 岳基隆,张庆杰,朱华勇. 微小型四旋翼无人机研究进展及关键技术浅析[J]. 电光与控制,2010,17(10):46-52.
- [5] 冯聪. 无人机集群编队交互式仿真平台的设计与实现[D]. 天津大学,2018.
- [6] 纪广,郝建国,张中杰,等. 四旋翼无人机飞行过程孪生仿真研究[J/OL]. 计算机工程与应用 :1-8[2022-05-27]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20211012.1412.004.html>
- [7] Chris Lovett. Unreal AirSim Setup from scratch[EB/OL]. 2017[2022-5-20]. <https://www.youtube.com/watch?v=1oY8Qu5maQQ>.
- [8] Microsoft Research. AirSim Setting[EB/OL]. [2022-5-20]. <https://microsoft.github.io/AirSim/settings/>.
- [9] PX4. Ubuntu Development Environment[EB/OL]. 2021[2022-5-20]. https://docs.px4.io/master/en/dev_setup/dev_env_linux_ubuntu.html.
- [10] Dronecode. Download and Install[EB/OL]. [2022-5-20]. https://docs.qgroundcontrol.com/master/en/getting_started/download_and_install.html.
- [11] Daniel Slotta. Global market share of consumer and commercial drone manufacturers in March 2021, based on sales volume[EB/OL]. 2022-1-13[2022-5-20]. <https://www.statista.com/statistics/1254982/global-market-share-of-drone-manufacturers/>.
- [12] Shital Shah, Debadeepta Dey, Chris Lovett, et al. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles[EB/OL]. 2017-5-15[2022-5-20]. <https://arxiv.org/abs/1705.05065>.
- [13] Kaveh Fathian, Emily Doucette, J. Willard Curtis, et al. Vision-Based Distributed Formation Control of Unmanned Aerial Vehicles[EB/OL]. 2018-9-1[2022-5-20]. <https://arxiv.org/abs/1809.00096>.

- [14]Zhao Xuhui. AirSim 笔记 1: AirSim 最速上手教程[EB/OL]. [2022-5-20].
<http://zhaoxuhui.top/blog/2021/11/30/airsim-note1-easiest-hand-on-tutorial.html>.
- [15]宁子安. airsimsim & unreal 仿真平台 [EB/OL]. [2022-5-20].
<https://www.zhihu.com/column/multiUAV>.

致 谢

非常谢谢钟冬老师愿意给我选题的机会，在这期间也耐心指导我，刚开始的时候一直担心自己问一些基本但自己不了解的问题可能会严重打扰到老师，但是老师都很愿意细心的跟我解释这些基础也让我产生了信心，谢谢秦同学在这期间跟我一起互相帮助一起努力做各自的毕业设计，有个伙伴一起打怪的感觉真的会让自己没这么压力，谢谢爸爸妈妈一直鼓励我，不管我做什么都能谅解并支持，谢谢弟弟常常给与我情绪上支持，最后我只想表达给那些所有帮助过我的人，我爱你们，衷心希望你们生活能持续安康快乐。

毕业设计小结

本次毕业设计真的让我学到了很多東西，我的专业是电子商务，在这期间从未接触过数字孪生和无人机的相关课题，因此在刚开始做毕业设计时真的感觉自己不知从何开始和没有相关基础知识而面临很大的压力，但是随着坚持研究和查找文献，会发现毕业设计并只要肯花时间做实际上并没有想象的难，当然这都要归功于老师和同学的帮助。其次，在本次毕业设计的过程中，我也渐渐的体会到很多乐趣，从一开始的一头雾水到实验中成功操控无人机，我真的兴奋的几天睡不着，尽管系统主要借助于很多前人的研究，但是这个过程依然给了我满满的满足感，而且也是阶梯式的，从一开始不知道怎么要通过API接口操控系统中的无人机，到后面实现无人机编队飞行、添加各种功能甚至实现远程通信，每一点对我来说都是困难点，但是每次完成都像打过了一个重要BOSS一样让人充满成就，希望以后我在面对任何困难也能像现在一样有挑战的思维。

附 录