



西北工业大学

# 本科毕业设计论文

题 目

集群无人机协同 SLAM

专业名称 飞行器控制与信息工程

学生姓名 刘昭宏

指导教师 布树辉

毕业时间 2022 年 6 月

## 摘 要

摘要正文

**关键词：** 无人机集群，协同 SLAM

## ABSTRACT

摘要正文-英文

**Key Words:** Flight control model, Quaternion, Three-dimensional visual simulation system, Loop simulation

## 目录

<b>第一章 绪论</b>	<b>4</b>
1.1 研究背景	4
1.2 研究内容及论文结构	4
<b>第二章 ROS 与 PX4 介绍</b>	<b>7</b>
2.1 ROS 介绍	7
2.1.1 ROS 的消息机制	7
2.1.2 gazebo 仿真	9
2.2 PX4 AutoPilot	9
2.2.1 FailSafe 机制	10
2.2.2 EKF 与飞行模式	10
2.2.3 联合 MAVROS 的 Offboard 模式	10
<b>第三章 SLAM 系统设计</b>	<b>11</b>
3.1 SLAM 系统	11
3.1.1 SLAM 的分类	11
3.1.2 相机参数及成像原理	11
3.1.3 视觉 SLAM 的基本步骤	11
3.2 ORB-SLAM2	11
3.2.1 ORB 特征点及描述子	11
3.2.2 ORB-SLAM2 的主要进程	11
3.3 CCM-SLAM	11
3.3.1 CCM-SLAM 的结构	11
3.3.2 Client 与 Server 机制	11
3.4 多机协同及地图融合方案	11
3.4.1 算法原理	11
3.4.2 编程实现	11
<b>第四章 多无人机 SLAM 仿真</b>	<b>12</b>
4.1 gazebo 仿真环境配置	12
4.1.1 场景	12
4.1.2 launch 文件	12
4.2 单机 SLAM 仿真	12
4.2.1 EKF 设置及启动仿真	12
4.2.2 视觉定位的坐标变换	12
4.3 多机 SLAM 仿真	12

第五章 实验与评估 .....	13
第六章 总结与展望 .....	14
6.1 全文总结 .....	14
6.2 对未来工作的展望 .....	15
参考文献 .....	16
致谢 .....	17
毕业设计小结 .....	18

## 第一章 绪论

### 1.1 研究背景

当今阶段，无人机（Unmanned Aerial Vehicle）技术迅速发展，在单架无人机上可以集成更多的系统，意味着对于单机更强大的功能。单架无人机也已经被广泛应用于灾害救援、监控巡查、环保监测、电力巡检、交通监视、农业植保等领域。但是，面对复杂的应用环境和多样化的需求，单架无人机受自身软硬件条件的限制，仍然具有一些局限性；为了弥补单架无人机的局限性，由多架相同或不同型号的无人机组成多无人机系统，即无人机集群，协同定位，共同完成任务；

通过集群的方式，能最大地发挥无人机的优势，又能避免由于单架无人机执行任务不佳或失败造成的不良后果，提高任务执行效率，扩展新的任务执行方式，从而达到提高系统可靠性，增强任务执行效果的目的。对于无人机的自主导航，能够在进入未知的环境时掌握无人机的位置和姿态是使其成功的关键。尽管 GPS 对于掌握无人机的位置有巨大的帮助，但仍存在普适性有限和准确度不高的问题；在一些特定场景下，比如室内狭小空间，对定位精度要求很高，GPS 定位的局限性就被显露出来。而 SLAM 技术则可以仅通过自身携带的传感器，来完成这一任务，同时达到一定的精度；

SLAM (simultaneous localization and mapping) 技术，即同时定位与建图，已有三十多年的研究；SLAM 最早由 Smith、Self 和 Cheeseman 于 1988 年提出。SLAM 指的是机器人在未知环境中从一个未知位置开始移动，在移动过程中根据位置和地图进行自身定位，同时在自身定位的基础上建造增量式地图，实现机器人的自主定位和导航。由于其重要的理论与应用价值，被很多学者认为是实现真正全自主移动机器人的关键 [1]。

但是单平台 SLAM 受到传感器性能的限制，存在两点不足：一是测量距离受限，单平台常用的传感器如激光雷达，其最远有效距离为 200 米，不能够满足大场景定位建图的需要，任务效率比较低；二是单平台构建出的三维点云相对稀疏，不能表现出足够的场景信息 [2]。

因此集群无人机协同 SLAM 方案有望解决单机存在的制约问题，主要表现在两个方面：一是多机意味着多传感器，能在大范围场景进行同时定位与建图；二是多机协同 SLAM 可以通过建图覆盖的检测，构建更加稠密和精细的点云地图。

目前单机同时定位与建图已经相对成熟，但是多机 SLAM 由于其控制复杂、数据传输量大、信息处理速度受限、关键数据融合效率低等问题，仍然需要大量的理论研究和实验。

### 1.2 研究内容及论文结构

本文研究目标旨在实现一套能够在室内高精度环境或 GPS 拒止环境下使用视觉进行多机定位和大范围建图的多无人机协同 SLAM 的方案；其中：

1. 在 SLAM 方面：掌握一些优秀的开源方案，选择各自优点做出一定的融合。并且有一套针对地图融合的方法。
2. 在仿真方面：在 ROS 的 gazebo 仿真平台中实现一定的集群控制方法，能够控制多个无人机协同完成同时定位与建图的任务。
3. 在真机方面：实现单机的视觉 SLAM；在安全的前提下实现双机协同 SLAM，将仿真环境下的协同 SLAM 算法在真机上完成验证，得到场景地图

本研究内容是多机协作进行定位与建图，多机协同 SLAM 能大大提高任务进行的效率，但由于无人机数量多，协同上存在一定困难；研究内容分为三个模块：SLAM 模块、仿真模块和真机模块。

SLAM 模块的主要内容是实现一套可协作的 SLAM 方案，实现的步骤有：

1. 研究传统的视觉 SLAM 的特征点提取、匹配、初始化、后端优化等技术，研究机器人的位姿估计技术；研究并了解 SLAM 技术的整体框架
2. 研究 CCM-SLAM 方案，重点研究其协同的机制和方法，服务端到子端的信息传递和接口设计等
3. 研究 VINS-Fusion 方案中的 VIO 方法，研究如何利用 IMU 与相机数据联合进行更加准确的位姿估计

仿真模块的主要内容是在 ROS 的 gazebo 中研究如何实现多机协同的同时定位与建图，实现的步骤有：

1. 首先研究 PX4 和 MAVROS 之间的通信方式，ROS 的话题发布和订阅方式，研究如何用程序解锁一架无人机、使其进入 Offboard 模式、起飞悬停并降落
2. 研究如何用程序发布话题，控制无人机按照航路点飞行
3. 研究如何构建多机的仿真环境，如何对多机进行控制，其控制策略的选择，即集中式或分布式的多机编队控制
4. 研究如何更改无人机的定位设置，将其从 GPS 定位改为视觉 SLAM 定位；并且完成单机的摄像头内容读取
5. 研究如何在 gazebo 中载入其他场景，在场景中控制无人机飞行，并且对拍摄到的画面进行建图，完成自身定位
6. 研究如何在 gazebo 中完成多机基于视觉的同时定位与建图，并且拼合地图，用第三方软件显示；研究多机的联合优化与协同方法

真机模块的主要内容是控制无人机的协同飞行及通信，实现的步骤有：

1. 研究无人机通过 MAVROS，MAVLINK 与地面站的通信方法，尤其是用于 SLAM 的关键数据的传输
2. 研究多无人机与地面站之间的、多无人机之间的数据传输

3. 研究多无人机之间的可变基线控制技术，如何设计一个详细的算法控制基线距离

基于研究内容的层级关系，本论文的结构主要由四层构成，如图1-1所示。

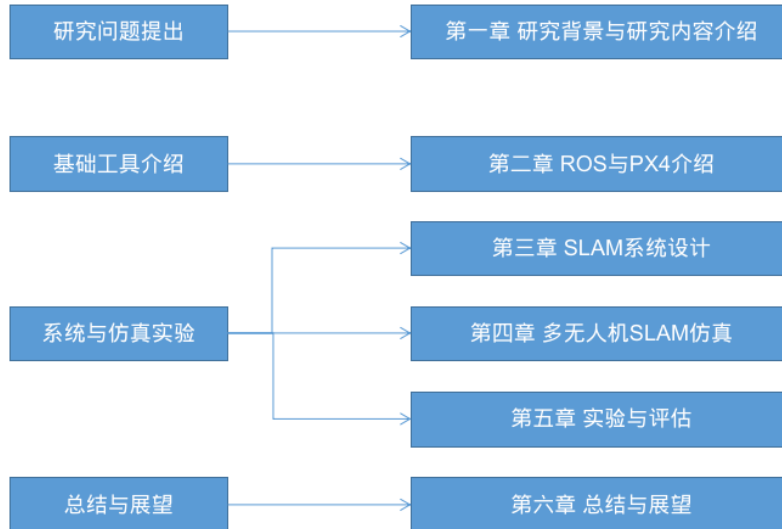


图 1-1 论文结构

第一章：主要介绍研究背景，进行问题的提出，从而引出本文的研究内容和研究目标；从宏观的角度概括本文的内容，并且对问题做出概述。

第二章：主要介绍研究中用到的 ROS（Robotics Operating System，机器人操作系统）和 PX4 AutoPilot 飞控软件系统。

第三章：主要介绍 SLAM 的原理和作用，SLAM 系统的基本流程，优秀的开源 SLAM 方案和地图融合的设计。

第四章：主要介绍仿真实验的情况，在 ROS 的 gazebo 环境中进行。

第五章：主要介绍真机实验的情况，并且做出相应的评估。

第六章：总结研究过程中的创新点和不足之处，提出进一步研究的大致方向，展望未来的研究工作。



## 第二章 ROS 与 PX4 介绍

### 2.1 ROS 介绍

本节主要对 ROS 平台进行介绍，包括 ROS 核心的消息机制和研究中要用到的 gazebo 仿真平台。

21 世纪开始，随着人工智能研究的发展，催生出了一批智能机器人的研究项目；ROS 诞生于 2007 年斯坦福大学 AI 实验室 Morgan Quigley 的 STAIR (Stanford Artificial Intelligence Robot) 项目，其期望构建一个基于移动机器人 + 机械臂的原型；该项目于 2008 年受到 Willow Garage 公司关注，其决定用商业化手段来推进机器人的发展，使机器人平台能够更快地走进人们的日常生活；Willow Garage 接手该项目后两年，2010 年第一代 ROS 即 ROS1.0 发布；2013 年，OSRF (Open Source Robotics Foundation) 接管了 ROS 的维护工作和版本的升级工作，随后至 2018 年间，ROS 的 Indigo、Kinetic 和 Melodic 版本相继发布。

ROS 即 Robotics Operating System，是一个针对机器人的开源、元级操作系统，在某些方面，ROS 更像是一种机器人框架 (robot framework)；它提供类似于操作系统的服务，包含底层的驱动程序管理、底层的硬件描述，随后上升到软件程序之间的消息传递、功能包的管理和发布、也提供用于获得、编译、编写和多设备跨计算机运行代码所需的库等。换言之，ROS 是由一套通信机制，开发工具，一系列应用功能和一个庞大的生态系统组成的集合，其目标为提高机器人研发中的软件复用率，不断完善他人的工作，进行更好的开发。

#### 2.1.1 ROS 的消息机制

ROS 提供了一套松耦合分布式通信机制，这种分布式处理框架 (又名 Nodes)，是以多个节点及节点之间的通信组成的。其中，节点 (Node) 和节点管理器 (ROS Master) 是 ROS 的核心概念，若干个节点在节点管理器下构建起来，共同实现特定的功能。

每一个节点是一个独立的执行单元，由可执行文件构成，在程序中需要声明节点的名称；节点的名称必须唯一，否则 ROS 会舍弃掉时间节点靠前的节点；节点执行具体的任务进程，比如单目的 ORB-SLAM2 中，其节点为 Mono，SLAM 的任务仅靠一个节点完成。

节点管理器是节点的控制中心，其作用是辅助节点的查找，帮助节点之间建立通信连接；还能提供节点的命名和注册等服务，以及提供了能够存储全局变量的配置的参数服务器。

如图2-1所示，节点在经过节点管理器注册后，可以建立节点之间的通信；常用的节点之间通信方式有两种，为话题 (Topic) 通信和服务 (Service) 通信：

1. 话题通信是异步通信机制，数据为单向传输；数据的流向为发布者 (Publisher) 到订阅者 (Subscriber)；完成话题通信需要定义一个话题 (Topic) 及其消息 (Message) 的内容，之后通过发布者 (Publisher) 发布该话题，并且订阅者

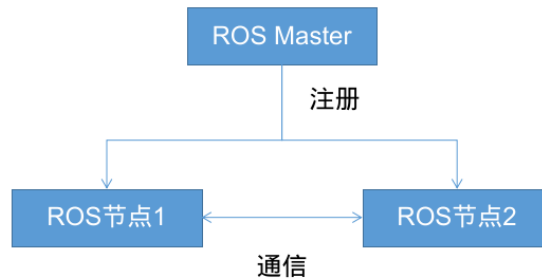


图 2-1 ROS 中的节点及通信

(Subscriber) 订阅该话题的操作，完成数据的传输，消息的数据结构由.msg 文件定义；话题通信可以完成多对多的信息传递。

2. 服务的通信机制则为同步，数据为双向传输；数据的流向为客户端 (Client) 与服务器 (Server) 之间的交互；完成服务的通信需要客户端向服务器发送请求，服务器完成任务处理后，向客户端返回应答数据，表示请求和应答的数据结构定义在.srv 文件中；服务通信一般用于逻辑判断，比如询问一项任务是否执行完毕，是一对多的节点处理关系。

发布和订阅话题的方法，发布者和订阅者类似，以发布者为例：先实例化一个发布者对象，定义发布的话题名称、数据类型和队列长度，最后对消息进行定义并发送，简单的逻辑代码如下：

```

ros::NodeHandle n; // define ros node handle
// define a publisher
ros::Publisher pub = n.advertise<'message type'>
    ("topic name", 'queue length');
// publish message
pub.publish(message);
  
```

需要注意的是，订阅者则需要声明并定义一个回调函数，在实例化 Subscriber 的对象后，通过 ROS 的 spin() 函数，循环等待回调函数获得话题消息。

客户端-服务器模型下的服务通信，则比话题的发布和订阅复杂；客户端的编程实现中，需要设置阻塞函数，其作用是直到发现对应的服务时才向下进行，否则程序被截止在该位置；如果对应的服务被发现，阻塞函数通过，之后创建客户端并且进行数据的设置，完成服务调用的请求，其代码实现如下：

```

// wait for right service
ros::service::waitForService("service name");
// create a client, connecting to service
ros::ServiceClient client = n.serviceClient
    <'data type'>("service name");
// call service
client.call(srv);
  
```

服务器的实现与订阅者类似，需要一个回调函数，如果收到了客户端发来的请求，则会触发回调函数，程序向下进行，否则将循环等待回调函数收到客户端发来的请求。

除此之外，ROS 中还有参数（Parameter）或参数服务器的概念，其作用类似全局共享字典，节点可以进行访问，适合存储一些和系统配置相关的静态非二进制的参数，以供节点读取。

### 2.1.2 gazebo 仿真

## 2.2 PX4 AutoPilot

本文将飞行摇杆（Joystick）数据读取作为四旋翼飞控模型的输入，Joystick 数据的数据主要是四旋翼的姿态角（滚转角，偏航角，俯仰角）和四旋翼的推力。因此，本文通过 C++ 编程，实现飞行摇杆数据的输入。由如下代码所示：

```
class HAL_JoyStick: public ZY::Thread {
public:
    HAL_JoyStick() {
        m_devID = -1;
    }
    HAL_JoyStick(int dev) {
        m_devID=dev;
    }
    virtual ~HAL_JoyStick() {}
    int open(int devID);
    void run();
    int close(void);
    int read(JS_Val *jsv)
    {
        data_mutex.lock();
        *jsv= m_JSVal;
        m_JSVal.dataUpdated = 0;
        data_mutex.unlock();
        return 0;
    }
public:
    int          m_devType;
    int          m_devID;
    int          m_devFD;
    char         number_of_axes;
    char         number_of_btns;
protected:
    ZY::Mutex    data_mutex;
    JS_Val       m_JSVal;
```

}

2.2.1 FailSafe 机制

2.2.2 EKF 与飞行模式

2.2.3 联合 MAVROS 的 Offboard 模式

## 第三章 SLAM 系统设计

### 3.1 SLAM 系统

#### 3.1.1 SLAM 的分类

#### 3.1.2 相机参数及成像原理

#### 3.1.3 视觉 SLAM 的基本步骤

### 3.2 ORB-SLAM2

#### 3.2.1 ORB 特征点及描述子

#### 3.2.2 ORB-SLAM2 的主要进程

### 3.3 CCM-SLAM

#### 3.3.1 CCM-SLAM 的结构

#### 3.3.2 Client 与 Server 机制

### 3.4 多机协同及地图融合方案

#### 3.4.1 算法原理

#### 3.4.2 编程实现

## 第四章 多无人机 SLAM 仿真

### 4.1 gazebo 仿真环境配置

#### 4.1.1 场景

#### 4.1.2 launch 文件

### 4.2 单机 SLAM 仿真

#### 4.2.1 EKF 设置及启动仿真

#### 4.2.2 视觉定位的坐标变换

### 4.3 多机 SLAM 仿真

## 第五章 实验与评估

## 第六章 总结与展望

### 6.1 全文总结

飞行控制是保证飞机安全飞行的核心部分，是衡量飞机飞行品质的重要指标。飞行控制技术的高低决定了无人机的飞行品质。本文主要研究了四旋翼无人机及固定翼无人机的飞行控制，基于 FlightGear 飞行模拟软件搭建飞行仿真平台，实现半物理仿真。本文主要做的工作如下：

1. 简述飞行控制过程中坐标系转换的原理以及坐标系旋转矩阵公式推导过程。
2. 推导四旋翼旋转矩阵，为六自由度四旋翼飞控模型铺垫。
3. 概述 FlightGear 软件组件，程序框架及软件优势。
4. 进行四旋翼无人机的飞控建模，分为三个方面。第一个方面，是实现飞行摇杆的数据传入过程，对于四旋翼而言，主要是飞行姿态角传入及油门数据读取。第二部分，是建立六自由度非线性的飞行动力学模型，实现 FlightGear 外部飞控模型的实时解算，达到实时控制四旋翼无人机效果。第三部分，是基于 FlightGear 搭建的三维视景仿真系统的介绍。
5. 基于 FlightGear 内部的飞行动力学模型 JSBSim，对固定翼无人机进行控制。其过程同样分为三部分，第一部分与第三部分分别于四旋翼无人机相同，主要是第二部分的飞行动力学模型。本文对 JSBSim 模型进行配置，实现 JSBSim 与飞行摇杆数据之间的接口通信编程。
6. 详细讲述 FlightGear 飞行器驱动的步骤以及系统实施的条件。以四旋翼为对象，从飞行器模型载入到 FlightGear 通信模块的实现最后到 FlightGear 三维视景系统的仿真效果展现。介绍了在 Linux 操作系统下，如何使用 FlightGear 进行半物理仿真的操作步骤。
7. 本文对无人机飞行控制的工作作出了一些展望，尤其是基于视觉的无人机飞行控制，结合 FlightGear 飞行仿真软件，实现功能更为强大的半物理仿真的飞行控制。



## 6.2 对未来工作的展望

根据本文的分析，可以发现无人机飞行控制技术已经非常成熟，但也存在不少可以改进的地方，基于视觉的无人机飞行控制，可以作为未来研究的重点，概括起来主要有如下几个方面：

1. 可行性。无人机内置水平和竖直两个摄像头，可以完成对无人机所处环境图像的采集，这满足了引入计算机视觉方法的前提条件；
2. 高效性。无人机采集的图像，能够通过无人机内部自建的 wifi 无线网络实时传送至计算机，从而可在计算机上运行视觉处理算法，充分发挥计算机强大的计算能力，实现对有用信息进行解算。这是非常关键的一点，解决了计算机视觉方法数据处理量大，难以利用无人机自带的处理器芯片进行求解的问题；
3. 自主飞行性。无人机在实际飞行时，必须由人实时发出控制信号才能保证其飞行。而我们希望无人机在需要很少的人为引导，甚至是没有人干预的情况下，同样可以安全平稳飞行，即减弱人在整个控制系统中所扮演的角色。利用计算机视觉技术取代人在控制系统中的作用，就显得尤为重要。
4. 通用性。摄像机善于捕捉运动信息，而传统的传感器则较吃力，从应用的角度来看，视觉信号的抗干扰性能很好。此外，视觉导航既适用于室内环境，也适用于室外环境，通用性好。
5. 合理性。无人机设计的完整合理性，使我们不需要考虑其电子元件级的实现、气动布局、力学建模以及电机转速的控制方法，可直接通过对俯仰角、滚转角、偏航角以及竖直方向速度的控制实现对无人机的各种控制，这大大简化了我们的工作，使我们可以专注于无人机与计算机视觉方法的结合。

## 参考文献

- [1] 于琰平. 基于 FlightGear 的四旋翼无人机三维可视仿真系统研究 [J]. 天津大学, 2010.
- [2] 刘鹏. 基于 FlightGear 的无人直升机飞行仿真技术研究 [J]. 南京航空航天大学, 2011.
- [3] 李海泉. 小型无人机飞行力学建模及虚拟训练平台的建立 [J]. 南京航空航天大学, 2011.

## 致 谢

首先要感谢我的导师布树辉老师。感谢布老师在整个毕设过程中的耐心指导，感谢布老师在整个论文进展过程中，提供的文献资料和实验平台，感谢布老师在整个毕设过程中的宝贵意见。与布老师交流过程中，不断的加深对问题的理解与认识，不断的提高自己解决问题的能力。对于很多女同学而言，并不会去选择编程方向作为自己的毕设，我只是因为一时的兴趣才去选择了它。回想起来，自己从一月份的一无所知到三月份的懵懵懂懂再到如今六月份的豁然开朗，从开始的压力山大到中途的排斥抵触再到现在的一往无前，在布老师的悉心教导下，让我懂得科研的路途毕竟是曲折而坎坷的，需要一份对待科研的严谨与热情，去迎风破浪，最终，必会柳岸花明。

同时要感谢教研室的赵勇师兄，在坐标系变换的编程实践中，赵勇师兄提供的四元数算法，让问题瞬间豁然开朗，感谢赵勇师兄的不吝赐教。感谢韩鹏程，程少光，王磊师兄，在我代码调试过程中，悉心的指导，在我压力很大的时候，热心的开导，分享他们的科研经历，感谢你们的鼓励与支持，让我可以一步一步慢慢的成长。

另外要感谢 Curt Olson 等飞行爱好者们，是他们创造了 FlightGear 这个功能强大的开源的飞行模拟软件；感谢为 Linux 贡献代码的程序员们，这个自由免费的平台为我完成毕设提供了不少便利；感谢清华大学王磊博士，他创作的 L<sup>A</sup>T<sub>E</sub>X 模板使我的论文的排版得以顺利完成。

最后感谢我的家人对我一如既往的关心和支持，感谢我的男朋友对我一如既往的支持与鼓励。

## 毕业设计小结

本次毕业设计不仅仅是对我大学四年来学习知识的一个总结，更是对我知识学习能力一次拓展与提高的过程。本次毕设主要以四旋翼无人机飞行控制为核心，学习了如何通过 C++ 编程实现坐标系的转换，旋转矩阵的运算。实践了通过编程实现半物理仿真的过程，以及提高了如何将数学公式转换为编程语言的能力。同时，了解了 FlightGear 飞行模拟软件的通信过程，学习如何实现 UDP 的通信编程。因为 FlightGear 是一款功能强大的开源软件，更适合在 Linux 操作系统下进行使用，所以，学习了如何在 Linux 操作系统下进行 FlightGear 软件的安装，启动及 Shell 文件的编写。在 Linux 系统下，更方便的使用 FlightGear 飞行模拟软件与 Joystick 摇杆结合，实现对四旋翼无人机的飞行控制。同时，本次毕设也对固定翼无人机的飞行控制进行了学习，通过 FlightGear 自带的 JSBSim 飞行动力学模型，结合 Joystick 飞行摇杆，通过 TCP/IP 编程实现对 FlightGear 的通信过程，完成对固定翼无人机的飞行控制。

本次毕业设计中印象比较深刻的有飞行控制过程中，坐标系转换的问题，必须保证坐标系选择的一致性，学习了如何通过 C++ 编程实现对向量矩阵的运算过程。同时，通过对 FlightGear 通信模块的学习，了解了 UDP, TCP/IP 协议，通过 C++ 编程实现 Joystick 飞行摇杆数据到 FlightGear 通信的过程。对六自由度非线性飞行动力学方程的推导过程，进行了十分深刻的了解。同时，对如何将数学公式变成编程语言的编程能力，有了一定提高。

经历了本次毕设，我对无人机飞行控制的过程有了一定了解，对 Linux 操作系统有了一定认识。在今后的学习过程中，会更加深入的学习飞行控制理论，提高自己的编程水平，熟练使用 FlightGear 飞行模拟软件，实现功能更强大的飞行仿真。