

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR



论文题目：工业巡检用六旋翼飞行器的设计与开发

学生姓名：陈杰

学生学号：5110209369

专 业：机械工程及自动化（试点班）

指导教师：盛鑫军

学院(系)：机械与动力工程学院

工业巡检用六旋翼飞行器设计与开发

摘要

近年来,随着新型材料、微机电、微惯导技术以及飞行控制等技术的进步,微小型无人飞行器正快速发展并且受到越来越多的关注。尤其是旋翼式无人机,在近几年更是受到了极大的关注与研究。除了航空拍摄,多旋翼无人机在工业、农业、公共安全、科学研究等方面都有很大的应用前景。另一方面,自改革开放以来,我国的桥梁建设不断地发展,为了预防桥梁事故的发生,需要经常性对桥梁进行检测,而目前的检测手段主要由人工完成,比较繁琐且不安全,对新型的桥梁检测手段的需求比较迫切。

本课题主要基于以上两个背景展开研究,旨在设计与开发一款工业巡航用的六旋翼飞行器用于桥梁检测。相比于传统的人工的巡检,六旋翼无人机可以不受限于空间、地形等因素,从更多角度和位置来评估巡检结果,简化巡检过程,提高巡检效率,保障人员安全,降低高危作业时事故的发生率。

本课题针对桥梁检测的特点,基于多旋翼无人机的工作原理,设计开发了一款六旋翼机架,并进行了仿真分析;对机载的动力系统、控制系统、传感器系统进行了选型与测试,并集成到设计制造出来的六旋翼无人机样机上进行了测试;编写了无人机的避障算法、定轨迹巡航算法,设计了桥底下检测过程的定位方案。最终课题给出了一套较为完整的桥梁检测系统方案,并搭建一套用于工业巡检的硬件平台。

关键词: 六旋翼, 无人飞行器, 工业检测, 桥梁检测系统, 避障

DESIGN AND DEVELOPMENT OF MULTICOPTER FOR INDUSTRIAL INSPECTION

ABSTRACT

In recent years, with the development of new materials, MEMS, MIMU and flight control technology, micro UAVs is growing rapidly and drawing more attention. Particularly, multi-rotor UAV is subject to a great deal of attention and research. In addition to aerial photography, multi-rotor UAV also has great potential applications in industry, agriculture, public security, scientific research and so on. On the other hand, since the reform and opening up policy, China's bridge construction continue to develop. In order to prevent the occurrence of bridge accidents, frequent bridge inspections are required to find the defects of the bridges. And current detection methods are mainly done manually, which are cumbersome and unsafe. So new bridge detection methods are urgently demanded.

The main topic of the research is based on the above two background. This issue aims to design and develop a six-rotor on industrial inspection for bridge inspection. Compared to the traditional inspection methods, the six-rotor UAV won't be limited to space, topography and other factors, and assess inspection results from more angles and positions, which will simplify the inspection process, improve the inspection efficiency, guarantee personnel safety, and reduce the incidence of high-risk job accidents.

According to the characteristics of the bridge detection and the principle of multi-rotor UAV, the issue will design a six-rotor frame, and do some simulation analysis. Suitable on-board power system, control system, sensor system will be selected and tested, and integrated into the frame. What's more, obstacle avoidance algorithm and given trajectory flight algorithm will be designed for bridge inspection. And the issue will also come up with some special positioning scheme under the bridge for inspection process. Final issue will give a complete bridge inspection system and build a hardware platform for industrial inspection.

Key words: Six-rotor unmanned aerial vehicle, industrial inspection, bridge inspection, obstacles avoidance

目录

第一章 绪论.....	1
1.1 课题背景.....	1
1.2 国内外研究现状.....	1
1.2.1 高校研究现状.....	1
1.2.2 国内外科技公司研究现状.....	3
1.2.3 无人机桥梁检测研究现状.....	5
1.3 课题研究目的.....	6
第二章 六旋翼的总体方案与结构设计.....	7
2.1 总体设计方案.....	7
2.2 六旋翼的结构设计.....	8
2.2.1 六旋翼的三维建模.....	8
2.2.2 结构的仿真分析.....	13
第三章 动力系统、控制系统及传感器的选型与集成.....	17
3.1 动力系统选型.....	17
3.2 飞行控制系统的选型与测试.....	18
3.2.1 飞行控制器主控器的选型.....	18
3.2.2 辅助控制器的选型.....	19
3.2.3 飞行控制器的测试.....	20
3.3 传感器的选型与测试.....	23
3.3.1 超声传感器的选型与测试.....	23
3.3.2 激光测距传感器的选型与测试.....	26
3.4 其他配件的选型.....	31
3.4.1 相机及云台的选型.....	31
第四章 六旋翼飞行器动力学模型及算法设计.....	33
4.1 六旋翼无人机的工作原理.....	33
4.1.1 六旋翼运动原理.....	33
4.1.2 六旋翼的数学模型.....	34
4.1.3 六旋翼的动力学模型.....	35
4.2 避障系统设计.....	37
4.2.1 避障系统结构设计.....	37
4.2.2 避障系统算法设计.....	38
4.3 定位方式设计.....	39
4.3.1 基于加速度传感器的 GPS 盲区定位方案.....	39
4.3.2 激光传感器定位方案.....	40
4.3.3 微基站定位方案.....	42
4.4 定轨迹飞行算法设计.....	44
4.4.1 定航向角模式.....	45
4.4.2 航向转动模式.....	45
第五章 实验测试与系统优化.....	47



5.1	机架的制造与装配.....	47
5.2	六旋翼续航能力测试.....	50
5.3	六旋翼基本飞行测试.....	51
5.4	避障测试.....	52
5.5	定轨迹巡航测试.....	53
5.6	系统优化.....	54
5.6.1	机架结构的优化.....	54
5.6.2	避障系统的完善.....	55
第六章	总结.....	56
6.1	论文总结.....	56
6.2	展望.....	56
	谢辞.....	58
	参考文献.....	59
	附录.....	61

第一章 绪论

1.1 课题背景

近年来,随着新型材料、微机电(MEMS)、微惯导(MIMU)及飞行控制等技术的进步^[1],微小型无人飞行器(MUAVs)正快速发展并且受到越来越多的关注。通常飞行器被分为三类,固定翼、直升机和多旋翼飞行器。多旋翼无人机作为微小型无人飞行器的一种,其概念在二十世纪初就被提出^[2],但由于科技水平的限制,一直没有得到发展,直到上世纪80年代开始才逐渐开始成为研究的热点。多旋翼无人机的尺寸一般小于1m,质量在1kg以内,因其尺寸小、机械结构简单、能够垂直起落与悬停的特点,在军事、民用上都有很广泛的应用前景^[3]。在军事上,作为侦查和武器平台,多旋翼无人机可以通过携带不同的设备,执行侦查监视、电子干扰、通信中继、目标定位、攻击损伤有效评估等任务^[4]。在民用上,可用于航空摄影、工业检测、农业巡检、公共安全与管理、地理测绘、科学研究等领域。多旋翼无人机作为一个平台,通过搭载不同的设备,可以实现广泛的应用。相比于传统的固定工业机器人,微小型无人飞行器作为新型的移动机器人^[5],可在三维空间中工作以及广泛的应用范围,使之成为未来几十年里研究的热点。

另一方面,我国的公路、市政桥梁事业在近些年发展迅速,桥梁建设作为国家重要的基础设施建设之一,关系到整个国家经济与社会的发展。自改革开放以来,我国的桥梁建设不断地发展,至2014年底我国的公路桥梁和城市桥梁就已分别建成71.3万座和5.8万座^[6]。然而公路桥梁等基础设施在自然条件的风吹日晒以及车辆行驶的作用下,会出现一定问题的劣化,这种劣化问题如果没有及时检测出来,进一步恶化后就会出现严重的事故。尤其是桥梁,桥梁垮塌事故的频繁发生,严重威胁了人员的安全以及社会的稳定。所以必须要对桥梁等进行经常性的检测与维护。目前桥梁检测的方式主要有两种^[7]:一种就是检测人员通过搭建临时的脚手架或者乘坐小艇到桥下进行观察,这种方式效率低下,检测人员的安全也存在隐患,而且检测范围有限,无法到达高度比较高的地方;另一种就是通过大型桥梁检测工程车进行,但是这种方式设备成本比较高,而且对于跨江桥的检测也是无能为力。总体来说,传统的桥梁检测方式效率低且检测过程麻烦,检测人员的安全也存在隐患,而且存在许多检测的盲区检测人员无法进行检测。目前对新型的桥梁检测手段的需求还是比较迫切的。

本课题基于以上两个背景进行探索与研究,开发一款工业巡检用的六旋翼无人机用于桥梁的检测。

1.2 国内外研究现状

1.2.1 高校研究现状

目前在多旋翼的理论研究上,国外有很多高校在研究。

典型的有宾夕法尼亚大学GRASP机器人实验室的Vijay Kumar Lab,他们的研究课题涉及很多方面,比较受关注的就是机群的协调飞行^[8]、智能手机与飞行器的协作飞行、多传感器融合的自主飞行^{[9][10][11]}等。其中机群的协调飞行主要通过高速运动捕捉系统VICON获取

飞行器的位置对飞行器进行协调控制。多传感器融合的自主飞行主要通过惯性单元、GPS、激光测距传感器和双目摄像机共同实现飞行器的定位。



图 1-1 多传感器融合的自主飞行器

麻省理工大学在容错飞行上有比较突出的成果，采用了自适应等算法，保障了飞行器在驱动系统局部损坏的情况下依然能够维持与原系统相近的性能^[12]；MIT SENSEable City Lab 还研制的一款能在水上飞行的四旋翼飞行器 Waterfly。



图 1-2 MIT Waterfly

斯坦福大学在微型无人机上有较多的研究，在 1999 年就研制了一款厘米尺寸大小微型无人机 Mesicopter，在 2014 年还研制了世界上第一款可穿戴的四旋翼 Nixie。



图 1-3 斯坦福大学的 Mesicopter

明尼苏达大学对无人机与脑机接口有比较深入的研究,通过脑机接口控制四旋翼飞行器实现各种复杂的动作。

总结一下,目前对多旋翼的研究主要可以分为两类,一类是针对飞行器本身的研究,包括控制算法、容错飞行等,另一类是飞行机器人的研究,即通过在多旋翼上搭载各类传感器实现其他功能和应用。

1.2.2 国内外科技公司研究现状

在多旋翼的产品方面,目前国内外很多科技公司都在致力于多旋翼无人机的研究。

国内比较成熟的就是大疆创新公司,从事于全球领先的无人飞行器控制系统及无人机解决方案的研发,其产品主要针对专业级航拍应用,销量占领全球的一半以上。其最新发布的Phantom3集成了视觉定位系统和超声波传感器,实现了室内飞行的稳定悬停。



图 1-4 DJI Phantom3

国外,除了航拍应用,很多公司都在积极探索多旋翼飞行器在其他领域的应用。

德国的 Ascending Technologies 公司的 AscTec Falcon 8 这款八旋翼飞行器的高精度与可靠性,使其在工业巡检领域表现很出色,其结构为一个“V”型的八旋翼,机体的前端固定一个摄像机,可拍摄上方、前方、下方获得图像。该公司的另外三款飞行器 AscTec Hummingbird、AscTec Pelican 和 AscTec Firefly 更是成为了很多高校与研究机构研究多旋翼的平台。



图 1-5 AscTec Falcon 8

同样是来自德国的 Aibotix 公司在工业巡检、测绘、救灾及其他一些领域都有应用。其主要的产品是一款通过保护架包围的并搭载摄像头的六旋翼 Aibot X6,这款产品 在电力线检测、风机检测、桥梁检测等工业领域的巡检上用的比较多。



图 1-6 Aibot X6

另外一家来自德国的 Microdrones 公司，其公司的产品主要有 MD4-200、MD4-1000 和 MD4-3000，分别为轴径从小到大的三款飞行器。其中 MD4-1000 这款飞行器空载最长飞行时间可达 88min，是一般飞行器续航时间的 2-3 倍。该公司的产品在工业巡检、公共安全、搜索救援等领域都有比较多的应用。



图 1-7 MD4-1000

同样是来自德国的 AiDrones、AirRobot 等公司在工业巡检、公共安全、测绘等领域也有许多优秀的产品及应用解决方案。

加拿大的 Aeryon Labs 公司、Draganfly Innovations 公司等 在多旋翼无人机方面也有比较出色的产品。Aeryon Labs 公司的产品在军事，公共安全，商业上都有很多的应用。Draganfly Innovations 公司一款以工业检测为导向的三轴六旋翼 Draganflyer X6 因其独特的结构在工业检测领域有比较多的应用。



图 1-8 Draganflyer X6

法国的 Novadem 公司的产品在工业巡检、室内巡检上也是做的比较出色。旗下的产品 U130 在桥梁检测、风机检测、电力线检测上都有应用，并且能够实现室内环境的巡检。



图 1-9 U130

1.2.3 无人机桥梁检测研究现状

在工业检测的桥梁检测方面，法国道路及桥梁中心实验室的于 2007 年利用微小型直升机系统进行了桥梁安全检测方面的尝试并取得了早期的成功^[16]。该系统采用单目摄像头进行局部定位，并在飞行器的前端用可俯仰摄像头来进行桥梁检测。法国铁路公司于 2013 年采用四旋翼飞行器进行了桥梁检测的尝试^[13]。德国的 AIBOTIX 公司利用六旋翼飞行器进行了一系列的桥梁、高压线缆的检测并取得了很大的成功。其基本方案为在具备保护框架的六旋翼飞行器的基础上搭载具有仰俯功能的摄像头，并在地面操作人员的控制下完成一定的检测工作。另外，2011 年 MIT 实现使用四旋翼无人机进行涵洞检修方面的应用^[14]，该应用基于 AscTec Pelican 四旋翼系统。在涵洞中，飞行器依靠两组激光雷达以及图像识别技术组合的方式实现局部精确定位。两组雷达分别对上下两个方位同时扫描，以实现精确定位，对桥梁检测很有参考价值。国内方面，近年来也有少数机构开展了桥梁检测方面的活动与研究。如沪渝高速路段就针对清江大桥实现了相较于人工检测更加高速可靠的检测方法。哈尔滨工程大学的莫宏伟教授也推出了可贴近桥底飞行的四旋翼无人机系统，但其广泛应用可行性还有待进一步验证。从目前的研究与应用现况来看，在桥梁检测方面的应用一般在较远距离下

对桥梁的缺陷等进行检测，还没有对超近距离的检测进行尝试。

无论在国内还是国外，采用多旋翼无人机进行桥梁安全检测还处于起步阶段。

1.3 课题研究目的

本课题的研究目的主要是设计与开发一款工业巡航用的六旋翼飞行器用于桥梁检测。相比于传统的人工的巡检，六旋翼无人机可以不受限于空间、地形等因素，能够到达检测员没法到达的地方，从更多角度和位置来评估巡检结果，大大简便了巡检过程，提高了巡检效率，保障了巡检人员的人身安全，降低高危作业时事故的发生率。

本课题的研究目标主要有以下几点：

- a) 设计并制造出一台适合工业巡检用的六旋翼飞行器原型机；
- b) 基于开源飞控实现飞行器的基本飞行控制；
- c) 实现飞行器的自主避障功能与定轨迹飞行功能；
- d) 设计应用于桥梁环境下的定位方式。

第二章 六旋翼的总体方案与结构设计

2.1 总体设计方案

六旋翼由六个旋翼组成，每个旋翼上都有一个电机带动桨叶为飞行器提供动力，一般采用无刷电机。电机连接电调，电调的作用就是将飞控板的控制信号转变为电流的大小，以控制电机的转速，电机的工作电流很大，通常每个电机正常工作时，平均有 3A 左右的电流，飞控板无法提供这样大的电流，所以需要电调来提供比较大的电流。飞行器上的传感器系统主要由惯性测量单元（IMU，包括加速度计和陀螺仪）、地磁计、气压计、GPS 组成，通过这些传感器获得飞行器的姿态与位置信息，其中 IMU 主要获得飞行器的加速度与角速度信息，气压计获得高度信息，GPS 获得位置信息。

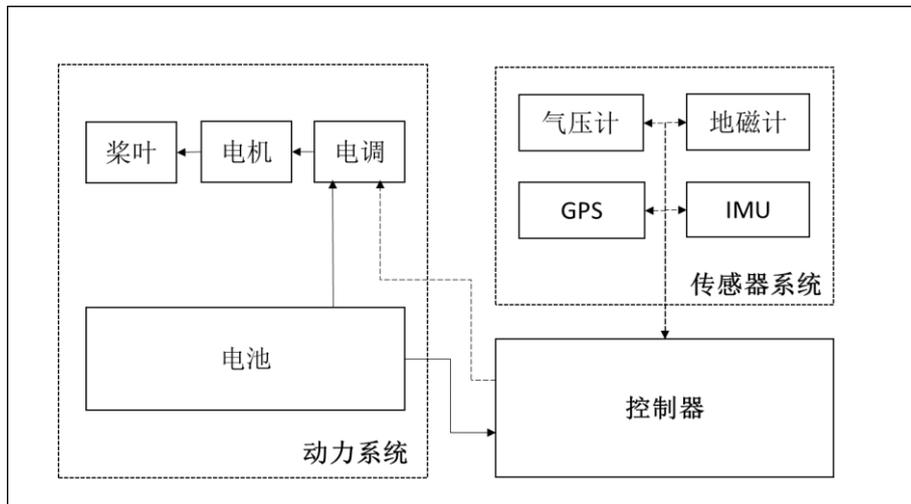


图 2-1 六旋翼飞行器系统结构图

针对桥梁检测的用途，主要针对普通高架桥与跨江桥之前的引桥部分进行底面、梁柱的检测，采集桥面与梁柱的图像并进行分析处理，找到桥面与梁柱的裂纹等缺陷，以便及时进行修复。所以飞行器一方面能够在桥下稳定悬停，另一方面能够获得清晰稳定的桥面特征的图像。同时，为了能够简化巡检过程，六旋翼无人机需要能够更具设定的轨迹自动巡检飞行，并能够在巡检过程中自动避开桥下的桥墩等障碍。

针对以上的这些功能需求，首先对六旋翼无人机系统进行整体的设计。对于获得清晰稳定的桥梁缺陷的图像，需要有一套适合于桥梁检测用的仰视的消抖云台，能够实现相机想前向上的取景与拍摄，并获得稳定清晰的图像；对于桥下的稳定悬停，因为桥下的 GPS 信号不好，无法实现飞行器的精确定位，所以需要设计一套或多套桥下的定位方案，实现无人机在桥下的稳定悬停；对于自动巡检飞行的功能，则需要设计一套适合于不同定位方式的定轨迹飞行的算法，实现六旋翼无人机的自动飞行功能；对于实现飞行过程中避开桥墩或者其他的障碍物，则需要选择合适的传感器，设计一套避障系统，实现无人机的自主避障功能。

2.2 六旋翼的结构设计

2.2.1 六旋翼的三维建模

六旋翼机架尺寸为直径 800mm，算上避碰结构整体直径为 1200mm，高度为 420mm（包括脚架，不包括机架上的传感器）。机身主体的直径为 220mm，机翼部分长度为 300mm。机架整体采用可折叠设计，便于使用的收放与携带。

六旋翼的顶上为固定了相机的仰视消抖云台，可以朝上拍摄桥梁地面的缺陷或者超前拍摄桥梁侧面的缺陷，云台的下方是一个 270°扫描范围的激光测距传感器，用于检测周围环境中的障碍物实现飞行器的自主避障。再往下是主飞行控制器和辅助飞行控制器。周围一圈是保护桨叶的避碰结构，上面安装有超声波传感器，辅助激光传感器的障碍物检测与飞行器的避障。

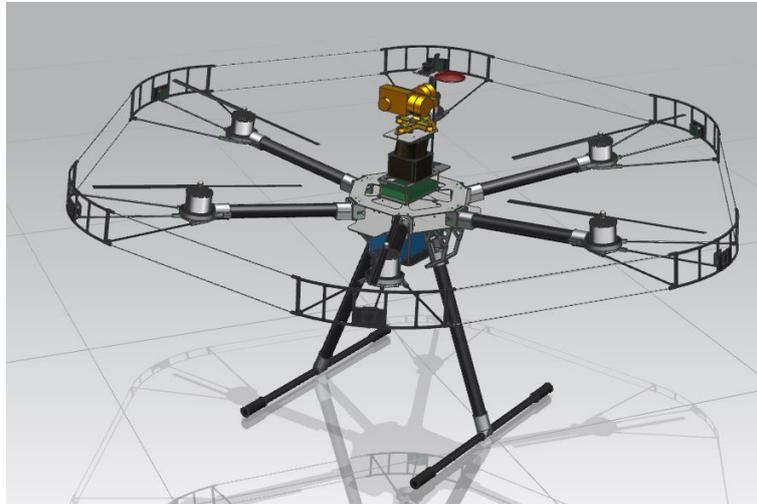


图 2-2 六旋翼整体效果图

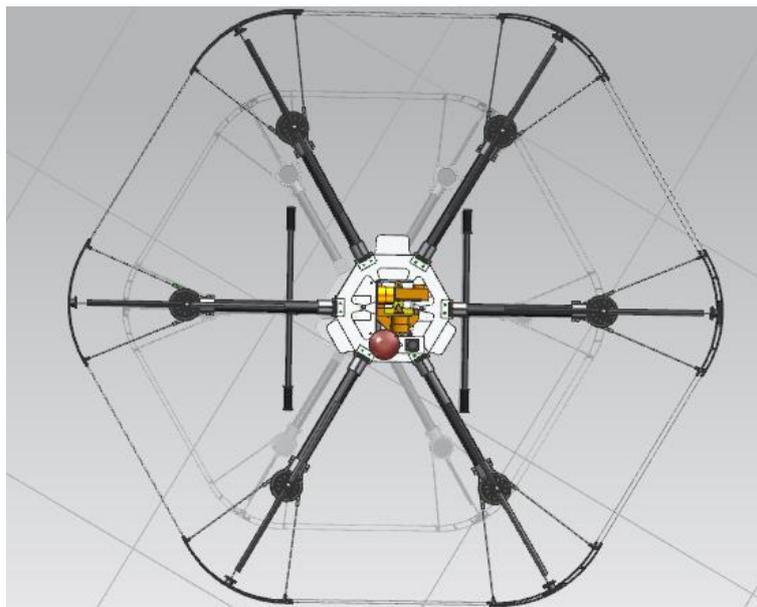


图 2-3 六旋翼俯视图

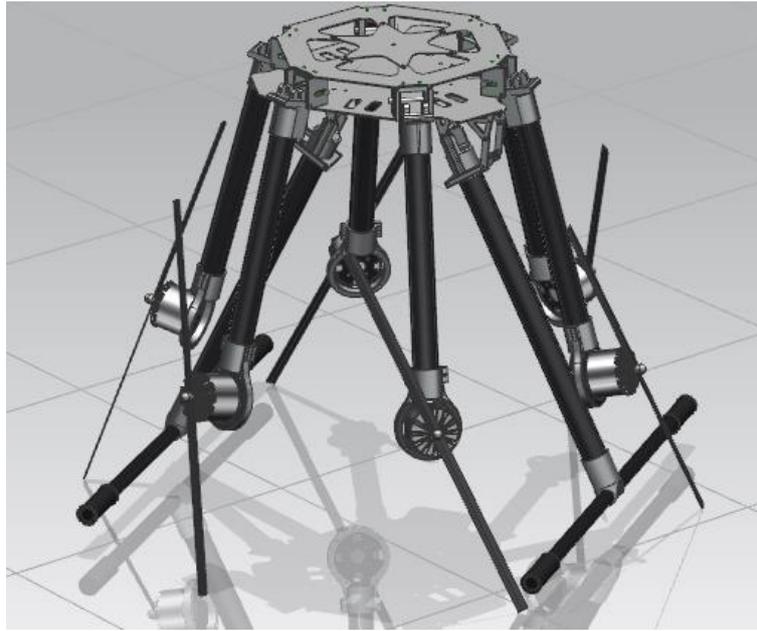


图 2-4 六旋翼机架收放图

2.2.1.1 机翼建模

相比于尺寸较小的四旋翼来说，六旋翼机架的收放与携带上很不方便了，所以在六旋翼机翼的结构设计上，设计了一个可折叠的结构，该结构上有一个可以滑动并手动拧紧的的销的结构，通过与机翼基部的槽的配合实现机翼的打开与收放，在一定程度上大大提高了六旋翼机架的便携性。

机翼的主体部分采用重量小但强度高的碳纤维管，连接部分采用铝合金材料加工的结构。

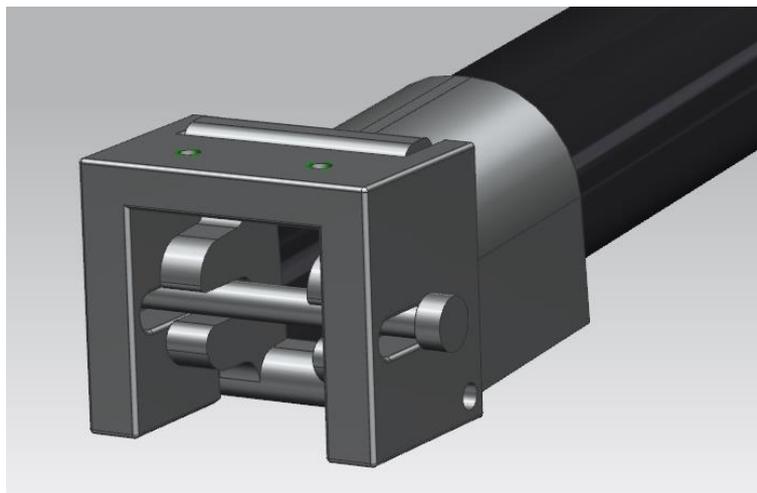


图 2-5 机翼连接结构

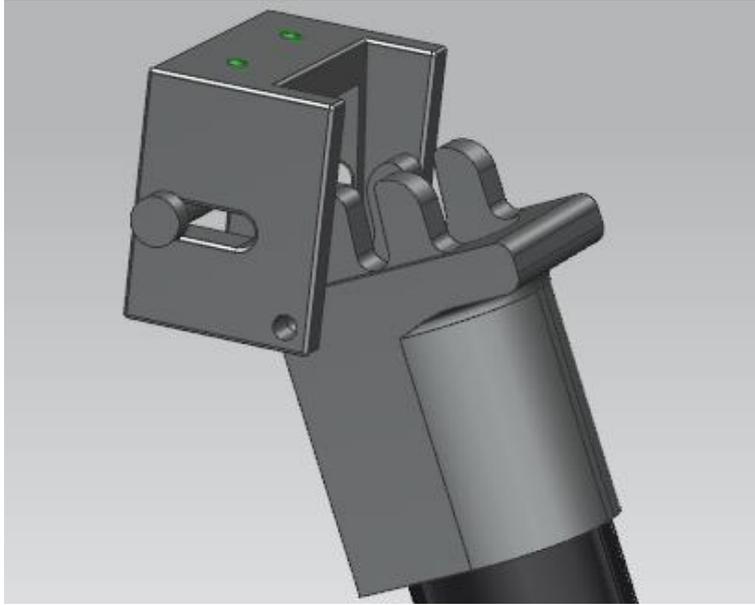


图 2-6 机翼连接结构折叠示意图

电机与机翼通过一块碳纤维板连接，碳纤维板与基座之间通过减震球连接，以此来减小电机转动产生的振动对机身的影响，从而提高六旋翼在飞行过程中的稳定性。

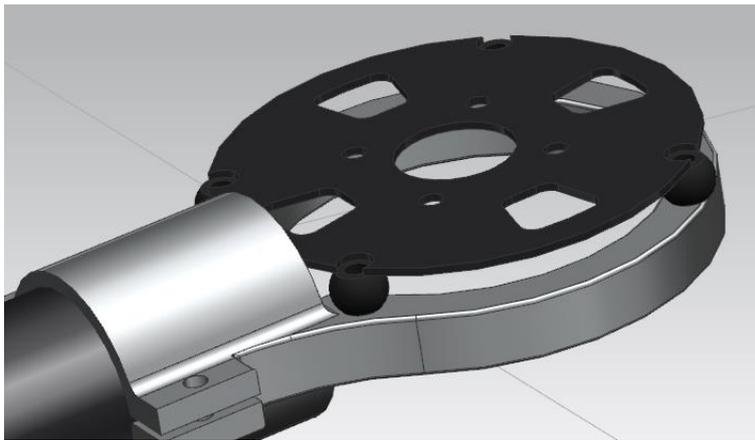


图 2-7 机翼的电机座与基座的示意图

2.2.1.2 机身建模

机身部分采用双层碳纤维板的设计。碳纤维板在抗压和抗拉上强度比较好，但是在抗弯曲上刚度和强度很差，很容易发生弯曲和断裂。所以通过双层碳纤维板的设计，可以将对碳纤维板弯曲的力转换到抗拉压上，保证机体良好的强度和刚度。

机身顶部设计一个可拆卸的顶板，方便组装时元器件的安装与放置，不必把整个机身拆掉进行安装操作。

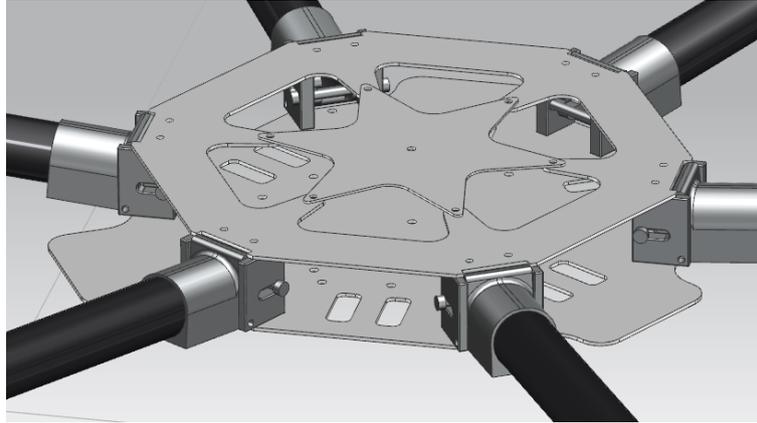


图 2-8 机身示意图

2.2.1.3 起落架建模

起落架采用了平面四杆机构设计，为一个曲柄摇杆机构。

其中杆 1 为摇杆，长度为 48mm，代表起落架的主支撑杆，杆 2 为连杆，长度为 37.5mm，杆 3 为曲柄，长度为 30mm，杆 4 为机架，长度为 50mm。通过舵机驱动杆 3 实现起落架的收起与放下。使舵机驱动杆 3 转动，转动角度范围在 0~150°，摇杆 1 的转动范围为 0~75°。

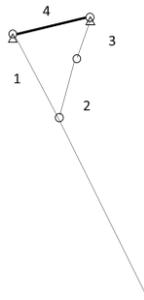


图 2-9 起落架机构简图

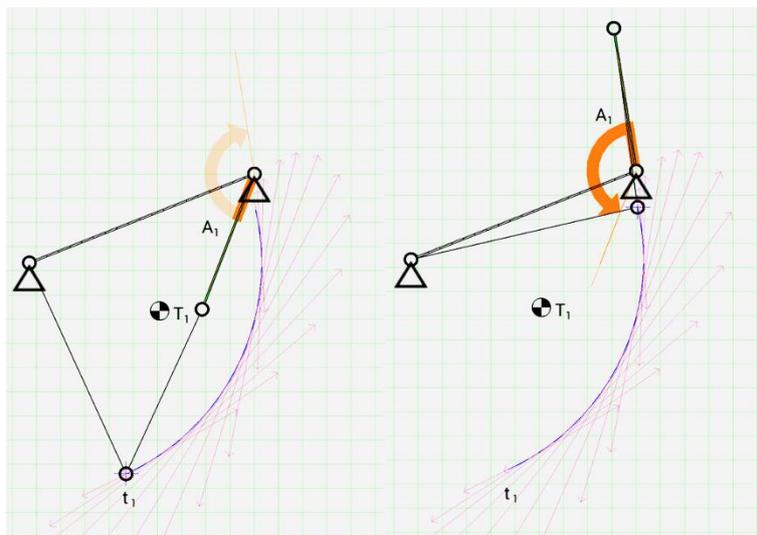


图 2-10 起落架运动示意简图

起落架主体材料采用碳纤维管，连接部分材料采用铝合金加工。



图 2-11 起落架示意图

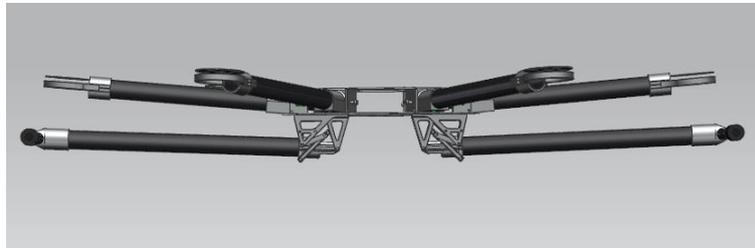


图 2-12 起落架收起示意图

2.2.1.4 避碰结构建模

避碰结构的主要作用是一方面保护桨叶，防止六旋翼桨叶碰到障碍物后直接导致飞行器失衡坠落，另一方面也可以保护操作人员与周围人的安全，防止桨叶伤人。

避碰结构主要采用栅栏式设计，可以有效保护桨叶与减轻重量。上面设计有超声波传感器的固定孔位，用于固定超声波传感器进行障碍物检测。



图 2-13 避碰结构设计

避碰结构拟采用注塑加工，采用 ABS 材料。避碰结构之间采用韧性较好的玻璃纤维棒进行连接。

2.2.2 结构的仿真分析

六旋翼机架及零部件的仿真分析主要在 solidworks 中进行, 通过仿真分析比较了最终版的机架与之前设计的机架, 分析部分的零部件, 并做了相关的改进。

2.2.2.1 机架质量属性仿真分析

首先在软件中定义材料属性, 机体部分采用碳纤维板, 机翼部分采用碳纤维管, 碳纤维的密度为 1780kg/m^3 , 连接件采用铝合金, 铝合金的密度为 2700kg/m^3 ; 避碰结构采用 ABS 塑料, ABS 的密度为 1020kg/m^3 ; 起落架主体部分采用碳纤维管, 连接部分采用铝合金。

以机架底板中心作为坐标原点, 机架的质量属性分析结果如表所示:

表 2-1 机架质量属性

测量项目	参数
质量(kg)	0.854
重心(mm)	X = 0.000 Y = 23.359 Z = -0.367
惯性主轴	$I_x = (-0.000, 0.000, 1.000)$ $I_y = (1.000, 0.000, 0.000)$ $I_z = (0.000, 1.000, -0.000)$
惯性张量($\text{kg} \cdot \text{mm}^2$)	$I_{xx} = 24771.021$ $I_{xy} = -0.010$ $I_{xz} = -0.003$ $I_{yx} = -0.010$ $I_{yy} = 48208.666$ $I_{yz} = -0.232$ $I_{zx} = -0.003$ $I_{zy} = -0.232$ $I_{zz} = 24728.202$

另外, 对其他的部件包括起落架、避碰结构等进行了分析, 质量如下表所示:

表 2-2 六旋翼各部件质量

组件	质量
机架(kg)	0.854
起落架(kg)	0.484
避碰结构(kg)	0.029

2.2.2.2 机架应力应变仿真分析

在 Solidworks 中导入机架模型, 定义材料, 材料包括铝合金和碳纤维, 输入材料的参数, 参数如下表所示。碳纤维的参数和碳纤维丝的排布方向与制造的工艺都有关系, 这里取了一个其参数范围内的一般值进行分析。

表 2-3 材料参数

材料	密度(kg/m^3)	弹性模量(GPa)	泊松比
铝合金	2700	69	0.33

然后定义连接方式，机体与机翼连接件处使用螺栓固定，其他连接处选择全局接触连接。

然后定义夹具约束。因为六旋翼整体的重量主要集中在中间机体的位置，力主要在六个机翼端部，所以在机体中间顶板位置定义一个固定约束。

在六个机翼的端部定义力，力的大小根据电机最大升力确定，每个机翼施加一个 30N 的力，方向为垂直电机座平面向上。

最后划分网格，网格密度选择良好。完成网格划分后，进行计算。

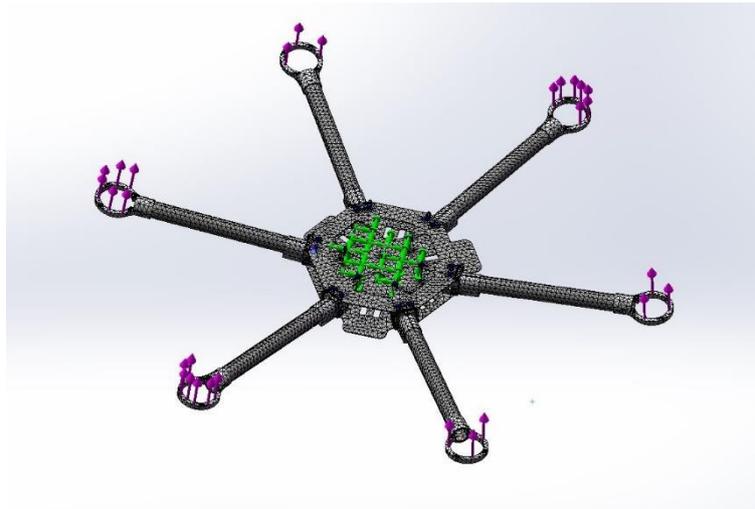


图 2-14 网格化后的机架

计算的结果主要有应力、应变和位移三部分。

分析应力、应变的结果，可以看到应力应变比较大的部分主要在两个地方：一个是机翼碳管根部与铝合金连接件的位置，另一个是机体上板与顶板的连接处，但是这两处的应力应变都在安全范围内。机架整体的应力分布比较均匀，没有出现明显的应力集中的地方。



图 2-15 机架应力分布情况

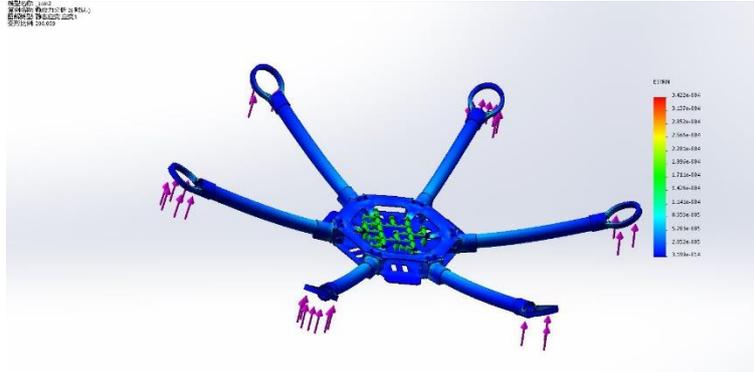


图 2-16 机架应变分布情况

分析位移的结果，可以看到机翼端部电机座中心的位移为 0.25mm，机架的刚度大概在 120000N/m 左右。

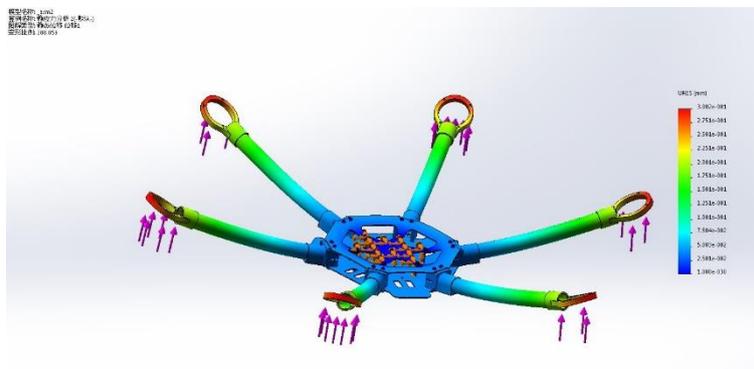


图 2-17 机架位移分布情况

2.2.2.3 避碰结构的分析与改进

避碰结构主要用于防撞和保护桨叶，所以对强度和刚度的要求也是比较大。对避碰结构进行了分析，并通过分析结果做了相应的改进，达到了较好的效果。

对第一版避碰结构进行了分析，固定基部的四个螺栓孔，在端部的顶部施加 1N 的作用力。分析结果发现圆环和支撑杆的连接处出现了应力比较集中区域，最大应力达到了 $1.0 \times 10^{10} Pa$ 。据此可以推断根部比较容易断裂。对第一版的结构进行了分析与调整，将连接处分布在基部的两端，尽量分散这一区域应力的分布。在相同的约束和外力的作用下，对第二版结构进行了分析。由应力图可以看到，相比于第一版，第二版结构的应力分布分散了很多，而且没有明显的应力集中区域，最大的应力在 $4.2 \times 10^9 Pa$ ，相比较于第一版减小了很多。

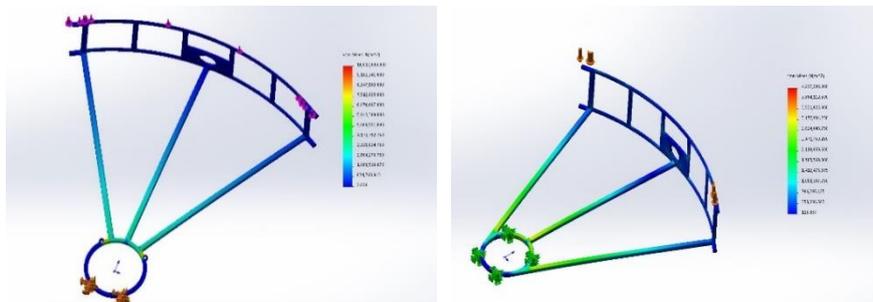


图 2-18 第一版与第二版避碰结构应力对比图

在考虑到整体结构的加工使用 ABS 注塑，结构的刚度可能会比较差。分析第二版的位

移结果,发现在 1N 的作用力下,端部的最大位移大概在 10mm 左右,整体的刚度在 100N/m,显然刚度比较差。对第二版结构进行了改进,将其分为三部分,基座、连接杆与防护栏。基座采用铝合金,连接杆采用碳纤维管,防护栏采用 ABS 注塑加工。在相同约束下,端部的最大位移大概在 0.5mm 左右,整体的刚度在 2000N/m,相比于原来的结构提高了 20 倍左右。

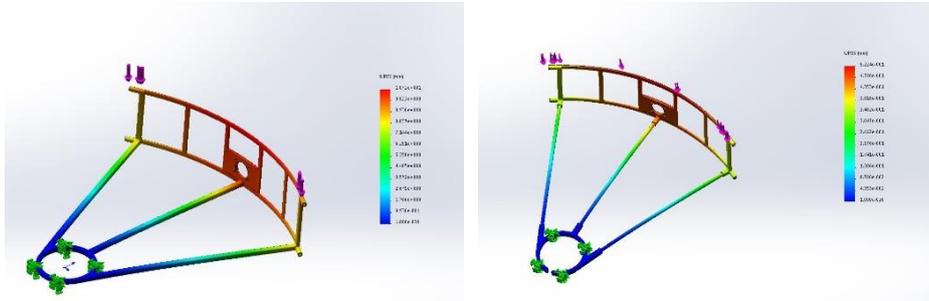


图 2-19 第二版与第三版避碰结构位移对比图

第三章 动力系统、控制系统及传感器的选型与集成

3.1 动力系统选型

六旋翼的动力套件主要包括电机、桨叶、电调和电池。电机方面，在多旋翼上目前主流使用无刷电机作为动力来源，电机 KV 值表示外加 1v 电压对应的每分钟空转转速。不同 KV 值的电机配备不同的桨叶，螺旋桨越大，升力越大，但对应需要更大的扭矩来驱动，这时需要采用 KV 值小的电机。反之，大 KV 值的电机配备小尺寸的螺旋桨。电调全称电子调速器，作用是将飞控板的控制信号，转变为电流的大小，以控制电机的转速。电机的电流是很大的，通常每个电机正常工作时，平均有 3A 左右的电流，飞控板无法承受这样大的电流，所以需要电调来提供比较大的电流。

根据软件的质量仿真计算，机架重量在 1.5kg 左右，脚架 1kg 左右，电池 1.3kg 左右，电机、电调和主控系统加起来重量在 1kg 左右，为不同型号的相机和云台、其他传感器或者器件等预留 2kg 的重量，整机的起飞重量在 6.8kg 左右，所以 6 个电机加起来的升力至少在 7kg 左右。

通过查找相关电机的参数，最终采用 JFRC-U411-KV420 款无刷电机作为动力，这款电机轴直径 5mm，槽极结构为 12N14P，电机尺寸为直径 48mm，高 31mm，重量为 140g。该电机在 22.2V 电压、配备 15inch 桨的条件下，最大提供 2855g 升力，最大电流为 22.1A，最大转速为 6980rpm，功率 500w，效率为 5.72g/W。



图 3-1 电机示意图

表 3-1 U4110 电机参数表

项目	参数
轴承	NMB
轴直径(mm)	5
槽极结构	12N14P
电机尺寸(mm)	Φ48 × 31

项目	参数
电机重量(g)	140

表 3-2 U4110 电机在满油门下的参数

项目	参数
型号	U4110
电压(V)	22.2
桨叶(inch)	15
电流(A)	22.6
拉力(G)	2855
转速(RPM)	6980
功率(W)	500
效率(G/W)	5.72

因为电机最大电流为 22.1A，所以选择 30A 以上的电调即可满足要求。选择好盈 Platinum-30A-Pro 电调，该电调重量为 31.6g，持续电流 30A，最大瞬间电流 45A（不超过 10s）。

电源采用 6S 锂电池，容量 10000mAh，放电倍率为 25C，重量 1390g。

3.2 飞行控制系统的选型与测试

3.2.1 飞行控制器主控器的选型

本课题基于开源的 Pixhawk 飞行控制器进行开发与测试，Pixhawk 使用的 MCU 是 32 位 STM32F427 芯片，168MHz/ 252MIPS 的 Cortex-M4F 核心，带有浮点运算器，256KB 的内存，2MB 的闪存，相比于目前使用较多的另一个开源飞控 APM 采用的是 8 位 16MHz/16MIPS 的 Atmega1280/2560 芯片，Pixhawk 的处理能力远远高于 APM，所以课题选择 Pixhawk 飞控来作为开发的平台。

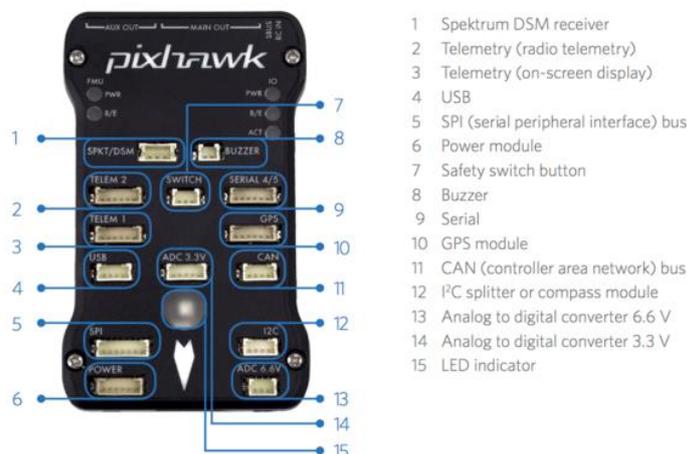


图 3-2 Pixhawk 飞行控制器正面说明

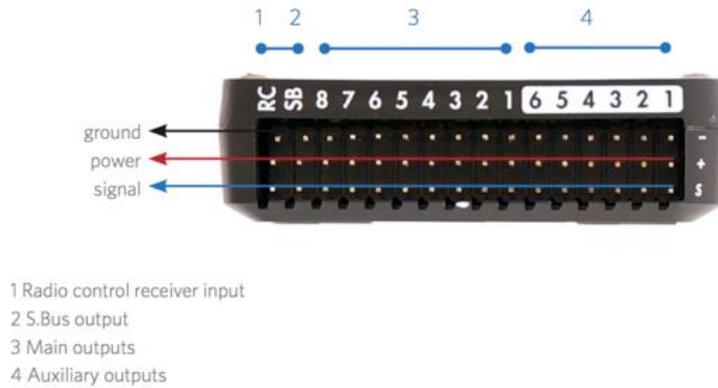


图 3-3 Pixhawk 飞行控制器侧面说明

Pixhawk 上有 5 个 UART 串口，2 个 CAN 接口，一个 DSM / DSM2 / DSM-X 接口，一个 SPI 接口，一个 I2C 接口，两个 3.3V 的 ADC 和一个 6.6V 的 ADV 接口，支持 Futaba 遥控器的 S.BUS 接口，支持 PWM 和 PPM 的输入。有 14 个通道的 PWM 输出，其中 8 个主通道，6 个辅助通道。主通道主要输出给 PWM 信号给电调驱动电机，辅助通道输出可以输出 PWM 信号给其他装置使用，如云台、起落架等。

Pixhawk 板载的传感器有 L3GD20H 16 bit 陀螺芯片获得角速度信息，LSM303D 14 bit 加速度/磁场芯片获得加速度和磁场信息，和 MEAS MS5611 气压芯片来获得飞行器高度信息，外接 GPS 芯片可在室外获得飞行器位置信息用于定位。

Pixhawk 上运行的是 Nuttx，Nuttx 是一个实时嵌入式操作系统（RTOS），通过 Makefile 文件组织编译，可以编写相关应用程序，然后编译后就可以在系统中运行。该系统下的应用程序通过“uORB 话题”进行通讯，应用程序可以从话题中订阅相关消息，经过处理之后，也可以把消息发布到话题中被其他应用程序订阅，从而很方便地实现了各个应用程序之间的通讯。

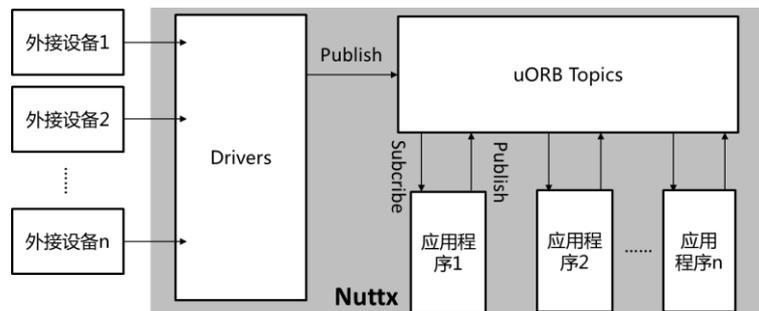


图 3-4 Pixhawk 系统框架

3.2.2 辅助控制器的选型

考虑到要安装激光测距传感器、超声波传感器等多个传感器并处理它们的数据，Pixhawk 一方面处理能力有限，而且无法提供足够的接口，所以另外选择了一块辅助控制器，主要用于处理其他传感器的数据。这里选择了 Raspberry Pi 2（树莓派）作为辅助控制器。

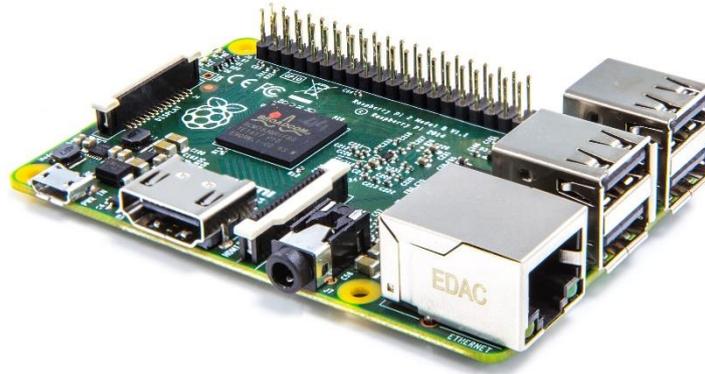


图 3-5 Raspberry Pi 2

树莓派拥有 900MHz 的基于四核的 ARM Cortex-A7 的处理器，1G 的内存，4 个 USB 接口，一个 Ethernet 接口，一个 HDMI 接口，还有 40 个通用 I/O 口，运行 LINUX 系统，相当于一台微型的计算机。

Name	Header	Name
3.3v	1 2	5v
SDA	3 4	5v
SCL	5 6	0v
GPIO7	7 8	TxD
0v	9 10	RxD
GPIO0	11 12	GPIO1
GPIO2	13 14	0v
GPIO3	15 16	GPIO4
3.3v	17 18	GPIO5
MOSI	19 20	0v
MISO	21 22	GPIO6
SCLK	23 24	CE0
0v	25 26	CE1

图 3-6 GPIO 引脚说明

40 个 GPIO 中，有几个特殊功能的引脚，分别可作为 I2C、SPI 和 UART 的接口，其余的可以作为通用输入输出。对于树莓派和 Pixhawk 的通讯，选择 UART 串口进行。

3.2.3 飞行控制器的测试

3.2.3.1 Pixhawk 测试

Pixhawk 飞行控制器的控制代码主要基于开源的 PX4 固件，PX4 的固件中包含了基本飞行控制的姿态计算、位置计算、姿态控制器、位置控制器等。本文关于 Pixhawk 的开发，主要在 Ubuntu 系统下搭建开发环境进行。

程序中主要有三种飞行控制模式：手动模式、辅助控制模式以及自动模式，其中辅助控制模式包括高度控制模式和位置控制模式。手动模式和辅助控制模式都需要遥控器控制，自动模式下，通过给飞行器设定的航点，飞行器按照航点自动飞行。手动模式下，遥控器通道直接控制飞行器的姿态与油门；高度控制模式下，油门通道通过积分，油门通道对应与控制飞行器的速度，来控制飞行器的高度，高度控制下可以使飞行器保持一定高度悬停；位置控

制模式需要有外部辅助定位系统，PX4 主要使用的是 GPS，遥控器通道对应于飞行器每个方向上的速度，来控制飞行器的位置实现飞行器的移动与定点悬停。

在手动模式下，遥控器直接控制飞行器翻滚、俯仰和偏航角，通过与加速度计、罗盘测得的姿态求差值，经过一个姿态控制器，算出角速率的值，然后与陀螺仪测得的角速率求差值，经过一个速率控制器，算出一个角加速度值，输出给电机控制飞行器的姿态。这里的姿态控制器和角速率控制器用的是 PID 控制器，姿态控制器用的是 P 控制器，角速率控制器用的是 PD 控制器。

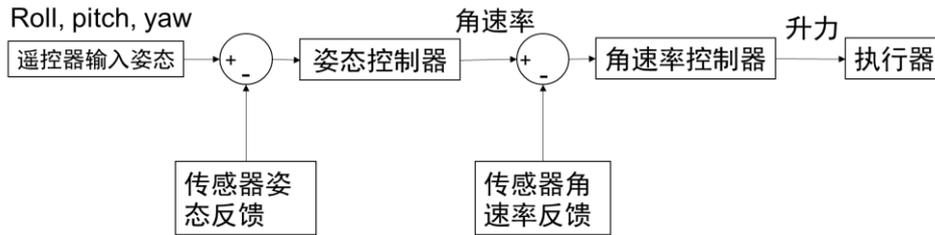


图 3-7 手动模式下的控制框图

在位置模式下，遥控器通道控制飞行器的位置，输入位置与传感器（GPS、气压计）得到的水平方向和高度方向的位置求差值，经过一个位置控制器，得到一个速度值，将这个速度与得到的飞行器的速度求差值，经过速度控制器得到一个姿态值，然后将姿态值输入到姿态控制器和角速率控制器中进行计算，最终输出给电机控制飞行器的姿态，实现位置的控制。这里位置控制器和速度控制器用到都是 PID 控制器，其中位置控制器用的是 P 控制器，速度控制器用的是 PID 控制器。

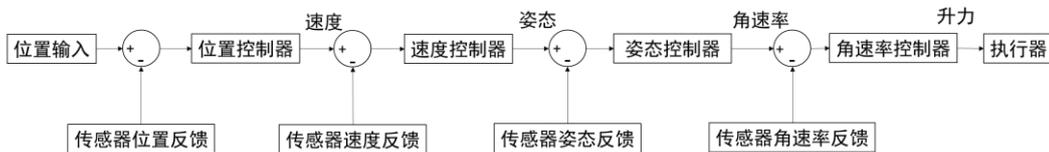


图 3-8 位置模式下的控制框图

在使用遥控器对飞行器进行控制时，无论在手动控制模式还是辅助控制模式下，遥控器俯仰、横滚、偏航三个通道的值会归一化到 $(-1, 1)$ 之间，遥控器油门通道的值会归一化到 $(0, 1)$ 之间，这里分别表示为 `manual.x`、`manual.y` 和 `manual.r`。手动模式下，归一化后的值乘以设定的最大的姿态值，就是飞行器的姿态输入。在位置控制下，归一化后的值乘以设定的最大的运动速度值，速度值经过积分，即可得到位置控制的输入。

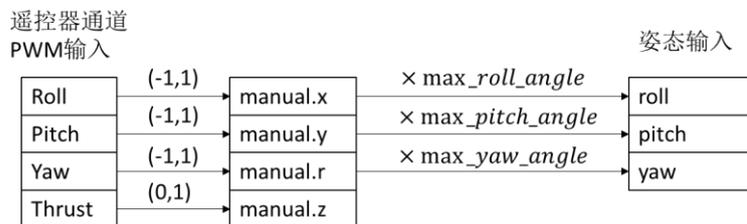


图 3-9 手动模式遥控器的输入

将 Pixhawk 飞行控制器搭载到第三章中设计制造的机架上，除了要进行初始的传感器校准，包括加速度计、陀螺仪、罗盘和遥控器，最重要的是调节姿态控制和位置控制的 PID 参数。参数调节的过程分为姿态参数调节和位置参数调节。首先在室内调节姿态参数，涉及姿态的参数主要有俯仰和横滚方向的角度参数 `Pitch_P`、`Roll_P` 和 `Yaw_P`，俯仰和横滚方向的角速度参数 `Pitch_Rate_P`、`Pitch_Rate_D`、`Roll_Rate_P`、`Roll_Rate_D`、`Yaw_Rate_P` 和

Yaw_Rate_D。将六旋翼飞行器绑在铝合金架子上，将 6 个参数都清零。首先逐渐增大 Pitch_Rate_P、Roll_Rate_P，直到飞行器刚开始出现振动，保持 Pitch_Rate_P、Roll_Rate_P 的值，逐渐增大 Pitch_Rate_D、Roll_Rate_D 的值，直至振动消失，则得到一个比较合适的角速度的参数值。然后调节 Pitch_P、Roll_P 的值，直到飞行器在俯仰和横滚方向有一个比较快速但没有出现振荡。然后调节偏航方向的参数，同样先调角速率参数，然后调节角度参数直至飞行器偏航方向的运动有一个比较合适的相应速度。这样，即可得到一组比较合适的姿态参数。位置参数需要在位置模式下进行，必须要室外有 GPS 的环境下调节，调节过程类似姿态调节。得到合适的姿态参数和位置参数后，即可进行飞行测试。



图 3-10 调节六旋翼的姿态参数

除了飞行功能的测试，为了能够方便后期的开发测试，编写了一个飞行器参数串口输出程序，通过 USB 和 Pixhawk 进行串口通讯，将需要查看的数据通过串口发送出来。在 Ubuntu 下通过 Kermit 与 Pixhawk 进行串口通讯，读取相关数据。PX4 固件中的应用程序通过“uORB 话题”进行通讯，所有的应用程序都会把处理后的数据以“消息”的形式发布到“话题”中，任何应用程序之间都可以通过“话题”订阅相关消息，实现应用程序之间的通讯。所以，只要在串口数据程序中，从“话题”中订阅想要了解的参数，然后通过计算机显示出具体的值，即可实现参数的输出。

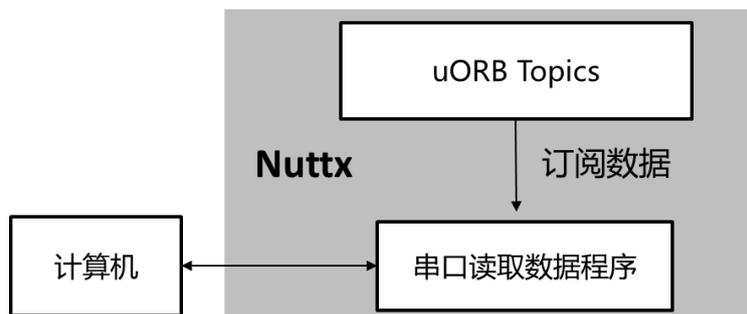


图 3-11 串口读取数据程序框图

3.2.3.2 Pixhawk 与 RaspberryPi 通讯测试

Pixhawk 与 RaspberryPi 通过 UART 串口进行通讯，通讯波特率设置为 115200，无奇偶校验位，无停止位。树莓派发送数据，数据格式以“M”开头、以“\n”结尾分别作为数据的起始和终止的标记，Pixhawk 使用一个 Ringbuffer（环形缓冲区）接收数据，防止数据的丢包，并将数据发布到“话题”中。

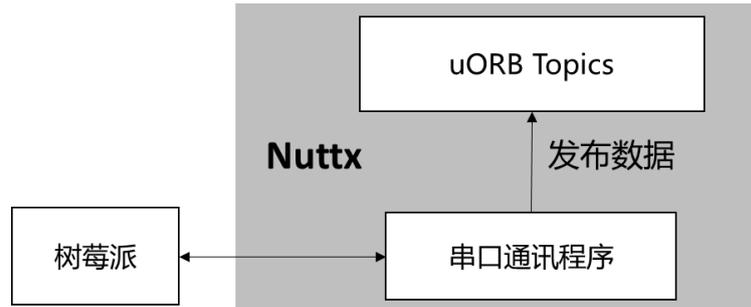


图 3-12 Pixhawk 与 RaspberryPi 通讯框图

3.3 传感器的选型与测试

要实现六旋翼的自主避障，除了机载的 IMU、地磁计、气压计、GPS 等传感器外，还需要其他能检测障碍物信息的传感器才能实现。这里，选择了超声传感器和激光测距传感器。

3.3.1 超声传感器的选型与测试

对 MaxBotics 公司 XL-MaxSonar-EZ 系列超声传感器进行了调研，该系列传感器体积小，重量轻，检测范围为 0 至 7.6 米，分辨率为 1cm，10HZ 的读取速率，可以通过三种方式输出数据，分别是脉宽输出、模拟电压输出以及串口数据输出。



图 3-13 XL-MaxSonar-EZ 系列超声传感器

表 3-3 XL-MaxSonar-EZ 系列超声传感器引脚说明

引脚	功能
PIN1	设置为高电平时，引脚 5 作为串口输出；设置为低电平时，引脚 5 输出脉冲，适用于低噪声链。
PIN2	脉宽输出。对于 MB1200 (EZ)系列，引脚 2 输出代表距离的脉冲宽度。计算距离时，每 cm 的脉冲宽度为 58us。
PIN3	模拟电压输出。对于测量范围在 7.6 米的声纳传感器，这个引脚以 (Vcc/1024) per cm 输出模拟电压。由于硬件的限制，最大的输出范围 5V 下为 700cm，3.3V 下为 600cm。

引脚	功能
PIN4	串口 RX。这个引脚在内部被拉高。如果不连接或者保持高电平，传感器会不断地测量并输出数据。如果置于低电平，则停止。20us 以上的高电平才能使其进行测量。
PIN5	串口 TX。当引脚 1 不连接或者置于高电平，以异步串口 RS232 的形式输出。
PIN6	VCC， 工作电压范围为 3.3V - 5V
PIN7	GND

对 XL-MaxSonar-EZ 系列的超声波传感器进行了对比，在测量范围在 7.6m 范围内的超声传感器型号有 MB1200、MB1210、MB1220、MB1230、MB1240 这几款，通过查看该系列超声传感器的数据手册，这几款声纳的主要区别在于其灵敏度和波束的带宽不同，灵敏度从 MB1200 到 MB1240 依次降低，波束宽度依次减小。

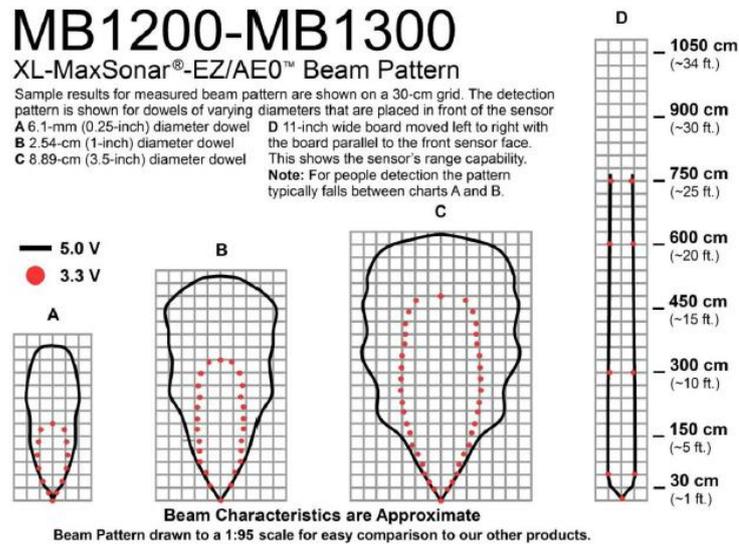


图 3-14 MB1200 波束测试图

MB1240/MB1340: XL-MaxSonar-EZ/AE4

The XL-MaxSonar-EZ/AE4 is the least sensitive and narrowest beam pattern in the XL-MaxSonar-EZ/AE sensor series. This makes the sensor great for applications in which there is a large amount of acoustic. The MB1240/MB1340 sensor is also recommended in applications where only large target detection is required.

MB1240-MB1340 XL-MaxSonar®-EZ/AE4™ Beam Pattern

Sample results for measured beam pattern are shown on a 30-cm grid. The detection pattern is shown for dowels of varying diameters that are placed in front of the sensor. A 6.1-mm (0.25-inch) diameter dowel B 2.54-cm (1-inch) diameter dowel C 8.89-cm (3.5-inch) diameter dowel D 11-inch wide board moved left to right with the board parallel to the front sensor face. This shows the sensor's range capability. Note: For people detection the pattern typically falls between charts A and B.

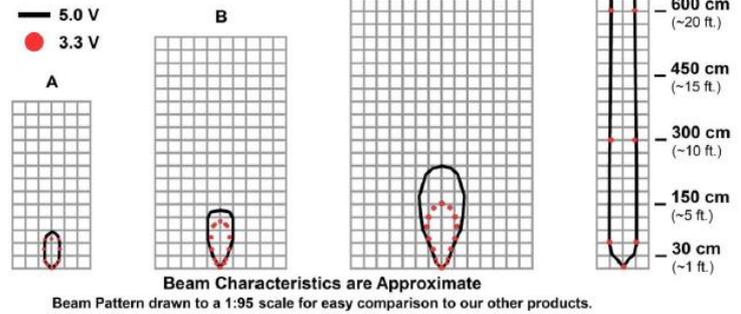


图 3-15 MB1240 波束测试图

MB1240 的灵敏度较低，波束比较窄，故不易受到其他干扰，适用于检测地面和桥面。MB1200 灵敏度较高且波束较宽，适用于检测六旋翼周围的障碍物。所以选择 MB1200 的声纳用于前后左右的避障，MB1240 的声纳用于地面与桥面距离的检测

因为主控器 Pixhawk 上没有足够的接口同时再接多个超声传感器，所以将超声波传感器接入到树莓派中进行处理，通过读取超声传感器输出的脉冲宽度计算距离信息，然后计算后的距离通过串口发给 Pixhawk 主控芯片，由主控器控制飞行器的姿态与位置。

分别在模拟电压输出和脉宽输出下，对超声波传感器进行了测试。在两种方式测试下，超声传感器均能输出较为准确的距离数据。

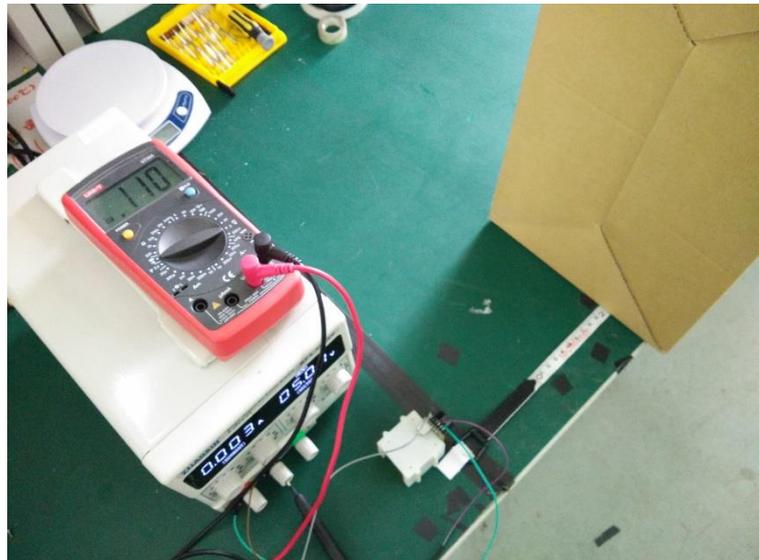


图 3-16 模拟电压输出下的测试

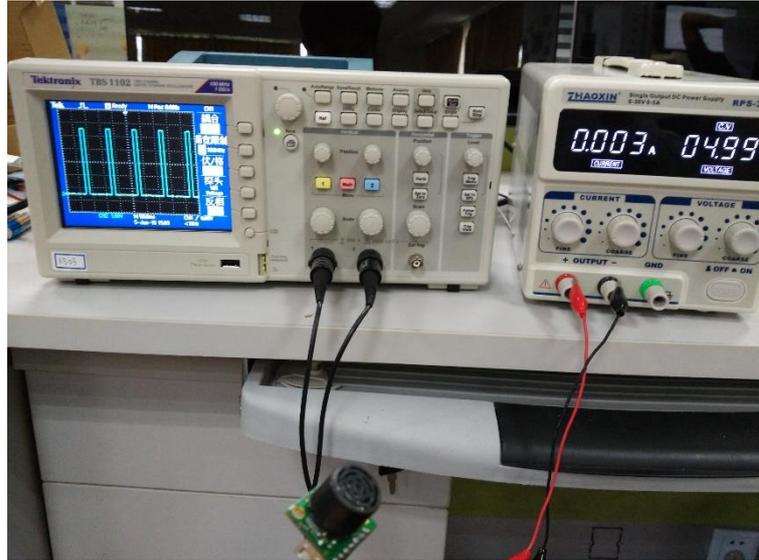


图 3-17 脉宽输出下的测试

3.3.2 激光测距传感器的选型与测试

分别对 Hokuyo 公司的几款激光测距传感器进行了调研，包括 URG-04LX、UBG-04LX-F01、UTM-30LX、UST-10LX/20LX 等，考虑到激光传感器要适应于室外环境的使用以及使用的距离，选择 UTM-30LX 型号激光测距仪作为障碍检测传感器。

UTM-30LX 激光测距传感器具有 25ms/scan 的扫描时间，30m 的测距范围，270° 的扫描范围，适用于室外环境，测量范围大，响应快，适用于高速运动的机器人。其尺寸为 60mm*60mm*87mm，重量 370g，在 0.1 到 10m 的精度为 $\pm 30\text{mm}$ ，10 到 30m 的精度为 $\pm 50\text{mm}$ 。其采用的通讯方式为 USB2.0。

表 3-5 UTM-30LX 激光测距传感器参数

类别	参数
型号	UTM-30LX
电源	12VDC $\pm 10\%$ (电流：最大 1A，正常情况：0.7A)
光源	半导体激光二极管($\lambda=905\text{nm}$) 激光安全等级 1(FDA)
检测范围	0.1-30m，最远可达 60m 扫描范围 270°
精度	0.1-10m: $\pm 30\text{mm}$, 10-30m: $\pm 50\text{mm}$
角度精度	0.25°(360°/1,440 steps)
扫描时间	25msec/scan
噪音	< 25dB
通讯接口	USB2.0
指令系统	专用指令 SCIP Ver.2.0
重量	370g

UTM-30LX 激光测距传感器在扫描时，从顶部看以逆时针在旋转。如图 3-18 所示，传

传感器的检测范围为 Step0 到 StepD, 为 270°。Step0 为第一个测量点, StepA 为设定的初始测量点, StepB 为传感器正前方测量点, StepC 为设定的终点的测量点, StepD 为最后一个测量点。UTM-30LX 激光测距传感器的检测范围 270°, 精度为 0.25°, 所以激光最多能够输出 1081 个测量点。

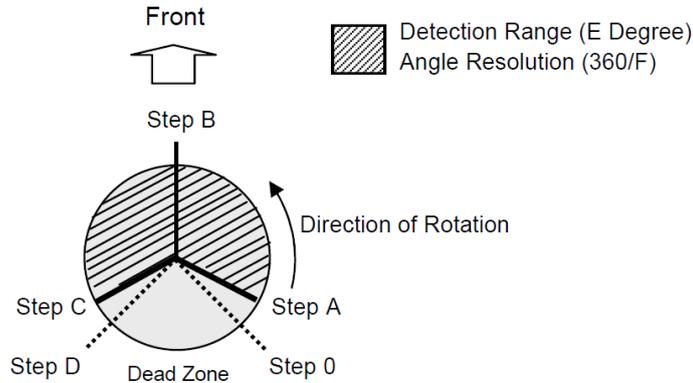


图 3-18 激光传感器的扫描范围说明

UTM-30LX 激光测距传感器采用 SCIP2.0 指令系统, 通过串口发送相应的指令, 就可以接收到相关的数据。如表 3-5 所示为主要的一些指令。

表 3-6 SCIP2.0 主要指令表

指令	功能
VV	传感器传输版本细节, 如序列号、固件版本等
PP	传感器规格信息
II	传感器运行状态
MD/MS	激光扫描指定次数并发回数据, 其中 MD 为三字符编码, MS 为二字符编码
GD/GS	激光扫描一次并发回数据, 其中 GD 为三字符编码, GS 为二字符编码
BM	使能激光
QT	关闭激光
RS	重置激光
TM	同步激光与主机

激光传感器发回给主机的数据是经过编码的, 距离数据主要有两种编码方式: 二字符编码和三字符编码。二字符编码应用于最大长度为 12 位的数据。编码通过将数据分为上下 6 位, 然后与 30H 相加, 转化为 ASCII 字符。三字符编码应用于最大长度为 18 位的数据。编码通过将数据分为上中下 6 位, 然后与 30H 相加, 转化为 ASCII 字符。所以, 在读取激光的数据后, 需要对其进行解码。以三字符编码为例, 图 3-19 和图 3-20 分别展示了编码与解码的过程。

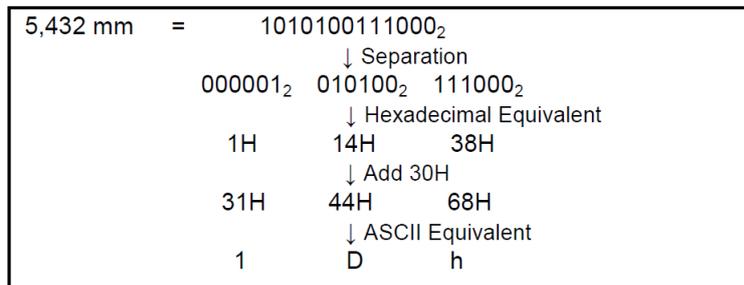


图 3-19 三字节编码的编码过程

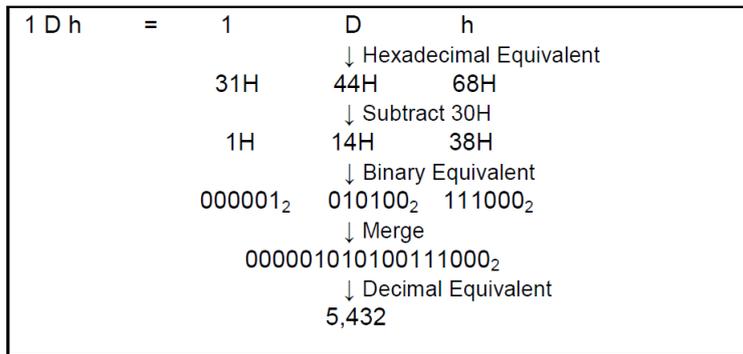


图 3-20 三字节编码的解码过程

使用 GD/GS 或者 MD/MS 指令读取激光传感器的数据时，通过指定起始测量点，终止测量点以及步长，激光传感器就会通过一定的格式发回数据。当数据长度小于 64 位时，数据部分通过一个数据块发送回来，数据块内容为经过编码的 ASCII 码，后三位是一个校验位和两个回车。当数据长度大于 64 位时，数据没 64 位组成一个数据块，每个数据块后两位为一个校验位和一个回车，最后一行的最后三位为一个校验位和两个回车。

M	D or S	Starting Step	End Step	Cluster Count	Scan Interval
Remaining Scans		LF	String Characters	LF	
9	9	b	LF	Time Stamp (4byte)	Sum LF
Data	Sum	LF	LF		

图 3-21 数据小于 64 位时发回的数据结构

M	D or S	Starting Step	End Step	Cluster Count	Scan Interval
Remaining Scans		LF	String Characters	LF	
9	9	b	LF	Time Stamp (4byte)	Sum LF
Data Block 1 (64 byte)			Sum	LF	
-----			Sum	LF	
Data Block N-1 (64 byte)			Sum	LF	
Data Block N (n byte)			Sum	LF	LF

图 3-22 数据大于 64 位时发回的数据结构

利用树莓派通过 USB 串口读取激光传感器的数据，获得每个角度激光扫描到的距离信息。为了更直观的看到激光扫描到的周围信息，分别在一个直径 320mm 的圆柱体内和一个 300mm*330mm 长方体内对激光的扫描精度进行了测试，并将数据导入到 matlab 里做了处理。

测试一：直径 320mm 的圆柱体内

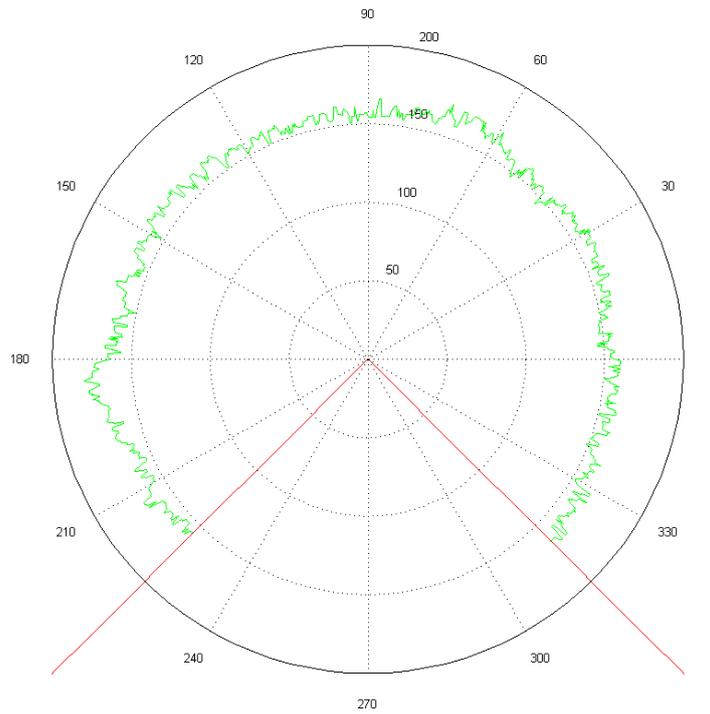


图 3-23 圆柱体内 2D 数据图像

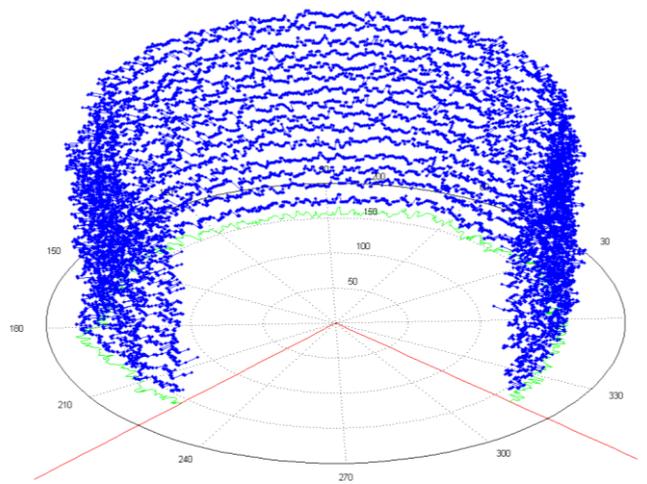


图 3-24 圆柱体内 3D 数据图像

通过MATLAB 计算,测得的直径的平均值为318mm,最大值为333mm,最小为302mm,精度在 $\pm 15\text{mm}$ 以内。

测试二: 300mm*330mm 长方体内

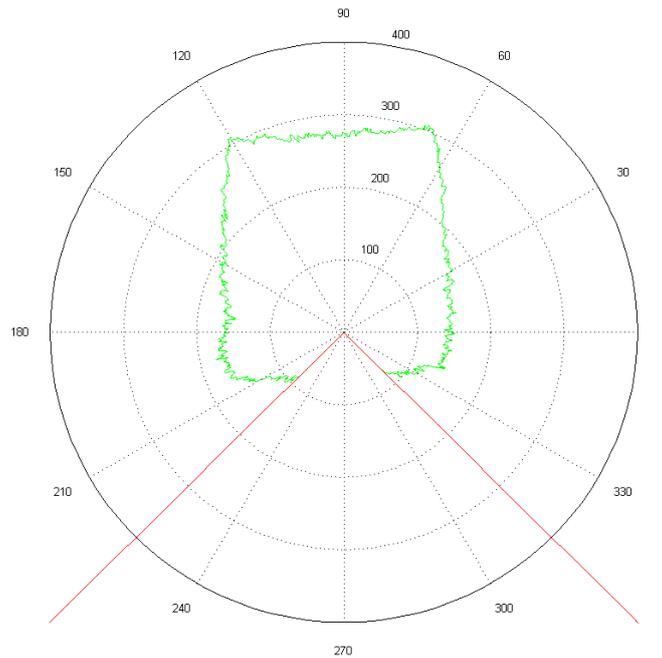


图 3-25 长方体内 2D 数据图像

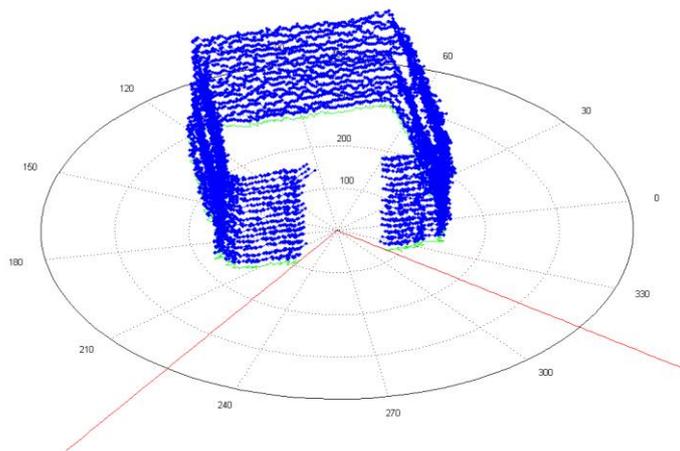


图 3-26 长方体内 3D 数据图像

通过 MATLAB 计算，测得的矩形宽度的平均值为 303mm，最大值为 316mm，最小为 282mm，精度在 $\pm 21\text{mm}$ 以内；矩形的平均长度为 330mm，最大为 344mm，最小为 310mm，精度在 $\pm 20\text{mm}$ 以内。

激光测距传感器的标称精度在 10 米以内为 $\pm 30\text{mm}$ ，故两次的测试结果在精度范围内。

将激光传感器接入到树莓派的 USB 接口中，利用树莓派读取激光的数据，树莓派将激光数据进行处理，得到扫描一圈得到的距离的最小值，以及最小距离对应的角度位置，然后

通过“M*****\n”的格式发送给 Pixhawk，其中距离值的位数为 5 位，不足 5 位前面用 0 补齐，角度值的位数为 3 位。例如扫描到的最小距离为 1200mm，角度为 120°，则发送给 Pixhawk 的数据为“M01200120\n”。

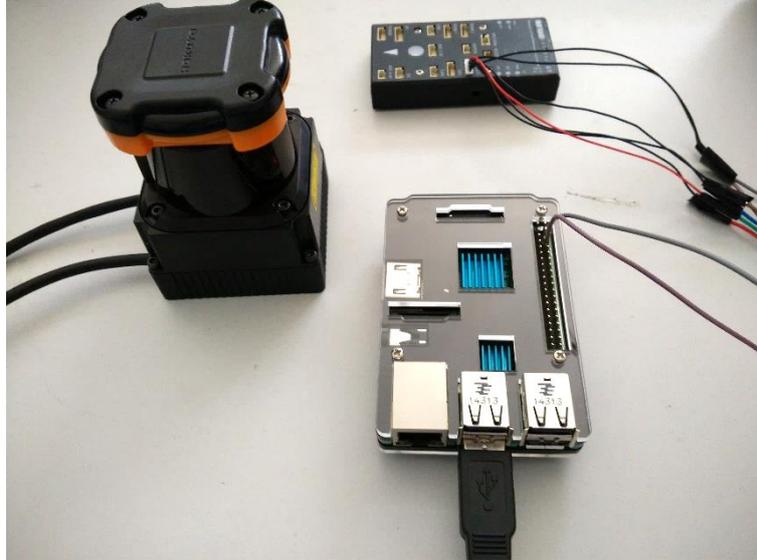


图 3-27 激光传感器、树莓派与 Pixhawk 连线图

3.4 其他配件的选型

3.4.1 相机及云台的选型

在使用六旋翼进行桥梁检测时，需要用相机拍摄桥梁底面和侧面，分析桥梁的缺陷情况。相机需要能够在运动情况下，既能通过拍照清晰地呈现桥面的缺陷裂纹情况，又能够录制清晰稳定的视频。所以相机和云台的选型，对桥梁检测的效果起到了至关重要的作用。

相机方面，选择 GoPro Hero 4 相机，该相机是一款小型的放水防震运动相机，能够录制 4K 高清视频，可以高达每秒 30 帧的速度拍摄 1200 万像素照片，基本可以满足拍摄桥梁的需求。



图 3-28 GoPro 相机

云台主要用于安装固定相机，因为检测桥梁的过程中只需对桥梁的底面和桥墩侧面进行检测，所以选择较低自由度的云台即可实现。本课题选择了一个 2 自由度的云台，该云台拥有俯仰和横滚两个方向的自由度，俯仰运动范围 240°，横滚运动范围 110°。将云台安装

于六旋翼的顶部朝上放置，拍摄桥梁底面时，保持相机始终朝上，拍摄桥墩侧壁时，保持相机始终朝前，以稳定地获得较为清晰的图像。



图 3-29 云台

第四章 六旋翼飞行器动力学模型及算法设计

4.1 六旋翼无人机的工作原理

4.1.1 六旋翼运动原理

六旋翼无人机是一个六个自由度的系统，包括 X 、 Y 、 Z 方向上的位置运动以及俯仰、翻滚、偏航三个方向上的姿态运动。通过调节六个旋翼上的电机的转速实现飞行器的姿态与位置的控制。

由电机的布置方式相对于机头方向的不同，可以把六旋翼分为“X型”和“+型”，以图 2-2 中的“X型”六旋翼为例，六旋翼是通过以下几种方式实现飞行器的运动：

(1) 俯仰运动：飞行器绕电机 1、2 所在的轴转动。电机 1、2 转速不变，电机 3、5 转速上升，电机 4、6 转速下降，且每个电机转速的改变值相同，使飞行器整体的扭矩保持不变，即可实现飞行器机头的上升。

(2) 横滚运动：飞行器绕垂直于电机 1、2 所在的轴转动。电机 1、4、5 转速上升，电机 2、3、6 转速下降，且电机 1 和 2 的改变值相同，电机 3、4、5、6 的改变值相同，使飞行器整体的扭矩保持不变，即可实现飞行器向左横滚。

(3) 偏航运动：飞行器绕竖直方向轴旋转。六旋翼飞行器的偏航运动主要是靠旋翼产生的反扭矩实现。旋翼转动过程中由于空气阻力作用会对飞行器形成与转动方向相反的反扭矩，为了克服反扭矩影响，可使六个旋翼中的三个正转，三个反转，且对角上的旋翼转动方向相反。反扭矩的大小与旋翼转速有关，当六个电机转速相同时，四个旋翼产生的反扭矩相互平衡，四旋翼飞行器不发生转动；当六个电机转速不完全相同时，反扭矩不平衡，就会引起飞行器转动。电机 1、3、6 转速增大，电机 2、4、5 转速减小，且电机转速的改变值相同，则可以实现飞行器的顺时针偏航运动。

(4) 垂直升降：飞行器沿竖直方向运动。同时增加六个电机的转速，且转速改变值相同，则飞行器垂直上升；反之减小电机转速，飞行器垂直下降。

(5) 水平移动：飞行器沿着水平方向运动。飞行器的水平运动通过改变飞行器的姿态实现。如果飞行器需要实现前后运动，只要使飞行器的姿态俯仰一定角度即可实现。同理通过改变飞行器姿态的翻滚实现飞行器的左右运动。

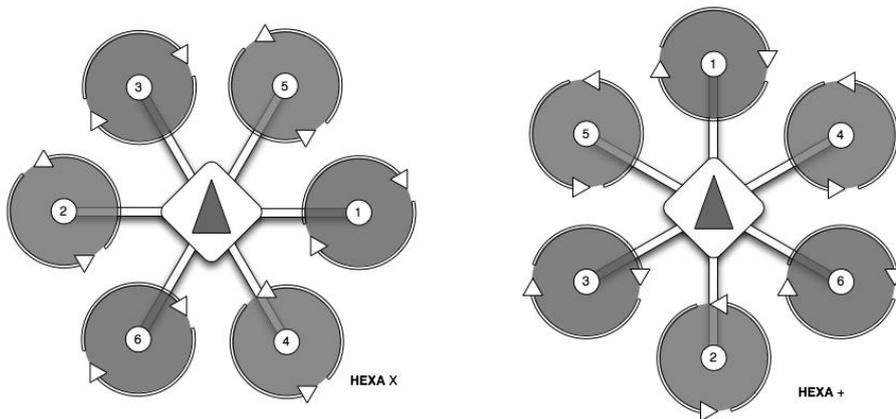


图 4-1 六旋翼的飞行模式

4.1.2 六旋翼的数学模型

定义一个惯性坐标系和一个机体固定坐标系，其中机体固定坐标系中的 X_b 的正方向为六旋翼的机头方向，垂直与机头方向定义为 Y_b ，根据右手定则定义 Z_b 。将六旋翼起飞时的原点作为惯性坐标系的原点，以地面的东北方向通过右手定则建立一个惯性坐标系（如图 2-3）。定义六旋翼的相对于惯性坐标系的坐标为 x 、 y 、 z ，定义位置向量为 \mathbf{r} ，

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4-1)$$

六旋翼的姿态信息通过定义绕机体坐标系旋转的三个欧拉角给出，分别为绕 X_b 轴转动的横滚角 φ ，绕 Y_b 轴转动的俯仰角 θ ，和绕 Z_b 轴转动的偏航角 ψ （如图 2-4）。定义姿态向量为 \mathbf{q} ，

$$\mathbf{q} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} \quad (4-2)$$

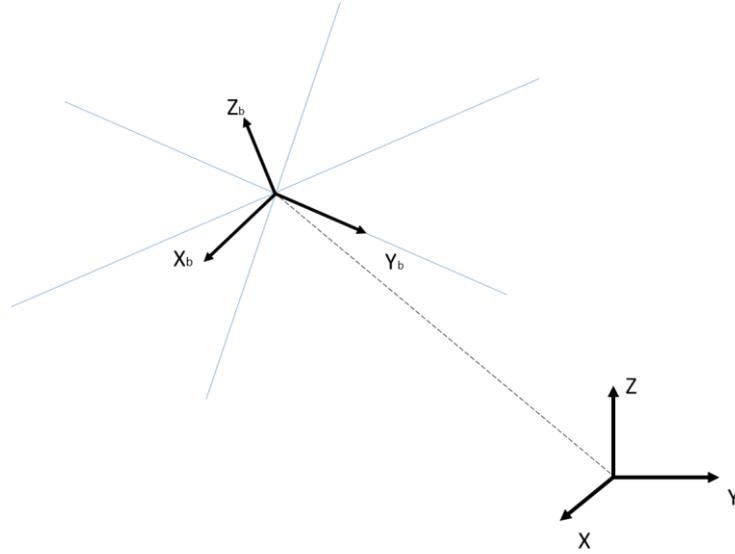


图 4-2 定义惯性坐标系与刚体固定坐标系

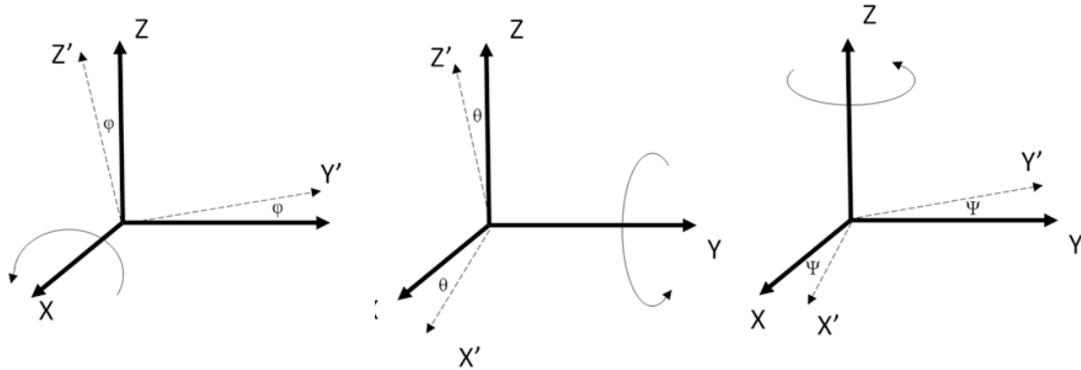


图 4-3 定义飞行器的横滚角、俯仰角及偏航角

通过定义姿态角后，即可推导出机体坐标系相对于惯性坐标系的旋转矩阵为

$$R = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\sin\theta\cos\psi - \cos\psi\sin\psi & \cos\psi\sin\theta\cos\psi + \sin\psi\sin\psi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\psi + \cos\psi\cos\psi & \cos\psi\sin\theta\sin\psi - \sin\psi\cos\psi \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix} \quad (4-3)$$

4.1.3 六旋翼的动力学模型

参考文献资料^[15]，因为地球自转的影响比较小，所以假设在地球坐标下为惯性坐标系，设惯性坐标系为 $E = \{x, y, z\}$ ，机体坐标系为 $B = \{x_b, y_b, z_b\}$ 。在机体坐标系下，六旋翼六个电机产生的六个升力且指向 z_b 正方向，六个电机的升力分别为 F_1 、 F_2 、 F_3 、 F_4 、 F_5 、 F_6 ，同时产生对应的六个反扭矩，分别为 M_1 、 M_2 、 M_3 、 M_4 、 M_5 、 M_6 。由前面六旋翼的工作原理的分析，要实现六旋翼的姿态与位置控制，可以定义以下 4 个控制输入

$$U_1 = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 \quad (4-4) \quad U_2 = \frac{\sqrt{3}}{2}(F_2 + F_3)$$

$$U_3 = \left(F_1 + \frac{1}{2}F_2 + \frac{1}{2}F_6\right)L - \left(F_4 + \frac{1}{2}F_3 + \frac{1}{2}F_5\right)L \quad (4-6)$$

$$U_4 = M_1 + 3 + M_5 - M_2 - M_4 - M_6 \quad (4-7)$$

其中 L 为电机轴心到六旋翼中心的距离。

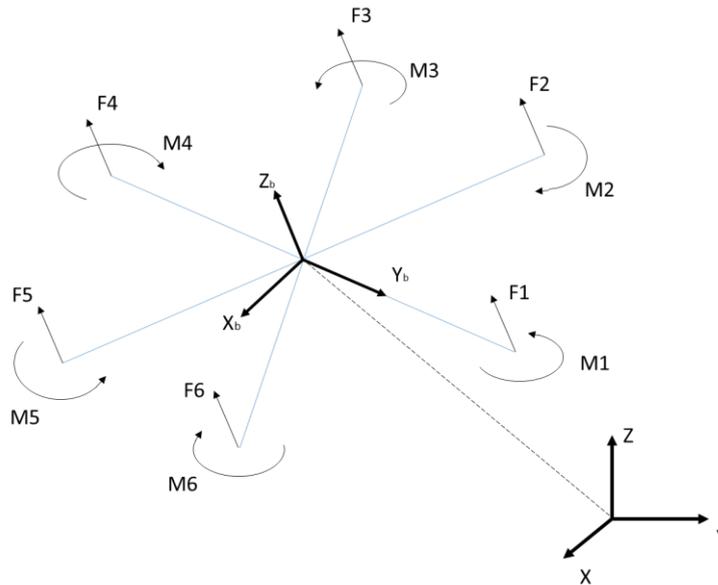


图 4-4 六旋翼的升力、扭矩示意图

六旋翼的刚体动力学方程可用如下形式表达

$$\begin{cases} m\dot{r} = R \begin{bmatrix} 0 \\ 0 \\ U1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \\ I\ddot{q} = \begin{bmatrix} U3 \\ U2 \\ U4 \end{bmatrix} \end{cases} \quad (4-8)$$

其中, m 为六旋翼的质量, g 为重力加速度, I 为转动惯量, r 为六旋翼的位置矩阵, q 为六旋翼的姿态矩阵, R 为机体坐标系到惯性坐标系的旋转矩阵。

展开可得

$$\begin{cases} \ddot{x} = \frac{U1(\sin\psi\sin\varphi + \cos\psi\sin\theta\cos\varphi)}{m} \\ \ddot{y} = \frac{U1(\sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi)}{m} \\ \ddot{z} = \frac{U1\cos\varphi\cos\theta}{m} - g \\ \ddot{\varphi} = \frac{U2}{I_{xx}} \\ \ddot{\theta} = \frac{U3}{I_{yy}} \\ \ddot{\psi} = \frac{U4}{I_{zz}} \end{cases} \quad (4-9)$$

其中 I_{xx} 、 I_{yy} 、 I_{zz} 分别为六旋翼绕 X、Y、Z 轴的转动惯量。

4.2 避障系统设计

4.2.1 避障系统结构设计

对于避障系统的设计，需要通过传感器得知飞行器周围的障碍物的距离与位置信息。如果仅仅使用激光传感器，激光传感器的扫描范围只有前面的 270° ，后面存在 90° 的盲区，而且飞行器上下区域也无法检测，故无法实现全面的避障。

如果仅仅使用超声波传感器，通过在六个机翼上分别固定六个超声波传感器，用于检测飞行器四周的障碍，在上面和下面分别可以固定一个超声波传感器，用于检测地面和桥面的距离。这样虽然可以得到左右方向的障碍物信息，但是六个旋翼上的超声传感器之间会存在部分的盲区，也无法获得全面的障碍物信息。一种解决方案是布置数量更多的超声传感器，但这种解决方案需要在结构上有特殊的设计，使机架上能够安装很多数量的超声传感器，而且这种方式需要开发板有更多的 I/O 口来读取超声数据。另一种解决方式是设计一个结构，用电机驱动超声传感器使其按照一定周期旋转，获得 360° 范围内障碍物的信息以避免盲区的存在，但是这种方式一方面会使整个飞行器的结构更加负责，而且转动机构会对飞行器产生一定的扭矩，如果没有其他扭矩来平衡，就会对飞行器的姿态产生干扰。

所以，单独使用激光传感器或者超声传感器，可能无法达到一个很全面的避障效果，这里选择了将激光传感器和超声传感器融合的方式，使用激光传感器检测前面 270° 的范围，超声传感器检测飞行器后方及上方与下方的障碍物距离信息，综合两种传感器的信息进行避障。如图所示为激光传感器和超声传感器布置与检测范围的示意图，激光传感器放置于飞行器的中央位置，在飞行器后方的两个机翼及之间的位置布置 3 个 MB1200 超声传感器，通过激光传感器与这三个超声传感器来检测六旋翼周围障碍物信息，放置在飞行器下方的 MB1240 传感器检测地面的距离，辅助气压计进行高度控制，放置在飞行器上方的 MB1240 超声传感器主要检测六旋翼与桥面的距离，一方面可以防止六旋翼撞上桥面，另一方面可以使六旋翼保持固定距离飞行。通过这样的方式，可以比较全面地检测周围的障碍物信息，从而进行避障处理。

激光传感器和超声波传感器的数据通过树莓派读取，然后通过串口将激光传感器的角度和距离数据、超声波传感器的数据发送给 Pixhawk 处理，控制飞行器的姿态从而防止六旋翼碰上障碍物。

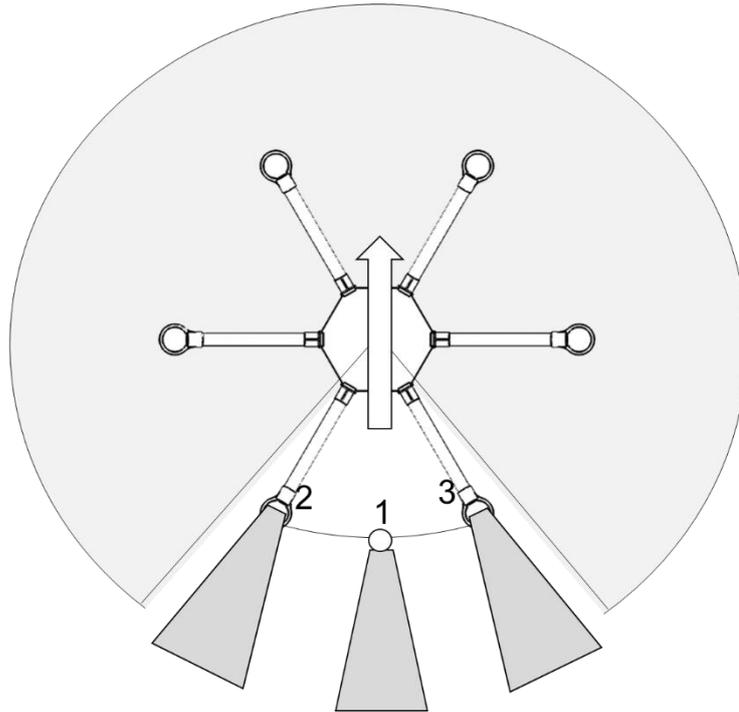


图 4-5 六旋翼激光与超声传感器检测范围示意图

4.2.2 避障系统算法设计

设定一个安全距离，当激光和超声传感器检测到周围物体距离大于安全距离时，飞行器正常飞行，当检测到的距离小于安全距离时，得到此最小距离的角度位置，切断遥控器前进方向通道的信号，并控制六旋翼后退一定距离并悬停，控制飞行器使其不撞上障碍物。

机架尺寸为直径 800mm，算上避碰结构，整体直径为 1200mm，设定障碍物到六旋翼中心的安全距离为 1200mm，当障碍物距离小于 1200mm 时，进入避障操作。设激光传感器检测到的最近的障碍物距离为 x ，角度为 θ ，1 号超声传感器检测到的障碍物距离为 x_1 ，角度为 270° ，2 号超声传感器检测到的障碍物距离为 x_2 ，角度为 240° ，3 号超声传感器检测到的障碍物距离为 x_3 ，角度为 300° 。

比较 x 、 x_1+600 、 x_2+600 、 x_3+600 的大小：

1) x 为最小值，且 $x < 1200$ 时，激光传感器有效：

$$\text{laser.}x = (1200 - x) \times P \times \cos\theta \quad (4-10)$$

$$\text{laser.}y = (1200 - x) \times P \times \sin\theta \quad (4-11)$$

其中， $0 < P < \frac{1}{600}$ ，通过调节 P 的大小，可以调节六旋翼的响应程度。

2) x_1+600 为最小值，且 $x_1 < 600$ 时，1 号超声传感器有效：

$$\text{sonar.}x = (600 - x) \times P \times \cos\frac{3\pi}{2} = 0 \quad (4-12)$$

$$\text{sonar.}y = (600 - x) \times P \times \sin\frac{3\pi}{2} = (600 - x) \times P \quad (4-13)$$

其中， $0 < P < \frac{1}{600}$ ，通过调节 P 的大小，可以调节六旋翼的响应程度。

3) x_2+600 为最小值, 且 $x_2 < 600$ 时, 2 号超声传感器有效:

$$\text{sonar.x} = (600 - x) \times P \times \cos \frac{4\pi}{3} \quad (4-14)$$

$$\text{sonar.y} = (600 - x) \times P \times \sin \frac{4\pi}{3} \quad (4-15)$$

4) x_3+600 为最小值, 且 $x_3 < 600$ 时, 3 号超声传感器有效:

$$\text{sonar.x} = (600 - x) \times P \times \cos \frac{5\pi}{3} \quad (4-16)$$

$$\text{sonar.y} = (600 - x) \times P \times \sin \frac{5\pi}{3} \quad (4-17)$$

在手动模式下, 当检测到距离小于安全距离时, 切断遥控器俯仰和横滚控制通道, 把激光或者超声传感器的计算结果赋给六旋翼的姿态设定值。

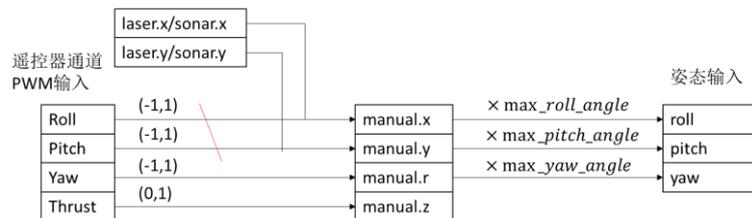


图 4-6 避障控制示意图

在位置模式下, 当检测到距离小于安全距离时, 切断遥控器 x 、 y 方向控制通道, 把激光或者超声传感器的计算结果赋给六旋翼的位置设定值。

4.3 定位方式设计

飞行器在室外飞行时主要依靠 GPS 实现精准的定位与位置控制。GPS 具有覆盖全面、定位精度较好、使用方便等优点, 但是在有障碍物阻挡的情况下, GPS 的定位精度大大的下降, 甚至无法搜到 GPS 信号。利用六旋翼无人机进行桥梁检测时, 因桥梁的阻挡, GPS 信号较差, 定位精度会不可避免地受到很大的影响。所以仅仅靠 GPS 无法实现无人机精确的定位, 需要引入新的定位方案。

4.3.1 基于加速度传感器的 GPS 盲区定位方案

利用加速度传感器获得飞行器飞行方向和加速度, 积分得到速度, 推算出飞行器位置。设飞行器的起飞位置为 (x_0, y_0) , 某一时刻测到的三个轴的加速度分别为 a_x 、为 a_y 和为 a_z , 两次测量加速度的时间差为 d_t , 则利用加速度及估算的飞行器的位置和速度分别为:

$$x(t) = x(t - d_t) + \frac{1}{2} a_x d_t^2 \quad (4-18)$$

$$y(t) = y(t - d_t) + \frac{1}{2} a_y d_t^2 \quad (4-19)$$

其中 $x(t - d_t)$ 和 $y(t - d_t)$ 为前一时刻积分得到的位置值。

当 GPS 定位有效时, 微处理器将惯性数据和 GPS 定位数据通过 Kalman 滤波融合在

一起，得到最优的定位结果。当 GPS 进入盲区或可见卫星少于 3 颗时，切换到惯性导航系统，利用 GPS 提供的初始速度和航向，通过航位推算来维持正常的导航定位。

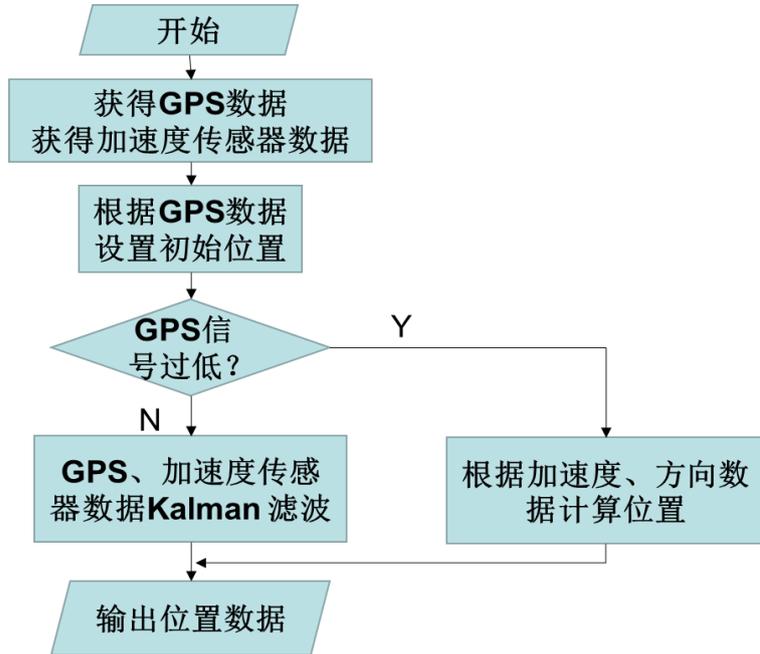


图 4-7 基于加速度传感器的 GPS 盲区定位方案

这种方案存在的一个问题就是，如果长时间没有接收到 GPS 信号，飞行器估算位置的误差会不断累积增大，所以可以通过设定飞行器飞行的轨迹，使其在巡航过程中不断从桥面出来接收更新 GPS 信号，以减小加速度计估算的位置累积误差。

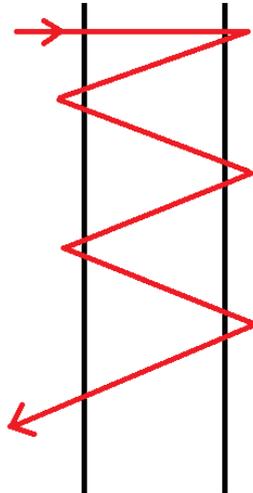


图 4-8 基于加速度传感器的 GPS 盲区定位飞行轨迹

4.3.2 激光传感器定位方案

激光传感器通过扫描可以得知周围环境的二维信息，将扫描到的二维信息点进行重构，可以得到周围环境的二维地图，在同一平面内，将激光扫描到的信息与构建的地图进行匹配，即可获得激光传感器在整个环境中的位置。

激光传感器的周围环境地图重构过程主要在 ROS (Robot Operating System) 中进行。ROS 是由 Willow Garage 公司发布的一个开源机器人操作系统，该系统主要是为了提高机器人研

发领域的代码复用率。ROS 能够为开发者提供一些标准操作系统服务，包括硬件抽象、底层设备驱动、库文件、进程间消息以及应用包管理等，开发者可以将更多的精力放在应用功能的开发上。

ROS 系统的架构主要分成了三个层级，第一个层级为文件系统级，第二个层级为计算图级，第三个为开源社区级^[16]。

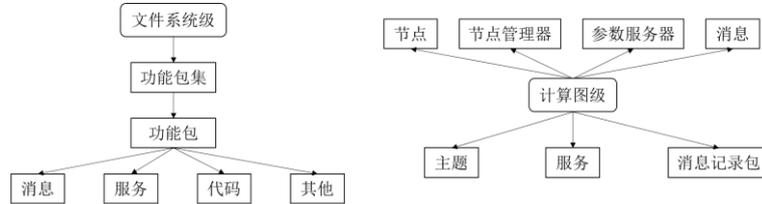


图 4-9 文件系统级和计算图级

文件系统级主要涉及一个功能包集的文件组成，功能包集下有不同的功能包，每个功能包下不同的组建包括消息、服务、代码及其他放在不同的文件夹下。功能包是 ROS 软件组织的基本形式。

ROS 的计算图级主要包括节点、节点管理器、参数服务器、消息、服务、主题和消息记录包。ROS 系统会创建一个连接所有进程的网络，系统中所有的节点能访问此网络，并与其他节点交互，获取其他节点发布的信息，并将自身的数据发布到网络中。节点之间通过消息互相通讯，主题是对所有消息进行管理的数据总线。

开源社区级则是主要关于 ROS 资源方面的内容。

使用激光传感器进行周围环境地图的重构主要由以下几个功能包实现：`hoyuko_node`、`laser_scan_matcher`、`gmapping`。`Hoyuko_node` 是一个用于获得激光数据的功能包，`laser_scan_matcher` 这个功能包用于接收激光扫描的数据，并对每两次激光扫描的数据进行对比和匹配，估算出激光传感器相对于起始时的位置坐标并发布出来，使用激光传感器进行位置估计主要由这个功能包完成，`gmapping` 这个功能包主要用于接收 `hoyuko_node` 发布的激光数据和 `laser_scan_matcher` 发布的位置数据，然后构建出周围环境的地图。

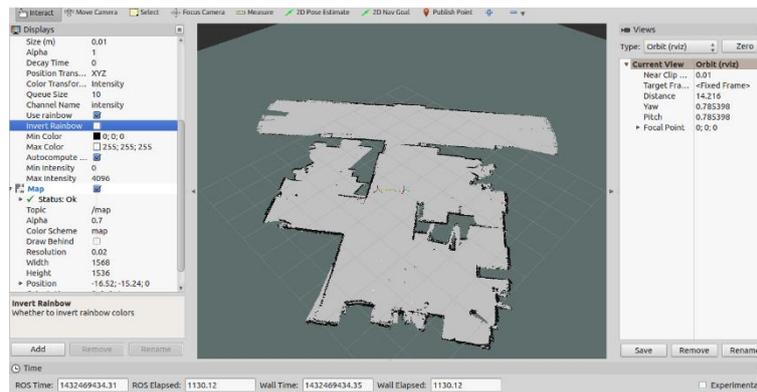


图 4-10 室内扫描的地图

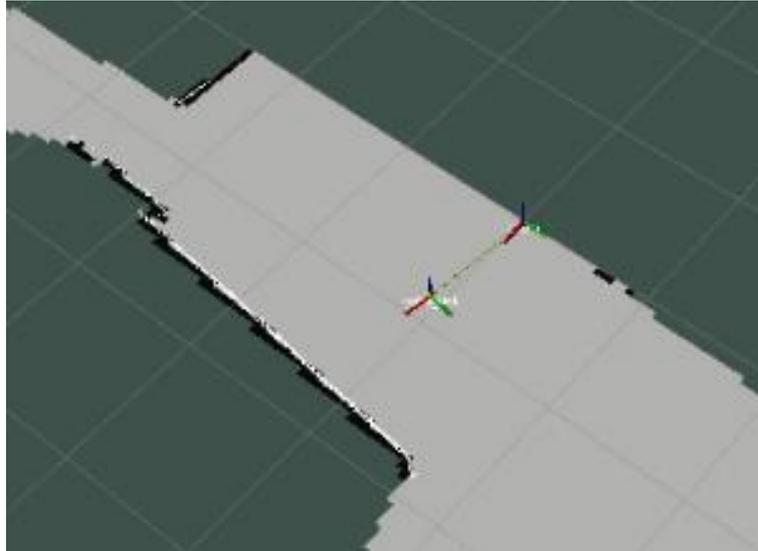


图 4-11 扫描过程激光传感器相对于起点的位置

利用激光扫描测距传感器进行定位存在的一个问题就是扫描的过程中必须保证传感器的姿态的水平，而且扫描的过程基本在同一平面内，如果只是简单得将激光传感器固定在飞行器上，飞行器的运动是通过改变姿态来实现的，不能保证激光传感器的水平，则定位会出现一定的问题。解决方式就是通过一个类似云台的装置来固定激光，飞行器运动改变姿态时，能够使激光保持水平状态

4.3.3 微基站定位方案

对清研讯科的无人机、机器人高精度定位解决方案进行了调研，其产品采用无线脉冲技术，通过在空间区域内布设有限数量的微基站，实时精确地定位无人机和机器人上的微标签位置，定位精度达到 ± 5 厘米以下，室外无遮挡情况下 50*50 米的区域可布置 4 个可获得较好的定位效果，最远可达 200 米。系统由四个及以上的微基站、微标签和相关软件组成。通过在特定区域布置四个（或者以上）的微基站，确定微基站之间的相对距离，飞行器上贴上微标签，通过解算微标签相对于四个（或者以上）的微基站的距离，即可算出飞行器相对于微基站的位置坐标。这样，微基站就可以取代 GPS 直接为飞行器进行定位，解决了桥梁底下 GPS 信号差的问题。



图 4-12 微基站系统

在微基站系统下，无人机有两种方式获得自身在区域内的相对位置：

a. 在 PC 端解算位置信息

微基站通过以太网连入到主服务器中，初始时输入布置的微基站之间组成的区域范围的距离（矩形区域的长宽值），使用其 windows 下的软件系统可解算出标签相对于 4 个微基站的位置坐标（x, y），通过无线传输方式发送给无人机处理。

b. 在无人机端解算位置信息

初始时确定布置的微基站之间组成的区域范围的距离，机载的微标签可以通过串口直接输出到每个微基站之间的距离值，机载的处理器可通过串口读取微标签输出的值，然后通过算法解算出在区域内的位置坐标。

无论是 PC 端解算位置信息，还是无人机端解算位置信息，都需要事先知道微基站布置的距离，这种方式比较适合固定环境下的定位，但是用于桥梁检测这种非固定环境下的定位，每次操作都要先确定基站之间的距离再输入就比较麻烦。所以需要在这套微基站系统进行改进以适合桥梁检测的定位。

具体的改进如下：使用 4 个微基站，每个微基站上固定一个微标签，四个微基站上的微标签可以输出其本身相对于其他三个基站的位置，将微标签连接 X-Bee（一种基于 Zig-Bee 的短距离、低功耗无线通信技术），通过 X-Bee 将距离数据发送到机载处理器，从而计算出四个微基站之间的相对距离。这样，在桥下固定 4 个微基站后，无需人工测量四个微基站的距离。无人机上固定一个微标签，无人机上的微标签通过串口将无人机相对于 4 个微基站的距离值发送到机载处理器，在知道了四个微基站的相对距离以及无人机距离四个基站的距离值后，就可以计算出无人机相对于 4 个基站的坐标。

以 A、B、C、D 四个基站中的一个基站作为坐标原点建立直角坐标系，这里以 A 点为原点，设得到矩形区域的长宽 a 和 b，标签距离四个微基站的距离为 d1、d2、d3 和 d4。通过计算，可以得到高度数据 h，然后可以得到飞行器在 x-y 平面的投影位置(x0,y0)为：

$$h = \sqrt{d_1^2 - \frac{a^2 + d_1^2 - d_2^2}{2a} - \left(\frac{d_1^2 - d_3^2 + \frac{d_2^2 + d_3^2 - d_1^2 - d_4^2}{2a}}{2b} \right)^2} \quad (4-20)$$

$$x_0 = \sqrt{d_2^2 - h^2 - \left(\frac{a^2 + d_2^2 - d_1^2}{2a} \right)^2} \quad (4-21)$$

$$y_0 = \frac{a^2 + d_2^2 - d_1^2}{2a} \quad (4-22)$$

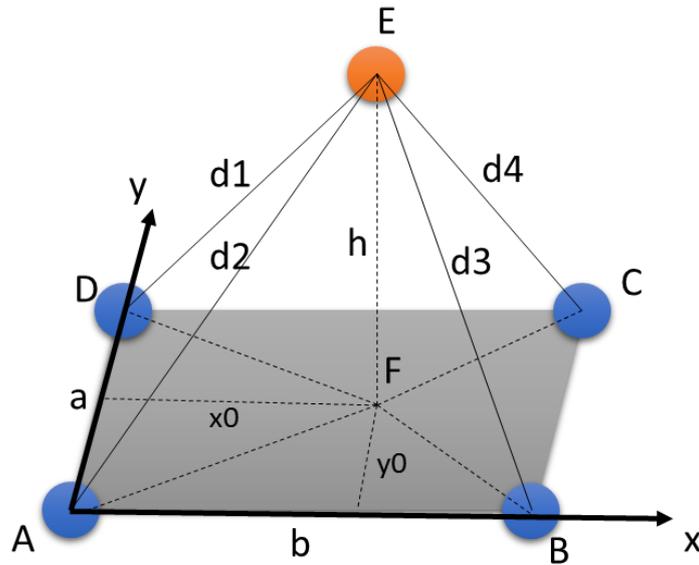


图 4-13 微基站定位示意图

4.4 定轨迹飞行算法设计

要实现六旋翼的定轨迹巡航，需要有一套定位系统给六旋翼输出位置坐标，本课题以 GPS 作为位置输出为例进行分析与设计。使用其他的定位方式，如利用微基站定位时，也可通过此算法实现飞行器的定轨迹巡航。

这里设计了两种定轨迹巡航的算法。第一种是定航向模式，即飞行器在飞行过程中机头方向始终保持不变，通过比较当前位置与目标点的相对位置，给飞行器在 X、Y 方向运动的指令来实现定轨迹飞行。第二种是航向转动模式，飞行过程中机头方向始终朝向目标点，每到一个位置，比较当前的航向与目标位置，飞行器进行偏航运动使飞行器机头指向目标位置，然后飞行器只要向前运动即可。

在使用 GPS 进行定轨迹飞行时，首先要给定一系列的航点，航点信息由经度纬度和高度组成。所以首先要把地理坐标系下的经度纬度信息转化为飞行器当地坐标系。以飞行器起飞位置作为飞行器当地坐标系的原点，以地理的东作为 X 轴，地理的北作为 Y 轴，上作为 Z 轴建立坐标系。设原点的经纬度分别为 LON_0 、 LAT_0 ，当前的经纬度坐标为 LON 、 LAT ，地球半径为 R 。

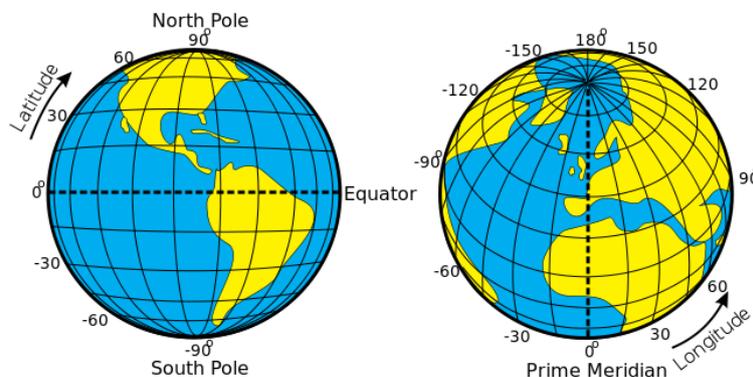


图 4-14 地理坐标系示意图

当前位置与原点的位置的经纬度差值为:

$$\Delta\text{LON} = \text{LON} - \text{LON}_0 \quad (4-23)$$

$$\Delta\text{LAT} = \text{LAT} - \text{LAT}_0 \quad (4-24)$$

将其转化为当地坐标系下的坐标:

$$x = \Delta\text{LON} \times \frac{\pi}{180} \times R \quad (4-25)$$

$$y = \Delta\text{LAT} \times \frac{\pi}{180} \times R \quad (4-26)$$

这样就可以把飞行器在地理坐标系下的位置转化为当前坐标系下的位置。

4.4.1 定航向角模式

定航向模式下,飞行器在飞行过程中不改变飞行器的航向角,通过控制飞行器的俯仰和横滚直接从当前位置飞向目标位置。设当前飞行器的位置坐标为 $A(x_1, y_1)$, 目标航点的位置坐标为 $B(x_2, y_2)$ 。

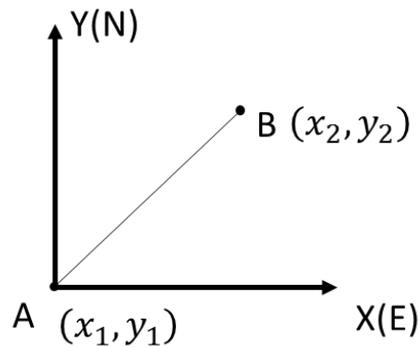


图 4-15 定航向飞行示意图

当飞行器从前一个航点飞行到达 A 点后,首先悬停稳定,然后计算目标航点与当前位置在 x 和 y 方向上的偏差, $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$,然后将目标位置与当前位置的偏差输入到位置控制器中进行计算,控制飞行器飞向目标位置。到达目标位置后,继续执行以上相同程序。

定航向角模式的定轨迹飞行,特点是飞行过程中机头始终朝向一个方向,不会因为位置的改变而改变航向,在需要相机拍摄角度始终朝向一个方向的情况下,可以使用此飞行模式。

4.4.2 航向转动模式

设机头方向与地理北极顺时针的夹角为飞行器的航向角 ψ ($0^\circ \leq \psi < 360^\circ$),设当前位置为 A,下一个目标点位置为 B,当前的航向角为 ψ_1 ,由电子罗盘给出,当前位置与下一个目标点之间的航向角为 ψ_2 ,由当前的位置与目标点的位置计算得出。

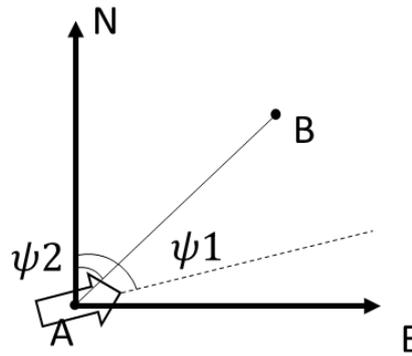


图 4-16 航向角示意图

当飞行器从前一个位置到达 A 点后，先悬停稳定，然后计算目前航向与目标航向的偏差，求得飞行的偏航角 $\Delta\psi = \psi_2 - \psi_1$ ，其中 $\Delta\psi$ 为负时，逆时针转动， $\Delta\psi$ 为正时，顺时针转动。完成偏航运动后，飞行器向前运动，直至到达下一个航点，然后执行相同程序。

这种定轨迹飞行的模式，特点就是可以保证机头方向实时朝着前进方向，在某些需要拍摄角度始终朝向机头的情况下，可以使用此模式。

第五章 实验测试与系统优化

5.1 机架的制造与装配

六旋翼机体部分及电机座的加工主要采用切割碳纤维板，机翼与起落架的主体部分、避碰结构的支撑杆的加工主要采用切割碳纤维管，其他的铝合金连接件通过加工中心加工完成。然后进行装配，完成六旋翼机架的搭建。

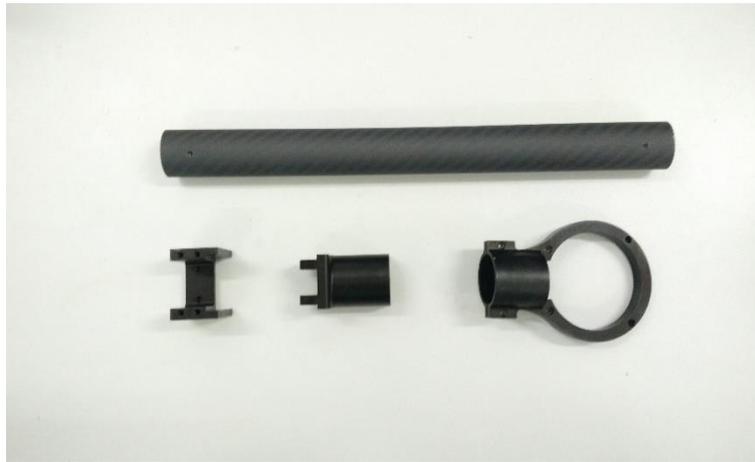


图 5-1 机翼碳管与连接件



图 5-2 机翼的装配

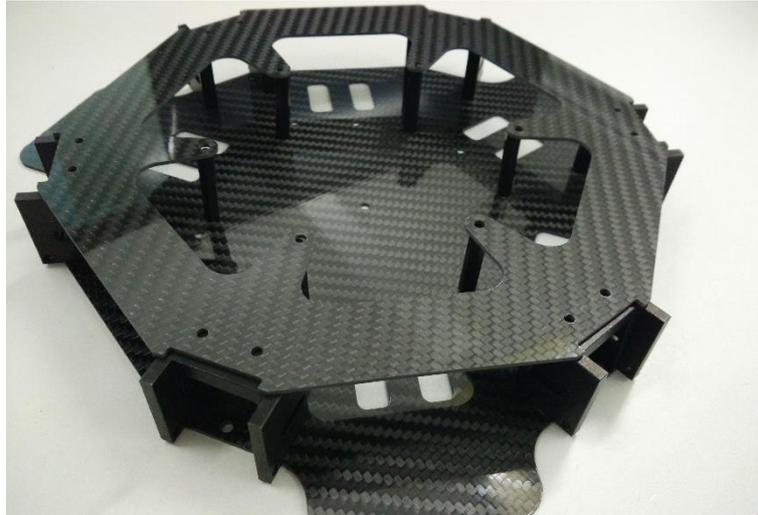


图 5-3 机体的装配



图 5-4 六旋翼整体机架的装配

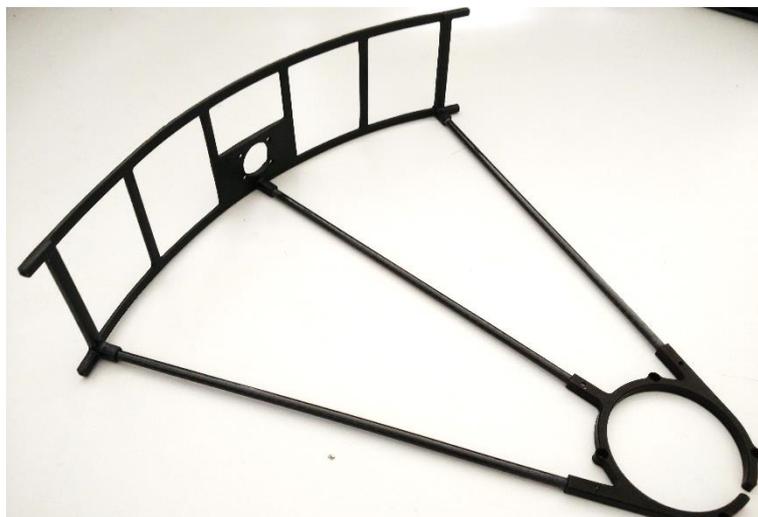


图 5-5 避碰结构



图 5-6 折叠状态的六旋翼



图 5-7 搭载动力系统与控制系统的六旋翼



图 5-8 六旋翼最终实物图

测量整体机架的重量，机翼、机体与脚架的总质量为 1938kg，搭载动力系统（包括电机、电调、电池）及飞控系统，总质量为 4.7kg，搭载所有的传感器（激光传感器、超声传感器）、相机云台等，总质量为 5.7kg。各部件的质量如下表所示。

表 5-1 六旋翼部件质量参数表

部件	质量
机架	1938g
电机（6 个）	840g（单个 160g）
电调（6 个）	210g（单个 35g）
电池	1360g
飞控系统	167g
树莓派	100g
激光传感器	370g
云台和相机	211g
避碰结构（6 个）	168g（单个 28g）

5.2 六旋翼续航能力测试

搭载动力系统（包括电机、电调、电池）及飞控系统总质量为 4.7kg 的情况下对六旋翼进行续航测试，采用 10000mAh 的电池，在室外无风环境下，使六旋翼稳定悬停，在空载情况下对六旋翼的续航进行测试，并于 DJI S800EVO 进行对比，其中 DJI S800EVO 的带电池的空载质量为 5kg。

测试条件分别为空载，相同的电池、控制系统以及测试环境，6S 电池满电情况下单节为 4.2V，总电压为 25.2V，测试结束的标志为单节电池为低于 3.5V 即整个电池电压低于 21V，整个测量过程根据电压报警器来确定电压情况。测试结果如下表所示。

表 5-2 六旋翼续航测试结果

负载情况 (kg)	续航时间 (min)
设计的机架	25
DJI S800EVO	19

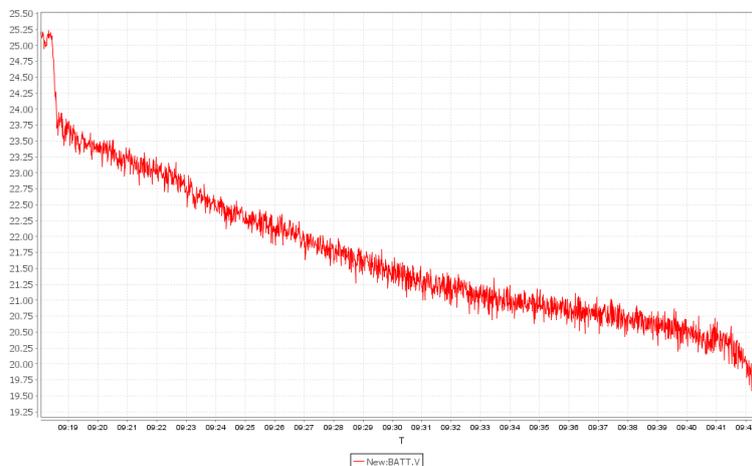


图 5-9 设计的机架续航测试电池电压图

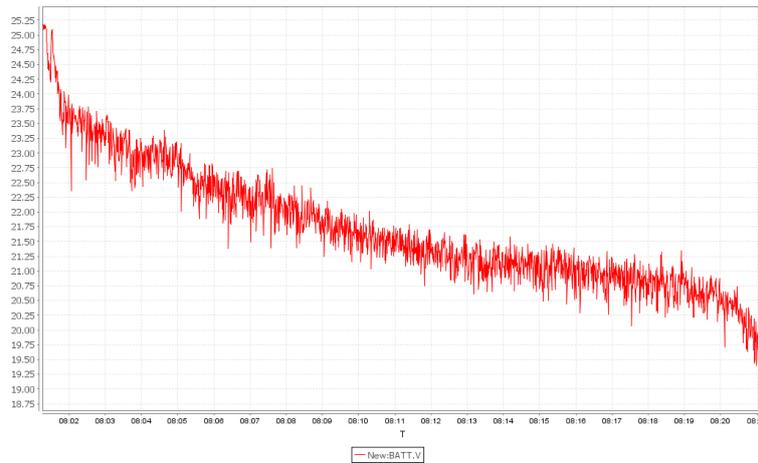


图 5-10 DJI S800EVO 续航测试电池电压图

可以看到，设计的六旋翼比 DJI S800EVO 的续航时间长了 5 分钟左右，可见设计的六旋翼空载的效率比 DJI S800EVO 的要好一些。

5.3 六旋翼基本飞行测试

完成加速度计、罗盘、磁力计、遥控器的校准，调节合适的姿态与位置 PID 参数后，在室外空旷环境下分别在手动、定高和定点模式下试飞，对飞行器进行测试，观察其悬停状态下姿态、位置的稳定性，以及进行俯仰、翻滚与偏航动作的相应程度。

首先测试了六旋翼的定点悬停，导出高度与位置的数据，并导入 matlab 进行处理，选取悬停时候的数据进行分析。

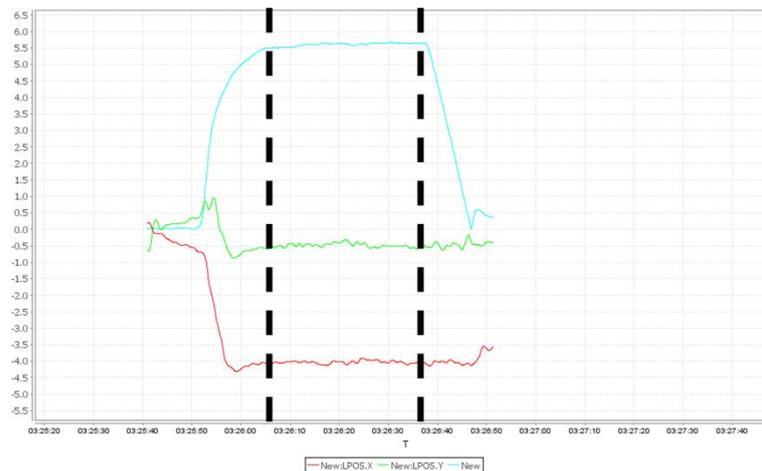


图 5-11 定点悬停下高度与位置变化图

首先分析定高精度，通过 matlab 计算，在虚线区间定点悬停的平均高度为 5.60m，最大高度为 5.67m，最小高度为 5.49m，高度标准差为 0.05m，在高度方向的定位精度大概在 $\pm 0.07m$ 以内。

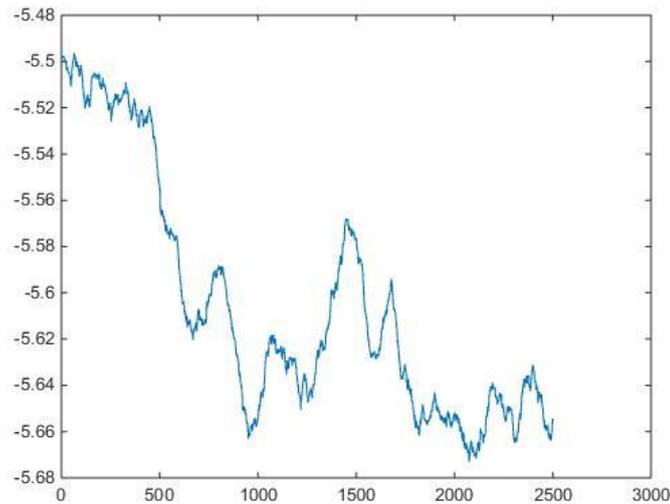


图 5-12 定点悬停下的高度变化图

分析水平方向的定位精度，在 matlab 中画出定点悬停时的六旋翼位置的分布图，利用计算出这些点 x 和 y 的平均值，然后画一个圆包围所有的点，圆的半径为 0.2m ，所以在水平方向六旋翼的定位精度大概在 0.2m 以内。

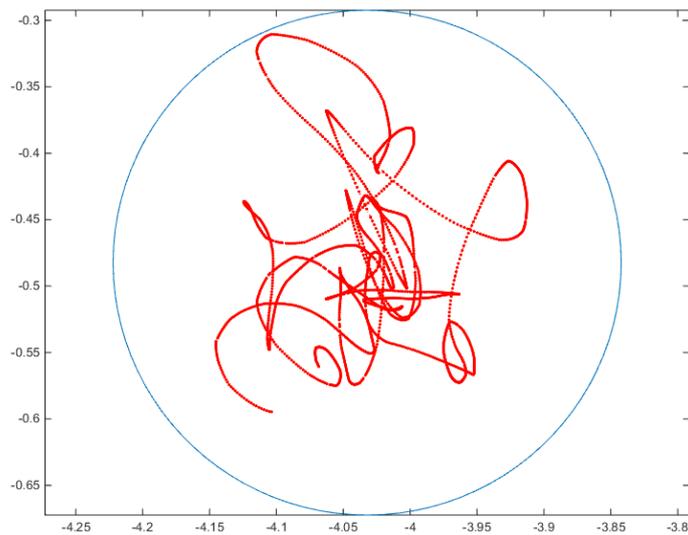


图 5-13 定点悬停下的位置变化图

经测试，飞行器在定点悬停下，高度控制的精度在 $\pm 0.07\text{m}$ 内，位置的控制的精度在 0.2m 内。

5.4 避障测试

在室外空旷环境下，使六旋翼处于定点悬停，然后人从不同方向靠近六旋翼，观察六旋翼的响应。



图 5-14 避障测试

经过测试，飞行器具有良好的避障效果，在人距离飞行器小于一定的距离后，飞行器都能够准确及时得做出响应，往原理障碍物的方向运动，当大于安全距离后，飞行器继续保持悬停状态。此过程没有其他遥控信号的输入。

5.5 定轨迹巡航测试

在室外空旷环境下，给六旋翼设定一些航点，航定如下图所示，测试六旋翼的定轨迹巡航，将测试的结果导出到 Google Earth 中画出轨迹，与设定的航点进行对比，观察偏差。

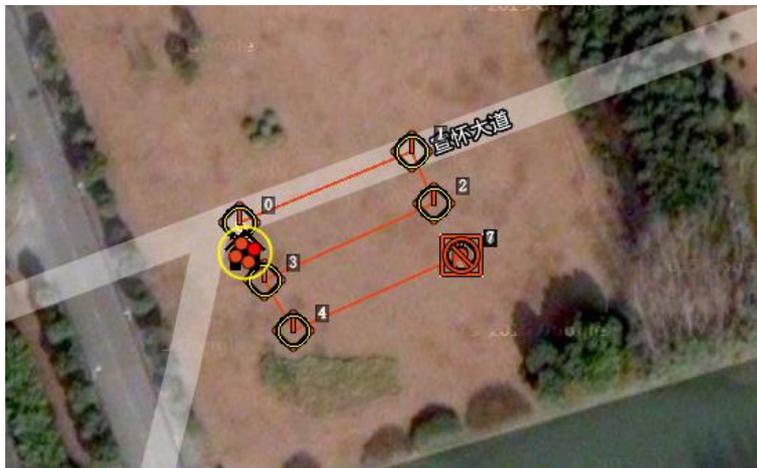


图 5-15 设定的飞行轨迹

导入到 Google Earth 的轨迹与设定的轨迹进行对比，可以看到六旋翼基本按照设定的轨迹飞行，并最终安全着陆，其中红线为设定轨迹，黄线为飞行轨迹。



图 5-16 设定轨迹与飞行轨迹对比图

5.6 系统优化

5.6.1 机架结构的优化

经过飞行测试，发现多次的飞行后，由于电机的振动，对电机连接处、机身其他连接处都有很大的影响，致使螺丝松动，造成结构安全的隐患，所以在结构的装配上，对电机座的螺栓连接处采用了弹簧垫圈，以避免电机的振动致使螺栓的松落。机架的其他螺栓螺母装配处也增加弹簧垫圈，或者使用放松螺母。

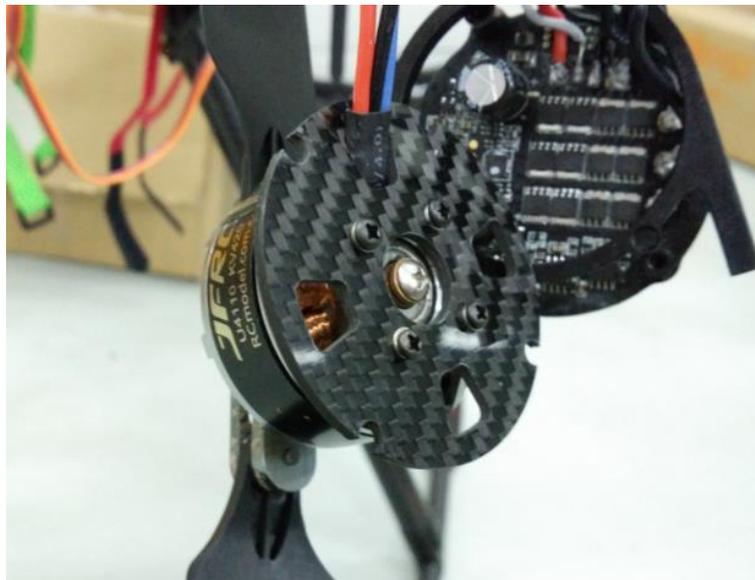


图 5-17 电机座连接处增加弹簧垫圈

由于电机座装减震球，导致连接座固定的稳定性不是很好，所以需要进一步对电机座的连接结构与减震结构进行优化，一方面保证良好的连接稳定，另一方面起到一个良好的减震效果



图 5-18 电机座减震装置

另外，在机翼的连接处设计了一个便于收放的可折叠结构，由于加工精度的问题，所以机翼的连接处会有一些小的误差，导致某些零件的连接会有一些部分的偏差，导致装配上有时会有一些松动，对飞行会有一些影响，所以这一部分的结构也是有待优化，使其能够更方便地折叠收放，并且在展开时连接比较牢固。



图 5-19 机翼连接处

5.6.2 避障系统的完善

目前避障系统能够在飞行过程中检测到障碍物不装上去，保证飞行器或者周围人的安全，但是目前还没办法实现飞行过程中检测到障碍物并自主避障并绕过去。所以这一点需要在未来进一步探究并完善，能够使飞行器自主飞行并绕开障碍物。

第六章 总结

6.1 论文总结

本课题对工业巡检用六旋翼无人机进行了设计与开发，主要针对桥梁缺陷的检测，提出解决方案。设计并加工制作了一台六旋翼样机，集成了开源的飞行控制系统完成基本飞行控制，通过超声与激光测距传感器实现了避障功能，基于开源飞行控制系统实现定轨迹飞行，提出了桥面下飞行的定位方案，基本完成了课题的目标。本课题的主要工作总结如下：

- 1) 对国内外无人机的研究现状进行了调研，调查并总结了国内外无人机公司开发的无人机的特点，并对目前无人机做桥梁检测方面的研究现状进行了调研，阐述了本课题的研究意义。
- 2) 基于六旋翼飞行器的原理，设计并开发了一款适合于工业巡检用的六旋翼无人机，并通过软件进行了应力应变分析，对机架的性能进行仿真，并对部分结构进行了优化，最终完成了样机的制作与测试飞行。
- 3) 对开源飞行控制器进行了选型，并配合主控制器，完成辅助控制器的选型，并完成主控制器与辅助控制器的通讯，基本搭建了一套完整的控制系统。
- 4) 对避障传感器进行了选项，并对超声传感器和激光扫描测距传感器进行了深入的开发，基于超声传感器和激光扫描测距传感器实现了六旋翼的避障功能。
- 5) 基于开源的飞控，在 GPS 定位方式下，实现了六旋翼无人机的定轨迹飞行，并设计其他定位方式下的定轨迹飞行算法。
- 6) 提出了几套桥梁下飞行的定位方案，第一种采用基于加速度计的 GPS 盲区定位方案，第二种采用激光扫描测距传感器的扫描与匹配进行定位，第三种采用微基站定位。

6.2 展望

工业巡检用的无人机的开发是一个较为复杂的系统开发，本文仅仅针对桥梁检测用的无人机开发了一个硬件与软件平台，并实现部分的避障功能与定轨迹飞行功能，未来还有更多需要研究的方面，主要有以下几点：

- 1) 六旋翼机架的进一步完善优化。虽然本课题设计开发的六旋翼机架基本可以实现稳定的飞行，并具有较高的刚度和强度。但是机架的性能是一个影响无人机飞行的很重要的因素，机架的重量、刚度、强度、特殊的结构等等，对飞行器的续航、飞行的稳定性等都有至关重要的影响，所以值得未来对机架进行进一步的优化。
- 2) 定位方案的进一步验证与测试。本文虽然提出了桥底下的飞行的定位方案，但限于时间与设备等原因，并未能进一步进行测试验证。然而桥底下的定位对桥梁检测的整个过程是至关重要的，所以在未来值得进一步的探索与研究。
- 3) 无人机的自主飞行。本课题目前仅仅实现了飞行过程中的避障，以及在结构性环境下人为输入航点进行定轨迹飞行的功能，未来无人机需要更加智能，通过给定一个目标，它能够在非结构性环境下自己根据周围环境规划路径进行自主飞行。本课题已经实现通过激光测距传感器



- 4) 更多工业领域的检测。本课题仅针对桥梁检测提出了一套无人机解决方案，然而像电网检测、风机检测、输油管道的检测等等，无人机都有很大的应用前景，所以未来如何能够让无人机适应不同任务的检测，也是值得未来进行探索的一个方面。

谢辞

白驹过隙，转眼已度过大学四年的时光，大学四年的生活让我收获颇丰。尤其是在做毕业设计的过程中，我将所学应用到实际的课题中，学以致用，并且在整个毕业设计的过程中，学习到很多的知识。

值此毕业论文完成之际，我首先要特别感谢指导我毕业设计的盛鑫军老师。老师渊博的专业知识，严谨的科研态度，执着的探索精神，犹如一盏明灯，为在黑夜中摸索的我指引了前进的方向。从论文选题，到资料收集，从模型设计，到制造加工，至始至终，我都得到了老师的悉心帮助，非常感谢老师。

同时，我也要感谢一直指导我的博士师兄董伟，董伟师兄在我整个毕业设计的过程中，给予了我莫大的帮助。每当我碰到技术性的难题，师兄都能耐心指导，帮我解决关键性的问题。

另外，我也要感谢实验室的硕士师兄方家豪、学弟张乔宇、叶欣、陈刚在整个毕设过程中给予我的帮助，他们总能为我伸出援手，尽管自己也都很忙，可他们还是挤出了时间来为我解答疑惑。他们的宝贵建议让我少走了不少弯路。

我还要感谢爱我的爸爸妈妈和姐姐。做毕业设计时间紧，任务重，他们或许不懂我的课题，但他们给予我生活上的无私关怀让我在劳累时总能提起精神，给我前进的动力。他们在我整个本科期间，全力支持我的学习生活，无论是经济上的还是精神上的，都给了我莫大的支持。

最后再一次对帮助我顺利完成毕业设计的导师、师兄、学弟和家人们表示由衷的感谢！

参考文献

- [1] 聂博文,马宏绪,王剑,王建文. 微小型四旋翼飞行器的研究现状与关键技术[J]. 电光与控制,2007,06:113-117.
- [2] Bouabdallah S, Murrieri P, Siegwart R. Design and control of an indoor micro quadrotor[C]//Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. IEEE, 2004, 5: 4393-4398.
- [3] Vijay Kumar and Nathan Michael. Opportunities and challenges with autonomous micro aerial vehicles[J]. The International Journal of Robotics Research, 31(11) 1279–1291
- [4] 崔秀敏,王维军,方振平. 小型无人机发展现状及其相关问题分析[J]. 飞行力学,2005,01:14-18.
- [5] Mahony R, Kumar V, Corke P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor[J]. IEEE Robotics & Automation Magazine, 2012 (19): 20-32.
- [6] 马建,孙守增,杨琦,赵文义,王磊,马勇,刘辉,张伟伟,陈红燕,陈磊,康军. 中国桥梁工程学术研究综述 2014[J]. 中国公路学报,2014,05:1-96.
- [7] 戴启凡. 桥梁检测爬壁机器人及其自适应控制技术研究[D].南京理工大学,2014.
- [8] Alex Kushleyev, Daniel Mellinger, Vijay Kumar. Towards a swarm of agile micro quadrotors[J]. Auton Robot (2013) 35:287-300
- [9] Shen S, Mulgaonkar Y, Michael N, et al. Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor[C]//Robotics: Science and Systems. 2013.
- [10] Shen S, Mulgaonkar Y, Michael N, et al. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav[C]//Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014: 4974-4981.
- [11] Shen S, Michael N, Kumar V. Autonomous indoor 3D exploration with a micro-aerial vehicle[C]//Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012: 9-15.
- [12] Dydek Z T, Annaswamy A M, Lavretsky E. Adaptive control of quadrotor UAVs: A design trade study with flight evaluations[J]. Control Systems Technology, IEEE Transactions on, 2013, 21(4): 1400-1406.
- [13] Metni N, Hamel T. A UAV for bridge inspection: Visual servoing control law with orientation limits[J]. Automation in construction, 2007, 17(1): 3-10.

- [14] Serrano N E. Autonomous quadrotor unmanned aerial vehicle for culvert inspection[D]. Massachusetts Institute of Technology, 2011.
- [15] Dong W, Gu G, Zhu X, et al. Modeling and Control of a Quadrotor UAV with Aerodynamic Concepts[J]. World Academy of Science, Engineering and Technology, 2013, 77.
- [16] (西)马丁内斯等著; 刘平杰译. ROS 机器人的程序设计[M]. 北京: 机械工程出版社, 2014.8.

附录

附录一 树莓派读取激光数据并串口发送程序

```
#include <stdio.h> /*标准输入输出定义*/
#include <stdlib.h> /*标准函数库定义*/
#include <unistd.h> /*Unix 标准函数定义*/
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h> /*文件控制定义*/
#include <termios.h> /*PPSIX 终端控制定义*/
#include <errno.h> /*错误号定义*/
#include <string.h>

#define BAUDRATE B19200 ///Baud rate : 19200
#define DEVICE "/dev/ttyACM0"
#define SERIAL_PORT "/dev/ttyAMA0"
#define MAX_DISTANCE 30000
#define DANGER_DISTANCE 1600
#define WARNING_DISTANCE 2000

int read_fd,serial_fd;
char databuf[4096];
int frame_len;
bool frame_start;
char distancebuf[3243];
int distance[1081];
char parse_head[2]={'G','D'};
char parse_end[2]={'\n','\n'};

char send_buf[50];

int set_serial(int fd,int nSpeed, int nBits, char nEvent, int nStop)
{
    struct termios newtio,oldtio;
    if ( tcgetattr( fd,&oldtio) != 0) {
        perror("SetupSerial 1");
```



```
    return -1;
}
bzero( &newtio, sizeof( newtio ) );
newtio.c_cflag |= CLOCAL | CREAD;
newtio.c_cflag &= ~CSIZE;

switch( nBits )
{
    case 7:
        newtio.c_cflag |= CS7;
        break;
    case 8:
        newtio.c_cflag |= CS8;
        break;
}

switch( nEvent )
{
    case 'O':
        newtio.c_cflag |= PARENB;
        newtio.c_cflag |= PARODD;
        newtio.c_iflag |= (INPCK | ISTRIP);
        break;
    case 'E':
        newtio.c_iflag |= (INPCK | ISTRIP);
        newtio.c_cflag |= PARENB;
        newtio.c_cflag &= ~PARODD;
        break;
    case 'N':

        newtio.c_cflag &= ~PARENB;
        break;
}

switch( nSpeed )
{
    case 2400:
        cfsetispeed(&newtio, B2400);
        cfsetospeed(&newtio, B2400);
        break;
    case 4800:
        cfsetispeed(&newtio, B4800);
        cfsetospeed(&newtio, B4800);
        break;
}
```



```
        case 9600:
            cfsetispeed(&newtio, B9600);
            cfsetospeed(&newtio, B9600);
            break;
        case 19200:
            cfsetispeed(&newtio, B19200);
            cfsetospeed(&newtio, B19200);
            break;
        case 115200:
            cfsetispeed(&newtio, B115200);
            cfsetospeed(&newtio, B115200);
            break;
        case 460800:
            cfsetispeed(&newtio, B460800);
            cfsetospeed(&newtio, B460800);
            break;
        default:
            cfsetispeed(&newtio, B9600);
            cfsetospeed(&newtio, B9600);
            break;
    }

    if( nStop == 1 )
    {
        newtio.c_cflag &= ~CSTOPB;
    }
    else if ( nStop == 2 )
    {
        newtio.c_cflag |= CSTOPB;
    }

    newtio.c_cc[VTIME] = 100;//重要
    newtio.c_cc[VMIN] = 0;//返回的最小值 重要
    tcflush(fd,TCIFLUSH);
    if((tcsetattr(fd,TCSANOW,&newtio))!=0)
    {
        perror("com set error");
        return -1;
    }
    return 0;
}

//设备使能
void switch_on()
```



```
{
    int ret;
    char write_buf[20];

    memset(write_buf,0,20);
    sprintf(write_buf, "BM\n");
    ret = write(read_fd,write_buf,3);
    if(-1 == ret)
    {
        perror("Switch_on Write Data Error!\n");
        exit(1);
    }
}

void switch_off()
{
    int ret;
    char write_buf[20];

    memset(write_buf,0,20);
    sprintf(write_buf, "QT\n");
    ret = write(read_fd,write_buf,3);
    if(-1 == ret)
    {
        perror("Switch_off Write Data Error!\n");
        exit(1);
    }
}

//获得一次数据
void get_data_by_GD(int inStart, int inEnd, int inClusterCnt)
{
    int ret;
    char write_buf[20];

    memset(write_buf,0,20);
    sprintf(write_buf, "GD%04d%04d%02d\n", inStart, inEnd,
inClusterCnt);
    ret = write(read_fd,write_buf,13);
    if(-1 == ret)
    {
        perror("Get Data Write Data Error!\n");
        exit(1);
    }
}
```



```
}

//获得设备信息
void get_data_by_VV()
{
    int ret;
    char write_buf[20];

    memset(write_buf,0,20);
    sprintf(write_buf, "VV\n");
    ret = write(read_fd,write_buf,3);
    if(-1 == ret)
    {
        perror("Get Data Write Data Error!\n");
        exit(1);
    }else
        printf("OK\n");
}

void read_data()
{
    int ret;

    memset(databuf,0,4096);
    ret = read(read_fd, databuf, 4096);
    if(-1 == ret)
    {
        perror("Read Data Error!\n");
        exit(1);
    }
}

long urg_decode(const char *data, int data_byte)
{
    long value = 0;
    for (int i = 0; i < data_byte; ++i)
    {
        value <<= 6;
        value &= ~0x3f;
        value |= data[i] - 0x30;
    }
    return value;
}
```



```
void parse(char *data)
{
    int nData;
    int nBlock;
    int nLeft;
    int nstart;
    int distance_num;
    char* pDataBuf;

    frame_len = 0;
    frame_start = false;
    memset(distance,0,1081);

    for(int i=0; i<4096; i++)
    {
        if(frame_start ==false)
        {
            if( (data[i]=='G') && (data[i+1]=='D') )
            {
                frame_start = true;
                nstart = i;
                frame_len++;
            }
        }
        else
        {
            frame_len++;
            if( (data[i]==parse_end[0]) &&
(data[i+1]==parse_end[1]) )
            {
                frame_len++;;
                break;
            }
        }
    }

    nData = frame_len - 23/*cmd*/ - 1/*最后一个LF*/;
    nBlock = nData/66;
    nLeft = nData%66 - 2/*sum+LF*/;

    pDataBuf = data + nstart +23 ;
    //data block
    for (int i=0;i<nBlock;i++)
    {
```



```
        memcpy((distancebuf+i*64),pDataBuf,64);
        pDataBuf += 66;
    }
    //left
    if (nLeft>0)
    {
        memcpy((distancebuf+nBlock*64),pDataBuf,nLeft);
    }

    distance_num = (nBlock*64+nLeft)/3;
    pDataBuf = distancebuf;

    for (int i=0;i<distance_num;i++)
    {
        int temp;
        temp = urg_decode(pDataBuf,3);
        if (temp < 10)
        {
            distance[i] = 30000;
        }
        else
            distance[i] = temp;
        pDataBuf += 3;
    }
}

void record()
{
    int ret,fd;
    char record_buf[11891];
    char *p;

    fd = open("/home/chenjie/test" ,O_RDWR);
    ret = write(fd,databuf,4096);
    if (ret == -1)
    {
        perror("write Error!\n");
        exit(1);
    }
    close(fd);

    p = record_buf;
    for(int i=0;i<1081;i++)
    {
```



```
        sprintf(p, "%04d:%05d\n", i, distance[i]);
        p+=11;
    }
    fd = open("/home/chenjie/test1" ,O_RDWR);
    ret = write(fd,record_buf,11891);
    if (ret == -1)
    {
        perror("write Error!\n");
        exit(1);
    }
    close(fd);
}

void send(char*data, int data_byte)
{
    int ret;

    ret = write(serial_fd,data,data_byte);
    if (ret == -1)
    {
        perror("write Error!\n");
        exit(1);
    }
}

void init()
{
    int ret;
    read_fd = open(DEVICE, O_RDWR);//打开串口
    if (read_fd == -1)
    {
        perror("Open Error!\n");
        exit(1);
    }
    ret = set_serial(read_fd,BAUDRATE, 8, 'N', 1);//设置串口属性
    if (ret == -1)
    {
        perror("Set Error!\n");
        exit(1);
    }

    serial_fd = open(SERIAL_PORT, O_RDWR);//打开串口
    if (serial_fd == -1)
    {
```



```
        perror("Open Error!\n");
        exit(1);
    }
    ret = set_serial(serial_fd,115200, 8, 'N', 1);//设置串口属性
    if (ret == -1)
    {
        perror("Set Error!\n");
        exit(1);
    }
}

void calculate()
{
    int min_distance;
    int angle;

    min_distance = distance[0];
    angle = 0;
    memset(send_buf,0,50);
    for(int i=1;i<1081;i++)
    {
        if(distance[i]<min_distance)
        {
            min_distance = distance[i];
            angle = i*0.25;
        }
    }
    sprintf(send_buf,"M%03d%05d\n" , angle , min_distance);
}

int main(void)
{
    int ret;

    init();
    switch_on();

    while(1)
    {
        get_data_by_GD(0,1080,1);
        usleep(75000);
        read_data();
        parse(databuf);
        calculate();
    }
}
```

```
        ret = write(serial_fd, send_buf, 10);
        if(ret == -1)
        {
            perror("write error!\n");
            exit(1);
        }
        printf("send OK\n");
        for(int i=0; i<10; i++)
        {
            printf("%c", send_buf[i]);
        }
    }
    close(serial_fd);
    return 0;
}
```

附录二 树莓派读取声纳数据并显示

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <wiringPi.h>

int sonar0_start_time;
int sonar0_stop_time;
int sonar0_time;

void Interrupt0()
{
    if(digitalRead(0)==1)
    {
        sonar0_start_time = micros();
    }
    if(digitalRead(0)==0)
    {
        sonar0_stop_time = micros();
    }
}

int main(void)
{
```

```
int distance0;
wiringPiSetup();
wiringPiISR(0, INT_EDGE_BOTH, &Interrupt0);
for(;;)
{
    if(sonar0_stop_time > sonar0_start_time){
        sonar0_time = sonar0_stop_time - sonar0_start_time;
    }

    distance0 = sonar0_time/58;
    //printf("start_ime = %d\n",start_time);
    //printf("stop_ime = %d\n",stop_time);
    //printf("time = %d\n",time);
    printf("distance0 = %d\n",distance0);
    delay(500);
}
}
```

附录三 Pixhawk 接收激光数据并发布

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include <nuttx/config.h>
#include <nuttx/sched.h>

#include <systemlib/systemlib.h>
#include <systemlib/err.h>

#include <sys/types.h>
#include <stdint.h>
#include <stdbool.h>
#include <semaphore.h>
#include <fcntl.h>
#include <poll.h>
#include <errno.h>
#include <math.h>
#include <termios.h>
#include <nuttx/arch.h>
#include <nuttx/clock.h>
#include <drivers/drv_hrt.h>
#include <drivers/device/device.h>
```

```
#include <uORB/uORB.h>
#include <board_config.h>
#include <uORB/topics/laser_msg.h>

/* oddly, ERROR is not defined for c++ */
#ifdef ERROR
# undef ERROR
#endif
static const int ERROR = -1;

// designated SERIAL4/5 on Pixhawk
#define SERIAL_PORT    "/dev/ttyS6"
#define MAXSIZE 30

static bool thread_should_exit = false;    /**< serial_test exit
flag */
static bool thread_running = false;    /**< serial_test status
flag */
static int serial_task;    /**< Handle of serial_test task /
thread */

int _serial_fd;
char readbuf[50];

char ringbuf[MAXSIZE];
int read_addr=0;
int write_addr=0;

int angle = 0;
int distance = 30000;

extern "C" __EXPORT int serial_test_main(int argc, char *argv[]);
int serial_test_thread_main(int argc, char *argv[]);
void serial_init();
int read();
int set_serial(int fd,int nSpeed, int nBits, char nEvent, int nStop);
static void usage(const char *reason);

int next_data_handle(int addr);
void write_data(char data);
void read_data();

static void
usage(const char *reason)
```



```
{
    if (reason) {
        warnx("%s\n", reason);
    }

    errx(1, "usage: serial_test {start|stop|status} [-p <additional
params>]\n\n");
}

/*ringbuffer*/
int next_data_handle(int addr)
{
    return (addr+1) == MAXSIZE ? 0:(addr+1) ;
}

void write_data(char data)
{
    *(ringbuf+write_addr) = data;
    write_addr = next_data_handle(write_addr);
}

void read_data()
{
    if(ringbuf[read_addr] == 'M')
    {
        angle = 100*(ringbuf[read_addr+1]-48) +
10*(ringbuf[read_addr+2]-48) + (ringbuf[read_addr+3]-48);
        distance = 10000*(ringbuf[read_addr+4]-48) +
1000*(ringbuf[read_addr+5]-48) + 100*(ringbuf[read_addr+6]-48)+
10*(ringbuf[read_addr+7]-48) +
(ringbuf[read_addr+8]-48);
        for(int i=0 ; i<10; i++)
        {
            read_addr = next_data_handle(read_addr);
        }
    }else
    {
        for(int i=0 ; i<10; i++)
        {
            read_addr = next_data_handle(read_addr);
            if(ringbuf[read_addr] == 'M')
            {
                break;
            }
        }
    }
}
```



```
    }  
  }  
}  
  
/*ringbuffer*/  
int set_serial(int fd,int nSpeed, int nBits, char nEvent, int nStop)  
{  
    struct termios newtio,oldtio;  
    if ( tcgetattr( fd,&oldtio) != 0) {  
        perror("SetupSerial 1");  
        return -1;  
    }  
    bzero( &newtio, sizeof( newtio ) );  
    newtio.c_cflag |= CLOCAL | CREAD;  
    newtio.c_cflag &= ~CSIZE;  
  
    switch( nBits )  
    {  
        case 7:  
            newtio.c_cflag |= CS7;  
            break;  
  
        case 8:  
            newtio.c_cflag |= CS8;  
            break;  
    }  
  
    switch( nEvent )  
    {  
        case 'O':  
            newtio.c_cflag |= PARENB;  
            newtio.c_cflag |= PARODD;  
            newtio.c_iflag |= (INPCK | ISTRIP);  
            break;  
  
        case 'E':  
            newtio.c_iflag |= (INPCK | ISTRIP);  
            newtio.c_cflag |= PARENB;  
            newtio.c_cflag &= ~PARODD;  
            break;  
  
        case 'N':  
            newtio.c_cflag &= ~PARENB;  
            break;  
    }  
}
```



```
}  
  
switch( nSpeed )  
{  
    case 2400:  
        cfsetispeed(&newtio, B2400);  
        cfsetospeed(&newtio, B2400);  
        break;  
  
    case 4800:  
        cfsetispeed(&newtio, B4800);  
        cfsetospeed(&newtio, B4800);  
        break;  
  
    case 9600:  
        cfsetispeed(&newtio, B9600);  
        cfsetospeed(&newtio, B9600);  
        break;  
  
    case 19200:  
        cfsetispeed(&newtio, B19200);  
        cfsetospeed(&newtio, B19200);  
        break;  
  
    case 115200:  
        cfsetispeed(&newtio, B115200);  
        cfsetospeed(&newtio, B115200);  
        break;  
  
    case 460800:  
        cfsetispeed(&newtio, B460800);  
        cfsetospeed(&newtio, B460800);  
        break;  
  
    default:  
        cfsetispeed(&newtio, B9600);  
        cfsetospeed(&newtio, B9600);  
        break;  
}  
  
if( nStop == 1 )  
{  
    newtio.c_cflag &= ~CSTOPB;  
}
```



```
else if ( nStop == 2 )
{
    newtio.c_cflag |= CSTOPB;
}

newtio.c_cc[VTIME] = 100;
newtio.c_cc[VMIN] = 0; //返回的最小值
tcflush(fd,TCIFLUSH);
if((tcsetattr(fd,TCSANOW,&newtio))!=0)
{
    perror("com set error");
    return -1;
}
return 0;
}

void serial_init(){
    _serial_fd = open(SERIAL_PORT, O_RDWR | O_NOCTTY | O_NONBLOCK);
    warnx("serial open: %d",_serial_fd);
    if (_serial_fd < 0) {
        warnx("FAIL: serial fd");
        exit(0);
    }
}

int read()
{
    int ret ;

    memset(readbuf,0,50);
    ret = read(_serial_fd, readbuf, 10);
    if (ret < 0) {
        warnx("read err: %d\n", ret);
        return -1;
    }
    else
        return 0;
}

int serial_test_main(int argc, char *argv[])
{
    if (argc < 1) {
```



```
        usage("missing command");
    }

    if (!strcmp(argv[1], "start")) {

        if (thread_running) {
            warnx("serial_test already running\n");
            /* this is not an error */
            exit(0);
        }

        thread_should_exit = false;
        serial_task = task_spawn_cmd("serial_test",
                                     SCHED_DEFAULT,
                                     SCHED_PRIORITY_DEFAULT,
                                     2000,
                                     serial_test_thread_main,
                                     (argv) ? (char * const *)&argv[2] : (char * const
*)NULL);
        exit(0);
    }

    if (!strcmp(argv[1], "stop")) {
        thread_should_exit = true;
        exit(0);
    }

    if (!strcmp(argv[1], "status")) {
        if (thread_running) {
            warnx("\trunning\n");

        } else {
            warnx("\tnot started\n");
        }

        exit(0);
    }

    usage("unrecognized command");
    exit(1);
}

int serial_test_thread_main(int argc, char *argv[])
{

    int ret;
```

```
warnx("[serial_test] starting\n");
thread_running = true;

/* advertise laser topic */
struct laser_msg_s laser;
memset(&laser, 0, sizeof(laser));
orb_advert_t _laser_pub = orb_advertise(ORB_ID(laser_msg),
&laser);

serial_init();
ret = set_serial(_serial_fd, 115200, 8, 'N', 1);
if (ret == -1)
{
    warnx("set error:%d\n", ret);
    exit(0);
}
while (!thread_should_exit) {
    angle = 0;
    distance = 30000;

    ret = read();
    if (ret == -1)
    {
        laser.laser_distance = distance;
        laser.laser_angle = angle;
        orb_publish(ORB_ID(laser_msg), _laser_pub,
&laser);

        printf("angle:%d\n", angle);
        printf("distance:%d\n", distance);
        printf("\n[message] Read:OK\n");
    } else
    {
        for (int i=0; i<10; i++)
        {
            write_data(readbuf[i]);
        }
        read_data();
        laser.laser_distance = distance;
        laser.laser_angle = angle;
        orb_publish(ORB_ID(laser_msg), _laser_pub,
&laser);

        printf("angle:%d\n", angle);
```



```
        printf("distance:%d\n", distance);
        printf("\n[message] Read:OK\n");
    }
    usleep(75000);
}

warnx("[serial_test] exiting.\n");
thread_running = false;
return 0;
}
```

DESIGN AND DEVELOPMENT OF MULTICOPTER FOR INDUSTRIAL INSPECTION

In recent years, with the development of new materials, micro-electromechanical systems (MEMS), micro inertial measurement unit (MIMU) and flight control technology, micro UAVs is growing rapidly and drawing more and more attention. Micro UAVs are divided into three categories, including fixed wing, helicopters and multi-copters. Particularly, multi-rotor UAV is subject to a great deal of attention and researches because of its small size, simple mechanical structure, capable of vertical landing and hovering characteristics. In addition to aerial photography, multi-rotor UAV also has great potential applications in industry, agriculture, public security, scientific research and even military. Compared to the traditional stationary industrial robots, micro UAVs, which is one kind of new mobile robot, can work in three-dimensional space and have a wide range of applications, such advantage make it become the focus of research in the next few decades.

On the other hand, since the reform and opening up policy, China's bridge construction continue to develop. In order to prevent the occurrence of bridge accidents, frequent bridge inspections are required to find the defects of the bridges. And current detection methods are mainly done manually, which are cumbersome and unsafe. So new bridge detection methods are urgently demanded.

The main topic of the research is based on the above two background. This issue aims to design and develop a six-rotor unmanned aerial vehicle on industrial inspection for bridge inspection. It can acquire the image of the bridge by an onboard camera, so that we can analyze the image to find the cracks and other defects of the bridge. Compared to the traditional inspection methods, the six-rotor UAV won't be limited to space, topography and other factors, and assess inspection results from more angles and positions, which will simplify the inspection process, improve the inspection efficiency, guarantee personnel safety, and reduce the incidence of high-risk job accidents.

An industrial six-rotor UAV system is designed and developed in this issue, which is mainly used for bridge inspection applications. The issue comes up with some solutions to make the UAV realize the following functions: Get a clear image of a stable bridge defects; Hover stably under the bridge; Do automatic flight mission in accordance with the specified track; Automatically avoid the obstacle during the flight.

To complete the task, the issue first does some research on the micro UAVs, and makes a brief description of the product of some UAV companies. The result of the investigation shows that there is great potential to do some industrial inspection such as bridge inspection using the UAVs.

Then, based on the principle of six-rotor vehicle, the issue designs a six-rotor frame, and simulates the performance of the frame by stress-strain analysis in the software. After that, some optimization is carried out to make the structure more and more reliable. Finally a six-rotor frame

prototype is manufactured and tested.

Thirdly, the issue builds a control system. The control system is based on an open source project named PX4 project, and uses Pixhawk as the main controller of the system. There are some basic sensors for attitude and position estimate on Pixhawk, including accelerometer, gyroscope, magnetometer, barometer and GPS. With the help of the PX4 project, the system can realize the basic control of the vehicle, including roll, pitch and yaw. To integrate more sensors to the system, a secondary controller is also selected. We use Raspberry Pi 2 to read and handle the data of the laser scan sensor and the ultrasonic sensors, and communicate with Pixhawk to send the result of the additional sensors data. Here, we use the laser scan sensor and the ultrasonic sensors to detect the surroundings and find the obstacle, so that the vehicle can automatically avoid the obstacle.

The obstacle avoidance system is combined with a laser scanner and five ultrasonic sensors. UTM-30LX laser scanner is use to scan and get the distance of the obstacle in a range of 270° in the front of the vehicle, and three ultrasonic sensors are arranged at the back of the vehicle to get the information back of the vehicle where the laser scanner can't reach. And the other two ultrasonic sensors is arranged at the up and down of the vehicle to detect the distance of the ground and the bridge. Set a safe distance, when the distance detecting by laser scanner and ultrasonic sensor is greater than the safe distance around, aircraft fly normally. However, when the distance is less than the safe distance, cut off the forward direction channel remote control signal and give a control signal according to the angle and distance information of the obstacles to the vehicle and control the vehicle to move back a certain distance and hover, so that the vehicle won't crash into the obstacles.

To hover stably under the bridge, the issue also comes up with three solution of positioning the vehicle under the bridge. The first is combining the IMU and the GPS to estimate the position of the vehicle. GPS can get the initial position, and when the GPS signal is poor, accelerator and the compass will be used to estimate the position until the GPS signal is fine again. The second solution is using the laser scanner to estimate the position of the vehicle. Laser scanner can get two dimensional point cloud of the surroundings, we can match every two scan to estimate the position of the vehicle. However, we must guarantee that the laser scanner is in the same plane when using the laser scanner to estimate the position. The third solution is using micro base station to positioning the vehicle. At least four micro base station will be used to make a rectangular region, and there is a label on the vehicle, which can output the distance between the label and the each base station. By the distance information, we can calculate the position of the vehicle relative to the base station.

The issue designs two automatic cruise flight algorithm. One is flight with fixing head direction, in this flight mode, the aircraft won't yaw during the process of automatic cruise flight, it moves from one point to the next waypoint just by roll and pitch. The other mode is flight with rotary head direction, in this mode, the head of the aircraft is always point to the next waypoint, when it arrive at the waypoint, it yaws to make the head point to the next waypoint, and then just pitch forward. This two flight mode can meet the demand of different situation of photographing.

Finally, the issue makes some test on the six-rotor aircraft, including weight testing, flight time testing, obstacle avoidance testing, and automatic cruise flight testing, to verify the performance of the vehicle.

The issue comes up with a primary solution of micro UAV for bridge inspection, however, there

are more aspects to research in the future. First is the optimization of the structure, the weight, stiffness, strength, and some special structures will contribute a lot to the performance of the aircraft. Second is that the positioning solutions need further verification and testing, because it's the key point for the aircraft to fly under the bridge. Third is the autonomous flight of the UAV. In the future, the UAV needs to be more intelligent, by giving a target, it can autonomous flight to the destination according to the surrounding environment by itself.

To make a conclusion, the innovation of issue is that it put forward an unmanned aerial vehicle system for bridge inspection, which will simplify the inspection process, improve the inspection efficiency, guarantee personnel safety, and reduce the incidence of high-risk job accidents. Such industrial inspection system will absolutely change the traditional inspection process, and can be widely applied to some other region such as power lines inspection, windmills inspection, oil and gas pipelines inspection and so on. So such unmanned aerial vehicle inspection system has a great potential in the future.