

摘要

近年来，得益于MEMS传感器技术的发展与微型处理器性能的提高，四轴飞行器受到许多研究机构或公司的关注。但是，目前能够实现自主飞行的小型飞行器很少。在室外环境中，飞行器可以通过GPS定位来实现自主飞行，但是GPS定位的定位精度较低，更新速率慢，延迟大，难以实现精确的轨迹跟踪与进行机动性强的飞行。在室内环境中，由于GPS信号衰减很大，飞行器往往需要依靠复杂的图像处理算法来进行定位，由于算法运算量大导致数据更新速率低，飞行器需要搭载高性能的处理器，飞行器的飞行负载往往亦随之增加。较大的负载量和较低的数据更新频率，使飞行器的机动性受到限制，飞行器在快速跟踪路径时会存在较大的误差。由于上述的问题，在没有精确的定位数据和较高的数据反馈速率的情况下，要深入研究飞行器的运动控制，路径规划，团队合作等问题时，会显得十分困难。

本设计希望能够实现飞行器在室内环境下的高机动性飞行，构建一个相对理想的飞行机器人室内实验平台。通过对各种室内定位系统的调查分析，最终确定参考光学运动捕捉系统的原理，构建了由双目红外相机与主动标记物构成的室内定位系统，该系统有定位精度高，更新速率高，系统鲁棒性强等优点。

对于四轴飞行器，本文根据所导出的飞行器的动力学模型，设计并实现了用于小型四轴飞行器的控制器与状态估计器，与室内定位系统组成了一个完整的室内飞行机器人实验平台。在设计中，首先根据真实飞行器模型参数，通过MATLAB[®]软件进行仿真，然后在实验中验证结果。实验结果显示，飞行器的控制器与状态估计器性能良好。同时，也证明了本设计中的室内定位系统具有较高的定位精度，与较好的噪声特性。

在构建了飞行器及其定位系统后，为实现飞行器的自主飞行，本文还对飞行器的路径规划问题进行了研究，给出了飞行器生成最优路径两种方法，实验证明对所生成的路径，飞行器跟踪误差小，并且能够实现较高速的路径跟踪。

关键词： 室内导航 四轴飞行器 控制 路径规划 状态估计

Abstract

In recent years, thanks to the developments of the MEMS sensor and the micro processor, quadrotor helicopter had caught the attention of many research institutions or companies, however, very few of them can fly autonomously. While in outdoor environment some quadrotor helicopter can achieve autonomous flying ability relying on GPS data, but the positioning precision and update rate of GPS base system is low, therefore, it is difficult to achieve precise trajectory tracking and maneuverable flight. In indoor environment, due to the high attenuation of GPS signals, the position and velocity of the aerial vehicle often need to estimate by some advance algorithms cooperates with the vehicle-mounted cameras. Because of the heavy computational burden, the update rate of such positioning system is low and the vehicle also need a more powerful processor which leads to increase the payload. For above reasons, the maneuverability and the tracking precision of the vehicle is limited. These problems make it difficult for the further studying of the motion control, path planning and team cooperation etc. of the aerial vehicle.

In this thesis I have developed a aerial robot experimental platform including an indoor positioning system and a quadrotor helicopter which can achieve high performance. Through investigation and analysis the performances of various kinds of indoor positioning systems, an optical motion capture system like system that can navigate my quadrotor in indoor environment had been presented. The indoor positioning system consists of two infrared cameras, a desktop computer and a group of active markers. During the experiment it reported that the indoor positioning system have high precision and high robustness.

During the discussion, a dynamic model of the quadrotor helicopter had been derived, with the derived model, an attitude controller, a position controller and a state estimator had been designed. Some MATLAB[®] simulations had been carried out, base on the simulated results the algorithms have been applied in real tests. The tested result showed that the designed controller and state estimator worked well, at the same time it showed that the indoor positioning system has high precision and

the noise characteristic is acceptable.

At the latter part of the thesis, in order to achieve the autonomous flying ability of the quadrotor, the path planning problem of the quadrotor had been studied. It gave two kinds of methods to generate optimum trajectories. The experiment results reported that the quadrotor can achieve fast tracking while maintaining the tracking error small.

Keywords: indoor positioning quadrotor control path planning state estimation

目 录

摘要	i
Abstract	iii
目录	v
符号和缩写列表	xv
第一章 绪论	1
1.1 相关发展和研究	1
1.2 设计动机	4
第二章 硬件结构	7
2.1 四轴飞行器	7
2.1.1 机架	8
2.1.2 飞行控制器	8
2.1.3 传感器	8
2.1.4 动力系统	9
2.1.5 飞行控制软件	9
2.2 室内定位系统	11
2.2.1 红外相机	11
2.2.2 计算机软件	11
2.2.3 标记物	11
2.2.4 摄像机布局	12
2.2.5 数据延时	12
第三章 动力学模型与控制	15
3.1 模型建立	15
3.1.1 电机模型	15
3.1.2 四轴飞行器的运动方程	16
3.2 控制律	19
3.2.1 姿态控制	19
3.2.2 位移控制	21
3.3 仿真与实验结果	22

3.3.1 MATLAB [®] 仿真	22
3.3.2 实际测试	26
第四章 状态估计	35
4.1 姿态估计	36
4.1.1 状态转移方程	36
4.1.2 测量方程	37
4.1.3 次优KF	37
4.2 位移估计	40
4.2.1 状态转移方程	40
4.2.2 测量方程	41
4.3 测量延迟	41
4.4 仿真与实验结果	44
4.4.1 MATLAB [®] 仿真	44
4.4.2 实际测试	48
第五章 室内定位系统	53
5.1 摄像机校正	53
5.1.1 内参数校正	53
5.1.2 齐次变换	54
5.1.3 确定摄像机坐标	57
5.2 标记点提取	59
5.3 确定空间坐标	61
5.3.1 立体匹配	62
5.3.2 确定空间坐标	63
5.4 确定姿态	64
5.5 测试结果	65
5.5.1 校正结果	65
5.5.2 位移测量精度	66
5.5.3 摄像机覆盖区域	66
第六章 路径规划	69
6.1 欧拉-拉格朗日方程	69
6.2 分段多项式插值	70

6.3 二次规划 (QP)	71
6.4 路径生成结果	73
第七章 结束语	77
7.1 总结	77
7.2 展望	78
致谢	81
附录 A 欧拉角与旋转矩阵	83
A.1 欧拉角	83
A.2 旋转矩阵	83
附录 B 欧拉角微分与机体角速度的关系	87
附录 C 由加速度计数据计算欧拉角	89
参考文献	91

表 格

3.1 飞行器物理参数	26
4.1 传感器噪声特性	44
5.1 定位系统 x 轴测量结果	66
5.2 定位系统 y 轴测量结果	66
5.3 定位系统 z 轴测量结果	66

插 图

1.1 (a)AR.Drone; (b)DJI Phantom;	2
1.2 (a)宾夕法尼亚大学GRASP实验室; (b)苏黎世联邦理工学院Flying Machine Arena;	3
1.3 Hummingbird四轴飞行器	4
2.1 本设计中的飞行器	7
2.2 Quanton 飞控板	8
2.3 (a)电机;(b)电子调速器;(c)螺旋桨;(d)电池;	10
2.4 飞行控制程序开发环境	10
2.5 JHSM36BF-R近红外工业相机	11
2.6 主动标记	12
2.7 摄像机布局	12
2.8 数据延时	13
3.1 电机阶跃响应	16
3.2 机体坐标系与参考坐标系	16
3.3 飞行控制系统结构	19
3.4 姿态控制器结构	20
3.5 位移控制器结构	21
3.6 横滚角控制仿真, $k_p^\Omega = 30.2, k_p^\omega = 72.3, k_d^\omega = 4.43$	23
3.7 航偏角控制仿真, $k_p^\Omega = 22.3, k_p^\omega = 63.5, k_d^\omega = 3.32$	23
3.8 位移控制x轴仿真, $k_p^r = 25.2, k_i^r = 5.23, k_d^r = 8.58$	24
3.9 位移控制z轴仿真, $k_p^r = 28.2, k_i^r = 6.73, k_d^r = 9.33$	25
3.10 跟踪曲线轨迹仿真	25
3.11 仿真中跟踪曲线轨迹时的误差, 误差平方和 $x = 0.0026\text{m}, y = 0.0336\text{m}, z = 0.00\text{m}$	26
3.12 横滚角控制实际响应, $k_p^\Omega = 14.28, k_p^\omega = 65.4, k_d^\omega = 4.82$	27
3.13 俯仰角控制实际响应, $k_p^\Omega = 14.28, k_p^\omega = 65.4, k_d^\omega = 4.82$	27
3.14 航偏角控制实际响应, $k_p^\Omega = 18.74, k_p^\omega = 76.5, k_d^\omega = 3.72$	28

3.15 位移控制 x 轴实际响应, $k_p^r = 15.3, k_i^r = 8.43, k_d^r = 10.2$	29
3.16 位移控制 y 轴实际响应, $k_p^r = 15.3, k_i^r = 8.43, k_d^r = 10.2$	29
3.17 位移控制 z 轴实际响应, $k_p^r = 18.4, k_i^r = 10.4, k_d^r = 15.3$	30
3.18 悬停时的位置误差, $\sigma_x = 0.0025\text{m}, \sigma_y = 0.0062\text{m}, \sigma_z = 0.0024\text{m}$	30
3.19 跟踪直线轨迹时 x 轴响应, 误差平方和= 0.0927m	31
3.20 跟踪直线轨迹时 y 轴响应, 误差平方和= 0.0997m	31
3.21 跟踪直线轨迹时 z 轴响应, 误差平方和= 0.0651m	32
3.22 跟踪曲线轨迹实际结果.....	32
3.23 实际跟踪曲线轨迹时的误差, 误差平方和 $x = 0.0916\text{m}, y = 0.0878\text{m}, z = 0.1054\text{m}$	33
3.24 高速直线跟踪.....	33
 4.1 无测量延迟时的速度估计误差	42
4.2 有测量延迟时的速度估计误差	42
4.3 补偿测量延迟后的速度估计误差	43
4.4 横滚角估计仿真	45
4.5 仿真中横滚角估计误差	45
4.6 仿真中陀螺仪零偏估计	46
4.7 位移估计 x 轴仿真	46
4.8 速度估计 x 轴仿真	47
4.9 仿真中 x 轴位置估计误差	47
4.10 仿真中 x 轴加速度零偏	48
4.11 横滚角的真实估计效果	49
4.12 俯仰角的真实估计效果	49
4.13 航偏角的真实估计效果	50
4.14 x 轴真实估计效果	50
4.15 y 轴真实估计效果	51
4.16 z 轴真实估计效果	51
4.17 三轴速度真实估计效果	52
 5.1 摄像机1图像	55
5.2 地面投影的坐标	56

5.3 (a)图像1变换前;(b)图像1变换后;(c)图像2变换前;(d)图像2变换后	56
5.4 图像1, 图像2融合	57
5.5 异面直线	57
5.6 计算摄像机坐标	59
5.7 标记点提取结果:(a)原图像;(b)二值化图像;	61
5.8 确定标记物空间坐标	62
5.9 极线约束	62
5.10 机体上的标记	63
5.11 由标记物坐标计算飞行器位移	64
5.12 摄像机覆盖空间区域	67
6.1 分段多项式插值	70
6.2 使用分段多项式插值生成的路径,(a)三维视图;(b) xy 平面轨迹	74
6.3 (a) x 轴数据;(b) y 轴数据;(c) z 轴数据;	74
6.4 使用QP生成的路径(未加入不等式约束),(a)三维视图;(b) xy 平面轨迹	75
6.5 使用QP生成的路径(加入不等式约束),(a)三维视图;(b) xy 平面轨迹	75
7.1 飞行器进行演奏	78
A.1 欧拉角	83
A.2 二维旋转	83

符号和缩写列表

符号列表

Γ_w	参考坐标系
Γ_b	机体坐标系
Ω	飞行器姿态欧拉角
${}^b R_a$	从坐标系 a 到坐标 b 的旋转矩阵
ω	飞行器角速度
r	飞行器在参考坐标系中的位移
I	飞行器惯量矩阵
m	飞行器质量
g	重力加速度
\mathcal{F}	螺旋桨产生的升力和
\mathcal{M}	飞行器转动力矩
L	飞行器螺旋桨中心到机体坐标轴的距离
k_F	电调控制信号与螺旋桨产生升力的比例系数
k_M	电调控制信号与电机产生反扭矩的比例系数

缩写列表

IMU(Inertial Measurement Unit)	惯性测量单元
MEMS(Micro-Electro-Mechanical System)	微机电系统
FPU(Float Point Unit)	浮点运算单元
SLAM(Simultaneous Localization And Mapping)	同步定位与建图
EKF(Extended Kalman Filter)	扩展卡尔曼滤波器
KF(Kalman Filter)	卡尔曼滤波器
PID(Proportion Integration Differentiation)	比例-积分-微分
LQ(Linear Quadratic)	线性二次型
QP(Quadratic Programming)	二次规划

第一章 绪论

相对于其它飞行器，多轴飞行器具有结构简洁，成本低廉，机动性强的优点，并且容易小型化，适合于室内飞行。

依靠地面移动的传统轮式机器人，相对于飞行机器人有：结构稳定，负载能力强，续航时间长等优点，但是其缺点也非常明显，对于复杂的地形环境，轮式机器人往往难以通过，或者容易出现故障，如在建筑物中的楼梯，灾区中的废墟，水塘等环境中，往往难以顺利到达其作业地点，此外轮式机器人通常需要依靠较为复杂的机械结构去适应各种地形。

固定翼飞行器，则具有续航长，行程远的优点，适合完成一些远程，并且定位精度要求较低的一些任务。其缺点是：飞行器必须一直移动，不能进行悬停作业，对于一些需要固定点进行操作的任务不能实现。

直升飞机和多轴飞行器则具有轮式机器人与固定翼飞行器所没有的优点，能够穿越复杂的地形，并且能够悬停作业，但是直升飞机为了实现六自由度的飞行，需要十字盘等复杂精密的传动机构，其出现故障几率相对较高，并且不易于维护。而多轴飞行器或者是四轴飞行器其机械结构非常简洁，四个螺旋桨直接由电机驱动，中间不需要经过任何复杂的传动机构即可实现了六自由度的控制，制作成本低，维护方便。并且由于结构简单，螺旋桨的响应非常迅速，机动性强，可以实现在障碍物间快速的穿越，或者跟踪复杂的路径。

近年来，多轴飞行器以其优点，引起了许多商业或研究机构的关注，如今，多轴飞行器已经广泛的用于航拍，交通管制，电力线巡逻，农业管理等领域。

然而，目前的投入使用的多轴飞行器大多数需要有专业人员来操作，并不能够实现完全自主飞行。而对于能实现完全自主飞行的空中机器人，在将来将具有广阔的应用场景，包括协助建筑物的架设，路桥的修建，灾区搜救，物资运输，数据采集等。

1.1 相关发展和研究

四轴飞行器最早出现于20世纪90年代初^[1]，然而当时飞机只能飞离地面不过10米，而且飞行时间只有1分钟。在往后的近百年时间里，四轴飞行器由于其较大的体积，过短的续航能力，以及不稳定的动力学结构，并没有引起研究者

过大的关注。直到21世纪初，由于微型处理器的性能提高，以及小体积，高精度的MEMS^[2]传感器出现，使多轴飞行器的小型化成为可能。并且由于其结构简洁，制造成本低，小型的多轴飞行器开始引起研究者的关注。另一方面，由于开源飞控的兴起，四轴飞行器逐渐在航模圈中流行起来，同时由于其极轻的机身发生故障时也不容易伤害到使用者，一部分微型四轴飞行器以玩具的形式出现在公众的视野。

2010年，法国Parrot[®]公司发布了其第一款具备航拍功能的小型四轴飞行器AR.Drone^[3]（图1.1），飞行器利用智能手机进行控制。2013年初，中国大疆创新(DJI)公司^[4]发布了其第一款小型四轴航拍机Phantom^[5]（图1.1）后不久，四轴航拍飞行器迅速风靡全国甚至全球。小型四轴航拍机由于其便携性，灵活的飞行能力，以其相对于专业航拍设备较低廉的价格，受到广大消费者的欢迎。到目前，一些消费级别的四轴飞行器已经搭载了高性能的处理器和高分辨率的图像采集模块，并通过复杂的图像处理算法实现飞行器在丢失GPS信号的室内环境中自主悬停，或者自主躲避障碍物。



图 1.1 (a)AR.Drone; (b)DJI Phantom;

在近十年内，小型四轴飞行器，由于其极小的体积以其极强的机动性，引起了不少研究者与相关企业的关注。四轴飞行器由于其结构简洁，动力学模型容易构建，并且能够实现悬停，许多研究者以四轴飞行器为平台，来研究飞行机器人在缺乏GPS信号的室内环境中控制，感知和规划问题，包括飞行机器人的室内定位，姿态和位移控制，路径规划以及同步建图与定位(SLAM)^[6]等。

2012年初，美国宾夕法尼亚大学的机器人学家Vijay Kumar，在TED上发表了名为“Robots that fly and cooperate”^[7]的演讲，向世界展示了实验室中具有

惊人机动性的四轴飞行器，具有团队合作能力飞行机器人，以及目前处于研究阶段的飞行机器人SLAM技术。在宾夕法尼亚大学的GRASP实验室（图1.2）中配备了Vicon Motion Systems^[8]公司一套具有20个高速红外相机的光学运动捕捉系统，用于飞行机器人的室内空间定位，系统能以200Hz以上的频率运行，能实时反馈多个飞行机器人的三维位置信息，三维姿态信息，是研究飞行机器人运动控制与规划问题的一个非常理想的试验平台。此外，GRASP实验室成员S.Shaojie et al.对利用机载摄像机进行飞行器状态估计和控制的问题进行了研究^[9]，其中四轴飞行器上安装了两个小型摄像机，一个以较高频率运行，配合IMU完成SLAM任务，另一个摄像头以低频率运行，负责参数校正。S.Tang et al.研究了四轴飞行器在软线悬挂负载状态下的路径规划问题^[10]，将负载物体的运动轨迹考虑到问题当中。L.Giuseppe et al.^[11]研究了飞行机器人团队合作SLAM的问题。

2013年6月，苏黎世联邦理工学院(ETH Zürich)教授Raffaello D'Andrea以及其团队在TED大会上进行了名为“*The astounding athletic power of quadcopters*”^[12]的演讲，向人们展示了四轴飞行器强大的运动能力，在演讲上展示了飞行机器人极强的鲁棒性以及四轴飞行器控制的倒立摆^[13]，飞行机器人团队合作表演^[14]等成果。观众们纷纷被其四轴飞行机器人灵活，高机动的运动性能所震撼，视频不久在互联网上被迅速传播，引起了许多爱好者和相关研究人员的关注。该演讲中的飞行机器人控制技术主要在苏黎世联邦理工学院的Flying Machine Arena（图1.2）进行研究，类似宾夕法尼亚大学的GRASP实验室，该实验室配备了一套具有8个高速红外相机VICON光学运动捕捉系统，图像运算频率能达200Hz以上，飞行器采用Ascending Technologies^[15]公司的Hummingbird（图1.3）小型四轴飞行器。

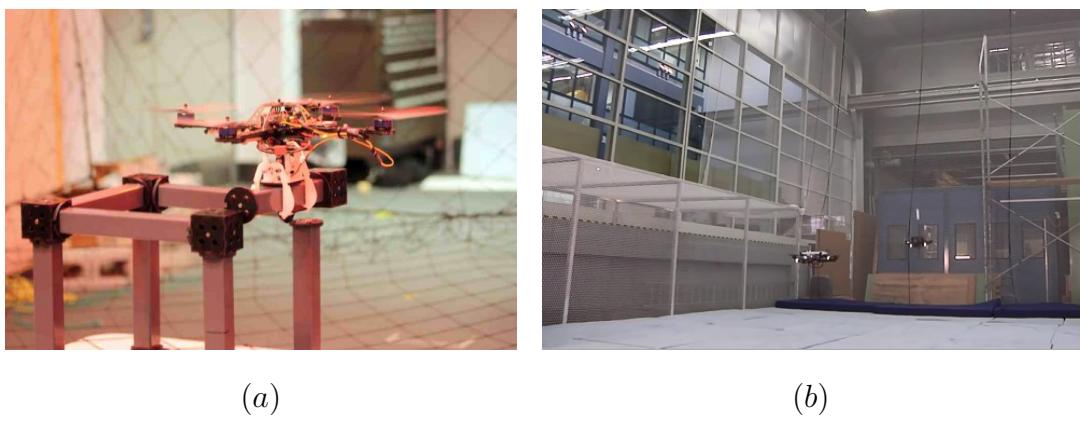


图 1.2 (a)宾夕法尼亚大学GRASP实验室; (b)苏黎世联邦理工学院Flying Machine Arena;



图 1.3 Hummingbird四轴飞行器

1.2 设计动机

四轴飞行器要在室内自主飞行，其中一个主要问题是进行定位。目前，在室内环境中一方面有基于无线电的定位方法，如RFID，iBeacon，无线电指纹等；也有基于机械波的定位方法，如超声波，麦克风阵列等，而这类的定位方法往往精度较低，并且延迟大，更新速率低，不适合做为飞行器的位置信号反馈。另一方面，也有基于图像处理的室内定位方法，包括光流法，基于双目摄像头或单目摄像头的SLAM算法等，然而，对于光流法，由于其检测的是飞行器的速度，长时间飞行后会产生较大的累积误差。对于SLAM一类的算法，其算法复杂，运算量大，难以获得高速率，低延迟的数据反馈，一般用于室内飞行器的低动态飞行运动控制。

本设计旨在建立一个室内飞行机器人实验平台，能够完成室内飞行机器人的高动态运动控制，精确的运动轨迹跟踪等任务。为了获得高速率，低延迟，低噪声的位置信息反馈，本设计采用类似光学运动捕捉系统的结构，利用外部相机对飞行器进行定位，给飞行器提供高质量的空间位置与姿态信息反馈。在解决了飞行器室内控制与定位问题后，本文探讨了飞行器的路径规划问题，利用计算机为飞行器生成最优的路径，实现飞行路径的精确跟踪，并且提供了一些演示场景。

本文的内容分布如下：第二章介绍了四轴飞行器与室内定位系统的硬件配置，整个系统的结构，以及一些硬件性能参数。第三章推导了四轴飞行器的动力学模型，姿态和位置控制的控制律，讨论了四轴飞行器在大角度下的控制的方法，最后，在MATLAB上进行了仿真，并进行了实际测试。第四章探讨了飞行器的状态估计问题，设计了次优滤波器估计四轴飞行器的姿态与位置信息，并对测量延迟

问题进行了讨论，最后给出了仿真效果与实际效果。第五章介绍了由红外相机构建的飞行器室内定位系统，讨论了室内定位的原理，并测试了实际性能。第六章探讨了飞行器的三维路径规划问题，给出了生成最优（最小抖动）飞行路径的几种方法，并给出在仿真环境中和实际环境中的跟踪效果。第七章为全文的总结，探讨该设计中目前存在的一些不足，与一些发展潜能。

第二章 硬件结构

2.1 四轴飞行器

四轴飞行器的物理结构，动力系统与飞行器的运动性能及稳定性有着密切的关系。四轴飞行器在飞行过程中，飞行器主要受到两个力的影响，一个是与飞行器机体平面垂直的升力，一个是重力。飞行器在三维空间中线运动所需要的加速度来自升力与重力的合力，飞行器的旋转运动所需的力矩则来自飞行器每个螺旋桨产生的升力差。一般来说飞行器尺寸越大，其质量与转动惯量越大，要使飞行器获得强的机动性，飞行器的必须要有一套高输出功率的动力系统，而高功率的动力系统又往往体积大，质量大。因此，一般来说尺寸大的飞行器的机动性往往不如小尺寸的飞行器。文献^[16]中指出，随着多轴飞行器的尺寸减少，飞行器能产生的最大角加速度越大，能产生的最大线加速度也会越大或至少不变。



图 2.1 本设计中的飞行器

本设计的飞行器主要在室内进行飞行，并且需要较强的机动性，因此飞行器的体积需要尽量的小。但是体积过小的飞行器其负载能力非常弱，往往只能满足单纯的飞行功能，难以继续在飞行器上进行上层应用的开发。考虑到飞行器的机动性，安全性，负载能力等各方面的要求，本设计的飞行器（图2.1）具体硬件配置如下：

2.1.1 机架

综合飞行器体积与负载能力要求的考虑，本设计采用了轴距为250mm的X型，高强度塑料四轴飞行器机架。

2.1.2 飞行控制器

本设计中机载的飞行控制器需要完成传感器数据采集，与上位机通信，飞行状态估计，飞行控制等几个主要任务。飞行控制系统一般由主控制器（一般是高性能的单片机）与一些传感器以及电源管理电路组成，目前许多开源飞控采用STM32F103系列的单片机作为主控制器，STM32F103系列单片机具有非常高的性价比，适合用于一些相对简单的飞行控制处理，但是对于实验研究中用到的一些更复杂的算法，或者还没经过运算速度优化的算法，其运算能力往往难以满足要求。对于本设计中的应用，带有浮点运算单元(FPU)的STM32F4系列的单片机是一个较好的选择。经过相关资料的搜索，发现目前有一些开源飞控板也采用了STM32F4系列的单片机，包括Pixhawk^[17]，Taulab^[18]的Quanton等，经过分析比较，本设计直接在Quanton飞控(图2.2)的硬件电路基础上进行开发，这样可以省去了许多调试底层硬件的麻烦，把更多的时间用于算法的研究上。飞控板上搭载了一片意法半导体(ST)公司的STM32F405RGT6 32位单片机，ARM Cortex-M4内核，具有浮点运算单元(FPU)，主频168MHz，具有192kB的RAM，1MB的Flash储存空间。

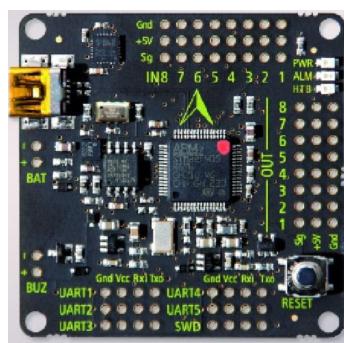


图 2.2 Quanton飞控板

2.1.3 传感器

飞控板上配备了10轴传感器，由以下几个传感器组成：

- Invensense MPU6000:

6轴IMU，包含3轴加速度计与3轴陀螺仪，内置低通滤波器，陀螺仪最高输出频率：8kHz（未开启低通滤波），1kHz（开启低通滤波），加速度数据最高输出频率：1kHz，支持IIC或SPI总线协议通信。本设计中，采用SPI总线进行通信，开启低通滤波，截至频率设置为42Hz，采样率1kHz。

- Honeywell HMC5883:

三轴磁阻传感器，最大输出速率160Hz，罗盘精度： $1^\circ \sim 2^\circ$ ，由于室内环境中电子罗盘十分容易受到干扰，并且每次飞行器前需要进行复杂的校正过程，本设计中并没有应用电子罗盘。

- Measurement Specialties MS5611:

气压计，内置数字温度计，支持IIC或SPI总线协议通信，最大气压分辨率：0.012mba，高度分辨率： $> 10\text{cm}$ ，设置为最高分辨率时转换时间为8 ~ 9ms。由于气压计获得的高度数据噪声大，精度相对较低，并不适用于室内的精确定位，本设计中并没有应用气压计。

2.1.4 动力系统

为了获得较高的输出功率，本设计采用无刷电机驱动螺旋桨。飞行器的动力系统包括以下部件（图2.3）：

- 电机：采用T-Motor的MN1806电机，最大持续输出功率150W，最大持续工作电流14A，重量18g。
- 电子调速器：采用DALRC BL12A电子调速器，最大持续工作电流12A。
- 螺旋桨：采用半径60mm，螺距为45mm的塑料螺旋桨。
- 电池：采用11.1V，1300mAh的锂聚合物动力电池，放电倍率为25C。

经实验测试，四个螺旋桨产生的最大升力约为13.8N。

2.1.5 飞行控制软件

本设计的飞行控制软件由本人自主开发，为兼顾飞行控制程序的实时性，可靠性与可扩展性，本设计采用Micrium[®]公司^[19]的可剥夺式实时操作系统μC/OS – III[®]。操作系统上主要运行了几大任务：遥控器接收解码，上位机通信，传感器采集，状态估计，姿态控制，位置控制和飞行状态监测。其中状态



图 2.3 (a)电机;(b)电子调速器;(c)螺旋桨;(d)电池;

估计任务的运算频率为1000Hz，受到电子调速器的限制姿态控制任务运行频率为400Hz，位置控制任务运行频率为200Hz。所有程序在Keil μ Vision[®]4环境下进行开发。

The screenshot shows the CodeScape IDE interface. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, Help. Below the menu is a toolbar with icons for file operations like Open, Save, Build, Run, and Debug. The left sidebar displays the 'Project' tree, which contains the 'Quad FC' project, specifically the 'APP' folder, and lists various source files such as includes.h, app_cfg.h, rpu_cfg.h, os_cfg.h, os_cfg_app.h, os_app_hooks.h, os_app_hooks.c, GlobalVariable.h, GlobalVariable.c, app.c, StateControlTask.c, RTCTasks.c, AttitudeTask.c, ControlTask.c, SerPrintTask.c, ESCCalibrateTask.c, AccCalibrateTask.c, CommTask.c, PosControlTask.c, NavigateTasks.c, and BSP. The right side of the interface is a large code editor window titled 'ControlTask.c'. The code itself is written in C and defines a task function for controlling the quadcopter's attitude and position. It includes imports for Constraint.h, math.h, and various internal module headers. The code uses constraints to limit the throttle value, handles sensor fusion for roll and pitch, and performs attitude control using Euler angles and rate controllers. It also manages motor PWM settings and applies minimum thrust constraints.

```
49     throttle = Constraint_f(g_RCThrottle + g_PosThrottle, 0, 0.7f);
50 
51     else
52         throttle = Constraint_f(g_RCThrottle, 0, 0.7f);
53 
54     /* Start control */
55     if (g_FlagArmed == RESET) {
56 
57         if (g_FlagTakeOff == SET) {
58             /* tilt compensation */
59             cos_a_bg = arm_cos_f32(euler_att.pitch) * arm_cos_f32(euler_att.roll);
60 
61             if (cos_a_bg > 0.5f)/* 60 degree */
62                 throttle = throttle / cos_a_bg;
63 
64             CTRL_SetTargetRate(target_gyro, euler_att, itarget_rate); /* angle control */
65             CTRL_RateController(target_rate, gyro_rate, &control_data, dt, SET); /* rate control */
66             CTRL_SetThrottle(throttle, &control_data);
67         }
68         /* still not take off */
69     } else {
70         CTRL_RateController(target_rate, gyro_rate, &control_data, dt, RESET);
71         CTRL_SetThrottle(throttle, &control_data);
72 
73         Motor_SetPWM(&control_data, &motor_data);
74     }
75     else
76         Motor_MinThrust(&motor_data);
77 
78     /* End control */
79 
80 //RCTC_Next_Sig();
81 }
```

图 2.4 飞行控制程序开发环境

2.2 室内定位系统

按照本设计的应用要求，需要一套高精度，低噪声，高速率的室内定位系统，符合这几个要求的定位方法目前一般只有用外部摄像机来进行定位，对于摄像机的选择，本设计采用了红外摄像机，配合主动标记的方法。因为，对于一般的RGB摄像机或者黑白摄像机，考虑到室内环境的复杂性，往往会给定位带来很多的干扰，这样算法的复杂度往往较高，并且鲁棒性低。而使用红外摄像机，则能把非红外波段的自然光或灯光，利用光学的方法滤除，使干扰大大降低，算法的复杂度降低，系统的鲁棒性也得到很大提高，使高速率，高质量的信息反馈成为可能。

2.2.1 红外相机

本设计采用了两个京航JHSM36BF-R近红外工业相机(图2.5)，有效像素为36万，最高输出帧率为62fps，响应波段范围为800nm ~ 1000nm。



图 2.5 JHSM36BF-R近红外工业相机

2.2.2 计算机软件

室内定位系统的图像处理过程在计算机上进行处理，本设计结合了开源计算机视觉库OpenCV^[20]和MATLAB[®]来进行了图像处理与摄像机校正。开发环境为VisualStudio[®]2013。计算机获得飞行器的位置信息与姿态信息后通过蓝牙进行数据反馈，反馈频率为40Hz。

2.2.3 标记物

为了让红外相机能够检测到飞行器的状态，飞行器顶部安装了3颗波长为820nm的红外LED(图2.6)。

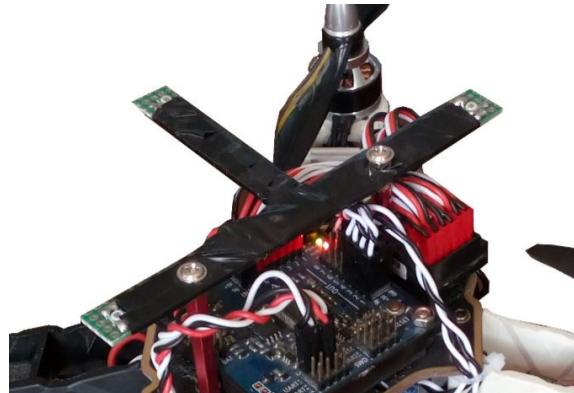


图 2.6 主动标记

2.2.4 摄像机布局

摄像机在实验室中安装如（图2.7）。



图 2.7 摄像机布局

2.2.5 数据延时

（图2.8）显示了定位系统中的数据延时，飞行器的姿态和位移反馈数据的延迟大约为40ms，其中延迟较大的部分是摄像机，和蓝牙。摄像机延迟的部分主要由感光器的曝光过程和摄像机内部数据处理造成，蓝牙的延迟主要来自于数据的空中传输时间以及数据的调制和解调过程。

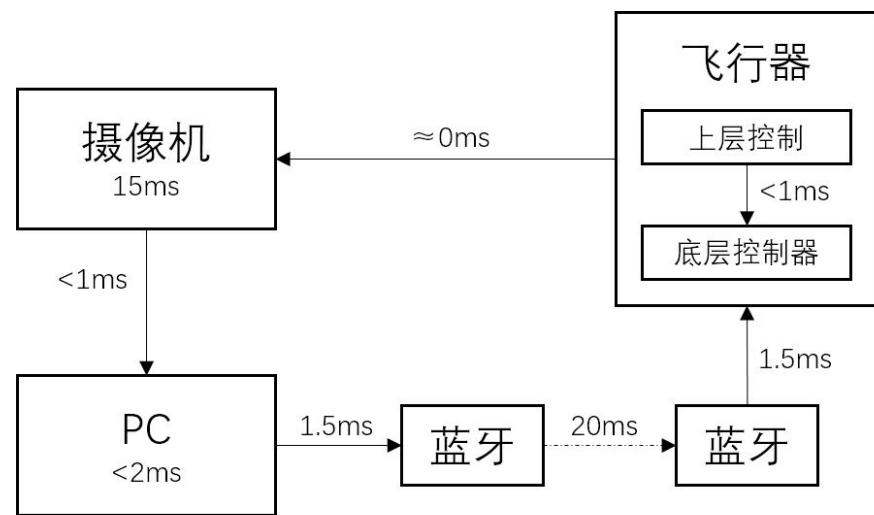


图 2.8 数据延时

第三章 动力学模型与控制

目前对四轴飞行器的模型与控制问题有较多相关的研究，文献^[21]中，研究了四轴飞行器的姿态控制，以及轨迹跟踪控制问题，其中采用了根轨迹法设计了控制器，考虑了电机模型，利用角加速度反馈来提高姿态控制器的带宽。文献^[22]中研究了采用混合控制系统令飞行器做出高机动性的动作的方法，文中考虑了二维平面中的飞行器模型，设计了飞行器进行空翻动作的策略。文献^[23]对四轴飞行器的控制进行了十分详细的讨论，其中尝试了各种控制方法，包括PID，LQ控制，反步法控制和滑模控制。文献^[24]中，研究了飞行器在不同负载的情况下进行物理参数实时估计的方法，实现了飞行器在进行物体运输时的自适应控制。本章推导出四轴飞行器的动力学模型，在计算机中进行了仿真，并根据飞行器的动力学模型得到飞行器姿态，位移控制的控制律，并展示了仿真以及实际的测试结果。

3.1 模型建立

3.1.1 电机模型

实验中发现电机带动螺旋桨产生的升力与输入到电子调速器的控制信号 u 接近线性关系，电机输入信号与输出升力可以用看作为一个一阶惯性环节，令系统开环传递函数为：

$$G(s) = \frac{k_F}{Ts + 1}$$

利用MTALAB系统辨识工具箱进行系统辨识，得到 $k_F = 3.174, T = 0.0835s$ ，阶跃响应如（图3.1）。

由于相对于飞行控制系统，电机的时常数较小，这里将简化电机模型，令每个螺旋桨产生的升力 F_i 与电子调速器输入的信号 u_i 的关系为：

$$F_i = k_F u_i \quad (3.1)$$

此外，螺旋桨加速时产生的反扭矩 M_i 也与输入信号 u_i 呈线性关系：

$$M_i = k_M u_i \quad (3.2)$$

通过实验测试得到 $k_M = 1.063$ 。

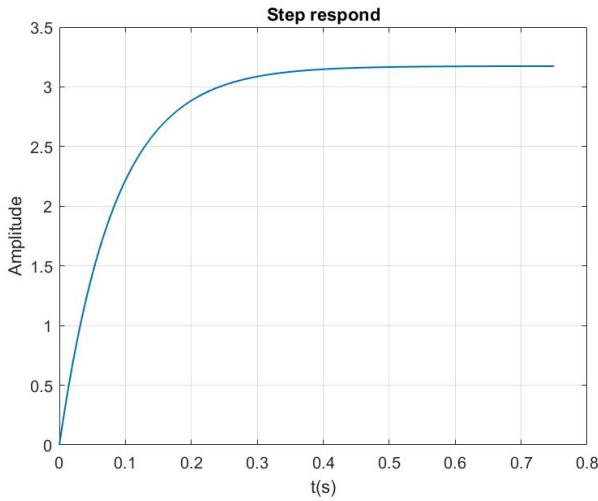


图 3.1 电机阶跃响应

3.1.2 四轴飞行器的运动方程

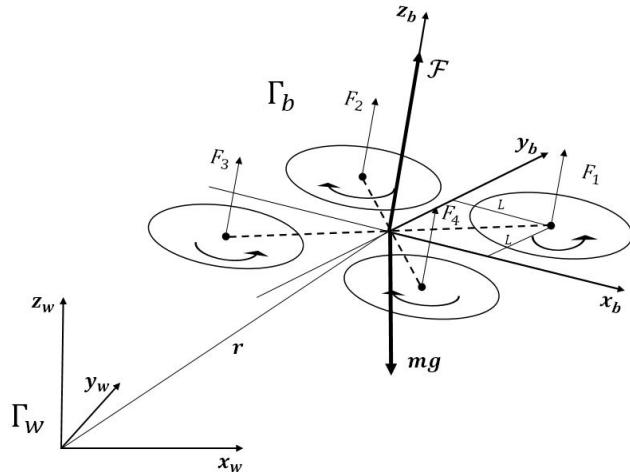


图 3.2 机体坐标系与参考坐标系

飞行器机体坐标系 Γ_b 与参考坐标系 Γ_w 如 (图3.2) 定义。四轴飞行器具有六个自由度，包括飞行器机体坐标 Γ_b 相对于参考坐标 Γ_w 的旋转与位移。本文以 $Z - Y - X$ 次序的欧拉角 Ω (附录A) 来描述飞行器的姿态 (即旋转)。

$$\Omega = \begin{pmatrix} \varphi \\ \theta \\ \psi \end{pmatrix}$$

φ, θ, ψ 分别表示飞行器的横滚角、俯仰角和航偏角。

机体坐标到参考坐标的旋转矩阵为：

$${}^w\mathbf{R}_b = \begin{pmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - c\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{pmatrix} \quad (3.3)$$

其中s和c分别表示sin和cos。

对于 Γ_b 中的任意向量 \mathbf{v}_b , 与其在 Γ_w 的投影为 \mathbf{v}_w 有如下关系:

$$\mathbf{v}_w = {}^w\mathbf{R}_b \mathbf{v}_b \quad (3.4)$$

飞行器的位移以 \mathbf{r} 来表示:

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}$$

r_x, r_y, r_z 为飞行器重心在 Γ_w 中的对应的三个坐标坐标, 飞行器在 Γ_w 中的速度与加速度则分别为 $\dot{\mathbf{r}}$ 和 $\ddot{\mathbf{r}}$ 。

牛顿方程

飞行器位移的运动方程可以通过牛顿方程来描述。忽略空气动力学因素的影响, 可以认为作用在飞行器上的力只有两个, 一个是四个螺旋桨提供的升力的合力 \mathcal{F} , 并且该力永远指向 z_b 方向。另一个是机体受到的重力 $m\mathbf{a}$, 方向为 $-z_w$ 。利用旋转矩阵(式3.3), 关系(式3.4), 以及牛顿定律 $\mathbf{F} = m\mathbf{a}$, 可以获得飞行器在 Γ_w 中的加速度 $\ddot{\mathbf{r}}$:

$$m\ddot{\mathbf{r}} = {}^w\mathbf{R}_b \begin{pmatrix} 0 \\ 0 \\ \mathcal{F} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (3.5)$$

进一步得:

$$\ddot{\mathbf{r}} = {}^w\mathbf{R}_b \begin{pmatrix} 0 \\ 0 \\ \mathcal{F}/m \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (3.6)$$

欧拉方程

飞行器的旋转运动由四个螺旋桨的升力差产生，提供了绕 x_b, y_b, z_b 轴的旋转力矩，飞行器的旋转运动可以用欧拉方程来描述。飞行器绕 x_b, y_b, z_b 三个轴旋转的角速度用 ω 来表示：

$$\boldsymbol{\omega} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

p, q, r 分别为绕 x_b, y_b, z_b 轴旋转的角速度。

根据欧拉方程有：

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \mathcal{M} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}$$

即：

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\mathcal{M} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}) \quad (3.7)$$

其中 \mathbf{I} 为飞行器的惯量矩。再由欧拉角的定义，可以得出欧拉角微分 $\dot{\Omega}$ 与机体角速度 $\boldsymbol{\omega}$ 的关系(附录B)：

$$\dot{\boldsymbol{\Omega}} = \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan(\theta) \sin(\varphi) & \tan(\theta) \cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sec(\theta) \sin(\varphi) & \sec(\theta) \cos(\varphi) \end{pmatrix} \boldsymbol{\omega}$$

以 \mathbf{W} 来表示该关系矩阵：

$$\mathbf{W} = \begin{pmatrix} 1 & \tan(\theta) \sin(\varphi) & \tan(\theta) \cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sec(\theta) \sin(\varphi) & \sec(\theta) \cos(\varphi) \end{pmatrix}$$

即：

$$\dot{\boldsymbol{\Omega}} = \mathbf{W}\boldsymbol{\omega} \quad (3.8)$$

利用(式3.6), (式3.7)以及(式3.8)可以建立飞行器的非线性状态方程:

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \\ \boldsymbol{\Omega} \\ \boldsymbol{\omega} \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{r}} \\ {}^w\boldsymbol{R}_b \begin{pmatrix} 0 \\ 0 \\ \mathcal{F}/m \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \\ \boldsymbol{W}\boldsymbol{\omega} \\ I^{-1}(\boldsymbol{\mathcal{M}} - \boldsymbol{\omega} \times I\boldsymbol{\omega}) \end{pmatrix} \quad (3.9)$$

3.2 控制律

四轴飞行器具有六个自由度, 四个控制量, 为一个欠驱动系。飞行器的控制主要分为两部分: 姿态控制和位移控制。其中姿态控制器作为飞行器飞行控制的底层控制器, 位移控制器根据目标位移, 速度以及加速度计算出飞行器的目标姿态, 通过改变飞行器的姿态来产生相应的线加速度, 从而实现位移控制。姿态控制器应具有较快的响应速度, 以及较强的鲁棒性, 以获得较好的位移控制效果。本设计中的飞行器姿态控制频率设置为400Hz, 位移控制频率设置为200Hz的情况下, 能达到较好的控制效果。

四轴飞行器控制系统结构如下(图3.3):

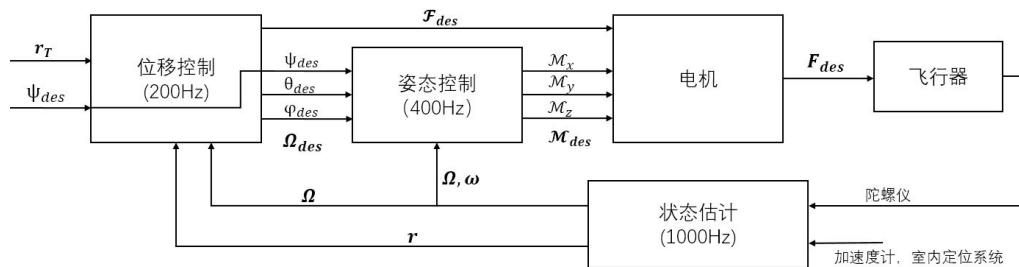


图 3.3 飞行控制系统结构

3.2.1 姿态控制

通过简单分析, 可得每个螺旋桨的升力 F_i , 反力矩 M_i 与绕机体坐标轴的力矩 $\boldsymbol{\mathcal{M}}$ 的关系如下:

$$\mathbf{M} = \begin{pmatrix} \mathcal{M}_x \\ \mathcal{M}_y \\ \mathcal{M}_z \end{pmatrix} = \begin{pmatrix} (-F_1 - F_2 + F_3 + F_4)L \\ (-F_1 + F_2 + F_3 - F_4)L \\ (-M_1 + M_2 - M_3 + M_4)\sqrt{2}L \end{pmatrix} \quad (3.10)$$

由（式3.1）和（式3.2）可以得到 F_i 和 M_i 之间的关系：

$$M_i = \frac{k_M}{k_F} F_i$$

令 $k_m = \frac{k_M}{k_F}$ 得：

$$M_i = k_m F_i \quad (3.11)$$

将（式3.11）代入（式3.10）可以得：

$$\mathbf{M} = \begin{pmatrix} -L & -L & L & L \\ -L & L & L & -L \\ -k_m\sqrt{2}L & k_m\sqrt{2}L & -k_m\sqrt{2}L & k_m\sqrt{2}L \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix} \quad (3.12)$$

令：

$$\mathbf{X} = \begin{pmatrix} -L & -L & L & L \\ -L & L & L & -L \\ -k_m\sqrt{2}L & k_m\sqrt{2}L & -k_m\sqrt{2}L & k_m\sqrt{2}L \end{pmatrix}, \mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{pmatrix}$$

飞行器的误差欧拉角 Ω_{error} 为：

$$\Omega_{error} = \Omega_{des} - \Omega \quad (3.13)$$

其中 Ω_{des} 为飞行器的目标欧拉角。

飞行器的姿态控制采用角速度PD控制，结构如（图3.4）：

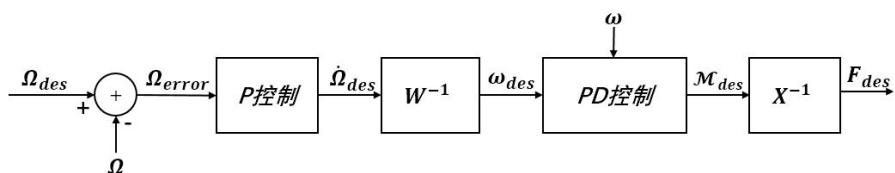


图 3.4 姿态控制器结构

姿态控制器的控制过程大概如下：首先由目标欧拉角 Ω_{des} 与估计得到的欧拉角 Ω 得到误差欧拉角 Ω_{error} ，误差欧拉角 Ω_{error} 与一个比例系数 k_p^Ω 相乘后得到目标欧拉角速度 $\dot{\Omega}_{des}$ （该系数可以在仿真或实验中调试得出），利用欧拉角微分与机体角速度的关系（式3.8）得到绕机体轴的目标角速度 ω_{des} ，再通过基于PD控制的角速度控制器输出目标扭矩 \mathcal{M}_{des} ，最后根据螺旋桨升力与机体旋转力矩的关系（式3.12）得到四个螺旋桨的目标升力 \mathbf{F}_{des} 。

由于机体的三轴角速度独立，基于PD控制的角速度控制器可以看作是三个独立的PD控制器，分别对应 Γ_b 中的 x_b, y_b, z_b 轴的角速度控制，输入 ω_{des} 与输出 \mathcal{M}_{des} 关系如下：

$$\omega_{error} = \omega_{des} - \omega$$

$$\mathcal{M}_i = k_{p,i}^\omega \omega_{error,i} + k_{d,i}^\omega \dot{\omega}_{error,i}, (i = x, y, z) \quad (3.14)$$

3.2.2 位移控制

飞行器的位移控制主要分为两步，一是位移的PID控制，二是目标加速度到目标欧拉角的转换，控制器结构如下（图3.5）：

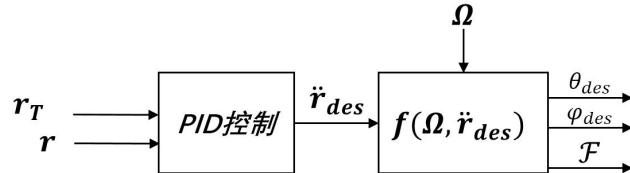


图 3.5 位移控制器结构

对于每一个 Γ_w 中每一个轴的位移，控制器应满足以下等式：

$$(\ddot{r}_T - \ddot{r}_{des}) + k_d^r (\dot{r}_T - \dot{r}) + k_p^r (r_T - r) + k_i^r \int (r_T - r) dt = 0$$

由此得出：

$$\ddot{r}_{des} = \ddot{r}_T + k_d^r (\dot{r}_T - \dot{r}) + k_p^r (r_T - r) + k_i^r \int (r_T - r) dt \quad (3.15)$$

其中 r_T 为目标位移， \ddot{r}_{des} 为飞行器要产生的加速度， \ddot{r}_T 为控制器的前馈部分。

由于三轴位移独立，对于飞行器在 Γ_w 中的 x_w, y_w, z_w 轴的位移控制，可以采用3个与（式3.15）相同的PID控制器进行控制。控制器输出飞行器的目标加速

度 \ddot{r}_{des} , 利用 (式3.6) 可以得到以下关系:

$$\begin{aligned}\ddot{r}_{des,x} &= \frac{\mathcal{F}}{m}(\cos(\psi)\sin(\theta_{des})\cos(\varphi) + \sin(\psi)\sin(\varphi)) \\ \ddot{r}_{des,y} &= \frac{\mathcal{F}}{m}(\sin(\psi)\sin(\theta)\cos(\varphi_{des}) - \cos(\psi)\sin(\varphi_{des})) \\ \ddot{r}_{des,z} &= \frac{\mathcal{F}}{m}\cos(\theta)\cos(\varphi) - g\end{aligned}\quad (3.16)$$

容易的到 \mathcal{F} 的表达式:

$$\mathcal{F} = \frac{m\ddot{r}_{des,z}}{\cos(\theta)\cos(\varphi)} + mg \quad (3.17)$$

进一步计算 θ_{des} 和 φ_{des} 。为了简化计算, 先将 \ddot{r}_{des} 绕 z_w 轴旋转 ψ , 得到 \ddot{r}'_{des} :

$$\begin{aligned}\ddot{r}'_{des,x} &= \cos(\psi)\ddot{r}_{des,x} + \sin(\psi)\ddot{r}_{des,y} \\ \ddot{r}'_{des,y} &= -\sin(\psi)\ddot{r}_{des,x} + \cos(\psi)\ddot{r}_{des,y} \\ \ddot{r}'_{des,z} &= \ddot{r}_{des,z}\end{aligned}\quad (3.18)$$

此时, 再利用 (式3.16) 可以得到:

$$\begin{aligned}\ddot{r}'_{des,x} &= \frac{\mathcal{F}}{m}\sin(\theta_{des})\cos(\varphi) \\ \ddot{r}'_{des,y} &= -\frac{\mathcal{F}}{m}\sin(\varphi_{des})\end{aligned}$$

进一步得:

$$\begin{aligned}\varphi_{des} &= \arcsin\left(-\frac{\ddot{r}'_{des,y}m}{\mathcal{F}}\right) \\ \theta_{des} &= \arcsin\left(\frac{\ddot{r}'_{des,x}m}{\mathcal{F}\cos(\varphi)}\right)\end{aligned}$$

上式中 \mathcal{F} 由 $\ddot{r}_{des,z}$ 计算得出, 为了尽量避免 x_w, y_w 轴的位移控制与 z_w 轴耦合, 将上式中的 \mathcal{F} 近似为 mg , 得:

$$\begin{aligned}\varphi_{des} &= \arcsin\left(-\frac{\ddot{r}'_{des,y}}{g}\right) \\ \theta_{des} &= \arcsin\left(\frac{\ddot{r}'_{des,x}}{g\cos(\varphi)}\right)\end{aligned}\quad (3.19)$$

至此得到位移控制器的所有输出 (式3.17) (式3.19)。

3.3 仿真与实验结果

本节给出了飞行器姿态控制器与位移控制器在MATLAB®中的仿真结果, 与在实验中的真实测试结果。

3.3.1 MATLAB®仿真

根据飞行器的状态空间方程 (式3.9), 并根据电机传递函数考虑了电机的延迟, 利用MATLAB®里面的ode45微分方程求解器进行仿真, 时间步长为0.01s。

姿态控制

采用上述的姿态控制器，1s后给定目标 $\varphi = 0.2\text{rad}$ ，2s后再给定目标 $\varphi = 0.0\text{rad}$ ，仿真中飞行器的响应如下：

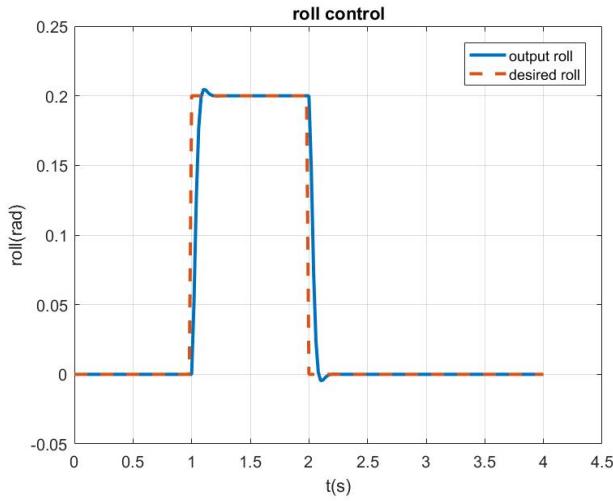


图 3.6 横滚角控制仿真, $k_p^\Omega = 30.2, k_p^\omega = 72.3, k_d^\omega = 4.43$

其中控制器的参数通过实验手工粗略整定后，利用梯度下降法，最小化跟踪误差平方的积分，得到相应控制器参数。可以看出在仿真（图3.6）中， φ 角度控制的响应较快，上升时间大约为100ms，超调量小于5%。俯仰角 θ 的响应基本与横滚角控制的响应一致。

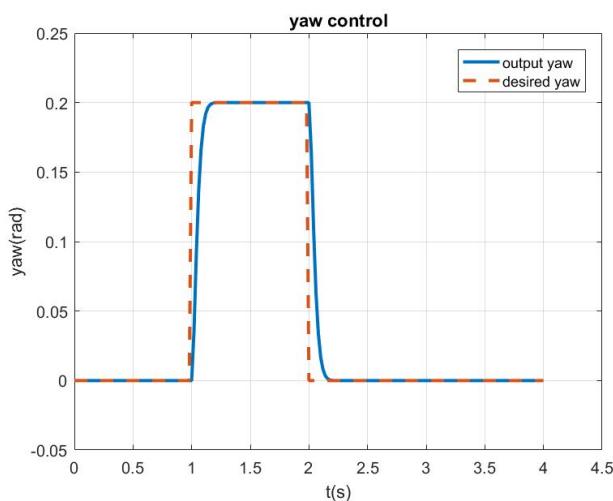


图 3.7 航偏角控制仿真, $k_p^\Omega = 22.3, k_p^\omega = 63.5, k_d^\omega = 3.32$

(图3.7) 为仿真中的航偏角控制，在实际中希望航偏角的响应有较小的过冲，因此选取了响应稍慢，超调量少的一组参数。

位移控制

根据上述设计的位移控制器，在MATLAB[®]中进行位移控制的仿真，仿真中通过设置阶跃的目标位移，得到位移控制器的阶跃响应，没有加入控制器中的前馈项。

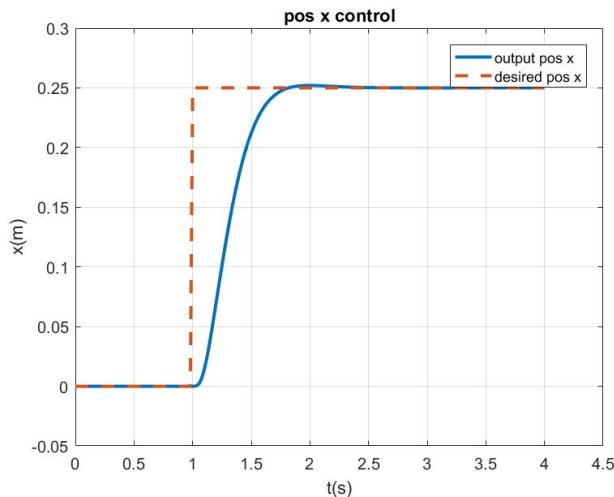


图 3.8 位移控制 x 轴仿真， $k_p^r = 25.2, k_i^r = 5.23, k_d^r = 8.58$

(图3.8) 为仿真中位移控制的 x 轴响应，由于飞行器的位移控制需要通过改变飞行器姿态来实现，在第六章中导出飞行器的电机的输出与飞行器位移为4阶导数的关系，从仿真中也可以看出飞行器的位移控制响应与姿态控制相比延迟较大，响应较慢。 y 轴的响应基本一致。

(图3.9) 为仿真中位移控制的 z 轴响应。特性与 x, y 轴的响应类似。

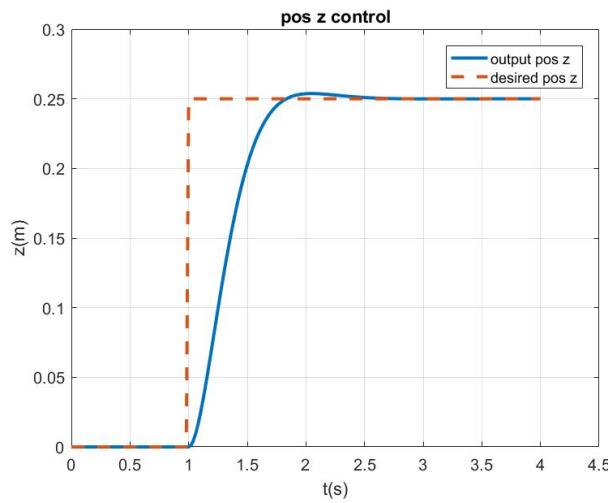


图 3.9 位移控制 z 轴仿真, $k_p^r = 28.2, k_i^r = 6.73, k_d^r = 9.33$

轨迹跟踪

在仿真环境中利用第七章所介绍的方法生成曲线路径并使飞行器跟踪, 得到以下仿真结果:

(图3.10) 为飞行器飞行轨迹与路径轨迹, 在仿真中, 飞行器的路径跟踪效果比较理想。(图3.11) 显示了三个轴的跟踪误差。

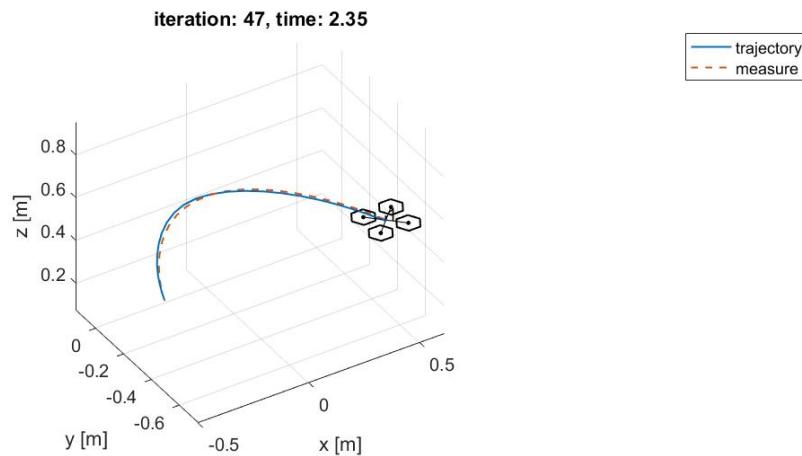


图 3.10 跟踪曲线轨迹仿真

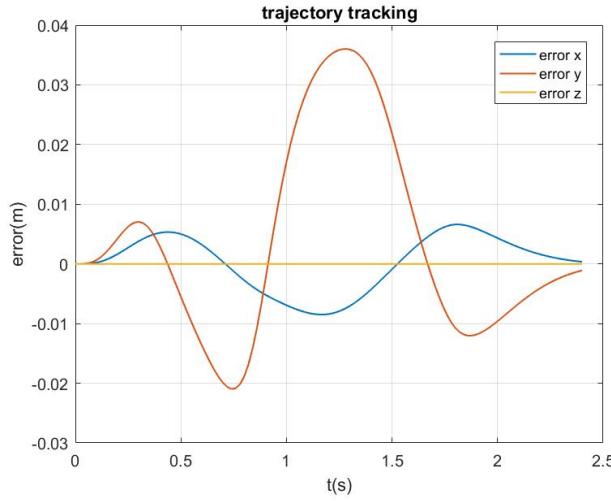


图 3.11 仿真中跟踪曲线轨迹时的误差，误差平方和 $x = 0.0026\text{m}$, $y = 0.0336\text{m}$, $z = 0.00\text{m}$

3.3.2 实际测试

按照所设计的控制器，在真实的飞行器进行测试，飞行器配置和结构如第二章所介绍，各项物理参数为（表3.1）：

物理量	测量结果
m	0.460kg
L	0.0883m
g	9.81m/s ²
I_{xx}	0.000354kg · m ²
I_{yy}	0.000365kg · m ²
I_{zz}	0.000817kg · m ²
k_F	3.1744
k_M	8.653

表 3.1 飞行器物理参数

由于飞行器结构基本对称，飞行器惯量矩阵中的非对角线元素近似为0。测试中以遥控器输入为目标值，测量值为飞行器响应，得到以下测量结果：

姿态控制

以仿真中的控制器参数为参考，手动调节真实飞行控制器中的参数，得到以

下几组较为适合的控制器参数。

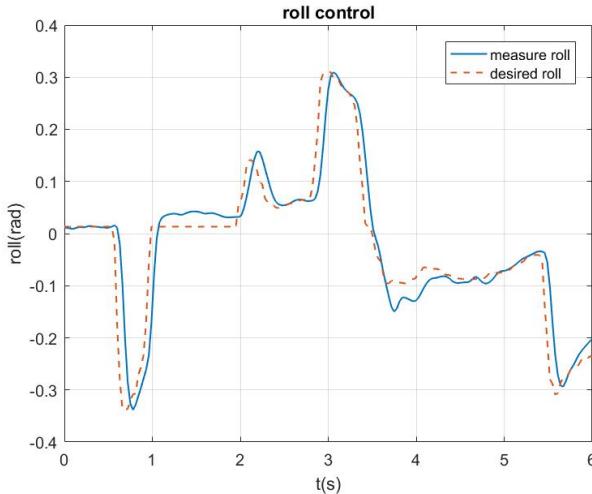


图 3.12 横滚角控制实际响应, $k_p^\Omega = 14.28, k_p^\omega = 65.4, k_d^\omega = 4.82$

(图3.12) 为真实飞行器的横滚角响应, 由数据可以看出, 控制器跟踪较好, 但跟仿真结果相比, 响应有一定的滞后。

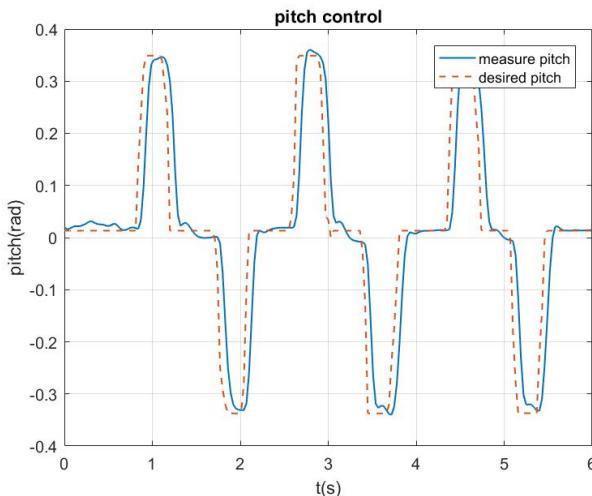


图 3.13 俯仰角控制实际响应, $k_p^\Omega = 14.28, k_p^\omega = 65.4, k_d^\omega = 4.82$

(图3.13) 为真实飞行器的俯仰角响应, 类似横滚角的响应, 曲线跟踪较好。

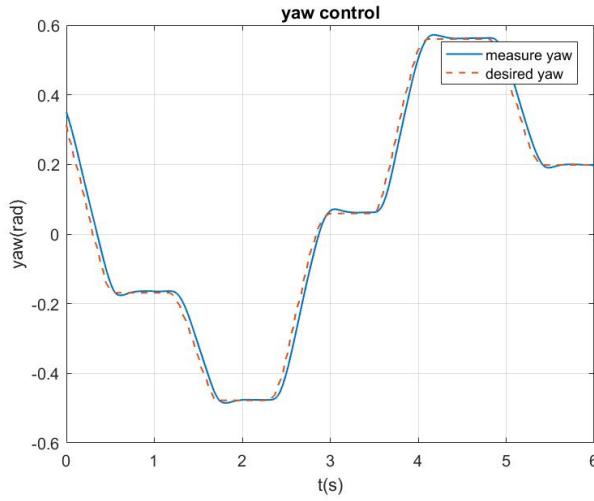


图 3.14 航偏角控制实际响应, $k_p^\Omega = 18.74$, $k_p^\omega = 76.5$, $k_d^\omega = 3.72$

(图3.14) 为真实飞行器的航偏角控制的响应, 由曲线可以看出航偏角度控制器的跟踪效果比较好。

由以上三个结果可以看出, 飞行器的姿态控制效果不错, 响应较快, 跟踪比较准确, 姿态曲线平滑, 噪声较小。在实验中, 飞行器的姿态横滚角度超过 $\pm 60^\circ$ 时, 飞行器仍能迅速恢复到悬停状态, 挂载100g重物时没有出现振荡现象, 控制器鲁棒性比较好。

位移控制

以仿真中的控制器参数为参考值, 调节真实控制器中的参数, 得到以下测量结果:

(图3.15) 为x轴位移控制效果, 由曲线可以看出位移控制响应的超调量大, 响应比较慢。这是因为在实验中发现, 当位移控制器在进行轨迹跟踪时, 由于控制器加入了前馈, 对控制器的响应速度要求并不高, 较慢的响应对轨迹跟踪的误差影响并不大 (图3.19)。而较快的位移控制响应, 在高速跟踪轨迹, 或者飞行器大角度飞行时, 时容易引起控制器不稳定, 使飞行器状态震荡或者发散。因此, 希望在控制器响应能接受的范围内, 令控制器鲁棒性尽量高, 以获得较好, 较安全的轨迹跟踪结果。

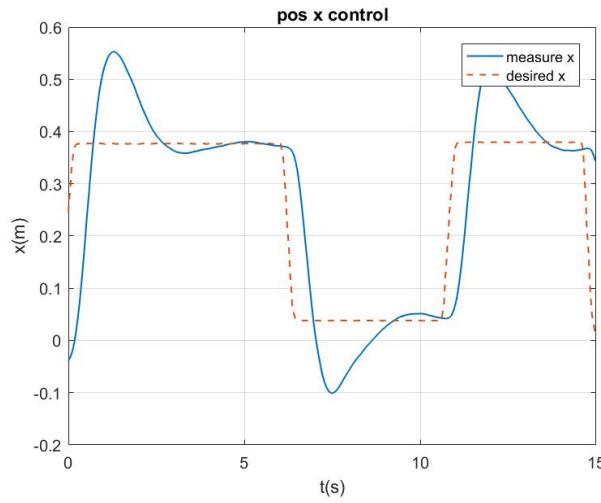


图 3.15 位移控制 x 轴实际响应, $k_p^r = 15.3, k_i^r = 8.43, k_d^r = 10.2$

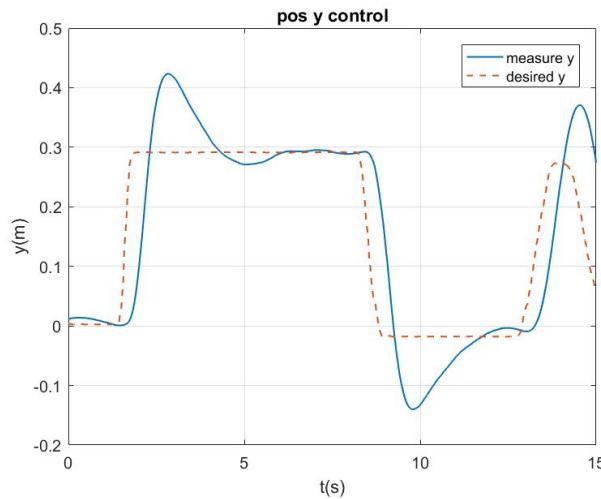


图 3.16 位移控制 y 轴实际响应, $k_p^r = 15.3, k_i^r = 8.43, k_d^r = 10.2$

(图3.16) (图3.17) 为 y, z 轴位移控制效果, 与 x 轴结果相似。

(图3.18) 为飞行器悬停时测得的位置误差, 从数据可以看出, 飞行器的悬停精度大概在 $\pm 1\text{cm}$, 得到 x, y, z 轴误差的标准差为 $\sigma_x = 0.0025\text{m}, \sigma_y = 0.0062\text{m}, \sigma_z = 0.0024\text{m}$ 。飞行器的悬停效果比较理想。

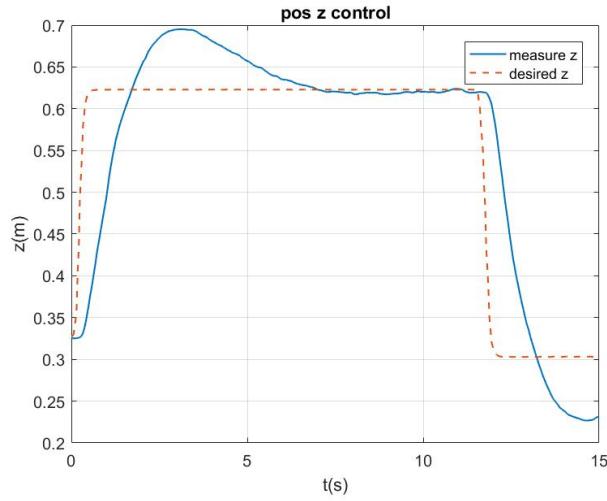


图 3.17 位移控制 z 轴实际响应, $k_p^r = 18.4, k_i^r = 10.4, k_d^r = 15.3$

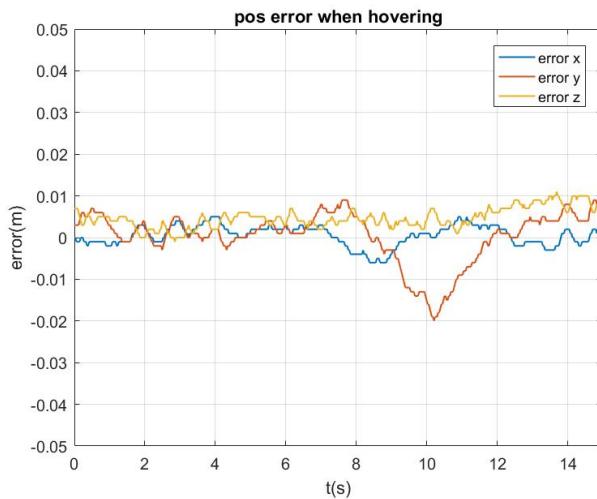
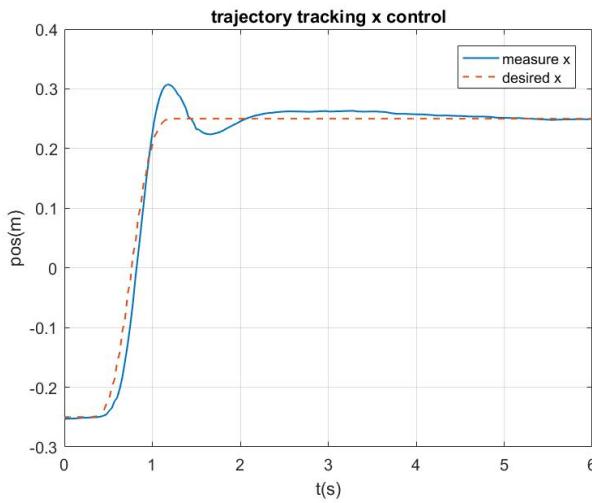


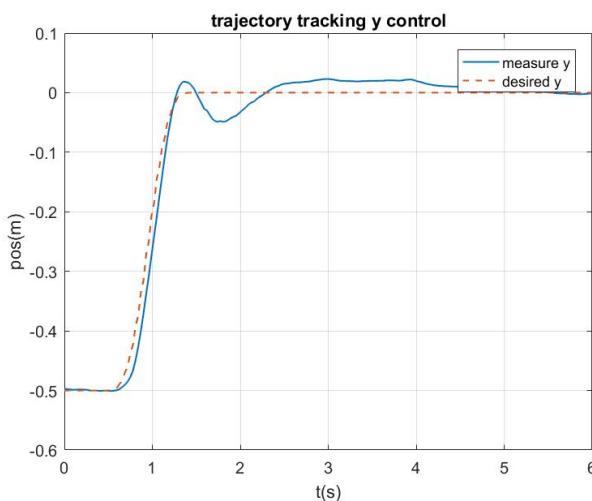
图 3.18 悬停时的位置误差, $\sigma_x = 0.0025\text{m}, \sigma_y = 0.0062\text{m}, \sigma_z = 0.0024\text{m}$

轨迹跟踪

利用第六章中介绍的路径生成方法, 令飞行器跟踪给定的飞行轨迹, 得到以下测试结果:

图 3.19 跟踪直线轨迹时 x 轴响应，误差平方和 = 0.0927m

(图3.19) 为飞行器跟踪轨迹时 x 轴位移的响应。从图中曲线可以看出，跟踪轨迹时的效果相比单独给定位移目标值时的跟踪效果好，这是因为控制器中加入了前馈控制，在模型参数比较准确的情况下，加入前馈控制可以有效提高控制器的响应速度以及减少超调量。(图3.20) 和 (图3.21) 为 y, z 轴跟踪效果，跟踪误差较小，飞行器的三维直线跟踪效果比较理想。

图 3.20 跟踪直线轨迹时 y 轴响应，误差平方和 = 0.0997m

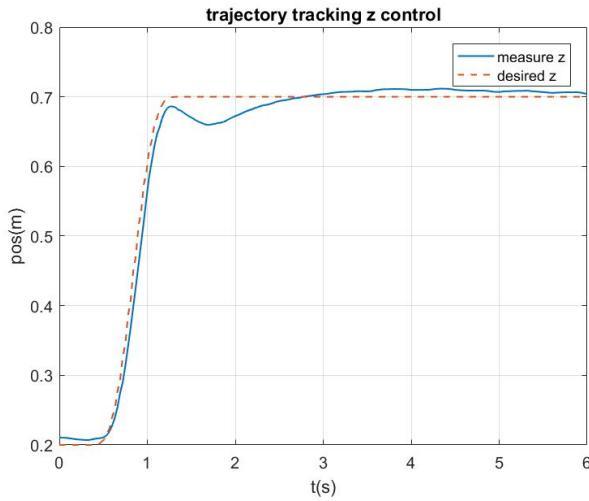


图 3.21 跟踪直线轨迹时 z 轴响应, 误差平方和 = 0.0651m

(图3.22) 为飞行器跟踪曲线轨迹的结果:

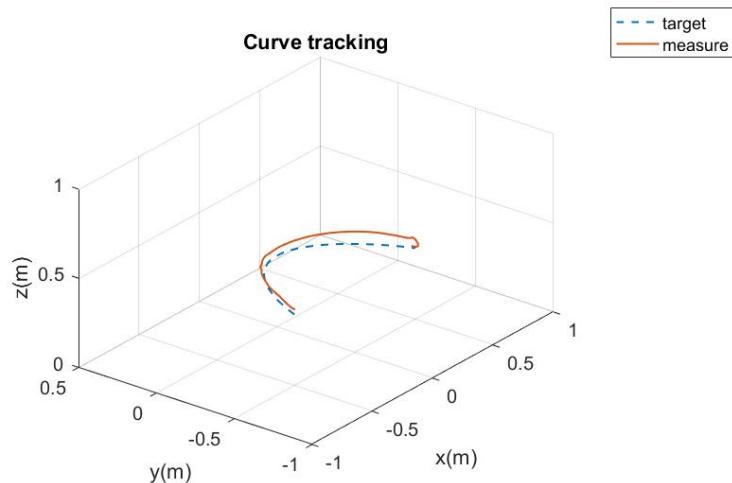


图 3.22 跟踪曲线轨迹实际结果

飞行时间为3s,从三维轨迹图看出, 飞行器 z 轴跟踪误差稍大, x, y 轴跟踪较好。(图3.23) 显示了各轴的误差, 并给出了跟踪误差的平方和。

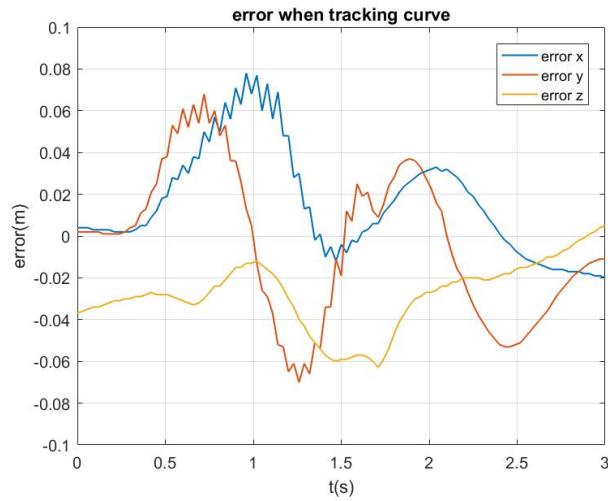


图 3.23 实际跟踪曲线轨迹时的误差, 误差平方和 $x = 0.0916\text{m}$, $y = 0.0878\text{m}$, $z = 0.1054\text{m}$

(图3.24) 显示了飞行器进行高速直线跟踪时 x 轴的响应, 路径长度为1m, 完成时间为1s, 跟踪时最大加速度达到 $6\text{m}/\text{s}^2$, 期间飞行器最大倾角约为 30° 。

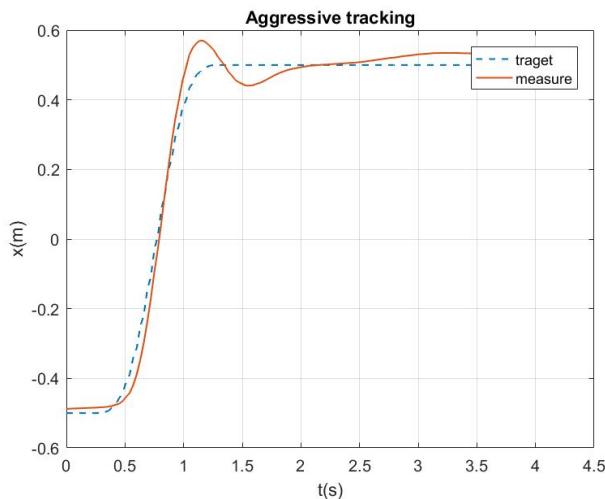


图 3.24 高速直线跟踪

第四章 状态估计

状态估计为闭环控制系统的一个重要的环节，反馈信号的噪声，延迟对控制系统的性能有着重要的影响。文献^{[25][26][27]}中对利用机载传感器进行飞行器状态估计的问题进行了研究。四轴飞行器进行状态估计，可以由四轴飞行器的运动方程（式3.9）入手，从飞行器的运动方程可以看出，飞行器的状态为一个12维的列向量，也就是说要按照第二章中的方法，实现对飞行器的姿态以及位移进行控制需要设计一个状态估计器来估计这12个状态。

另外，飞行器上的传感器，以及室内定位系统能提供以下几个数据：

- 陀螺仪：飞行器的三维角速度 ω_{meas}
- 加速度计： Γ_b 中测得的加速度 acc_{meas}
- 室内定位系统：飞行器在 Γ_w 中的位移 r_{meas} ，以及飞行器的姿态 Ω_{cam}

一般来说，利用飞行器的运动方程（式3.9），把传感器数据以及室内定位系统作为观测量，可以构建扩展卡尔曼滤波器（EKF），对运动方程中的12个状态以及各传感器误差进行在线估计。然而，滤波器中有12个以上的状态（比如加入对陀螺仪及加速度计的零偏进行估计，一共18个状态），其状态误差的协方差矩阵 P 为一个 18×18 的方阵，对矩阵进行求逆运算时运算量非常大（对于单片机来说），此外，对系统状态进行线性化时同样占用较大的运算时间，要在单片机上以几百Hz的频率运行一个十几维的EKF进行在线状态估计不太现实（实验中进行一次12维的EKF更新，运算时间大概4ms）。因此，本设计中，为了得到较小的估计误差，以及较高的更新频率，主要做了以下处理：

- 通过实验测试发现，经过低通滤波(截至频率42Hz)后陀螺仪的数据噪声较低，动态性能非常好，但是存在零点漂移，因此可以用去掉零偏的陀螺仪数据作为控制器的角速度反馈，不再作为被估计的状态。即：

$$\omega = \text{DLPF}(\omega_{meas}) - b_\omega$$

- 将整个飞行器的状态估计问题分为姿态估计问题和位移估计问题。
- 姿态估计中设计次优的KF。

- 位移估计中应用KF。

4.1 姿态估计

由（式3.9）有：

$$\dot{\Omega} = \mathbf{W}\omega = \begin{pmatrix} 1 & \tan(\theta) \sin(\varphi) & \tan(\theta) \cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sec(\theta) \sin(\varphi) & \sec(\theta) \cos(\varphi) \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (4.1)$$

离散化后得到：

$$\Omega_k = \mathbf{I}_{3 \times 3} \Omega_{k-1} + dt \mathbf{W}(\Omega_{k-1}) \omega_{k-1} \quad (4.2)$$

其中 $\mathbf{I}_{3 \times 3}$ 为 3×3 的单位矩阵， dt 为采样周期。

4.1.1 状态转移方程

设陀螺仪模型为：

$$\omega_{meas} = \omega + \mathbf{b}_\omega + \mathbf{w}_\omega$$

其中 \mathbf{b}_ω 为陀螺仪零偏，假设为常数。 \mathbf{w}_ω 为陀螺仪测量噪声，假设每个轴的噪声独立并服从零均值高斯分布：

$$\begin{aligned} w_{\omega,i} &\sim \mathcal{N}(0, \sigma_\omega), (i = x, y, z) \\ E(w_{\omega,i} w_{\omega,j}) &= 0, (i = x, y, z; j = x, y, z; i \neq j) \end{aligned}$$

σ_ω 为陀螺仪噪声的标准差。

令不包含噪声的陀螺仪测量值为： $\omega'_{meas} = \omega + \mathbf{b}_\omega$ 。

令状态量为：

$$\mathbf{x} = \begin{pmatrix} \Omega \\ \mathbf{b}_\omega \end{pmatrix} = \begin{pmatrix} \varphi \\ \theta \\ \psi \\ b_{\omega,x} \\ b_{\omega,y} \\ b_{\omega,z} \end{pmatrix} \quad (4.3)$$

根据（式4.2），建立状态方程：

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \omega'_{meas,k-1}, \mathbf{w}_{\omega,k-1}) \quad (4.4)$$

$$f(\mathbf{x}_{k-1}, \omega'_{meas,k-1}, \mathbf{w}_{\omega,k-1}) = \\ \begin{pmatrix} I_{3 \times 3} \Omega_{k-1} + dt \mathbf{W}(\Omega_{k-1})(\omega'_{meas,k-1} - b_{\omega,k-1} + w_{\omega,k-1}) \\ b_{\omega,k-1} \end{pmatrix}$$

4.1.2 测量方程

观测量如下：

$$\mathbf{y} = \begin{pmatrix} \Omega_{acc} \\ \Omega_{cam} \end{pmatrix} = \begin{pmatrix} \varphi_{acc} \\ \theta_{acc} \\ \varphi_{cam} \\ \theta_{cam} \\ \psi_{cam} \end{pmatrix}$$

其中 Ω_{acc} 和 Ω_{cam} 分别为加速度计与室内定位系统测得的欧拉角，由于加速度计的特点，加速度计只能测得 φ 角度和 θ 角度（附录C）。

测量方程为：

$$\mathbf{y} = \mathbf{Hx} + \mathbf{v} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \varphi \\ \theta \\ \psi \\ b_{\omega,x} \\ b_{\omega,y} \\ b_{\omega,z} \end{pmatrix} + \begin{pmatrix} v_{\Omega,acc} \\ v_{\Omega,acc} \\ v_{\Omega,cam} \\ v_{\Omega,cam} \\ v_{\Omega,cam} \\ v_{\Omega,cam} \end{pmatrix} \quad (4.5)$$

\mathbf{v} 为测量噪声，为包含5个元素的列向量，并假设测量噪声为零均值独立的高斯白噪声：

$$v_{\Omega,acc} \sim \mathcal{N}(0, \sigma_{\Omega,acc})$$

$$v_{\Omega,cam} \sim \mathcal{N}(0, \sigma_{\Omega,cam})$$

测量噪声协方差矩阵为：

$$\mathbf{R} = E(\mathbf{vv}^T) \quad (4.6)$$

4.1.3 次优KF

若直接利用状态转移方程（式4.4）进行状态估计，则需要采用EKF等复杂的非线性状态估计算法，在EKF对状态转移方程进行线性化步骤中，需要求解偏微分矩阵 $\frac{\partial f}{\partial \mathbf{x}}$ 和 $\frac{\partial f}{\partial \mathbf{w}_\omega}$ ，由于矩阵 \mathbf{W} 中涉及大量三角函数，EKF线性化过程的运算量将非常大，不利于高频率运行，并且滤波器的鲁棒性也不高。因此，接下来的讨论，

将简化状态转移方程，进而设计次优的状态估计器，大大减少状态估计过程的运算量。

对方程（式4.4），利用假设 $\varphi \approx 0, \theta \approx 0$ ，可简化为：

$$\boldsymbol{x}_k = \begin{pmatrix} \mathbf{I}_{3 \times 3} \boldsymbol{\Omega}_{k-1} + dt(\boldsymbol{\omega}'_{meas,k-1} - \boldsymbol{b}_{\omega,k-1} + \boldsymbol{w}_{\omega,k-1}) \\ \boldsymbol{b}_{\omega,k-1} \end{pmatrix}$$

进一步写成一般线性系统的形式：

$$\boldsymbol{x}_k = \mathbf{F}\boldsymbol{x}_{k-1} + \mathbf{G}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \quad (4.7)$$

其中：

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 & -dt & 0 & 0 \\ 0 & 1 & 0 & 0 & -dt & 0 \\ 0 & 0 & 1 & 0 & 0 & -dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{G} = \begin{pmatrix} dt \\ dt \\ dt \\ 0 \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{u} = \begin{pmatrix} \boldsymbol{\omega}'_{meas,x} \\ \boldsymbol{\omega}'_{meas,y} \\ \boldsymbol{\omega}'_{meas,z} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

过程噪声的协方差矩阵 \mathbf{Q} 为：

$$\mathbf{Q} = \begin{pmatrix} E(\boldsymbol{w}_{\omega}\boldsymbol{w}_{\omega}^T) & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \end{pmatrix} \quad (4.8)$$

其中 $\mathbf{O}_{3 \times 3}$ 为 3×3 全零矩阵。

注意（式4.8）为连续系统中的协方差矩阵，其对应得离散系统中的协方差矩阵为：

$$\mathbf{Q}_k = \int_0^{dt} \mathbf{F}\mathbf{Q}\mathbf{F}^T dt$$

在采样间隔很小时，可以近似为：

$$\mathbf{Q}_k = dt\mathbf{Q}$$

利用(式4.7)(式4.8)(式4.5)和(式4.6)，递推求解Riccati方程：

$$\begin{aligned} \mathbf{M}_k &= \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k\mathbf{H}^T(\mathbf{H}\mathbf{M}_k\mathbf{H}^T + \mathbf{R}_k)^{-1} \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{M}_k \end{aligned} \quad (4.9)$$

其中 \mathbf{P} 为状态误差的协方差矩阵， \mathbf{M} 为先验估计得到的状态误差的协方差矩阵，即未考虑测量方程时的 \mathbf{P} 矩阵。 \mathbf{K} 为KF的增益。

整个姿态估计KF过程为：

$$\begin{aligned}\mathbf{x}_k^- &= \mathbf{F}\mathbf{x}_{k-1}^+ + \mathbf{G}\mathbf{u}_{k-1} \\ \mathbf{M}_k &= \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k\mathbf{H}^T(\mathbf{H}\mathbf{M}_k\mathbf{H}^T + \mathbf{R}_k)^{-1} \\ \mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-) \\ \mathbf{P}_k &= (\mathbf{I}_{6 \times 6} - \mathbf{K}_k\mathbf{H})\mathbf{M}_k\end{aligned}\quad (4.10)$$

其中 \mathbf{x}_k^- 为先验估计的状态， \mathbf{x}_k^+ 是引入测量后的后验估计。

注意到，在第一步计算先验估计 \mathbf{x}_k^- 中，直接利用了线性化的 \mathbf{F} 进行状态转移，在飞行器偏离悬停状态较大时，或者瞬间加速时，模型误差会比较大，造成先验估计的误差比较大。在KF中，模型的准确性对滤波器的性能有重要的影响，因此，在第一步应该采用非线性模型（式4.4）计算先验估计，得到如下的估计流程：

$$\begin{aligned}\mathbf{x}_k^- &= f(\mathbf{x}_{k-1}^+, \omega_{meas,k-1}, \mathbf{O}_{3 \times 1}) \\ \mathbf{M}_k &= \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k\mathbf{H}^T(\mathbf{H}\mathbf{M}_k\mathbf{H}^T + \mathbf{R}_k)^{-1} \\ \mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-) \\ \mathbf{P}_k &= (\mathbf{I}_{6 \times 6} - \mathbf{K}_k\mathbf{H})\mathbf{M}_k\end{aligned}$$

需要注意，在实际中，由于观测数据的更新周期往往与状态转移的周期不一样，对应的 \mathbf{H} 矩阵也应相应改变。

再进一步观察可以看出，递推求解Riccati方程（式4.9）时与 ω_{meas} 和 \mathbf{y} 无关，若 \mathbf{Q}, \mathbf{R} 矩阵在滤波器运行过程中为常数，递推求解Riccati方程实际上可以离线运行，取其稳态解得到滤波器稳态增益 \mathbf{K}_∞ ，滤波器进一步简化为：

$$\mathbf{x}_k^- = f(\mathbf{x}_{k-1}^+, \omega_{meas,k-1}, \mathbf{O}_{3 \times 1}) \quad (4.11)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_\infty(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-) \quad (4.12)$$

4.2 位移估计

若姿态估计完成，则可以得到估计的飞行器的欧拉角 $\hat{\Omega}$ ，在位移估计步骤中假设姿态估计的欧拉角 $\hat{\Omega}$ 为真实欧拉角 Ω ，即 $\hat{\Omega} = \Omega$ ，由（式3.4）可以把 Γ_b 中的加速度投影到 Γ_w 中：

$$\mathbf{a}_w = {}^w\mathbf{R}_b \mathbf{a}_b$$

令三轴加速度计的测量值为 \mathbf{a}_{meas} ，并将其投影到 Γ_w ，减去重力加速度常量 \mathbf{g} ，得到飞行器在 Γ_w 中的加速度测量值 \ddot{r}_{meas} ：

$$\ddot{r}_{meas} = -{}^w\mathbf{R}_b \mathbf{a}_{meas} - \mathbf{g}$$

由于三个坐标轴 x_w, y_w, z_w 相互独立，采用KF对飞行器位移进行估计时，可以分别对每个坐标轴的位移，速度等的状态估计应用相同结构的KF，因此以下就其中一个坐标轴来设计KF。

4.2.1 状态转移方程

令位移估计KF的状态量为：

$$\mathbf{x} = \begin{pmatrix} r \\ \dot{r} \\ b_{\ddot{r}} \end{pmatrix}$$

其中 $r, \dot{r}, b_{\ddot{r}}$ 分别为飞行器在 Γ_w 中的位移，速度，以及加速度测量值的零偏。

状态转移方程为：

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (4.13)$$

其中：

$$\mathbf{F} = \begin{pmatrix} 1 & dt & -\frac{1}{2}dt^2 \\ 0 & 1 & -dt \\ 0 & 0 & 1 \end{pmatrix}, \mathbf{G} = \begin{pmatrix} \frac{1}{2}dt^2 \\ dt \\ 0 \end{pmatrix}, \mathbf{u} = \ddot{r}_{meas}$$

\mathbf{w} 为过程噪声，为加速度测量值引入的高斯白噪声，均值为零，标准差为 $\sigma_{\ddot{r}}$ ，并且假设测量值的零偏 $b_{\ddot{r}}$ 为常量，过程噪声协方差为：

$$\mathbf{Q} = E(\mathbf{w}\mathbf{w}^T) \quad (4.14)$$

4.2.2 测量方程

测量值为室内定位系统返回的对应坐标轴的位移信息：

$$y = r_{cam} \quad (4.15)$$

测量方程为：

$$y = \mathbf{H}\mathbf{x} + v_{r,cam} \quad (4.16)$$

其中：

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

$v_{r,cam}$ 为室内定位系统测量数据的噪声，假设为零均值高斯白噪声，标准差为 $\sigma_{r,cam}$ ，方差为：

$$R = E(vv^T) = \sigma_{r,cam}^2 \quad (4.17)$$

最后，由（式4.13）（式4.14）（式4.16）和（式4.17），根据KF算法进行飞行器的位移估计：

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{M}_k &= \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{M}_k\mathbf{H}^T(\mathbf{H}\mathbf{M}_k\mathbf{H}^T + R_k)^{-1} \\ \mathbf{x}_k^+ &= \mathbf{x}_k^- + \mathbf{K}_k(y_k - \mathbf{H}\mathbf{x}_k^-) \\ \mathbf{P}_k &= (\mathbf{I}_{3 \times 3} - \mathbf{K}_k\mathbf{H})\mathbf{M}_k \end{aligned} \quad (4.18)$$

在实际中，由于观测数据的更新周期往往与状态转移的周期不一样，对应的 \mathbf{H} 矩阵也应相应改变。

4.3 测量延迟

在第二章2.2.5中提到，由于摄像机感光元件曝光，数据传输和处理，无线通信等原因，室内定位系统的反馈数据存在40ms左右的延迟。对于控制系统来说，反馈数据的延迟会使控制系统的鲁棒性下降，容易引起系统振荡。

通过仿真发现定位系统的数据延迟对飞行器的速度估计影响尤为明显，在实验中发现若直接使用带延迟的测量来估计飞行器状态，飞行器依然能完成悬停，

路径跟踪等任务，但是在飞行器需要产生较大线加速度（倾斜角度大）时系统容易振荡，甚至发散。

（图4.1）为仿真环境中无测量延迟情况下的飞行器速度估计误差，（图4.2）为加入测量延迟40ms时的速度估计误差，从结果可以看出，由于测量延迟的引入，速度估计误差明显增大。

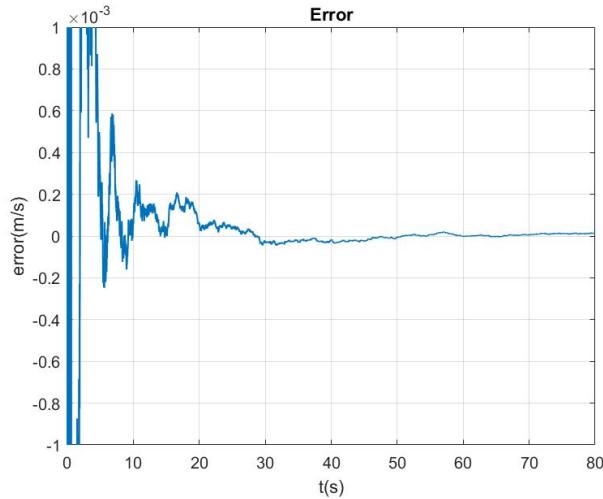


图 4.1 无测量延迟时的速度估计误差

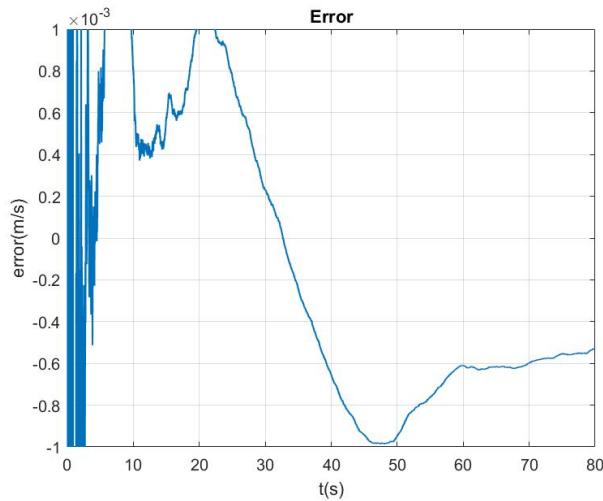


图 4.2 有测量延迟时的速度估计误差

为尽量避免由测量延迟引入的额外的噪声，应该对状态估计器做相应的修改。实验测试中发现，测量数据的延迟基本是一个常量，因此在这里假设测量延迟固定。

考虑如（式4.13）的线性系统，假设当前时刻为 k ，实际测量时刻为 k_0 ，由当前的状态 \mathbf{x}_k 以及 $k_0 \sim k-1$ 时刻记录下的控制量 \mathbf{u}_j , ($j = k_0, \dots, k-1$)利用下式，可以追溯到 k_0 时刻的状态 \mathbf{x}_{k_0} :

$$\mathbf{x}_{k-1} = \mathbf{F}^{-1}(\mathbf{x}_k - \mathbf{G}\mathbf{u}_{k-1})$$

同时根据（式4.18）中的第二行，误差协方差矩阵的传播方程，可以由 k 时刻的误差协方差矩阵 \mathbf{P}_k 追溯到 k_0 时刻的误差协方差矩阵 \mathbf{P}_{k_0} :

$$\mathbf{P}_{k-1} = \mathbf{F}^{-1}(\mathbf{P}_k - \mathbf{Q}_k)(\mathbf{F}^T)^{-1}$$

将追溯得到的状态 \mathbf{x}_{k_0} 和误差协方差矩阵 \mathbf{P}_{k_0} 作为 k_0 时刻的先验估计，分别记为 $\mathbf{x}_{k_0}^-$ 和 \mathbf{M}_{k_0} ，使用测量值 \mathbf{y}_{k_0} 和卡尔曼增益 \mathbf{K}_{k_0} 得到后验估计:

$$\begin{aligned}\mathbf{K}_{k_0} &= \mathbf{M}_{k_0} \mathbf{H}^T (\mathbf{H} \mathbf{M}_{k_0} \mathbf{H}^T + \mathbf{R}_{k_0})^{-1} \\ \mathbf{x}_{k_0}^+ &= \mathbf{x}_{k_0}^- + \mathbf{K}_{k_0} (\mathbf{y}_{k_0} - \mathbf{H} \mathbf{x}_{k_0}^-) \\ \mathbf{P}_{k_0} &= (\mathbf{I} - \mathbf{K}_{k_0} \mathbf{H}) \mathbf{M}_{k_0}\end{aligned}$$

然后，由（式4.18）中第一，二行，以及 $\mathbf{x}_{k_0}^+$ 和 \mathbf{P}_{k_0} 复原当前状态，得到 k 时刻的后验估计 \mathbf{x}_k^+ 和 \mathbf{P}_k 。

对测量延迟进行修正后的仿真结果如（图4.3），可以看出，曲线与无测量延时情况下的误差曲线非常相似。

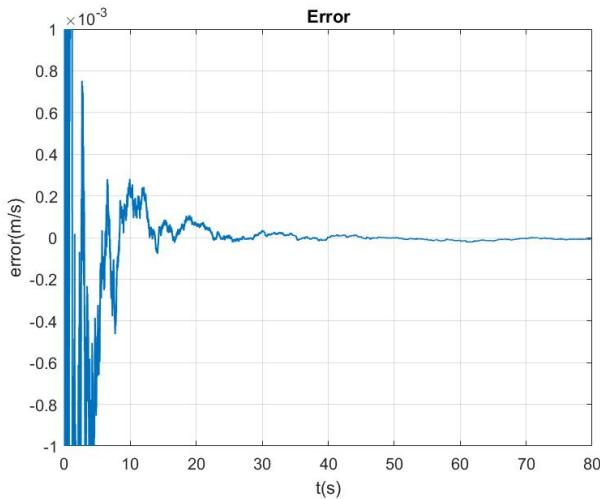


图 4.3 补偿测量延迟后的速度估计误差

4.4 仿真与实验结果

本节提供了采用上述方法设计的状态估计器的仿真与实际测量结果，实验证明上述的方法有较好的效果。

由飞行器返回的数据得到各传感器数据噪声特性如下：

测量	噪声标准差
加速度计($\sigma_{\ddot{r}}$)	1.8282m/s ²
陀螺仪(σ_ω)	0.042rad/s
加速度姿态($\sigma_{\Omega,acc}$)	0.528rad
摄像机姿态($\sigma_{\Omega,cam}$)	0.024rad
摄像机位移($\sigma_{r,cam}$)	0.012m

表 4.1 传感器噪声特性

上表（表4.1）中传感器噪声的标准差取三轴中噪声最大的一轴的标准差，并且噪声基本服从高斯分布。

4.4.1 MATLAB[®]仿真

仿真数据来自飞行器传感器采集到的数据，估计器在MATLAB[®]中运行，由于无线通信硬件和单片机性能的有限，数据返回的周期为30ms，因此在仿真环境中的估计器采样周期为30ms而实际飞行器中的采样周期为1ms。所以，在应用到实际飞行器中，需要对参数进行相应调节。

姿态估计

（图4.4）为飞行器横滚角估计的仿真结果，数据来自处于悬停状态下的飞行器，图中与估计值对比的是室内定位系统得到的欧拉角，另外加速度计得到的欧拉角噪声相对要大的多（图中并未显示），从结果可以看出滤波器能去除掉大量的观测噪声。

（图4.5）为飞行器横滚角估计误差，状态误差协方差矩阵 P 初始为单位矩阵，当滤波器趋于稳态时，理论上的误差标准差为 $\sigma_{error,roll} = 0.0178\text{rad}$ ，从图中可以看出，实际的估计误差基本在理论值以内。俯仰角和航偏角估计也得到相似的结果。

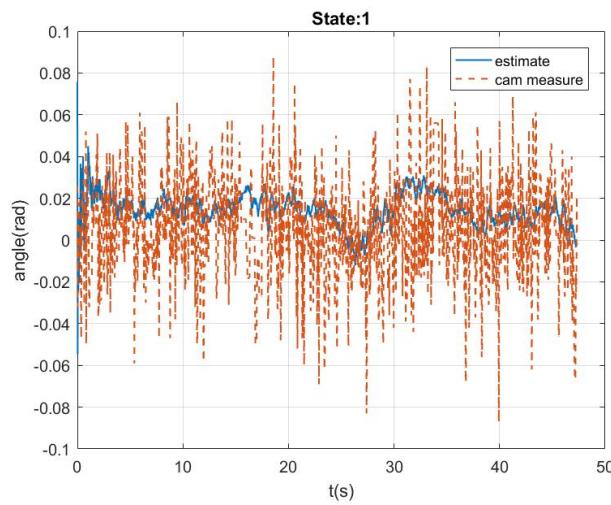


图 4.4 横滚角估计仿真

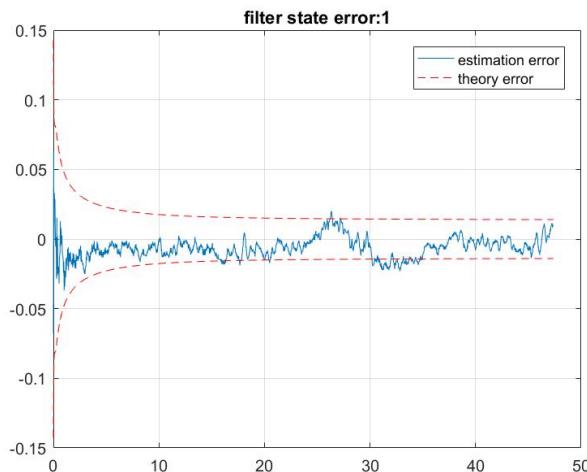


图 4.5 仿真中横滚角估计误差

此时的滤波器增益为：

$$\mathbf{K} = \begin{pmatrix} 0.0011 & 0 & 0.0054 & 0 & 0 \\ 0 & 0.0011 & 0 & 0.0054 & 0 \\ 0 & 0 & 0 & 0 & 0.0061 \\ 0 & 0 & -0.0001 & 0 & 0 \\ 0 & 0 & 0 & -0.0001 & 0 \\ 0 & 0 & 0 & 0 & -0.0001 \end{pmatrix}$$

(图4.6) 为仿真中的估计得到的陀螺仪零偏。由于陀螺仪在每次飞行前进行自

动校正，校正后零偏较小，仿集中得到陀螺仪 x, y, z 轴的零偏分别为 0.0004rad/s , -0.0002rad/s , 0.0011rad/s 。

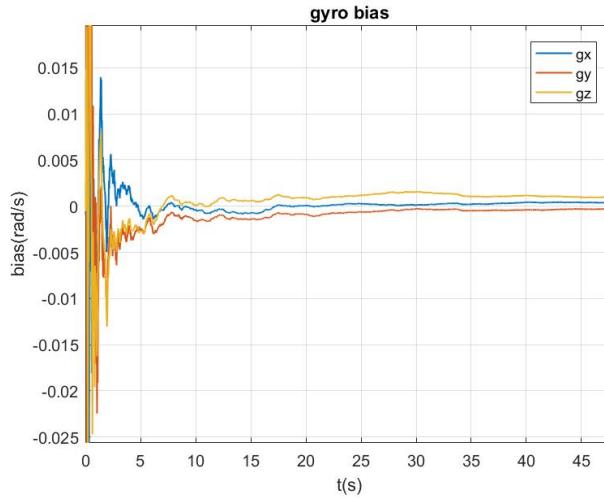


图 4.6 仿集中陀螺仪零偏估计

位移估计

位移估计仿集中，输入为转换到 Γ_w 中并去除重分量的飞行器加速度测量值 $\ddot{\mathbf{r}}_{meas}$ 以及室内定位系统的位移测量值 \mathbf{r}_{cam} 。

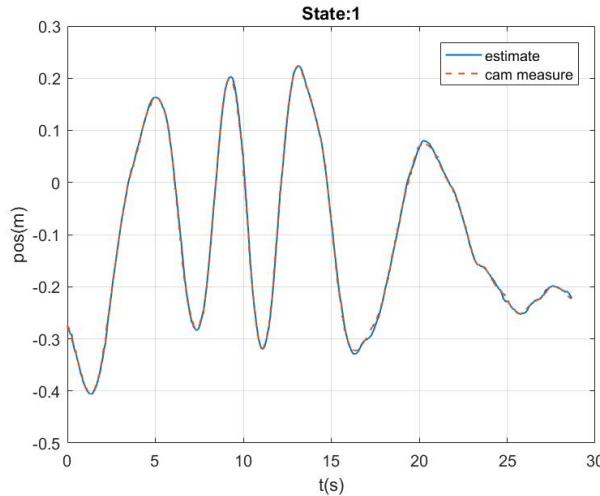


图 4.7 位移估计 x 轴仿真

(图4.7) 中显示了仿集中 x 轴位移的估计值，由结果可看出估计值与测量值基本一致，曲线比较平滑，当然另一方面说明室内定位系统的位置数据噪声已经

非常小。(图4.8)为相应的x轴的速度估计,可以看出,速度估计噪声比较大,这是仿真时使用的采样间隔较大造成的,在实际中采用较短的采样周期,可以得到较好的速度估计结果。

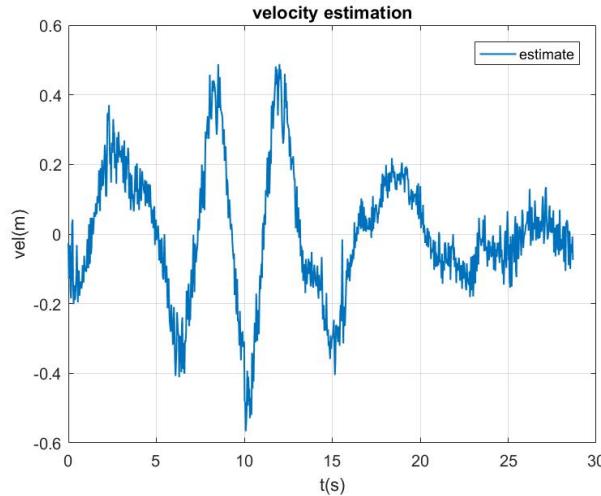


图 4.8 速度估计x轴仿真

(图4.9)为x轴的估计误差,由于没有更准确的测量作为真实值,在这里采用经过滑动窗口平滑后的室内定位系统数据近似为真实值,以对估计误差提供一个参考值。

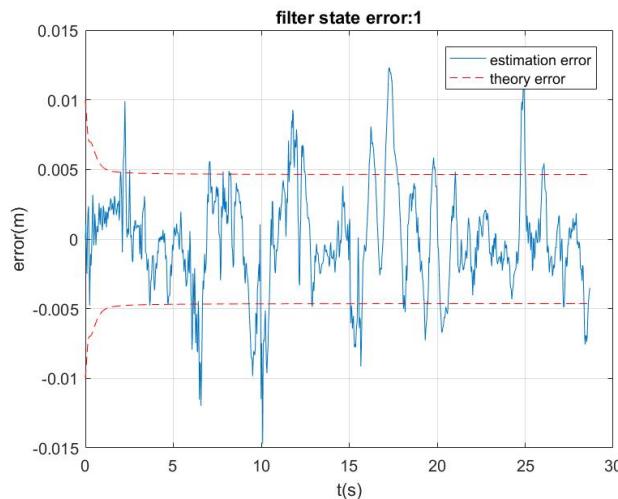


图 4.9 仿真中x轴位置估计误差

(图4.10)为仿真中的x轴加速度零偏估计结果,稳态值大概为 0.008m/s^2 。对于y, z轴的仿真结果,与x轴结果类似。

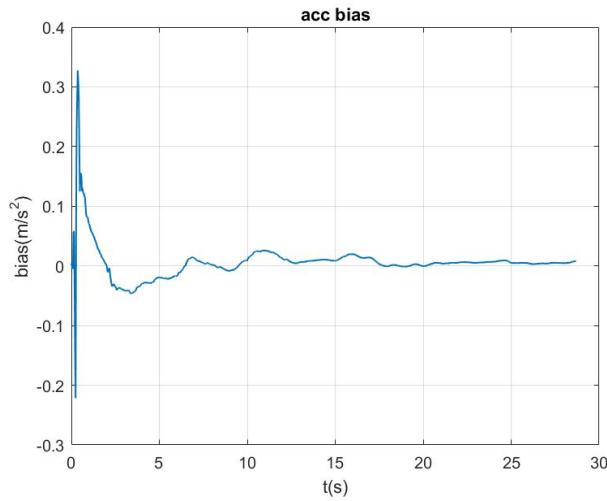


图 4.10 仿真中 x 轴加速度零偏

与上述结果响应的滤波器增益为：

$$\mathbf{K} = \begin{pmatrix} 0.0955 \\ 0.1596 \\ -0.0057 \end{pmatrix}$$

4.4.2 实际测试

实际测试中姿态估计运行频率为1000Hz，其中陀螺仪与加速度计采样率为1000Hz，室内定位系统反馈速率为40Hz，由于没有更精确的测量数据，实验主要通过估计值与测量值比较来判断估计效果。

姿态估计

(图4.11) ~ (图4.13) 为真实中飞行器的估计结果，相对于仿真结果，由于采样频率较高高，估计结果比较理想。为了提高系统鲁棒性，在实际中过程噪声的标准差设置值稍大于实验测量值。

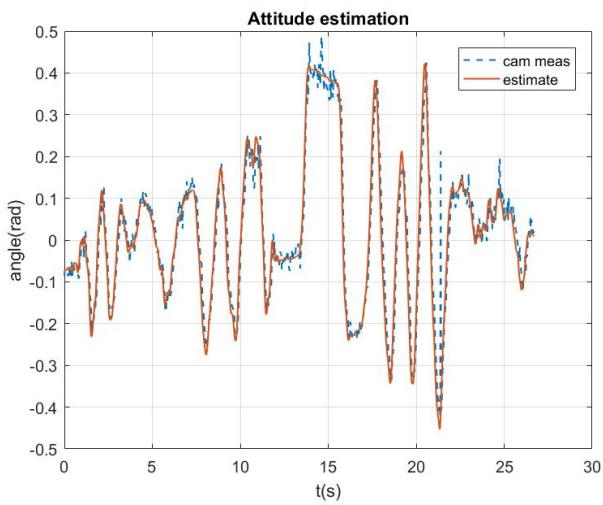


图 4.11 横滚角的真实估计效果

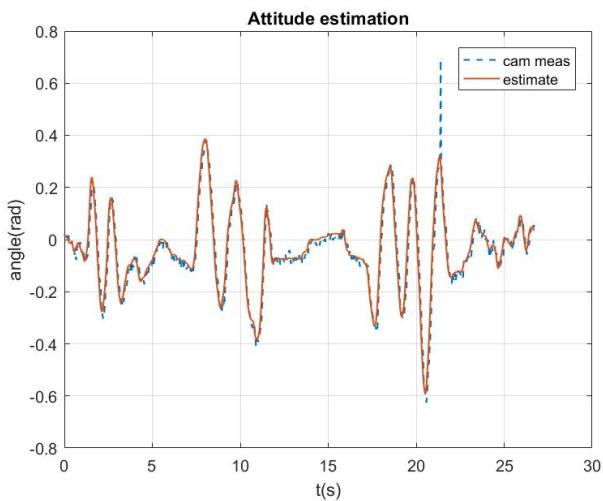


图 4.12 俯仰角的真实估计效果

与上述估计结果相应的滤波器增益为：

$$K = \begin{pmatrix} 0.0002 & 0 & 0.0024 & 0 & 0 \\ 0 & 0.0002 & 0 & 0.0024 & 0 \\ 0 & 0 & 0 & 0 & 0.0025 \\ -0.0002 & 0 & -0.0022 & 0 & 0 \\ 0 & -0.0002 & 0 & -0.0022 & 0 \\ 0 & 0 & 0 & 0 & -0.0022 \end{pmatrix}$$

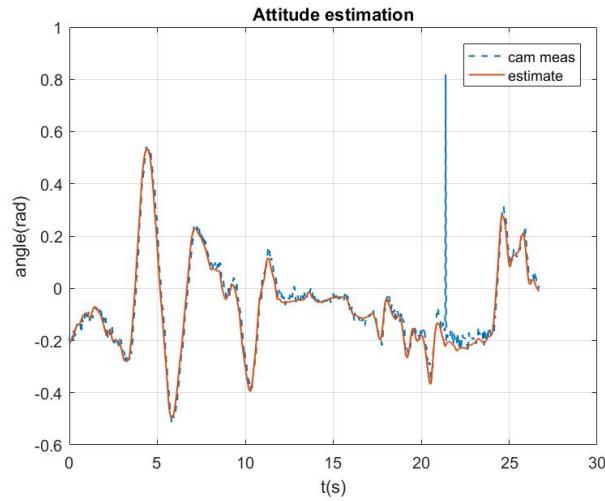
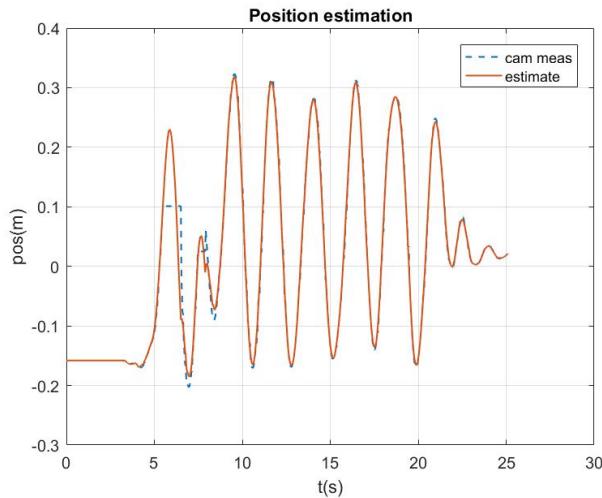


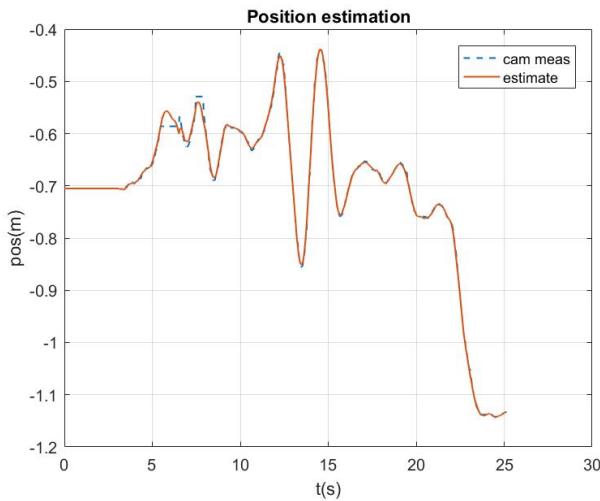
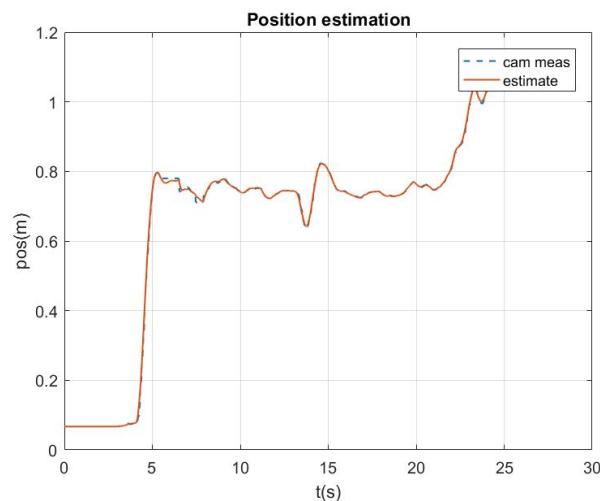
图 4.13 航偏角的真实估计效果

位移估计

位移估计的实际效果如下：

图 4.14 x 轴真实估计效果

由（图4.14）~（图4.16）可以看出，位移估计的实际效果较好。（图4.17）为相应的速度估计，可以看出，在实际中由于采样间隔缩短，相对于仿真结果，速度估计的噪声明显降低。

图 4.15 y 轴真实估计效果图 4.16 z 轴真实估计效果

对三轴数据采用相同结构和参数的KF，与上述结果相应的滤波器增益为：

$$\boldsymbol{K} = \begin{pmatrix} 0.151 \\ 0.628 \\ -0.00621 \end{pmatrix}$$

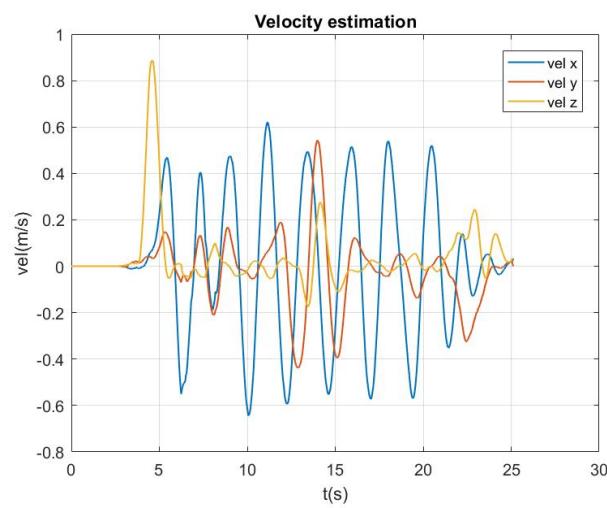


图 4.17 三轴速度真实估计效果

第五章 室内定位系统

目前室内定位定位系统种类较多，文献^[28]中对目前各类的室内定位系统进行了详细的介绍，比较，其中指出对于不同的应用场景，室内定位系统的性能要求差距甚大。学位论文^[29]中对基于标记物的运动跟踪系统进行了较详细的探讨，包括对摄像机的校正，标记点的定位，模型的匹配等问题。本设计的室内定位系统从基础的立体几何理论入手，实现三维空间的定位与姿态测量，系统能保证有较好的定位精度，较高的数据更新频率，以及系统有较高的鲁棒性，能够为飞行器提供高质量的反馈数据。

本设计中基于双目红外摄像机的室内定位系统主要内容为以下几点：

- 确定摄像机的内参数。
- 确定图像坐标与地平面坐标的变换关系（齐次变换）。
- 确定摄像机的空间坐标。
- 标记物的提取。
- 确定所有标记点的空间坐标。
- 确定飞行器位移与姿态。

5.1 摄像机校正

定义摄像机1的坐标系为 Γ_{cam1} ，摄像机2的坐标系为 Γ_{cam2} ，空间参考坐标系为 Γ_w

5.1.1 内参数校正

摄像机的内参数校正在MATLAB[®]中的Camera Calibrator软件中利用棋盘法进行，校正后得到摄像机的内参数矩阵，以及镜头的径向和切向畸变系数。利用校正结果，可以得到校正后的图像像素坐标 m 。

5.1.2 齐次变换

三维空间中某个坐标系中的点 \mathbf{M} 与其投影到另外一个坐标系中对应的点 \mathbf{M}' 之间的关系可以由单应矩阵 \mathbf{H} 来描述：

$$\begin{pmatrix} \mathbf{M}' \\ 1 \end{pmatrix} = s\mathbf{H} \begin{pmatrix} \mathbf{M} \\ 1 \end{pmatrix} \quad (5.1)$$

其中：

$$\mathbf{H} = \begin{pmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t}_{3 \times 1} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

\mathbf{H} 为一个 4×4 的矩阵，其中包含一个旋转矩阵 \mathbf{R} 和平移矩阵 \mathbf{t} ， s 为一个缩放系数。

本小节的目标是将图像平面 Γ_{cam1} 中的任意一点 \mathbf{m} 映射到 Γ_w 中 $z_w = 0$ 平面（即室内地面）上一点 \mathbf{M} ，该变换是二维点之间的映射（ $m_z = 0, M_z = 0$ ），由（式5.1）得：

$$\begin{pmatrix} M_x \\ M_y \\ 1 \end{pmatrix} = s \begin{pmatrix} r_{11} & r_{12} & t_{11} \\ r_{21} & r_{22} & t_{21} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m_x \\ m_y \\ 1 \end{pmatrix} \quad (5.2)$$

该单应矩阵 \mathbf{H} 为一个 3×3 的方阵，若把缩放系数也写进 \mathbf{H} 矩阵里，得到：

$$\begin{pmatrix} M_x \\ M_y \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} m_x \\ m_y \\ 1 \end{pmatrix}$$

接下来需要求解 \mathbf{H} 矩阵，求解 \mathbf{H} 矩阵即求解一个线性方程组，由于测量噪声的存在，方程组的系数矩阵可能存在奇异的情况，造成方程组没有精确解，因此，需要使用最小二乘法或其它近似的方法来求解该方程组。

OpenCV中提供了求解单应性矩阵的函数：

```
Mat findHomography(InputArray srcPoints,
                    InputArray dstPoints,
                    int method=0,
                    double ransacReprojThreshold=3,
                    OutputArray mask=noArray())
```

求解方法采用基于RANSAC的鲁棒方法：`method = CV_RANSAC`。

求取摄像机 \mathbf{H} 矩阵的具体过程如下：

1. 如（图5.1）在地面上做标记，并且记录标记物在 Γ_w 中的坐标，图中的1, 2, 3, 4标记为实际地面上边长为170cm的正方形的四个顶点，正方形的四边均与坐标轴平行，且其中心为坐标原点。
2. 识别图中标记物的图像坐标，可以通过特征提取，手动选择等方法获得4个标记物在图中的坐标，在本设计中是通过在标记物上放置红外发光点，并经过红外标记物提取后获得地面标记物的图像坐标。
3. 根据地面标记物的实际坐标与其一一对应的图像坐标，把对应点对输入到findHomography()函数中，获得摄像机的单应 \mathbf{H} 。

对摄像机1，摄像机2进行相同的操作，获得矩阵 \mathbf{H}_1 和 \mathbf{H}_2 。

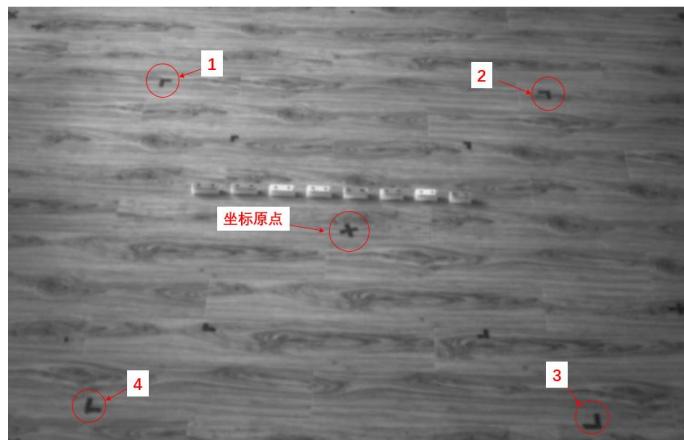


图 5.1 摄像机1图像

\mathbf{H} 矩阵获得后，图像中的每一个像素可以与 Γ_w 中 $z_w = 0$ 平面上的点一一对应起来，即可以通过图像坐标来计算空间中任意一点 \mathbf{P} 在室内地面上的投影的坐标 \mathbf{M} （图5.2）。

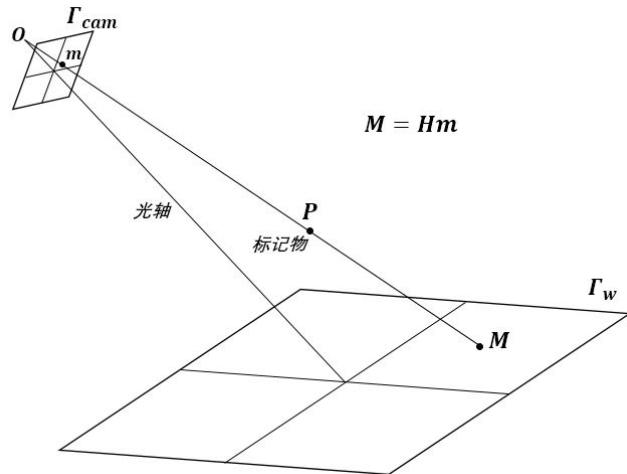


图 5.2 地面投影的坐标

变换结果

将摄像机1，摄像机2对应的图像进行齐次变换后（图5.3），把得到的图像融合到一幅图上，得到（图5.4）结果，可以看出图像1，2中相应的物体完全重合。

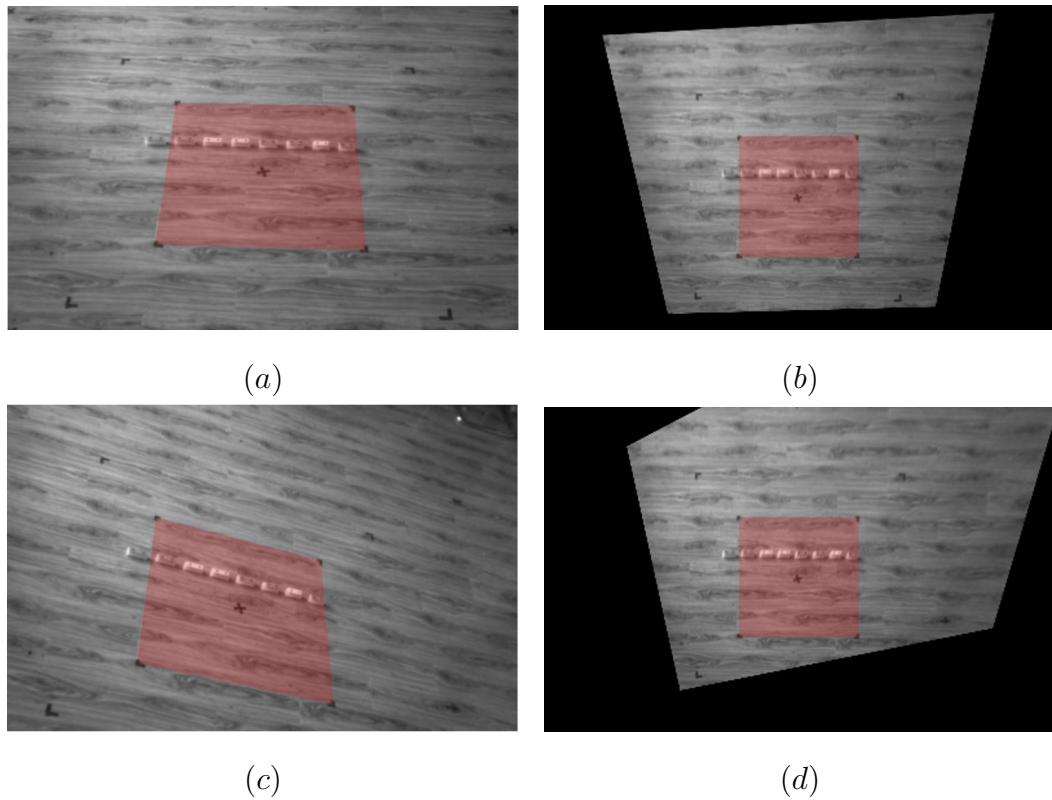


图 5.3 (a)图像1变换前;(b)图像1变换后;(c)图像2变换前;(d)图像2变换后

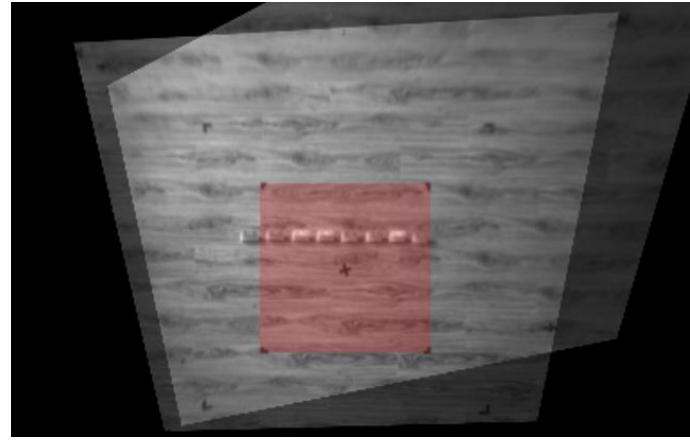


图 5.4 图像1, 图像2融合

5.1.3 确定摄像机坐标

空间中两直线的交点

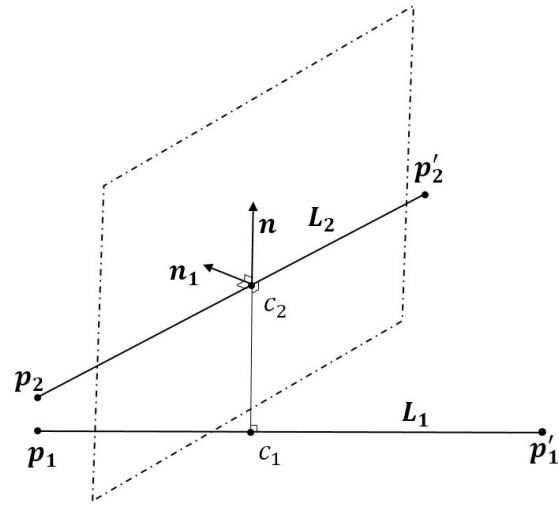


图 5.5 异面直线

对于给定空间中的两条相交线 L_1 和 L_2 , 可以通过联立方程组来求解两直线的交点, 然而, 在实际应用中, 由于测量数据存在噪声与误差, 测得的两直线往往不共面(图5.5), 导致两直线的交点没有精确解。因此需要将问题转化为: 求空间中距离两直线最近的点。该问题的解决方法有多种, 包括最小二乘法, 二分法搜索, 梯度法等, 在本设计中采用立体几何的方法解决该问题:

将空间中的直线以一个点与一个向量来表示，对于 L_1, L_2 :

$$L_1 = p_1 + t_1 d_1$$

$$L_2 = p_2 + t_2 d_2$$

其中 d_1, d_2 为直线的方向向量:

$$d_1 = p'_1 - p_1$$

$$d_2 = p'_2 - p_2$$

计算与 L_1, L_2 垂直的向量 n :

$$n = d_1 \times d_2$$

n 的方向即为连接直线 L_1, L_2 最短线段的方向。计算向量 n_1 :

$$n_1 = d_2 \times n$$

n_1 即为 d_2, n 所在平面的法向量。

以相同的方法计算 n_2 :

$$n_2 = d_1 \times n$$

得 d_1, n 所在平面的法向量。

得到 L_1 上离 L_2 最近的点的坐标 c_1 :

$$c_1 = p_1 + \frac{(p_2 - p_1) \cdot n_2}{d_1 \cdot n_2} d_1$$

同样得 L_2 上离 L_1 最近的点的坐标 c_2 :

$$c_2 = p_2 + \frac{(p_1 - p_2) \cdot n_1}{d_2 \cdot n_1} d_2$$

最后，取线段 $c_1 c_2$ 的中点得到距离 L_1, L_2 最近的点 c :

$$c = \frac{c_1 + c_2}{2}$$

通过观察 c_1, c_2 的计算过程，可以发现只有在两直线平行时，该方法才会出现无解的情况。

计算摄像机坐标

摄像机的空间坐标最少可以由通过两次测量获得：在摄像机可视区域内放置标记点，标记点的三维坐标为预先已知，或能手动测量，将标志点先后在两个不同的位置 p_1, p_2 放置，放置时记录摄像机经过齐次变换得到的标记点在地平面投影的坐标 p'_1, p'_2 ，如（图5.6）。

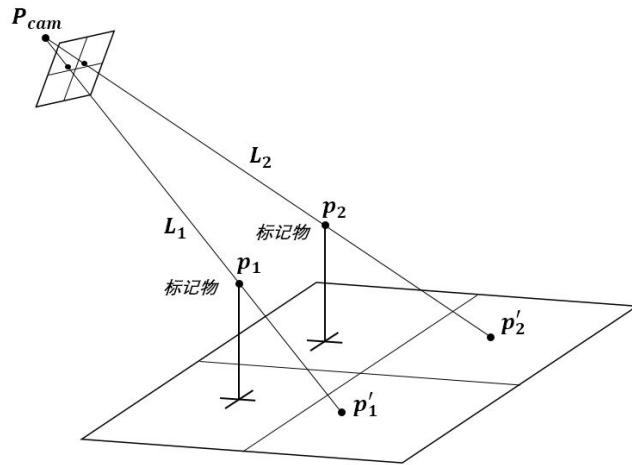


图 5.6 计算摄像机坐标

由以上步骤可以得到两条直线 L_1, L_2 ，利用上述求空间直线交点的方法求出 L_1, L_2 的交点即得摄像机的三维空间坐标 P_{cam} 。摄像机1和摄像机2的坐标确定可以同时进行以加快标定过程，只需令标记点在其公共可视区域内放置。

在实际应用中，若只在两个位置的采样，得到的结果往往误差较大，因此，可以稍作修改，通过2个以上的样本来确定摄像机的坐标。假设采集了 m 个位置的样本，得到了直线 $L_1 \dots L_m$ ，求出所有直线两两之间的交点，取其均值作为摄像机的坐标：

$$P_{cam} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m \text{findIntersection}(L_i, L_j), (m \geq 2)$$

其中函数`findIntersection()`返回直线交点坐标。

5.2 标记点提取

该步骤需要在摄像机返回的图像中识别出标记点的图像坐标，由于采用的是红外摄像机，镜头中的滤光片对非红外波段的电磁波进行了有效滤除，大大简化

了特征提取算法的处理。本设计中标记物的提取相对简单，主要分三部分：

- 图像降噪（二值化）。
- 轮廓检测。
- 特征点定位。

二值化

对图像中的所有像素进行以下操作：

$$I(x, y) = \begin{cases} 255, & I(x, y) > \text{threshold} \\ 0, & \text{else} \end{cases}$$

其中， $I(x, y)$ 为图像坐标中 (x, y) 处的像素的亮度，threshold为二值化阈值。

一般来说二值化可以采用动态阈值以适应更复杂的环境，但实际测试中原始图像噪声非常低，采用固定阈值已经能达到较好的效果，因此本设计中采用固定阈值。通过测试，图像二值化的阈值设置在100左右可以达到较好的效果。

在OpenCV库中，提供了相应的二值化函数：

```
double threshold(InputArray src,
                  OutputArray dst,
                  double thresh,
                  double maxval,
                  int type );
```

提取轮廓

经过对图像二值化后，可以对图像中的标记点的轮廓进行提取，OpenCV中提供了提取轮廓的函数：

```
void findContours(InputOutputArray image,
                  OutputArrayOfArrays contours,
                  OutputArray hierarchy, int mode,
                  int method,
                  Point offset = Point());
```

函数返回了图像中所有的轮廓序列contours，配置参数为：

```
mode = CV_RETR_EXTERNAL, method = CV_CHAIN_APPROX_SIMPLE
```

特征点定位

获取了所有特征点的轮廓（点的序列）后，可以利用特征点的轮廓来确定特征点在图像中的坐标，OpenCV中提供了函数minEnclosingCircle()来寻找一个能包围其所有输入的点的面积最小的圆形：

```
void minEnclosingCircle(InputArray points ,
                        CV_OUT Point2f& center ,
                        CV_OUT float& radius );
```

函数返回了该圆形的中心坐标 c 与半径 r ，圆形的中心即为所要求的特征点在图像的坐标。

提取结果

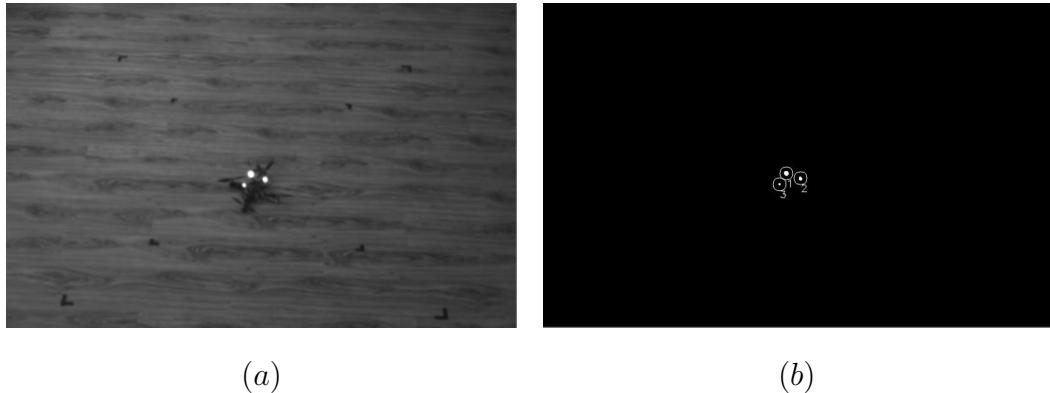


图 5.7 标记点提取结果:(a)原图像;(b)二值化图像;

5.3 确定空间坐标

确定标记物空间坐标的方法可以采用类似确定摄像机坐标(5.1)中的方法，通过计算两个摄像机位置坐标与标记物在地上的投影点坐标连线所形成的交点，即得标记点的空间坐标，具体如下：（图5.8）中 $\mathbf{P}_{cam1}, \mathbf{P}_{cam2}$ 为标定中获得的两个摄像机的空间坐标， $\mathbf{p}'_1, \mathbf{p}'_2$ 分别为标记物相对于摄像机1和摄像机2，在地面上的投影的坐标，可以由（5.1）中提到的单应矩阵获得。通过计算直线 $\overline{P_{cam1}p'_1}$ 和直线 $\overline{P_{cam2}p'_2}$ 的交点，可以得到标记物的空间坐标 \mathbf{p} 。

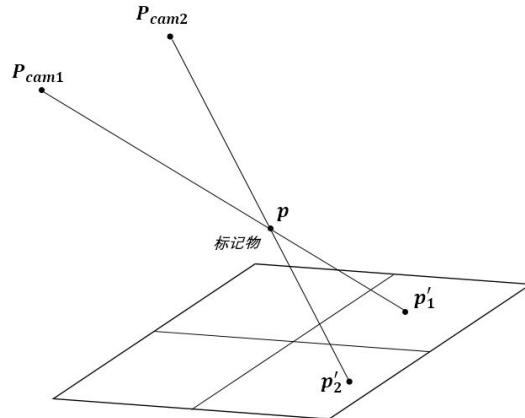


图 5.8 确定标记物空间坐标

5.3.1 立体匹配

当双目摄像机可视区域内只有一个标记点时，可以直接利用上述的方法来确定标记点的空间坐标。然而，如果飞行器上只有一个标记点，室内定位系统只能得到飞行器三维位置坐标，并不能得到飞行器的姿态信息，在本设计中，飞行器上安装的标记物有三个标记点以确定飞行器的位置与姿态。图像中有多个特征点，需要涉及到立体匹配问题，立体匹配问题是立体视觉中比较重要的一个环节，主要任务就是对于两个摄像机看到的同一个点，在两个图像中建立起联系。一般来说，在立体视觉中会用极线约束配合其他特征匹配的算法进行立体匹配。通过极线约束（图5.9），排除了大部分非对应的点，可以把特征点的搜索范围从2维将为1维，大大提高了搜索的效率。

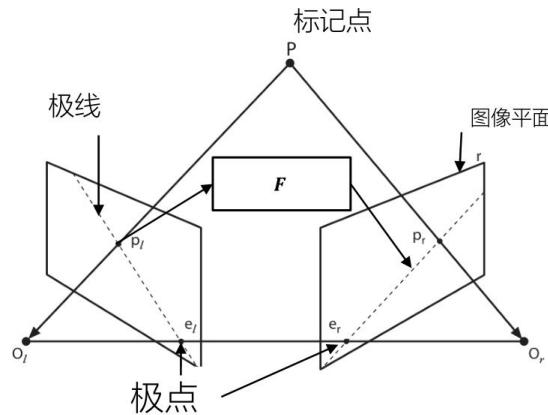


图 5.9 极线约束

利用对极几何，可以将图像中的一点通过基本矩阵 \mathbf{F} 映射到另一幅图像中的一条直线上，并且其对应点必在这条直线上。

在实验中发现即使基于极线约束缩小了搜索范围，但在两极线距离很近的情况下，特征匹配这一步中成功率并不高，实验中采用了基于直方图相关性，亮度差的平方和（SSD）等算法，成功率大概在50% ~ 60%，总体效果并不理想。

基于标记物发光面积进行匹配

为了解决这个问题，采取了另外可靠性更强的方法，实验结果表明，该方法鲁棒性较高：在（5.2）特征点定位这一步中，不但得到了标记点在图像中的坐标 \mathbf{c} ，还得到了最小包围圆的半径 r ，可以从这个半径入手进行特征匹配。实验中发现，该包围圆的半径与红外主动标记物的亮度成正相关关系，通过调整标记物的亮度，使每个标记物的亮度不一致，即可根据最小包围圆的半径来进行立体匹配。具体实现方法很简单，即对每个摄像机图像中的标记物根据最小包围圆的半径 r 进行排序，并编号，即可将两幅图像中的点对应起来。在实验中，该方法匹配的成功率接近100%，并且可以通过 r 来筛选掉一些可能的干扰，算法的鲁棒性非常高。在本设计的应用当中，本方法能提高非常可靠的定位数据。

然而，该方法也有一个比较明显的缺点，即当标记物过多时，匹配的成功率会下降，因为每一个标记物都要分一个亮度等级，并且亮度差异小会影响到匹配的成功率。但在本设计的应用中，依然是一个较为可靠的方法。

5.3.2 确定空间坐标

标记物的空间坐标

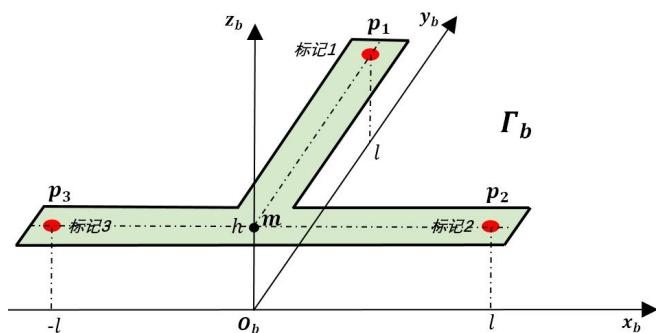


图 5.10 机体上的标记

三个标记点在机体坐标上如（图5.10）所示。图像经过上述几个步骤的处理后，已经可以得到飞行器上三个标记点的三维空间坐标。最后一步就是确定整个标记物的空间坐标，本设计中取标记点 p_2, p_3 的中点 m 作为标记物的中心，以表示标记物的空间位置，其在 Γ_b 中的坐标已知为 $P_{b,m} = \begin{pmatrix} 0 & 0 & h \end{pmatrix}^T$ ， m 在 Γ_w 中的坐标 $P_{w,m}$ 可以如下计算：

$$P_{w,m} = \frac{p_2 + p_3}{2}$$

飞行器的空间坐标

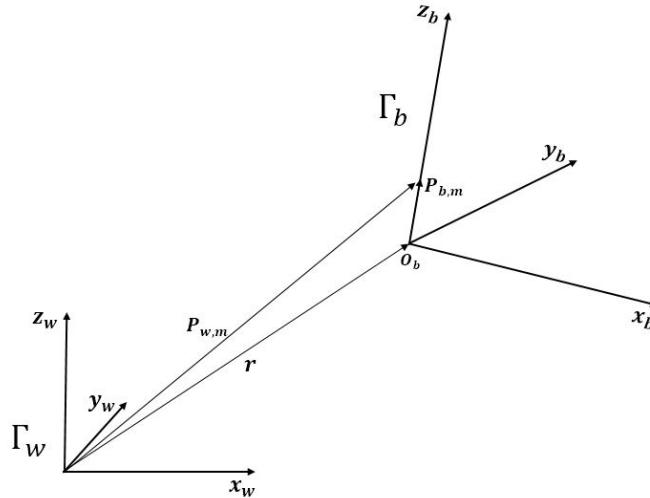


图 5.11 由标记物坐标计算飞行器位移

明显地， $P_{w,m}$ 不等于机体坐标原点 O_b 在 Γ_w 中的坐标 r 。此步骤，需要将 $P_{w,m}$ 转换为 r ，如（图5.11）所示，要算 r 只需要计算 $P_{w,m} - P_{b,m}$ ，但是，图中的 $P_{b,m}$ 为 Γ_b 中的坐标，因此，需要把 $P_{b,m}$ 投影到 Γ_w 中，再进行计算，利用旋转矩阵的特性（式3.4）得：

$$r = P_{w,m} - {}^w R_b P_{b,m}$$

5.4 确定姿态

定位系统测得标记点1，标记点2，以及标记点2和标记点3的中点在 Γ_w 中的坐标 p_1, p_2 和 m ，向量 $\hat{m}p_1, \hat{m}p_2$ 为对应的单位方向向量，记为 \hat{d}_1, \hat{d}_2 ，其在 Γ_b 中

坐标分别为 $\begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$ 和 $\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$ 利用旋转矩阵可得：

$$\hat{d}_1 = {}^w\mathbf{R}_b \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} c\psi s\theta s\varphi - s\psi c\varphi \\ s\psi s\theta s\varphi + c\psi c\varphi \\ c\theta s\varphi \end{pmatrix} \quad (5.3)$$

$$\hat{d}_2 = {}^w\mathbf{R}_b \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} c\psi c\theta \\ s\psi c\theta \\ -s\theta \end{pmatrix} \quad (5.4)$$

观测以上两式可得定位系统测得的欧拉角 Ω_{cam} ：

$$\varphi_{cam} = \arcsin\left(\frac{\hat{d}_{1,z}}{\cos(\theta)}\right) \quad (5.5)$$

$$\theta_{cam} = -\arcsin(\hat{d}_{2,z}) \quad (5.6)$$

$$\psi_{cam} = \arctan\left(\frac{\hat{d}_{2,y}}{\hat{d}_{2,x}}\right) \quad (5.7)$$

在实际应用中反正切函数应该使用 $\text{atan2}(y, x)$ 函数，以使输出角度在 $(-\pi, \pi)$ 范围。同时，注意到 θ 的范围为 $(-\frac{\pi}{2}, \frac{\pi}{2})$ 而非 $(-\pi, \pi)$ ，但实际上当飞行器横滚角旋转90°后，标记物已被遮挡，因此在实际应用中影响不大。

5.5 测试结果

本节中给出了室内定位系统的校正结果和测量数据与实际数据的差异，并给出了定位系统的覆盖区域。

5.5.1 校正结果

单应矩阵：

$$\mathbf{H}_1 = \begin{pmatrix} 0.0043 & 0.0002 & -1.6549 \\ 0.0002 & -0.0058 & 1.3484 \\ -0.0000 & 0.0001 & 1.0000 \end{pmatrix}$$

$$\mathbf{H}_2 = \begin{pmatrix} 0.0042 & -0.0004 & -1.8606 \\ 0.0007 & -0.0054 & 1.2387 \\ -0.0003 & 0.0009 & 1.0000 \end{pmatrix}$$

摄像机位置坐标：

$$P_{cam,1} = \begin{pmatrix} 0.0112 \\ -2.8296 \\ 3.0649 \end{pmatrix} \quad P_{cam,2} = \begin{pmatrix} -1.0143 \\ -2.8186 \\ 3.0496 \end{pmatrix}$$

5.5.2 位移测量精度

在实验室中，利用卷尺测量实际位置，与定位系统测得的数据进行比较，大概估计定位系统的误差范围（表5.1）～（表5.3）。

	x 轴, $\sigma_x = 0.0081\text{m}$					
手工测量值(m)	-1.000	-0.600	-0.200	0.200	0.600	1.000
定位系统测量值(m)	-1.003	-0.610	-0.207	0.206	0.598	1.011

表 5.1 定位系统 x 轴测量结果

	y 轴, $\sigma_y = 0.0268\text{m}$					
手工测量值(m)	-1.000	-0.600	-0.200	0.200	0.600	1.000
定位系统测量值(m)	-1.007	-0.573	-0.174	0.222	0.658	0.984

表 5.2 定位系统 y 轴测量结果

	z 轴, $\sigma_z = 0.0104\text{m}$					
手工测量值(m)	0.000	0.200	0.400	0.600	0.800	1.000
定位系统测量值(m)	-0.011	0.190	0.413	0.597	0.799	0.983

表 5.3 定位系统 z 轴测量结果

定位系统的精度依赖于校正过程，由测试结果可以看出定位的精度大概为1cm～2cm。并且根据上述定位原理可知，距离摄像机越近的区域，定位精度越高。

5.5.3 摄像机覆盖区域

（图5.12）显示了该定位系统覆盖的空间区域。

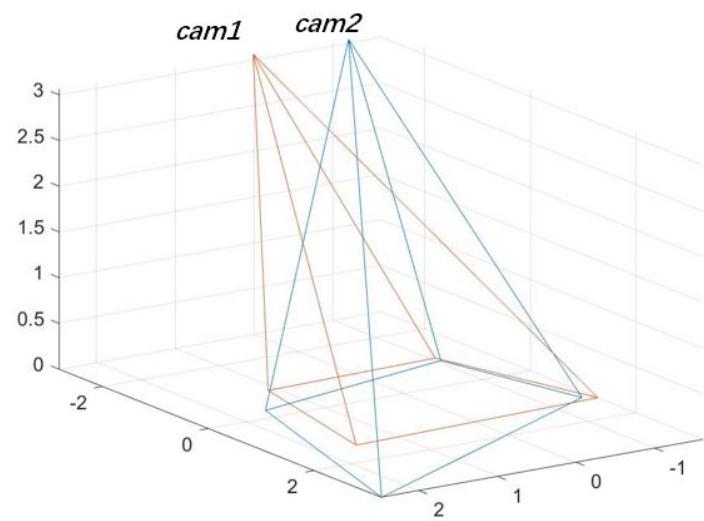


图 5.12 摄像机覆盖空间区域

第六章 路径规划

路径规划在机器人进行自主控制时尤为重要。文献^[13]中，对飞行器的轨迹生成方法进行了研究。文献^[30]中，考虑了飞行器进行高速翻滚时的路径规划问题。文献^[31]中，通过对于人类的运动轨迹观察，发现人类运动中符合加速度导数平方的积分最小的运动轨迹。文献^[32]中发现，人类肢体的运动轨迹倾向于最小化力矩导数平方的积分。在四轴飞行器中，电机是飞行器的执行机构，在飞行器进行路径跟踪时，希望能够最小化电机的输出力矩，以让飞行器尽量平稳地飞行。

一般来说，轨迹跟踪最简单的方法是依次向位移控制器输入目标位移，判断飞行器实际位移与目标位移的距离，当距离小于一定值后切换到下一个目标。该方法虽然能让飞行器依次经过所有的目标点，但是在跟踪过程中其路径的位移，速度，加速度并不连续，会使飞行器的动作非常不流畅，跟踪误差大，而且难以实现高速的路径跟踪。

本章根据飞行器位移与电机输出的关系，讨论位移，速度，加速度以及加速度的导数均连续的路径的生成方法，并且构建代价函数，令所生成的路径满足一定的最优条件和约束条件。由于飞行器的三维路径互相独立，以下介绍的路径生成方法均只针对一维的路径，其他两个维度的路径可以使用完全相同的方法生成。文中讨论了两个生成最优路径的方法，分别有其优缺点。

6.1 欧拉-拉格朗日方程

对于给定的两个时间点($t = a, b$)，设路径为函数 $f(t)$ ，有以下代价函数：

$$J(f) = \int_a^b \mathcal{L}(t, f, \dot{f}, \ddot{f}, \dots, f^{(n)}) dt \quad (6.1)$$

其中 \mathcal{L} 为 t, f 以及 f 的各阶导数 $f^{(n)}$ 的函数。

以上代价函数 J 取得极值的必要条件是 \mathcal{L} 符合欧拉-拉格朗日方程：

$$\frac{\partial \mathcal{L}}{\partial f} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{f}} \right) + \frac{d^2}{dt^2} \left(\frac{\partial \mathcal{L}}{\partial \ddot{f}} \right) - \dots + (-1)^n \frac{d^n}{dt^n} \left(\frac{\partial \mathcal{L}}{\partial f^{(n)}} \right) = 0 \quad (6.2)$$

为了给飞行器生成最优的路径，首先应该选取一个适当的代价函数。按照(3.2)中设计的控制器，由(式3.13)，(式3.14)，(式3.15)以及(式3.19)，取小角度近

似($\theta \approx 0, \varphi \approx 0$)，可以得出以下结论：

$$\left. \begin{array}{l} \ddot{\mathbf{r}}_{des} \propto \boldsymbol{\Omega}_{des} \\ \ddot{\boldsymbol{\Omega}}_{des} \propto \boldsymbol{\mathcal{M}}_{des} \end{array} \right\} \Rightarrow \mathbf{r}_{des}^{(4)} \propto \boldsymbol{\mathcal{M}}_{des} \quad (6.3)$$

其中 \propto 表示成正比例关系。也就是说进行位移控制时，飞行器的力矩与给定的路径的四阶导成正比例关系，为了使飞行器进行路径跟踪时动作流畅，跟踪误差小，飞行姿态的抖动小，生成的路径应该要使控制器力矩输出尽量小。因此，可以选择以下代价函数：

$$\begin{aligned} J(f) &= \int_a^b \mathcal{L} dt \\ \mathcal{L} &= (f^{(4)})^2 \end{aligned} \quad (6.4)$$

当 J 取得最小值时， \mathcal{L} 应该服从欧拉-拉格朗日方程，将 \mathcal{L} 代入（式6.2）得：

$$f^{(8)} = 0 \quad (6.5)$$

即生成的最优路径所应该满足的条件。

解以上微分方程可得7阶多项式：

$$f(t) = c_1 + c_2 t + c_3 t^2 + c_4 t^3 + c_5 t^4 + c_6 t^5 + c_7 t^6 + c_8 t^7 \quad (6.6)$$

其中 c_1, c_2, \dots, c_8 为常数。

6.2 分段多项式插值

为了使用（式6.6）生成路径，可以采用多段多项式连接给定的起点，终点和中间点（图6.1），使每一小段路径均满足最小化代价函数的条件。

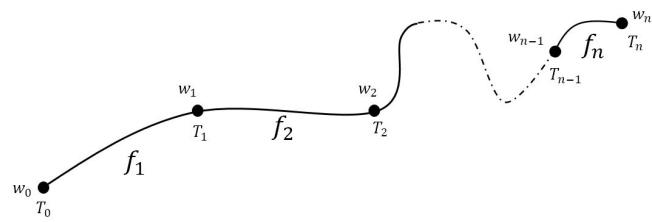


图 6.1 分段多项式插值

生成路径时，给定路径经过的起始位置，中间位置，终点位置序列 $w_0, w_1, w_2, \dots, w_n$ ，以及其对应的时间点 $T_0, T_1, T_2, \dots, T_n$ 。

生成的路径为如下形式：

$$f(t) = \begin{cases} f_1 = \sum_{j=1}^8 c_{1,j} t^{j-1}, & T_0 \leq t < T_1 \\ f_2 = \sum_{j=1}^8 c_{2,j} t^{j-1}, & T_1 \leq t < T_2 \\ \vdots & \vdots \\ f_n = \sum_{j=1}^8 c_{n,j} t^{j-1}, & T_{n-1} \leq t < T_n \end{cases} \quad (6.7)$$

可以看出上式含有 $8n$ 个未知常数，可以通过给定边界条件，构建线性方程组来求解未知常数：

首先，可以由已知的 $w_0, w_1, w_2, \dots, w_n$ 和 $T_0, T_1, T_2, \dots, T_n$ 得到以下 $2n$ 个边界条件：

$$\begin{aligned} f_i(T_{i-1}) &= w_{i-1}, \quad (i = 1, 2, \dots, n) \\ f_i(T_i) &= w_i, \quad (i = 1, 2, \dots, n) \end{aligned} \quad (6.8)$$

再通过令路径在起点和终点时的 $1 \sim 3$ 阶导数为零：

$$f_i^{(k)}(T_i) = 0, \quad (k = 1, 2, 3; i = 0, n) \quad (6.9)$$

得到6个边界条件。

到此，还需要有 $8n - 2n - 6 = 6(n - 1)$ 个边界条件。由于路径在各个中间点前后是连续的，因此可以利用路径在中间点时的各阶导数建立边界条件。令路径在中间点的 $1 \sim 6$ 阶导数连续，可以获得 $6(n - 1)$ 个边界条件：

$$f_i^{(k)}(T_i) = f_{i+1}^{(k)}(T_i), \quad (k = 1, 2, \dots, 6; i = 1, 2, \dots, n - 1) \quad (6.10)$$

由（式6.8），（式6.9），（式6.10）一共获得 $8n$ 个边界条件，构成线性方程组，即可得出每一段多项式的系数。

采用上述方法，可以生成经过任意点的最优路径，路径生成速度较快，并且基本上不会出现无解的情况。然而，上述方法也存在缺点，就是给定的路径约束都是等式约束，即通过给定中间点的方式来给路径进行约束。在某些情况下，比如，希望飞行器穿过一个狭窄的隧道，仅仅依靠给定的中间点并不能保证飞行器的路径不与障碍物发生碰撞，因为在两个中间点之间的路径，没有加入任何的约束。

6.3 二次规划 (QP)

在飞行器穿越障碍物时，若障碍物的位置已知，一条合理的路径应该能够在障碍物之间流畅地穿越，并且与障碍物保持一定的距离，简单来说就是希望生成

的路径能控制在某一个空间范围内。这种情况下，可以看作是给生成的路径加入了不等式约束，而这种带不等式约束的优化问题，可以使用二次规划（QP）的方法来解决。

二次规划问题可以归结为以下问题：

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{f}^T \mathbf{x} \quad (6.11)$$

\mathbf{x} 满足以下边界条件：

$$\begin{aligned} \mathbf{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{A}_{eq}\mathbf{x} &= \mathbf{b}_{eq} \\ l\mathbf{b} &\leq \mathbf{x} \leq u\mathbf{b} \end{aligned}$$

即寻找满足条件的 \mathbf{x} ，使 $\frac{1}{2}\mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{f}^T \mathbf{x}$ 最小。

类似上一节，给定路径的固定点之间可以用多项式进行连接，但是在使用QP方法进行路径规划中，并不需要多项式有特定的阶数，一般来说若多项式阶数太低在加入不等式约束后问题可能无解。阶数太高会导致运算时间过长，或者也会引起求解器不稳定。在本设计中使用了14阶多项式：

$$f(t) = \begin{cases} f_1 = \sum_{j=1}^{15} c_{1,j} t^{j-1}, & T_0 \leq t < T_1 \\ f_2 = \sum_{j=1}^{15} c_{2,j} t^{j-1}, & T_1 \leq t < T_2 \\ \vdots & \vdots \\ f_n = \sum_{j=1}^{15} c_{n,j} t^{j-1}, & T_{n-1} \leq t < T_n \end{cases} \quad (6.12)$$

应用QP来求解最优路径问题，令路径在起点和终点处 $1 \sim 4$ 阶导数为0，中间点处 $1 \sim 4$ 阶导数连续，得到以下优化问题：

$$\min_f \int_{T_0}^{T_n} \left\{ \left(\frac{d^4 f_1}{dt^4} \right)^2 + \left(\frac{d^4 f_2}{dt^4} \right)^2 + \cdots + \left(\frac{d^4 f_n}{dt^4} \right)^2 \right\} dt \quad (6.13)$$

f 服从边界条件：

$$\begin{aligned} f_i(T_{i-1}) &= w_{i-1}, \quad (i = 1, 2, \dots, n) \\ f_i(T_i) &= w_i, \quad (i = 1, 2, \dots, n) \\ f_i^{(k)}(T_i) &= 0, \quad (k = 1, 2, 3, 4; i = 0, n) \\ f_i^{(k)}(T_i) &= f_{i+1}^{(k)}(T_i), \quad (k = 1, 2, 3, 4; i = 1, 2, \dots, n-1) \end{aligned}$$

令（式6.11）中的 x 为路径多项式的系数：

$$\boldsymbol{x} = \begin{pmatrix} c_{1,1} \\ c_{1,2} \\ \vdots \\ c_{1,15} \\ \vdots \\ c_{n,1} \\ c_{n,2} \\ \vdots \\ c_{n,15} \end{pmatrix}$$

根据 f 的边界条件，可以构造矩阵 \mathbf{H} , \mathbf{A}_{eq} 和 \mathbf{b}_{eq} ，并令 \mathbf{f} 为 $n \times 1$ 全零列向量，求解该QP问题。若需要在某段路径限制在某个范围内，则可对相应的多项式（例如 f_m ）加入不等式约束加入不等式约束：

$$lb \leq f_m(t) \leq ub, \quad T_{m-1} \leq t \leq T_m$$

实际中若对 $f_m(t)$ 中的所有时间点进行不等式约束，QP求解运算量将会很大，可以通过取 f_m 中等间隔的 $N + 1$ 个时间点进行约束：

$$lb \leq f_m(T_{m-1} + \frac{n}{N}(T_m - T_{m-1})) \leq ub, \quad (n = 0, 1, 2, \dots, N)$$

根据该不等式约束，可以构造相应的 \mathbf{A}, \mathbf{b} 矩阵。

以上介绍的两种方法都是基于用多项式进行插值，方法一直接利用边界条件构成线性方程组，运算量少，适应面较广，但不能进行不等式约束。利用QP方法进行路径生成，则可以进行不等式约束，适用于生成使飞行器穿越障碍物的最优路径，但是运算量较大，并且在实际应用中，当中间点个数较多时，会出现无解的情况，相对于方法一来说鲁棒性较低。

6.4 路径生成结果

路径生成过程在MATLAB®中进行，给定起点，终点与中间点序列为：

$$\boldsymbol{w} = \begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \\ \boldsymbol{w}_z \end{pmatrix} = \begin{pmatrix} -0.75 & -0.5 & 0.5 & 0.75 \\ -0.5 & 0 & 0 & 0.5 \\ 0.2 & 0.5 & 0.5 & 0.2 \end{pmatrix}$$

求解3段7阶多项式中的24个系数后，生成路径如图（图6.2）。相应的三轴位移，速度，加速度如（图6.3）所示。

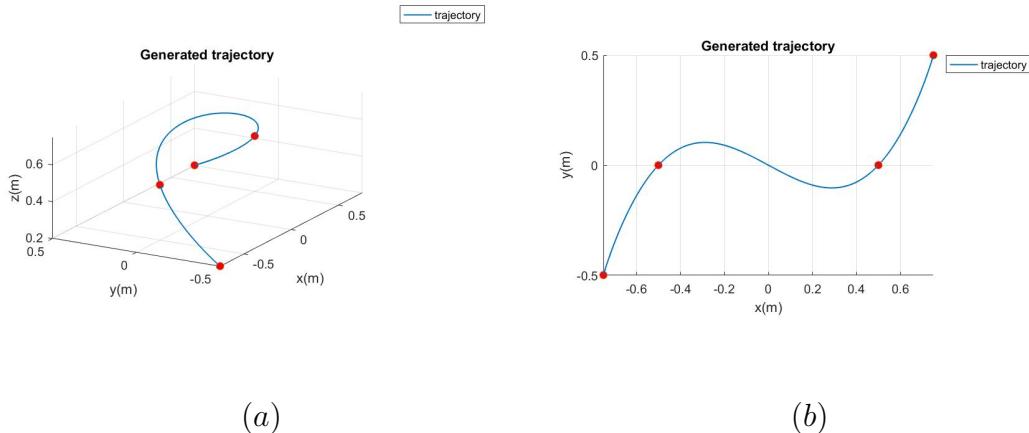


图 6.2 使用分段多项式插值生成的路径,(a)三维视图;(b)xy平面轨迹

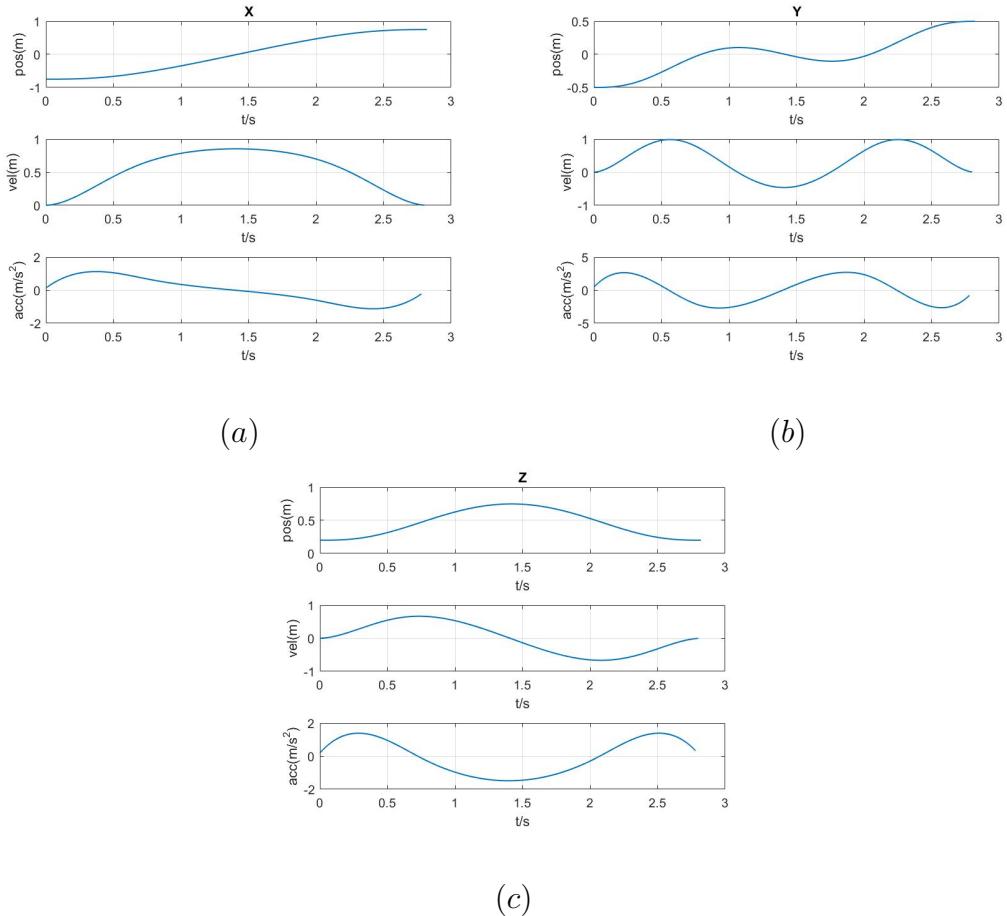


图 6.3 (a)x轴数据;(b)y轴数据;(c)z轴数据;

应用QP方法，利用MATLAB[®]中的quadprog()函数，生成未加不等式约束的路径如（图6.4）所示。

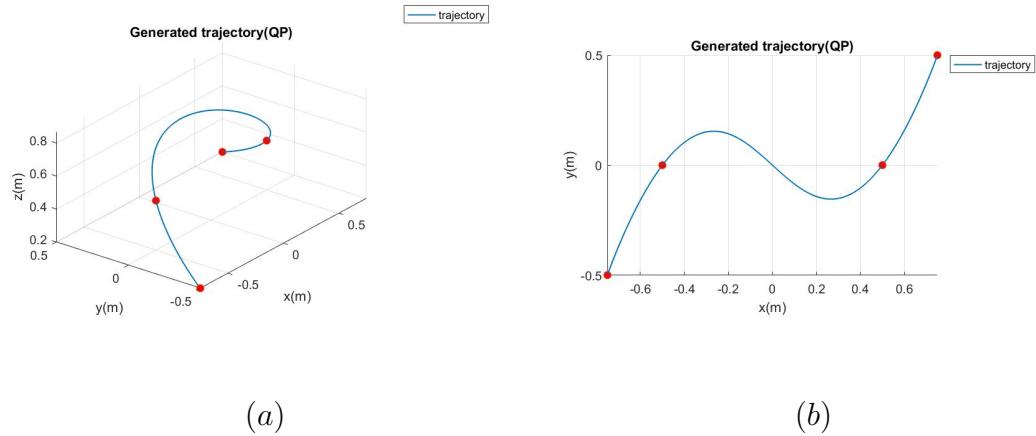


图 6.4 使用QP生成的路径(未加入不等式约束),(a)三维视图;(b)xy平面轨迹

加入不等式约束，令第二段路径中：

$$-0.01m \leq y \leq 0.01m$$

$$-0.49m \leq z \leq 0.51m$$

得到（图6.5）的结果。

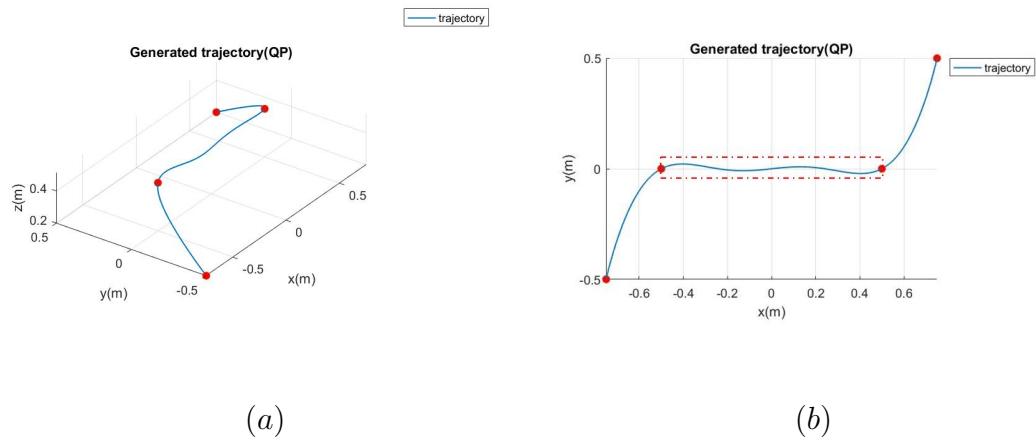


图 6.5 使用QP生成的路径(加入不等式约束),(a)三维视图;(b)xy平面轨迹

第七章 结束语

7.1 总结

本文详细介绍了一个实现飞行机器人室内自主飞行的具体设计与实现方案。

第二章中介绍了组成该系统的具体硬件需求，以及其性能配置，包括飞行器，与室内定位系统两大部分。通过调查，分析，室内定位系统采用了定位精度高，更新频率高的光学运动捕捉系统方案。目前，商用的光学运动捕捉系统价格普遍较高，一般的系统价格高达数万元到数十万元人民币，一般只有较大型的研究所或商业机构才能够配备。而本文提供的室内定位系统方案，成本相对较低，整套系统的价格能限制在5000元人民币以内，并且有较好的定位性能。配合设计中的飞行器系统，构成了一个较理想的飞行机器人实验平台。

第三章中，详细介绍了四轴飞行器的动力学模型，推导了飞行器的状态方程，设计了飞行器姿态与位移控制的控制律，并通过在MATLAB[®]软件中仿真实验，得到近似的控制器参数后，在实际飞行器中验证，取得较好的控制效果，在大角度控制时飞行器的姿态控制器依然性能良好。在跟踪给定飞行路径中，跟踪误差较少。在实验中，飞行器进行路径跟踪时最大加速度能达到 7m/s^2 ，最大倾斜角度超过 60° 。

第四章，讨论了飞行器的状态估计问题，考虑到飞行器控制器的处理能力，以及高采样率的需要，设计了次优滤波器对飞行器的姿态和位移进行估计，同时考虑了测量延迟带来的影响，设计了补偿方案。实际中取得了较好的估计效果。

第五章，讨论了室内定位系统的设计与实现原理，通过在飞行器上安装主动标记物，利用基于双目红外摄像机的光学定位系统获得飞行器的位移与姿态信息，测量的数据以40Hz的频率通过蓝牙发送给飞行器。经过实际测量，定位系统的精度大约为1cm，噪声标准差大约为 $\sigma = 0.012\text{m}$ 。

第六章，研究了飞行器控制中的上层控制算法，路径规划。通过给定的起始位置，终点位置，以及中间点的位置，以使飞行器产生以最小抖动为目标，光滑，合理的的飞行轨迹。文中提供了两种生成方法，一种基于求解线性方程组来得到飞行轨迹多项式系数，该方法运算量少，适用面广，但是不能进行不等式约束。另一种方法基于二次规划(QP)来求解路径多项式的系数，通过加入不等式约束，

能够让飞行路径限制在某个空间区域内，适用于飞行器需要穿越障碍物的情况，但运算时间相对方法一较长，同时当路径的中间点数较多时，会出现求解器不稳定，导致无解的情况。

最后，为了验证飞行器以及室内定位系统的综合性能，尝试利用飞行器进行乐器演奏（图7.1），总体结果比较理想。



图 7.1 飞行器进行演奏

7.2 展望

目前，人们对四轴飞行器已经不陌生，近几年与四轴飞行器相关的公司大量涌现。然而，目前消费级的多轴飞行器基本上只适用于室外环境，即使能够在室内飞行，也需要专人来手动操作才能完成任务。目前的消费级小型四轴飞行器中，基本上还没有应用到同步建图与定位(SLAM)技术或者动态路径规划等相对高级的机器人控制算法，要求飞行器能在室内环境中自主完成任务就更难了。然而，室内自主飞行机器人，在未来有广阔的应用前景，比如，在灾区或恶劣地形的搜救中，可以发挥四轴飞行器能够悬停，穿越能力强的优点，配合工作人员高效地完成相关的任务；在工厂中利用飞行器来高效搬运货物；在反恐行动中，利用自主飞行的机器人对未知内部结构的建筑物建立三维室内地图，协助军队高效完成任务。

本设计中的室内定位系统一方面为飞行器提供了精确可靠的反馈信息，但另一方面它限制了飞行器的飞行范围，这也是光学运动捕捉系统的优点和缺点。但是，对于研究者来说，该定位系统为飞行机器人的实验提供了良好的平台，以该

平台为基础，可以进行对飞行器控制，路径规划，团队协作等问题进行深入研究，而且研究的结果往往也适用于使用其他定位系统进行定位的情况下。目前，本设计中控制的是一个飞行器，当能加入更多的飞行器进行团队合作的时候，飞行器就能完成更多更复杂的任务了，例如，可以配合机械臂进行模拟的物资运输，或者合作搭建建筑物等。当然，这样涉及的问题就更多了，例如，对于飞行控制器：飞行器在运送不同的物体时，如何实时估计飞行器的物理模型参数，在线调整控制器的参数；当飞行器之间路径有冲突，飞行器如何自主调整最优路径，以避免发生碰撞。对于室内定位系统：当区域内标记点数量较多的时候如何进行快速准确的立体匹配，等等。

随着机器人控制技术和人工智能技术的进一步深入研究与发展，在我们将来的生活中一定会出现更多各种形式的机器人，他们将会服务于人类，使我们的生活变得更加便捷。

致 谢

在进行毕业设计这将近半年的时间里我获益匪浅，本毕业设计题目具有较高的综合性，使我对大学四年里所学的一些理论知识有了更深入的了解，同时也大大提高了我的探索分析能力。值此论文完成之际，我要感谢在我的大学生涯中尤其在我完成毕业设计期间给予了我帮助的所有人。

我尤其要感谢的我的指导老师王新怀老师，他在大学期间里给我提供了许多的帮助和支持，给我提供了许多动手实践，提高我自身能力的机会。在我完成毕业设计期间，王老师给我提供了良好的实验场地与充足实验设备。在我遇到困难的时候，与我分析探讨了相关问题，并给予了我许多具有指导性的意见和许多宝贵的经验。

同时，我也要感谢实验室里的同学，在我搭建相关实验场地时给予了我许多帮助，在我遇到难题时，分享他们的见解。也感谢我的室友，给我提供了良好的生活环境。

另外，我还要感谢我的家人，他们在我的大学生涯里给我提供了许多物质上与精神上的支持。在我遇到困惑时给予我引导，在我作出选择时给予我支持。

附录 A 欧拉角与旋转矩阵

A.1 欧拉角

对于三维空间中一个坐标系一次任意的旋转，可以由按次序的三次独立旋转来表示（图A.1），本文中使用的旋转次序为 $Z - Y - X$ 。

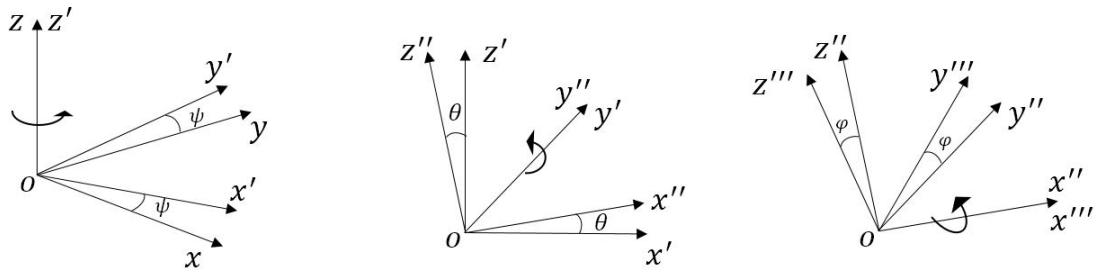


图 A.1 欧拉角

旋转前的坐标系为 $o-xyz$ ，第一次绕 z 轴旋转角度 ψ ，得到坐标系 $o-x'y'z'$ ， ψ 称为航偏角。第二次，绕 y' 轴旋转角度 θ ，得到坐标系 $o-x''y''z''$ ， θ 称为俯仰角。第三次，绕 x'' 轴旋转角度 φ ，得到最终的坐标系 $o-x'''y'''z'''$ ， φ 称为横滚角。

A.2 旋转矩阵

首先，考虑二维平面中的旋转，如（图A.2）：设点 p 在坐标 $o-xy$ 中的坐标

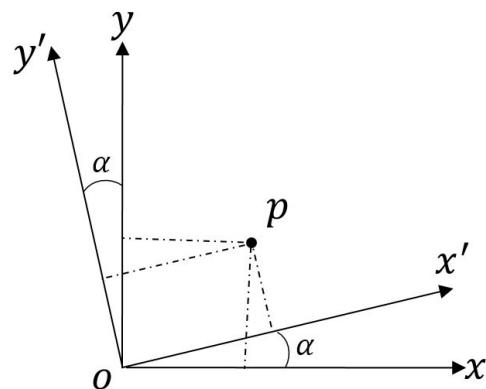


图 A.2 二维旋转

为 $\begin{pmatrix} p_x & p_y \end{pmatrix}^T$, 在 $o - x'y'$ 中的坐标为 $\begin{pmatrix} p'_x & p'_y \end{pmatrix}^T$ 。根据几何关系可以得到:

$$\begin{pmatrix} p'_x \\ p'_y \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

表示将点 \mathbf{p} 从坐标系 $o - xy$ 投影到坐标系 $o - x'y'$ 中。其中:

$$\begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} = \mathbf{R}$$

称为旋转矩阵，并且旋转矩阵为正交矩阵有:

$$\mathbf{R}^{-1} = \mathbf{R}^T, |\mathbf{R}| = 1$$

根据欧拉角的定义，三次旋转中每次旋转都可以视为一次二维旋转，因此，一共可以得到三个旋转矩阵:

$$\mathbf{R}_\psi = \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_\theta = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix}$$

$$\mathbf{R}_\varphi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

可以将 $o - xyz$ 中的任意一点 \mathbf{p} 投影到 $o - x'''y'''z'''$ 中:

$$\begin{pmatrix} p'''_x \\ p'''_y \\ p'''_z \end{pmatrix} = \mathbf{R}_\varphi \mathbf{R}_\theta \mathbf{R}_\psi \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$$

若以 $o - xyz$ 为参考坐标系 Γ_w , $o - x'''y'''z'''$ 为机体坐标系 Γ_b , ${}^b\mathbf{R}_w$ 表示 Γ_w 到 Γ_b 的旋转, 得:

$${}^b\mathbf{R}_w = \mathbf{R}_\varphi \mathbf{R}_\theta \mathbf{R}_\psi = \begin{pmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ c\psi s\theta s\varphi - s\psi c\varphi & s\psi s\theta s\varphi + c\psi c\varphi & c\theta s\varphi \\ c\psi s\theta c\varphi + s\psi s\varphi & s\psi s\theta c\varphi - c\psi s\varphi & c\theta c\varphi \end{pmatrix}$$

其中s和c分别表示sin和cos。

如果要进行相反的旋转，由 Γ_b 旋转到 Γ_w ，只需利用旋转矩阵的性质：

$${}^w\boldsymbol{R}_b = ({}^b\boldsymbol{R}_w)^{-1} = ({}^b\boldsymbol{R}_w)^T$$

附录 B 欧拉角微分与机体角速度的关系

由（附录A）中的欧拉角定义和旋转矩阵的性质可以得到：

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} \dot{\varphi} \\ 0 \\ 0 \end{pmatrix} + \mathbf{R}_\varphi \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \mathbf{R}_\varphi \mathbf{R}_\theta \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}$$

进一步计算可得：

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\varphi) & \cos(\theta)\sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\theta)\sin(\varphi) \end{pmatrix} \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (\text{B.1})$$

以及：

$$\begin{aligned} \begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\varphi) & \cos(\theta)\sin(\varphi) \\ 0 & -\sin(\varphi) & \cos(\theta)\sin(\varphi) \end{pmatrix}^{-1} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \\ &= \begin{pmatrix} 1 & \tan(\theta)\sin(\varphi) & \tan(\theta)\cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sec(\theta)\sin(\varphi) & \sec(\theta)\cos(\varphi) \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \end{aligned}$$

需要注意的是，当 $\theta = \pm 90^\circ$ 时，（式B.1）中的矩阵不可逆，在实际应用中需要做相应的处理。

附录 C 由加速度计数据计算欧拉角

在加速计只受重力加速度影响，忽略外力干扰的假设下，利用（附录A）中的旋转矩阵以及重力加速度常量 $\begin{pmatrix} 0 & 0 & -g \end{pmatrix}^T$ ，可以得到：

$$\begin{pmatrix} acc_{meas,x} \\ acc_{meas,y} \\ acc_{meas,z} \end{pmatrix} = {}^b\mathbf{R}_w \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix}$$

即

$$\begin{pmatrix} acc_{meas,x} \\ acc_{meas,y} \\ acc_{meas,z} \end{pmatrix} = \begin{pmatrix} c\psi c\theta & s\psi c\theta & -s\theta \\ c\psi s\theta s\varphi - s\psi c\varphi & s\psi s\theta s\varphi + c\psi c\varphi & c\theta s\varphi \\ c\psi s\theta c\varphi + s\psi s\varphi & s\psi s\theta c\varphi - c\psi s\varphi & c\theta c\varphi \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix}$$

其中 acc_{meas} 为加速度计的测量值。

简化得：

$$\begin{pmatrix} acc_{meas,x} \\ acc_{meas,y} \\ acc_{meas,z} \end{pmatrix} = g \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \sin(\varphi) \\ -\cos(\theta) \cos(\varphi) \end{pmatrix} \quad (C.1)$$

由（式C.1）可以解得欧拉角：

$$\varphi = \tan^{-1}\left(\frac{acc_{meas,y}}{acc_{meas,z}}\right)$$

$$\theta = -\sin^{-1}\left(\frac{acc_{meas,x}}{g}\right)$$

或

$$\theta = \tan^{-1}\left(\frac{acc_{meas,x}}{\sqrt{acc_{meas,y}^2 + acc_{meas,z}^2}}\right)$$

参考文献

- [1] “Breguet Richet Gyroplane.” https://en.wikipedia.org/wiki/Breguet-Richet_Gyroplane.
- [2] “Microelectromechanical Systems.” https://en.wikipedia.org/wiki/Microelectromechanical_systems.
- [3] “Parrot AR.Drone.” <http://ardrone.parrot.com>.
- [4] “DJI,Inc.” http://www.dji.com.
- [5] “DJI,Phantom.” <http://www.dji.com/product/phantom-4>.
- [6] “Simultaneous Localization And Mapping.” https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping.
- [7] “Robots that fly and cooperate.” https://www.ted.com/talks/vijay_kumar_robots_that_fly_and_cooperate.
- [8] “Vicon Motion Systems,Ltd.” http://www.vicon.com.
- [9] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor,” *Robotics:Science and Systems*, vol. 1, June 2013.
- [10] S. Tang and V. Kumar, “Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload,” *IEEE International Conference on Robotics and Automation(ICRA)*, pp. 2216–2222, 2015.
- [11] G. Loianno and a. V. K. Justin Thomas, “Cooperative Localization and Mapping of MAVs using RGB-D Sensors,” *IEEE International Conference on Robotics and Automation(ICRA)*, pp. 4021–4028, 2015.
- [12] “The astounding athletic power of quadcopters.” https://www.ted.com/talks/raffaello_d_andrea_the_astounding_athletic_power_of_quadcopters.
- [13] M. Hehn and R. D’Andrea, “Quadrocopter trajectory generation and control,” *In IFAC World Congress*, 2011.

- [14] A. Schoellig, F. Augugliaro, and R. D'Andrea, "A platform for dance performances with multiple quadrocopters," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)-Workshop on Robots and Musical Expressions*, 2010.
- [15] "Ascending Technologies,GmbH." <http://www.asctec.de/>.
- [16] R. Mahony, V. Kumar, and P. Corke, "Multirotor areial vehicles modeling,estimation, and control of quadrotor," *IEEE ROBOTICS and AUTOMATION MAGAZINE*, August 2012.
- [17] "Pixhawk." <http://www.pixhawk.com/>.
- [18] "Taulab." <https://github.com/taulabs>.
- [19] "Micrium." <https://micrium.com/>.
- [20] "OpenCV." <http://opencv.org/>.
- [21] G. Hoffman, S. Waslander, and C. Tomlin, "Quadrotor helicopter trajectory tracking control," *In AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 1–22, April 2008.
- [22] J. Gillula, H. Huang, M. Vitus, and C. Tomilin, "Design and analysis of hybrid system,with applications to robotic aerial vehicles," *Proc. of the Int. Symposium of Robotics Research*, pp. 1–22, September 2009.
- [23] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," *PhD thesis,Ecole Ploytechnique Federale de Lausanne*, 2007.
- [24] D. W. Mellinger, "Trajectory generation and control for quadrotors," *Publicly Accessible Penn Dissertations*, p. 547, 2012.
- [25] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Marc Pollefeyns, "Autonomous obstacle avoidance and maneuvering on a vision-guided mav using on-board processing," *In Robotics and Automation (ICRA), 2011 IEEE International Conference on*, p. 2472 – 2477, May 2011.
- [26] S. Shen, N. Michael, and V. Kumar, "3d estimation and control for autonomous flight with constrained computation," *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, May 2011.

-
- [27] A. Bachrach, R. He, and N. Roy, “Autonomous flight in unknown indoor environments,” *International Journal of Micro Air Vehicles*, p. 1(4):217 – 228, December 2009.
 - [28] D. Mautz, “Indoor positioning technologies,” *Habilitation Thesis submitted to ETH Zurich*, February 2012.
 - [29] A. Gaschler, “Real-time marker-based motion tracking: Application to kinematic model estimation of a humanoid robot,” Master’s thesis, Technische Universitaet München, February 2011.
 - [30] S. Lupashin, A. Schollig, M. Sherback, and R. D’ Andrea, “A simple learning strategy for high-speed quadrocopter multi-flips,” *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, p. 1642 – 1648, May 2010.
 - [31] T. Flash and N. Hogan, “The coordination of arm movements:an experimentally confirmed mathematical model,” *The Journal of Neuroscience*, p. 5:1688 – 1703, 1985.
 - [32] M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki, “Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion,” *Biological Cybernetics*, p. 62:275 – 288, 1990.