



单位代码 10006

学 号 16721099

分 类 号 V279

北京航空航天大学
B E I H A N G U N I V E R S I T Y

毕业设计（论文）

基于视觉的四旋翼自主着陆系统设计

学 院 名 称 宇航学院

专 业 名 称 探测制导与控制技术(航天)

学 生 姓 名 沈天琦

指 导 老 师 张 冉

2020 年 5 月

论文封面书脊

基于视觉的四旋翼自主着陆系统设计

沈天琦

北京航空航天大学

北京航空航天大学

本科生毕业设计（论文）任务书

I、毕业设计（论文）题目：

中文：基于视觉的四旋翼自主着陆系统设计

英文：DESIGN OF VISION-BASED QUADROTOR AUTONOMOUS LANDING
SYSTEM

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

1、原始资料：四旋翼飞行器，单目相机，着陆平台

2、设计技术要求：基于视觉传感器的信息，识别移动的着陆平台上的标志，计算相对位置及姿态，设计降落制导算法，使得四旋翼飞行器能缓慢、稳定、自主降落到着陆平台上。着陆平台移动速度暂定 0.1m/s，降落精度约 20 厘米。

III、毕业设计（论文）工作内容：

1、设计着陆平台上的视觉标志及利用标志解算相对位姿的算法；

2、根据视觉提供的相对位姿信息，设计制导律，实时规划降落轨迹；

3、对四旋翼进行动力学建模，并设计控制律，使得四旋翼能够缓慢平稳降落于移动平台之上；

4、对四旋翼自主着陆系统进行仿真与实验验证，综合分析算法性能；

IV、主要参考资料：

(1) 王兆哲. 基于视觉定位的四旋翼无人机移动平台自主降落技术研究[D].哈尔滨工业大学,2019.

(2) Borowczyk A , Nguyen D T , André Phu-Van Nguyen, et al. Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle[J]. Journal of Guidance Control and Dynamics, 2017, 40(9):1-8.

宇航学院（系）探测制导与控制技术（航天） 专业类 161514 班

学生 沈天琦

毕业设计（论文）时间： 2020 年 1 月 7 日至 2020 年 5 月 30 日

答辩时间： 2020 年 6 月 4 日

成 绩：

指导教师： 张 冉

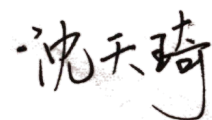
兼职教师或答疑教师（并指出所负责部分）：

航天制导导航与控制系（教研室）主任（签字）：

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者： 沈天琦

签字：

时间： 2020 年 6 月



基于视觉的四旋翼自主着陆系统设计

学 生：沈天琦

指导老师：张 冉

摘要

随着无人机应用技术的不断扩展，空地协同运动控制技术已成为研究热点，并迫切需要无人机具有在运动体上的自主着陆能力。随着图像处理技术的不断进步，已经能够获得低成本、高自主和实时的视觉导航传感器，使得基于视觉的无人机自主着陆技术成为可能。

本文进行了基于视觉的四旋翼自主着陆系统设计。内容包括系统架构搭建、视觉导航算法设计，制导律设计以及控制律设计。视觉导航部分，采用 AprilTag 视觉库的视觉标志和位姿估计算法，在着陆平台的表面贴 AprilTag 标志，通过四旋翼底部单目相机获取图像，通过 AprilTag 标志识别与位姿估计，并对相对位姿信息进行卡尔曼滤波，获得导航信息；制导律部分，采用 Minimum Snap 轨迹规划算法，以轨迹多项式的高阶导数为最优指标，利用相对位姿信息，构建优化问题，进行降落轨迹的实时求解；控制律部分，以制导部分求解出的轨迹为参考状态量，采用 MPC 方法，构建递推预测模型，滚动优化求解控制量，反馈至四旋翼运动模型进行控制。并在 ROS 中搭建仿真环境，进行了运动体分别进行直线运动和圆周运动的虚拟仿真实验，四旋翼都成功降落在了移动着陆平台，实验所得的误差上限分别为 10cm 和 30cm。

本文搭建了一套基于视觉的四旋翼自主着陆系统，验证了基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼自主着陆的可行性，可以为无人机的空地协同运动控制技术提供技术借鉴。

关键词：AprilTag 视觉导航，Minimum Snap 轨迹规划，MPC，ROS 仿真



Design of vision-based quadrotor autonomous landing system

Author : SHENTian-qi

Tutor : ZHANGRan

Abstract

With the continuous expansion of UAV application technology, air ground cooperative motion control technology has become a research hotspot and it is urgent for UAV to have the ability of autonomous landing on the moving body. With the continuous progress of image processing technology, it has been able to obtain low-cost, high autonomous and real-time vision navigation sensors, making vision based autonomous landing technology of UAV possible.

In this paper, a vision-based quadrotor autonomous landing system is designed. The content includes system architecture build, visual navigation algorithm design, guidance law design and control law design. In the visual navigation part, the visual mark and pose estimation algorithm of the AprilTag visual database are used, the AprilTag mark is pasted on the surface of the landing platform, the image is obtained by the quadrotor bottom monocular camera, and the navigation information is obtained by the AprilTag mark recognition and pose estimation, and the relative pose information is filtered by Kalman filter; in the guidance law part, the Minimum Snap trajectory planning algorithm is used and takes higher derivative of trajectory polynomial as the optimal index, uses the relative pose information to build the optimization problem and solve the landing trajectory in real time; the control law part takes the trajectory solved by the guidance part as the reference state quantity, uses the MPC Method to build the recursive prediction model, rolls the optimization solution control quantity and feeds back to the quadrotor motion model for control. The simulation environment is built in ROS, and the virtual simulation experiments of linear motion and circular motion of the moving body are carried out. The quadrotor are successfully landed on the mobile landing platform, and the upper error limits of the experiment are 10 cm and 30 cm respectively.

In this paper, a vision based quadrotor autonomous landing system is built to verify the



feasibility of realizing quadrotor autonomous landing based on AprilTag vision navigation, Minimum Snap trajectory planning and MPC, which can provide technical reference for UAV air ground cooperative motion control technology.

Key words: AprilTag visual navigation, MinimumSnap trajectory planning, MPC, ROS



目 录

1 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 视觉导航研究现状	2
1.2.2 制导与控制研究现状	3
1.3 本文采取的研究方法	4
1.4 论文构成及研究内容	4
2 数学建模	6
2.1 相机成像建模	6
2.1.1 坐标系	6
2.1.2 小孔成像模型	7
2.2 四旋翼运动建模	8
2.2.1 坐标系	8
2.2.2 状态空间模型	9
2.2.3 状态空间的线性化	12
2.3 本章小结	13
3 视觉导航	14
3.1 AprilTag 相对位姿估计原理	14
3.1.1 边缘检测	14
3.1.2 四边形检测	15
3.1.3 四边形编码与位姿估计	16
3.2 卡尔曼滤波	18
3.2.1 卡尔曼滤波方程建立	18
3.2.2 相对位置的卡尔曼滤波	19
3.3 本章小结	24
4 制导	25
4.1 Minimum Snap 轨迹规划原理	25



4.1.1 位置轨迹相关量的表示方法	25
4.1.2 优化问题的构造	26
4.2 本章小结	29
5 模型预测控制	31
5.1 MPC 原理	31
5.1.1 预测模型	31
5.1.2 滚动优化与反馈校正	34
5.2 本章小结	35
6 系统仿真验证	36
6.1 ROS 仿真平台的搭建	36
6.1.1 四旋翼模型	36
6.1.2 着陆平台模型	37
6.2 仿真一：着陆平台静止	38
6.2.1 仿真一的实验结果	39
6.2.2 仿真一的实验结论	40
6.3 仿真二：着陆平台匀速圆周运动	40
6.3.1 仿真二的实验结果	41
6.3.2 仿真二的实验结论	42
6.4 仿真三：着陆平台随机运动	42
6.4.1 仿真三的实验结果	42
6.4.2 仿真三的实验结论	44
6.5 本章小结	44
总结与展望	45
致谢	47
参考文献	49

1 绪论

1.1 研究背景及意义

近些年来无人机技术已经有了相当的发展，这为无人机的应用开辟很多新的领域。然而，受制于电池技术的发展不足^[1]以及无人机载荷的有限等不利因素，实现无人机长时间、广范围作业仍在研究当中。一种解决思路是实现无人机与地面单位的空地协同^[2]，具体表现为无人机起飞作业、无人机降落补给、两者之间通信等。其中，无人机自主降落已成为该方案的实现难点，而随着计算机视觉技术与图像传感器的发展，基于视觉的无人机自主着陆逐步成为该领域的研究热点^[3]。

基于视觉的无人机自主着陆的任务过程涵盖了导航、制导和控制三部分。基于图像信息的导航滤波目前已有相当的研究，因此本次毕业设计研究的重点放在获得图像解算导航信息之后，制导律与控制律的设计，即规划出一条合理的降落轨迹，同时选用合适的控制律控制无人机按照规划出的轨迹进行降落。

本次研究的内容总框图如图 1.1 所示。

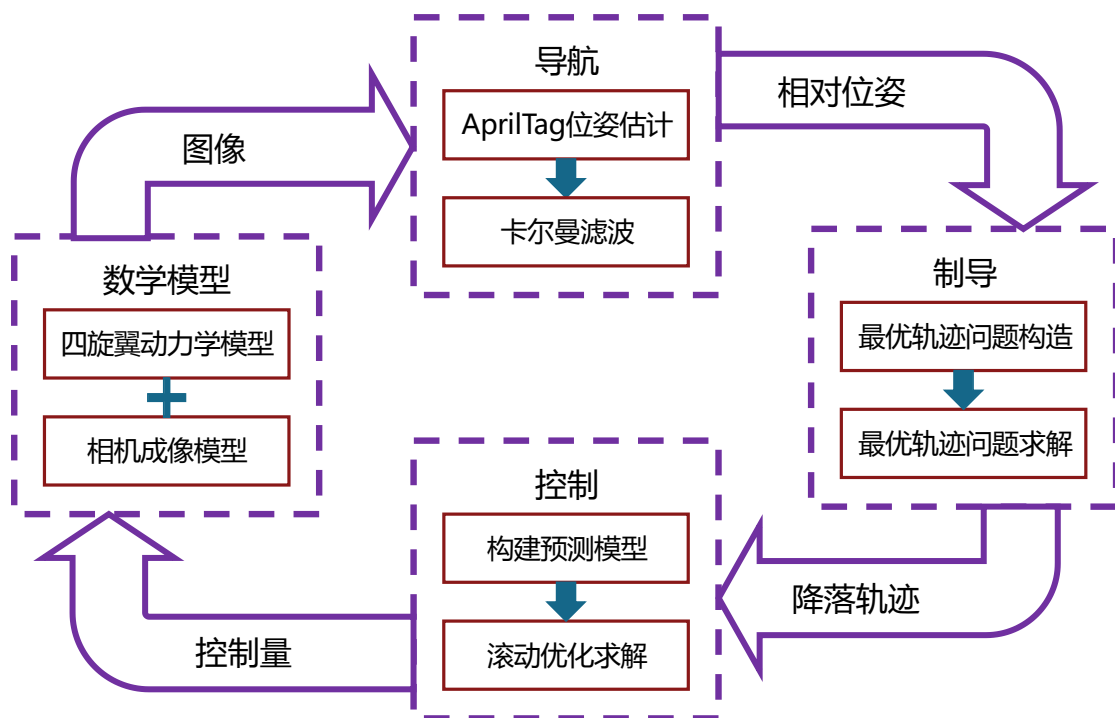


图 1.1 着陆系统总框图

1.2 国内外研究现状

1.2.1 视觉导航研究现状

视觉导航的关键应当是视觉标志的设计，以及针对该标志的相应的视觉算法。

用于四旋翼无人机视觉着陆的地面识别标志，应满足以下设计要求^[4]：第一，标志应能包含无人机自主降落所需的信息，保证可以有效的识别出降落标志并解算出无人机的相对位置；第二，标志不应太复杂，无需进行复杂的模式识别和学习，从而保证识别算法的实时性；第三，标志要容易识别，易于区别其他地面轮廓，保证不会出现误识别而发生不可预测的错误。基于上述设计要求，目前已出现各种各样的着陆标志，其中有自行设计并验证可行性的着陆标志，也有已经形成成熟视觉算法库的着陆标志。

自主设计的着陆标志往往不采用常见的二维码图像。这些设计方案为了提高识别效率，设计具有独特风格的图形，然而这种着陆标志没有经过大量的实验验证，没有形成完善的视觉识别库，不能保证它们的普适性，如图 1.2 便是其中一些例子^[5-6]。

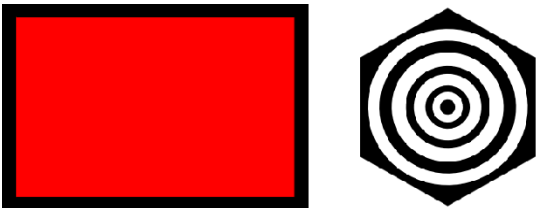


图 1.2 自主设计的着陆标志例子

可以看到，该例设计的着陆标志存在问题有两个，一是矩形、圆形、六边形等形状的着陆标志由于其中心对称性，在跟踪移动平台时容易获得错误的方位信息，二是而带有颜色的着陆标志易受着陆时环境光线条件的影响。

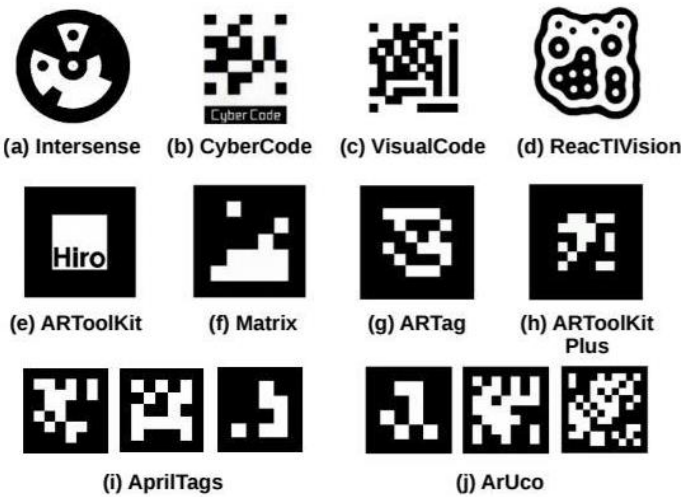


图 1.3 一些典型的视觉库

形成成熟的视觉算法库的着陆标志也有很多种，它们以其经济实用、鲁棒性好的特点被广泛的应用，一些典型的标志如图 1.3 所示^[7]。其中，Intersense 具有良好的更新频率以及较低的位置误差，但其代码不开源并且只能在 Windows 操作系统下使用，与本次研究选用 Ubuntu+ROS 的形式冲突；CyberCode 同样因为代码不开源而不被使用；虽然 VisualCode 与 ReachTIVision 拥有各自的优点，但因其实时性不高，不适用于本次研究；实时性高的 Matrix 则因为抗干扰能力差不被使用；因此，最终选用代码开源，适用于 Ubuntu+ROS 系统，鲁棒性强，实时性与准确性较高的 Apriltag 标识码^[8-10]。

1.2.2 制导与控制研究现状

无人机降落过程的制导需要根据自己的位置、速度以及内部性能和外部环境的约束条件，规划出一条抵达着陆平台的合理路径，因而这实质上是一种路径规划问题，显然传统或改进的比例导引^[11]等方法都不能满足本次研究的需求。同时，本次研究考虑到任务系统的复杂性，暂不考虑实时地图的三维建模，也不考虑障碍物的存在，故依赖地图进行轨迹规划的一类算法不在本次研究的调研范围内。

针对无人机轨迹规划的 Minimum Snap 方法^[12]是一种能够通过一系列三维位置和偏航角实时生成最优轨迹，同时确保满足速度、加速度和输入量约束的方法。因此选择 Minimum Snap 方法作为制导律。

线性控制方法中 PID 控制和 LQR 控制最具有代表性。其中 PID 方法^[13]原理简单，只需要进行参数整定即可，但是在本次的毕业设计中，四旋翼的运动是多通道耦合的，这意味 PID 控制需要整定的参数呈三倍增加；LQR 控制^[14]易于构成闭环系统最优控制，但是其只能在一个固定时域上求解，不能及时更新最优解。其它非线性控制方法将问题复杂化，且不便于求解，因此也不采用。

模型预测控制（MPC, Model Predictive Control）方法^[15]也是对是状态方程进行处理，但可以一次性输出多个参数，减少整定的参数量，利用滚动优化，不在固定时域求解，常更新最优控制。因此选择 MPC 作为控制律。



1.3 本文采取的研究方法

本文采取的研究方法为理论与仿真相结合。对四旋翼欠驱动系统进行数学建模，并相应地在 Gazebo 中建立对应的四旋翼模型。同时在 Gazebo 中搭建相机模型（安装在四旋翼底部）和着陆平台（表面贴上 AprilTag 标志）模型，在模型建立的基础上，在机器人操作系统（ROS, Robot Operating System）中进行仿真。首先编写 AprilTag 标志识别与位姿解算部分工作包并仿真测试，再单独利用 Python 编写 Minimum Snap 轨迹生成和 MPC 控制的文件，最后整合在 ROS 仿真系统中。

1.4 论文构成及研究内容

本文通过建立四旋翼的运动模型，再基于 AprilTag 视觉标志，解算着陆平台与四旋翼之间的相对位姿；然后根据相对位姿关系，使用 Minimum Snap 方法实时规划出无人机降落到着陆平台的降落轨迹；最后利用 MPC 实时控制无人机沿规划出的轨迹进行降落。

各章主要内容如下：

第一章为绪论。阐述了本文的研究背景和目的，简要描述了国内外的研究现状。

第二章为数学建模。对于四旋翼对象，首先定义描述四旋翼运动的坐标系（本体系和惯性系），再对作用在四旋翼上的力和力矩进行建模，得到运动学和动力学方程，最后构建状态空间模型。对于相机对象，同样定义成像过程中的四个坐标系，并建立小孔成像的模型。

第三章为视觉导航及滤波。首先介绍识别 AprilTag 着陆标志并解算相对位姿的算法，再对相对位姿关系进行卡尔曼滤波，最后给出仿真结果。

第四章为制导。介绍 Minimum Snap 方法的原理，以及改进该原理以适用于本次毕业设计。

第五章为模型预测控制。介绍 MPC 方法的原理，以及改进该原理以适用于本次毕业设计。

第六章为系统仿真验证。即在 ROS 系统中整合完整的四旋翼建模、视觉导航滤波、制导和模型预测控制，给出着陆平台匀速直线运动和匀速圆周运动两种仿真条件下的四旋翼自主着陆仿真结果。

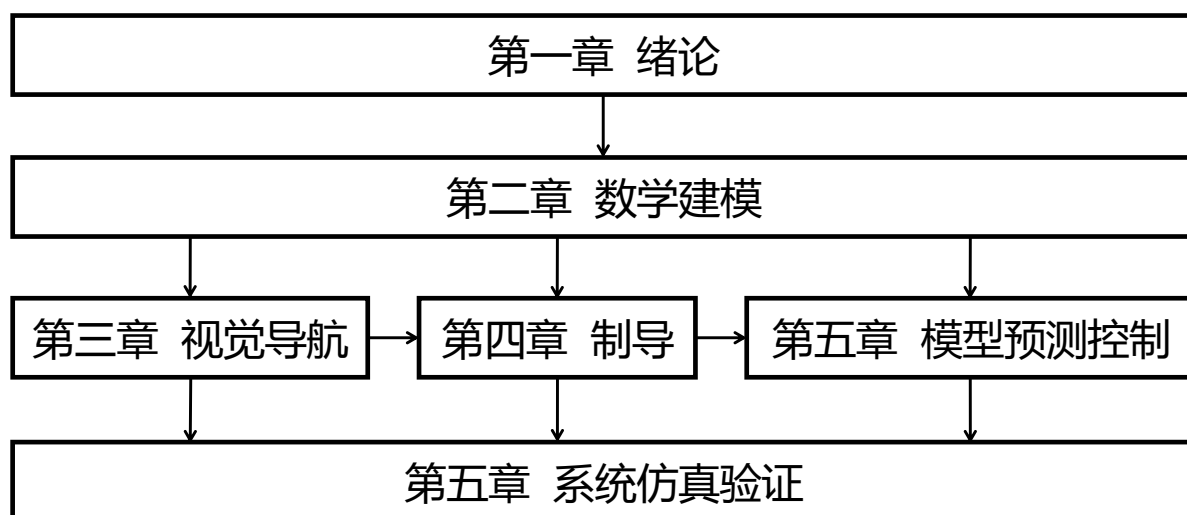


图 1.4 论文内容结构

2 数学建模

本章主要分为相机成像数学建模^[16]和四旋翼数学建模^[17]两个部分，分别为后续的视觉导航和制导律与控制律的设计提供依据。

为了简化问题，对研究对象作以下假设：

- a) 四旋翼机身结构为刚体
- b) 四旋翼结构完全对称
- c) 质心与四旋翼本体坐标系原点重合
- d) 四旋翼本体坐标系沿惯性主轴

2.1 相机成像建模

2.1.1 坐标系

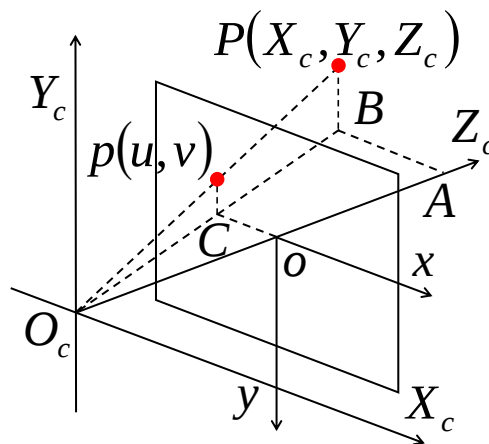


图 2.1 相机成像示意图

①世界坐标系 $O_wX_wY_wZ_w$

同四旋翼运动建模中的惯性系。

②相机坐标系 $O_cX_cY_cZ_c$

相机坐标系的原点和相机的光心相重合， O_cZ_c 垂直于成像平面且与光轴重合，指向相机的前方， O_cX_c 与 O_cY_c 的正方向与物体坐标系平行，其中 f 为相机的焦距，单位为米。

③图像物理坐标系 oxy

为了描述成像过程中物体从相机坐标系到图像坐标系的投影投射关系而引入，方便进一步得到像素坐标系下的坐标。以 CCD 图像平面的中心为坐标原点， ox 与 oy 分别平

行于图像平面的两条垂直边，单位为毫米，表示像素在图像中的位置。

④图像像素坐标系 uv

为了描述物体成像后的像点在数字图像（即相片）上的坐标而引入，是真正从相机内读取到信息所在的坐标系。以图像平面的左上角顶点为原点， u 轴和 v 轴分别平行于图像坐标系的 ox 和 oy 。

2.1.2 小孔成像模型

①从世界坐标系到相机坐标系

从世界坐标系变换到相机坐标系属于刚体变换，即物体不会发生形变，只需要进行旋转和平移。

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \mathbf{T} \Rightarrow \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.1)$$

其中 $\mathbf{R}$ 为 3×3 的旋转矩阵， $\mathbf{T}$ 为 3×1 的平移矩阵。

②从相机坐标系到图像物理坐标系

这个变换过程是从三维到二维的透视投影过程，满足三角形的相似定理。如图 2.1 有如下关系：

$$\begin{cases} \Delta ABO_c \sim \Delta oCO_c \\ \Delta PBO_c \sim \Delta pCO_c \end{cases} \Rightarrow \frac{AB}{oC} = \frac{AO_c}{oO_c} = \frac{PB}{pC} = \frac{X_c}{x} = \frac{Z_c}{f} = \frac{Y_c}{y} \quad (2.2)$$

$$\begin{cases} x = f \frac{X_c}{Z_c} \\ y = f \frac{Y_c}{Z_c} \end{cases} \Rightarrow Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2.3)$$

③从图像物理坐标系到图像像素坐标系

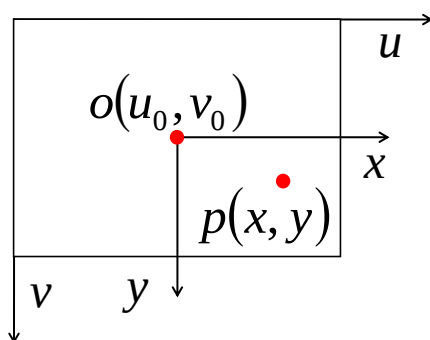


图 2.2 图像物理坐标系与图像像素坐标系的关系图

这个变换过程没有涉及旋转，但原点位置不一致，大小不一致，故属于伸缩变换及平移变换。如图 2.2 有如下关系：

$$\begin{cases} u = \frac{x}{dx} + u_0 \\ v = \frac{y}{dy} + v_0 \end{cases} \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.4)$$

④完整的模型

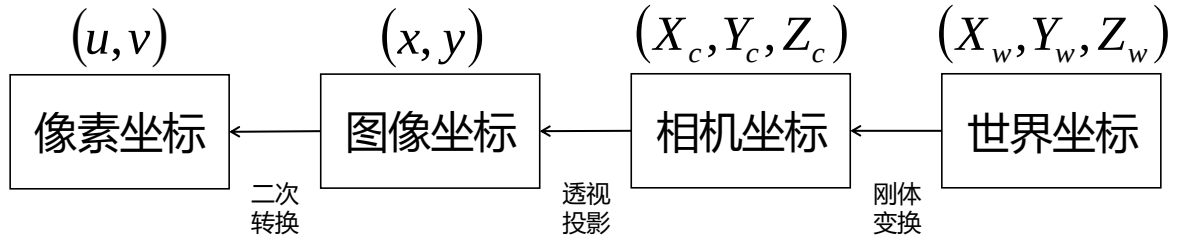


图 2.3 完整的相机成像流程框图

世界坐标系中的物体被相机拍摄成二维图像的完整的坐标系变换的流程框图如图 2.3 所示，该过程经历的刚体变换、透视投影和二次转换三个环节各可用矩阵表示，因此可以用矩阵相乘表示完整的变换过程：

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.5)$$

2.2 四旋翼运动建模

四旋翼的运动建模既涉及运动学建模也涉及动力学建模，而且涉及惯性系与本体系两个坐标系，考虑到这样一种 6 个自由度的系统的复杂性，为了充分描述关键的变量，本次毕业设计采用如下示例的记号：

$${}^b\dot{v}_{CM|i}^b$$

其中，基本变量 $\dot{v}$ 是线加速度，左上标 b 指在本体系中进行求导操作，右上标 b 指该加速度是本体系中的矢量，下标指该加速度是相对于惯性系的质心。

2.2.1 坐标系

①惯性系 $O_iX_iY_iZ_i$

惯性系如图 2.4 所示，实际仿真时也可以根据需要自定义惯性系。

②本体系 $oxyz$

四旋翼的本体系选择十字型配置， ox 轴与电机 1 的机臂重合， oy 轴与电机 2 的机臂重合， oz 由右手系确定且向上，变量 d 为任一电机到旋转轴的距离。

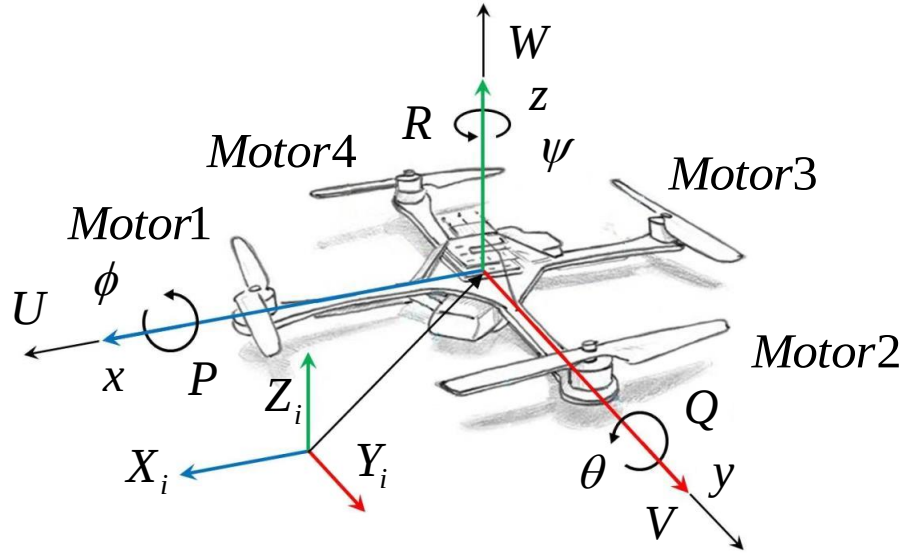


图 2.4 四旋翼建模的坐标系图

2.2.2 状态空间模型

①质量惯性矩阵

质量惯性矩阵描述四旋翼本体系各轴上的转动惯量（又称质量惯性矩），该矩阵定义如下：

$$J^b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (2.6)$$

其中： J_{xx} 是四旋翼在本体系 ox 轴上的转动惯量， J_{yy} 和 J_{zz} 同理。

②推力系数与扭矩系数

四旋翼运动的主要驱动力由四个电机的推力提供，单个电机的推力公式如下：

$$T = C_T \rho A_r r^2 \omega^2 \quad (2.7)$$

其中： C_T 是该电机的推力系数， ρ 是空气密度， A_r 是螺旋桨旋转的横截面积， r 是该电机的半径， ω 是该电机旋转的角速度。对于简单的飞行建模，可以采用集总参数法减少物理量：

$$T = c_T \omega^2 \quad (2.8)$$

其中： c_T 是集总参数， $c_T = C_T \rho A_r r^2$ ，由该电机具体物理参数决定，推力 T 垂直于本体系的 xoy 平面，方向为 oz 轴正向。

电机旋转产生的扭矩会对四旋翼的偏航产生影响，故类似于上述推力的方式，可得电机旋转产生的扭矩 Q 的集总参数表达如下式所示：

$$Q = c_Q \varpi^2 \quad (2.9)$$

其中： c_Q 是集总的扭矩系数。该电机旋转产生的扭矩 Q 会产生使四旋翼绕 oz 轴偏航的力。

③陀螺力

电机旋转产生的陀螺进动力同样作用在四旋翼本体系，其大小由电机的转动惯量 J_m ，滚转角度率 P 和俯仰角速率 Q ，以及电机的转速 ϖ_i 共同决定，表达式如下所示：

$$\tau_{\phi_{gyro}} = J_m Q \left(\frac{\pi}{30} \right) (\varpi_1 - \varpi_2 + \varpi_3 - \varpi_4) \quad (2.10)$$

$$\tau_{\theta_{gyro}} = J_m P \left(\frac{\pi}{30} \right) (-\varpi_1 + \varpi_2 - \varpi_3 + \varpi_4) \quad (2.11)$$

其中： $\pi/30$ 是从 RPM 单位变换到弧度制的转换量。

④矩阵构造

最终将①②③中的讨论结合，可以将作用在四旋翼上的力和力矩用矩阵构造成如下形式：

$$\mathbf{M}_{A,T}^b = \begin{bmatrix} d_+ c_T \varpi_2^2 - d_+ c_T \varpi_4^2 + J_m Q \left(\frac{\pi}{30} \right) (\varpi_1 - \varpi_2 + \varpi_3 - \varpi_4) \\ -d_+ c_T \varpi_1^2 + d_+ c_T \varpi_3^2 + J_m Q \left(\frac{\pi}{30} \right) (-\varpi_1 + \varpi_2 - \varpi_3 + \varpi_4) \\ -c_Q \varpi_1^2 + c_Q \varpi_2^2 - c_Q \varpi_3^2 + c_Q \varpi_4^2 \end{bmatrix} \quad (2.12)$$

$$\mathbf{F}_{A,T}^b = \begin{bmatrix} 0 \\ 0 \\ c_T (\varpi_1^2 + \varpi_2^2 + \varpi_3^2 + \varpi_4^2) \end{bmatrix} \quad (2.13)$$

其中： $\mathbf{M}_{A,T}^b$ 是由空气动力、推力和扭矩而引起的在四旋翼本体系中的力矩， $\mathbf{F}_{A,T}^b$ 是由于空气动力和电机推力而在四旋翼本体系中产生的力。

⑤状态方程

整个四旋翼运动的状态方程有四个，分别是角速度状态方程、欧拉运动方程、速度状态方程和位置状态方程，具体方程如下：

角速度状态方程：

$${}^b\dot{\omega}_{b|i}^b = (J^b)^{-1} [M_{A,T}^b - \Omega_{b|i}^b J^b \omega_{b|i}^b] = \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} \quad (2.14)$$

欧拉运动方程:

$$\dot{\Phi} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t(\theta)s(\phi) & t(\theta)c(\phi) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)/c(\theta) & c(\phi)/c(\theta) \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = H(\Phi) \omega_{b|i}^b \quad (2.15)$$

速度状态方程:

$${}^b\dot{v}_{CM|i}^b = \left(\frac{1}{m}\right) F_{A,T}^b + g^b - \Omega_{b|i}^b v_{CM|i}^b = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} \quad (2.16)$$

位置状态方程:

$${}^i\dot{P}_{CM|i}^i = C_{i|b} v_{CM|i}^b = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad (2.17)$$

其中，四旋翼在惯性系的三轴位置为 $[X \ Y \ Z]^T$ ，在本体系的三轴速度为 $[U \ V \ W]^T$ ，滚转、俯仰和偏航角分别为 $[\phi \ \theta \ \psi]^T$ ，本体系相对于惯性系三轴的转动角速度为 $[P \ Q \ R]^T$ ，上述 12 个量便为描述四旋翼运动的全部的状态量。

此外，还有其它一些量需要说明:

为了便于计算叉积，定义旋转叉积矩阵 $\Omega_{b|i}^b$ 和位移叉积矩阵 $\Lambda_{b|i}^b$:

$$\Omega_{b|i}^b = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}, \Lambda_{b|i}^b = \begin{bmatrix} 0 & -W & V \\ W & 0 & -U \\ -V & U & 0 \end{bmatrix} \quad (2.18)$$

从四旋翼惯性系到四旋翼本体系的坐标系变换矩阵 $C_{b|i}$ ($C_{i|b} = C_{b|i}^T$):

$$C_{b|i} = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ -c(\phi)s(\psi) + s(\phi)s(\theta)c(\psi) & c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi) & s(\phi)c(\theta) \\ s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) & -s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi) & c(\phi)c(\theta) \end{bmatrix} \quad (2.19)$$

以及欧拉角 $\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$ 的三角函数，以滚转角 ϕ 为例（ θ 和 ψ 同理）:

$$s(\phi) = \sin(\phi), c(\phi) = \cos(\phi), t(\phi) = \tan(\phi)$$

作用在四旋翼本体系的重力加速度 g^b :

$$g^b = C_{b|i} g^i = \begin{bmatrix} -s(\theta)g \\ s(\phi)c(\theta)g \\ c(\phi)c(\theta)g \end{bmatrix} \quad (2.20)$$

2.2.3 状态空间的线性化

从上一条中得到的 4 个状态方程可以看出，四旋翼运动的系统存在状态量的耦合及非线性，为了便于后续的 MPC 设计，有必要将 4 个状态空间方程整合成一个矩阵方程，并将该方程表述为

$$\dot{\mathbf{x}} = \mathbf{A}_c \mathbf{x} + \mathbf{B}_c \mathbf{u} \quad (2.21)$$

的形式，以完成状态空间模型的线性化。

为了方便表达完整的 12×12 维状态矩阵，记矩阵 $\mathbf{E}$ 和 $\mathbf{F}$ 如下：

$$\mathbf{E} = \begin{bmatrix} 0 & -g \left(1 - \frac{\theta^2}{6} + \frac{\theta^4}{120} \right) & 0 \\ gc(\theta) \left(1 - \frac{\phi^2}{6} + \frac{\phi^4}{120} \right) & 0 & 0 \\ g \frac{1+c(\theta)}{2} \left(-\frac{\phi}{2} + \frac{\phi^3}{24} + \frac{\phi^5}{720} \right) & g \frac{1+c(\phi)}{2} \left(-\frac{\theta}{2} + \frac{\theta^3}{24} + \frac{\theta^5}{720} \right) & 0 \end{bmatrix} \quad (2.22)$$

$$\mathbf{F} = \begin{bmatrix} 0 & \frac{(J_{yy} - J_{zz})R}{2J_{xx}} & \frac{(J_{yy} - J_{zz})Q}{2J_{xx}} \\ \frac{(J_{zz} - J_{xx})R}{2J_{yy}} & 0 & \frac{(J_{zz} - J_{xx})P}{2J_{yy}} \\ \frac{(J_{xx} - J_{yy})Q}{2J_{zz}} & \frac{(J_{xx} - J_{yy})P}{2J_{zz}} & 0 \end{bmatrix} \quad (2.23)$$

则整合后的状态矩阵 $\mathbf{A}_c$ 及控制量的输入矩阵 $\mathbf{B}_c$ 如下所示：

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{C}_{b|i}^T & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{1}{2} \boldsymbol{\Omega}_{b|i}^b & \mathbf{E} & -\frac{1}{2} \mathbf{A}_{b|i}^b \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{H}(\Phi) \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{F} \end{bmatrix} \quad (2.24)$$

$$\mathbf{B}_c = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c_T/m & c_T/m & c_T/m & c_T/m \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & c_T d_+/J_{xx} & 0 & -c_T d_+/J_{xx} \\ -c_T d_+/J_{yy} & 0 & c_T d_+/J_{yy} & 0 \\ -c_Q/J_{zz} & c_Q/J_{zz} & -c_Q/J_{zz} & c_Q/J_{zz} \end{bmatrix} \quad (2.25)$$

需要注明的是，上述将状态空间模型线性化的过程，忽略了上一条建立状态空间模型时对陀螺力的分析，由于陀螺的转动惯量与四旋翼的转动惯量有数量级上的差异，因此在非精确建模的情况下，这是可以接受的。

综上所述，建立了用 12 个状态量描述四旋翼运动系统的完整的线性状态方程如下：

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = A_c \begin{bmatrix} X \\ Y \\ Z \\ U \\ V \\ W \\ \phi \\ \theta \\ \psi \\ P \\ Q \\ R \end{bmatrix} + B_c \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = A_c \mathbf{x} + B_c \mathbf{u} \quad (2.26)$$

其中： $\mathbf{x}_{12 \times 1}$ 为状态量， $\mathbf{u}_{4 \times 1}$ 为（输入）控制量，仅由四个电机的转速确定。

2.3 本章小结

本章主要对相机成像过程和四旋翼运动过程这两个对象进行了数学建模，得到了相机成像中的内参和外参矩阵（后续仿真中可以手动设置内参矩阵）等数学定义，以及描述四旋翼运动过程的整合的状态方程。

综上所述，本章为后续各章的内容提供数学依据，在第 3 章中将利用相机成像原理，阐述视觉标志 AprilTag 解算位姿的原理，在第 4 章和第 5 章中将利用四旋翼运动的模型设计制导律（Minimum Snap 轨迹规划）和控制率（MPC）。

3 视觉导航

视觉导航的任务是通过视觉传感器实时获取着陆标志的图像，再通过相应的图像算法根据所获图像及相机成像模型解算相应的导航信息。在本次毕业设计任务中，视觉传感器采用简单的单目相机，图像算法采用标准的 AprilTag 库算法，导航信息为四旋翼与着陆平台之间的相对位置和相对姿态（简称相对位姿）。只要是传感器，则必然存在噪声，因此外导航的另一个还必须进行的步骤是对解算出的相对位姿进行滤波，在本次毕业设计任务中采用传统的卡尔曼滤波。

3.1 AprilTag 相对位姿估计原理

3.1.1 边缘检测

边缘检测的目的是找到图像中亮度变化剧烈的像素点构成的集合。如果单目相机获取的边缘能够被精确地测量与定位，那么意味着着陆平台在相机坐标系中也可以被精确地定位。AprilTag 视觉标志采用的是二维码型的标志，故边缘检测方便从场景中提取出该标志。传统的边缘检测算子有 Sobel 算子、Laplacian 算子和 Canny 算子，在本次毕业设计任务中，采用局部二值模式（LBP, Local Binary Pattern）的方法^[18]进行边缘检测。

①基本的 LBP

基本的 LBP 算子为 3×3 的核，核中心的像素值作为阈值，将相邻的 8 个像素点的像素值与阈值比较，小于等于阈值记为 0，大于阈值记为 1，将这些 0 和 1 按顺序可以组织成 8 位二进制数，再将二进制数换算成十进制数，得到的数便作为该 LBP 核作用在该块 3×3 区域中心像素点的 LBP 值。

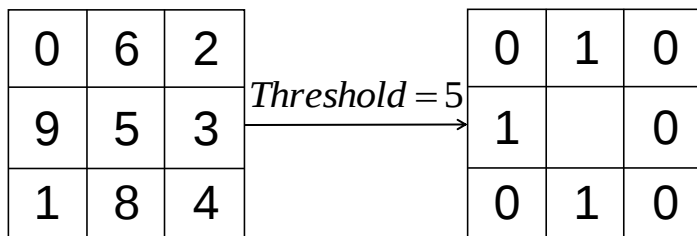


图 3.1 基本的 LBP 算子作用原理图

若设 LBP 作用的 3×3 区域中心像素为 (u_i, v_j) ，对应像素值为 $s_{i,j}$ ，则公式表达为：

$$\text{LBP}(u_i, v_j) = 2^7 s_{i-1, j-1} + 2^6 s_{i, j-1} + 2^5 s_{i+1, j-1} + 2^4 s_{i+1, j}$$



$$+2^3s_{i+1,j+1} + 2^2s_{i,j+1} + 2^1s_{i-1,j+1} + 2^0s_{i-1,j} \quad (3.1)$$

将 LBP 核遍历整幅图像后会得到经过 LBP 滤波的图像，该图像展示的就是原始相机采集图像中的边缘，从而完成边缘检测的任务。

②改进的 LBP

考虑到基本的 LBP 有如下 2 个缺点：

- a、只覆盖固定范围的小区域，对不同尺寸、频率的纹理特征检测效果差。
- b、不满足旋转不变性，即若图像旋转，同一块区域的 LBP 值将变化。

针对以上两个缺点，AprilTag 中采用改进的 LBP 以更鲁棒地提取图像边缘，具体做法是：

a、将 3×3 邻域的限定放宽到任意邻域，同时不再采用正方形的 LBP 核，而采用圆形的 LBP 核，圆周经过的所有像素点都是该 LBP 核作用的对象。

b、将上述的圆形邻域的 LBP 核旋转一周，旋转过程中将会产生若干不同的 LBP 值，取其中最小的值作为该 LBP 核作用后的 LBP 值。

综上所述，改进的 LBP 检测并提取到不同尺寸、频率的纹理特征（在本次毕业设计的任务中是边缘），且该检测具有旋转不变性。

3.1.2 四边形检测

在上一条中，边缘检测通过 LBP 算子获得了边缘纹理特征，但是这些边缘可能并非直线，可能是直线但并不能构成封闭的多边形，可能是多边形但并非四边形，即使是四边形也不能保证是凸四边形。为了解决上述问题，真正地检测出 AprilTag 标志的四边形，AprilTag 算法运用了多边形分析，处理的对象是 LBP 算子作用后再经过二值化处理得到的二值化边缘图像，具体的处理方法如下所述。

①边缘结构分析（Topological structural analysis of digitized binary images）^[19]

LBP 算子得到的边缘在原图的位置是有相应的对应关系的，因此不同的边缘相对原图的位置也不同。边缘结构分析则利用这一原理，确定二值图像边缘的围绕关系，即确定外边缘、内边缘以及它们之间嵌套关系。

为了方便梳理多边形的层次关系，首先对所有边缘进行编码区分。编码完成之后，从给定起始的像素点开始，遍历边缘中的像素点，搜寻与起始的像素点属于同类的边缘点，当再次遍历至起始的像素点时，便得到一个封闭的多边形。同理可将上述的过程应



用到所有的二值像素点，便得到若干多边形，筛选出其中边数大于等于 4 的多边形，以便其进行下一步处理。

②多边形凸包寻找算法（Finding the convex hull of a simple polygon）^[20]

上述过程得到的多边形不一定是凸的，但是任一多边形顺序连接其所有顶点必然可以构成凸多边形，若这个由原多边形顶点构成的凸多边形的面积大于原多边形，那么原多边形必然非凸，多边形凸包寻找算法舍弃这些多边形。

③道格拉斯-普克算法（Douglas-Peucker algorithm）^[21]

该算法用于对上述得到的凸四边形进行逼近，伪代码如表 3.1 所示。

Douglas-Peucker algorithm
输入与输出：凸四边形顶点集合 & 逼近的凸四边形
function POLYGON_APPROXIMATION(Points[], Threshold)
pointA, pointB $\in$ Points[]
line(以 pointA, pointB 为端点的曲线)
curve(以 pointA, pointB 为端点的直线)
find pointC $\in$ curve(pointC 距离 line 最远, 距离为 distance)
if distance < Threshold then
curve 可作为 line 的近似
else then
curve1 = pointApointC, curve2=pointBpointC
repeat 步骤 3456
end if
repeat 步骤 2 直到遍历 Points[] 中的点
return 所有 line
end function

表 3.1 道格拉斯-普克算法

3.1.3 四边形编码与位姿估计

在上一条中检测出的四边形已经是相机获取的场景中所有凸四边形，并且通过逼近获得了这些四边形的所有像素点，但是并不能保证这些四边形就是 AprilTag 标志，为此



需要将这些四边形进行编码^[22]，且将编码的结果与 AprilTag 库中所存的标志编码进行比对，以确认这就是本次毕业设计需要的 AprilTag 着陆标志。

在已得到的四边形中，可以根据四边形顶点确定四边形内的像素点点阵的坐标，进而对这些点阵进行编码。编码方式有很多种，具体编码方式在对物体进行标定时确定。不同的编码方式处理的点阵的规模不相同（有 8×8 、 7×7 、 6×6 可选），因此可以生成不同的内点坐标。编码过程的伪代码如表 3.2 所示。

Quadrilateral Coding algorithm
输入与输出：四边形 Quad 和灰度图像 Gray & 编码结果 Tagcode
function CODEQUAD(Quad, Gray)
Tagcode $\leftarrow 0$
PointsIndex $\leftarrow$ 取得四边形顶点(Quad)
BlackValue $\leftarrow$ 计算黑色块的灰度值均值(PointsIndex, Gray)
WhiteValue $\leftarrow$ 计算白色块的灰度值均值(PointsIndex, Gray)
Threshold $\leftarrow$ 计算阈值(BlackValue, WhiteValue)
for Point in PointsIndex do
Tagcode 左移一位
if Point 处的灰度值 > Threshold then
Tagcode 与 1 进行与运算
end if
end for
return Tagcode
end function

表 3.2 四边形编码算法

将四边形编码结果与 AprilTag 编码库进行比对，若码值之间的汉明距离小于阈值，那么可以认为该四边形即是库中编码对应的 AprilTag 着陆标志，至此 AprilTag 检测的完整过程已经结束。

通过将 AprilTag 编码的结果与库中编码结果比对，并对码值进行相应的算法处理与解码（解码过程对应编码的逆过程），便可计算该 AprilTag 着陆标志的其他参数，包括该标志相对于原标志的旋转方位，以及着陆标志的单应矩阵（Homography Matrix）^[23]。

单应矩阵可分解出旋转矩阵与平移向量，利用第2章所述的相机成像模型，便可以解算出着陆标志在相机坐标系中的位姿，进而通过坐标系变换可以得到四旋翼与着陆平台之间的相对位姿。

3.2 卡尔曼滤波

由相机传感器解算获得的相对位姿必然存在误差，为了尽量地获得更加可靠的相对位姿关系，有必要对相对位姿进行滤波，其中主要是对相对位置进行滤波，滤波方法将采用传统的卡尔曼滤波^[24]。

3.2.1 卡尔曼滤波方程建立

考虑到着陆平台的移动规律是未知的，因此无法构建四旋翼与着陆平台之间运动的状态方程，因此考虑如下假设：

假设：在一个滤波周期内，四旋翼与着陆平台之间的相对位置 (x, y, z) 匀速变化。

若将 (x, y, z) 离散化，并设一个滤波周期为 Δt ，则上述假设可用下式表示：

$$\begin{cases} \frac{x_k - x_{k-1}}{\Delta t} = \frac{x_{k+1} - x_k}{\Delta t} = \frac{x_{k+2} - x_{k+1}}{\Delta t} = \dots \\ \frac{y_k - y_{k-1}}{\Delta t} = \frac{y_{k+1} - y_k}{\Delta t} = \frac{y_{k+2} - y_{k+1}}{\Delta t} = \dots \\ \frac{z_k - z_{k-1}}{\Delta t} = \frac{z_{k+1} - z_k}{\Delta t} = \frac{z_{k+2} - z_{k+1}}{\Delta t} = \dots \end{cases} \quad (3.2)$$

令 $\mathbf{\varepsilon}$ 为卡尔曼滤波的状态量：

$$\mathbf{\varepsilon}_k = \left[x_k \quad y_k \quad z_k \quad \frac{x_{k+1} - x_k}{\Delta t} \quad \frac{y_{k+1} - y_k}{\Delta t} \quad \frac{z_{k+1} - z_k}{\Delta t} \right]^T \quad (3.3)$$

则完整的卡尔曼滤波方程建立如下：

预测方程：

$$\hat{\mathbf{\varepsilon}}_k^- = \mathbf{A} \hat{\mathbf{\varepsilon}}_k \quad (3.4)$$

$$\mathbf{P}_k^- = \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q} \quad (3.5)$$

更新方程：

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.6)$$

校正方程：

$$\hat{\mathbf{\varepsilon}}_k = \hat{\mathbf{\varepsilon}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{\varepsilon}}_k^-) \quad (3.7)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (3.8)$$

其中：

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.9)$$

经过调参，设置状态噪声的协方差矩阵 $\mathbf{Q}$ 和观测噪声的协方差矩阵 $\mathbf{R}$ 如下：

$$\mathbf{Q} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

至此构建了整个卡尔曼滤波方程。

3.2.2 相对位置的卡尔曼滤波

为了具体展示卡尔曼滤波的效果，将采用两种展示方式：

①与真实值的曲线直接比较

将 AprilTag 解算出的相对位置，以及 AprilTag 解算出并经过卡尔曼滤波之后的相对位置，分别与仿真中的实际的相对位置进行比较，如图 3.2、3.3 和 3.4 所示。

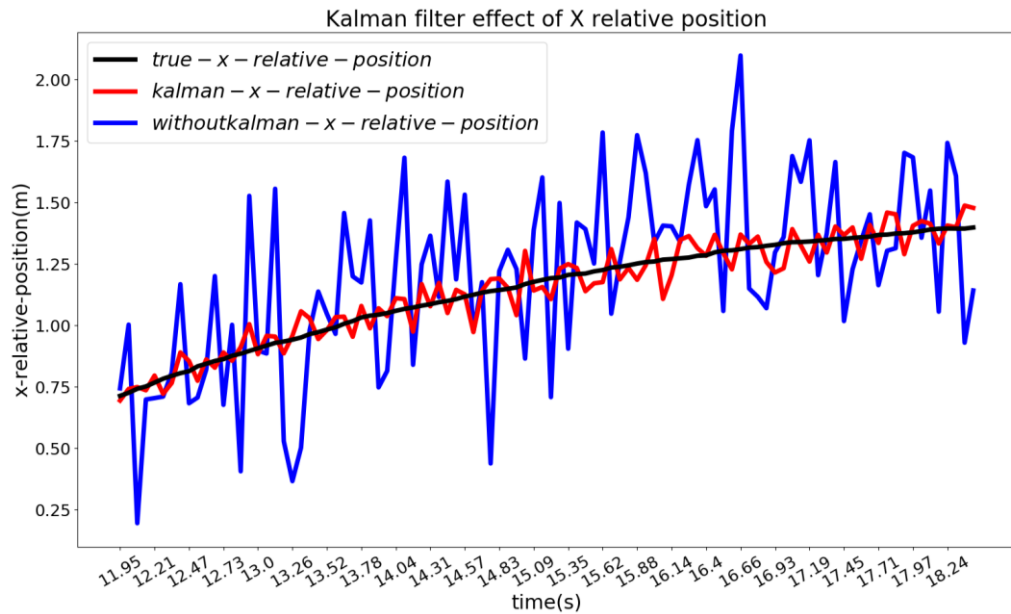


图 3.2x 向相对位置的卡尔曼滤波效果图

其中：黑色曲线是在仿真过程中，着陆平台在四旋翼本体系中实际的 ox 轴距离；蓝色曲线是通过 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 ox 轴距离（未经过滤波处理）；红色曲线是将 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 ox 轴距离，再经过卡尔曼滤波处理之后的距离。

可以看出，红色曲线比蓝色曲线更接近真实的距离，即着陆平台在四旋翼本体系中实际的 ox 轴距离。经过计算得到红色曲线与黑色曲线的平均误差约为 13cm，蓝色曲线与黑色曲线的平均误差约为 32cm。

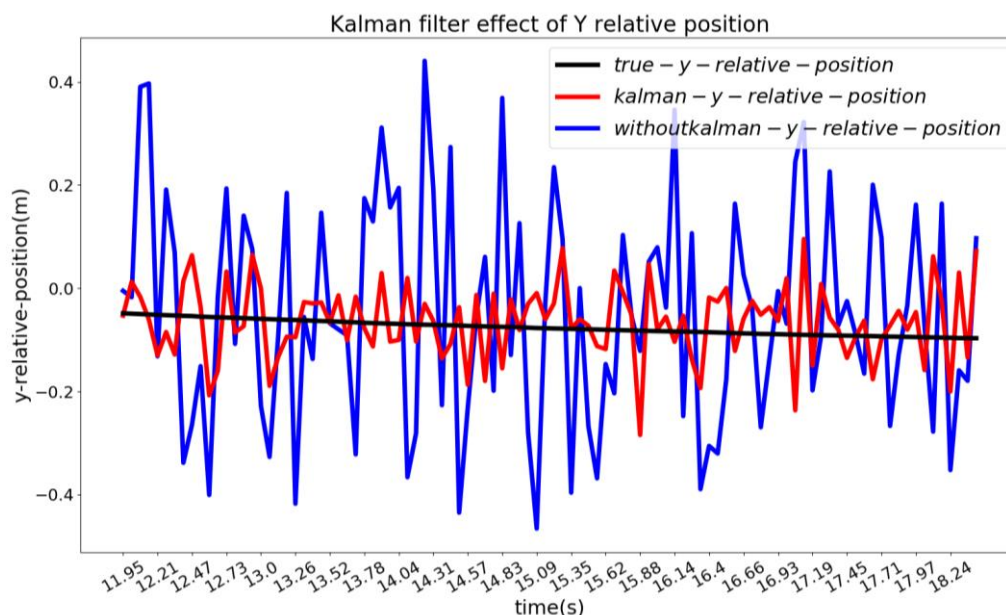


图 3.3y 向相对位置的卡尔曼滤波效果图

其中：黑色曲线是在仿真过程中，着陆平台在四旋翼本体系中实际的 oy 轴距离；蓝色曲线是通过 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oy 轴距离（未经过滤波处理）；红色曲线是将 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oy 轴距离，再经过卡尔曼滤波处理之后的距离。

可以看出，红色曲线比蓝色曲线更接近真实的距离，即着陆平台在四旋翼本体系中实际的 oy 轴距离。经过计算得到红色曲线与黑色曲线的平均误差约为 16cm，蓝色曲线与黑色曲线的平均误差约为 38cm。

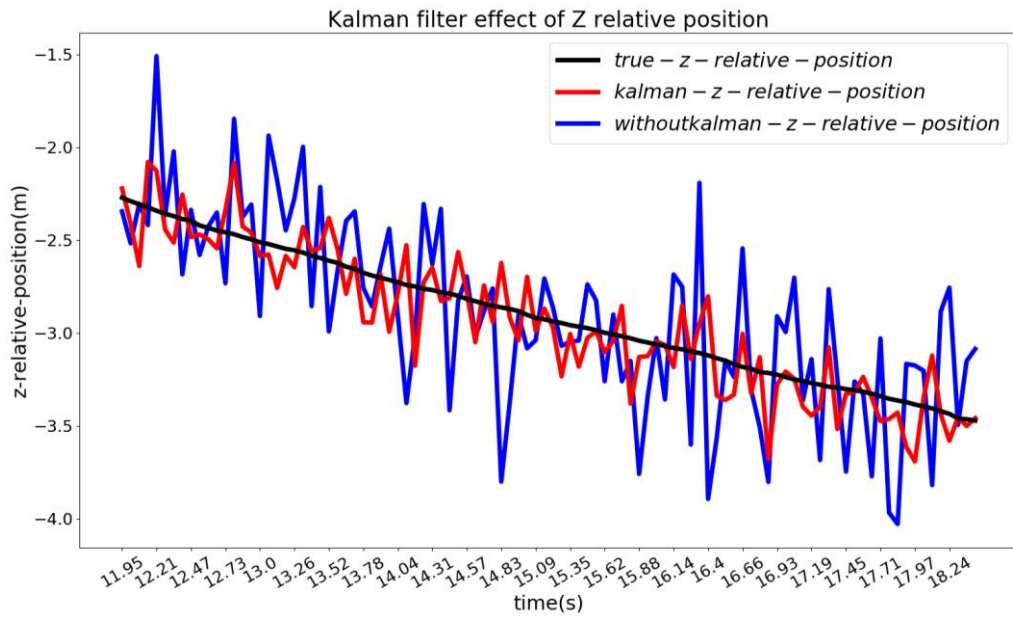


图 3.4z 向相对位置的卡尔曼滤波效果图

其中：黑色曲线是在仿真过程中，着陆平台在四旋翼本体系中实际的 oz 轴距离；蓝色曲线是通过 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oz 轴距离（未经过滤波处理）；红色曲线是将 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oz 轴距离，再经过卡尔曼滤波处理之后的距离。

可以看出，红色曲线比蓝色曲线更接近真实的距离，即着陆平台在四旋翼本体系中实际的 oz 轴距离。经过计算得到红色曲线与黑色曲线的平均误差约为 19cm，蓝色曲线与黑色曲线的平均误差约为 47cm。

经过多次实验，得到相对位置的卡尔曼滤波误差在 20cm 以内，而未经过卡尔曼滤波处理的相对位置误差在 50cm 以内，因此可以认为经过卡尔曼滤波的相对位置更接近真实的相对位置。

②绘制与真实值之间的均方根误差曲线

为了进一步对比有无卡尔曼滤波的相对位置估计精度，对整个降落过程仿真进行的若干次滤波求取均方根误差（Root Mean Square Error, RMSE），对 x 向相对位置的均方根误差的求取方法如下（ y 向和 z 向同理）：

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x - x_{truth})^2}{n}} \quad (3.11)$$

其中： x 是经过（或未经过）卡尔曼滤波的四旋翼与着陆平台的 x 轴相对位置， x_{truth} 是仿真过程中四旋翼与着陆平台的真实的 x 轴相对位置， n 指进行第 n 次卡尔曼滤波。

绘制出的“RMSE—时间”曲线如图 3.5、3.6 和 3.7 所示。

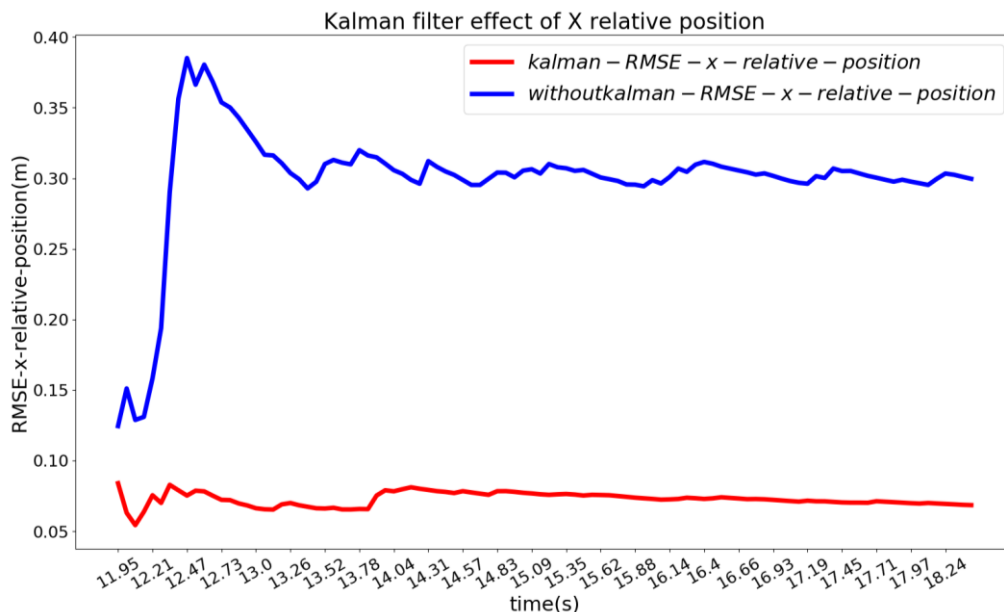


图 3.5x 向相对位置的 RMSE 对比图

其中：蓝色曲线是通过 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 ox 轴距离（未经滤波处理），与仿真过程中着陆平台在四旋翼本体系中实际的 ox 轴距离，求取 RMSE 值的曲线；红色曲线是将 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 ox 轴距离再经过卡尔曼滤波处理之后的距离，与仿真过程中着陆平台在四旋翼本体系中实际的 ox 轴距离，求取 RMSE 值的曲线。

可以看出红色曲线明显低于蓝色曲线，即经过滤波处理后的相对位置误差减小。

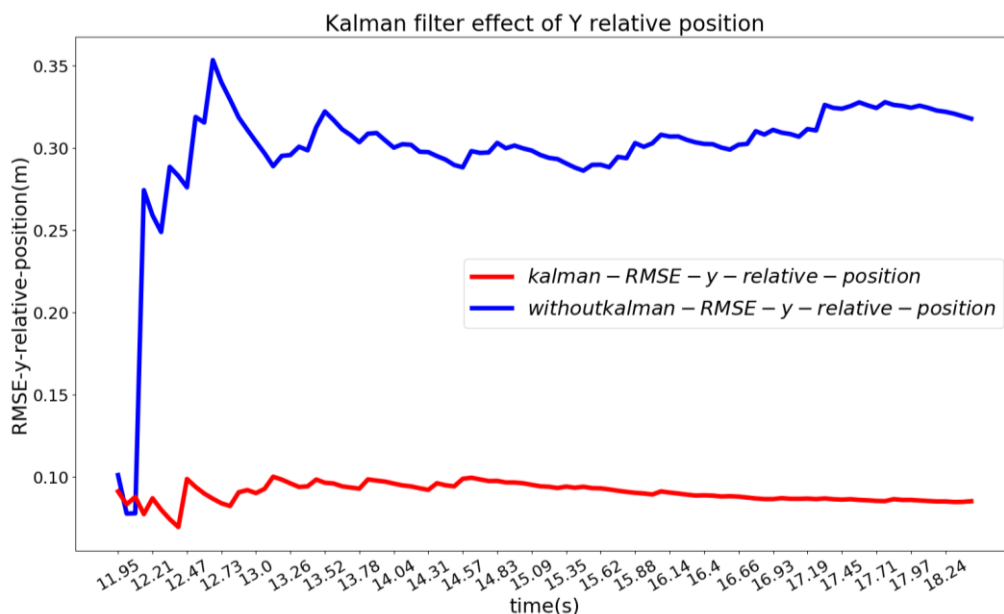


图 3.6y 向相对位置的 RMSE 对比图

其中：蓝色曲线是通过 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oy 轴距离（未经滤波处理），与仿真过程中着陆平台在四旋翼本体系中实际的 oy 轴距离，求取 RMSE 值的曲线；红色曲线是将 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oy 轴距离再经过卡尔曼滤波处理之后的距离，与仿真过程中着陆平台在四旋翼本体系中实际的 oy 轴距离，求取 RMSE 值的曲线。

可以看出红色曲线明显低于蓝色曲线，即经过滤波处理后的相对位置误差减小。

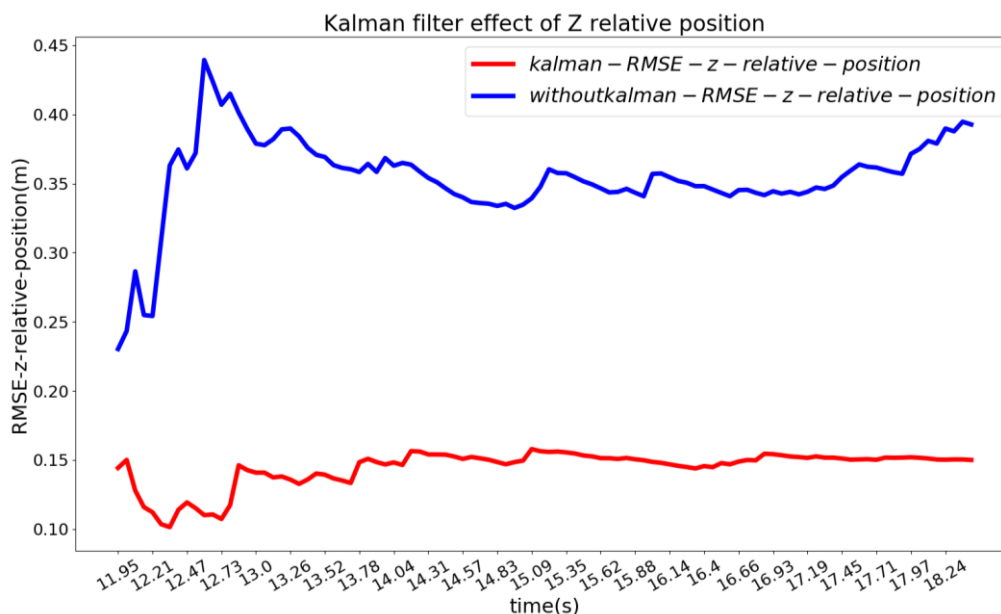


图 3.7z 向相对位置的 RMSE 对比图

其中：蓝色曲线是通过 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oz 轴距离（未经滤波处理），与仿真过程中着陆平台在四旋翼本体系中实际的 oz 轴距离，求取 RMSE 值的曲线；红色曲线是将 AprilTag 视觉导航解算出的着陆平台在四旋翼本体系中的 oz 轴距离再经过卡尔曼滤波处理之后的距离，与仿真过程中着陆平台在四旋翼本体系中实际的 oz 轴距离，求取 RMSE 值的曲线。

可以看出红色曲线明显低于蓝色曲线，即经过滤波处理后的相对位置误差减小。

结论：经过卡尔曼滤波的相对位置更接近真实的相对位置。

相对速度的卡尔曼滤波结果同理相对位置，只需对相对位置进行一次差分即可；相对姿态的滤波结果也同理相对位置，本章将不再赘述。



3.3 本章小结

本章节主要介绍了 AprilTag 视觉库利用其标志进行相对位姿估计的原理，即先进行边缘检测寻找边缘；接着从所有边缘中筛选可以构成凸四边形的边缘；再对凸四边形内的像素点进行编码，与 AprilTag 库中编码比对，确定视觉标志的种类；最后根据编码结果得到单应矩阵，并利用第 2 章中的相机成像建模解算出相对位姿。

在得到相对位姿之后，本章利用“在一个滤波周期内，四旋翼与着陆平台之间的相对位置 (x, y, z) 匀速变化”的假设，对相对位姿进行卡尔曼滤波的原理进行了推导。接着采用了“直接与真实值的曲线进行比较”和“求取与真实值之间的均方根误差进行比较”两种方式，证明了实际仿真中卡尔曼滤波后的相对位置更接近真实的相对位置，即上述假设下的卡尔曼滤波确实是有效的。此外，相对姿态的滤波同理，本章不再赘述。

综上所述，本章得到了四旋翼与着陆平台之间较为精确地相对位姿信息，这为第 4 章设计制导律，即进行轨迹规划提供了依据。



4 制导

在制导部分，本次毕业设计的主要目的是：根据第3章中获得的卡尔曼滤波后的四旋翼与着陆平台之间的相对位姿关系，人为地设置优化指标与一些约束条件，构造一个优化问题，求解出一条前往着陆平台的降落轨迹。

考虑本次毕业设计的任务的一个核心问题：在四旋翼着陆过程中，着陆平台也是移动的，当四旋翼按照前一时刻规划的轨迹跟踪降落了一段距离后，着陆平台的位置已经发生了改变，如果不及时调整规划的轨迹，那么势必降落失败。因此，在降落过程中，不断根据新的视觉导航信息（滤波后的相对位姿），实时规划新的轨迹，这是完成降落任务的目标，故构造轨迹多项式的高阶导数作为轨迹规划的优化指标。同时，考虑到为了精准控制着陆过程中的状态，第2章中得到的描述四旋翼运动状态的12个量需要尽可能地加入到轨迹规划中。综上考虑，本次毕业设计选取改进的 Minimum Snap 方法进行轨迹规划。

4.1 Minimum Snap 轨迹规划原理

4.1.1 位置轨迹相关量的表示方法

①位置轨迹相关量的表示方法：

位置轨迹相关量，即包括位置、加速度、加加速度和加加加速度等物理量。

在 Minimum Snap 方法中，轨迹采用关于时间 t 的 n 阶多项式表示：

$$\mathbf{p}(t) = p_0 + p_1 t + p_2 t^2 + \cdots + p_n t^n = \sum_{i=0}^n p_i t^i \quad (4.1)$$

其中： $p_0, p_1, \cdots, p_n$ 为 $n+1$ 个轨迹参数，设为参数向量 $\mathbf{p}$ ，即：

$$\mathbf{p} = [p_0 \ p_1 \ \cdots \ p_n]^T \quad (4.2)$$

轨迹规划的目标就是求解出合适的参数向量 $\mathbf{p}$ ，也就构造出了合适的轨迹。轨迹的表示同样可以用向量形式改写如下：

$$\mathbf{p}(t) = [1 \ t \ t^2 \ \cdots \ t^n] \cdot \mathbf{p} \quad (4.3)$$

事实上，上述的轨迹其实就是三维空间中的位置的时间序列，因此可以将其用 x 、 y 和 z 重写如下，注意此时 $\mathbf{p}$ 为参数矩阵：

$$\mathbf{p} = [p_0 \ p_1 \ \cdots \ p_n]^T = \begin{bmatrix} X_0 & Y_0 & Z_0 \\ X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}, \mathbf{p}(t) = [1 \ t \ t^2 \ \cdots \ t^n] \cdot \mathbf{p} \quad (4.4)$$

②其它相关量的表示方法:

在任意时刻 t , 对轨迹多项式 $\mathbf{p}(t)$ 进行求导, 便可以依次得到速度 $\mathbf{v}(t)$ 、加速度 $\mathbf{a}(t)$ 、加加速度 $\mathbf{jerk}(t)$ 和加加加速度 $\mathbf{snap}(t)$, 具体表达式如下:

$$\mathbf{v}(t) = \mathbf{p}'(t) = [0 \ 1 \ 2t \ 3t^2 \ 4t^3 \ \cdots \ nt^{n-1}] \cdot \mathbf{p} \quad (4.5)$$

$$\mathbf{a}(t) = \mathbf{p}''(t) = [0 \ 0 \ 2 \ 6t \ 12t^2 \ \cdots \ n(n-1)t^{n-2}] \cdot \mathbf{p} \quad (4.6)$$

$$\mathbf{jerk}(t) = \mathbf{p}'''(t) = [0 \ 0 \ 0 \ 6 \ 24t \ \cdots \ n(n-1)(n-2)t^{n-3}] \cdot \mathbf{p} \quad (4.7)$$

$$\mathbf{snap}(t) = \mathbf{p}''''(t) = [0 \ 0 \ 0 \ 0 \ 24 \ \cdots \ n(n-1)(n-2)(n-3)t^{n-4}] \cdot \mathbf{p} \quad (4.8)$$

考虑到简单的轨迹可以用多项式进行表示, 但是一段复杂的轨迹仅仅用一段多项式没有办法进行精确刻画。因此采用分段函数的思想, 用分段的多项式曲线来表示一段复杂的轨迹:

$$\mathbf{p}(t) = \begin{cases} [1 \ t \ t^2 \ \cdots \ t^n] \cdot \mathbf{p}_1, & t_0 \leq t < t_1 \\ [1 \ t \ t^2 \ \cdots \ t^n] \cdot \mathbf{p}_2, & t_1 \leq t < t_2 \\ \cdots & \cdots \\ [1 \ t \ t^2 \ \cdots \ t^n] \cdot \mathbf{p}_k, & t_{k-1} \leq t < t_k \end{cases} \quad (4.9)$$

其中: k 为分段轨迹的段数, $\mathbf{p}_i = [p_{i0} \ p_{i1} \ \cdots \ p_{in}]^T$ 为第 i 段轨迹的参数向量。

综上, 本次毕业设计已经获得可以构造基本 Minimum Snap 优化问题的相关量。

4.1.2 优化问题的构造

为了求解轨迹的多项式参数 $\mathbf{p}$, 轨迹规划需要针对参数构建一个带约束优化问题, 求解该优化问题, 便可以得到本次毕业设计想要的轨迹多项式参数 $\mathbf{p}$ 。如前所述, 这个问题的最优指标是时间 t 的相关量。除此之外, Minimum Snap 轨迹规划还需要的是轨迹上已知的一些三维路径点, 以及相应的约束。

①优化指标:

设 Minimum Snap 轨迹规划的优化指标与高阶导数相关, 表达式如下:

$$\min \int_0^T [\mathbf{p}''''(t)]^2 dt \quad (4.10)$$

再设给定的包含起始点与终端点在内的 $k+1$ 个三维路径点为:

$$\mathbf{p}(t_0) \ \mathbf{p}(t_1) \ \cdots \ \mathbf{p}(t_k)$$

因此可以构建优化函数如下所示：

$$\begin{aligned}
 \min \int_0^T [\mathbf{p}''''(t)]^2 dt &= \min \sum_{i=0}^{k-1} \int_{t_i}^{t_{i+1}} [\mathbf{p}_i''''(t)]^2 dt \\
 &= \min \sum_{i=0}^{k-1} \int_{t_i}^{t_{i+1}} \left\{ \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] \cdot \mathbf{p}_i \right\}^T \left\{ \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] \cdot \mathbf{p}_i \right\} dt \\
 &= \min \sum_{i=0}^{k-1} \mathbf{p}_i^T \cdot \left\{ \int_{t_i}^{t_{i+1}} \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right]^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] dt \right\} \cdot \mathbf{p}_i \\
 &= \min \sum_{i=0}^{k-1} \mathbf{p}_i^T \mathbf{Q}_i \mathbf{p}_i \tag{4.11}
 \end{aligned}$$

其中：

$$\begin{aligned}
 \mathbf{Q}_i &= \int_{t_i}^{t_{i+1}} \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right]^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] dt \\
 &= \begin{bmatrix} 0_{4 \times 4} & 0_{4 \times (n-3)} \\ 0_{(n-3) \times 4} & \frac{r!}{(r-4)!} \frac{c!}{(c-4)!} \frac{1}{(r-4) + (c-4) + 1} (t_{i+1}^{r+c-7} - t_i^{r+c-7}) \end{bmatrix} \tag{4.12} \\
 \text{其中：} r \text{ 和 } c \text{ 是矩阵 } \mathbf{Q} &= \begin{bmatrix} \mathbf{Q}_0 & & \\ & \mathbf{Q}_1 & \\ & & \ddots \\ & & & \mathbf{Q}_{k-1} \end{bmatrix} \text{ 的行索引与列索引，均从 } 0 \text{ 开始。}
 \end{aligned}$$

在本次毕业设计的任务中，虽然第 3 章得到的卡尔曼滤波结果是四旋翼与着陆平台之间的相对位姿，但是站在四旋翼的角度去看，在第 2 章的视觉导航中得到的就是着陆平台在四旋翼本体系中的位姿。因此规划轨迹时，已知的是起始点，即四旋翼本体系的原点，以及终端点，即着陆平台相对四旋翼本体系原点的位置。

基于上述考虑，在降落过程中，本次毕业设计使用的 Minimum Snap 轨迹规划方法不断规划的是两点之间的路径，即 $k = 1$ ，并记初始时间为 t_i ，终端时间为 t_f ，代入可得优化指标即为：

$$\begin{aligned}
 \min \int_0^T [\mathbf{p}''''(t)]^2 dt \\
 = \int_{t_i}^{t_f} \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right]^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] dt \tag{4.13}
 \end{aligned}$$

又如果考虑从开始着陆到着陆结束的时间差 $\Delta t = t_f - t_i$ ，这样 Minimum Snap 轨迹规划就不必设计起始时间和终端时间两个量，因此最终设计的优化指标为：

$$\begin{aligned} & \min \int_0^T [\mathbf{p}''''(t)]^2 dt \\ & = \int_0^{\Delta t} \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right]^T \left[0, 0, 0, 0, 24, \dots, \frac{n!}{(n-4)!} t^{n-4} \right] dt \end{aligned} \quad (4.14)$$

其中参数 Δt 便是本次毕业设计任务的待调参数，如果选取的 Δt 过大，则四旋翼规划的路径不足以让四旋翼跟得上不断移动的着陆平台；如果选取的 Δt 过小，则上述的优化问题可能无解，即使有解并规划出了轨迹，四旋翼按照这条轨迹飞行的时候也很容易出现失稳的情况。

优化问题的求解可以使用 CVX 凸优化工具包进行求解。

②等式约束方程：

由于在本次毕业设计的任务中不考虑着陆过程中存在障碍物这一情况，因此优化问题不存在不等式约束，而只存在等式约束方程，且等式约束方程仅由起始点与终端点确定。又如前所述，将着陆平台看成在四旋翼本体系中运动，因此起始点的各约束可以设为没有，只需要设置终端点的约束。

设定终端点的位置、速度、加速度或者更高阶的量为一定值，即可构成一个等式约束：

终端位置约束：

$$\begin{bmatrix} 1 & t_f & t_f^2 & \dots & t_f^n & 0 & \dots & 0 \end{bmatrix} \cdot \mathbf{p} = \mathbf{p}_f \quad (4.15)$$

终端速度约束：

$$\begin{bmatrix} 0 & 1 & 2t_f & \dots & nt_f^{n-1} & 0 & \dots & 0 \end{bmatrix} \cdot \mathbf{p} = \mathbf{v}_f \quad (4.16)$$

终端加速度约束：

$$\begin{bmatrix} 0 & 0 & 2 & \dots & n(n-1)t_f^{n-2} & 0 & \dots & 0 \end{bmatrix} \cdot \mathbf{p} = \mathbf{a}_f \quad (4.17)$$

③姿态轨迹相关量的表示及优化问题的构造：

姿态轨迹相关量，即包括四旋翼欧拉角、角加速度等物理量。

由于四旋翼的运动是多状态量耦合的，因此着陆过程势必存在四旋翼欧拉角的变动。若考虑角度约束的问题，即四旋翼降落到移动的着陆平台的时候，其俯仰角和滚转角不可过大，但是偏航角并无大角度约束。角度约束问题的引入，将姿态相关的量引入到轨迹规划问题中，为了解决这一问题，本次毕业设计改进 Minimum Snap 轨迹规划方

法仿照先前的位置轨迹相关量进行的轨迹规划，建立姿态轨迹相关量进行的轨迹规划。

设姿态轨迹同样是由关于 t 的多项式刻画：

$$\boldsymbol{\theta}(t) = [1 \quad t \quad t^2 \quad \dots \quad t^n] \cdot \boldsymbol{\vartheta} \quad (4.18)$$

$$\text{其中：} \boldsymbol{\vartheta} = [\vartheta_0 \quad \vartheta_1 \quad \dots \quad \vartheta_n]^T = \begin{bmatrix} \phi_0 & \theta_0 & \psi_0 \\ \phi_1 & \theta_1 & \psi_1 \\ \vdots & \vdots & \vdots \\ \phi_n & \theta_n & \psi_n \end{bmatrix} \text{为} n+1 \text{个轨迹参数。}$$

在任意时刻 t ，对轨迹多项式 $\boldsymbol{\theta}(t)$ 进行求导，便可以依次得到角速度 $\boldsymbol{o}(t)$ 、角加速度 $\boldsymbol{\kappa}(t)$ ，具体表达式如下（注意姿态轨迹相关量的轨迹规划更关注角度量的二阶导）：

$$\boldsymbol{o}(t) = \boldsymbol{\theta}'(t) = [0 \quad 1 \quad 2t \quad 3t^2 \quad 4t^3 \quad \dots \quad nt^{n-1}] \cdot \boldsymbol{\vartheta} \quad (4.19)$$

$$\boldsymbol{\kappa}(t) = \boldsymbol{\theta}''(t) = [0 \quad 0 \quad 2 \quad 6t \quad 12t^2 \quad \dots \quad n(n-1)t^{n-2}] \cdot \boldsymbol{\vartheta} \quad (4.20)$$

优化指标为：

$$\min \int_0^T [\boldsymbol{\theta}''(t)]^2 dt =$$

$$\int_0^{\Delta t} \left[0, 0, 2, 6t, 12t^2, \dots, \frac{n!}{(n-2)!} t^{n-2} \right]^T \left[0, 0, 2, 6t, 12t^2, \dots, \frac{n!}{(n-2)!} t^{n-2} \right] dt \quad (4.21)$$

终端角度约束：

$$[1 \quad t_f \quad t_f^2 \quad \dots \quad t_f^n \quad 0 \quad \dots \quad 0] \cdot \boldsymbol{\vartheta} = \boldsymbol{\theta}_f \quad (4.22)$$

终端角速度约束：

$$[0 \quad 1 \quad 2t_f \quad \dots \quad nt_f^{n-1} \quad 0 \quad \dots \quad 0] \cdot \boldsymbol{\vartheta} = \boldsymbol{o}_f \quad (4.23)$$

终端角加速度约束：

$$[0 \quad 0 \quad 2 \quad \dots \quad n(n-1)t_f^{n-2} \quad 0 \quad \dots \quad 0] \cdot \boldsymbol{\vartheta} = \boldsymbol{\kappa}_f \quad (4.24)$$

综上便是姿态轨迹相关量进行的轨迹规划。

优化问题的求解可以使用 CVX 凸优化工具包进行求解。

4.2 本章小结

本章节的主要内容是设计了制导律，即设计了如何从四旋翼当前位置，降落至着陆平台上的降落轨迹。主要运用的方法是 Minimum Snap 轨迹规划方法。主要原理即是：构建轨迹的时间多项式表达，通过求导获得高阶导数量，以轨迹多项式的高阶导数为优



化指标，在已知多项式的若干点的参数的情况下，构建带约束的优化问题并求解，获得轨迹多项式的未知参数，确定规划的轨迹。

从上述主要思想可以看出，该轨迹是一条尽量快的轨迹，为了实时地控制四旋翼按照该轨迹运动，第 5 章将介绍 MPC 方法，以轨迹为参考的状态量，实现上述目标。

5 模型预测控制

模型预测控制（MPC, Model Predictive Control）的作用原理可以简述为：在每一个采样时刻，根据当前获得的量测信息，在线求解一个有限时间开环优化问题，并将得到的控制序列的第一个元素作用于被控对象；在下一个采样时刻，重复上述过程，即用新的量测值作为此时预测系统未来状态的初始条件，刷新优化问题并重新求解。

本次毕业设计已经在第2章中得到了四旋翼运动的线性化的状态方程，第4章中得到了实时规划的降落轨迹，因此本章的核心任务就是：在线性化的状态方程及规划的降落轨迹的基础上，使用 MPC 方法，控制四旋翼沿着降落轨迹运动。为此，本章节需要在了解 MPC 原理的基础上，构建本任务的预测模型，求解滚动优化问题，得到所需的控制量，并进行反馈校正。

5.1 MPC 原理

在第2章中，本次毕业设计通过数学建模获得了如下的线性化的状态空间方程：

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \mathbf{A}_c \begin{bmatrix} X \\ Y \\ Z \\ U \\ V \\ W \\ \phi \\ \theta \\ \psi \\ P \\ Q \\ R \end{bmatrix} + \mathbf{B}_c \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \mathbf{A}_c \mathbf{x} + \mathbf{B}_c \mathbf{u} \quad (2.26)$$

但是，在本次毕业设计的任务中，并直接不关心四旋翼本身的运动状态，而更关注的是四旋翼当前运动的轨迹，与第4章中获得的规划的轨迹相差多少，即需要建立误差的状态空间模型，以构建适用于自主着陆任务的模型预测控制。

5.1.1 预测模型

①建立参考的状态量 $\mathbf{x}_r$

在第4章中，规划的轨迹涉及如下4个相关量，需要注意的是，这些量都是用关于

t 的多项式描述，即是时域连续的：

$$\mathbf{p}(t) = [1 \quad t \quad t^2 \quad \dots \quad t^n] \cdot \mathbf{p} \quad (5.1)$$

$$\mathbf{v}(t) = \mathbf{p}'(t) = [0 \quad 1 \quad 2t \quad 3t^2 \quad 4t^3 \quad \dots \quad nt^{n-1}] \cdot \mathbf{p} \quad (5.2)$$

$$\boldsymbol{\theta}(t) = [1 \quad t \quad t^2 \quad \dots \quad t^n] \cdot \boldsymbol{\theta} \quad (5.3)$$

$$\mathbf{o}(t) = \boldsymbol{\theta}'(t) = [0 \quad 1 \quad 2t \quad 3t^2 \quad 4t^3 \quad \dots \quad nt^{n-1}] \cdot \boldsymbol{\theta} \quad (5.4)$$

考虑到规划的轨迹属于三维空间，而且为了建立参考的状态量 $\mathbf{x}_r$ ，则必须采用描述四旋翼运动的12个状态量对轨迹进行重新刻画：

$$\mathbf{p}(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = [1 \quad t \quad t^2 \quad \dots \quad t^n] \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5.5)$$

$$\mathbf{v}(t) = \begin{bmatrix} U(t) \\ V(t) \\ W(t) \end{bmatrix} = [0 \quad 1 \quad 2t \quad 3t^2 \quad 4t^3 \quad \dots \quad nt^{n-1}] \cdot \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (5.6)$$

$$\boldsymbol{\theta}(t) = \begin{bmatrix} \phi(t) \\ \theta(t) \\ \psi(t) \end{bmatrix} = [1 \quad t \quad t^2 \quad \dots \quad t^n] \cdot \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (5.7)$$

$$\mathbf{o}(t) = \begin{bmatrix} P(t) \\ Q(t) \\ R(t) \end{bmatrix} = [0 \quad 1 \quad 2t \quad 3t^2 \quad 4t^3 \quad \dots \quad nt^{n-1}] \cdot \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (5.8)$$

此时：

$$\mathbf{x}_r =$$

$$[X(t) \quad Y(t) \quad Z(t) \quad U(t) \quad V(t) \quad W(t) \quad \phi(t) \quad \theta(t) \quad \psi(t) \quad P(t) \quad Q(t) \quad R(t)]^T \quad (5.9)$$

将上式离散化可得：

$$\mathbf{x}_r = \begin{bmatrix} X_1 & Y_1 & Z_1 & U_1 & V_1 & W_1 & \phi_1 & \theta_1 & \psi_1 & P_1 & Q_1 & R_1 \\ X_2 & Y_2 & Z_2 & U_2 & V_2 & W_2 & \phi_2 & \theta_2 & \psi_2 & P_2 & Q_2 & R_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_m & Y_m & Z_m & U_m & V_m & W_m & \phi_m & \theta_m & \psi_m & P_m & Q_m & R_m \end{bmatrix}^T \quad (5.10)$$

其中：因为当前时刻规划出的轨迹，四旋翼只会跟踪该轨迹的一段，等下一时刻的轨迹已规划完毕，四旋翼再跟踪新的轨迹，同样也是只跟踪一段，重复上述过程。所以参数 m 即是表征每次跟踪的轨迹长度，实际通过调整确定的参数。

②误差的状态空间模型

整个控制环节关注的重点是让四旋翼尽可能沿规划出的轨迹降落至移动的着陆平台上。因此需要构建如下的误差的状态空间模型：

误差的状态量：

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_r \quad (5.11)$$

误差模型的离散化:

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k \quad (5.12)$$

③误差模型的递推预测

可以看出，离散后的误差的状态空间模型是关于 k 的递推表达式，可以向后递推若干步，递推4次的结果如下：

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k \quad (5.13)$$

$$\tilde{\mathbf{x}}_{k+2} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k+1} + \tilde{\mathbf{B}}\mathbf{u}_{k+1} = \tilde{\mathbf{A}}(\tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k) + \tilde{\mathbf{B}}\mathbf{u}_{k+1} \quad (5.14)$$

$$\tilde{\mathbf{x}}_{k+3} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k+2} + \tilde{\mathbf{B}}\mathbf{u}_{k+2} = \tilde{\mathbf{A}}(\tilde{\mathbf{A}}(\tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k) + \tilde{\mathbf{B}}\mathbf{u}_{k+1}) + \tilde{\mathbf{B}}\mathbf{u}_{k+2} \quad (5.15)$$

$$\tilde{\mathbf{x}}_{k+4} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k+3} + \tilde{\mathbf{B}}\mathbf{u}_{k+3} = \tilde{\mathbf{A}}(\tilde{\mathbf{A}}(\tilde{\mathbf{A}}(\tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k) + \tilde{\mathbf{B}}\mathbf{u}_{k+1}) + \tilde{\mathbf{B}}\mathbf{u}_{k+2}) + \tilde{\mathbf{B}}\mathbf{u}_{k+3} \quad (5.16)$$

仿照上述递推过程，依次可以往后预测 n_p 次，即预测到状态 $\tilde{\mathbf{x}}_{k+n_p}$ 。

其中：参数 n_p 为 MPC 的预测时域，在本次毕业设计任务中属于待调参数。需要注意的是，如果预测时域选择过大，那么越往后预测，模型将越不准；相反，如果预测时域选择过小，那么后续构建的优化问题求解出的控制量参数，也将不能很好地控制四旋翼沿着规划轨迹降落。

将上述 $\tilde{\mathbf{x}}_{k+1}, \tilde{\mathbf{x}}_{k+2}, \dots, \tilde{\mathbf{x}}_{k+n_p}$ 整合成一个新的状态向量，则向后预测 n_p 步的状态方程整合成大矩阵表示如下：

$$\mathbf{Y} = \begin{bmatrix} \tilde{\mathbf{x}}_{k+1} \\ \tilde{\mathbf{x}}_{k+2} \\ \vdots \\ \tilde{\mathbf{x}}_{k+n_p} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}} \\ \tilde{\mathbf{A}}^2 \\ \vdots \\ \tilde{\mathbf{A}}^{n_p} \end{bmatrix} \tilde{\mathbf{x}}_k + \begin{bmatrix} \tilde{\mathbf{B}} & 0 & \dots & 0 \\ \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{B}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}}^{n_p-1}\tilde{\mathbf{B}} & \tilde{\mathbf{A}}^{n_p-2}\tilde{\mathbf{B}} & \dots & \tilde{\mathbf{B}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n_p-1} \end{bmatrix} = \boldsymbol{\Psi}\tilde{\mathbf{x}}_k + \boldsymbol{\Xi}\mathbf{U}_k \quad (5.17)$$

进一步考虑问题：通过预测时域 n_p 的选择，可以获得 n_p 个待求解的控制量：

$$[\mathbf{u}_k \quad \mathbf{u}_{k+1} \quad \dots \quad \mathbf{u}_{k+n_p-1}]^T$$

然而事实上，在实际的控制时不会将全部 n_p 个控制量运用到四旋翼沿轨迹降落的控制过程中，而是会人为地选择前 n_c 个控制量参与到滚动优化的环节中（ $n_c \leq n_p$ ），即：

$$[\mathbf{u}_k \quad \mathbf{u}_{k+1} \quad \dots \quad \mathbf{u}_{k+n_c-1}]^T$$

其中：参数 n_c 为 MPC 的控制时域，在本次毕业设计任务中属于待调参数。需要注意的是，如果控制时域选择过小，那么解算出的控制量将不足以精确控制四旋翼运动；

相反，如果预测时域选择过大，那么后续构建的优化问题的求解耗时将明显增加，将不能保证四旋翼控制的实时性。

如果将控制时域 n_c 也考虑在内，则向后预测 n_p 步并取前 n_c 个控制量参与到滚动优化环节中的状态方程，整合成大矩阵的表示改写如下：

$$\begin{aligned} \mathbf{Y} = \begin{bmatrix} \tilde{\mathbf{x}}_{k+1} \\ \tilde{\mathbf{x}}_{k+2} \\ \vdots \\ \tilde{\mathbf{x}}_{k+n_p} \end{bmatrix} &= \begin{bmatrix} \tilde{\mathbf{A}} \\ \tilde{\mathbf{A}}^2 \\ \vdots \\ \tilde{\mathbf{A}}^{n_p} \end{bmatrix} \tilde{\mathbf{x}}_k + \begin{bmatrix} \tilde{\mathbf{B}} & 0 & \cdots & 0 \\ \tilde{\mathbf{A}}\tilde{\mathbf{B}} & \tilde{\mathbf{B}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}}^{n_p-1}\tilde{\mathbf{B}} & \tilde{\mathbf{A}}^{n_p-2}\tilde{\mathbf{B}} & \cdots & \tilde{\mathbf{A}}^{n_p-n_c}\tilde{\mathbf{B}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n_c-1} \end{bmatrix} \\ &= \boldsymbol{\Psi} \tilde{\mathbf{x}}_k + \boldsymbol{\Xi} \mathbf{U}_k \end{aligned} \quad (5.18)$$

上式即为最终的误差模型的递推预测。

5.1.2 滚动优化与反馈校正

本章的核心任务是求解出让四旋翼尽量沿规划的降落轨迹进行着陆的控制量，为此需要构建一个优化问题，通过在线求解该优化问题来实时获得控制量，并将控制量反馈回四旋翼运动的状态空间模型中。

①滚动优化

一方面，求解出的控制量应当使得四旋翼当前运动轨迹与规划的降落轨迹之间的误差尽可能地小；另一方面，考虑到四旋翼的电机动力提供有限，故求解出的控制量本身也需要尽可能地小，或者说待求解的控制量与零控制量的偏差也需要尽可能地小。基于上述两点考虑，选取优化指标如下：

$$\min \mathbf{Y}^T \mathbf{M} \mathbf{Y} + \mathbf{U}^T \mathbf{N} \mathbf{U} \quad (5.19)$$

其中： $\mathbf{M}$ 和 $\mathbf{N}$ 分别是 $\mathbf{Y}$ 和 $\mathbf{U}$ 的加权矩阵，在本次毕业设计的任务中属于待调参数。

此外，可以在求解优化问题的时候提供控制量的约束，约束形式为不等式约束：

$$\mathbf{U}_a \leq \mathbf{U} \leq \mathbf{U}_b \quad (5.20)$$

其中参数 $\mathbf{U}_a$ 和 $\mathbf{U}_b$ 根据自主着陆任务过程中的实际情况进行设定。

优化问题的求解可以使用 CVX 凸优化工具包进行求解。

②反馈校正

上述求解出的控制量 $\mathbf{U}$ 为有 n_c 个元素的向量，即：

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n_c-1} \end{bmatrix} \quad (5.21)$$



选择其中第一个元素 u_k ，反馈到四旋翼运动的状态空间模型 $\dot{x} = A_c x + B_c u$ 中进行控制，即完成了反馈校正环节。

5.2 本章小结

本章节重点介绍了 MPC 控制四旋翼按照第 4 章规划的轨迹进行降落的原理。主要原理即是：将时域连续的轨迹及其导数离散化为参考状态量，构建当前状态量与参考状态量的误差的状态空间模型，并将其递推预测，选择误差最小化与控制量最小化为优化指标，进行滚动优化求解，将求解出的控制量反馈回四旋翼状态空间模型，以保证其沿轨迹运动，最终实现自主降落。

为了检验本章设计的 MPC 控制方法是否可行，以及第 2 章的建模、第 3 章的导航和第 4 章的轨迹规划是否正确可行，将在第 6 章中进行仿真实验。

6 系统仿真验证

本次毕业设计与前人工作^[7,25]中的仿真方法保持一致，选用机器人操作系统（ROS, Robot Operating System）进行仿真。四旋翼的模型、着陆平台的模型以及自主着陆过程的可视化的效果都通过 ROS 内置的 Gazebo 物理引擎进行仿真和展示。

在搭建仿真平台的基础上，本章涉及的仿真实验包括如下三种：

- a、四旋翼自主降落于静止的着陆平台；
- b、四旋翼自主降落于合作运动的着陆平台，着陆平台做匀速圆周运动；
- c、四旋翼自主降落于非合作运动的着陆平台，着陆平台做随机运动。

仿真的主要目标即是，验证基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼自主降落的可行性。

6.1 ROS 仿真平台的搭建

为了在 ROS 中搭建完整的仿真系统，需要在 Gazebo 中搭建出四旋翼模型、着陆平台模型。由于 Gazebo 的插件库中提供了相关模型的插件，因此插件不必从头开始编写，只需要在其中选取合适的插件，修改其中的参数并进行使用即可。

6.1.1 四旋翼模型

①物理模型及相关参数

四旋翼模型选择 TUM ARDrone 模型，TUM ARDrone 是慕尼黑工业大学的视觉团队开发的一款飞行器自主导航的 ROS 程序包，其中包括了使用的四旋翼模型。这款四旋翼的 SimuLink 模型如图 6.1 所示：

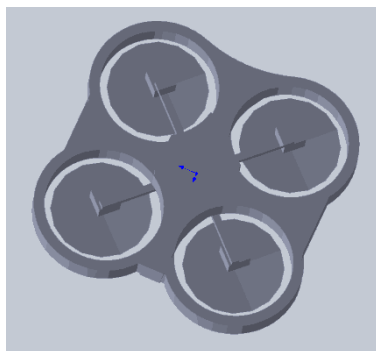


图 6.1 四旋翼的 SimuLink 模型



四旋翼的相关物理量如表 6.1 所示：

质量(kg)	质量惯性矩(kg · m ²)			d ₊ (m)
1.477	J _{xx} = 0.01152	J _{yy} = 0.01152	J _{zz} = 0.02180	0.18707

表 6.1 四旋翼模型参数

②在 ROS 中的控制

ROS 中的信息是以“消息”的形式在“话题”中传播，即一个话题可接收若干个发来的消息，在这个话题进行处理之后，将新的若干消息再发布出去。TUM ARDrone 四旋翼的控制话题有多个，其中最直接的是“/ardrone/cmd_vel”话题，该话题直接向四旋翼发布速度指令，控制四旋翼按照指定速度运动。但实际仿真的时候并不采用该话题控制四旋翼，这是因为在第 5 章设计 MPC 的时候，已经确定将四个电机的转速作为控制量。基于上述考虑，最终采用“/ardrone/navdata/motor1”话题控制 1 号电机的转速，2 号、3 号和 4 号电机的控制话题类似。

6.1.2 着陆平台模型

①物理模型及相关参数

由于仿真中需要着陆平台移动，因此着陆平台实际上是由平台板和移动机器人两部分组成。其中平台板提供导航所需的视觉标志（平台板表面贴上 AprilTag 标志）和足够四旋翼降落区域，移动机器人则负责搭载平台板，静止或者按照仿真设定的运动规律进行移动。

移动机器人模型选择 Summit XL 模型。Summit XL 是 Robotnik 公司开发制作的一款多功能、高移动性的室内/室外移动机器人，它在 Gazebo 中的模型已经被收入 Gazebo 插件库，因此直接导入使用即可。平台板即是一块扁平的板，它直接固连在移动机器人的顶部，平台板表面直接贴上 AprilTag 标志，选择的是该标志库中的 TAG36H11 系列的 0 号标志。

整个着陆平台（移动机器人及其顶部的平台板）的 Gazebo 模型如图 6.2 所示：

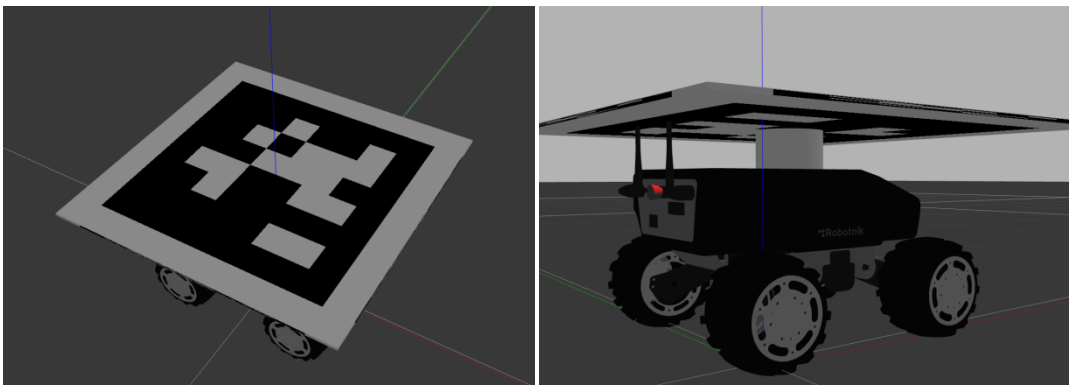


图 6.2 着陆平台的 Gazebo 模型

着陆平台的相关物理量如表 6.2 所示：

平台板			
质量(kg)	长(m)	宽(m)	高(m)
0.5	1	1	0.01
移动机器人底座			
质量(kg)	质量惯性矩(kg · m ²)		
25	$J_{xx} = 1.391$	$J_{xx} = 6.853$	$J_{xx} = 6.125$
移动机器人轮子			
质量(kg)	半径(m)	轮宽(m)	
6.5	0.117	0.14	

表 6.2 着陆平台模型参数

②在 ROS 中的控制

着陆平台的控制主要是对移动机器人的控制，Summit XL 移动机器人的运动控制主要通过底盘下的四个轮子的转速控制实现。

6.2 仿真一：着陆平台静止

为了初步验证四旋翼能够跟踪移动的着陆平台并进行降落，本节设计的仿真任务的具体说明如下：

- a、四旋翼从地面起飞，飞至惯性系的(0,0,-10)点处，即在惯性系原点正上方 10 米处并悬停（在仿真中惯性系的 O_iZ_i 轴正向设置为向下）；

b、着陆平台在惯性系的(0,0,0)点处静止不动，即停在惯性系原点；

c、四旋翼停止悬停，垂直降落在着陆平台上。

整个仿真过程在 ROS 中进行，但利用消息的通信机制，将四旋翼状态信息发送给 Python 脚本，用以绘制各状态量的曲线。

6.2.1 仿真一的实验结果

实验结果主要包括：制导律规划的轨迹与四旋翼实际降落的轨迹的比较图，描述四旋翼运动状态的状态量随时间变化的图。

①制导律规划的轨迹与四旋翼实际降落的轨迹的比较图：

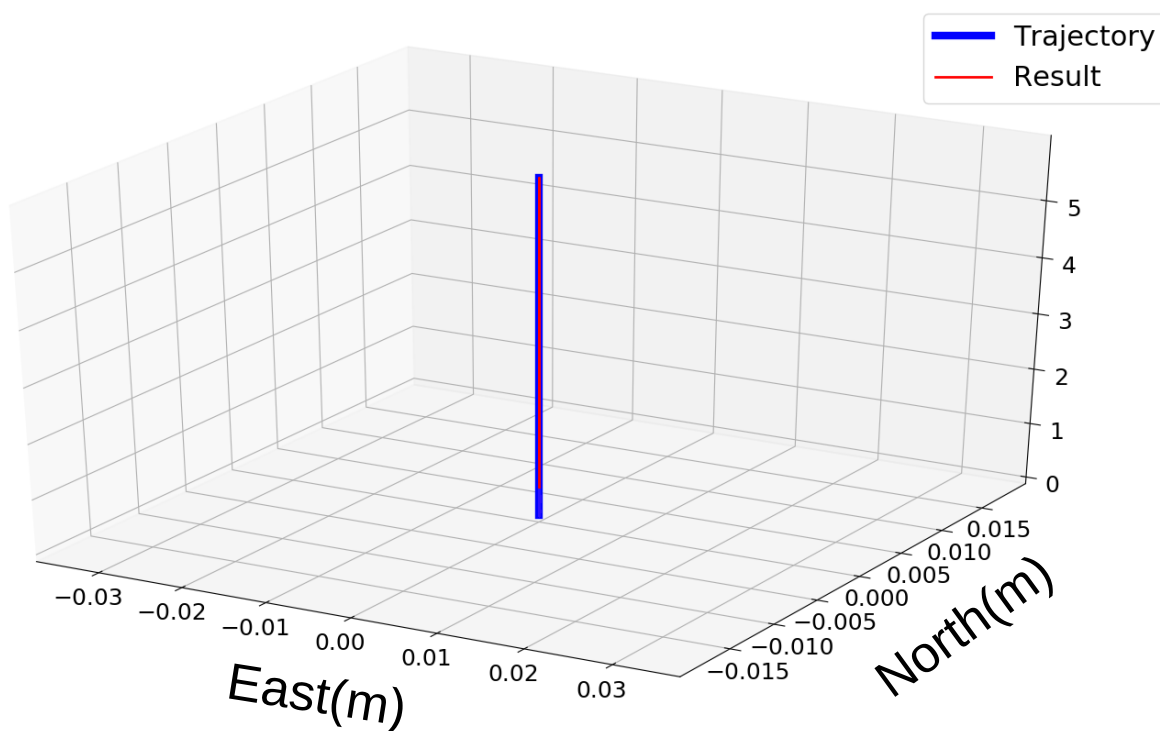


图 6.3 制导律规划的轨迹与四旋翼实际降落的轨迹的比较图

如图 6.3 所示，红色的曲线为四旋翼在上述任务中实际运动的轨迹；蓝色的曲线为制导律实时规划的轨迹。可以看出，红色的曲线首先从惯性系原点起飞，并飞至 10 米高处，该段曲线对应任务环节 a；随后四旋翼从悬停状态变为跟踪降落状态，其运动轨迹是新的红色曲线，与此同时制导律开始规划轨迹，即出现蓝色的线，该部分对应任务环节 c。需要说明的是，最终红色曲线并非停止在 0 米高处，这是因为着陆平台自身是有高度的。

②描述四旋翼运动状态的状态量随时间变化的图

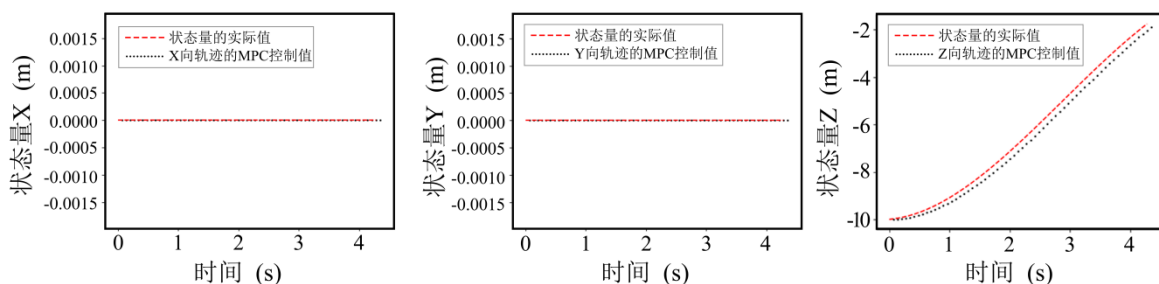


图 6.4 仿真中四旋翼位置状态量随时间变化图

如图 6.4 所示，红色的线是四旋翼状态量(X,Y,Z)在仿真过程中的实际的值，黑色的线是规划出的轨迹经 MPC 递推预测、滚动优化，再反馈回模型中推算出的下一刻的状态量(X,Y,Z)，可以看出红色的线精密跟随黑色的线，这说明 MPC 确实在控制着四旋翼按照规划的轨迹进行降落，且从 ROS 中的几次实验中测得降落的误差上限约为 1cm。

6.2.2 仿真一的实验结论

基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼降落静止的着陆平台的算法是可行的。

6.3 仿真二：着陆平台匀速圆周运动

为了进一步验证四旋翼能够跟踪移动的着陆平台并进行降落，本节设计的仿真任务的具体说明如下：

- a、四旋翼从地面起飞，飞至惯性系的(0,0,-10)点处，即在惯性系原点正上方 10 米处并悬停（在仿真中惯性系的 O_iZ_i 轴正向设置为向下）；
- b、着陆平台在惯性系的(0,0,0)点处静止不动，即停在惯性系原点；
- c、着陆平台开始从惯性系的(-5,5,0)点处开始作匀速圆周运动，同时四旋翼停止悬停，跟踪着陆平台并降落。

整个仿真过程在 ROS 中进行，但利用消息的通信机制，将四旋翼状态信息发送给 Python 脚本，用以绘制各状态量的曲线。

6.3.1 仿真二的实验结果

实验结果主要包括：制导律规划的轨迹与四旋翼实际降落的轨迹的比较图，描述四旋翼运动状态的状态量随时间变化的图。

①制导律规划的轨迹与四旋翼实际降落的轨迹的比较图：

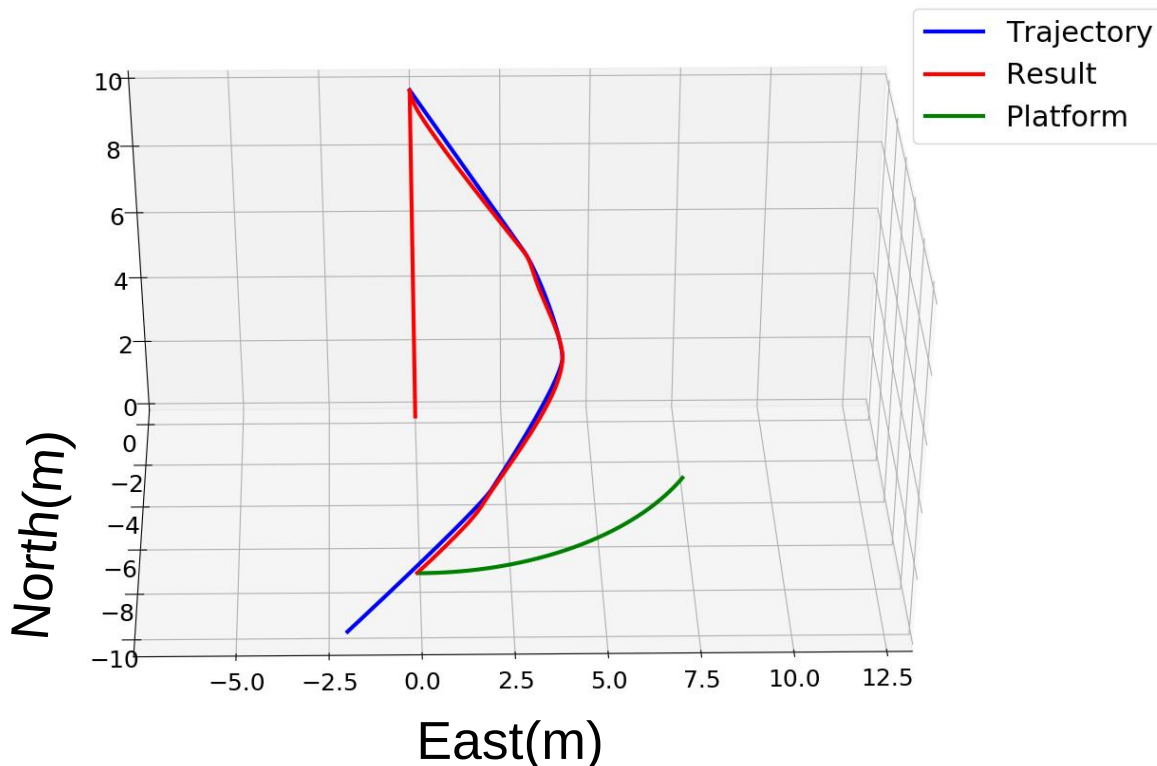


图 6.5 制导律规划的轨迹与四旋翼实际降落的轨迹的比较图

如图 6.5 所示，红色的曲线为四旋翼在上述任务中实际运动的轨迹；蓝色的曲线为制导律实时规划的轨迹；绿色的线是着陆平台的运动轨迹。可以看出，红色的曲线首先从惯性系原点起飞，并飞至 10 米高处，该段曲线对应任务环节 a；随后由于着陆平台的移动，四旋翼从悬停状态变为跟踪降落状态，其运动轨迹是新的一段红色曲线，与此同时制导律开始规划轨迹，即出现蓝色的线，该部分对应任务环节 c。需要说明的是，最终红色曲线并非停止在 0 米高处，这是因为着陆平台自身是有高度的。

②描述四旋翼运动状态的状态量随时间变化的图

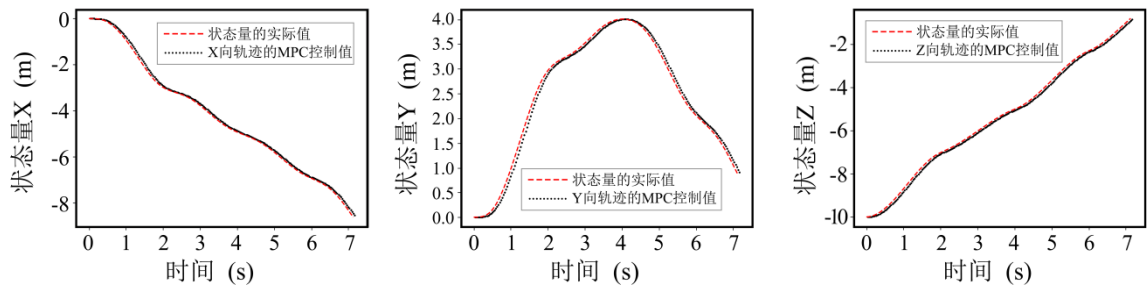


图 6.6 仿真中四旋翼位置状态量随时间变化图

如图 6.6 所示，红色的线是四旋翼状态量(X,Y,Z)在仿真过程中的实际的值，黑色的线是规划出的轨迹经 MPC 递推预测、滚动优化，再反馈回模型中推算出的下一刻的状态量(X,Y,Z)，可以看出红色的线精密跟随黑色的线，这说明 MPC 确实在控制着四旋翼按照规划的轨迹进行降落，且从 ROS 中的几次实验中测得降落的误差上限约为 30cm。

6.3.2 仿真二的实验结论

基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼跟踪合作运动的着陆平台并降落的算法是可行的。

6.4 仿真三：着陆平台随机运动

为了进一步验证四旋翼能够跟踪非合作运动的着陆平台并进行降落，本节设计的仿真任务的具体说明如下：

- a、四旋翼从地面起飞，飞至惯性系的(0,0,-10)点处，即在惯性系原点正上方 10 米处并悬停（在仿真中惯性系的 O_iZ_i 轴正向设置为向下）；
- b、着陆平台在惯性系的(0,0,0)点处静止不动，即停在惯性系原点；
- c、着陆平台开始随机运动，同时四旋翼停止悬停，跟踪着陆平台并降落。

整个仿真过程在 ROS 中进行，但利用消息的通信机制，将四旋翼状态信息发送给 Python 脚本，用以绘制各状态量的曲线。

6.4.1 仿真三的实验结果

实验结果主要包括：制导律规划的轨迹与四旋翼实际降落的轨迹的比较图，描述四旋翼运动状态的状态量随时间变化的图。

①制导律规划的轨迹与四旋翼实际降落的轨迹的比较图：

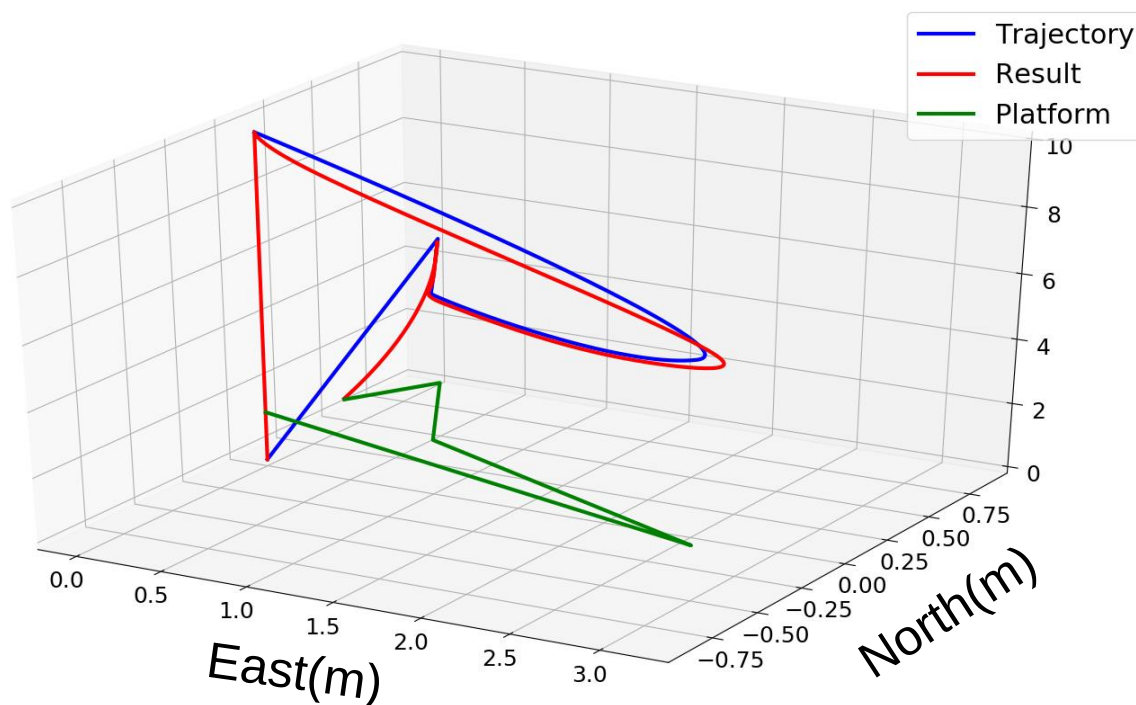


图 6.5 制导律规划的轨迹与四旋翼实际降落的轨迹的比较图

如图 6.5 所示，红色的曲线为四旋翼在上述任务中实际运动的轨迹；蓝色的曲线为制导律实时规划的轨迹；绿色的线是着陆平台的运动轨迹。可以看出，红色的曲线首先从惯性系原点起飞，并飞至 10 米高处，该段曲线对应任务环节 a；随后由于着陆平台的移动，四旋翼从悬停状态变为跟踪降落状态，其运动轨迹是 new 的一段红色曲线，与此同时制导律开始规划轨迹，即出现蓝色的线，该部分对应任务环节 c。需要说明的是，最终红色曲线并非停止在 0 米高处，这是因为着陆平台自身是有高度的。

②描述四旋翼运动状态的状态量随时间变化的图

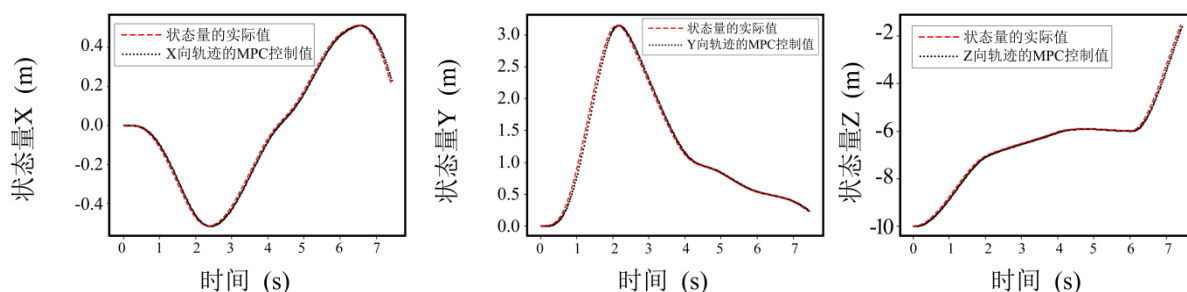


图 6.6 仿真中四旋翼位置状态量随时间变化图

如图 6.6 所示，红色的线是四旋翼状态量(X,Y,Z)在仿真过程中的实际的值，黑色的



线是规划出的轨迹经 MPC 递推预测、滚动优化，再反馈回模型中推算出的下一刻的状态量(X,Y,Z)，可以看出红色的线精密跟随黑色的线，这说明 MPC 确实在控制着四旋翼按照规划的轨迹进行降落，且从 ROS 中的几次实验中测得降落的误差上限约为 1m。

6.4.2 仿真三的实验结论

基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼跟踪非合作运动的着陆平台并降落的算法是可行的。

6.5 本章小结

本章节主要对基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼自主降落的可行性进行了验证。首先对整个仿真环境的搭建做了介绍，重点说明了 ROS 中四旋翼模型与着陆平台模型的搭建与展示；接着设计三个仿真实验，即四旋翼自主降落于静止的着陆平台实验、四旋翼自主降落于匀速圆周运动的着陆平台实验和四旋翼自主降落于随机运动的着陆平台实验，并分析实验结果的相关图，得到“基于 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 实现四旋翼跟踪着陆平台并降落的算法是可行的”这一结论。

总结与展望

本文针对四旋翼无人机基于视觉信息实现自主着陆的问题，建立了一套涉及视觉导航、制导和控制的自主降落系统，实现了四旋翼平稳降落在移动的着陆平台上的任务。

通过建立四旋翼动力学模型，采用 AprilTag 视觉库的视觉标志和位姿估计算法，实时解算四旋翼与着陆平台的相对位姿，并利用卡尔曼滤波减小误差。再根据相对位姿信息，使用 Minimum Snap 轨迹规划算法，实时规划从四旋翼到着陆平台的降落轨迹，并将该轨迹作为参考状态量，构建实际状态量与参考状态量之间的误差的状态空间模型，采用 MPC 方法，以较高的精度完成了四旋翼自主降落在移动的着陆平台的任务，且降落过程平稳。

本文的主要工作与贡献如下：

一、基于 AprilTag 视觉标志及位姿估计算法，实现了四旋翼与着陆平台的相对位姿估计。经由“边缘检测—四边形检测—四边形编码”流程获得 AprilTag 标志的检测结果，通过编码信息的处理获得相对位姿信息，再利用卡尔曼滤波对相对位置信息进行处理，且位置误差在 20cm 以内。

二、设计了包含姿态信息的 Minimum Snap 轨迹规划算法。在原算法的位置轨迹多项式的基础上，构建姿态轨迹多项式，在轨迹规划中考虑角度及其约束，构建优化问题，利用 CVX 进行求解，实时规划出以轨迹多项式的高阶导数为优化指标，带有位置量和角度量约束的降落轨迹。

三、设计了基于“实际-参考状态量误差”的模型预测控制方法。以规划的轨迹为四旋翼参考状态量，构建参考状态量与实际状态量之间的误差的状态空间模型，滚动构建优化问题并利用 CVX 进行求解，得到控制量并反馈回四旋翼运动模型成功进行控制。

四、在 ROS 中搭建仿真环境，分别针对静止、匀速圆周运动和随机运动平台进行着陆虚拟仿真实验，四旋翼都以较高的精度成功降落在了移动着陆平台，实验所得的误差上限分别为 1cm、30cm 和 1m。

本文成功地将 AprilTag 视觉导航、Minimum Snap 轨迹规划和 MPC 结合，完成了四旋翼自主降落移动的着陆平台的任务，验证了该方法的可行性，为无人机的空地协同运动控制技术提供技术借鉴。

考虑到本次毕业设计是国内首次尝试将 AprilTag 视觉导航、Minimum Snap 轨迹规



划和 MPC 结合在一起实现四旋翼的自主降落，导航、制导和控制三个模块的理论衔接还有待优化，除此之外实际仿真中也有很多需要改善的地方。总之，这是一次初步的尝试，但是有很多值得展望的地方。

将来的研究还需要对以下几个方面进行改善：

（1）在四旋翼建模环节，从质量惯性矩矩阵的定义可以看出，该矩阵描述是四旋翼在不挂载的情况下，其本体系各轴上的转动惯量。然而事实情况是，四旋翼要利用视觉信息，则必然需要在四旋翼底部安装相机。那么如何精确地对安装相机后的四旋翼的质量惯性矩矩阵进行建模有待进一步考虑。

同样，四旋翼建模部分的陀螺力在建立线性化的状态空间模型时忽略，但是如果精确刻画作用在四旋翼上的力和力矩，这显然是不合理的。因此，如何建立含陀螺力的线性化状态空间模型也值得考虑^[26]。

（2）在视觉导航环节，可以尝试对 AprilTag 算法做适用于具体任务的改进。比如用于边缘检测的 LBP 算子，它对光照条件的变化具有很强的鲁棒性，但是它会检测出很多明显不能构成多边形的边缘，同时边缘图像的噪点较多；倘若实际任务场景中光照条件变化不明显，则可以尝试使用 Canny 算子^[27]进行边缘检测，减少检测出的边缘，降低噪点。此外，如何对 AprilTag 位姿估计算法解算出的四旋翼与着陆平台相对位姿信息做更为优良的滤波也有待进一步探究。

（3）在制导与控制环节，一个非常值得探究的问题是：Minimum Snap 轨迹规划采用了构建优化问题的方式求解轨迹参数，而 MPC 同样采用了构建优化问题的方式求解控制量，考虑到两次求解优化问题势必会影响实时性，因此需要找到简化优化问题的求解的方法，或者思考是否能将这两个优化问题整合成一个优化问题进行求解。

（4）在仿真环节，本次毕业设计提及的所有的待调参数，均可待进一步调整。可以通过进一步调参，研究其背后的经验规律，甚至对其进行精确建模。

（5）与前人的工作^[7,28-30]相比，缺少实物系统的搭建与实验。从仿真到实物，还有一段路需要走，比如如何得到一个四旋翼实物的相关模型参数，此时系统参数辨识的相关方法的运用也值得考虑。



致谢

在北航的四年时光真如白驹过隙。弹指一瞬间，我已经到了毕业的季节。在这四年里的喜怒哀乐、所学所感所悟，也都随着“毕业设计”这四个字在任务手账上的划去，而逐渐封存在我的青春信封里，将信口火漆热烈的红色，收藏在我的记忆的百宝箱的最里层。但是，在合上百宝箱之前，我的脑海中又浮现出毕业设计这6个月以来的点点滴滴，心中不免又泛起感激的涟漪。

首先，我必须感谢我的指导老师张冉老师。张老师的博学的专业知识，丰厚的工程经验，一丝不苟的学术态度，以及热情友好的师长风范，俨然是矗立在我面前的四座高峰，令我不禁心生敬畏，却又激起我也去攀登这些高峰的憧憬和动力。在毕业设计之初，我的文献调研能力及展示 PPT 的能力十分欠缺，张老师不辞辛劳，逐字逐句地指导我改进开题报告及开题答辩的 PPT，令我十分感动。此后的每周进度汇报的组会上，张老师总是能为我的毕业设计遇到的问题，提出一针见血的见解，每次都能让我的思绪如“柳暗花明又一村”般豁然开朗。此外，犹记得年初还在校的那段时光里，有一次和张老师同乘校车从沙河返回学院路，在路上张老师借月喻人，对当时做毕业设计没有信心的我的一番的开导，让我醍醐灌顶，投入到毕业设计的工作当中。

接着，我要感谢李惠峰老师。李老师给我的感觉一直是像个宽厚温柔的长辈，她总是站在切实为学生考虑的角度，发自肺腑地对我的毕业设计、乃至将来的学业发展，提出非常有用的指导；总是站在一个过来人的角度、一个视野更为开阔的角度，对正处于迷茫当中的我们，用温柔的话语，召唤驱散黑暗的明光。除了平时的教导，李老师在毕业设计的后期，更是再百忙之中抽出时间，为我的毕业设计最终论文进行修改、提供指导性的意见，对此我不胜感激。

我还要感激负责指导我的毕业设计的张柯柯和陈星伦两位学长，以及机械学院的陈成学长。在我看来，“学长”二字，其负甚重。学，意味着更多的知识、更多的技能、更多的经验；长，意味着更多的阅历、更宽的眼界、更阔的胸襟。而张柯柯和陈星伦两位学长，无愧于“学长”二字。张柯柯学长比我拥有更多的图像方面的知识，并在我的毕业设计的视觉导航部分亲力亲为，把手相教；他对 ROS 运用的熟练程度也比我更胜一筹，在我搭建 ROS 仿真系统的时候，甚至逐行阅读我那没有注释的代码，推敲可能出现的问题。陈星伦学长则是比我拥有更多的制导、控制方面的理论知识，在我尝试使



用 MPC 控制的时候，指导我如何用代码实现控制，在我对设计制导律完全没有头绪的时候，立即找出若干篇文献，并陪我阅读文献，寻找思路。陈成学长是我早先在做“冯如杯”项目时认识的前辈，并且完全出于一个学长对学弟的关心，在毕业设计期间凭借他丰富的 ROS 工程经验，对我的毕业设计提供了大力的援助，可以说如果没有陈成学长的帮助，我根本不可能独自完成整个 ROS 仿真的实现。

我也要感谢我的同学朋友们。都冠全、刘一平同学凭借其专业知识，对我的 SimuLink 模型的测量与绘制提供帮助；何林坤、刘子博同学慷慨分享他们的毕业设计的一些答辩 PPT 模板，为我的答辩提供帮助；王舒阳、张有驰同学由于毕业设计同为处理四旋翼对象，对我的四旋翼运动的数学模型的构建提供帮助；张孟希、赵欣玥、邓阳艳、李晓涵、郭威威等同学在我毕业设计的攻关阶段为我排除来自考研复试的忧虑；赵恩泽同学为我的数据图表的绘制提供帮助；好友沙昊、陈学颖陪我度过每天的晚饭后出门消食的放松时光；好友鲍振宇、罗绮则是与我一起分享兴趣爱好。

感谢我的本科的学生辅导员王艳和王子涵，在毕业设计期间，他们也十分辛劳，尤其是考研复试的那段时间，辅导员的高效率工作，减轻了我的复试压力，间接为我完成毕业设计提供了帮助。

最后，我要重点感谢我的家人。父母一直是我坚强的后盾，他们永远是最懂我、最为我着想的人。在我的毕业设计期间，他们减少我的家务量，及时对我进行疏导，备齐各种文件，带我出去放风……他们的爱是细致的、周到的、完备的，是推动我完成毕业设计的关键保障。

希望我在后续的研究生生活中可以获得更大的进步、更多的快乐，我十分憧憬！



参考文献

- [1] Fouad Y , Nassim R , Omar B , et al. Optimization of Energy Consumption for Quadrotor UAV[C]// International Micro Air Vehicle Conference and Flight Competition (IMAV). 2017.
- [2] 周思全,化永朝,董希旺,李清东,任章.面向空地协同作战的无人机-无人车异构时变编队跟踪控制[J].航空兵器,2019,26(04):54-59.
- [3] Borowczyk A, Nguyen D T, Nguyen A P V, et al. Autonomous landing of a quadcopter on a high-speed ground vehicle[J]. Journal of Guidance, Control, and Dynamics, 2017, 40(9): 2378-2385.
- [4] 孟学斌. 基于机器视觉的旋翼无人机自主着陆系统设计与实现[D].内蒙古工业大学,2019.
- [5] 单一. 基于视觉导航的四旋翼无人机自主着陆控制研究[D].南京航空航天大学,2018.
- [6] Lange S , Sunderhauf N , Protzel P . A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments[C]// Advanced Robotics, 2009. ICAR 2009. International Conference on. IEEE, 2009.
- [7] 王兆哲. 基于视觉定位的四旋翼无人机移动平台自主降落技术研究[D].哈尔滨工业大学,2019.
- [8] Olson E . AprilTag: A robust and flexible visual fiducial system[C]// Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
- [9] Richardson A , Strom J , Olson E . AprilCal: Assisted and repeatable camera calibration[C]// Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013.
- [10] Wang J , Olson E . AprilTag 2: Efficient and robust fiducial detection[C]// IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE, 2016.
- [11] 盖文东,张宁,张婧,张桂林,黄梁松. 一种基于负比例导引的无人机自动避撞方法[P]. 山东省: CN106647810B,2019-06-18.
- [12] Mellinger D , Kumar V . Minimum snap trajectory generation and control for quadrotors[C]// Robotics and Automation (ICRA), 2011 IEEE International Conference



- on. IEEE, 2011.
- [13]石喜玲,孙运强,李静,余红英.四旋翼动力学建模及非线性 PID 轨迹跟踪控制[J].科学技术与工程,2020,20(06):2489-2493.
- [14]陈洪亮,张向文.基于 LQR 控制器设计的无人机姿态控制[J].自动化应用,2019(10):1-4.
- [15]周杰. 基于模型预测控制的四旋翼飞行控制算法研究[D].南京航空航天大学,2017.
- [16]David A Forsyth. Computer Vision: A Modern Approach 2nd Edition[J]. 2012.
- [17]陈骥. 四旋翼无人飞行器动力学建模及控制技术的研究[D]. 2016.
- [18]Ojala T , Matti Pietikäinen, Topi Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns[J]. Classification with Local Binary Patterns, 2002.
- [19]Suzuki S , Be K . Topological structural analysis of digitized binary images by border following[J]. Computer Vision Graphics & Image Processing, 1985, 30(1):32-46.
- [20]Mccallum D , Avis D . A linear algorithm for finding the convex hull of a simple polygon[J]. Information Processing Letters, 1979, 9(5):201-206.
- [21]Douglas D H , Peucker T K . ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE[J]. Cartographica the International Journal for Geographic Information & Geovisualization, 1973, 10(2):112-122.
- [22]Hamming, R. W . Error Detecting and Error Correcting Codes[J]. Bell System Technical Journal, 1950, 29(2):147-160.
- [23]Run L , Zong-Cai R , Sui W . Research on Algorithms of Homography Matrix in Plane Measuring[J]. Acta Simulata Systematica Sinica, 2001.
- [24]陈玉寅,张文安,杨旭升. 基于视觉传感器和无迹卡尔曼滤波器的运动刚体位姿估计 [C]. 中国自动化学会控制理论专业委员会.第 36 届中国控制会议论文集（G）.中国自动化学会控制理论专业委员会:中国自动化学会控制理论专业委员会,2017:1266-1269.
- [25]Palafox P R , Mario Garzón, Valente J , et al. Robust Visual-Aided Autonomous Takeoff, Tracking, and Landing of a Small UAV on a Moving Landing Platform for Life-Long Operation[J]. Applied ences, 2019, 9(13):2661.



- [26]邵鹏杰, 董文瀚, 马骏,等. 考虑桨叶陀螺效应的四旋翼定点飞行动态面控制[J]. 飞行力学, 2015(05):43-47.
- [27]Canny J . A Computational Approach To Edge Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, PAMI-8(6):679-698.
- [28]Borowczyk A , Nguyen D T , André Phu-Van Nguyen, et al. Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle[J]. Journal of Guidance Control and Dynamics, 2017, 40(9):1-8.
- [29]Feng Y , Zhang C , Baek S , et al. Autonomous Landing of a UAV on a Moving Platform Using Model Predictive Control[J]. Drones, 2018, 2(4).
- [30]Maces-Hernandez J A , Defay F , Chauffaut C . Autonomous landing of an UAV on a moving platform using model predictive control[C]// 2017 11th Asian Control Conference (ASCC). 2017.