



单位代码 10006

学 号 16231235

分 类 号 TP249

密 级 公开

**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)

## 基于视觉定位的多无人机编队 控制方法研究

院（系）名称 高等理工学院

专 业 名 称 自动化

学 生 姓 名 李 谨 杰

学院指导教师 郑建英

工业部门导师

2020 年 5 月

# 北京航空航天大学

## 本科毕业设计（论文）任务书

I、毕业设计（论文）题目：

基于视觉定位的多无人机编队控制方法研究

---

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

毕业设计使用的原始资料为 Google Scholar 中与视觉定位、强化学习和编队控制有关的参考文献, 计算机视觉与强化学习领域的英文书籍, 与 OpenCV, ROS, Gazebo 等软件库的官网文档, 并依托于实验室的 smart-swarm 无人机仿真平台开展。应实现的要求为建立仿真系统, 完成定位-决策-控制的系统闭环, 编队节点 5 个, 编队过程考虑避障任务, 定位精度小于 10 厘米。

---

III、毕业设计（论文）工作内容：

毕设任务主要围绕无人机编队仿真系统的实现展开, 主要包括以下三部分:

1. 建立视觉模型, 通过计算机视觉技术处理二维码获得视觉定位信息, 同时融合惯导传感器信息, 以实现在稀疏二维码环境中的无人机定位任务。
  2. 在无人机定位的基础上, 将编队任务合理转化为马尔科夫决策过程, 应用强化学习算法实现相应任务的训练与部署, 实现多架无人机的编队控制。
  3. 在仿真环境中, 通过多架小型无人机开展基于视觉定位的多无人机编队控制算法实验验证, 并分析实验结果。
-

#### IV、主要参考资料：

1. Speck C, Bucci D J. Distributed UAV Swarm Formation Control via Object-Focused, Multi-Objective SARSA[A]. 2018 Annual American Control Conference (ACC)[C]. Washington, DC, USA: IEEE, 2018: 6596-6601.
2. Krogus M, Haggenmiller A, Olson E. Flexible Layouts for Fiducial Tags[A]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[C]. Washington, DC, USA: IEEE Computer Society, 2019: 1898-1903.
3. Knopp M, Aykın C, Feldmaier J, et al. Formation Control Using GQ ( $\lambda$ ) Reinforcement Learning[A]. 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)[C]. Washington, DC, USA: IEEE, 2017: 1043-1048.

\_\_\_\_高等理工学院\_\_\_\_院（系）\_\_\_\_自动化\_\_\_\_专业类\_\_\_\_162320\_\_\_\_班

学生姓名：\_\_\_\_李谨杰\_\_\_\_

指导教师：\_\_\_\_郑建英\_\_\_\_

毕业设计（论文）时间：自\_\_\_\_2020\_\_\_\_年\_\_\_\_02\_\_\_\_月\_\_\_\_10\_\_\_\_日至\_\_\_\_2020\_\_\_\_年\_\_\_\_05\_\_\_\_月\_\_\_\_31\_\_\_\_日

答辩时间：\_\_\_\_2020\_\_\_\_年\_\_\_\_06\_\_\_\_月\_\_\_\_03\_\_\_\_日 成绩：\_\_\_\_

指导教师签字：\_\_\_\_\_\_\_\_

兼职教师或答疑教师（并指出所负责部分）：

系（教研室）主任签字：\_\_\_\_

注：任务书应该附在已完成的毕业设计（论文）的首页。

## 关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本文不包含其他人已经发表或撰写过的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中做出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：\_\_\_\_\_李谨杰

日期： 2020 年 06 月 02 日

## 学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

本论文不涉及任何保密内容。若有涉密内容，本人愿意承担相关法律责任。

学位论文作者签名：\_\_\_\_\_李谨杰

日期：2020 年 06 月 02 日

学院指导教师签名：\_\_\_\_\_郑建英

日期：2020 年 06 月 02 日

基于视觉定位的多无人机编队控制方法研究

李谨杰

北京航空航天大学



## 基于视觉定位的多无人机编队控制方法研究

学 生：李谨杰

指导教师：郑建英

### 摘 要

随着近几年计算机视觉技术和人工智能技术的蓬勃发展，集群化、自主化、智能化将成为无人机发展的重要方向。本论文将视觉定位技术与人工智能算法应用于无人机集群领域，实现了一种基于视觉定位与强化学习的多无人机编队控制系统，提升了无人机执行任务的自主性与抗干扰能力。论文的主要研究内容如下：

首先，研究视觉定位的基本原理并建立视觉模型，进而应用计算机视觉技术，实现摄像头标定、二维码检测、解码和位姿求解等步骤以获得视觉定位信息，并与惯性导航方式进行信息融合，以在 Gazebo 仿真环境中搭建的稀疏二维码场景下实现自主、连续、精确的定位效果。其次，研究强化学习的基本原理，将编队问题分解为目标跟踪任务与避碰任务并分别设计状态空间、动作空间和奖励函数，并结合 Dyna-Q 强化学习算法设计训练流程。进而基于 ROS/Gazebo 仿真环境搭建虚拟现实仿真平台，分别完成对目标追踪算法与避碰算法的训练。最后，应用视觉定位算法与强化学习编队算法，完成五架无人机包含内部避碰的协同定位与控制任务，取得良好的协同编队效果。

**关键词：**编队控制，二维码定位，强化学习，无人机



# The formation control method of multiple UAVs based on a visual positioning system

Author : LI Jinjie

Tutor : ZHENG Jianying

## Abstract

The vigorous development of computer vision and artificial intelligence technologies in recent years has created conditions for enhancing the intelligence of drones. By applying these technologies to the field of UAV formation, this paper implements a multi-UAV formation control system based on visual positioning and reinforcement learning algorithms, which improves the autonomy and anti-interference of UAVs in the complex environment. The main research contents are as follows:

For the first part, this paper initially studies the basic principles of visual positioning and establishes a visual model, then applies computer vision technology to achieve camera calibration, tags detection, decoding, and pose solving steps to obtain visual positioning information. Next, the author integrates visual navigation with inertial navigation to provide continuous and accurate positioning information in the sparse-tag scenario built in the Gazebo simulation environment. For the second part, the paper initially studies the basic principles of reinforcement learning, then decomposes the formation problem into target tracking task and collision avoidance task and designs the Markov decision process, respectively. After that, the paper provides the training process based on the Dyna-Q reinforcement learning algorithm and trains the target tracking task and collision avoidance task respectively in a ROS / Gazebo based virtual reality simulation platform. Finally, the paper completes the collaborative positioning and control process of five drones including the collision avoidance task, achieving a great formation effect.

**Key words:** Formation control, Localization by QR-code, Reinforcement learning, Unmanned Aerial Vehicle



## 目 录

1	绪论 .....	1
1.1	课题背景及目的 .....	1
1.2	国内外研究状况 .....	3
1.2.1	无人机定位方法研究状况 .....	3
1.2.2	无人机编队控制研究现状 .....	5
1.3	论文构成及研究内容 .....	8
2	视觉定位原理及实现 .....	10
2.1	视觉模型与定位流程 .....	10
2.2	摄像头的标定 .....	14
2.3	计算机视觉算法 .....	15
2.3.1	AprilTags 的检测算法 .....	15
2.3.2	AprilTags 的解码算法 .....	17
2.3.3	PnP 求解算法 .....	18
2.4	视觉定位与 IMU 组合导航 .....	19
2.4.1	组合导航的基本原理 .....	19
2.4.2	apriltag_ros 工具包 .....	20
2.4.3	robot_localization 工具包 .....	21
2.5	本章小结 .....	22
3	强化学习编队原理及实现 .....	23
3.1	编队问题分析 .....	23
3.2	马尔科夫决策过程设计 .....	24
3.2.1	目标追踪问题的 MDP 设计 .....	25
3.2.2	避碰问题的 MDP 设计 .....	26
3.3	强化学习编队算法 .....	28
3.3.1	训练阶段的算法 .....	28
3.3.2	编队执行阶段的算法 .....	30
3.4	本章小结 .....	30
4	仿真实验及结果分析 .....	31





4.1 总系统介绍 .....	31
4.2 视觉定位部分的实验 .....	32
4.2.1 检测解码算法测试 .....	33
4.2.2 组合导航定位实验 .....	34
4.3 强化学习编队部分的实验 .....	37
4.3.1 目标追踪与避碰能力的训练 .....	37
4.3.2 编队效果测试 .....	41
4.4 本章小结 .....	44
结论 .....	45
致谢 .....	46
参考文献 .....	47
附录 .....	51
附录 A: apriltag_ros 配置文件 .....	51
附录 B: robot_localization 配置文件 .....	52

## 1 绪论

### 1.1 课题背景及目的

无人飞行器 (Unmanned Aerial Vehicle, UAV), 俗称无人机, 是一种靠无线电设备遥控或由自身程序操控的不需要驾驶员的飞行器。相比有人驾驶飞机, 无人机具有机身尺寸小巧、飞行机动性高、作业隐蔽性好、应用成本低等优点<sup>[29]</sup>。

近年来, 随着电子元件的体积与价格不断降低与嵌入式设备计算能力的提升, 全世界越来越多的研究机构开始研究小型、与人交互良好的无人机, 以实现其在不同场景下的应用, 如图 1.1<sup>[1]</sup>。



(a) 大疆 T20 植保无人机



b) 亚马逊物流无人机



c) Zipline 无人机正在投递药物

图 1.1 无人机在各行各业的应用

随着应用场景的复杂化和执行任务的多样化, 无人机对导航性能的要求也越来越高。传统的导航方法基于惯性测量单元 (IMU) 和全球卫星导航系统 (GNSS) 提供定位信息。但是, 卫星导航系统依赖于卫星信号, 当信号被干扰或被遮挡时不再可用。军用方面, 伊朗曾利用 GPS 干扰技术, 成功捕获了美军 RQ-170 “哨兵” 无人机, 如图 1.2 (a)。民用方面, 2018 年 5 月 1 日晚, 从西安古城墙起飞进行灯光秀表演的 1374 架无人机由于 GPS 信号遭到恶意干扰, 最终纷纷掉下, 如图 1.2 (b)。在室内场景中, GNSS 系统则完全无法使用。因此, 为保证无人机在卫星信号受到干扰的情况或室内场景中正常飞行, 研制不依赖外部信号的自主导航技术受到了各国的重视。



(a) 伊朗捕获的 RQ-170 “哨兵” 无人机



(b) 西安受到干扰的无人机群

图 1.2 西安受到干扰的无人机群

自然界中的沙蚁、鸽子、海龟等生物仅通过地球自然场信息即可进行远距离导航，这为自主导航的研究提供了很多思路。有研究表明，人和动物至少有 80% 以上的外界信息经视觉获得。因此，在新兴的自主导航技术中，视觉导航（Visual Navigation），即利用视觉传感器感知周围环境信息作为航空器飞行依据的导航技术，得到最为广泛的关注。

另一方面，单架无人机越来越显现出其能力的局限性：由于飞行速度、电池续航等限制，导致作业覆盖面积小、效率低；由于单个无人机的抗风险能力差，导致执行任务时容错性较低。为解决这些局限，多无人机编队的概念应运而生，成为多无人机系统协同控制领域的一个热门研究方向。多无人机编队指的是，每个无人机根据任务要求进行动作，最终形成期望队形的过程。根据执行任务的不同，无人机编队分为固定编队（如多边形编队、圆形编队等）和时变编队（包括形状变化或位置变化的编队），如图 1.3。多无人机编队研究既包括编队的形成、保持和动态变化，又包含编队任务规划、分配、优化和动态调节等环节。与单架无人机执行任务相比，多无人机编队可以覆盖更大的作业面积，也具备较强的系统抗风险能力。因此，针对多无人机系统编队问题的研究能够充分挖掘无人机的潜力，使其拥有更广阔的应用前景。



(a) “红箭” 飞机组成固定编队



(b) 自然界中的鸟群组成时变编队

图 1.3 不同的编队类型

本项目主要研究视觉定位与编队形成两方面的问题。

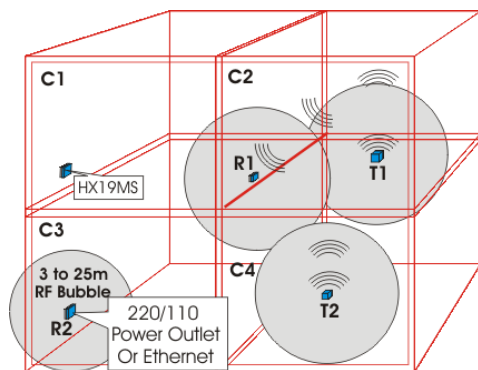
定位部分的主要目标为实现视觉定位技术,基于计算机视觉算法解算无人机自身的位置信息。这种定位方式一方面计算量较小,即使部署在计算能力较弱的嵌入式设备中,也可以满足实时解算的导航要求;另一方面仅依赖环境中的二维码标识,环境配置成本低,自主性强。这种定位方式从一定程度上解决了无人机的室内导航问题,又具备较高的可行性,对解决无人机在室内环境的定位问题具有一定意义。

编队控制部分的主要目标为将强化学习算法应用在无人机编队控制领域,以实现编队组成与内部避碰任务,并在物理仿真平台上实现。传统算法,如基于行为或基于虚拟结构的编队控制算法,往往将被控对象进行了一定程度的简化以便于数学推导,这导致其难以应用于实际问题。而强化学习是一种以环境反馈作为输入的、适应环境的机器学习方法,通过试错与环境交互获得策略的改进,具有自学习和在线学习的特点<sup>[2]</sup>。这种自学习的特点可以针对不同环境与机体的差异进行实时优化,也许可以较好解决传统多智能体理论在实际应用中遇到的问题,具有深远的研究意义。

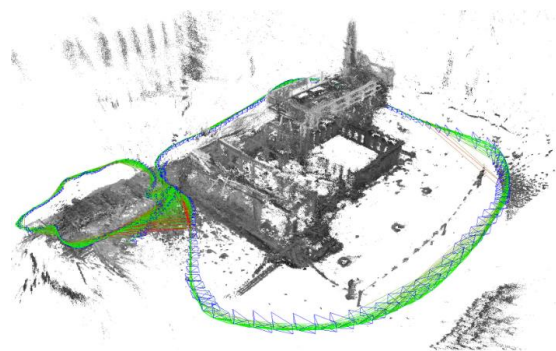
## 1.2 国内外研究状况

### 1.2.1 无人机定位方法研究状况

国内外研究的室内定位方法主要有两类,分别是基于通讯原理的定位方式,如 WI-FI 定位、超宽带(UWB)定位、蓝牙(Bluetooth)定位、射频识别(RFID)定位等技术,如图 1.4a,和基于计算机视觉的定位方式,如即时定位与地图构建(SLAM)和二维码定位等技术,如图 1.4b。二维码定位技术对环境要求简单,定位效果准确,计算资源消耗少,因此本项目采用二维码定位技术作为无人机的定位方法。



(a) RFID 室内定位原理



(b) 双目视觉 SLAM<sup>[32]</sup>

图 1.4 几种典型的室内定位技术

对于二维码定位技术,一部分的研究集中在二维码的设计与检测上。由于传统的二维码包含信息过多,组成复杂,存在容错性差、距离远时难以检测等问题,往往不适用于机器人实验系统。因此,世界各地的研究机构提出了多种简化编码方式,典型代表有 ARToolkit, ArTag 和 AprilTags 等。ARToolkit 是采用图像作为内部编码的代表,自 1999 年提出后在学术界得到广泛使用<sup>[4]</sup>。它的标识物由外部的黑色边框和内部的图像组成,如图 1.5 (e)所示。ARTag<sup>[5]</sup>是在 ARToolkit 基础上发展起来的一种正方形标识物系统,由包含了校验位的二进制编码方案来进行错误检测和校正,如图 1.5 (g)。而本项目采用的 AprilTags 二维码是由密歇根大学 Edwin Olson 教授及其实验室团队于 2011 年提出的一种标记识别算法,如图 1.5 (i) <sup>[6]</sup>。AprilTags 经历了 2016 年 AprilTags 2<sup>[7]</sup>和 2019 年 AprilTags 3<sup>[8]</sup>的两次迭代,检测速度更快,容错性更高,且有开源代码进行参考,因此我将采用 AprilTags 二维码作为本项目的标识物。

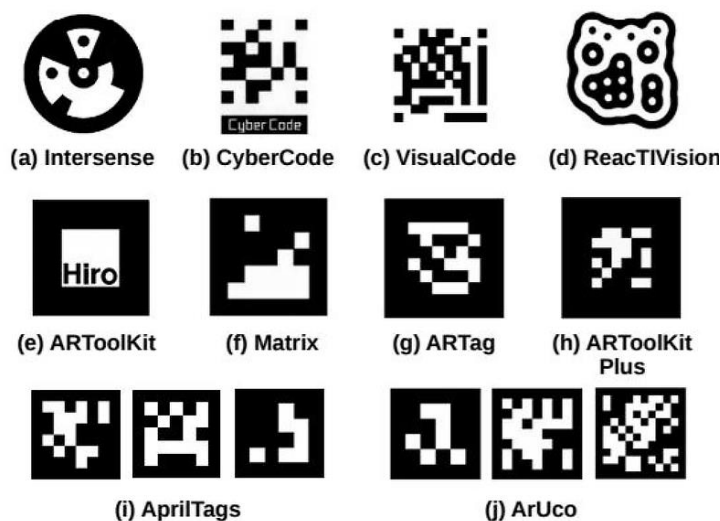


图 1.5 一些典型标识物<sup>[3]</sup>

另一部分的研究则集中于从图像中的标识物形态提取摄像头的相对位置和姿态信息,即求解 PnP 问题。解法主要分为迭代法和非迭代法两类,迭代法速度慢但精度高,非迭代法速度快,但精度一般略低于迭代法。非迭代法中较为常用的算法是 Lepetit V.等人提出的 EPnP 算法,该算法的复杂度为  $O(n)$ ,对于点数较多的 PnP 问题非常高效<sup>[9]</sup>。另外, Toby Collins 等人则于 2014 年提出了 IPPE 算法,专门针对共面的 PnP 问题进行优化,大大提升了针对共面正方形标识物进行姿态检测的精度、速度和抗干扰能力,是目前求解 P4P 问题最高效的解法之一<sup>[10]</sup>。

还有一部分的研究则集中在如何将二维码定位信息融入机器人导航。业界普遍使用

卡尔曼滤波组合导航技术进行惯导信息和视觉信息的融合<sup>[12]</sup>。Lei Cheng 团队提出一种基于惯导信息与立体视觉的组合导航算法，通过线性卡尔曼滤波（KF）进行惯性传感器信息与视觉信息的融合，在移动机器人平台进行测试，取得了较好的效果<sup>[11]</sup>。Sungsik Huh 团队则设计了一种装有 GPS、相机、激光传感器和惯性传感器的无人机导航模块，通过扩展卡尔曼滤波（EKF）技术估计无人机的状态，模块的系统框图如图 1.6<sup>[13]</sup>。

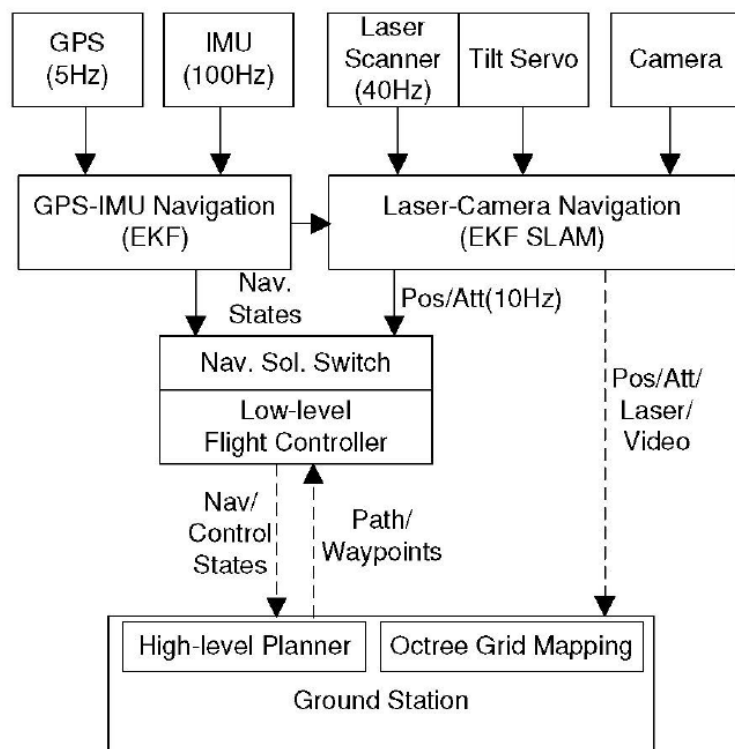


图 1.6 Sungsik Huh团队设计的组合导航模块的系统框图<sup>[13]</sup>

总的来说，基于二维码的视觉定位技术比较成熟，已经有大量研究提出各种各样的标记物，并将视觉定位信息与多传感器信息相融合，以应用在机器人研究中。但是在无人机编队领域，大多数的研究依赖动作捕捉装置完成无人机的定位任务，成本十分高昂，且无法在大型室内环境使用。很少有研究将视觉定位方式应用于无人机编队领域，为编队中的每架无人机提供自主的高精度定位信息。因此，本课题的重点在于将视觉定位算法实现，并与惯导信息组合导航，以应用于无人机编队任务。

### 1.2.2 无人机编队控制研究现状

国外内针对不同任务背景下的多无人机协同编队技术已有大量的研究成果，部分研究成果已经通过小型无人机集群实验的形式进行了算法验证。传统的无人机编队控制算法主要基于一致性理论。吴晋团队基于 $\Omega$ 图，即领头者只发送信息，不接收信息，而其



他的相邻个体彼此通信的通信拓扑结构,并结合一致性理论设计控制率,分析稳定性与收敛性,最终成功进行仿真验证,几类简单的 $\Omega$ 图如图 1.7<sup>[14]</sup>。Han 同样基于 leader-follower 模型结合一致性理论设计控制率,所不同的是 Han 的控制算法关注了时变构型,并且考虑了时延因素对编队控制的影响<sup>[15]</sup>。总的来说,基于一致性的编队控制理论已经取得了较好的仿真结果,但在实际应用时往往受限于各种物理环境的约束。这些约束既包括无人机之间的通信延迟、通信拓扑网络改变、空气阻力干扰等系统外界约束,又包括模型不确定、系统非线性、控制输入饱和等系统内部约束。

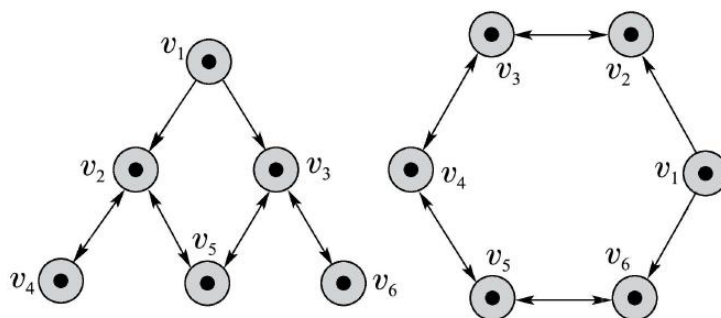
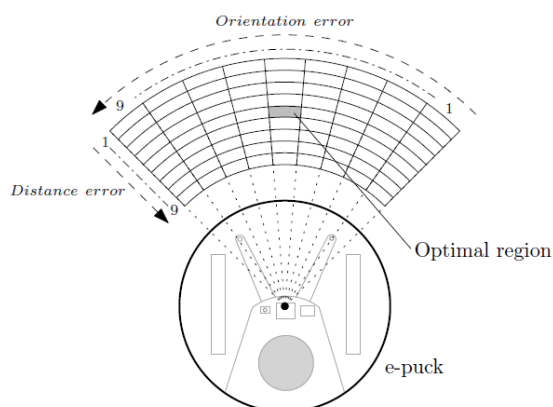


图 1.7 几类简单的 $\Omega$ 图<sup>[14]</sup>

而强化学习不受预设的控制框架限制,可以根据实际情况学习控制策略,是编队控制实物应用的理想方案<sup>[20]</sup>。强化学习经过几十年的发展,已经在解决机器人与环境交互的问题上取得一定的成就,特别是 2015 年 Google DeepMind 将深度网络引入强化学习体系提出的深度强化学习算法,大大提升了强化学习的性能。近几年的研究已经证明,良好训练的深度强化学习算法在复杂任务上的表现,例如在电脑游戏<sup>[16]</sup>和围棋<sup>[17]</sup>领域,甚至可以超过人类专家。强化学习在具有连续的状态、动作空间的学习任务中也具有良好的效果,这预示着它在机器人领域的应用潜力<sup>错误!未找到引用源。</sup>。苏黎世联邦理工学院的 Jemin Hwangbo 等人完全基于强化学习训练了针对四旋翼无人机四个电机转速的控制算法,使得从不同方向、不同初始速度扔出的四旋翼均可以回到初始位置并悬停,飞行过程如图 1.8。Hwangbo 基于强化学习训练的控制率可以直接从仿真环境迁移到实物上,计算速度只有  $7\mu s$ ,且在极端条件下依然具有很好的控制效果<sup>[20]</sup>。

图 1.8 从扔出到稳定悬停过程中的无人机姿态变化<sup>[20]</sup>

针对强化学习在机器人编队中的应用问题,德国慕尼黑工业大学的 Martin Knopp 等人合理划分状态空间和动作空间,设计奖励函数,基于 leader-follower 模型将  $GQ(\lambda)$  强化学习算法应用在轮式机器人 e-puck 上进行编队控制,较好解决了四个机器人的编队跟踪问题。其关于状态的划分和实物实验如图 1.9<sup>[21]</sup>。



(a) 状态空间的划分



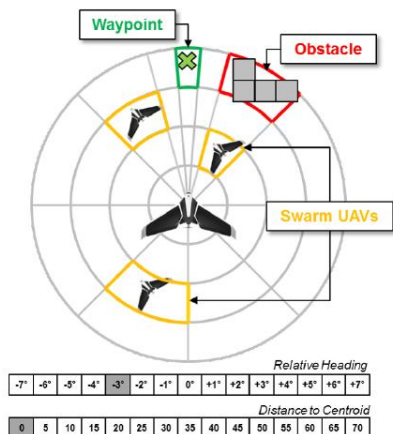
(b) 实物实验

图 1.9 通过强化学习在e-puck机器人上实现编队控制<sup>[21]</sup>

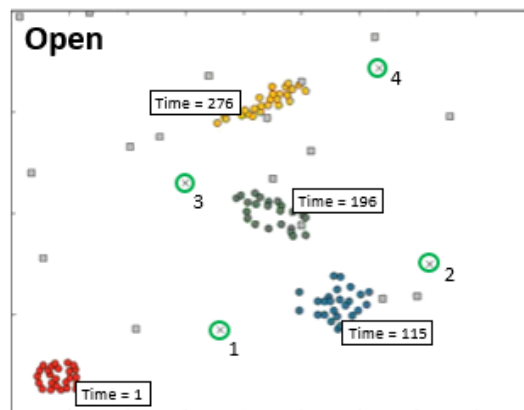
将强化学习算法与无人机编队结合的研究数量较少,比较有代表性的是加拿大皇家军事学院的 Shao-Ming Hung 和美国洛克希德马丁公司高等技术实验室 Christian Speck 等人的工作。Hung 将 Q-Learning 算法应用于固定翼编队保持。在其实验中,飞机飞行的速度大小恒定,通过控制固定翼飞机滚转的角度来影响飞机的左转或右转,进而使 follower 飞机均保持在 leader 周围的一个圆环内<sup>[22]</sup>。而 Speck 从 Reynolds 于 1987 年提出的 Boid 鸟群模型出发,将无人机的集群任务分为向中心靠拢(Cohesion),速度匹配(Alignment),内部避碰(Cohesion),目标追踪(Target Seek)和外部避障(Obstacle Avoidance)共五个子任务,为每一个子任务通过 SARSA 算法训练一个值函数表(Q table),最终通过几个值函数的加和来决定最终的动作。最终 Speck 实现了几十架无人机抽象模型在仿真



中的集群飞行，其状态空间的划分如图 1.10a，实验结果如图 1.10b<sup>[23]</sup>。



(a) 状态空间划分



(b) 在开放区域的仿真结果

图 1.10 Speck等人通过强化学习进行编队控制的实验<sup>[23]</sup>

总的来说，将强化学习算法应用在无人机编队控制领域的研究还很少，已有的研究主要基于高度抽象的仿真模型，且多为随机时变编队，无法完成较为精确的固定编队任务。因而本课题的研究重点在于，通过强化学习控制无人机完成精确编队任务，并考虑无人机间的避碰情况。

### 1.3 论文构成及研究内容

本论文的研究内容可分为以下三个部分

#### 1) 视觉定位算法

针对视觉定位问题，首先对无人机建立视觉模型。其次，对镜头进行摄像头标定以获得相机内参数矩阵。接着拍摄图像，编写二维码检测程序、解码程序以获得图片中二维码四个角点的像素坐标和世界坐标。之后通过求解 PnP 算法与坐标变换获得无人机机体坐标系相对于世界坐标系的位置。最后，将视觉定位结果与惯性测量单元(IMU)的数据进行组合导航，得到最终的定位信息。

#### 2) 强化学习编队控制算法

针对编队问题，首先学习强化学习基础知识，接着将编队任务拆分为目标追踪问题与避碰问题，分别设计状态空间，动作空间和奖励函数。之后编写程序，通过 gazebo 物理仿真平台搭建无人机训练环境并训练。训练过程分为两个阶段：第一阶段训练无人机的目标追踪能力；第二阶段在目标追踪能力的基础上训练避障能力。最终将结果应用于五架无人机上实现带避碰效果的编队形成任务。



### 3) 仿真实验的设计与结果分析

通过 python 实现上述算法,借助实验室在 Gazebo 物理仿真引擎开发的 smart-swarm 无人机集群仿真平台,实现定位-决策-控制的系统闭环。最后设计实验,分析验证算法的效果。

本论文主要分为四章进行论述,具体构成如下所述。

第一章是绪论部分。主要介绍了本论文的课题背景及目的,叙述了目前国内外在无人机定位和编队控制领域的研究现状,总述了全文的研究内容。

第二章介绍了视觉定位的原理及实现过程。该章主要分为四个部分,分别是视觉模型的建立,摄像头的标定,计算机视觉算法和 IMU 与视觉定位组合导航。计算机视觉算法部分详细介绍了二维码检测算法、解码算法和求解 PnP 算法。

第三章介绍了强化学习编队的原理及实现过程。该章主要分为三个部分,分别是编队问题的分析,马尔科夫决策过程设计和强化学习算法。算法部分介绍了训练采用的 Dyna-Q 算法,环境交互的流程以及编队执行时的算法。

第四章介绍了仿真实验系统,并对实验结果进行分析。该章主要分为三个部分,分别是总系统的介绍,定位部分的实验和编队部分的实验。定位部分测试了检测算法的速度和准确度,以及视觉与惯导组合导航的定位精度。编队部分展示了目标跟踪和避障任务的训练结果,以及无人机编队避障实验的结果。

## 2 视觉定位原理及实现

为了克服传统卫星导航方式的局限，给室内定位问题提供全自主、低成本、高精度的解决方案，本章节将研究视觉定位的原理与实现流程，以在室内环境中为无人机提供连续准确的定位信息。本章开始介绍了视觉模型的建立过程，接着介绍了摄像头标定，AprilTags 二维码的检测、解码和位姿求解等步骤。为让无人机在视野中没有二维码时也可以获得定位信息，在最后一部分引入惯性测量单元与视觉定位进行组合导航，并介绍了相应原理与工具包的使用。

### 2.1 视觉模型与定位流程

解决视觉定位问题的第一步是建立视觉模型。本论文中涉及到的坐标系定义如下：

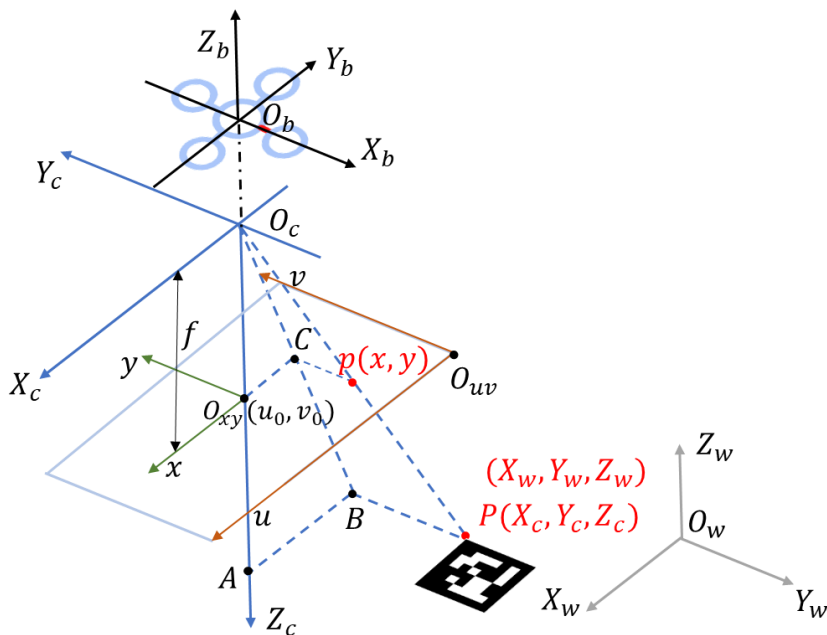


图 2.1 论文中使用的坐标系示意图

**机体坐标系 (Base Coordinates):** 以无人机中心为原点，以前进方向为  $x$  轴，以垂直机体平面向上为  $z$  轴建立的坐标系，以下标为  $b$  的大写字母表示，如  $(X_b, Y_b, Z_b)$ 。

**世界坐标系 (World Coordinates):** 真实世界的立体空间三维直角坐标，以下标为  $w$  的大写字母表示，如  $(X_w, Y_w, Z_w)$ 。

**相机坐标系 (Camera Coordinates):** 以相机的聚焦中心为原点，以光轴为  $Z$  轴建立的三维直角坐标系，以下标为  $c$  的大写字母表示，如  $(X_c, Y_c, Z_c)$ 。

**图像坐标系 (Image Coordinates):** 将相机呈现的三维坐标投影到屏幕上而建立的新

坐标系，不含高度信息，是一个二维坐标系，以不加下标的小写字母表示，如 $(x, y, z)$ 。

**像素坐标系 (Pixel Coordinates):** 将投影的图像坐标离散抽样形成的，以左上角为原点的二维坐标系，也是我们得到的图片上最直观的的坐标系。以小写字母 $(u, v)$ 表示， $u$ 表示水平坐标， $v$ 表示垂直坐标。

本论文根据针孔相机模型建立视觉模型。从物体在真实世界中的物理位置到图像平面的像素位置，主要按该顺序坐标变换：世界坐标系 $(O_w) \rightarrow$  相机坐标系 $(O_c) \rightarrow$  图像坐标系 $(O_{xy}) \rightarrow$  像素坐标系 $(O_{uv})$ 。

### 步骤一：从世界坐标系转换到相机坐标系

该变换属于刚体变换，包括旋转和平移两种变换。根据齐次坐标变换理论，方程为：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & \vec{T} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2.1)$$

在该方程中， $R$  为旋转矩阵，形式为：

$$R = R_1 \cdot R_2 \cdot R_3 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \quad (2.2)$$

其中， $\theta$ ， $\varphi$ 和 $\phi$ 分别为绕世界坐标系的  $x$  轴， $y$  轴和  $z$  轴转动的角度。而 $\vec{T}$ 为平移矩阵，形式为：

$$\vec{T} = \begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T \quad (2.3)$$

其中， $x_0$ ， $y_0$ 和 $z_0$ 为世界坐标系原点在相机坐标系中的坐标。

### 步骤二：从相机坐标系转换到图像坐标系

根据三角形相似关系  $\begin{cases} \Delta ABO_c = \Delta oCO_c \\ \Delta PBO_c = \Delta pCO_c \end{cases}$  得到  $\frac{AB}{oC} = \frac{AO_c}{OO_c} = \frac{PB}{pC} = \frac{X_c}{x} = \frac{Z_c}{f} = \frac{Y_c}{y}$ 。整理成

矩阵形式：

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2.4)$$

其中,  $f$  为相机镜头的焦距。

### 步骤三: 从图像坐标系转换到像素坐标系

如图 2.2,  $xoy$  是图像坐标系,  $uo_{uv}v$  是像素坐标系,  $p(x, y)$  是图像中的任意一点。

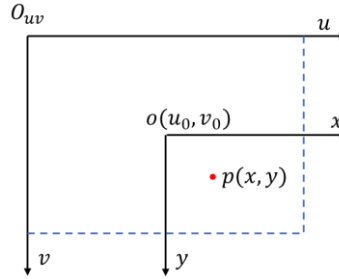


图 2.2 从图像坐标系转换到像素坐标系示意图

假设  $u$  方向的一格像素和  $v$  方向的一格像素在  $x$  坐标,  $y$  坐标中对应的长度分别为  $dx$ ,  $dy$ ,

则可得到几何公式:  $\begin{cases} u = \frac{x}{dx} + u_0 \\ v = \frac{y}{dy} + v_0 \end{cases}$ 。转换为矩阵形式:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.5)$$

其中  $dx$ ,  $dy$  是感光元件每一颗像素的物理尺寸,  $u_0$  和  $v_0$  分别是图像中心点在像素坐标系的坐标。这些参数都是相机本身的参数, 均可通过标定得到。至此, 综合公式(2.1) - (2.5), 可以得到从世界坐标系到像素坐标系的变换公式:

$$\begin{aligned} Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \vec{T} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{f}{dx} & 0 & u_0 & 0 \\ 0 & \frac{f}{dy} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \vec{T} \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_1 \cdot M_2 \cdot X_w \end{aligned} \quad (2.6)$$

其中 $M_1$ 为相机内参数矩阵,  $M_2$ 为相机外参数矩阵。由以上推导可知, 如果已知相机内参数矩阵 $M_1$ , 世界坐标系中的坐标 $(x_w, y_w, z_w)^T$ , 图像坐标系中的坐标 $(u, v)^T$ , 就可以解算位姿矩阵 $M_2$ <sup>[25]</sup>。位姿矩阵 $M_2$ 是我们解算的最终目标, 也是视觉定位的结果。为减少对环境中二维码的依赖, 在最后加入组合导航环节, 本项目视觉定位的总流程如下图所示。

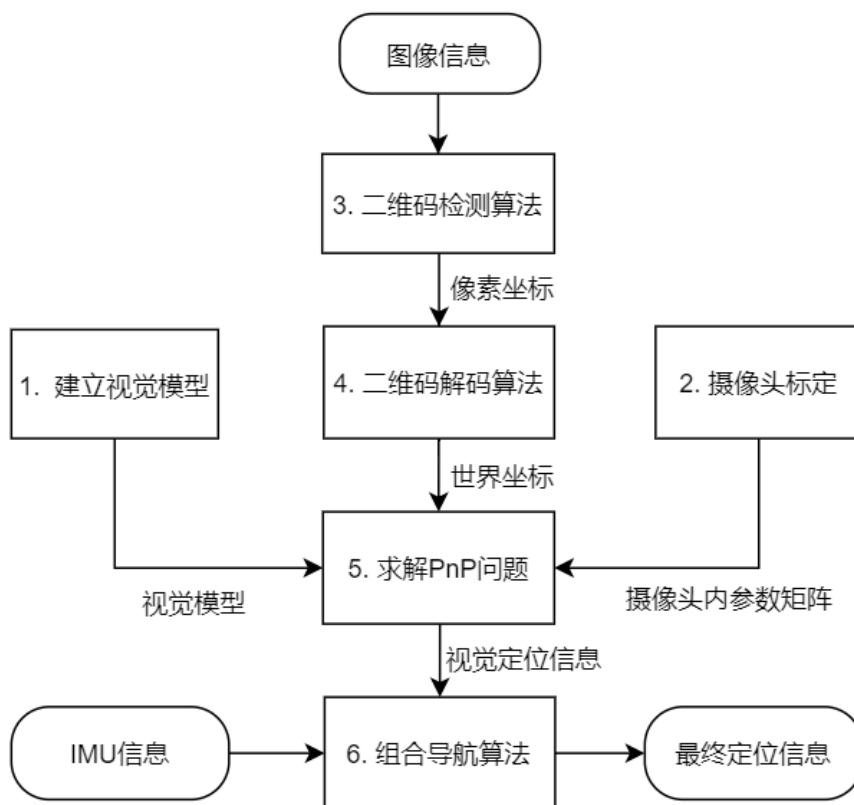


图 2.3 视觉定位程序流程图

总的视觉定位算法流程梳理如下:

---

#### 算法：视觉定位

---

1. 建立视觉模型
2. 摄像头标定, 载入相关数据

若仿真没有停止, 则一直循环:

- (1) 从摄像头或仿真环境获得 *图片信息*
- (2) 通过二维码检测算法, 获得二维码的 *像素坐标*
- (3) 通过二维码解码算法, 综合地图信息, 获得二维码的 *世界坐标*
- (4) 根据视觉模型方程和求解 PnP 算法得到 *视觉定位信息*
- (5) 获得 *IMU 数据*

## (6) 将视觉定位信息和 IMU 数据通过组合导航得到最终的定位信息

下面详细介绍每一步的具体过程。

### 2.2 摄像头的标定

在仿真中和在实际应用中获取相机内参数矩阵的方法不完全相同。

在仿真中，内参数矩阵 $M_1$ 主要根据仿真相机参数通过式(2.7)，(2.8)获取：

$$\frac{f}{dx} = \frac{width}{2 \times \tan\left(\frac{HFOV}{2}\right)} \quad (2.7)$$

$$u_0 = \frac{width}{2} \quad (2.8)$$

其中 $\frac{f}{dx}$ 为 $M_1$ 中的待求项，具体含义见上文。 $u_0$ 是图像中心点在像素坐标系中的横坐标， $width$ 为图像水平分辨率， $HFOV$ 为镜头的水平焦距。 $\frac{f}{dy}$ 和 $v_0$ 的获取与此类似，将水平参数换为垂直参数即可<sup>[35]</sup>。

在实际应用中，虽然在上一节推导了相机内参数矩阵，但由于制作精度、装配误差和相机畸变等因素的影响，通常采用摄像头标定的方法以得到更精确的相机内参数矩阵 $M_1$ 。标定的过程采用 MATLAB 的镜头标定工具箱。首先打印 OpenCV 提供的标定板图片，测量标定板每个方格的大小。接着，从不同角度拍摄大约 15 到 20 张标定板图像。最后，将这些图片导入 MATLAB，即可通过 `cameraCalibrator` 命令得到畸变参数 $(k_1, k_2, p_1, p_2, k_3)$ 与相机内参数矩阵 $M_1$ ，并通过 ROS 的相机畸变校正节点 `image_proc` 进行校正。对树莓派 P5V04A 相机的标定结果见图 2.4。

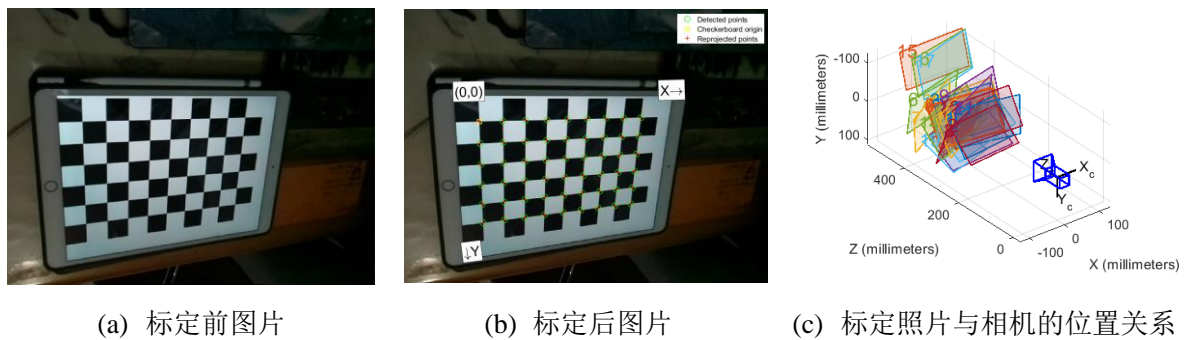


图 2.4 对树莓派相机的标定结果

平均标定误差为 0.09 像素，得到的相机内参数矩阵 $M_1$ 为

$$M_1 = \begin{pmatrix} 577.1543 & -0.4591 & 322.1148 & 0 \\ 0 & 576.6490 & 242.6257 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.9)$$

畸变参数为 $(k_1, k_2, p_1, p_2, k_3) = (0.0666 \quad -0.2983 \quad 0 \quad 0.0027 \quad 0)$ 。从标定前后的图片变化可以看出, 图片变化很小。这是因为树莓派 P5V04A 相机焦距为 65mm, 属于标准焦距镜头, 镜头畸变较小。

## 2.3 计算机视觉算法

本项目选用 AprilTags 的 36h11 族作为定位使用的二维码。AprilTags 是一种结构简单的二维码, 因其识别简单, 容错性好, 已经在机器人研究中得到广泛的应用。一张典型的 36h11 族图片如图 2.5。

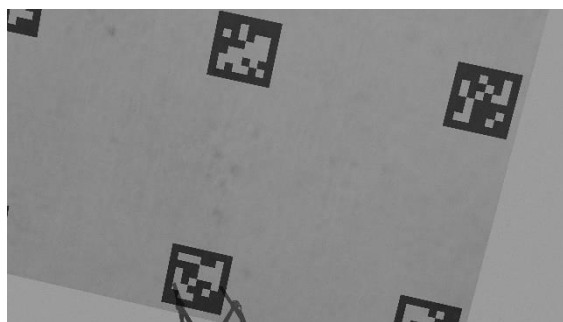


图 2.5 一张属于 AprilTags 36h11 族的二维码图片

每个二维码的边长为 8 像素。最外边的一圈用来检验有效性, 因此全部是黑色。中间共  $6 \times 6 = 36$  个像素, 占用 36bit, 用来编码。通信中常用 Hamming 距离来衡量两个编码的一致性。Hamming 距离指的是对两个编码进行异或运算后, 结果为 1 的个数。这一系列二维码可以确保即使旋转  $90^\circ$ ,  $180^\circ$  或  $270^\circ$ , 每一个二维码依然与其他二维码具有最小 Hamming 距离 11 的差距, 因而称作 36h11 族<sup>[6]</sup>。

### 2.3.1 AprilTags 的检测算法

检测的目的是通过计算机视觉算法, 从原始图片中获得二维码的像素坐标, 并分离出二维码的正视图以进行后续的解码操作。检测过程中图像的变化如图 2.6。



(a) 预处理



(b) 阈值分割与形态学处理



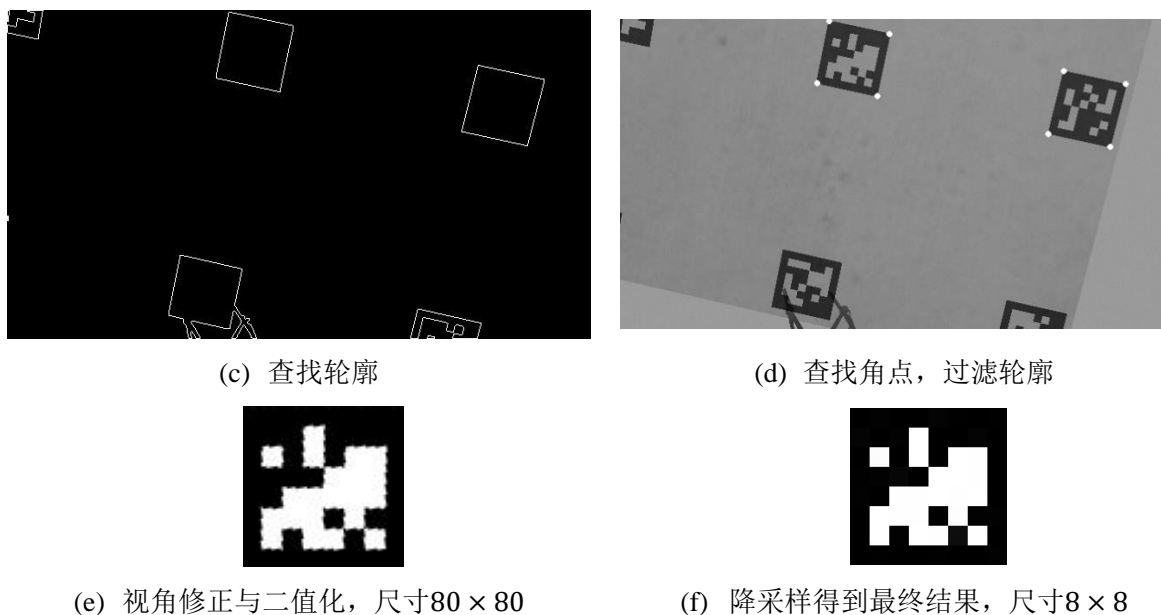


图 2.6 检测过程中图像的变化

参考 AprilTags 的原始文献<sup>[6][7][8]</sup>, 结合 OpenCV 库进行算法简化<sup>[35]</sup>, 介绍二维码检测的基本步骤如下:

#### a) 图像预处理

图像的预处理操作包括降噪、转为灰度图和图像缩放。将彩色图片转为灰度图并进行缩放处理, 可以减少数据量, 加快检测速度。常用的图像缩放算法有两种: 双线性差值法与最近邻插值法。在实际测试中, 与双线性差值法相比, 最近邻插值法可以更完善地保留二维码细节, 因此选取最近邻插值法。另外, 图像缩放的程度可以权衡计算速度与正确率, 图像缩放程度越大, 计算速度越快, 正确率越低, 反之亦然。目前选用的图像尺寸是 $640 \times 360$ 。

#### b) 阈值分割与形态学处理

对图像进行阈值分割 (Threshold) 可以使图片呈现黑白效果, 过滤掉对比不明显的信息, 从而达到增强目标轮廓的效果。由于本项目中设计的二维码环境较为单一, 因此采用大津阈值 (Ostu) 方法进行阈值分割。而形态学处理 (Morphology) 中的开运算 (Open Operation) 对图像进行先腐蚀后膨胀处理, 可以消除小物体, 平滑大物体边界。项目中选用的处理核为 $3 \times 3$ 单位矩阵。

#### c) 查找轮廓

对二值化图像应用查找轮廓算法, 可以获得图像中每个目标的轮廓点集。原文献通过查找并集的方式获取轮廓, 本项目采用 OpenCV 函数 `cv2.findContours()`。

#### d) 查找角点, 过滤轮廓

查找角点是检测算法中的关键步骤。经查阅资料, 从查找角点主要有四种算法: 1. 使用 OpenCV 函数 `cv2.approxPolyDP()`, 应用 Ramer–Douglas–Peucker 算法以迭代的方式进行多边形拟合。2. 根据 apriltags 文献, 以迭代的方式对轮廓上的点进行直线拟合, 拟合方差最大的即为角点。3. 通过霍夫变换 (Hough) 寻找。4. 通过 Harris 角点检测算法寻找角点。

本项目采用方案 1, 实现简单且对于仿真环境中的图片有较好的效果。主要思路为: 以  $(0.04 \times \text{轮廓周长})$  的精度拟合轮廓, 如果拟合之后是外接矩形的四边均大于 20 像素的四边形, 且外接矩形的宽高比介于  $[0.8, 1.2]$  之间, 则保留结果, 否则排除。

#### e) 视角修正算法

视角修正的目的是通过一个  $3 \times 3$  的单应性矩阵 (Homography matrix) 将二维码转化为正视图。单应性矩阵的原理如式 (4.10):

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.10)$$

其中  $u, v$  为变换前的像素坐标, 通过单应性矩阵映射后即得到变换后的像素坐标  $u', v'$ 。由于二维码四个角点变换前后的坐标均已知, 因此可以直接应用直接线性变换 (Direct Linear Transform, DLT) 方法计算单应性矩阵  $H$  以进行视觉修正。在程序中使用 `cv2.findHomography()` 函数计算单应性矩阵, 通过 `cv2.warpPerspective()` 函数完成视角变换与像素点插值。最后需要进行二值化处理, 增加图像对比度。

#### f) 降采样

通过降采样将  $80 \times 80$  的二值图像缩放为  $8 \times 8$  的二值图像, 以进行后续解码处理。

经过这一部分的算法, 得到了二维码四个角点在图像中的像素坐标, 并获得了可以解码的二维码图像。

### 2.3.2 AprilTags 的解码算法

解码的主要步骤如下:

---

**算法: 二维码解码**

---

经二维码检测算法获得 $8 \times 8$ 的二值图像后：

1. 将 AprilTags36h11 族图像的二进制编码保存在 python 列表中，制作成字典
2. 分别旋转图像 $0^\circ$ ， $90^\circ$ ， $180^\circ$ 和 $270^\circ$ ，进行编码
3. 循环遍历字典中的二进制编码：
  - (a) 将第 2 步得到的编码与字典中的编码计算 Hamming 距离
  - (b) 记录最小的 Hamming 距离，以及此时图像的旋转角度和字典中编码的 id 号
4. Hamming 距离是否小于 8bit?
  - Yes:
    - (a) 根据旋转角度调整二维码的像素坐标顺序，保存
    - (b) 该对应的 id 号与地图信息相结合，即可得到该二维码四个角点的世界坐标
  - No: 排除

其中在步骤 2 编码时，根据图 2.7 箭头的顺序进行二进制编码。

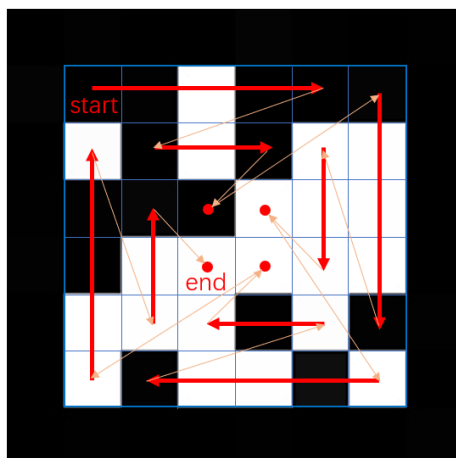


图 2.7 AprilTags 的编码顺序

如图，最外面一圈主要用于完整性检查。编码从左上角 start 开始，黑色部分为 0，白色部分为 1。该图片的二进制编码结果为：0010,0010,0011,1011,1110,1101,0111,1001,1101。

另外，步骤 4 中的过滤阈值根据文献确定。根据 AprilTags 文献的测试资料，即使存在 3bit 数据出错，误判率仅为%0.008714<sup>[6]</sup>。因此，我将 Hamming 距离的门限值设置为  $11-3=8$  bit。

### 2.3.3 PnP求解算法

给定二维码四个角点的像素坐标和世界坐标, 求解相机外参数矩阵 $M_2$ , 属于求解 P4P 问题。在 OpenCV 中, 已提供了 solvePnP()函数解决 PnP 问题。经查询, 该函数自带了一种于 2014 年提出的 IPPE 方法: SOLVEPNP\_IPPE\_SQUARE, 是当前速度最快、精度最高, 且专门针对共面的正方形标记进行优化的 PnP 问题解法<sup>[10]</sup>。求解 PnP 问题不是本项目的重点, 因而我将直接使用 OpenCV 自带的函数完成这一环节。

至此, 得到相机坐标系 $O_c$ 相对于世界坐标系 $O_w$ 的位姿变换关系。

最后, 根据相机的安装位置, 经过坐标变换得到机体坐标系 $O_b$ 相对于世界坐标系 $O_w$ 的位姿变换关系, 视觉定位解算完毕。

## 2.4 视觉定位与 IMU 组合导航

尽管上述内容已经可以让无人机获得视觉定位信息, 但这种定位方式要求摄像头内必须拍到至少一个完整的二维码, 对环境中的二维码密度要求较高, 使用条件比较苛刻。为了减少环境中二维码的密度, 让无人机在从一个二维码飞向另一个二维码的过程中依然可以获得较精确的定位信息, 下面引入组合导航算法。

### 2.4.1 组合导航的基本原理

本课题采用视觉定位与惯性测量单元进行组合导航。惯性测量单元 IMU (Inertial Measurement Unit), 是测量物体三轴姿态角的角速度及三轴加速度的装置。一般的 6 轴 IMU 包括三轴陀螺仪及三轴加速度计, 某些 9 轴 IMU 还包括三轴磁力计。其中陀螺仪测量  $x, y, z$  三方向的角速度, 加速度计测量  $x, y, z$  三方向的加速度, 分别经过积分可以得到角度信息和速度、位置信息。IMU 获得的导航信息具有数据连续, 信息完备的特点, 但其定位精度随时间会有较大程度的漂移。

为从 IMU 原始数据解算较为精确的位置和姿态信息, 常加入其它高精度导航元件进行数据校准, 并通过卡尔曼滤波技术进行信息估计。卡尔曼滤波(Kalman Filter)算法是由 R.E. Kalman 于 1960 年第一次提出的针对离散数据的最优估计技术, 其本质是基于统计公式得出的一种递推线性最小方差估计, 可以通过迭代的方式对真值进行准确预测。卡尔曼最初提出的滤波算法只适用于线性系统。Bucy, Sunahara 等人提出了扩展卡尔曼滤波算法 EKF, 将卡尔曼滤波理论进一步应用到非线性领域。EKF 的基本思想是将非线性函数的 Taylor 展开式进行一阶线性化, 忽略其余高阶项, 然后进行卡尔曼滤波, 因此 EKF 是一种次优滤波。

卡尔曼滤波在组合导航中的应用分为两种：直接法与间接法。在直接法滤波中,卡尔曼滤波器接收各导航子系统的导航参数,经过滤波计算,得到导航参数的最优估计。在间接法滤波中,卡尔曼滤波器接收两个导航子系统对同一导航参数输出值的差值,经过滤波计算,估计出各误差量的最优估计<sup>[27]</sup>。直接法系统方程简单,但其非线性特性使得结果误差较大;间接法是线性系统,可以直接应用卡尔曼滤波,但误差传递的状态方程推导复杂。本论文采用 EKF 滤波算法进行直接法组合导航。

#### 2.4.2 apriltag\_ros工具包

本项目上文中的检测解码算法由 python3.7 编写,难以与 ROS kinetic 版本兼容,且根据实验 4.2.1,本项目算法在真实环境中的效果略逊于 AprilTag3 源码算法。因此,经与导师协商,在组合导航系统中我将使用源码算法完成二维码的检测与解码任务。

AprilTags 源码算法团队提供了该算法的 ROS 封装,名为 apriltag\_ros。与纯 AprilTag3 算法相比, apriltag\_ros 针对多个二维码的情况进行了定位优化<sup>[33]</sup>。整体流程如图 2.8:

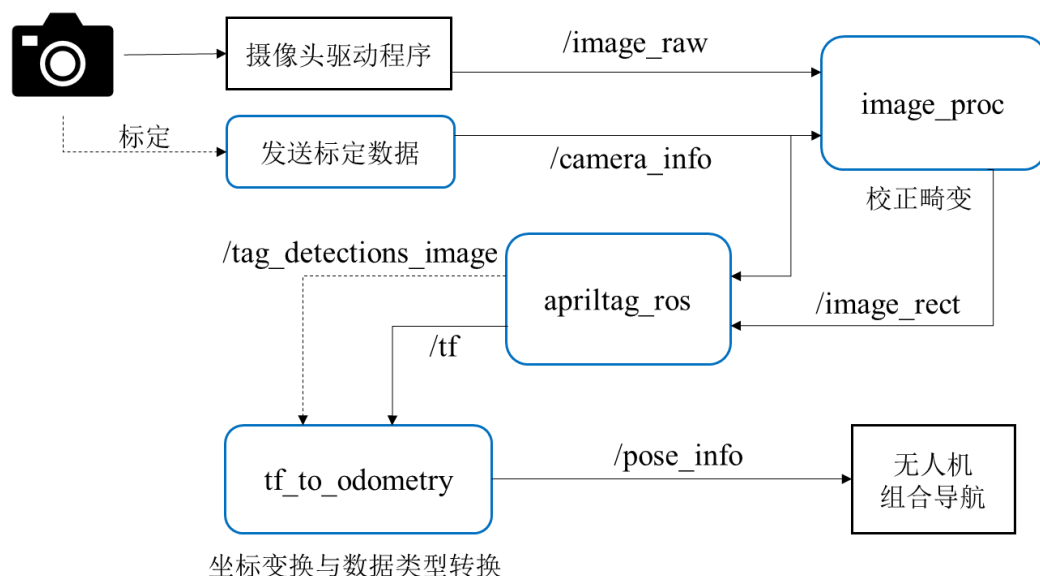


图 2.8 使用apriltag\_ros的流程

其中,蓝色框表示需要自行配置或编写的 ROS 节点(node),以“/”开头的字符表示 ROS 话题(topic)。摄像头获取的图像信息经由“image\_proc”节点进行畸变校正后得到无畸变的图像/image\_rect,与相机信息一起送入“apriltag\_ros”节点进行二维码相对位姿的检测,最后由“tf\_to\_odometry”节点进行坐标变换,得到机体坐标系在世界坐标系中的位姿,送入无人机组合导航节点。对各个话题的介绍如表 2.1,配置文件请见附录 A。

表 2.1 apriltag\_ros 使用过程中涉及的 ROS topic

名称	/image_raw	/camera_info	/image_rect	/tf	/pose_info	/tag...image
内容	原始图像	相机参数	校正图像	位姿信息	位姿信息	检测后图像
数据类型	Image	CameraInfo	Image	tfMessage	Pose	Image

### 2.4.3 robot\_localization工具包

robot\_localization 是由美国机器人智能公司 Charles River Analytics 的研究员 Thomas Moore 和 Daniel Stouch 开发的非线性组合导航 ROS 工具包。与其他组合导航工具相比，该工具包不限制输入传感器的数量，提供 EKF 和 UKF 两种滤波算法，且可以在使用时选择全部或部分信息进行融合，因而在机器人领域得到广泛的应用。

使用时应使各传感器信息的单位与坐标系定义满足 ROS 的 REP-103 和 REP-105 标准。使用过程中存在三个坐标系之间的坐标系转换关系如图 2.9<sup>[37]</sup>。

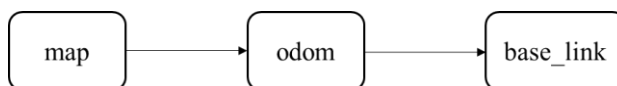


图 2.9 坐标系转换关系

其中，map 指由 GPS、二维码定位等传感器生成的非连续数据所依托的不会飘移的世界坐标系；odom 指由 IMU、视觉里程计等传感器生成的连续数据所依托的不断飘移的坐标系；base\_link 坐标系指机体坐标系。由于该程序包基于 ROS 的 tf 包实现坐标变换，而 tf 要求每一个坐标系只能存在一个父坐标系，因此经典用法如图 2.10：

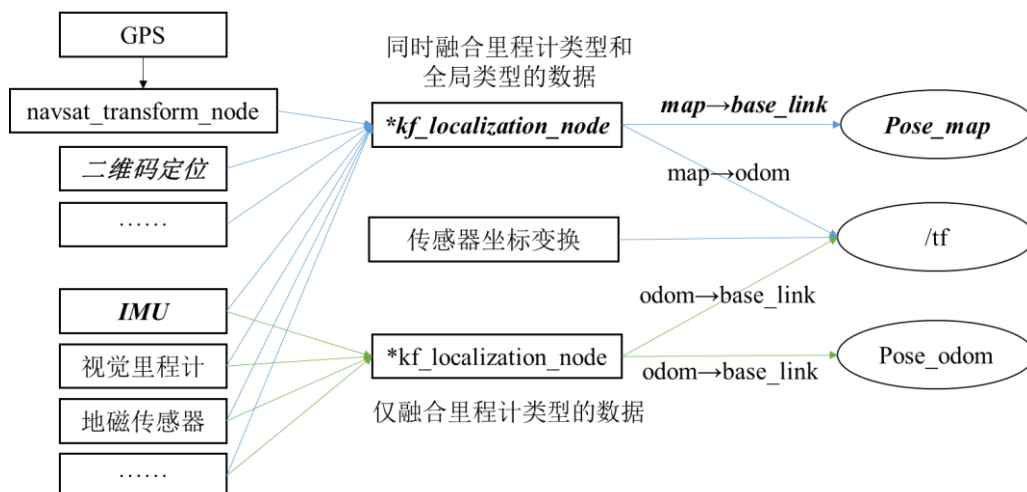


图 2.10 robot\_localization使用框图

图中斜体加粗部分为本项目用到的融合路线。本项目只有一个 odom 系传感器 IMU 和一个 map 系传感器二维码定位，因此将应用一个 KF 节点进行信息融合，生成机体系

base\_link 在 map 系中的位姿。

对传感器融合信息的选择如下表。

表 2.2 对传感器信息的选择

Sensor	配置向量, $true = 1, false = 0$														
	$x$	$y$	$z$	$\phi$	$\theta$	$\psi$	$x'$	$y'$	$z'$	$\phi'$	$\theta'$	$\psi'$	$x''$	$y''$	$z''$
IMU	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
Apriltag	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

使用卡尔曼滤波的关键是误差协方差矩阵初值 $P_0$ ，系统噪声协方差矩阵 $Q$ 和观测噪声协方差矩阵 $R$ 的正确配置。 $Q$  阵与  $R$  阵的大小可以调节系统对 IMU 信息和量测信息的置信度，一般减小  $Q$  阵增大对原状态的置信度，减小  $R$  阵增大对量测数据的置信度。本工具包的  $R$  阵可以由传感器精度推算量级， $R$  阵为对角阵，其对角线上的元素即为该量测量误差的平方。仿真设定的具体参数请见附录 B。至此完成对组合导航部分的介绍。

## 2.5 本章小结

本章介绍了无人机编队系统所采用的定位方法。本章开始介绍了视觉模型的建立过程，并在此基础上说明了视觉定位的总流程。接着依次介绍了摄像头标定、二维码检测、解码和求解 PnP 问题中用到的计算机视觉步骤，完成了视觉定位算法。为减少无人机对环境中二维码的依赖程度，引入惯性测量元件与视觉定位信息进行融合，以在稀疏二维码环境中为无人机提供连续、精确的定位信息。

### 3 强化学习编队原理及实现

为解决单个无人机执行任务效率低,容错性差的问题,多无人机协同编队控制已经成为无人机研究的新热点。传统的编队控制算法往往将无人机模型进行一定程度的简化以便于数学推导,但在实际应用时常常受到真实环境的约束。而强化学习算法具备自学习的特性,也许可以较好地应用于真实环境。因此,本章节开始介绍了编队问题的定义以及使用强化学习算法解决该问题的思路,接着针对目标追踪任务和避障任务设计状态空间、动作空间和奖励函数,最后结合 Dyna-Q 强化学习算法给出了编队问题的训练算法与应用算法。

#### 3.1 编队问题分析

在无人机的实际应用中,由于单个无人机续航里程的限制,测量范围往往十分有限,通常需要利用多架无人机组成一个特定队形来实现更大范围的覆盖和检测。另外在公路运输等应用场景中,也需要多无人机系统自主形成一个队形,并保持这个队形来完成特定任务。因此,编队控制成为了多智能体协调控制中的一个研究热点。

多智能体的编队控制问题分为两类,一类是运动编队控制问题,指一组多智能体通过局部的相互作用(如通信、合作、竞争等),使它们在运动过程中保持预先指定的几何图形,向指定的目标运动;另一类是静止编队控制问题,指多智能体经过运动静止在某一位置组成几何图形。多智能体系统的编队控制主要解决以下问题:1. 各智能体间如何相互作用,才能生成指定的队形。2. 在队形移动的过程中,智能体之间如何保持指定队形。3. 在运动的过程中,队形中的个体如何才能躲避障碍物。4. 当外界环境突然改变时,如何自适应地改变队形或者保持队形,以适应环境<sup>[40]</sup>。

由于传统强化学习算法的局限性,本论文中提到的编队控制属于静止编队控制,重点在于无人机的位置控制,暂不考虑对姿态和速度的控制。针对多智能体编队控制存在的问题,本论文着重解决问题一队形形成和问题三编队避障。

队形定义是解决编队控制问题的第一步。按照传感器设备测量和通讯能力的强弱,几何队形的定义可以被分为四类:基于绝对位置定义的队形,基于相对位置定义的队形,基于相对距离定义的队形以及基于相对角度定义的队形。本论文中的无人机采用第二章介绍的视觉惯导组合定位系统提供定位信息,精度较高,因而采用基于绝对位置的编队方法进行队形定义。这种编队控制方法对传感器检测能力的要求很高,但对无人机之间



的交互程度要求不高。

本项目基于 leader-follower 模型给出模型定义,并将无人机简化为二维平面的飞行。Leader-follower 指的是 follower 以 leader 为基准保持一定构型的模型定义方法,是目前多无人机编队控制中最常用的方法之一<sup>[29]</sup>。在本课题中,首先给定各 follower 相对于 leader 的目标位置信息,然后每个 follower 根据定位信息将该相对位置转化为地图中的绝对位置,再控制自身飞去。

具体来说,本课题的无人机编队问题拆解到每个无人机上有两个子任务:1. 目标追踪任务:发给无人机一个位置,无人机可以自行飞到该目标地点。2. 避碰任务:在目标追踪的过程中,无人机要注意彼此的避碰。当每个无人机实现这两个子任务后,将多架无人机组合起来完成带避障的编队任务。

而强化学习算法在实际问题上的应用,关键是通过合理设计状态空间、动作空间和奖励函数,将实际问题转化为马尔科夫决策过程(Markov Decision Process, MDP),再使用已有的强化学习算法解决问题。在本课题中,我将分别针对目标追踪任务和避障任务设计状态空间和奖励函数。最终无人机的动作由这两部分的值函数表(Q-table)对应的值加和得出。

### 3.2 马尔科夫决策过程设计

马尔科夫决策过程(MDP)是强化学习算法适用的基本框架,如图 3.1 所示。

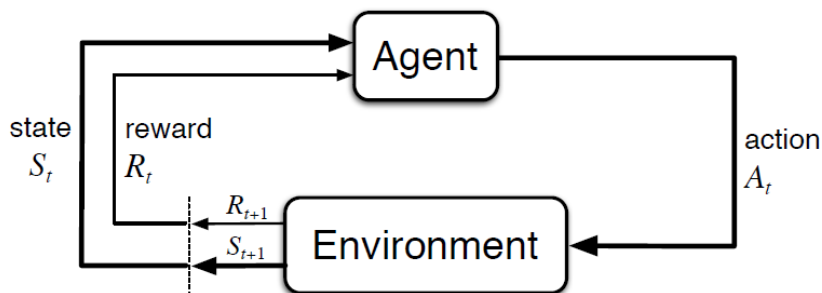


图 3.1 MDP过程示意图<sup>[38]</sup>

在 MDP 过程中,决策者被称为智能体(agent),与智能体交互的一切外部因素被统称为环境(environment)。在交互过程中,智能体每执行一个动作(action),环境将反馈给智能体这个动作对应的奖励(reward)和智能体所处的新状态(state)。这种交互过程将在强化学习中一直循环。另外,此过程还应该满足马尔科夫条件(Markov property),即现在的状态包含了预测未来的全部信息<sup>[38]</sup>。如要将强化学习算法运用在实际任务中,最关键的

问题是如何将目标任务转变为马尔科夫决策过程。

### 3.2.1 目标追踪问题的MDP设计

**状态空间:** 目标追踪任务中,可以被设定为状态的信息有相对位置、速度和加速度,其中相对位置信息对于该任务必不可少。由于本项目拟采用基于表格的强化学习算法,状态量不宜过多,因此不考虑引入相对速度和加速度作为状态。为确保目标追踪的精度,应满足离目标越近,状态划分越精细的要求。结合对文献[21][23]的参考,我定义目标追踪问题的状态空间如下:

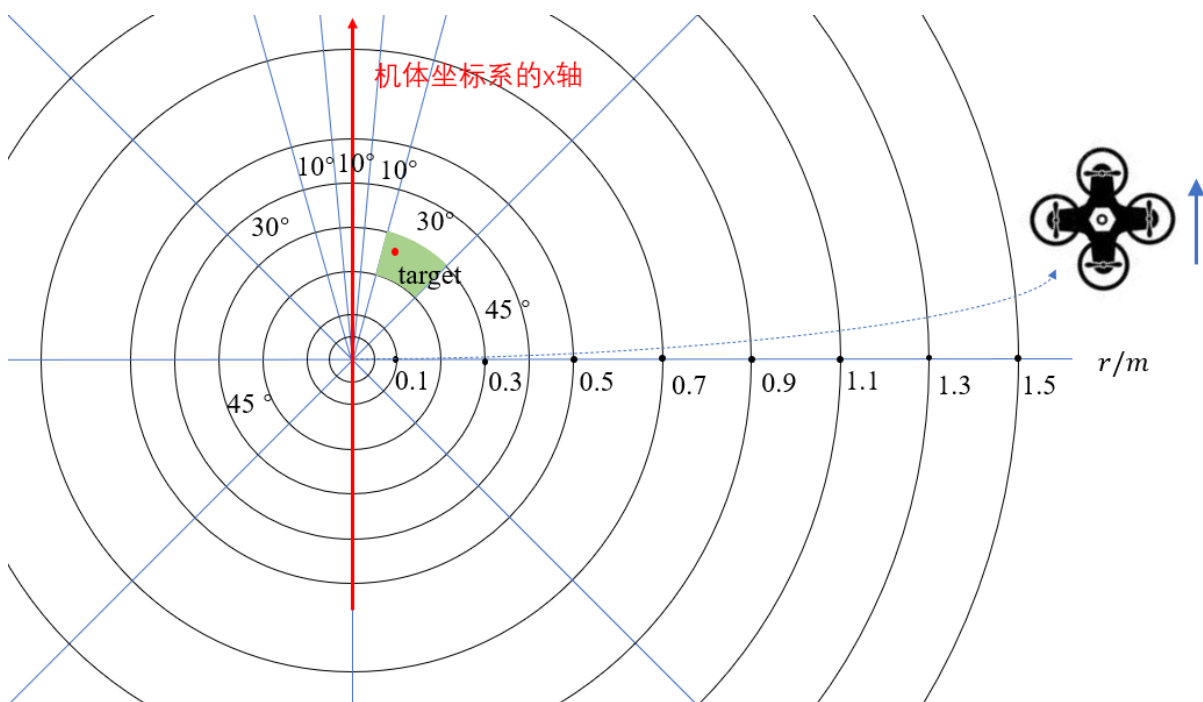


图 3.2 目标追踪问题的状态定义

如图 3.2,以无人机机体中心为原点,以机体的  $x$  轴为极轴,建立极坐标系,根据目标点与无人机的相对位置确定状态。状态划分的具体参数如表 3.1。

表 3.1 目标追踪任务的状态定义

圈数	0	1	2	3	4	5	6	7	8	9	10
距离/m	0.05	0.1	0.2	0.3	0.4	0.5	0.7	0.9	1.1	1.3	1.5
扇形数	0	1	2	3	4	5	6	7	8	9	10
角度/°	5	15	45	90	135	180	-135	-90	-45	-15	-5

从内到外共 11 圈,共 11 个扇形,从最内圈  $0^\circ$  所在的状态从 0 开始逆时针标号,总共 121 个状态。如果目标点与飞机的相对位置超过 1.5m,则都算在最外圈对应的扇形

状态内。

**动作空间：**由于状态空间仅包含无人机和目标点的相对位置，为使整个过程满足马尔科夫条件，则无人机的动作空间只能包含位置量的改变。文献中基于位置定义的动作空间有两种，一种是按照前后左右进行动作，不考虑机体偏航角度的改变，另一种是按照前后和机体的偏航进行动作。考虑到在实际应用中，无人机存在正面安装前置摄像头等传感器件的情况，第二种动作方式具有更长远的研究价值。因此，我定义无人机在机体坐标系下的动作如表 3.2:

表 3.2 目标跟踪问题的动作空间定义

动作标号	0		1		2		3	
前进距离/m	0.2		0.1		0.05		0	
动作标号	4	5	6	7	8	9		
偏转角度/°	-30	-10	-3	3	10	30		

**奖励函数：**奖励函数定义的好坏直接决定了系统的训练难度。经过在实验中的反复尝试，我确定了奖励函数的五条规则：

1. 若无人机与目标点的相对位置小于 0.05m，则获得奖励+50。
2. 若目标所在的状态离无人机接近一圈，则奖励+5；若远离一圈，则奖励-5。
3. 若目标所在的状态与极轴接近了一个扇形，则奖励+5；若远离一个扇形，则奖励-5。
4. 若无人机处于最外圈，且下一次动作后的位置更加远离目标，则奖励-10。
5. 若无其余奖励情况，则奖励-1。这条规则可以让无人机尽快完成目标。

最开始的奖励函数仅设置了第一条和第五条规则，但在训练阶段出现了“稀疏奖励”问题，即无人机因长时间得不到奖励，难以学习策略的情形。解决办法就是增加奖励函数的条件，尽量让无人机每做一个动作都可以获得相应的奖励。

### 3.2.2 避碰问题的MDP设计

**状态空间：**避碰问题的状态空间中也包含了障碍物和无人机机体的相对位置信息。与目标跟踪任务不同的是，避障不要求精度，作用范围也很小，因此状态空间的定义更为均衡，圈数更少。仿真中无人机模型的直径为 0.8m。我定义避碰任务的状态空间如下图。

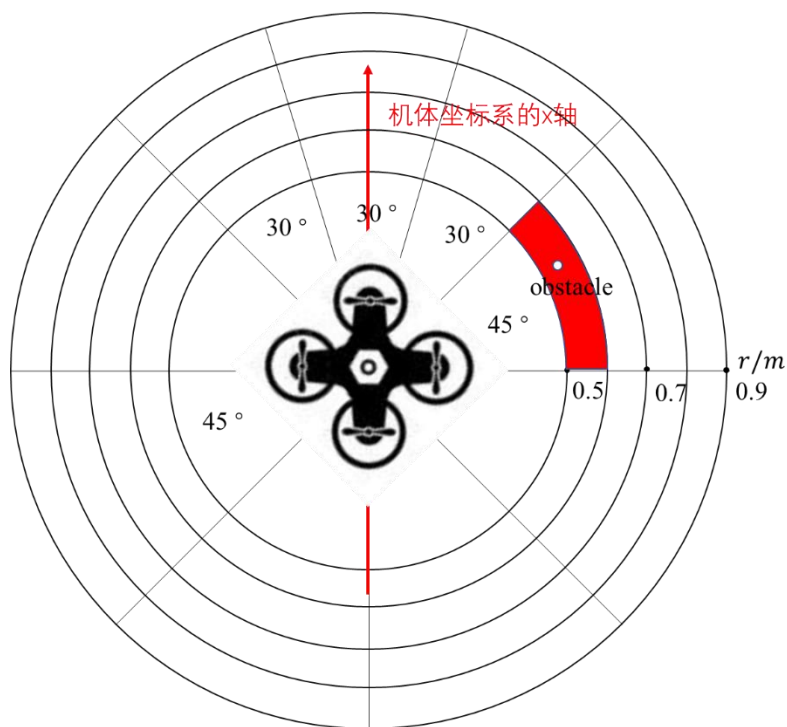


图 3.3 避碰任务状态空间的定义

如图 3.3，以机体的 x 轴为极轴，以机体中心为原点，建立极坐标系，根据障碍物与无人机的相对位置确定状态。状态划分的具体参数如表 3.3。

表 3.3 避碰任务的状态

圈数		0		1		2		3		4	
距离/m		0.5		0.6		0.7		0.8		0.9	
扇形数	0	1	2	3	4	5	6	7	8		
角度/°	15	45	90	135	180	-135	-90	-45	-15		

从内到外共 5 圈，一共 9 个扇形，从最内圈  $0^\circ$  所在的状态从 0 开始逆时针标号，总共 45 个状态。如果障碍物与飞机的相对位置超过 0.9m，则都算在最外圈对应的状态内。

**动作空间：**与目标追踪任务相同。

**奖励函数：**

1. 若障碍物处于最外圈及以外的部分，奖励 0。
2. 障碍物远离飞机，奖励+10；反之-5。
3. 若飞机远离避障任务的作用范围，则奖励+20。
4. 若障碍物位于正前方的三个扇形内，奖励-5。

5. 若障碍物与无人机中心的距离小于 0.5m, 则给予奖励-100。

至此, 本课题编队任务的马尔科夫决策过程均定义完毕。

### 3.3 强化学习编队算法

#### 3.3.1 训练阶段的算法

经过几十年的发展, 基于表格的强化学习算法(Tabular Methods)中涌现出很多种经典的学习算法 (learning algorithm), 包括 Sarsa, Q-Learning, Expected Sarsa 等。这些算法在与环境的交互中进行学习, 不需要环境模型, 但对样本的利用率较低。为提高样本利用率, 我将采用 Dyna 架构进行值函数表(Q-table)的学习。Dyna 结构如图 3.4。

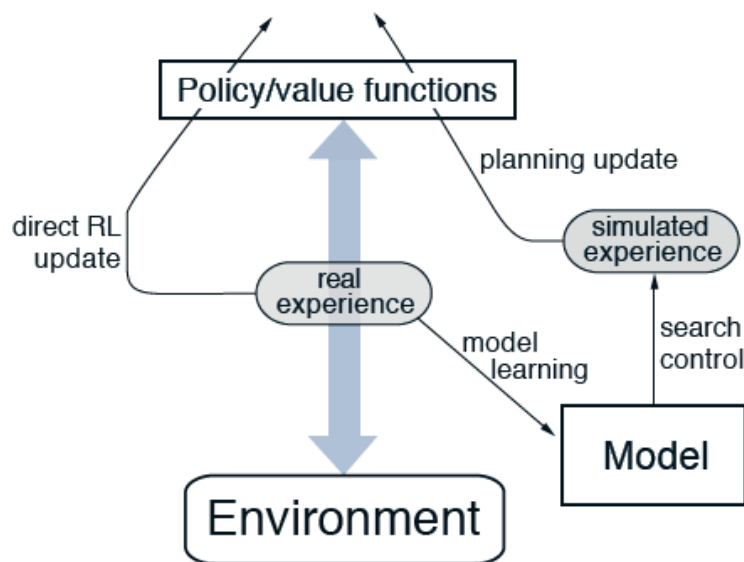


图 3.4 通用Dyna结构<sup>[38]</sup>

Dyna 结构与 Q-Learning 等经典强化学习算法的区别在于, 除了存在对值函数(action value)的学习, 还存在对模型(model)的学习。智能体(agent)在每次做决策前, 首先会基于模型, 迭代值函数表(Q table)。在基于模型的迭代(planning)和基于实际交互的迭代(learning)的双重作用下, 系统对交互样本的利用率得到了大大提高。将 Q-Learning 算法和 Dyna 结构结合即可得到 Dyna-Q 算法。

在实际课题中, 需要编写程序完成智能体(agent)与环境(environment)的交互。训练智能体采用的算法与具体任务无关, 而与 gazebo 环境交互的内容则放在环境部分的算法中。

训练目标追踪任务时对智能体采用的算法如下:

---

**算法: Tabular Dyna-Q<sup>[38]</sup>**

---

1. 针对所有的状态  $s \in \mathcal{S}$  和动作  $a \in \mathcal{A}(s)$ , 初始化  $Q(s, a)$  和  $Model(s, a)$
2. 若不到结束状态, 则一直循环:
  - (1)  $S \leftarrow$  现在的状态 (非结束状态)
  - (2)  $A \leftarrow$  根据  $\epsilon - greedy$  算法选出的动作
  - (3) 执行动作  $A$ : 观察奖励  $R$  和新状态  $S'$
  - (4)  $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
  - (5)  $Model(S, A) \leftarrow R, S'$ , 假设是确定性环境(deterministic environment)
  - (6) 循环  $n$  步:
    - a)  $S \leftarrow$  随机选择过去经历过的状态
    - b)  $A \leftarrow$  随机选择在状态  $S$  经历过的动作
    - c)  $R, S' \leftarrow Model(S, A)$
    - d)  $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

---

训练避障任务时, 除了第(2)步依据的  $Q$  值改为根据  $Q = Q_{target}(S, A) + 1.5 \times Q_{obstacle}(S, A)$  得出, 其余步骤保持不变。上式中的1.5为比例系数, 根据试验调参得出。在训练 agent 算法的第(3)步中, agent 需要将动作发送给 environment, 并等待反馈的新状态与奖励。在这一过程中, environment 经历的流程如下:

---

**算法: environment 的交互过程**

---

1. 接收 agent 发来的动作序号  $A$
  2. 从动作空间索引对应的动作组合
  3. 将上述动作坐标变换为世界坐标系下的位姿命令
  4. 执行命令, 直到无人机飞到指定位置
  5. 根据现在无人机的状态, 目标位置和障碍物位置评估状态  $S'$
  6. 获得对应的奖励  $R$
  7. 判断是否到达终点
  8. 将现在的状态  $S'$  和对应的奖励  $R$  反馈给 agent
- 

实际编程时我采用 RL-Glue 强化学习框架完成编程。RL-Glue 是加拿大阿尔伯塔大学开发的强化学习软件包, 通过模板类的形式定义了强化学习的关键环节, 极大提升不

同算法之间的通用性<sup>[39]</sup>。RL-Glue 内部主要完成了 agent 与 environment 之间交互的代码，而 agent 与 environment 的属性和动作需要继承 RL-Glue 的模板类并重构代码。

### 3.3.2 编队执行阶段的算法

训练完成后，每架无人机根据目标追踪阶段和避障阶段训练的值函数表进行动作。执行的算法是训练阶段算法的简化版，具体流程如下：

---

#### 算法：每架无人机根据强化学习算法执行编队任务

---

1. 初始化目标追踪的值函数表 $Q_{target}$ 和避障的值函数表 $Q_{obstacle}$
  2. 若程序不结束，则循环：
    - (1) 接收编队命令（相对位置），同时获取 leader 无人机的位置
    - (2) 根据坐标变换得到世界坐标系下的目标位置
    - (3) 若没有到达目标位置，则循环：
      - a) 接收其他无人机的位置信息
      - b) 选择距离最近的无人机，记录位置
      - c) 设定无人机实物半径为 0.5m，计算该无人机上距离本机最近的点的世界坐标，该点即为障碍物
      - d) 根据自身位置和该点的世界坐标获得避障任务的状态观测 $S_1$
      - e) 根据自身位置和目标点的世界坐标获得目标追踪任务的状态观测 $S_2$
      - f) 根据  $A = \underset{a}{\operatorname{argmax}} (Q_{target}(S_1) + 1.5 \times Q_{obstacle}(S_2))$  选择动作标号A
      - g) 根据动作标号索引相应动作并执行
- 

### 3.4 本章小结

本章介绍了无人机编队系统所采用的编队控制方法。本章开始介绍了编队问题的研究方向，进而确定本论文研究的是基于绝对位置的静态编队问题，并将该问题拆分为目标追踪问题和避碰问题。接着分别对这两个问题定义了马尔科夫决策过程。最后结合 Dyna-Q 强化学习算法给出训练阶段的算法和编队执行阶段的算法。编队控制是利用多无人机执行任务的关键步骤，而强化学习的引入为应对避障等非结构化问题提供了新的思路，提升了编队算法的实用性。

## 4 仿真实验及结果分析

本论文的第二章和第三章分别介绍了视觉定位任务和编队控制任务所采用的算法原理与实现步骤。在实际使用中，定位与编队控制组成了系统闭环，因此将在此章节集中展示对这两部分功能进行的实验与取得的结果。本章开始介绍了整个系统的基本构成与运作逻辑。接着在定位部分，测试了二维码检测算法的速度与精度，测试了组合导航定位的最终效果。最后在编队部分，展示了目标追踪能力和避碰能力的训练过程，测试了五架无人机的编队效果。

### 4.1 总系统介绍

本项目最终期望达到的硬件框图如图 4.1 所示：

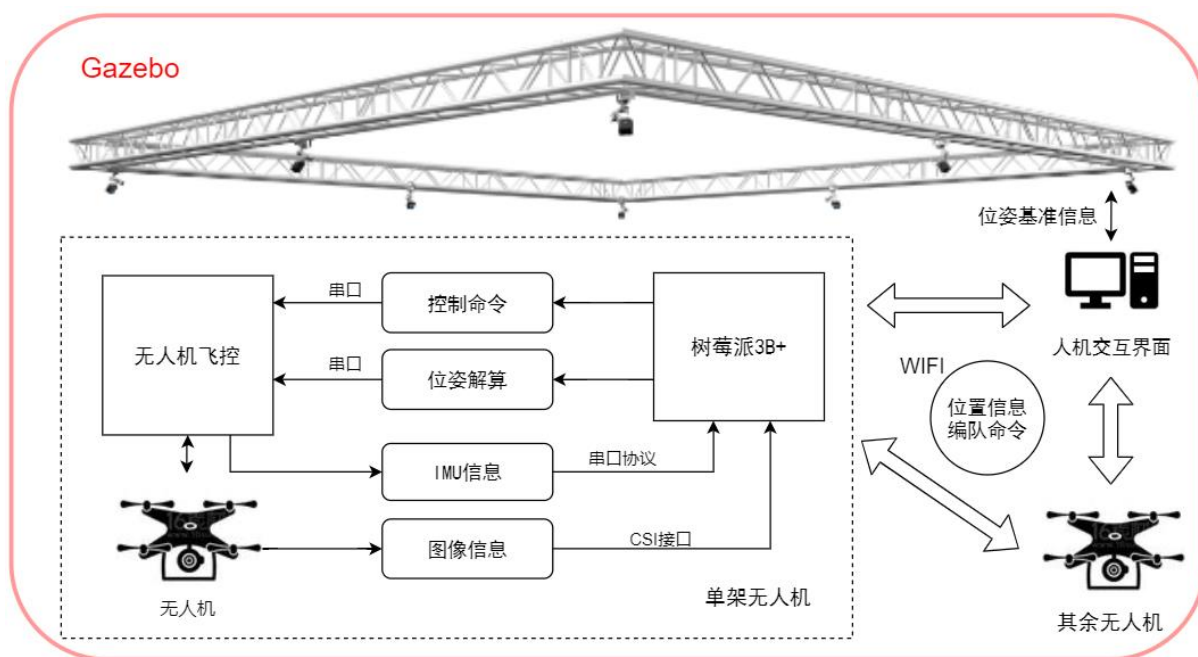


图 4.1 系统硬件框图

如图，每个无人机上都搭载了一个无人机飞控，一个树莓派 3B+设备和一个摄像头。整个定位-编队的流程逻辑如下：

#### 算法：无人机进行定位-编队的流程

1. 人机交互界面程序发送编队命令（在 leader 无人机坐标系下，每架无人机需要到达的位置）
2. 无人机接收编队命令，更新自身的目标位置



3. 在无人机内部循环, 直至到达目标位置:
- a) 接收图像和 IMU 数据
  - b) 通过定位算法解算自身的位置信息, 并发送给公共话题(topic)
  - c) 从公共话题订阅其他无人机的位置信息
  - d) 将目标位置、自身位置和其他无人机的位置输入强化学习算法, 得到动作指令  
(在机体坐标系下的偏转角度或前进距离)。
  - e) 将动作指令发送给底层飞控, 控制无人机飞行。

初期硬件选型如下:

**表 4.1 实验原拟采用的硬件设备**

处理器	飞控	无人机	摄像头	定位基准设备
树莓派 3B+	课题组自研飞控	QAV250 mini Quadcopter	P5V04A	Optitrack

因疫情影响, 本项目主要基于仿真开展实验。如图 4.1 所示, 无人机模型、飞行环境和传感器等改为 Gazebo 仿真, 而系统框图逻辑不变。仿真参数尽可能与表 4.1 中的硬件选型保持一致。用到的软件介绍如下:

**表 4.2 仿真实验拟采用的软件**

机器人操作系统	仿真模拟器	无人机及飞控模型	树莓派操作系统	主机系统
Robot Operating System (ROS)	Gazebo7	Hector_quadrotor	Raspbian 2020.02	Ubuntu 16.04

**ROS:** 用于编写机器人软件程序的一种具有高度灵活性的开源软件架构。它包含了大量工具软件、库代码和约定协议, 旨在简化跨机器人平台创建复杂、鲁棒的机器人行为这一过程的难度与复杂度<sup>[31]</sup>。

**Gazebo:** 一种常用的机器人仿真软件, 使机器人研究者可以使用逼真的环境快速测试算法、设计机器人与训练 AI 系统<sup>[34]</sup>。

**Hector\_quadrotor:** 由德国达姆施塔特工业大学的 hector 团队基于 ROS 与 Gazebo 开发的无人机建模、控制和仿真程序包<sup>[33]</sup>。

## 4.2 视觉定位部分的实验

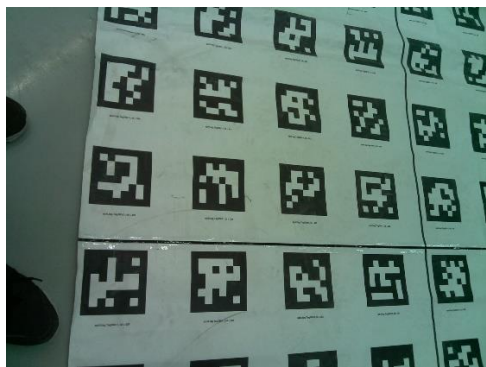
定位部分主要进行了两个实验: 本项目检测解码算法与 AprilTag3 源码算法的性能比较实验, 和视觉 IMU 组合导航定位实验。

### 4.2.1 检测解码算法测试

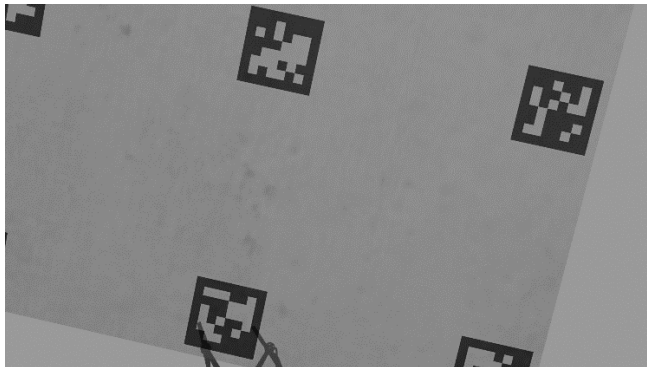
为比较本项目编写程序与源码程序的检测效果,选择实物与仿真环境下的两张图片进行测试,测试条件如表 4.3,测试图片如图 4.2。每组数据测试三次取平均值,测试结果如图 4.3~4.4。

表 4.3 检测算法测试条件

环境	硬件设备	图像分辨率	摄像头	完整二维码数量
实物	树莓派 3B+	2592 × 1944 (4:3)	OV5647	18
仿真		1920 × 1080 (3:2)		2



(a) 实物测试图片



(b) 仿真测试图片

图 4.2 检测算法测试图片

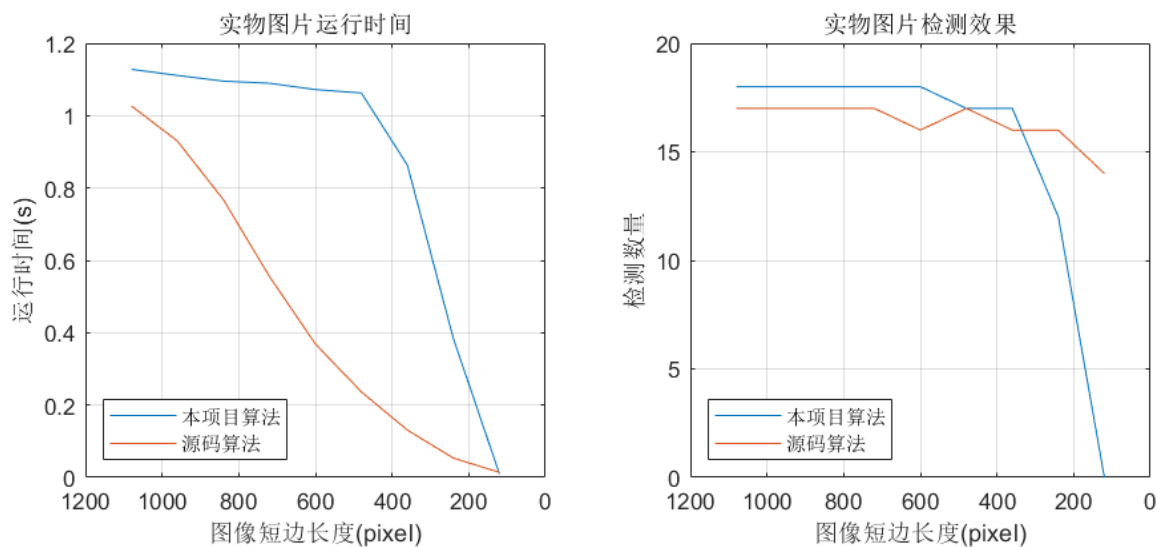


图 4.3 实物图片测试结果

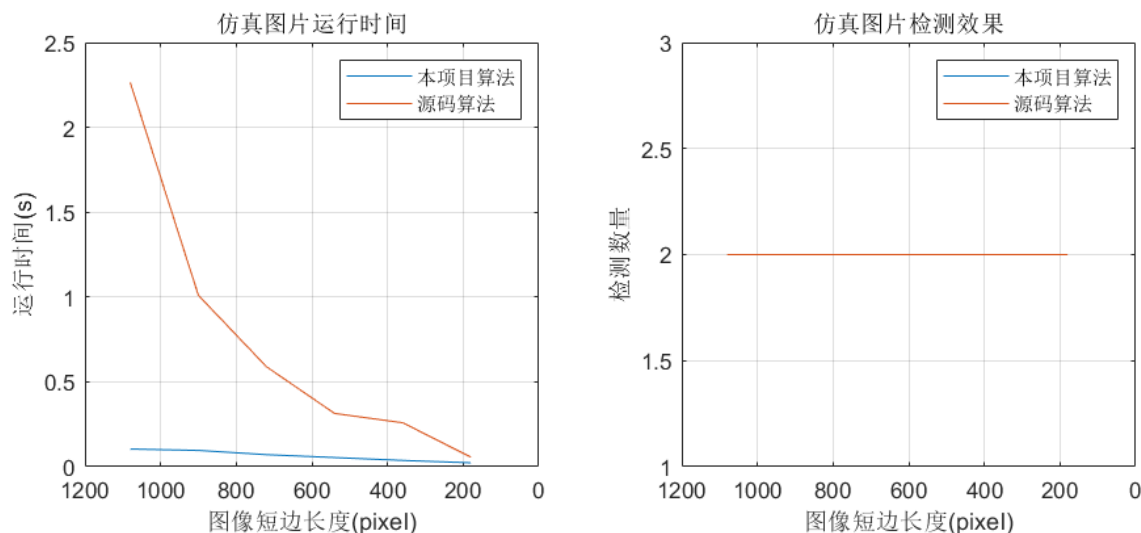


图 4.4 仿真图片测试结果

由测试结果可以得出以下结论：

- 1) 两种算法在不同场景下的检测精度接近，图像尺寸较小时（短边小于 360 像素）源码精度较高。
- 2) 源码算法的检测速度受检测场景影响较小，本项目算法受检测场景影响很大。仿真环境的图片较简单，真实场景的图片较复杂。本项目算法在仿真场景中检测速度远快于源码算法，而在真实场景中检测速度慢于源码算法。
- 3) 总的来说，源码算法更适合真实场景，而本项目算法更适合仿真环境，如果要将本项目算法应用于真实场景，仍需进一步调参。

#### 4.2.2 组合导航定位实验

##### 1. 二维码地图

仿真地图力求还原实验室场景。考虑到实验室的尺寸，设计地图为边长 4 米左右的正方形，无人机飞行高度在 1~2 米。我希望地图尺寸满足如下要求：飞行高度为 2 米时，画面中至少有 3~4 个完整的二维码；飞行高度为 1.5 米时，画面中至少有 1 个完整的二维码；飞行高度为 1 米时，画面中隔一段时间才出现一个完整的二维码。根据摄像头视角设计地图尺寸如图 4.5：

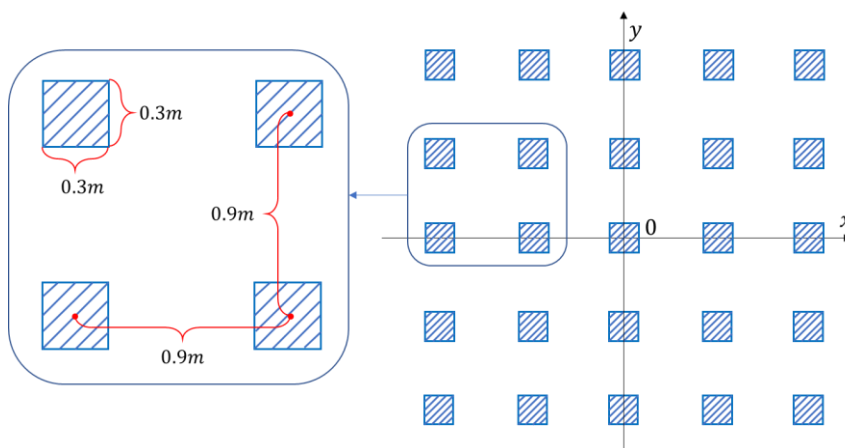


图 4.5 二维码地图的尺寸

## 2. 仿真实验

关于组合导航原理、`apriltag_ros` 工具包和 `robot_localization` 工具包的使用请见 2.4 节。实验室无人机上使用的 IMU 为 `mpu6050`，但查询 `datasheet` 发现没有详细的 IMU 参数。故参考高精度的惯性元件 `ADIS16488` 的参数和 `P5V04A` 摄像头的参数设置仿真参数如表 4.4。需要注意设置 IMU 的参数单位应与 REP-103 标准一致，角度为弧度制<sup>[33]</sup>。

表 4.4 组合导航实验的仿真参数

陀螺仪飘移	陀螺仪噪声的标准差		加速度计的飘移		IMU 更新速率	加速度计噪声的标准差
<i>radius/sec</i>	<i>radius/sec</i>		<i>meter/sec<sup>2</sup></i>		<i>Hz</i>	<i>meter/sec<sup>2</sup></i>
0.0002	0.0047		0.0018		100	0.0500
焦距(mm)	视频参数	视频格式	飞行高度(m)	飞行半径(m)	飞行速度(秒/圈)	
65	720p60	L8	1.0	1.3	20	

测试时长为 60 秒，实验结果如图 4.6~4.8。

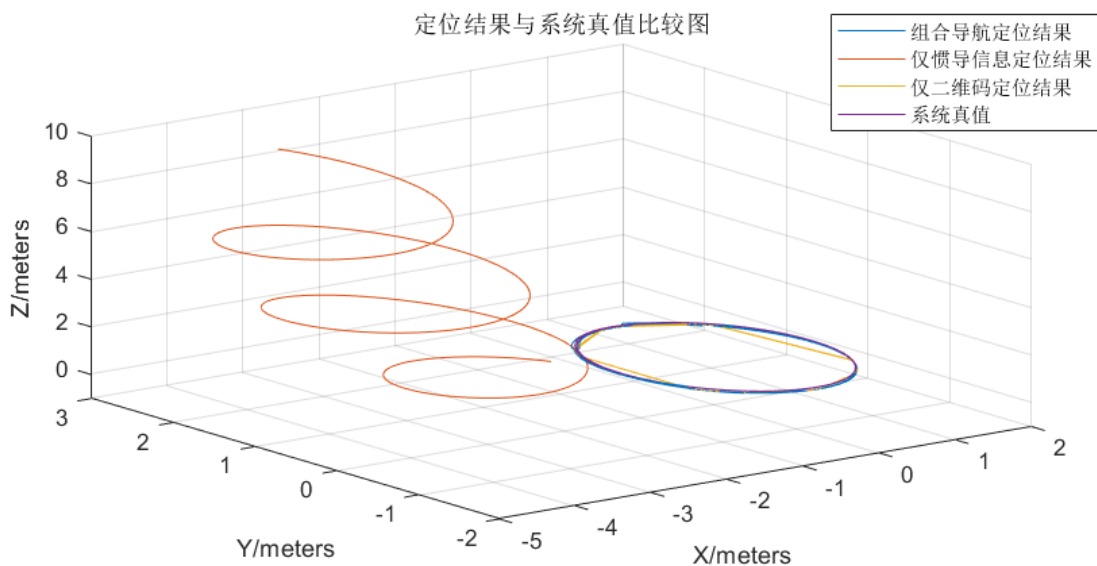


图 4.6 各种定位结果的三维比较图

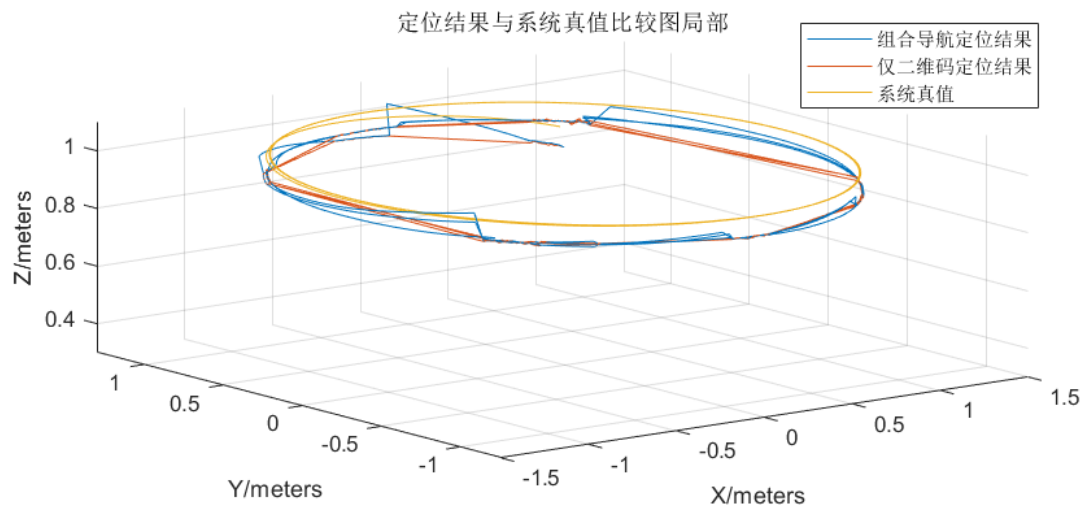
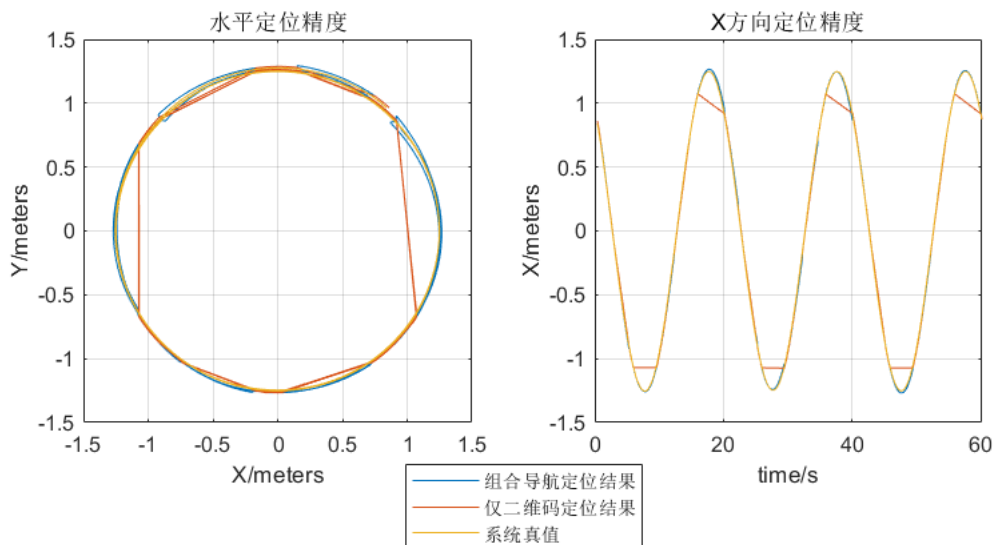


图 4.7 定位结果与三维比较图局部



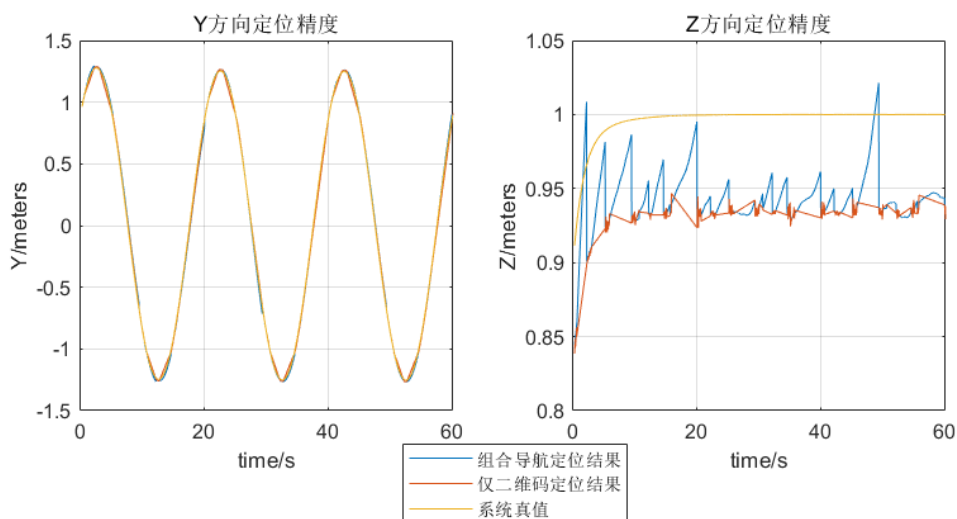


图 4.8 定位结果单独比较图

根据实验结果，可以得出以下结论：

1. 从图 4.6 和图 4.7 可以看出，在没有视觉信息进行校正时，仅凭惯导信息解算得到的导航信息以惊人的速度发散。而加入二维码信息进行组合导航以后，定位精度显著提高，飘移问题也得到解决。
2. 从图 4.8 可以看出，没有融入惯导信息时，若视野中没有完整的二维码，则无法进行视觉定位，因此难以得到连续的定位信息。而加入惯导信息进行组合导航后，在无人机运行到没有二维码的区域时，仍能在短时间内以较高精度连续提供定位信息。
3. 从精度上来说，仅视觉定位的水平定位精度约为 2~6 厘米，组合导航的水平定位精度约为 1.5~4 厘米，精度与运动速度相关：飞机运动速度越快，误差越大。推测导致误差的原因是对姿态的估计不够精准。
4. 高度定位精度在 7 厘米左右。值得注意的是，z 轴方向上的高度信息出现频繁的跳变。推测原因：第一，对重力信息去除不完全，第二，二维码定位算法只依靠 PnP 解法确定高度，对二维码形态的变化十分敏感。

总的来说，IMU 和二维码定位组合导航可以在较为稀疏的二维码地图中提供厘米级水平定位信息，且使用方便，精度较高。但如果要用于实际导航，需要引入高度传感器以矫正高度信息。

### 4.3 强化学习编队部分的实验

#### 4.3.1 目标追踪与避碰能力的训练

### 1. 目标追踪能力的训练

训练无人机的目标追踪能力分为两个阶段。第一阶段采取固定目标点的训练模式，目的在于尽快让无人机获得目标追踪的初步能力。若无人机与目标点的距离小于 5cm，则判断为追踪成功。实验参数设定如表 4.5，训练过程中每个周期(episode)获得的总奖励(reward)与训练次数(episode)的关系如图 4.9。

表 4.5 目标追踪第一阶段训练参数

中文	贪心率	学习率	折损因子	目标位置	训练趟数	每趟的训练次数
英文	Epsilon	step size	discount factor	(x, y)	Number of run	Number of episodes
值	0.1	0.5	0.9	(1.2m, -1.2m)	5	100

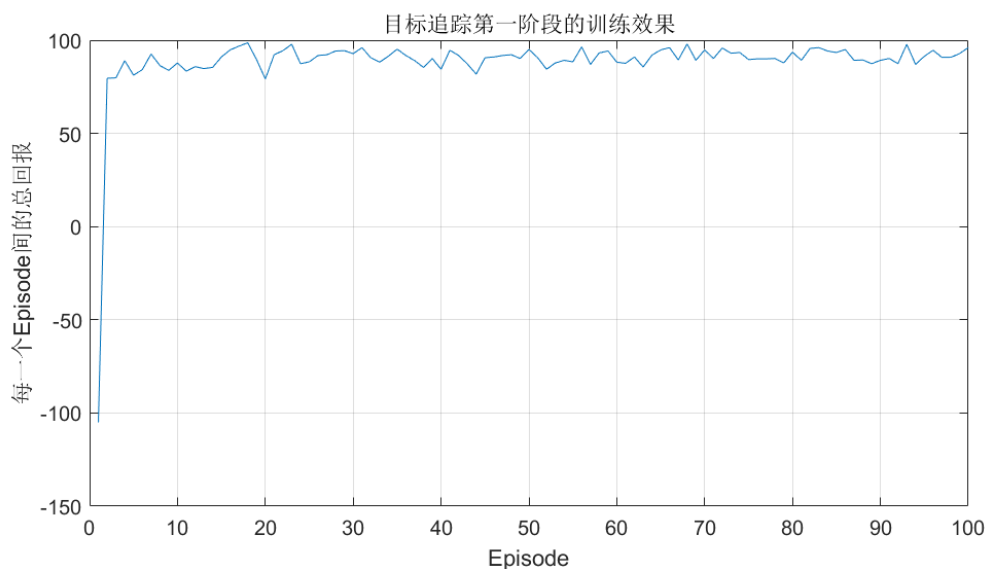


图 4.9 目标追踪第一阶段的训练效果

根据上图可以看出，第一个周期(episode)因处于探索过程，总回报非常低。之后总回报迅速上升，在 10 个周期后，目标追踪的效果已经趋于稳定，总回报均处于 90 上下。另一方面可以看到，即使已经平均了 5 趟(run)训练的结果，依然存在较大的振荡。这是因为在训练阶段，无人机根据  $\epsilon - greedy$  算法选择策略，具有一定程度的随机性。运行更多的趟数可以进一步平滑该结果。

训练无人机追踪能力的第二阶段在第一阶段 Q 表的基础上进行，采取随机目标点的训练模式，目的是尽可能遍及无人机的各个状态，提升目标追踪算法的泛化能力。实验参数中的贪心率、学习率和折损因子与表 4.5 相同，其余设定如表 4.6，训练过程的效果如图 4.10，访问到的状态如图 4.11。

表 4.6 目标追踪第二阶段训练参数

中文	目标位置	训练趟数	每趟的训练次数	基于模型的迭代步骤
英文	(x, y)	Number of run	Number of episodes	Planning steps
值	在[-3.6m, 3.6m]范围保留一位小数随机产生	2	50	5

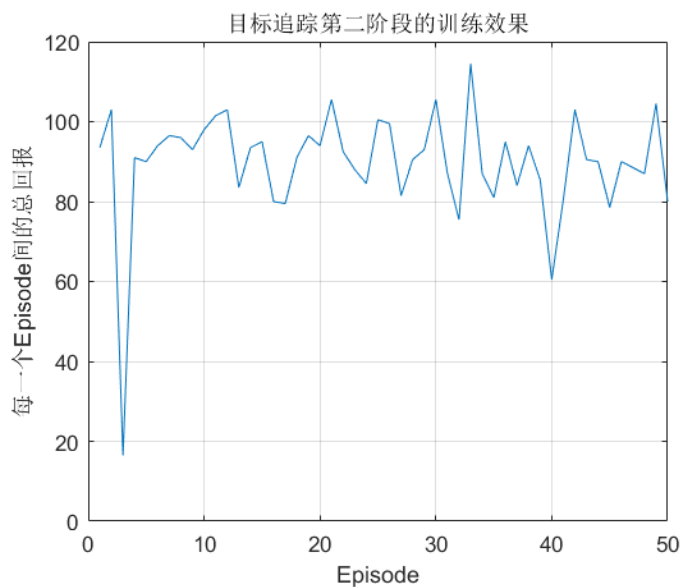


图 4.10 目标追踪第二阶段的训练效果

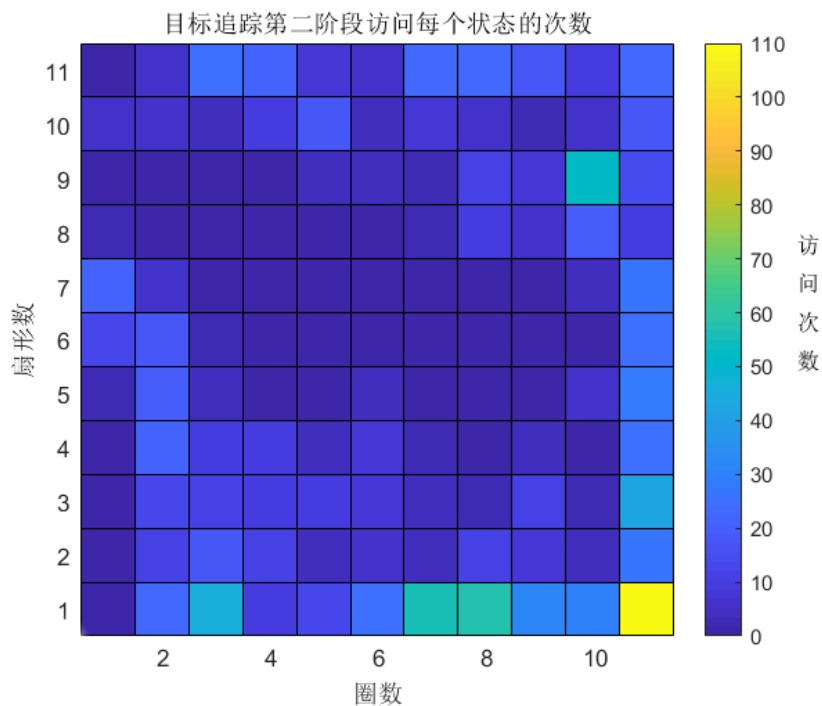


图 4.11 目标追踪第二阶段每个状态被访问到的情况

由于目标点的位置产生和无人机的动作选择均具有随机性，图 4.10 显示的训练效



果振荡更加严重,但追踪效果的泛化性在逐渐改善。从图 4.11 可以看出,几乎所有状态在随机训练期间均有被实际访问到,结合 Dyna-Q 算法 planning 的特性,证明这种训练方式可以充分训练值函数表,有效提升目标追踪算法的泛化能力。之后,又按照与表 4.6 相同的参数训练了 150 个周期,进一步提升性能。

## 2. 避碰能力的训练

下面在目标追踪的基础上,训练无人机的避碰能力。训练时,给定一个目标点和处于前进路线上的障碍点,保持目标追踪的 Q 表不变,训练负责避障的 Q 表。在执行任务时,依据两个 Q 表加和得出的 Q 值进行动作选择。当无人机与目标位置的距离小于 5cm,或无人机与障碍物的距离小于避碰最短距离,则该次训练结束。训练的具体参数如表 4.7, 初期五十次的训练效果如图 4.12。

表 4.7 避障能力训练参数

中文	贪心率	避碰最短距离	障碍位置	目标位置	训练趟数	每趟的训练次数
英文	Epsilon	Distance	(x', y')	(x, y)	Number of run	Number of episodes
值	0.2	0.5m	(0.8m, 0.6m)	(2.0m, 2.0m)	1	50

其余参数和表 4.7 相同。

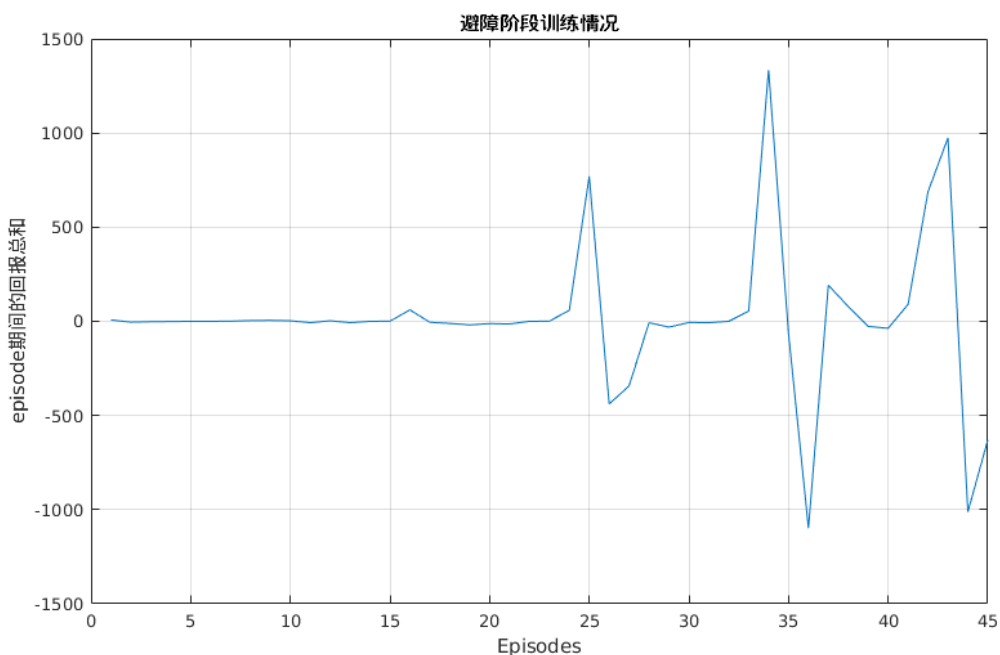


图 4.12 避障任务初期训练情况

训练中总回报陡然下降的情况是因为无人机与障碍物相撞而重新开始。除了这种相撞的情况,总的来说还是逐渐向更好的趋势发展。而相撞的情形随着 Q 表的逐渐稳定也获得了改善。之后,我又选择不同位置的障碍点训练了 200 次,完成避障能力的训练。

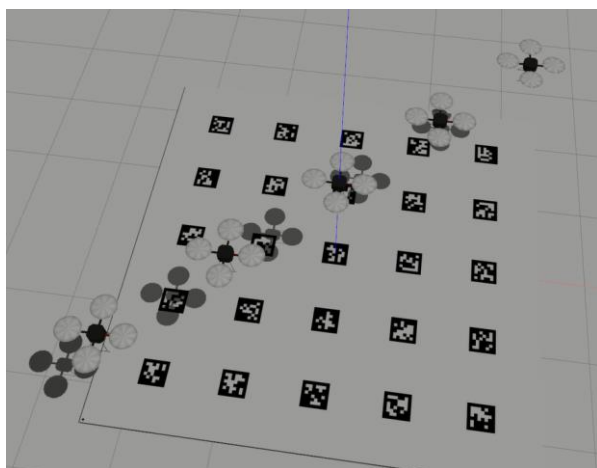
#### 4.3.2 编队效果测试

训练完单架无人机后,需将训练结果部署在多架无人机上组成编队,编队流程如下:

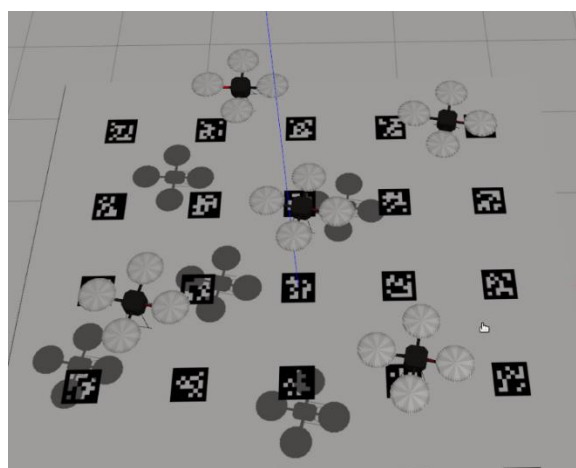
- (a) 人机交互界面程序将相对于 leader 无人机的编队位置发送给 follower 无人机。
- (b) Follower 无人机将相对位置坐标变换为世界坐标系下的位置。
- (c) 根据内部的强化学习算法控制无人机飞行。

##### 1. 仅目标跟踪的编队效果

为测试控制效果,将一字排开的无人机(如图 4.13a)控制为围绕 leader 的正方形构型(如图 4.13b),设定自身位置与目标位置误差小于 5cm 则停止动作。



(a) 编队之前的五架无人机



(b) 编队之后的五架无人机

图 4.13 仿真环境中的编队过程

Uav1(leader)的位置为(0.1, -0.1),朝 z 轴旋转-0.2rad。最终编队位置与实际位置的误差如表 4.8,编队结果如图 4.14。

表 4.8 编队结果及误差情况

名称	Uav2	Uav3	Uav4	Uav5	均值
编队命令	(-1, -1)	(1, -1)	(1, 1)	(-1, 1)	—
期望 x 坐标/m	-0.6814	1.2787	0.8814	-1.0787	—
期望 y 坐标/m	1.0787	0.6814	-1.2787	-0.8814	—

实际 x 坐标/m	-0.6828	1.2378	0.8785	-1.0370	—
实际 y 坐标/m	1.0735	0.6928	-1.3165	-0.8697	—
距离误差/m	0.0055	0.0425	0.0379	0.0434	<b>0.0323m</b>

从表中可以得出,编队的距离误差均值为 0.0323m,即 3cm 左右,总的来说定位误差均小于 5cm。

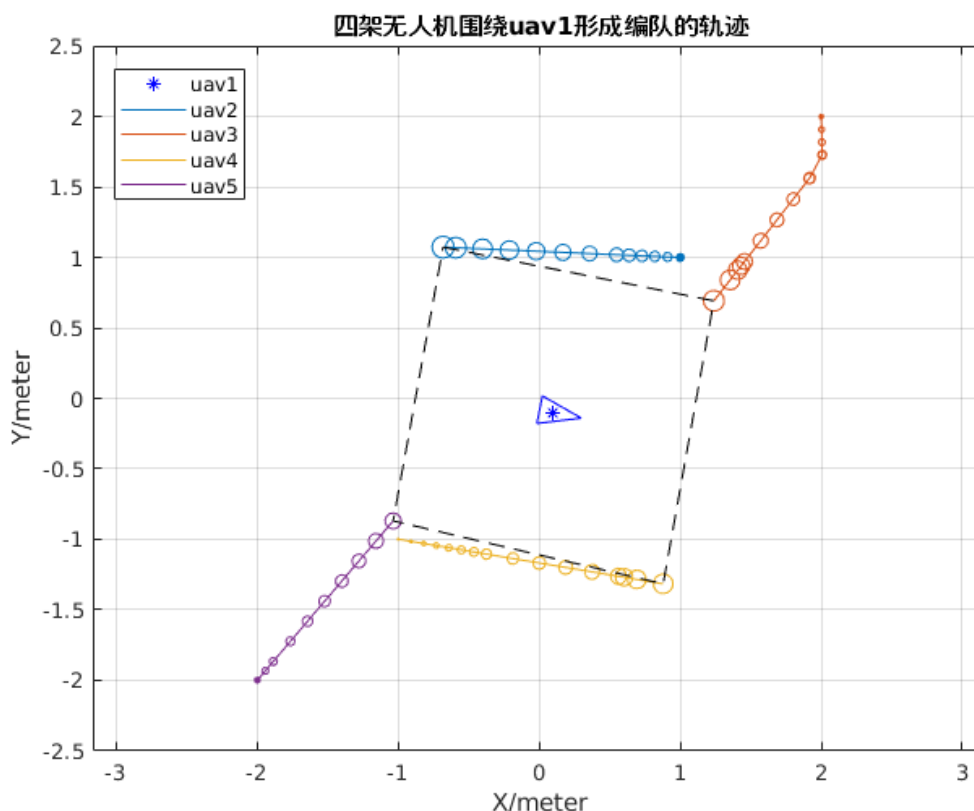
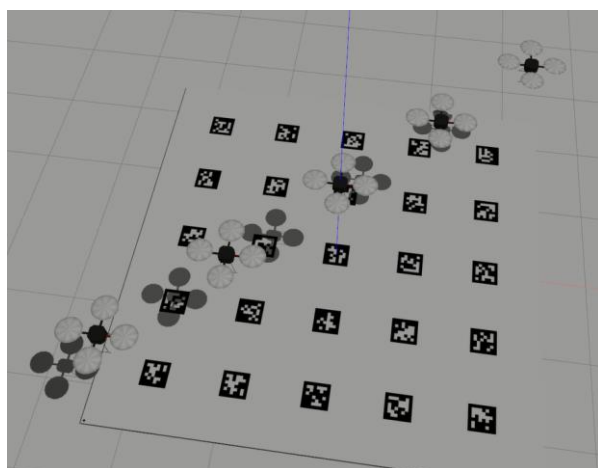


图 4.14 仅通过目标追踪实现的编队效果

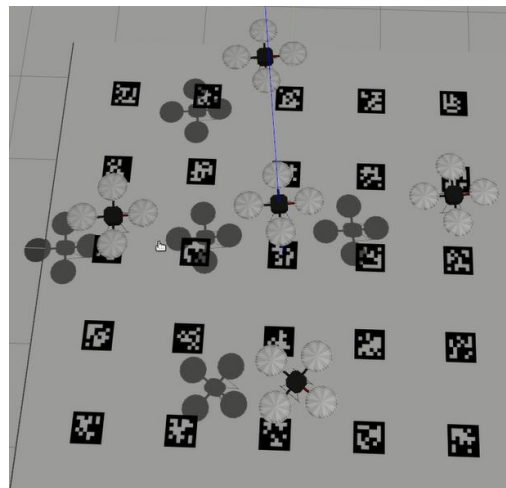
图中蓝色三角代表 leader 无人机所处的位置与朝向,其他四条颜色的轨迹为无人机的飞行轨迹。无人机从小圈向大圈飞去,相同大小的圆圈代表同一步(step)的动作。四架无人机在最开始呈一字排列,经过一定时间的运动,完成了围绕 leader 无人机的正方形编队构型,且正方形偏转的角度与 leader 无人机一致。总的来说,基于强化学习的目标追踪算法可以使无人机完成编队任务,编队的位置误差在 3cm 左右。

## 2. 带避碰的编队效果

下一步,将避障任务的训练效果应用于编队任务中。为测试编队效果,将一字排开的无人机(如图 4.15a)控制为围绕 leader 无人机的正方形(如图 4.15b),但对应的位置点无法只通过目标追踪算法到达。设定自身位置与目标位置误差小于 8cm 则停止动作。



(a) 编队之前的五架无人机



(b) 编队之后的五架无人机

图 4.15 仿真环境中带避障的编队过程

Uav1(leader)的位置为(0.0, 0.0), 朝 z 轴旋转+0.1rad。编队位置与实际位置的误差如表 4.9, 编队过程中的轨迹如图 4.16。

表 4.9 带避障的编队结果及误差情况

名称	Uav2	Uav3	Uav4	Uav5	均值
编队命令	(0, -1.5)	(1.5, 0)	(0, 1.5)	(-1.5, 0)	—
期望 x 坐标/m	0.1498	1.4925	-0.1498	-1.4925	—
期望 y 坐标/m	-1.4925	0.1498	1.4925	-0.1498	—
实际 x 坐标/m	0.2001	1.5228	-0.1919	-1.4305	—
实际 y 坐标/m	-1.5405	0.1324	1.4570	-0.1601	—
距离误差/m	0.0696	0.0349	0.0551	0.0629	<b>0.0556</b>

从表中可以得出, 编队的距离误差均值为 0.0556m, 即 5.5cm 左右, 总的来说定位误差均小于 8cm。

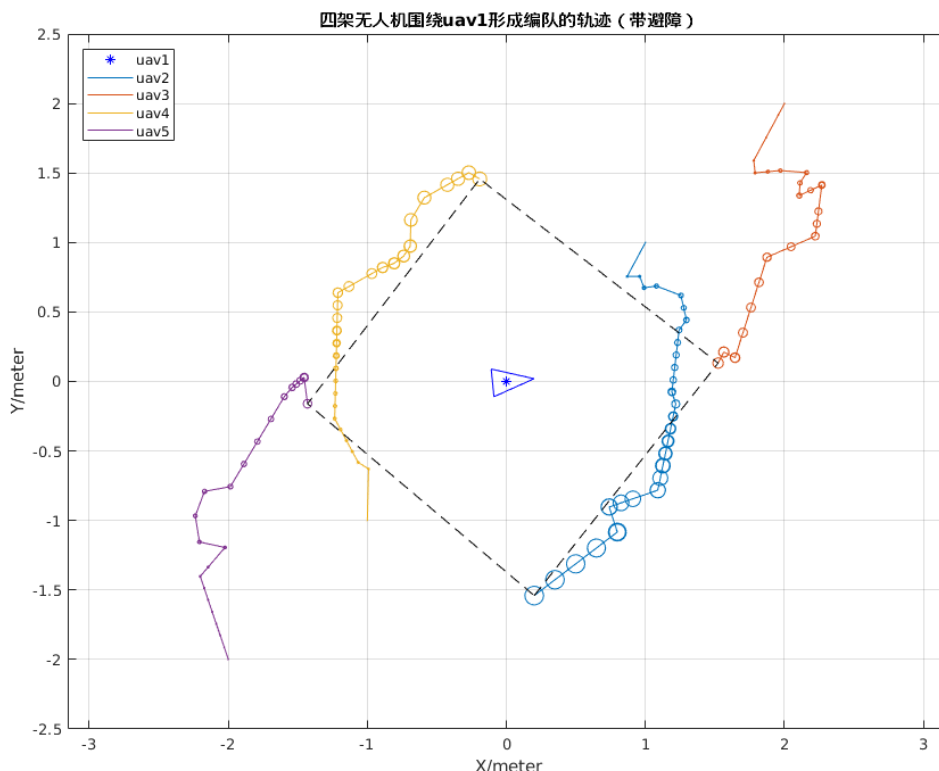


图 4.16 包含避障功能实现的编队效果

图中蓝色三角代表 leader 无人机所处的位置与朝向，其他四条颜色的轨迹为无人机的飞行轨迹。无人机从小圈向大圈飞去，相同大小的圆圈代表同一步(step)的动作。四架无人机在最开始呈一字排列，经过一定时间的运动，完成了围绕 leader 无人机的正方形编队构型，且正方形偏转的角度与 leader 无人机一致。

可以看到 UAV2 和 UAV4 无人机在飞行的过程有避开 UAV1 的动作，同时 UAV3 和 UAV5 在开始阶段为避免相撞，有一个向外飞等待的过程。最终的定位精度由目标追踪算法和对目标设定的阈值决定。目标算法越好，精度越高，时间越少。目标设定阈值越小，精度越高，但花费的时间越长。

#### 4.4 本章小结

本章介绍了整个多无人机编队系统的构成及相关测试。本章开始介绍了仿真系统所模拟的真实无人机系统框图，进而总结了无人机进行定位-编队的流程。接着介绍视觉定位部分进行的实验，包括二维码检测解码算法的测试，和最终组合导航的定位效果。最后介绍了与强化学习编队相关的实验，包括目标追踪与避碰能力的训练过程，和五架无人机包含内部避碰的协同定位与控制实验。总的来说，本系统完成了基于机器视觉与强化学习的协同定位与控制任务，实现了本课题最初制定的全部目标。

## 结论

随着无人机任务复杂度的提升,传统的卫星导航方式与单机执行任务的模式均面临着挑战,集群化、智能化成为无人机新的发展方向。在此背景下,研究自主性强的视觉定位算法,并与编队控制算法结合,实现完整的多无人机定位-决策-控制系统,对于发挥无人机的应用潜力具有重要意义。本论文从实际应用角度出发,以实验室现有的 smart-swarm 仿真平台为基础,一方面实现基于二维码的视觉定位流程,并借助工具包完成视觉导航与惯性导航的组合导航;另一方面基于强化学习完成对目标追踪和避碰任务的任务划分与训练,最终将算法应用于多架无人机,实现带避碰的协同定位与编队控制任务。

本论文的主要研究工作及研究成果简述如下:

- 1) 在国内外关于二维码定位技术研究的基础上,结合计算机视觉知识,研究了建立视觉模型、相机标定、对 apriltag 二维码进行检测、解码和求解位姿的过程。接着使用 apriltag\_ros 工具包将算法融入无人机仿真平台,最后使用 robot localization 工具包完成视觉定位与惯性测量元件的信息融合。该方法的优点是自主性强,精度较高,成本低廉,可用于室内定位场景。
- 2) 在国内外关于强化学习和编队任务研究的基础上,结合强化学习基础知识,完成编队任务的分解、Dyna-Q 算法的实现、目标追踪任务和避碰任务的状态划分与训练,最终将训练结果应用于多架无人机,完成编队任务。与其他编队控制算法相比,该方法考虑了无人机的避碰情况,更具有实用意义。
- 3) 基于实验室 smart-swarm 平台实现上述内容与相应算法,并进行测试。经测试,组合导航精度小于 4cm,带避障的编队控制误差小于 8cm。

由于时间有限,本论文在以下几个方面有待进一步尝试和改进:

1. 本论文中关于视觉定位的实验都是在仿真环境下完成的,应用于实际场景可能需要进一步优化。日后可以将算法迁移到真实无人机上进行测试,对实际照片中存在的种种问题进行改进,并通过 Motion Capture 动作捕捉装置测试精度。
2. 本论文中关于强化学习的内容是在离散空间,二维平面下进行的,控制速度慢。日后可以引入深度强化学习网络,将状态空间和动作空间拓展到三维的连续空间。
3. 本论文的代码没有详细的文档说明。日后可以加入教程,将本论文实现的功能整合到实验室的无人机仿真平台中,方便其他同学进行更深入的研究。



## 致谢

春去秋来，四年的大学时光转瞬即逝。我设想过多许多大学毕业的可能，但从未想过因为一场百年难遇的疫情，最后一个学期我竟会完全在家度过。除了关于毕业设计铺天盖地的各种通知、答辩与讨论在提醒着我“快毕业了”，其他方面完全没有毕业的感觉。在此，要特别感谢郑老师、韩老师在这次毕业设计中的指导。虽然是远程指导我的毕业设计，但在每周一次的毕设进度交流中，老师都会听取我的进度与遇到的困难，及时纠正研究方向，和我一起解决软件或硬件上的问题。印象最深的是老师为让我进行无人机调试，提出要给我寄一架无人机，类似这样的事情点点滴滴还有很多。此外，由于疫情对我未来出国读研究生的计划影响非常严重，老师还不断与我讨论未来的规划，为我出谋划策，我从心底里感到非常感激！之前我曾在校外的科研机构进行过一段短期科研，两相比较，我深深感到了北航的老师对学生的认真负责。再一次感谢郑老师与韩老师在毕设中的指导！

毕业设计也是对大学四年的总结与体现。首先感谢我的本科导师，郭雷教授，对我整个大学生涯的关怀与指导。简单总结我的大学：大一学数理，大二搞科创，大三学英语，大四做申请。感谢大一教我数分、高代、物理的几位老师，不仅为我打下坚实的数理基础，更让我体会到数学与物理的美。感谢大二唐老师给我们做模电项目的机会，让我第一次体验到自动化学科做项目的快乐。感谢在北航航模队遇到的两位教练以及那么多志同道合的小伙伴，我永远也忘不了亲手制作的飞机成功飞上蓝天的快乐，永远也忘不了那两个暑假我们在准备比赛时，共同经历的熬夜、挨批、聚餐、胜利，更忘不了第二年自己当飞手摔飞机最后调整心态参加比赛的全过程。这两年的经历完成了我儿时的梦想，也真正教会我在逆境中永不言弃，教会我如何赢得光彩，输得大气。感谢大三大四学英语和准备出国材料中帮助过我的学长学姐与老师们，这段经历不仅让我迈出“国际视野”的第一步，更教会我如何做时间的朋友，明白“复利”的重要性。总的来说，我感觉自己的大学生活十分充实而丰富，我没有遗憾。感谢母校北航与高等理工学院为我提供如此优质的平台与如此充足的机会！

此外，还要感谢女朋友在这四年的陪伴，没有你的支持就不会有今天的我。感谢我的父母一直关照我，让我在生活上没有后顾之忧。感谢一路走来遇到的良师益友。

前路漫漫，希望我能永保赤子之心，未来为国家机器人领域的发展贡献自己的力量！

## 参考文献

- [1] Floreano D, Wood R J. Science, Technology and the Future of Small Autonomous Drones[J]. Nature, 2015, 521(7553):460-466.
- [2] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, 030(001):86-100.
- [3] 赵文一. 无人机视觉辅助自主降落系统研究[D]. 哈尔滨:哈尔滨工业大学, 2018.
- [4] Kato H, Billinghurst M. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System[A]. Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99) [C]. Washington, DC, USA: IEEE Computer Society, 1999: 85-94.
- [5] Fiala M. Artag, an Improved Marker System Based on ARtoolkit[J]. National Research Council Canada, 2004: NRC, 2004, 47419.
- [6] Olson E. AprilTag: A Robust and Flexible Visual Fiducial System[A]. 2011 IEEE International Conference on Robotics and Automation[C]. Washington, DC, USA: IEEE Computer Society, 2011: 3400-3407.
- [7] Wang J, Olson E. AprilTag 2: Efficient and Robust Fiducial Detection[A]. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[C]. Washington, DC, USA: IEEE Computer Society, 2016: 4193-4198.
- [8] Krogus M, Haggemiller A, Olson E. Flexible Layouts for Fiducial Tags[A]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[C]. Washington, DC, USA: IEEE Computer Society, 2019: 1898-1903.
- [9] Lepetit V, Moreno-Noguer F, Fua P. Epnp: An Accurate  $O(n)$  Solution to the PnP Problem[J]. International Journal of Computer Vision, 2009, 81(2): 155.
- [10] Collins T, Bartoli A. Infinitesimal Plane-Based Pose Estimation[J]. International Journal of Computer Vision, 2014, 109(3):252-286.
- [11] Cheng L, Song B, Dai Y, et al. Mobile Robot Indoor Dual Kalman Filter Localization Based on Inertial Measurement and Stereo Vision[J]. CAAI Transactions on Intelligence Technology, 2019, 2(4): 173-181.
- [12] 肖献强, 程亚兵, 王家恩. 基于惯性和视觉复合导航的自动导引小车研究与设计[J].



中国机械工程, 2019, 30(22):2734-2740.

[13] Huh S, Shim D H, Kim J. Integrated Navigation System Using Camera and Gimbaled Laser Scanner for Indoor and Outdoor Autonomous Flight of UAVs[A]. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems[C]. Washington, DC, USA: IEEE Computer Society, 2013: 3158-3163.

[14] 吴晋, 张国良, 曾静, 等. 多机器人编队离散模型及队形控制稳定性分析[J]. 控制理论与应用, 2014, 31(3):293-301.

[15] Han L, Dong X, Li Q, et al. Formation Tracking Control for Time-delayed Multi-agent Systems with Second-Order Dynamics[J]. Chinese Journal of Aeronautics, 2017, 30(1):348-357.

[16] Mnih V, Kavukcuoglu K, Silver D, et al. Human-Level Control through Deep Reinforcement Learning[J]. Nature, 2015, 518(7540): 529-533.

[17] Silver D, Huang A, Maddison C J, et al. Mastering the Game of Go with Deep Neural Networks and Tree Search[J]. Nature, 2016, 529(7587): 484.

[18] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous Control with Deep Reinforcement Learning[J]. Computer Science, 2015, 8(6):A187.

[19] 史豪斌, 徐梦. 基于强化学习的旋翼无人机智能追踪方法[J]. 电子科技大学学报, 2019, 048(004):553-559.

[20] Hwangbo J, Sa I, Siegwart R, et al. Control of a Quadrotor with Reinforcement Learning[J]. IEEE Robotics and Automation Letters, 2017, 2(4): 2096-2103.

[21] Knopp M, Aykın C, Feldmaier J, et al. Formation Control Using GQ ( $\lambda$ ) Reinforcement Learning[A]. 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)[C]. Washington, DC, USA: IEEE, 2017: 1043-1048.

[22] Hung S M, Givigi S N. A Q-learning Approach to Flocking with UAVs in a Stochastic Environment[J]. IEEE Transactions on Cybernetics, 2016, 47(1): 186-197.

[23] Speck C, Bucci D J. Distributed UAV Swarm Formation Control via Object-Focused, Multi-Objective SARSA[A]. 2018 Annual American Control Conference (ACC)[C]. Washington, DC, USA: IEEE, 2018: 6596-6601.

[24] Koch W, Mancuso R, West R, et al. Reinforcement Learning for UAV Attitude Control[J].



ACM Transactions on Cyber-Physical Systems, 2019, 3(2): 1-21.

[25] 罗高, 袁亮. 基于二维码的单目视觉测距移动机器人定位研究[J]. 组合机床与自动化加工技术, 2017(10).

[26] Koubâa, Anis. Robot Operating System (ROS)[M]. Berlin: Springer-Verlag, 2017:595-617.

[27] 秦永元, 张洪钺, 王叔华. 卡尔曼滤波与组合导航原理[M]. 西安:西北工业大学出版社, 2012.

[28] Du T, Guo L, Yang J. A Fast Initial Alignment for SINS Based on Disturbance Observer and Kalman Filter[J]. Transactions of the Institute of Measurement and Control, 2016, 38(10): 1261-1269.

[29] 宗群, 王丹丹, 邵士凯, 等. 多无人机协同编队飞行控制研究现状及发展[J]. 哈尔滨工业大学学报, 2017(3).

[30] Groves P D. GNSS 与惯性及多传感器组合导航系统原理[M]. 练军想, 唐康华等译. 北京:国防工业出版社, 2011:145-152.

[31] Quigley M, Conley K, Gerkey B, et al. ROS: An Open-Source Robot Operating System[A]. ICRA Workshop on Open Source Software[C]. Washington, DC, USA: IEEE Computer Society, 2009, 3(3.2): 5.

[32] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[A]. European Conference on Computer Vision[C]. Cham: Springer, 2014: 834-849.

[33] Brian Gerkey. ROS Wiki[DB/OL]. <https://wiki.ros.org/>, 2018-11-29.

[34] Andrew Howard. Gazebo Website[DB/OL]. <http://gazebo.org/>, 2020-03-11.

[35] Dmitry Kurtaev. OpenCV Documentation[DB/OL]. <https://docs.opencv.org/3.4.9/>, 2019-12-19.

[36] Moore T, Stouch D. A Generalized Extended Kalman Filter Implementation for the Robot Operating System[A]. Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)[C]. Cham: Springer, 2016: 335-348.

[37] Moore T. Robot\_localization Documentation[DB/OL]. [http://docs.ros.org/melodic/api/robot\\_localization/html/index.html](http://docs.ros.org/melodic/api/robot_localization/html/index.html), 2020-03-15.

[38] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[M]. Cambridge, MA:



MIT press, 2018: 47-50, 159-168.

[39] Tanner B, White A. RL-Glue: Language-Independent Software for Reinforcement-Learning Experiments[J]. Journal of Machine Learning Research, 2009, 10(Sep): 2133-2136.

[40] 苗国英, 马倩. 多智能体系统的协调控制研究综述[J]. 南京信息工程大学学报(自然科学版), 2013, 5(5):385-396.

## 附录

### 附录 A: apriltag\_ros 配置文件

settings.yaml:

```
1. tag_threads:      4          # default: 2
2. tag_decimate:     1.0        # default: 1.0
3. tag_blur:         0.0        # default: 0.0
4. tag_refine_edges: 1          # default: 1
5. tag_debug:        0          # default: 0
6. # Other parameters
7. publish_tf:       true       # default: false
```

tags.yaml:

```
1. tag_bundles:
2.   [
3.     {
4.       name: 'map_frame',
5.       layout:
6.         [
7.           {id: 0, size: 0.30, x: -1.8000, y: -
8.             1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
9.           {id: 1, size: 0.30, x: -0.9000, y: -
10.            1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
11.          {id: 2, size: 0.30, x: 0.0000, y: -
12.            1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
13.          {id: 3, size: 0.30, x: 0.9000, y: -
14.            1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
15.          {id: 4, size: 0.30, x: 1.8000, y: -
16.            1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
17.          {id: 5, size: 0.30, x: -1.8000, y: -
18.            0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
19.          {id: 6, size: 0.30, x: -0.9000, y: -
20.            0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
21.          {id: 7, size: 0.30, x: 0.0000, y: -
22.            0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
23.          {id: 8, size: 0.30, x: 0.9000, y: -
24.            0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
25.          {id: 9, size: 0.30, x: 1.8000, y: -
26.            0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},
```

```
17.      {id: 10, size: 0.30, x: -  
    1.8000, y: 0.0000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},  
18.      {id: 11, size: 0.30, x: -  
    0.9000, y: 0.0000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},  
19.      {id: 12, size: 0.30, x: 0.0000, y: 0.0000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
20.      {id: 13, size: 0.30, x: 0.9000, y: 0.0000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
21.      {id: 14, size: 0.30, x: 1.8000, y: 0.0000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
22.      {id: 15, size: 0.30, x: -  
    1.8000, y: 0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},  
23.      {id: 16, size: 0.30, x: -  
    0.9000, y: 0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},  
24.      {id: 17, size: 0.30, x: 0.0000, y: 0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
25.      {id: 18, size: 0.30, x: 0.9000, y: 0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
26.      {id: 19, size: 0.30, x: 1.8000, y: 0.9000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
27.      {id: 20, size: 0.30, x: -  
    1.8000, y: 1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},  
28.      {id: 21, size: 0.30, x: -  
    0.9000, y: 1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz: 0.0},  
29.      {id: 22, size: 0.30, x: 0.0000, y: 1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
30.      {id: 23, size: 0.30, x: 0.9000, y: 1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0},  
31.      {id: 24, size: 0.30, x: 1.8000, y: 1.8000, z: 0.0, qw: 1.0, qx: 0.0, qy: 0.0, qz  
    : 0.0}  
32.    ]  
33.  }  
34. ]
```

## 附录 B: robot\_localization 配置文件

tags\_imu\_ekf.yaml

```
1. ekf_se_map:  
2.   frequency: 30  
3.   sensor_timeout: 0.1  
4.   two_d_mode: false  
5.   transform_time_offset: 0.0
```

```
6.   transform_timeout: 0.0
7.   print_diagnostics: true
8.   debug: false
9.   # Defaults to "robot_localization_debug.txt" if unspecified. Please specify the full
    ll path.
10.  debug_out_file: /home/ljj/Desktop/robot_localization_debug.txt
11.
12.  map_frame: map
13.  odom_frame: odom
14.  base_link_frame: base_link
15.  world_frame: map
16.
17.  pose0: /uav1/tags_pose
18.  pose0_config: [true, true, true,
19.                false, false, false,
20.                false, false, false,
21.                false, false, false,
22.                false, false, false]
23.  pose0_differential: false
24.  pose0_relative: false
25.  pose0_queue_size: 5
26.  pose0_rejection_threshold: 10 # Note the difference in parameter name
27.  pose0_nodelay: false
28.
29.  imu0: /uav1/raw_imu
30.  # imu0: /imu_new
31.  imu0_config: [false, false, false,
32.               true, true, true,
33.               false, false, false,
34.               true, true, true,
35.               true, true, true]
36.  imu0_nodelay: true
37.  imu0_differential: false
38.  imu0_relative: false
39.  imu0_queue_size: 10
40.  imu0_remove_gravitational_acceleration: true
41.  gravitational_acceleration: 9.80
42.
43.  use_control: false
44.
45. # Q matrix. difficult to tune.
46.  process_noise_covariance: [0.05, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
```



```
47.          0, 0.05, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
48.          0, 0, 0.05, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
49.          0, 0, 0, 0.03, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
50.          0, 0, 0, 0, 0.03, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
51.          0, 0, 0, 0, 0, 0.03, 0, 0,
    0, 0, 0, 0, 0, 0,
52.          0, 0, 0, 0, 0, 0, 0.01, 0,
    0, 0, 0, 0, 0, 0,
53.          0, 0, 0, 0, 0, 0, 0, 0.01,
    0, 0, 0, 0, 0, 0,
54.          0, 0, 0, 0, 0, 0, 0, 0, 0.01,
    0, 0, 0, 0, 0, 0,
55.          0, 0, 0, 0, 0, 0, 0, 0,
    0.03, 0, 0, 0, 0, 0,
56.          0, 0, 0, 0, 0, 0, 0, 0,
    0, 0.03, 0, 0, 0, 0,
57.          0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0.03, 0, 0, 0,
58.          0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0.01, 0, 0,
59.          0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0.01, 0,
60.          0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0.015]
61.
62. # P0
63. initial_estimate_covariance: [1.0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
64.          0, 1.0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
65.          0, 0, 1.0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
66.          0, 0, 0, 1.0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
67.          0, 0, 0, 0, 1.0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
68.          0, 0, 0, 0, 0, 1.0, 0, 0, 0,
    0, 0, 0, 0, 0, 0,
69.          0, 0, 0, 0, 0, 0, 1.0, 0, 0,
    0, 0, 0, 0, 0, 0,
```



70.					0,	0,	0,	0,	0,	0,	0,	1.0,	0,
	0,	0,	0,	0,	0,	0,							
71.					0,	0,	0,	0,	0,	0,	0,	0,	1.0,
	0,	0,	0,	0,	0,	0,							
72.					0,	0,	0,	0,	0,	0,	0,	0,	0,
	1.0,	0,	0,	0,	0,	0,							
73.					0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	1.0,	0,	0,	0,	0,							
74.					0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	1.0,	0,	0,	0,							
75.					0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	1.0,	0,	0,							
76.					0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	1.0,	0,							
77.					0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	1.0]							