



東南大學

本科毕业设计（论文）报告

题目： 基于视觉伺服的共享遥操作
系统控制设计

学 号： 22020206

姓 名： 张子哲

学 院： 仪器科学与工程学院

专 业： 测控技术与仪器

指导教师： 杨缘

起止日期： 2023年12月-2024年6月

东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____ 日期：_____年____月____日

东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内 容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘 要

近年来，随着工业自动化和智能制造技术的迅猛发展，工业机器人在各个领域得到了广泛应用。尤其是在制造业中，机械臂作为一种关键的自动化设备，极大的提高了生产效率和产品质量。然而，传统的机械臂控制方法通常依赖于预设的关节轨迹，这种方法仅在关节空间内实现闭环控制，其末端或本体的控制依赖于精确且经过标定的机械臂模型。这种控制方法存在诸多不足，如缺乏对机械臂末端位姿的实时反馈控制，难以应对复杂和动态的作业环境。为了实现对机械臂精确度更高，实时性更好，自主性更强的控制，使其执行复杂度更高，难度更大的任务，本文结合视觉伺服与机器人遥操作技术，提出了一种基于视觉伺服的共享遥操作方案，完成了完整的系统设计并进行仿真和实物实验验证。该系统融合了机器人技术与计算机视觉技术，视觉伺服实现了机械臂对外界环境的感知，提升了机器人的自主性，并且通过实时获取图像信息，实现了对机械臂末端位姿的实时反馈控制；遥操作则弥补了视觉伺服控制精度不够，收敛域小等问题，实现了对机械臂更加精细的控制，多重控制方式的融合也增强了系统的灵活性。

关键词：视觉伺服，实时控制，遥操作，共享控制

ABSTRACT

In recent years, with the rapid development of industrial automation and intelligent manufacturing technology, industrial robots have been widely used in various fields. Especially in the manufacturing industry, robotic arms, as a key automation equipment, have greatly improved production efficiency and product quality. However, traditional robotic arm control methods usually rely on preset joint trajectories, which realize closed-loop control only in the joint space. The end or body control relies on an accurate and calibrated robot arm model. This approach has a number of shortcomings. For example, it lacks real-time feedback control of the end position of the arm, and it is difficult to cope with complex and dynamic operating environments. In order to achieve more accurate, real-time, and autonomous control of the robot arm, and to enable it to perform more complex and difficult tasks, this paper combines visual servoing with robot teleoperation to provide a more accurate, real-time, and autonomous control of the robot arm. This paper combines visual servo and robot teleoperation technology, proposes a shared teleoperation program based on visual servo, completes the complete system design and carries out simulation and physical experiment verification. The system integrates robotics and computer vision technology. The visual servo realizes the robotic arm's perception of the external environment and enhances the robot's autonomy. And through the real-time acquisition of image information, it realizes the real-time feedback control of the position of the end of the robotic arm; remote operation makes up for the problems such as insufficient precision of visual servo control, small convergence domain, etc., and realizes the control of the end of the robot. Remote operation compensates for the lack of precision of visual servo control and the small convergence domain, and realizes a more fine control of the robot arm, and the integration of multiple control modes also enhances the flexibility of the system.

KEY WORDS: Visual servo, Real-time control, Teleoperation, Shared control

目 录

摘 要	I
ABSTRACT	II
目 录	III
第一章 绪论	1
1.1 课题背景和意义	1
1.2 研究现状	4
1.3 本文研究内容	7
第二章 理论基础	8
2.1 成像几何	8
2.1.1 引言	8
2.1.2 相机坐标系	8
2.1.3 透视投影	9
2.2 视觉伺服控制理论基础	10
2.2.1 引言	10
2.2.2 视觉伺服的基本组成	10
2.2.3 基于图像的视觉伺服控制 (IBVS)	11
2.2.4 基于位置的视觉伺服控制 (PBVS)	13
2.3 遥操作控制原理	15
2.3.1 引言	15
2.3.2 遥操作系统的基本组成和工作原理	15
2.3.3 基于无源性的控制	15
2.3.4 共享控制	17
2.4 本章小结	18
第三章 系统设计	19
3.1 引言	19
3.2 系统架构	19
3.2.1 硬件设计	19

3.2.2 软件设计	20
3.3 机械臂建模	21
3.3.1 引言	21
3.3.2 正运动学建模	21
3.3.3 速度运动学建模	22
3.4 视觉伺服控制算法	24
3.4.1 引言	24
3.4.2 图像特征识别	24
3.4.3 基于位置的方法	24
3.4.4 基于图像的方法	25
3.5 直接控制算法	26
3.6 共享控制策略	27
3.7 本章小结	28
第四章 实验验证	30
4.1 实验设置	30
4.2 视觉伺服仿真实验	30
4.2.1 实验目的	30
4.2.2 实验过程	30
4.2.3 实验结果与分析	33
4.3 视觉伺服实物实验	33
4.3.1 实验目的	33
4.3.2 实验过程	34
4.3.3 实验结果与分析	37
4.4 共享控制实验	38
4.4.1 实验目的	38
4.4.2 实验过程	38
4.4.3 实验结果与分析	39
4.5 本章小结	41
第五章 总结与展望	42

5.1 工作总结	42
5.2 工作展望	42
参考文献	44
致 谢	47

第一章 绪论

1.1 课题背景和意义

在工业生产制造领域，能够进行长时间高强度作业的工业机器人得到了广泛应用。这一变化不仅提高了现代生产力和工作效率，还降低了生产成本，为人们的日常生活提供了便利，进一步改善了生活质量。随着科学技术的不断进步和人们生活水平的提升，机器人应用的范围逐渐扩展到日常生活的各个方面。机械臂作为最基本的工业机器人之一，在制造领域中扮演着不可或缺的自动化设备角色。然而，迄今为止，大多数机械臂的控制方法仍然依赖于预设关节轨迹来实现。这类控制方法仅在关节空间内实现闭环控制，其末端或本体的控制依赖于精确且经过标定的机械臂模型，缺乏针对机械臂末端位姿反馈的真正闭环控制系统。这主要是因为传统控制方案未能充分利用传感器从外部获取位姿信息。

人们希望机械臂不仅能够代替人类完成低难度、重复性、体力导向的劳动，还能够发挥人类认知判断能力和机器优势。这不仅包括高效准确的工作能力，还包括对外部环境状态进行分析和判断的能力。尤其是在医疗、航天、智能家居以及不适宜人类工作的恶劣环境下，人们期望机器人能够替代人类完成各种高难度的任务。在这种背景下，视觉伺服技术结合机器人技术与机器视觉技术，通过增强机械臂的感知能力，实现对外界环境的感知，从而构建了闭环控制系统。这种结合不仅提升了机械臂的运动平稳性，还显著增强了其环境适应能力和任务执行能力。



(a) 达芬奇手术机器人



(b) 工业机器人

图 1-1 应用于多种环境的机器人

与传统的接触式传感器相比，视觉传感器可以实现无需接触目标物体的测量和控制，避免了对目标物体造成影响的可能性。在适用范围上，视觉伺服技术适用于不同尺度、形状和表面特性的目标物体，具有很强的适应性和灵活性。视觉伺服系统可以实时地获取图

像信息，并根据实时反馈进行控制，适用于需要快速响应和高速运动的应用场景^[1]。随着图像处理算法和计算能力的不断提高，视觉伺服系统可以实现对目标位置和姿态的高精度控制。同时，一些视觉伺服系统可以在无需对目标物体进行标记或预先建模的情况下实现目标检测和跟踪，简化了系统的部署和维护过程。由于上述优点，视觉伺服控制系统的研究对机器人领域的发展具有十分重要的意义，并且已经成为了当前机器人研究领域的热点。

一个简单视觉伺服控制系统的实现最少需要一台机械臂和一个视觉传感器来实现系统的搭建。作为一个非线性，强耦合的系统，与其相关的研究领域包括计算机视觉，图像处理，运动学，动力学控制科学等。在视觉伺服系统中，物体的图像信息通过视觉传感器获取，然后通过计算机视觉技术进行处理，提取目标像素特征点像素坐标，再通过计算，得到物体的位置和姿态等信息。同时，读取机器人的状态，通过计算运动学，得到机器人的运动状态，并依据物体的位姿信息指导机器人执行下一步动作^[2]，如图1-2所示。

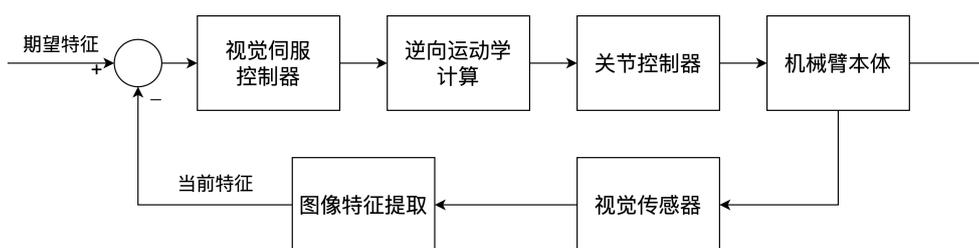


图 1-2 视觉伺服框图

遥操作技术是一种通过通信网络和输入输出设备，将人类的感知和操作意图远程传递并实现的操作手段^[3]。这项技术的核心在于将操作者纳入控制回路，对远端机器人进行实时在线操作，如图1-3所示：



图 1-3 遥操作系统回路

这种方法不仅让操作者能够直接控制远端机器人，还能够实时获取操作反馈，从而实现了对复杂任务的精准执行。

在遥操作系统中，操作者通过主端控制器的操作机构将控制指令传递给从端机器人。机器人接收到这些指令后，立即执行相应的操作，并将当前的状态信息以及自身传感器采

集到的数据，通过通信回路反馈给主端。这种双向信息传递机制，一方面通过人工参与提升了机器人“智能化”的水平，弥补了当前机器人自主能力的不足；另一方面，通过从端反馈的信息增强了操作者对操作环境的感知，使其能够实时掌握现场情况。

遥操作技术不仅使机器人能够接收操作员的控制指令，实现运动控制，还显著提升了系统的整体灵活性和效率。由于人类操作员的认知决策能力与机器人的负载操作能力相结合，遥操作机器人被广泛应用于复杂、非结构化环境中，执行远程接触目标物体的任务。

在空间探索领域，遥操作技术发挥了重要作用。地面控制人员或空间站中的宇航员可以在安全的环境中使用遥操作机器人，完成在轨航天器的捕获、维修和燃料加注等高风险任务^[4]。通过回传和反馈机器人的接触作用力，遥操作技术能够显著改善操作员对远程环境的感知能力，增强临场感，从而提升任务完成的效果。

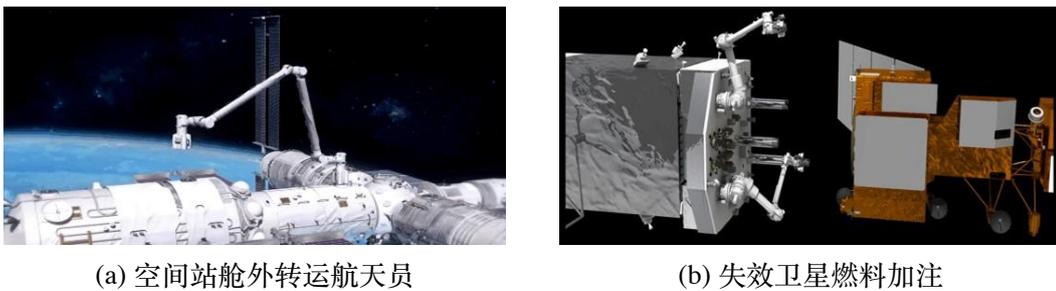


图 1-4 空间遥操作机器人的两种典型应用场景

然而，遥操作系统的一个关键挑战是通信延迟。当系统存在通信延迟时，力反馈遥操作系统的稳定性和安全性将无法得到有效保障。通信延迟会导致操作员对机器人动作的感知和实际执行之间产生时间差，从而影响任务的精确性和安全性。为了解决这一问题，研究人员正在开发多种技术，包括更快速的通信协议、先进的预测算法和本地自主控制功能，以提高系统的整体性能和可靠性。

总体来说，遥操作技术通过将人类的智慧和机器的力量有机结合，极大地扩展了机器人技术的应用范围。从工业制造到空间探索，从医疗手术到灾害救援，遥操作技术正在不断改变我们的生活和工作方式，并推动着科技进步的步伐。

基于视觉伺服的共享遥操作系统结合了机器人的局部自主控制（视觉伺服）和操作员的远程控制（遥操作），实现了人机智能的协同。这种系统利用视觉伺服技术，使机器人能够在局部自主完成一定任务，从而减轻操作员的负担。在共享控制方法下，操作员可以实时获取机器人的运动状态和目标物体的位置信息，这使得操作员能够对机器人进行更加精

确的控制，从而提高任务的成功率和效率。

通过共享控制方式，机器人不仅可以在复杂和动态的作业环境中进行高精度的运动，还能在极限条件下保持高效率的工作状态。这种协同工作模式不仅提升了机器人在各种复杂环境中的适应能力，还扩展了其在不同行业和应用场景中的潜力。操作员和机器人之间的这种互动和协作，使得机器人能够在更加多样化的任务中发挥重要作用，进一步推动了机器人技术在工业、医疗、空间探索等领域的应用和发展。

此外，基于视觉伺服的共享操作系统还具有高度的灵活性和扩展性。操作员可以根据任务需求调整控制策略，使得机器人能够更好地适应不同的工作环境和任务要求。随着技术的不断进步，这种共享控制模式将会变得更加智能和高效，为未来的自动化和智能化发展提供重要支持。总的来说，基于视觉伺服的共享操作系统，通过人机协作的方式，显著提升了机器人的作业能力和应用范围，推动了各个领域的技术革新和进步。

1.2 研究现状

基于视觉伺服的共享操作系统是一个新兴的技术领域，目前国内外对此研究尚处于发展阶段。

在国外，一些知名大学和科研机构已经开展了相关研究。有关视觉伺服的研究中，法国国家信息与自动化研究所的 Francois Chaumette 团队提出了一种基于光度高斯混合模型的稠密图像特征^[5]，Eric Marchand 团队提出了一系列基于主成分分析^[6]、离散余弦变换^[7]和卷积自动编码器^[8]的紧凑图像表征方法。美国麻省理工学院的研究团队设计了一种基于深度学习的视觉伺服算法^[9]，能够在复杂环境中实现高效的目标跟踪和操作。然而，该算法对计算资源要求较高，难以在资源受限的嵌入式系统中应用。此外，深度学习模型的鲁棒性在某些极端情况下仍需进一步提升。斯坦福大学提出了一种结合视觉伺服和触觉反馈的遥操作系统^[10]，能够在未知环境中进行自主导航和操作。然而，该系统在多传感器融合方面仍有改进空间，尤其是传感器数据的实时性和准确性对系统性能有较大影响。有关的遥操作的研究中，意大利理工学院的研究团队提出了一种基于无源性的遥阻抗控制框架^[11]，能够在遥操作过程中实时测量操作员的阻抗参数并据此调整机器人的柔顺性。然而，该框架在处理复杂操作环境时的灵活性仍需提升。法国国家科学研究中心开发了一种基于无源性的共享控制策略，通过自主控制托盘方位来辅助操作员平稳传送易滑动物体^[12]。尽管该策略在实验中表现出色，但在实际操作中的可靠性和适应性仍有待进一步验证。加拿大西

安大略大学的研究人员提出了一种基于无源性的反馈力预测方法^[13]，通过频域特征预测机器人的接触作用力，从而降低通信延时对系统稳定性的影响。然而，这种方法在高频率操作下的预测精度还有待提高。美国纽约大学的研究团队设计了一种基于无源性的变结构控制方法^[14]，利用操作员无源性的估计值降低了机器人控制设计的保守性。尽管该方法提高了控制灵活性，但在多变环境中的适应性仍需优化。斯坦福大学提出了一种基于无源性的双代理控制框架，利用力空间中的粒子滤波器实时估计远地环境的几何特征，从而增强机器人的感知能力^[15]。然而，该框架在处理动态变化环境中的实时性和准确性尚需进一步验证。德国宇航中心和欧洲空间局的研究团队通过天地遥操作实验证实，基于无源性的力反馈遥操作控制方法能够有效应对 800 毫秒的通信延时并完成星表采样任务^[16]。尽管取得了显著进展，但在更复杂和多变的空间环境中，系统的鲁棒性和稳定性仍需进一步提升。

(a) 光度高斯混合^[5](b) 离散余弦变换^[7](c) 离散正交矩^[17]

图 1-5 基于图像的自主机器人视觉伺服控制

在国内，一些高校和研究机构也积极开展基于视觉伺服的遥操作机器人控制方法的研究。例如，中山大学的研究团队利用线性参数化和组合学习技术为基于单应性的视觉伺服控制设计了一种图像深度值估计方法^[18]。线性参数化和组合学习技术的应用可能导致模型复杂性增加，对计算资源和时间的需求较高。南方科技大学研究团队提出了一种基于离散正交矩的图像表征策略^[17]，能够增强直接视觉伺服控制方法对冗余数据的鲁棒性。离散正交矩的表征策略在不同类型的图像和不同应用场景中的适应性和效果需要进一步验证，可能存在适应性不足的问题。

总的来说，视觉伺服技术为自动化系统提供了一种强大的感知和控制手段，可以实现对目标物体的高效跟踪和精确控制。基于视觉伺服的共享遥操作控制系统是一个具有广泛应用前景和技术挑战性的研究领域，吸引了全球众多学术机构和研究人员的关注。

这种技术在医疗、制造、航天等多个领域都展现出巨大的潜力。例如，在医疗领域，基于视觉伺服的远程手术系统可以让医生在远程进行高精度的手术操作，从而提高医疗服务的可及性和精度。在制造业中，视觉伺服技术可以用于自动化生产线，实现对生产过程的精确控制和质量监测。而在航天领域，视觉伺服技术可以应用于太空遥操作任务^[19]，帮助完成精细的组装和维护工作。

然而，当前的研究仍存在一些显著的不足之处，需要进一步的改进和优化。首先，实验环境的局限性使得许多技术难以在实际复杂的应用场景中得到验证。例如，许多实验是在控制良好的实验室环境中进行的，这与实际应用中的复杂和动态环境存在较大差异。这种局限性可能导致实验结果与实际应用效果之间存在差距，影响技术的实际推广和应用。

其次，系统的实时性问题也是一个重要挑战。许多基于视觉伺服的系统在处理复杂任务时，响应速度较慢，难以满足实时操作的要求。例如，在医疗手术或工业机器人操作中，任何微小的延迟都可能导致严重的后果。因此，提高系统的实时性和响应速度，是未来研究的一个关键方向。

此外，计算资源的需求也是一个重要的限制因素。许多先进的视觉伺服算法依赖于深度学习和其他计算密集型技术，这需要大量的计算资源。然而，在许多实际应用中，如嵌入式系统和资源受限的环境中，提供足够的计算资源是一个挑战。如何在有限的计算资源下实现高效的视觉伺服控制，是一个亟待解决的问题。

最后，多传感器融合带来的挑战也不容忽视。为了提高系统的精度和鲁棒性，许多视觉伺服系统需要融合来自多个传感器的数据。然而，不同传感器的数据往往具有不同的特性和噪声水平，如何有效地融合这些数据，确保系统的实时性和准确性，是一个复杂的问题。此外，多传感器融合还可能带来数据处理和传输的额外负担，进一步增加系统的复杂性。

综上所述，基于视觉伺服的共享遥操作控制系统在多个领域展现出广泛的应用前景，但也面临着诸多技术挑战。未来的研究需要在多个方面进行改进和优化，包括实验环境的多样化和真实性、系统实时性的提高、计算资源需求的优化以及多传感器融合的有效实现。通过这些努力，基于视觉伺服的共享遥操作控制系统有望在更多实际应用中得到广泛推广，推动自动化系统向更加智能和高效的方向发展。

1.3 本文研究内容

本文旨在设计并实现一种基于视觉伺服的共享遥操作系统，通过理论分析和实验验证相结合的方法，研究并优化视觉伺服算法和共享控制策略，以提高系统的实时性和精度，并验证其在实际应用中的有效性和可靠性。为了实现上述目标，本文将利用计算机视觉技术和机器人技术，设计并实现共享控制策略并通过模拟和实物实验验证系统的性能。全文的结构安排如下：

第一章绪论：介绍视觉伺服和遥操作系统的基本概念，以及在现代工业和机器人中的应用。强调共享遥操作系统在实际应用中的重要性，如在危险环境下操作机器人。简要介绍国内外在这一领域的研究进展，指出现有研究的不足之处。最后简要介绍本文的研究内容。

第二章理论基础：解释视觉伺服的基本原理，包括位置基视觉伺服（PBVS）和图像基视觉伺服（IBVS）。详细介绍遥操作系统的组成、工作原理和分类。讨论共享控制的概念及其在遥操作系统中的应用。

第三章系统设计：描述共享遥操作系统的总体架构，包括硬件和软件部分。详细介绍所采用的视觉伺服算法，说明共享控制策略的设计和实现，包括操作员和自动系统之间的任务分配。

第四章实验与验证：描述实验设备和环境，包括 ROS、CoppeliaSim、UR3e 机器人和 RS D435 深度相机等。详细说明实验过程，如系统的初始化、图像处理 and 速度控制等。展示实验数据和结果，并用图表进行分析。对实验结果进行讨论，指出系统的优点和不足，并提出改进建议。

第五章总结与展望：总结本文的主要工作和成果。提出未来的研究方向，如系统的优化和在其他应用场景中的推广。

第二章 理论基础

2.1 成像几何

2.1.1 引言

一张数字图像可以看作是一个二维矩阵，其中每个元素称为像素（pixel）。在本节中，我们用图像（Image）表示一个维度为 $N_{\text{rows}} \times N_{\text{cols}}$ 的矩阵。图像的形成是因为光线聚焦到一个二维的感光元件阵列上，每个像素的数值对应于入射到特定感光元件上的光强度。使用焦距为 λ 的透镜将光聚焦到感光阵列上，该阵列通常由 CCD（电荷耦合器件）传感器组成。透镜和感光阵列一同封装在相机中，然后与数字化转换器（digitizer）或图像采集卡（frame grabber）连接。在模拟相机中，数字化转换器将模拟视频信号转换为离散值，并通过图像采集卡传输到像素阵列。在数码相机中，图像采集卡仅传输相机中的数字信号到像素阵列。

在机器人应用中，仅考虑几何因素通常就能满足成像需求。因此，在本节中，我们只介绍成像过程中的几何知识，不讨论光度测量（photometric）如景深、镜头型号或辐射测量（radiometry）等方面的问题。

2.1.2 相机坐标系

首先，我们给成像系统分配一个坐标系。接着，讨论针孔模型的图像形成过程，并推导出一个公式，将一个点的世界坐标与它的图像坐标联系起来。为了简化本章中的许多公式，将物体坐标表示为相对于相机中心的坐标系是非常有用的。

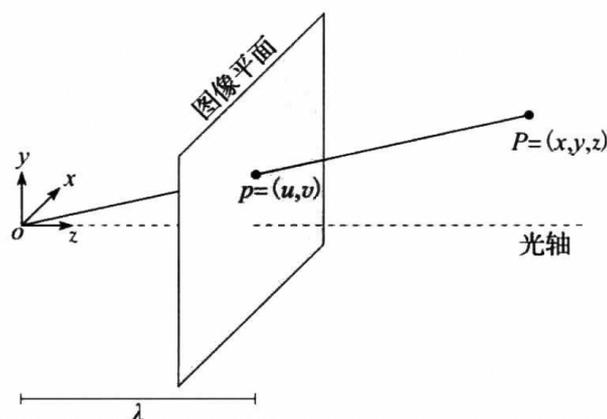


图 2-1 相机坐标系

为此，定义相机坐标系如下。我们将包含检测阵列的平面定义为图像平面。坐标轴 x_c

和 y_c 构成图像平面的坐标基，通常分别平行于图像的水平轴和垂直轴。坐标轴 z_c 垂直于图像平面，并且与镜头的光轴对齐，即它通过镜头的焦点。相机坐标系的原点位于图像平面后方距离为 λ 的位置，该点也称为投影中心（center of projection）。光轴与图像平面的交点称为主点（principal point）。这个坐标系如图2-1所示。

在上述坐标系中，图像平面内的任意一点都可以用坐标 (u, v, λ) 表示。因此，我们可以用 (u, v) 来参数化图像平面，并且用 (u, v) 来指代图像平面坐标。

2.1.3 透视投影

图像形成过程通常通过针孔镜头模型来建模描述。在该近似方法中，镜头被看作是处于镜头焦点处的一个理想针孔。光线穿过该针孔并与图像平面相交。

令 P 表示相对于相机参考系的坐标为 (x, y, z) 的一个点。令 p 表示点 P 在图像平面中的投影，其坐标为 (u, v, λ) 。在针孔模型假设下，点 P 、 p 以及相机参考系的原点共线，如图2-1所示。因此，对于未知的正值常数 k ，我们有

$$k \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} u \\ v \\ \lambda \end{bmatrix} \quad (2.1)$$

它可被写成下列方程组

$$\begin{cases} kx = u \\ ky = v \\ kz = \lambda \end{cases} \quad (2.2)$$

这给出 $k = \lambda/z$ ，该关系可被代入到公式 (2.2) 中得出

$$u = \lambda \frac{x}{z}, \quad v = \lambda \frac{y}{z} \quad (2.3)$$

这些是众所周知的透视投影（perspective projection）方程。

2.2 视觉伺服控制理论基础

2.2.1 引言

视觉伺服控制指的是使用计算机视觉数据来控制机器人的运动。视觉数据可以从直接安装在机器人操作臂或移动机器人上的摄像机获取，这种情况下机器人的运动会引起摄像机的运动，或者摄像机可以固定在工作空间中，以便它可以从一个固定的配置观察机器人的运动。本文中我们主要关注前一种，即所谓的手眼情况。

2.2.2 视觉伺服的基本组成

所有基于视觉的控制方案的目标都是最小化一个误差 $\mathbf{e}(t)$ ^[20]，通常定义为

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^* \quad (2.4)$$

公式中的参数定义如下：向量 $\mathbf{m}(t)$ 是一组图像测量值（例如，特征点的图像坐标或对象质心的图像坐标）。这些图像测量值用于计算一个由 k 个视觉特征组成的向量 $\mathbf{s}(\mathbf{m}(t), \mathbf{a})$ ，其中 \mathbf{a} 是一组表示关于系统可能的额外知识的参数（例如，粗略的摄像机内部参数或对象的 3-D 模型）。向量 \mathbf{s}^* 包含特征的期望值^[21]。

在本文中，我们考虑固定目标姿态的情况（和目标静止的情况）。即 \mathbf{s}^* 是常数， \mathbf{s} 的变化仅依赖于摄像机运动。此外，我们在这里考虑控制具有六个自由度（6 DOF）的摄像机的运动；例如，一个附加到六自由度臂末端的摄像机。

视觉伺服方案主要在于 \mathbf{s} 的设计方式不同。在本文中，我们将看到两种非常不同的方法。首先，我们描述基于图像的视觉伺服控制（IBVS），在这种方法中， \mathbf{s} 由一组直接在图像数据中可用的特征组成。然后，我们描述基于位置的视觉伺服控制（PBVS），在这种方法中， \mathbf{s} 由一组必须从图像测量值估计出的 3-D 参数组成^[22]。

一旦选择了 \mathbf{s} ，控制方案的设计可以相当简单，也许最直接的方法是设计一个速度控制器。为此，我们需要 \mathbf{s} 的时间变化与摄像机速度之间的关系。设摄像机的空间速度由 $\mathbf{v}_c = (v_c, \omega_c)$ 表示，其中 v_c 是摄像机框架原点的瞬时线速度， ω_c 是摄像机框架的瞬时角速度。 $\dot{\mathbf{s}}$ 和 v_c 之间的关系由下式给出

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c, \quad (2.5)$$

其中 $\mathbf{L}_s \in \mathbb{R}^{k \times 6}$ 称为与 \mathbf{s} 相关的交互矩阵，在视觉伺服文献中，特征雅可比一词在某

种程度上也可以与之互换使用。

由公式(2.4)和(2.5)，我们可以得到摄像机速度和误差时间变化之间的关系：

$$\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{v}_c, \quad (2.6)$$

其中 $\mathbf{L}_e = \mathbf{L}_s$ 。考虑 \mathbf{v}_c 作为机器人控制器的输入，如果我们想要尝试确保误差的指数解耦递减（即， $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ ），我们使用(2.6)获得：

$$\mathbf{v}_c = -\lambda \mathbf{L}_e^+ \mathbf{e}, \quad (2.7)$$

其中 $\mathbf{L}_e^+ \in \mathbb{R}^{6 \times k}$ 被选择为 \mathbf{L}_e 的 Moore-Penrose 伪逆，即当 \mathbf{L}_e 具有满秩 6 时 $\mathbf{L}_e^+ = (\mathbf{L}_e^T \mathbf{L}_e)^{-1} \mathbf{L}_e^T$ 。这个选择允许 $\|\dot{\mathbf{e}} - \lambda \mathbf{L}_e \mathbf{L}_e^+ \mathbf{e}\|$ 和 $\|\mathbf{v}_c\|$ 是最小的。当 $k = 6$ 时，如果 $\det(\mathbf{L}_e) \neq 0$ ，可以直接求逆 \mathbf{L}_e ，得到控制 $\mathbf{v}_c = -\lambda \mathbf{L}_e^{-1} \mathbf{e}$ 。

在实际的视觉伺服系统中，不可能准确地知道 \mathbf{L}_e 或 \mathbf{L}_e^+ 。因此，必须实现对这两个矩阵之一的近似或估计。在后续中，我们用符号 $\widehat{\mathbf{L}_e^+}$ 表示交互矩阵近似的伪逆和近似交互矩阵的伪逆。使用这个符号，控制律实际上是：

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_e^+} \mathbf{e}. \quad (2.8)$$

这是大多数视觉伺服控制器实现的基本设计。

2.2.3 基于图像的视觉伺服控制（IBVS）

传统的基于图像的控制方案使用一组点的图像平面坐标来定义集合 \mathbf{s} 。图像测量值 \mathbf{m} 通常是图像点集的像素坐标（但这不是唯一可能的选择），参数 \mathbf{a} 在公式(2.4)中定义 $\mathbf{s}(\mathbf{m}, \mathbf{a})$ 中仅仅是摄像机内部参数，用于从以像素为单位的图像测量值转换到特征。

更具体地说，对于在摄像机框架中坐标为 $\mathbf{X} = (X, Y, Z)$ 的 3-D 点，它在图像中投影为 2-D 点，坐标为 $\mathbf{x} = (x, y)$ ，我们有：

$$\begin{cases} x = X/Z = (u - c_u)/f\alpha \\ y = Y/Z = (v - c_v)/f, \end{cases} \quad (2.9)$$

这里 $\mathbf{m} = (u, v)$ 给出了以像素单位表示的图像点坐标， $\mathbf{a} = (c_u, c_v, f, \alpha)$ 是摄像机内部参数集： c_u 和 c_v 是主点坐标， f 是焦距， α 是像素尺寸比例。在这种情况下，我们取

$\mathbf{s} = \mathbf{x} = (x, y)$ ，即点的图像平面坐标。关于成像几何和透视投影的细节可以在许多计算机视觉文本中找到。

取投影方程(2.9)的时间导数，我们得到：

$$\begin{cases} \dot{x} = \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} = \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z. \end{cases} \quad (2.10)$$

我们可以将 3-D 点的速度与摄像机空间速度联系起来，使用以下众所周知的方程：

$$\dot{\mathbf{X}} = -\mathbf{v}_c - \boldsymbol{\omega}_c \times \mathbf{X} \Leftrightarrow \begin{cases} \dot{X} = -v_x - \omega_y Z + \omega_z Y \\ \dot{Y} = -v_y - \omega_z X + \omega_x Z \\ \dot{Z} = -v_z - \omega_x Y + \omega_y X. \end{cases} \quad (2.11)$$

将公式(2.11)代入公式(2.10)，整理后我们得到：

$$\begin{cases} \dot{x} = -\frac{v_x}{Z} + x\frac{v_z}{Z} + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ \dot{y} = -\frac{v_y}{Z} + y\frac{v_z}{Z} + (1+y^2)\omega_x - xy\omega_y - x\omega_z \end{cases} \quad (2.12)$$

这可以写为：

$$\dot{\mathbf{x}} = \mathbf{L}_x \mathbf{v}_c, \quad (2.13)$$

其中与 \mathbf{x} 相关的交互矩阵 \mathbf{L}_x 为：

$$\mathbf{L}_x = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix}. \quad (2.14)$$

在矩阵 \mathbf{L}_x 中， Z 是点相对于摄像机框架的深度。因此，任何使用这种形式的交互矩阵的控制方案都必须估计或近似 Z 的值。同样，摄像机内部参数涉及到 x 和 y 的计算。因此， \mathbf{L}_x 不能直接用于控制，必须使用估计或近似的 $\hat{\mathbf{L}}_x$ 。

为了控制系统的 6 个自由度，至少需要三个点（即，我们需要 $k \geq 6$ ）。如果我们使用特征向量 $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ ，通过简单地堆叠三个点的交互矩阵，我们得到：

$$\mathbf{L}_x = \begin{bmatrix} \mathbf{L}_{x1} \\ \mathbf{L}_{x2} \\ \mathbf{L}_{x3} \end{bmatrix}.$$

在这种情况下，存在一些配置使得 \mathbf{L}_x 奇异。此外，存在四个不同的摄像机姿态使得 $\mathbf{e} = 0$ ，即存在四个全局最小值，我们无法区分它们。因此，通常考虑超过三个点。

有几种选择可用于构建控制律中使用的估计交互矩阵 $\widehat{\mathbf{L}}_e^+$ 。一种流行的方案是，如果 $\mathbf{L}_e = \mathbf{L}_x$ 已知，就选择 $\widehat{\mathbf{L}}_e^+ = \mathbf{L}_e^+$ ；也就是说，如果每个点的当前深度 Z 是可用的。在实践中，这些参数必须在控制方案的每次迭代中被估计。本文介绍的基本方法使用经典的姿态估计方法，这些方法将在下一节中简要介绍。另一种流行的方法是选择 $\widehat{\mathbf{L}}_e^+ = \mathbf{L}_{e^*}^+$ ，其中 \mathbf{L}_{e^*} 是在期望位置 $\mathbf{e} = \mathbf{e}^* = 0$ 时 \mathbf{L}_e 的值。在这种情况下， $\widehat{\mathbf{L}}_e^+$ 是常数，只需要设置每个点的期望深度，这意味着在视觉伺服过程中不需要估计变化的 3-D 参数。最后，选择 $\widehat{\mathbf{L}}_e^+ = 1/2(\mathbf{L}_e + \mathbf{L}_{e^*}^+)$ 最近在文献中被提出。由于 \mathbf{L}_e 在这种方法中被涉及，也必须估计每个点的当前深度。

2.2.4 基于位置的视觉伺服控制（PBVS）

基于位置的视觉伺服控制方案使用相机相对于某个参考坐标系的姿态来定义 \mathbf{s} 。从一组图像测量中计算该姿态需要知道相机的内部参数和被观察物体的三维模型。这个经典的计算机视觉问题被称为三维定位问题。

接下来，通常会根据用于表示相机姿态的参数化方法来定义 \mathbf{s} 。注意，涉及到定义(2.4)中 \mathbf{s} 的参数 \mathbf{a} 现在是相机的内部参数和物体的三维模型。

考虑三个坐标系：当前相机坐标系 F_c ，期望相机坐标系 F_c^* ，以及附着在物体上的参考坐标系 F_o 。这里我们采用标准符号，使用前导上标来表示一组坐标是相对于哪个坐标系定义的。因此，向量 ${}^c\mathbf{t}_o$ 和 ${}^{c^*}\mathbf{t}_o$ 分别给出了相对于当前相机坐标系和期望相机坐标系的物体原点的坐标。此外，让 $\mathbf{R} = {}^{c^*}\mathbf{R}_c$ 表示当前相机坐标系相对于期望坐标系的旋转矩阵。

我们可以将 \mathbf{s} 定义为 $(\mathbf{t}, \theta\mathbf{u})$ ，其中 \mathbf{t} 是一个平移向量， $\theta\mathbf{u}$ 给出了旋转的角度/坐标轴表示。现在讨论两种选择 \mathbf{t} 的方法，并给出相应的控制律。

如果 \mathbf{t} 是相对于物体坐标系 F_o 定义的，我们得到 $\mathbf{s} = ({}^c\mathbf{t}_o, \theta\mathbf{u})$ ， $\mathbf{s}^* = ({}^{c^*}\mathbf{t}_o, 0)$ ，以及 $\mathbf{e} = ({}^c\mathbf{t}_o - {}^{c^*}\mathbf{t}_o, \theta\mathbf{u})$ 。在这种情况下，与 \mathbf{e} 相关的交互矩阵由下式给出：

$$\mathbf{L}_e = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_o]_{\times} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}} \end{bmatrix} \quad (2.15)$$

其中 \mathbf{I}_3 是 3×3 的单位矩阵， $\mathbf{L}_{\theta\mathbf{u}}$ 由下式给出：

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_{\times} + \left(1 - \frac{\sin c\theta}{\sin c^2\theta}\right)[\mathbf{u}]_{\times}^2 \quad (2.16)$$

根据本章开头的讨论，我们得到控制方案：

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_e^{-1}} \mathbf{e},$$

由于 \mathbf{s} 的维数 k 为 6，这对应于相机的自由度数。通过设置：

$$\widehat{\mathbf{L}_e^{-1}} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_o]_{\times} L_{\theta\mathbf{u}}^{-1} \\ \mathbf{0} & L_{\theta\mathbf{u}}^{-1} \end{bmatrix}, \quad (2.17)$$

我们经过简单的推导得到：

$$\begin{cases} \mathbf{v}_c = -\lambda(({}^{c^*}\mathbf{t}_o - {}^c\mathbf{t}_o) + [{}^c\mathbf{t}_o]_{\times}\theta\mathbf{u}), \\ \omega_c = -\lambda\theta\mathbf{u}, \end{cases} \quad (2.18)$$

因为 $L_{\theta\mathbf{u}}$ 是这样的，即 $L_{\theta\mathbf{u}}^{-1}\theta\mathbf{u} = \theta\mathbf{u}$ 。

这个 PBVS 方案导致旋转运动沿着具有指数递减速度的测地线，使得涉及到 s 中的平移参数以相同的速度减小。这解释了相机速度分量的指数递减。

可以通过使用 $s = ({}^{c^*}\mathbf{t}_c, \theta\mathbf{u})$ 设计另一个 PBVS 方案。在这种情况下，我们有 $s^* = 0$ ， $e = s$ ，以及

$$L_e = \begin{bmatrix} R & 0 \\ 0 & L_{\theta\mathbf{u}} \end{bmatrix}. \quad (2.19)$$

注意平移和旋转运动之间的解耦，这允许我们获得一个简单的控制方案：

$$\begin{cases} v_c = -\lambda R^T({}^{c^*}\mathbf{t}_c), \\ \omega_c = -\lambda\theta\mathbf{u}. \end{cases} \quad (2.20)$$

在这种情况下，如果涉及到(2.20)的姿态参数被完美估计，相机轨迹应该是一条纯直线，而图像轨迹不如之前那样令人满意。甚至可以找到一些特殊配置，使得一些点离开相机视野。

2.3 遥操作控制原理

2.3.1 引言

遥操作（Teleoperation）是一种允许操作人员通过远程控制系统操作设备或机械装置，实现对远程目标的操控的技术，为在一些特殊的环境，如空间站舱外，火灾现场中的操作提高了便利性和安全性。

2.3.2 遥操作系统的基本组成和工作原理

遥操作系统通常由以下几个主要部分组成：主端控制器，包括操作者接口，操作人员使用主端控制器进行操作，它可以是操纵杆、控制台、手控器等，用于输入操作指令。主端控制器负责接收操作人员的指令并将其转换为电信号，通过通信回路传输到从执行端。通信回路，控制指令和反馈信息通过通信网络在主控制端和从执行端之间传输，通信网络可以是有线的（如以太网）或无线的（如 Wi-Fi、卫星通信）。从端控制器，根据接收到的控制指令操作执行机构，通常是一个机器人或机械臂，用于执行具体任务。反馈系统负责将从端执行器的状态信息和操作环境信息传输回主端控制器，使操作人员能够实时监控和调整操作。反馈信息可以包括视觉反馈（如摄像头）、力反馈（如力传感器）^[23]等。

遥操作的工作原理包括以下几个步骤：首先，操作人员通过主端控制器输入操作指令，例如移动手控器。接着，这些指令通过通信网络传输到从端控制器。从端控制器接收到控制指令后，执行相应的操作，例如移动机械臂或启动设备。与此同时，传感器系统会采集从执行端的状态信息和操作环境信息，例如设备位置、速度、外界力等。采集到的信息通过通信网络传输回主控制端，主控制端接收反馈信息，并在操作界面上显示，使操作人员能够实时监控和调整操作。

2.3.3 基于无源性的控制

远程操作控制设计的一个主要挑战是人类和远程环境的动态特性在很大程度上是未知的，它们都是全局控制回路的一部分。基于无源性的控制方法解决了这些挑战，此外还促进了模块化设计方法，可以从子系统特性推导出整体稳定性证明。

触觉闭环控制中的人类动态只能近似估计。这样的近似模型可以分为非参数模型和参

数模型，根据非参数模型，受过训练的人可以被认为是与无源环境被动交互的，参数模型也是如此。在远程操作的背景下，基于无源性的非参数模型由于其稳定性、鲁棒性以及能够将机器人操作器（HSI、远程操作器）的非线性子系统动态包括在稳定性分析中的能力，取得了最成功的应用。HSI 和远程操作器是无源的，或者可以通过适当的局部控制使其无源，而大多数环境也可以被认为是无源的。

无源性是一种基于能量的概念，通过系统的输入/输出行为来表征的动态系统^[24]。它提供了充分的输入/输出稳定性条件，并且假设零状态为可检测的，同时它还提供了原点的稳定性。我们考虑的系统是描述为连续时间、非线性时间不变的动态系统：

$$\Sigma \begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases} \quad (2.21)$$

其中 $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}^p$ 分别是状态、输入和输出向量，并且 $f(0, 0) = h(0, 0) = 0$ 。函数 f 假设局部 Lipschitz 连续，和 h 连续。因此，对于每个固定的初始状态 $x(0)$ ，公式(2.21)定义了从输入信号 $u(\cdot)$ 到输出信号 $y(\cdot)$ 的因果映射。

首先定义无源性：如果系统 Σ (2.21) 存在一个半正定函数 $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ ，称为储能函数，使得对于每一个可接受的 u 和每一个 $t \geq 0$

$$V(x(t)) - V(x(0)) \leq \int_0^t u^T y d\tau. \quad (2.22)$$

为了简化符号表示，未显式写出 u 和 y 随时间 τ 的变化关系。

无源性可以视为一种能量不生成条件：进入系统的能量，由输入向量 u 和输出向量 y 的标量积的右侧积分表示，要么被存储在内部状态 x 中，要么被耗散。注意，如果输入/输出变量没有选择为共置的力/流变量（例如，机械系统中的力和速度），输入能量不一定对应于任何实际的物理能量。

无源形式的一个重要特性是其闭合性。这意味着两个无源系统连接在一起，无论是反馈还是并联配置，组合系统仍然是无源的。这种无源性论点在典型的速度-力和力-速度远程操作架构中适用。

然而，对于存在时间延迟的通信，无源性论点不再有效。实际上，即使是小的时间延迟，也会使整个系统变得不稳定。

为了解决基于无源性的控制存在的问题，存在几种控制方案^[25]，包括：

- 散射变换控制：通过将通信通道模拟为无损传输线，确保系统的无源性^[26]。

- 阻尼注入控制：通过在系统中引入阻尼来稳定系统^[27]。
- 自适应控制：通过自适应算法估计系统的物理参数，以实现位置跟踪和延迟无关的稳定性^[28]。

2.3.4 共享控制

遥操作系统控制方案可以根据不同的分类方式进行分类^[29]，《Handbook of Robotics》中，遥操作的控制结构被分为监督控制、共享控制和直接控制^[30]。划分层次如表2.1所示：

表 2.1 遥操作控制方案划分

从端自主性	控制方案	控制模式
高	监督控制	\
中	共享控制	\
低	直接控制	单向控制/双向控制

在监督控制模式下，机械臂主要依赖自身的自主性来完成任任务^[31]，操作者仅在突发事件和复杂规划时进行实时监控和干预。而在直接控制模式中，操作者被直接纳入控制回路，这是目前遥操作研究中较为常见的一种模式。在这种模式下，机械臂的自主性通常不被考虑，操作者会将控制指令直接发送给机械臂。根据机械臂在执行指令后是否会将操作信息反馈给操作者，控制方法可以分为单向控制和双向控制两种类型。共享控制模式介于自主控制和直接控制之间，既能够向操作者提供直接控制的临场感，又能利用机械臂的自主性确保操作的安全性^[32]。当操作有延迟时，机械臂需要自主处理一部分任务，以弥补人类操作者在实时规划中的不足。

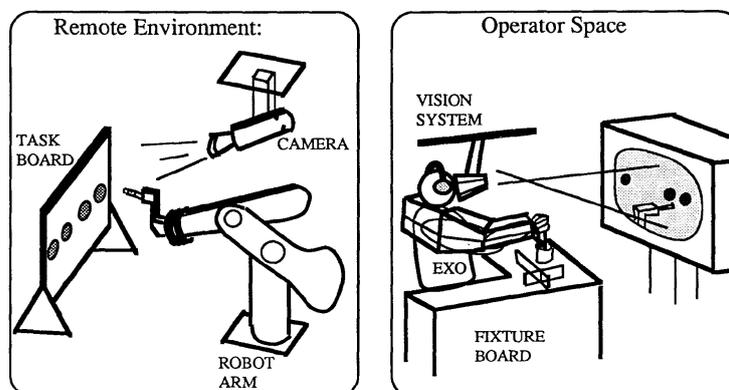


图 2-2 共享控制模式^[33]

共享控制模式的主要特点包括混合控制、增强的安全性和临场感反馈。混合控制将操作者的高层次指令与自动系统的低层次操作相结合，操作者提供目标和方向，自动系统处理细节操作和执行^[34]。增强的安全性体现在自动系统能够利用传感器数据和内置算法，实时监控和调整操作，确保安全性，并在发生意外或错误时及时干预，防止事故发生。临场感反馈则允许操作者通过视觉、触觉等多种反馈方式，实时感知操作环境和设备状态，操作者可以在必要时进行干预和调整，从而提高操作的精确性和效率。

2.4 本章小结

本章深入探讨了成像几何、视觉伺服控制理论及遥操作控制原理，为后续研究奠定了坚实的理论基础。

成像几何部分介绍了数字图像的基本概念及其形成过程，详细解析了相机坐标系的定义和针孔模型的图像形成过程。透视投影方程的推导，阐明如何将三维世界的点映射到二维图像平面上，揭示了图像生成背后的几何关系。

在视觉伺服控制理论部分探讨了视觉伺服控制的基本组成和两种主要方法：基于图像的视觉伺服控制（IBVS）和基于位置的视觉伺服控制（PBVS）。通过对误差最小化目标的解析，展示了如何利用视觉数据来实现对机器人的精确控制。交互矩阵的分析和控制律的设计，解释了视觉伺服控制的实现原理和关键技术。

在遥操作控制原理部分详细介绍了遥操作系统的基本组成和工作原理，包括主端控制器、通信回路、从端控制器和反馈系统的功能和作用。对基于无源性控制和共享控制的讨论，揭示了在远程操作环境下，如何通过合理的控制策略来实现高效、安全的操作。基于无源性的控制方法强调了能量不生成的稳定性原则，而共享控制模式则结合了操作者的高层次指令和自动系统的低层次执行，提供了更为灵活和安全的控制方案。

第三章 系统设计

3.1 引言

本章主要介绍基于视觉伺服的共享遥操作系统的的设计过程，涵盖硬件设计、软件设计、机械臂运动学建模、控制算法及控制策略的实现。该系统希望实现通过视觉伺服和手控器直接控制，使机械臂定位二维码标记的指定区域并完成按压。要求整个系统运行稳定，定位精准，响应快速。

3.2 系统架构

3.2.1 硬件设计

系统硬件部分包括主端控制器，从端执行器，反馈系统和通信回路。主端控制器采用 3DSystems Haptic Touch 手控器以及计算机作为操作者接口，视觉伺服控制时，程序传输末端移动速度给计算机，直接控制时，操作者移动手控器末端，手控器传输末端位姿信息给计算机，计算机接收到信息，计算关节速度，发送给从端执行器；从端执行器由一台 UR3e 机械臂以及组装在其末端执行器上的一台 Intel RealSense D435 深度相机组成，使用关节速度控制器，通过接受主端控制器传来的关节速度信息，控制机械臂的运动。反馈信息包括通过深度相机采集的视觉反馈信息，机械臂末端力传感器采集的力和力矩信息，机械臂关节传感器采集的关节速度和位置信息等。UR3e 与计算机之间使用以太网接口有线连接，手控器与计算机之间通过 USB 接口有线连接。硬件组件之间均采用有线连接，可以有效降低组件之间数据传输的延时。系统的硬件框图如图3-1所示：

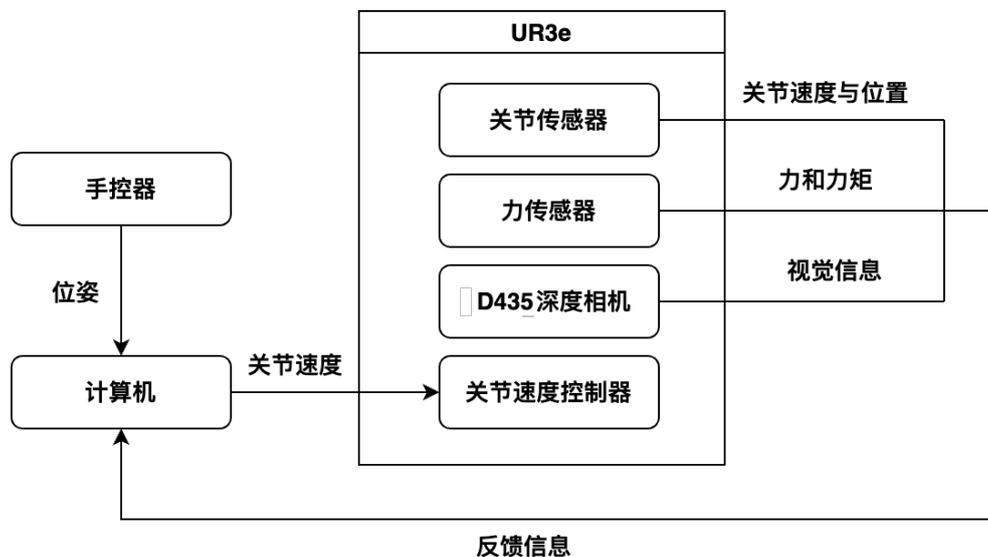


图 3-1 硬件框图

3.2.2 软件设计

系统软件部分包括初始化模块，数据处理与传输模块，视觉伺服模块，直接控制模块，共享控制模块以及手控器和机械臂的驱动程序。整个系统使用 ROS（Robot Operate System），采用 C++ 语言编写软件包，进行数据处理以及通过话题通信实现多机数据传输。

初始化模块使用 UR3e 示教器，通过设置关节角度，初始化机械臂位置；数据处理与传输模块订阅机械臂驱动程序发布的关节信息，建立机械臂正运动学以及速度运动学模型，计算得到机械臂转换矩阵，在基座坐标系下的当前位姿信息和雅各比矩阵等信息，并发布给视觉伺服模块和直接控制模块。视觉伺服模块使用视觉传感器获取的图像信息，得到特征点与目标的偏差，订阅数据处理与传输模块发布的转换矩阵和雅各比矩阵信息，计算末端执行器速度，通过订阅转化矩阵再转化为关节速度，发布给共享控制模块；直接控制模块通过手控器驱动，实时读取手控器末端位姿信息，同时订阅数据处理与传输模块发布的机械臂末端当前位姿信息，计算机械臂末端期望的位姿，订阅转换矩阵和雅各比矩阵信息，通过两者偏差计算末端运动速度，并转化为关节速度，发布给共享控制模块。共享控制模块订阅视觉伺服模块以及直接控制模块发布的关节速度信息，通过共享控制策略，选择需要使用的关节速度，并发布到机械臂驱动程序的关节速度控制器话题，实现机械臂的控制。

整个系统的软件框图如图3-2所示：

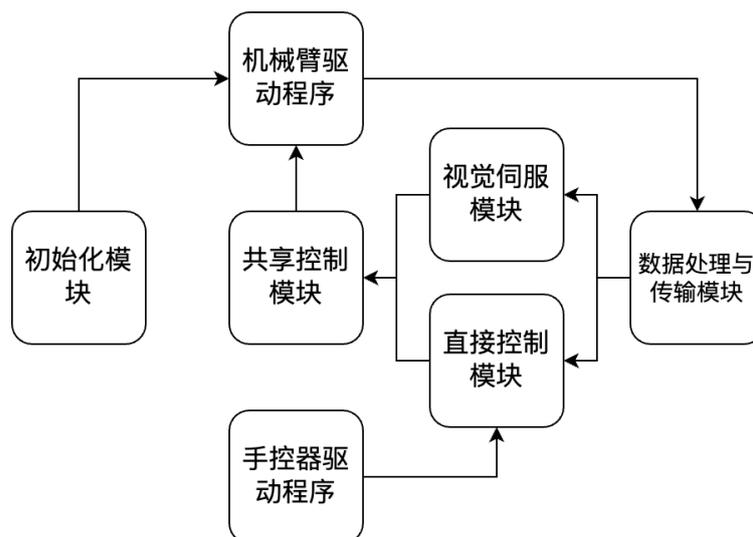


图 3-2 软件框图

3.3 机械臂建模

3.3.1 引言

本节主要介绍机械臂的正运动学建模和速度运动学建模过程，并在数据处理与传输模块中通过代码实现，将从机械臂驱动程序中订阅的关节位置信息转化为关节转换矩阵，末端位姿矩阵以及雅各比矩阵。

3.3.2 正运动学建模

从官网查阅可知，UR3e 机械臂 MDH (Modified Denavit Hartenberg) 参数如表3.1所示：

表 3.1 UR3e MDH 参数表

连杆序号 i	连杆长度 α_{i-1}/m	连杆扭角 a_{i-1}/rad	连杆距离 d_i/m	关节转角 θ_i/rad
1	0	0	0.15185	θ_1
2	0	$\pi/2$	0	θ_1
3	-0.24355	0	0	θ_1
4	-0.2132	0	0.13105	θ_1
5	0	$\pi/2$	0.08535	θ_1
6	0	$-\pi/2$	0.0921	θ_1

UR 机器人的 DH 参数被描述在示意图3-3中。

根据 MDH 方法建立的坐标系中的连杆关系，有：

$$T_i^{i-1} = Trans(x, a_{i-1})Rot(x, a_{i-1})Trans(z, d_i)Rot(z, \theta_i) \quad (3.1)$$

将公式(3.1)右边展开得到

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

式中 $i = 1, 2, 3, 4, 5, 6$ 。

根据表3.1中的 DH 参数以及公式(3.2)可得各连杆之间的变换矩阵 T_i^{i-1} ，将各变换矩阵相乘可得 UR3e 机械臂的变换矩阵 T_6^0 ：

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \quad (3.3)$$

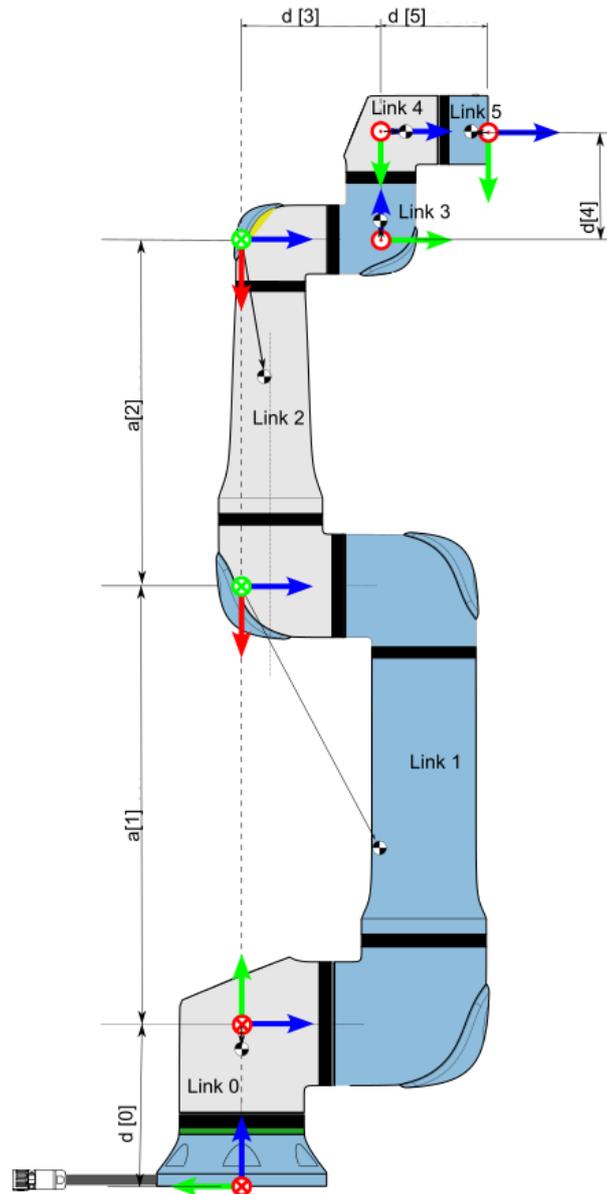


图 3-3 UR 机器人 DH 参数示意图

3.3.3 速度运动学建模

在正运动学建模中可知，我们只要知道机器人的结构以及每个关节的位置（平动关节的位移或旋转关节的角度），就可以求出末端执行器的位姿，因此，UR3e 末端执行器位姿和关节位置可用公式(3.4)表示：

$$\mathbf{x} = f(\vec{\theta}) \quad (3.4)$$

其中 \mathbf{x} 表示末端执行器的位姿， $\vec{\theta}$ 表示关节角度（UR3e 中仅有旋转关节）。对公式(3.4)两边同时对时间 t 求导可得：

$$\frac{d\mathbf{x}}{dt} = \frac{d\vec{\theta}}{dt} = \frac{df(\vec{\theta})}{d\vec{\theta}} \cdot \frac{d\vec{\theta}}{dt} \quad (3.5)$$

即：

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{d\theta} \cdot \dot{\theta} = \mathbf{J}(\vec{\theta}) \cdot \dot{\theta} \quad (3.6)$$

矩阵 $\mathbf{J}(\vec{\theta})$ 即为机械臂的雅各比矩阵，它将机器人关节空间的速度与操作空间的速度联系起来。由于 \mathbf{x} 可以表示在不同坐标系下，为了方便起见，定义基本雅各比矩阵，即使用笛卡尔坐标系描述线速度和角速度，以机械臂基座坐标系作为参考系来描述末端执行器速度所求得的雅各比矩阵。后文中出现的雅各比矩阵如无特殊说明，均为基本雅各比矩阵。

根据基本雅各比矩阵的定义，末端执行器的速度可以表示为：

$$\dot{\mathbf{x}} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \vec{v} \\ \vec{\omega} \end{bmatrix} \quad (3.7)$$

雅各比矩阵应为 6×6 矩阵，可以写成：

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_\omega \end{bmatrix} \quad (3.8)$$

矩阵上半部分对应线速度，下半部分对应角速度。由正运动学建模部分可以得到末端执行器位姿矩阵为：

$$T_6^0 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

其中 p_x 、 p_y 、 p_z 为末端执行器在基座坐标系下的位置，要求雅各比矩阵线速度部分 \mathbf{J}_v ，只需要将 p_x 、 p_y 、 p_z 分别对应关节角求导即可：

$$\mathbf{J}_v = \begin{bmatrix} \frac{dp_x}{d\theta_1} & \frac{dp_x}{d\theta_2} & \frac{dp_x}{d\theta_3} & \frac{dp_x}{d\theta_4} & \frac{dp_x}{d\theta_5} & \frac{dp_x}{d\theta_6} \\ \frac{dp_y}{d\theta_1} & \frac{dp_y}{d\theta_2} & \frac{dp_y}{d\theta_3} & \frac{dp_y}{d\theta_4} & \frac{dp_y}{d\theta_5} & \frac{dp_y}{d\theta_6} \\ \frac{dp_z}{d\theta_1} & \frac{dp_z}{d\theta_2} & \frac{dp_z}{d\theta_3} & \frac{dp_z}{d\theta_4} & \frac{dp_z}{d\theta_5} & \frac{dp_z}{d\theta_6} \end{bmatrix} \quad (3.10)$$

在三维空间中，角速度定义一个指向旋转轴的向量，在正运动学建模部分中，我们定义机械臂的每一个旋转关节均绕自身 z 轴旋转，所以当关节转动速度为 ω 时，在关节自身

坐标系下，角速度向量应为 $[0, 0, \omega]^T$ 。末端执行器的实际转速可以由每个关节的角速度线性叠加得到。

由于我们的雅各比矩阵是以基座坐标系作为参考系的，所以为了写出 \mathbf{J}_ω ，需要将每个关节在自身坐标系下的角速度向量转换到基座坐标系下，另外，由于平动关节运动不会改变末端执行器姿态，所以 p_x 、 p_y 、 p_z 不影响 \mathbf{J}_ω 的计算。因此，雅各比矩阵角速度部分 J_ω 为：

$$\mathbf{J}_\omega = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 & \mathbf{z}_4 & \mathbf{z}_5 & \mathbf{z}_6 \end{bmatrix} \quad (3.11)$$

由正运动学建模部分分析可知， \mathbf{z}_i 即为齐次变换矩阵 T_i^0 第三列前三行元素组成的 3×1 向量。

3.4 视觉伺服控制算法

3.4.1 引言

在视觉伺服模块中，涉及视觉伺服算法的选取，由第二章可知，视觉伺服主要分为基于位置的方法和基于图像的方法。本节主要介绍视觉伺服模块基于这两种方法基于 ViSP (Visual Servoing Platform)^[35] 库具体实现的过程。

3.4.2 图像特征识别

采用视觉伺服控制方法，首先需要在相机捕获的图像中检测到一组图像特征，即一组相关的可测量属性的图案定义^[36]。本系统中采用易于检测的二维码 (QR Code) 作为标记物，使用 ViSP 库 `vpDetectorQRCode` 类中的 `detect` 和 `getPolygon` 函数，可以检测到图像中的二维码，并按顺序读取二维码四个角点的图像坐标。随着相机捕获的图像以较高的频率不断更新，我们就得到了近乎实时的图像坐标。

3.4.3 基于位置的方法

在得到了四个特征点的实时图像坐标后，首先使用 ViSP 库 `vpPixelMeterConversion` 类中的 `convertPoint` 函数，使用相机内参将读取到的二维码角点图像坐标（以像素为单位）转化为图像平面中的二维坐标（以米为单位），然后使用 `vpPose` 类中的 `DMENTHON_LAGRANGE_VIRTUAL_VS` 方法估计二维码姿态，该方法首先运行 `Dmenthon` 和 `Lagrange` 线性姿态估计算法，然后保留具有最低残差的结果，并使用该结果初始化非线性虚拟视觉伺服算法来估计姿态。该算法具有比线性算法更低的残差。通过以上方法，我们得到了二维码在相机坐标系下的实时位姿，由于相机直接装配在末端执行器上，我们配置

相机坐标系与末端坐标系相同，我们就得到了二维码在末端坐标系下的实时位姿。

通过订阅数据处理与传输模块得到的转化矩阵和当前末端位姿信息，将二维码位姿转换到基坐标系下，并与当前位姿做差，乘以一定增益，得到末端运动速度，再转化为关节速度，发送给共享控制模块，实现对机械臂的控制。

3.4.4 基于图像的方法

同样在得到了四个特征点的实时图像坐标后，首先使用 ViSP 库 `vpFeatureBuild` 类中的 `create` 函数，读取相机内参，将图像坐标转化为相机坐标系下的三维点坐标，使用该函数需要手动设置深度，用于计算交互矩阵 \mathbf{L}_x 。我们指定四个特征点来定义任务，将得到的特征点与任务点分别作差并组合，得到了公式(2.4)中定义的误差 $\dot{\mathbf{s}}$ ， $\dot{\mathbf{s}}$ 为一个 8×1 的向量。接着使用公式(2.14)计算交互矩阵 \mathbf{L}_x ，由于存在 4 个特征点，我们将四个交互矩阵组合得到了一个 8×6 的矩阵 \mathbf{L} 。我们可以计算出相机速度：

$$\xi = \mathbf{L}^{-1}\dot{\mathbf{s}} \quad (3.12)$$

其中相机速度

$$\xi = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$$

对于具有 k 个特征值并且相机速度具有 m 个分量的情形，我们有 $\mathbf{L} \in \mathbb{R}^{k \times m}$ 。一般 $m = 6$ ，在本系统中，我们有四个特征点，每个特征点提供 (x, y) 两个特征值，所以 $k = 8$ 。当 $k > m$ ，并且矩阵 \mathbf{L} 满秩时我们通常会得到一个不一致系统。在视觉伺服系统中，这意味着我们观察到比唯一确定相机速度所需更多的特征速度。在这种情况下，矩阵 \mathbf{L} 的归零空间的秩为零，这是由于 \mathbf{L} 的列空间维度等于 $\text{rank}(\mathbf{L})$ ^[36]。在这种情况下我们可以使用最小二乘解

$$\xi = \mathbf{L}^+\dot{\mathbf{s}} \quad (3.13)$$

使用左伪逆矩阵，由

$$\mathbf{L}^+ = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T \quad (3.14)$$

给出。此时得到的相机速度 ξ 在末端坐标系下，通过订阅数据处理与传输模块得到的转化矩阵将 ξ 转换到基座坐标系下。

由公式(3.6)可以计算出关节速度

$$\dot{\theta} = \mathbf{J}^{-1}\xi \quad (3.15)$$

同样的，矩阵 \mathbf{J} 不一定满秩，这里同样适用伪逆矩阵

$$\dot{\theta} = \mathbf{J}^+ \xi \quad (3.16)$$

3.5 直接控制算法

直接控制时，主要是将手控器末端的位移映射到机械臂末端执行器中，实现移动手控器末端时机械臂跟随其运动。从手控器驱动程序中读取末端位姿信息（手控器末端坐标系下），得到末端位姿矩阵

$$\mathbf{T}_h = \begin{bmatrix} \mathbf{R}_h & \mathbf{p}_h \\ 0 & 1 \end{bmatrix}$$

其中 \mathbf{R}_h 为齐次矩阵姿态部分， \mathbf{p}_h 为位置部分。

再从数据处理与传输模块中订阅机械臂末端执行器到基座的转换矩阵，以及机械臂末端的位姿矩阵。记订阅得到的转换矩阵为

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix}$$

其中 \mathbf{R} 为齐次矩阵旋转部分， \mathbf{p} 为位移部分。

机械臂当前末端位姿为

$$\mathbf{T}_{ur} = \begin{bmatrix} \mathbf{R}_{ur} & \mathbf{p}_{ur} \\ 0 & 1 \end{bmatrix}$$

其中 \mathbf{R}_{ur} 为齐次矩阵姿态部分， \mathbf{p}_{ur} 为位置部分。

将手控器末端位移信息转化到基座坐标系下，得到期望位置

$$\mathbf{p}_d = \mathbf{R}\mathbf{p}_h \quad (3.17)$$

将期望位置与机械臂末端当前位置做差并乘以增益 λ_v ，得到近似的末端执行器在基座坐标系下的线速度

$$\hat{\mathbf{v}} = \lambda_v(\mathbf{p}_d - \mathbf{p}) \quad (3.18)$$

同理，将手控器末端姿态信息转化到基座坐标系下，得到表示期望姿态的旋转矩阵

$$\mathbf{R}_d = \mathbf{R}\mathbf{R}_h \quad (3.19)$$

再乘以机械臂末端当前表示当前姿态的旋转矩阵的逆，得到近似的末端执行器在基座坐标

系下的旋转矩阵

$$\mathbf{R}_b = (\mathbf{R}_d \mathbf{R}^{-1}) \quad (3.20)$$

现在我们需要将旋转矩阵转化为角速度 ω , 由罗德里格斯公式 (Rodrigues' rotation formula) 可以推导得到若时间间隔为 Δt , 旋转矩阵为 \mathbf{R} , 角速度向量为 $\omega = (\omega_x, \omega_y, \omega_z)$,

$$[\omega]_{\times} = \frac{\mathbf{R} - \mathbf{R}^T}{2\Delta t} \quad (3.21)$$

其中 $[\omega]_{\times}$ 为 ω 的反对称矩阵, 定义为

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (3.22)$$

由公式(3.21), 取增益为 λ_{ω} , 计算近似角速度向量的反对称矩阵为

$$[\hat{\omega}]_{\times} = \lambda_{\omega} [\omega]_{\times} \Delta t \quad (3.23)$$

由此可以得到近似的末端速度为

$$\hat{x}i = \begin{bmatrix} \hat{v} \\ \hat{\omega} \end{bmatrix} \quad (3.24)$$

取由公式(3.16), 可以解算出近似关节速度

$$\hat{\theta} = \mathbf{J}^+ \hat{\xi} \quad (3.25)$$

将 $\hat{\theta}$ 发布到机械臂关节速度控制器, 实现对机械臂的控制。

3.6 共享控制策略

共享控制模块, 将两种控制模块发布的速度信息通过特定的共享控制策略融合, 并发送给机械臂的关节速度控制器。这里仅仅给出一个非常简单的策略, 即在两种控制方法同时工作时, 从手控器驱动程序中读取末端按钮的状态, 若按钮没有被按下, 采用视觉伺服模块发布的速度控制信息, 若按钮被按下, 则采用直接控制模块发布的速度控制信息, 但此时速度的计算方式与前一小节稍有不同, 目的是可以实时介入姿态控制, 不同之处将在下文中详细说明。

当按钮被按下时记为零时刻, 程序自动记录零时刻手控器末端位姿矩阵, 并实时计算

当前时刻手控器末端位姿到零时刻末端位姿的转化矩阵

$$\mathbf{T}_h = \begin{bmatrix} \mathbf{R}_h & \mathbf{p}_h \\ 0 & 1 \end{bmatrix}$$

以及记录机械臂零时刻末端位姿

$$\mathbf{T}_{ur0} = \begin{bmatrix} \mathbf{R}_{ur0} & \mathbf{p}_{ur0} \\ 0 & 1 \end{bmatrix}$$

再从数据处理与传输模块中订阅机械臂末端执行器当前位姿（基座坐标系下）

$$\mathbf{T}_{ur} = \begin{bmatrix} \mathbf{R}_{ur} & \mathbf{p}_{ur} \\ 0 & 1 \end{bmatrix}$$

和转换矩阵

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix}$$

计算出机械臂末端期望位姿，并与当前位姿做差，得到机械臂末端执行器的近似速度，再计算出关节运动速度。

机械臂末端期望位置由零时刻机械臂位置加上手控器末端位移（基座坐标系下）得到，

$$\mathbf{p}_d = \mathbf{p}_{ur0} + \mathbf{R}\mathbf{p}_h \quad (3.26)$$

机械臂末端近似线速度 $\hat{\mathbf{v}}$ 可由公式(3.18)求得，

机械臂末端期望姿态由手控器末端姿态（手控器坐标系下）乘以零时刻机械臂姿态得到，

$$\mathbf{R}_d = \mathbf{R}_h \mathbf{R}_{ur0} \quad (3.27)$$

由公式(3.20)(3.21)(3.23)可以得到近似的末端角速度 $\hat{\xi}$ 。

又由公式(3.16)可以求得近似的关节速度 $\hat{\theta}$ 。

3.7 本章小结

本章系统设计详尽地介绍了基于视觉伺服的共享遥操作系统的整体设计与实现，涵盖了硬件设计、软件设计、机械臂运动学建模、控制算法及控制策略的实现过程。这些详细的描述，为整个系统的实现提供了明确的技术路线和具体的操作步骤。

在硬件设计部分，明确了系统的主要组件，包括主端控制器、从端执行器、反馈系统和通信回路。通过详细描述每个组件的功能和相互连接方式，确保了系统的各个部分可以

高效、可靠地协同工作，从而保证了系统的稳定性和响应速度。软件设计部分详细介绍了系统的软件架构，阐述了各个模块的功能及其实现方法。通过 ROS 框架，实现了各模块之间的数据处理和通信，使得系统能够实时响应操作者的指令，精确控制机械臂的运动。

机械臂建模部分涵盖了正运动学和速度运动学建模的详细过程，通过建立机械臂的运动学模型，能够准确计算机械臂的位姿和速度，为控制算法提供了坚实的理论基础。视觉伺服控制算法部分，介绍了基于位置的方法和基于图像的方法，通过使用 ViSP 库实现了二维码特征点的实时检测和姿态估计。这些算法使得系统能够利用视觉数据精确控制机械臂的运动，实现了对目标区域的精确定位和操作。直接控制算法部分介绍了如何通过手控器实现对机械臂的直接控制，将手控器末端的位移和姿态信息映射到机械臂的运动中，实现了高效、直观的远程操作。最后，共享控制策略部分详细描述了如何将视觉伺服和直接控制的速度信息融合，通过特定的共享控制策略，实现了两种控制模式的无缝切换。这不仅提高了系统的灵活性和安全性，也增强了操作者的对系统的掌控力。

总之，本章通过系统的设计和实现，充分展示了基于视觉伺服的共享遥操作系统的技术原理和实现方法。通过硬件与软件的有机结合、运动学建模与控制算法的精确实现，以及共享控制策略的优化应用，构建了一个功能强大、性能可靠的遥操作系统，为进一步的研究和实验提供了重要的技术支持和参考。

第四章 实验验证

4.1 实验设置

本节将详细描述实验设备和环境的配置，实验设置包括硬件设备的配置和软件环境的搭建。实验设备包括 UR3e 机械臂一台，RealSense D435 深度相机一台，一台笔记本电脑一台，3DSystems Haptic Touch 手控器一台，粘贴有二维码标记和操作区域的塑料板一块以及用于将上述硬件搭建在其上的实验平台一座。软件环境选用了 Ubuntu 20.04 LTS 作为操作系统，稳定发行版的 Linux 系统可以确保实验环境的兼容性和稳定性；控制软件使用 ROS Noetic 版本，与操作系统适配，并且可以利用其丰富的库和工具集成各种功能；代码部分，编程语言使用 C++，便于快速开发和测试，并且主要使用了 OpenCV 库，ViSP 库实现视觉伺服的部分功能，使用 Eigen 库实现机械臂建模和数据处理等过程。仿真环境使用 CoppeliaSim 4.6.0 版本，可以轻松搭建本实验的环境，并且具有 ROS 接口，方便进行软件的测试。

4.2 视觉伺服仿真实验

4.2.1 实验目的

在学习了视觉伺服的相关理论后，合理运用，验证 PBVS 和 IBVS 两种视觉伺服方法实现视觉伺服控制机械臂的可行性，并比较两种视觉伺服方法的优劣，为后续的实物实验选择更加合适的方法。

4.2.2 实验过程

首先在 CoppeliaSim 中搭建实验环境，包括 UR3 机械臂，视觉传感器，二维码标记物等；然后分别编写视觉传感器和机械臂脚本，由于在仿真环境中主要测试视觉伺服控制，实验简化了机械臂正运动学建模和关节速度控制部分，直接利用仿真软件获得机械臂的转换矩阵和末端位姿，进行速度计算和机械臂控制。搭建的场景如图4-1所示。

其中机械臂初始位置关节角为 $(90, 40, 100, -140, -90, 0)$ （单位为角度），在世界坐标系中的位置为 $(0.12488, 0.64991, 0.04303)$ （单位为米），朝向为 $(0, 0, 0)$ （单位为角度）。二维码在世界坐标系中的位置为 $(-0.38824, 0.55244, 0.22089)$ （单位为米），朝向为 $(100.417, 65.186, -128.076)$ （单位为角度），大小为 10 厘米。机械臂末端视觉传感器设置剪裁平面距离为 0.01 米到 10 米，视场角为 120 度，分辨率为 1024×1024 。

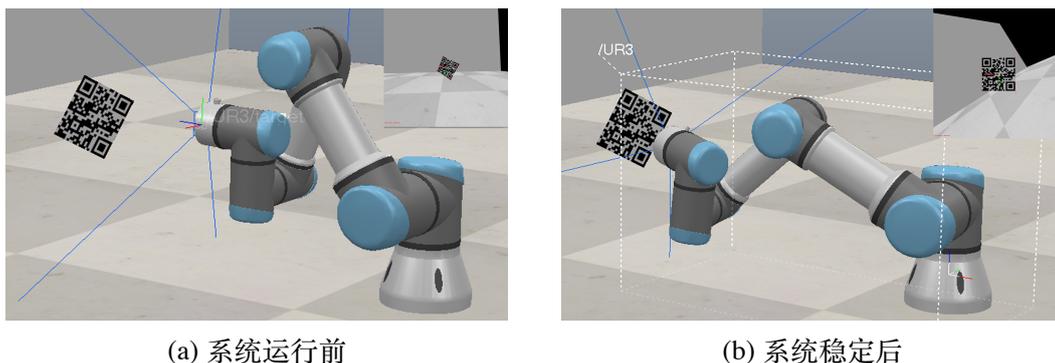


图 4-1 仿真场景搭建（以 PBVS 为例）

完成场景搭建后，考虑到 PBVS 方法更加直观，首先尝试 PBVS 方法，在仿真环境中编写脚本，使用 ROS 订阅视觉传感器发布的图像数据，接着使用视觉视觉伺服模块对图像中的二维码进行位姿估计，得到其位姿后直接更新机械臂末端位姿，并在脚本中记录系统运行过程中到系统稳定时的位姿偏差，如图4-2所示：

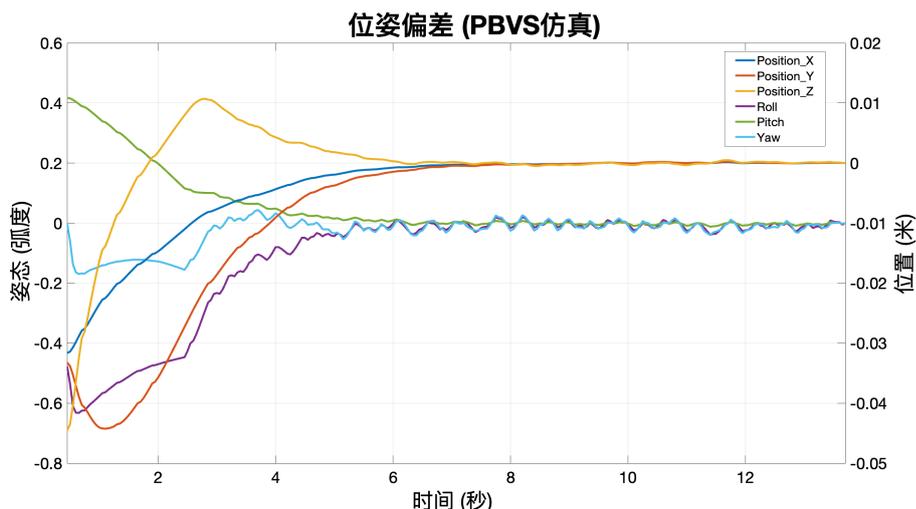


图 4-2 PBVS 仿真末端位姿偏差

图中深蓝色、红色、黄色 3 条曲线分别对应末端执行器 x, y, z 方向上位置与稳定时刻位置的偏差，单位为米，紫色、绿色、浅蓝色三条曲线分别对应末端执行器 x, y, z 方向上欧拉角表示的姿态与稳定时刻姿态的偏差。首先我们观察到位姿的偏差在系统运行过程中逐渐收敛到零值附近，说明系统逐渐趋于稳定，同时结合图4-1(b)，我们也可以看到，系统稳定后二维码正对着相机且位于相机视野中央，说明了 PBVS 方法进行机械臂控制的可行性。其中位置的波动幅度更小，而姿态的波动幅度较大，并呈现出类似于周期性的特征，在实验中的直观体现就是系统趋于稳定后机械臂末端仍在小幅转动，考虑到在 PBVS

方法中深度信息是必须的并且需要进行实时估计，而这个信息往往通过相机模型从二维图像中推算得到。深度估计的不准确会直接影响到控制系统的性能，尤其是在使用单目相机的情况下，深度估计本身就是一个具有挑战性的问题。同时，估计深度信息需要使用准确的相机内参和外参等，这些参数的微小误差都可能导致位姿估计不准确。由于上述原因，尽管 PBVS 方法更加直接，它的鲁棒性可能不如 IBVS 方法。

接下来验证 IBVS 方法，在 CoppeliaSim 中完成环境的搭建以及脚本的编写，需要注意的是，由于视觉伺服模块中 IBVS 方法发布的是机械臂末端的速度信息，故需要在脚本中进行速度运动学建模，计算雅各比矩阵。首先同样在 ROS 中订阅视觉传感器发布的图像数据，在视觉伺服模块中进行末端速度的计算，脚本中订阅模块计算得到的末端速度，使用建立的速度运动学模型，代入读取的转换矩阵信息，计算得到关节速度，并控制机械臂。在视觉伺服模块中记录特征点与任务点的偏差，如图4-3所示：

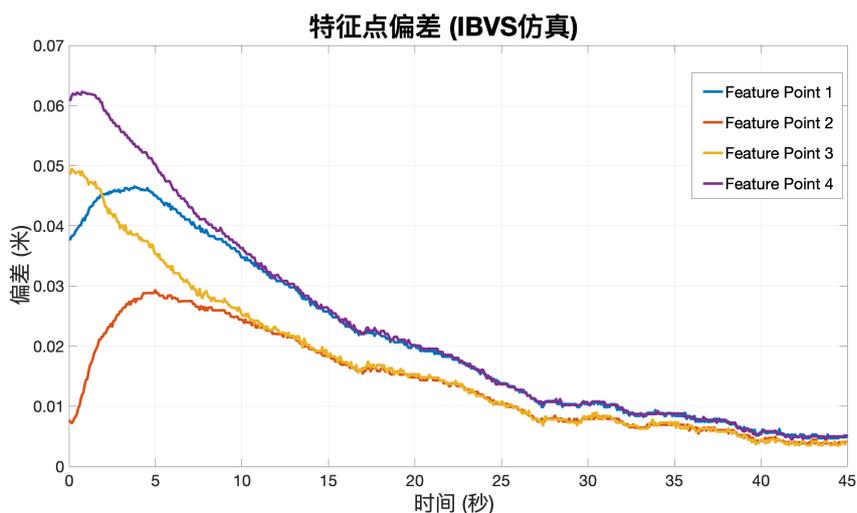


图 4-3 IBVS 仿真特征点偏差

图中蓝色、红色、黄色、紫色四条曲线分别代表记为特征点的二维码右下、左下、左上、右上四个角点与对应四个任务点之间的偏差。偏差为对应特征点和任务点之间在相机参考系中成像平面上的欧式距离，单位为米。我们可以发现四个特征点的偏差在一段时间后都逐渐减小，收敛到零值附近，说明 IBVS 方法进行机械臂控制的可行性。同样在脚本中记录系统运行过程中到系统稳定时的位姿偏差，如图4-4所示：

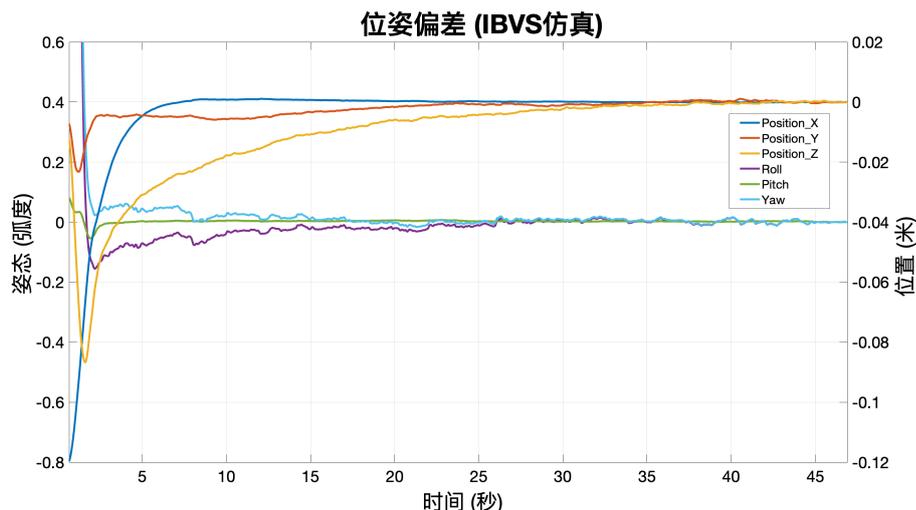


图 4-4 IBVS 仿真末端位姿偏差

图中深蓝色、红色、黄色 3 条曲线分别对应末端执行器 x , y , z 方向上位置与稳定时刻位置的偏差, 单位为米, 紫色、绿色、浅蓝色三条曲线分别对应末端执行器 x , y , z 方向上欧拉角表示的姿态与稳定时刻姿态的偏差。我们同样可以观察到位姿的偏差在系统运行过程中逐渐收敛到零值附近, 说明系统逐渐趋于稳定, 与图4-2比较可以发现, 两种方法记录的位置的偏差都表现良好, 收敛到非常接近零值的区域并且, 然而观察两种方法记录的姿态的偏差可以发现, 在表示姿态的纵轴 (左侧 y 轴) 采用相同区间 $-0.8rad$ 到 $0.6rad$ 的情况下, 使用 IBVS 方法得到的姿态偏差波动幅度明显更小, 说明姿态的稳定性比 PBVS 方法更高。从原理上看, IBVS 直接使用图像特征进行控制, 较少受到相机标定误差和深度估计错误的影响, 具有更高的鲁棒性。同时 IBVS 方法实现视觉伺服的过程中逐渐缩小特征点与任务点之间的偏差, 所以在目标和相机的相对运动逐渐变小后, IBVS 提供更加稳定的反馈控制。

4.2.3 实验结果与分析

从以上实验过程可知, 两种方法均能实现视觉伺服控制, 但从记录的实验数据来看, IBVS 具有更高的鲁棒性, 且受到实验器材的精度限制, 很可能无法精确估计目标的深度和位姿信息, 故在实物实验中选择 IBVS 方法。

4.3 视觉伺服实物实验

4.3.1 实验目的

在完成视觉伺服仿真实验后, 视觉伺服方法控制机械臂的可行性得到了验证, 接着进行视觉伺服的实物实验, 检查视觉伺服在完成按压指定区域任务时是否存在不足。

4.3.2 实验过程

首先在操作台上搭建实物实验场景，主要有机械臂与相机还有操作杆的装配，使用3-D打印技术打印相机与机械臂之间的连接件和操作杆，如图4-5和图4-6所示：

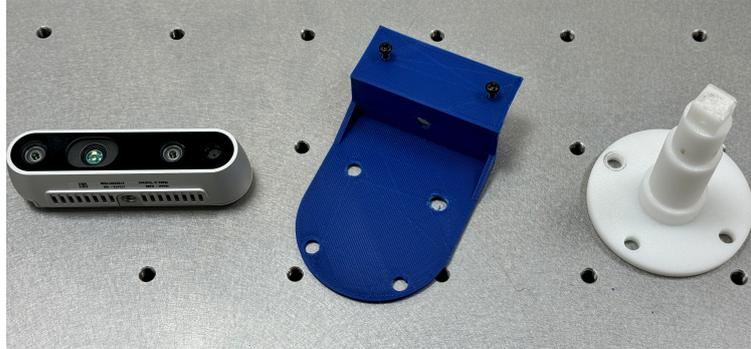


图 4-5 从左至右依次为深度相机、连接件和操作杆

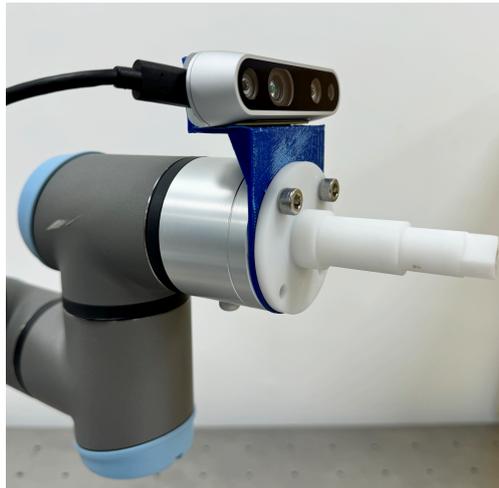


图 4-6 装配好的机械臂末端

以及目标打印和固定,如图4-7所示,其中二维码边长为4厘米,操作区域直径为2.5厘米,且在二维码右侧3.5厘米,下侧4厘米处。

在场景搭建的过程中,需要考虑相机内参的选取和外参的计算。相机内参直接调用ViS库中vpRealSense2类的函数getCameraParameters得到,

$$\begin{cases} p_x = 1366.75, \\ p_y = 1366.52, \\ u_0 = 961.355, \\ v_0 = 565.91. \end{cases}$$

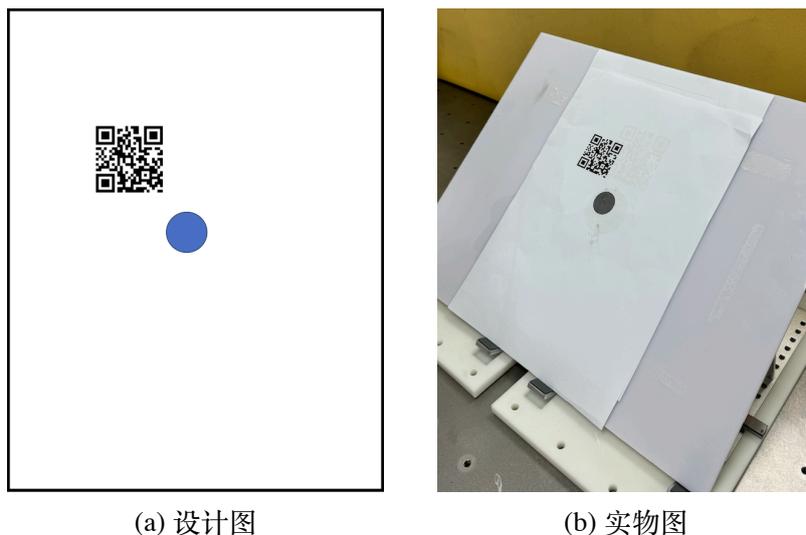


图 4-7 标志物与操作区域

这四个参数代表的相机内参分别是：

- p_x : 水平方向的焦距（以像素为单位）。
- p_y : 垂直方向的焦距（以像素为单位）。
- u_0 : 主点的水平坐标（图像中心的水平位置）。
- v_0 : 主点的垂直坐标（图像中心的垂直位置）。

在相机内参中，焦距通常会被表示为 f_x 和 f_y ，即水平方向和垂直方向上的焦距，它们与图像平面的物理尺寸和相机传感器上的像素大小有关。主点坐标 (u_0, v_0) 则表示成像平面上的光轴中心点在图像坐标系中的位置，通常接近图像的中心。

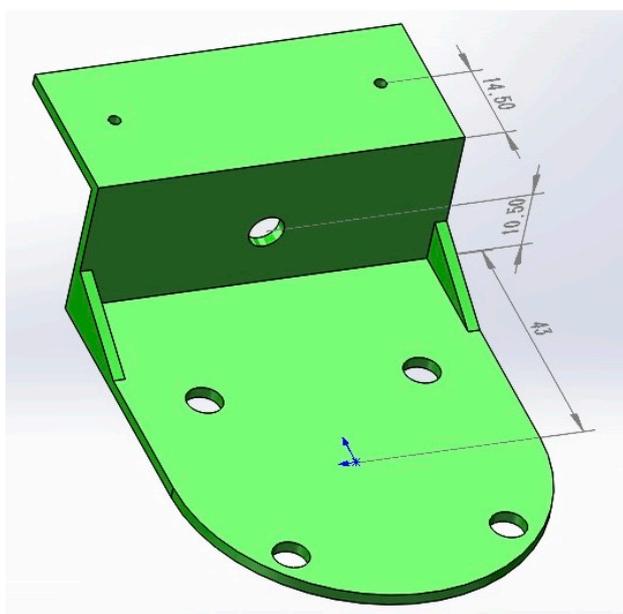


图 4-8 连接件建模图（尺寸单位为毫米）

相机的外参反映了相机在世界坐标系中的位置和方向，由于相机通过连接件连接在机械臂末端，记相机的外参齐次矩阵为

$$\mathbf{T}_c = \begin{bmatrix} \mathbf{R}_c & \mathbf{p}_c \\ 0 & 1 \end{bmatrix}$$

其中 \mathbf{R}_c 代表相机坐标系到机械臂末端坐标系的旋转部分， \mathbf{p}_c 代表位移部分。由连接件 3 维建模图 4-8 可知。

通过连接件的连接，机械臂末端坐标系和相机坐标系朝向相同，故外参矩阵旋转矩阵为单位矩阵 \mathbf{I} ，参考图 4-8 中所标注的尺寸，可以得到相机的外参齐次矩阵为

$$\mathbf{T}_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.575 \\ 0 & 0 & 1 & -0.105 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

完成场景搭建后，首先启动机械臂配置与计算机的通讯，完成后在 ROS 打开机械臂驱动，进行机械臂的标定，接着在机械臂示教器中完成机械臂的初始化，角度设定为 $(-36, -110, -135, 30, 90, 195)$ ，单位为角度。机械臂基座与操作区域在操作台平面内的偏差为 $(11, 44)$ ，操作区域约在操作台平面上方 11.5 厘米处，塑料板与桌面夹角约为 40 度。通过改变关节角度设定机械臂的位姿，如图 4-9 所示：



图 4-9 机械臂初始状态

完成初始化后，打开机械臂远程控制模式并启动关节速度控制器。依次启动数据处理与传输模块，视觉伺服模块，开始视觉伺服控制过程并记录数据。

4.3.3 实验结果与分析

在视觉伺服模块中记录特征点与任务点的偏差，如图4-10所示：

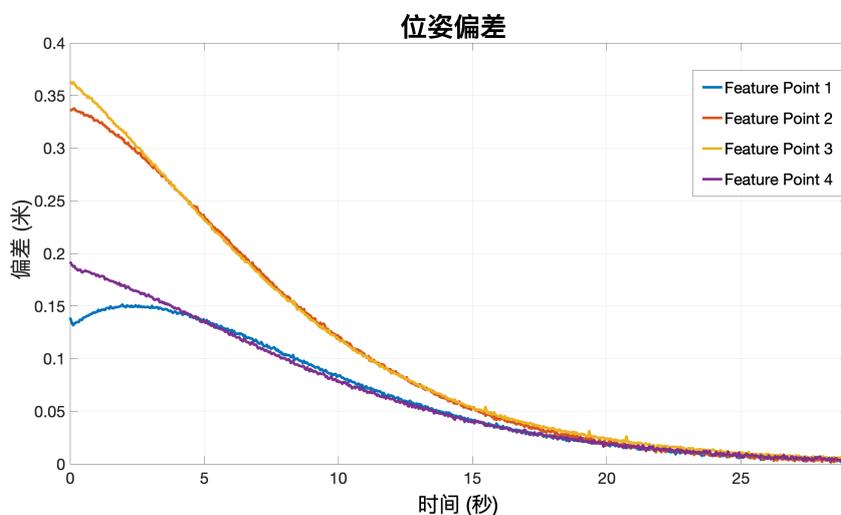


图 4-10 特征点偏差

图中蓝色、红色、黄色、紫色四条曲线分别代表记为特征点的二维码右下、左下、左上、右上四个角点与对应四个任务点之间的偏差。偏差为对应特征点和任务点之间在相机参考系中成像平面上的欧式距离，单位为米。由图像可知，四个特征点的偏差在一段时间后都收敛到零附近，同时观察系统稳定后的相机视角和末端位姿，如图4-11所示：

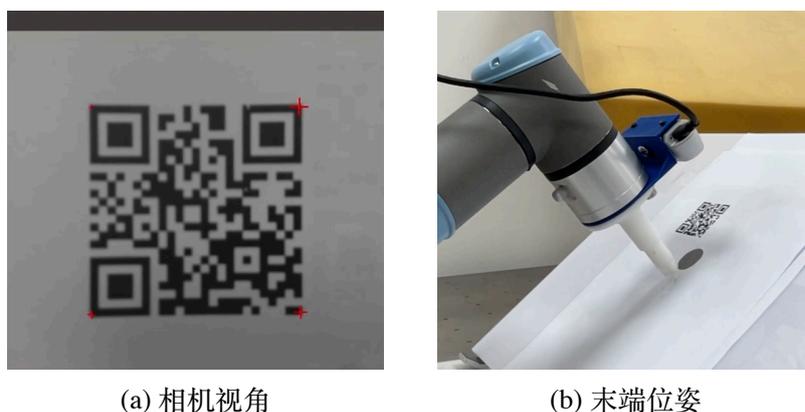


图 4-11 视觉伺服系统稳定后

IBVS 方法较好的完成了二维码的定位任务，但是仔细观察相机末端位姿，如图4-11(b)所示：不难发现操作杆末端与需要按压的区域仍存在一定距离，同时末端也没有准确悬停在需要按压的操作区域上方，存在细微的偏差。出现误差的原因包括操作杆末端与相机主点位置不重合，使用 IBVS 方法进行控制的精确度不够高等。

上述结果说明仅仅使用视觉伺服对机械臂进行控制仍然存在定位不够精确的问题，也从侧面证明了引入直接控制，将其与视觉伺服融合进行共享控制的必要性。

4.4 共享控制实验

4.4.1 实验目的

由视觉伺服实物实验可知，仅使用单一的视觉伺服控制机械臂并不能完成设计的按压指定区域任务，需要引入别的控制方法与之融合，进行共享控制，并完成任务。

4.4.2 实验过程

在视觉伺服实物实验的基础上，加入手控器实现机械臂的直接控制功能，并采用系统设计章节中提到的控制策略实现共享控制。首先完成手控器与计算机的连接和配置，并启动直接控制模块和共享控制模块。在视觉伺服完成控制过程，机械臂稳定后，观察末端操作杆与指定区域的偏差，按压手控器按钮，移动手控器末端，人工介入机械臂的控制，完成更为精细的定位和按压任务。在实验过程中记录机械臂末端速度，末端位置，末端力传感器，手控器按钮状态，末端位置等数据。

记手控器按键按下时状态为真，视觉伺服运行时状态为真，共享控制的伪代码如下：

算法 1: 如何实现共享控制

Data: 手控器按键状态，视觉伺服状态，关节速度

Result: 选择合适的控制方法，实现共享控制

- 1 共享控制模块启动;
 - 2 设置关节速度为零;
 - 3 **while** 视觉伺服状态 **do**
 - 4 读取手控器按键状态;
 - 5 **if** 手控器按键状态 **then**
 - 6 关节速度设置为直接控制模块发布的值;
 - 7 **else**
 - 8 关节速度设置为视觉伺服模块发布的值;
 - 9 **end**
 - 10 **end**
 - 11 向控制器发送关节速度;
-

4.4.3 实验结果与分析

观察记录的机械臂末端速度与时间的折线图4-12:

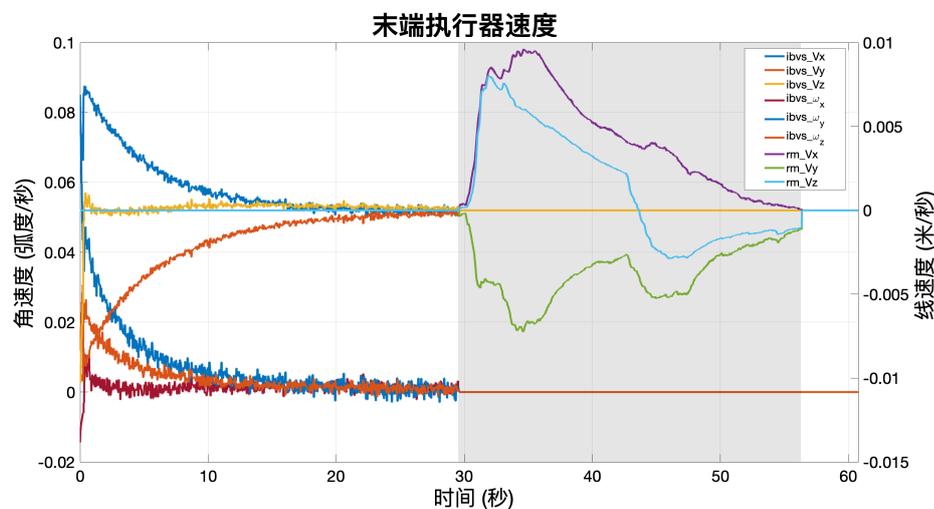


图 4-12 机械臂末端速度

图中上半部分记录了机械臂末端 x, y, z 方向上的线速度, 下半部分记录了机械臂末端 x, y, z 方向上的角速度。图像中阴影部分为按下按键, 使用手控器直接控制机械臂的时间段。分析图像可以发现, 按下按键之前, 在视觉伺服控制下, 机械臂末端速度逐渐收敛到零附近, 系统趋于稳定。当按下按钮后, 由于直接控制介入, 机械臂末端跟随手控器末端移动, 速度首先变大, 在对准操作区域按压, 接触到目标后速度逐渐减小。随后松开按键, 保持末端操作杆对目标的按压。

读取记录的机械臂末端力传感器数据并画图4-13:

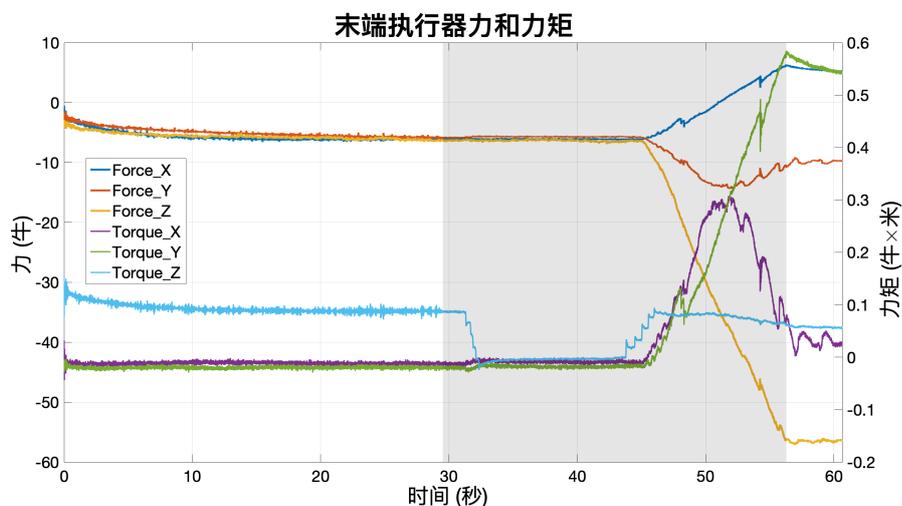


图 4-13 机械臂末端力和力矩

图中深蓝色、红色、黄色三条曲线代表了机械臂末端力传感器采集到的 x , y , z 方向上的力数据, 紫色、绿色、浅蓝色三条曲线代表了力传感器采集到的 x , y , z 方向上的力矩数据。由图像可以看到, 系统启动时, 由于机械臂的震动, 力和力矩数据发生了一定幅度的抖动, 随后视觉伺服工作的过程中, 力和力矩的数据随着末端姿态的变化发生了微小的改变, 直接控制接管后, 由于机械臂末端速度突然变大, 绕 z 轴方向力矩数据发生了一定幅度的改变, 在按压过程中, 力和力矩数据出现的大幅变化, 反映了末端操作杆与目标发生接触。最后力和力矩的数据又趋于稳定则反映了按键被松开, 控制结束。

我们将读取的手控器末端位置与机械臂末端位置画在一张图4-14下。图中三条实线代表了手控器末端位置, 而三条虚线代表了机械臂末端位置, 相同颜色的实线和虚线对应相同方向的位置。观察图片可以发现, 三条实线代表的手控器末端位置与三条虚线代表的机械臂末端位置一一对应, 机械臂末端位置曲线逐渐与手控器末端位置曲线重合, 体现了机械臂末端跟随手控器末端运动。然而对应曲线之间的偏差反应了直接控制的迟滞, 平均在 7 秒左右, 虽然考虑到为了系统的稳定性, 直接控制时设定的速度增益处较小, 机械臂运动较为缓慢, 但无论如何, 在更为精细和复杂的任务中存在这么高的控制延时是不可接受的。

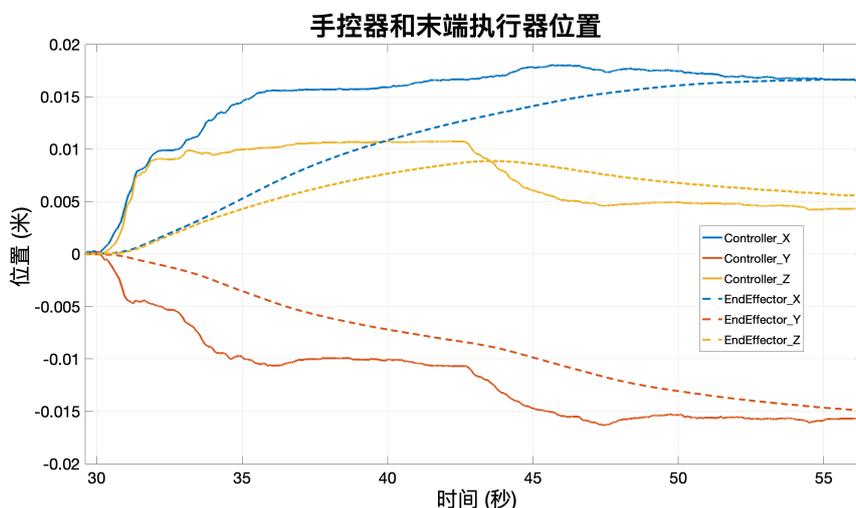


图 4-14 手控器和机械臂末端位置

在最终的共享控制实验中, 我使用简单的策略把视觉伺服和直接控制融合, 实现了不同控制方法对机械臂的协同控制, 并完成了按压指定区域的任务。然而从采集到的数据来看, 整个控制过程较为粗糙, 尤其是直接控制部分的延时问题仍有很大改进空间。

4.5 本章小结

本章通过一系列实验验证了基于视觉伺服的共享遥操作系统的设计和实现。实验设置部分详细描述了硬件设备的配置和软件环境的搭建，为后续实验奠定了基础。

在视觉伺服仿真实验中，验证了 PBVS 和 IBVS 两种视觉伺服方法的可行性，并通过数据对比发现 IBVS 方法在鲁棒性和稳定性方面具有优势。实验过程展示了如何在仿真环境中进行视觉伺服控制，并通过记录和分析特征点与任务点之间的偏差，验证了两种方法的性能。

视觉伺服实物实验进一步验证了 IBVS 方法在实际应用中的可行性。实验结果显示，尽管 IBVS 方法能够较好地完成二维码的定位任务，但由于操作杆末端与相机主点位置不重合以及控制精度的限制，系统在按压指定区域时存在一定误差。这一结果强调了仅使用视觉伺服控制仍存在定位不够精确的问题，体现了引入直接控制进行共享控制的必要性。

共享控制实验通过结合视觉伺服和直接控制，实现了对机械臂的精细控制。采用共享控制策略可以有效地弥补单一控制方法的不足，实现了按压指定区域的任务。然而，在对实验数据分析的过程中也发现直接控制部分存在较高的延时问题，平均延时较大，这在精细和复杂的任务中是不可接受的。

综上所述，本章通过仿真实验和实物实验，系统地验证了基于视觉伺服的共享遥操作系统的设计和实现。实验结果展示了系统在实际应用中的可行性和存在的问题，为进一步优化和改进系统提供了宝贵的参考。未来的研究将着重解决直接控制的延时问题，并探索更复杂的共享控制策略，以提升系统的整体性能和应用范围。

第五章 总结与展望

5.1 工作总结

本论文主要研究基于视觉伺服的共享操作系统的控制设计。通过对当前机器人的控制方法及其局限性的分析，提出了一种结合视觉伺服技术的闭环控制方案，以增强机械臂在复杂环境下的适应性和任务执行能力。

本文的主要创新点在于，结合视觉伺服技术与遥控技术，提出了一种新的共享控制方法，实现了两种控制模式的无缝切换。这不仅提高了系统的灵活性和安全性，也增强了操作者对系统的控制，并且通过实时感知机械臂末端的位姿信息，实现真正的闭环控制。

本文的主要成果在于，在进行完善的系统设计之后，通过仿真研究，首先验证了基于位置的视觉伺服和基于图像的视觉伺服两种方法的可行性。通过对比实验数据，我们发现 IBVS 方法在鲁棒性和稳定性方面表现出明显优势。为了进一步验证 IBVS 方法在实际应用中的可行性，继续进行了视觉伺服实物实验。实验结果表明，尽管 IBVS 方法能够较好地完成二维码的定位任务，但由于操作杆末端与相机主点位置不重合以及控制精度等的限制，系统在按压指定区域时存在一定误差。这一结果强调，仅依靠视觉伺服控制，定位精度仍不够，体现了引入直接控制进行共享控制的必要性。最后，提出了共享控制策略，将视觉伺服和直接控制相结合，实现了对机械臂的精细控制，并完成了设计的任务并对系统的性能进行了评估。

综合实验结果表明，IBVS 方法在鲁棒性和稳定性方面具有明显优势，但在实际应用中仍存在定位误差的问题。共享控制策略通过结合视觉伺服和直接控制，弥补了单一控制方法的不足，实现了对机械臂更佳精细地控制。然而，直接控制部分的延时问题需要进一步优化，以满足精细和复杂任务的需求。未来的研究方向将着重于提高直接控制的响应速度和精度，以进一步提升系统的整体性能。

5.2 工作展望

尽管本论文取得了一定的研究成果，但仍有许多问题值得进一步探讨。未来的研究工作可以从以下几个方面进行深入：

1. 系统优化与改进：进一步优化控制算法，降低控制的延迟，提高系统的鲁棒性，以适应更复杂和多变的工作环境。

2. 人机协同研究：优化共享控制策略，探索人机协同控制的新模式，充分发挥人类的

认知优势和机器的执行优势，实现更高效的任务完成。

3. 实际应用推广：将本论文提出的控制方法应用于实际工业环境中，进行大规模实验测试，验证其在真实工作场景下的有效性和可靠性。

4. 智能化发展：引入人工智能技术，研究基于深度学习的自适应控制方法，进一步提升系统的智能化水平。

通过以上研究工作，相信基于视觉伺服的共享遥操作系统在未来将会有更广泛的应用前景，并对社会发展产生积极影响。

参考文献

- [1] LI D, LIU H, WEI T, et al. Robotic grasping method of bolster spring based on image-based visual servoing with yolov3 object detection algorithm[J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2022, 236(3): 1780-1795.
- [2] LIPPIELLO V, SICILIANO B, VILLANI L. Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration[J]. IEEE Transactions on Robotics, 2007, 23(1): 73-86.
- [3] 鹿振宇. 面向协同操作的多人多机遥操作控制方法研究[D]. 西北工业大学, 2019.
- [4] VERTUT J. Ma23m contained maintenance servomanipulator with computer-aided control[J]. Trans. Am. Nucl. Soc.:(United States), 1984, 47(CONF-841105-).
- [5] CROMBEZ N, MOUADDIB E M, CARON G, et al. Visual servoing with photometric gaussian mixtures as dense features[J]. IEEE Transactions on Robotics, 2018, 35(1): 49-63.
- [6] MARCHAND E. Subspace-based direct visual servoing[J]. IEEE Robotics and Automation Letters, 2019, 4(3): 2699-2706.
- [7] MARCHAND E. Direct visual servoing in the frequency domain[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 620-627.
- [8] FELTON S, BRAULT P, FROMONT E, et al. Visual servoing in autoencoder latent space[J]. IEEE Robotics and Automation Letters, 2022, 7(2): 3234-3241.
- [9] LEVINE S, FINN C, DARRELL T, et al. End-to-end training of deep visuomotor policies[J]. Journal of Machine Learning Research, 2016, 17(39): 1-40.
- [10] KRAGIC D, CHRISTENSEN H I, et al. Survey on visual servoing for manipulation[J]. Computational Vision and Active Perception Laboratory, Fiskartorpsv, 2002, 15: 2002.
- [11] LAGHI M, AJOUDANI A, CATALANO M G, et al. Unifying bilateral teleoperation and tele-impedance for enhanced user experience[J]. The International Journal of Robotics Research, 2020, 39(4): 514-539.
- [12] SELVAGGIO M, CACACE J, PACCHIEROTTI C, et al. A shared-control teleoperation architecture for nonprehensile object transportation[J]. IEEE Transactions on Robotics, 2021, 38(1): 569-583.
- [13] FEIZI N, PATEL R V, KERMANI M R, et al. Adaptive wave reconstruction through regulated-bmflc for transparency-enhanced telerobotics over delayed networks[J]. IEEE Transactions on Robotics, 2022, 38(5): 2928-2942.

- [14] PAIK P, THUDI S, ATASHZAR S F. Power-based velocity-domain variable structure passivity signature control for physical human-(tele) robot interaction[J]. IEEE Transactions on Robotics, 2022, 39(1): 386-398.
- [15] JORDA M, VULLIEZ M, KHATIB O. Local autonomy-based haptic-robot interaction with dual-proxy model[J]. IEEE Transactions on Robotics, 2022, 38(5): 2943-2961.
- [16] PANZIRSCH M, PEREIRA A, SINGH H, et al. Exploring planet geology through force-feedback telemanipulation from orbit[J]. Science robotics, 2022, 7(65): eabl6307.
- [17] CHEN Y, MENG M Q H, LIU L. Direct visual servoing based on discrete orthogonal moments[J]. IEEE Transactions on Robotics, 2024.
- [18] LAI B, LI Z, LI W, et al. Homography-based visual servoing of eye-in-hand robots with exact depth estimation[J]. IEEE Transactions on Industrial Electronics, 2023.
- [19] KUBAN D, MARTIN H. Advanced remotely maintainable force-reflecting servomanipulator concept[R]. Oak Ridge National Lab., TN (USA), 1984.
- [20] HUTCHINSON S, HAGER G D, CORKE P I. A tutorial on visual servo control[J]. IEEE transactions on robotics and automation, 1996, 12(5): 651-670.
- [21] CHAUMETTE F, HUTCHINSON S. Visual servo control. i. basic approaches[J]. IEEE Robotics & Automation Magazine, 2006, 13(4): 82-90.
- [22] YU P, TAN N, MAO M. Position-based visual servo control of dual robotic arms with unknown kinematic models: A cerebellum-inspired approach[J]. IEEE/ASME Transactions on Mechatronics, 2023.
- [23] OLIVA A A, GIORDANO P R, CHAUMETTE F. A general visual-impedance framework for effectively combining vision and force sensing in feature space[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 4441-4448.
- [24] HIRCHE S, BUSS M. Human-oriented control for haptic teleoperation[J]. Proceedings of the IEEE, 2012, 100(3): 623-647.
- [25] NUÑO E, BASAÑEZ L, ORTEGA R. Passivity-based control for bilateral teleoperation: A tutorial[J]. Automatica, 2011, 47(3): 485-495.
- [26] NITZSCHE N, SCHMIDT G. Force-reflecting telepresence in extensive remote environments[J]. Journal of Intelligent and Robotic Systems, 2007, 50: 3-18.
- [27] NUÑO E, ORTEGA R, BARABANOV N, et al. A globally stable pd controller for bilateral teleoperators

- [J]. IEEE Transactions on Robotics, 2008, 24(3): 753-758.
- [28] POLUSHIN I G, TAYEBI A, MARQUEZ H J. Control schemes for stable teleoperation with communication delay based on ios small gain theorem[J]. Automatica, 2006, 42(6): 905-915.
- [29] CRANDALL J W, GOODRICH M A. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation[C]//IEEE/RSJ international conference on intelligent robots and systems: volume 2. IEEE, 2002: 1290-1295.
- [30] SICILIANO B, KHATIB O, KRÖGER T. Springer handbook of robotics: volume 200[M]. Springer, 2008.
- [31] SHERIDAN T. Human supervisory control of robot systems[C]//Proceedings. 1986 IEEE International Conference on Robotics and Automation: volume 3. IEEE, 1986: 808-812.
- [32] ABBINK D A, MULDER M, BOER E R. Haptic shared control: smoothly shifting control authority?[J]. Cognition, Technology & Work, 2012, 14: 19-28.
- [33] ROSENBERG L B. Virtual fixtures: Perceptual tools for telerobotic manipulation[C]//Proceedings of IEEE virtual reality annual international symposium. IEEE, 1993: 76-82.
- [34] LI Y, NG W Y, LI W, et al. Towards semi-autonomous colon screening using an electromagnetically actuated soft-tethered colonoscope based on visual servo control[J]. IEEE Transactions on Biomedical Engineering, 2023.
- [35] MARCHAND É, SPINDLER F, CHAUMETTE F. Visp for visual servoing: a generic software platform with a wide class of robot control skills[J]. IEEE Robotics & Automation Magazine, 2005, 12(4): 40-52.
- [36] SPONG M W, HUTCHINSON S, VIDYASAGAR M. Robot modeling and control[M]. John Wiley & Sons, 2020.

致 谢

转眼间，我四年本科的旅程已经来到了毕业设计的致谢部分，尤其在疫情的影响下，这四年仿佛一晃就过去了，有种体验卡四年，封号两年的感觉。正好借着这个机会，感谢一下陪伴在我身边的大家。

首先感谢在我毕业设计过程中给予我极大帮助的我的指导老师，杨缘。虽然没有上过杨老师的课，但有幸在本科的最后阶段选择了杨老师作为我的毕设指导老师。杨老师给我提供了非常优渥的环境，包括但不限于让全寝室都无比眼红的单人间（虽然后来因为某些原因无了）、新买的显示器和昂贵的机械臂，同时也给我提供了悉心的指导，总是理直气壮（其实问多了也不好意思）地拿一些蠢问题打扰老师，老师也不厌其烦得回答。接下来，我要感谢同样一直陪伴我做毕设，甚至陪我一起熬大夜的幸倩影同学，虽然和你吵吵闹闹，但每天学习实验之余，最开心的就是和你待在一起的时间，我们要好好在一起哦！Emm，先定个小目标，我也要出现在你的毕设致谢里！

当然，我还要感谢非常非常爱我的爸爸妈妈，作为我最为坚实的后盾，你们各自以不同的方式表达着对我的爱，我都能感受的到。我也爱你们！虽然我可能不是勇于拼搏的性格，但以后也会尽力走好自己的路，不辜负你们的培养和期望。

最后，我要感谢我的舍友们，鲁振松、石晓东、张宇恒、周喆琦、李超然、杨源旭，你们可是这四年里陪我最久的人了，很开心能够遇到你们！还有陪我打了无数局游戏的周天悦学弟，我要是有显示屏和好鼠标垫，单挑肯定是轻松赢你的。祝你，好好学习，天天向上，期待以后与你再见。

致谢就写到这里了，很想唱一首林志炫的《凤凰花开的路口》作为结束，但是现在一是已经 27 号凌晨 2 点了，我连论文都还没交；二是还在实验室，杨老师还在我旁边努力工作，就算咯，等到真毕业的时候，得边喝酒边唱边哭一下。

对了，也祝自己以后在科研的道路上越走越远，多写点能发表的致谢，哈。