

分类号: TP27

单位代码: 10335

密级: 公开

学号: 21225086

浙江大学

硕士学位论文



中文论文题目: 轮式移动机器人建模与运动控制策略研究

英文论文题目: Modeling and Motion Control Strategy of Wheeled Mobile Robot

申请人姓名: 邓哲宇

指导教师: 姚斌教授

合作导师: 朱笑丛副教授

专业名称: 机械电子工程

研究方向: 机器人运动控制

所在学院: 机械工程学院

论文提交日期 2015 年 1 月

轮式移动机器人建模与运动控制策略研究



论文作者签名:_____

指导教师签名:_____

论文评阅人 1: _____

评阅人 2: _____

评阅人 3: _____

评阅人 4: _____

评阅人 5: _____

答辩委员会主席: _____

委员 1: _____

委员 2: _____

委员 3: _____

委员 4: _____

委员 5: _____

答辩日期: _____

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。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 浙江大学 或其他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谢意。

学位论文作者签名：

签字日期：

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：

导师签名：

签字日期： 年 月 日

签字日期： 年

致谢

感谢国家重点实验室基金、国家千人计划基金对本课题的资助。

岁月如梭，如歌。转眼间，二年半的研究生求学生涯即将结束，回首往昔，奋斗和辛劳成为丝丝的记忆，甜美和欢笑也尘埃落定。值此毕业论文完成之际，我谨向所有关心、爱护、帮助我的人们表示最诚挚的感谢与最美好的祝愿。

本学位论文是我攻读硕士学位期间，导师国家千人计划、长江学者姚斌教授悉心指导与亲切关怀下完成的，论文凝聚了导师的智慧与心血。感谢导师姚斌教授，作为机电控制领域的国际知名专家，能够细心指导我这样一名普通硕士生。您敏捷睿智的思维、高深的学术造诣和严谨的治学态度使我在知识层面、科研方法以及学术作风等方面都受益良多。这两年多，你一直担当我科研道路的领路人，对我未来的生活产生着积极深远的影响。

感谢导师朱笑丛副教授一直以来的指导、支持和照顾。无论导师姚斌教授是否在国外教学，您都一直给予我悉心的指导，解决我在科研与生活上的困惑，也倾注了大量的关心和帮助，同时也为我提供了优越宽松的科研环境。是您在我人生迷茫的时候开导与鼓励我，也是您在关键时刻的定力支持让我度过各种难关。

感谢课题组王庆丰老师对课题的宝贵建议和帮助。

感谢课题组陈正博士在课题最初开展时给予的指导和帮助，使我快速融入课题组的研究生活中来，感谢课题组陈珊博士对我研究中问题不断耐心的解答及生活上的关怀，感谢高德东博士对我科研上工具使用的指导及生活上的帮助，感谢李聪博士对课题研究上的建议，感谢李超博士在我入校时生活上的帮助及实验室科研上的探讨，感谢实验室其他研究生对我生活上的帮助。

感谢浙江大学对我的培养，浙大这百年名校所凝聚出来的求是创新、开拓进取的精神和深厚的底蕴，将深深影响我今后的人生旅程。

感谢我远方的父母、家人对我学业和生活上的关怀、鼓励和支持，您们是我前进的强大后盾。

最后，谨向所有给予我关心、指导和帮助的老师、亲人、同学和朋友们致以最诚挚的谢意！

邓哲宇

二零一五年一月于浙大求是园

摘要

本论文受国家重点实验室基金、国家千人计划基金资助。

自 60 年代末期开始的轮式移动机器人 (Wheeled Mobile Robot, WMR) 研究起, 机器人技术逐渐成熟, 并逐渐运用于人们生活的各个方面。针对日益高要求的控制精度, 本文针对一个具体的轮式移动机器人平台提出了两整套的控制策略, 通过算法的优化方法提高系统的运动精度。

首先, 轮式移动机器人是一个典型的多输入多输出非完整约束系统, 系统中也存在参数不确定性和模型不确定性。本文针对轮式移动机器人建立了包含直流电机电磁力学模型在内的完整的动力学模型, 并且进行了系统辨识与解耦分析。

其次, 根据得到的简化系统模型与实际系统的对比, 发现系统存在低频摩擦力影响与高频模型不确定性影响。为此设计了 μ -synthesis 控制器, 该控制器在系统所能达到的频宽范围内, 能最大程度地削弱摩擦力的影响, 并且其中的参数不确定性模型能够较好的涵盖系统相对高频的模型不确定性, 达到了一个较好的鲁棒特性。其中在轨迹线跟踪实验中设计了内外环控制框架, 将 μ -synthesis 控制器很好地应用到了实际平台。

最后, 针对实际中系统辨识存在误差, 系统特性也会受到负载、工况、环境等因素影响的情况, 在动力学中进一步设计了自适应鲁棒控制器 (ARC), 以满足自适应和鲁棒性能方面的要求。该控制器中使用投影自适应率实时更新模型参数, 结合滑模的控制思想额外附加一个非线性反馈控制项, 该项可以使系统具有较强的鲁棒性能, 并且使系统达到一个很快的响应性能。其中在避障实验中设计了三层控制框架, 涵盖了轮式移动机器人系统的路径规划、运动学与动力学控制。在最上层使用了改进的二叉树路径规划算法, 底层的自适应鲁棒控制保证了上层算法的实际应用性, 各层相辅相成, 达到避障路径最短的效果。

关键词: 轮式移动机器人, 建模与辨识, μ -synthesis 控制, 自适应鲁棒控制, 二叉树路径规划

Abstract

Robot technology has become more and more mature and has been applied to every aspect of people's life since the beginning of research on Wheeled Mobile Robot (WMR). This thesis will propose two kinds of control strategy applied to a specific WMR in dynamic to improve motion precision with the method of algorithm optimization.

Firstly, the WMR is a typical multiple input-multiple output nonholonomic system, and parameter uncertainty and model uncertainty exist in the system, so this thesis gives a completed modeling of a specific WMR, including electromagnetic model of DC motor. The parameter of the system is obtained, then the completed model is simplified and compared with the actual system.

Secondly, according to the contrast of the simplified system model and actual system, friction is found existed in low frequency and flexible model uncertainty is found existed in relative high frequency. In order to eliminate influence of the two factors, μ -synthesis control is designed. This control can eliminate the influence of friction in a high efficiency and use a parameter uncertainty model to cover the uncertainties in the relative high frequency, so the proposed μ -synthesis control achieve a good robust performance. A inner-outer control structure is designed to apply μ -synthesis to real system in trajectory tracking experiment.

Lastly, friction, flexibility features and sensor accuracy in the actual system affected the parameter identification results. If the WMR need to carry different objects, model parameters will change, so Adaptive Robust Control (ARC) is designed. A three level control structure is also designed to cover the path planning, kinematics and dynamics of the WMR. The designed control strategy can achieve high motion precision so as to promise the binary-tree path algorithm in high level can applied to actual system successfully.

Keywords: Wheeled Mobile Robot, modeling and identification, μ -synthesis control, ARC control, binary tree path planning

目录

致谢.....	I
摘要.....	II
Abstract.....	III
目录.....	IV
插图和附表清单.....	VII
第 1 章 绪论.....	1
1.1 课题研究背景.....	1
1.2 国内外研究现状.....	3
1.2.1 移动机器人的运动学和动力学建模.....	4
1.2.2 移动机器人的运动学和动力学控制.....	6
1.2.3 移动机器人的路径规划.....	9
1.3 课题研究意义及研究内容.....	11
1.3.1 课题的研究意义.....	11
1.3.2 课题的研究内容.....	12
1.4 本章小结.....	13
第 2 章 机器人平台系统搭建及建模.....	14
2.1 实验平台系统.....	14
2.1.1 基本架构及系统连接设计.....	14
2.1.2 传感器标定.....	16
2.2 轮式移动机器人模型推导.....	21
2.2.1 运动学模型.....	21
2.2.2 非完整约束建模分析.....	22
2.2.3 完整动力学模型.....	25
2.3 系统辨识.....	27
2.3.1 物理测量法辨识.....	27
2.3.2 时频域辨识.....	29
2.4 本章小结.....	32
第 3 章 μ -synthesis 鲁棒控制器设计.....	33

3.1	模型分析	33
3.1.1	模型解耦.....	33
3.1.2	模型分析.....	35
3.2	μ -synthesis 控制器.....	36
3.2.1	控制方法选择.....	36
3.2.2	控制方法思想.....	38
3.2.3	控制器设计过程.....	40
3.3	实验验证及应用控制框架设计	43
3.3.1	PID 控制器调试	43
3.3.2	动力学速度控制实验.....	44
3.3.3	内外环控制框架设计及位置跟踪控制实验.....	46
3.4	本章小结	48
第 4 章	自适应鲁棒控制器设计.....	49
4.1	控制系统分析	49
4.1.1	问题阐述.....	49
4.1.2	自适应控制.....	50
4.1.3	确定性鲁棒控制.....	51
4.2	自适应鲁棒控制 (ARC)	52
4.2.1	控制方法优势.....	52
4.2.2	设计过程.....	53
4.3	实验验证及应用控制框架设计	55
4.3.1	ARC 参数调节	55
4.3.2	动力学速度控制实验.....	56
4.3.3	三层控制框架设计及路径规划与跟踪实验.....	58
4.4	本章小结	64
第 5 章	总结及展望.....	65
5.1	总结	65
5.2	展望	66
参考文献		67

附录 A.....	71
A.1 电机辨识多组数据自动频域分析 matlab 程序.....	71
A.2 ARC 仿真参数定义 C 程序.....	73
A.3 二叉树路径规划算法程序实现.....	77

插图和附表清单

插图清单

图 1.1	工厂全自动搬运机器人.....	1
图 1.2	排爆机器人.....	1
图 1.3	五类轮式移动机器人结构示意图.....	4
图 1.4	WMR 模型分解图.....	5
图 1.5	直接线性反馈自适应控制框架.....	7
图 1.6	轮式移动机器人的自适应模糊滑模控制框架.....	9
图 1.7	人工势能法示意图.....	9
图 1.8	目标点、障碍物一维示意图.....	10
图 2.1	NI Robotics Starter Kit 2.0	14
图 2.2	NI Single Board 9632	15
图 2.3	平台整体硬件联系图.....	15
图 2.4	图像坐标系.....	16
图 2.5	图像坐标系与摄像机坐标系.....	17
图 2.6	摄像机坐标转换关系图.....	18
图 2.7	棋盘格子标定图像.....	20
图 2.8	车体示意图.....	24
图 2.9	轮轴上力矩传递图.....	26
图 2.10	电机模型.....	28
图 2.11	电机特性曲线.....	28
图 2.12	稳态直流电机 T-S 图.....	29
图 2.13	电机频域响应曲线.....	30
图 3.1	频率响应对比.....	36
图 3.2	一般闭环控制系统.....	37
图 3.3	传统闭环系统 (a)、不确定系闭环系统 (b)	39
图 3.4	权重函数闭环系统方框图.....	40
图 3.5	速度控制框架.....	40
图 3.6	带权重函数控制框架.....	41
图 3.7	不确定性模型幅频图.....	42
图 3.8	闭环 bode 图.....	42
图 3.9	误差跟踪对比.....	45
图 3.10	机器人位姿控制框架.....	46
图 3.11	机器人跟踪椭圆轨迹.....	47
图 3.12	移动机器人摄像头坐标系.....	48
图 4.1	系统辨识影响因素.....	49
图 4.2	自适应控制框架.....	51
图 4.3	自适应鲁棒控制.....	52
图 4.4	无负载角速度跟踪误差对比图.....	57
图 4.5	有负载角速度跟踪误差对比图.....	57
图 4.6	三层路径跟踪控制系统.....	58

图 4.7 二叉树路径规划结果.....	59
图 4.8 二叉树基准路径图.....	59
图 4.9 内外切线图.....	60
图 4.10 障碍物判断.....	60
图 4.11 二叉树路径规划程序流程图.....	61
图 4.12 目标点与障碍物.....	62
图 4.13 二叉树路径跟踪实验.....	63

附表清单

表 1.1 WMR 的一般运动学模型.....	5
表 2.1 系统模型参数.....	31
表 3.1 Z-N 振荡参数调整公式.....	44
表 3.2 实验性能参数.....	44
表 4.1 实验性能参数.....	56
表 4.2 位置误差实验结果.....	64

第 1 章 绪论

1.1 课题研究背景

移动机器人是一个集环境检测、运动决策与规划、行为控制与执行等多功能于一体的综合系统。它涉及了多学科的研究领域，集成了信号处理、机械设计、计算机工程、电子工程、自动化控制工程以及人工智能等多学科的研究成果，可作为机电一体化的最高成就代表之一，是目前科学研究最热门的领域之一^[1]。

移动机器人的研究最早可以追溯到上世纪 60 年代末期。在 1966 年至 1972 年中，斯坦福研究院研发出了命名为 **shakey** 的移动机器人。随着移动机器人周边技术的急速发展与成熟，其应用范围出现大的拓展，由最先涉及的工业、农业、医疗、服务等常规行业的应用拓展到了城市安全、国防和空间探测领域等有害危险的场合中的应用，人们因此可以从重复的体力劳动和危险的工作环境中解放出来。轮式移动机器人由于其高的灵活性，在工业柔性制造系统、柔性装配系统、自动化搬运系统和物流仓储系统中成为难以取代的重要部分，使得离散型物流系统能够更有效地连续化作业，如图 1.1。在军事上，战场上出现了各种先进的无人侦察机器人、拆弹排爆机器人等类型机器人。美国 **Remotec** 技术公司设计开发的自主排爆机器人 **ANDROS F6-A** 和美国机器人系统技术公司研制的保安机器人 **MDARS-E**，可以取代人类进行侦察和爆炸物处理等危险工作，有效减少人员伤亡，如图 1.2。在民用上，轮式移动机器人在抢险救灾、监控安保、家居服务



图 1.1 工厂全自动搬运机器人



图 1.2 排爆机器人

等场合给予人类的帮助越为重要。在空间探测上，美国移动机器人也走在前列，其中 2004 年成功登陆火星的“机遇号”和“勇气号”火星车可以作为轮式结构的典型代表。此外，在医疗上，移动机器人可以帮助人们完成助老助残等任务。

因此,更为广泛的应用及更高的要求促使移动机器人成为全世界研究机构的研究热点。

机器人在不同领域相对传统人工表现出来的强大优势也吸引了大量的科学工作者的关注。移动机器人可以以不同的方式进行移动,包括行走、跳跃、滑动和滚动等等。一般来说,移动机器人采用轮式结构或腿式结构进行移动。腿式结构可以使机器人跨越凹坑和障碍物,在极端复杂的情况移动,但是腿式机器人结构复杂,自由度大幅增加^[2]。

相对于腿式移动机器人,轮式移动机器人(Wheeled Mobile Robot, WMR)具有操作快捷、性能稳定、机构和控制相对简单和能量利用效率高等特点,是用途最广泛的一类机器人^[2]。

运动控制是指轮式移动机器人能够在指定的环境中达到一定的运动要求,这也是移动机器人控制系统的最基本也是最核心的功能。随着应用环境的扩展,要求机器人的智能化程度越高,能够在充满动态或静态的障碍物环境中正常工作,同时在特定的任务中达到的运动精度也逐渐提高。这些吸引了大量研究人员从事运动控制系统的研究工作,主要可分为两方面,一方面是运动学和动力学上的运动控制,另一方面是在不同环境中的路径规划。

由于含有非完整约束的轮式移动机器人具有运动学模型和动力学模型,它的运动控制问题包括运动学控制问题和动力学控制问题。根据应用的需要,轮式移动机器人往往需要在更为复杂的环境下工作,而现实环境中具有很多未知信息,这将会带来很多的不确定性和复杂性。机器人需要在未知环境下工作,需要有很好的抵抗未知干扰的能力,而相应算法的实现要得到机器人本身处理速度的支持,因此也给轮式移动机器人的控制系统在抗干扰能力和实时性等方面带来了巨大的挑战。在运动学方面,轮式移动机器人受到非完整约束的影响,运动控制难度较大,并且现实实际情况中轮子与地面往往会发生滑动,会造成运动学模型的不确定性,小的滑动会影响移动机器人的轨迹跟踪精度,大的滑动则可能导致系统在运动中失稳,所以在运动学的控制律设计方面需要较强的鲁棒性。在动力学方面,轮式移动机器人是一个多变量、强耦合、参数时变的非线性对象,难以对其进行高性能位置跟踪的控制。之前,在性能要求高的轮式移动机器人运动系统中,机器人的运动控制主要采用基于精确模型的反馈控制律方法。在实际应用中,由

于测量和建模不精确,加上参数时变和负载扰动等因素的影响,很难获得精确完备的系统模型,因此轮式移动机器人的轨迹跟踪控制性能的提高受到很大挑战。

路径规划是指规划出一条移动机器人从出发点到终点的与周围环境无碰撞路径,评价路径的指标往往包括了路程距离、时间和能量等项目,而路程距离是最重要的一项评价指标。根据环境信息的获取情况,可以分为离线和在线两种路径规划方法。离线路径规划是指机器人所在环境都是事先已知的,包括障碍物的位置、大小及移动轨迹。而一般情况下,环境的全部信息是不能都提前已知的,需要移动机器人在环境运行中不断通过传感器的信息反馈获得,这就是在线路径规划或局部路径规划^[3]。

本文首先针对实际具体的轮式移动机器人平台进行了完整的运动学和动力学建模,其中动力学模型涵盖了驱动直流电机的电磁力学方程、车轮扭矩传递方程和车体动力学模型。通过实际的参数辨识对模型进行解耦简化,并在动力学上先后设计了 μ -synthesis 控制器和自适应鲁棒控制,两者控制器能使具体一定硬件限制的机器人平台分别在不同工作环境中的控制精度得到大幅度提高,控制性能得到提升,当然两控制器的具体适合的使用场景有不同侧重。随后,首先通过动力学上的速度控制实验验证了两者控制器对速度控制精度的提高,然后针对轨迹跟踪和在含有障碍物环境下的路径规划两个具体应用实例中设计两套控制框架,分别融合 μ -synthesis 控制器与自适应鲁棒控制器 (ARC),体现了控制器在实际控制环境中的优势。其中在路径规划方面设计了改进的二叉树路径规划算法,该算法得到的路径由直线和圆弧组成,能够在已知环境下规划出最短路径。上层的二叉树路径规划算法与底层运动学动力学上的先进控制方法组成了整个轮式移动机器人控制系统,运动学动力学上的控制方法减少机器人运动误差,保证了上层路径规划的准确运行,两者相辅相成,为移动机器人运动控制提供了一个提高精度的整体控制策略。

1.2 国内外研究现状

针对轮式移动机器人这个典型的非完整约束系统,国内外对其运动控制系统进行了大量的科学研究,并呈现逐步升温状态,尤其在轮式移动机器人的非完整约束特性下的运动学动力学建模、先进控制策略与路径规划等相关方面。

1.2.1 移动机器人的运动学和动力学建模

轮式移动机器人结构相对于腿式机器人简单,但是其本身作为一个典型的非完整约束系统,吸引了不少科学家对其建模研究^[4-13]。非完整约束对 WMR 运动学和动力学都有体现,不少学者针对运动学、动力学进行建模研究,也有学者考虑了模型中不同类型轮子的力学特性及影响^{[5][7-12]}。

早期 Guy Campion 等学者对不同类型 WMR 的运动学和动力学进行了分析^[4]。通过大量的统计,根据轮式移动机器人不同的轮式结构而具有的运动自由度 DOM (Degree of Mobility) 和舵性自由度 DOS (Degree of Steerability) 可分为五大类机器人,以 (δ_m, δ_s) 的形式可分为写为 (3,0), (2,0), (2,1), (1,1) 和 (1,2) 类型,其中 δ_m 代表的是运动自由度数, δ_s 代表的是舵性自由度,如图 1.3。

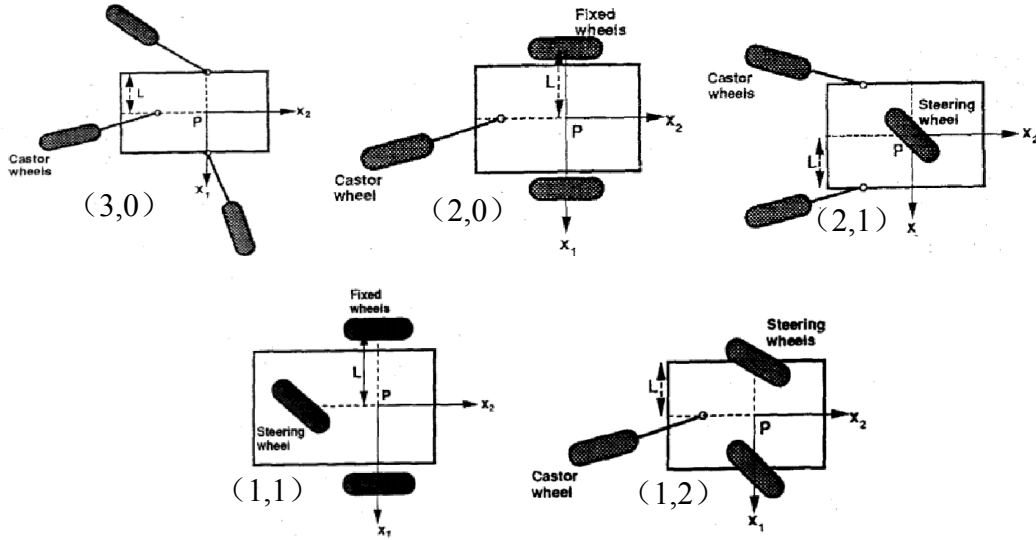


图 1.3 五类轮式移动机器人结构示意图

Guy Campion 等学者对这五类机器人的运动学进行了运动学的推导,统一可用公式 (1.1) 代表,如下:

$$\dot{z} = B(z)u \quad (1.1)$$

其中, z 代表 WMR 的位置自由度,包括 x, y, θ 三个平面位姿信息或舵性轮的旋转角度,用 β 表示, $B(z)$ 表示与自由度 z 相关的转换矩阵, u 表示控制输入,其中的 η 代表的是不同类型移动机器人所具有的移动速度,对应于运动自由度, ς 表示的舵性轮的旋转速度,对应于舵性自由度,具体的五类机器人运动学表达式可见表 1.1。

表 1.1 WMR 的一般运动学模型

类型	位姿坐标	运动学模型 $\dot{z} = B(z)u$
(3,0)	x y θ	$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix}$
(2,0)	x y θ	$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\sin \theta & 0 \\ \cos \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}$
(2,1)	x y θ β_{c1}	$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_{c1} \end{pmatrix} = \begin{pmatrix} -\sin(\theta + \beta_{c1}) & 0 & 0 \\ \cos(\theta + \beta_{c1}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \varsigma_1 \end{pmatrix}$
(1,1)	x y θ β_{c3}	$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_{c3} \end{pmatrix} = \begin{pmatrix} -L \sin \theta \sin \beta_{c3} & 0 \\ L \cos \theta \sin \beta_{c3} & 0 \\ \cos \beta_{c3} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \varsigma_1 \end{pmatrix}$
(1,2)	x y θ β_{c1} β_{c2}	$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\beta}_{c1} \\ \dot{\beta}_{c2} \end{pmatrix} = \begin{pmatrix} -L(\sin \beta_{c1} \sin(\theta + \beta_{c2}) + \sin \beta_{c2} \sin(\theta + \beta_{c1})) & 0 & 0 \\ L(\sin \beta_{c1} \cos(\theta + \beta_{c2}) + \sin \beta_{c2} \cos(\theta + \beta_{c1})) & 0 & 0 \\ \sin(\beta_{c2} - \beta_{c1}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \varsigma_1 \\ \varsigma_2 \end{pmatrix}$

以上的五种类型涵盖住了轮式移动机器人的所有类型，本论文的轮式移动机器人作为上述中的 (2,0) 类型。

近年来，为了获得 WMR 的更高控制精度，不少研究者不仅仅关注 WMR 的运动学模型，同时更加关注 WMR 系统运动学与动力学建模的完整统一性。Kozłowski K 将四轮机器人分为了运动学、动力学和驱动模块，并对每个子系统单独建模^[7]。模型框架如图 1.4 所示。

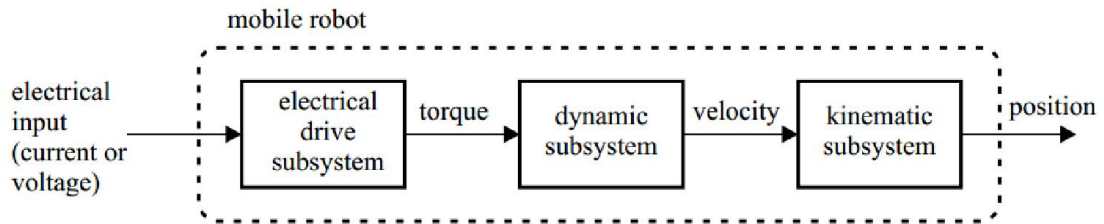


图 1.4 WMR 模型分解图

这种完整性的建模考虑较为全面，理论推导比较严谨，相对单独只针对车身动力学建模考虑了驱动系统电力学，但是未考虑到扭矩在驱动轴上的损失。

1.2.2 移动机器人的运动学和动力学控制

针对轮式移动机器人运动学模型上，已有 pure pursuit^[14]、神经网络控制^[15]、神经-模糊控制^[16]和线性化^[17-18]等方法被用来解决跟踪问题。其中 Y. Kanayama 针对机器人运动学中的非完整约束特性提出了一种稳定跟踪控制律^[19]，将机器人参考位姿 (x_r, y_r, θ_r) 与参考速度 (v_c, w_c) 联系起来，较多地被后面研究者采用，具体控制率公式如下：

$$\begin{pmatrix} v_c \\ w_c \end{pmatrix} = \begin{pmatrix} v_r \cos \theta_e + K_x y_e \\ w_r + v_r (K_y x_e + K_\theta \sin \theta_e) \end{pmatrix} \quad (1.2)$$

其中 x_e, y_e 为车体在世界坐标系中实际位置与参考轨迹的误差， θ_e 为车体实际转角与参考轨迹切线的角度误差， v_r, w_r 为车体设定的一个基础速度值， v_c, w_c 为车体理论速度， K_x, K_y, K_θ 为需待选定的三个正参数。最后 Y. Kanayama 用李雅普诺夫方程证明该控制率的渐进稳定性^[19]。此种方法只考虑了机器人的运动学。

在动力学上，K. Watanabe、A. Saffiotti 等研究人员针对轮式移动机器人在实际运行中所处的不确定环境条件和系统具有的模型不确定特性，采用了不同的模糊控制^[20,21]。虽然模糊控制相对易于实现，并且具有一定的鲁棒性，但是模糊控制有不少缺点：模糊控制缺乏系统理论性，对复杂系统的控制是难以奏效的，并且模糊控制方程的设计往往需要依赖于个人的设计经验，参数的选择也需要反复调试。若要进一步提高控制精度就需要增加量化级数，从而导致规则搜索范围扩大，增加计算负荷，这将大大降低算法的实时性，使其实用性变差。同时滑模控制不能很好解决模型参数不确定性，并且缺乏定性地解决摩擦力的影响。

自适应控制能够在系统运行中去不断地测量系统的性能、状态或参数，进而做出改变控制器的结构、参数等决策或根据自适应律来改变控制作用，以保证系统运行在实时情况下达到优化的控制性能。由于实际轮式移动机器人的动力学是一个多变量、强耦合、参数时变的非线性对象，各式自适应控制被广泛应用到其控制中来^[22-28]。其中 K. Shojaei 提出了一种直接输入输出线性反馈的自适应控制方法，他采用了一种 look-ahead 控制方法^[29] 解决了非完整约束中不能直接线性

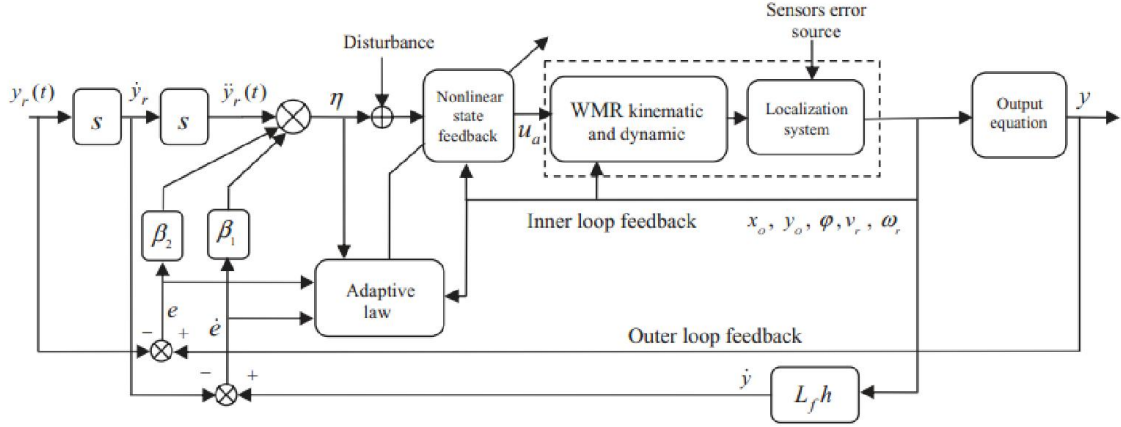


图 1.5 直接线性反馈自适应控制框架

反馈的问题，并且针对由建模误差、轮子滑动、表面摩擦力和定位误差造成的系统不确定性设计一种自适应控制率，控制框架如图 1.5。K. Shojaei 对系统输出量选取的 look-ahead 控制方法如下：

$$y = h(x) = [h_1(q), h_2(q)]^T = [x_o + L \cos \varphi, y_o + L \sin \varphi]^T \quad (1.3)$$

其中， (x_o, y_o, φ) 是移动机器人的位置和姿态信息， L 是根据需要选定的长度。

对输出 y 其进行双重积分得到：

$$\ddot{y} = L_f^2 h(x) + L_q L_f h(x) + L_g L_f h(x) \cdot u_a \quad (1.4)$$

假设系统的状态量 $q = [x_o \ y_o \ \varphi]$ 和 $V = [v(t) \ w(t)]$ （机器人直线速度和角速度）

在实时运行中都能测量到，并且它们及它们的微分都是有界的，可选择控制率如下：

$$\left. \begin{aligned} u_a &= \hat{D}^{-1}(x) \cdot (\eta - L_f^2 h - L_q \hat{L}_f h) \\ \eta &= \ddot{y} + \beta_1 (\dot{y}_r - \dot{y}) + \beta_2 (y_r - y) \\ \hat{\theta} &= \Gamma W^T E_1 \end{aligned} \right\} \quad (1.5)$$

其中 W 是回归矩阵， E_1 是误差信号向量， Γ 是自适应增益参数，为正定对称矩阵， β_1 和 β_2 是对角矩阵，表示线性控制率的微分和比例增益。

自适应控制方法虽然有可以自行调整参数的优点，然而在系统有不确定非线性影响时会造成未知瞬态和失稳可能性，同时 K. Shojaei 并未考虑实际机器人运行中存在大幅值摩擦力影响的作用。

Backstepping(逐步后推，反推)是一种处理不确定性系统的相对系统化的综合控制方法，是将 Lyapunov 函数的选取与控制器的设计相结合的一种回归设计方法。它引入虚拟控制的概念，通过从系统的最低阶次微分方程开始，一步一步设

计满足要求的虚拟控制，最终得出真正的控制律。Backstepping 的控制方法也被广泛运用到移动机器人运动控制上来^[30-32]。Dongkyuong Chwa 根据类似 Backstepping 反馈线性化的思想设计了一种控制器，这种控制器以机器人运动学和动力学线性化的级联形式以得到简单模块化的控制结构。首先，从车体位置和姿态坐标到车体前进直线速度和转向角度的虚指令被设计出来，如式 1.6。然后，实际扭矩控制输入被设计出来使轮式移动机器人的实际前进直线速度和实际转向角度追随上级产生的虚指令。由于系统动力学方程可写为 $H(q)\dot{z} + F(q, z) = \tau$ ，所以该级的控制方程如式 1.7。最后一个稳定性分析说明车体跟踪误差（位置和转向角度）是全局渐进有界的，并且这个有界的范围与选取的控制参数有关。

$$\begin{cases} v_{pseudo} = \cos(\theta_c)X + \sin(\theta_c)Y \\ \theta_{pseudo} = \theta^* \text{ or } \theta^* + \pi \end{cases} \quad (1.6)$$

$$\begin{cases} \tau = F(q, z) + H(q)u \\ u = -Az + z_c \\ z_c = [k_v v_{pseudo} \quad b \sin(\theta_c - \theta_{pseudo})] \end{cases} \quad (1.7)$$

反推法技术采用级联的设计思想，但其控制器的结构和设计过程都很复杂，因此应用到实际中有一定的难度，且缺乏对低频摩擦力与模型参数不确定的考虑。

滑模控制(sliding mode control, SMC)本质上是一类特殊的非线性控制，控制的不连续性即为其非线性表现，它可称为变结构控制。滑模模态本身是可以设计的并且和对象参数及扰动无关，因此具有了快速响应、对应参数变化及扰动不灵敏、无需系统在线辨识、物理实现简单等优点。由于滑模控制能够克服系统的不确定性，对于干扰和未建模动态具有很强的鲁棒性，特别是对非线性系统的控制具有不错的控制效果，很多类型的滑模控制也应用到轮式移动机器人的运动控制中来^[33-36]。Ying-Chieh Yeh 针对轮式移动机器人动力学设计了一种自适应模糊滑模控制器 (AFSMDC)。这种控制方法可以有效地减少外部干扰和系统不确定性的影响以达到好的轨迹跟踪效果，控制框架如图 1.6。

滑模控制方法具有设计过程简便、较好的鲁棒性、快速响应等优点。但是，在实际中这种方法容易产生震荡的控制信号，这在实际应用中是很危险的^[38]，

在最近的三年里，国内对轮式移动机器人的轨迹跟踪问题也相继有了很多的研究。张和胡两学者对轮式移动机器人轨迹跟踪控制的最新研究特点和现状进行了分析^[37,38]。对跟踪轨迹控制方法的研究，有些学者提出了神经网络自适应控制

和一种基于迭代学习的移动机器人轨迹跟踪控制方法^[39,40]；也有学者提出了一些反演控制方法^[41,42]；最多研究的轨迹跟踪控制策略是滑模控制，国内近三年提出了很多针对轮式移动机器人运动学或动力学的滑模控制方法^[43-45]。但是国内设计的控制方法大多数是通过仿真来验证，较少运用到实际系统中去，而且针对的动力学模型都是大量简化而无相关建模说明，没有同时考虑到实际系统的摩擦力、模型参数不确定性。

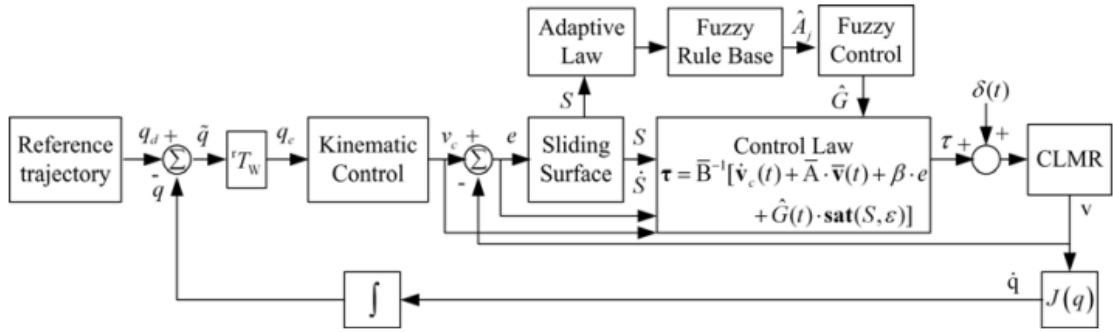


图 1.6 轮式移动机器人的自适应模糊滑模控制框架

1.2.3 移动机器人的路径规划

在机器人路径规划方面，人工势场法^[46-50]和网格法^[51]是较早被研究并被广泛使用的两种路径规划方法。两种方法简单直观，在日益复杂的环境下却都有自己的缺陷。

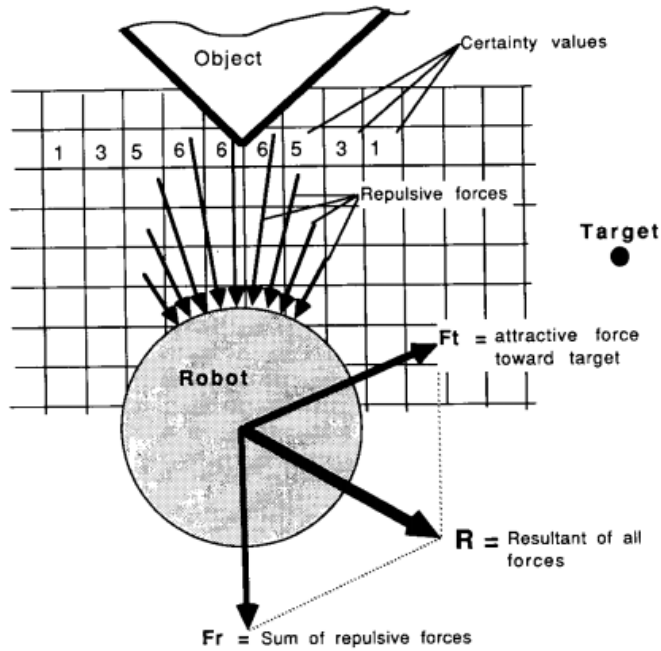


图 1.7 人工势场法示意图^[46]

人工势场法假设目标点给予机器人一个吸引力，吸引力的大小与机器人和目标点的距离有关，同时周围环境中的障碍物给予机器人排斥力，排斥力与机器人和障碍物的距离有关，最后通过吸引力和排斥力的矢量合成，得到机器人不同位置的一个运动方向，最终规划出机器人到达目标点这样一个轨迹图。传统人工势场法设定吸引势能为：

$$U_{att}(q) = \frac{1}{2} \xi \rho^2(q, q_{goal}) \quad (1.13)$$

其中 ξ 一个正随机参数， $\rho^2(q, q_{goal}) = (q_{goal} - q)^2$ 表示的是机器人位置 q 与目标点 q_{goal} 距离的平方，吸引力即为吸引势能的负梯度：

$$F_{att}(q) = -\nabla U_{att}(q) = \xi(q_{goal} - q) \quad (1.14)$$

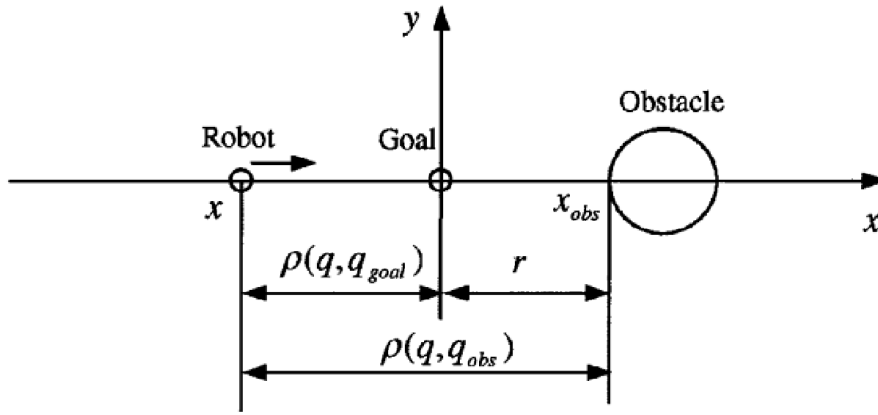


图 1.8 目标点、障碍物一维示意图

环境中的障碍物对机器人产生的排斥引力可表示为：

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right), & \text{if } \rho(q, q_{obs}) \leq \rho_0 \\ 0 & \text{if } \rho(q, q_{obs}) > \rho_0 \end{cases} \quad (1.15)$$

其中 η 为与障碍物位置相关的一个正参数， $\rho(q, q_{obs})$ 表示的是机器人与障碍物的距离， ρ_0 表示障碍物与机器人的考虑距离范围，距离小于 ρ_0 则表示障碍物对机器人有排斥力。排斥力为排斥势能的负梯度：

$$F_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} \eta \left(\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right) \\ \frac{1}{\rho^2(q, q_{obs})} \nabla \rho(q, q_{obs}), & \text{if } \rho(q, q_{obs}) \leq \rho_0 \\ 0 & \text{if } \rho(q, q_{obs}) > \rho_0 \end{cases} \quad (1.16)$$

最后通过两者合力得到机器人最终一个牵引力，即机器人当前运动方向，合力方程为：

$$F_{rep} = F_{att} + F_{rep} \quad (1.17)$$

考虑到人工势能法存在局部极小点、无法计算紧邻障碍物之间的通道和与障碍物发生相撞等情况，S.S.Ge 根据机器人与目标点和障碍物两者的相对距离定义了一种新的排斥力方程^[48]。Ahmad A.Masoud 把势能法结合到机器人底层驱动，进一步规划了 WMR 在路径中的速度^[49]。之前的势能法针对的都是静态环境，规划出来的结果也往往是机器人的运动方向，L. Huang 提出了针对目标点和障碍物运动情况下的人工势场法，能同时规划出机器人的运动方向和速度^[50]。

网格法是通过给地图划分网格并按照一定规律赋值，划分方法与车体当前的位置和目标点位置、车体当前的位置和障碍物位置的两项距离值有关，然后通过网格的数值大小比较得到前进的方向^[51]。

1.3 课题研究意义及研究内容

1.3.1 课题的研究意义

随着轮式移动机器人在工业、农业、医疗、服务、国防和空间探测等众多领域的应用，所要求运动控制精度也越来越高。一般机器人的运动控制系统包含了运动学与动力学上的控制两个方面，运动学控制起步较早，已有具有实际应用的一定研究成果，而在动力学上，未知时变的质量和惯量、摩擦力和控制信号的外干扰和本身非线性的强耦合特性降低了轮式移动机器人的轨迹跟踪控制精度。在这之前虽提出了不少控制策略，也在仿真中验证了其控制性能，但是它们有各自的限制，在实际系统中的应用方法也未具体说明。

此外在路径规划方面上，最常用的两种方法存在各自的局限性。势能法存在计算路径时往往会有局部势能点，在大障碍物面前无法计算出到达终点的轨迹等缺点。而网格法对复杂地图划分困难，随着所需精度增加其计算量将剧增，这也限制了该方法的研究。所以需要一种直观有效的方法用来规划机器人的轨迹。

综上所述，本文将针对 WMR 不同的工作要求在动力学上设计高性能的控制器，并且根据工作与实验场景设计涵盖运动学、动力学以及更上层的路径规划的

运动控制系统, 综合解决实际情况中遇到的各类干扰的影响, 达到很高的一个控制精度, 从而适应移动机器人更为广泛的运用。

1.3.2 课题的研究内容

本课题针对轮式移动机器人实验平台的不同工作要求设计了两种控制器, 即 μ -synthesis 控制器和自适应鲁棒控制器, 并且为 μ -synthesis 控制器跟踪实际黑线的应用中设计了两层控制框架, 为自适应鲁棒控制器在含有障碍物环境中的自主运行情景中设计了三层控制框架, 其中上层部署了改进的二叉树路径规划算法。主要内容包含以下部分:

第一部分为机器人实验平台的具体搭建工作, 其目的为搭建一个最为普遍的三轮机器人, 并且为后续设计的控制系统附加安装相应的传感器, 为其控制器提供所需的速度位置等信号。其中在本实验平台中, 通过使用 CCD 摄像头去捕捉实验室设定的环境特征, 如地面黑线轨迹, 在通过几何计算得到与实际轨迹的位姿误差以获得移动机器人在位置和角度上的信息。

第二部分内容为对三轮机器人的一个整体性的建模, 该建模涵盖了机器人的车身动力学和直流电机电磁力学。在完成整体性建模之后, 针对不同的参数使用了不同的辨识方法, 有物理测量法、部分最小二乘法辨识和整体最小二乘法辨识, 最后得到带有实际参数的整体模型。

第三部分内容为移动机器人动力学上的控制器设计, 在控制器设计之前对已辨识得到的机器人模型进行相应的分析, 了解到由于科氏力和离心力导致的耦合项在系统中的影响较小, 进行忽略并相乘以解耦矩阵进行简化, 得到的解耦简化的模型与实际系统的频域响应情况进行对比, 发现在低频存在摩擦力的影响和在高频有模型参数不确定性的影响。后续设计的 μ -synthesis 控制器在系统能够达到的频域范围内, 兼容性地解决了两个频域段存在问题的影响。

第四部分内容为自适应鲁棒控制器 (ARC) 的设计, 该控制器的设计相对于上部分 μ -synthesis 控制器性能更侧重于在线的模型参数估计, 同时具有快速的瞬态响应性能。该控制器融合了自适应控制和鲁棒控制的优点, 具有很高的控制性能。其中在避障实验中设计了三层控制框架, 涵盖了轮式移动机器人系统的路径规划、运动学与动力学控制。在最上层使用了改进的二叉树路径规划算法, 底层

的自适应鲁棒控制保证了算法的实际应用性，各层相辅相成，达到避障路径最短的效果。

1.4 本章小结

本章以大量国内外文献为基础，分类阐述了轮式移动机器人在动力学与运动建模、动力学与运动学先进控制策略与路径规划等方面的研究背景及研究现状，并且结合各类机器人研究的特点并指出不足之处。随后简要说明了本课题的研究意义，概述了论文的主要研究内容及架构。

第2章 机器人平台系统搭建及建模

2.1 实验平台系统

实验平台是课题研究的基础，搭建一个比较正确的完善的科学的实验平台，才能模拟实际轮式移动机器人运行状况，对其进行后续的建模分析与控制才显得更有意义。

2.1.1 基本架构及系统连接设计

本课题实验平台的基础架构为NI公司开发的NI Labview Robotics Starter Kit 2.0。该机器人开发包为一个典型的三轮移动机器人，平台包括了易拆卸的硬件支撑框架、驱动轮及万向支撑轮、双电机驱动器、直流电机、光学编码器和超声波传感器等，如图2.1所示。车体三轮中的前两轮分别由两个直流电机驱动，通过两轮差速旋转从而带动车体进行相应的直线和旋转运动，第三个轮为万向轮，起到辅助支撑的作用。三轮中相应的力学特性因此也不同，两驱动轮与地面产生的静摩擦力给车体是驱动力，而支撑轮上的摩擦力为阻力。

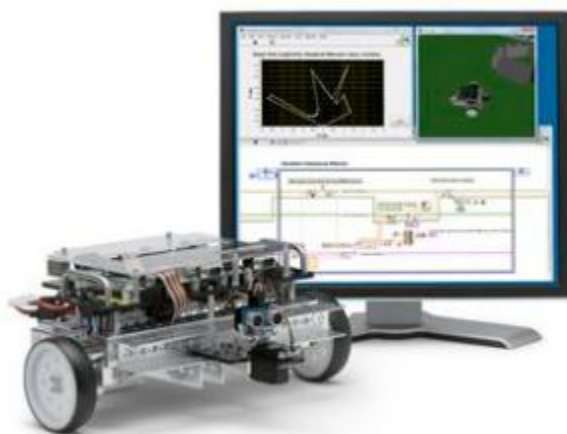


图 2.1 NI Robotics Starter Kit 2.0

该机器人车体上承载的电路硬件是NI Single Board 9632，采用的是图像化编程语言 Labview，能够为本课题研究中的算法设计、原型建立和算法验证同时提供软硬件的支持，如图2.2所示。该机器人系统可以建立上位机远程电脑与下位机 Single Board 9632 的联系，通过便捷的图形化编程可将在上位机的 Labview 程序直接导入到下位机，利于实际嵌入式系统的开发。在 NI Single Board 9632 硬

件上，集成了实时模块和 FPGA 模块，同时还预留了一些数字和模拟 I/O 接口，便于系统的扩展。实时模块主要是编写系统的一些较复杂的主程序，而在 FPGA 模块上主要用来采集各传感器信号，并对其简单处理与融合，并且能达到一个较快的处理速度。

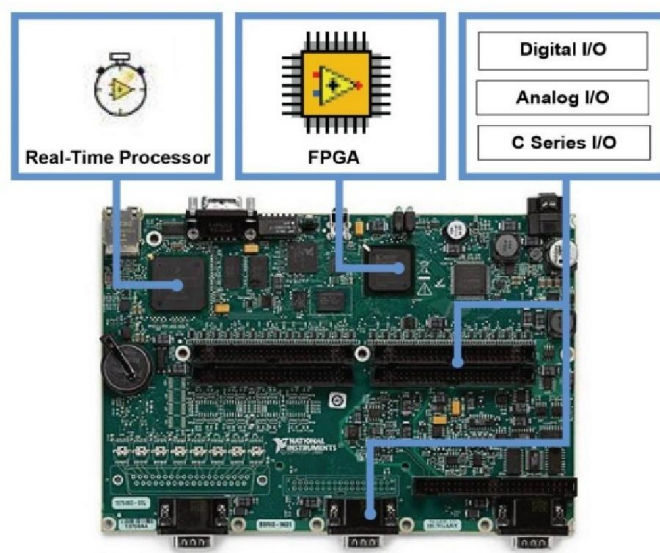


图 2.2 NI Single Board 9632

NI Robotics Starter Kit 2.0 只含光电编码器和超声波传感器，光电编码器一周线圈数仅为 100CPR，每周脉冲数仅为 400CPR，同时超声波传感器一个循环周期只能测量一个方向与周围物体的距离值，并且测量精度受周围物体表面质量影响很大，用来获取机器人的绝对位置误差将很大。可知，机器人平台本身附带的传感器精度和性能有限。

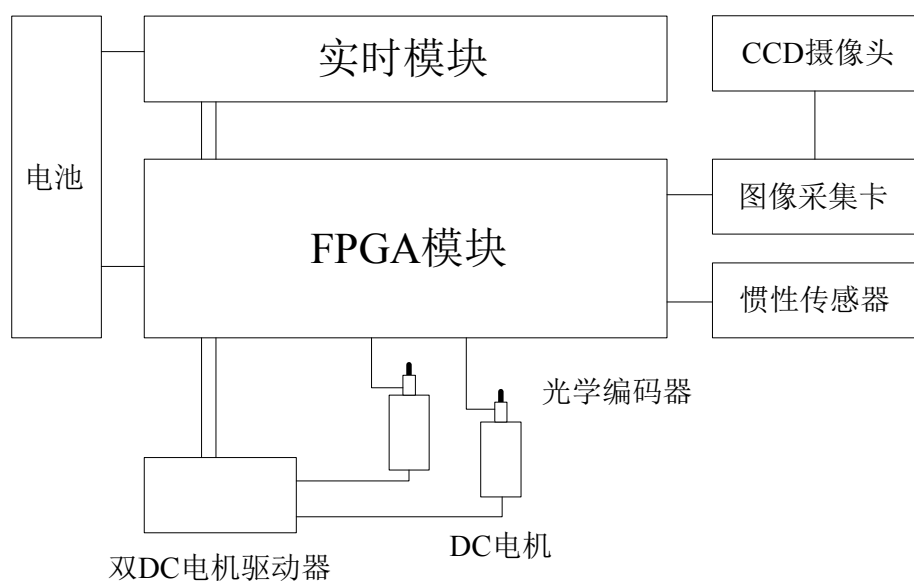


图 2.3 平台整体硬件联系图

为了更好地获得车体绝对位置和姿态参数,为后续控制系统的研究开发提供 WMR 位置和速度上的信息,一个 CCD 摄像头被安装在车体前端获取一定的环境图像特征,通过识别已在实验室环境中设定的点与线位置信息并进行反推以便用来获取轮式移动机器人本身的绝对位姿信息。机器人角速度将会是后面的建模控制过程中的一个较大耦合量,影响到控制系统的性能,本课题附加了一个六自由度惯性传感器 ADIS16375 用来检测车体的一个实际转角速度,其角速度分辨率为 $0.013108^{\circ}/s$ 。系统中各硬件的联系关系如图 2.3 所示。

2.1.2 传感器标定

搭建过程中在已有机器人平台上额外附加了 CCD 摄像头传感器和 ADIS16375 六自由度惯性传感器,其中 ADIS16375 由 ADI 公司提供的标定模板进行标定,而 CCD 摄像头传感器的内部参数需要自己标定获得。

CCD 摄像机一般都为针孔模型,并且采用的摄像机为普通广角,故忽略摄像头畸变的情况。摄像头模型中,涉及到三个坐标系,即图像坐标系、摄像机坐标系和世界坐标系。图像坐标系可分为两种,见图 2.4,一种坐标系中坐标是以

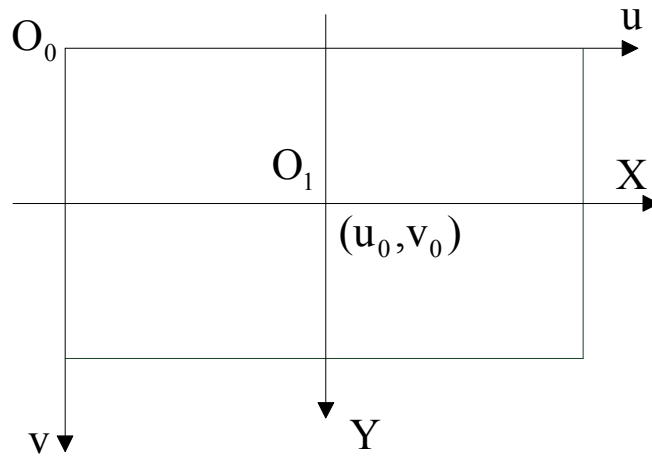


图 2.4 图像坐标系

像素为单位,而另一种坐标系中是以毫米为单位。摄像机的图像信息都是存储给像素点的像素值,为了方便与后面世界坐标系的联系,同时建立了以毫米为单位的坐标系。在以毫米为单位的坐标系中,原点 O_1 定义在摄像机光轴与图像平面的交点,一般情况下为图像中心处,设以像素值为单位的坐标系中的每一个像素在摄像机坐标系中 X 轴与 Y 轴方向上的物理尺寸分别为 dX 、 dY ,则两坐标系之间的转换关系为:

$$\begin{aligned} u &= \frac{X}{dX} + u_0 \\ v &= \frac{Y}{dY} + v_0 \end{aligned} \quad (2.1)$$

其中, (u, v) 为图像坐标系中的像素坐标值, (X, Y) 为图像坐标系中的毫米坐标值。

该关系式用齐次方程可表示为:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dX} & 0 & u_0 \\ 0 & \frac{1}{dY} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.2)$$

同样, 两坐标系的逆关系式可表示为:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} dX & 0 & -u_0 dX \\ 0 & dY & -v_0 dY \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.3)$$

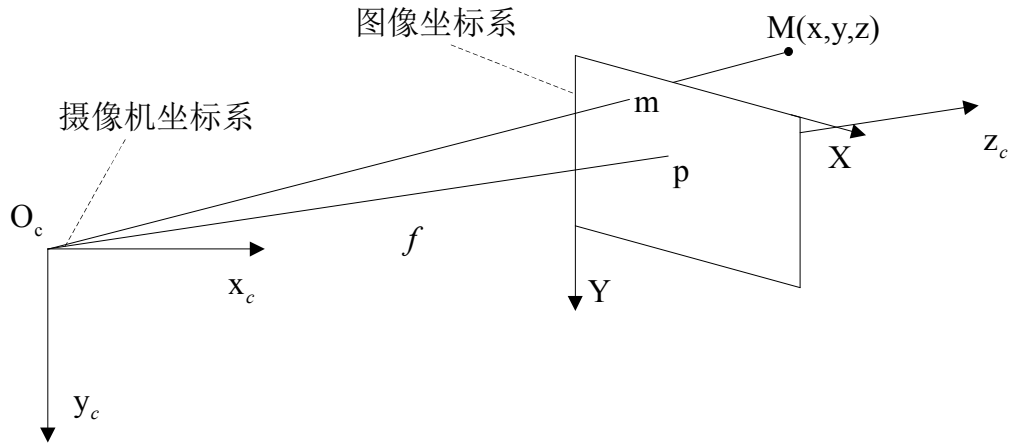


图 2.5 图像坐标系与摄像机坐标系

图像坐标系与摄像机坐标系的关系可见图 2.5, 假设 M 点在摄像机坐标值为 (x, y, z) , 该点 M 在图像坐标系中的对应点为 $m(X, Y)$, 图像坐标系与摄像机坐标系的绝对距离为焦距 f , 根据图中的比例关系有:

$$\begin{aligned} X &= \frac{fx}{z} \\ Y &= \frac{fy}{z} \end{aligned} \quad (2.4)$$

使用齐次方程表示图像坐标系与摄像机坐标系的关系如下:

$$s \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \bar{\mathbf{P}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.5)$$

其中， $s=z$ ，为一比例因子， $\mathbf{P}$ 为透视投影矩阵。

摄像机坐标和世界坐标系之间关系为刚性转换，可表示如下：

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{R}} & t \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.6)$$

结合式 2.3、式 2.5 和式 2.6，得到线性摄像机整体成像模型方程如下，转换关系图可见图 2.6：

$$\begin{aligned} s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} \frac{1}{dX} & 0 & u_0 \\ 0 & \frac{1}{dY} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{R}} & t \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \alpha & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{R}} & t \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \overline{\mathbf{A}\mathbf{M}_1\mathbf{X}_w} = \overline{\mathbf{M}\mathbf{X}_w} \end{aligned} \quad (2.7)$$

其中 $\alpha = \frac{f}{dX}$ 为图像坐标系中 u 轴上尺度因子，或称为 u 轴上归一化焦距； $\beta = \frac{f}{dY}$

为图像坐标系中 v 轴上尺度因子，或称为 v 轴上归一化焦距； $\mathbf{M}$ 为 3×4 矩阵，称为投影矩阵； $\mathbf{A}$ 由 α 、 β 、 u_0 、 v_0 决定，只与摄像机内部参数有关，称为内部参数； $\mathbf{M}_1$ 由摄像机相对于世界坐标系的方位决定，称为外部参数。摄像机的内部参数往往是未知的，也是接下来需要通过摄像头标定获得的参数。

$$\begin{pmatrix} \text{2维点矩阵} \\ 3 \times 1 \end{pmatrix} = \begin{pmatrix} \text{物理图像坐标系} \rightarrow \text{像素图像坐标系} \\ \text{转换矩阵} \\ 3 \times 3 \end{pmatrix} \begin{pmatrix} \text{摄像机坐标系} \rightarrow \text{物理图像坐标系} \\ \text{转换矩阵} \\ 3 \times 4 \end{pmatrix} \begin{pmatrix} \text{世界坐标系} \rightarrow \text{摄像机坐标系} \\ \text{转换矩阵} \\ 4 \times 4 \end{pmatrix} \begin{pmatrix} \text{3维点矩阵} \\ 4 \times 1 \end{pmatrix}$$

图 2.6 摄像机坐标转换关系图

摄像机标定的方法有很多，包括摄影测量标定和自标定等等方法，不过传统

的标定方法需要价格贵重的专门标定仪器，本课题采用的是张正友标定法^[52]，该方法只需摄像机观察不同方位的2维棋牌格子标定板。该标定板的制作也非常简便，可打印出来一个棋盘格子样板黏贴到一平整平面即可，如图2.7所示。

不失一般性，假设在标定环境世界坐标系为 $Z=0$ ，并将转换矩阵 $\mathbf{R}$ 的第 i 列向量用 r_i 表示，则等式2.7可写为：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \bar{A} \begin{bmatrix} \bar{r}_1 & \bar{r}_2 & \bar{r}_3 & \bar{t} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} = \bar{A} \begin{bmatrix} \bar{r}_1 & \bar{r}_2 & \bar{t} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \quad (2.8)$$

用在世界坐标系中的 M 点来进行分析，不过 Z 坐标值等于零，所以 $\tilde{M} = [X_w \ Y_w \ 1]^T$ ，在图像坐标系中用 m 点来表示，有：

$$s\tilde{m} = \bar{H}\tilde{M} \quad (2.9)$$

其中 $\bar{H} = \bar{A} \begin{bmatrix} \bar{r}_1 & \bar{r}_2 & \bar{t} \end{bmatrix}$ ， s 为一比例因子。定义 $\bar{H} = [\bar{h}_1 \ \bar{h}_2 \ \bar{h}_3]$ ，有：

$$\begin{bmatrix} \bar{h}_1 & \bar{h}_2 & \bar{h}_3 \end{bmatrix} = \bar{A} \begin{bmatrix} \bar{r}_1 & \bar{r}_2 & \bar{t} \end{bmatrix} \quad (2.10)$$

在等式2.10中，由于向量 $\bar{r}_1$ 和 $\bar{r}_2$ 是正交的，有：

$$\bar{h}_1(\bar{A}^T)^{-1}\bar{A}^{-1}\bar{h}_2 = 0 \quad (2.11)$$

$$\bar{h}_1(\bar{A}^T)^{-1}\bar{A}^{-1}\bar{h}_1 = \bar{h}_2(\bar{A}^T)^{-1}\bar{A}^{-1}\bar{h}_2 \quad (2.12)$$

由于整个转换有8个自由度，外部参数有6个，其中包括3个旋转参数和3个平移参数。等式2.11和2.12就是摄像机内部参数的两个约束方程。

为了获得摄像机内外部参数，这里设：

$$B = (\bar{A}^T)^{-1}\bar{A}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{c}{\alpha^2\beta} & \frac{cv_0 - u_0\beta}{\alpha^2\beta} \\ -\frac{c}{\alpha^2\beta} & \frac{c^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{cv_0 - u_0\beta}{\alpha^2\beta} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(cv_0 - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (2.13)$$

由于 B 是对称矩阵，使用一个6维向量可定义如下：

$$\mathbf{b} = [B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33}] \quad (2.14)$$

令 $\mathbf{H}$ 矩阵中第 i 个列向量为 $\bar{h}_i = [h_{i1} \ h_{i2} \ h_{i3}]^T$ ，有：

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \quad (2.15)$$

其中 $\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$ ，因此根据等式 2.11 和 2.12 两个一般约束方程可写为：

$$\begin{bmatrix} \mathbf{v}_{ij}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \quad (2.16)$$

通过读取 n 幅不同位置的图像，根据已知的 $\mathbf{H}$ ，便可推导出 $\mathbf{b}$ 矩阵中的内部参数，得到内部参数矩阵 $\mathbf{A}$ 后，可根据等式 2.8 反推出外部矩阵，由于本次标定只为获得内部参数，反推外部参数不在此做讨论。

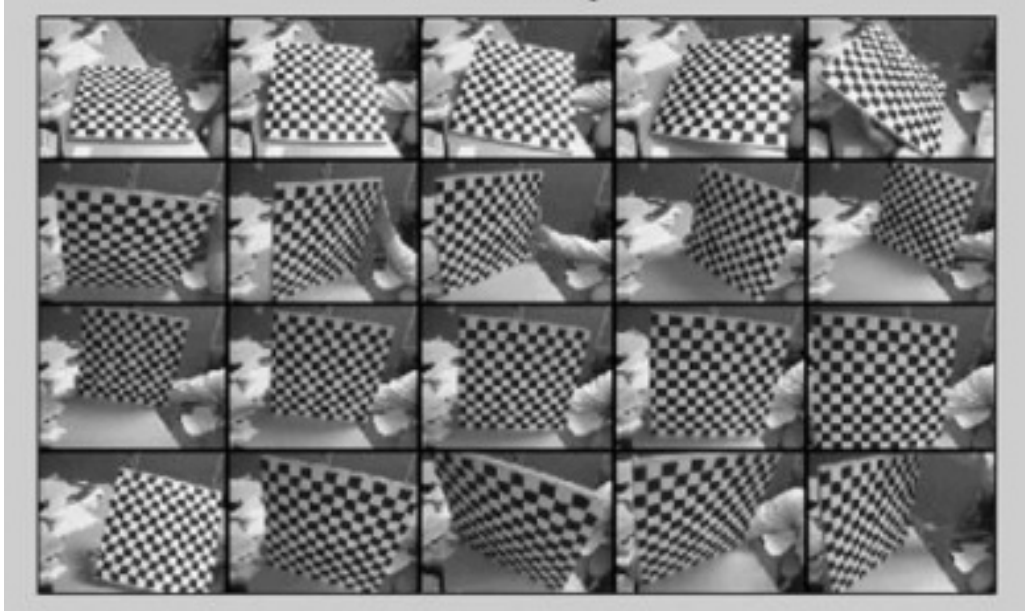


图 2.7 棋盘格子标定图像

一般情况下，读取三幅不同位置的标定图像便可获得所需的内部参数，不过所读的标定图像数量越多，便可通过最大似然法估计出相对准确的摄像机内部参数。本次辨识过程读取了 20 幅不同方位的标定图像，并且由于内部参数中的 c 项太小，在本次标定中不作考虑，因此通过 openCV 下编程获得摄像机内部参数矩阵为：

$$\mathbf{A} = \begin{bmatrix} 1031.0731 & 0 & 374.6596 \\ 0 & 936.3638 & 239.2664 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

本次实验平台中的摄像机安装与标定，是为后续验证动力学上控制器性能的实验做准备工作。在后续动力学控制闭环中，所需的角速度信号由惯性编码器获得，而位姿信号由对摄像头捕捉周围环境特征的图像信号转化成世界坐标位置信息，并进行反推得到，与标定过程为一相似逆过程。

2.2 轮式移动机器人模型推导

2.2.1 运动学模型

本课题的轮式移动机器人是由双直流电机带动两个驱动轮运动，两轮轴在同一直线上但不同轴，通过两轮的同速和差速运转分别完成车体的直线前进和转向运动，如图 2.8 所示。假设车体运行的地面是水平的，车体姿态总共有 5 个运动状态量，包括三个自由度 $q = (x_o, y_o, \varphi)$ 和两个内状态量 θ_r, θ_l ，其中 x_o, y_o 为两轮中点的坐标， φ 为车体对称轴与 X 轴的夹角， θ_r, θ_l 为两轮的转角。

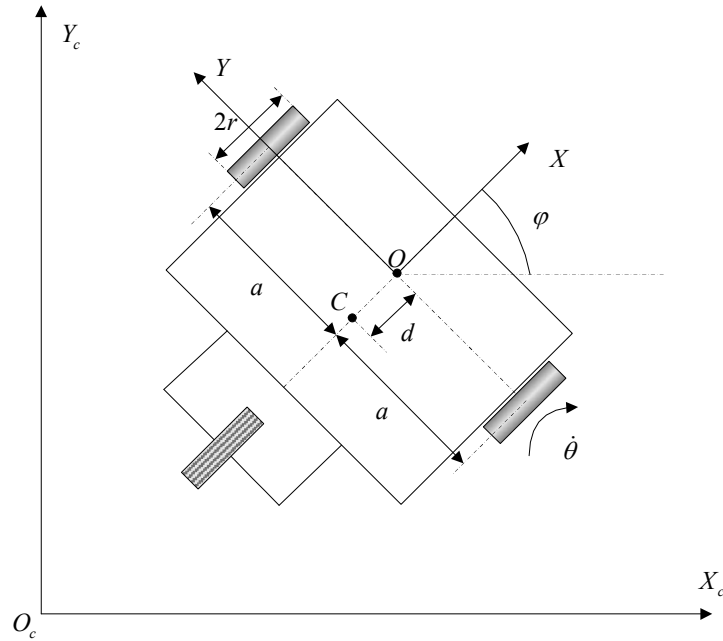


图 2.8 车体示意图

理想情况下，车轮与地面无相对滑动（包括横向与纵向）或者滑动可以忽略不计，则有以下约束条件：

$$\dot{y}_o \cos \varphi - \dot{x}_o \sin \varphi = 0 \quad (2.18)$$

$$\begin{aligned} r\dot{\theta}_r &= \dot{x}_o \cos \varphi + \dot{y}_o \sin \varphi \\ r\dot{\theta}_l &= \dot{x}_o \cos \varphi + \dot{y}_o \sin \varphi \end{aligned} \quad (2.19)$$

其中式 2.18 是轮子横向无滑动约束条件，式 2.19 是纵向无滑动约束条件，本课题对轮子状态不作详细分析，重点讨论式 2.18，其写成矩阵形式如下：

$$A(q)\dot{q} = \begin{bmatrix} -\sin \varphi & \cos \varphi & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_o \\ \dot{y}_o \\ \dot{\varphi} \end{bmatrix} = 0 \quad (2.20)$$

选择 $S(q)$ 为一个 3×2 维的光滑线性独立的矩阵, 与 $A(q)$ 有如下关系:

$$A(q)S(q) = \begin{bmatrix} -\sin \varphi & \cos \varphi & 0 \end{bmatrix} \begin{bmatrix} \cos \varphi & 0 \\ \sin \varphi & 0 \\ 0 & 1 \end{bmatrix} = 0 \quad (2.21)$$

参照图 2.8, 并结合式 2.20 与式 2.21, 可得到车体速度向量与车体广义坐标有如下一个关系, 该方程即为 (2,0) 类型 WMR 系统的运动学方程:

$$\dot{q} = S(q)\mathbf{v} = \begin{bmatrix} \cos \varphi & 0 \\ \sin \varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (2.22)$$

其中 v 、 w 分别为车体的直线速度和转角速度, 这也是后面动力学模型中需要控制的两个速度量。从运动学模型中可以看出, 通过控制 v 、 w 两个速度量从而达到控制车体 $q = (x_o, y_o, \varphi)$ 位姿状态的三个量, 与传统的完整系统有所区别。

2.2.2 非完整约束建模分析

一般而言, 对于一个由 N 个质点组成的离散力学系统, 可以针对具体的点采用坐标 $x_i, y_i, z_i (i=1, 2, \dots, N)$ 和时间 t 的解析方程或非微分方程来表示它的约束, 其一般约束表达形式为:

$$f_\alpha(q_i, t) = 0 \quad \left(\begin{array}{l} \alpha = 1, 2, \dots, d < 3N \\ i = 1, 2, \dots, N \end{array} \right) \quad (2.23)$$

其中 q_i 表示系统的广义坐标。当系统存在此类约束时, 系统不能在每一时刻在空间取任意位置, 也就是在时间 t 对系统位置上的限制。只包含此类约束的系统就为完整约束系统。

从上可知, 本三轮移动机器人平台所具有的约束, 如式 2.18, 不仅仅是广义坐标 q 的线性关系, 而含有 x_o 和 y_o 的微分项, 并且在式中是不可积分的形式出现。这类约束被称之为非完整约束或者非全定约束。为了便于后续分析非完整约束特性对系统动力学上的影响, 将式 2.18 写为一般表达形式, 如下:

$$f_\beta(q_1, q_2, \dots, q_i, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_i, t) = 0 \quad \left(\begin{array}{l} \beta = 1, 2, \dots, g \\ i = 1, 2, \dots, N \end{array} \right) \quad (2.24)$$

其中 g 表示系统含有非完整约束的个数。式 2.24 由于左边式子不能变为只是坐标的函数全微分因而具有不可及积性。若含有该类约束条件时, 对于任意时刻 t ,

系统点可出现在空间的任意地方,但是在特定的空间位置下该点的速度就有限制,不是随意任取的了。如果在式 2.24 中,函数 f_β 相对于 $\dot{q}_1, \dot{q}_2, \dots, \dot{q}_N$ 是线性的,则为线性非完整约束,否则为非线性非完整约束。含有相应约束条件的系统也分别称为线性非完整约束系统和非线性非完整约束系统。

对于一个符合完整系统要求的物理系统,即所有广义坐标都互相独立,则拉格朗日方程成立:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0 \quad (2.25)$$

其中, $L(q, \dot{q}, t)$ 是拉格朗日量, $q = (q_1, q_2, \dots, q_N)$ 是广义坐标, $\dot{q} = (\dot{q}_1, \dot{q}_2, \dots, \dot{q}_N)$ 是广义速度。

在分析力学里,一个动力系统的拉格朗日量,又称为拉格朗日函数,用来表述整个物理系统的动力状态,对于一般经典物理系统,具体形式通常为动能减去势能,其方程表达式为 $L=T-V$ 。在具体分析一个系统的力学特征时,在其拉格朗日量已知情况下,就将拉格朗日量直接代入到拉格朗日方程,进行相应地推导,便计算出系统的动力学方程。

完整约束系统可用第二类拉格朗日方程来描述,而非完整约束系统分析则相对复杂,需要用到更复杂的运动微分方程组表述。Roth 方程是针对一阶线性非完整约束系统带有约束乘子的运动方程。

假设一系统位姿由 n 个广义坐标 $q = (q_1, q_2, \dots, q_n)$ 确定,并且受到如下的一阶线性非完整约束:

$$\sum_{s=1}^n a_{\varepsilon+\beta,s} \dot{q}_s + a_{\varepsilon+\beta,n+1} = 0 \quad \left(\begin{array}{l} \beta=1,2,\dots,g \\ \varepsilon=n-g \end{array} \right) \quad (2.26)$$

其中 $a_{\varepsilon+\beta,s}$ 和 $a_{\varepsilon+\beta,n+1}$ 是两个约束系数,其中 $\varepsilon=n-g$ 是一定值。根据时间 t 不变原理,可以改写为:

$$\sum_{s=1}^n a_{\varepsilon+\beta,s} \delta q_s = 0 \quad \left(\begin{array}{l} \beta=1,2,\dots,g \\ \varepsilon=n-g \end{array} \right) \quad (2.27)$$

上式为对虚位移的约束,由于约束中虚位移不是完全独立的,因此该种系统是不能直接采用第二类拉格朗日方程进行分析。对式 2.27 两边同时乘以不定乘子 λ_β 并且对 β 的所有项相加,有:

$$\sum_{s=1}^n \left(\sum_{\beta=1}^g \lambda_{\beta} a_{\varepsilon+\beta,s} \right) \delta q_s = 0 \quad (2.28)$$

将式 2.28 与传统的 Alembert-Lagrange 方程相加, 有

$$\sum_{s=1}^n \left\{ -\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_s} \right) + \frac{\partial T}{\partial q_s} + Q_s + \sum_{\beta=1}^g \lambda_{\beta} a_{\varepsilon+\beta,s} \right\} \delta q_s = 0 \quad (2.29)$$

在之前的假设中, 式 2.27 是线性的, 所以下面矩阵的秩为 $g = n - \varepsilon$ 。

$$\begin{vmatrix} a_{\varepsilon+1,1}, a_{\varepsilon+1,2}, \cdots, a_{\varepsilon+1,n} \\ a_{\varepsilon+2,1}, a_{\varepsilon+2,2}, \cdots, a_{\varepsilon+2,n} \\ \vdots \\ a_{n,1}, a_{n,2}, \cdots, a_{n,n} \end{vmatrix} \quad (2.30)$$

简化后可得到:

$$\begin{vmatrix} a_{\varepsilon+1,\varepsilon+1}, a_{\varepsilon+1,\varepsilon+2}, \cdots, a_{\varepsilon+1,n} \\ \vdots \\ a_{n,\varepsilon+1}, a_{n,\varepsilon+2}, \cdots, a_{n,n} \end{vmatrix} \quad (2.31)$$

行列式 2.31 的值不为零。由于系统受到式 2.27 的约束, 广义坐标中有 $\varepsilon = n - g$ 个变分广义坐标可作为独立的, 其余的变分 $\delta q_{\varepsilon+1}, \delta q_{\varepsilon+2}, \cdots, \delta q_n$ 则不是独立的, 但可以通过选取不定乘子 λ_{β} 的值使式 2.28 前括号的值为零, 即

$$-\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_{\varepsilon+\gamma}} \right) + \frac{\partial T}{\partial q_{\varepsilon+\gamma}} + Q_{\varepsilon+\gamma} + \sum_{\beta=1}^g \lambda_{\beta} a_{\varepsilon+\beta,\varepsilon+\gamma} = 0 \quad (\gamma = 1, 2, \cdots, g) \quad (2.32)$$

根据行列式 2.31 的值不为零, 并且式 2.32 是线性的, 因此这样选取不定乘子 λ_{β} 是可行的, 这样式 2.29 就剩下 ε 项, 如下:

$$\sum_{\sigma=1}^{\varepsilon} \left\{ -\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_{\sigma}} \right) + \frac{\partial T}{\partial q_{\sigma}} + Q_{\sigma} + \sum_{\beta=1}^g \lambda_{\beta} a_{\sigma+\beta,\sigma} \right\} \delta q_{\sigma} = 0 \quad (2.33)$$

而在等式 2.33 中, $\varepsilon = n - g$ 个变分广义坐标为独立的, 便有:

$$-\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_{\sigma}} \right) + \frac{\partial T}{\partial q_{\sigma}} + Q_{\sigma} + \sum_{\beta=1}^g \lambda_{\beta} a_{\sigma+\beta,\sigma} = 0 \quad (\sigma = 1, 2, \cdots, \varepsilon) \quad (2.34)$$

结合式 2.32 和式 2.34 则有:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_s} \right) - \frac{\partial T}{\partial q_s} = Q_s + \sum_{\beta=1}^g \lambda_{\beta} a_{s+\beta,s} \quad (s = 1, 2, \cdots, n) \quad (2.35)$$

方程 2.35 与约束方程 2.26 可求解 n 个自由度 $q_1, q_2, \dots, q_n$ 和 g 个不定乘子 $\lambda_1, \dots, \lambda_g$, 共 $n+g$ 个未知量。

2.2.3 完整动力学模型

根据式 2.35 中的非完整约束的 Roth 方程, 可推导得到两轮驱动力到车体广义坐标的动力学方程^[22]:

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) = B(q)\tau - A^T(q)\lambda \quad (2.36)$$

其中:

$$M(q) = \begin{bmatrix} m & 0 & md \sin \varphi \\ 0 & m & -md \cos \varphi \\ md \sin \varphi & -md \cos \varphi & I \end{bmatrix}$$

$$V_m(q, \dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\varphi} \cos \varphi \\ 0 & 0 & md\dot{\varphi} \sin \varphi \\ 0 & 0 & 0 \end{bmatrix} \quad B(q) = \frac{1}{r} \begin{bmatrix} \cos \varphi & \cos \varphi \\ \sin \varphi & \sin \varphi \\ b & -b \end{bmatrix}$$

$M(q)$ 是对称正定惯性矩阵, $V_m(q, \dot{q})$ 是离心和科式力矩阵, $F(\dot{q})$ 是表面摩擦力矩阵, $G(q)$ 是重力矩阵, 由于假设机器人在平面上运行, 这项为零, $B(q)$ 是输入转换矩阵, τ 是驱动摩擦力, $A^T(q)$ 是与约束有关的矩阵, λ 是约束力向量。

将式 2.22 进行微分, 有

$$\ddot{q} = \dot{S}(q)\dot{v} + S(q)\ddot{v} \quad (2.37)$$

为了方便处理, 先把式 2.36 中 $F(\dot{q})$ 看成一个干扰项 τ_d , 然后将式 2.22 和式 2.37 代入式 2.36 中, 有:

$$M(q)[\dot{S}(q)\dot{v} + S(q)\ddot{v}] + V_m(q, \dot{q})S(q)\dot{v} + \tau_d = B(q)\tau - A^T(q)\lambda \quad (2.38)$$

在式 2.38 的左右两边同时乘以 $S^T(q)$, 有

$$\begin{aligned} & [S^T(q)M(q)S(q)]\ddot{v} + [S^T(q)M(q)\dot{S}(q) + S^T(q)V_m(q, \dot{q})S(q)]\dot{v} + S^T(q)\tau_d \\ & = S^T(q)B(q)\tau - S^T(q)A^T(q)\lambda \end{aligned} \quad (2.39)$$

由式 2.21 可消除未知的约束力向量 λ , 上式可重新写为:

$$\bar{M}(q)\ddot{v} + \bar{V}_m(q, \dot{v})\dot{v} + \bar{\tau}_d = \bar{B}(q)\tau \quad (2.40)$$

其中

$$\begin{aligned}
\bar{M}(q) &= S^T(q)M(q)S(q) \\
\bar{V}_m(q, \dot{q}) &= S^T(q)M(q)\dot{S}(q) + S^T(q)V_m(q, \dot{q})S(q) \\
\bar{\tau}_d &= S^T(q)\tau_d \\
\bar{B}(q) &= S^T(q)B(q)
\end{aligned} \tag{2.42}$$

最后可通过化简得到车体动力学方程如下：

$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} - \begin{bmatrix} 0 & mdw \\ -mdw & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{a}{r} & -\frac{a}{r} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} F_d \\ \tau_d \end{bmatrix} \tag{2.43}$$

其中 m 是轮式移动机器人总体质量， I 是系统绕 O 点的转动惯量， v 和 w 分别为车体前进和转向速度， a 是两轮距离的一半， τ_1 和 τ_2 分别是左轮和右轮提供的驱动力， F_d 和 τ_d 是由式 2.40 中衍生过来的干扰项。

如图 2.9，从电机输出扭矩到车轮的驱动力矩方程为：

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} - J_w P \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} - b_w P \begin{bmatrix} v \\ w \end{bmatrix}, P = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \tag{2.44}$$

其中， τ_r 和 τ_l 是电机输出力矩， τ_1 和 τ_2 是轮子受到的摩擦力矩，与式 2.43 中的 τ_1 和 τ_2 相同， J_w 是轮子绕转轴的转动惯量， b_w 是转动粘性系数， P 是从两轮转速到车体前进速度和转向速度的一个转换矩阵。式 2.44 中间的右边两项可分别看作与轮子惯性力和粘性力的抵消项。

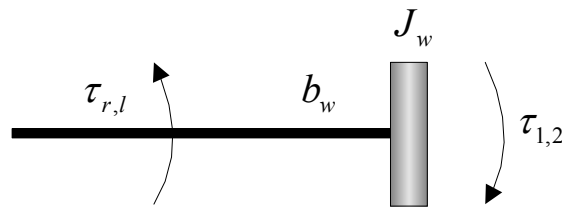


图 2.9 轮轴上力矩传递图

直流电机的电磁力学模型如下：

$$\begin{bmatrix} U_r \\ U_l \end{bmatrix} - K_e P \begin{bmatrix} v \\ w \end{bmatrix} = \frac{L}{K_i} \begin{bmatrix} \dot{\tau}_r \\ \dot{\tau}_l \end{bmatrix} + \frac{R}{K_i} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{2.45}$$

其中 U 是电机输入电压， K_e 是电动势常数， K_i 是力矩常数， L 是电机等效电感， R 是电机等效电阻。

综合式 2.43 的车体动力学、式 2.44 的力矩传递方程与式 2.45 的直流电机电磁力学模型，得到一个完整的轮式移动机器人的动力学模型：

$$U = X_v V + X_{\dot{v}} \dot{V} + X_{\ddot{v}} \ddot{V} - D \quad (2.46)$$

其中

$$\begin{aligned} U &= \begin{bmatrix} U_r \\ U_l \end{bmatrix}, V = \begin{bmatrix} v \\ w \end{bmatrix} \\ X_v &= \begin{bmatrix} \beta_1 + \beta_2 + \beta_3 w + \beta_4 \dot{w} & a(\beta_1 + \beta_2 - \beta_3 w - \beta_4 \dot{w}) \\ \beta_1 + \beta_2 - \beta_3 w - \beta_4 \dot{w} & a(-\beta_1 - \beta_2 - \beta_3 w - \beta_4 \dot{w}) \end{bmatrix} \\ X_{\dot{v}} &= \begin{bmatrix} \beta_5 + \beta_6 w + \beta_7 + \beta_8 & a(\beta_5 - \beta_6 w + \beta_8) + \beta_9 \\ \beta_5 - \beta_6 w + \beta_7 + \beta_8 & -a(\beta_5 - \beta_6 w - \beta_8) - \beta_9 \end{bmatrix} \\ X_{\ddot{v}} &= \begin{bmatrix} \beta_{10} + \beta_{11} & a\beta_{10} + \beta_{12} \\ \beta_{10} + \beta_{11} & -a\beta_{10} + \beta_{12} \end{bmatrix} \end{aligned} \quad (2.47)$$

$$\begin{aligned} \beta_1 &= \frac{K_e}{r}, \beta_2 = \frac{Rb_w}{rK_i}, \beta_3 = \frac{Rdmr}{2aK_i}, \beta_4 = \frac{Ldmr}{2aK_i} \\ \beta_5 &= \frac{Lb_w}{rK_i}, \beta_6 = \frac{Ldmr}{2aK_i}, \beta_7 = \frac{Rmr}{2K_i}, \beta_8 = \frac{J_w R}{rK_i} \\ \beta_9 &= \frac{RIr}{2aK_i}, \beta_{10} = \frac{J_w L}{rK_i}, \beta_{11} = \frac{Lmr}{2K_i}, \beta_{12} = \frac{LIr}{2aK_i} \end{aligned} \quad (2.48)$$

矩阵 D 是从式 2.43 中 F_d ， τ_d 衍生出来的未知干扰项。

2.3 系统辨识

在系统模型式 2.47 中，包含了系统几何物理参数、电机电磁力学参数和系统力学参数共 11 个未知参数，具体见参见表 2.1。本课题将通过几何测量、说明书资料曲线拟合和时频域辨识的方法获得模型参数值。

2.3.1 物理测量法辨识

在本课题机器人系统中，车体总质量 m 、车轮半径 r 、两驱动轮间距 a 都可以通过直接测量获得，而车体质心与两驱动力轮中心轴距离 d 需要知道车体的质心所在。由于机器人车体基本呈扁平状，可采用的悬挂法来测量质心所在。悬挂法即用一绳索悬吊车体至半空中，该状态下车体仅受重力和拉力作用，故质心在悬挂绳索的直线上。通过两次不同位置的悬挂，两悬挂线所在直线的交点即为该机器人平台的质心，这样可得到 d 的具体数值。

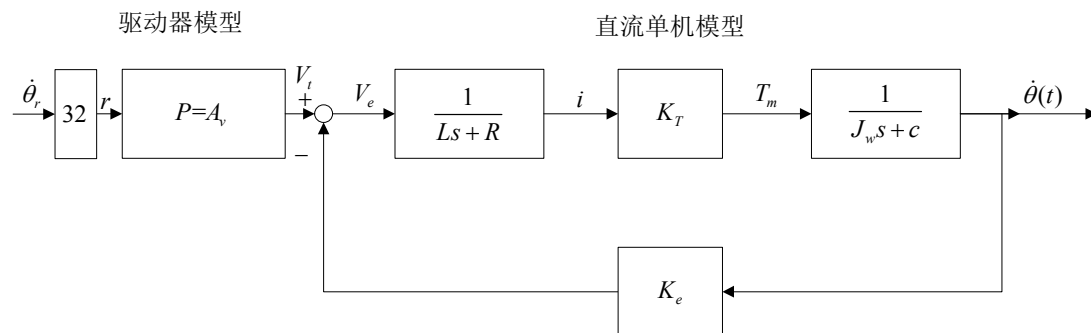


图 2.10 电机模型

本实验平台中的电机模型如图 2.10 所示，其中考虑到驱动器中电路的频宽远远高于平台的动力学频宽，此处用一个比例参数 A_v 代表驱动器模型。直流电机的特性曲线如图 2.11，可以通过特性曲线拟合获得电机的等效电阻 R 、扭矩系数 K_t 和电动势系数 K_e 等参数。

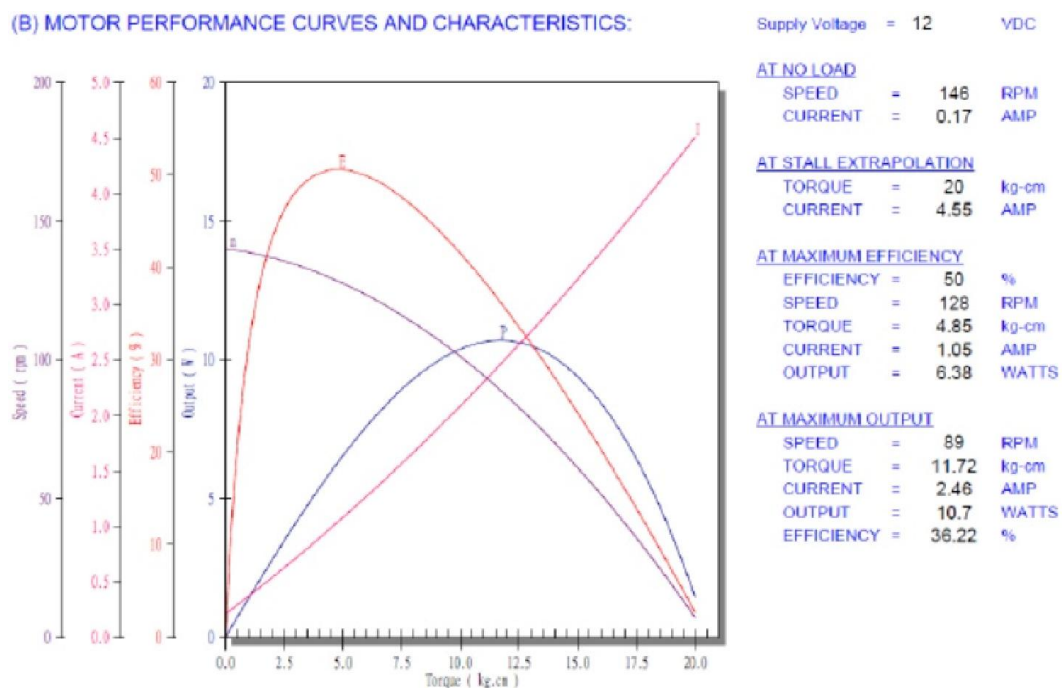


图 2.11 电机特性曲线

从电机特性图中，电机正常工作电压为 12V，停转时电流将达到 4.55A，由于没有转动，电机不产生反转电动势，所以等效总电阻 R 可记为电压与停转电流之比 2.637 欧姆。另外由于电机扭矩 $\tau = K_t \cdot i$ ，图中电机的转速-扭矩曲线基本呈线性，拟合得其斜率的倒数即为电机力矩系数。在电机空转时，电机有 0.17 的电流值，产生的微小扭矩可认为抵抗电机转动的粘性摩擦力，则 b_w 可由等式 $b_w = K_t i / w$ 获得。

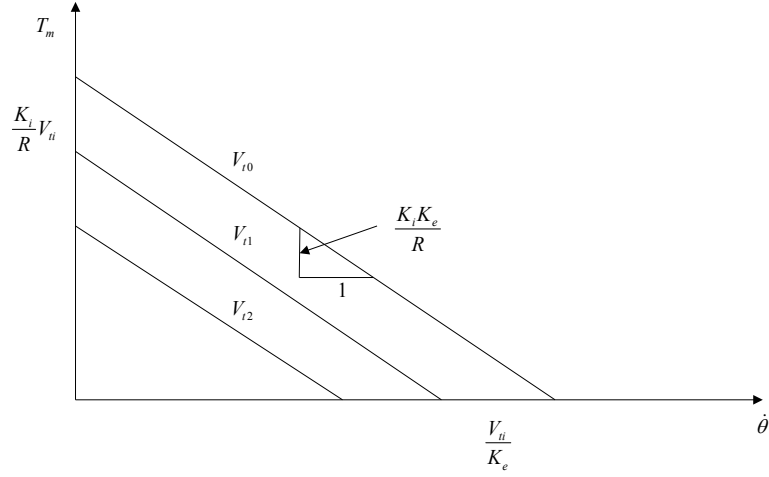


图 2.12 稳态直流电机 T-S 图

对于电机的电动势系数 K_e 而言，根据图 2.10 的电机模型，有：

$$\begin{aligned} V_i(t) &= L \frac{di}{dt} + Ri + K_e \dot{\theta}(t) \\ T_m(t) &= K_i i(t) \end{aligned} \quad (2.49)$$

在电机处于稳态时，电感的作用可以忽略，将式 2.49 中两等式结合，可得到电机扭矩与转速的线性关系式，关系如图 2.12：

$$T_m(t) = \frac{K_i}{R} V_i(t) - \frac{K_i K_e}{R} \dot{\theta}(t) \quad (2.50)$$

而此时，电机力矩系数与电机等效电阻由之前步骤已获得，通过拟合图 2.12 的扭矩-转速曲线的斜率即可获得电机电动势系数值。

2.3.2 时频域辨识

通过前小节的参数辨识，还剩下电机电感 L 、电机转子和轮子的转动惯量之和 J_w 与车体绕两轮中心 O 点转动惯量 I 未知。接下来将通过选取相应的输入信号获得平台相关的输出信号，并通过 matlab 上的系统辨识包对其时域和频域进行分析，进而获得相关未知参数和系统模型。

考虑到电机电感 L 和电机转子和轮子的转动惯量之和 J_w 两个参数仅存在于电机子系统中，可通过仅针对直流电机做辨识获得。针对图 2.10 的电机模型，选择正弦扫频电压信号作为直流电机的输入信号，其频率段涵盖到 0-14Hz，并且每个正弦信号频率相差 0.2Hz，一共 70 组数据。

首先，通过在 Labview 上的编程，实现给电机依次输入频率递增的正弦电压信号，并通过 Labview 进行相应的数据采集，采集得到的是由电机上光学编码器

测量得到的电机转速。本次针对的模型为一个开环的从输入电压到电机转速的直

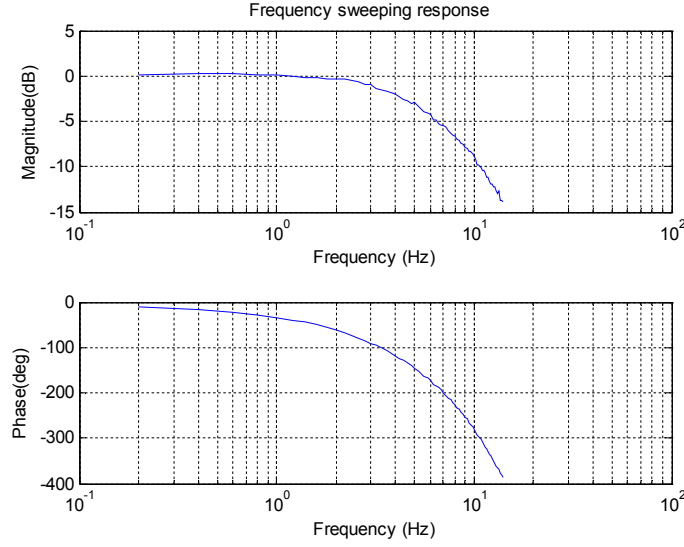


图 2.13 电机频域响应曲线

流电机模型，未有反馈项。将采集到的 70 组数据通过 Matlab 去分别计算每一组数据的幅值相应和相位相应情况，再通过得到的 70 组数据的相位相应情况画出电机的频域相应曲线，如图 2.13。最后将得到的电机频域相应数据导入到 matlab 上的系统辨识包中进行参数拟合，其基本思想为通过最小二乘法原理匹配到与实际最为接近的参数模型。该种采用扫频信号作为输入的辨识方法得到的数据量较为全面，并且不同频域段的信号之间无互相干扰，因此得到的辨识结果可靠，不足之处由于数据量大，处理起来相对较为繁琐，不过可通过编程实现自动处理数据的方法来提高效率，相应的 matlab 频域分析程序可见附录 A.1。

由于如图 2.10 所示的电机模型为两阶，并且考虑到实际系统存在 0.04s 的延迟，其中平台中实时系统和底层 FPGA 的循环周期都为 0.02s。因此选用式 2.51 带有 0.04s 延时环节的二阶模型去拟合电机曲线参数，得到的结果如式 2.52 所示：

$$G = \frac{K}{(1 + T_{p1} \cdot s)(1 + T_{p2} \cdot s)} e^{-0.04 \cdot s} \quad (2.51)$$

$$\begin{aligned} G &= \frac{0.97086}{(1 + 0.017635s)(1 + 0.020142s)} e^{-0.04s} \\ &= \frac{0.97086}{0.0003552s^2 + 0.037777s + 1} e^{-0.04s} \end{aligned} \quad (2.52)$$

图 2.10 的电机模型加入比例驱动器环节后共有三个未知参数，如 2.53 所示：

$$\frac{w}{\theta_r(s)} = \frac{32A_v K_i}{J_w L s^2 + (Lb_w + J_w R)s + (b_w R + K_i K_e)} \quad (2.53)$$

对比式 2.52 与式 2.53，可得到以下三个等式：

$$\begin{cases} \frac{32A_v K_i}{X} = 0.97086 \\ \frac{J_w L}{X} = 0.000352 \\ \frac{Lb_w + J_w R}{X} = 0.037777 \end{cases} \quad (2.54)$$

其中 $X = b_w R + K_i K_e = 0.27062$ ，三个等式获得三个未知参数有：

$$\begin{cases} A_v = 0.018274 \\ J_w = 0.0038293 \quad (1) \\ L = 0.0251026 \end{cases} \text{ 或 } \begin{cases} A_v = 0.018274 \\ J_w = 4.75616 \times 10^{-5} \quad (2) \\ L = 2.02107 \end{cases} \quad (2.55)$$

显然，第一组数据与实际相符，故采用第一组数据。剩下的车体绕两轮中心 O 点转动惯量 I 可通过类似方法获得，仅针对机器人平台的输入电压到转角速度 u-w 模型，输入为根据参考车体正弦旋转速度转换的电压，使用陀螺仪测量车体实际转速输出，得到的时域响应结果导入到 matlab 辨识工具箱，拟合得到的 u-w 数学表达式与式 2.46 对比即可得到 I 数值。最后所有参数辨识结果如表 2.1 所示。

表 2.1 系统模型参数

参数名称	代表字符	标称值
车体总质量	m	3.6Kg
质心与两驱动力轮中心轴距离	d	0.44m
轮子半径	r	0.04918m
两轮间距一半	a	0.179m
电机电动势系数	K_e	0.573V·s/rad
电机力矩系数	K_i	0.4493N·m/A
电机等效总电阻	R	2.637Ω
电机粘性阻尼系数	b_w	0.00499N·m·s/rad
电机电感	L	0.0251H
电机转子和轮子的转动惯量之和	J_w	0.00383Kg·m ²
车体绕两轮中心 O 点转动惯量	I	0.07946Kg·m ²

2.4 本章小结

本章主要内容为搭建了一个轮式移动机器人平台,并且针对具体的实验平台在动力学上进行了整体性的建模,通过系统辨识的方法获得模型参数。

1) 本次实验平台的搭建是围绕 NI Robotics Starter Kit 2.0 为基本框架搭建的,在一个基本的机器人平台上附加了六自由度惯性传感器和摄像头传感器分别去获得机器人角速度和绝对位姿信息,整个机器人实验平台的软硬件框架也随之体现出来。随后对机器人上的摄像头传感器采用张正友标定法去获得摄像头的内部参数。

2) 轮式移动机器人是一个典型的非完整约束系统,本章首先分析了非完整约束系统的一般特性及相应的拉格朗日方程模型。随后针对具体的 3 轮轮式移动机器人平台进行了运动学和动力学模型的推导,其中动力学为重点研究对象。采用了全面整体的分析方法,涵盖了平台中直流电机电磁力学模型、驱动轮力矩传递模型及机器人车体力学模型。

3) 得到机器人的动力学系统后,采用了基于具体实验平台的物理测量方法和时频域辨识的方法获得了模型中的各参数,最后得到一个带有离线参数的具体动力学模型。

本章的平台搭建和系统辨识工作为后面的控制策略设计做出了铺垫性的工作,也为一般机器人建模提供了一个系统的参考方法。

第3章 μ -synthesis 鲁棒控制器设计

3.1 模型分析

3.1.1 模型解耦

在得到系统整体模型式 2.46 中，为了全面了解系统特性建模覆盖面较大，在具体控制器设计时，可依据实际情况忽略影响甚微的部分。代入已辨识的参数，为了便于模型结构分析，忽略未知干扰矩阵 D ，得到以下具体参数耦合模型：

$$\begin{bmatrix} \ddot{v} \\ \ddot{w} \end{bmatrix} = \begin{bmatrix} 41.6667 & 41.6667 \\ 341.9973 & -341.9973 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} 105.833 & -0.01813w \\ 0.831737w & 105.746 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} - \begin{bmatrix} 1020.608 & -1.55w - 0.01475\dot{w} \\ 70.999w + 0.6765\dot{w} & 1499.521 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (3.1)$$

在式 3.1，包含了较多的耦合项 w 、 $\dot{w}$ ，这些是在车体转动时由于质心和几何中心不重合而产生的离心力和广义加速度导致的耦合项。将式 3.1 转化成标准的多输入多输出方程有：

$$\begin{bmatrix} \dot{v} \\ \ddot{v} \\ \dot{w} \\ \ddot{w} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1020.608 & -105.833 & 1.55w + 0.01475\dot{w} & 0.01813w \\ 0 & 0 & 0 & 1 \\ -70.999w - 0.6765\dot{w} & -0.831737w & -1499.521 & -105.746 \end{bmatrix} \begin{bmatrix} v \\ \dot{v} \\ w \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 41.667 & 41.667 \\ 0 & 0 \\ 341.9973 & -341.9973 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.2)$$

其中直线速度 v 和转角速度 w 就是在动力学模型中所需要控制的输出。在实际系统中，转角速度范围在 $0-3\text{rad/s}$ ，大多数时间在 $0-1.5\text{rad/s}$ 范围内，同时由于实际系统的限制，转角加速度在 $0-18\text{rad/s}^2$ 。从式中的耦合项中可以看出，乘以实际系统的转角速度与转角加速度后耦合项的值相对其它项很小，故可暂时忽略这些耦合项，得到简化的系统动力学模型并写成传递函数的模型如下：

$$\begin{bmatrix} v \\ w \end{bmatrix} = G(s) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{41.6667}{(s+95.1011)(s+10.7318)} & \frac{41.6667}{(s+91.211)(s+14.444)} \\ \frac{341.9973}{(s+88.8735)(s+16.8725)} & \frac{-341.9973}{(s+87.640)(s+18.016)} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.3)$$

对于一个多维的传递函数矩阵 $M(s)$ ，可以通过一系列左乘矩阵 $\{L_1(s), L_2(s), \dots, L_{k_1}(s)\}$ 和右乘矩阵 $\{R_1(s), R_2(s), \dots, R_{k_2}(s)\}$ 将转换成秩与矩阵 $M(s)$ 一样为 r 的多项式对称矩阵 $M_f(s)$ ，其中矩阵 $M_f(s)$ 形式为：

$$M_f(s) = [E(s) \ 0] \text{ 或 } M_f(s) = \begin{bmatrix} E(s) \\ 0 \end{bmatrix}$$

$$E(s) = \begin{bmatrix} \bar{\varepsilon}_1(s) & & & \\ & \ddots & & \\ & & \bar{\varepsilon}_1(s) & \\ & & & 0 \\ & & & & 0 \end{bmatrix} \quad (3.4)$$

如果矩阵 $M(s)$ 为满秩的行列数一致的矩阵， $M_f(s)$ 则为满秩的对称矩阵。

要对一个复杂的多输入多输出多项式系统矩阵解耦成对称矩阵，需要选取一定规律的多项式矩阵。需要知道，对一个多项式矩阵，左乘一个参数矩阵，就等于对该矩阵的行向上的数据进行相应的处理，同样右乘一个参数矩阵，就等于对列向上的参数进行相应处理。

针对任意一个多项式矩阵 $M(s)$ ，可以通过以下方式获得转化后的多项式对称矩阵 $M_f(s)$ 及左右乘矩阵：

- 1) 对多项式矩阵 $M(s)$ 选取相应的左右乘矩阵把矩阵中系数最少的一个多项式搬运到一行一列的位置，并把该最小的系数定为 γ_1 ；
- 2) 同样采用矩阵相乘的方法，把第二行第一列多项式的系数 γ_2 变换成少于 γ_1 。
如果该项为零，则直接进入下一步，否则通过交换第一行和第二行并重复之前的步骤把该项变为零；
- 3) 重复第 2 步步骤把第一列的其余所有项变换成零；
- 4) 再重复第 2 步步骤把第一行的除第一项外其他所有项变换成零；
- 5) 若第 4 步中不能顺利进行，重新调到第 2 步，选择合适的第一行第一列元素；
- 6) 如式 3.5，若变换过程中在 $M_{22}(s)$ 中产生新的多项式项系数比 $\bar{\pi}_{11}(s)$ 项的系数小，需通过行列变换把系数最小项提到 $\bar{\pi}_{11}(s)$ 的位置，最后得到式 3.5 形式且保证 $\bar{\pi}_{11}(s)$ 是新的矩阵中最小项。
- 7) 对矩阵 $M_{22}(s)$ 重复 1 到 5 的步骤，最终得到解耦的矩阵 $M_f(s)$ 。

$$M(s) = \begin{bmatrix} \bar{\pi}_{11}(s) & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & M_{22}(s) & \\ 0 & & & \end{bmatrix} \quad (3.5)$$

对本课题的系统模型式 3.3 中进行以上的 Smith 模型转换步骤，两阶系统相对比较简单， $G(s)$ 右侧加个补偿矩阵 C_{pre} ，即 $C = \bar{C}C_{pre}$ ，则可得到解耦的系统模型：

$$\bar{G}(s) = G(s)C_{pre} = \begin{bmatrix} \frac{41.6667}{(s+95.1011)(s+10.7318)} & 0 \\ 0 & \frac{341.9973}{(s+88.8735)(s+16.8725)} \end{bmatrix} \quad (3.6)$$

其中补偿右乘矩阵为 $C_{pre} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$ 。

3.1.2 模型分析

式 3.6 中的等式是忽略了未知干扰和耦合项得到的一个简化的解耦方程，之前分析忽略项中耦合项较小，不过尚须验证未知干扰项对实际系统的影响。本次对比实验的对象是整个 WMR 实验平台，输入是两驱动直流电机的电压，之前模型分析系统有前进速度 v 和转角速度 w 两个输出，不过在实际系统中，机器人的直线速度只能由两轮转速计算得到，而两驱动轮的编码器精度不高，每转最高只能产生 400 个脉冲，所以直线速度的所测结果精度也不高，普通 PID 控制器就能使系统达到传感器测量分辨率级别的控制精度，高性能控制器在该环上体现不出性能，并且耦合项都与转角速度有关，对整个系统性能的影响较高，所以下面工作只考虑从输入电压到转角速度的 u - w 模型。输入采用正弦叠加输入信号，输入为 0.1Hz 到 20Hz 间隔为 0.1Hz 峰值为 10V 的两电机电压信号，输出转角速度 w 通过六自由度惯性传感器测量获得。

使用 matlab 系统辨识包对上述实验采集到的 w 信号进行频域分析，得到实际系统的频率响应图，并与式 3.6 中的 u - w 模型进行对比，可见图 3.1。在对比图中，红色实线是简化解耦后的理论推导模型，蓝色点状图为实际系统的频域响应，可以看出，从对比可得两者相位上的值是基本相吻合的，而幅值上的结果虽

大致吻合，个别区域确有较大区别。首先，在 0.1Hz 到 2Hz 的较低频率段，实验结果的幅值基本在统一水平面上，并且低于理论推导的模型结构一定幅度。这是之前模型推导过程中忽略的未知干扰对系统的影响，实际分析为万向支撑轮摩擦力的影响；然后在相应的 8Hz 到 20Hz 的较高频率段，两者的幅值基本相当，但是在实验的频率响应过程中，随着频率的增加，幅值的波动越来越大，这是在相应的频率段可能存在噪声的干扰或未知力学的影响，这都会导致在辨识过程中造成辨识结果准确性，理论推导得到的模型具有参数不确定性。

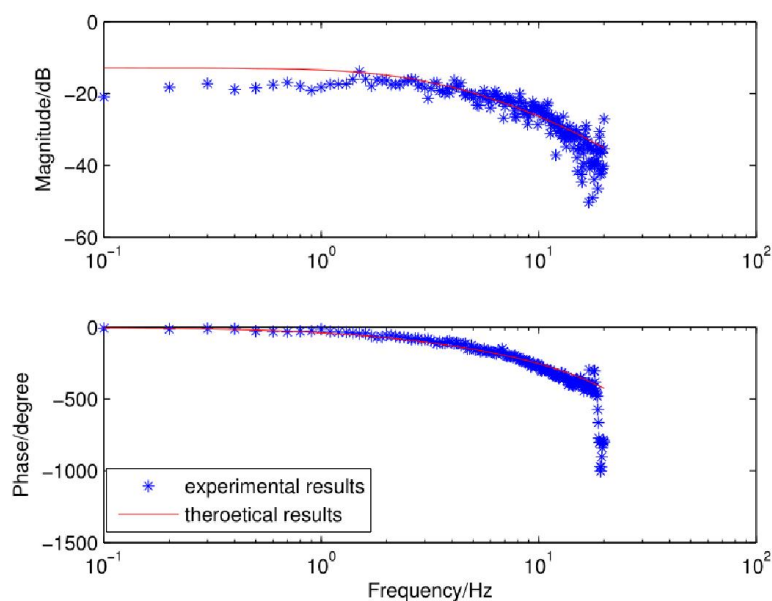


图 3.1 频率响应对比

从上图的对比分析看出，在系统达到的相对高频和低频域内分别存在模型参数不确定性和摩擦力的影响，也是与理论简化模型的区别。相应的控制器的设计需要解决这两个影响因素，这样设计的控制器效果才能达到基于简化模型设计的控制效果，也就是需要控制器有一定抗干扰能力和鲁棒性能，具体针对本移动机器人系统，为对低频摩擦力的抗干扰能力和对相对高频模型参数不确定性的鲁棒性能。

3.2 μ -synthesis 控制器

3.2.1 控制方法选择

针对一般生活与工业上的应用机器人，由于成本等因素的影响，硬件能达到的处理周期往往有限，采用 NI 公司提供的 Robotics Starter Kit 2.0 为例，在底层

FPGA 上能达到的最高处理周期为 20ms, 采样周期仅为 50Hz, 根据采样定理, 理论上能恢复的实际采样信号的频宽为 25Hz, 也就是理论上该系统能达到的最高频宽。一般的控制系统需要处理的最常见的外干扰项为摩擦力和传感器噪声, 它们分别存在于系统低频段和高频段, 在有限的系统频宽内, 这两者在整个系统的频率范围内则覆盖率很大, 处理这个问题将变得棘手。具体在本次实验系统中, 低频摩擦力的影响比较突出, 同时也需考虑传感器噪声, 还需处理高频柔性力学的影响, 传统的 PID 控制将达不到理想的控制效果。

针对一个传统的闭环控制系统, 有如图 3.2 的结构, 其中 $P(s)$ 是需要控制的系统, 前馈函数 $F(s)$ 和控制器 $C(s)$ 是设计的控制系统, $R(s)$ 是参考输入, $\bar{R}(s)$ 是经过前馈环节处理后的参考输入, $E(s)$ 控制误差, $U(s)$ 是控制输入, $d_i(s)$ 和 $d_o(s)$ 分别为内部干扰和外部干扰, $n(s)$ 是传感器误差, $Y(s)$ 为系统输出。

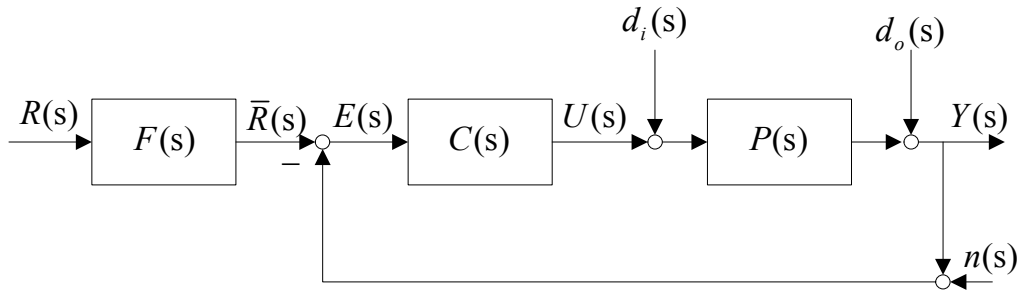


图 3.2 一般闭环控制系统

在该系统中, 有关系式

$$U(s) = C(s)E(s) = C(s)[F(s)R(s) - (Y(s) + n(s))] \quad (3.7)$$

根据 3.7 式, 代入到系统的输出函数中, 则有

$$\begin{aligned} Y(s) &= d_o(s) + P(s)[u(s) + d_i(s)] \\ &= d_o(s) + P(s)C(s)F(s)R(s) - P(s)C(s)Y(s) - P(s)C(s)n(s) + P(s)d_i(s) \end{aligned} \quad (3.8)$$

将式 3.8 针对不同的输入进行整理, 有

$$\begin{aligned} Y(s) &= \frac{P(s)C(s)F(s)}{1 + P(s)C(s)} R(s) + \frac{1}{1 + P(s)C(s)} d_o(s) \\ &\quad + \frac{P(s)}{1 + P(s)C(s)} d_i(s) + \frac{-P(s)C(s)}{1 + P(s)C(s)} n(s) \end{aligned} \quad (3.9)$$

$$\begin{aligned} U(s) &= \frac{C(s)F(s)}{1 + P(s)C(s)} R(s) + \frac{-C(s)}{1 + P(s)C(s)} d_o(s) \\ &\quad + \frac{-P(s)C(s)}{1 + P(s)C(s)} d_i(s) + \frac{-C(s)}{1 + P(s)C(s)} n(s) \end{aligned} \quad (3.10)$$

在式 3.9 中, 分别得到系统的各类输入到输出的传递函数, 而从 3.10 中可以得到系统的各类输入到控制输入的传递函数, 而这两个是控制系统设计需要重点考虑的两个量。输出需要使其尽可能快和准的跟踪参考输入, 而控制输入往往是系统中的一个有界的一个量, 控制系统计算出来的控制输入需保证在该界限范围内, 否则将导致系统失稳。根据不同输入对这两个量的影响, 可列出以下四个传递函数:

$$\begin{aligned} T(s) &= \frac{P(s)C(s)}{1+P(s)C(s)} \\ S(s) &= \frac{1}{1+P(s)C(s)} \\ S_i(s) &= \frac{P(s)}{1+P(s)C(s)} \\ S_u(s) &= \frac{C(s)}{1+P(s)C(s)} \end{aligned} \quad (3.11)$$

其中 $T(s)$ 是互补灵敏函数, 代表的是传感器噪声到系统输出的敏感程度, $S(s)$ 是敏感函数, 表达的是系统外部干扰到系统输出的敏感程度, $S_i(s)$ 是内部干扰敏感函数, 表达的是内部干扰到系统输出的敏感程度, $S_u(s)$ 是控制敏感函数, 表达的是外部干扰和噪声到控制输入的敏感程度。

设计的好的控制器系统是需要尽可能把上述 4 个等式都最小化的, 其中最重要的两个传递函数为 $T(s)$ 和 $S(s)$, 使其都为最小消减外部干扰和传感器噪声对系统输出的影响, 但是在式 3.11 可知 $T(s)+S(s)=1$, 同时使两者都为零是做不到的。实际系统中可根据外部干扰和传感器噪声存在的频率范围的不同, 有针对的使两者在不同频率段为零。本机器人系统中, 摩擦力的覆盖范围在整个系统能达到的频率范围比重较大, 也就是需要机器人系统在有限的频宽范围内达到很好的抗干扰能力。采用 μ -synthesis 控制器的设计方法可以很好地解决以上系统分析得到的问题。

3.2.2 控制方法思想

μ -synthesis 控制器能在系统具有参数不确定性和模型不确定性达到抗干扰、噪声抑制和高精度跟踪等多目标的一个特点, 这些优点在实际环境中轮式移动机器人的运行中具有很大的优越性, 本次课题将首先设计 μ -synthesis 控制器, 观察

其能达到的控制性能。

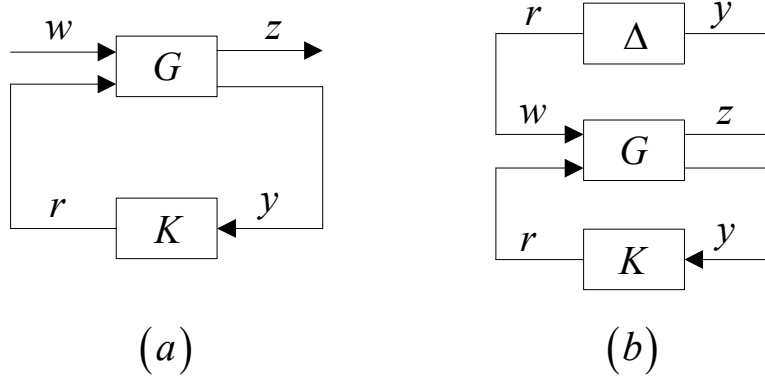


图 3.3 传统闭环系统 (a)、不确定系闭环系统 (b)

在理论上, H -infinity 优化控制系统如图 3.3 (a) 所关心的是找到一个能最小化输入输出传递函数 T_{zw} 的 H -infinity 范数的控制器。一个传递函数的 H -infinity 范数可以定义为如下形式:

$$\|G\|_{\infty} := \sup_{\omega \in R} \bar{\sigma}[G(j\omega)] = \sup_{u \in L_2} \frac{\|Gu\|_2}{\|u\|_2} \quad (3.12)$$

对一个正弦输入信号, $\|T_{zw}\|_{\infty}$ 为在输入到输出的所有频率点的最高一个稳态状态增益。 H -infinity 控制理论是最小化这个尖峰值或者改善最坏情况以达到 T_{zw} 的增益在所有频率段都很小。通常一个控制器 $K(s)$ 可以设计出来去保证一个界限 $\|T_{zw}\|_{\infty} < \frac{1}{\gamma}$ 。这个样的一个界限, 通过最小增益理论证明能同时保证不确定性

系统的稳定性, 如图 3.3 (b), 条件系统的稳态不确定性 $\|\Delta\|_{\infty} < \frac{1}{\gamma}$, 即这个闭环系统所能达到的鲁棒稳定性。根据上面提到的一些理论, μ -synthesis 控制方法被提出来解决不确定性系统, 同时设计的控制器能够达到一个整个闭环系统的鲁棒控制性能。

H -infinity 控制器设计通常是通过选取权重函数来实现的, 而权重函数是根据信号的相对幅值、频率独立性和它们的相对重要性而选择的。一个普遍的权重表现方框图可以表达成图 3.4 的样子。权重函数的选择是 H -infinity 控制器设计过程中的重要一点。在实际系统中, 系统的不确定性往往包括参数波动和未建模的动力学。通常, 我们可以选取一些参数范围来描述参数不确定性。对于未考虑的动力学, 附加多重的模型不确定性机构可以用来解决它的问题。

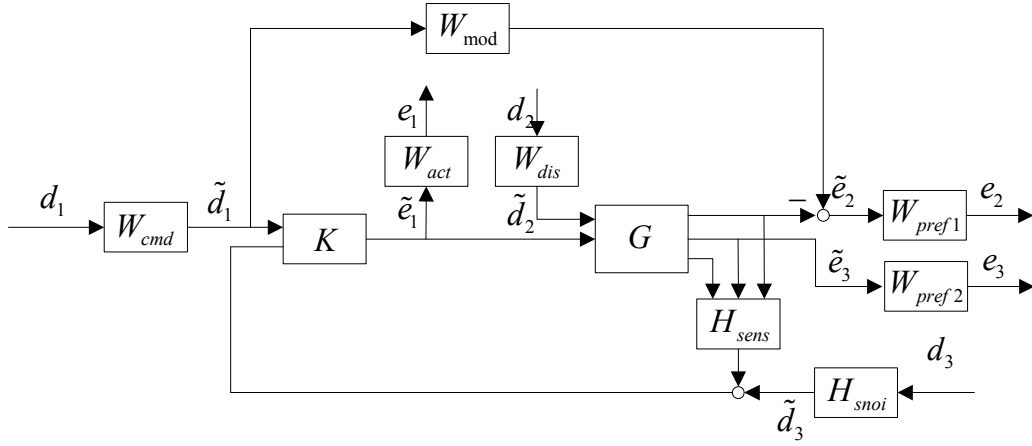


图 3.4 权重函数闭环系统方框图

使用 MATLAB 中的“dksyn”命令，一个不确定系统的 μ -synthesis 控制器可以被设计出来。稳定鲁棒性边界和性能鲁棒边界的定义如下：

a) 如果稳定鲁棒边界大于 1，则不确定系统对于每个模型不确定性都是稳定的；相反，如果稳定鲁棒边界小于 1，表示存在的一些不确定性元素在某个频率段可能会导致系统失稳。

b) 如果性能鲁棒边界大于 1，则不确定性系统的输入输出的反馈增益对于每个模型不确定性都小于 1；相反，性能鲁棒边界小于 1 表示某个模型不确定性在特定的频率段使得输入输出的反馈增益大于 1。

3.2.3 控制器设计过程

本课题首先采用 H-infinity 控制器设计方法在系统能达到的频宽内协调解决摩擦力和传感器噪声的影响，然后通过建立不确定性模型，使用 μ -synthesis 控制器的设计思想使控制系统达到一个很高的一个鲁棒性能。

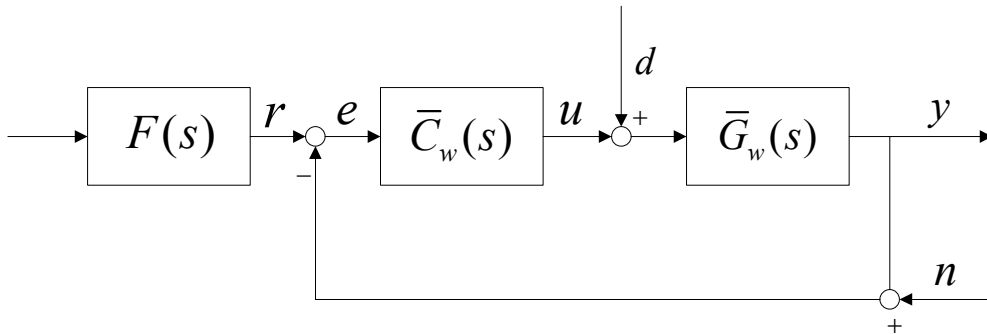


图 3.5 速度控制框架

本课题采用图 3.5 结构的控制框架，该框架前面有一个前馈函数，可以用来调配控制系统的零点，主要用于改善系统的瞬态反应。从控制框架中可得：

$$y = \bar{G}_w(u + d) = \bar{G}_w \left[\bar{C}_w(s)(r - y - n) + d \right] \quad (3.13)$$

为了方便后续权重函数的选择，对式 3.13 进行变换，使 $e = r - y$ ，则有：

$$-e = \bar{G}_w \left[\bar{C}_w(s)(e - n) + d \right] \quad (3.14)$$

进行以上变化后，可以有利于系统权重函数的选择。一个主要的权重函数的选择标准是相应信号重要性越大，所取的权重函数值将越大。 μ -synthesis 控制器性能的优劣主要也在于图中各权重函数的选择。

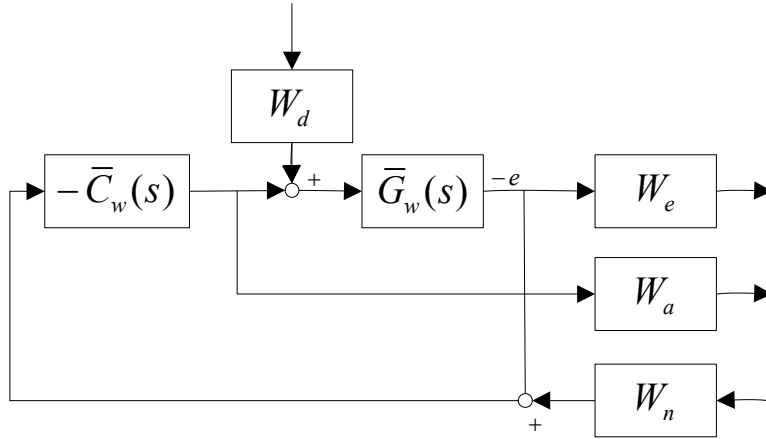


图 3.6 带权重函数控制框架

W_n 代表的是噪声的频域内容的权重。通常参照实验中的测量结果得到。由于由惯性传感器 ADIS16375 测得的旋转速度噪声幅值约为 0.02rad/s ，所以 W_n 取值为 0.02 ，即 $W_n = 0.02$ 。

W_d 通常用来代表外界干扰对系统影响的频率内容和幅值内容的权重。通过上述模型对比，知道摩擦力在 0.1Hz 到 1Hz 的低频段影响较大，我们选择权重函数在其频率段为一个较大值 1000 来抑制摩擦力。由于较宽的权重函数 W_d 频宽会导致最后较宽的闭环系统频宽，而实际系统能达到的频宽有限，如达不到设计的闭环频宽将会导致失稳。所以在本系统中， W_d 在 1Hz 后的幅值需要迅速减小，至少为二阶，因此选取权重函数：

$$W_d = \frac{10}{s^2 + 0.8s + 0.01} \quad (3.15)$$

W_e 为系统输出与参考输入误差的权重。通常选择一个较大常数值，代表最后的控制误差得到的重视程度。这里我们选择 $W_e = 80$ 。

W_a 用来代表控制输入的使用量的权重，该系统不许考虑该种情况，因此选

择 $W_a = 1$ 。

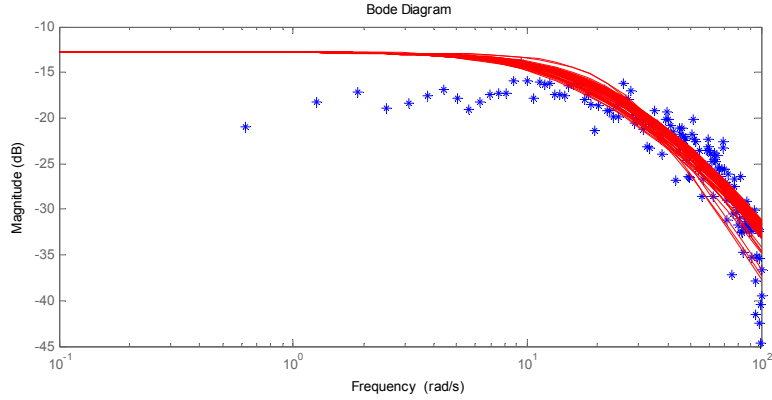


图 3.7 不确定性模型幅频图

在所有的权重函数选择后, 建立了一个参数不确定性模型用来涵盖在较高频率段的模型参数不确定性, 如图 3.7。建立的参数不确定性模型包括两个变动参数, 参数 1 的波动范围为 $[0.0005968, 0.001968]$, 参数 2 的波动范围为 $[0.0656, 0.0856]$, 模型如下:

$$\bar{G}_w = \frac{0.2287}{\text{para1} \cdot s^2 + \text{para} \cdot s + 1} \quad (3.23)$$

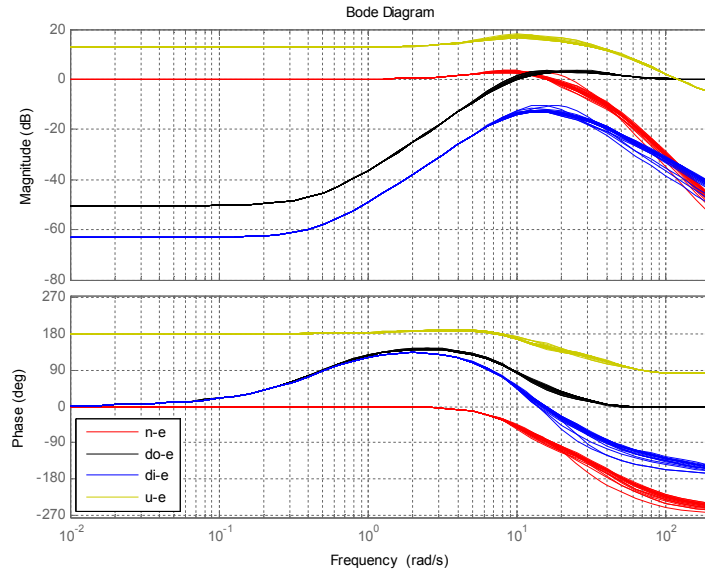


图 3.8 闭环 bode 图

使用 μ -synthesis 设计方法得到的控制器如式 3.24, 使闭环频率响应如图 3.8。如图所示, 各曲线分别为系统闭环传函, 即 $n-e$; 从外部干扰到输出误差的传函, 即 d_0-n ; 从内部干扰到输出误差的传函, 即 d_i-n ; 控制输入到输出误差的传函, 即 $u-e$ 。从图中可看出在系统能达到的频宽范围内很好的抑制了大频率范围且大

幅值的干扰。

$$C_u(s) = \frac{98.31(s+91.89)(s+15.6)(s+7.025)}{(s^2+0.8156s+0.2215)(s^2+81.75s+3028)} \quad (3.24)$$

3.3 实验验证及应用控制框架设计

3.3.1 PID 控制器调试

PID 结构简单，现场调试便捷，故在工业应用中有广泛的应用，不过针对复杂的系统，PID 控制器受限于本身结构往往达不到很好的控制效果。完整的 PID 控制器结构形式如下：

$$C_{PID}(s) = K_p \left(1 + \frac{1}{T_r s} + \frac{T_d s}{\tau_D s + 1} \right) \quad (3.25)$$

其中 T_r 是复归时间， T_d 是微分时间。

在 PID 控制器中，比例环节的作用与控制误差的瞬态值有关。一个单纯的比例控制器可以控制任何稳定系统，但是它不能达到零稳定误差，这是因为它的频域响应涵盖了整个频域。

积分环节是与累积误差成正比而产生的控制输入，这导致它是一个慢的响应环节。积分环节针对于阶跃的干扰和误差最后可以使系统达到零稳定误差。但是它有两个缺点：第一是在原点的极点不利于系统的稳定性；第二是该环节可能会使控制输入饱和从而导致系统失稳。

微分环节与根据控制误差的变化有关。该环节比较快速，并且会随着控制误差的消失而消失。由于它与控制误差的变化趋势有关，因此也为预测环节。主要的一个缺点是：当参考信号稳定点变化或噪声变化较大时，会产生大的瞬时控制输入变化，导致控制输入饱和从而可能会导致系统失稳。

对 PID 控制器的调试，比较通用的方法是 Ziegler-Nichols(Z-N)振荡法。该方法只针对开环稳定系统有效，正好与本实验平台相符合，主要调节步骤如下：

- 1) 给真实系统设定一个增益很小的比例反馈控制闭环；
- 2) 增加比例系数直到系统开始振荡；
- 3) 记录系统开始振荡时的比例系数 $K_p = K_c$ 和控制输入的振荡周期 P_c ；
- 4) 根据表 3.1 中的公式调整各 PID 控制器参数。

表 3.1 Z-N 振荡参数调整公式

	K_p	T_r	T_d
P	$0.5K_c$		
PI	$0.45K_c$	$\frac{P_c}{1.2}$	
PID	$0.60K_c$	$0.5P_c$	$\frac{P_c}{8}$

根据以上步骤，得到 PID 各参数之后，再根据实际平台调整 PID 参数，最后得到相对能达到最好控制效果的 PID 控制器如式 3.26。为了与 μ -synthesis 控制器对比的公平性，两控制器的达到的闭环频宽一致。

$$C_{PID}(s) = \frac{0.00888s^2 + 0.333s + 3.7}{0.0008s^2 + 0.08s} \quad (3.26)$$

3.3.2 动力学速度控制实验

为了全面验证所设计控制算法的性能，以下性能参数将会被用到：

$e_M = \max_t \{|e|\}$ ，表示跟踪误差的最大绝对值，用来描述系统的瞬态响应；

$e_F = \max_{T-1 \leq t \leq T} \{|e|\}$ ，表示在稳定运行的一段时间内跟踪误差的最大绝对值，用来描述系统最后的跟踪精度；

$\|e\|_{rms} = \left(\frac{1}{T} \int_0^T |e|^2 dt \right)^{1/2}$ ，跟踪误差的有效值，用来描述平均跟踪性能。

在本实验装置中，由于驱动器带动的直流电机转速过零点时有较大的死区，导致系统的瞬时相应较差， $\|e\|_{rms}$ 将由未过零点的数据计算得到。

表 3.2 实验性能参数

	e_M (rad/s)	e_F (rad/s)	$\ e\ _{rms}$ (rad/s)
PID	1.9653	0.3492	0.1275
μ -synthesis	0.9601	0.2154	0.0709

1) 实验对象及目的

在本实验中，实验对象针对的是移动机器人动力学部分，测试 μ -synthesis 控制器在图 3.5 所示动力学反馈闭环中的性能，通过与 PID 控制器对比得到，被控量是机器人转角速度 w 。

2) 实验方法

实验中的参考输入信号是频率为 2Hz 幅值为 1.5rad/s 的正弦转角速度信号，使用六自由度惯性传感器测得机器人实验台的实际转速与参考输入信号做差，得到机器人角速度的跟踪误差。依次在图 3.5 式中的动力学反馈环路中分别写入 PID 与 μ -synthesis 控制器，两者控制器的跟踪误差对比情况如图 3.9。由于简化后的动力学模型为二阶模型，相对简单，加上控制器每个循环周期为 20ms，具有较好的实时性。

3) 实验结果及分析

将 PID 与 μ -synthesis 控制器代入机器人实验平台中得到的角速度跟踪误差按照 e_M 、 e_F 和 $\|e\|_{rms}$ 三者性能参数计算，具体值见表 3.2。对于性能 e_M 而言，PID 控制误差结果为 1.9635rad/s，而 μ -synthesis 控制误差结果为 0.9601 rad/s，两者都比较大，这是因为实际系统中电机过零点有较大停滞死区导致过零点的瞬态响应差；对性能 e_F 而言，速度平缓运行的一段数据，得到 PID 和 μ -synthesis 控制器性能参数分别为 0.3492rad/s 和 0.2154rad/s，性能有大幅提升；对 $\|e\|_{rms}$ 而言，该实验中速度过零点较多，为避免瞬态带来的较大影响，选取速度未过零点一段数据，得到 PID 和 μ -synthesis 性能参数分别为 0.1275rad/s 和 0.0709rad/s。从上述参数可知，如表 3.2， μ -synthesis 控制的最大误差和波动范围都比 PID 的控制结果小，说明了 μ -synthesis 控制器有更好地抗干扰能力和一定的鲁棒性能。

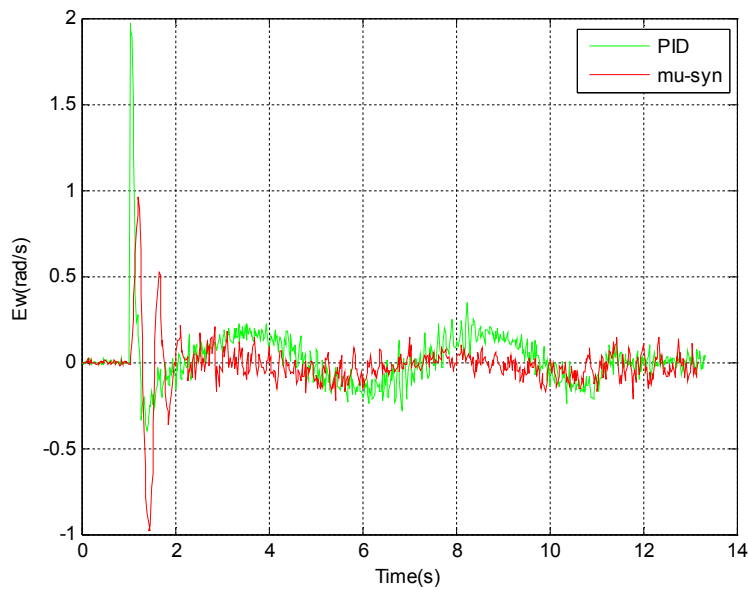


图 3.9 跟踪误差对比

3.3.3 内外环控制框架设计及位置跟踪控制实验

1) 实验对象及目的

由于轮式移动机器人在具体的工作中直接关注的往往是机器人的位置，本次第二个实验将具体针对机器人跟踪地面黑色轨迹线这一实际场景，具体测试 μ -synthesis 控制器在轮式移动机器人位置控制精度上的提高。

2) 实验方法

整个机器人位置和姿态的控制框架如图 3.10。在图中，由于采用的是内外环的控制框架，内部环路即为第一个实验中的动力学速度闭环，其中的控制器也分别使用 PID 和 μ -synthesis 控制器，放在 FPGA 中运行，处理周期短；外部环路为位姿控制环路，图中的运动学控制率采用的是式 1.2 中轮式移动机器人一种稳定的运动学跟踪控制率，为了方便 PID 与 μ -synthesis 控制器的对比，控制率中的 K_x, K_y, K_θ 三个参数在实验中都选为 1，该外环放在 Labview 实时系统中运行，由于图像检测周期较长，实时性较差，约为 200ms 一周期。如此内外环的控制框架在有限的硬件条件下提高了动力学上的频宽，使控制器的设计达到了更好跟踪性能。

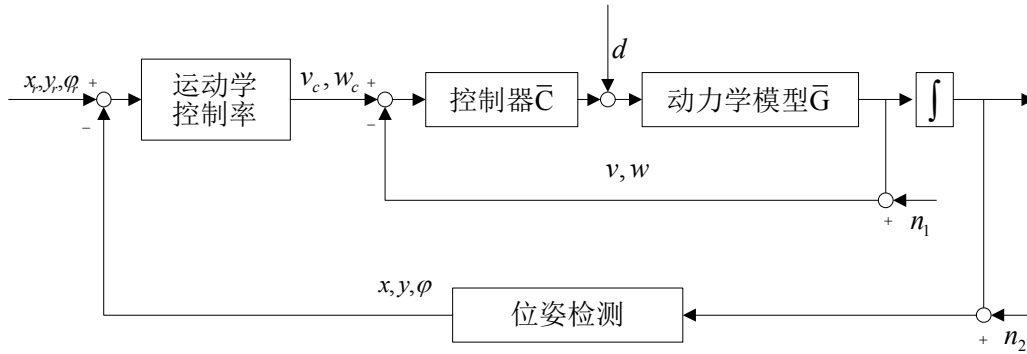


图 3.10 机器人位姿控制框架

在本次实验中，针对机器人位姿控制框架的内外环反馈信号，使用的测量方法有所不同。内环中的反馈量前进速度 v 可以通过使用光学编码器测得两驱动轮的转速进行推算得到，而反馈量转速 w 则可以直接使用六自由度惯性传感器获得，之前已提到由于实验平台光学编码器测量精度的限制，本次控制器设计与实验仅针对转速 w 做具体研究。外环中 x_r, y_r, φ_r 的三个位姿偏差量的测量是让移动机器人去跟踪地面上一条黑色椭圆轨迹线，通过安装在机器人上的摄像头传感器捕捉地面上的轨迹图像信号，并通过实时处理计算得到，如图 3.11。

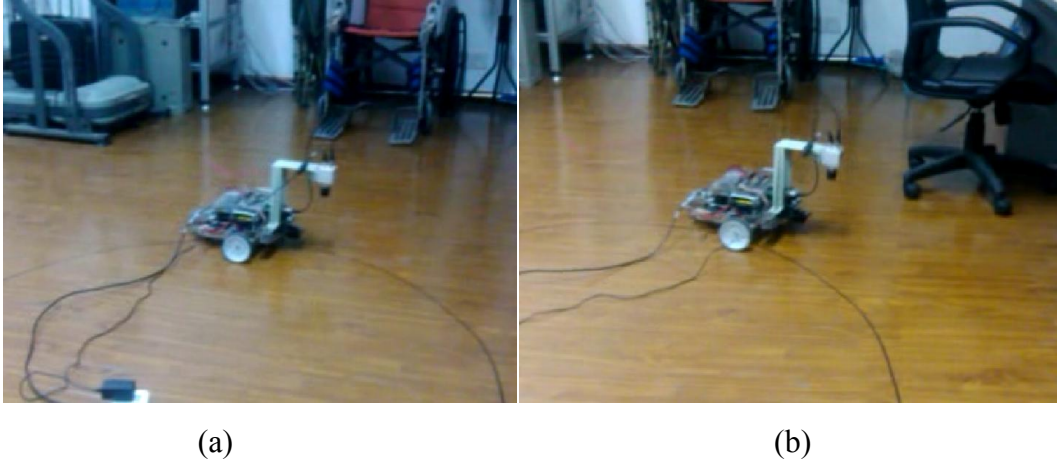


图 3.11 机器人跟踪椭圆轨迹

为了方便位姿信息的推算,摄像头竖直对地面,保证了摄像头坐标系中的 X 、 Y 轴与机器人坐标系中的 X 、 Y 轴平行,如图 3.12 所示。当摄像头捕捉一条线性轨迹后,使用阈值过滤算法保证直线轨迹特征突出,然后采用边缘提取算法去提取直线边缘像素点,采用最小二乘法用一条斜线去拟合这些像素点,得到在摄像机图像坐标系中的一条轨迹方程 $Ax + By + C = 0$, 然后需要利用第二章辨识的摄像头内部参数获得机器人的位姿偏差。由于具体跟踪一条轨迹线,前进速度可具体设定,故该方向上的偏差 x_r 令其为零,横向上的偏差 y_r 计算公式如式 3.27, 角度偏差 φ_r 计算公式如式 3.28,。

$$y_r = Y_r / \left(\frac{\alpha_x}{Z_c} \right) = Y_r \cdot \frac{214.2699}{1031.0731} = 0.2078 Y_r (mm) \quad (3.27)$$

$$\varphi_r = \begin{cases} -90^\circ + \arctan(-\frac{A}{B}) & \text{当}(-\frac{A}{B} > 0) \\ 90^\circ + \arctan(-\frac{A}{B}) & \text{当}(-\frac{A}{B} < 0) \end{cases} \quad (3.28)$$

其中 Y_r 是以像素为单位图像坐标系中的 Y 轴值, α_x 是棋盘格子一个网格的边长, Z_c 是式 2.7 中内部参数中的 $\alpha = \frac{f}{dX}$ 项, A 和 B 是上述拟合直线方程中对应的系数。

3) 实验结果

通过上述的位姿与速度闭环中的反馈信息,使用移动机器人跟踪椭圆轨迹,并且在内环的控制器分别采用 μ -synthesis 和 PID 控制器,并进行性能对比。椭圆轨迹的长轴为 2.26m, 短轴为 1.6m, 机器人系统上的 CCD 摄像头在本次实验

中能捕捉的宽度为 0.13m ，即当机器人的横向误差大于 0.0675m 时，系统将不能完成跟踪任务。在实验中，逐渐增加机器人的前进速度，当机器人前进速度达到 0.4m/s 时，在椭圆轨迹斜率较大处，附加 PID 控制器的系统会因为跟踪误差太大而不能完成跟踪任务，如图 3.11 的(a)图所示，而附加 μ -synthesis 控制器的系统能够正常完成跟踪任务，如图 3.11 的(b)图所示。实验说明了 μ -synthesis 控制器在该闭环系统对角速度 w 控制比 PID 能达到更快更高的性能。

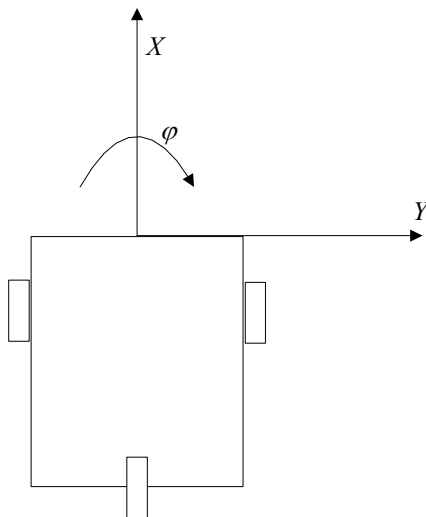


图 3.12 移动机器人摄像头坐标系

3.4 本章小结

本章在第二章建立的动力学模型上设计了 μ -synthesis 控制器，并通过实验验证了其控制性能。

1) 在第二章中分析得到的机器人动力学模型中，了解系统的具有模型耦合项，对其进行了相应的模型解耦和简化，并通过使得到的简化模型与实际系统的频率响应图进行对比，分析得到了系统在低频段存在一定的摩擦力的影响，在相对高频段有模型参数不确定性，为影响系统控制精度的主要因素。

2) 根据分析得到的系统中摩擦力和模型参数不确定性的特征，在动力学中设计了 μ -synthesis 控制器，该控制器能够在系统达到的频宽内兼容地解决低频摩擦力和较高频率段模型不确定的影响，针对具体的实验平台提高了其速度控制精度。同时，进行了相应的动力学速度控制实验和位置跟踪控制实验验证了其控制性能，其中位置跟踪控制实验采用的是跟踪椭圆轨迹线这一实际场景，采用的内外环控制框架将 μ -synthesis 很好应用到一般硬件的实验平台上。

第4章 自适应鲁棒控制器设计

4.1 控制系统分析

在上章节中,设计的 μ -synthesis 控制器在实际机器人能达到的频宽中很好地综合解决了低频摩擦力与高频模型不确定的影响,在已往的控制方法中仅解决摩擦力等外干扰的基础上有所提高,不过都是基于固定的系统平台,并且通过了详细的建模获得系统较为准确的模型参数。如图 4.1 所示,低频的摩擦力和相对高频的噪声干扰会在系统辨识过程中影响辨识结果,使系统存在参数和模型不确定性,同时,在不同负载的工作情况下,模型参数也会较大地改变。综上所述,之前设计的 u -synthesis 控制器尚存在两个局限性:

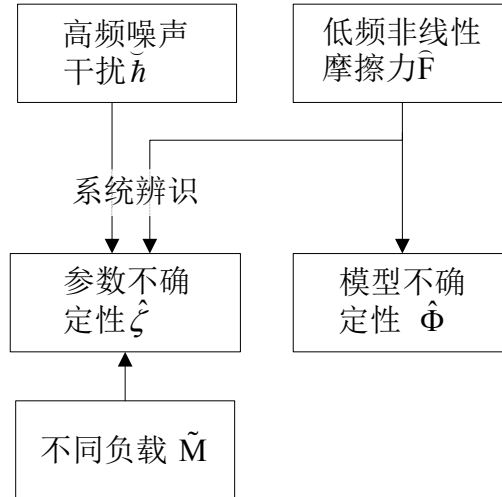


图 4.1 系统辨识影响因素

- 1) 控制器虽有一定的鲁棒性能,不过瞬态响应较差;
- 2) 控制器针对的已有实验数据得到的模型,若系统在不同工作条件下,出现较大的模型参数不确定性时,不能实时调整模型参数,影响控制性能。

综上所述,为了适应系统在不同的工作环境中的工作,自适应控制是能够在线更新模型参数,不过这仅限于系统模型参数不确定性,缺乏对于未建模不确定性与非线性环节的考虑。

4.1.1 问题阐述

一般的不确定非线性模型系统可表述为:

$$\dot{x} = f(x, t) + u, \quad f = \varphi(x)^T \theta + \Delta(x, t) \quad (4.1)$$

其中 $x \in R$ 和 $u \in R$ 分别是系统的输入和输出, $f(x, t)$ 是一个未知的非线性方程。一般情况下, 该方程很难有确定的表达式, 所以采用式 4.1 中的形式用两部分表示。第一部分采用线性近似的格式表示, 为一系列已知基本函数 $\varphi(x) \in R^p$ 与一系列未知权重 $\theta \in R^p$ 相乘的形式。一般而言, 这部分模型项事先可以根据系统的动力学模型或者其它形式的物理规律获得, 不过模型中系数是不确定的。第二部分是实际系统与第一部分近似处理后的误差。

一般在控制系统设计之前, 需要根据事先获得的系统信息作出一些合力的假设。对系统了解的越多, 则可以作出更多明确的假设, 这样设计的控制器理论上可以获得更好的控制效果。当然, 如果这些假设过于苛刻, 将很难与实际系统进行匹配, 所以依据此些假设设计出来的控制器也未必能够用到实际系统中。所以, 针对式 4.1 的模型, 一般有以下假设:

参数不确定和非线性不确定的范围是可知的, 有如下形式:

$$\begin{aligned}\theta &\in \Omega_\theta \triangleq \{\theta: \theta_{\min} < \theta < \theta_{\max}\} \\ \Delta &\in \Omega_\Delta \triangleq \{\Delta: |\Delta(x, t)| \leq \delta(x, t)\}\end{aligned}\quad (4.2)$$

其中 $\theta_{\max}$ 和 $\theta_{\min}$ 分别是权重系数向量 θ 的最大和最小界, $\delta(x, t)$ 是一个已知的函数。

4.1.2 自适应控制

一般的自适应控制只适用于只有参数不确定的情况^[53], 这里假设式 4.2 中的模型只有参数不确定, 即 $\Delta(x, t) = 0$ 。这样, 自适应控制的设计框架可以如图 4.2 所示。

自适应控制是在线去试图辨识获得未知权重参数 θ , 这样随着稳态误差的缩小, 最终的模型误差也可以被消除。在这种控制律中, 实时的参数估计是在系统除参数不确定性没有其他模型不确定性条件下进行的。这样设计的控制律如下:

$$u = u_m + u_s, \quad u_m = \dot{x}_d(t) - \varphi(x)^T \hat{\theta}(t), \quad u_s = -kz \quad (4.3)$$

其中 $\hat{\theta}(t)$ 是在线估计的权重参数, $z = x - x_d(t)$ 是跟踪误差, 以上的控制律可以得到如下的跟踪误差动力学方程:

$$\dot{z} + kz = -\varphi(x)^T \tilde{\theta}(t) \quad (4.4)$$

其中 $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$ 表示权重系数估计误差。在仅有参数不确定性的系统中，式 4.5 形式的梯度参数自适应率可以将等式 4.4 中右边的参数估计误差消除^[54]。

$$\dot{\hat{\theta}}(t) = \Gamma \varphi(x) z \quad (4.5)$$

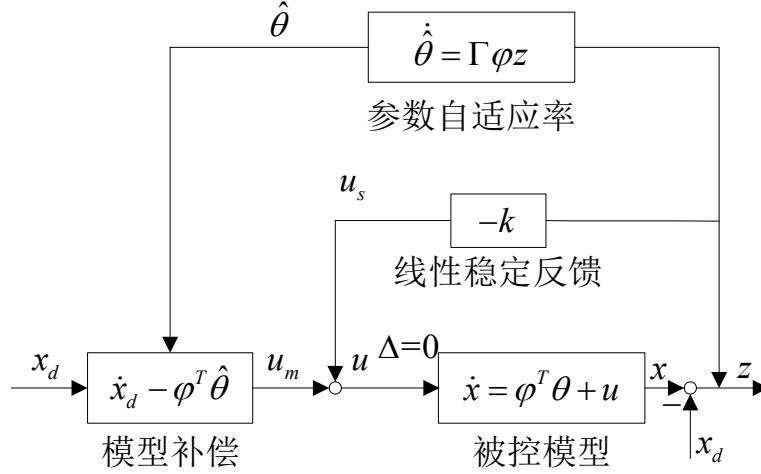


图 4.2 自适应控制框架

4.1.3 确定性鲁棒控制

确定性鲁棒控制根据已知的确定性模型特征设计一个非线性反馈项去消除各类形式的模型不确定性影响以达到一定的鲁棒稳定性。由于不确定性 $\Delta(x, t)$ 存在，采用如下形式控制率：

$$u = u_m + u_s, \quad u_m = \dot{x}_d(t) - \varphi(x)^T \hat{\theta}_o, \quad u_s = u_{s1} + u_{s2}, \quad u_s = -kz \quad (4.6)$$

其中 $\hat{\theta}_o \in \Omega_\theta$ 是权重系数 θ 的离线估计参数， u_{s2} 是针对鲁棒稳定性和鲁棒表现性能设计的一个额外反馈项。这样，误差动力学为：

$$\dot{z} + kz = u_{s2} - [\varphi(x)^T \tilde{\theta}_o - \Delta(x, t)] \quad (4.7)$$

其中 $\tilde{\theta}_o = \hat{\theta}_o - \theta$ 。左边是一个普通的稳定闭环动力学。方括号中的项表示系统不确定性。虽然它们未知，但是根据式 4.2 的假设都是有界的，可以找到一个已知函数 $h(x, t)$ 满足：

$$|\varphi(x)^T \tilde{\theta}_o - \Delta(x, t)| \leq h(x, t) \quad (4.8)$$

这样，鲁棒反馈控制项可以选为：

$$u_{s2} = -h(x, t) \operatorname{sgn}(z) \quad (4.9)$$

其中 $\operatorname{sgn}(z)$ 为一个不连续方程，如下：

$$\text{sgn}(z) = \begin{cases} 1 & z > 0 \\ -1 \text{ 到 } 1 \text{ 的一个随机值} & \text{if } z = 0 \\ -1 & z < 0 \end{cases} \quad (4.10)$$

根据滑模控制思想^[55]，式 4.1 的模型满足式 4.2 的假设时，以上控制率可以使系统的跟踪误差 z 最终趋于 0。

4.2 自适应鲁棒控制（ARC）

4.2.1 控制方法优势

上节的分析中可知系统可以采用在线的参数估计方法最终把模型参数不确定消除。这样不需要采用很高的一个增益反馈就可以达到系统的零稳态跟踪误差，理论上使用任意正参数 k 便可达到。但是这种方法尚有两个缺陷：第一，闭环控制系统的瞬态响应没有充分考虑，若存在干扰的情况下会导致跟踪误差出现突变的现象；第二，由于系统中把未知非线性状态 $\Delta(x, t)$ 忽略，这样控制器在线参数估计方法得到的参数存在漂移的现象，这样导致控制系统很容易受到干扰的情况影响从而导致控制性能变差。

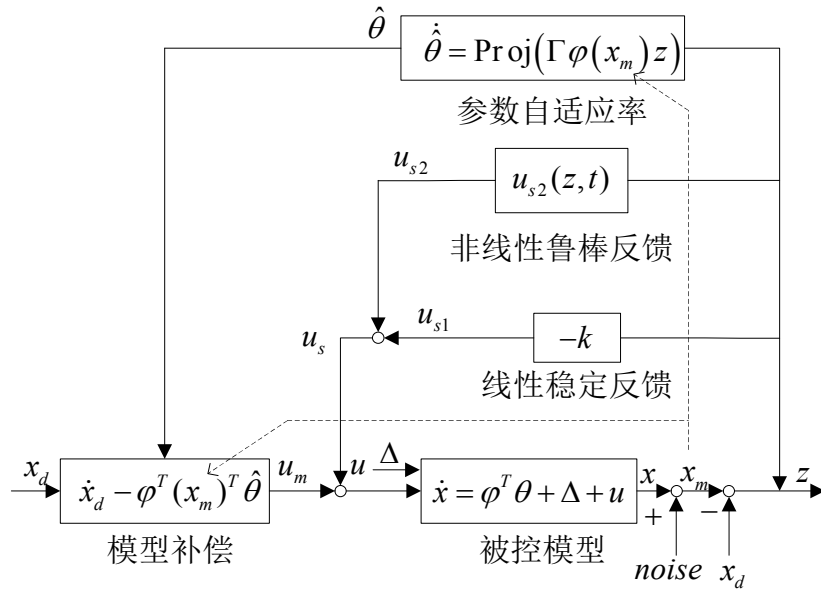


图 4.3 自适应鲁棒控制

确定性鲁棒控制相对自适应控制虽然能够在系统存在未知非线性环节时达到好的瞬态响应和稳态跟踪误差，这也使确定性鲁棒控制应用广泛，但是控制器

的渐进跟踪在控制率不能满足非连续的条件或能使某个增益区域无穷的条件下无法达到，这些在系统存在高频忽略的未建模动力学的环境下都是无法实现的。

以上两种控制器有很大的不同，但是通过分析它们之前是存在联系的，并且可以进行互补。自适应鲁棒控制(ARC)^[56-58]从机理上结合两种控制方法的优点，从而使存在各类不确定性环节的控制系統能够达到很好的瞬态和稳态控制效果，结构形式如图 4.3 所示。

4.2.2 设计过程

本章采用自适应鲁棒控制同第三章一样对式 3.6 中的 $u-w$ 的角速度环节模型进行设计，将未知的摩擦力环节考虑进来改写系统模型成如下形式：

$$\xi_1 \dot{w} = u - \xi_2 \dot{w} - \xi_3 w + \xi_4 + \tilde{d} \quad (4.11)$$

其中 ξ_1 、 ξ_2 和 ξ_3 分别是式 3.6 的 $u-w$ 模型中对应过来的系数， ξ_4 用来表示对摩擦等干扰的补偿项， $\tilde{d} = d - \xi_4$ 为实际外干扰与补偿项 ξ_4 之间的误差。由于在之前建立的线性模型中，参数辨识过程中受到忽略干扰的影响，参数 ξ 的值跟实际值有差别，并且系统工作环境变化时，参数 ξ 也会改变，然而在本次控制器设计中，作出以下假设后，参数可以通过相应自适应获得。

假设 1：系统的参数不确定性和非线性不确定的范围都是已知的，有：

$$\xi \in \Omega_\xi \triangleq \{\xi : \xi_{\min} \leq \xi \leq \xi_{\max}\} \quad (4.12)$$

$$\tilde{d} \in \Omega_d \triangleq \{\tilde{d} : |\tilde{d}| \leq \delta_d\} \quad (4.13)$$

其中 $\xi_{\min} = \{\xi_{1\min}, \dots, \xi_{4\min}\}^T$ ， $\xi_{\max} = \{\xi_{1\max}, \dots, \xi_{4\max}\}^T$ ， δ_d 是已知的。在式 4.12 中，符号 $\leq$ 是针对两个向量中每项分别操作的。

让 $\hat{\xi}$ 表示 ξ 的估计值， $\tilde{\xi}$ 表示相应的估计误差，即 $\tilde{\xi} = \hat{\xi} - \xi$ 。未获得的式 4.12 的相应模型系数值，可以采用以下修正的投影自适应率获得：

$$\dot{\hat{\xi}} = Proj_{\hat{\xi}}(\Gamma \tau) \quad (4.14)$$

其中 $\Gamma > 0$ 是一个对角矩阵， τ 是一个自适应方程，后面将会介绍。投影映射

$Proj_{\hat{\xi}}(\bullet) = [Proj_{\hat{\xi}_1}(\bullet_1), \dots, Proj_{\hat{\xi}_4}(\bullet_4)]$ 可以表示为：

$$Proj_{\hat{\xi}_i}(\bullet_i) = \begin{cases} 0, & \text{if } \hat{\xi}_i = \xi_{i \max} \text{ and } \bullet_i > 0 \\ 0, & \text{if } \hat{\xi}_i = \xi_{i \min} \text{ and } \bullet_i < 0 \\ \bullet_i, & \text{otherwise} \end{cases} \quad (4.15)$$

其中对于任何自适应方程 τ , 采用投影映射可以保证: $\hat{\xi} \in \Omega_{\xi} \triangleq \{\hat{\xi} : \xi_{\min} \leq \hat{\xi} \leq \xi_{\max}\}$ 和 $\tilde{\xi}^T (\Gamma^{-1} Proj_{\hat{\xi}}(\Gamma \tau) - \tau) \leq 0, \forall \tau$ 。

完整的自适应鲁棒控制和在线自适应方程可表示如下:

$$u = u_a + u_s, u_a = -\varphi_d^T \hat{\xi}, \tau = \varphi_d p \quad (4.16)$$

其中 u_a 是一个调整后的模型补偿环节, u_s 是鲁棒控制率, $\varphi = [-\dot{w}_{eq}, -\dot{w}, -w, 1]$ 是需要跟踪轨迹的回归量, $p = \dot{e} + k_1 e$ 是转换量, k_1 是一个正值, $e = \theta - \theta_d$ 是跟踪误差。鲁棒控制率 u_s 可写为以下形式:

$$u_s = u_{s1} + u_{s2}; u_{s1} = -k_{s1} p \quad (4.17)$$

其中 k_{s1} 是非线性增益, 它的值需要足够大使以下矩阵正定。

$$\begin{bmatrix} k_{s1} - k_2 - \xi_1 k_1 + \xi_2 + \xi_3 g & -\frac{1}{2}(\xi_2 k_1 + \xi_3 g k_1) \\ -\frac{1}{2}(\xi_2 k_1 + \xi_3 g k_1) & -\frac{1}{2}\xi_1 k_1^3 \end{bmatrix} \quad (4.18)$$

其中 k_2 是一个名义稳定反馈, u_{s2} 是一个鲁棒反馈项, 并满足:

$$\begin{aligned} \text{i} \quad & p \{u_{s2} - \varphi_d^T \hat{\xi} + \tilde{d}\} \leq \varepsilon \\ \text{ii} \quad & p u_{s2} \leq 0 \end{aligned} \quad (4.19)$$

这样, 最终的跟踪误差动力学如下:

$$\xi_1 \dot{p} = u_s - \varphi_d^T \tilde{\xi} + (\xi_1 k_1 - \xi_2) \dot{e} + \xi_3 e + \tilde{d} \quad (4.20)$$

该形式可以证明有以下特征^[59]:

A. 一般而言, 所有信号有界。正定矩阵可以表示为:

$$V(t) = \frac{1}{2} \xi_1 p^2 + \frac{1}{2} \xi_1 k_1^2 e^2 \quad (4.21)$$

并且有界:

$$V(t) \leq \exp(-\lambda t) V(0) + \frac{\varepsilon}{\lambda} [1 - \exp(-\lambda t)] \quad (4.22)$$

其中 $\lambda = \min\{2k_2 / \xi_{1 \max}, k_1\}$ 。

B. 如果经过一个有限的时间 t_0 ，系统只存在参数不确定性，即 $\tilde{d}=0, \forall t > t_0$ ，则根据 A 中的结果，可以获得系统最终的零稳态跟踪误差，即当 $t \rightarrow \infty$ 时， $e \rightarrow 0$ 。

4.3 实验验证及应用控制框架设计

4.3.1 ARC 参数调节

自适应鲁棒控制能在含有非线性不确定的系统中达到很好的瞬态和稳态响应，但是控制器设计中需要调节的参数包括自适应矩阵 Γ 和相应的稳态比例反馈值，对于稍复杂的系统其需调节的参数个数也大于普通的 PID 控制。一般而言，控制器中的参数调节都是随着控制系统设计过程中逐步实现的，随着对系统物理机制的了解加深，通过不断调试的方法得到适当参数从而到达好的一个控制效果。整的一个流程可现在仿真中调试，然后到应用到实际系统中去。

在 ARC 参数调节中，理论上式 4.19 中的参数 ε 选取的越小越能获得很好的一个稳态闭环控制精度，然后在实际中该参数不能选的很小，这样得到等效闭环反馈值可以保证在一定的范围中，防止在实际系统中因为忽略的高频动力学而导致系统失稳。通过以上的了解，大致的参数调节可以有以下两点一般方法：

A. 首先通过不断调试的方法针对实际系统的进行一些实验获得等效局部反馈增益存在的范围。这个过程就类似于 3.3 节中的 PID 的 Z-N 调节法把增益逐渐提高到系统开始振荡这一环节。用 K_M 表示实验中能够达到的最大等效局部反馈值， ε 可以根据等效增益在 $e=0$ 处的约束决定，即：

$$\varepsilon = \frac{1}{4(k_M - k)} h_M^2, h_M = \sup \{h(x, t)\} \quad (4.23)$$

其中 $h(x, t)$ 是系统不确定模型的一个范围函数。ARC 的自适应率调节参数不能太大，这样系统参数在快速的动态响应中才能得以缓慢变化。参数自适应矩阵需要根据等效反馈增益来决定，等效公式如下：

$$\varphi_d^T \Gamma \varphi_d = k_I \approx w_d^2 = \frac{k_{eq}^2}{4\zeta_d^2} \quad (4.24)$$

其中 ζ_d 是目标阻尼系数。

具体针对不同参数不确定情况的具体情况，自适应参数矩阵 Γ 的各项权重可以选择以下结构形式：

$$\Gamma = \gamma W_\theta^2, W_\theta = \text{diag}\{\theta_{\max} - \theta_{\min}\} \quad (4.25)$$

式 4.24 和式 4.25 组成了自适应矩阵参数的系统调节方法。

B. 由在 A 中的 $k_{eq} \approx k_M$ 和目标阻尼系数 $\zeta_d \approx 0.707 \sim 1.414$ ，式 4.25 中的保守参数 γ 可以选择为：

$$\gamma = \frac{k_M^2}{4\zeta_d^2 s_{\bar{\varphi}}}, s_{\bar{\varphi}} = \sup\{\varphi_d^T W_\theta^2 \varphi_d\} \quad (4.26)$$

4.3.2 动力学速度控制实验

根据上节的 ARC 一般参数调节方法，调节得到的自适应率参数矩阵和等效反馈系数，具体数值可见附录 A.2 中 ARC 仿真程序。

1) 实验对象及目标

本次实验与第三章的动力学实验类似，实验对象针对的是移动机器人动力学部分，测试 ARC 控制器在图 3.5 所示动力学反馈闭环中的性能，通过与 PID 控制器对比得到，被控量是机器人转角速度 w 。

2) 实验方法

在机器人 $u-w$ 环节中采用图 3.5 的控制框架，控制器采用自适应鲁棒控制器 (ARC)，参考输入采用的是频率为 2Hz 幅值为 1.5rad/s 的正弦转角速度信号，并使用其跟踪精度与 μ -synthesis 控制器、PID 控制器进行对比，如图 4.4。为了验证控制器在不同工况下的自适应性能，增加了机器人在额外附加 2.2kg 重物下 ARC 与 μ -synthesis 控制器、PID 控制器性能的对比实验，如图 4.5。同样采用 3.3.2 节中的三个性能评判参数，得到性能参数表如 4.1

表 4.1 实验性能参数

	e_M (rad/s)	e_F (rad/s)	$\ e\ _{rms}$ (rad/s)
PID(空载)	1.9653	0.3492	0.1275
μ -synthesis (空载)	0.9819	0.2251	0.0709
ARC(空载)	2.0628	0.2104	0.0672
PID(负载)	1.9542	0.2398	0.1245
μ -synthesis(负载)	1.2631	0.1818	0.0741
ARC(负载)	2.0984	0.1118	0.0407

3) 实验结果

从图中和表中可以得知 μ -synthesis 控制性能优于 PID，而 ARC 的瞬态或稳态控制性能相对两者均有更大的提高。由于实验平台中电机驱动器存在几个周期的死区，三者控制器均会出现大的误差跳动，以此可观察控制器的瞬态响应性能。在未加负载速度跟踪实验中，从表 4.1 可得，ARC 与 μ -synthesis 的 e_F 和 $\|e\|_{rms}$ 性能参数基本相当，且都比 PID 相应的参数小很多，代表前两者控制器的跟踪误差绝对最大值和波动范围都比 PID 小，说明了 ARC 具有与 μ -synthesis 同样甚至更好的抗摩擦力干扰能力。不过从图 4.4 可知， μ -synthesis 的瞬态响应情况稍差于

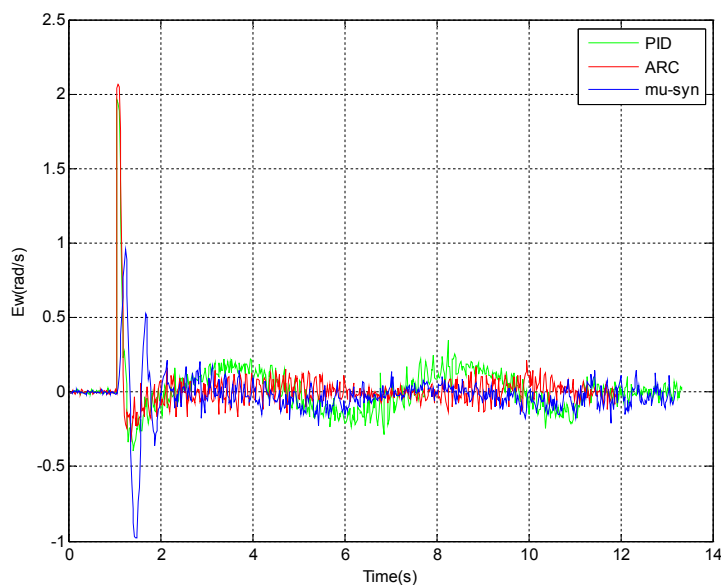


图 4.4 无负载角速度跟踪误差对比图

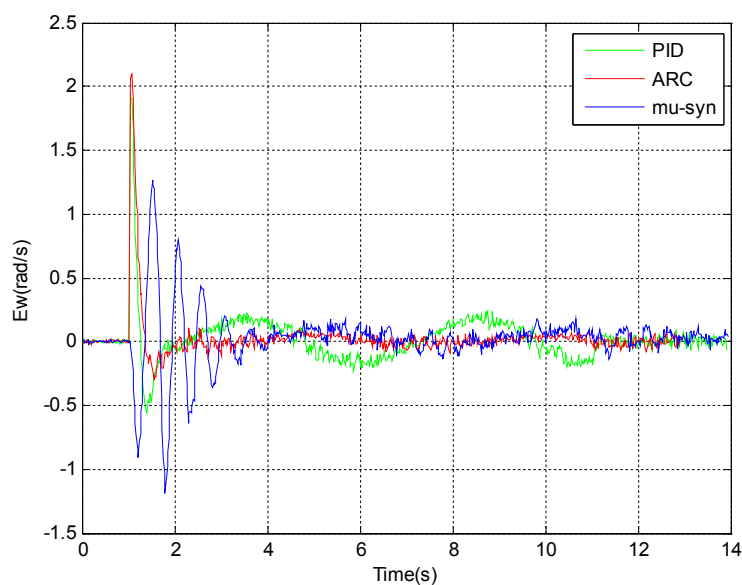


图 4.5 有负载角速度跟踪误差对比图

其它两控制器。在加上负载 2.2kg 重物后的实验中，从图 4.5 和表 4.1 可知，ARC 的 e_F 和 $\|e\|_{rms}$ 性能参数均明显高于其它两个控制器，这是因为机器人加上负载后，模型参数发生了变化，即具有了参数不确定性，实验数据说明了 ARC 在线估计模型参数的自适应性能。 μ -synthesis 的稳态性能虽然高于 PID，随着负载的增加，瞬态性能变得较差，而 ARC 的瞬态性能依然很好。图 4.5 中 ARC 的跟踪误差更稳定些，是因为添加重物后，相机支撑架晃动减小，柔性得以抑制，这样在线参数估计更为准确。综上所述，ARC 很好解决了系统中的抗干扰能力并且对系统不确定性有很强的鲁棒性能，另外自适应功能适应不同的工作负载，具有优秀的瞬态与稳态速度跟踪性能。

4.3.3 三层控制框架设计及路径规划与跟踪实验

1) 实验目标

在上次实验中，验证了 ARC 在动力学上对速度控制的精度，作为一个轮式移动机器人整体，本次实验将结合二叉树路径规划算法以体现 ARC 在更为普遍的机器人路径规划及跟踪实例中的运动控制效果。

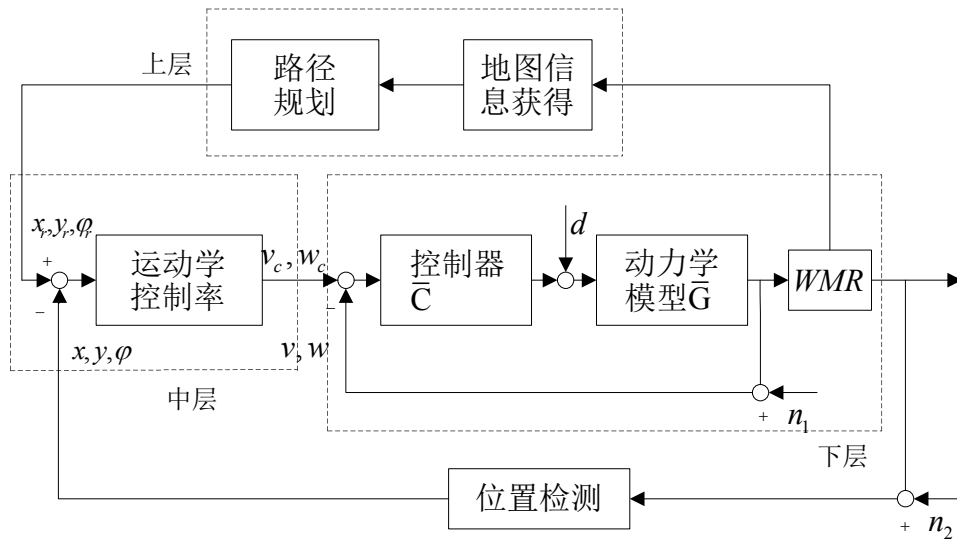


图 4.6 三层路径跟踪控制系统

2) 实验方法

为将 ARC 很好应用到路径规划及跟踪实例中去，本次实验设计了三层移动机器人自主运动控制系统，涵盖了机器人路径规划、运动学控制和动力学控制，如图 4.6 所示。上层就是二叉树路径规划算法，完成由周围含有障碍物的环境信息生成一条路程最优的路径任务，由于实验器材限制，本实验该部分离线完成；

中层是运动学控制部分,采用的是式 1.2 中轮式移动机器人一种稳定的运动学跟踪控制率,完成由三个位姿量 x_r, y_r, φ_r 到机器人两个速度量 v_c, w_c 的渐进稳定转换;下层是动力学控制部分,在本次实验中将分别采用 ARC 和 PID 控制器,ARC 结合了自适应和鲁棒控制的优越性能,能够解决实际系统中如图 4.1 所示的系统参数不确定性 $\hat{\zeta}$ 和模型不确定性 $\hat{\Phi}$,使系统达到更高的控制精度,实验中将与工业中常用的 PID 控制器进行对比验证。

中层和下层的运动学动力学控制前面章节已经介绍,下面将详细介绍上层改进后的二叉树路径规划算法。

A. 二叉树路径规划

本实验采用的二叉树路径规划使用圆形去代替相应的障碍物,得到的路径由直线和圆弧组成,直线与圆弧是以切点形式相连,保证路径平滑性并且路程最短,最后得到的路径如图 4.7。

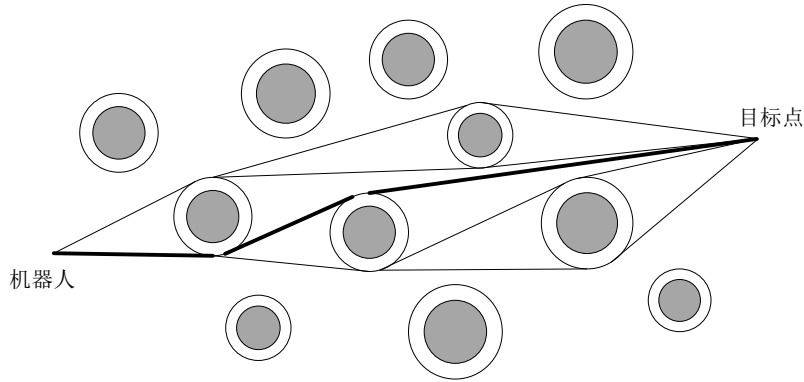


图 4.7 二叉树路径规划结果

二叉树路径的生成步骤^[60]如下:

第一步:获得确定工作环境的地图信息,即确定车体的位置 $P_s(x_s, y_s)$, 车身半径 r_s , 转向 θ_s ; 障碍物的位置及半径 $(P_1, P_2, \dots, P_n, r_1, r_2, r_n)$, 目标点 $(P_g(x_g, y_g))$;

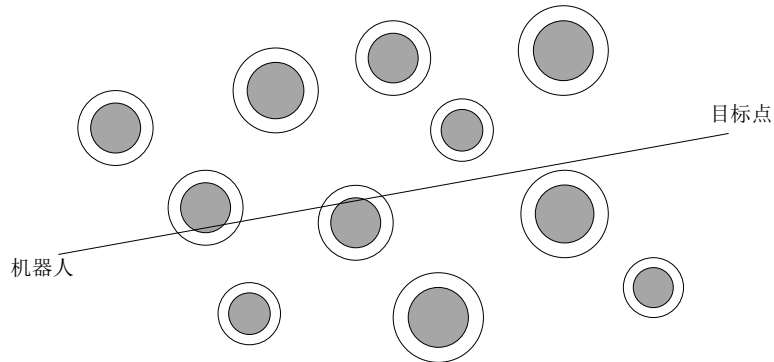


图 4.8 二叉树基准路径图

第二步：先计算从开始点到目标点的一个简单基准路径，包括一个圆弧和直线，如图 4.8 所示；

第三步：检查基准路径上是否存在障碍物，如果有，则设立一个二叉树节点，其中障碍物范围包含了障碍物实体和一段安全距离；

第四步：如果建立一个二叉树节点后，做相邻两个圆的内切线和外切线作为该节点的两个分支；

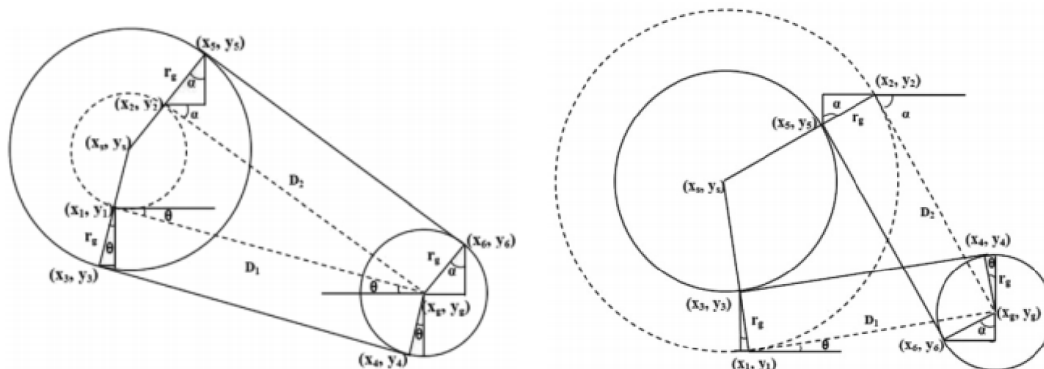


图 4.9 内外切线图

第五步：以与上一步障碍物圆的切线交点为起点，重复 2-4 步，直到最后分支的所有直线段汇交于目标点。

B. 规划算法改进

根据以上二叉树路径规划算法的一般步骤，可依据图 4.11 的流程图完成程序的编写，其中虚线表示的内嵌部分，为统一循环部分，故使用虚线框表示。程序中非常重要的一个性能是计算效率，相应程序效率的提高可以减少计算时间和计算过程中的存储空间，清晰的编程思路也利于程序的阅读与二次开发。针对以

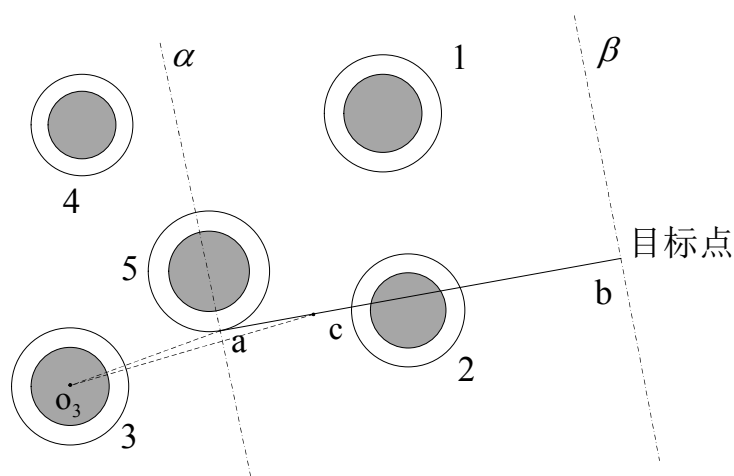


图 4.10 障碍物判断

上程序特性，在本次二叉树的具体程序编写中，提出了以下三点的算法改进方法：

1. 在二叉树路径规划过程中, 需要找到点 a 到终点 b 的一段子二叉树路径, 如图 4.10 所示。当判断 ab 线段是否与障碍物范围相交时, 对地图中所有的障碍物依次进行判断会大大增加计算负荷, 尤其是在障碍物比较多的场景中。在实际场景中, 有很多的障碍物与当前的规划路径是无关的, 即在图 4.10 中这个例子中,

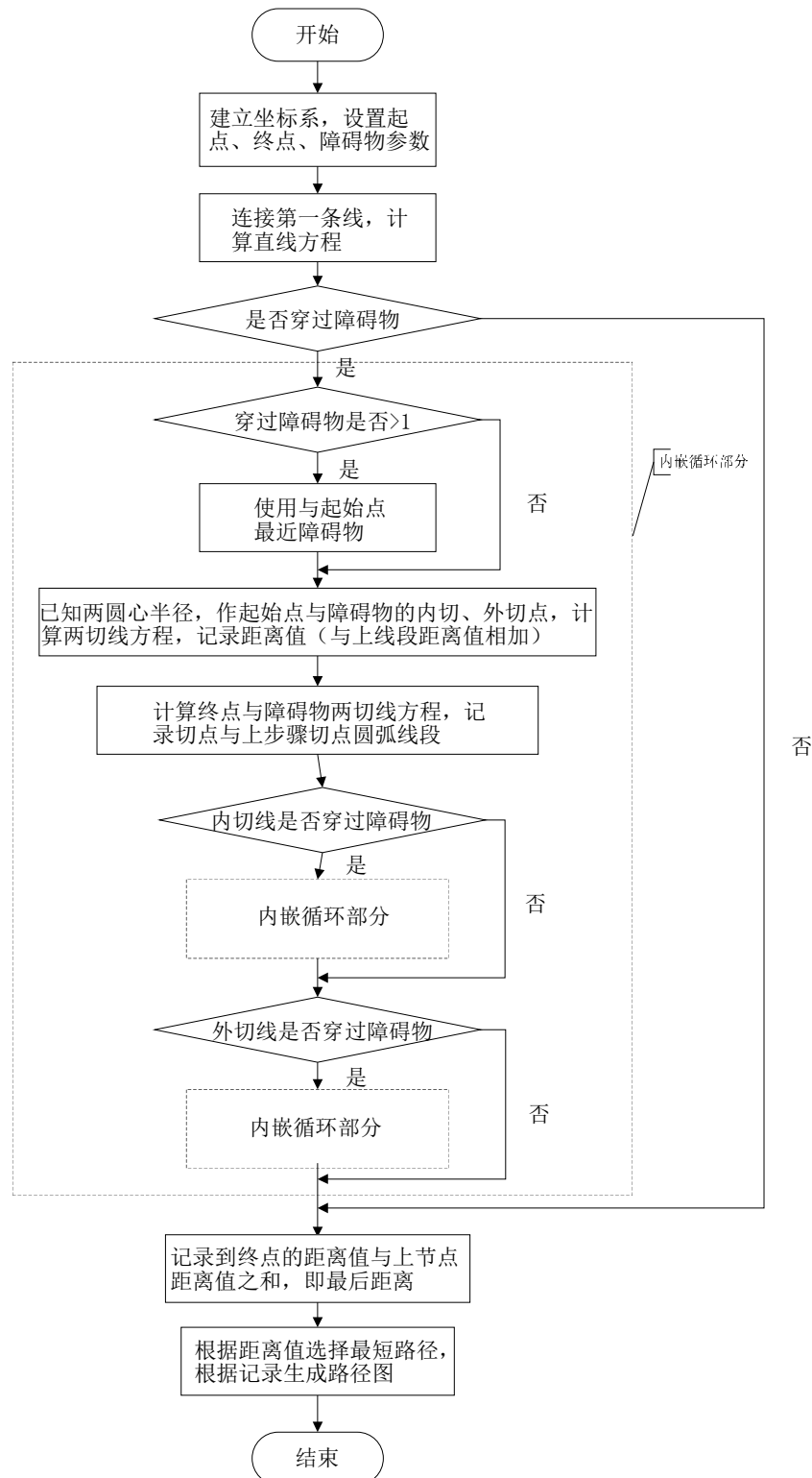


图 4.11 二叉树路径规划程序流程图

这些障碍物是不可能与 ab 线段有相交情况的。其中障碍物 1 和 2 是可能与线段 ab 相交，障碍物 3 和 4 是不可能与线段 ab 相交，障碍物 5 是上一二叉树规划路径所相交的，取其中一切点 a 为起点。

给出一个无关障碍物的判定准则：在 a 点和 b 点分别做与线段 ab 垂直的两条直线 α 和直线 β ，障碍物圆心在直线 α 和 β 之间的为有可能与线段 ab 相交的障碍物，而在直线 α 和 β 之外的为不可能与 ab 直线有相交情况，算法执行过程中可以忽略。

证明：在实际的环境地图中，障碍物之间是没有交点的，定义为：

$$\Theta(o_3) \cap \Theta(o_5) = \emptyset \quad (4.27)$$

其中 $\Theta(o_3)$ 表示障碍物 3 的范围， $\Theta(o_5)$ 表示障碍物 5 的范围，由之前的定义存在以下的几何关系：

$$ab \cap \Theta(o_5) = a \quad (4.28)$$

表示障碍物 5 的范围与 ab 线段交于切点 a 。假设圆心在直线 α 和 β 外的障碍物 3 与线段有交点 c ，有图中的几何关系知 $\angle o_3ac > 90^\circ$ ，线段 o_3c 在钝角三角形中对应 $\angle o_3ac$ ，故有：

$$\begin{cases} L(o_3c) = R_{o_3} \\ L(o_3c) > L(o_3a) \end{cases} \quad (4.29)$$

由于障碍物范围为圆心，式 4.29 表示 a 点包含在了障碍物 3 的范围内，合并式 4.28 与式 4.29 可得：

$$a \in \Theta(o_3) \cap \Theta(o_5) \neq \emptyset \quad (4.30)$$

式 4.30 与式 4.27 明显相悖，故反证得到在圆心直线 α 和 β 外的障碍物与线段 ab 无交点，规划过程中不需要考虑。

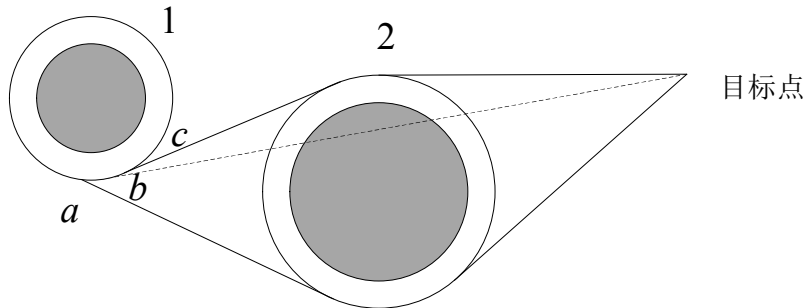


图 4.12 目标点与障碍物

2. 当发现线段与障碍物有交点时, 需要计算两障碍物之间的内切线和外切线。两个切线会在与障碍物 1 有 a 和 c 两个切点, 不同于最开始的切点 b , 如图 4.12。实际计算过程中, a 、 b 、 c 三点之间的相差很小, 采用 b 代替后续的切点 a 和 c 既能够减少计算过程, 又能减少存储空间。

3. 二叉树路径规划算法得到的一系列路径图结构类似于 C 语言数据结构中的二叉树结构, 故在编程实现中采用二叉树的结构来存储结构, 一方面使规划算法思路清晰明显, 利于之后的二次开发, 另一方面有利于算法的迭代处理, 提高程序的处理效率, 节约了存储结构。具体的二叉树程序实现代码可见附录 A.3。

3) 实验结果及分析

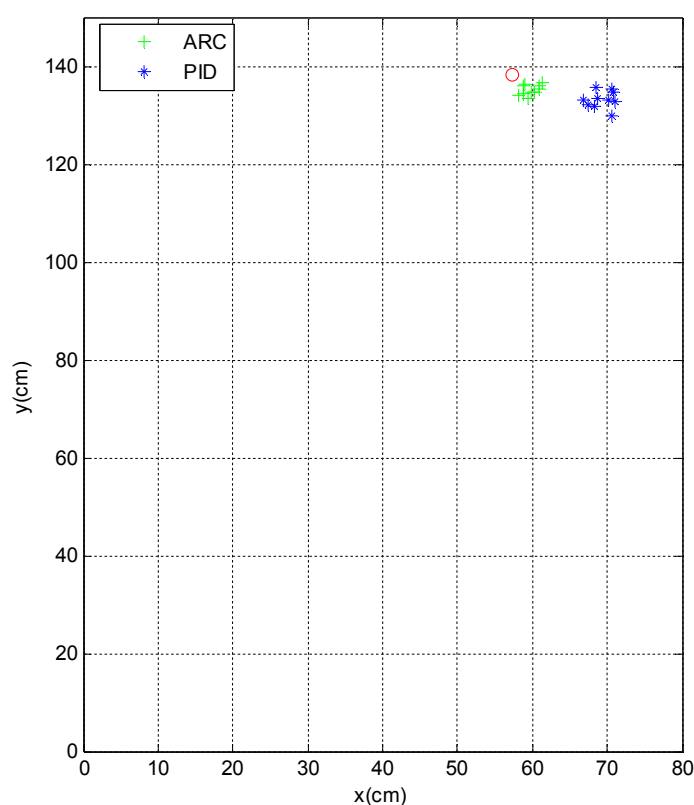


图 4.13 二叉树路径跟踪实验

本实验让移动机器人分别带有 ARC 和 PID 控制器的三层控制系统去跟踪一小段二叉树路径, 由于实验系统缺乏实时的机器人在线绝对位姿信息检测装置, 故实验中先根据已知的环境离线产生一条二叉树路径, 并计算出机器人需达到的运行速度, 然后输入给轮式移动机器人, 并且通过测量机器人最后停止时的位置与目标的误差进行分析。在 ARC 和 PID 的动力学控制下, 分别对机器人进行了

10组有效的二叉树路径跟踪实验,得到机器人停止点与终点信息如图4.13所示,其中红点为目标终点,绿色十字点为ARC控制下机器人最终位置,蓝色星形点为PID控制下机器人最终位置。误差参数的最大值与平均值可见表4.2。

表 4.2 位置误差实验结果

	e_M (cm)	$\ e\ _{rms}$ (cm)
PID	15.71	13.160
ARC	5.22	4.188

从表中的实验结构可得到由于ARC在动力学上对速度的高精度的控制,反应到上层的位置时,其精度比PID有了很大的提高。最大误差由PID的15.71cm减小到了5.22cm,平均误差由PID的13.160cm减小到了4.188cm。这也反映了在三层控制系统中,运动学与动力学上较高精度的控制性能能够很大提高路径跟踪的精度,也就是说,在底层实现ARC控制,能够使上次的二叉树路径规划在实际系统中能够更好的实现。

4.4 本章小结

本章针对轮式移动机器人在多变的工作环境中,即系统模型参数具有实时变动等情况下而设计了自适应鲁棒控制器(ARC),该控制器结合了自适应控制与鲁棒控制两者的优点,后面通过实验验证了其性能。

1) 在机器人实验平台的系统辨识中,未知摩擦力与未知柔性都会影响离线参数辨识的准确性,并且在机器人需要搬运不同重物即有不同负载的情况下,其系统模型参数也随之变化,给出了一个控制器需要更进一步考虑的因素。

2) 针对模型参数实时变化并且具有未知模型不确定性的情况下,设计了自适应鲁棒控制器(ARC),控制器中使用投影自适应率去实时估计模型参数,另外采用了具有滑模思想的一种鲁棒反馈控制项,使控制器具有一定的鲁棒性能。

3) 使用动力学速度控制实验验证了ARC在速度上的控制精度,另外针对避障这一具体应用场景设计了一种三层的控制框架,上层采用了改进的二叉树路径规划算法,下层结合了ARC,整个框架形成一体,验证了ARC在速度控制精度的提高上保证了上层二叉树路径规划算法在实际中的顺利应用,两者相辅相成。

第5章 总结及展望

5.1 总结

随着轮式移动机器人(WMR)在工业生产、家具生活、科技探险等多领域的广泛应用,机器人所处工作环境多变并更具复杂性。本文针对一般三轮移动机器人设计了两套完整的控制策略以适应于机器人在不同环境中的需要,整个控制策略涵盖了 WMR 的运动学与动力学,并在具体的实验平台中部署,控制性能得以验证,为 WMR 的具体应用提供了更高精度的控制策略参考。论文的主要内容如下:

1. 实验平台的搭建与完整建模

具体的控制策略需要在实际平台中得以验证才有意义,本文前期工作搭建了一个能通过六自由度惯性传感器测量角速度和摄像头传感器获得绝对位姿信息的轮式移动机器人平台。随后对该 WMR 系统进行了整体性的动力学建模,涵盖了直流电机电磁力学模型、驱动轮力矩传递模型及机器人车体力学模型。最后采用了基于具体实验平台的物理测量方法和时频域辨识的方法获得了模型中的各参数。

2. μ -synthesis 控制策略设计

在得到的机器人动力学模型中,了解系统的由于科氏力和离心力而具有模型耦合项,对其进行了相应的模型解耦和简化,并通过使得到的简化模型与实际系统的频率响应进行对比,分析得到了系统在低频段存在一定的摩擦力的影响,在相对高频段存在模型参数不确定性,为影响系统控制精度的主要因素。设计了 μ -synthesis 控制器,该控制器能够在系统达到的频宽内兼容地解决两者的影响。最后,进行了动力学速度控制实验和位置跟踪控制实验验证了其控制性能。为轨迹跟踪提供了一种有效的控制策略。其中位置跟踪控制实验采用的是跟踪椭圆轨迹线这一实际场景,采用的内外环控制框架将 μ -synthesis 控制器很好应用到一般硬件的实验平台上。

3. 自适应鲁棒控制(ARC)策略设计

由于离线模型参数辨识精度受各种因素影响以及变化的负载工况导致系统具有参数与模型不确定性。本文针对此情况设计了自适应鲁棒控制器(ARC),

控制器中使用投影自适应率去实时估计模型参数,另外采用了具有滑模思想的一种鲁棒反馈控制项,使控制器具有一定的鲁棒性能。最后,使用动力学速度控制实验验证了 ARC 在速度上的控制精度,另外设计了一种三层的控制框架,上层采用了改进的二叉树路径规划算法,下层结合了 ARC,整个框架形成一体,验证了 ARC 在路径规划方面给予整个机器人运动控制精度提高的作用。

以上的机器人模型建立与控制系统设计针对的是两轮差速驱动的三轮移动机器人,即运动自由度为 2,舵性自由度为 0 的一类机器人,具体平台采用的是 NI 机器人开发包,不过整个设计思路可以给该类 (2,0) 机器人控制系统设计提供高精度且具有实际应用价值的参考。

5.2 展望

目前课题处于初期发展阶段,已完成的控制策略研究主要偏向于 WMR 底层动力学上的研究,保证了系统在动力学上具有很好的自适应和抗干扰能力,但由于实验条件及时间上的限制,尚有一些研究工作可待开展,现提出希望对同领域的科研者有所帮助。

1. 移动机器人在线路径规划

本文中在 WMR 动力学控制策略方向提出了 μ -synthesis 和 ARC 两种控制方法,提高 WMR 在不同工况中的控制精度,也完成了轨迹线跟踪与二叉树路径规划的验证实验,不过由于实验平台的限制,路径规划尚在离线中完成。在日益智能化的生产及日常生活中,高效可靠的在线路径规划将能更大范围地提高 WMR 的应用。

2. 多机器人协调运动控制

WMR 所处的环境往往是实际变动的,多 WMR 的协调控制更为复杂。通常来说,相互协调的 n 个 WMR 系统的能力远大于一个 WMR 的 n 倍,其控制算法要求具有更高的实时性,具体很高的研究价值。

参考文献

- [1] 徐国保, 尹怡欣, 周美娟. 智能移动机器人技术现状及展望[J]. 机器人技术与应用, 2008 (2): 29-34.
- [2] Siegwart R, Nourbakhsh I R, Scaramuzza D. Introduction to autonomous mobile robots[M]. MIT press, 2011.
- [3] Raja P, Pugazhenth S. Optimal path planning of mobile robots: A review[J]. International Journal of Physical Sciences, 2012, 7(9): 1314-1320.
- [4] Campion G, Bastin G, D'Andréa-Novel B. Structural properties and classification on kinematic and dynamic models of wheeled mobile robots[J]. Nelineinaya Dinamika [Russian Journal of Nonlinear Dynamics], 2011, 7(4): 733-769.
- [5] Moret E N. Dynamic modeling and control of a car-like robot[D]. Virginia Polytechnic Institute and State University, 2003.
- [6] DeSantis R M. Modeling and path-tracking control of a mobile wheeled robot with a differential drive[J]. Robotica, 1995, 13(04): 401-410.
- [7] Kozłowski K, Pazderski D. Modeling and control of a 4-wheel skid-steering mobile robot[J]. Int. J. Appl. Math. Comput. Sci, 2004, 14(4): 477-496.
- [8] Tian Y, Sidek N, Sarkar N. Modeling and control of a nonholonomic wheeled mobile robot with wheel slip dynamics[C]//Computational Intelligence in Control and Automation, 2009. CICA 2009. IEEE Symposium on. IEEE, 2009: 7-14.
- [9] d'Andrea-Novel B, Bastin G, Campion G. Modelling and control of non-holonomic wheeled mobile robots[C]//Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on. IEEE, 1991: 1130-1135.
- [10] Gao H, Chen F, Li D, et al. Movement Simulation for Wheeled Mobile Robot Based on Stereo Vision[M]//Advanced Research on Electronic Commerce, Web Application, and Communication. Springer Berlin Heidelberg, 2011: 396-401.
- [11] Muir P F, Neuman C P. Kinematic modeling of wheeled mobile robots[J]. Journal of robotic systems, 1987, 4(2): 281-340.
- [12] Low K H, Leow Y P. Kinematic modeling, mobility analysis and design of wheeled mobile robots[J]. Advanced Robotics, 2005, 19(1): 73-99.
- [13] Gracia L, Tornero J. Kinematic modeling and singularity of wheeled mobile robots[J]. Advanced Robotics, 2007, 21(7): 793-816.
- [14] Ollero A, Heredia G. Stability analysis of mobile robot path tracking[C]//Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on. IEEE, 1995, 3: 461-466.
- [15] Yuan G, Yang S X, Mittal G S. Tracking control of a mobile robot using a neural dynamics based approach[C]//Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. IEEE, 2001, 1: 163-168.

- [16] Hu Y, Yang S X. A fuzzy neural dynamics based tracking controller for a nonholonomic mobile robot[C]//Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on. IEEE, 2003, 1: 205-210.
- [17] Kim D H, Oh J H. Tracking control of a two-wheeled mobile robot using input-output linearization[J]. Control Engineering Practice, 1999, 7(3): 369-373.
- [18] Yamamoto Y. Control and coordination of locomotion and manipulation of a wheeled mobile manipulators[D]. Graduate School of Arts and Sciences, University of Pennsylvania, 1994.
- [19] Kanayama Y, Kimura Y, Miyazaki F, et al. A stable tracking control method for an autonomous mobile robot[C]//Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on. IEEE, 1990: 384-389.
- [20] Watanabe K, Tang J, Nakamura M, et al. A fuzzy-Gaussian neural network and its application to mobile robot control[J]. Control Systems Technology, IEEE Transactions on, 1996, 4(2): 193-199.
- [21] Saffiotti A, Ruspini E H, Konolige K. Using fuzzy logic for mobile robot control[M]//Practical applications of fuzzy technologies. Springer US, 1999: 185-205.
- [22] Mohareri O, Dhaouadi R, Rad A B. Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks[J]. Neurocomputing, 2012, 88: 54-66.
- [23] Klančar G, Škrjanc I. Tracking-error model-based predictive control for mobile robots in real time[J]. Robotics and Autonomous Systems, 2007, 55(6): 460-469.
- [24] Shojaei K, Shahri A M, Tarakameh A, et al. Adaptive trajectory tracking control of a differential drive wheeled mobile robot[J]. Robotica, 2011, 29(3): 391-402.
- [25] Yeh Y C, Li T H S, Chen C Y. Adaptive fuzzy sliding-mode control of dynamic model based car-like mobile robot[J]. International Journal of Fuzzy Systems, 2009, 11(4): 272-286.
- [26] Kühn F, Gomes J, Fetter W. Mobile robot trajectory tracking using model predictive control[C]//II IEEE latin-american robotics symposium. 2005.
- [27] Park B S, Park J B, Choi Y H. Adaptive observer-based trajectory tracking control of nonholonomic mobile robots[J]. International Journal of Control, Automation and Systems, 2011, 9(3): 534-541.
- [28] Park B S, Yoo S J, Park J B, et al. A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots[J]. Control Systems Technology, IEEE Transactions on, 2010, 18(5): 1199-1206.
- [29] Sarkar N, Yun X, Kumar V. Control of Mechanical Systems With Rolling Constraints Application to Dynamic Control of Mobile Robots[J]. The International Journal of Robotics Research, 1994, 13(1): 55-69.
- [30] Fierro R, Lewis F L. Control of a nonholonomic mobile robot: backstepping kinematics into dynamics[C]//Decision and Control, 1995., Proceedings of the 34th IEEE Conference on. IEEE, 1995, 4: 3805-3810.
- [31] Dierks T, Jagannathan S. Control of nonholonomic mobile robot formations: backstepping kinematics into dynamics[C]//Control Applications, 2007. CCA 2007. IEEE International Conference on. IEEE, 2007: 94-99.

- [32] Chwa D. Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization[J]. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2010, 40(6): 1285-1295.
- [33] Solea R, Filipescu A, Minzu V, et al. Sliding-mode trajectory-tracking control for a four-wheel-steering vehicle[C]//Control and Automation (ICCA), 2010 8th IEEE International Conference on. IEEE, 2010: 382-387.
- [34] Park B S, Yoo S J, Park J B, et al. Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty[J]. Control Systems Technology, IEEE Transactions on, 2009, 17(1): 207-214.
- [35] Yeh Y C, Li T H S, Chen C Y. Adaptive fuzzy sliding-mode control of dynamic model based car-like mobile robot[J]. International Journal of Fuzzy Systems, 2009, 11(4): 272-286.
- [36] Liu Y, Zhang Y, Wang H. Tracking control of wheeled mobile robots based on sliding-mode control[C]//Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on. IEEE, 2011: 1787-1790.
- [37] 陈鹏程. 轮式移动机器人轨迹跟踪控制研究[D]. 江南大学, 2007.
- [38] 张元良. 轮式移动机器人轨迹跟踪控制的特点与方法[J]. 数字技术与应用, 2011, (11):18-19.
- [39] 刘钰, 陆剑锋, 何云红, 等. 移动机器人神经网络自适应控制[J]. 哈尔滨理工大学学报, 2012, 17(6): 30-35.
- [40] 韩龙, 刘国栋. 一种基于迭代学习的移动机器人轨迹跟踪控制方法[J]. 计算机系统应用, 2012, (4):59-63.
- [41] 席雷平, 段莲飞, 江涛. 非完整四轮式移动机器人反演轨迹跟踪控制[J]. 计算机测量与控制, 2012, 20(008): 2143-2145.
- [42] 吴剑, 黄晓娟, 江维. 基于 Back-stepping 的轮式机器人轨迹跟踪[J]. 南昌航空大学学报 (自然科学版) ISTIC, 2012, 26(4).
- [43] 黄麟, 奚茂龙, 孙俊. 改进 QPSO 算法的移动机器人轨迹跟踪控制方法[J]. 计算机工程与应用, 2012, 48(34): 230-236.
- [44] 张鑫, 刘凤娟, 闫茂德. 基于动力学模型的轮式移动机器人自适应滑模轨迹跟踪控制[J]. 机械科学与技术, 2012, 31(001): 107-112.
- [45] 张扬名, 刘国荣, 杨小亮. 基于滑模变结构的移动机器人轨迹跟踪控制[J]. 计算机工程, 2013, 39(5): 160-164.
- [46] Borenstein J, Koren Y. Real-time obstacle avoidance for fast mobile robots[J]. Systems, Man and Cybernetics, IEEE Transactions on, 1989, 19(5): 1179-1187.
- [47] Koren Y, Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation[C]//Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on. IEEE, 1991: 1398-1404.
- [48] Ge S S, Cui Y J. New potential functions for mobile robot path planning[J]. IEEE Transactions on robotics and automation, 2000, 16(5): 615-620.

- [49] Masoud A A. A harmonic potential field approach for joint planning and control of a rigid, separable nonholonomic, mobile robot[J]. *Robotics and Autonomous Systems*, 2013, 61(6): 593-615.
- [50] Huang L. Velocity planning for a mobile robot to track a moving target—a potential field approach[J]. *Robotics and Autonomous Systems*, 2009, 57(1): 55-63.
- [51] Boschian V, Pruski A. Grid modeling of robot cells: a memory-efficient approach[J]. *Journal of Intelligent and Robotic Systems*, 1993, 8(2): 201-223.
- [52] Zhang Z. Flexible camera calibration by viewing a plane from unknown orientations[C] //Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. IEEE, 1999, 1: 666-673.
- [53] Narendra K S, Annaswamy A M. Robust adaptive control in the presence of bounded disturbances[J]. *Automatic Control, IEEE Transactions on*, 1986, 31(4): 306-315.
- [54] Yao B, Tomizuka M. Smooth robust adaptive sliding mode control of manipulators with guaranteed transient performance[J]. *Journal of dynamic systems, measurement, and control*, 1996, 118(4): 764-775.
- [55] VADIM I U. Survey paper variable structure systems with sliding modes[J]. *IEEE Transactions on Automatic control*, 1977, 22(2).
- [56] Yao B, Tomizuka M. Smooth robust adaptive sliding mode control of manipulators with guaranteed transient performance[J]. *Journal of dynamic systems, measurement, and control*, 1996, 118(4): 764-775.
- [57] Yao B, Tomizuka M. Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form[J]. *Automatica*, 1997, 33(5): 893-900.
- [58] Yao B, Tomizuka M. Adaptive robust control of MIMO nonlinear systems in semi-strict feedback forms[J]. *Automatica*, 2001, 37(9): 1305-1321.
- [59] Xu L, Yao B. Adaptive robust precision motion control of linear motors with negligible electrical dynamics: theory and experiments[J]. *Mechatronics, IEEE/ASME Transactions on*, 2001, 6(4): 444-452.
- [60] Rashid A T, Ali A A, Frasca M, et al. Path planning with obstacle avoidance based on visibility binary tree algorithm[J]. *Robotics and Autonomous Systems*, 2013, 61:1440–1449.

附录 A

在轮式移动机器人控制系统的研究中，包含了实验平台的搭建、系统建模、控制器设计与实验验证等几大块工作，下面附上工作中设计的 matlab 与 C 语言等程序代码内容

A.1 电机辨识多组数据自动频域分析 matlab 程序

在第二章的电机电感 L 和电机转子和轮子的转动惯量之和 J_w 参数辨识中，采集了 70 组电机输入电压与输出转速数据，每组数据对应不同的频率值，需对每组进行频域分析，编写以下程序可以由该 70 组数据得到直流电机的频域响应值。

```
%逐步扫频信号频域处理 MATLAB 函数%
i=0;%用于记录相幅值比，相位差等等
first=input('请输入计算开始频率：');
step=input('请输入步长：');
last=input('请输入计算结束频率：');
for n=first:step:last;%循环计算
i=i+1;
%-----文件循环命名-----
number=n;
numberstr=num2str(number);
name=strcat(numberstr,'.mat');%文件名
%-----读取文件-----
load(name);%读取 mat 文件
Time=ans(3,:);%分别赋值
U=ans(1,:);
Vs=ans(2,:);
%-----读取文件-----
a=mean(Vs);
V=Vs-a;
%-----自动获取长度-----
L=length(Time);%长度数值
Fs=50; % Sampling frequency
T=1/Fs; % Sample time
%-----自动获取长度-----
t=Time; % Time vector
x=U;
y=V;
% plot(t,x,'r',t,y,'b--')
% title('Signal ')
```

```

% xlabel('time (seconds)')
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
X = fft(x,NFFT)/L;
Y = fft(y,NFFT)/L;
%-----求取 X,Y 绝对值的最大值以及相对位置-----
X1=2*abs(X);
[XC,XI]=max(X1); %XC 表示幅值最大点, XI 表示最大点所在位置
Y1=2*abs(Y);
YI=XI;
YC=Y1(YI); %YC 表示幅值最大点, YI 表示最大点所在位置
Mag(i)=YC/XC;%幅值比
%-----求取 X,Y 绝对值的最大值以及相对位置-----

f = Fs/2*linspace(0,1,NFFT/2+1);
Pha1=angle(X);
Pha2=angle(Y);
%-----求相位差-----
Phase(i)=(Pha2(XI)-Pha1(YI))*180/pi; % 相位差
%-----求相位差-----

%-----频率-----
Fre(i)=n;
%-----频率-----
end

%下面是文档存储, 以频率开始 first 命名
numberstr=num2str(first);
name=strcat('Fre',numberstr,'.mat');%文件名

%如果已经存在该文档, 先删除以前的文档, 然后存储现在的
if ~isdir(name)
    delete(name)
end
Fre=Fre*0.2;
Phase(11:70)=Phase(11:70)-360;
save(name,'Fre','Mag','Phase')
clear;
clc;

```

A.2 ARC 仿真参数定义 C 程序

下面程序为第四章的 ARC 控制器设计各自适应率及反馈项的相关定义，代入到 matlab 可进行相应的仿真。

```
#define S_FUNCTION_NAME  ARC_sf
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include "math.h"

#define NUM_INPORTs      1
#define NUM_INPORTWIDTH  5
#define NUM_OUTPORTs     1
#define NUM_OUTPORTWIDTH 5
#define NUM_RWORK         5

#define NUM_PARAMS        13
#define Param1  ssGetSFcnParam(S,0) /*A1_hat_0 */
#define Param2  ssGetSFcnParam(S,1) /* A2_hat_0 */
#define Param3  ssGetSFcnParam(S,2) /* A3_hat_0 */
#define Param4  ssGetSFcnParam(S,3) /* d_hat_0 */
#define Param5  ssGetSFcnParam(S,4) /* Ts */
#define Param6  ssGetSFcnParam(S,5) /* gamma_A1 */
#define Param7  ssGetSFcnParam(S,6) /* gamma_A2 */
#define Param8  ssGetSFcnParam(S,7) /* gamma_A3*/
#define Param9  ssGetSFcnParam(S,8) /* gamma_d*/
#define Param10 ssGetSFcnParam(S,9) /* k1 */
#define Param11 ssGetSFcnParam(S,10) /* ks1 */
#define Param12 ssGetSFcnParam(S,11) /* epsilon */
#define Param13 ssGetSFcnParam(S,12) /* maximum
disturbance*/

/*
 * S-function routines
 */

static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, NUM_PARAMS); /* Number of expected
parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S))
    {
        return;
    }

    /* Input : ym , ym_dot, ym_ddot, y, y_dot */
    if (!ssSetNumInputPorts(S, NUM_INPORTs)) return;
    ssSetInputPortWidth(S, 0, NUM_INPORTWIDTH);
```

```

    ssSetInputPortDirectFeedThrough(S, 0, 1);

/* Output : u, Me_hat, B_hat, Fsc_hat */
if (!ssSetNumOutputPorts(S, NUM_OUTPUTS)) return;
ssSetOutputPortWidth(S, 0, NUM_OUTPUTWIDTH);

ssSetNumSampleTimes(S, 1);
ssSetNumRWork(S, NUM_RWORK);

ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

static void mdlInitializeSampleTimes(SimStruct *S)
{
    real_T ts = *mxGetPr(Param5);

    ssSetSampleTime(S, 0, ts);
    ssSetOffsetTime(S, 0, 0.0);
}

#define MDL_INITIALIZE_CONDITIONS

static void mdlInitializeConditions(SimStruct *S)
{
    real_T A1_hat0 = *mxGetPr(Param1);
    real_T A2_hat0 = *mxGetPr(Param2);
    real_T A3_hat0 = *mxGetPr(Param3);
    real_T d_hat0 = *mxGetPr(Param4);

    real_T *z = ssGetRWork(S);

    z[0] = A1_hat0;
    z[1] = A2_hat0;
    z[2] = A3_hat0;
    z[3] = d_hat0;
    z[4] = 0.0;

}

static void mdlOutputs(SimStruct *S, int_T tid)
{
    real_T *out =
ssGetOutputPortRealSignal(S, 0);
    InputRealPtrsType uPtrs =
ssGetInputPortRealSignalPtrs(S, 0);
    real_T *z = ssGetRWork(S);

    real_T Ts = *mxGetPr(Param5);
    real_T gamma_A1 = *mxGetPr(Param6);
    real_T gamma_A2 = *mxGetPr(Param7);
    real_T gamma_A3 = *mxGetPr(Param8);

```

```

real_T  gamma_d  = *mxGetPr(Param9);

real_T  k1       = *mxGetPr(Param10);
real_T  ks1      = *mxGetPr(Param11);
real_T  epsilon  = *mxGetPr(Param12);
real_T  delta    = *mxGetPr(Param13);

real_T  u, wm, wm_dot, wm_ddot, w, w_dot;
real_T  e, e_dot, p, PI;
real_T  h, ua, us1, us2;
real_T  phi[5], phi_lim;
real_T  theta[5], theta_dot[5], theta_max[5],
theta_min[5], theta_lim, Iabort;
int      j;

/* get the boundary values for the estimates */
theta_max[0] = 0.006;      /* maximun value for A1*/
theta_min[0] = 0.002;      /* minimum value for A1*/
theta_max[1] = 0.45;       /* maximun value for A2 */
theta_min[1] = 0.25;       /* minimum value for A2 */
theta_max[2] = 5;          /* maximun value for A3 */
theta_min[2] = 3;          /* minimum value for A3 */
theta_max[3] = 1;          /* maximun value for d */
theta_min[3] = -1;         /* minimum value for d */
theta_lim    = 6;          /* max. varation of parameters */

/* get the current input and system states*/
wm          = *uPtrs[0];    /* the desired trajectory */
wm_dot      = *uPtrs[1];    /* the desired velocity */
wm_ddot     = *uPtrs[2];    /* the desired acceleration */
w           = *uPtrs[3];    /* the position signal */
w_dot       = *uPtrs[4];    /* the velocity signal */

/* restore the variables */
theta[0]    = z[0];
theta[1]    = z[1];
theta[2]    = z[2];
theta[3]    = z[3];
Iabort      = z[4];

/* output */
out[0] = theta[0];          /* A1_hat*/
out[1] = theta[1];          /* A2_hat */
out[2] = theta[2];          /* A3_hat */
out[3] = theta[3];          /* d_hat */

/* obtain the error signal */
e = w - wm;
e_dot = w_dot - wm_dot;
p = e_dot + k1*e;

```

```

/* form the current regressors */
phi[0] = -(wm_ddot );
phi[1] = -wm_dot;
phi[2] = -wm;
phi[3] = 1;

/* theta_dot = gamma*phi*p */
theta_dot[0] = gamma_A1*phi[0]*p;
theta_dot[1] = gamma_A2*phi[1]*p;
theta_dot[2] = gamma_A3*phi[2]*p;
theta_dot[3] = gamma_d*phi[3]*p;

for (j=0;j<=3;j++)
    theta[j] = theta[j] + theta_dot[j]*Ts;

/* discrete projection */
for (j=0; j<=3; j++)
{
    if (theta[j] > theta_max[j])
        theta[j] = theta_max[j];
    else if (theta[j] < theta_min[j])
        theta[j] = theta_min[j];
}

/* store the updated theta and control input*/
z[0] = theta[0];
z[1] = theta[1];
z[2] = theta[2];
z[3] = theta[3];

/* control law */
ua=0;
for (j=0;j<=3;j++)
    ua = -phi[j]*theta[j]+ua;

// h = phi_lim*theta_lim;
us1 = - ks1*p;
u = ua + us1 ;
if (fabs(e)>1)
    Iabort=1;
if (Iabort > 0.5)
    u=-ks1*w_dot;

out[4] = u;          /* control input */
z[4]= Iabort;
}

static void mdlTerminate(SimStruct *S){}

#ifdef MATLAB_MEX_FILE    /* Is this file being compiled as

```

```

a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfuns.h"      /* Code generation registration
function */
#endif

```

A.3 二叉树路径规划算法程序实现

下面程序段采用 C 语言中的二叉树结构在已知的地图环境信息中根据改进的二叉树路径规划算法生成最短路径，包含了多层数据和算法的迭代。

```

#pragma once

double Start_p[2]={0,0}; //起始点
double End_p[2]={4000,20}; //终点(定义规则，终点x值大于起始点)
int Num_ob=6;
double
P_ob[6][2]={{900,-50},{1200,300},{2540,60},{3040,-500}};
double R_ob[6]={516,400,7421,700};

#include <stdio.h>
#include <math.h>
#include "map.h"
#include <stdlib.h>
#include "malloc.h"

#define NO_OB 999

double* Line(double start[2],double end[2])
//计算直线方程,a*x+b*y+c=0
{
    double line[3];
    line[0]=(end[1]-start[1]);
    line[1]=start[0]-end[0];
    line[2]=(end[0]-start[0])*start[1]-(end[1]-start[1])*s
tart[0];
    return(line);
}

int Judge(double sta[2], double end[2], double a, double
b,double c)
//判断在两点内与直线相交的的障碍物，并记录相交障碍物序号（按序号顺序
执行障碍物操作）
{
    int n=NO_OB,i;
    int Y_N; // 障碍物是否需要考虑标示

```

```

double aa[2],bb[2],cc[2];
double Num,Den,Dis;
aa[0]=-(b/a);
aa[1]=-(b/a);
bb[0]=1;
bb[1]=1;
cc[0]=-(aa[0]*sta[0]+bb[0]*sta[01]);
cc[1]=-(aa[1]*end[0]+bb[1]*end[01]);
for(i=0;i<Num_ob;i++)
{
    if (cc[1]>cc[0])
    {
        if ((aa[0]*P_ob[i][0]+bb[0]*P_ob[i][1]+cc[0]<-1)
            &(aa[1]*P_ob[i][0]+bb[1]*P_ob[i][1]+cc[1]>0))
        {
            Y_N=1;
        }
        else
        {
            Y_N=0;
        }
    }
    else
    {
        if ((aa[0]*P_ob[i][0]+bb[0]*P_ob[i][1]+cc[0]>1)
            &(aa[1]*P_ob[i][0]+bb[1]*P_ob[i][1]+cc[1]<0))
        {
            Y_N=1;
        }
        else
        {
            Y_N=0;
        }
    }
    if (Y_N)//判断障碍物中心是否在两点内
    {
        Num=fabs(a*P_ob[i][0]+b*P_ob[i][1]+c);
        Den=sqrt(a*a+b*b);
        Dis=Num/Den;
        if (Dis<R_ob[i])
        {
            n=i;
            break;
        }
    }
}
return(n);
}
// 点
struct point_t {
    double x, y;

```

```

};
// 圆
struct circle_t {
    struct point_t center;
    double r;
};
// 浮点数判同
int double_equals(double const a, double const b)
{
    static const double ZERO = 1e-3;
    return fabs(a - b) < ZERO;
}
// 两点之间距离的平方
double distance_sqr(struct point_t const* a, struct point_t
const* b)
{
    return (a->x - b->x) * (a->x - b->x) + (a->y - b->y) * (a->y
- b->y);
}

// 两点之间的距离
double distance(struct point_t const* a, struct point_t
const* b)
{
    return sqrt(distance_sqr(a, b));
}

double* intersection(struct circle_t circles[])
{
    double a, b, c, p, q, r;
    double cos_value[2], sin_value[2];
    double points[4];
    struct point_t ppoints[2];

    a = 2.0 * circles[0].r * (circles[0].center.x -
circles[1].center.x);
    b = 2.0 * circles[0].r * (circles[0].center.y -
circles[1].center.y);
    c = circles[1].r * circles[1].r - circles[0].r *
circles[0].r
        - distance_sqr(&circles[0].center,
&circles[1].center);
    p = a * a + b * b;
    q = -2.0 * a * c;

    r = c * c - b * b;
    cos_value[0] = (sqrt(q * q - 4.0 * p * r) - q) / p / 2.0;
    cos_value[1] = (-sqrt(q * q - 4.0 * p * r) - q) / p / 2.0;
    sin_value[0] = sqrt(1 - cos_value[0] * cos_value[0]);
    sin_value[1] = sqrt(1 - cos_value[1] * cos_value[1]);
}

```

```

    points[0] = circles[0].r * cos_value[0] +
circles[0].center.x;
    points[2] = circles[0].r * cos_value[1] +
circles[0].center.x;
    points[1]= circles[0].r * sin_value[0] +
circles[0].center.y;
    points[3] = circles[0].r * sin_value[1] +
circles[0].center.y;

    ppoints[0].x=points[0];
    ppoints[0].y=points[1];
    ppoints[1].x=points[2];
    ppoints[1].y=points[3];
    if (!double_equals(distance_sqr(&ppoints[0],
&circles[1].center),
        circles[1].r * circles[1].r)) {
        points[1] = circles[0].center.y - circles[0].r *
sin_value[0];
    }
    if (!double_equals(distance_sqr(&ppoints[1],
&circles[1].center),
        circles[1].r * circles[1].r)) {
        points[3]= circles[0].center.y - circles[0].r *
sin_value[1];
    }
    ppoints[0].x=points[0];
    ppoints[0].y=points[1];
    ppoints[1].x=points[2];
    ppoints[1].y=points[3];
    if (double_equals(ppoints[0].y, ppoints[1].y)
        && double_equals(ppoints[0].x, ppoints[1].x)) {
        if (points[1]> 0) {
            points[3] = -points[3];
        } else {
            points[1] = -points[1];
        }
    }
    return (points);
}

typedef struct Node          /* 二叉链表存储结构 */
{
    double Best_points[2]; // 存放路径点
    double Best_cicles[2]; //存放路径障碍物圆形
    double Best_distance; //存放障碍物
    struct Node *lchild,*rchild;
    struct Node *parent;
}BiTree;

void EndBiTree(BiTree *T) //无相交最后节点为终点

```

```

{
    BiTree *T1;
    T1=(BiTree *)malloc(sizeof(BiTree));
    T->lchild=T1;
    T->rchild=NULL;
    T1->Best_points[0]=End_p[0];
    T1->Best_points[1]=End_p[1];
    T1->Best_cicles[0]=0;
    T1->Best_cicles[1]=0;
    T1->Best_distance=distance(&T->Best_points,
&T1->Best_points)+T->Best_distance;
    T1->lchild=NULL;
    T1->rchild=NULL;
    T1->parent=T;
}

void TwoBiTree(BiTree *T,int N_ob)//与终点连线与障碍物有交点时
{
    BiTree *T11=(BiTree *)malloc(sizeof(BiTree));
    BiTree *T12=(BiTree *)malloc(sizeof(BiTree));
    BiTree *T21=(BiTree *)malloc(sizeof(BiTree));
    BiTree *T22=(BiTree *)malloc(sizeof(BiTree));

    struct circle_t circles[2];//临时存放两圆参数
    double *points;//临时存放交点
    double *line;
    struct point_t ppoints[2];

    circles[0].center.x=T->Best_points[0];
    circles[0].center.y=T->Best_points[1];
    circles[1].center.x=P_ob[N_ob][0];
    circles[1].center.y=P_ob[N_ob][1];
    circles[1].r=R_ob[N_ob];
    circles[0].r=sqrt(distance_sqr(&circles[0].center,&cir
cles[1].center)-circles[1].r*circles[1].r);
    points=intersection(circles);

    ppoints[0].x=points[0];
    ppoints[0].y=points[1];
    ppoints[1].x=points[2];
    ppoints[1].y=points[3];

    T->lchild=T11;
    T->rchild=T12;
    T11->Best_points[0]=ppoints[0].x;
    T11->Best_points[1]=ppoints[0].y;
    T11->Best_cicles[0]=0;
    T11->Best_cicles[1]=0;
    T11->Best_distance=distance(&T->Best_points,
&T11->Best_points)+T->Best_distance;

```

```

T12->Best_points[0]=ppoints[1].x;
T12->Best_points[1]=ppoints[1].y;
T12->Best_cicles[0]=0;
T12->Best_cicles[1]=0;
T12->Best_distance=distance(&T->Best_points,
&T11->Best_points)+T->Best_distance;

circles[0].center.x=End_p[0];
circles[0].center.y=End_p[1];
circles[1].center.x=P_ob[N_ob][0];
circles[1].center.y=P_ob[N_ob][1];
circles[1].r=R_ob[N_ob];
circles[0].r=sqrt(distance_sqr(&circles[0].center,&cir
cles[1].center)-circles[1].r*circles[1].r);
points=intersection(circles);

ppoints[0].x=points[0];
ppoints[0].y=points[1];
ppoints[1].x=points[2];
ppoints[1].y=points[3];

//T21
if
(distance(&ppoints[0],&T11->Best_points)<distance(&ppoint
s[1],&T11->Best_points))
{
    T11->lchild=T21;
    T11->rchild=NULL;
    T11->parent=T;
    T21->Best_points[0]=ppoints[0].x;
    T21->Best_points[1]=ppoints[0].y;
    T21->Best_cicles[0]=P_ob[N_ob][0];
    T21->Best_cicles[1]=P_ob[N_ob][1];
    T21->Best_distance=distance(&T11->Best_points,
&T21->Best_points)+T11->Best_distance;
}
else
{
    T11->lchild=T21;
    T11->rchild=NULL;
    T11->parent=T;
    T21->Best_points[0]=ppoints[1].x;
    T21->Best_points[1]=ppoints[1].y;
    T21->Best_cicles[0]=P_ob[N_ob][0];
    T21->Best_cicles[1]=P_ob[N_ob][1];
    T21->Best_distance=distance(&T11->Best_points,
&T21->Best_points)+T11->Best_distance;
}
}

```

```

//T22
if
(distance(&ppoints[0], &T12->Best_points) < distance(&ppoints[1], &T12->Best_points))
{
    T12->lchild=T22;
    T12->rchild=NULL;
    T12->parent=T;
    T22->Best_points[0]=ppoints[0].x;
    T22->Best_points[1]=ppoints[0].y;
    T22->Best_cicles[0]=P_ob[N_ob][0];
    T22->Best_cicles[1]=P_ob[N_ob][1];

    T22->Best_distance=distance(&T12->Best_points, &T22->Best_points)+T12->Best_distance;
}
else
{
    T12->lchild=T22;
    T12->rchild=NULL;
    T12->parent=T;
    T22->Best_points[0]=ppoints[1].x;
    T22->Best_points[1]=ppoints[1].y;
    T22->Best_cicles[0]=P_ob[N_ob][0];
    T22->Best_cicles[1]=P_ob[N_ob][1];

    T22->Best_distance=distance(&T12->Best_points, &T22->Best_points)+T12->Best_distance;
}

//T21
line=Line(T21->Best_points, End_p);
N_ob=Judge(T21->Best_points, End_p, line[0], line[1], line[2]);

if (N_ob==NO_OB)
{
    T21->parent=T11;
    EndBiTree(T21);
}
else
{
    T21->parent=T11;
    TwoBiTree(T21, N_ob);
}

//T22

```

```

    line=Line(T22->Best_points,End_p);
    N_ob=Judge(T22->Best_points,End_p,line[0],line[1],line
[2]);
    if (N_ob==NO_OB)
    {
        T22->parent=T12;
        EndBiTree(T22);
    }
    else
    {
        T22->parent=T12;
        TwoBiTree(T22,N_ob);
    }
}

BiTree* PreOrder_FT(BiTree *T,BiTree *Best_T)    // 前序遍历
二叉树的递归算法 找到距离最短节点
{
    if (T)
    {
        if
(T->lchild==NULL&&T->rchild==NULL&&T->Best_distance<Best_
T->Best_distance)
        {
            Best_T=T;
        }
        Best_T=PreOrder_FT(T->lchild,Best_T);
        Best_T=PreOrder_FT(T->rchild,Best_T);
    }
    return(Best_T);
}

void PreOrder_pri(BiTree *Best_T)
{
    if(Best_T)
    {
        printf("路径点\n");
        printf("%f ",Best_T->Best_points[0]);
        printf("%f\n",Best_T->Best_points[1]);
        printf("圆心点\n");
        printf("%f ",Best_T->Best_cicles[0]);
        printf("%f\n",Best_T->Best_cicles[1]);
        PreOrder_pri(Best_T->parent);
    }

}

void main()
{

```

```
BiTree * T;
BiTree * Best_T; //建立二叉树, 必须放在最前面, 否则出错
double Sum_dis[50]={0};
double *line;
int N_ob;
line=Line(Start_p,End_p);
N_ob=Judge(Start_p,End_p,line[0],line[1],line[2]);
T=(BiTree *)malloc(sizeof(BiTree));
Best_T=(BiTree *)malloc(sizeof(BiTree));
T->Best_points[0]=Start_p[0];
T->Best_points[1]=Start_p[1];
T->Best_cicles[0]=0;
T->Best_cicles[1]=0;
T->Best_distance=0;
T->parent=NULL;
Best_T->Best_distance=1000000000000000;
if (N_ob==NO_OB) //全程没有交点
{
    EndBiTree(T);
}
else
{
    TwoBiTree(T,N_ob); //生成二叉树路劲图
}
Best_T=PreOrder_FT(T, Best_T);
PreOrder_pri(Best_T);
}
```