

浙江大学

本科生毕业论文(设计)



题目 两轮机器人的运动规划与平衡控制算法设计与实现

姓名/学号 李川/3100102839

指导教师 熊蓉、吴永海

年级与专业 自动化 2010 级

所在院系 控制科学与工程学系

致 谢

我在此谨以此部分感谢那些在大学生活中直接或者间接给予我帮助的人。

首先，感谢我的指导老师熊蓉老师。在大学生活中熊蓉老师教会了我很多知识，同时给予了我很多学习和生活上的重要建议和引导。熊蓉老师对我毕设提供的帮助及指导也是至关重要的，尤其是过程中关键的阶段性安排和具体的细节思路让我能有条不紊的完成本次毕设。在本次毕设中，从查找资料，选题，研究到最终毕业设计定稿，熊蓉老师都付出了大量的心血和智慧。我同样要感谢我的企业导师吴永海博士。吴永海博士在机器人设计，两轮机器人平衡控制仿真等方面对我进行了关键性的指导，并帮助我顺利实现从仿真到实物调试的过渡。吴永海博士在其他机器人软硬件创新与维护方面也为我提供了大量的帮助。我也要感谢朱秋国老师。朱秋国老师在控制方案的选择，控制效果优劣评判方面给予我许多关键性的指导，同时朱秋国老师的许多建议与想法使我对许多问题都有了更深入的理解。我还要感谢李文华工程师，李文华工程师在两轮机器人的机械结构设计与具体改进方面给予了我很多关键性的帮助。

其次，感谢实验室的学长学姐们。毛翊超学长在我开始设计之初为我讲解了他在两轮机器人设计与实现过程中的经验，让我在本次设计中避免了许多弯路。也在后续设计过程中和我深入许多理论和创意，让我对整体系统有更深入的理解。李千山学长和赵越学长在多种方法的细节实现上为我提供了他们的研究经验，并从他们研究的角度分析我的控制方法的得失，为我的控制算法改进提供了帮助。毛曙源学长、戴萧何学长、王亚亮学长在我的仿真与机器人的调试过程中也为我提供了巨大的帮助。

再次，感谢我的同学们。方立同学为本次两轮机器人的电路设计做出了巨大的贡献，同时在很多实现细节上与我进行了深入探讨。吴伟男同学在机械结构计算方面与多种方法取舍方面给予我很大帮助。陶宇伦同学在实物调试方面给了我许多有效的参考建议。还有我的室友和其他小伙伴们，在本次设计最困难的时期给予我许多的鼓励。

最后我要感谢在我背后默默支持我的父母和家人，你们的支持是我最大的动力！

摘要

两轮机器人由于其两轮驱动具有结构简单、运动灵活的特点，适用于在狭小空间的执行任务，在家庭办公环境和室外野外环境均具有广泛的应用前景。而两轮机器人系统是一个本质不稳定系统，其运动规划和平衡控制是机器人研究领域的热点问题。其系统具有的复杂性及非线性特性，对于非线性系统控制算法研究和智能控制器的设计有很大的意义。

本文对两轮机器人系统的设计与实现、系统建模、平衡控制以及平衡约束下的运动规划进行了研究，完成了以下四方面工作：

1. 完成了两轮机器人的机械、电路构建，设计并实现了基于状态反馈控制的两轮机器人平衡控制。
2. 设计并实现了电脑端与手机端的上位机控制系统：电脑端基于 Labview 语言建立了数据采集监控系统，基于 c 语言建立了轨迹规划系统；手机端基于 java 语言实现了手机 app 运动控制。
3. 提出了一种基于“虚拟质心”假设，在平衡约束下，基于状态反馈内环实现了机器人运动控制，并通过状态维护与平滑切换，实现了平衡状态下的复杂运动控制
4. 针对两轮机器人设计了对于其质心变化的系统辨识与卡尔曼滤波算法，初步实现质心自适应。

关键词： 移动倒立摆，状态反馈，运动规划，虚拟质心，自适应

Abstract

Driven by two wheels, the structure of two-wheeled robot is simple, and it could move flexibly, so it has a advantage in the task execution in small space. It also has a wide application prospect in the indoor and outdoor environment. Two-wheeled robot system is a natural unstable system, the motion planning and control of which is a hot topic in robotics. Its high complexity and nonlinearity are important to the research on nonlinear system control and design of intelligent controller.

This paper studies on system establishment, model construction, balance control and balance constrained motion planning on the two-wheeled robot system. Totally, the paper has done four aspects of work:

1. The system constructs the mechanical and electrical system for the two-wheeled robot, and realizes the balance control of the robot based on state-feedback control.

2. The system constructs upper control system in computer and mobile phone. In computer, it establishes a record subsystem based on Labview language and a motion planning subsystem based on c language. In mobile phone, it establishes motion control subsystem based on java language.

3. The paper presents an assumption about “virtual centroid”, realizes the balance constrained motion control of the robot based on state-feedback control by State maintaining and switching smoothly.

4. The system identification and Kalman filter is designed for the robot’s centroid adaption.

Keywords: mobile inverted pendulum, state feedback, motion planning, virtual centroid, self-adaption

目 录

1 绪论.....	1
1.1 研究背景和意义	1
1.2 研究现状	2
1.2.1 两轮机器人来源	2
1.2.2 国外实物机器人研究现状	3
1.2.3 国内实物机器人研究现状	5
1.2.4 国内外理论方法研究概述	6
1.3 本文研究内容思路	7
1.4 本文结构	9
2 系统设计.....	11
2.1 系统设计的要求	11
2.2 机械结构设计	11
2.2.1 机器人整体设计	11
2.2.2 质心设计分析	13
2.2.3 IMU 位置设计分析	14
2.3 电气结构设计	15
2.3.1 整体电气结构	15
2.3.2 控制器	15
2.3.3 角度及角速度数据测量	16
2.3.4 电机控制与测速	16
2.3.5 无线通讯	17
2.3.6 电路设计	17
2.4 底层程序设计与实现	18
2.4.1 底层软件程序综述	18
2.4.2 硬核程序模块	18
2.4.3 软核程序模块	20
2.5 上层程序设计与实现	21
2.5.1 监控程序结构	22
2.5.2 路径规划程序结构	26
2.5.3 手机控制程序结构	26
2.6 系统测试	27
2.6.1 机器人运行测试	27
2.6.2 上层监控程序测试	28
2.6.3 上层路径规划程序测试	30
2.6.4 手机控制程序测试	31
2.7 本章小结	32
3 平衡运动建模与控制方法.....	33
3.1 机器人平衡建模	33
3.1.1 Lagrange 方程建模	33
3.1.2 非线性模型表述	36
3.1.3 线性模型表述	36

3.1.4 LQR 反馈矩阵计算与仿真	37
3.2 电机建模	38
3.2.1 电机模型表述	38
3.2.2 实际电机数据分析	39
3.3 机器人控制方法设计	40
3.3.1 机器人控制方法综述	40
3.3.2 基础平衡控制算法设计	40
3.3.3 直行运动控制算法设计	41
3.3.4 旋转运动控制算法设计	42
3.4 机器人控制效果	43
3.4.1 原地平衡控制效果	43
3.4.2 “虚拟质心”直行运动控制效果	45
3.4.3 原地旋转运动控制	51
3.4.3 带半径旋转运动控制	53
3.5 本章小结	55
4 路径规划及复合运动控制	57
4.1 路径规划	57
4.1.1 路径规划的意义	57
4.1.2 A 星算法基本思想和方法阐述	57
4.2 基于航点的复杂运动控制	59
4.2.1 基于航点的复杂运动控制方法	59
4.2.2 基于航点的运动控制效果	60
4.3 基于光滑连续轨迹的复杂运动控制	61
4.3.1 光滑连续轨迹复杂运动控制方法	61
4.3.2 手绘轨迹跟踪控制	62
4.3.3 手绘轨迹跟踪效果	62
4.4 本章小结	64
5 机器人质心自适应功能设计	65
5.1 机器人系统参数辨识	65
5.1.1 参数辨识方法设计	65
5.1.2 参数辨识实验结果	66
5.2 机器人自适应算法设计	68
5.2.1 自适应算法设计综述	68
5.2.2 卡尔曼滤波器设计	69
5.2.3 卡尔曼滤波效果	70
5.2.4 辅助自适应算法设计	74
5.2.5 自适应算法最终设计思路	76
5.3 本章小结	76
6 结论	77
6.1 总结	77
6.2 展望	77
参考文献	79

1 绪论

1.1 研究背景和意义

移动机器人是一类集环境感知、动态决策与规划、行为控制等多功能于一体的机器人综合系统。由于其智能性及安全性，移动机器人越来越多的应用于家用服务、民用物流、军用侦察、环境勘探等领域。两轮机器人由于其两轮驱动具有结构简单、运动灵活的特点，适用于在狭小空间的执行任务，在家庭办公环境和室外野外环境均具有广泛的应用前景。

移动机器人的研究涉及多个方面，其中比较重要的就是其运动控制。通常移动机器人选择三轮或四轮驱动，保证自身稳定。而两轮机器人作为特殊的移动机器人，在运动控制的同时，必须保证其平衡。所以对于两轮机器人这种本质不稳定系统而言，其运动规划和平衡控制是机器人研究领域的热点问题。

两轮机器人运动控制是通过两个轮子电机的控制来控制整个机器人的倾角、前行速度以及转向速度，是一个典型的低维控制高维的系统，并且几个被控量间还存在耦合。在研究时，通常将两轮自平衡机器人抽象为倒立摆系统。倒立摆系统在平衡控制领域是一类典型的非线性系统，此类系统采用简单的控制策略无法得到理想的控制效果，其复杂性、非线性让众多研究机构投身其中，希望通过更复杂的算法使得两轮平衡机器人系统在不同的状态及环境下有更强的稳定性和适应性。状态反馈控制、自适应控制、模糊控制等多种控制算法先后不断应用到类似系统的控制中，并取得了一定的控制效果。

但现阶段国内外对于两轮机器人研究的重点更多倾向于先进控制算法在两轮机器人设计中的实现，特别是 Segway 式的载人机器人方面的研究。较少涉及两轮机器人在平衡约束下在不同运动状态下进行运动规划的问题。由于两轮机器人前进后退与平衡相耦合，即机体倾角与前进速度相耦合，无法实现独立的平衡控制与速度控制。而 Segway 一类的两轮机器人由于存在“人”这个高级模糊控制器，因而其速度控制仅需响应人的操作，实现平衡功能，其倾斜所造成的额外的合外力自然能使得机器人做加速、减速运动，无需考虑给定速度追踪的问题。故在平衡约束下的运动控制与规划问题仍值得研究。

同时，本次毕业设计还涉及两轮机器人的设计与实现、以及对现有文献各类控

制方法的合理选择和技术实现。由于能查阅到的几乎所有的文献仅阐述其控制原理及实验结果，缺乏实现过程以及具体的技术细节，有些甚至仅在仿真中实现过。因此有必要开展这方面的工作。

1.2 研究现状

1.2.1 两轮机器人来源

两轮机器人的想法来源于倒立摆模型^[1]。一般常见的线性倒立摆使用的是四轮小车，在其上部加装倒立摆，通过小车的运动保持倒立摆的平衡。其结构如图 1.1 所示。

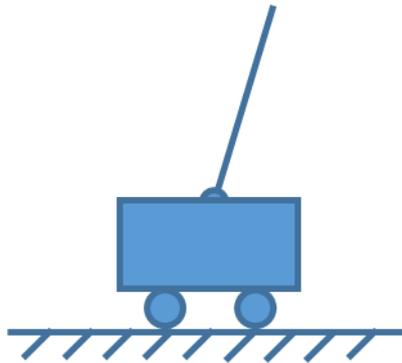


图 1.1 倒立摆小车模型

对于上述倒立摆小车系统，摆通过活动件与小车连接，只能在平面内运动，整个系统为二维系统。为了保持摆的平衡，当摆向左侧倾斜时，小车需迅速向左运动，来保持摆的平衡。同理，摆向右侧倾斜时，小车需向右运动来保持平衡。

将上述倒立摆小车系统的小车去除，改为倒立摆与两个带动力的轮子直接连接，就成为了两轮自平衡机器人，抽象而言，即移动倒立摆系统，如图 1.2 所示。

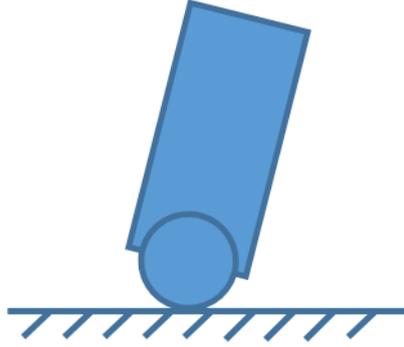


图 1.2 两轮机器人模型

两轮机器人和倒立摆小车类似，也通过相应的向前与向后的运动来控制机器人自身的平衡。所不同的是，两轮机器人轮子与机身通过电机连接，即电机产生的力矩直接作用于摆体。另外两轮机器人可通过两轮差速控制，实现原地旋转等倒立摆小车无法实现的三维运动。这一特性也就决定了除平衡控制外，两轮机器人可进行相应的二维平面内的运动规划。

1.2.2 国外实物机器人研究现状

对于两轮自平衡机器人的研究，美国、日本、瑞士等国起步较早，其研究也达到了一个比较高的阶段，而近几年来，随着 Segway 的成功，两轮机器人的研究在各国都得到了快速发展。

2002 年，瑞士联邦工业大学工业电子实验室的 Felix Grasser 等人研究出利用 DSP 控制的基于倒立摆控制原理的两轮移动机器人 JOE^[2]，如图 1.3 所示。研究人员对移动倒立摆模型进行建模分析，将整个控制系统解耦为平衡前后运动子系统与转动子系统，通过陀螺仪和光电编码器测量的数据，用线性状态反馈控制器来控制整个系统的平衡，并通过不同的极点配置达到不同的控制性能。

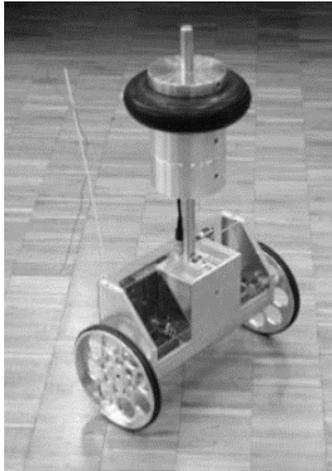


图 1.3 瑞士联邦工业大学的 JOE 两轮移动机器人

美国 Segway 公司的 Dean Kamen 发明了世界上第一步可以载人的两轮机器人^[3]，如图 1.4 所示。驾驶者在 Segway 小车上，以车辆的总体纵轴作为参考线。当人前倾时，这条轴前倾，Segway 内置的会产生向前的运动，一方面平衡前倾时重力所产生的扭矩，另一方面，提供动力使得整车加速。相反，当驾驶者的重心后倾时，也会产生相应的动作以达到相应的控制效果^[4]。Segway 的传感器采用了五个陀螺仪，两个倾角传感器和一个电机编码器。人在控制 Segway 前进或后退时，正是这些传感器来获取相应的位姿信息，提供给处理器做出对应的控制。



图 1.4 Segway

美国南曼瑟迪斯特大学地球物理学图像研究所的 David P. Anderson 设计了一个名为 Nbot 的两轮平衡机器人^[5]。如图 1.5 所示。其姿态传感器是由 FAS-G 的惯性测量单元构成。通过 FAS-G 的惯性测量单元测量机器人的倾斜角度，传递给控

制单元来处理并对其自身做出控制。Nbot 可以实现零半径回转，同时能实现不平的路面的情况下前进，也可在一定坡度上前进。在实验原型机中，Nbot 是比较具有代表性的。



图 1.5 Nbot

1.2.3 国内实物机器人研究现状

相对于国外而言，国内起步较晚，但近几年的快速发展下，也出现了一些实验用的原型机。

2005 年哈尔滨工程大学的尹亮制作了一个双轮直立自平衡机器人 Sway^[6]，如图 1.6 所示。该系统采用的核心处理单元为 C8051 单片机。利用 MEMS 加速度计及反射式红外传感器测量车体倾斜角度，反馈给控制单元，运算得到相应的 PWM 控制两台直流电机的转速达到控制机器人平衡的目的。

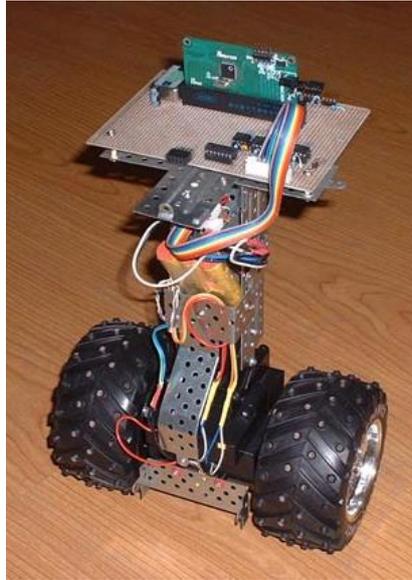


图 1.6 Sway

2008 年, 哈尔滨工程大学研制了一款变质心两轮自平衡代步电动车^[7], 如图 1.7 所示。该系统采用了 16 位工业级单片机作为主控芯片, 通过硅微陀螺和加速度计信号融合成的倾角信息, 经主控芯片计算, 控制直流无刷电机的转动, 实现系统的动态平衡。当人站立于此款电动车上时, 可以通过控制自身的重心来控制电动车的前进或后退, 通过摆动转向杆, 控制左右转向。



图 1.7 哈尔滨工程大学两轮自平衡代步车

1.2.4 国内外理论方法研究概述

就这几年国内外对两轮机器人的发展, 为了两轮机器人平衡控制, 其使用的控制策略越来越复杂。瑞士的实验室在研究 JOE 机器人时充分分析了机器人的物理

特性,根据其运动学方程进行建模,并将模型解耦为两个子系统数学,通过不同的简单极点配置实现不同的控制性能。而此后的两轮机器人建模中更多的采用了拉格朗日方程,从对象整体入手分析广义力与广义速度,结合机器人自身运动约束,推导出数学模型。在中科大发表的系列论文中就多采用了该方法^[8]。但这种建模方法往往基于自身摆角的偏移量为小角度的前提下。

但当出现大角度时,两轮机器人的非线性特性会表现的非常明显,通过单一参数的极点配置得不到理想的控制效果。所以陆续有学者提出模糊控制的方法^[9]。在模糊控制中,模糊化不同的状态变量,并通过规则进行模糊推理,获得各个等级模糊化输出的隶属度。根据隶属度以及经验设定的各个等级的输出实际数值大小,获得最终的实际输出数值。

为了解决模糊控制规则指数增长的问题,中科大实验室在两轮机器人系统中应用了分层模糊控制^[10],将自平衡车控制分为平衡控制以及速度控制两部分,通过平衡角度的误差与电机速度误差的比较,对两个控制子系统的重要性进行评估,赋予两个子系统模糊控制的输出以不同的权重,以达到平衡控制的效果。使得模糊控制器的规则数目随着状态变量的数目线性增加,降低模糊规则的复杂度。

而为了使得机器人的参数能适应机器人自身物理参数变化,在两轮机器人的设计中加入了基于利亚普若夫稳定性的自适应模糊控制方法^[11]。通过设计自适应控制率,可以使控制系统的各等级对应的输出随着系统状态的改变而改变,从而实现参数自适应。

针对其非线性,有学者开始在两轮机器人的设计中加入了滑模控制方法^[13],设计了基于模糊切换增益自调节的滑模控制器并应用以用于两轮自平衡机器人的控制,该控制器利用模糊逻辑对滑模控制的切换增益进行实时调整,既能充分发挥滑模控制对参数变化和干扰不灵敏的特点,又具有模糊自调节的能力,从而达到削弱抖振的目的。

1.3 本文研究内容思路

在本次毕业设计中,研究目标是在两轮机器人硬件系统的基础上,实现两轮机器人的平衡控制,优化控制性能,并完成平衡约束下的运动控制。本文主要完成了两轮机器人的硬件搭建、平衡控制和平衡约束下的运动控制以及机器人质心自适应方面的研究。主要工作包括:

(1)完成机器人硬件系统搭建。完善电路系统,选用 Xsens Mti 系列 IMU(Inertial measurement unit, 惯性测量单元)作为测量单元,并对应建立独立供电模块为之供电,建立串口通信模块,实现主控芯片与测量单元的数据交互。在机器人机械及电路系统的基础上,设计底层控制芯片编程结构,利用 verilog 语言以及 QuartusII 编程环境自定义硬件资源,自由选择 IO 口数量、定时器设置、中断顺序等,构建基础硬核;在硬核基础上采用 NiosII 嵌入式编程,利用 c 语言构建软核,运行控制程序。

(2)完成上层控制系统程序。上层控制程序主要利用 Labview 配合蓝牙通讯,实现实时数据上传、测量与记录,在数据的支持下优化控制策略。利用 c 语言实现路径规划及实时指令下发,控制运动。手机端利用 java 编写对应 app 实现手机端遥控控制。

(3)完成两轮机器人平衡运动建模及在平衡约束下的运动控制。在平衡控制方面,研究了系统建模和平衡控制方法。采用 Lagrange 方程建模,推导并整理出两轮机器人非线性模型。加上小角度线性化处理得到其线性模型。在线性模型的基础上,利用状态反馈设计控制器并编程实现平衡控制。在平衡约束下的运动控制方面,提出了一种“虚拟质心”的运动控制方法,利用“虚拟质心”的向前向后偏移,配合状态反馈约束实际质心运动,能很好的实现两轮机器人在平衡约束下直行运动控制。并在此基础上为旋转设计控制算法,实现原地旋转与带半径旋转功能。

(4)完成复杂轨迹规划及复杂运动控制。借助 A*算法实现两点间的路径规划,获得复杂运动路径。设计阶段式复杂路径分解办法,在完成简单运动的基础上,考虑复杂运动与简单运动的联系,设计控制方法完成简单运动间的平滑切换,将复杂运动划分为若干简单运动,利用简单运动组合实现复杂运动过程。并考虑利用直行过程中的角度跟踪优化复杂路径运动,设计连续式复杂路径分解办法,在保持机器人直行的情况下追踪理想朝向角实现复杂光滑路径的跟踪。

(5)完成机器人质心自适应功能,优化机器人控制效果。为了适应自身结构改变等引起的质心变化,需要获得系统较为准确的物理参数数据,建立质心自适应系统。为此引入了基于最小二乘的系统辨识,针对两轮机器人系统进行推导并导出系统转动惯量的最小二乘估计表达。利用该估计,设计卡尔曼滤波器对其实际质心位置进行滤波估计,研究其估计效果。同时针对卡尔曼滤波器的不足提出了辅助质心适应算法的设计,通过对路程的大反馈系数衰减控制和小反馈系数不衰减控制的

结合，实现一定范围内的质心自适应，辅助卡尔曼滤波器完成质心自适应功能。

1.4 本文结构

本文第一章为绪论，整体介绍了本次设计的研究背景、研究思路以及国内外研究现状。第二章为系统设计，对两轮机器人系统的结构及功能进行了阐述。第三章为平衡运动建模与控制方法，对系统建模并设计了对应的平衡控制算法与简单运动控制算法。第四章为路径规划及复杂运动控制，阐述了路径规划的方法和从简单运动到复杂运动的设计。第五章为自适应算法设计，阐述了最小二乘辨识的思路、自适应卡尔曼滤波器的设计及提出了辅助自适应算法的设计。

2 系统设计

2.1 系统设计的要求

为了保证两轮机器人的灵活性,适用于在狭小空间的执行,提高在任务环境复杂、变化剧烈环境下的执行能力,结构上,设计要求其结构简单,车体尺寸在高 450mm×宽 250mm×长 200 mm 内,并要求整体重量不超过 5kg。为了保证以后可能在两轮机器人上加装的传感器系统和其他操作设备的信息采集和控制的需要,要求保证其平衡和运动过程中的稳定性,在直立平衡控制过程中,其调整摆动角不能超过 $\pm 3^\circ$;功能上,设计要求其实现原地自平衡功能,速度受控的直线运动功能,以及定点转动和沿圆弧旋转功能。其中,要求直线运动误差小于运动距离的 10%,定点旋转角度误差小于旋转角度的 10%。同时具有一定的不平整路面运动抗干扰能力。

2.2 机械结构设计

2.2.1 机器人整体设计

所设计机器人机械结构如图 2.1 所示。机器人主要分为机器人主体支架、电路板、车轮、驱动电机、IMU 等。

机器人整体关于其中轴面对称。轮子设计采用六轮辐镂空合金钢轮壳,减小了轮子重量同时保证了轮子整体的刚度。胎面采用包胶工艺,保证了合适的软硬程度及对地摩擦力。两轮胎对称的设计在主体支架两侧,通过 10:1 的减速箱与无刷电机相连。电机由电机支架固定并固定在主体支架上。支架主体前方设有平面供电电路板的安装。支架内部设有电池仓以供电池安放。电池仓上部平面供独立供电电路板安装。主体支架上部预留数字舵机安装位,为后续开发提供空间。

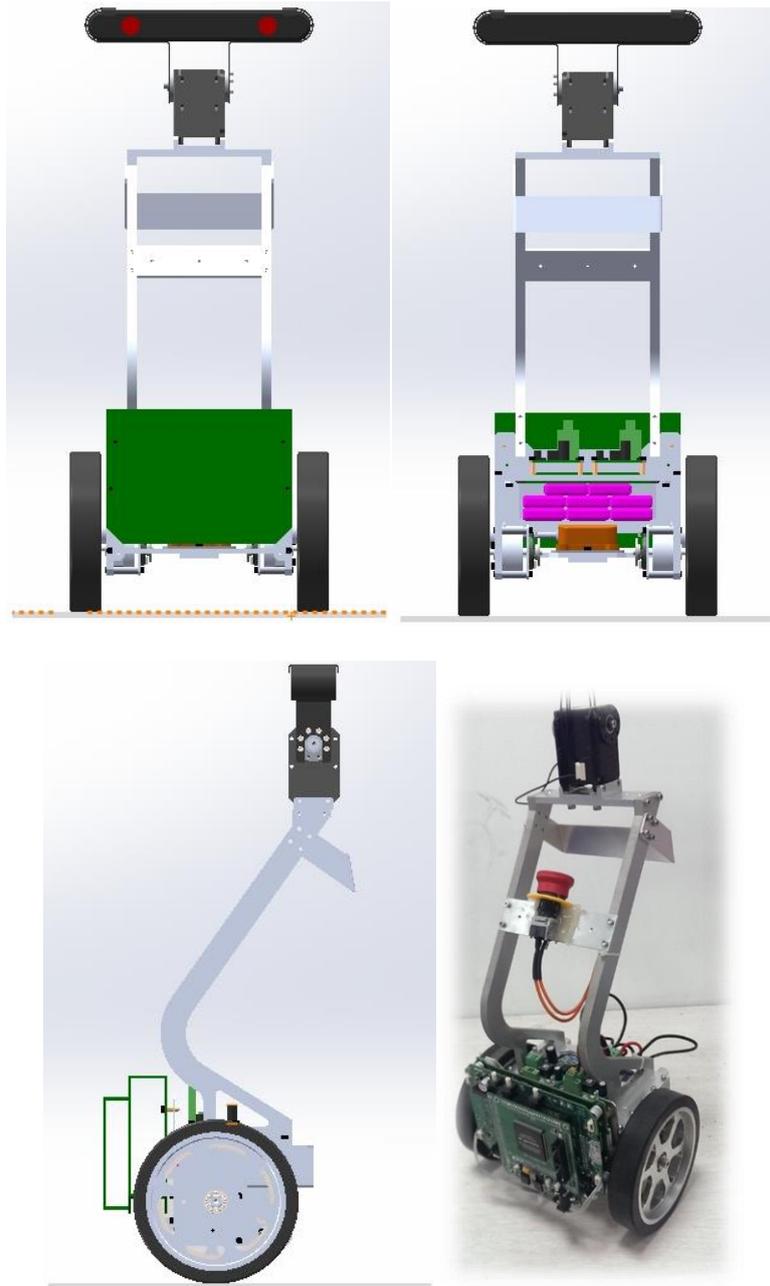


图 2.1 两轮机器人模型主视图（左上）后视图（右上）侧视图（左下）及实物图（右下）

驱动电机配合电路设计，选择无刷直流有感电机；为保证设计尺寸符合要求，选择外形尺寸较小的盘式电机；为保证其驱动能力，选择 50W 电机。综上，最终选择 Maxon EC 45 flat brushless 50W motor，查阅电机手册，其连续驱动负载 96.9mNm，最大堵转负载 1100mNm，能为平衡控制输出较大的力矩；额定工作电压 18V，与机器人设计的电池供电电压 16.8V 相近，能较好的发挥电机性能。



图 2.2 Maxon EC 45 flat brushless 50W motor

IMU 作为两轮机器人的核心测量单元，与机器人刚性连接，可实时测量机器人的三轴角度和三轴角速度，即位姿描述中常用的 roll、pitch、yaw 三个角度及其角速度。这些信息是平衡控制的重要依据。同时 IMU 的响应速度、角度及角速度测量精度、角度漂移程度这些性能与获得的位姿信息的准确度息息相关，对平衡控制也有重大影响。故在本次设计中，采用在大仿人所使用的 Xsens MTi IMU。作为一款成熟的 IMU 产品，其本身可高速测量三轴的加速度、角速度以及磁场强度，最大测量频率可达 512Hz，并通过内部设计好的信号融合算法融合滤波后输出，其 roll 经长时间测量基本无零漂，角度数值测量在 $^{\circ}$ 单位下小数点后三位，动态误差小于 0.1° ，大大保证了机器人位姿信息的准确程度。



图 2.3 Xsens Mti IMU

2.2.2 质心设计分析

两轮机器人总体可抽象为一个移动倒立摆，其具体建模详见后续章节。在此仅以一个粗糙的倒立摆模型做说明，假设其下端固定，所受到的重力将引起其角加速度变化，即：

$$J\ddot{\theta} = mgl \sin \theta \quad (2.1)$$

式中 m 为倒立摆质量，假设倒立摆质量集中于质心， l 为转动中心到倒立摆质量中心的距离， J 为倒立摆的转动惯量， θ 为摆偏离平衡位置的摆角。同时有：

$$J = ml^2 \quad (2.2)$$

将式(2.2)带入式(2.1)可得：

$$\ddot{\theta} = g \sin \theta / l \quad (2.3)$$

由上式可以看出，当质心离轮轴过近时， l 很小，在同样的角度下会引起较大的角加速度，即会导致整体系统对控制作用敏感，对控制精度要求增大；反之，质心离轮轴过远会导致整体系统对控制作用响应滞后，增加控制难度。因此合理的质心设计对于控制效果影响较大。经过测算，这种设计使得其总体质心偏离轮轴约 0.06m。后续实验证明，这种质心设计和质量分布较为合理。

2.2.3 IMU 位置设计分析

在机器人各部件中，IMU 的位置设计直接关系到后续测量与控制的效果。因此对陀螺仪位置的设计做分析。

机器人所采用的陀螺仪为 Xsens 公司生产的 Mti 系列 IMU。该系列 IMU 含 3 轴加速度计、3 轴陀螺仪和 3 轴磁力计。可以同时测量 xyz 三轴的角速度、加速度和磁场强度。其内部处理芯片可由采集到的信息，通过内部自带的信息处理算法，给出整体的 IMU 姿态信息，包括角速度、加速度和三轴的角度。在整个系统中，IMU 和机器人刚性连接，IMU 的姿态信息直接反应了机器人的姿态信息。

本次设计中，IMU 安装在两轮轴线上，IMU 的三轴与机器人定义的三轴方向重合。其角速度的测量是根据内部振子来回振动过程中，测量在科里奥里力的作用下产生的偏移来获得的，因而跟 IMU 安装的角度有关，本设计中的安装角度能够正确测量出机器人的各轴角速度。而在加速度测量的是外界对 IMU 的合加速度。本次设计中的安装位置在轴线上，最大程度的避免了因旋转等引起的其他加速度对重力加速度的干扰。又因为 IMU 与机器人的三轴重合，也保证了其磁力计测量的信息的准确性。

2.3 电气结构设计

2.3.1 整体电气结构

整体电气结构框架如图 2.4 所示。采用 Altera 的 CycloneIII 作为主控芯片，利用专用软件对其编程，采集 IMU 反馈的角度与角速度信号、码盘反馈的电机速度信号，运算产生控制信号，配合电路驱动 2 路 50W 电机，控制机器人运动。同时利用蓝牙模块与上位机通讯，进行信息交互。

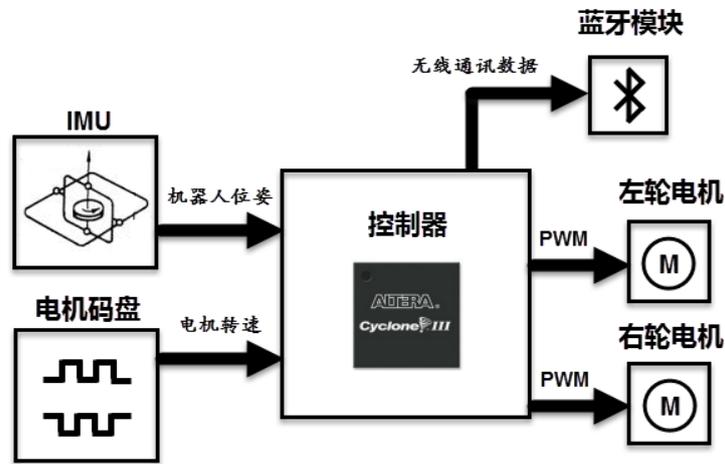


图 2.4 电气结构框架图

2.3.2 控制器

控制芯片采用 Altera 的 CycloneIII 系列 FPGA，通过专用的软件编程软件可配置 FPGA 芯片内的逻辑关系，定制机器人所需的硬件处理单元片上与片外资源。通过芯片管脚与信号相连，直接或者间接采集各个反馈信号，运算并输出控制信号控制电机运动，无线通讯等功能。为实现更好的效果，看可以对不同运动过程编写不同的控制策略，并加以条件跳转，优化控制效果。

2.3.3 角度及角速度数据测量

机器人的角度与角速度数据通过 IMU 进行测量。IMU 内置有陀螺仪。机械设计上，IMU 与车轮共轴，因而陀螺仪测量的角速度基本上可以表征车体角速度。陀螺仪的角速度测量利用的是科里奥里力原理，通过陀螺仪内置的敏感元件可以测得当前小车的倾斜角速度。在陀螺仪内部积分即得当前的角度。

IMU 内部另有磁传感器，可通过陀螺仪所处位置三个方向上地磁场强弱来确定小车相对于世界坐标系的 roll、pitch 和 yaw 角。陀螺仪的角速度的测量是相对准确的，但长期积分会存在积分误差以及固定的零漂误差，磁力计长期测量是不会有误差的，但每次测量由于磁场变化都存在误差。即磁力计的低频分量较为准确，陀螺仪的高频分量较为准确，则可以通过一组高通低通滤波器，将两者信号融合得角度值，比起单一使用一种传感器信号所获得的数据更可信。

2.3.4 电机控制与测速

两轮机器人采用的是 Maxon 的 50W 盘式直流无刷电机，在电路设计中采用 H 桥控制，在程序中模仿 33035 控制芯片原理，根据电机反馈的霍尔信号，确定当前电机转子所在位置，通过所模仿的 33035 控制过程，下发相应的控制信号和 PWM(Pulse Width Modulation, 脉宽调制)波，控制电路中各相通路 MOS 管的通断，使直流电机中的绕组在永磁体的作用下产生定向转动，从而驱动电机转动，使机器人产生所需要的控制动作。

电机速度运用码盘进行测量，配合 512 线光栅，通过测量码盘在一个测量周期内产生的脉冲信号即可确定码盘转速。速度测量采用 MT 法，即当转速很高时，两个脉冲信号时间间隔测量误差大，此时通过测量一个周期内的脉冲信号的个数，进行相应换算得电机转速；当转速偏低时，脉冲信号个数测量误差大，此时通过测量相邻脉冲信号间隔，进行相应换算得电机转速。同时，由于码盘拥有 A、B 两路相位差为 90 度的脉冲信号，因此可以通过确定 A、B 相位的顺序先后确定车轮转动的方向。

2.3.5 无线通讯

无线通讯利用蓝牙通讯原理。建立蓝牙通讯后，在上位机设备和机器人间建立了一条通过蓝牙连接的串口通道，遵循 UART(Universal Asynchronous Receiver and Transmitter, 通用异步通讯)协议，可以与带蓝牙的手机或电脑进行通讯。确定对应的通讯包格式，两轮机器人按照格式解析通讯包，即可实现无线通讯。手机与两轮机器人通过蓝牙连接，进行相应的通讯，通过手机端串口应用发包，可完成手机对两轮机器人的遥控控制；电脑与两轮机器人通过蓝牙连接，进行相应的通讯，配合相应的 Labview 程序，可完成小车运行情况监控，保存实时运行数据等。

2.3.6 电路设计

电路总体主要功能是完成用电模块供电、IMU 信号采集、电机信号采集与驱动、蓝牙无线通讯。其总体框架见图 2.5。

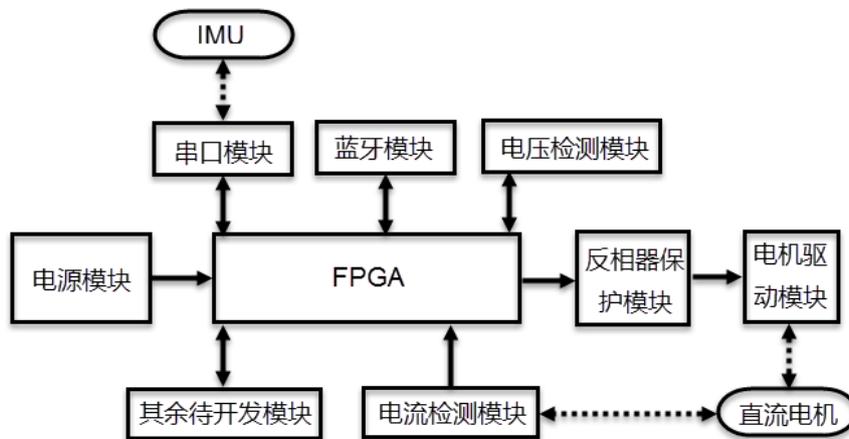


图 2.5 电路结构框架图

上述电路功能分布在两块电路板上。第一块为母版，包含了电源模块、串口模块、电压检测模块及反相器保护模块。第二块为驱动板，包含了电机驱动模块、电流检测模块、蓝牙模块，并预留了一些待开发模块的接口。

2.4 底层程序设计与实现

2.4.1 底层软件程序综述

底层软件程序由两部分组成，分别是由 verilog 硬件描述语言及 soc 模块构建的底层硬核，以及借助 NiosII 环境，由 c 语言构建的软核。

底层软核控制周期为 10ms，每个控制周期中，FPGA 会通过 IO 口获取电机码盘信心，通过硬核码盘测速模块计算电机速度，并反馈给软核进行计算。软核程序还会通过 UART 串口接收来自 IMU 的位姿信息，主要是角度与角速度的位姿信息。通过对速度及位姿信息的计算，将得到该控制周期内，两轮应输出的力矩的大小，并将之转换为相应输出的 PWM 波的大小，传输给硬核电机驱动模块，控制板上电机驱动电路，驱动电机运动。其中硬核中的码盘测速、电机驱动等模块的控制周期均小于 2ms，软硬核配合能有效的实现平衡控制。同时在机器人运行过程中，每 50ms 将自身信息通过蓝牙上传给上位机处理，上位机根据需要通过蓝牙下发控制指令。硬核同时驱动 A/D 芯片采集电池电压信息供软核处理使用。

2.4.2 硬核程序模块

为了提高控制精度，减小规律性复杂运算对于软核性能的消费，做到短周期实时响应，本次设计中将电机驱动模块、码盘测速模块及 A/D 转换模块用 verilog 语言在硬核中编程实现。在这样也利用了 verilog 语言并行运算的特点，使多个规律性复杂运算同时进行，做到实时响应，提高数据采集及运动控制的精度。

2.4.2.1 电机驱动模块

硬核的电机驱动模块模拟了 33035 电机驱动芯片的工作原理，采集电机反馈的霍尔信号，确定当前电机转子位置，来确定下一控制周期的相电流方向，产生能够驱动电机转子转动的磁场来驱动电机转动。同时将输入的 PWM 波耦合到控制信号上，改变施加在电机上的等效电压的大小，控制电机旋转速度。其模块结构如图 2.6 所示。图中码盘控制信号共有 6 路，分别控制三相 H 桥上下桥臂 MOS 管的通断，使得电机内电流定向流动，产生所需的控制效果。

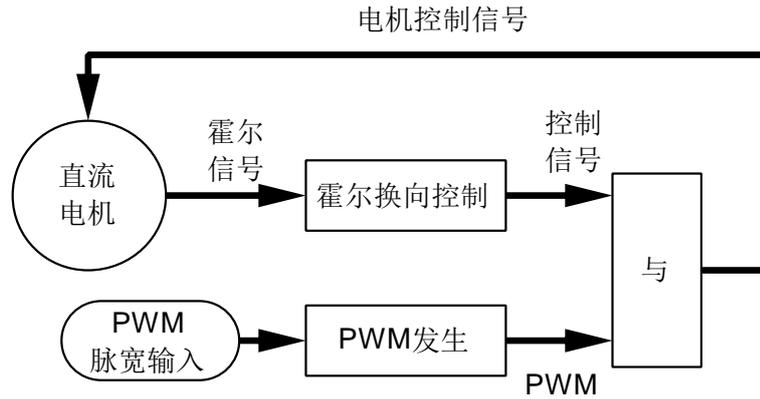


图 2.6 电机驱动模块结构

2.4.2.2 码盘测速模块

码盘测速模块是根据码盘反馈给 FPGA 的脉冲的个数或频率进行测速的。本次设计中选用的是 512 线的码盘，电机旋转 1 圈将产生 512 个脉冲，每个脉冲表示电机已转过 $2\pi/512$ 弧度。码盘信号分 A、B 两相，可根据 A、B 两相的相位情况判断此时的电机是正转或反转。

码盘测速有两种方式，一种为测定其固定时间周期内产生的脉冲个数，通过运算确定电机转动速度，另一种为测定两脉冲的间隔时间，通过运算确定电机的转动速度。在电机高速旋转的情况下，两脉冲的间隔时间短，测量的相对误差较大，而固定时间周期内的脉冲个数较多，测量误差小。高速情况下应选用后者。反之，低速情况下固定时间周期内脉冲数少，误差大，两脉冲间隔时间长，误差小，低速应测算两脉冲间隔。因此需依据电机转动情况选用不同的测速方式。同时由于硬核运算周期与软核运算周期不同步，需要将数据缓存供软核读取。

故本设计中码盘测速模块由一个计数模块、一个定时器模块、一个比较模块和一个数据缓存模块构成，定时器模块每隔 2ms 向计数模块发送一个信号，将当前计数数据缓存并清空计数。软核可根据需要随时在缓存模块中读取当前转速信息。其结构如下图所示。

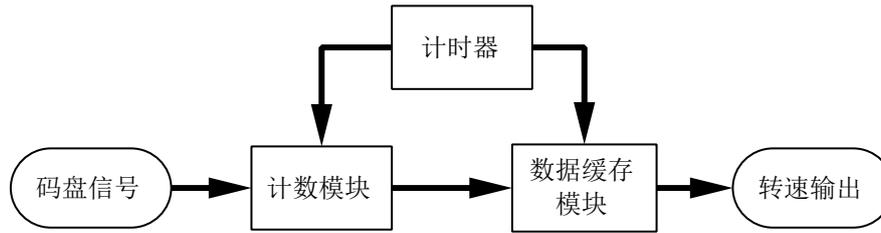


图 2.7 码盘测速模块结构

2.4.2.3 A/D 转换模块

电池电压信号是模拟量信号，但 FPGA 只能读取数字量信号，故采用 A/D 转换将电压值转换成数字量。本设计中采用的 A/D 芯片是一块多通道的 A/D 转换芯片，其内部可完成模拟量到数字量的转换，但读取转换后的数字量需要一定的通讯协议。

A/D 芯片输入配置在多路器寻址时序中进行，多路器地址选择模拟输入通道，也决定输入是单端输入还是差分输入。故在获取 A/D 转换结果前，需对配置输入模式，并选择输入通道。随后 A/D 芯片对输入的模拟量信号进行转换，转换完成会有相应标志位置位，此时可获取对应转换结果，并将结果缓存，供软核读取。完成转换后再一次进入前四个环节的循环，不断获取电压值。其结构如图 2.8 所示。

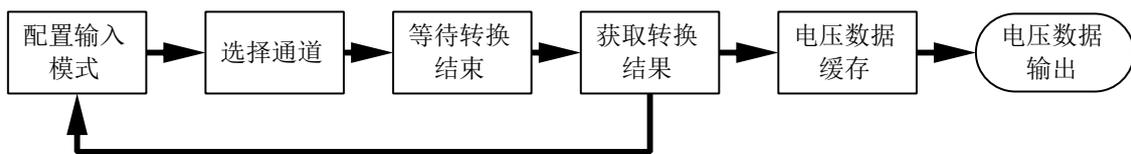


图 2.8 A/D 转换模块结构

2.4.3 软核程序模块

软核程序通过 c 语言完成，在主循环中获取硬核及 IMU 所采集的电机转速与机器人位姿信息，经过设计好的平衡控制器运算得控制信号，并传输给硬核做出相应的电机驱动控制。其具体程序框图如图 2.9 所示。

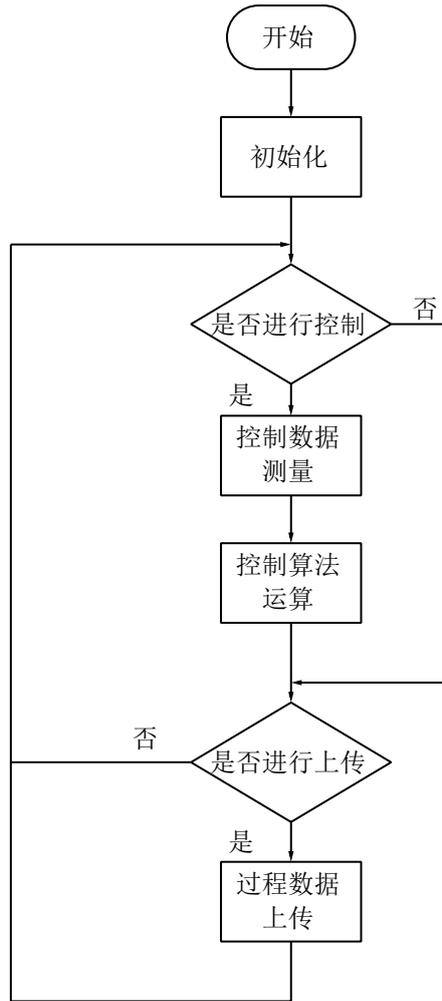


图 2.9 软核程序结构

初始化程序主要对程序中所涉及的一些结构进行初始化，对蜂鸣器和 LED 等进行关闭、闪烁等初始化操作，对电机模块所需的一些参数进行初始化设置。

主循环过程中是否进行控制或是否进行上传的标志是在定时器中进行改变的，即达到预设的控制周期时，将其置位，使循环中执行控制或上传模块，在控制或上传模块执行完成后，会将标志位清零，防止重复进入，直到下一控制周期到时计时器才会重新置位标志位，如此重复循环进行。

2.5 上层程序设计与实现

上层程序主要有三部分，分别是电脑端的由 Labview 编写的监控程序，电脑端由 c 语言编写的路径规划程序和手机端由 java 编写的手机控制程序。

2.5.1 监控程序结构

监控程序由 Labview 编写, Labview 语言在编程过程中会同时产生后台的图形化程序和前台的控制面板。故两者都需同时进行设计。另外为了减少蓝牙通讯收发之间的干扰,本次设计中将数据收发分别在两个程序中完成。因程序编程需要,在 Labview 监控程序项目中设计了多个 vi 和 Subvi, 如下图所示。

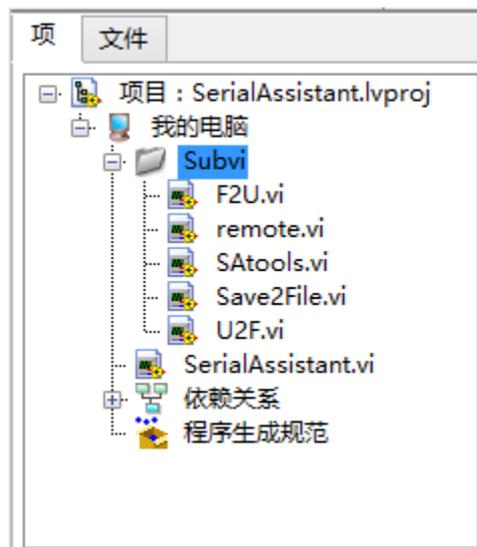


图 2.10 Labview 项目构成

2.5.1.1 主监控程序结构

主监控程序为 SerialAssistant.vi, 主要功能为接受机器人通过蓝牙发送上来的数据包, 判断数据包是否有错误并解包, 计算出数据包中所包含数据, 显示在前面板中, 并保存到指定的文件中。其流程如图 2.11 所示。

程序中串口读取部分程序选用顺序结构, 根据包格式顺序解包, 保证程序的稳定性。经串口读取程序解包完成的数据将带有包类型、包长度、包内容等信息, 这些信息在程序中打包成簇, 整体进入数据队列供数据队列解析程序读取和分析。串口读取程序和数据队列分析程序相互独立, 并行运算, 防止数据处理运算和显示时对系统计算资源的长时间占用导致的通讯问题, 保证了数据通讯的实时性。

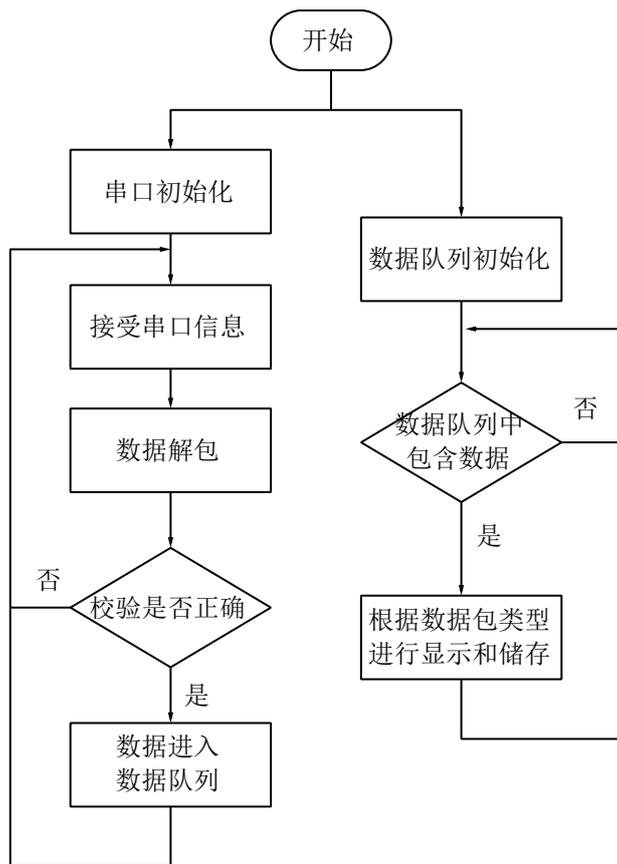


图 2.11 SerialAssistant.vi 程序流程图

对应后台程序，前面板结构如图 2.12 所示。前面板左侧是串口初始化相关参数，主要在启动程序前设置串口号及波特率。前面板右侧是一个 4 选项卡的控件，每个选项卡对应一种监控类型。在本次设计中针对数据需求，设计了 4 种监控模式，图中为其中之一的需求保存数据的面板。

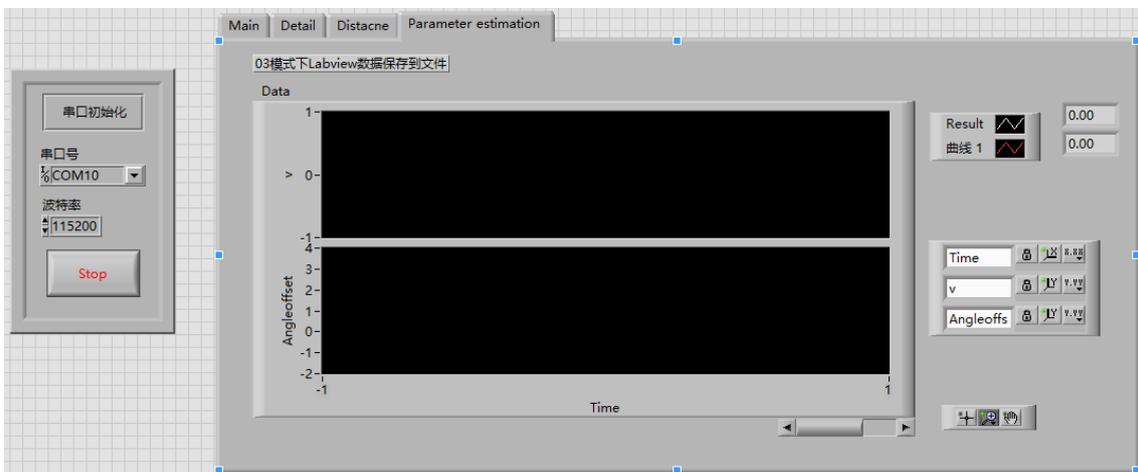


图 2.12 SerialAssistant.vi 程序流程图

2.5.1.2 遥控程序结构

遥控监控程序为 `remote.vi`，主要功能为通过蓝牙串口对机器人发包，控制机器人的前进、转弯、驻停等动作。

该程序图形化程序如图 2.13 所示，程序中预先根据通讯包格式设置了不同动作所对应的通讯包内容。在程序中存在一个类似于按键响应的中断，当按键按下时，自动跳入中断响应程序，根据按下键的键值执行不同的发包指令，发出不同的包内容来控制机器人不同的行为。

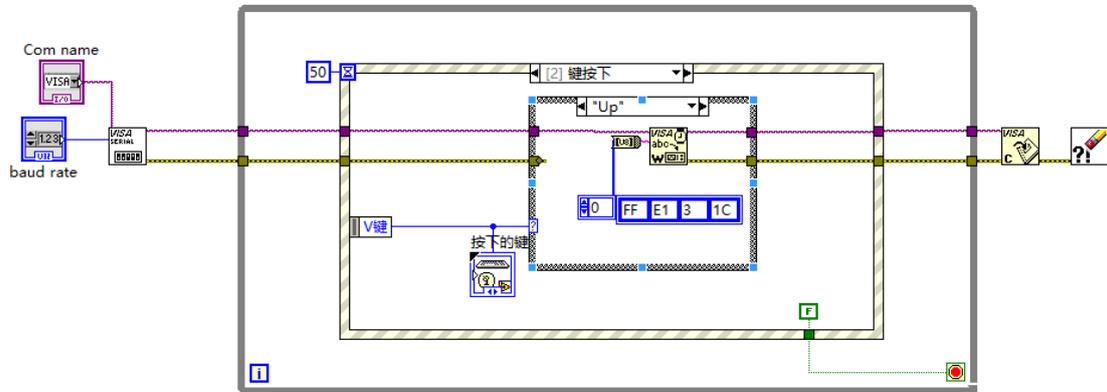


图 2.13 `remote.vi` 程序图

其前面板如图 2.14 所示。前面板中主要是串口通讯相关的参数配置，以及一个现实所按下的键的显示框。具体的遥控操作通过键盘按键实现，上键为前进，左键为左转，右键为右转，`shift` 键为停止。当机器人运动时，按下左转键，机器人将以一个指定的半径进行左转；当机器人停止时按下左转键，机器人将原地左转，右转亦然。



图 2.14 remote.vi 面板图

2.5.1.3 其余子 vi 程序

为完成主监控程序的功能，依照需求还完成了 4 个子 vi，其具体结构不再赘述，仅列举其名称和作用。

F2U.vi

浮点数转化为四个字节无符号数子 vi，转换过程遵循 IEEE754 浮点数定义规则，用于程序中浮点数数据打包过程。

U2F.vi

四个字节无符号数转化为浮点数子 vi，转换过程遵循 IEEE754 浮点数定义规则，用于程序中浮点数数据解包过程。

SAtools.vi

辅助计算子 vi，人为定义数据时计算整包数据格式所用，避免复杂的人工计算过程。

Save2File.vi

数据储存到文件子 vi，用于过程数据解包完成后依次存入同一个新建文件。避免重复覆盖文件，简化文件操作过程。

2.5.2 路径规划程序结构

路径规划程序由 c 语言编写，主要功能为在地图和障碍物已知的情况下，完成从初始点到目标点的路径规划，并下发执行。其机构框图如图 2.15 所示。

本设计中，在程序开始时需导入采集好的地图及障碍物信息，并设置好起始点与终点，通过路径规划得到在起始点和终点间可以通行的一段路径，并下发执行。下发执行过程中用到了机器人程序中累计的里程计，根据里程信息判断机器人此时所处的位置，实时下发基于当前位置的后续控制命令。本次设计进行的同时还邀请到了实验室其他成员，对在车上设置的深度相机进行开发，可以根据深度相机采集到的信息，在地图中实时显示出前方的实际地图信息。



图 2.15 路径规划流程图

2.5.3 手机控制程序结构

手机控制程序由 java 语言编写，运行环境为 android 4.2 手机。手机控制程序由两大部分组成，一个部分简单手机遥控，另一部分是手绘轨迹响应。

简单手机遥控主要是通过手机蓝牙串口对机器人发送命令包，控制机器人的

基本运动，如前进，左转，等，其结构功能与 labview 程序中的 remote 子 vi 类似，同样能完成 remote 子 vi 的控制效果。其程序界面截图如图 2.16 左图所示。该程序中还有很多空闲按钮，可根据后续控制需求进行进一步开发使用。

手绘轨迹响应程序主要是在屏幕中设置一块画布，当手在其上运动时可以记录下手绘轨迹。程序将手绘轨迹解析为一段段直线段的连接，并计算出每个转折点的轨迹长度与朝向，将这段轨迹信息根据机器人运动情况逐一下发，实现手绘轨迹响应。其程序截图如图 2.16 右图所示。



图 2.16 手机控制程序

2.6 系统测试

2.6.1 机器人运行测试

机器人系统启动并完成初始化后，将采集 IMU 测量得到的机器人姿态和码盘测量得到的机器人速度，通过平衡控制器计算控制两个电机所需的 PWM 波占空比，输出控制信号并驱动电机转动，实现平衡及运动控制。运动过程参数通过板载

蓝牙与上位机进行通讯。机器人自平衡如图 2.17 所示。



图 2.17 机器人自平衡图

2.6.2 上层监控程序测试

机器人启动后，当开启向上传输数据的标志时，将定时将自身运动过程信息上传。Labview 编写的上层监控程序将实时显示上传数据。Labview 显示面板如图 2.18 所示。其中上半部分表示机器人当前速度，下半部分表示机器人当前角度。

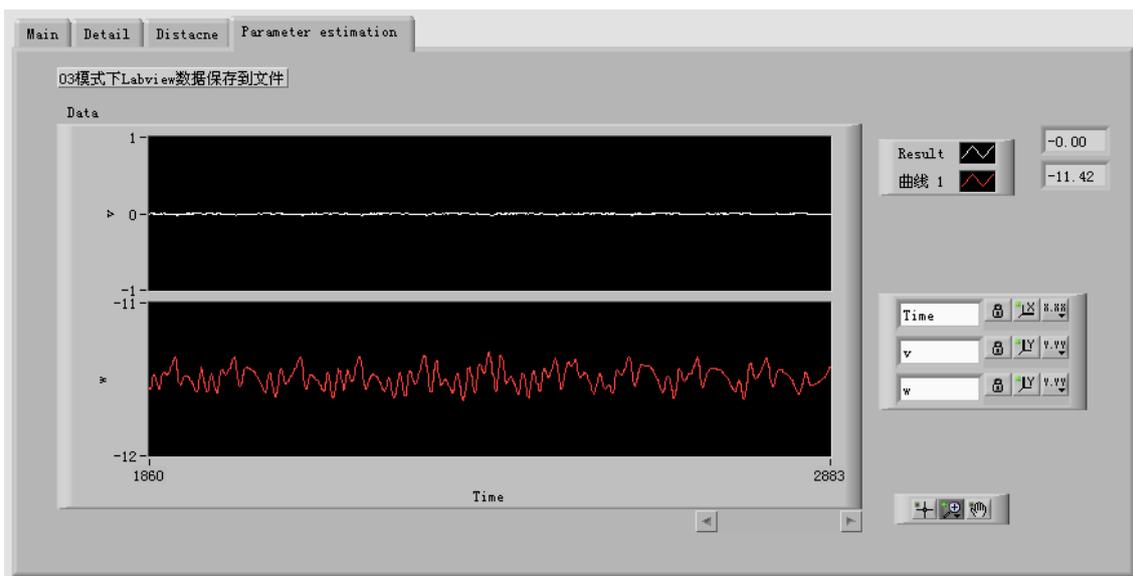


图 2.18 Labview 监控程序显示界面

上传的数据同时将保存在本地 txt 文件中，供后续分析使用。将本地 txt 文件中的数据用 Matlab 对应编写的函数进行绘制后可得图如 2.19 所示。可以看出上传的数据绘制后和 Labview 显示的波形相同。

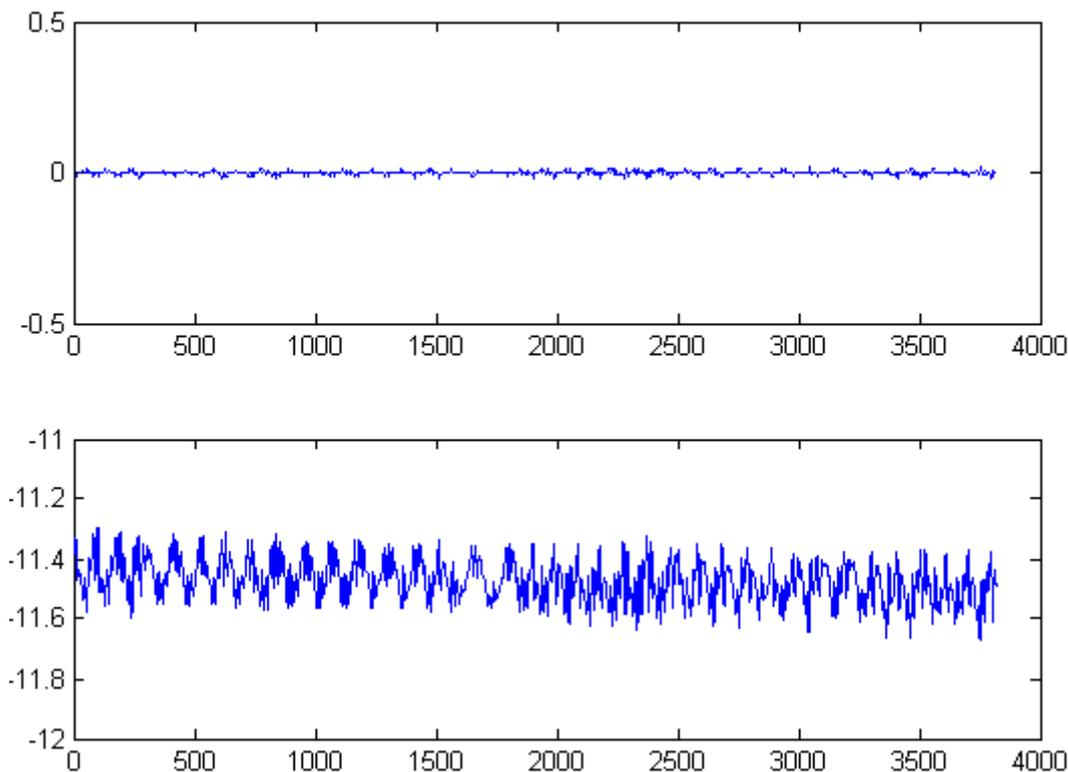


图 2.19 matlab 数据绘制显示

监控程序遥控部分启动后,通过按键可实时控制机器人执行简单运动,当按键按下时,程序面板将显示按下的键值,如图 2.20 所示。



图 2.20 Labview 遥控程序运行面板

2.6.3 上层路径规划程序测试

上层路径规划程序开启后,首先要导入地图信息和障碍物信息。导入后路径规划程序界面如图 2.21 所示。

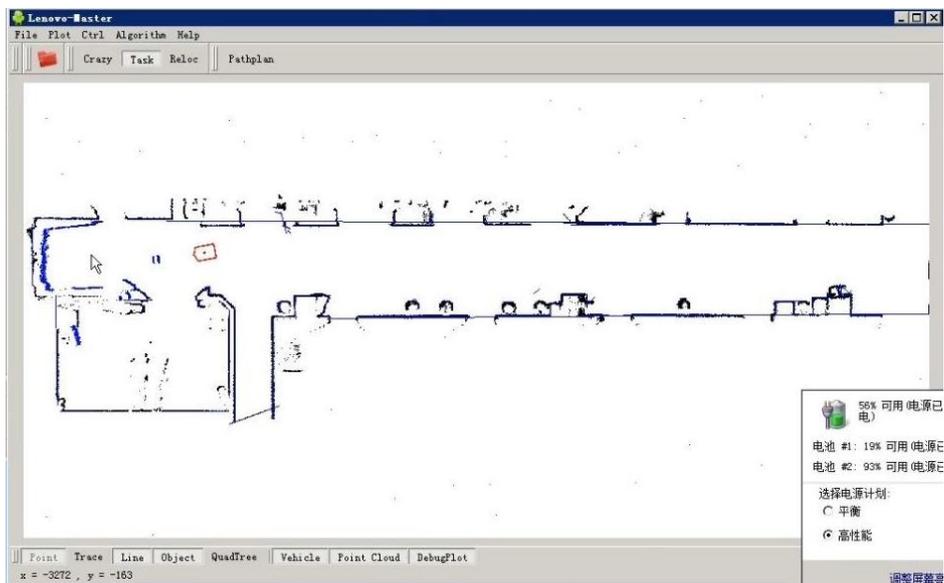


图 2.21 路径规划程序运行界面

用鼠标确定起点并选取终点后，将由轨迹规划模块自动生成从起点到终点之间的路径的路径，如图 2.22 所示。路径下发后机器人将循着这条路径运动。

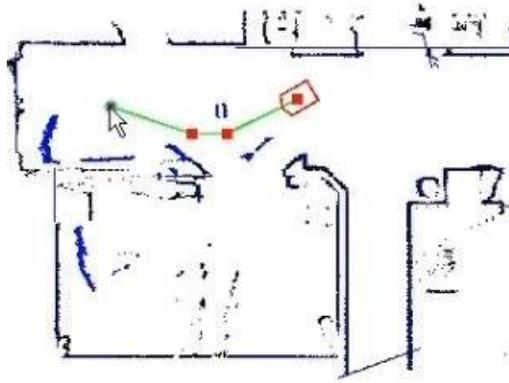


图 2.22 路径规划结果

2.6.4 手机控制程序测试

开启手机蓝牙并与机器人建立通讯后，在简单遥控程序界面按下对应控制键后，将对机器人下发相应的控制包，控制机器人运动。实际测试过程中，按下相应的键后，机器人将对应进行相应的运动。

在手绘轨迹跟踪界面，用手描绘一条随机的轨迹后，按下计算按钮，程序将自动解析路径信息，并根据底层控制需求，计算出轨迹中的关键控制点对，并在程序下方的显示窗口中。如图 2.23 所示，这一系列控制信号随机器人运动下发后，机器人能跟踪出对应的光滑轨迹。

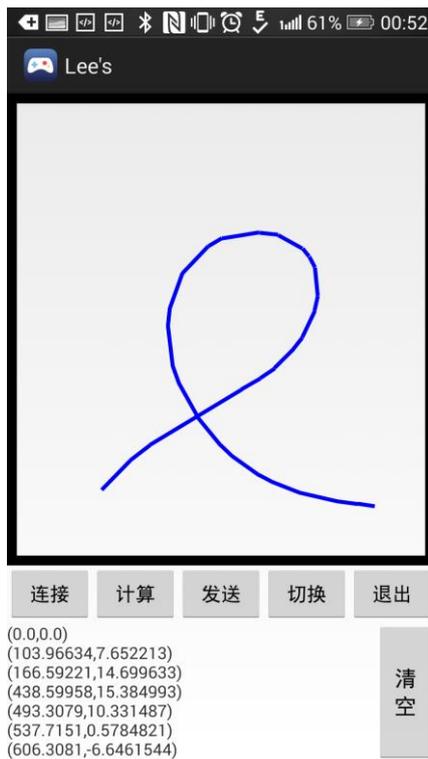


图 2.23 手绘轨迹计算结果

2.7 本章小结

本章阐述了两轮机器人整体硬件设计原理、底层硬件结构、底层软件结构以及多平台下程序结构，搭建了一个能实现平衡功能，并能实现实时数据上传，实时上层软件控制的两轮机器人系统。经测试，所设计的系统各部分功能实现良好。

3 平衡运动建模与控制方法

要对两轮机器人进行控制，需先对机器人整体建模，了解其系统特性，再根据建好的模型设计控制器。而在实际的控制器应用到实际的过程中，需要将计算所得的不可直接控制的力矩转换为可以直接控制的 PWM 波占空比，这一转化工程需要用到电机模型。

故本次设计中先需要对机器人整体和电机分别建模，并将机器人非线性模型线性化得到在小角度近似下的系统线性模型，然后依据线性模型提出状态反馈的控制方法。状态反馈控制器的反馈矩阵利用 LQR 进行整定并根据实际运动状况进行调整。

针对平衡约束下的运动控制，本文提出了“虚拟质心”控制的控制方法。利用虚拟质心的前后调整，通过状态反馈使机器人整体加速或减速，来达到平衡约束下的速度控制，完成机器人直行运动。本文还专门设计旋转控制算法实现机器人原地旋转与带半径旋转控制。

3.1 机器人平衡建模

3.1.1 Lagrange 方程建模

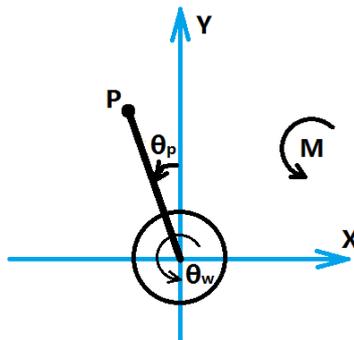


图 3.1 两轮机器人抽象模型

各物理量正方向如图 3.1 所示，其含义如下表所示：

表 3.1 两轮机器人参数表

参数符号	参数意义
l	机器人主体（除轮外）质心 p 到轮轴距离
R	轮子半径
x_p	机器人质心位移
θ_p	机器人质心倾斜角
m_p	机器人主体（除轮外）质量
J_p	机器人主体（除轮外）绕轮轴转动惯量
x_w	两轮质心位移
θ_w	两轮转角
m_w	一个轮子的轮子质量
J_w	一个轮子的转动惯量
E_{kp1}	机器人主体（除轮外）平动动能
E_{kp2}	机器人主体（除轮外）转动动能
E_{up}	机器人主体（除轮外）势能
E_{kw1}	两轮总平动动能
E_{kw2}	两轮总转动动能
E_{uw}	两轮总势能
M	一个电机输出的转矩

由于机器人前进后退和旋转是不耦合的^[1]，此处仅分析和直立与前进后退，则两轮运动状态完全一致。于是对于机器人主体 p 有

$$\begin{cases} x_p = x_w - l \sin \theta_p \\ y_p = l \cos \theta_p \end{cases} \rightarrow \begin{cases} \dot{x}_p = \dot{x}_w - l \cos \theta_p \dot{\theta}_p \\ \dot{y}_p = -l \sin \theta_p \dot{\theta}_p \end{cases} \quad (3.1)$$

故车体动能为

$$\begin{aligned} E_{kp1} &= \frac{1}{2} m_p \left[(\dot{x}_w - l \cos \theta_p \dot{\theta}_p)^2 + (-l \sin \theta_p \dot{\theta}_p)^2 \right] \\ &= \frac{1}{2} m_p \dot{x}_w^2 - m_p \dot{x}_w l \cos \theta_p \dot{\theta}_p + \frac{1}{2} m_p l^2 \dot{\theta}_p^2 \end{aligned} \quad (3.2)$$

$$E_{kp2} = \frac{1}{2} J_p \dot{\theta}_p^2 \quad (3.3)$$

车体势能为

$$E_{up} = m_p g l \cos \theta_p \quad (3.4)$$

对于轮子 W

$$E_{kw1} = 2 * \frac{1}{2} m_w \dot{x}_w^2 \quad (3.5)$$

$$E_{kw2} = 2 * \frac{1}{2} J_w \dot{\theta}_w^2 \quad (3.6)$$

$$E_{up} = 2m_w g R = C \quad (3.7)$$

假定轮子与地面无相对滑动，则

$$\dot{x}_w = -R\dot{\theta}_w \quad (3.8)$$

能量耗散不计，总动能为 E_k ，总势能为 E_u ，代入上式可得 Lagrange 方程为

$$\begin{aligned} L &= E_k - E_u \\ &= \left(\frac{m_p R^2}{2} + m_w R^2 + J_w \right) \dot{\theta}_w^2 + m_p l R \cos \theta_p \dot{\theta}_w \dot{\theta}_p + \frac{1}{2} (2m_p l^2 + J_p) \dot{\theta}_p^2 \\ &\quad - m_p g l \cos \theta_p \\ &= K_1 \dot{\theta}_w^2 + K_2 \cos \theta_p \dot{\theta}_w \dot{\theta}_p + K_3 \dot{\theta}_p^2 + K_4 \cos \theta_p \end{aligned} \quad (3.9)$$

式中

$$K_1 = \frac{m_p R^2}{2} + m_w R^2 + J_w$$

$$K_2 = m_p l R$$

$$K_3 = \frac{1}{2} (2m_p l^2 + J_p)$$

$$K_4 = -m_p g l$$

由 Lagrange 方程得：

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_w} \right) - \frac{\partial L}{\partial \theta_w} = M \quad (3.10)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_p} \right) - \frac{\partial L}{\partial \theta_p} = -2M \quad (3.11)$$

则

$$\frac{\partial L}{\partial \dot{\theta}_w} = 2K_2 \dot{\theta}_w + K_2 \cos \theta_p \dot{\theta}_p \quad (3.12)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_w} \right) = 2K_1 \ddot{\theta}_w + K_2 \cos \theta_p \ddot{\theta}_p - K_2 \sin \theta_p \dot{\theta}_p^2 \quad (3.13)$$

$$\frac{\partial L}{\partial \theta_w} = 0 \quad (3.14)$$

可得

$$2K_1 \ddot{\theta}_w + K_2 \cos \theta_p \ddot{\theta}_p - K_2 \sin \theta_p \dot{\theta}_p^2 = M \quad (3.15)$$

同理

$$\frac{\partial L}{\partial \dot{\theta}_p} = K_2 \cos \theta_p \dot{\theta}_w + 2K_3 \dot{\theta}_p \quad (3.16)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_p} \right) = K_2 \cos \theta_p \ddot{\theta}_w - K_2 \sin \theta_p \dot{\theta}_w \dot{\theta}_p + 2K_3 \ddot{\theta}_p \quad (3.17)$$

$$\frac{\partial L}{\partial \theta_p} = -K_2 \sin \theta_p \dot{\theta}_w \dot{\theta}_p - K_4 \sin \theta_p \quad (3.18)$$

可得

$$K_2 \cos \theta_p \ddot{\theta}_w + 2K_3 \ddot{\theta}_p + K_4 \sin \theta_p = -2M \quad (3.19)$$

即通过以上可得可得其运动模型为

$$\begin{cases} 2K_1 \ddot{\theta}_w + K_2 \cos \theta_p \ddot{\theta}_p - K_2 \sin \theta_p \dot{\theta}_p^2 = M \\ K_2 \cos \theta_p \ddot{\theta}_w + 2K_3 \ddot{\theta}_p + K_4 \sin \theta_p = -2M \end{cases} \quad (3.20)$$

3.1.2 非线性模型表述

将上述运动模型式(3.20)改写，得：

$$\begin{cases} 2K_1 \ddot{\theta}_w + K_2 \cos \theta_p \ddot{\theta}_p = M + K_2 \sin \theta_p \dot{\theta}_p^2 \\ K_2 \cos \theta_p \ddot{\theta}_w + 2K_3 \ddot{\theta}_p = -2M - K_4 \sin \theta_p \end{cases} \quad (3.21)$$

即

$$\begin{cases} b_1 \ddot{\theta}_w + b_2 \ddot{\theta}_p = b_3 \\ b_4 \ddot{\theta}_w + b_5 \ddot{\theta}_p = b_6 \end{cases} \quad (3.22)$$

式中

$$\begin{aligned} b_1 &= 2K_1, b_2 = K_2 \cos \theta_p, b_3 = M + K_2 \sin \theta_p \dot{\theta}_p^2 \\ b_4 &= K_2 \cos \theta_p, b_5 = 2K_3, b_6 = -2M - K_4 \sin \theta_p \end{aligned}$$

则其非线性模型表述如下

$$\frac{d}{dt} \begin{bmatrix} \theta_p \\ \theta_w \\ \dot{\theta}_p \\ \dot{\theta}_w \end{bmatrix} = \begin{bmatrix} \dot{\theta}_p \\ \dot{\theta}_w \\ \frac{b_3 b_4 - b_1 b_6}{b_2 b_4 - b_1 b_5} \\ \frac{b_3 b_5 - b_2 b_6}{b_1 b_5 - b_2 b_4} \end{bmatrix} \quad (3.23)$$

以上非线性模型可应用于 Matlab 仿真。

3.1.3 线性模型表述

将上述运动模型线性化，即取

$$\begin{aligned} \cos \theta_p &= 1 \\ \sin \theta_p &= \theta_p \end{aligned}$$

$$\dot{\theta}_p^2 = 0$$

得

$$\begin{cases} 2K_1\ddot{\theta}_w + K_2\ddot{\theta}_p = M \\ K_2\ddot{\theta}_w + 2K_3\ddot{\theta}_p = -2M - K_4\theta_p \end{cases} \quad (3.24)$$

分别消去可得

$$\begin{cases} (K_2^2 - 4K_1K_3)\ddot{\theta}_p = (K_2 + 4K_1)M + 2K_1K_4\theta_p \\ (4K_1K_3 - K_2^2)\ddot{\theta}_w = (2K_2 + 2K_3)M + K_2K_4\theta_p \end{cases} \quad (3.25)$$

即

$$\begin{cases} a_1\ddot{\theta}_p = a_2M + a_3\theta_p \\ a_4\ddot{\theta}_w = a_5M + a_6\theta_p \end{cases} \quad (3.26)$$

式中

$$\begin{aligned} a_1 &= K_2^2 - 4K_1K_3, a_2 = K_2 + 4K_1, a_3 = 2K_1K_4 \\ a_4 &= 4K_1K_3 - K_2^2, a_5 = 2K_2 + 2K_3, a_6 = K_2K_4 \end{aligned}$$

则其线性模型表述如下：

$$\frac{d}{dt} \begin{bmatrix} \theta_p \\ \theta_w \\ \dot{\theta}_p \\ \dot{\theta}_w \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{a_3}{a_1} & 0 & 0 & 0 \\ \frac{a_6}{a_4} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_p \\ \theta_w \\ \dot{\theta}_p \\ \dot{\theta}_w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{a_2}{a_1} \\ \frac{a_5}{a_4} \end{bmatrix} M \quad (3.27)$$

3.1.4 LQR 反馈矩阵计算与仿真

利用 Matlab 相应工具箱可计算 LQR 反馈矩阵，基于 LQR 反馈矩阵系数，通过编程实现状态反馈控制。

利用 LQR 确定反馈矩阵，其中

$$R = 1, Q = \begin{bmatrix} 250 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, x = \begin{bmatrix} \theta_p \\ \theta_w \\ \dot{\theta}_p \\ \dot{\theta}_w \end{bmatrix}$$

则在偏离平衡位置时(6°)，对应控制效果如下：

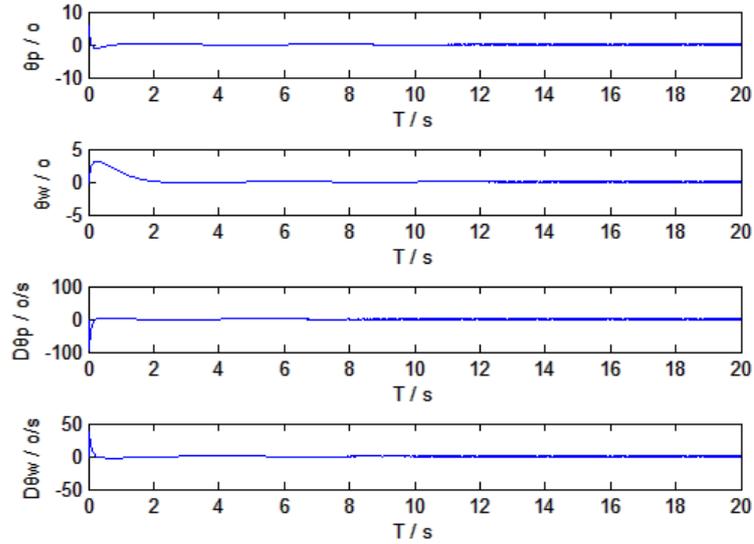


图 3.2 平衡控制效果仿真

图 3.2 中各参数从上到下依次为机器人车身偏角、车轮转角、车身偏角角速度、车轮转角角速度。当车身偏角有一个偏移量时，机器人将快速向前加速，使其主体产生一个与偏角方向相反的角速度，矫正车身偏角，并能在 2s 的调整时间内快速稳定。

3.2 电机建模

3.2.1 电机模型表述

在机器人硬件设计中，可以通过控制向电机输出的 PWM 波的占空比来控制电机动作，但电机输出扭矩不直接可控，为了使得状态反馈控制器输出的期望扭矩转化为输出的 PWM，需要用到电机的理论模型，建立扭矩与 PWM 间的关系。电机的抽象模型如图 3.3 所示。

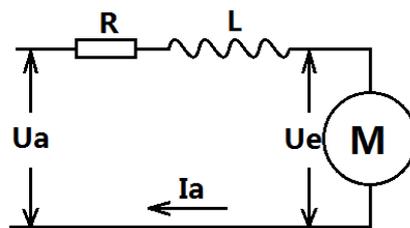


图 3.3 抽象电机模型

电机相关的物理量如下表所示：

表 3.2 电机模型参数表

参数符号	参数意义
T_m	电机输出转矩
ω_m	转子转速
U_a	电枢电压
U_e	反电动势
I_a	电枢电流
K_m	电机扭矩系数
K_e	电机反电动势系数
R	电枢电阻

故从电机模型可得

$$\begin{cases} U_a = RI_a + U_e \\ U_e = K_e \omega_m \\ T_m = K_m I_a \end{cases} \quad (3.28)$$

电机转速不高时，可以忽略电枢电感 L ，粘性阻尼系数 B ，从而得到上述方程，

化简得：

$$T = -\frac{K_e K_m}{R} \omega_m + \frac{K_m}{R} u_a \quad (3.29)$$

查阅对应 Maxon 电机手册可得

- ◆ $R=1.2\Omega$
- ◆ $K_m=25.5 \text{ mNm/A}$
- ◆ $K_e=1/374 \text{ V/rpm}$

即

$$T = -5.425 * 10^{-4} * \omega_m + 0.02125 u_a \quad (3.30)$$

3.2.2 实际电机数据分析

采用实际电机数据分析，电压用占空比代替，则应有

$$T = -K_w \omega_m + K_{pwm} PWM \quad (3.31)$$

对测控测试数据分析，把扭矩关于转速做线性拟合，得 K_w ，则对应截距即为 $K_w * K_{pwm} * PWM$ ，除即得 K_{pwm} ，则大致有如下关系：

$$T = -4 * 10^{-4} * \omega_m + 1 * 10^{-4} * PWM \quad (3.32)$$

又

$$u_a = 16 * PWM / 2500 \quad (3.33)$$

则

$$T = -4 * 10^{-4} * \omega_m + 0.015625 * u_a \tag{3.34}$$

与理论值近似。

3.3 机器人控制方法设计

3.3.1 机器人控制方法综述

总体控制算法基于状态反馈，形成基础平衡控制模块作为内环，在不同的期望下，采取不同的控制策略，形成外环对机器人进行控制，用类似串级的控制结构优化平衡约束下的运动控制效果。为了完成状态反馈控制器，需要对机器人进行建模与分析。分析中所用的系统输入为电机扭矩 M ，相应控制器的输出为期望扭矩 M ，并通过理论电机模型建立 PWM 与电机扭矩的关系，实现对期望扭矩的控制。控制算法框架图如图 3.4 所示。

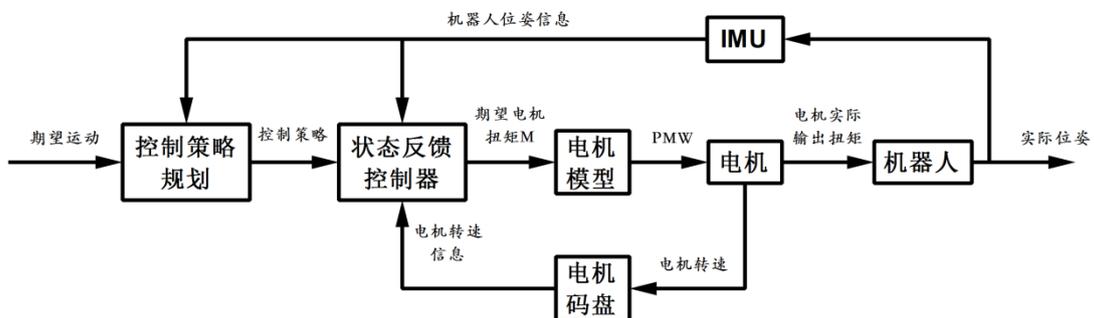


图 3.4 控制算法框架图

3.3.2 基础平衡控制算法设计

基础平衡控制算法采用状态反馈法，四个状态量分别为机器人倾斜角度、机器人倾斜角速度、机器人位移及机器人平均轮速。通过实际物理参数测量与 Lagrange 方程建模，得出两轮机器人的数学模型，利用 Matlab 工具箱求解反馈矩阵，并进行对应编程，在实际两轮机器人上实现。综合考虑机器人平衡特性及实际调试参数的不同效果，最终确定的机器人倾斜角度、机器人倾斜角速度、机器人位移反馈较大，机器人平均轮速反馈较小。

3.3.3 直行运动控制算法设计

机器人直行控制的基本要求是在期望速度下，机器人平稳直立直行。故需对其速度进行反馈。但在调试中发现，若基于上述平衡算法，调整速度反馈参数大小，速度控制效果始终不理想。因为在两轮机器人上单纯对速度反馈系数做调整的过程中，未考虑机器人运动过程的平衡约束。在平衡控制与速度控制有较强耦合的情况下无法利用单一反馈实现平衡约束下的速度控制。

故改变速度控制算法设计思路。考虑到Segway式的两轮机器人在运动过程中，机器人的速度控制主要靠人的前倾与后倾实现。则研究采取类似的策略对应调整本设计中所使用的两轮机器人是否也可实现机器人整体的加速与减速。

基于此思路，算法设计引入“虚拟质心”概念，采用类似串级结构，基础平衡为内环，“虚拟质心”作为外环速度控制的计算结果，同时最为内环质心位置的给定量。有别于实际质心，虚拟质心是机器人控制算法对机器人进行控制调整后，希望实际质心所处的位置。而控制策略总会将质心调整并保持到虚拟质心附近。该种算法的控制过程如图 3.5 所示。图中橙点为虚拟质心，物理上不存在；蓝色为实际质心，物理上存在。大箭头为前进方向。

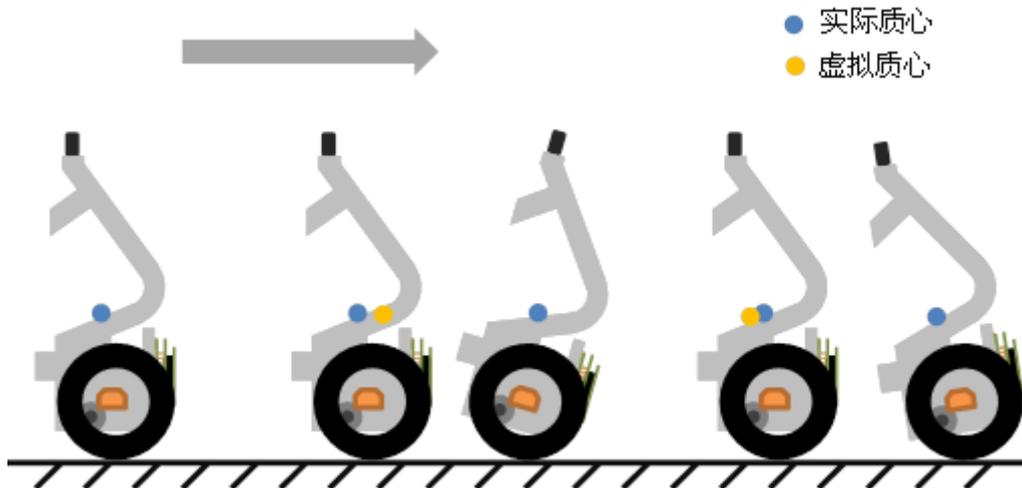


图 3.5a 直立或匀速

图 3.5b 加速

图 3.5c 减速

当要求机器人直立驻停或匀速运动时，虚拟质心与实际平衡时的质心重合，如图 3.5a，此时机器人本体质心合外力为 0，运动状态不改变；当要求机器人加速时，虚拟质心将偏向机器人加速方向，如图 3.5b，此时控制策略将使机器人保持前倾的

姿态，则其质心受水平向前的合外力，存在向前的加速度。故其质心会向前加速导致倾角变大，控制策略为了保持当前倾角，会控制轮子也相应加速，使得整车加速；当要求机器人减速时，同理如图 3.5c。

3.3.4 旋转运动控制算法设计

由于旋转与直行不耦合，因此对于原地旋转较处理好，在平衡不运动的基础上，即均速为 0 的基础上，增加一个轮子控制信号输出，减小另一个轮子的控制信号输出，即可实现朝一个方向原地旋转。加减换号，即可实现朝另一个方向旋转。如图 3.6 所示。



图 3.6 原地旋转

而在机器人运动过程中，由于实际两轮着地点不明确，两轮各自速度难以分别准确控制。因此需按期望半径转弯时，利用两轮距离和半径关系，计算出两轮各自所需的速度，并加以控制这样的策略不可取。但鉴于上述加减速可行，考虑通过整体速度进行转弯控制。即利用转弯半径和圆弧长度的关系，两轮机器人自行累积自转弯开始时所运动的路程，与期望的转弯半径一同可以计算出当前期望朝向角。如图 3.7 所示。在均速不为 0 的情况下，增加单一侧轮子的控制信号输出，使机器人向目标方向偏转，控制当前朝向角追踪期望朝向角。

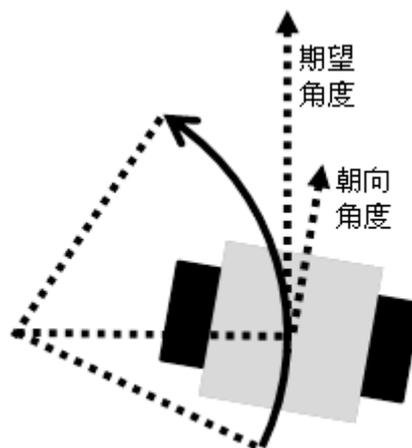


图 3.7 半径旋转

3.4 机器人控制效果

3.4.1 原地平衡控制效果

原地平衡时，机器人姿态如图 3.8 所示。



图 3.8 机器人原地平衡

通过机器人板载蓝牙数据上传，Labview 数据记录，最终可以得到原地平衡时，其速度和角度的变化关系，经 Matlab 绘制如图 3.9 所示。

从图中可以看出，机器人原地平衡时，机器人平衡效果良好，其在平衡过程中，速度有微小波动，但其波动幅度小于 0.01m/s ，整体表现为在原地直立，并几乎不发生前后移动，最大的前后位移不超过 1cm 。在平衡过程中，其车身偏角同样有微小波动，车身偏角波动幅度小于 0.2° ，通过肉眼观察，其整体将进行极微小的振动。

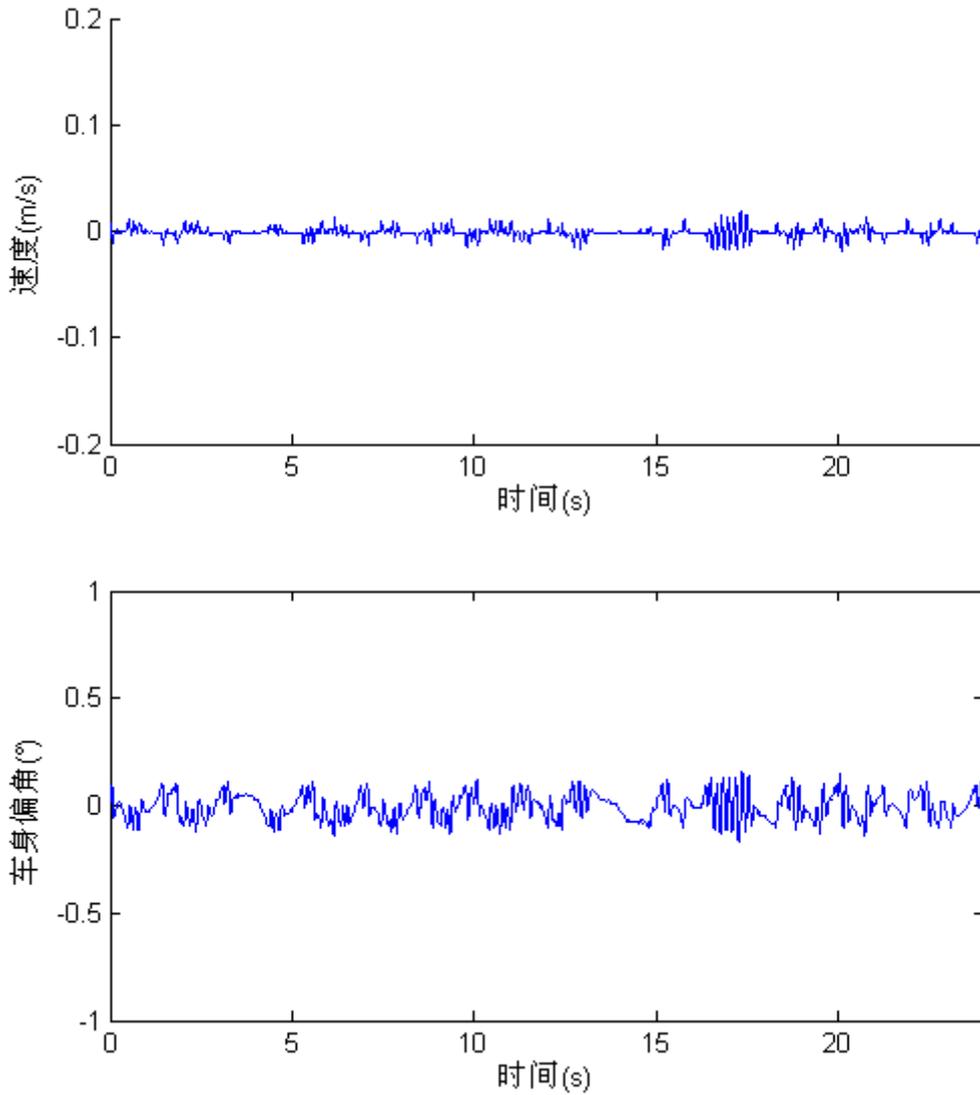


图 3.9 机器人原地平衡过程状态

在原地自平衡状态下，测试其抗扰动能力。在 120s 附近对机器人系统施加扰动。可到其响应如图 3.10 所示。

从图中可知，当在 120s 附近对机器人施加干扰时，其车身偏角因干扰达到过 6° ，但在调整过程中很快恢复到平衡状态，大概在 121.5s 左右完全恢复平衡。整体恢复时间小于 2s。观察其速度变化情况，因扰动速度激增至 0.3m/s 附近，后经调整速度反向增长，并最终趋于稳定。观察速度在正负方向大致呈对称形状，其宏观表现为，机器人因扰动偏离原位置，后经调整后回到受扰动前的位置，保持在原地平衡。

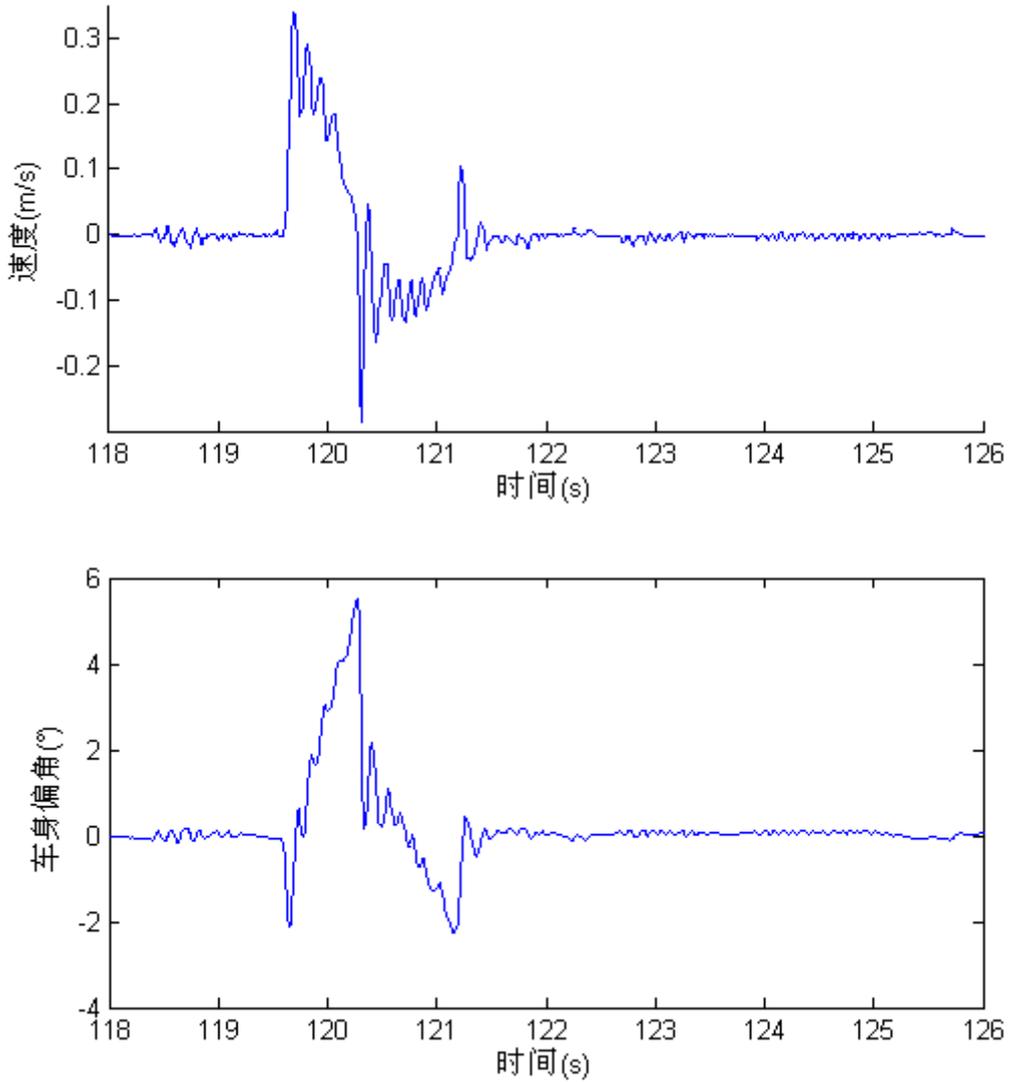


图 3.10 机器人原地平衡抗扰动过程状态

综上所述，通过状态反馈设计的平衡控制控制器维持机器人平衡效果良好。

3.4.2 “虚拟质心”直行运动控制效果

3.4.2.1 加速与匀速控制效果

在给定速度为 0.5m/s 时，机器人在加速与匀速运动过程中，其速度和虚拟质心偏置量如下图所示。

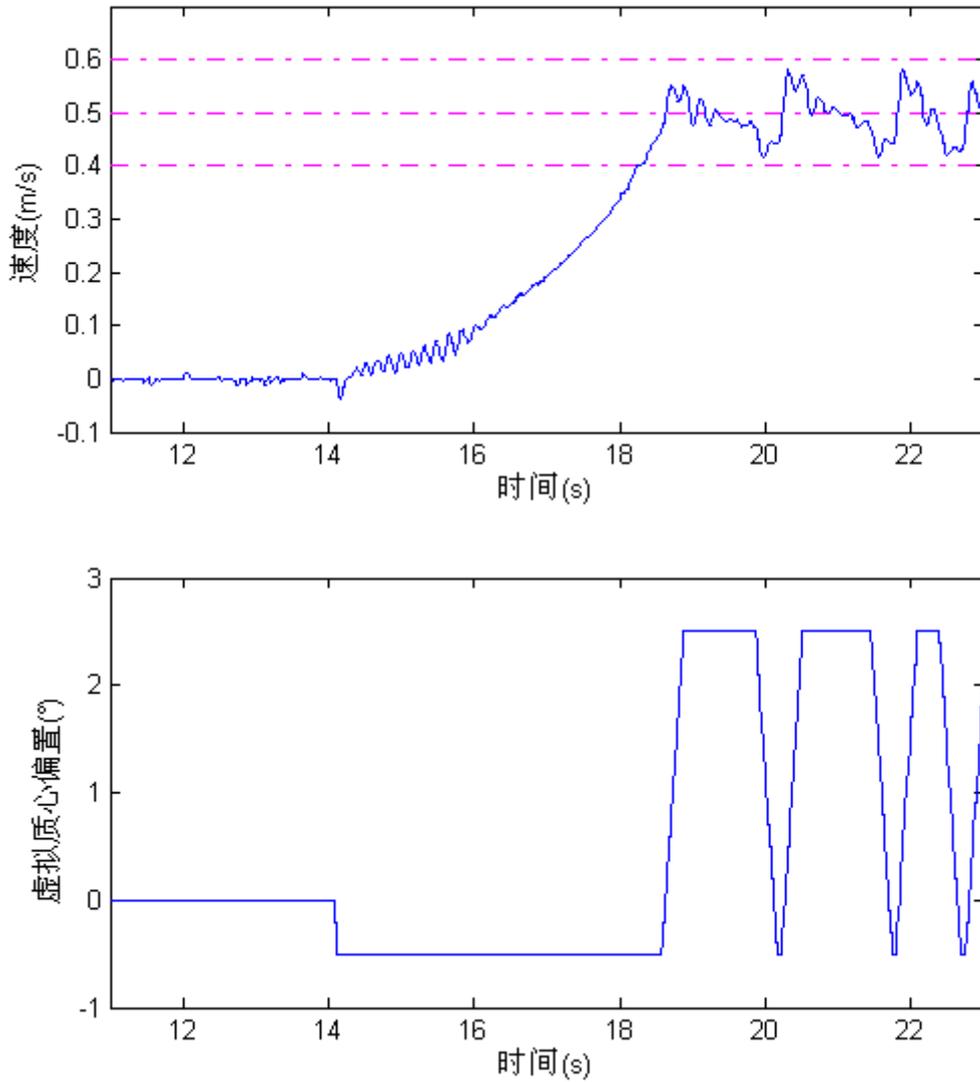


图 3.11 速度设定值为 0.5m/s 时加速与匀速运动过程状态

速度随时间变化的图中，三条虚线分别代表速度值为 0.4m/s、0.5m/s 和 0.6m/s 的标志线，从图中可以看出，当加速过程开始时，虚拟质心偏置迅速下降至 -0.5° ，并在加速过程中一直维持该值不变，在此期间，速度在初期有一个微小振荡，随后平稳增加至期望速度 0.5m/s。当进入匀速状态时，虚拟质心随速度不断调整，速度随之在 0.5m/s 上下波动，其波动幅度在 0.1m/s 内。

在给定速度为 1m/s 时，机器人在加速与匀速运动过程中，其速度和虚拟质心偏置量如下图所示。

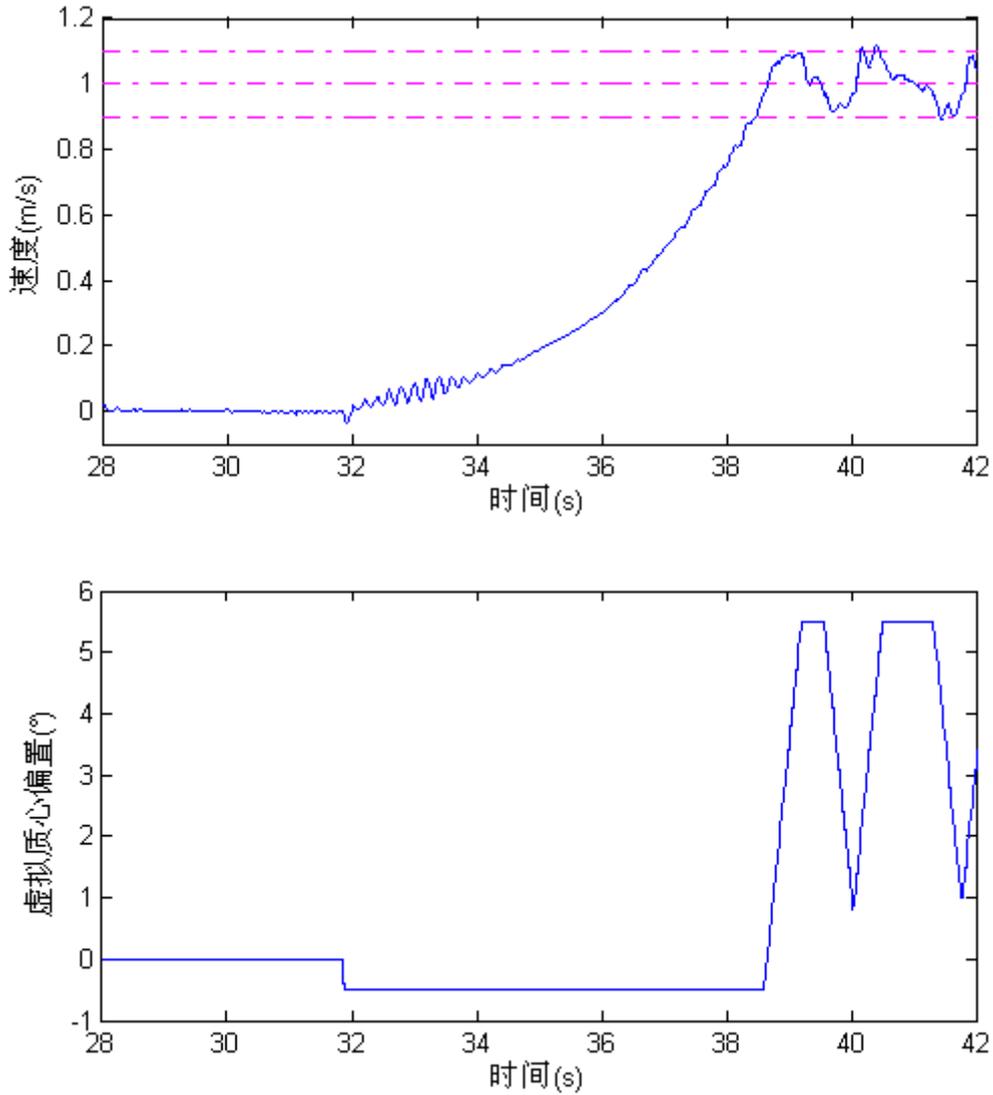


图 3.12 速度设定值为 1.0m/s 时加速与匀速运动过程状态

速度随时间变化的图中，三条虚线分别代表速度值为 0.9m/s、1.0m/s 和 1.1m/s 的标志线，从图中可以看出，当加速过程开始时，虚拟质心偏置迅速下降至 -0.5° ，并在加速过程中一直维持该值不变，在此期间，速度在初期有一个微小振荡，随后平稳增加至期望速度 1.0m/s。当进入匀速状态时，虚拟质心随速度不断调整，速度随之在 1.0m/s 上下波动，其波动幅度在 0.1m/s 内。

综上，在“虚拟质心”控制方法中，能很好的实现加速阶段机器人的平稳加速，由于目前对于加速阶段的虚拟质心偏置上限设置较小，故还需要稍长的加速时间。在匀速阶段，速度设定值为 0.5m/s 与 1.0m/s 的运动过程中，其匀速时的速度控制的绝对误差均在 0.1m/s 的范围内。0.5m/s 对机器人来说，运动速度较慢，其过程中因外界干扰引起的轻微摆动也有可能引起机器人速度相对较大的变化，故在低

速下,机器人的速度控制较为困难,速度控制相对误差较大。而当速度提高到 1.0m/s 时,外界干扰对于速度的影响相对减小,其绝对误差值不变,故其相对误差减小。故“虚拟质心”构想对于两轮机器人在平衡约束下的速度控制是有效的,而且随设定速度的增大,其相对误差有减小的趋势。

3.4.2.2 加速控制效果优化

根据“虚拟质心”控制,当“虚拟质心”在加速过程中偏置越大,机器人的姿态越倾斜,其获得的加速度应越大,故若加速过程过慢不符合控制要求,则可对调整加速过程“虚拟质心”偏置上限。上述控制中,其加速过程如下图所示

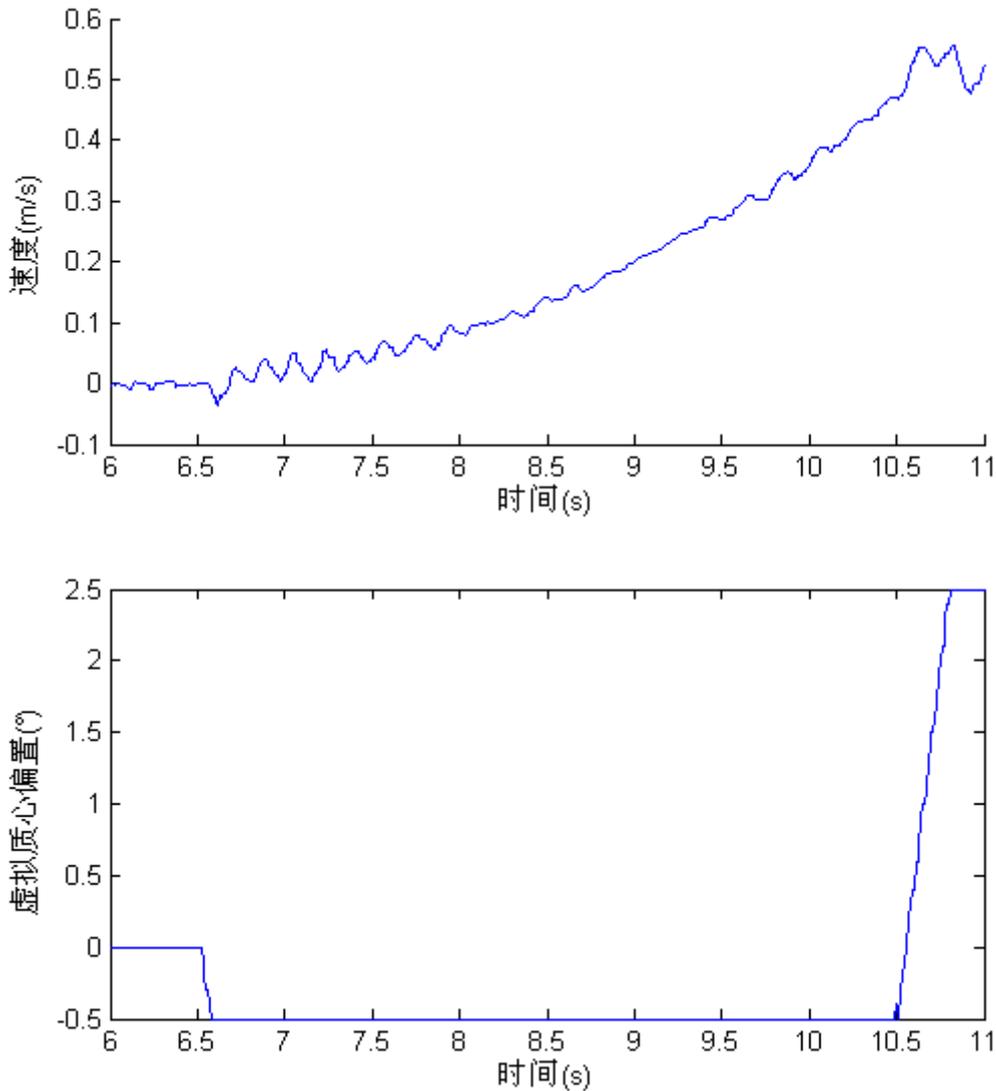


图 3.13 速度设定值为 0.5m/s 时加速过程速度变化

加速过程大概从 6.5s 开始到 10.5s 结束，加速过程共计约 4s，最终达到 0.5m/s 的设定速度。

增大“虚拟质心”在加速过程中的偏置上限后，其加速过程如下图所示。其加速过程大概在 8.5s 开始，在 10.7s 结束，加速过程共计 2.2s，最终达到 0.5m/s 的设定值，其加速过程明显缩短，故可采取调解“虚拟质心”在加速过程中的最大偏置来调解机器人加速性能。

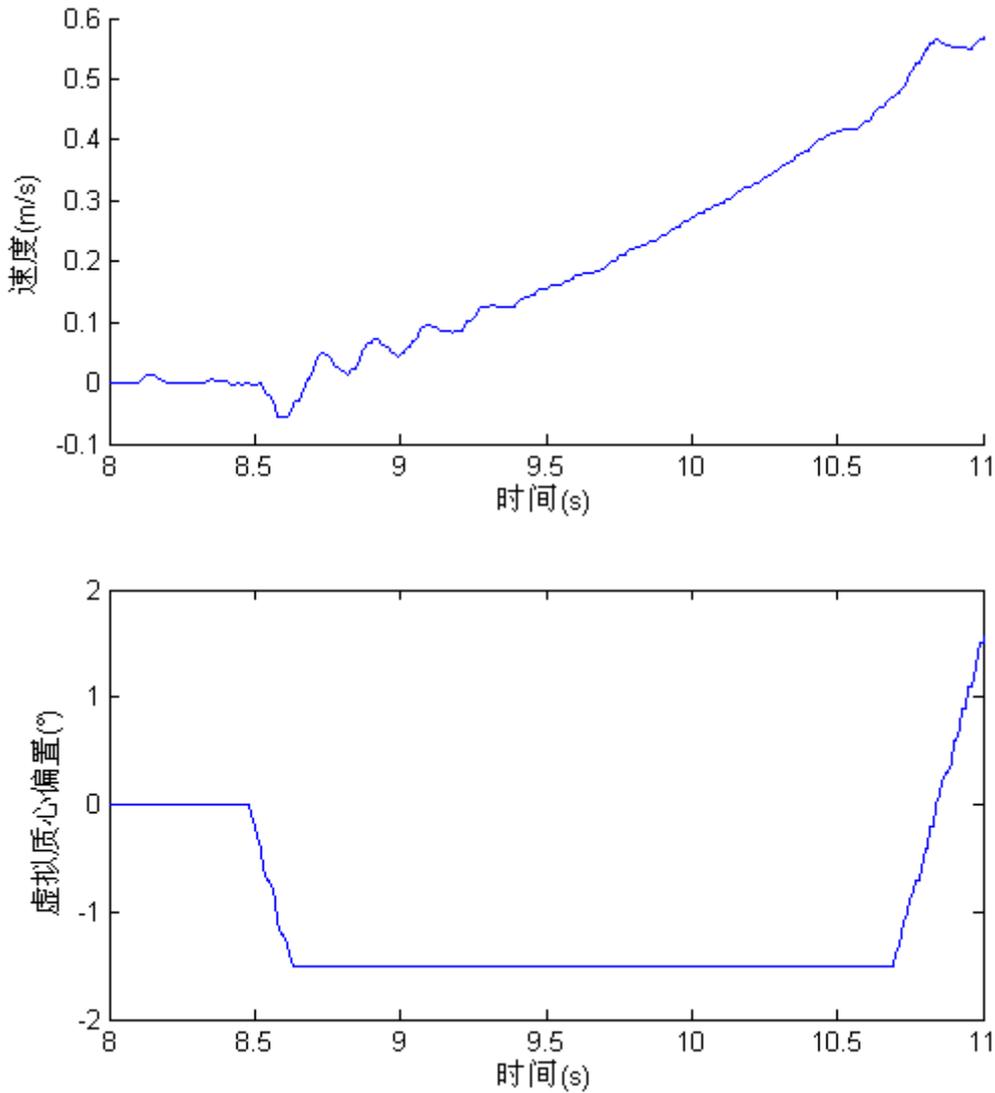


图 3.14 速度设定值为 0.5m/s 时加速过程速度变化

3.4.2.3 减速控制效果及优化

在给定速度为 0.5m/s 时，机器人在减速运动过程中，其速度和“虚拟质心”

偏置量如下图所示。其大概在 12s 左右开始减速，到 19s 左右减速过程完成。减速过程维持 7s 左右。在减速初期，速度对于“虚拟质心”偏置的响应不大，即速度变化不明显，在减速末期，为了将姿态调整回平衡状态，产生了一个速度的振荡，表现在机器人上则是减速末期有一个车身的振荡。其振荡幅度与速度的振荡幅度正相关，振荡及调整过程维持 2s。

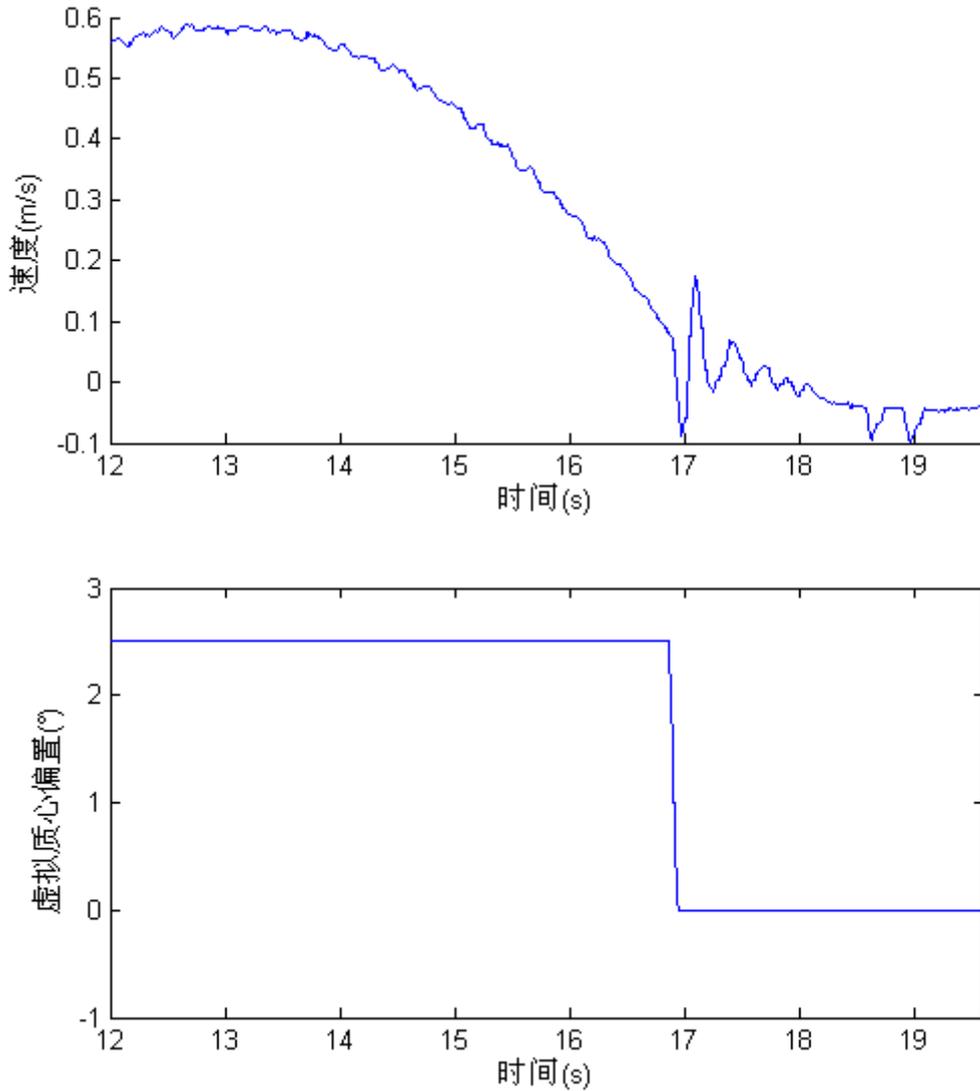


图 3.15 速度设定值为 0.5m/s 时小偏置减速过程状态

同样参考上述加速优化的思路，在下发减速命令后，设置不同于正常运行过程中的“虚拟质心”偏置上限，加快减速过程，其减速过程如图 3.16 所示。其大概在 67.5s 左右开始减速，到 69.5s 左右减速过程完成。减速过程维持 2s 左右。相比于不特别设置减速过程偏置而言，减速过程持续时间大大缩短。在减速初期，速度

对于“虚拟质心”偏置的响应明显，但会产生一个小振荡，并很快消失，振荡持续时间约为 0.5s，对系统影响不大。在减速末期，为了将姿态调整回平衡状态，同样产生了一个速度的振荡，但该振荡仅持续 0.5s，并表现在机器人上是直接将机器人从倾斜状态调整回平衡姿态，无明显振荡反应。故相比而言，单独提高在减速过程中的“虚拟质心”偏置上限，对于减速控制过程优化效果显著。

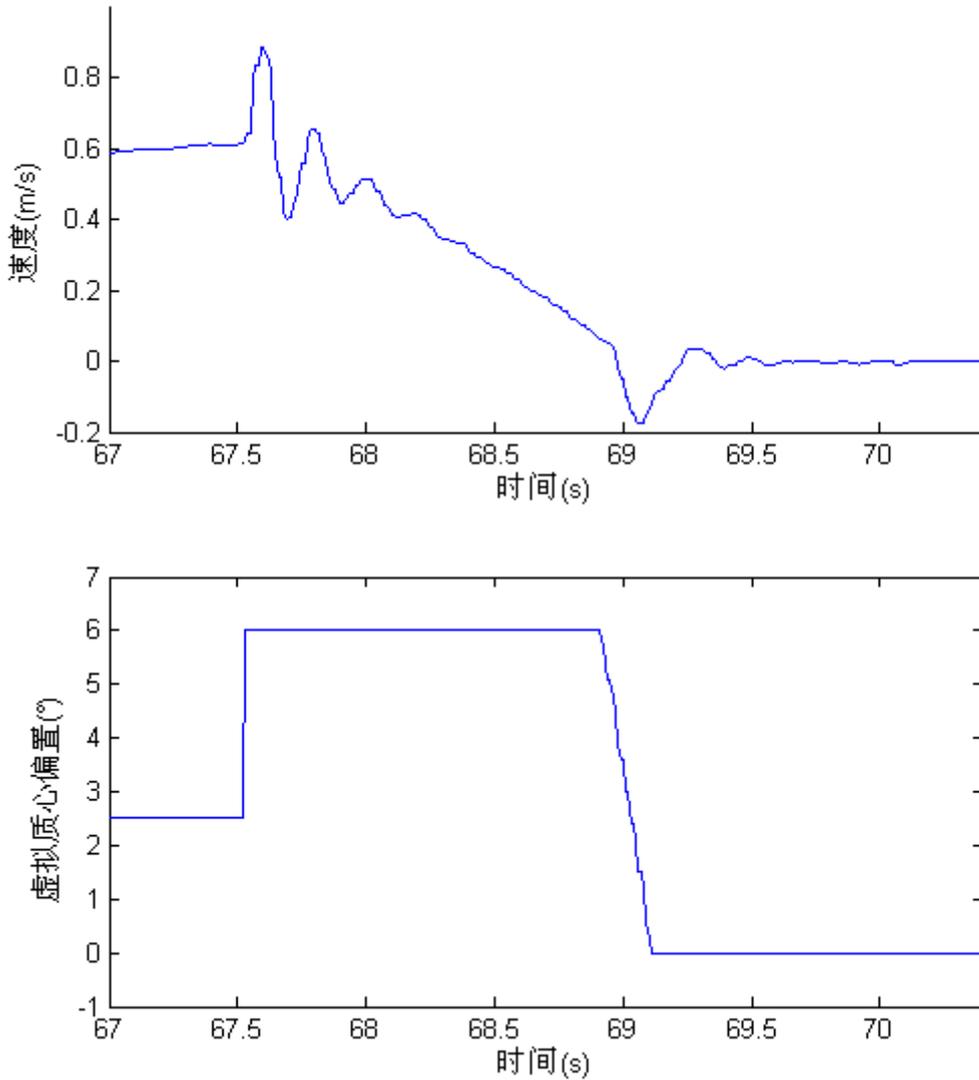


图 3.16 速度设定值为 0.5m/s 时大偏置减速过程状态

3.4.3 原地旋转运动控制

原地旋转时，机器人姿态变化如图 3.17 所示。

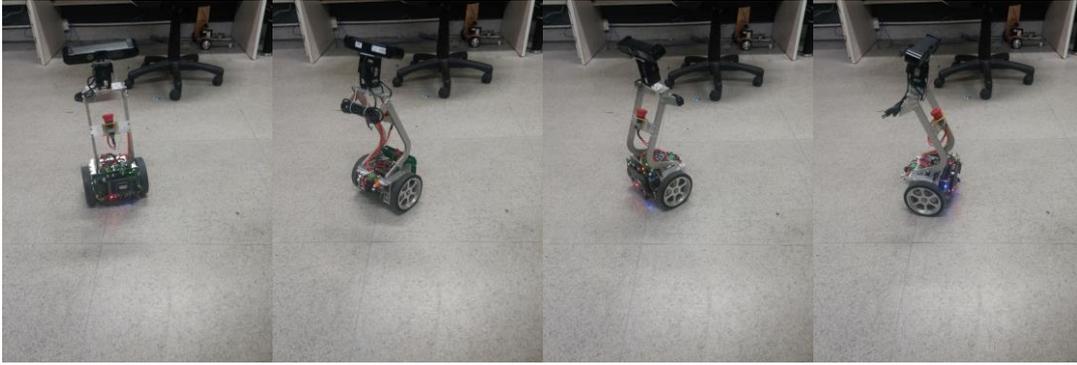


图 3.17 机器人原地旋转姿态变化

原地旋转时，机器人通过板载蓝牙上传过程数据。经 Labview 记录，Matlab 绘制后，其路程与朝向角度变化如图 3.18 所示。由图可见，机器人朝向在 -166.1° 到 $+193.9^{\circ}$ 间近似均匀变化，说明机器人一直在进行 360° 旋转。

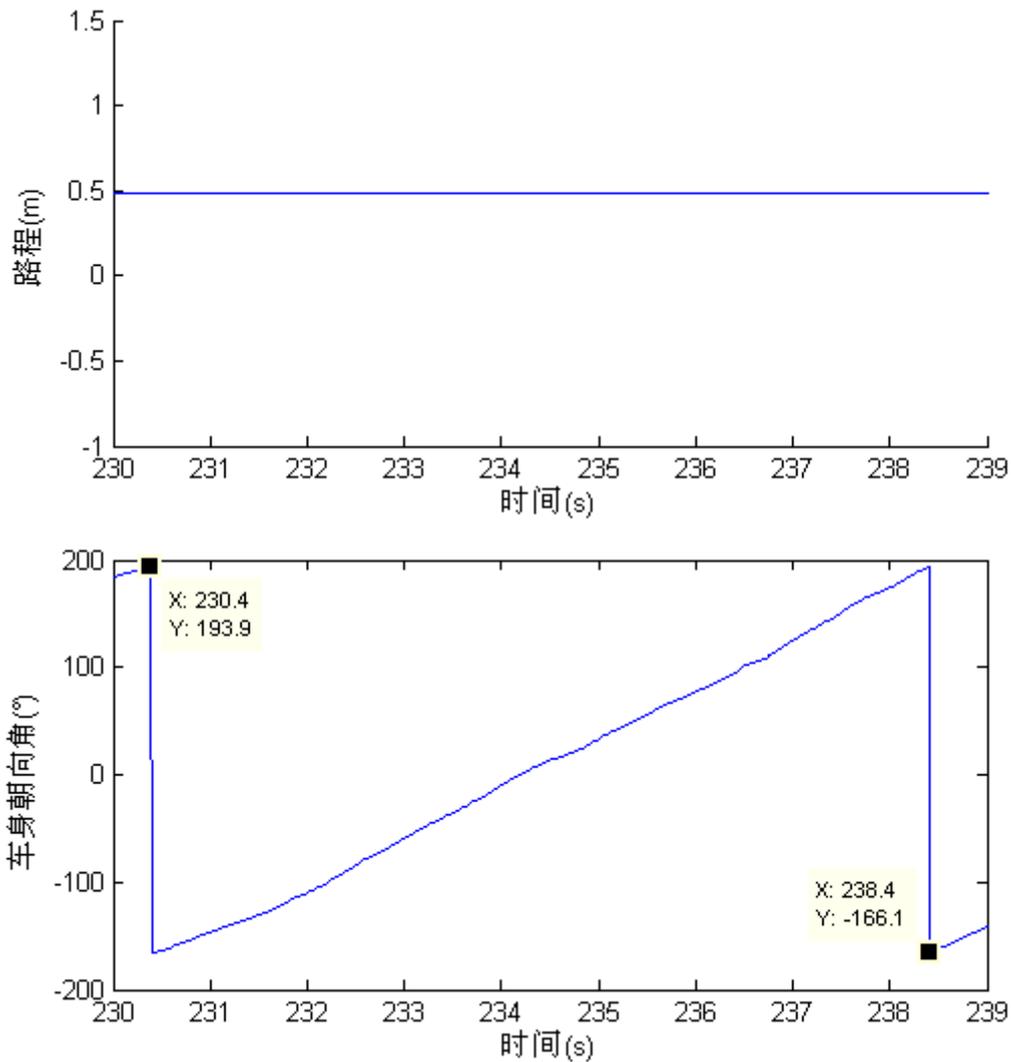


图 3.18 机器人原地旋转过程状态

而且其路程维持在 0.5m 附近保持不变，说明其左右轮速度的共模分量一直为 0，0.5m 的初值是由上一个阶段运动带来的。综上，机器人原地旋转效果良好。

3.4.3 带半径旋转运动控制

带半径旋转即机器人沿一定半径的圆弧运动，机器人姿态变化如图 3.19 所示。

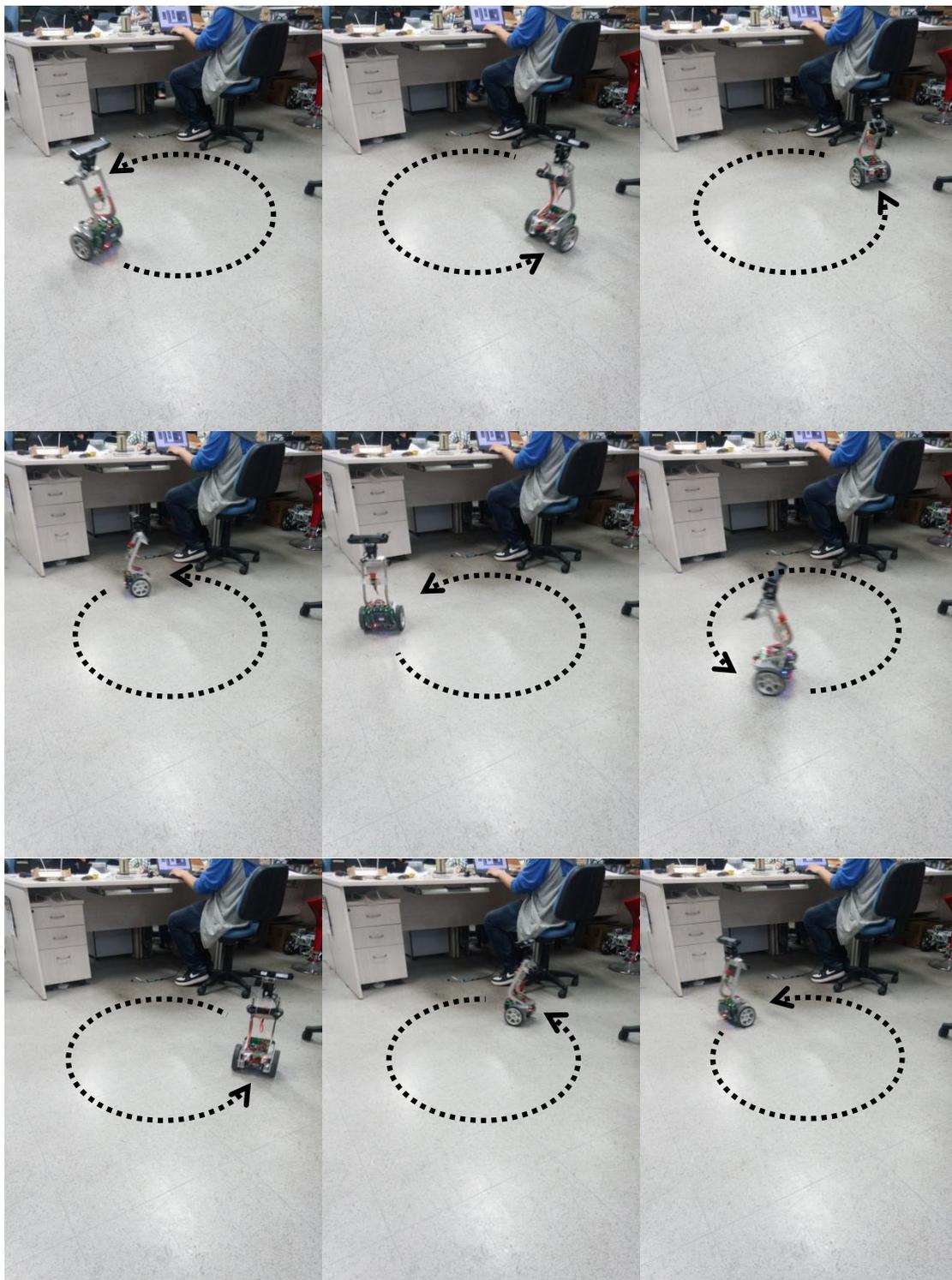


图 3.19 机器人带半径旋转姿态及位置变化

带半径旋转时，机器人路程与朝向角度变化如图 3.20 所示。由图可见，机器人在 260.1 到 -98.74 间运动，能在约 360° 内旋转，且角度随路程的增加大致均匀增加，可以得知实际机器人在沿着某个特定的圆进行旋转。其角度并非严格的 360° 与 IMU 所测得的 yaw 角存在一定的零漂有关。每次路程均由 0 变化到 0.8m 附近，程序中一次追踪最多持续 90° ，超过 90° 即将路程清零，重新开始计算追踪，故完整的 360° 需要四个追踪的周期，在图像中显示为 4 个峰。四个峰值加和即旋转一圈机器人所走过的路程，共 3.21m，对应计算出其旋转圆的半径为 0.511m，实际测量其旋转圆的半径为 0.52m 与理论计算相近，且两者与设定值 0.5m 误差均在 5% 以内。综上，机器人带半径旋转效果良好。

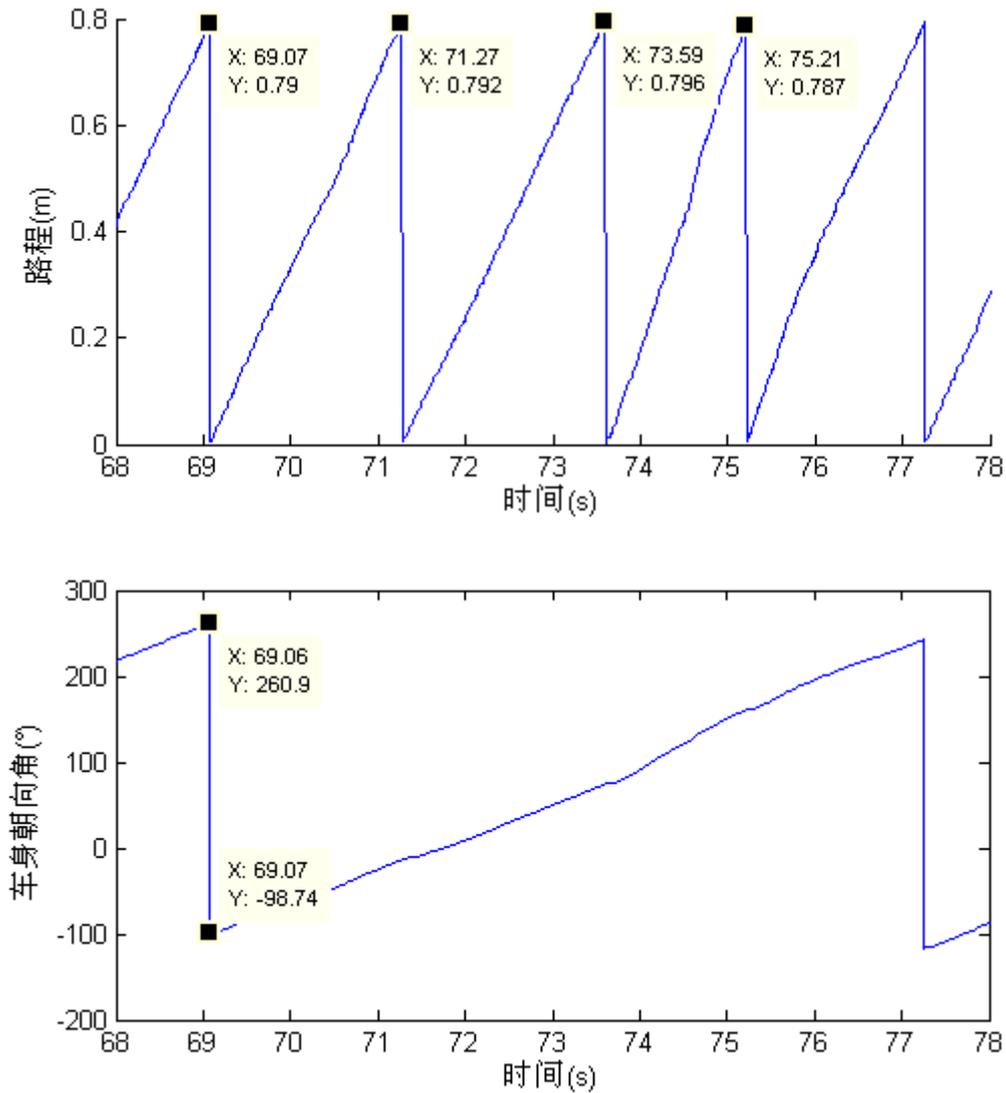


图 3.20 机器人带半径旋转过程状态

3.5 本章小结

本章利用 Lagrange 方程对两轮机器人整体进行建模，得到了两轮机器人非线性动力学模型，在小角度近似的假设下得到了线性模型。利用线性模型进行了状态反馈控制器的设计，并采用 LQR 进行反馈矩阵的整定，最终在两轮机器人上进行应用，取得了良好的控制效果。

运动控制方面，在考虑平衡约束的情况下，不直接对状态量进行闭环反馈控制，引入“虚拟质心”概念，通过速度外环“虚拟质心”的计算，实现内环平衡控制与速度控制兼顾。同时针对原地旋转与带半径旋转分别设计不同的控制算法，都取得良好的控制效果。

4 路径规划及复合运动控制

4.1 路径规划

4.1.1 路径规划的意义

移动机器人路径规划是移动机器人技术研究中的关键问题，其主要任务是让移动机器人在某种环境中运动时，寻求一条从已知起点到已知终点之间的最优路径。对于本次设计而言，地图信息和障碍物信息事先都已获得，要求两轮机器人在已知环境中实现从起点到终点的运动，并且在运动期间不触碰障碍物。这也是两轮机器人完成其他更复杂任务的基础。同时路径规划所得的路径为后续复杂运动的规划提供了依据。

4.1.2 A 星算法基本思想和方法阐述

本次设计中在电脑端用 c 语言实现了相应的路径规划模块。其路径规划采用了 A 星算法。A 星算法的基本思想和实现方法如下所述。

A 星算法结合了启发式方法和形式化方法，它通过一个估价函数来估计图中的当前点到终点的距离，并由此决定它的搜索方向，当这条路径失败时，它会尝试其它路径。

得到地图建立相应的搜索区域后，基于 A 星算法的特点，建立了 ClosedList 和 OpenList。OpenList 用来记录所有被考虑来寻找最短路径的区域，ClosedList 用来记录不会再被考虑的方块。并对每一个在计算范围内的方块给定一个 G+H 值。其中 G 是从开始点到当前方块的移动量。本次规划中不考虑路面差异，故相邻方块间的移动量均为 1。而 G 值是移动量的和，故会随着路径增长而不断增加。H 是当前方块到目标点的移动量估算值，该值仅为到目标点的估计值。H 这个估计量的准确性将影响最终路径的精确程度。但为了简单起见，本次设计中 H 量的估计使用“曼哈顿距离方法”，即仅忽视过程中的障碍物，仅计算到终点的水平和垂直距离。

在路径规划的过程中，将不断将搜索区域添加到 OpenList 和 ClosedList 中，并计算其 G+H 值来决策下一段路径的规划。其规划过程如图 4.1 所示。

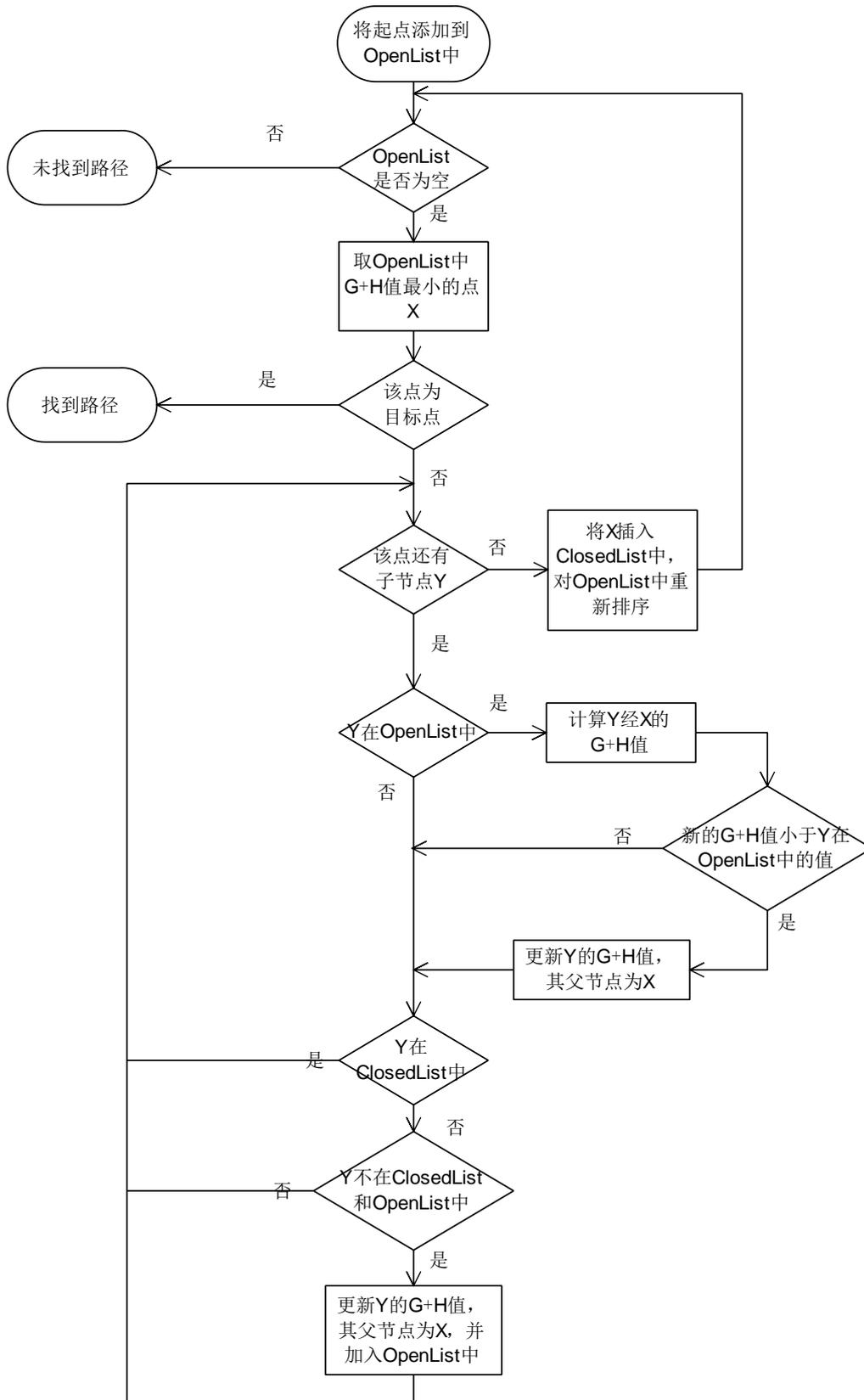


图 4.1A 星算法计算流程

4.2 基于航点的复杂运动控制

4.2.1 基于航点的复杂运动控制方法

比较常见的运动控制是在路径规划完成后，分析规划所得路径，计算出路径中每个点的运动状态，做速度规划并下发，底层收到相应的控制命令后实现速度控制，以完成最终的路径。但在两轮机器人上，由于整体速度控制受限于平衡约束，无法很好的保证速度控制命令的实时响应，其最终速度控制精度也无法达到要求，且运动过程的加速度限制复杂，故无法通过速度控制实现最终的路径。

考虑到上述完成的直行、旋转等单元动作，在本次设计中，考虑利用简单路径组合形成复杂路径以实现所规划路径。故利用 A 星算法完成初步路径规划，再将路径切分，得到关键航点，利用直线连接关键航点，即可将整条路径切分为若干直线段，仅将直线的控制指令下发，仅做运动距离控制。在底层机器人系统中，利用自身里程信息反馈，在未达到目标距离前完成直行运动，运动到点后停止，并原地旋转至下一直线朝向，并继续执行直行运动。重复若干次后即可实现所规划复杂路径。但该方法完成的路径为一组折线，路径不光滑。同时在速度控制精度受限的情况下无法保证其刹车距离。故对该方法做出优化，利用光滑圆弧连接两直线段，在光滑圆弧处机器人可利用带半径旋转实现。

该方法将速度规划下放到底层实现，避免速度控制精度受限引起的轨迹无法跟踪。利用光滑圆弧连接直线段后，理论上在到达目标点前都不需要进行刹车，大大提高了机器人实现轨迹的精度。综上，在路径规划完成后，将各个航点间路径拆分为基本的前进、停止和带半径旋转命令下发，机器人根据收到的命令顺序依次执行，即可完成整条路径的跟踪。

但在实际实现中发现，运动控制中引入了“虚拟质心”概念，在每个小运动完成后，“虚拟质心”可能还残留一定的量，并不归零，故下段小运动开始时，“虚拟质心”的初值不应从零开始，而应根据计算量和上段运动的残留量进行平滑滤波，以防运动状态切换时，因“虚拟质心”的突变造成机器人运动的“顿卡”现象。此处的平滑滤波采用加权平均的方法进行。

4.2.2 基于航点的运动控制效果

软件界面上，利用 A 星算法进行路径规划，可得如下效果：

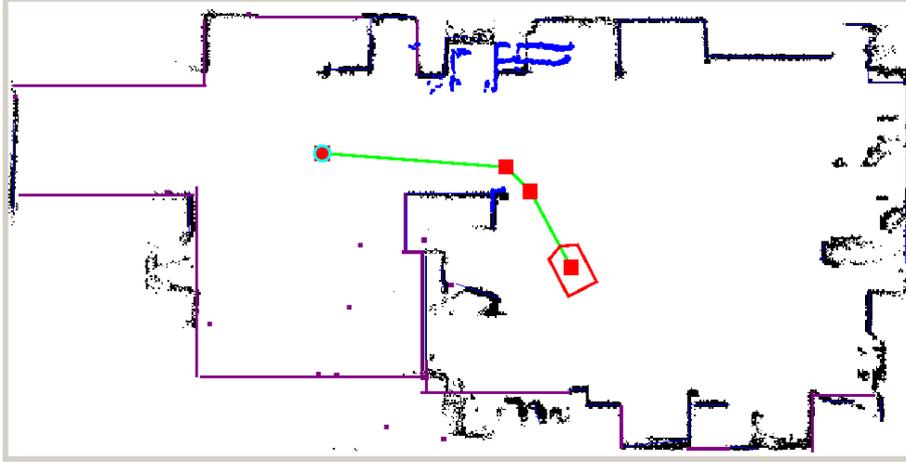


图 4.2 A 星算法路径规划结果

图中地图为实验室地图，增加额外的障碍物信息后，可得路径规划如下图所示。通过 A 星算法所规划的路径避开了障碍物。

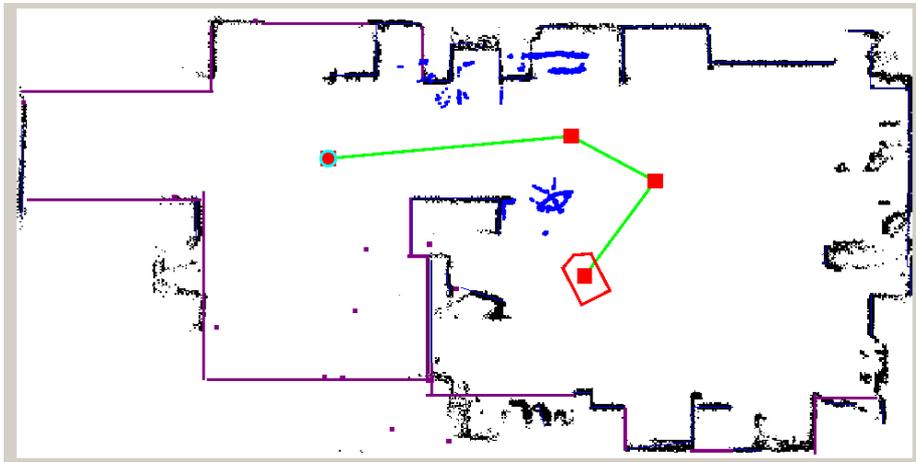


图 4.3 加障碍物后 A 星算法路径规划结果

图中所示的红色方块即为求得的关键航点，最终在基于航点复杂运动控制后，在无障碍物的地图中进行规划，实际运动效果如下图所示。实际运动所经过的路线与规划的基本一致，基于航点的复杂运动控制方法有效。



图 4.4 无障碍物实际运动效果

4.3 基于光滑连续轨迹的复杂运动控制

4.3.1 光滑连续轨迹复杂运动控制方法

上述运动控制的实现是在路径规划完成后，将整条路径规划完成后拆分为一个个小的简单运动从而实现整体的运动控制。而考虑到两轮机器人的直行和旋转是解耦的，即两轮驱动的共模分量影响直行的速度和平衡控制，两轮驱动的差模分量影响机器人的旋转。则通过旋转来完成路径将减小平衡约束对路径实现的干扰。

由于连续光滑路径处处均可计算其路径长度，同时处处均可计算出其切线方向，即获得其点的朝向，连续光滑路径可以通过计算抽象出其实时点的朝向关于路径长度的函数。而机器人能获得较为准确的里程信息，包含其自身移动路程以及朝向角度信息。故考虑在获得复杂路径后，分析路径可得朝向关于其路径长度的函数

关系,将该函数关系下发,让机器人获得在运动过程中实时追踪的目标角度的大小,从而实现整条光滑连续轨迹。

该方法避免了在复杂运动控制中对路径长度的直接控制,而将其转化为实时运动过程中朝向角关于目标角度的追踪,而目标角度的追踪是通过与直行解耦的旋转实现,大大降低平衡约束对复杂运动控制的影响。

实际实现则应将已知的轨迹规划为一长段直行运动,在直行的过程中不断改变机器人目标角度,并通过直行中的旋转实现对目标角度的追踪,从而完整连续地实现光滑轨迹的跟踪。

4.3.2 手绘轨迹跟踪控制

为验证上述轨迹跟踪思路的可行性,特在手机端编写控制程序。利用程序画布记录手绘轨迹。程序将手绘轨迹解析为一段段直线段的连接,并计算出每个转折点的轨迹长度与朝向,即得出朝向角关于路径长度的函数。接收机器人在运动过程中上传的里程计信息,根据机器人的运动区间计算机器人理想的朝向角。并将朝向角下发。机器人在直行的过程中不断调整自身朝向,最终完成整条平滑轨迹的跟踪。

4.3.3 手绘轨迹跟踪效果

利用手机端软件进行手绘轨迹绘制,其手绘结果如图 4.5 所示。在对应的 java 软件设计中,设计了计算按钮,通过计算按钮可分析当前轨迹,计算出机器人轨迹朝向角关于轨迹长度的函数关系,部分结果如图中所示,并下发给机器人。机器人通过该函数关系不断追踪朝向角实现期望轨迹。其轨迹完成效果如图 4.6 所示。机器人所实现的轨迹与手绘轨迹相似且光滑,故该种基于光滑连续轨迹的复杂运动控制方法有效。

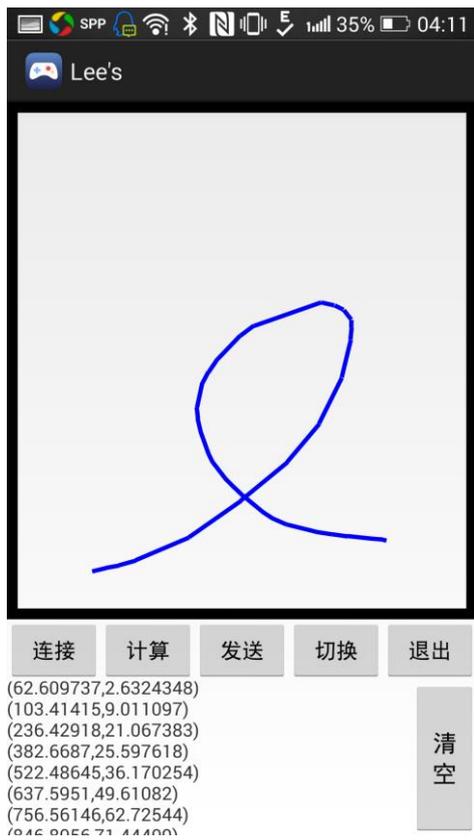


图 4.5 手绘轨迹结果

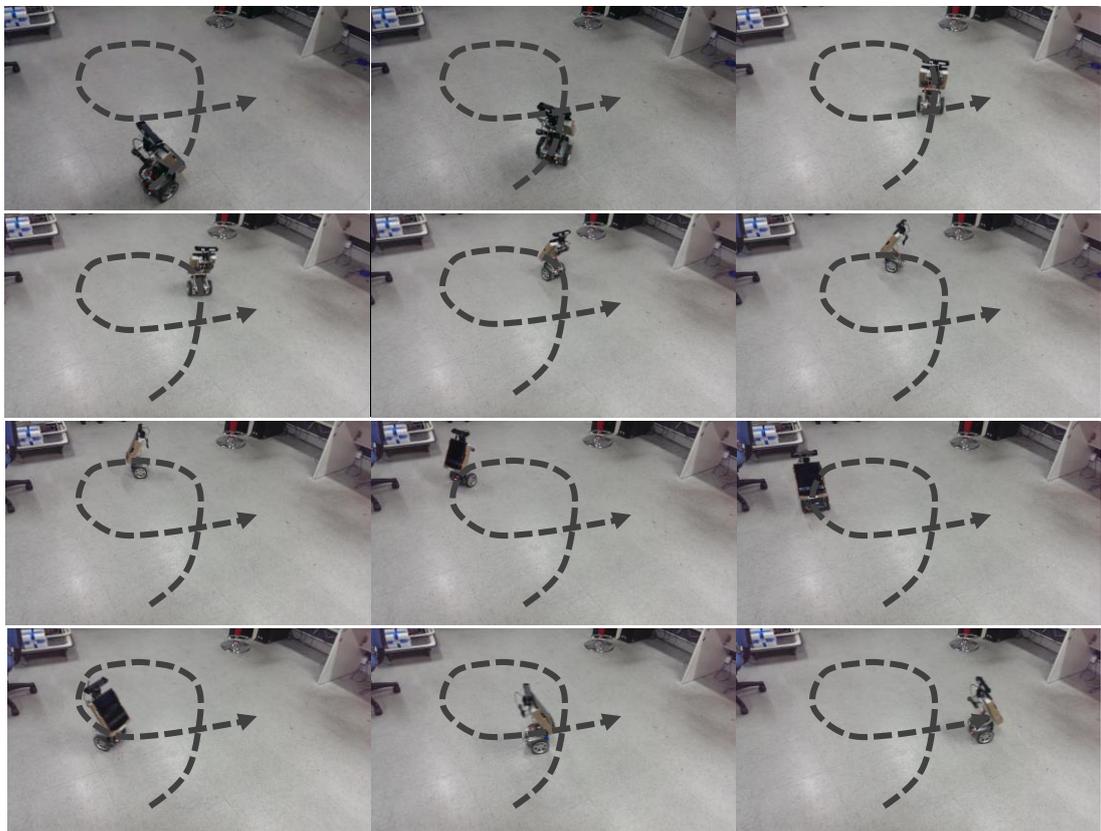


图 4.6 手绘轨迹实现结果

4.4 本章小结

本章主要阐述了路径规划所用到的 A 星算法的规划思路和计算流程，完成了基于导航点的复杂运动控制。并在独立运动跟踪轨迹的基础上提出了在直行过程中追踪目标角度的轨迹跟踪方法，能很好的完成整条平滑轨迹的跟踪。

5 机器人质心自适应功能设计

本设计中定义质心的投影落在机器人支撑面上时，IMU 测量的车身偏角大小为零位偏角。机器人在控制过程中，其车身偏角是由当前 IMU 所测量的角度减去预设零位偏角得到的。机器人系统在运行过程中，可能由于临时悬挂物体、遇到稍倾斜路面、IMU 测量数据漂移等问题引起程序中预设的零位偏角与实际零位偏角不一致。这种情况的出现将引起机器人向一个方向倾斜并不断加速运动。因而需要实现在原定直立情况下的质心自适应功能设计。本设计中考虑采用卡尔曼滤波计算实际控制程序所需的零位偏角。但卡尔曼滤波的设计需要机器人的相关物理参数，由于机器人结构复杂，物理参数不能直接测量得到，需要对机器人系统进行辨识。

5.1 机器人系统参数辨识

5.1.1 参数辨识方法设计

机器人系统中，共有三个物理参数有待测算，分别是转动惯量 J ，整车质量 m 与质心偏角 θ 。考虑转动惯量 J 与整车质量 m 存在相关关系，若这两者同时在卡尔曼滤波中进行估计还需分析器相关关系，为降低卡尔曼滤波器的设计难度，在此对转动惯量 J 进行参数辨识，后续利用卡尔曼滤波估计整车质量。

考虑能量守恒：

$$\frac{1}{2}J\omega^2 \Big|_{\theta_0}^{\theta} = \int_{\theta_0}^{\theta} M * d\theta \quad (5.1)$$

式中 J 为转动惯量， ω 为转动角速度， θ 为转动角， M 为合外力矩。

在实验中电机力矩可根据电机模型推算，而重力矩的测量涉及实际的零位偏角，故在转动惯量的测算中需排除重力矩的干扰，应将机器人横卧，仅靠电机力矩驱动旋转。则在多次数据测量的情况下，令

$$A = \begin{bmatrix} \frac{1}{2}\omega^2 \Big|_{\theta_0}^{\theta_1} \\ \frac{1}{2}\omega^2 \Big|_{\theta_1}^{\theta_2} \\ \vdots \\ \frac{1}{2}\omega^2 \Big|_{\theta_{n-1}}^{\theta_n} \end{bmatrix}, \quad B = \begin{bmatrix} \int_{\theta_0}^{\theta_1} M * d\theta \\ \int_{\theta_1}^{\theta_2} M * d\theta \\ \vdots \\ \int_{\theta_{n-1}}^{\theta_n} M * d\theta \end{bmatrix}$$

代入式(5.1)可得

$$A * J = B \quad (5.2)$$

设 J 的最小二乘估计为 \hat{j} ，则其最小二乘误差满足

$$E = (B - A * \hat{j})' * (B - A * \hat{j}) = 0 \quad (5.3)$$

因为 \hat{j} 为一个数，故上述式有唯一解，解为

$$B = A * \hat{j} \quad (5.4)$$

由于 A 矩阵列满秩，故可对应求取伪逆，则对应 J 的最小二乘估计为

$$\hat{j} = \text{pinv}(A) * B = (A' * A)^{-1} * A' * B \quad (5.5)$$

代入测量所得数据即可确定转动惯量的估计值。

5.1.2 参数辨识实验结果

应用上述推导结果实际实验，将机器人侧卧并固定其一侧轮子，程序设计中仅驱动被固定侧轮子，在轮子被固定的情况下，使电机产生的驱动力完全作用于轮子，排除重力做功的影响。同时保证足够大，以减小运动过程中摩擦引起的相对误差。

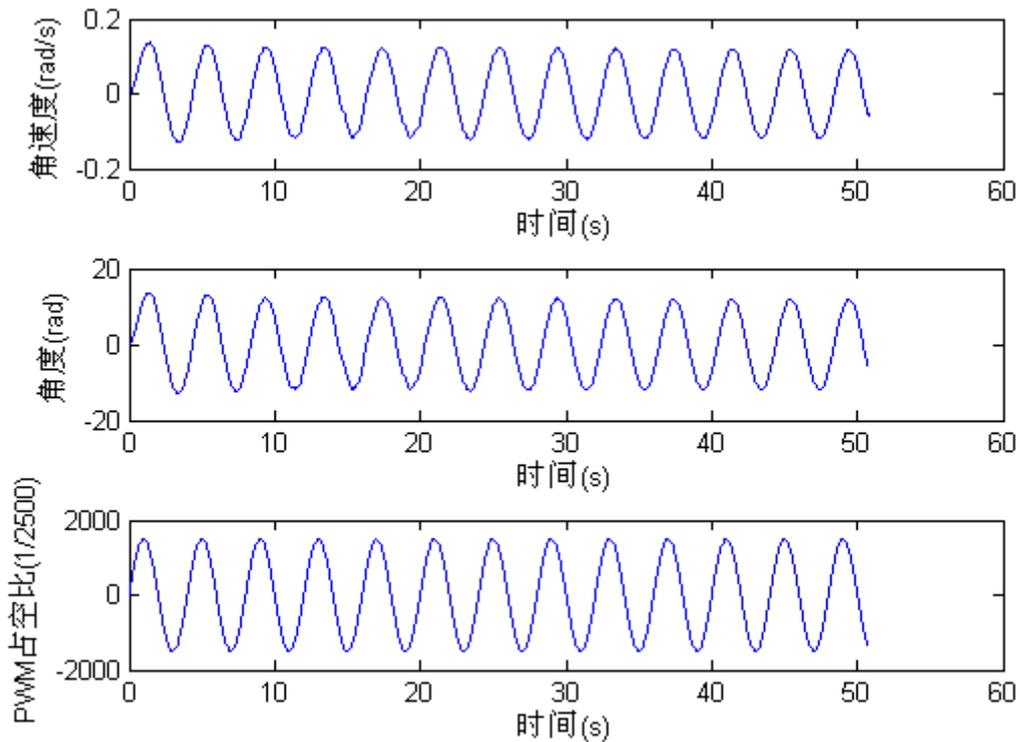


图 5.1 机器人过程参数

分析上述推导过程，要获得 A 、 B 矩阵，需获得机器人实时的速度、角速度和 PWM 波占空比。故编程对上述三个机器人参数进行采集，得到的实验数据结果如下图所示。为了使实验结果更准确全面，本次辨识输入激励采用的是正弦激励，故得到的各个状态量的响应曲线都近似于正弦波波形。

借用电机模型计算其输出力矩 M ，同时利用上述获得的过程参数，可以计算出 A 、 B 随时间变化的关系如下图所示。可以看出，输入为正弦激励时，计算所获得的 A 、 B 随时间变化曲线也有正弦曲线的特征。同时由于数据采集的过程中，角速度作为角度的一阶微分量，其测量误差应比角度大的多，实验结果中 A 的计算与角速度相关，而 B 的计算与角度相关，其中 A 值的噪声要显著大于 B 值的噪声。

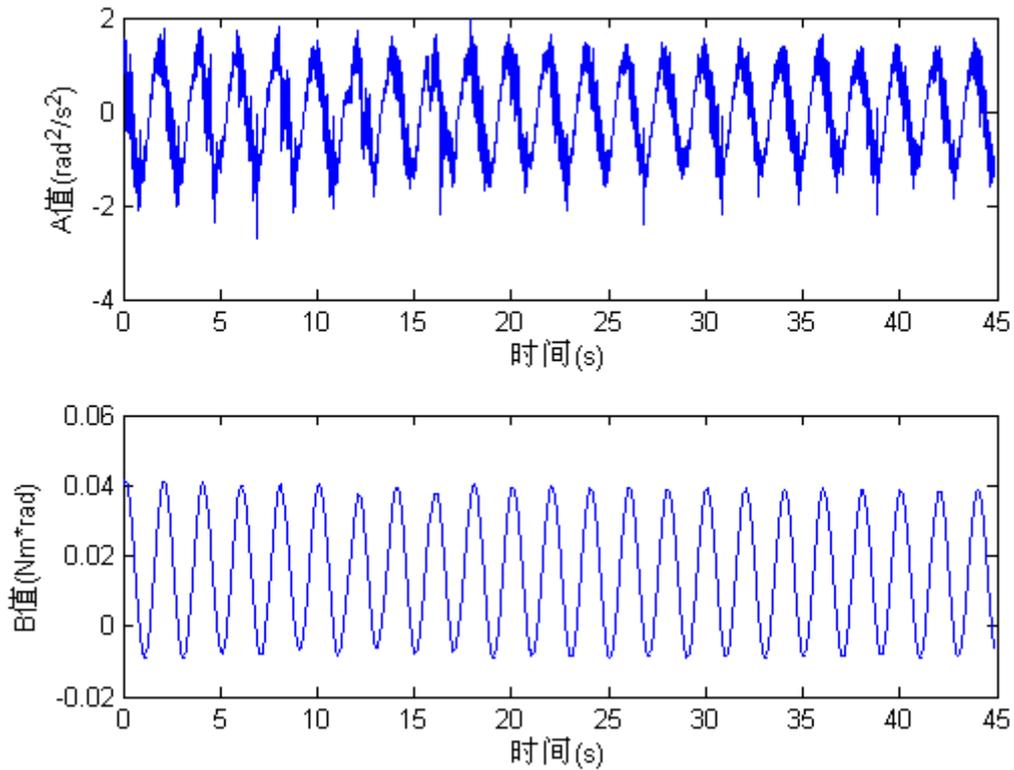


图 5.2 A 、 B 计算结果

在获得 A 、 B 两值的基础上，可以通过 A 、 B 与转动惯量 J 的关系同一时刻相除得到 J 随时间变化的图，其关系如图 5.3 所示。从图中可以看出， A 、 B 直接计算所得的转动惯量 J 存在很大的噪声。通过上述式子所推导的办法，利用 A 、 B 形成的矩阵整体计算转动惯量 J 的最小二乘估计，其估计值为 0.0139kgm^2 。将其中较明显的噪声去除后再进行估计，得到其估计值为 0.0137kgm^2 。可以看到，利用最

小二乘估计能有效的减少噪声对估计结果的影响。利用 Solidworks 可从侧面验证本次估计的结果。在 Solidworks 中对本设计中的机器人进行质量属性分析,可以得到其惯性张量矩阵,其中绕 Y 轴旋转的转动惯量为 14684692.10gmm^2 ,即 0.0149kgm^2 ,虽然该值由于模型与实际有出入等问题可能并不是完全正确,但该值与本次估计所得的转动惯量的值接近。因此,利用最小二乘对机器人的转动惯量进行估计是可行有效的。

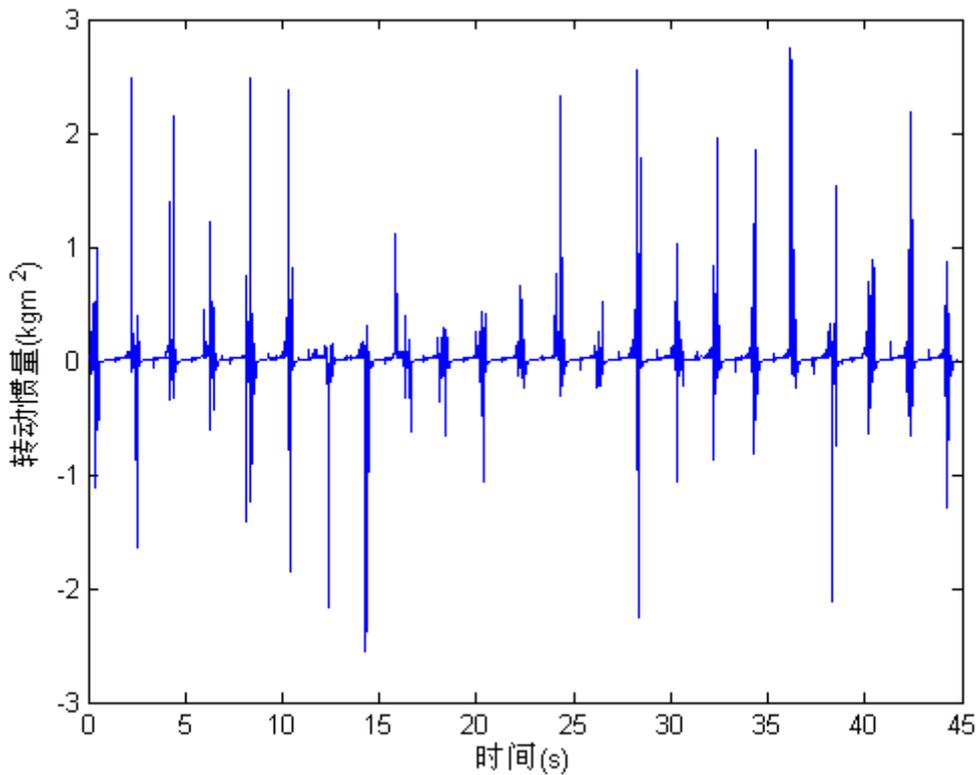


图 5.3 转动惯量计算结果

5.2 机器人自适应算法设计

5.2.1 自适应算法设计综述

机器人由于种种原因会引起零位偏角的改变,导致车身偏角计算偏大或偏小,似的控制器的运算结果不能得到最佳的控制效果,使其往零位偏角漂移的方向倾斜。因此如果能获得实际的零位偏角值,修正车身偏角的计算,就能实现机器人质心自适应。

故本次设计中利用卡尔曼滤波器，根据预设的零位偏角与观测所得的零位偏角融合滤波，得到实际零位偏角值的大小，从而实现质心自适应。

5.2.2 卡尔曼滤波器设计

首先建立系统的观测方程和状态方程。考虑能量守恒，当机器人直立时

$$\frac{1}{2}J\omega^2 \Big|_{\theta_0}^{\theta} = \int_{\theta_0}^{\theta} (M - mgl \sin(\theta - \theta_0)) * d\theta \quad (5.6)$$

式中 J 为转动惯量， ω 为转动角速度， θ 为转动角， M 为电机力矩， m 为机器人质量， l 为质心到轮轴的距离， θ_0 为待求的零位偏角。当机器人在小角度内保持平衡时，可线性化近似获得

$$\frac{1}{2}J\omega^2 \Big|_{\theta_0}^{\theta} = \int_{\theta_0}^{\theta} (M - mgl(\theta - \theta_0)) * d\theta \quad (5.7)$$

将式(5.7)右侧展开整理可得

$$\frac{1}{2}J\omega^2 \Big|_{\theta_0}^{\theta} - \int_{\theta_0}^{\theta} M * d\theta = -mgl \int_{\theta_0}^{\theta} \theta * d\theta + mgl\theta_0 \int_{\theta_0}^{\theta} d\theta \quad (5.8)$$

则不妨设滤波的状态量为 mgl 与 $mgl\theta_0$ ，令

$$\begin{cases} K_1 = \frac{1}{2}J\omega^2 \Big|_{\theta_0}^{\theta} - \int_{\theta_0}^{\theta} M * d\theta \\ K_2 = - \int_{\theta_0}^{\theta} \theta * d\theta \\ K_3 = - \int_{\theta_0}^{\theta} d\theta \end{cases}$$

代入式(5.8)得观测方程，观测噪声为 V

$$K_1 = [K_2 \quad -K_3] \begin{bmatrix} mgl \\ mgl\theta_0 \end{bmatrix} + V \quad (5.9)$$

假设零位偏角不变，则得到状态方程，过程噪声为 W

$$\begin{bmatrix} \dot{mgl} \\ \dot{mgl\theta_0} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} mgl \\ mgl\theta_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} W \quad (5.10)$$

将上述观测方程与状态方程离散得：

$$\begin{cases} K_1(k) = [K_2 \quad -K_3]X(k) + V \\ X(k-1) = X(k) \end{cases} \quad (5.11)$$

要估计 k 时刻的两个状态量，就必须根据 $k-1$ 时刻的两个状态量，在根据预测得到的 k 时刻的两个状态量运算得出 k 时刻的高斯噪声的方差，并在此基础上进行递归运算。故运算过程中需要提前获得过程噪声协方差矩阵 Q 和观测噪声协方差矩阵 R ，两者形式如下：

$$\begin{cases} Q = \begin{bmatrix} q_{mgl} & 0 \\ 0 & q_{mgl\theta} \end{bmatrix} \\ R = r_{mgl} \end{cases}$$

式中 q_{mgl} 和 $q_{mgl\theta}$ 分别为对量估计量的协方差，表示对这两个数据的置信程度，越小，置信程度越高。在本次设计中，对于角度值存在较大的噪声，因此 q_{mgl} 取小值， $q_{mgl\theta}$ 取大值。

对于当前状态，根据系统状态方程与上一状态值可以做出预测：

$$X(k|k-1) = AX(k-1|k-1) + BU(k) \quad (5.12)$$

式中 $A=I_{2 \times 2}$, $B=[0 \ 0]^T$ ，相应可计算出上述预测对应的协方差为：

$$P(k|k-1) = AP(k-1|k-1)A' + Q \quad (5.13)$$

式中 Q 即上述提及的协方差矩阵， $P(k-1|k-1)$ 为上一状态对应协方差矩阵。当前时刻的估计值为：

$$X(k|k) = X(k|k-1) + K(k)(Z(k) - HX(k|k-1)) \quad (5.14)$$

式中 $Z(k) = K_1(k)$, $H = [K_2 \ -K_3]$, $K(k)$ 为卡尔曼增益，表述如下：

$$K(k) = P(k|k-1)H' / (HP(k|k-1)H' + R) \quad (5.15)$$

式中 R 即测量噪声协方差矩阵，至此已获得 k 状态下的状态量滤波值，但为了使卡尔曼滤波不断循环下去，还需对 $X(k|k)$ 对应的协方差矩阵进行更新，即：

$$P(k|k) = (I - K(k)H)P(k|k-1) \quad (5.16)$$

至此，利用式(5.12)~(5.16)即可循环进行卡尔曼滤波运算，同样可以通过编程实现。

在循环滤波过程中，将两个状态量相除即可得所求零位偏角。

5.2.3 卡尔曼滤波效果

根据上述式子，要进行卡尔曼滤波，需实时求取观测方程中的 K_1 , K_2 和 K_3 ，这几个变量可根据参数上传获得，需要车身偏角，车身偏角角速度以及电机提供的力矩。电机提供的力矩可通过电机模型得到。计算 K_1 所需的转动惯量 J 使用辨识的估计值，取转动惯量 $J=0.0135\text{kgm}^2$ 。

为获得实验所需要的数据，在遥控的情况下，让车在平面上保持平衡并做前后摇摆运动，得到相关参数如图 5.4 所示。通过上述推导，借助 Matlab 编程离线计算得 Kalman 状态量计算结果如图 5.5 所示。最终将两状态量相除得零位偏角的滤波值，其随时间变化情况如图 5.6 所示。从中可以看出，经过一小段时间的滤波后，

其零位偏角的滤波值最终稳定在实际偏角 -2.2° 附近，说明卡尔曼滤波在零位偏角的估计上有一定的效果。

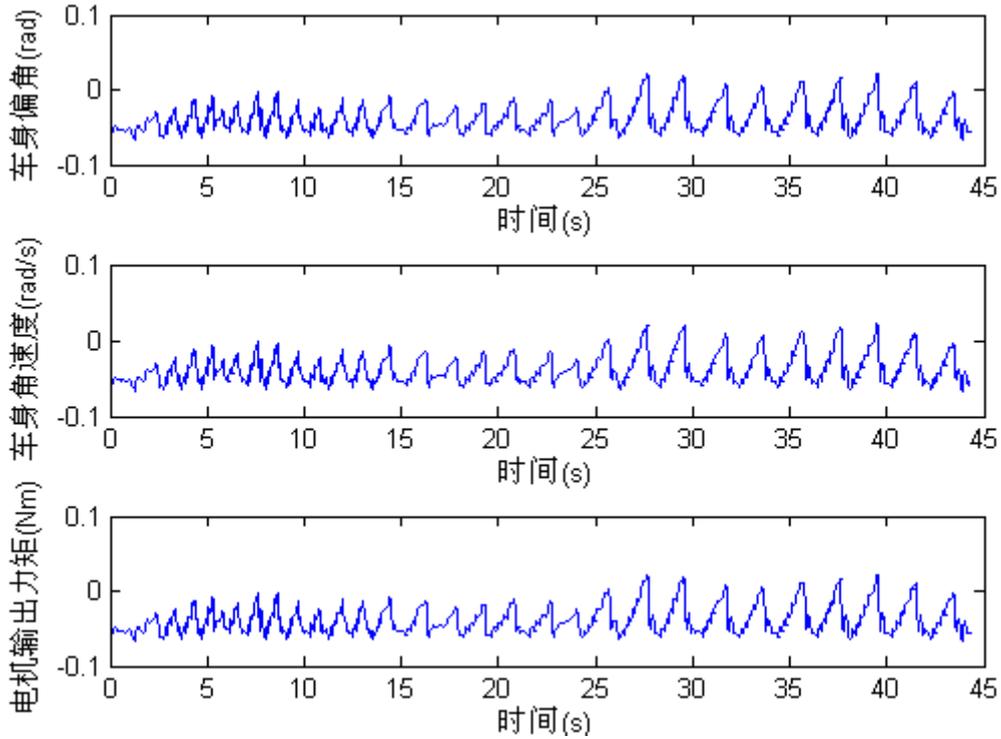


图 5.4 机器人状态信息

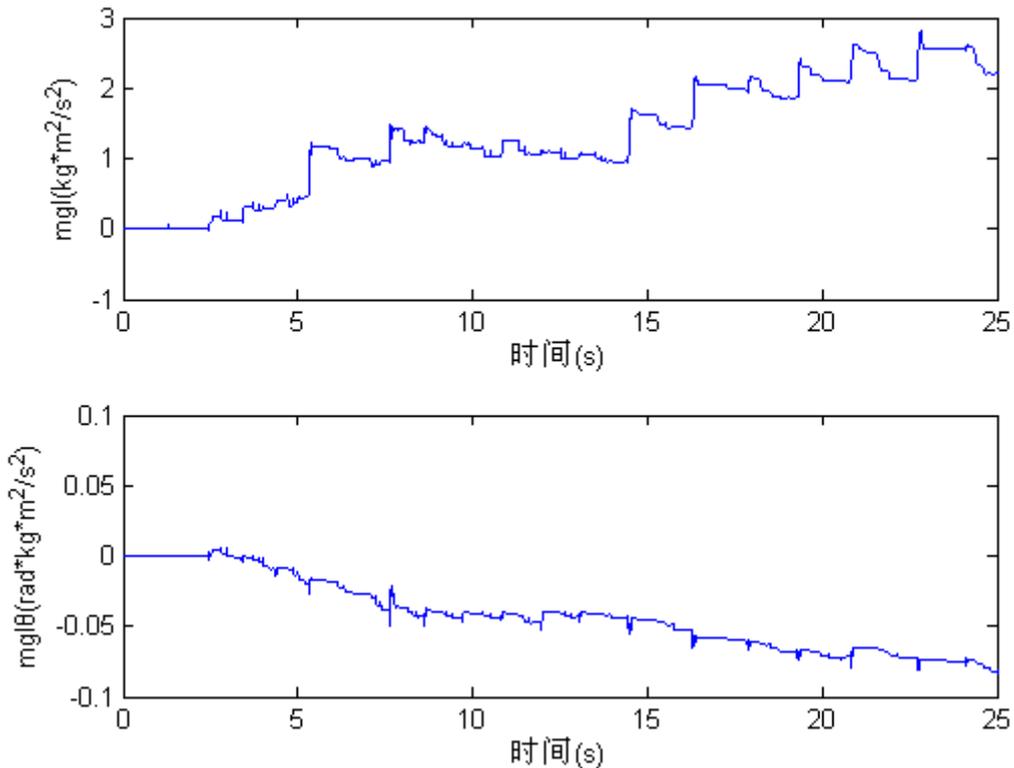


图 5.4 卡尔曼滤波过程状态量

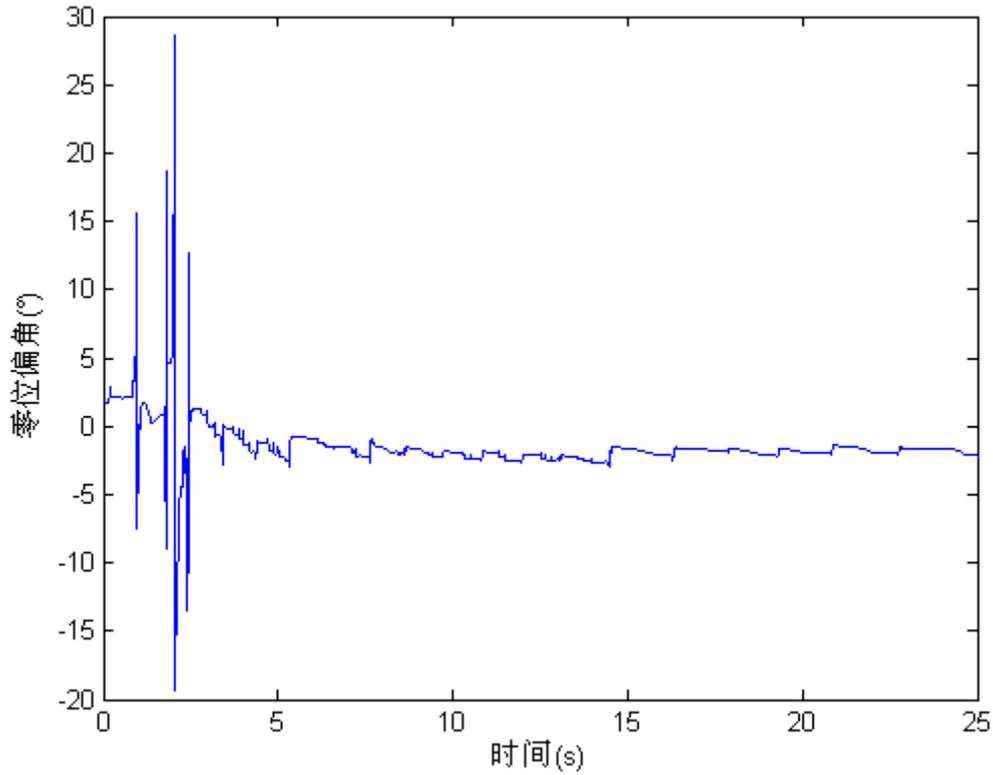


图 5.5 卡尔曼滤波结果

考虑提高滤波效果，事先赋给状态量初值，经测量，车身重量 $m=3.42\text{kg}$ ，质心距转轴约 $l=0.06\text{m}$ ，零位偏角 $\theta=-2.2/180*3.14\text{ rad}$ ，即

$$x(0) = [mgl \quad mgl\theta] = [0.513 \quad -0.0197]$$

代入得零位偏角卡尔曼滤波结果如图 5.6 所示。可以看出，给定估计初值后，卡尔曼滤波收敛速度大大提高，并且滤波开始阶段不会出现大幅度振荡。观察滤波效果，其虽然在实际零位偏角 -2.2° 附近振荡，但其振荡范围有将近 $\pm 1^\circ$ ，对于平衡控制器而言，这种滤波的效果还不是很适用。甚至可能由于零位偏角的振荡引起两轮机器人的振荡，导致失去平衡。

尝试给定错误估计初值，模拟实际零位偏角漂移情况，测试卡尔曼滤波效果，将零位偏角的初值从 -2.2 修改为 -10 ，测试结果如图 5.7 所示。从测试结果可以看出，在初值错误的情况下卡尔曼滤波经过大约 5s 左右的计算，依旧能收敛到正确的实际零位偏角附近。但其运算结果仍存在比较大的振荡。

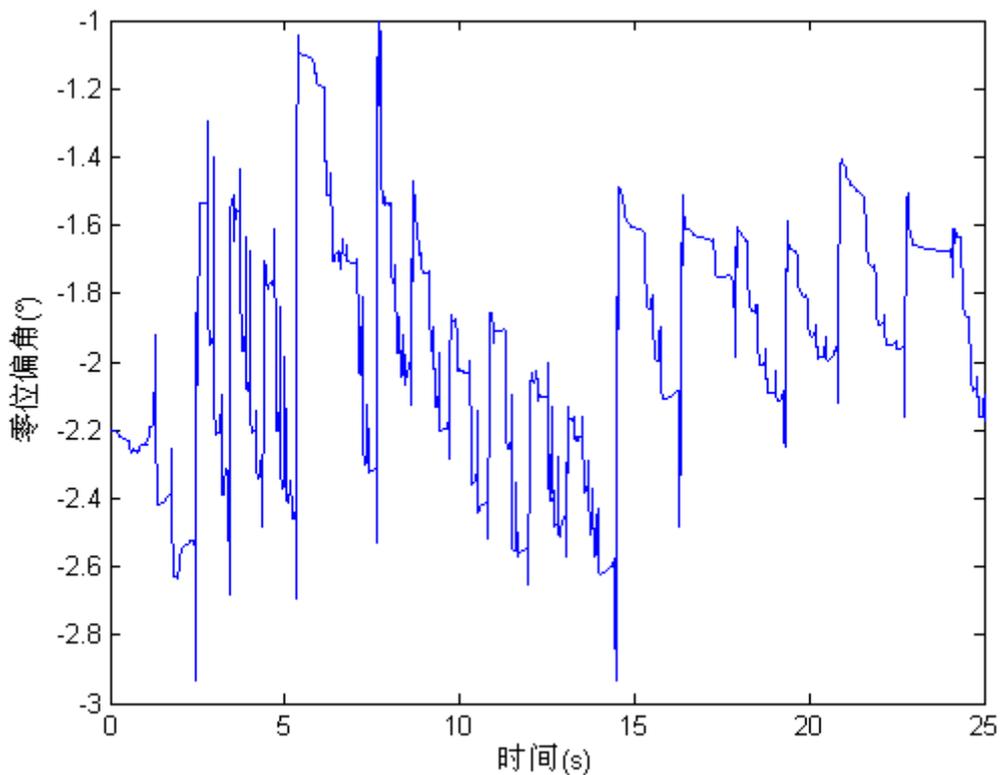


图 5.6 设置估计初值的卡尔曼滤波结果

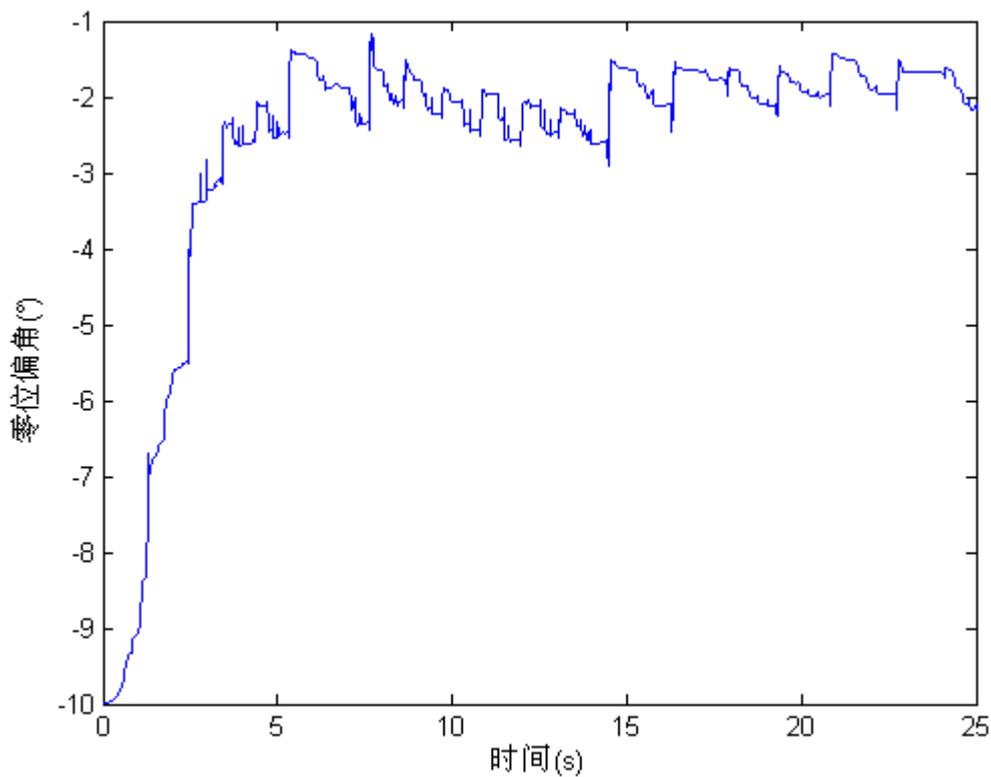


图 5.6 设置错误估计初值的卡尔曼滤波结果

综上所述,卡尔曼滤波能在较短的时间内修正较大的零位偏角误差,但其最终收敛结果存在振荡,其幅度还不能适应平衡控制器的控制要求。鉴于此种情况,还需要更多的实验去研究其振荡的原因和减小收敛振荡的方法。故暂不在控制器中设计卡尔曼滤波器。而考虑到其小幅振荡的情况,可能需要另一种能在小幅度能准确修正质心的算法辅助卡尔曼滤波,进行整体质心自适应算法的设计。

5.2.4 辅助自适应算法设计

在实时卡尔曼滤波实现前,由于无法时刻准确获得两轮机器人零位偏角,两轮机器人的平衡控制依然有质心自适应的需要,同时卡尔曼滤波在小范围内存在振荡,故要求实现一种辅助自适应算法,使得机器人在一定小范围内适应质心的改变。

在机器人零位偏角偏离初始设定值时,机器人将向零位偏角漂移的方向倾斜,在角度状态反馈平衡控制的情况下,机器人并不会产生很大的车身倾角,在调整车身倾角的过程中,会产生向零位漂移方向的移动。故可以考虑利用机器人在这种状态下的移动距离,进行反馈线性调整机器人的初始零位偏角。

在设计之初,为保证在平衡受到干扰时,能快速稳定的停下来,在平衡控制中加入了路程里程信息的反馈,设置了一个相当大的反馈系数。即机器人自身计算其运动路程,当运动路程增大时,通过反馈,控制器输出增大,使机器人姿态朝反方向倾斜,使机器人能及时减速并停止。但对于运动路程的大反馈易导致在某些情况下的振荡,故在机器人运动速度较小,即认为干扰所产生的运动已得到控制的情况下,逐渐衰减所积累的路程值。其机器人在该种控制下,在有零位偏角漂移时产生的运动过程如下图所示。

从图中可以看出,当有零位偏角漂移时,其里程数据不断积累,积累到对控制器的反馈足够大时,车身偏角会产生一个剧烈的减小,即车身剧烈后倾,迫使运动停止,运动速度降低。而后调整回来,开始衰减里程数据。但由于零位偏角漂移的存在,其又会开始前进,里程数据积累,不断产生上述循环。整体来看,机器人处于一种一抖一抖地前进的效果,不能在一定范围内保持相对静止的稳定平衡,且其预设的零位偏角数值保持不变。

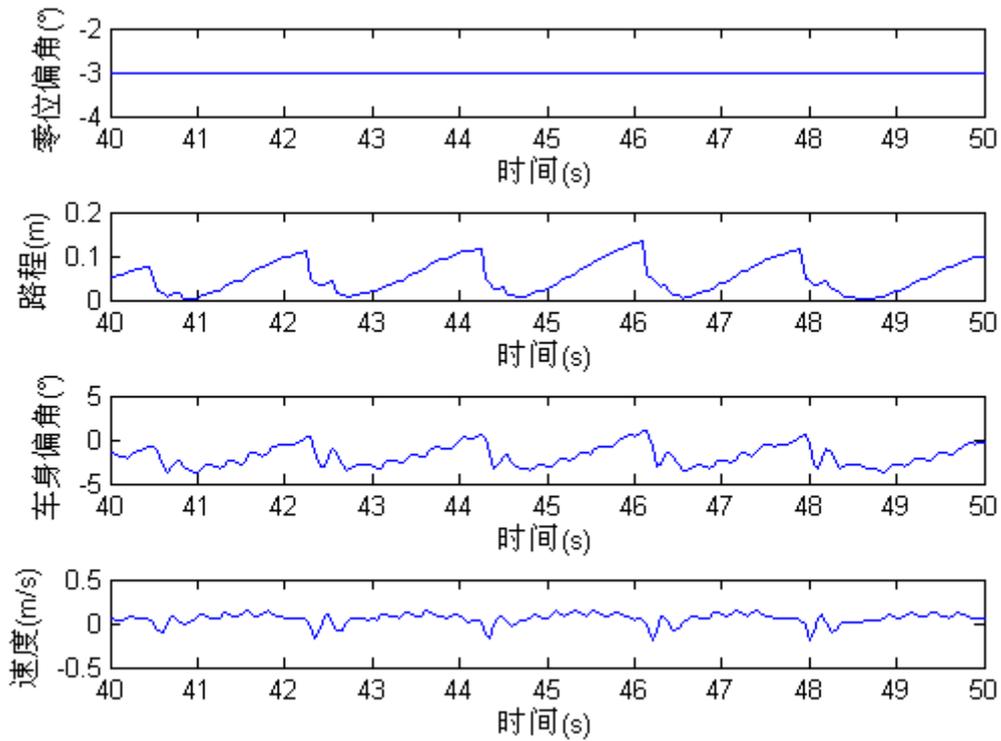


图 5.7 路程大反馈系数控制效果

加入移动距离对初始零位偏角的反馈调整，相对于路程里程信息大反馈系数的控制，这种反馈调整是小反馈系统，不衰减的对初始零位偏角的微调。在这种情况下，大反馈系数的控制仍然存在，故机器人会间断性地向前运动，而零位偏角在不衰减小反馈的情况下，随着路程的增加缓慢被向后修正，直到零位偏角被修正到实际零位偏角的值，此时在原地平衡，路程不再增加，则零位偏角也不再会被修正，实现在运动一段过程后的质心自适应。大概在 15s 内通过上述控制作用，修正了 0.5° 的零位偏角漂移，其作用效果如图 5.8 所示。

从调整的作用效果来分析，路程大反馈系数衰减控制是为了短时间内将姿态调整至向反方向倾斜，其控制的效果对于零位漂移来说是过大的，但是起作用花费的时间短，并且会调整过度。而小反馈不衰减控制是为了长时间将零位偏角逐渐修正至实际值，其起作用的时间花费的时间长，但能缓慢逼近。两者控制作用相结合，最终能实现辅助自适应算法的设计。

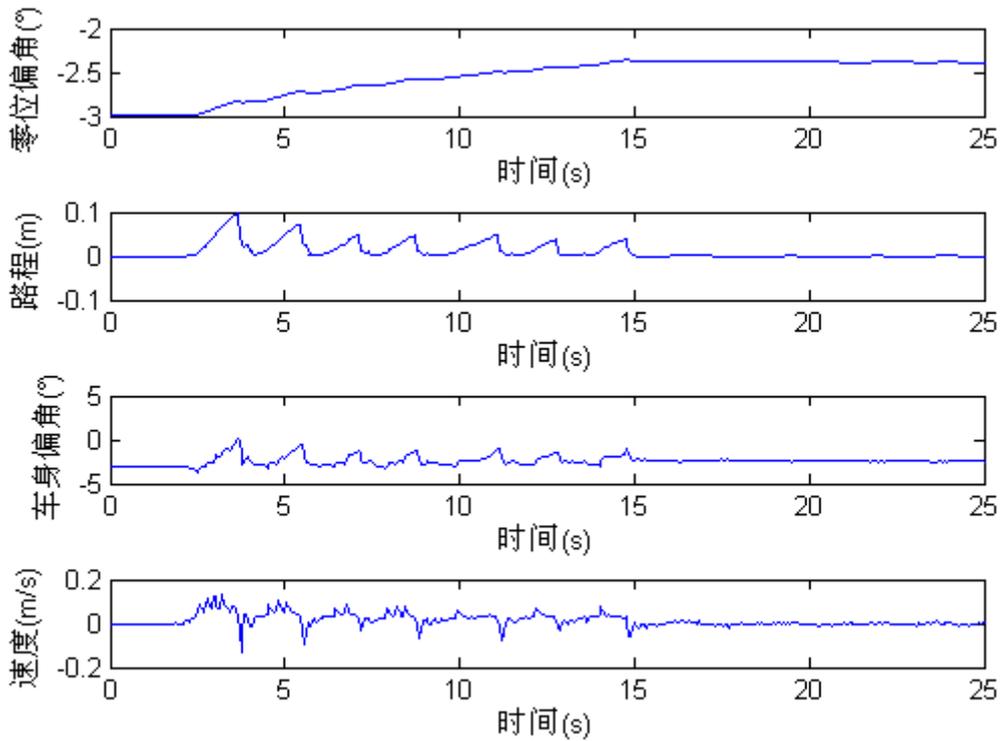


图 5.8 路程大反馈系数加路程小反馈系统控制效果

5.2.5 自适应算法最终设计思路

经上述实物实验证实，辅助自适应算法能实现一定范围内的质心自适应，但零位偏角不能太大，因为这种自适应会相应消耗其运动路程，如果过大的零位偏角需要经过比较大的路程才能修正，故对于大零位偏角的自适应，实时的卡尔曼滤波器还是需要的，最后设计时可考虑采用卡尔曼滤波将零位偏角修正到误差在小范围内，再通过辅助自适应算法实现小零位偏角误差的修正。

5.3 本章小结

本章阐述了机器人质心自适应功能的设计过程，通过最小二乘估计辨识出通过物理测量方法难以得到的系统物理参数，利用物理参数估计值，结合力学方程观测，设计卡尔曼滤波器获得零点偏角的大小，研究其滤波效果。同时设计辅助自适应算法，在小范围内实现质心自适应。

6 结论

6.1 总结

本次毕业设计主要完成的工作内容可分为四大部分。

第一部分为两轮机器人系统的搭建。底层在机械设计和电路设计的基础上，完成了直流无刷电机驱动、IMU 模块机器人位姿读取、电机码盘测速等机器人基本功能。并在整体机器人系统中加入上层监控系统，实现多平台和机器人间的通信与控制。

第二部分为两轮机器人运动控制器的设计与实现。通过对机器人系统与电机的建模，设计了状态反馈控制器，借用 LQR 整定反馈矩阵，在 FPGA 中编程实现了对机器人的平衡控制。后引入“虚拟质心”概念，优化控制结构，进一步实现了在平衡约束下的运动控制。

第三部分为两轮机器人的路径规划与复合运动控制。在电脑和手机两个不同的平台编程实现了对机器人的运动控制。电脑端主要完成的是利用 A*算法，在地图已知的情况下完成路径规划并下发控制。手机端完成的是在随机的手绘路径下解析路径，完成路径规划并下发控制。

第四部分为两轮机器人的质心自适应控制。在系统物理参数未知的情况下，利用最小二乘辨识系统未知参数，得到对系统转动惯量的估计值。利用辨识结果设计卡尔曼滤波器，估计系统质心位置，研究其滤波效果，同时结合路程大反馈衰减控制和小反馈不衰减控制设计辅助自适应方法，实现小范围质心自适应。

6.2 展望

本次设计所完成的两轮机器人，抽象模型为移动倒立摆，而移动倒立摆在平衡控制领域是一个典型的非线性不稳定系统。加上本次设计中对应设计了上传数据采集显示和记录的模块，因此本系统还可作为其他非线性控制算法的实验平台。

本次设计主要针对的是实验室简单环境，水平地面及比较小的外部干扰，如果要进一步研究可开展关于倾斜地面，多干扰复杂环境下两轮机器人的控制与运动规划。

另外在设计实现过程中发现,本设计还有三处可改进的地方:一是可在机械设计上增加驻停装置,在两轮机器人需要长时间稳定直立或是掉电待机的情况下放下驻停装置,依靠驻停装置与两轮三点稳定支撑,也可在驻停装置中增加类似导轮的结构,在平衡被破坏的极限条件下依靠三轮运动,防止意外倾倒对机器人本体的破坏。二是改进电脑端轨迹规划方案,在完成 A*规划后进一步优化路径,参考手机端任意平滑路径的控制方法,使机器人连续平滑运动。三是针对质心自适应算法可在后续继续利用本文所提出的,利用卡尔曼滤波修正大范围零位偏角误差后再利用辅助自适应算法修正小范围零位偏角误差的思路进行设计和实现。

参考文献

- [1] Felix Grasser, Aldo D'Arrigo, Silvio Colombi, Member, Alfred C. Rufer, JOE: A Mobile, Inverted Pendulum, IEEE Transactions on Industrial Electronics[J]. Vol. 49(1).
- [2] Lee Howon, Lee Jangmyung. Driving Control of Mobile Inverted Pendulum[J]. International Conference Ubiquitous Robots and Ambient Intelligence, 2012, 449-453.
- [3] Voth Danna. Segway to the future[J]. IEEE Intelligent Systems, 2005,(3).
- [4] Segway Human Transporter[Z/OL]. <http://www.segway.com>, 2004.
- [5] D.P.Anderson. nBot[Z/OL]. <http://www.geology.smu.edu/~dpa-www/robo/nbot/2007>.
- [6] 尹亮 . 双轮直立自平衡机器人 Sway 研究报告 [R/OL]. <http://www.robotdiy.com/article.php?sid=194>, 2005.4.8.
- [7] Sun Hua, Zhou Haixu. Design of two-wheel self-balanced electric vehicle based on MEMS technology[J]. IEEE Int.Conf.on Nano/Micro Engineered and Molecular Systems, 2008.
- [8] 张华宾, 张培仁, 李昊华等. 基于 DSP 的两轮式倒立摆自平衡控制系统[J]. 测控技术 2007. Vol. 26(7).
- [9] Cheng-Hao Huang, Wen-June Wang, Chih-Hui Chiu, Design and Implementation of Fuzzy Control on a Two-Wheel Inverted Pendulum[J], IEEE Transactions on Industrial Electronics [J], Vol. 58(7).
- [10] 丁学明, 张培仁, 杨兴明等. 分层模糊控制在两轮移动式倒立摆中的应用[J]. 电机与控制学报. Vol. 9(4).
- [11] Li-Xin Wang. Stable Adaptive Fuzzy Control of Nonlinear Systems[J], IEEE Transactions on Industrial Electronics , Vol. 1(2).
- [12] Chaoquan Li, Fangxing Li, Shusan Wang, Fuquan Dai, Yang Bai, Gao Xueshan, Kejie Li. Dynamic Adaptive Equilibrium Control for a Self-Stabilizing Robot[J]. Robotics and Biomimetics (ROBIO) IEEE International Conference on, 2010, 609-614.
- [13] 孙泓. 两轮自平衡小车的模糊滑模控制研究[D]. 西安电子科技大学, 2008.
- [14] Pousti, A, Bodur, M. Kinematics and Dynamics of a Wheeled Mobile Inverted Pendulum[J]. Computational Intelligence for Modelling Control & Automation, 2008 International Conference on, 2008, 409-413.
- [15] Goldstein, H. Classical Mechanics (3rd ed.)[M]. Addison-Wesley, 2001, 35.
- [16] Louis N. Hand, Janet D. Finch. Analytical mechanics. Cambridge University Press.

- 1998, 140–141.
- [17] Wimpenny, J.W.T. The Validity of Models[J]. *Adv Dent Res*, Vol11(1), 150–159.
- [18] Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems[J]. *Journal of Basic Engineering*, 1960, 82 (1), 35–45.
- [19] Lee C.R. High-performance FPGA-based Implementation of Kalman Filter[J]. *Microprocessors and Microsystems*, 1997, 57-65.
- [20] 杨元喜. 动态 Kalman 滤波模型误差的影响[J]. *测绘科学*. 2006, Vol131(1).