

独创性声明

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

研究生签名：周栋

时间：2014年6月4日

关于论文使用授权的说明

本人完全了解宁夏大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同意宁夏大学可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

(保密的学位论文在解密后应遵守此协议)

研究生签名：周栋

时间：2014年6月4日

导师签名：刘永贵

时间：2014年6月4日



摘要

沙漠化是当今世界严重的环境问题，为了实现自然生态的保护，预防沙尘暴，防止沙漠化，需要更好的研究沙漠，因此需要获取沙漠中沙粒振动频率、风速、风向、风蚀等实时数据，使用数据融合和时间序列的分析方法，建立沙尘暴动力学重构模型，探索沙尘暴起因机制和预测方法。由于轮式机器人技术的快速发展，并且非常成熟，采用轮式机器人在沙漠中采集沙漠中的各种信息已经是一种可能。由于沙漠环境特别复杂，必须设计一个快速、有效、可靠的沙漠信息感知机器人运动控制系统。利用嵌入式 Linux 系统、ModbusRTU 协议，研究实现沙漠信息感知机器人运动控制系统，研究的内容主要包括如下方面：

分析沙漠信息感知机器人运动模型。沙漠机器人采用四轮独立驱动，基本运动包括直行、左前转弯、左后转弯、右前转弯、右后转弯。在理想的环境下，对沙漠机器人直行和转弯模型分析，推导得出车速与机器人电机转速的关系，为实际控制系统编程提供理论依据。最后，分析沙漠机器人轮胎打滑的原因和机器人运行的控制算法。

利用 matlab 软件的 simulink 工具建立沙漠信息感知机器人运动控制系统模型，并且仿真。首先，根据直流伺服电机的动态数学模型，建立 simulink 模型，在此基础上建立直流伺服电机伺服系统 simulink 模型，仿真得出调速系统转速响应曲线，最后根据沙漠机器人运行的控制算法建立运行控制系统，仿真当机器人负载加重或上坡时，系统调节转速，增加电机输出转矩，以保证系统稳定。

设计沙漠机信息机器人运动控制系统的硬件接口电路。根据控制系统的设计的方案，分别设计控制系统电源电路，时钟和复位电路，网口电路，串口电路，SD 卡接口电路，直流伺服电机驱动电路。

设计沙漠信息感知机器人运动控制系统的软件。软件设计主要包括机器人直行控制程序，机器人差速转弯控制程序，机器人运行故障检测程序，机器人运行控制程序。

关键词：四轮独立驱动，ModbusRTU，伺服电机，差速转弯

Abstract

Desertification is the serious environmental problem in the world. In order to protect natural ecology, prevent dust storms and desertification, a much deeper research of desert is necessary. So we must obtain real-time data ,such as the vibration frequency of desert sand, wind speed, wind direction and wind erosion, then use the analysis of data fusion and time series to build sandstorm dynamic reconstructed model and explore the mechanisms of causing sandstorm and the forecasting sandstorm methods. Due to the rapid development of wheeled robot technology and its mature, it is possible to use wheeled robot to collect a variety of information in the desert. Because the desert environment is particularly complex, so we must design a fast, efficient, and reliable robot motion control system to collect information in the desert. Using embedded Linux system, Modbus RTU protocol to research and realize desert information perception machine motion control system. The main contributions of dissertation are as follows:

Analyse desert information perception robot motion model. The desert robot uses four-wheel independent drive, the basic motions include: go straight, front-left turn, back-left turn, front-right turn ,back-right turn. In an ideal environment, we analyze desert robot straight moving and turning model, deduce the relationship between robot's speed and robot's motor speed in order to provide a theoretical basis for the actual control system programming.

Using matlab simulink software tools to build the desert information perception robot motion control system model and simulate. First, according to the DC servo motor dynamic mathematical model to build DC motor servo system simulink model based on motor simulink model and simulate to get response curves of speed. Last, build control system in terms of desert robot running control algorithm. Simulating when the robot loads heavier or uphill. The system can adjust the speed to increase the motor output torque to ensure the system's stability.

Design the hardware interface circuit of desert information perception robot's motion control system. According to the designing scheme of control system, design control system power supply circuit, clock and reset circuits, network interface circuits, serial circuit, SD card interface circuit, DC servo motor drive circuits.

Design desert information perception robot motion control system software. Software design includes desert robot information perception straight control program, differential speed swerve control program, running fault detect program, running control program .

keywords: four-wheel independent drive, Modbus RTU, servo motor, differential speed turning

目 录

第一章 绪论	1
1.1 课题的意义	1
1.2 轮式机器人研究现状	2
1.3 本文研究内容	5
1.4 本章小结	5
第二章 沙漠信息感知机器人动力学分析及仿真	7
2.1 机器人运动学分析	7
2.2 机器人力学分析	10
2.3 机器人轮胎转动滑动分析	11
2.4 机器人运行控制算法	12
2.5 机器人控制系统仿真	13
2.6 本章小结	17
第三章 沙漠信息感知机器人运动控制系统总体设计	18
3.1 机器人控制系统方案	18
3.2 机器人运动控制系统硬件选择	19
3.3 机器人运动控制系统通信方式选择	21
3.4 沙漠信息感知机器人控制系统方总体方案	22
3.5 本章总结	22
第四章 沙漠信息感知机器人运动控制系统硬件电路设计	23
4.1 机器人控制系统主要硬件	23
4.2 机器人控制系统主控制器硬件电路设计	25
4.3 直流伺服电机驱动器电路设计	29
4.4 本章小结	31
第 5 章 沙漠信息感知机器人运动控制系统软件设计	32
5.1 嵌入式操作系统	32
5.2 伺服电机驱动器 Modbus 协议	34
5.3 Linux 串口通信编程	37
5.4 机器人运动控制程序设计	38
5.5 本章小结	44
第六章 结论与展望	45
6.1 结论	45
6.2 展望	45
参考文献	46
致谢	48
附录	49
个人简介及攻读硕士学位期间的研究成果	54

第一章 绪论

1.1 课题的意义

沙漠是当今世界很严重的环境问题之一，其是由土地荒漠化恶性发展导致。长期的土地荒漠化最终使土地变成沙漠，对土地资源造成极大的破坏，影响植被生长，使气候变得干燥。因此，为了保护土地资源，需要更好的认识沙漠、了解沙尘暴形成的原理，需深入沙漠获取沙漠中沙粒的振动频率、风蚀、风向、风力大小及气压等数据，对环境变化规律研究，但是大量的数据获取会消耗很多人力物力，并且具有一定的危险性，同时，长期的现场采集数据使得工作效率低，不利于长期进行。因此借助沙基机器人深入沙漠，不仅能代替人类完成这些危险的工作，而且具有很高的工作效率，其以受到人们的重视并且得到广泛的应用。

现在设计的机器人不仅要求能够适应沙漠、沼泽等环境，更要求其能适应未来不可预知的环境，要求其能根据自身携带的大量传感器信息和摄像机感知环境，自主做主路径规划、目标识别等。总之，现在机器人要具备很好的穿越障碍能力、很强的生存能力和自主运动能力，以适应各种复杂的环境^[1]。

目前，可移动式机器人分为轮式、履带式 and 足式移动方式，虽然足式机器人具有很强穿越障碍能力，但其控制特别复杂，而轮式机器人技术特别成熟，结构简单可靠、易控制、高速稳定、能量利用率高、操作性好、通过性和平顺性好，加上适当的车轮配置方式可使其稳定而灵活地在各种环境下运动越障等^[2]。因此，利用成熟的技术，采用轮式机器人应用于沙漠传感信息采集的可实现性较强，开发周期也短，在实际应用中具有非常重要的意义。

由于本次设计中采用的主控制器不仅要控制电机，而且还要接收沙漠环境信息采集控制器发送过来的沙漠传感信息，并且还要收到的数据远程发送到远程上位机，这就要求主控制器的速度非常快，因此，主控制采用飞思卡尔公司以 CortexA8 为体系架构的 iMX53 的 ARM 工业级的处理器，其主频上限达到 1GHz，具有 512M RAM、2G Flash，是一款高性能的处理器，具有快速运行、易于扩展的特点 ARM 处理器，且支持嵌入式 Linux 系统和 WinCE 系统。采用 ARM 控制器主要有以下几点原因：

- (1)采用 ARM 控制器与物联网的相关技术结合，可以处理机器人运动过程中遇到的问题；
- (2)ARM 控制器系统资源特别丰富，主频高，可以快速运行，实时性较好；
- (3)在 ARM 控制器移植操作系统，实现多任务处理，系统 API 函数丰富，减少控制系统开发时间；
- (4)ARM 控制器与无线 3G 通信模块结合，在操作系统上移植无线 3G 通信模块的驱动程序方便快捷，还可以在此基础上设计远程监控平台，利用无线数据通信技术实现远程操作，获取前方数据^[3]。

1.2 轮式机器人研究现状

1.2.1 轮式机器人国内研究现状

轮式机器人由于结构简单,控制容易实现,运行速度较快,稳定、能量利用率高、可操作性好,具有其优越性,因此,国内外学者深入研究了轮式机器人。轮式机器人可以分为一轮式、二轮式、三轮式、四轮式、五轮式、六轮式。一轮式机器人利用圆的几何特性,适应负载地面,实现全方位运动;二轮式机器人具有体积较小、重量小、避障非常灵活,性能好,并且可以将二轮式机器人组装为四轮式或者六轮式机器人;三轮式机器人,车体设计采用三角锥结构,三轮独立驱动,结构较简单,可独立转向和全方位运动^[4];四轮式机器人具有结构简练、容易操作,可以独立控制转矩,且稳定性较好;六轮式机器人相对于四轮式机器人广泛用于太空探测,其具有更好的运行稳定性和越障能力^[5]。

在国内早期,研究设计了一些轮式机器人,其中哈尔滨工业大学设计的两轮并列式月球(图1-1)^[6],和多机动模式可重构移动机器人,采用四驱四转结构能变换出不同轴距、轮距、重心高度、净空和稳定裕量的构型;湖南大学设计的全新菱形布置四轮月球车(图1-2)^[7]。

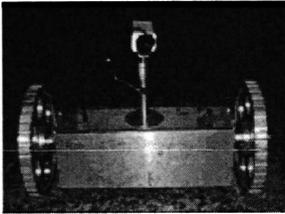


图1-1 两轮机器人

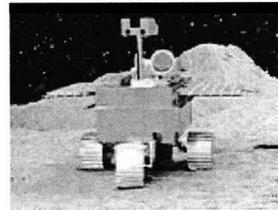


图1-2 四轮三轴月球车

2004年,比亚迪公司推出了四轮电动轮独立驱动轿车(图1-3),其采用直流无刷轮毂电机,为独立双后轮驱动,输出转矩最大可以500N/m,最大速度120km/h。2013年11月26日,我国自主研发的月球探测车“玉兔号”(图1-3)^[8],随着嫦娥三号运载火箭的升空,并且登陆月球。“玉兔号”月球车,长1.5米,宽1米,高1.1米,采用6轮独立弹性轮驱动,具备20度爬坡、20厘米越障能力,可耐受300摄氏度温差;其采用的轮式摇臂悬架设计使其具备前进、后退、原地转向、行进间转向、20度爬坡、20厘米越障能力;其身上携带大量的传感器和摄像机感知环境,确定其位置、速度等参数,通过这些参数实现规划路径、避障、监测等;其采用实时操作系统,并且实现遥测遥控、数据管理、导航控制、移动与机构的驱动控制等功能。

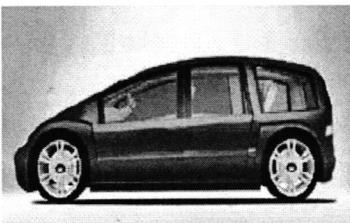


图1-3 比亚迪公司独立驱动轿车

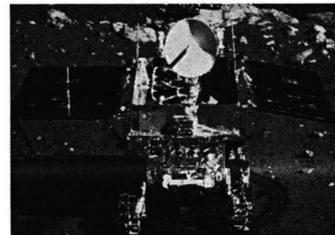


图1-4 “玉兔号”月球车

1.2.2 轮式机器人国外研究现状

在国外,轮式机器人的研究设计相对较早,其中,由于美国军方特别重视轮式机器人,因此美国的技术发展相对较快。在此情况下,美国国内一些公司开始设计轮式机器人,其中洛克希德马丁公司开发的六轮独立驱动的轮式机器人 MULE(图 1-5),卡内基梅隆大学开发的六轮独立驱动的轮式机器人 Crusher(图 1-6)。这两种机器人都采用了六轮独立驱动方式,采用轮毂电机驱动,底盘采用关节式独立悬挂系统,并且可以发送指令控制悬挂系统,使其具有很好的越野性能和越障能力。MULE 可越过 1m 高的障碍,可以通过 1.5 宽的的壕, 21.8 度爬坡 1.25 米的涉水深度,最主要是其可根据环境情况,自主导航,自动调节车身的高度,对运动路径进行自主规划。Crusher 轮式机器人采用涡轮混合动力系统,主动液压悬挂设计,可越过 1.2m 高的障碍,可以通过 2.1 宽的的壕,最大速度 62km/h,载荷 1.5 吨,最大动力 192kw,可谓动力十足^[9-12]。



图 1-5 Mule



图 1-6 Crusher

卡内基梅隆大学的机器人 Nomad (图 1-7)^[13],采用四轮独立驱动,负载均化悬挂系统,底盘可收缩;美国的月球车 Ratler (图 1-8)^[14];卡内基梅隆大学 NASA 研制了一款名为“圣甲虫”的月球漫游车(图 1-9),其悬挂装置可以使使车辆能跨过岩石或在爬坡时使车身倾斜;采用两厢式车体分体结构;日本宇宙科学研究所和明基大学研制的火星探测车 Micro5 (图 1-10)^[15],采用五轮轮式结构,其中一个车轮为支撑轮,动力驱动采用超声马达,可越 1.5 倍的轮径障;美国 JPL 的 Rocky 系列行星探测机器人 Sojourner (图 1-11)^[16]采用六轮独立驱动的摇杆悬吊结构,前后轮独立转向;瑞士联邦工学院设计的轮式机器人 Shrimp (图 1-12)^[17],采用六轮独立驱动,后轮转向,轮臂采用弹簧设计,具有灵活的越障能力;法国分析与系统实验室的机器人 Lama (图 1-13)采用六轮独立锥形轮驱动,主动铰接悬架结构,车身可起伏收缩,独特的的悬架结构使其可越过 2 倍轮径的障碍。前苏联月球探测机器人 Lunokhod (图 1-14)^[18],采用 8 轮独立驱动,弹性扭簧悬挂系统,驱动力大,超强驱动力。1997 年,全俄运输车辆研究所为欧洲宇航局 ESA 研制的 LRMC 机器人(图 1-15)^[19],采用四轮独立驱动、独立转向。瑞士 EPFL 的 Octopus (图 1-16)采用 8 轮独立全驱,并列双轴两段臂悬架,具有越障高度为自身升高。



图 1-7 Nomad 机器人

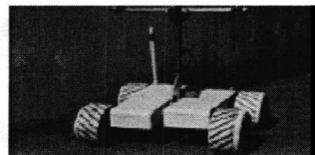


图 1-8 Ratler 机器人

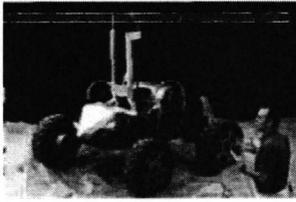


图 1-9 “圣甲虫”的月球漫游车

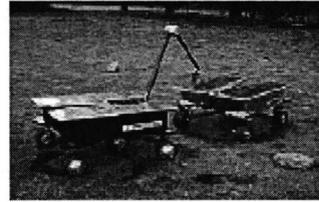


图 1-10 Micro5 机器人

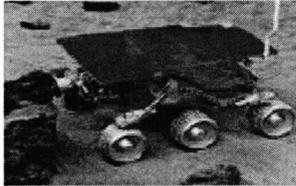


图 1-11 Sojourner 探险车

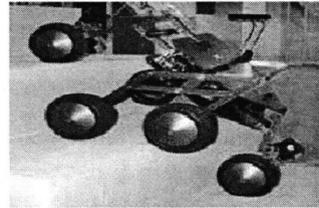


图 1-12 Shrimp 机器人



图 1-13 Lama 探险车

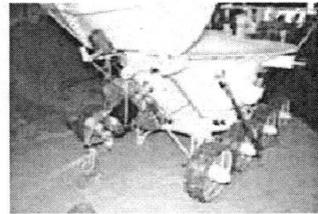


图 1-14 Lunarokhod 探险车

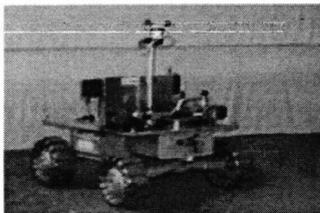


图 1-15 LRMV 机器人

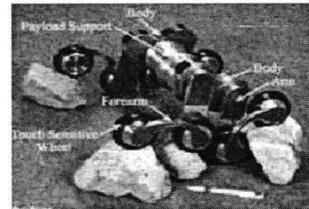


图 1-16 Octopus 机器人

1.2.3 轮式机器人驱动系统

目前轮式机器人驱动系统主要分为集中驱动和分布式驱动，其中集中驱动的这种方式应用比较广泛。如图 1-17 所示，集中式驱动系统，其需要离合器、变速箱、传动轴、差速器等器件，因此连接比较复杂，而且效率还是比较低。

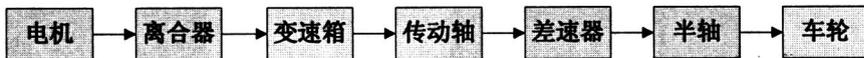


图 1-17 集中式驱动系统

分布式驱动系统是机器人的每一个驱动装置都独立可控，驱动装置之间没有关系，不存在任何机械器件之间的连接，相对于集中式驱动系统，结构简单，效率高，但是控制相对复杂^[20]。分布式驱动系统可以分为两种系统：独立式半轴驱动系统和独立式电动机系统。图 1-18 为独立式半轴驱动系统，图 1-19 为独立式电动机驱动系统。对于分布式驱动系统，由于结构简单，相对于集中式驱动控制系统，系统的控制不需要很多传感器，只需要驱动装置自身的传感器，就可以完成

系统的控制。采用集中式驱动系统的机器人的控制系统一般使用 ABS^[21] (防抱死制动系统) 对车轮控制, 而对于采用分布式驱动系统的机器人, 各车轮实现独立可控, 因此, 各个车轮输出的转矩可以测量和控制, 通过车轮输出转矩的值, 可以得到轮胎驱动力矩传递到地面切向力的大小, 实现车轮动力学的控制。

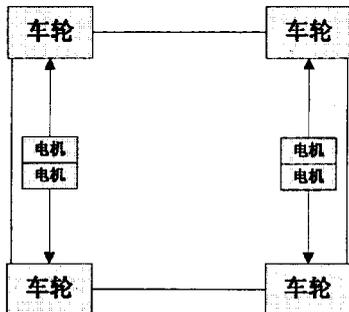


图 1-18 独立式半轴驱动系统

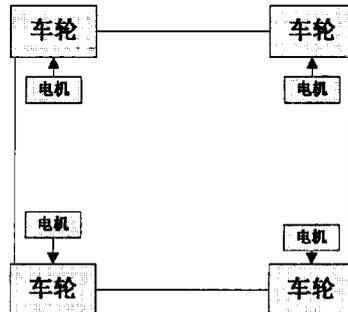


图 1-19 独立式电动机驱动系统

1.3 本文研究内容

1.3.1 本文内容

本文以沙漠腹地监测与信息获取背景, 以嵌入式 Linux 为系统, 利用 Linux 串口通信技术, 实现了沙漠信息感知机器人运动控制系统, 主要研究内容为:

- (1) 利用 Linux 串口通信技术, RS485 组网技术、ModbusRTU 协议, 实现机器人运动控制系统;
- (2) 研究机器人运动控制系统运行控制算法, 并且编程实现;
- (3) 设计机器人主控制器电路和直流伺服电机驱动器电路;

(4) 利用 matlab 软件的 simulink 工具建立机器人运动控制系统模型, 并且对模型仿真, 根据仿真结果, 分析机器人空载和负载时, 电磁转矩和转速的关系。

1.3.2 本文的组织结构

本文以沙漠腹地监测与信息获取背景, 以嵌入式 Linux 为系统, 利用 Linux 串口通信技术, 实现了沙漠信息感知机器人运动控制系统, 主要研究内容为:

- (1) 利用 Linux 串口通信技术, RS485 组网技术、ModbusRTU 协议, 实现机器人运动控制系统;
- (2) 研究机器人运动控制系统运行控制算法, 并且编程实现;
- (3) 设计机器人主控制器电路和直流伺服电机驱动器电路;

(4) 利用 matlab 软件的 simulink 工具建立机器人运动控制系统模型, 并且对模型仿真, 根据仿真结果, 分析机器人空载和负载时, 电磁转矩和转速的关系。

1.3.2 本文的组织结构

第 1 章: 介绍了轮式移动机器人的国内外现状和轮式驱动机器人的驱动系统。

第2章：建立机器人运动模型。首先本章分别建立机器人直行运动和转弯运动模型，分析其运动的过程，得出直流伺服电机转速和机器人运动速度的关系，并且对机器人进行力学分析，得出机器人运动的驱动力矩。之后，分析机器人沙地打滑打滑的原因，并且提出解决此问题的措施。最后，利用 matlab 软件的 simulink 工具分别建立直流伺服电机动态数学模型，直流伺服电机伺服系统模型、机器人模型、机器人运行控制系统模型，并且对各个模型理论仿真，对仿真的结果进行分析。

第3章：沙漠信息感知机器人运动控制系统总体设计。本章分别对两种系统控制方案对比分析、控制器分析、三种电机对比分析、两种通信方式对比，得出沙漠机器人控制系统的总体设计方案。

第4章：沙漠信息感知机器人运动控制系统硬件电路设计。本章根据第3章机器人运动控制系统的设计，设计了整个系统所需要的硬件电路。

第5章：沙漠信息感知机器人运动控制系统软件设计。本章建立了嵌入式 Linux 交叉编译环境，制作主控制器 LinuxFlash 系统启动盘，并且 Linux 系统串口进行编程的基础上，对机器人运动控制系统编程实现。

第6章：总结与展望。本章对本课题进行总结，并提出了系统所存在的不足，最后说明以后所需要完成的事项。

1.4 本章小结

本章主要阐述了本课题研究的目的地意义，详细分析了国内外轮式机器人的研究和发展现状，然后提出了目前轮式机器人主要采用的两种驱动方式的区别和各自的优缺点，最后表述了论文研究的主要内容。

第二章 沙漠信息感知机器人动力学分析及仿真

2.1 机器人运动学分析

机器人的运动具有六个动作，分别是前进、后退、左前转弯、左后转弯、右前转弯、右后转弯。对于前进和后退，控制相对比较简单，机器人前进时，先做匀加速运动，之后达到最大目标速度，做匀速运动；机器人后退时，机器人先做匀速运动，后座匀减速运动，最终速度变为 0，而停止；对于转弯，本文中采用的是差速转弯，利用机器人左右两侧车轮的速度之差实现转弯，其中四个轮胎中，同一侧轮胎的速度是相同的。由于轮式机器人运动受到路面状况的影响不，在本文中对机器人运动学分析之前，做如下假设，方便研究。

- (1)机器人的中心位于其几何中心；
- (2)机器人运动时，忽略空气阻力；
- (3)车轮运动为纯滚动，不发生滑动；
- (4)假设车轮半径严格一致；
- (5)地面为水平地面^[22]。

2.1.1 机器人运动学分析

如图 2-1 所示，机器人直线行走运动模型，O 为机器人的几何中心，R 为机器人车轮的半径， V_R 为车轮边缘的线速度， V 为机器人几何中心 O 点的速度，即机器人的速度， a 为车轮的加速度。

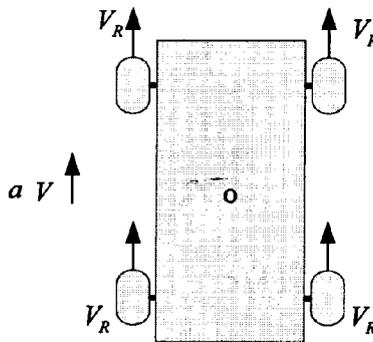


图 2-1 机器人直行行走模型

机器人直线行走采用同步控制，四个车轮在同步运动，并且四个车轮的线速度也相同，此时机器人的速度 $V = V_R$ 。

$$V_R = V_0 + at \quad (2-1)$$

$$V_R = \omega_R R \tag{2-2}$$

$$\omega_R = 2\pi n_R \tag{2-3}$$

$$n_R = \frac{n}{N} \tag{2-4}$$

由 2-1、2-2、2-2、2-4 式推导可得：

$$V = \frac{2\pi n R}{N} \tag{2-5}$$

$$a = \frac{\frac{2\pi n R}{N} - V_0}{t} \tag{2-6}$$

其中 ω_R 为车轮的角速度， n_R 为车轮的转速， n 为直流伺服电机的转速， N 为减速器的减速比。由 2-5 和 2-6 式可知，只要控制直流伺服电机的转速就可以控制机器人的运动的速度和加速度。

2.1.2 机器人转弯运动学分析

对于机器人转弯，本文中设计了四种转弯的动作，分别是左前转弯、右前转弯、左后转弯，右后转弯。如图 2-2 所示，机器人左前转弯运动模型图，其中 L 为机器人车体的长度， W 为机器人车体的宽度， D 为机器人侧面两轮的轮心距离， d 为机器人车轮到其车体边缘的距离， V_1 为机器人转弯内侧车轮的线速度， V_2 为机器人转弯外侧车轮的线速度， O' 为机器人转弯时圆弧的圆心，且其在机器人车体长度的中垂线上， O 为车体几何中心， h 为 O' 到机器人内侧一边中点的距离， θ 为机器人转弯圆弧所对的圆心角， $r_{内}$ 为机器人转弯左侧圆弧的半径， $r_{外}$ 为机器人左前转弯右侧圆弧的半径， V 为机器人几何中心 O 点的速度，即机器人转弯速度。若已知 $r_{内}$ 和 θ ，则由几何图形，可以计算得出：

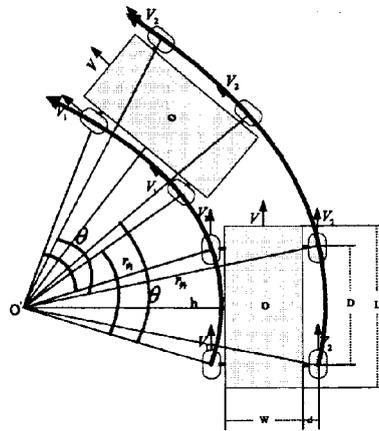


图 2-2 机器人左前转弯运动模型

$$h = \sqrt{r_{内}^2 - \left(\frac{D}{2}\right)^2} + d \tag{2-7}$$

$$r_{\text{外}} = \sqrt{\left(\sqrt{r_{\text{内}}^2 - \left(\frac{D}{2}\right)^2} + d + W \right)^2 + \left(\frac{D}{2}\right)^2} \quad (2-8)$$

又根据公式 2-9、2-10、2-11 推导可以得出 2-12、2-13 式，其中 ω 为机器人车轮的角速度， R 为机器人车轮的半径， n 为直流伺服电机的转速， N 为直流伺服电机减速器的减速比。

$$Tm = k_t I_d \quad (2-9)$$

$$V = \omega R \quad (2-10)$$

$$\omega = \frac{2\pi n}{N} \quad (2-11)$$

$$\frac{2\pi n_1 R t}{N} = \frac{\theta \pi r_{\text{内}}}{180} \quad (2-12)$$

$$\frac{2\pi n_2 R t}{N} = \frac{\theta \pi r_{\text{外}}}{180} \quad (2-13)$$

2-9 与 2-10 式相比可得：

$$\frac{n_1}{n_2} = \frac{r_{\text{内}}}{r_{\text{外}}} \quad (2-14)$$

再根据 2-8 式得出：

$$n_2 = \frac{n_1 \sqrt{\left(\sqrt{r_{\text{内}}^2 - \left(\frac{D}{2}\right)^2} + d + W \right)^2 + \left(\frac{D}{2}\right)^2}}{r_{\text{内}}} \quad (2-15)$$

若又知 n_1 ，则根据 2-12 或 2-13 式可以得出机器人转弯运行的时间为

$$t = \frac{N \theta r_{\text{内}}}{360 R n_1} \quad (2-16)$$

对于其它转弯的动作，和左前转弯类似，不用推导。根据图 2-2 的几何关系，可知机器人几何中心 O 点绕 O' 点做圆周运动的半径为 $h+w/2$ ，在根据上面的推导过程可以得出机器人的转弯时的速度为：

$$V = \frac{\left(\sqrt{r_{\text{内}}^2 - \left(\frac{D}{2}\right)^2} + d + \frac{W}{2} \right) V_1}{r_{\text{内}}} \quad (2-17)$$

2.2 机器人力学分析

轮式机器人在运动时，一般受到五种阻力：空气阻力、车轮滚动阻力、加速惯性力、上坡阻力和传动系统内部的阻力。对于空气阻力其大小与汽车行驶速度有关，因为沙漠轮式机器人在沙漠中的速度很慢，所以空气阻力很小，可以忽略；车轮滚动阻力产生的原因是车轮滚动与地面发生变形，这种变形分为轮胎变形和地面变形，但是在沙漠中轮胎变形可以忽略，地面变形非常严重，因此沙漠机器人在沙漠中运动中滚动阻力是非常大的，必须考虑；对于沙漠机器人上坡时，上坡阻力也非常大，也必须考虑，至于传动系统内部的摩擦阻力可以忽略。

在对沙漠机器人进行力学分析时，由于机器人在上坡时，阻力是最大的，因此以上坡为例分析，且忽略空气的阻力、机器人轮胎的变形、地面打滑等因素的影响，而且机器人车轮与路面为点接触，即可认为车轮只存在滚动，不存在横向运动和纵向运动^[23]。由于机器人是四轮独立驱动，四个轮胎的受力情况是一样的，因此我们只分析一个轮胎的受力情况，如图 2-3 所示，机器人轮胎受力分析图。

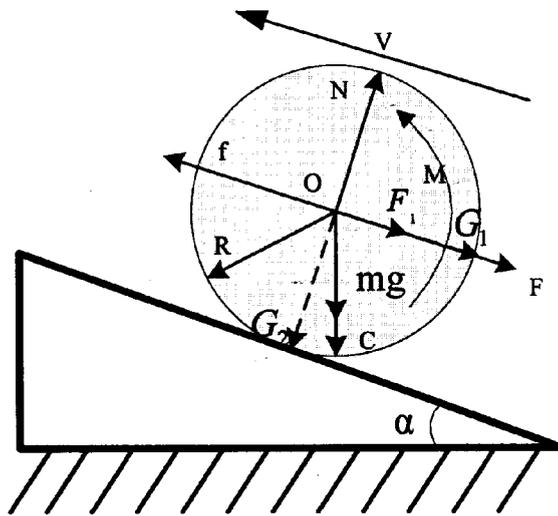


图 2-3 机器人轮胎受力分析图

其中： M 为驱动力矩， N 为地面对机器人车轮的支持力， C 为车轮的载荷， m_R 为车轮的质量， f 为地面对车轮的摩擦力， F 为车体对车轮的作用力， G_1 为上坡阻力， G_2 为车体负荷和轮胎的重力沿垂直斜面方向的分力， F_1 为车轮滚动阻力， V 为车轮轴心的运动速度， R 为车轮的半径。对轮胎受力分析可得：

$$f - F - F_1 - G_1 = m_R \frac{dv}{dt} \quad (2-18)$$

$$M - fR = J \frac{d\omega}{dt} \quad (2-19)$$

其中 J 为车轮的转动惯量， ω 为车轮的角速度， $F_1 = z_R C$ ， $G_1 = (m_R g + C) \sin \alpha$ 。又由

$v = \omega R$ ，可得：

$$\frac{dv}{dt} = R \frac{d\omega}{dt} \quad (2-20)$$

由 2-18、2-19、2-20 式可得：

$$F = \frac{M}{R} - z_R C - (m_R g + C) \sin \alpha - \frac{dv}{dt} \left(m_R + \frac{J}{R^2} \right) \quad (2-21)$$

当 $\alpha = 0$ 时，即机器人在平面内运行，此时

$$F = \frac{M}{R} - z_R C - \frac{dv}{dt} \left(m_R + \frac{J}{R^2} \right) \quad (2-22)$$

对车轮受力分析之后，下面对机器人主体进行受力分析，如图 2-4 所示。由受力可以得到：

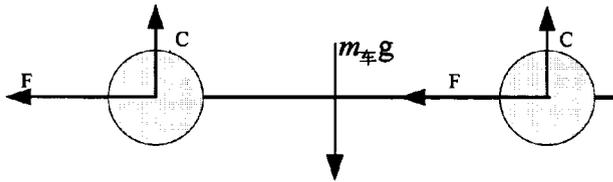


图 2-4 机器人主体受力分析图

$$4F = m \frac{dv}{dt} \quad (2-23)$$

由 2-21 式可得：

$$M = \frac{dv}{dt} \left(\frac{1}{4} m + m_R + \frac{J}{R^2} \right) R + (m_R g + C) \sin \alpha + z_R C R \quad (2-24)$$

根据公式 $P=FV$ 、 $F=M/R$ ，可以计算得到机器人的总功率。

2.3 机器人轮胎转动滑动分析

轮式机器人前进是通过电机输出的扭矩通过车轮作用于地面，从而获得推力，但是，由于地面的变化，导致轮胎与地面的摩擦力减小，使轮式机器人无法获得足够的推力，车轮只能空转打滑。根据公式 2-25 式，

$$f_{\max} = \mu C \quad (2-25)$$

其中 f_{\max} 为地面对车轮的最大静摩擦力， μ 是地面的附着系数， C 是车体负荷。当机器人在沙漠里运动，地面沙土非常松软，因而地面的附着系数非常小，当电机施加的力稍大，大于此时地面给轮胎所能提供的最大静摩擦力，轮胎就会出现打滑，长时间这样打滑，会使电池的功耗增加，轮胎磨损也会增大，并且使沙漠机器人的稳定性降低，同时会使机器人的动作调整产生影响。

针对沙漠机器人在沙漠中的打滑问题，提出如下两点措施：

(1) 增加车轮轮胎宽度，选择的轮胎花纹比较深，而且实地测试经验，还要选择充气轮胎，且轮胎的气要放掉，目的在于增加轮胎与沙地之间的附着系数；

(2) 实时监测，车轮的打滑问题，认为控制电机输出的转矩，避免这种现象的发生。在轮式机

器人控制方面，称轮胎与地面的最大摩擦系数 μ_{\max} 所对应的滑移率称之为最佳滑移率 λ_{\max} 。当实际运行时的滑移率 λ 小于 λ_{\max} 时，则证明轮胎无打滑现象，机器人运行稳定，反之，则发生打滑现象。对于滑移率又分为制动滑移率和驱动滑移率，本设计设计的机器人无制动装置，因此制动滑移率，不在这里讨论，只讨论驱动滑移率。一般定义驱动滑移率：

$$\lambda = \frac{v - v_0}{v} \tag{2-26}$$

其中 v 为机器人实际对地的速度， v_0 为轮胎转速。因此，只要实时监测 v 和 v_0 的值，控制其在 0 到 1 内变化，就可以避免轮胎打滑，但是对于本次设计 v_0 直接可以通过指令读取电机当前的转速进行计算而得，但是 v 的检测需要实际的传感器检测，本次设计没有这种传感器，在这里只做理论分析，在下一版的沙漠机器人的设计中，加入此传感器。

2.4 机器人运行控制算法

在沙漠机器人的运行中，需要实时监测电机的工作状态。当机器人遇到障碍物时，负载力矩增大，此时必须增大电机输出扭矩，则需要减小转速，反之，机器人负载力矩小时，则电机转速增加，达到最大速度，机器人以最大速度运动。由于机器人在沙漠环境，路面不可能是平坦的，四个轮子遇到的负载力矩也不一样，但是只需要检测负载力矩最大的一个轮胎，但是负载力矩无法检测，只需要检测每个电机线圈上的电流，即可实现负载力矩的检测。根据公式 2-27、2-28、2-29，可以得出式 2-30，其中 P 为电机功率， T 为电机输出转矩， n 为电机转速。

$$P = FV \tag{2-27}$$

$$F = \frac{T}{R} \tag{2-28}$$

$$V = 2\pi n \tag{2-29}$$

$$P = \frac{\pi R}{30} Tn \tag{2-30}$$

由于电机的功率一定，因此电机输出的转矩和转速成反比，因此通过电流的检测，实现对负载力矩的检测，只要负载力矩发生变化，通过调整转速，实现电机输出转矩的调整，防止出现电机负载过重而停机。如图 2-5 所示，机器人运行控制结构图，对转速的控制采用 PID 控制器，通过检测电流，将电流通过计算转换为转速，以此转速做为 PID 控制器的目标转速，以电机反馈的转速的反馈转速值，计算误差将误差送入 PID 控制器降低误差，调节转速从而实现随着负载转矩的变化，动态调整调整机器人的运行速度。

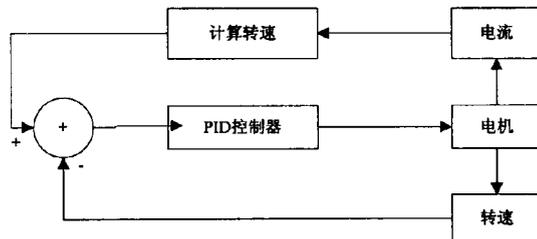


图 2-5 机器人运行控制结构图

2.5 机器人控制系统仿真

2.5.1 直流伺服电机数学模型建立

如图 2-6 所示, 直流电机等效电路图, 其中 U_c 是驱动装置的输入电压脉冲, 通常为占空比为 0-1 变化的 PWM 脉冲波, U_d 是驱动装置输出电压, 当 U_c 占空比在 0-1 变化时, U_d 在 0-24v 之间变化, 单位伏特 V, R_a 为直流电机等效电阻, L_a 为直流电机等效电感, I_d 为直流电机线圈电流, E 为直流电机感应电动势, T_m 为电机电磁转矩, ω_m 为电机机械角速度, T_L 为直流电机负载转矩^[24]。

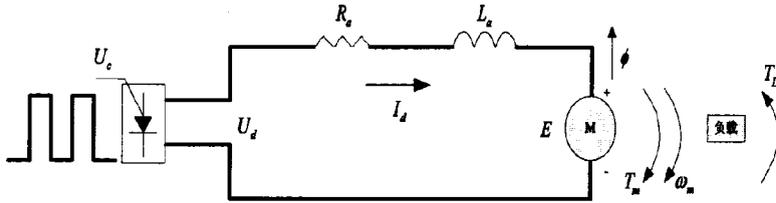


图 2-6 直流电机等效电路图

根据驱动装置的工作原理, 将驱动装置的传递函数等效为一个一阶惯性环节, 即

$$W(s) = \frac{U_d(s)}{U_c(s)} = \frac{K_s}{T_s s + 1} \quad (2-31)$$

其中, U_d 为驱动器输出的空载平均电压; U_c 为驱动器的控制电压; $K_s = U_d / U_c$ 为驱动器的放大系数; $T_s = 1/2f$ 为驱动器中开关周期。

根据直流电机等效电路图, 在假定气隙磁通恒定时, 直流电机的电路平衡方程为:

$$U_d = E + I_d R_a + L_a \frac{dI_d}{dt} \quad (2-32)$$

力矩平衡方程为:

$$T_m - T_L = J_t \frac{d\omega_m}{dt} + B_t \omega_m \quad (2-33)$$

在 2-32, 2-33 式中, ω_m 是电机输出的机械角速度; $J_t = J_m + J_L N^2$ 系统总转动惯量, 单位 $\text{kg}\cdot\text{m}^2$, J_m 电机转动惯量, 单位 $\text{kg}\cdot\text{m}^2$, J_L 负载转动惯量, 单位 $\text{kg}\cdot\text{m}^2$; T_L 系统的负载转矩, 单位 $\text{N}\cdot\text{m}$; $B_t = B_m + N^2 B_L$, 总摩擦系数, 单位 $\text{N}\cdot\text{m}/\text{rad}/\text{sec}$, B_m 电动机粘性摩擦系数, 单位 $\text{N}\cdot\text{m}/\text{rad}/\text{sec}$, B_L 负载粘性摩擦系数, 单位 $\text{N}\cdot\text{m}/\text{rad}/\text{sec}$ 。

由已知: $E = K_b \omega$ 伺服电机感应电势, 单位伏特 V; $T_m = k_t I_d$ 电机电磁转矩, 单位 $\text{N}\cdot\text{m}$, 可以在 matlab 的 simulink 中搭建模型如图 2-7 所示。

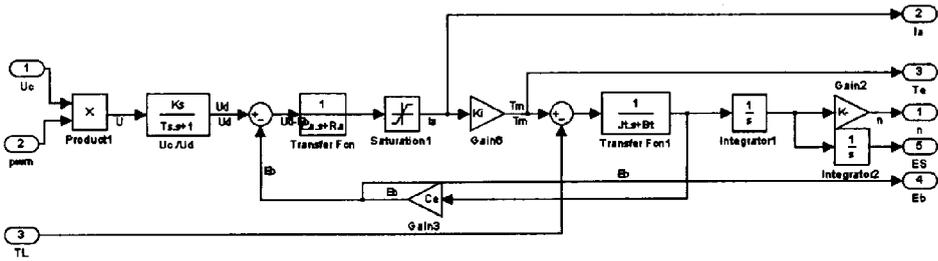


图 2-7 直流电机 simulink 等效模型

2.5.2 直流伺服电机伺服系统 simulink 模型建立及仿真

如图 2-8，直流电机伺服系统采用双闭环控制，内环为电流环，外环为转速环，且电流环和转速环都采用 PI 调节^[25-26]。转速环 PI 调节的基本作用是使转速 n 跟随给定电压变化，稳态无静差，对负载变化起抗干扰作用，对输出幅值决定最大允许电流。电流环 PI 调节的作用是在电机启动时保证获得允许的最大电流，从而加快系统的动态响应，保证启动的快速性；对电流环前向通道扰动能及时调节，增强系统的抗干扰性能；改造电流环对象，提高电流跟随性能；当点击过负载或堵转时，限制电枢电流最大值，可起到快速安全保护作用^[27-30]。

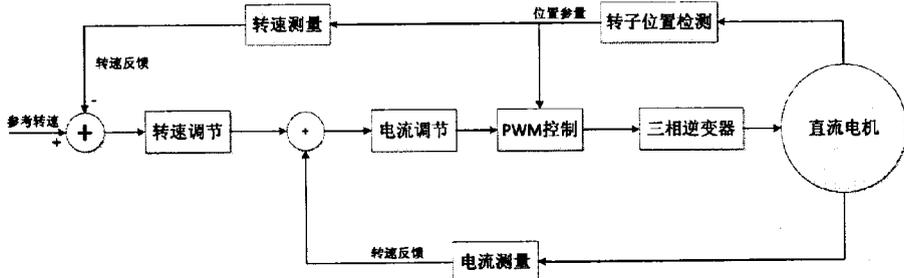


图 2-8 直流电机伺服系统

以 2.5.1 节对直流伺服电机建立 simulink 模型为基础，建立整个伺服系统模型如图 2-9 所示。

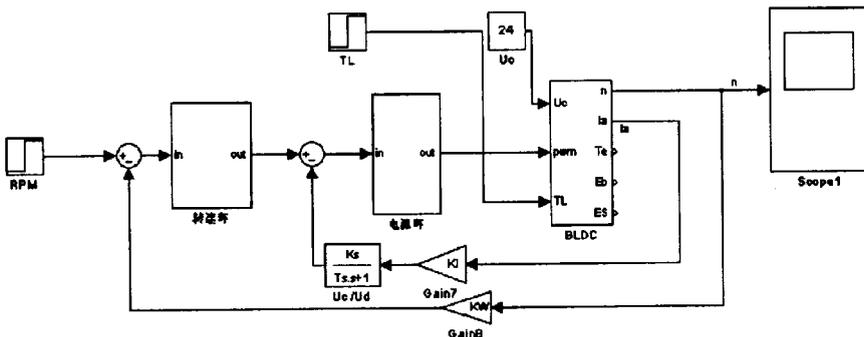


图 2-9 直流电机伺服系统 simulink 等模型

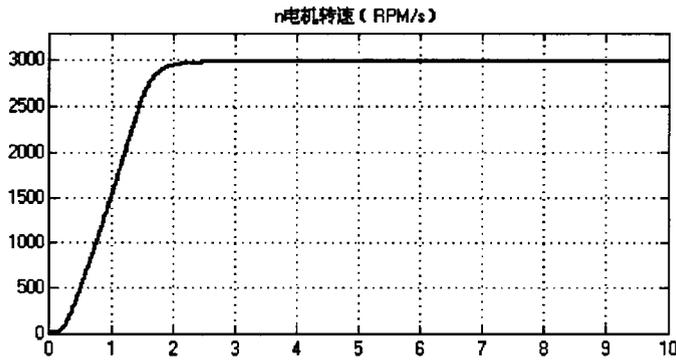


图 2-10 直流电机伺服系统 simulink 仿真速度响应曲线

在图 2-10 中，可以看出，目标转速为 3000r/min，系统快速响应，将转速调节到目标转速。

2.5.3 机器人控制系统仿真

机器人采用四轮独立驱动，需要四个电机和四个电机驱动器，如图 2-11 所示机器人模型，W_L_F 为左前轮电机系统，W_L_B 为左后轮电机系统，W_R_F 为右前轮电机系统，W_R_B 为右后轮电机系统，RPM 为参考转速，TL 为负载力矩。

图 2-12 中，为机器人的速度和加速度响应曲线，可以看出，当给定目标转速时，系统快速响应，达到目标速度，此时由于速度达到目标值，加速度也变为 0。

在第二章中，已经详细阐述运行控制系统的算法，系统设定目标转速，开始运行，同时系统检测四个电机的电磁转矩，然后比较 4 个电机的电磁转矩的大小，得到电磁转矩最大的一个电机（说明此机器人的车轮收到的负载力矩比较大），然后根据公式计算得到此转矩对应的转速，此时以这个转速为目标值，利用调速系统 PID 控制器调节转速，快速达到此设定值。图 2-13 为机器人运行控制系统 simulink 模型。

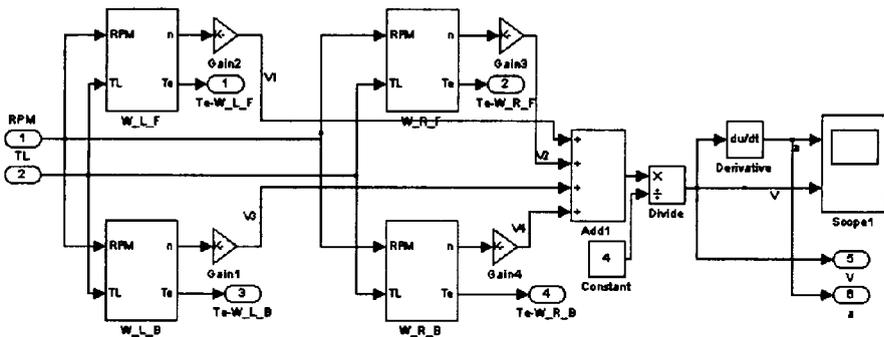


图 2-11 机器人 simulink 模型

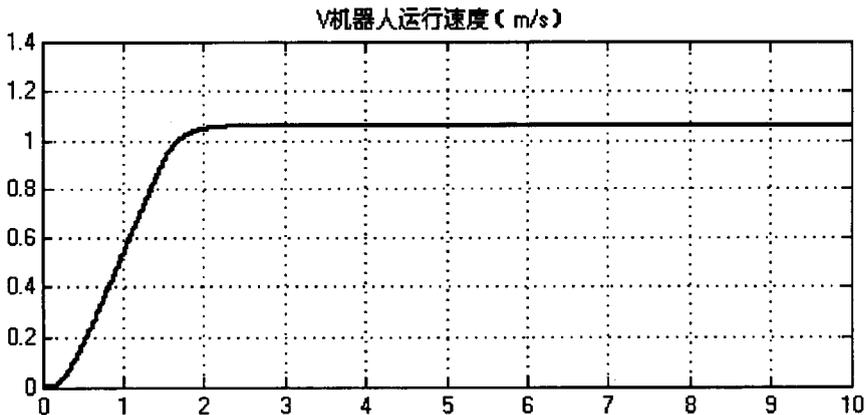


图 2-12 机器人速度响应曲线

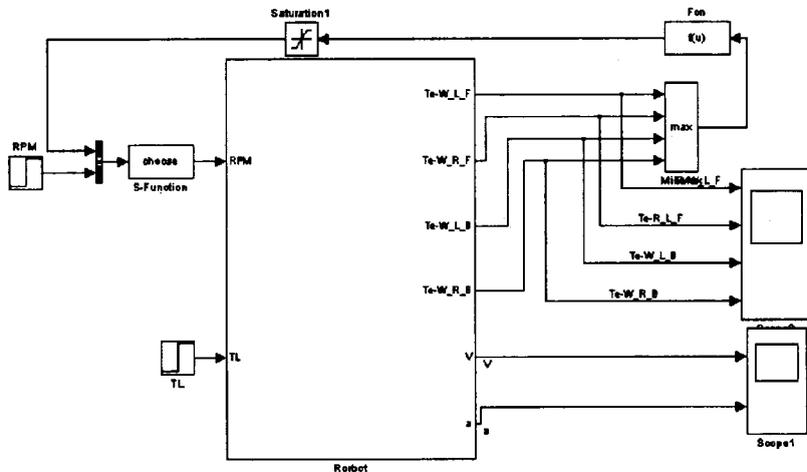


图 2-13 机器人运行控制系统 simulink 模型

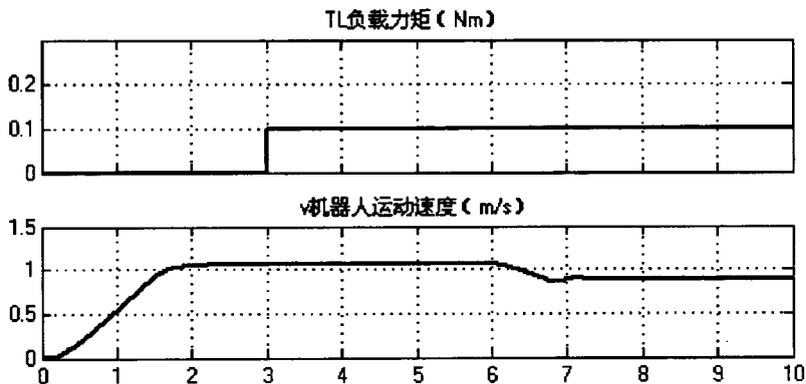


图 2-14 机器人运行控制系统 simulink 模型仿真结果

图 2-14 为机器人运行控制系统 simulink 模型仿真结果，图中在系统运行后，在 1.8s 达到最大速度，在 3s 时，系统负载力矩突然增加，系统快速调节，降低速度，达到调节后的速度值。

2.6 本章小结

本章的任务主要建立机器人运动模型，设计运行算法，并且利用 Simulink 工具仿真。对本次设计的机器人直行运动和转弯进行运动学分析，为机器人控制系统编程提供理论依据；对机器人进行力学分析，得出使机器人运动的驱动力矩，进而，可以根据设计需求得到所需的电机功率；分析沙漠机器人轮胎打滑的原因，并且提出解决措施；设计沙漠机器人运行的控制算法，并且建立机器人模型，对机器人运行算法仿真。

第三章 沙漠信息感知机器人运动控制系统总体设计

3.1 机器人控制系统方案

本文所设计的系统是基于 ARM 的机器人控制系统，机器人采用的是四轮独立驱动，四个伺服电机驱动器分别控制四个伺服电机。系统设计要求机器人具有前进、后退、左前转弯、右前转弯、左后转弯、右后转弯，一共 6 个动作：

(1)机器人前进控制：机器人前进必须让四个轮子同步，为了四个轮子同步，必须保证四个轮子的车速相同；

(2)机器人后退控制：机器人后退和机器人前进控制一样，采用的同样的方式，只不过此时的伺服电机的转速为负值；

(3)机器人转弯控制：机器人采用的是四轮独立驱动，转弯采用差速转弯^[31]。当机器人左前转弯时，左侧的两个轮子和右侧的两个轮子都向前转，但是左侧的轮子的转速小于右侧轮子的转速；当机器人右前转弯时，左侧的两个轮子和右侧的两个轮子都向前转，但是右侧的轮子的转速小于左侧轮子的转速；至于机器人左后转弯、右后转弯，其与左前转弯、右前转弯是一样的，只是车轮的方向相反。

3.1.1 集中式控制方案

如图 3-1 所示集中式控制方案，其将主控制器和四个直流伺服电机驱动器利用总线组成网络，实现主从机通信。对于这种控制方案，通信方式可以选择 RS485 总线或 CAN 总线。当使用 RS485 总线通信时，总线协议选择为 Modbus 协议，当使用 CAN 总线时，则总线协议为 CAN 总线协议。

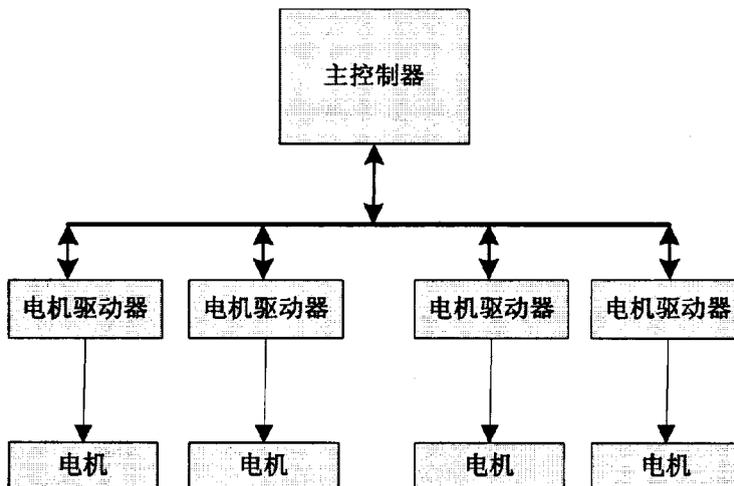


图 3-1 集中式控制方案

集中式控制方案优点在于使用总线的广播地址可实现各电机的同步操作，使四个电机同时实

现启停、保持四个电机的转速相同。但是，在通过指令读取各个电机的状态时，网络延时比较大。

3.1.2 分布式控制方案

如图 3-2 分布式控制方案，主控制器与各个直流伺服电机驱动器独立通信，通信方式可以选择 RS232、RS485、CAN 等总线。使用这种控制方案优点是网络延时较小，传输速度高，但是成本较高，而且对于控制电机不能实现各电机的同步控制。

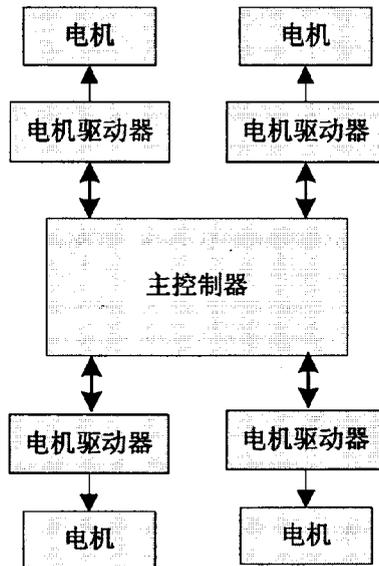


图 3-2 分布式控制方案

3.2 机器人运动控制系统硬件选择

3.2.1 主控制器选择

在嵌入式系统设计的领域，ARM 具有很强的竞争力，相对于其他处理，ARM 具有耗电省、功能强、成本低，而且支持 16 位 Thumb 指令、32 位的 ARM 指令和 Thumb-2 指令。除此之外，ARM 处理器采用 RISC 结构，具有多种处理器状态模式，而且采用先进的片内 AMBA 总线技术和低功耗技术。

ARM 处理器中分为 Cortex-R、Cortex-M、Cortex-A 三个系列^[32]。Cortex-R 系列的 ARM 处理器要求用到具有实时操作的环境，一般用于打印机、硬盘等。Cortex-M 系列的 ARM 处理器、最大的特点就是超低能耗，能量最省，处理器的门电路数量少、代码占用空间小，一般用于低端控制领域，如医疗器械、电子测量、照明、智能控制等。Cortex-A 系列的处理器为高端应用而生，其设计采用最先进的推测性 8 级流水线，因此使处理器的性能、功能达到前所未有的水平，广泛用于移动应用、网络和企业。Cortex-A 系列的处理器中，Cortex-A8 是 ARM 处理器开发以来，具有性能最高、功率效率最大的处理器，其处理器的主频可以在 600MHz 到 1GHz 之间动态调整，并且具有丰富的 I/O 口，内部集成 AD、DA、USB 控制器、LCD 控制器、RS485、RS232、SPI

等接口。其具有如下功能：

- (1) NEON 技术，用于多媒体处理和信号处理；
- (2) 运行时间编译目标技术；
- (3) 优化的 1 级高速缓存和集成的 2 级高速缓存；
- (4) 支持 Thumb-2 指令，以提高代码的高性能，减少代码存储空间；
- (5) 具有内存管理单元，因而可以运行操作系统及丰富的应用。

本此控制机器人使用的控制器不仅要求控制电机，还要接收各种传感器数据，并且将数据发送到远程控制中心，同时还要接收远程机器人控制指令，因此要求控制器具有较高的速度，而且能进行多任务处理。由于 Cortex-A8 优越性能，因此选用其做为机器人控制器的处理器，并在其上运行 Linux 系统，实现多任务处理。

3.2.2 电机选择

在机器人控制系统中，驱动电机可以选择为步进电机、直流伺服电机、交流伺服电机。步进电机驱动采用步进电机驱动器，直流伺服电机驱动采用为直流伺服电机驱动器、交流伺服电机驱动采用交流伺服电机驱动器。在实际选用电机时，要求控制机器人的电机具有以下特点：

- (1) 速度响应的快速性。电机从获得要求的工作指令到要求的工作状态，时间越短越好；
- (2) 启动转矩大。在具体负载情况下，要求电机具有高的启动转矩较低的转动惯量；
- (3) 能实现连续性和直线型的控制。要求电机的转速能使先连续变化，尽量与控制信号近似成正比；
- (4) 宽的调速范围；
- (5) 电机的质量和尺寸小；
- (6) 电机具有较高的过载能力。

步进电机是一种将电脉冲转换为角位移的机构，当步进电机收到一个脉冲信号，其就转动一定角度，其旋转就是控制其以固定角度运行，进而实现准确的定位。步进电机是一种特殊的电机，其没有累计误差，但是只是开环控制。

直流伺服电机实质就是一个直流电机，只是在其上面加入脉冲编码器、霍尔传感器，实现精确调速。其可以分为有刷直流伺服电机和无刷直流伺服电机。直流有刷直流伺服电机成本较高、结构复杂、启动转矩大、具有很宽的调速范围、控制较容易、需要经常维护、产生电磁干扰。直流无刷伺服电机具有体积小、重量轻、输出电磁力矩稳定、速度响应快、惯量小、不需要经常维护、电磁辐射小、寿命长^[33]。

交流伺服电机是一种特殊的直流伺服电机，驱动器控制三相线圈形成电磁场，转子在磁场的作用下产生振动，同时驱动器利用电机编码器的反馈转速实现闭环控制，达到精确的转速，精度取决于编码器的线数。但是，相对于直流伺服电机，交流伺服电机优点在于其一般应用于高速大力矩的应用，同时其控制较复杂。

在步进电机和伺服电机中，两者各有特点，下面从如下几个方面比较^[34]：

- (1) 不同的控制方式。步进电机采用脉冲的个数控制，实现精确的角度控制。而伺服电机采用脉冲的时间长短控制，实现精确的角度控制；

(2)不同低频工作特性。在低速工作时,步进电机容易发生低频振荡,伺服电机不会发生低频振荡,工作稳定;

(3)不同转矩输出特性。步进电机的转矩随着转速的升高,输出转矩会下降,当在高速使,步进的输出转矩会急速下降。伺服电机输出转矩不会随着转速的升高而变化,其可实现恒转矩输出。

(4)不同的过载能力。步进电机不具有过载能力,伺服电具有强的过载能力;

(5)不同的速度相应能力。步进电机达到额定转速一般需要几百 ms,而伺服电机只需要几 ms。

根据以上的比较,机器人驱动的电机选择为伺服电机。对直流伺服电机和交流四度电机,选择为直流伺服电机,原因在于本次设计的沙漠机器人在沙漠中不需要很高的速度,而交流伺服电机一般用于高速应用,同时交流伺服电机控制相对直流伺服电机控制较麻烦,因此选择为直流伺服电机。

3.3 机器人运动控制系统通信方式选择

3.3.1 CAN 总线通信方式

CAN 即控制器局域网络,常用于汽车电子各模块之间的通信,这种总线是标准的工业现场总线,具有实时性较高,很强的传输可靠性,采用两线传输 CANH 与 CANL。其具有很多特点^[35]:

- (1)CAN 总线可实现 10Km 距离的传输,传输速率可以达到 1Mbps;
- (2)CAN 总线采用多主结构,且各节点都可以成为主机,各节点没有主从之分;
- (3)CAN 总线具有较高的实时性,采用不具有破坏性的总线仲裁技术;
- (4)CAN 总线硬件设计具有消息滤波功能,依次减轻 CPU 的负担,减少软件编程;
- (5)当 CAN 节点出现错误时,其具有自动关闭出错节点,使出错节点不影响总线通信。

CAN 总线虽然具有很多的优点,但是成本较高,而且协议比较复杂,要实现 CAN 总线协议的编程,需要了解 CAN 总线协议的各层,软件开发的周期较长。

3.3.2 RS485 总线通信方式

RS485 总线是使用最广的串行总线标准,接口电平低于 RS232 总线,因此不易损坏接口芯片,另外,其兼容 TTL 电平,可直接与 TTL 电路连接,其具有如下特点:

(1)RS485 总线可实现 1Km 距离的传输,传输速率可以达到 1Mbps,但是以最大速度传输时,传输距离只能达到 100m;

- (2)RS485 总线采用单主结构,各节点之间通信只能有一个主机,其余为从机;
- (3)RS485 总线采用平衡发送和差分接收,因此,具有很强的抗干扰能力,共模抑制比高;
- (4)采用不同的芯片可实现 RS485 全双工和半双工通信;
- (5)硬件不具有滤波功能,而且传输过程中一个节点出现问题,整个总线全部瘫痪。

由于 RS485 总线,只定义了物理层没有数据链路层,造成总线错误无法识别,为了弥补这个缺点,一般在 RS485 总线网络通信时,采用 Modbus 协议经行数据通信,即可避免总线错误无法识别的缺点。其次,RS485 芯片比较便宜,而且 Modbus 协议简单,软件编程较易实现,开发周

期短。在 Modbus 协议中，具有 ASCII 模式和 RTU 模式，由于 ASCII 模式是将原信息的一字节转换为两字节传输，传输的字节多于 RTU 模式，因此选择 RTU 模式传输数据。

3.4 沙漠信息感知机器人控制系统总体方案

在 3.1、3.2、3.3 节中分别对机器人控制系统的两种控制方案分析，主控制选择分析、电机选择的比较分析、通信方式的比较分析，得出如图 3-4 所示的沙漠信息感知机器人控制系统总体设计，系统控制方案采用集中式控制方案，主控制采用 Cortex-A8 主控制器，驱动电机采用直流伺服电机，通信方式采用 RS485 通信，协议采用 ModbusRTU 协议^[36]。尽管 CAN 总线在很多方面都比 RS485 总线有优势，但是本次控制电机使用 RS485 总线的原因有如下几点：

(1)CAN 总线收发芯片价格高于 RS485 总线收发芯片的 2 倍，同时 CAN 总线协议复杂，软件开发周期长；

(2)使用的 Cortex-A8 主控制器的 CAN 总线底层驱动存在问题，导致收发数据出现问题，厂家要求签保密协议，才能修复底层驱动问题；

(3)本次设计的控制系统就是单主机多从机模式，因此采用 RS485 总线足以使用。

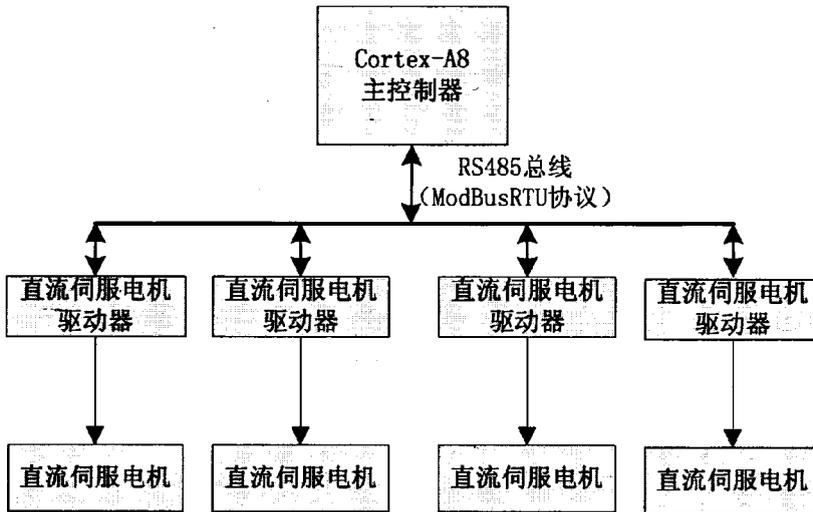


图 3-4 沙漠信息感知机器人控制系统总体设计

3.5 本章总结

本章分别分析机器人控制系统的两种方案的优缺点，选择主控制器的要求，机器人驱动电机的三种电机的优缺点，及两种通信方式的优缺点，然后根据实际的情况，设计了本次沙漠信息感知机器人控制系统的总体设计方案。

第四章 沙漠信息感知机器人运动控制系统硬件电路设计

本设计中,涉及到硬件电路有 ARM 主控制器、直流伺服电机驱动器、直流伺服电机。ARM 主控制器采用的是研华科技的 RSB4210,其具有丰富的资源,但是本次设计只用到了串口等资源。ARM 主控制器和电机驱动器之间采用 RS485 接口进行通信。

4.1 机器人控制系统主要硬件

4.1.1 机器人主控制器

本次设计采用的控制器是研华科技的 RSB4210,它采用飞思卡尔公司的 iMX53 处理器,这个处理器是以 ARM CortexA8 为体系架构的。iMX53 处理器是 32 位,且主频上限达到 1GHz,是一款高性能的处理器,具有快速运行、易于扩展的特点,而且具有灵活的 I/O 端口和完整的硬件和软件的解决方案,已方便用户开发自己的应用程序,不再考虑系统的集成。

RSB-4210 是一款工业级、不用散热、简单的控制板,具有可靠强大的电源管理,其主要特点有以下几点:

- (1)具有以 CortexA8 为体系架构的好性能处理器;
- (2)支持 OpenGL 2.0 和 OpneVG1.1 硬件加速;
- (3)支持高清晰度的 1080P 的视频编码和高清晰度的 720P 视频编码硬件引擎;
- (4)采用飞思卡尔公司智能快速技术,低功耗;
- (5)具有 3-3VI/O 端口,以工业设计的理念为出发点,具有宽广的工作温度;
- (6)具有高性能的 I/O 端口,其中具有 4 个串口、USB、以太网接口、SD 卡、30 个 GPIO、SATA、SPI、CAN 等;
- (7)支持 SATA 存储接口和 CAN 总线应用;
- (8)支持 Andriod2.3、嵌入式 Linux2.6 (本次实际使用)、winCE;
- (9)宽广的工作温度为-40~85 摄氏度;
- (10)系统 RAM 512M,板载 Flash 2G;
- (11)支持 RTC、看门狗、复位。

RSB-4210 具有的资源有,5 个串口,两个 3 线 RS232,一个 9 针 RS232,一个 RS485,一个 4 线 3.3V 串口;一个 10/100M 以太网接口;3 个 USB2-0 接口;1 个 SD 卡接口;一个 CAN 总线接口;一个 6*6 键盘接口;20 个 3-3V TTL 端口;CPU 内部集成 LCD 控制器,一个 HDMI 接口等。

4.1.2 直流伺服电机电机及驱动器

本次设计使用的直流伺服电机驱动器采用最新 32 位 dsp 处理器,支持 20~50V 输入,支持串

□ ModbusRTU 模式控制，支持串口 RS232 电平控制方式，具有欠压，过压，堵转，过热保护。使用的电机额定功率为 64W，额定转速为 3000rpm，额定电流为 4.0A，额定转矩为 0.2Nm，峰值电流为 14A，峰值转矩为 0.7Nm，极对数为 8，电机线圈电阻为 0.89 欧姆，电感为 0.62mH，编码器线数为 1000 线，减速器减速比为 40。

驱动器控制电机时，具有 4 种控制模式：指令脉冲+方向位置控制模式，正交指令脉冲位置控制模式，速度控制模式，Modbus 控制模式。

驱动器控制模式为脉冲+方向位置控制模式时，驱动器控制引脚选择 signal 部分的 PU 和 DIR，参数设定值为指令脉冲+方向。指令脉冲数=电机运行步数*电子齿轮，指令脉冲频率=(电机运行的转速/60)*电子齿轮*编码器线速。有一点需要特别注意，使用这种方式控制电机时，在改变 DIR 时，应该提前脉冲 5us 以上。

驱动器控制模式为正交指令位置控制模式时，驱动器控制引脚选择 signal 部分的 PU 和 DIR。图 4-2 是这种方式控制电机的 PU 金额 DIR 的波形。在此种模式下，指令脉冲数=PU 脉冲的个数。

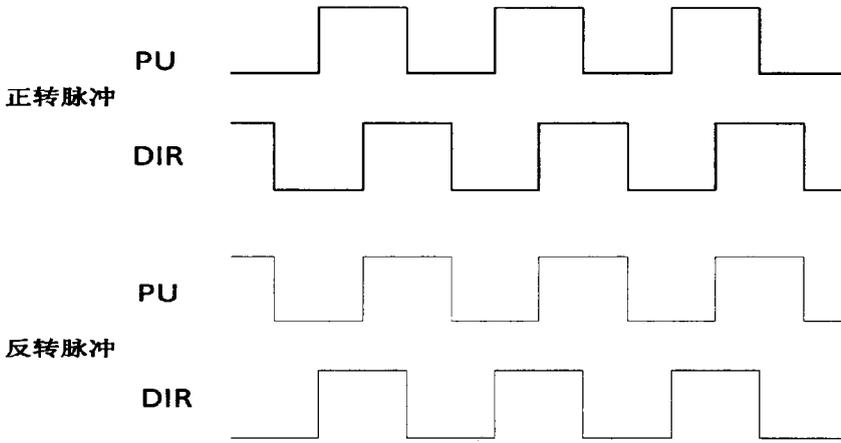


图 4-2 正交指令位置控制模式脉冲波形

对于正反转，图中可以看到，PU 上升沿超前 DIR 上升沿为正转，反之为反转。

驱动器控制模式为速度控制模式时，驱动器控制引脚选择 signal 部分的 PU 和 DIR。图 4-3 是这种方式控制电机的 PU 金额 DIR 的波形。在此种模式下，通过 PU 的脉冲占空比来控制转速，占空比转速范围为 10%~90%，代表 0~3000R/min，PU 的频率为 1KHZ（误差范围 10%），PU 占空比=(目标转速/3000)*80%+10%。

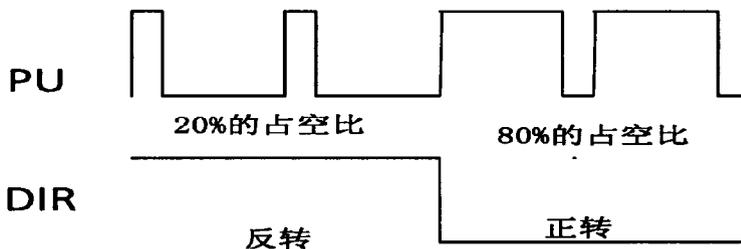


图 4-3 速度模式控制脉冲波形

驱动器控制模式为正交指令位置控制模式时，驱动器控制引脚选择 RS485 或串口 RS232。在

这种模式下，根据电机驱动器的 Modbus RTU 指令发送相应的命令控制电机，具体的指令在第五章中。

4.2 机器人控制系统主控制器硬件电路设计

4.2.1 电源电路

直流伺服电机驱动器采用直流 24V 电压供电，24V 电压由蓄电池提供，但是主控制器采用 3.3 直流电压供电，因此需要设计加电电路为主控制器供电。直流电压加电芯片采用 LM2596 开关调节器，开关频率为 150KHz^[37]，其既具有可调节电压输出，也具有具有固定的 3.3V、5V、12V 输出；其芯片内部集成了固定频率发生器；其工作环境温度-40 度~125 度；输出电流可达 3A，输入电压可达 40V，低功耗待机模式，可以用仅 80uA 的待机电流；具有 TTL 断电能力，过热保护盒限流保护能力；建立降压电路非常简单，只需要 4 个器件。

如图 4-4 所示，机器人控制系统电源电路，采用 LM2596 芯片，固定输出 5V 和 3.3V，为系统各模块供电。

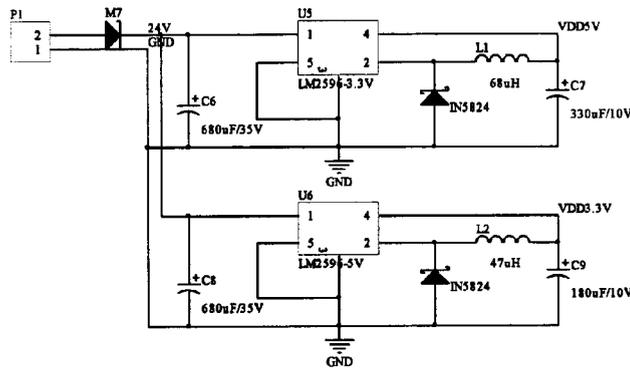


图 4-4 电源电路

4.2.2 时钟电路与复位电路

RSB4210 芯片集成了 RTC 和 PLL 的时钟发生器。RTC 是系统时钟，需要外接 32.768KHz 晶振。PLL 利用外部输入的参考信号控制环路内部振荡信号的频率和相位，RSB4210 支持三种 PLL：为 ARM 设备提供 ARMCLK 时钟的 APLL、为所有 AXI/AHB/APB 相连模块提供 HCLK 或 PCLK 时钟的 MPLL、为特殊的设备提供时钟的 EPLL。复位电路采用的是芯片 MAX811TEUS，具体时钟和复位电路电路如图 4-5 所示。

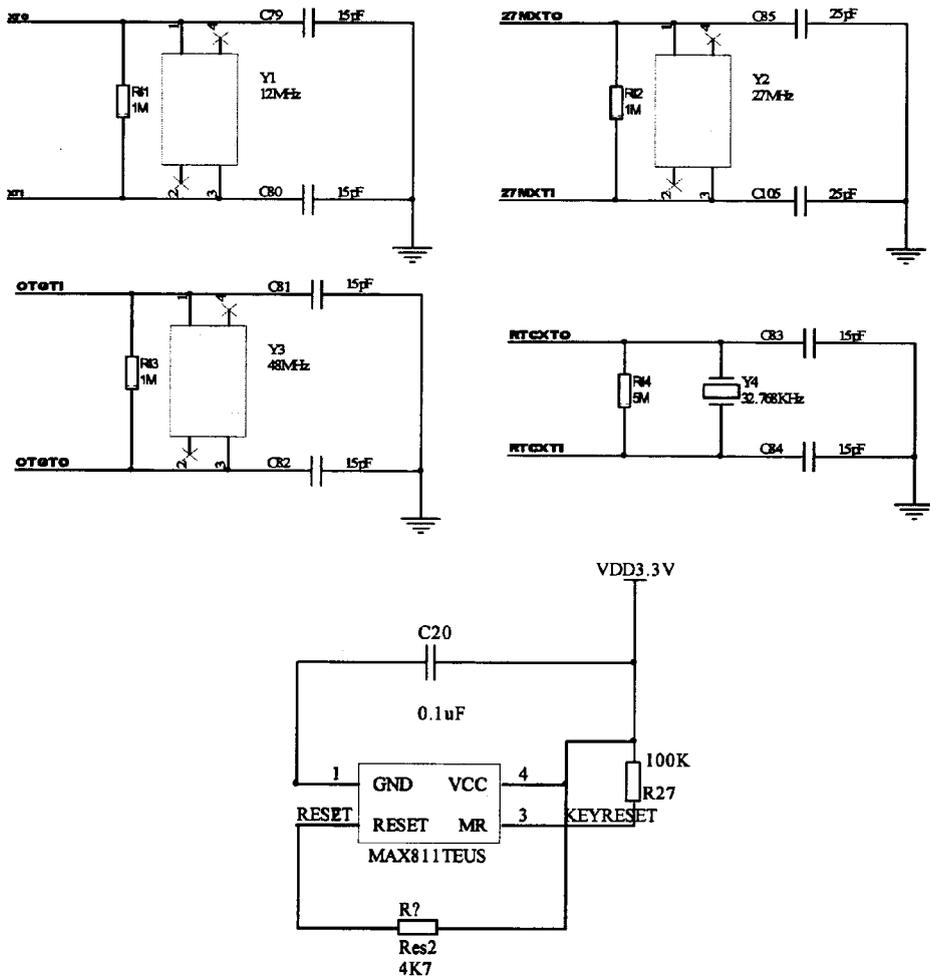


图 4-5 时钟与复位电路

4.2.3 网卡电路

RSB4210 开发板采用的网卡芯片是 DM9000，其不需要另接网络变压器，因为其 RJ45 接头内部已经包含了耦合线圈，因此，使用普通的网线即可连接本开发板至路由器或交换机，其可以自适应 10/100M 以太网，DM9000 是一款以太网 MAC 控制器，其具有低功耗和高性能的特点，其支持 8 位、16 位和 32 位宽度接口的内部存储器，支持不同的处理器的，支持 IEEE802.3x 全双工流量控制。

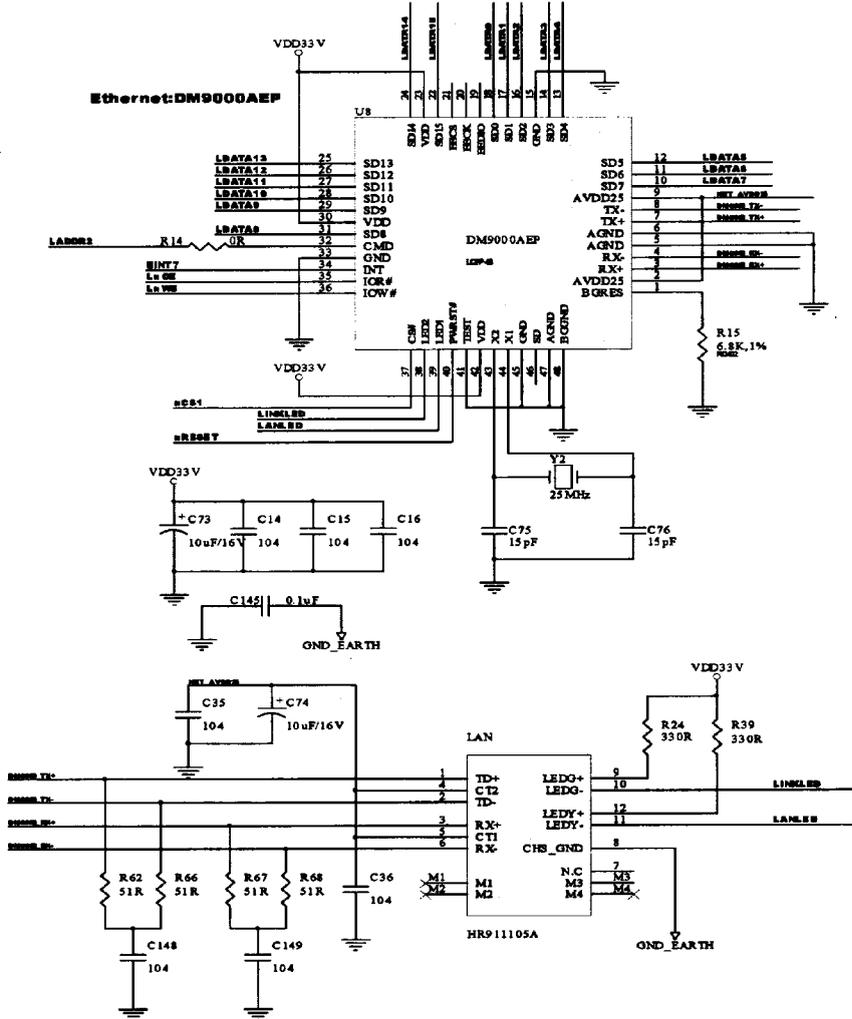


图 4-6 网卡电路

4.2.4 SD 卡接口电路

本次主控制器设计在操作系统移植时，必须将操作系统编译拷贝到 SD 卡，然后系统从 SD 卡启动，另外，SD 卡用于记录控制机器人电机的相关参数。如图 4-7，SD 卡接口电路很简单，其通信采用 SPI 总线。

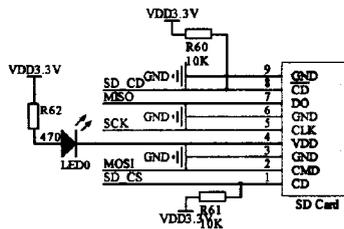


图 4-7 SD 卡接口电路

4.2.5 串口电路

如图 4-8 串口电路，主控制器具有 5 个串口，一个 DB9 针的 RS232 接口，两个三线 RS232 串口，一个三线 TTL 串口和一个 RS485 串口。TTL 转 RS232 电平的芯片采用 MAX3232CSE，TTI 转 RS485 电平采用 MAX485 芯片。5 个串口中，COM1 用于和电机驱动器通信，COM2 用于接收上位机发送机器人控制指令，COM4 用于接收所有传感器数据采集器发送的传感信息，COM5 用于系统调试，剩余串口做为扩展使用。

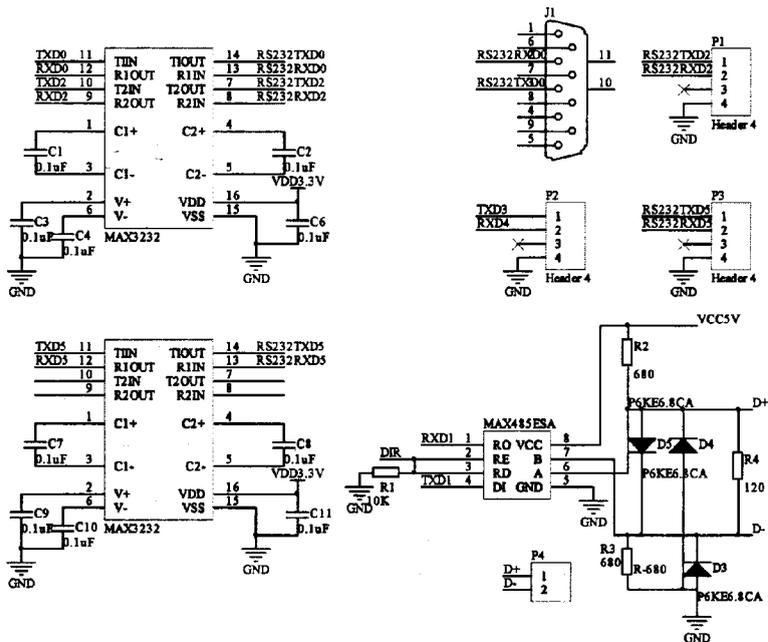


图 4-8 串口电路

4.2.6 直流伺服电机驱动器接口电路设计

直流伺服电机驱动器具有多种控制方式，但是使用 RS485 接口和串口 RS232 接口，采用 Modbus RTU 协议方便控制，接口电路简单。由于机器人采用四驱独立控制，因此主机和四个直流电机组建主存结构，且一个主机，多个从机，所以采用 RS485 接口。由于在实际编程测试中，主控制器采用 RS485 接口在发送数据后，立即接收数据，但是接收不到数据，原因在于主控制器的 RS485 接口发送数据和接收数据切换时，时间比较长，在此切换时间内，从机返回的数据已近返回，所以采用主控制器的三线 RS232 接口，将 RS232 电平转化为 RS485，再和四个电机组网，实现通信。图 4-9 是 RS232 转 RS485 电平电路，图 4-10 是实际的电机驱动器接口电路。

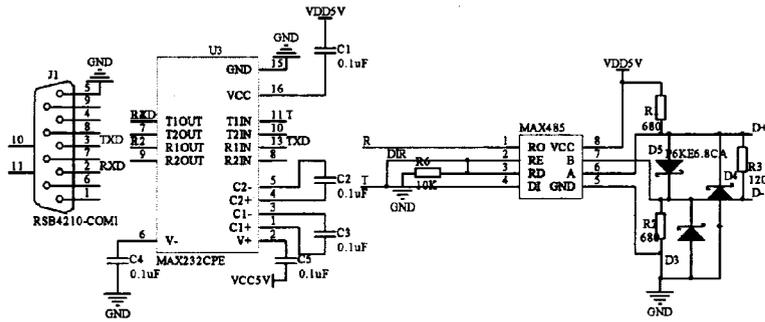


图 4-9 RS232 转 RS485 电平电路

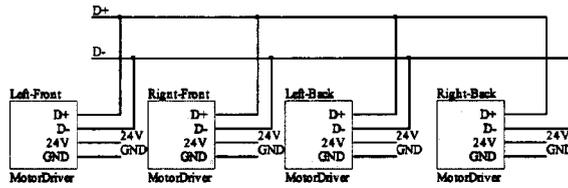


图 4-10 直流伺服电机驱动器接口电路

4.3 直流伺服电机驱动器电路设计

4.3.1 直流伺服电机驱动器电源电路

驱动器采用蓄电池的 24V 直流电源供电，其中一部分用于降压产生 3.3V 电源和 15V 电源，另一部分向三相桥式逆变电路提供 24V 直流电源。如图 4-11 所示，电源电路中采用 LM2596 开关电源芯片将 24V 直流电源降压为 5V，又利用 REG1117 芯片将 5V 电源转换为 3.3V 电源，为驱动器控制其提供电源，同时采用 MC78M15 芯片将 24V 直流电源转换为 15V 电源为功率驱动电路功率管提供电源。

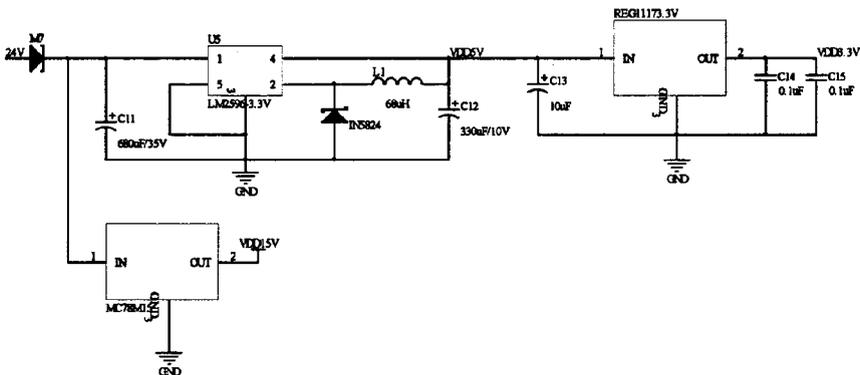


图 4-11 直流伺服电机驱动器电源电路

4.3.2 直流伺服电机驱动器控制器外围电路

如图 4-12 所示，直流伺服电机驱动器控制器外围电路主要包括时钟电路，电机霍尔信号接口和脉冲编码器接口，其余则是电机功率驱动电路和控制器的接口。主控制采用 TI 公司的 TMS320 DSP 处理器，其采用哈佛结构，具有专用的 DSP 指令和专用的硬件乘法器，使其可实现快速的 DSP 运算，其运行时钟采用 100M 晶振。

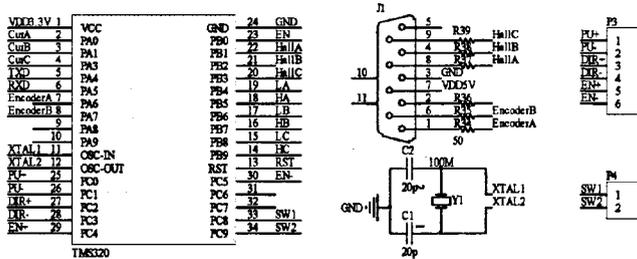


图 4-12 直流伺服电机驱动器控制器外围电路

4.3.3 直流伺服电机功率驱动电路

如图 4-13 所示，直流伺服电机功率驱动电路芯片采用 IR2136，电机采用三相星型连接，逆变电路采用 MOSFET IRF540。IR2136 芯片采用 15V 电源供电，其内部具有光耦隔离电路，支持高频工作，死区时间为 250ns，一般情况下开关频率为 400ns。另外，此芯片封装 6 个通道的传播延迟可相互配合，确保稳定的高频工作，从而在低速下，发挥完善电机的转矩性能，并且可以降低噪声。

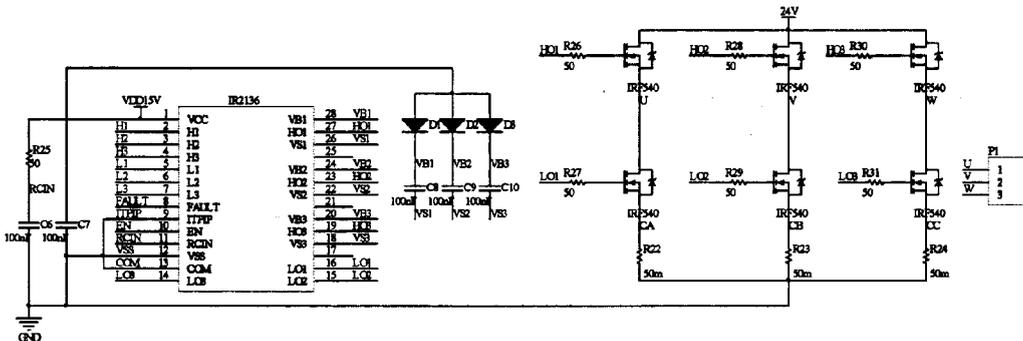


图 4-13 直流伺服电机功率驱动电路

4.3.4 电机电枢电流采集放大电路

在直流伺服电机驱动器控制中，必须实时采集电枢电流的大小，由于电机驱动控制器的 AD 转换器是电压型转换器，不能直接采集采集电流的大小，本次利用母线上的 100 毫欧的电阻采集电压，将电压放大采集，再变换为相应的电流大小。因为具有三相线圈，所以要采集三路电流，电流采集放大电路也需要三路，其放大倍数为 4.7。

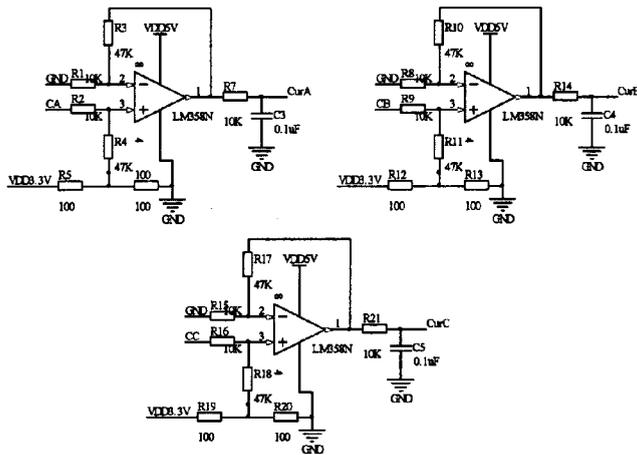


图 4-14 电机电枢电流采集放大电路

4.3.5 驱动器串口通信电路

实际控制电机采用 Modbus 指令控制模式，既可以采用 RS485 总线通信，也可以采用 RS232 通信，但是组网时，只能采用 RS485。如图 4-15 图所示，RS485 接口采用 MAX485 芯片做为收发器，RS232 接口采用 MAX232 芯片，图中的二极管用于 RS485 端口保护电路，其耐压为 6.8V。

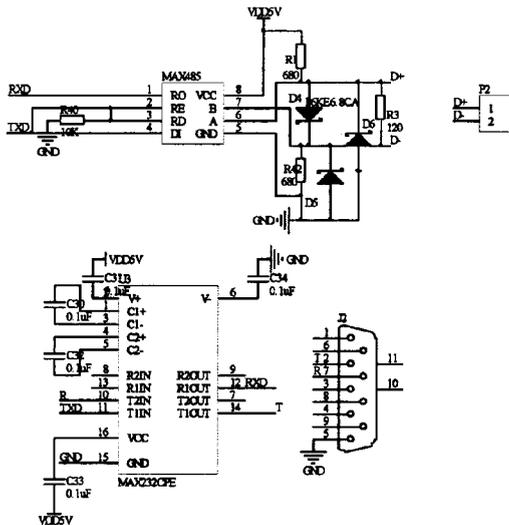


图 4-15 驱动器串口通信电路

4.4 本章小结

本章主要设计机器人控制系统主要硬件电路。整个系统硬件有主控制器、直流伺服电机、电机驱动器；主要接电路的设计包括电源电路的设计，时钟和复位电路设计，网卡电路设计，SD 卡接口电路设计，串口电路设计，直流伺服电机驱动器接口电路和直流伺服电机驱动器电路设计。

第 5 章 沙漠信息感知机器人运动控制系统软件设计

5.1 嵌入式操作系统

目前主流的嵌入式操作系统 VxWorks、RT-Linux、Linux、Android、uC/OS-II、WindowsCE 中, Linux 操作系统是一个稳定和普遍的, 功能强大, 支持多平台操作系统, 支持 TCP/IP 协议; 网络功能是非常强大的, 支持多用户和多进程; 其网络协议内置了丰富的免费的 Web 服务器软件, 数据库和 Web 开发工具; Linux 可以提供全方位的网络服务。Linux 操作系统具有良好的安全性和稳定性, Linux 操作系统本身具有良好的安全性能, 其开源, 漏洞很快就会被发现和完美; Linux 操作系统, 对硬件的要求比较低, 而且成本低^[38]。

5.1.1 嵌入式 Linux 操作系统

Linux 系统是一种自由和开放源代码的类 Unix 操作系统, 其是一个支持多线程、多用户、多任务的操作系统, 同时其也是基于 UNIX 和 POSIX 的操作系统; Linux 继承了 Unix 系统的特性, 其设计思想是以网络为整个核心, 其能够运行大部分 UNIX 的网络协议、工具软件 and 应用程序; Linux 系统也能支持多硬件的系统, 其同时能够支持 32 位和 64 位硬件设备; Linux 做为一个多用户网络操作系统, 且性能非常稳定; Linux 操作系统也是全世界各地的成千上万的程序员努力一起创新劳动的结晶, 其是以建立一个不受版本制约及商品软件的限制, 而且让全世界能够自由使用的 Unix 兼容产品为主要目的, 虽然存在着很多不同的 Linux 系统, 但是它们的本质都是使用的是 Linux 内核。大部分计算机硬件设备都安装了 Linux 系统, 比如手机、平板电脑、路由器、大型机和超级计算机等。Linux 操作系统运行速度是非常快的, 在世界上有运算最快的 10 台超级计算机运行的都是 Linux 系统。Linux 这个词本身从严格的意义上只表示 Linux 内核, 但实际上人们习惯了用 Linux 来形容整个基于 Linux 系统, 包括了数据库和 GNU 工程应用各种工具的操作系统。高效性和灵活性是 Linux 系统所具有的优势, 其设计结构采样了模块化的设计, 使得其既可以在价格相对高的工作站运行使用, 也可以在配置相对较低的一半计算机上实现具有多任务、多用户的能力的 Linux 操作系统。我们可以在 GNU 公共许可权下, 获得免费且符合了 POSIX 标准的 Linux 的操作系统。我们获得的 Linux 系统软件包, 不仅包括了完整的 Linux 操作系统, 而且还包括了很多工具软件, 如: VI 文本编辑器、GCC 编译器及 GDB 调试器等应用软件及具有 XWindows 图形用户界面多窗口管理器, 就和我们使用微软公司的 Windows 系统一样, 我们可以使用很多图形界面的操作, 如: 菜单、窗口图标^[39]。

5.1.2 交叉编译环境建立

由于 PC 机和主控制器的 ARM 是两个不同的架构平台, 它们具有不同指令集。在 PC 机上用 gcc 编译的可执行文件, 在 ARM 平台无法运行, 因此, 目前建立的嵌入式 Linux 开发模式是宿主

机-目标机开发模式，即宿主机为 PC 机，ARM 为目标机，目标机上的运行程序在宿主主机上编译好，拷贝到目标机上运行，但是 gcc 编译的程序不能在 ARM 平台运行，因此需要在宿主主机上面建立交叉编译工具 arm-linux-gcc，用此工具编译的程序才能在 ARM 平台运行。下面的步骤是在宿主主机上建立交叉编译工具的过程。

(1)切换目录 `cd /opt`;

(2)解压文件 `tar xvzf arm-linux-gcc-4-4-3-tar-gz`;

编辑文件，修改环境变量 `vim /etc/environment`;

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/opt/cross_compiler/arm-fsl-linux-gnueabi/bin"
```

(在此文件里面加入 `:/opt/cross_compiler/arm-fsl-linux-gnueabi/bin`)

输入命令是环境变量生效 `source environment`;

输入命令测试交叉编译工具 `arm-linux-gcc-v`，如果显示版本，证明配置成功，之后，可以用此工具编译在 ARM 平台运行的程序。

在宿主主机上使用 `arm-linux-gcc` 编译程序。编译好程序之后，利用网络共享文件 (NFS)，将程序拷贝到 ARM 控制板运行。

5.1.3 主控制器 Linux 系统 Flash 启动制作

主控制器 Linux 系统 Flash 制作的前提条件是嵌入式 Linux 系统、Uboot 及文件系统已近制作好，由于这个过程比较复杂，提前已近将这些工作完成，在这里不必赘述，只将系统 Flash 启动制作阐述。

本此使用的主控制器 Flash 的 Linux 系统制作首先要将系统烧写带 SD 卡上，然后主控制启动默认 SD 卡启动，系统启动之后，再将 SD 卡上的系统烧写到主控制器的 Flash 上面。对于 SD 卡的 Linux 系统的烧写，首先明白 Linux 系统的分区格式。对于 Linux 系统，文件系统分区分为两个部分：FAT32、ext3，系统存储介质的第一部分，也是系统启动部分，是 Uboot 和 Linux 系统内核，它们是 FAT32 的文件格式，之后的第二部分是根文件系统，其文件格式是 ext3。因此，制作 SD (最大 8G) 卡启动时，首先将 SD 卡分区：boot 分区和 file 分区，然后将 uboot 和 linux 系统内核拷贝到 boot 分区，然后将根文件系统拷贝到 file 分区，下面是详细的步骤。

(1)将 SD 卡插入 Linux 宿主机，打开宿主机终端，切入到 root 用户；

(2)输入命解压文件 `tar xvf RSB4210_Linux_image-tar-gz`;

(3)输入命令切换目录 `cd /home/dong3936533/Desktop/RSB-4210_Linux_image`;

(4)运行系统烧写脚本 (提前写好的 shell 脚本，用于系统分区和文件拷贝，具体参考附件)

```
~/mkmmc-linux-sh /dev/sda u-boot-bin uImage rootfs;
```

(5)输入 y，等待几分钟，SD 卡系统烧写结束；

(6)将 SD 卡插入到控制器，主控制器上电，系统启动 (默认 root 用户)；

(7)输入命令切换目录 `cd /mk_inand`;

(8)运行脚本文件 `sh mkmmc-linux-sh /dev/mmcbk0 u-boot-bin uImage rootfs-tar-gz`;

(9)输入 y，等待几分钟，控制器 Flash 系统烧写结束，之后移除 SD 卡，重新上电，系统从

Flash 启动。

经过上面的步骤之后，控制器 Linux 系统 Flash 启动制作完成，其中制作过程中的几个 shell 脚本都是提前写好，这里没有列出。

5.2 伺服电机驱动器 Modbus 协议

5.2.1 Modbus 协议

Modbus 协议广泛应用于工业控制中，是一种工业上标准的成熟协议^[40]。控制器利用自己组建的网络，和其它控制器或者其它设备之间，使用此协议通信。实际控制中，各厂商将控制设备组建成工业网络，利用 Modbus 协议，对设备进行监视控制。Modbus 协议描述的是一个过程，主控制器向其它设备发车请求，对应的设备如果收到相应的请求消息，会向主控制器返回相应的数据，主控制器根据返回的信息，判断请求是否成功。所有设备在通信时，都利用统一的数据格式，设备之间的设备是利用消息中的地址域，不同的设备具有不同的地址，所以主控制必须知道所有设备的地址，通过地址设备发来的消息，由此决定下一步的动作。至于，此协议的消息帧在什么网络上传输，就转换为这种网络的要求的消息帧结构。

(1) Modbus 协议传输方式

标准的 Modbus 协议有两种传输模式：ASCII 模式和 RTU 模式。不管选择什么模式都需要设置串口通信的参数，但是当在网络上通信时，每一个控制器在网络上具有相同的传输模式和串口参数。对于 ASCII 模式，采用的是 ASCII 码传输，消息的每一字节采用 2 字节 ASCII 码传输，且采用的是 LRC 校验。而对于 RTU 模式，消息的一个字节就是一个字节传输，校验采用的 CRC 校验。

(2) Modbus 协议消息帧

不管哪种传输模式，Modbus 的消息帧都有起始标志和结束标志，所以，各种接收设备根据起始标志，接收消息，根据结束标志停止接收消息。接收到消息后，读取地址信息，判断是否被选中，如果被选中，接收并且校验数据，最后返回相应请求的数据。

ASCII 模式中，Modbus 消息帧以“:”符为起始标志，以回车换行符为结束标志，消息帧的其它部分使用十六进制数。在网络的各设备不断检测“:”，如果接收到这个字符，然后解析地址域，看是否是自己的地址，如果是自己的地址，则接收消息，如果不是，则不接收后面的数据。一个典型消息帧结构如表 5-1 所示：

表 5-1 Modbus ASCII 消息帧

起始位 (字符)	设备地址 (字符)	功能代码 (字符)	数据 (字符)	LRC 校验 (字符)	结束符 (字符)
1	2	2	n	2	2

RTU 模式中，Modbus 消息帧以 3-5 个字符时间的停顿间隔为消息起始标志，因此消息帧中的第一个域就是地址域。在网络的各设备不断检测，当判断到接收到的第一个域为自己的地址时，再继续接收其它字符。当字符发送结束之后，停顿 3-5 个字符时间的停顿间隔为消息结束标志。RTU 模式典型的消息帧如表 5-2 所示：

表 5-2 ModbusRTU 消息帧

起始位 (字符)	设备地址 (字符)	功能代码 (字符)	数据 (字符)	CRC 校验 (字符)	结束符 (字符)
3-5	8	8	n	2	3-5

Modbus 协议的 ASCII 模式的地址域是两个字符，而 RTU 模式的地址域是 8Bit。设备的地址从 0 到 247，其中 0 地址为广播地址，所有设备都会识别此地址，因此单个设备的地址只能是 1 到 247。主控制器想要联系从设备，就将该设备的地址放入消息帧，从地址收到消息好，会将自己的地址放入消息帧，把主控制器请求的消息发送到主控制器，主控制器通过此地址识别是哪个从设备返回的数据。

Modbus 协议的 ASCII 模式的功能域是两个字符，而 RTU 模式的功能域是 8Bit。这两种模式的代码的取值范围为 1 到 255，但是具体的代码表示什么功能，就要看具体的应用，一般功能码 0x06 用于读寄存器，0x03 用于写寄存器。标准的协议定义 21 种功能代码，包括：读线圈状态、读输入状态、读保持寄存器、读输入寄存器、强制单线圈、预制单寄存器、读取异常状态、回送诊断校验、读取事件计数、读取通信事件记录、强置多线圈、预置多寄存器等功能。主控制器过功能代码通知从机发生什么时间，从设备受到消息后，使用功能代码通知主控制器是否发生错误，如果没有发生错误，则从机正常相应请求的功能，反之，返回异常代码。

数据域以字节为单位，当选择 ASCII 模式时，其以 ASCII 字符为单位，当选择 RTU 模式时，以单个数据的 16 进制字节为单位。对于数据域的具体内容，其随着功能码的不同发生变化，当要读寄存器时，数据域包括寄存器的起始地址和寄存器的数量，当要写寄存器时，数据域包括写寄存器的起始地址、寄存器的数量和写入寄存器的数值。如果传输发送错误，返回的数据域是发生错误类型的代码，如果没有发生错误，数据包括所请求的数据。

标准的 Modbus 网络具有两种错误检测的方法，当在 ASCII 模式时，使用 LRC 校验，而在 RTU 模式时，使用 CRC 校验。一般也可以采用奇偶校验，但奇偶校验对每个字符都可用，但是对于 ASCII 和 RTU 模式，LRC 与 CRC 校验，只对消息域校验，不对起始标志和结束标志校验。当主控制发送消息时，将消息内容校验，将校验后的校验码添加到消息后面，组成一个消息帧发送出去，相应的从设备收到消息帧，会先对消息进行同样的校验，如果产生的校验码和收到的校验码一致时，则证明消息帧传输无错误，但是当不一致时，从设备就不会返回数据，此时主控制器等待一段时间如果收不到返回数据，则证明发送的数据传输错误，则进行超时重传。

奇偶校验相对于 LRC 和 CRC 校验，就特别简单，ASCII 模式时，数据位只有 7 位，RTU 模式中数据位 8 位。所谓奇校验就是数据的二进制代码中 1 的个数为奇数，偶校验就是数据的二进制代码中的 1 的个数为偶数。除此之外，用户也可以配置无奇偶校验位。

LRC 校验用于 ASCII 模式，ASCII 模式的消息帧包括了错误检测域，当进行校验时，除了消息的“:”起始标志和回车换行的结束标志，其余地址域、功能域、数据域全部进行校验，将校验的结果添加到结束标志的前面。主控制器发出消息帧后，接收设备接收到消息后，会对消息进行 LRC 检测，将检测得到的校验码和接收到的代码比较，如果以样，说明接收正常，反之，则发生错误，特别注意一点，LRC 校验是将消息中的各字节求和，丢弃进位，最后得出结果。

CRC 校验用于 RTU 模式，RTU 模式的消息帧包括了错误检测域，其对整个消息帧进行检测，最后产生一个 16 位的二进制数，添加的消息的最后面，发送出去，接收设备收到消息，对消息

同样进行相同的 CRC 检测, 将产生的校验码和接收到的校验码比较, 如果相等, 则证明消息正确, 反之, 发生错误。CRC 的校验的原理比较复杂, 其原理是先用一个 16 位寄存器的值是 16 个 1, 将消息帧从起始的每一个字节与 16 为寄存器相或, 结果最高位填充 0, 其余向有效方向移动, 把最低位取出来, 判断是否为 1, 如果为 1, 则寄存器和预置的值相或, 反之, 不进行操作。一共重复 8 次, 当最后一个字节完成之后, 则去下一个字节单独和寄存器当前的值相或, 最后, 寄存器的值是产生的 CRC 校验码, 将 CRC 校验码的低字节取出来加入消息帧, 后把高字节取出来放到后面。

5.2.2 伺服驱动器 Modbus 协议

对于此次采用的伺服电机驱动器可以通过 Modbus (RTU 模式) 来控制驱动器。

表 5-3 伺服驱动器 Modbus 协议寄存器说明表

地址	参数名称	读写	参数范围	参数说明
0	Modbus 使能	读写	0~1	0: Modbus 禁止 1: Modbus 使能
1	驱动器输出使能	读写	0~1	0: Modbus 禁止 1: Modbus 使能
2	电机目标速度	读写	0~3000r/min	速度模式: 目标速度 位置模式: 最大速度
3	电机加速度	读写	0~30000r/min/s	速度模式: 加速度 位置模式: 位置缓存
4	电机起始速度	读写	0~500r/min	位置模式: 最小速度
5	速度环比例系数	读写	0~10000	代表 0~10-0
6	速度环积分时间	读写	2~2000ms	积分时间 2~2000ms
7	位置换比例系数	读写	60~5000	位置模式速度下降速率
8	速度前馈	读写	0~8-0V/KRPM	速度前馈 1KRPM 对应电压幅值
9	DIR 极性	读写	0~1	0: 外部 DIR 不导通顺时针旋转 1: 外部 DIR 导通顺时针旋转
10	电子齿轮分子	读写	0~65536	16 位电子齿轮分子
11	电子齿轮分母	读写	1~65536	16 位电子齿轮分母
12	目标位置低十六位	只读	-	需要走步数的高 16 位
13	目标位置高 16 位	只读	-	需要走步数的低 16 位
14	报警代码	只读	-	-
15	系统电流	只读	0~32767	实际电流为 X/2000(A)
16	电机当前速度	只读	-3000~3000r/min	实际电机转速
17	系统电压	只读	0~32767	实际电压 x/327(V)
18	系统温度	只读	0~100	摄氏度
19	系统输出的 PWM	只读	-32768~32767	代表-100%~100%
20	参数保存标志	读写	0~1	0: 参数未保存 1: 保存参数中 2: 保存完毕
21	设备地址	只读	0~255	设备地址

表 5-4 主机读取数据格式

设备地址 (字节)	功能码 (字节)	第一寄存 器高地址 (字节)	第一寄存 器低地址 (字节)	寄存 器高位 (字节)	寄存 器低位 (字节)	CRC 高位 (字节)	CRC 低位 (字节)
1	1	1	1	1	1	1	1

表 5-5 从机应答主机读取数据格式

设备地址 (字节)	功能码 (字节)	字节数 (字节)	数据高字节 (字节)	数据低节 (字节)	CRC 高位 (字节)	CRC 低位 (字节)
1	1	1	1	1	1	1

表 5-6 主机写数据格式

设备地址 (字节)	功能码 (字节)	第一个寄存 器的高地址 (字节)	第一个寄存 器的低地址 (字节)	数据高位 (字节)	数据低位 (字节)	CRC 高位 (字节)	CRC 低位 (字节)
1	1	1	1	1	1	1	1

表 5-7 从机应答主机写数据格式

设备地址 (字节)	功能码 (字节)	第一个寄存 器的高地址 (字节)	第一个寄存 器的低地址 (字节)	数据高位 (字节)	数据低位 (字节)	CRC 高位 (字节)	CRC 低位 (字节)
1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节

表 5-8 主机特殊功能码 0x78 格式

设备地址 (字节)	功能码 (字节)	PU:24~31 位(字节)	PU:16~23 位(字节)	PU:8~15 位(字节)	PU:0~7 位 (字节)	CRC 高位 (字节)	CRC 低位 (字节)
1 字节	0x78	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节

表 5-9 主机特殊功能码 0x7a

设备地址 (字节)	功能码 (字节)	无效 (字节)	无效 (字节)	无效 (字节)	新设备 的地址 (字节)	CRC 高位 (字节)	CRC 低位 (字节)
1 字节	0x7a	0x00	0x00	0x00	1 字节	1 字节	1 字节

主机可以通过 modbus 的读写寄存器功能来设置驱动器参数和控制运行。驱动器支持的功能码为 0x03 (读寄存器)、0x06 (写寄存器)、0x78 (写目标位置)、0x7a (修改设备地址)。表 5-3 为伺服驱动器 Modbus 协议寄存器说明表。

5.3 Linux 串口通信编程

由于本次主控制器使用 Linux 的 Ubuntu 系统,且此系统是经过裁剪,在此系统文件系统中,对于串口设备文件为/dev 目录下的 ttymxc0、ttymxc1、ttymxc2、ttymxc3、ttymxc4。在 Linux 系统下,通过使用 read()、write()函数来实现对串口的读写,但是在使用串口之前需要对串口的其他参数进行配置。

串口编程流程,首先在使用串口之前必须打开串口然后对串口进行设置,串口位于 linux 文

件系统目录/dev/下，打开串口是使用函数为 open()函数，fcntl()、isatty()、set_opt()这三个函数用于设置串口设备的文件一些标志或者属性。

串口的设置主要是对结构体 termios 的各个成员值进行设置，主要包括波特率、数据位停止位等设置。

该结构中 c_cflag 最为重要，通过它可以设置串口通信的数据位、波特率、停止位、校验位。使用其需通过与或操作方式。常用的串口编程函数函数：

Tcgetattr 取属性(termios 结构)

Tcsetattr 设置属性(termios 结构)

cfgetispeed 得到输入速度

Cfgetospeed 得到输出速度

Cfsetispeed 设置输入速度

Cfsetospeed 设置输出速度

波特率的设置是使用函数 cfsetispeed、cfsetospeed。

```
cfsetspeed(&uartio, B4800);
```

```
cfsetoutspeed(&uartio, B4800);
```

数据位的设置：

```
newtio.c_cflag |= CLOCAL |CREAD;
```

```
newtio.c_cflag &= ~CSIZE;
```

在 Linux 下串口的读写和普通文件一样，使用 read，write 函数，在确保串口打开和正确设置的情况下，下列函数可以实现将 buff 数组中长度为 ByteCount 的数据通过串口发送出去：

```
nwrite = write(fd1, buff, ByteCount);
```

使用 read()函数接收数据，read()函数返回的字符数是实际串口收到的字符数。函数的使用方法如下：

```
n = read(fd0, Uart0_Receive_Buff, MAX);
```

串口使用完毕之后，可以通过 close(fd0)函数将串口关闭。

5.4 机器人运动控制程序设计

5.4.1 机器人运动控制程序总体设计

本次设计所设计的机器人是四轮独立驱动，因此对于机器人的控制就是对四个直流伺服电机进行控制，而电机的控制，总线采用 RS485 总线，协议采用 ModbusRTU 协议。采用的主控制板的是研华科技的 iMX53，其是一款以 CortexA8 为内核的 ARM 处理器，整个控制程序的设计就是采用主控制器的两个串口进行控制，一个 RS232 串口用于连接上位机，接收上位机发送的机器人动作控制指令，另一个 RS485 串口用于向四个直流伺服电机发送控制指令。控制流程图如图 5-1 所示。

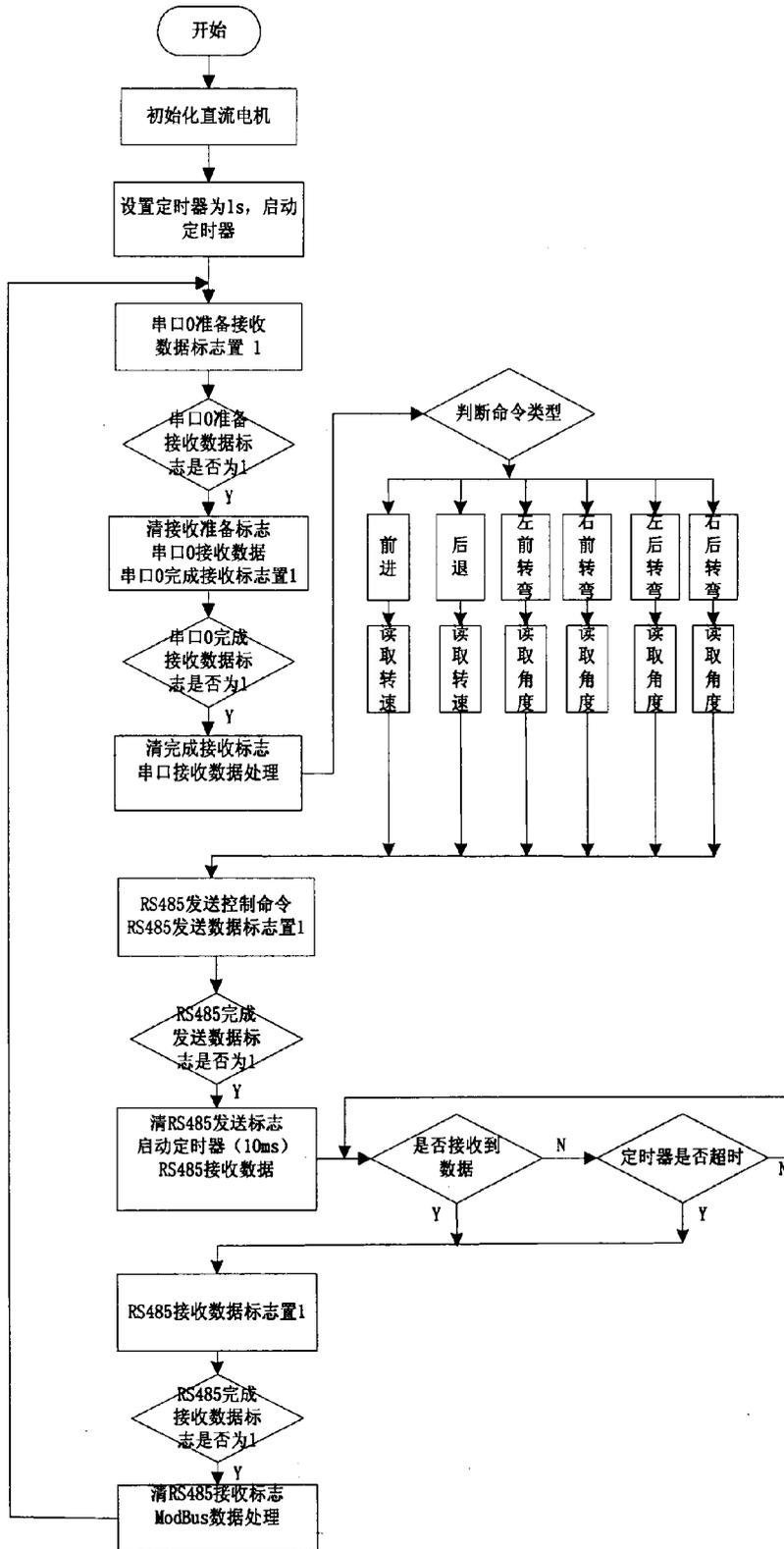


图 5-1 系统控制流程图

系统首先初始化四个直流伺服电机和定时器,执行 Init_Car()函数初始化电机,执行 SetTimer()初始化定时器。定时器用来定时检查电机是否出现故障报警和每个电机的电流。如果出现故障报警,检查报警原因并处理报警。然后,主控制器等待接收机器人控制指令,收到指令,执行相应的动作,执行 Receive_Dispose()函数。

对于 Receive_Dispose() 函数的编程,其是一个状态机,对于连个串口分别定义 Uart0_Receive_Ready_Flag、Uart0_Receive_Flag、Rs485_Receive_Flag、Rs485_Send_Flag 四个标致位实现状态机编程,其优点在于是系统有条不紊的执行。

5.4.2 机器人控制指令及数据格式

本设计所设计的机器人具有 7 个动作,分别是:前进、后退、停止左前转弯、右前转弯、左后转弯、右后转弯。接收指令的串口的缓冲区的大小为 8 字节,传输指令使用的 ASCII 码传输,不同的控制动作对应不同的指令格式,具体的指令格式如表 5-10。

表 5-10 机器人控制指令集

功能(字符)	速度(字符)					角度(字符)			功能说明
1	1	1	1	1	1	1	1	1	前进
2	1	1	1	1	1	1	1	1	后退
3	1	1	1	1	1	1	1	1	左前转弯
4	1	1	1	1	1	1	1	1	右前转弯
5	1	1	1	1	1	1	1	1	左后转弯
6	1	1	1	1	1	1	1	1	右后转弯
7	1	1	1	1	1	1	1	1	停止

对于前进和后退,速度大小为 0~3000 转,需要 4 字节,角度的 4 字节为 0 接收到的数据是字符,需要转换为整数;对于转弯,速度的 4 字节为 0,第 5 字节为 0,其后 3 字节的角度字节,其大小为 0~360 度,同样一样,接收的数据为字符,需要转换为整数;对于停止,速度为 0,角度也为 0。因此,对与程序的编写,主要是控制指令的识和速度或角度的转换,指令的识别采用 switch、case 语句完成,主控制器接收指令处理的函数是 Contrl_Car(const unsigned char *temp)。

5.4.3 机器人直行控制程序设计

本次设计采用的伺服电机驱动器控制电机既具有速度模式,又具有位置模式,但是这两种模式不能通过编程实现自动切换,必须手动切换。对于本次控制机器人,由于采用速度模式控制相对与位置模式比较方便,因此,采用速度模式控制电机。电机正反转时,将相应的速度指令发送给电机,正传时,电机速度为正值,反转时,电机的速度为负值。向电机驱动器发送速度值时,采用 ModbusRTU 协议的写命令,即 6 号命令。控制机器人前进,调用 Car_Forward(int data)函数,在此函数里面调用的是 Motor_Forward(char addr, int data)函数,而在 Motor_Forward 函数里面,调用的是 Write_Register(char addr, char Register_Addr, int data)写函数。

机器人直行运动时,必须保持四个轮胎的车速相同,否则机器人直线行走会发生转弯,因此,只要保持四个电机的转速相同,就可以保证机器人沿直线行走,如图 5-2 所示,机器人直线行走

控制算法程序流程图。

程序中，设计定时器，定时读取四个电机的转速，判断四个电机的转速是否相等，如果不相等则利用平均值法，计算四个电机转速的平均值，然后，给电机发送控制指令，控制四个电机以平均转速运行；如果四个电机的转速相等，则继续判断电机转速是否和上位机发送的控制转速相等，如果相等，计算四个电机转速的平均值，然后，给电机发送控制指令，控制四个电机以平均转速运行，如果不相等，则重新赋值四个电机的转速为 n ，用转速 n 控制电机。如此循环，最终的目的保持四个电机的转速相同，使机器人直线运动。

由于采用电机的最大转速为 3000r/min ，电机减速器减速比为 40 ，机器人的轮胎半径为 265mm ，根据第三章的公式推导，可以得到机器人运行的最大速度为 1.04m/s 。

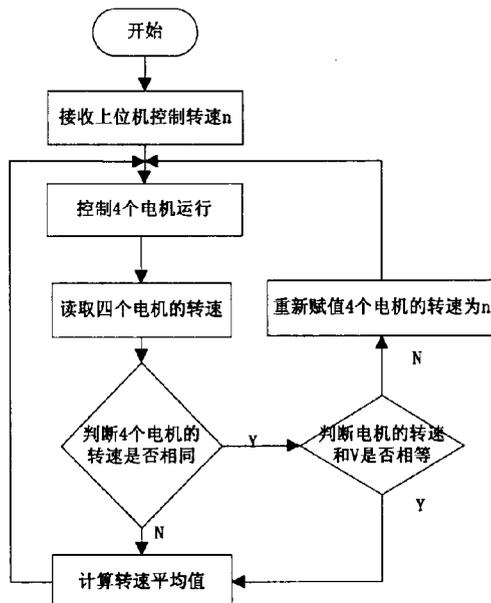


图 5-2 机器人直线行走控制算法程序流程图

5.4.4 机器人差速转弯控制程序设计

机器人转弯具有四个动作：左前转弯、左后转弯、右前转弯、右后转弯。由于本设计所设计的机器人没有专项机构，采用差速方法转弯。这种方法就是利用机器人左侧轮子的速度和右侧轮子的速度之差实现转弯，相对于原地转弯，机器人受到的阻力是比较小，而且转弯的效果也比较好。程序的编写是根据第三章所推导的方法，根据转弯的半径和转弯圆弧圆心角的度数，计算出转弯的时间，程序中只要控制时间，即可实现转弯。

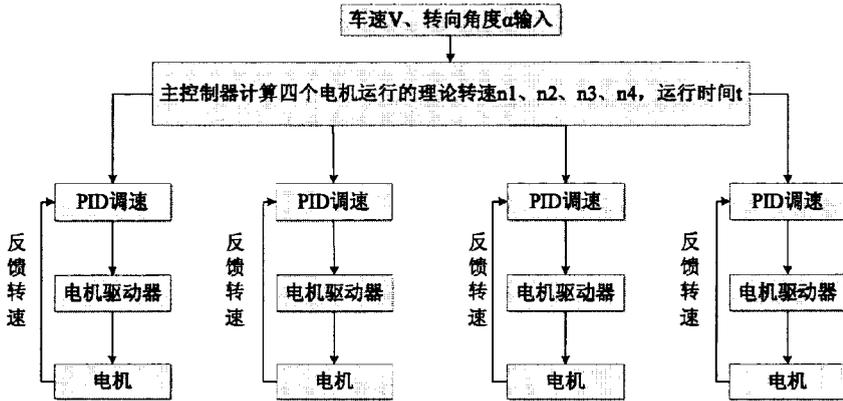


图 5-3 转弯控制结构图

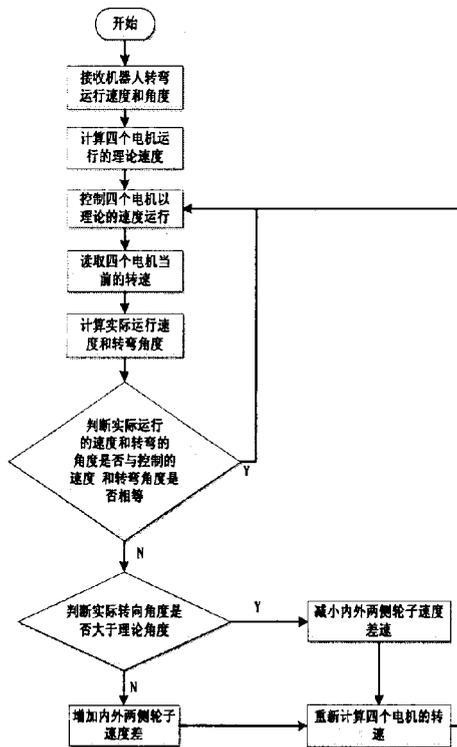


图 5-4 机器人转弯控制程序流程图

如图 5-3 所示，转弯控制结构图，主控制器根据上位机发送的机器人转弯运行的速度和角度，根据第二章机器人转弯推导的公式计算四个电机的理论转速分别为 n_1 、 n_2 、 n_3 、 n_4 ，转弯运行时间为 t ，主控制器根据计算的理论转速和反馈的实际转速，对转速实现调节，然后用调节后的转速控制电机。四个电机转速的调节采用 PID 控制器，使电机实际的转速和理论计算的转速接近，实现机器人平稳转弯。

图 5-4 为机器人转弯控制程序流程图，主控制根据接收到机器人转弯需要的速度和角度，计算四个电机的理论转速，用理论转速控制电机，然后再读取的实际转速，计算实际的机器人运行的速度和转弯角度，判断实际的速度和理论的计算的速度是否相等，如果相等，则控制机器人以

当前速度运行，如果不相等，则继续判断机器人转弯是否过度，如果过度，则减小两侧轮子的速度差，如果不相等，增加内外两侧轮子的速度差，之后重新计算四个电机的转速，用重新计算的转速控制电机运行。机器人转弯运行时间，转弯运行时间结束之后，机器人以设置的最小速度直行。

5.4.5 机器人运行故障监测程序设计

沙漠环境很复杂，机器人在运行会出现很多故障，对于这个环境，不可能人工去检测故障，如何实现自动检测故障、处理故障是非常重要的。本次机器人使用的直流伺服电机，电机驱动器会检测电机的运行状态，开机后，红灯绿灯都亮一次，用于检验 LED 是否工作正常。而后绿亮，红灯灭为正常状态。如果遇到故障，利用 Modbus 读取报警代码，检查故障类型。下表是故障的类型、相应的报警代码及驱动器对故障如何处理。

表 5-5 电机故障类型

报警代码	故障类型	驱动器处理结果
0x10	系统高温(60℃)	继续运行
0x20	写 flash 失败	继续运行
0x11	系统过热(90℃)	停机
0x12	系统堵转	停机
0x13	系统欠压	停机
0x14	失速、负载过重	停机

程序设计中，必须定时监测四个电机的故障，并且处理相应的故障。由于系统堵转、失速、负载过重，会导致驱动器使驱动器使能禁止和 Modbus 使能禁止，因而电机会停止转动。只要重新将驱动器使能和 Modbus 使能允许之后，电机又重新转动；对于系统过热，是由于温度高于 90 度，电机停转，当温度低于 90 度之后，电机会继续转动；对于系统欠压，是由于系统电压低于 20V，电机停转，当电压大于 20V 之后，电机继续运转。

本次设计中，利用 Linux 系统的内核定时器实现定时 1s，在定时器中断服务程序中检查四个电机的状态，检查电机的是否出现故障，如果出现故障，在 Modbus 返回数据中，对相应的故障进行处理。内核定时器数据结构体是 struct itimerval v，初始化定时器就要初始化这个结构体的相应成员变量，然后打开定时器，在中断服务程序进行处理，定时器初始化函数是 Set_On_Timer() 定时器中断服务函数是 sig_handler(int signo)。

5.4.6 机器人运行控制程序设计

根据 2.4 节的运行控制算法，主控制器设置定时器，通过电机 ModBusRTU 协议的 15 号指令，分别读取四个电机当前线圈的电流，比较得出四个电机线圈电流最大一个电机的电流，以这个电流值计算当前需要的转速，然后判断计算得到的转速于电机当前转速进行比较，如果大于当前转速，则增加转速，反之则减小转速。具体程序设计流程下图 5-4 所示。

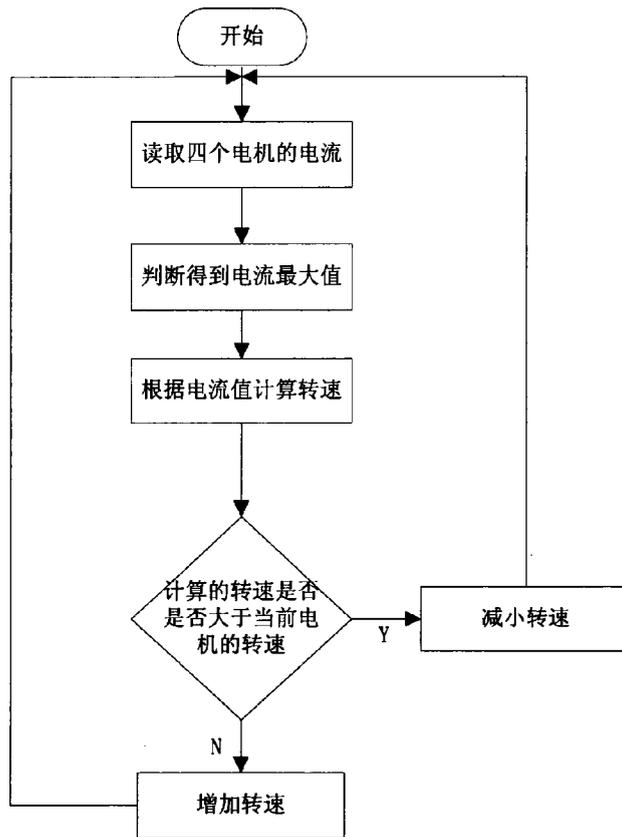


图 5-4 机器人运行算法流程图

5.5 本章小结

本章主要介绍了机器人控制系统的整个软件设计，其中包括嵌入式 Linux 系统交叉编译环境建立、控制器 Linux 系统 Flash 启动制作，伺服驱动器 ModbusRTU 协议，Linux 系统串口编程，最重要的是机器人运动控制程序设计。在机器人运动程序设计中，详述了机器人直行控制程序设计、转弯控制程序设计、故障检测程序设计及运行控制算法设计和程序实现。

第六章 结论与展望

6.1 结论

本系统以研华科技的RSB4210为主控制器，利用Linux串口通信技术，ModbusRTU协议，实现了沙漠信息感知机器人控制系统，完成机器人直行控制程序设计，机器人差速转弯控制程序设计，机器人故障检测程序设计，机器人运行控制程序的设计，并且利用matlab软件的simulink工具对机器人控制系统建模，对运行的控制算法进行仿真。在本次设计的过程中，从软件开发平台的搭建，到整个系统软件的编程实现是本人独立完成。在此过程中，也遇到很多的问题，通过自己的查资料全部解决了，自己也学到了很多知识。本次毕业设计主要完成了以下任务：

- (1)根据多轮独立驱动机器人的国内外发展现状，提出基于嵌入式Linux的机器人控制系统，利用RS485网络实现各个电机和主控制直接通信；
- (2)利用Linux串口编程技术，ModbusRTU协议，实现机器人运动控制系统，完成设计的动作；
- (3)对机器人运动控制系统的软件设计之后，为了验证系统运行控制算法，对机器人运动控制系统进行simulink仿真，并且分析仿真结果。

6.2 展望

由于时间的限制，本设计实现了机器人监控系统的基本控制，但是沙漠机器人在沙漠环境，收到各种因素的影响，整个系统没有用到传感器，对机器人的位置进行定位，导致机器人转弯时的角度不准确，因此系统需要传感器的数据反馈，使整个系统更精准。对在以后的学习中需要弥补以下几点不足：

- (1)系统中加入传感器，弥补系统的欠缺。系统六轴传感器对机器人的位置进行定位，加入超声传感器对机器人前面的障碍物进行检测；
- (2)为了对机器人周围的环境观察，为系统加入摄像头，方便对机器人的运程控制；
- (3)在直流伺服电机伺服系统的仿真中，转速环利用PI控制器，进行仿真，系统的响应不是很快，之后转速环更换其它算法，加快系统的响应速度。

参考文献

- [1] 朱磊磊. 轮式移动机器人研究综述[J]. 机床与液压. 2009, 39(8): 242-247.
- [2] 孙宏宇. 基于ARM的轮式机器人控制系统设计[D]. 大连: 大连理工大学, 2013.
- [3] 焦尚伟. 嵌入式Linux在轮式机器人控制中的应用[D]. 大连: 大连理工大学, 2005.
- [4] 高海波, 张鹏, 邓宗全. 新型八轮月球车悬架的研制[J]. 机械工程学报. 2008, 44(7): 85-92.
- [5] 张明路. 移动机器人的研究现状与趋势[J]. 河北工业大学学报. 2003, 33(2): 111-115.
- [6] 邓宗全, 高海波, 胡明. 行星越障轮式月球车的设计[J]. 哈尔滨工业大学学报. 2003, 35(2): 203-207.
- [7] 毕贞法, 邓宗全, 陶建国. 变驱动半径两轮并列式月球车的稳定性分析[J]. 上海交通大学学报. 2007, 41(7): 1204-1208.
- [8] <http://baike.so.com/doc/7333931.html>: 2013.12.
- [9] Rollins E, Luntz J, Foessel A. A Demonstration of the Transforming Chassis[Z]. 1998.
- [10] Wettergreen D, Thomas H. Initial results from vision-based control of Ames Lama[Z]. Grenoble, France: 1997.
- [11] T E, Y C, B M. An innovative Space Rover with Extended Climbing Abilities[Z]. 2000.
- [12] 赵云. 洛克希德_马丁MULE无人车[J]. 兵器知识. 2004(2).
- [13] <http://mars.jpl.nasa.gov/MPF/index1.html>: 2013.12
- [14] 刘兴杰, 江磊, 杨树岭. 星球车行走系统和它的研制者们-美国篇[J]. 机器人技术与应用. 2008(3): 14-16.
- [15] Zhiying W, Xilun D, Rovetta A. Structure Design and Locomotion Analysis of a Novel Robot for Lunar Exploration[J]. 12th IFToMM World Congress. 2007.
- [16] Franc, Michaud O, L'Etourneau D. Multi-Modal Locomotion Robotic Platform Using Leg-Track-Wheel Articulations[J]. Autonomous Robots. 2005(18).
- [17] Poulson E A. Design of A Rover with Six Intelligent Wheels[J]. M.S.Thesis in Mechanical Engineering. 2000.
- [18] 江磊, 姚其昌, 何亚丽. 星球车行走系统和它的研制者们-俄罗斯篇[J]. 机器人技术与应用. 2008(3): 17-19.
- [19] Rooney J, Hobbs J D. Towards kinematic classification schemes for planetary surface locomotion systems[J]. Mechatronics. 2003, 3(13): 153-174.
- [20] 费晓曦. 多轮独立驱动轮式移动机器人驱动控制研究[D]. 长沙市: 国防科学技术大学, 2011.
- [21] 翟丽, 董守全, 罗开宇. 四轮电机独立驱动车辆转向电子差速控制[J]. 北京理工大学学报. 2010, 30(8): 901-905.
- [22] 赵冬斌, 易建强. 全方位移动机器人结构和运动分析[J]. 机器人. 2003, 25(5).
- [23] 常勇, 马书根, 王洪光. 轮式移动机器人运动学建模方法[J]. 机械工程学报. 2003, 46(5): 30-36.
- [24] 李沁生, 于家凤. 基于Simulink的直流伺服电机PID控制仿真[J]. 船电技术. 2011, 31(3): 26-29.
- [25] 龙驹. 永磁无刷直流电动机调速控制系统的优化设计[M]. 成都: 西南交通大学出版社, 2006.
- [26] 王雷. 无刷直流电动机调速系统的研究[D]. 浙江: 浙江大学, 2008.
- [27] 贾东耀, 曾智刚. 基于模糊控制的直流电机调速系统MATLAB仿真[J]. 电机电器技术. 2002(5): 2-5.
- [28] 坤刘, 高少平, 张福军. 基于模糊PID的直流电机调速系统仿真研究[J]. 防爆电机. 2009, 44(2): 31-34.
- [29] 张莉, 李彦明, 马培芬. 基于模糊PID控制器的多电机同步控制装置的应用[J]. 工业仪表与自动化控制. 2003(4): 11-13.
- [30] 刘刚. 永磁无刷直流电机控制技术与应用[M]. 北京: 机械工业出版社, 2008.
- [31] 邱恒浪. 基于轮毂电机的四轮驱动差速转向控制系统的研究[D]. 重庆: 西南大学, 2012.
- [32] 张福炎. 嵌入式系统开发技术[M]. 北京: 高等教育出版社, 2012.
- [33] 任燕君. 电机与拖动[M]. 北京: 机械工业出版社, 2011.
- [34] 王勇. 步进电机与伺服电机的比较[J]. 西部煤化工, 2010, (3):47-49.

- [35] 邱恒浪. 基于轮毂电机的四轮驱动差速转向控制系统的研究[D]. 重庆: 西南大学, 2012.
- [36] 刘波. 基于CAN总线的侦察机器人控制系统设计[J]. 测控技术. 2013. 32(1): 65-68.
- [37] 潘传勇, 丁国臣, 陈世夏. 基于LM2596的不间断直流电源设计[J]. 现代电子技术. 2013, 36(17): 106-109.
- [38] 黄义文. Linux操作系统内核裁剪的分析[J]. 中国民航飞行学院学报. 2010, 21(3): 56-59.
- [39] 许青林, 解争龙. 基于ARM的Linux系统移植研究与实现[J]. 物联网技术. 2013(1): 37-42.
- [40] 王书根, 王振松, 刘晓云. Modbus协议的RS485总线通讯机的设计及应用[J]. 自动化仪表. 2011(5): 25-28.

致 谢

时光飞逝，两年的硕士研究生学习生活即将结束，在这即将毕业之际，真诚的向所有关心、帮助我的老师、同学以及朋友亲人表示最真挚的谢意。

在这里，首先特别感谢我的导师刘大铭教授，刘老师对我细心教导，循循善诱，教导我如何将理论用于实践，经常带我去实地做实验，增加我的工程经验。另外，从我的论文定题到现在，刘老师一直在认真指导，提出修改意见，在此，我真心的向刘老师说声谢谢。

感谢霍雯，赵九龙等舍友，在平时生活中给予我的帮助，在我失落沮丧时，给予我的勇气和信心，在我学习中遇到困难时，帮助我解决苦难，真诚的谢谢你们。

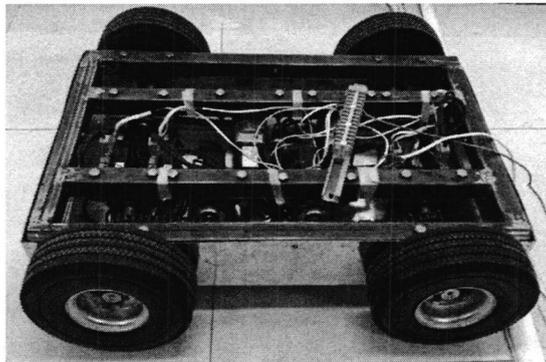
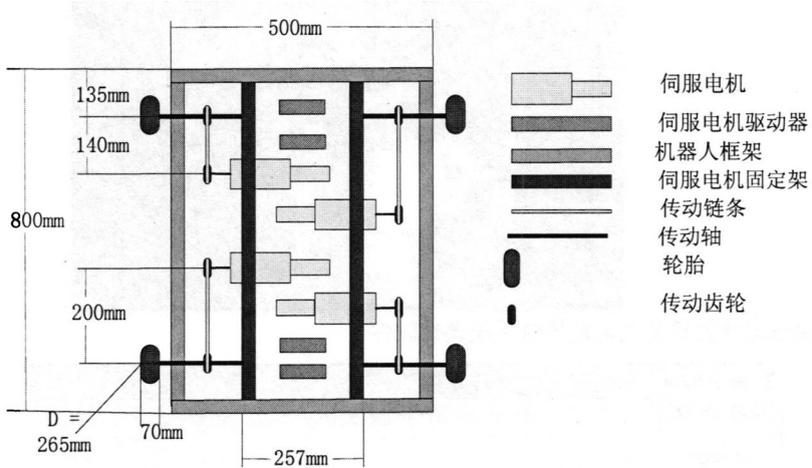
感谢我的父母，感谢他们二十四年对我的养育之恩，无私的关怀，教导我如何做人做事，你们的一言一行，牢记心中，你们对我的恩情无以回报，你们身体健康是我最大的心愿。

感谢我学习生活六年的母校——宁夏大学，母校给了我一个宽阔的学习平台，让我不断学习、吸取知识，充实我的头脑，不断提高我的科研能力。

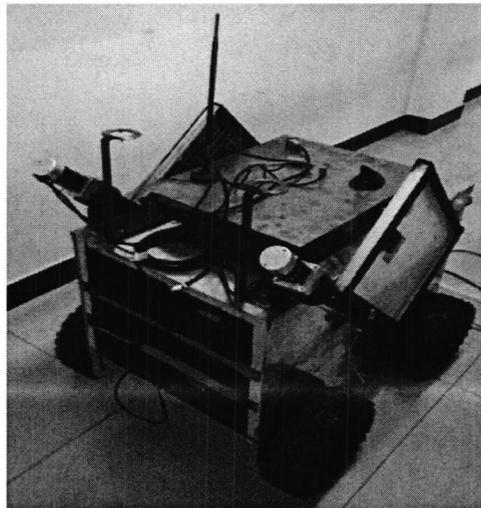
最后，祝愿学院所有老师、同学、朋友和家人平安健康，万事顺心！

附录

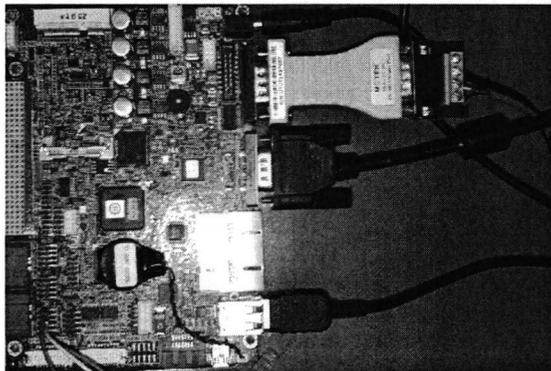
(1)沙漠机器人底盘结构及底盘实物图



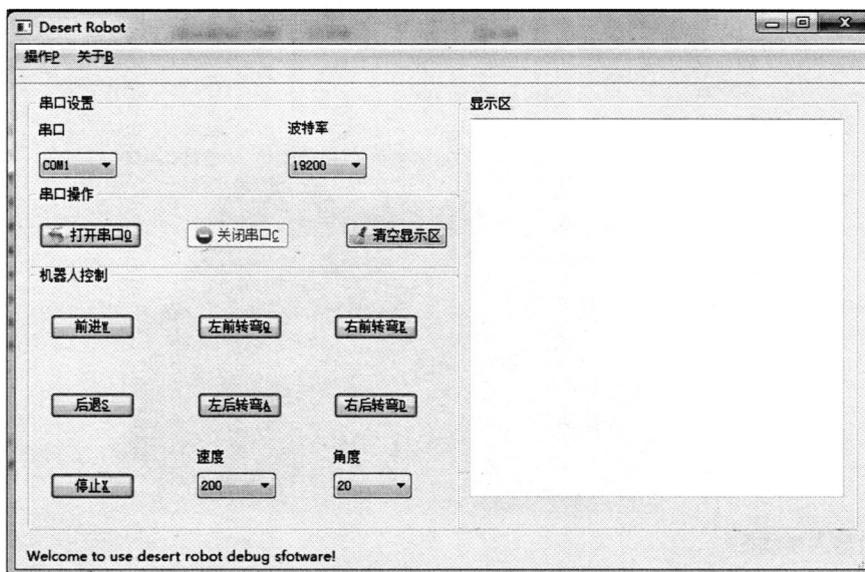
(2)机器人实物图



(3)沙漠机器人主控制器实物图



(4)沙漠信息感知机器人运动控制系统调试软件



(5)部分主要函数代码

```

void Receive_Dispose()
{
    if(Uart0_Receive_Ready_Flag)
    {
        Uart0_Receive_Ready_Flag = 0;
        Uart0_Receive();
        Uart0_Receive_Flag = 1;
    }
    if(Uart0_Receive_Flag)
    {
        Uart0_Receive_Flag = 0;
    }
}

```

```
        Contral_Car(Uart0_Receive_Buff);
        Rs485_Send_Flag = 1;

    }
    if(Rs485_Send_Flag )
    {
        Rs485_Send_Flag = 0;
        Rs485_Receive();
        Rs485_Receive_Flag = 1;
    }
    if(Rs485_Receive_Flag )
    {
        Rs485_Receive_Flag = 0;
        Check_Modbus(Rs485_Receive_Buff);
        Uart0_Receive_Ready_Flag = 1;
    }
}

Void Write_Register(char addr, char Register_Addr, int data)
{
    unsigned int crcData;

    sendbuf[0] = addr; //slaver address
    sendbuf[1] = 0x06; //funtion read
    sendbuf[2] = 0x00; //funtion read
    sendbuf[3] = Register_Addr; //register address
    sendbuf[4] = data >> 8;
    sendbuf[5] = data & 0xff; //data
    crcData = crc16(sendbuf, 6);
    sendbuf[6] =crcData >> 8;
    sendbuf[7] =crcData & 0xff;
    Rs485_Send(sendbuf, 8);
    usleep(10000);
}

void Read_Register(char addr, char Register_Addr)
{
    unsigned int crcData;

    sendbuf[0] = addr; //slaver address
```

```
    sendbuf[1] = 0x03; //function read
    sendbuf[2] = 0x00; //function read
    sendbuf[3] = Register_Addr; //register address
    sendbuf[4] = 0x00;
    sendbuf[5] = 0x01; //register number is 1
    crcData = crc16(sendbuf, 6);
    sendbuf[6] = crcData >> 8;
    sendbuf[7] = crcData & 0xff;
    Rs485_Send(sendbuf, 8);
    usleep(10000);
}
void Set_On_Timer()
{
    if(signal(SIGALRM, sig_handler) == SIG_ERR)
    {
        printf("Unable to create handker for SIGALRM!\n");

        return;
    }
    v1.it_interval.tv_sec = s;
    v1.it_interval.tv_usec = us;
    v1.it_value.tv_sec = s;
    v1.it_value.tv_usec = us;
    setitimer(ITIMER_REAL, &v1, NULL);

    return;
}

//Timer interrupt function

static void sig_handler(int signo)
{
    if(signo == SIGALRM)
    {
        Check_Driver();
        Check_Current()
    }
}
```

```
v1.it_interval.tv_sec = s;
v1.it_interval.tv_usec = us ;
v1.it_value.tv_sec = s;
v1.it_value.tv_usec = us;

setitimer(ITIMER_REAL, &v1, NULL);

return;
}
```

个人简介及攻读硕士学位期间的研究成果

基本情况

姓名：周栋 性别：男 民族：回族 出生日期：1989年7月 籍贯：宁夏

教育经历：

2008年9月—2012年6月	宁夏大学	通信工程	本科
2012年9月—2014年6月	宁夏大学	电子与通信工程	硕士

项目经验：

2012年12月—2014年5月参与《沙漠腹地沙尘检测与信息融合技术合作研究》科技部国际合作项目（No.2011DFA11780）

发表论文：

周栋、刘大铭、焦勤皓. 沙漠信息感知机器人运动控制系统设计. 宁夏大学学报. 2014.年5月(已录用)