

摘 要

移动机器人是现代科学技术的热点研究领域,包含了多个研究分支。其中,移动机器人控制技术致力于构造机器人的“控制中枢”,具有重要的研究意义和应用价值。本文主要涉及基于数字信号处理器(Digital Signal Processor, DSP)和嵌入式实时操作系统(Real Time Operating System, RTOS)的机器人控制器的构建,以及基于该控制器的机器人运动平衡、负趋光和避障控制的实现,取得了以下主要的研究结果:

第一,设计了一种基于数字信号处理器和嵌入式实时操作系统的移动机器人控制器。论文分析了 TMS320F2812 型数字信号处理器和嵌入式实时操作系统 $\mu C/OS-II$ 的体系结构。在此基础上,构造了一种嵌入式移动机器人控制决策系统。该系统既可独立执行具有一定复杂度的控制任务,又适用于主从式或者分布式控制结构,具有一定的开放性。

第二,基于本文所设计的控制系统实现了倒立摆的运动平衡控制。论文分析了单级直线倒立摆的机电结构及其不同运动阶段的运动特性,然后,利用所设计的控制器应用经典控制理论和现代控制理论实现了倒立摆起摆和稳摆控制。

第三,基于本文所设计的控制器实现了移动机器人避障与负趋光控制。论文分析了《控制论》中关于维纳虫的思想实验。然后,设计了环境信息采集电路,以一个三轮小车为载体实现了移动机器人的负趋光和避障控制。

本课题研究得到了国家自然科学基金(60375017)、北京市人才强教计划的支持。基于本课题的研究工作所撰写的两篇学术论文《嵌入式移动机器人控制器设计》、《基于DSP的倒立摆控制系统设计》已分别被中文核心期刊《计算机工程与应用》和《控制工程》录用,并取得了软件著作权一项,所申请的一项实用新型专利已通过初审。

关键词 移动机器人; 控制器; 数字信号处理器; 运动平衡; 负趋光

ABSTRACT

Research on mobile robot is a hotspot of the modern science and technology, comprising of several branches. One of these subfields is Robot control technology, which is dedicated to construct robots' central control hinge and of great value of both research and application. The paper mainly involves the construction of DSP&RTOS-based robot controller, robot balance control, light avoidance and obstacle avoidance on the basis of the above controller. The achievements in this thesis can be summarized as follows:

First, design and realization of mobile robot controller using DSP and RTOS. The paper analyzes the architecture of TMS320F2812 DSP and embedded real time operating system $\mu\text{C}/\text{OS-II}$, Using them constructs an embedded mobile robot control system. The system is not only capable of carrying out some tasks of certain complexity by itself, but also applicable to take a part in a master-slave or the distributed control system as a subsystem, so it's open in some degrees.

Second, realization of inverted pendulum control using the above mentioned controller. The paper analyzes the single pendulum's electromechanical structure and motion specialities separately related to its both stages of motion, Applies classical and modern cybernation theories to make a single pendulum to swing up and hold steady, using he above mentioned controller.

Third, realization of mobile robot obstacle avoidance and light avoidance. The paper analyzes the virtual experiment about electromechanical reflex arc described in *Cybernetics*. Designs circumstance detecting circuit and realizes a three-wheeled cart obstacle avoidance and light avoidance, using the above mentioned controller.

The research described in the paper was supported by National Nature Science Foundation of China(60375017)、Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (PHR(IHLB)).

Key words mobile robot; controller; DSP; motion balance; light avoidance

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，也不包含为获得北京工业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。

签名： 常青 日期： 2007.6.1

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签名： 常青 导师签名： 陈维新 日期： 2007.6.1

第1章 绪 论

1.1 移动机器人学及其研究内容

移动机器人学是机器人学的一个重要分支,主要研究在复杂环境下机器人系统的实时控制问题。对移动机器人技术的研究始于20世纪六十年代,以斯坦福研究所的自主式移动机器人 SHAKEY 为标志^{[1][2]}。目前,移动机器人技术已经成功应用于城市作战、排爆、反恐、消防、空间探测等领域。

一般地说,一个移动机器人通常由感知系统、控制决策系统、执行系统三大部分组成。其中感知系统获取并向控制决策系统提供有效的环境信息,控制决策系统则对这些信息进行运算,最后做出决策并向执行系统发出控制指令,执行系统根据指令执行诸如行走、转向、抓取等动作,最终达成目标。

按照移动方式的不同,可将移动机器人分为足式移动机器人和轮式移动机器人。足式移动机器人主要是模仿动物以及人类的行走方式,主要研究内容是机器人本体控制(运动平衡)问题,即机器人运动过程的平稳性。轮式移动机器人按其结构又可分为单轮、双轮、三轮、四轮及履带式。轮式移动机器人的研究热点是导航、路径规划以及单轮和双轮机器人的运动平衡等课题。

除按照移动方式分类外,移动机器人的分类方法还有很多,比如按移动机器人控制器的体系结构来分、按功能和用途来分、按作业空间来分、按“智能”程度来分。

不同种类的移动机器人由于其工作环境、任务目标等的不同会对机器人的各个组成部分产生不同的要求。譬如足式机器人的控制系统主要关注机器人的运动平衡控制。而空间探测机器人由于要携带大量的勘探设备,所以执行结构比较繁杂。而某些特种机器人要求系统具有视觉功能等。

1.2 移动机器人控制技术的现状与发展

移动机器人控制技术主要涉及控制器的体系结构、导航以及足式或轮式机器

人的运动平衡控制等课题。

1.2.1 机器人控制器的体系结构

控制器的体系结构是一个工程背景很强、涉及领域较多的综合性课题。包括硬件和软件两个方面。

从硬件的角度来看，移动机器人控制系统有如下三种结构方式：

- (1) 集中式控制结构，用一台功能比较强大的计算机实现全部的控制功能。
- (2) 主从式控制结构，有上下两级计算机，其中上位机利用其强大的运算能力和庞大的系统资源完成坐标变换、轨迹插补、图形仿真等功能，并把数据传送给下位机，下位机完成一些相对简单的具体控制，并把相关数据上传给上位机。
- (3) 分布式控制结构，主要为工业机器人、多机器人系统所采用。上一级计算机完成整个系统的管理和坐标变换以及轨迹插补运算等。下位机由多个微控制器组成，他们并行工作。上下位机之间以及下位机之间通过总线交换信息。这是一种开放型的控制系统^[3]。

从软件的角度来看，移动机器人控制体系主要有三种结构，分别是基于知识的功能式结构(又称为水平分解型)、基于行为的行为式结构(又称为垂直分解型)和混合式结构。

基于知识的体系结构是一种按信息流向将功能模块依序分解排列的结构，在这种系统中，感知、建模、规划、任务执行和驱动控制等模块次序分明，前者的输出结果即为后者的输入，所以又称为 SMPA (Sense-Model-Plan-Act) 结构。这种结构有利于研究人员按照模块功能与自己的专长划分设计任务，同时比较容易实现高层次的智能。

基于知识的体系结构遵循的是从感知到动作的串行功能分解控制路线，如图 1-1 所示。

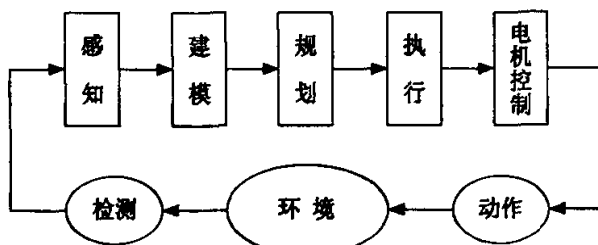


图 1-1 基于知识的体系结构
fig 1-1 Knowledge-based architecture

卡内基梅隆大学 (Carnegie-Mellon University, CMU) 研制的 Navlab 系统就是一种基于功能分解型结构的系统。这种结构存在一些严重问题,其中之一是可靠性低,一旦这种串行结构中的某个模块出现故障,就可能导致整个系统的瘫痪。另外,由于信息串行传递处理所导致的一个致命缺陷是实时性较差,无法实现特殊的快速反射功能。

基于行为的体系结构将控制器依据行为能力划分成在功能上逐层迭加的层次结构,每个层次都根据自身的目标处理相应的信息,并给出相应的控制命令。尽管高层次会对低层次施加影响,但低层次本身具有独立控制机器人运动的功能,而不必等待高层次完成处理。

基于行为的体系结构以移动机器人的被动式行为控制为算法框架,重点突出“感知-动作”行为控制(见图 1-2)。从物理结构上来说,系统中存在着多个并行控制回路,构成各种基本行为,传感数据根据需求以一种并行方式给出,各种行为通过协调配合后作用于驱动装置,产生有目的的动作。

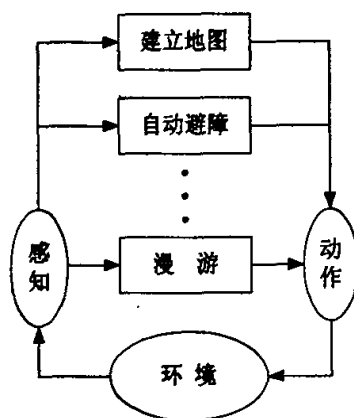


图 1-2 基于行为的体系结构
fig 1-2 Action-based architecture

基于行为的体系结构具有响应快、鲁棒性强与灵活性高等特点。但是这种结构需要设计一个协调机制来解决各个控制回路对同一执行装置控制权的竞争,工程难度较大。

近几年已经陆续出现了一些基于行为的体系结构的改进方法和变异结构。Pattie Maes 已在研制开发一种分节式的学习算法,试图寻求一条通过经验学习行为的控制与协调的途径。Steels 提出了一种突发框架识别的思想并据此建立了一个框架识别系统,在此基础上研制了具有避障、沿墙行走、目标追踪等感知

行为的机器人^[4]。

混合式体系结构的目标就是把基于知识的体系结构和基于行为的体系结构的优点有效地结合在一起。

1.2.2 移动机器人导航技术

导航是移动机器人研究的一个重要课题。移动机器人有多种导航方式,根据环境信息的完整程度、导航指示信号类型、导航地域等因素的不同,可以分为基于地图导航、基于路标导航、基于视觉导航、基于感知器导航^{[5][6]}等。

基于地图的导航是指机器人掌握完整的环境信息,在全局路径规划的基础上,采用路径跟踪和避障技术,实现机器人导航,涉及环境地图模型建造和模型匹配两大问题^{[7][8]}。当所掌握的环境信息并不完全时,则可采用基于路标的导航策略,将全局路线分解成陆标与陆标之间的片段,机器人通过对陆标的探测来定位^[9]。在环境信息完全未知的情况下,可通过摄像机或传感器探测环境信息以实现机器人导航。在相对规整的环境中,可以采用寻迹导航的方式,机器人在行走的过程中对标志线进行探测,并据此调整行进路线,最终在标志线的指引下到达指定目的地。

无论采用何种方式,移动机器人的导航主要涉及定位、路径规划和避障等技术^[10-13]。

1.2.3 移动机器人运动平衡控制与倒立摆

移动机器人运动平衡控制属于移动机器人本体控制范畴,是足式机器人和单轮或双轮机器人的研究热点,目的是保持机器人在行走过程中的整体平衡性。

足式机器人研究的难点是双足式机器人的运动平衡控制,目前已取得突破性进展,达到了实用水平。比如日本本田公司研制的服务机器人ASIMO和机器人足球比赛中使用的人形球员等。

轮式机器人的运动控制主要以双轮式机器人为研究对象。2002年,瑞士联邦工业大学的Felix Grasser等研制出可遥控的两轮自平衡机器人Joe(见图1-3),其最大运动速度可以达到1.5 m/s,超过了人的行走速度。2004年,中国科学技术大学成功研制了两轮自平衡电动代步车Free Mover(见图1-4)。它可以实现零半径回转,操作者经10分钟的训练后就可以驾驶其行走。身体前倾时车子自动加速,后倾时自动减速,时速可达10公里^[14]。双轮式移动机器人的运动平衡控制过

程与直线型倒立摆的控制过程类似,故又称倒立摆机器人或者轮式倒立摆^{[15][16]}。

倒立摆是20世纪50年代由麻省理工学院(MIT)的控制论专家根据火箭发射助推器原理设计的。这是一种典型的快速、多变量、非线性、自然不稳定的高阶系统,按运动方式可以分为直线型、平面型和环形倒立摆,按照摆杆级联的数量可以分为单级、二级和多级倒立摆。直线型倒立摆在结构和运动特性上与足式机器人、双轮机器人的相似性,使其能够比较真实地模拟机器人的运动平衡控制。因此,倒立摆系统的运动控制对于移动机器人控制技术的研究具有重要的参考价值。

我国对倒立摆系统的研究始于20世纪80年代,经过20余年的探索,已经走在了世界的前列。北京师范大学李洪兴教授分别于2001年6月和2002年8月分别完成了四级倒立摆系统的仿真和实物实验^{[17][18]},是目前世界上控制成功的多级倒立摆系统中级数最多的。



图 1-3 自平衡机器人Joe^[19]
Fig 1-3 Balancing robot: Joe^[19]



图 1-4 自平衡两轮代步电动车Free Mover^[20]
Fig 1-4 Self-balancing vehicle: Free Mover^[20]

1.2.4 机器人控制器研究的发展方向

目前市场上的机器人控制器主要存在以下4个方面的缺陷。首先是开放性

差,“专用计算机、专用机器人语言、专用微处理器”的封闭式结构使其具有特定的功能、适应于特定的环境,但是不便于对系统进行扩展和改进。第二是软件独立性差,软件功能及其逻辑结构依赖于处理器硬件,难以在不同的系统间移植。第三,容错性差,多处理器协同工作时,其中一个处理器出故障可能导致整个系统的瘫痪。第四,扩展性差,由于结构的封闭性,难以根据需要对系统进行扩展,如增加传感器等功能模块^[21]。

归根结底,这四种缺陷都是由于控制器体系结构的开放性差造成的,由此决定了今后机器人控制器的主要研究方向是开发“具有开放式结构的模块化、标准化机器人控制器”。目前,日本、美国和欧洲一些国家都已着手开发具有开放式结构的机器人控制器。

开放式结构机器人控制器是指控制器设计的各个层次对用户开放,用户可以方便的扩展和改进其性能,其主要思想是^[22]:

(1)采用非封闭式计算机平台。有效利用标准计算机平台的软、硬件资源为控制器扩展创造条件。

(2)采用标准的操作系统和标准的控制语言,从而改变各种专用机器人语言并存且互不兼容的局面。

(3)采用标准总线结构,简化为扩展控制器性能所需的集成工作。

(4)利用网络通讯,实现资源共享或远程通讯,提高系统变化的柔性。

因此,新型的机器人控制器应具有以下特点:

(1)开放式系统结构。采用开放式软件、硬件结构,可以方便地剪裁系统功能,使其适用于不同类型机器人^{[23][24]}。

(2)合理的模块化设计^[25]。对硬件来说,根据系统要求和电气特性,按模块化设计,这不仅方便安装和维护,而且提高了系统的可靠性,系统结构也更为紧凑。

(3)有效的任务划分。不同的子任务由不同的功能模块实现,以利于剪裁系统功能。

(4)实时性、多任务能力。机器人控制器必须能在确定的时间内完成对外部中断的处理,并且可以使多个任务同时进行。

(5)网络通讯功能。利用网络通讯功能,以便于实现资源共享或多机器人协

同工作。

(6) 形象直观的人机接口。

由于传感器和处理器等产品的新种类层出不穷,开发一个具有绝对开放式结构的标准化机器人控制器存在一定困难^{[26] [27]},因此,所谓开放式体系结构必然是一个历史的、相对的概念。

1.3 课题来源及研究内容

20 世纪中叶以来,计算机科学、控制理论、传感器技术、电子技术等的飞速发展极大地推动了机器人学的发展。时至今日,移动机器人的应用已经深入到国计民生的各个领域。同时,移动机器人控制技术对于机器人系统的整体性能有着决定性的作用,因此,对移动机器人控制系统进行研究具有重要的理论与实践意义。

本课题来源于国家自然科学基金项目《基于达尔文自动机和运动神经内膜理论的运动控制技能进化认知模型与自治机器人 Agent 的研究 (Cognitive Models Based On Darwin Automation For Motor Skill and Autonomous Robot Agents)》(60375017),并得到了北京市人才强教计划的支持。

本课题主要研究内容包括:

- (1) 基于 DSP 和 RTOS 构建具有一定开放性的移动机器人控制系统。
- (2) 单级直线倒立摆运动控制。
- (3) 移动机器人负趋光与避障控制。

第 2 章 控制器系统硬件设计

本章针对机器人控制中的常用器件和技术设计了移动机器人控制器的硬件系统。首先，确定了系统设计方案。然后，分析了 TMS320F2812 型数字信号处理器的体系结构并设计了 DSP 最小系统。最后，分析了与机器人控制相关的接口技术并设计了接口电路。

2.1 系统硬件设计方案

控制系统硬件设计遵循机器人控制器的开放性原则，在保证目标系统能够独立完成一定控制功能的基础上，兼顾系统硬件的可扩展性。在控制系统设计过程中，所谓目标系统的可扩展性包括向目标系统中添加传感器等资源 and 以目标系统为模块组成更高级系统的能力。

考虑到系统造价，本文确定目标系统的功能如下：第一，可实现两台直流电机的闭环调速，包括编码器信号采集和 PWM 脉冲信号的输出；第二，可采集 0~5V 的模拟信号、开关量信号；第三，可实现系统状态的光电显示和继电器控制；第四，可以适用于集中式（单板方式）、主从式和分布式控制结构。因此，采用如下设计方案。

首先，采用 TMS320F2812DSP 为核心设计最小系统。F2812 是美国德州仪器（Texas Instrument, TI）公司出产的一款面向控制的新型高性能数字信号处理器，由于 TI 的 DSP 产品在国际市场上所占份额较大，所以采用该公司产品作为控制核心可以增强系统软件的可移植性。

第二，充分利用 DSP 的片内资源，设计接口电路完成模拟信号采集、开关量信号采集、编码器脉冲信号采集、直流电机脉冲控制、状态显示、继电器控制。

第三，利用 DSP 的外部存储器扩展接口扩展一片 DA 转换器 DAC7724，作为控制电机驱动器的信号接口。

第四，利用 DSP 的串行通信接口（SCI），采用一片 MAX3232 设计 RS-232-C 串行通信接口，以实现目标系统的数据通信能力。

第五, 采用光电隔离方式确保系统安全。

第六, 硬件设计采用模块化方法。首先将系统划分为 DSP 最小系统和接口部分, 然后根据系统硬件资源(见表 2-1)将各部分细化为若干个子模块。

表 2-1 控制器系统硬件资源

Table 2-1 Hardware resources of controller

序号	资源	数量	功能
1	静态 RAM	256K×16 位	数据存储
2	模拟量输入通道	8 路	模拟信号采集
3	数字量/中断输入通道	3 路	开关量信号采集
4	QEP 输入通道	2 路	编码器信号采集
5	PWM 信号输出通道	4 路	电机控制
6	数字量输出通道	6 路	继电器控制
7	LED 阵列	4 只	光电显示
8	RS-232C 接口	2 路	数据通信
9	D/A 转换接口	4 路	电机控制

2.2 DSP 最小系统设计

DSP 的最小系统包括 DSP 及其时钟、JTAG 仿真接口、电源、存储器扩展和系统配置电路, 是整个系统硬件的核心部分。

2.2.1 TMS320F2812 DSP 的体系结构

DSP 是对信号和图像实现实时处理的一类高性能的处理器。所谓“实时(Real-time)处理”, 是指一个实际系统能在人们视觉、听觉或任务要求所允许的时间范围内实现对输入信号的处理并将运算结果输出。目前, DSP 已经广泛应用于通信、家电、航空航天、工业测量与控制、生物医学工程及军事等许多对实时性有较高要求的领域。

TI 自 1982 以来, 已先后推出了 C10、C20、C30、C40、C50、C80 等 6 代 TMS320 系列 DSP 产品和 C2000 系列、C5000 系列、C6000 系列三大主流产品。

TMS320F2812 是 TI 最新推出的 DSP 芯片, 是目前国际市场上最先进、功能最强大的 32 位定点型 DSP 芯片。它既有数字信号处理能力, 又具有强大的事件管

理能力和嵌入式控制功能, 特别适合像移动机器人控制、工业自动化控制等有大批量数据需要处理的测控场合。

2.2.1.1 DSP 的 CPU TMS320F2812 的 CPU 是一种低功耗的 32 位定点数字信号处理器。它集中了数字信号处理的诸多优秀特性, 具有精简指令集 (RISC) 功能、微型控制器结构、固件 (firmware) 及工具集等。

该 CPU 采用改进型的哈佛结构, 通过 6 条独立的数据和地址总线实现流水线操作。这 6 条总线分别是程序地址总线 (Program Address Bus, PAB)、数据读地址总线 (Data-Read Address Bus, DRAB)、数据写地址总线 (Data-Write Address Buss, DWAB)、程序读数据总线 (Program-Read Data Bus, PRDB)、数据读数据总线 (Data-Read Data Bus, DRDB) 和数据/程序写数据总线 (Data/program-write Data Bus, DWDB)。

CPU 达到最高工作频率 (150MHz) 时的核心电压仅需 1.9V, I/O 口电压为 3.3V。

F2812 片内有 3 个 32 位的 CPU 定时器 (TIMER0/1/2)。每个定时器可以产生一个溢出中断。

2.2.1.2 DSP 的中断系统和复位 F2812 的中断可以归结为两类:

(1) 可屏蔽中断 这些中断可以用软件加以屏蔽或解除屏蔽。

(2) 非屏蔽中断 这些中断不能被屏蔽, CPU 将立即响应该中断并转入相应的服务子程序 (Interrupt Service Routine, ISR)。

中断通常由硬件触发, 如果有多个硬件中断同时被触发, CPU 则按照它们的优先级来提供服务。此外, 使用 INTR、TRAP、OR IFR 指令可以由软件触发中断。

F2812 按如下的 4 个步骤处理中断。

(1) 接受中断请求。

(2) 测试当前中断源是否被允许, 若是, 转至第 3 步, 否则忽略该请求。

(3) 准备执行中断服务程序并保存寄存器值。

(a) 完整地执行完当前指令, 清除流水线中未到达第二阶段的所有指令。

(b) 将寄存器 ST0、T、AH、AL、PH、PL、AR0、AR1、DP、ST1、DBGSTAT、PC、IER 的内容保存到堆栈中, 以便自动保存主程序的大部分内容。

(c) 取回中断向量并将其放入到程序寄存器 PC 中。

(4) 执行中断服务子程序。

F2812 的中断采用 3 级控制, 分别是 CPU 级、PIE 级和设备级。可以通过程序在任一级别上使能或屏蔽相应的中断。

2.2.1.3 时钟和系统控制 F2812 内建了一个基于锁相环 (Phase-Locked Loop, PLL) 的时钟模块。该模块为芯片提供了所有必要的时钟信号, 还提供了低功耗方式的控制入口。PLL 具有 4 位比例控制, 用来选择不同的 CPU 时钟频率。

F2812 器件有 3 种低功耗工作方式, 分别是空闲 (Idle) 方式、挂起 (Standby) 方式、暂停 (Halt) 方式。

片内外设的时钟可以通过软件使能或关断。并可分别对高速外设和低速外设的时钟频率进行预定标 (Pre-Scale) 操作。

F2812 内置了一个 8 位的看门狗定时器, 该定时器可以通过软件使能或关断。当被使能时, 看门狗定时器将振荡器时钟 512 分频后计数, 当 8 位计数器溢出时则数出一个脉冲, 脉宽为 512 个振荡周期。

2.2.1.4 片内外设的中断扩展 F2812 芯片内部的外设资源非常丰富, 每种外设出于实时处理数据的需要往往占用一定数量的中断资源。因此, 设计了 3 级中断管理机制提供方便灵活的中断管理, 可以通过软件分别在 CPU 级、PIE 级和设备级对相应的中断源进行使能和关断控制。

外设中断扩展 (Peripheral Interrupt Expansion, PIE) 模块将许多中断源多路复用成一个较小的中断输入集。PIE 模块支持 96 个不同的中断, 这些中断被分成 12 个组, 每组 8 个中断, 每个组都被反馈到 CPU 的 12 条中断线的一条上。当前 F2812 的外设共使用了 45 个中断。

每个中断都得到了各自的向量的支持, 这些向量被保存在 RAM 中的 PIE 向量表中。PIE 向量表自首地址 0x00000D00 起连续存放在外设帧 0 中, 每个 32 位中断向量占用两个特定存储单元。当 CPU 响应中断时, 用 PIE 向量表中特定偏移量的地址中的内容覆盖程序计数器 PC 的数据。

2.2.1.5 事件管理器 事件管理器模块提供了众多面向控制的功能和特性, 他们在机器人运动控制和马达控制中是十分有用的。事件管理器模块包括 4 个 16 位通用定时器, 6 个全比较/PWM 单元、6 个捕捉单元和 2 路正交编码脉冲 (Quadrature-Encoder Pulse, QEP) 电路等。

6 个全比较/PWM 单元共可产生 6 对互补的 PWM 信号, 可控制 6 个 H 桥电路,

4 个通用定时器共可产生 4 路独立的 PWM 信号, 因此, F2812 可同时输出 16 路 PWM 信号, 并可编程控制互补 PWM 对的死区时间。事件管理模块的通用定时器共有 4 种工作模式, 分别是停止/保持模式、连续增计数模式、定向增/减计数模式和连续增减计数模式。通用定时器除完成自身的功能外, 还为事件管理模块中的其他设备提供时基。捕捉单元可以捕捉引脚上的电平变化并记录两次变化之间的时间延迟。正交编码电路由通用定时器 2/4 提供时基, 可以计数来自光电编码器的脉冲信号, 计数器的值是输入的脉冲信号的 4 倍。

2.2.1.6 模数转换模块 F2812 的 ADC 是一个 12 位分辨率的、具有流水线结构的模数转换模块。ADC 模块共有 16 路通道, 可配置为 2 个 8 通道的独立模块, 或级联成一个 16 通道的模块。

2.2.1.7 串行通信接口 串行通信接口 SCI 是一个双线的异步串口, 一般看作是 UART。SCI 模块支持 CPU 与采用 NRZ 标准格式的异步外围设备之间的全双工数据通信。

SCI 的接收和发送器各有一个 16 级深度的 FIFO, 发送 FIFO 宽度是 8 位而接收 FIFO 的宽度是 12 位。发送 FIFO 中各个单元的数据依次经发送移位寄存器逐位发送到数据线上, 而接收移位寄存器则逐位接收数据线上的数据位, 并将数据帧存储到接收 FIFO 中。

SCI 有两个多处理器通信协议, 分别是空闲线 (Idle Line) 多处理器模式和地址位 (Address Bit) 多处理器模式^[28-30]。

2.2.2 最小系统电路设计

如前所述, DSP 的最小系统包括 DSP 及其时钟、JTAG 仿真接口、电源、存储器扩展和系统配置电路。

系统电源采用二级稳压结构 (见图 2-1), 外电源经一片 7805 输出稳定的 +5V 电压, 然后由一片 TPS7333 和一片 TPS7301 分别提供 3.3V 的 I/O 口电压和 1.9V 的内核电压。

系统时钟及 JTAG 仿真接口电路如图 2-2 所示。目标层次的 TI 调试标准使用 5 个标准的 IEEE1149.1 (JTAG) 信号 (TRST、TCK、TMS、TDI、TDO) 和两个 TI 扩展口 (EMU0、EMU1)。

由于 F2812 内建 128K*16 位的 FLASH 存储器, 因此仅扩展了 256K 数据存储

器（见图 2-3）。IS61LV25616 是一款高速静态 RAM，读写周期是 10 微秒。数据存储扩展使用 DSP 存储器区域 2，以 XZCS2 为片选信号，基地址是 0x080000。

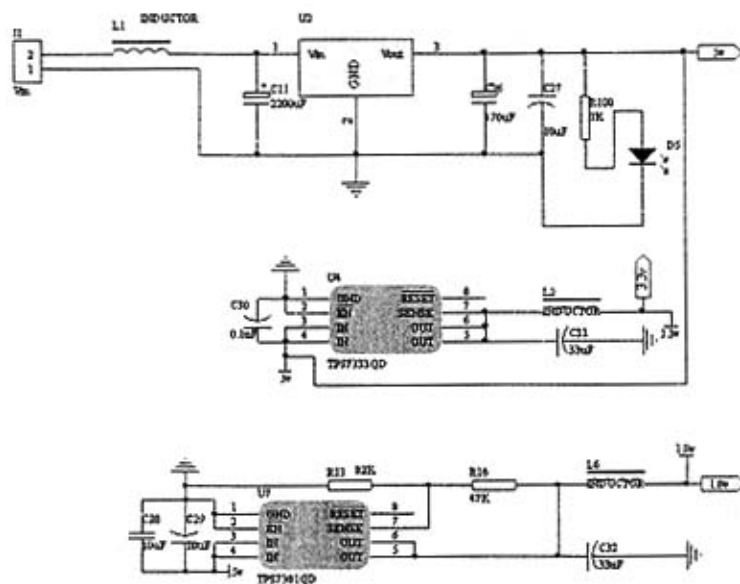


图 2-1 DSP 电源电路

fig 2-1 Schematic diagram of power supply

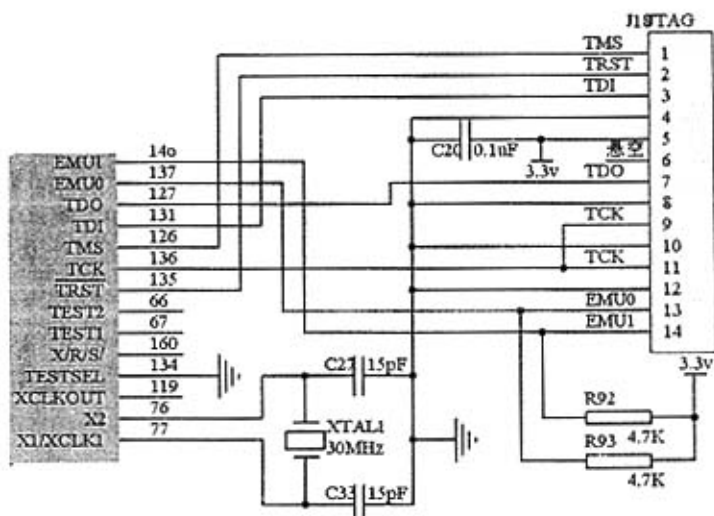


图 2-2 JTAG 仿真接口电路

fig 2-2 Schematic diagram of JTAG



fig 2-3 Schematic diagram of RAM extension

2.3.1 电机控制及脉宽调制

2.3.1 电机控制及脉宽调制

BG1、BG4 和 BG2、BG3 这两组晶体管分别由互补的 PWM 信号 P1 和 P2 控制。当 P1 的占空比大于 P2 的占空比时, 由于电枢电流 $I_a > 0$, 所以 BG2、BG3 不会导通, 电机正转。当 P1 和 P2 的占空比相同时, I_a 的平均值为 0, 电机停转。当 P1 的占空比小于 P2 的占空比时, $I_a < 0$, BG1、BG4 截止, 电机反转。

- 14 -

管理模块提供了 6 对带可编程死区的互补 PWM 输出电路。

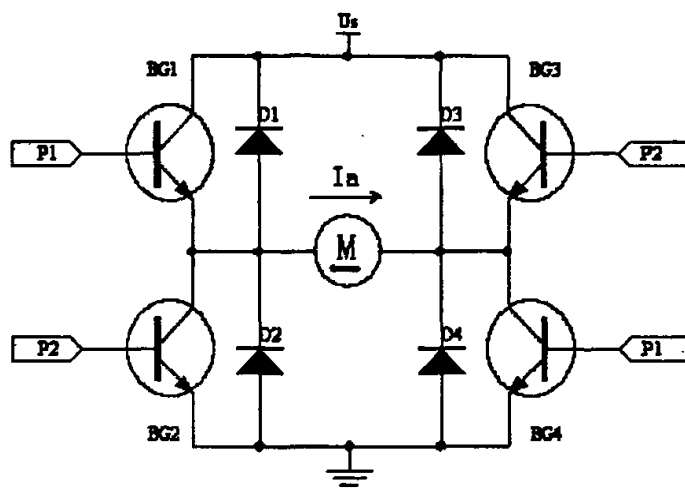


图 2-4 H 桥 PWM 驱动电路

fig 2-4 Schematic diagram of PWM-based H-structure

2.3.2 EIA-RS232-C 通信标准

串行通信不同于并行通信，其数据是一位接一位地传输的。串行通信的速率比并行通信低，但是结构简单，传输距离长。

EIA-RS232-C 通信标准是美国电子工业联合会 (Electronics Industry Association, EIA) 与 BELL 等公司一起开发并于 1969 年公布的通信协议。它适合于数据传输率在 0~20 000bps 范围内的通信。EIA-RS232-C 标准最初是为远程通信连接数据终端设备 (Data Terminal Equipment, DTE) 与数据通信设备 (Data Communication Equipment, DCE) 而制定的。该标准中所提到的“接收”和“发送”都是站在 DTE 的立场上的。

EIA-RS232-C 标准中，用数据收发线上的 -3~-15V 表示逻辑 1，而用 +3~+15V 表示逻辑 0。EIA-RS232-C 通信通常使用 DB9 连接器，DTE 端引脚的信号定义如表 2-2 所示。对于一般的双向连接仅需 3 根信号线，即数据接收 RXD，数据发送 TXD 和信号地 GND。

EIA-RS232-C 通信中，一帧数据最多可达 12 位，由以下几个部分组成：数据位，最多为 8 位；起始位，1 位；奇偶控制位，1 位；结束位，最多 2 位^[34-36]。

表 2-2 DTE 端引脚信号定义

Table 2-2 Definition of DTE pins

针号	功能说明	缩写
1	数据载波检测	DCD
2	接收数据	RxD
3	发送数据	TxD
4	数据终端准备	DTR
5	信号地	SG
6	数据设备准备好	DSR
7	请求发送	RTS
8	清除发送	CTS
9	振铃指示	RI

2.3.3 接口电路设计

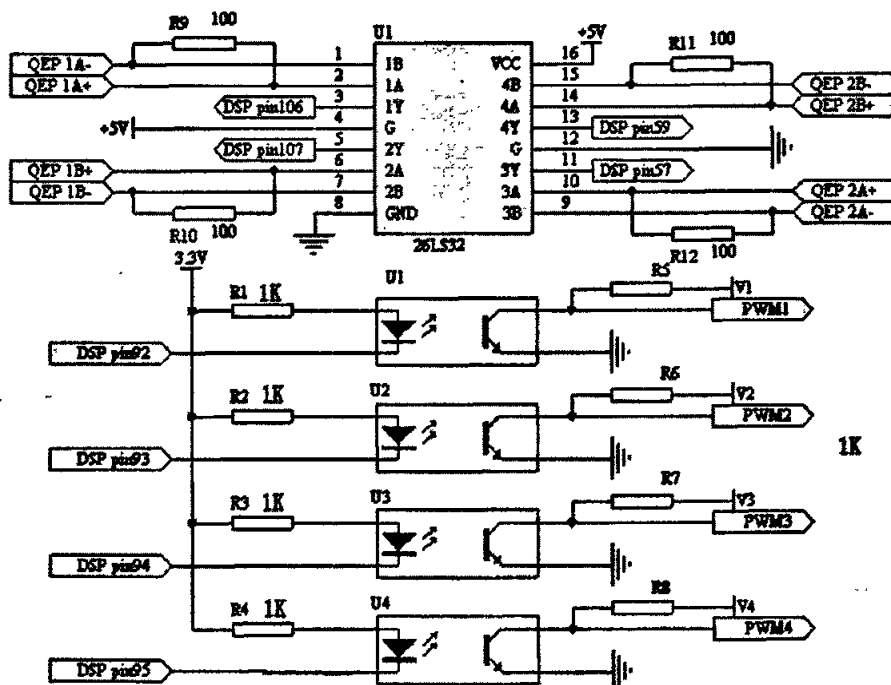


图 2-5 电机控制信号接口

fig 2-5 Interface to QEP and PWM

电机控制接口电路主要包括编码器脉冲信号检测和 PWM 脉冲输出电路(见图 2-5)。编码器输出的正交编码脉冲信号经整形电路 26LS32 整形后以单端方式送

入 DSP 的 QEP 接口。利用 DSP 的 106 脚、107 脚和 57 脚、59 脚可同时检测两对正交编码脉冲信号。利用 DSP 的 92 脚、93 脚和 94 脚、95 脚可输出两对带可编程死区的互补 PWM 信号。为了保护 DSP 的安全, PWM 信号经光电隔离器件与电机电路相连。图中 R5-R8 的阻值应根据具体要求选择。

模拟信号采集和输出接口电路如图 2-6 所示。共有 8 路输入通道和 4 路输出通道。输入的 0~5V 的信号经分压后送入 DSP 的 ADC 接口。DAC 通道利用一片 12 位 DAC 芯片 DAC7724 实现。

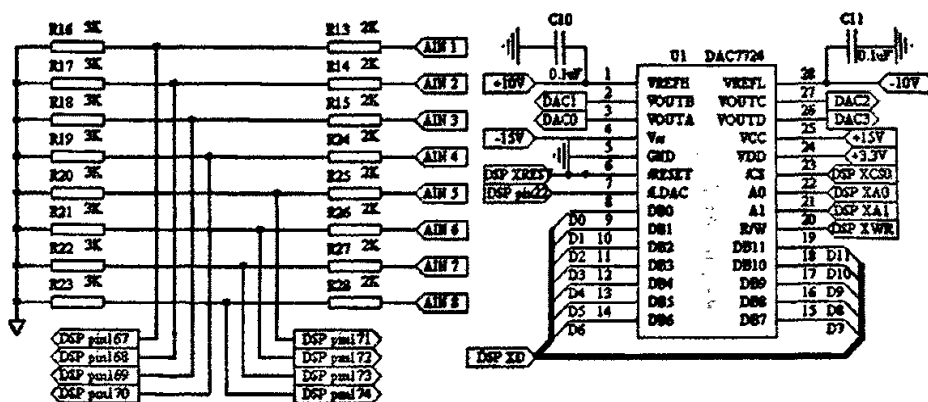


图 2-6 模拟信号接口

fig 2-6 Interface to analogue signals

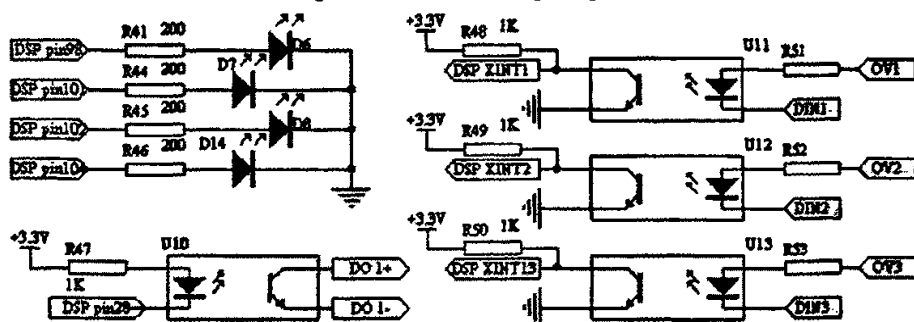


图 2-7 数字信号接口

fig 2-7 Interface to digital signals

图 2-7 所示为数字信号接口电路。图中绘出了全部 4 路 LED 控制电路、6 路数字量输出电路中的 1 路和全部 3 路数字量输入电路。数字量的输入/输出采用光电隔离方式以保证 DSP 芯片的安全。数字量输入信号分别接至 DSP 的 3 个外部中断接口, 因此, DSP 可以采用程序查询或者中断方式对输入信号进行处理。

RS-232-C 串行通信接口由 DSP 的串行通信接口 (SCI) 和一片 RS-232-C 控制芯片实现 (见图 2-8)。利用 RS-232-C 串行通信, 目标系统除可以与 PC/IPC 组成主

从式控制结构外，还可以适用于分布式控制结构。

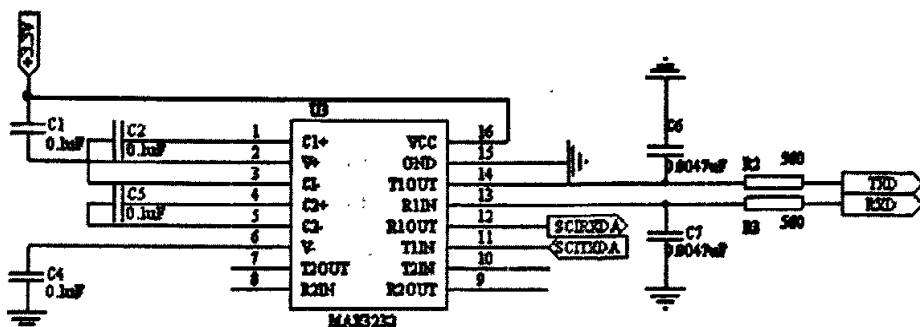


图 2-8 RS-232-C 串行通信接口

fig 2-8 Interface to RS-232-C communication

2.4. 本章小结

本章针对机器人控制中的常用器件和技术设计了移动机器人控制器的硬件系统。首先,确定了系统设计方案。然后,分析了 TMS320F2812 型数字信号处理器的体系结构并设计了 DSP 最小系统。最后,分析了机器人控制相关的接口技术并设计了接口电路。

该系统可独立执行不同传感器信号采集、电机闭环调速以及系统运行状态的光电显示,并具有数据通信接口,可以适用于集中式、主从式和分布式控制结构。

第3章 控制器系统软件设计

本章完成了移动机器人控制器的系统软件设计。首先,对实时系统与前后台式系统进行了比较,指出了实时系统的优势。第二,分析了 $\mu\text{C/OS-II}$ 的体系结构。第三,在F2812 DSP上移植了 $\mu\text{C/OS-II}$,并进行了系统测试。最后,分析了 $\mu\text{C/OS-II}$ 下驱动程序的设计方法,并设计了主要设备的驱动程序。

3.1 实时系统与前后台系统的比较

3.1.1 前后台系统

前后台系统又称超循环(Super-Loops)系统,其特征是应用程序是一个无限循环,循环中调用若干函数完成相应的操作,这部分可以看成后台行为(background)。中断服务程序处理异步事件,这部分可以看成前台行为(foreground)。后台也叫做任务级,前台也叫中断级。时间相关性很强(Time Critical)的关键操作是靠中断服务来完成的。因为中断服务提供的信息一直要等到后台程序走到该处理这个信息这一步时才能得到处理,这种系统在处理信息的及时性上,比实际可以做到的要差,这个指标称作任务级响应时间。最坏情况下的任务级响应时间取决于整个循环的执行时间。因为循环的执行时间不是常数,所以程序经过某一特定部分的准确时间也是不能确定的。更糟的是,如果程序修改了,循环的时序也会受到影响。

3.1.2 实时系统

基于实时内核的系统称为实时系统。实时内核也称为实时操作系统或RTOS。根据系统实时性的强弱,实时内核可分为软实时系统和硬实时系统。软实时系统的宗旨是使各个任务运行得越快越好,不限定某一任务的完成时间。在硬实时系统中,各任务不仅要执行无误而且要做到准时。大多数实时系统是二者的结合。

按照任务调度的方式的不同,可将实时内核分为可剥夺型(preemptive)内核和不可剥夺型(non-preemptive)内核。不可剥夺型内核允许每个任务占用

CPU 直到该任务自愿放弃 CPU 使用权。优点是响应中断快、任务级响应时间比前后台系统快、几乎不需要保护共享数据。最大缺陷在于任务级响应时间是不确定的。因此,当系统响应时间很重要时,要使用可剥夺型内核。在这种系统中,高优先级的任务一旦就绪,总能得到 CPU 的控制权。若一个运行着的任务使一个比它优先级高的任务进入了就绪态,当前任务的 CPU 使用权就被剥夺了,或者说被挂起了,那个高优先级的任务立刻得到了 CPU 的控制权。如果是中断服务子程序使一个高优先级的任务进入就绪态,中断完成时,中断了的任务被挂起,优先级高的那个任务开始运行。可剥夺型内核使得任务级响应时间得以最优化。

实时系统的结构如图 3-1 所示,系统由下向上可划分为 2 个层次,即硬件层和软件层。软件层又包括系统软件层和应用软件层。系统软件层包括与处理器相关部分、与处理器无关部分(内核)和与应用相关部分。其中,处理器相关部分用于实现实时内核与处理器硬件的接口。与应用相关的部分提供应用程序与实时内核的接口。实时内核提供系统系统服务,用户编写的应用程序则使用这些服务完成特定的功能。

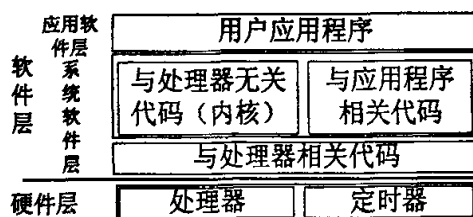


图 3-1 实时系统的结构

fig 3-1 Architecture of real-time system

实时系统,顾名思义是以运行的实时性见长的,尤为重要;实时操作系统支持系统功能的可剪裁性,极大地提高了控制系统的开放性。它的使用使得实时应用程序的设计和扩展变得容易,不需要大的改动就可以增加新的功能。通过将应用程序分割成若干独立的任务,RTOS 使得应用程序的设计过程大为简化。

内核通过一定的调度算法可以实现多任务运行。多任务运行很像前后台系统,但后台任务有多个。多任务运行使 CPU 的利用率得到最大的发挥,并使应用程序层次化、模块化、更易于设计与维护。

不过,使用实时内核会增加系统开支,包括内核的价格、额外的 ROM/RAM 开销、2 到 4 个百分点的 CPU 额外负荷。

考虑到移动机器人工作环境复杂、实时性和开放性要求较高以及价格因素,

本系统采用基于实时内核 $\mu\text{C/OS-II}$ 的结构^[37]。

3.2 $\mu\text{C/OS-II}$ 的体系结构

$\mu\text{C/OS-II}$ 是一款可移植、可剪裁、多任务、可剥夺型实时内核，提供了多项系统服务，包括任务管理、系统时钟服务、内存管理、消息机制和异常处理等。

3.2.1 $\mu\text{C/OS-II}$ 的任务管理机制

$\mu\text{C/OS-II}$ 内核的主要功能是对任务进行调度和管理。任务可分为系统任务和用户任务两类。内核将全部任务的优先权划分为 64 个等级，并为每个任务分配一个唯一的优先级。

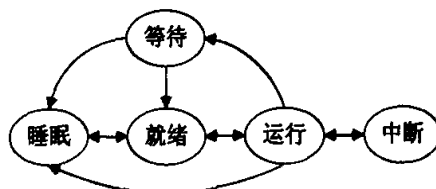
3.2.1.1 任务 $\mu\text{C/OS-II}$ 可以管理多达 64 个任务，其中包括两个优先级最低的系统任务。一个任务由 3 部分组成，包括任务代码和用来保存任务属性的任务控制块、用来保存任务工作环境的任务堆栈。

(1) 任务代码就是一个包括若干临界段的 C 函数，该函数返回值类型必须为 void，其结构通常是一个无限循环。代码的临界段也称为临界区，一旦这部分代码开始执行，则不允许被中断。为此，在进入临界段之前要关中断，代码执行完毕以后要立即开中断。临界段的运行时间会影响系统实时性。任务代码的一般结构如下：

```
void YourTask (void *pdata)
{
    for (;;) { ..... } /* 用户代码 */
}
```

(2) 任务状态 $\mu\text{C/OS-II}$ 中任务有五种状态，分别是睡眠、就绪、运行、等待和中断，在任一给定的时刻，任务一定处于这五种状态之一。睡眠态，指任务仅以代码形式驻留在程序空间之中，而不参与对 CPU 使用权的竞争。就绪态是指任务在等候 CPU 使用权，一旦获得 CPU 使用权即进入运行态。等待态指任务正在延时或等待某个事件的发生。处于运行态的任务在运行期间如果被中断则进入中断服务态，此时该任务被挂起。中断服务子程序可能会报告一个或多个事件的发生，而使一个或多个任务进入就绪态。因此，中断返回前内核要重新考量 CPU 使用权的归属，也就是说，被中断的任务未必能得以继续运行。任务状态的转换关系如图 3-2 所示。

(3) 任务控制块 是一个数据结构，每个任务有一个唯一的任务控制块。当

图 3-2 μ C/OS-II 中任务状态的关系fig 3-2 Relationship between tasks' states in μ C/OS-II

任务的 CPU 使用权被剥夺时， μ C/OS-II 用它来保存该任务的状态。当任务重新得到 CPU 使用权时，任务控制块能确保任务从当时被中断的那一点丝毫不差地继续执行。

(4) 就绪表 就绪表的结构如图 3-3 所示。全部 64 个优先级被分成 8 个组，每组 8 个优先级。每个优先级对应 OSRdyTbl[] 中的一位，每个组对应 OSRdyGrp 的一位。当某个组中的某个任务进入就绪态时，OSRdyTbl[] 中相应的位被置 1，同时，OSRdyGrp 中与该组对应的位被置 1。内核依次查询 OSRdyGrp 和 OSRdyTbl[] 即可确定就绪任务的最高优先级。

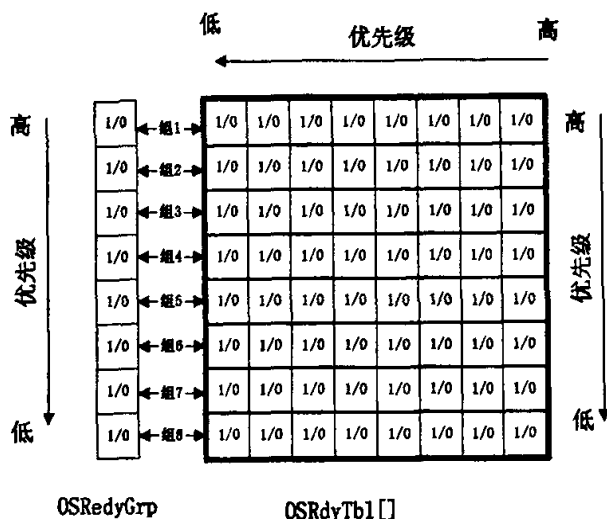


图 3-3 任务就绪表的结构

fig 3-3 Structure of Ready Table

3.2.1.2 任务调度 μ C/OS-II 中的任务调度包括任务级调度和中断级调度。在任务级任务调度中，当前任务主动放弃 CPU 的使用权。然后，由任务调度器找到具有最高优先权的就绪任务，并引发一次软件中断。在中断服务程序中将 CPU 各寄存器的当前值（即断点数据）保存在放弃 CPU 使用权的任务的堆栈中。最后，从待运行任务的堆栈中恢复其断点数据，实现任务的切换。在中断级任务调度中，

当前任务的断点数据在进入中断时被保存在其堆栈中，在退出中断时，则恢复当前时刻最高优先级任务的断点数据，使其获得 CPU 使用权，这一点正是 $\mu C/OS-II$ 内核中“可剥夺”性的体现。

任务级调度包括优先级查询和任务切换两个步骤。当内核决定执行任务调度时，首先查询当前就绪任务的最高优先级并判断该任务是否是当前正在运行的任务，以避免不必要的任务调度。如果最高优先级任务不是正在运行的任务，则内核执行任务切换。任务切换是通过一个软中断来实现的，首先，将被挂起任务的微处理器寄存器推入堆栈，然后将较高优先级的任务的寄存器值从栈中恢复到寄存器中。中断返回后，新的任务即取得了 CPU 使用权。

中断级任务调度发生在中断服务程序返回前。内核查询就绪的任务中是否有任务优先级高于被中断的任务。如果没有则直接返回，否则，执行任务切换。由于进入中断服务程序时已经保存了被中断任务的断点数据，因此此时只需恢复待运行的任务的断点数据并返回即可。

3.2.2 $\mu C/OS-II$ 的系统时钟

为实现延时和确认超时， $\mu C/OS-II$ 需要一个系统时钟。这是由硬件定时器产生的毫秒级定时中断来实现的，该定时中断称为时钟节拍。硬件定时器可以是独立于处理器的分立式定时器，也可以是处理器集成的定时器。定时中断的周期决定了操作系统可为应用软件提供的最小时间刻度，该周期越小则处理器的额外开销越大。

时钟节拍的中断服务函数的结构如下：

```
void OSTickISR(void)
{
    保存处理器寄存器的值；
    调用 OSIntEnter() 或是将 OSIntNesting 加 1；
    调用 OSTimeTick();
    调用 OSIntExit();
    恢复处理器寄存器的值；
    执行中断返回指令；
}
```

3.2.3 任务的同步与通信

任务间的同步依赖于任务间的通信，任务间的通信通过“事件”来实现。

$\mu C/OS-II$ 中用于任务间通信的事件有信号量、消息邮箱、消息队列以及信号量

集。比较常用的是信号量和消息邮箱。消息邮箱用于在任务间传递一个共享数据的指针。

信号量用于标志一个或一组共享资源是否可用， $\mu\text{C/OS-II}$ 中的信号量由一个 16 位计数器和等待任务表组成。根据信号量的用途的不同，在建立信号量的时候应给该计数器赋以不同的初值。如果信号量是用来表示一个或者多个事件的发生，那么该信号量的初始值应设为 0。如果信号量是用于对共享资源的访问，那么该信号量的初始值应设为 1（例如，把它当作二值信号量使用）。如果该信号量是用来表示允许任务访问 n 个相同的资源，那么该初始值显然应该是 n 。

消息邮箱可以使一个任务或者中断服务子程序向另一个任务发送一个指针型的变量。该指针指向一个包含了特定“消息”的数据结构。消息队列等效于若干个消息邮箱^{[38][39]}。

3.3 $\mu\text{C/OS-II}$ 在 F2812 上的移植与测试

3.3.1 实时内核的移植

$\mu\text{C/OS-II}$ 在 F2812 上的移植工作主要是编写系统软件层中与处理器相关部分的代码。这些代码包含在三个文件中，分别是 OS_CPU.H、OS_CPU_A.ASM、OS_CPU_C.C。

3.3.1.1 任务的临界段 在 $\mu\text{C/OS-II}$ 中用两个宏 OS_ENTER_CRITICAL() 和 OS_EXIT_CRITICAL() 来实现临界段。在文件 OS_CPU.H 中编写两个宏的定义如下：

```
#define OS_ENTER_CRITICAL() asm(" SETC INTM ")
#define OS_EXIT_CRITICAL()  sm("  CLRC INTM ")
```

3.3.1.2 任务堆栈初始化 任务建立以后，应对其堆栈进行初始化。在 F2812 在进入中断时会自动将关键寄存器按如下顺序压入堆栈：ST0:T、AL:AH、PL:PH、ARO:AR1、ST1:DP、IER:DBGSTAT 和 PC(L):PC(H)，用户程序负责将其他寄存器的内容按照自定义的顺序压栈。任务堆栈的初始化就是将任务所涉及的 CPU 寄存器的期望值按照上述顺序保存任务堆栈中。

任务堆栈初始化函数 OSTaskStkInit() 在 OS_CPU_C.C 文件中编写。

3.3.1.3 任务级任务切换 当前任务放弃 CPU 的使用权时，内核会执行一次任务调度。任务级任务调度函数 OS_Sched() 确定了即将取得 CPU 使用权的任务以后，

首先调用宏 `OS_TASK_SW()`，引发一次软件中断，在中断服务函数中执行任务切换。具体实现方式如下。

首先，在 `OS_CPU.H` 中定义一个宏 `OS_TASK_SW()`：

```
#define OS_TASK_SW() TRAP #31,
```

在 F2812 DSP 中，#31 中断是一个由用户定义的中断。将该中断向量设置为函数 `OSCtxSw()` 的入口，在该函数中实现任务级任务切换。

然后，在 `OS_CPU_A.ASM` 文件中编写 `OSCtxSw()` 函数，部分代码如下。

```
MOVL XAR2, @_OSTCBCur;
MOVB XAR0, #0;
MOV AL, @SP
MOV **XAR2[AR0], AL;
LCR _OSTaskSwHook;
MOVL XAR2, @_OSTCBHighRdy;
MOVL @_OSTCBCur, XAR2;
MOV ACC, @OSPrioHighRdy;
MOV @_OSPrioCur, ACC;
MOVL XAR2, @_OSTCBHighRdy;
MOV AL, **XAR2[0];
MOV SP, AL;
```

3.3.1.4 中断级任务切换 在 $\mu C/OS-II$ 中，退出中断服务程序之前会调用函数 `OSIntExit()`，这个函数判断就绪表中是否有比被中断的任务优先级更高的任务。在中断嵌套层数为 0，且调度器未被锁定的情况下，`OSIntExit()` 调用中断级任务切换函数 `OSIntCtxSw()` 实现任务切换。由于 TMS320F2812 DSP 在中断时会自动保护重要寄存器的内容，因此，在将 $\mu C/OS-II$ 移植到 TMS320F2812DSP 上时，如果不需额外保护其它寄存器的话，`OSIntCtxSw()` 与 `OSCtxSw()` 的函数体是相同的。

3.3.1.5 运行系统中优先级最高的任务 内核通过调用函数 `OSStart()` 启动多任务。在此之前，系统中应至少已建立了一个任务。`OSStart()` 调用函数 `OSStartHighRdy()` 使系统中最高优先级的任务开始运行。函数 `OSStartHighRdy()` 的工作是将系统中的最高优先级任务的堆栈栈顶指针传递给 SP 寄存器。

3.3.1.6 $\mu C/OS-2$ 的系统时钟 利用 TMS320F2812 DSP 的 CPU 定时器 1 产生 5ms 定时中断为 $\mu C/OS-II$ 提供时基，该定时中断的中断服务程序是 `OSTickISR()`。这个函数的作用有两个，其一是计数时钟节拍，其二是管理正在延时的任务，在

退出中断前, 还有可能引发一次中断级任务调度, 因此在函数入口处应确保相关寄存器的内容都受到保护^[40-42]。

3.3.2 系统测试

内核在目标系统上移植完成后, 应对系统进行测试。本文以 TMS320F2812 DSP 的模拟信号采集和输出实验对系统进行测试。实时内核移植与测试代码均在 TI 集成开发环境 (Integrated Development Environment, IDE) CCS2.0 下编写^{[43][44]}。

本测试建立了两个任务 Task1() 和 Task2(), 分别在通用 I/O 引脚 GPIOF3、GPIOF12 上产生方波。Task1() 和 Task2() 的优先级分别为 0 和 1, 通过一个互斥型信号量 flag 同步。这两个用户任务的工作流程如表 3-1 所示。

表 3-1 task1 和 task2 的工作流程

Table 3-1 Steps of task1 and task2

	task1()	Task2()
第一步	请求信号量	请求信号量
第二步	翻转 GPIOF3	翻转 GPIOF12
第三步	延时 30 个节拍	立即发送信号量
第四部	发送信号量	延时 10 个节拍
第五部	延时 10 个节拍	循环结束
第六部	循环结束	

表 1. 用户任务 Task1() 和 Task2() 的工作流程

图 3-4 所示为程序运行后, 在示波器上观察到的 GPIOF3(上)、GPIOF12(下) 引脚上的电平变化情况。其中, 图 3-4(a)所示为 Task2() 无限期待信号量 flag 情况下的引脚电平变化。图 3-4(b)所示是 Task2() 等待信号量 flag 的时限设为 20 个节拍时的引脚电平变化。

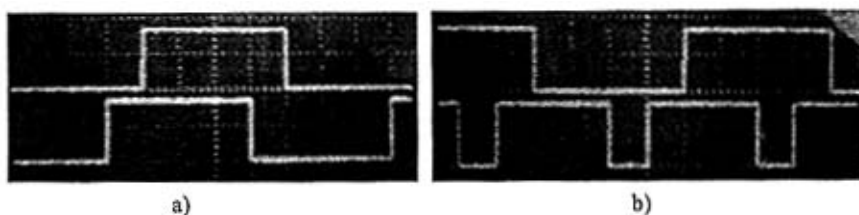


图 3-4 测试程序的运行结果

fig 3-4 Results of test routine

从图 3-4 (a)可见, Task1() 将 GPIOF3 上的电平翻转后, 延时 30 个节拍 (对应图中三格) 发送信号量, 由于 Task2() 在无限期地等待信号量, 所以, 当 Task1() 延时期满后, Task2() 立即获得信号量, 并翻转 GPIOF12 引脚电平。在图 3-4 (b) 中, Task2() 延时 10 个节拍 (一格) 后, 申请信号量, 但是由于 Task1() 未释放信

号量, 所以进入等待状态, 等待 20 个节拍后, 放弃等待, 继续运行。

程序运行结果表明, 实时内核已能正常工作, 移植成功。

3.4 驱动程序设计

驱动程序属于系统软件层中与应用相关的部分。

3.4.1 μ COS-II 中设备驱动的实现

μ COS-II 设备驱动程序的实现方式分为两种: 一种是轮询(Polling)方式, 另一种是中断方式。

最简单的设备驱动方式是轮询。在这种方式中, 对设备的操作完全由 CPU 掌握, 外部设备则完全处于被动的状态。当 CPU 要对某一外部设备进行操作, 但该设备暂时不可用时, CPU 有两种对策: 一种是睡眠一段时间(或先进行其他操作)以后再试; 另一种是锲而不舍地试探下去, 直至完成, 称为“连轴转(Spin)”。后者对系统实时性的破坏是致命性的。中断式方式中, 设备的工作是主动的, 当设备完成数据处理后通知 CPU 读取处理结果。

在 μ COS-II 下编写设备驱动程序包括如下 5 部分工作。(1) 把访问设备需要的相关数据信息封装成合适的数据结构, 便于管理。(2) 定义一组跟踪目前硬件和设备驱动状态的变量, 这些变量的定义是与第一点对应的。(3) 编写设备初始化函数。(4) 为用户端的有关操作编写接口函数, 仔细合理地定义接口函数的形式、参数和返回值, 最主要的用户程序操作就是读和写。(5) 编写中断服务程序^[45]。

3.4.2 驱动程序代码设计

针对第二章所设计的硬件系统中的主要设备, 设计了 7 组驱动程序。根据这些设备各自的工作特点, 采用轮询、中断分别实现设备控制。

3.4.2.1 AD 转换器驱动 包括初始化函数 `DRIVE_AD_INIT()`、转换启动函数 `DRIVE_AD_START()`、转换完成通知函数 `DRIVE_AD_COMPLETE()`。

(1) `DRIVE_AD_INIT()` 完成 ADC 工作方式选择、时钟配置、启动方式选择。

(2) `DRIVE_AD_START()` 启动 ADC, 首先将相关标志位清零, 表示数据暂时不可用, 然后将模拟信号转换为 12 位的无符号数, 并存入 FIFO 缓冲区。

(3) `DRIVE_AD_COMPLETE()` 这是一个 ISR 函数, 首先, 读取 FIFO 缓冲

区中的数据，并将其换算成 0~3V 的浮点型电压值。然后，将相关标志位置位，表示数据可用，其他任务可以共享这些数据。

3.4.2.2 QEP 驱动 包括初始化函数 `DRIVE_QEP_INIT()`、数据读取函数 `DRIVE_QEP_READ()`。

(1) `DRIVE_QEP_INIT()` 完成时基选择、计数器配置、启动。

(2) `DRIVE_QEP_READ()` 读取计数器的当前值，并重载计数器初值。当前值与计数器初值的差除以 4 就是实际的脉冲数，可以用来计算角位移或者线位移。

3.4.2.3 RS - 232 串行通信接口驱动 包括初始化函数 `DRIVE_SCI_INIT()`、编码函数 `DRIVE_SCI_ENCODE()`、解码函数 `DRIVE_SCI_DECODE()`、数据发送函数 `DRIVE_SCI_TXD()`、数据接收函数 `DRIVE_SCI_RXD()`。

(1) `DRIVE_SCI_INIT()` 完成时钟预定标、波特率设置、工作方式选择、设备级中断配置、数据格式设置。

(2) `DRIVE_SCI_ENCODE()` 将数据进行编码，以备传输。串口通信的数据以帧为单位，每帧中的数据最多为 8 位（一个字节），而实际系统中需传送的数据往往是整形数据、浮点型数据等多字节型数据。因此，在进行数据传输前，需将待传的数据编码，即用若干个字节的无符号数来表示一个实际的数值。

(3) `DRIVE_SCI_DECODE()` 如前所述，传输线上传来的数据仅是一组编码，就单个帧的数据而言是没有意义的。`DRIVE_SCI_DECODE()`的作用在于将接收到的数据帧复原为它原来的形式即整形或浮点型数据。

(4) `DRIVE_SCI_TXD()` 调用 `DRIVE_SCI_ENCODE()`将系统级缓冲区中的数据编码，然后发送到设备级 FIFO 缓冲区。FIFO 中的数据以字节为单位存放，每个字节的数据以移位方式被发送到传输线。

(5) `DRIVE_SCI_RXD()` 这是一个 ISR 函数。传输线上的数据以移位方式被下载到移位寄存器中，每下载 8 个数据位则将其以字节形式存入设备级接受 FIFO 缓冲区。当该缓冲区中的数据达到预先设置的阈值时，设备发出中断请求，表示数据接收完毕。ISR 函数 `DRIVE_SCI_RXD()`负责调用 `DRIVE_SCI_DECODE()`将设备级接受 FIFO 缓冲区中的数据解码，然后转移到系统级缓冲区。

3.4.2.4 DA 转换器驱动 包括初始化函数 `DRIVE_DA_INIT()`、数据转换函数

DRIVE_DA_UPDATE()。

(1) DRIVE_DA_INIT() 完成 GPIO 与 DAC 电路相关的引脚的工作方式的设置。

(2) DRIVE_DA_UPDATE() 将某个浮点型数据转换成 12 位无符号整数, 并用来更新 DAC 电路的某个输出通道的值。完成数据转换、通道选择、数据锁存等。

3.4.2.5 PWM 设备驱动 包括初始化函数 DRIVE_PWM_INIT()、输出控制函数 DRIVE_PWM_UPDATE()。

(1) DRIVE_PWM_INIT() 完成时基选择、定时器工作方式选择、死区时间设置、输出极性、比较器配置等。

(2) DRIVE_PWM_UPDATE() 通过修改比较器寄存器改变 PWM 信号的占空比, 载波频率等。

3.4.2.6 开关量 I/O 设备驱动 包括 LED 初始化函数 DRIVE_LED_INIT()、LED 控制函数 DRIVE_LED_ON()、开关量信号数据接口初始化函数 DRIVE_DO_INIT()和开关量信号输出函数 DRIVE_DO_OUT()。

(1) DRIVE_LED_INIT()和 DRIVE_DO_INIT() 完成相关 GPIO 引脚的工作方式设置和初始状态设置。

(2) DRIVE_LED_ON()、DRIVE_DO_OUT()将某个或某些引脚设置成期望状态, 以点亮/熄灭 LED 或者控制某个设备。

3.5 本章小结

基于 DSP 和 RTOS 的机器人控制系统包括硬件和软件两个方面。软件方面包括操作系统移植与测试、驱动程序设计和应用程序设计。

本章在第二章所设计的 DSP 硬件平台上移植了嵌入式实时内核, 并设计了主要设备的驱动程序。实时内核的主要功能是进行任务调度和管理, 是应用程序设计和运行的基础。本章首先比较了基于实时内核的实时系统与前后台式系统的特性, 指出了实时内核的优势。第二, 进一步分析了 $\mu\text{C}/\text{OS} - \text{II}$ 的体系结构。第三, 实现了 $\mu\text{C}/\text{OS} - \text{II}$ 在 DSP 上的移植, 并对系统进行了测试。最后, 分析了在 $\mu\text{C}/\text{OS} - \text{II}$ 下的驱动程序设计方法, 设计了主要设备的驱动程序。

第4章 直线倒立摆运动控制

本章基于第二、三章所设计的控制器实现了倒立摆的运动平衡控制。首先,分析了单级直线倒立摆的机电结构和运动特性。然后,以深圳固高科技有限公司生产的 GLIP2001 型倒立摆本体为控制对象,应用经典控制理论和现代控制理论实现了倒立摆起摆和稳摆控制。采用 VC 设计了人机界面程序,实现了倒立摆参数的图形化显示、数据存档和期望状态的手动设定。

4.1 倒立摆的机电结构及数学模型

二十世纪50 年代,麻省理工学院(MIT)的控制论专家根据火箭发射助推器原理设计出一级倒立摆实验设备。

倒立摆是一种典型的非线性自然不稳定系统,具有高阶次、多变量、强耦合等特点。倒立摆系统与足式机器人、双轮自平衡机器人在结构和运动特性上的相似性,使其能够比较真实地模拟机器人的运动平衡控制。因此,倒立摆系统的运动控制对于移动机器人控制技术的研究具有重要的参考价值。

4.1.1 倒立摆的数学模型

系统建模可以分为两种:实验建模和机理建模。实验建模就是在研究对象上施加一系列已知的激励信号并通过传感器检测其可观测的输出,应用数学手段建立起系统的输入-输出关系。机理建模就是在了解研究对象的运动规律的基础上,通过物理、化学和数学手段建立起系统内部的输入-输出关系。

对于倒立摆系统,由于其自身是结构性不稳定系统,实验建模存在一定的困难。但是,经过小心的假设忽略掉一些次要因素后,倒立摆系统就是一个典型的运动刚体系统,可以在惯性坐标系内应用经典力学理论建立系统的动力学方程。

在忽略了空气阻力和各种摩擦力之后,可将直线一级倒立摆抽象成由小车和匀质杆组成的系统(见图 4-1)。对摆杆和小车的受力进行分析可得到系统的两个

运动方程^{[46][47]}。

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \quad (4-1)$$

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -m\ddot{x}\cos\theta \quad (4-2)$$

式中: x —小车位移, 左方为正方向;

\dot{x} —小车速度;

ϕ —摆杆与竖直向上方向的夹角, 顺时针为正方向;

$\dot{\phi}$ —摆杆角速度。

F —作用在小车上的外力, 左方为正方向。

$M=1.0\text{Kg}$, 小车质量;

$m=0.1\text{Kg}$, 摆杆质量;

$b=0.1\text{N/m/s}$, 小车摩擦系数;

$l=0.25\text{m}$, 摆杆转动轴心到质心的长度;

$J_m=0.0034\text{Kg}\cdot\text{m}^2$, 摆杆转动惯量;

参数具体数值来自 GLIP2001 型倒立摆本体。

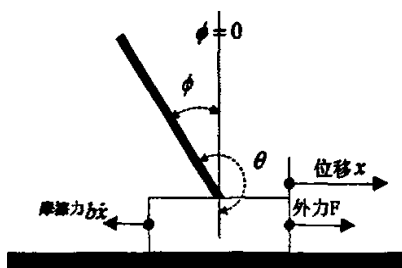


图4-1 倒立摆系统的简化模型

fig4-1 Simplified model of Inverted Pendulum

设 $\theta = \phi + \pi$ (ϕ 是摆杆与垂直向上方向之间的夹角), 当 $\phi \ll 1(\text{rad})$ 时, 有 $\cos\theta = -1, \sin\theta = -\phi, (\frac{d\theta}{dt})^2 = 0$ 。用 u 来代替 F 表示施加在系统上的外力, 可得线性化运动方程组如下:

$$\begin{cases} (I+ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \\ (M+m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \end{cases} \quad (4-3)$$

系统的状态空间模型如式 (4-4) 所示。

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(J_m+ml^2)b}{J_m(M+m)+Mml^2} & \frac{m^2gl^2}{J_m(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{J_m(M+m)+Mml^2} & \frac{mgl(M+m)}{J_m(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{J_m+ml^2}{J_m(M+m)+Mml^2} \\ 0 \\ \frac{ml}{J_m(M+m)+Mml^2} \end{bmatrix} u \quad (4-4)$$

4.1.2 倒立摆控制系统的机电结构

本文所述的倒立摆控制系统由第二、三章设计的机器人控制器和 GLIP2001 实验系统的倒立摆本体及驱动器组成。其机电结构（见图 4-2）包括小车、摆杆、滑轨、传动皮带、滑轮、行程开关、编码器、电机及驱动器、下位机（控制器）和上位机等。摆杆的一端（固定端）通过一个轴承安装在小车上，因此，摆杆可以固定端为圆心在竖直平面内作近似无阻尼旋转运动。电机输出的力矩通过皮带和滑轮作用在小车上，从而使由小车和摆杆组成的系统受到水平外作用力，通过小车与摆杆间的相互作用将外力作用在摆杆的固定端，使摆杆的两端产生相对运动，从而影响摆杆的位姿。

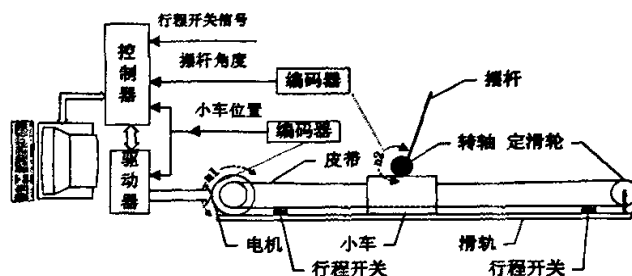


图 4-2 倒立摆的机电结构图

Fig4-2 Electromechanical structure of inverted pendulum

小车在水平外力的作用下会沿滑轨作直线运动，但是滑轨的长度是有限的（在本实验所使用的倒立摆系统中，滑轨长度为 0.6 米），所以在滑轨的两端各安装了一个光电开关，当小车滑过光电开关时，开关闭合发出报警信号，从而避免小车“撞墙”造成电机长时间堵转。

系统中共使用了 2 个增量式光电编码器用于检测角位移信号。其一与摆杆的固定端同轴，当摆杆绕固定端旋转时，编码器输出正交编码脉冲序列，脉冲的数量对应于摆杆转过的角度。摆杆每转过 360 度，编码器输出 600 个脉冲。其二是内置在电机内部的，与电机转子同轴，转子每转过 360 度编码器输出 2500 个脉冲，工作原理与摆杆编码器相同。

驱动器和电机组成倒立摆的执行系统。驱动器实际上执行电机的功率驱动和闭环调速双重功能，使电机按照期望的转速工作，因此须将电机的编码器信号反馈到驱动器。此外，驱动器还会将电机的状态参数如报警信号等传递给控制器，并接受控制器发出的控制指令。

控制器分别通过编码器检测摆杆和电机转子的角位移，并换算成摆杆的角度、小车的水平位移和速度以及加速度。此外，控制器的输入信号还包括行程开关的输出信号和驱动器上传的电机状态参数。在此基础上，依据一定的控制算法做出决策并发出控制指令。控制器的控制指令全部下达到驱动器，这些指令包括电机启动、警报清除、期望转速等。

PC 机提供人机界面，将摆杆角度、角速度和小车位置、速度参数进行实时图形化显示和存档，同时接受来自键盘和鼠标的设定指令，并下达给控制器。其本身不执行控制功能。

控制器与倒立摆本体的电气连接主要涉及控制器与驱动器、控制器与编码器、控制器与行程开关间的接口，如图 4-3 所示。

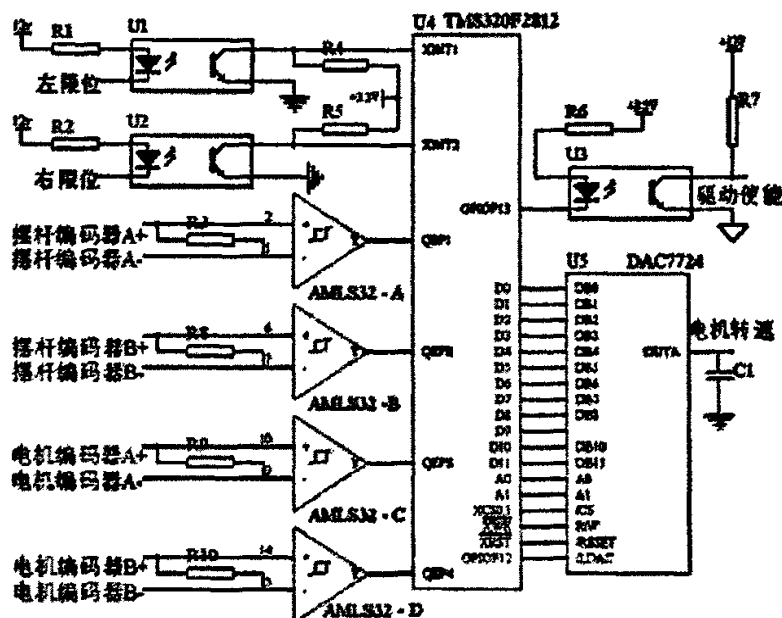


图 4-3 控制器与倒立摆的信号接口

Fig4-3 Interface between controller and Inverted Pendulum

(1) 行程开关信号接口 导轨的两端各安装了一个光电开关，当小车滑过光电开关时，开关闭合发出报警信号。因此，这两个光电开关起到了行程开关的作用，其输出信号是对小车的限位信号，即控制器在接收到此信号后，应立即限制小车继续滑向导轨的尽头。为保证控制器对限位信号处理的实时性，将两路光电开关输出的限位信号通过光电耦合器分别输入到 DSP 的外部中断接口 XINT1 和 XINT2。

(2) 编码器接口信号 GLIP2001 倒立摆本体采用 2 个增量式光电编码器来检测

电机和摆杆的旋转角位移。两个编码器的区别仅在于,摆杆编码器每周输出 600 个脉冲,电机编码器每周输出 2500 个脉冲。每个编码器的输出信号包括 A 相、B 相、Z 相共 3 路差分信号,分别由 A+和 A-、B+和 B-、Z+和 Z-组成,共 6 根线,其中 A、B 两相为正交编码脉冲信号。所谓正交编码脉冲信号即根据编码器轴的旋转方向的不同,A 相脉冲相位或超前或滞后于 B 相脉冲 90 度。Z 相为索引信号,其高低电平标志编码器轴的旋转方向^{[48] [49]}。在本系统中,由于 DSP 可以识别正交编码信号,所以无需使用索引信号。因此,只需分别将摆杆编码器和电机编码器的 A、B 两相差分信号连接到控制器接口电路中的 AMLS32 的 2 和 1、6 和 7、10 和 9、14 和 15 脚即可。

(3) 驱动器信号接口 驱动器与控制器间的信号包括开关量信号和模拟信号两种类型。驱动器发出的电机报警信号、来自控制器的电机启动信号、报警清除信号为开关量信号。控制器以模拟信号的形式向驱动器发出转速信号。由于控制器接口资源比较丰富,因此这些信号的连接比较容易。

(4) 串行通信接口 本实验采用 3 线制 RS-232 串口实现控制器与上位机通信。由于设计控制器的串口电路时已设定其为 DCE,考虑了其 PC 机的接口问题,因此,只需使用一条 9 芯电缆通过 DB9 连接器即可建立起控制器与上位机的通信链路。

4.2 倒立摆的运动控制

倒立摆运动控制涉及起摆和稳摆控制。倒立摆的起摆控制又称摆起控制,通常是使摆杆在外力作用下从自然下垂并静止的状态开始震荡,逐步增大振幅并使摆杆最终以适当角速度“冲”到竖直向上位置附近的过程。稳摆控制关注的是使摆杆保持在竖直向上位置附近,即倒立状态。

4.2.1 倒立摆控制的软件结构

倒立摆控制系统程序结构如图 4-4 所示,包括反馈信号 (x 、 ϕ) 拾取、起摆控制器、稳摆控制器、饱和控制单元和速度插补运算单元。在起摆过程中,当 ϕ 首次进入 $[-\phi_{TH}, \phi_{TH}]$ 区间时切换到稳摆控制器。饱和控制的作用是限制施加在小车上的外力,以确保系统的安全。

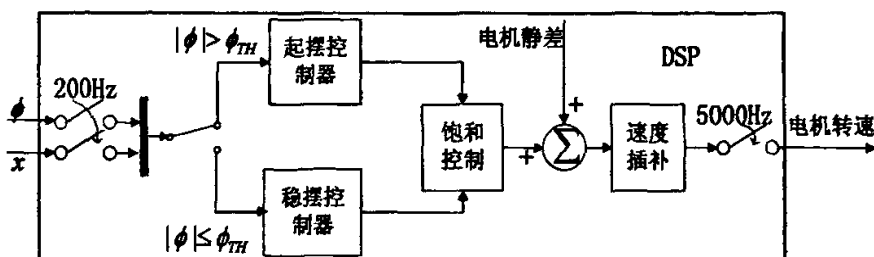


图 4-4 倒立摆控制系统的软件结构

Fig4-4 Software structure of Inverted Pendulum controller

由于伺服电机采用恒速工作方式，且物体所受外力可以由其加速度来表征，所以起摆控制器和稳摆控制器的输出为加速度信号。DSP 以 200Hz 的频率（采样频率）采样反馈信号，并以 5000Hz 频率（控制频率）向电机驱动器发出控制指令。速度插补运算的作用在于针对起摆控制器或稳摆控制器在每个“采样时刻”输出的期望加速度计算出在下一个采样时刻到来前的每个“控制时刻”电机的期望转速，并在每个“控制时刻”向驱动器发出相应的转速指令。

由软件结构图可知，倒立摆控制系统软件应包括以下任务：

(1) TASK_SAMP() 该任务首先调用驱动程序 DRIVE_QEP_READ() 读取两个编码器值，分别换算成摆杆角度、角速度、小车位置、小车速度信息。第二步，测试摆杆的角度，以判断接下来应该执行起摆控制程序还是稳摆控制程序。该任务每 5ms 执行一次。第三步，起摆控制器或稳摆控制器依据某一控制算法做出决策，输出“期望加速度”。第四步，对起摆或稳摆控制器的输出进行饱和控制。第五步，速度插补运算，其结果为共享数据。

(2) TASK_UPDT() 根据 TASK_SAMP() 的运算结果更新 DAC 的输出。该任务每 200us 执行一次。

(3) TASK_COMM() 首先调用串口编码函数 DRIVE_SCI_ENCODE() 将摆杆角度、角速度、小车位置、小车速度等参数打包成数据帧并组成数据块。第二步，调用数据发送函数 DRIVE_SCI_TXD() 将数据发送出去。该任务每 20ms 执行一次。

(4) TASK_GET_ORDER() 当上位机有数据包传下来时会引起 DSP 的串口中断，在 ISR 中，将互斥型信号量 FLAG_GET_ORDER 置位，以通知 TASK_COMM() 处理接收到的数据。TASK_COMM() 首先将数据包中的数据进行解码，并用来更新参考量。互斥型信号量 FLAG_GET_ORDER 用于串口 ISR 与

TASK_GET_ORDER()的通信。

此外, DSP 通过串行通信接口将摆杆角度、角速度和小车位置、速度共四个参数上传到 PC 机, 由人机界面程序进行可视化显示和数据存档(见图 4-5)。该界面采用 MFC 技术设计, 可以接受来自键盘或鼠标的设置指令, 并传达到下位机。这仅是一个界面程序, 不同于一般主从式系统中的上位机, 不执行控制任务, 倒立摆的全部控制由下位机完成。

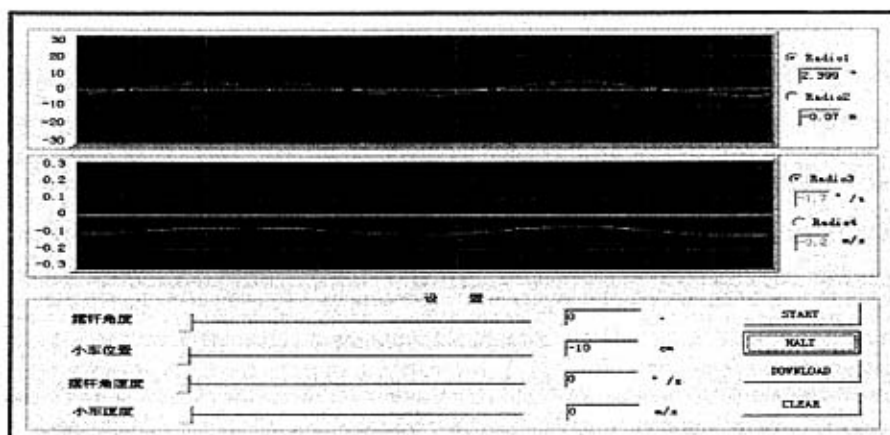


图 4-5 PC 上的监控界面

Fig4-5 Pc-based monitor window

4.2.2 起摆控制

众所周知, 在理想的实验条件下, 摆杆在振荡过程中的能量由动能和势能组成。初始状态下, 摆杆静止且质心处于最低点, 其动能、势能皆为0, 因此, 摆杆的初始能量为0。当摆杆摆到 $\phi = 0$ 位置时, 其质心位于最高点, 具有最大势能 mgL , 并同时具有一定的动能 $\frac{1}{2}J_m\dot{\phi}^2$, 由此可见摆杆的能量有所增加, 倒立摆的起摆控制实际就是一个向摆杆注入能量的过程。另一方面, 起摆过程中当摆杆接近平衡位置时其角速率 $\dot{\phi}$ 不能过大, 否则稳摆控制器的控制作用不足以“抓住”摆杆。综上所述, 倒立摆的起摆控制就是通过作用在小车上的外力向系统做功, 使摆杆的能量从零逐步增大到 mgL 的某个临域的过程。

起摆过程的控制流程如图4-6所示。其中 E 为摆杆动能与势能之和; a 为起摆控制器输出, 表征小车期望加速度, $a_+ > 0$ 、 $a_- < 0$; $c \geq 0$ 为常数, 用以补偿摆

起过程中的能量损耗。

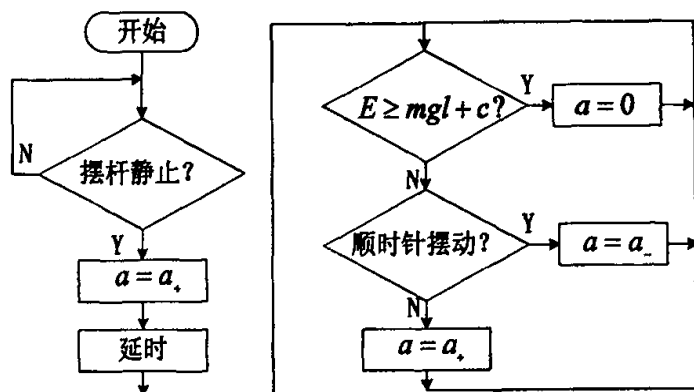


图 4-6 起摆控制流程图

Fig4-6 Steps of swinging-up

4.2.3 稳摆控制

本小节首先采用经典控制理论中的PID算法实现稳摆控制，然后应用现代控制理论中的LQR算法和4.2.1小节所述的起摆方法实现倒立摆的全过程控制。

4.2.3.1 基于PID方法的稳摆控制 PID 控制是最早发展起来的一种控制方法, 由于其结构简单、技术成熟, 至今仍广泛应用于各种控制系统中。

(1) 比例 (P) 调节 比例系数 K_p 的大小决定了比例调节器调节的快慢程度。

但 K_p 过大会使控制系统出现超调或振荡现象, K_p 过小又起不到调节作用。而且比例控制无法消除净差。

(2) 积分 (I) 调节 积分作用可消除余差。积分常数 K_i 的大小决定了积分作用强弱程度。积分作用通常使系统的稳定性下降。

(3) 微分 (D) 调节 当偏差 e 瞬间波动过快时, 微分调节器会立即产生响应, 来抑制偏差的变化, 使系统更趋于稳定, 改善了系统的动态性能。微分常数通常用 K_d 表示^[50]。

经过多次试验, 确定控制器参数为: $K_p = 0.1$, $K_i = 0.08$, $K_d = 0.01$ 。

(4) 稳摆控制实验 将摆杆缓慢扶起, 当摆杆的角度进入到 $[-20^\circ, 20^\circ]$ 临域时, 控制器开始进行稳摆控制。实验结果如图4-7所示。

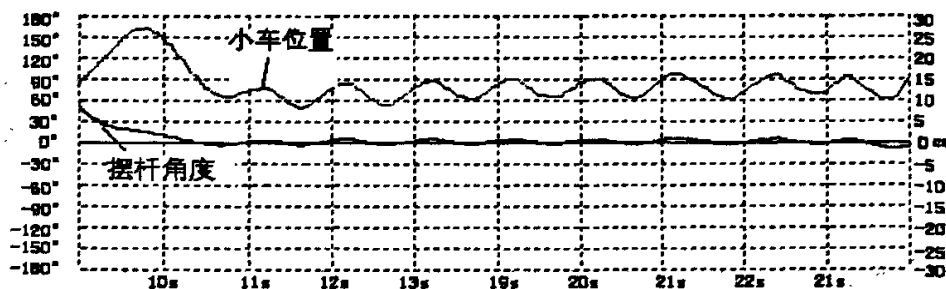


图 4-7 应用 PID 方法的倒立摆控制实验结果

Fig4-7 Experiment result of Inverted Pendulum control using PID

稳摆过程中摆杆的最大角度偏差在 $[-15^{\circ}, 15^{\circ}]$ 区间。小车起始位置距原点位置 13 厘米处，摆杆角度小于 60° 度时，控制器开始产生控制作用，使摆杆保持平衡。由图可见，小车的位置存在正方向的飘移，这是由于仅对摆杆的角度进行闭环调节，因此小车的位置是不可控的。运行一段时间后，小车会发生撞墙，然后在控制器的作用下回归原点位置。

4.2.3.2 基于 LQR 方法的稳摆控制

(1) LQR 控制器的工作原理 LQR 控制算法就是基于状态方程 $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ ，确定最优反馈增益矩阵 \mathbf{K}^* 和最优控制量 $\mathbf{u}^*(t) = -\mathbf{K}^* \mathbf{x}(t)$ ，使得控制性能指标

$$J = \int (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (4-5)$$

达到极小。其实质在于用较小的控制来维持较小的误差，达到能量和误差综合最优的目的。其中， \mathbf{Q} 为正定（或半正定）对称矩阵； \mathbf{R} 为正定对称矩阵。

式 (4-5) 中加权矩阵 \mathbf{Q} 和 \mathbf{R} 分别表征误差和能量损耗的相对重要性。由于误差与能量损耗之间存在着此消彼长的关系，所以 \mathbf{Q} 阵和 \mathbf{R} 阵的选择是相互制约、相互影响的。

一般来说， \mathbf{R} 阵固定，增大 \mathbf{Q} 阵中某一权值，可改善与之相对应的状态分量的动态响应过程，同时，加剧振荡，控制量的幅值也会增大； \mathbf{Q} 阵固定，增大 \mathbf{R} 阵中某一权值则控制量幅值相应减小，而动态性能指标的改善并不显著。

(2) 选择 \mathbf{Q} 阵和 \mathbf{R} 阵的一般原则 通常选用 \mathbf{Q} 和 \mathbf{R} 为对角线矩阵，实际应用中，通常将 \mathbf{R} 值固定，然后改变 \mathbf{Q} 的数值，最优控制的确定通常在经过仿真或实际比较后得到。当控制输入只有一个时， \mathbf{R} 成为一个标量数（一般可直接选 $\mathbf{R} = 1$ ）。 \mathbf{Q} 的选择不唯一。这表明当得到的控制器相同时，可以有多种 \mathbf{Q} 值的选择，其中

总有一个对角线形式的 $Q^{[51]}$ 。

(3) 全过程控制实验 首先, 使摆杆自然下垂并静止。第二步, 在上位机界面中点击“START”按钮, 开始起摆。第三步, 在上位机界面的文本框中键入小车期望位置, 点击“DOWNLOAD”按钮, 观察小车位置的变化。第四步, 给摆杆施加扰动观察系统的镇定过程。实验中系统参数如下:

$$a_+ = 0.2m/s^2, \quad a_- = -0.2m/s^2, \quad c = 0, \quad Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 600 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = 1,$$

$$K^* = [-31.63 \quad -24.05 \quad 57.92 \quad 10.92].$$

为确保起摆与稳摆过程的平稳过渡, 设置 $\phi_{TH} = 20^\circ$, 实验结果如图4-8所示。

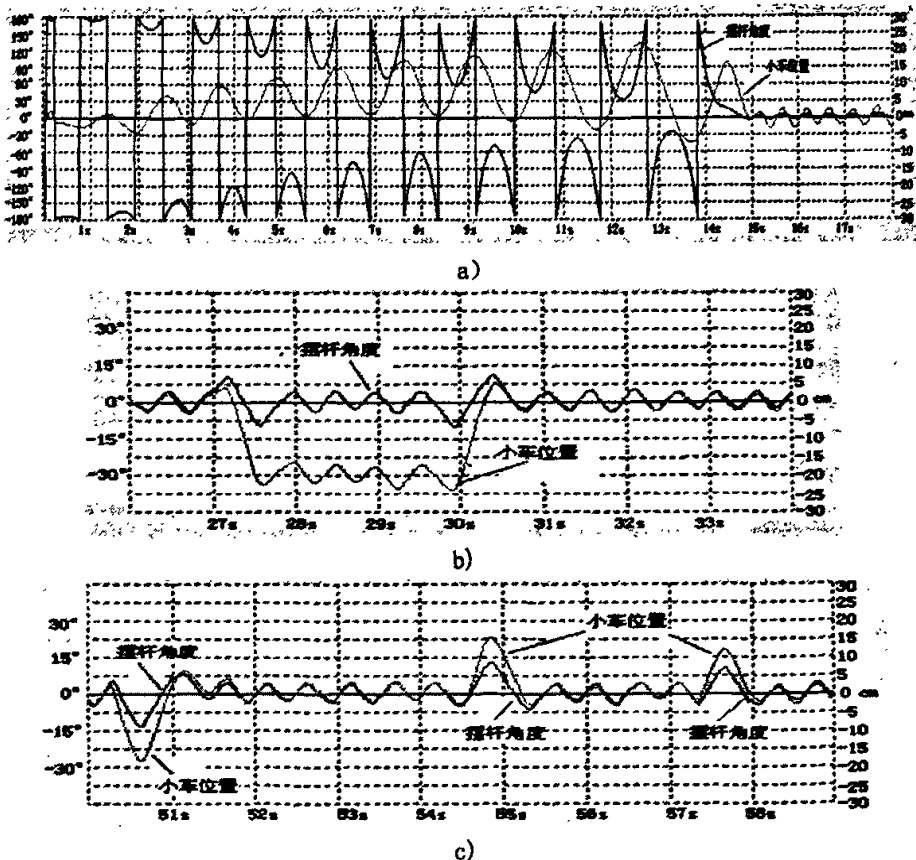


图 4-8 应用 LQR 方法的倒立摆控制实验结果

Fig4-8 Experiment result of Inverted Pendulum control using LQR

图4-8(a)所示为起摆控制向稳摆控制的过渡过程。摆杆经过多次振荡后, 在

14.5秒时起摆过程结束，进入稳摆阶段。此时，摆杆角度误差一直在 $\pm 8^\circ$ 以内，同时小车位置回归到原点附近，位置误差不大于 $\pm 0.05\text{m}$ 。

图4-8 (b) 所示为保持摆杆倒立的同时小车位置的随动过程。由图可见，当小车由原是位置项目标位置运动的过程中，摆杆的振幅仅有较小的增加，但不超过 $\pm 13^\circ$ 。

图4-8 (C) 所示为系统在外力扰动的响应。外力分别在第50s、55s、57s时刻作用在摆杆上，使摆杆角度产生增大的趋势。在这些外力的激励下，控制器驱动小车迅速做出反应，基本上将摆杆的最大振幅限制在了 $\pm 15^\circ$ 以内。

多次实验证明该系统在上述参数下起摆控制成功率在90%以上，小车位置误差在 $\pm 0.05\text{m}$ 以内，摆杆角度误差不大于 $\pm 8^\circ$ 。同时具有较高的抗干扰和实时性。

4.3 本章小结

倒立摆是一种典型的非线性自然不稳定系统，具有高阶次、多变量、强耦合等特点。倒立摆系统与足式机器人、双轮自平衡机器人在结构和运动特性上的相似性，使其能够比较真实地模拟机器人的运动平衡控制。因此，倒立摆系统的运动控制对于移动机器人控制技术的研究具有重要的参考价值。

本章以基于 DSP 和 RTOS 的机器人控制器为平台，设计了倒立摆运动控制应用软件。根据倒立摆起摆过程中的能量转换关系实现了倒立摆的起摆控制，分别应用经典控制理论中的 PID 算法和现代控制理论中的 LQR 算法实现了稳摆控制。并采用 RS-232 串行通信技术和 MFC 技术设计了友好的人机界面，实现了系统状态参数的可视化显示、数据存档和系统期望状态的手动设定。

第 5 章 移动机器人负趋光与避障控制

本章基于第二、三章所设计的控制器实现了移动机器人的负趋光和避障控制。首先,分析了《控制论》中关于维纳虫的思想实验。然后,设计了环境信息采集电路和用户任务,以一个三轮小车为载体实现了移动机器人的负趋光和避障控制。

5.1 控制论中关于“维纳虫”的思想实验

“维纳虫”是控制论的创始人维纳在其著作《控制论》中虚拟的一个具有一定仿生特性的人工装置,该装置具有类似某种低等生物的负趋光特性。所谓“负趋光性”就是指某些生命体所具有的避光倾向。

该人工装置验证了动物与机器在通信和控制方面的共通性。维纳认为,在生命体对环境进行感知并做出反应的过程中,起关键作用的是生物的神经反射弧;可以为机器设计一种机电系统以实现生命体中的神经反射弧的功能,他将这种机电系统称作机电反射弧。通过机电反射弧的工作,机器可以感知环境信息,产生类似生命体的神经反射的行为,从而具有了生命的特征^[52-54]。

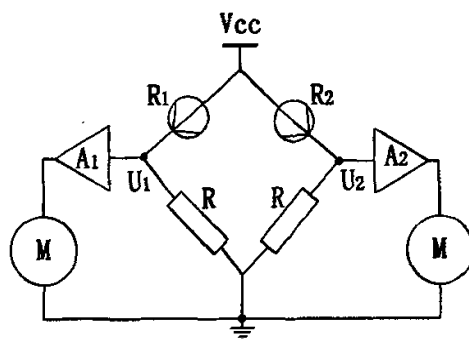


图 5-1 维纳设想的机电反射弧

fig 5-1 Electromechanical reflex arc described by Wiener

维纳设想,利用一个如图 5-1 所示的电路即可实现一种机电反射弧,从而使机器能表现出负趋光性这样一种生命特性。图中两个光电管 R_1 、 R_2 相当于生命体的感觉器官,惠斯登电桥相当于生命体的神经中枢,电机驱动电路相当于生命

体的传出神经，而电机就相当于生命体的动作器官。

在这种机电反射弧的作用下，“维纳虫”自动搜索环境中光照较弱的区域。当环境光强均匀时，由于 $U_1 = U_2$ 所以两个电机转速相同，维纳虫直行。当环境中光强不均匀时，若左侧光照较强，则 $U_1 > U_2$ ，左侧电机转速高于右侧电机，维纳虫转向右方，即光照较弱的一侧；反之亦然。维纳虫两侧光强差别越大，则转向速度越高。可见，维纳虫的负趋光性实质上是一种光线追踪。

该电路的优点是结构极其简单，但是致命的缺陷是系统性能严重依赖于器件参数的对称性。

因此，本文以第二、三章设计的移动机器人控制器重构维纳虫机电反射弧，实现其负趋光和避障控制。

5.2 维纳虫的信号采集电路设计

为实现维纳虫的负趋光和避障功能，系统应具备环境光检测电路和障碍物检测电路。

5.2.1 环境光检测电路设计

环境光检测电路包括信号拾取和信号处理两个模块，如图 5-2 所示。

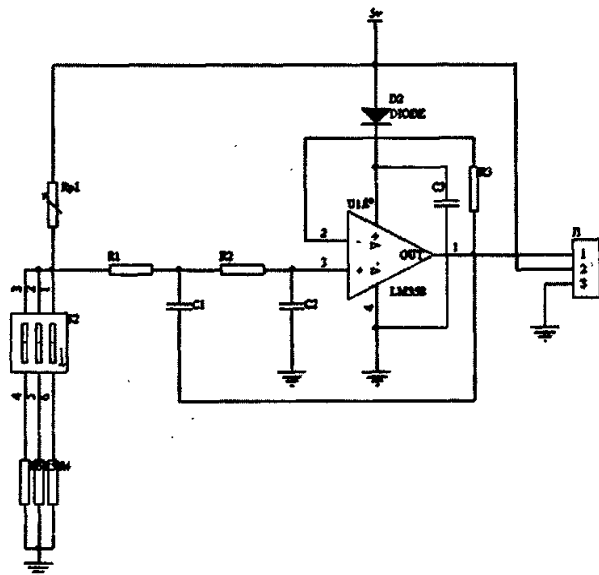


图 5-2 维纳虫环境光检测电路
fig 5-2 Worm Wiener's light-detect circuit

信号拾取模块由光敏电阻 R_{p1} 、三位拨码开关 $S2$ 和三个不同阻值的分压电阻 $R4$ 、 $R5$ 、 $R6$ 组成。 $S2$ 和 $R4$ 、 $R5$ 、 $R6$ 可以实现 8 种不同的等效分压电阻，以弥补光敏电阻的非线性特性，从而使系统能够适应不同的环境。

信号处理模块是一个由 $R1$ 、 $R2$ 、 $R3$ 、 $C1$ 、 $C2$ 、 $C3$ 、 $D2$ 和一片运算放大器 LM358 组成的放大倍数为 1 的二阶压控电压源型低通滤波器。该模块对信号拾取模块输出的信号进行滤波，因为该输出信号是非负的，所以放大器使用单电源供电，简化了电路结构。二极管 $D2$ 的作用是降低放大器电源电压，进而限制了本模块电路的输出信号的最大值不大于 3.3V。

该电路的工作原理比较简单。当环境光强变化时，光敏电阻 R_{p1} 的阻值随之变化，信号拾取模块的输出电压改变，所以滤波后的电压信号也随之变化。

5.2.2 障碍物检测传感器的选用

按照仿生学的观点，可以将机器人上各种传感器划分为感知型和非感知型两大类，如图 5-3 所示。

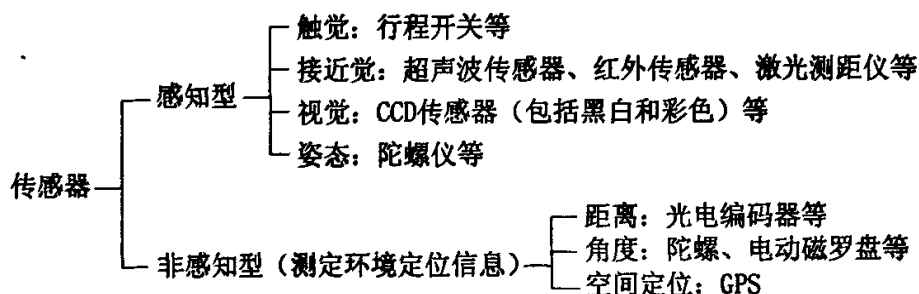


图 5-3 机器人传感器分类

fig 5-3 Catalogue of robot-related sensors

对环境中的障碍进行探测，通常采用感知型传感器。感知型传感器，可以通过自身的敏感元件获得环境辐射的信息（如声、光、电、磁）或向环境发射自身调制的辐射信号，并接受其返回信号以获取环境信息。移动机器人上常用的几种感知型传感器是超声波传感器、红外传感器、视觉传感器、激光测距仪器等。

对于维纳虫的避障而言，仅需获得其自身与障碍物的距离信息，因此，仅需配备一定数量的超声波传感器或者红外传感器即可满足要求。超声波传感器和红外传感器的工作原理相同，都是利用信号反射的渡越时间来计算障碍物距离。

超声波传感器是一种主动式传感器。其工作原理是，向环境中发射一定频率

(通常为 40kHz) 的机械波, 然后接收被障碍物反射回来的回波, 以回波的渡越时间乘以声速的 0.5 倍即可计算出障碍物的距离。由于声波在空气中不断扩散, 所以超声波传感器方向性较差。但是其优点是, 环境适应能力强, 在阴影、灰尘、烟雾等环境下, 应用不受限制。

红外传感器工作原理与超声波传感器类似, 采用发射固定波长红外线并接受同一回波的主动方式。其缺点主要是两方面, 一是受环境影响较大, 环境重的粉尘和荧光灯都会影响其工作的可靠性, 其二是探测距离短, 一般在 2 米以内。但其优点是探测视角小、方向性强、测量精度高、反应速度快^[55-59]。本文采用夏普 (SHARP) 公司生产的 GP2D12 型的红外测距传感器检测障碍物距离。该传感器实际上是一个变送器, 输出信号为模拟电压, 接口简单。

5.3 维纳虫的负趋光与避障控制

5.3.1 维纳虫的机电结构

本文采用一个 3 轮小车作为维纳虫的机械本体, 小车采用差速驱动方式。在车体前端安装两个环境光检测电路和红外测距传感器, 小车前视图如图 5-4 所示。

小车底盘为长方形, 在两侧分别有一个驱动轮 (3、4), 在前端安装一个从动轮 (5)。两个驱动轮分别有一个直流电机驱动, 直流电机的电枢电流由控制器 (9) 输出的 PWM 信号控制。

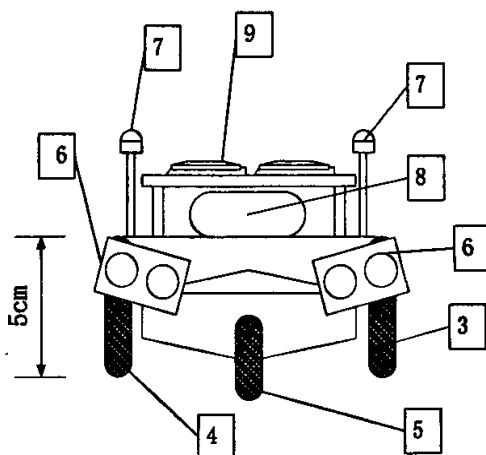


图 5-4 维纳虫小车前视图
fig 5-4 Front view of cart

在车体的前端安装两个夏普 (SHARP) GP2D12 型的红外测距传感器 (6), 用于检测前方的障碍物。该型传感器输出信号为 0~5V 模拟电压, 电压值与所测的距离成反比。红外测距传感器的输出信号送入控制器的 AD 电路接口。

两个环境光检测模块 (7) 安装于车体前端两侧, 并使光敏电阻的采光面向上, 以检测来自车体两侧上方的光照。环境光检测电路的输出信号送入控制器的 AD 电路接口。

系统由一个 8.4V 电池组 (8) 供电。控制器电路板通过定位孔安装在小车车体上方。

5.3.2 控制程序设计

基于实时内核的多任务系统的软件可划分为两个层次, 即系统层和应用层。系统层由内核和驱动程序库组成, 而应用层包括用于达成机器人任务目标的全部代码。系统层的设计已经在第三章完成, 本小节将完成实现维纳虫负趋光和避障控制的应用程序设计。

维纳虫的特性是以环境光为导航信号, 同时避开环境中的静态障碍物或者低速运动的障碍物。因此, 应用程序应包括 3 个任务, 即高优先级的避障任务 TASK_AROUND()、次优先级的导航任务 TASK_NAVIGAT() 和最低优先级的电机控制任务 TASK_MOTOR()。优先级的这种安排是因为维纳虫在遇到障碍物时, 应首先绕开障碍物, 然后再躲避光信号。另外, 考虑到电机采用 50Hz 的 PWM 信号控制, 因此反应时间较长, 所以为保证两个应用任务的决策得到有效执行, 任务的延时应该足够长。经过反复多次实验证明, 控制器在两次下达电机控制指令之间的时间间隔应不低于 200ms。

TASK_AROUND()、TASK_NAVIGAT() 和 TASK_MOTOR() 的程序结构和同步关系如图 5-5 所示, 其中共享数据 PARAMETER 包括小车的运动方式参数和一个标志该参数是否被锁定的互斥型信号量 FLAG。因为避障的优先级高于导航, 因此当前方出现障碍物时, 将由 TASK_AROUND() 控制小车的运动, 而 TASK_NAVIGAT() 的控制权被剥夺。只有当前方没有障碍物时, TASK_NAVIGAT() 才取得对小车的控制权。

TASK_AROUND() 通过红外测距传感器检测障碍物信息, 并依据反应式策略决定小车的运动方式。小车的运动方式包括左转、右转、倒车、直行。当前方无障碍物时, TASK_AROUND() 将信号量 FLAG 置 1, 表示可以由 TASK_NAVIGAT() 更改小

车的运动方式, 否则, FLAG 被清 0, 表示小车正在避障, 暂时放弃对光信号的追踪。

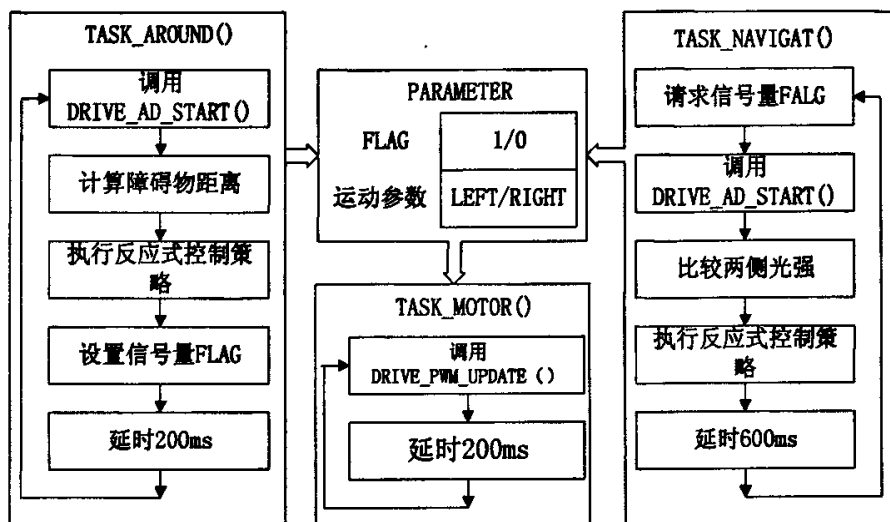


图 5-5 3 个用户任务的结构与同步关系

fig 5-5 Structures and synchronization of 3 application tasks

TASK_NAVIGAT() 首先查询信号量 FLAG 的状态, 如果 FLAG 为 0, 则挂起进入等待态。否则, TASK_NAVIGAT() 接管小车的控制权, 检测两侧的光强, 并依据反应式控制策略^{[60][61]}决定小车的运动方式。

TASK_AROUND() 或 TASK_NAVIGAT() 的决策结果保存在共享数据 PARAMETER 中。TASK_MOTOR() 根据这个共享数据设置 PWM 信号的占空比, 并调用 PWM 设备驱动函数 DRIVE_PWM_UPDATE() 改变相应电机的转速, 从而改变小车的位姿。

5.3.3 实验结果

维纳虫在小型足球机器人场地中漫游的场景如图 5-6 所示。

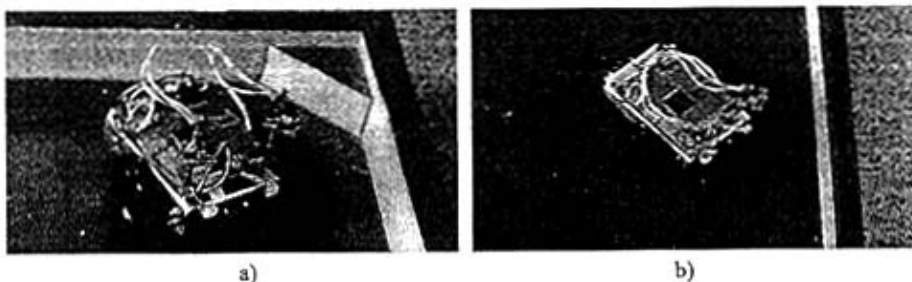


图 5-6 小车在机器人足球场地中漫游

fig 5-6 Cart wandering in the Robcup football ground

小车基本可以实现对光信号的反向运动, 并避开障碍物。图 5-6(a) 所示为小车在场地拐角处转向的场景, 图 5-6(b) 为小车在到达场点边缘时转向的场景。

实验中，小车与障碍物的碰撞较多，原因是小车的机械结构不合理，造成传感器对环境的覆盖率低、存在盲区，从而小车无法获得全面的障碍物信息。改进的方发是采用专用的机器人底盘，这种底盘的边缘应具有一定的弧度并有足够的传感器安装位置。

5.4 本章小结

本章首先分析了维纳在其《控制论》中所设想的机电反射弧的结构和工作原理。该电路的优点是结构非常简单，但是其对元件参数对称性的过分依赖却是致命性的。而且，该电路功能单一，不能实现避障这一移动机器人的基本技术。

然后，比较了常用的测距传感器并设计了用于环境光检测的专门电路。该电路通过光敏电阻检测环境光强，并通过可调的分压电阻在一定程度上弥补了光敏电阻的非线性特性。

最后，以本文第二、三章设计的移动机器人控制器为控制核心，以一个三轮小车为载体，应用上述环境光检测电路和红外测距传感器实现了维纳虫的负趋光和避障控制。

结 论

移动机器人控制技术在移动机器人学的一个重要分支,具有重要的研究意义和应用价值,主要研究移动机器人在复杂环境下的实时控制,包括移动机器人导航与定位、控制器体系结构和机器人本体控制(运动控制)等子课题。本论文在国家自然科学基金(60375017)、北京市人才强教计划的支持下,研究了基于TMS320F2812型数字信号处理器和嵌入式实时多任务操作系统 $\mu\text{C}/\text{OS} - \text{II}$ 的移动机器人控制器的设计与实现,在此基础上,以直线型倒立摆为对象模型研究了移动机器人的运动平衡控制、以三轮式小车为载体研究了移动机器人的负趋光控制和避障控制。

本文主要工作可以总结为如下三个方面:

第一,设计了一种基于数字信号处理器和嵌入式实时操作系统的移动机器人控制器。论文分析了TMS320F2812型数字信号处理器和嵌入式实时操作系统 $\mu\text{C}/\text{OS} - \text{II}$ 的体系结构。在此基础上,以DSP为核心设计了系统硬件,并在其上移植了 $\mu\text{C}/\text{OS} - \text{II}$,构造了一种嵌入式移动机器人控制决策系统。该系统既可独立执行具有一定复杂度的控制任务,又适用于主从式或者分布式控制结构,具有一定的开放性。

第二,基于本文所设计的控制系统实现了倒立摆的运动平衡控制。论文分析了单级直线倒立摆的机电结构和不同运动阶段的运动特性,然后,利用所设计的控制器应用经典控制理论和现代控制理论实现了倒立摆起摆和稳摆控制。此外,采用VC设计了人机界面软件,倒立摆系统的状态参数(角度、位置、速度等)可以通过RS232串口上传到PC/IPC进行图形化显示和数据存档。

第三,基于本文所设计的控制器实现了移动机器人避障与负趋光控制。论文分析了《控制论》中关于维纳虫的思想实验。然后,设计了环境信息采集电路,以一个三轮小车为载体实现了移动机器人的趋光和避障控制。

实验证明,采用主流高性能DSP和实时操作系统构建的移动机器人的控制器,具有实时性、快速性、开放性和板卡尺寸小型化等优点,具有很强的实用性。

移动机器人控制技术是一门交叉性学科,涉及电子、机械、计算机、控制论、

电机、传感器等理论与技术。本文所做工作仅是对移动机器人控制领域的初步探讨，仍有许多须改进和完善之处，主要包括如下两点：

第一，直线倒立摆虽然在机电结构和运动特性上与轮式自平衡机器人具有一定的相似性，但终究是一种简化的模型，因此，对机器人运动平衡控制技术的进一步研究应采用足式或轮式自平衡机器人实体作为控制对象。

第二，移动机器人的负趋光和避障试验中所用的三轮小车在结构、尺寸和传感器安装等方面不够理想，进一步的工作应专门设计合理的小车结构和传感器阵列以确保机器人尽可能全面地掌握环境信息。

参考文献

- 1 Nilsson. A mobile automation: An application of artificial intelligence techniques. In Proc IJCA
- 2 李磊, 叶涛, 谭民, 陈细军. 移动机器人技术研究现状与未来. 机器人. 2002 年, 24(5). 475-480
- 3 Hu H S, Brady M. A parallel processing architecture for sensor-based control of intelligent mobile robots. Robotics and Autonomous Systems, 1996, 1 (7):235-257P.
- 4 贺伟. 移动机器人体系结构的设计和研究. 哈尔滨工程大学硕士学位论文. 2006: 1-15
- 5 Jie Huang, Tadawate Supaongprapa. A model — based sound localization system and its application to robot navigation. Robotics and Autonomous Systems. 1999, 27:199 — 209P.
- 6 Robert L. French . Historical Overview of Automobile Navigation Technology. IEEE Technology Conference. 1986:350-358
- 7 Chong K S, Kleeman L. Sonar Based Map Building for a Mobile Robot. Proc of IEEE Intern. Conf. on Robotics and Automation, 1997, 2: 1700- 1705
- 8 E Bourque, G Dudek. Automated image-based mapping. In work shop on perception for creating virtual environments. In Proc IEEE Int Conf On Robotics and Automation, Leuven Belgium, 1998
- 9 Thrun S. Finding landmarks for mobile robot navigation. Proc of IEEE Inter Conf Robotics and Auto, 1998, 2:958-963
- 10 Yamauchi B. Inter. Conf. Mobile robot localization in dynamic environments using dead reckoning and evidence grids. Proc of IEEE on Robotics and Automation. 1996, 2:1401- 1406
- 11 Pere. Automatic Planning of Manipulator Movements. IEEE Trans on Sys Man and Cyb. 1981, 11(11):681-698
- 12 JCLatombe. Robot Motion Planning. Kluwer, Norwell. 1991:256-359
- 13 O. Khatib. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. Int J Robotics Research, 1986, 5 (1):90-98
- 14 袁泽睿. 两轮自平衡机器人控制算法的研究. 哈尔滨工业大学硕士学位论文. 2006: 1-62
- 15 段旭东, 魏衡华. 基于DSP 的二轮小车2倒立摆系统控制工程. 2004, 11 (6) : 521-524
- 16 Felix G, Aldo D A, Silvio C, et al . JOE: A mobile, inverted pendulum. IEEE

- Transaction on Industrial Electronics , 2002 ,49(1) :8-12
- 17 李洪兴等. 四级倒立摆的变论域自适应模糊控制. 中国科学, 2002, 32(1): 65~75.
- 18 李恺. 倒立摆伺服系统设计与实验研究. 清华大学硕士学位论文2004: 1-5
- 19 Felix Grasser,Aldo D'Arrigo,Silvio Colombi,Alfred C.Rufer. JOE: A Mobile Inverted Pendulum. IEEE Transactions on Industrial Electronics. 2002, 49(1)
- 20 屠运武, 徐俊艳. 自平衡控制系统的建模与仿真.系统仿真学报. 2004,16(4)
- 21 范永, 谭民. 机器人控制器的现状及展望. 机器人. 1999, 21 (1) : 75-80
- 22 William E. Ford. What is an open architecture robot controller[J].1994 international symposium on intelligent control, 1994, Columbus, Ohio,bS IEEE 16-18
- 23 Keum-Shik Hong, Kyung-Hyun Choi. A Pc-based open robot control system: PC-ORC. Robotics and computer integrated manufacturing, 17(2001):355-365
- 24 G. Prischow, Ch. Daniel, G. Junghans, et al. Open System Controller — a Challenge for the Future of the machine Tool Industry. Annals of the CIRP, 1993:44-945
- 25 周学才, 李卫平, 李强. 开放式机器人通用控制系统. 机器人.1998, 20(1) :25-31
- 26 J. Katupitiya, R. Radajewski, J.Sanderson, M. Tordon. Implementation of a PC Based Controller for a PUMA Robot[A].Proceeding of the 4th Annual Conference on Mechanatronics and Machine Vision Practice,1997
- 27 胡鹏. 开放式机器人控制系统研究与实现. 武汉理工大学硕士学位论文. 2006: 1-4
- 28 张卫宁.TMS320C28X 系列 DSP 的 CPU 与外设.清华大学出版社.2004: 1-400
- 29 苏奎峰 等.TMS320F2812 原理与开发.电子工业出版社.2005: 1-300
- 30 张培仁, 张志坚. 基于 16/32 位 DSP 机器人控制系统设计与实现.清华大学出版社, 2006: 1-380
- 31 黄立培. 电动机控制. 清华大学出版社. 2003: 42-43
- 32 刘胜, 彭侠夫. 现代伺服系统设计. 哈尔滨工程大学出版社, 2001: 239-269
- 33 杜坤梅 李铁才.电机控制技术.哈尔滨工业大学出版社.第 2 版.2002: 7-55
- 34 龚建伟, 熊光明. Visual C++/Turbo C 串口通信编程实践. 电子工业出版社.2005: 46-86
- 35 李现勇. Visual C++串口通信技术与工程实践. 第 2 版. 人民邮电出版社. 2004: 1-87
- 36 于英民, 于佳. 计算机接口技术. 第 3 版. 电子工业出版社, 2004: 154-200
- 37 邬可军 朱铭镭 等.DSP实时多任务操作系统设计与实现.电子工业出版社.2005: 1-325
- 38 任哲. 嵌入式实时操作系统 μ COS-II 原理与应用. 北京航空航天大学出版社, 2005: 1-200

- 39 Jean J. Labrosse. 嵌入式实时操作系统 μ C/OS2-II. 邵贝贝. 北京航空航天大学出版社, 2003: 1-560.
- 40 黄涛, 徐宏喆. 嵌入式实时操作系统移植技术的分析与应用. 计算机应用. 2003, 23(9): 88-98
- 41 赵伟国, 李文军. 实时嵌入式操作系统 μ COS-II 在 AT91 上的移植. 中国计量学院学报. 2005, 16(2): 137-139
- 42 杨麟祥, 岳继光. 实时内核 μ COS-II 在 TMS320F2407 上的移植. 计算机工程与应用. 2004(18) 129-130
- 43 尹勇, 欧光军等. DSP 集成开发环境 CCS 开发指南. 北京航空航天大学出版社. 2003: 8-145
- 44 张卫宁. TMS320C28X 系列 DSP 的指令和编程指南. 清华大学出版社. 2005:
- 45 张晓辉. 嵌入式操作系统驱动程序开发. 安徽电气工程职业技术学院学报. 2005, 10(1): 74-77
- 46 张霞, 张宁宁, 金元郁. 基于 DSP 实现的一阶倒立摆控制. 微计算机信息. 2006, 22(6-2)
- 47 关红洲. 倒立摆自动起摆和稳定控制系统及策略研究. 浙江大学硕士学位论. 2005: 1-21
- 48 梅晓榕, 柏桂珍. 自动控制元件与线路. 第 3 版. 科学出版社, 2005: 255-261
- 49 葛伟亮. 自动控制元件. 北京理工大学出版社, 2004: 326-330
- 50 Kiam Heong Ang, Gregory Chong, Yun Li. PID Control system Analysis, Design, and Technology. IEEE Transactions on control systems technology, 2005, 13(4)
- 51 朱文凯 袁桂嫦 朱学峰. 倒立摆的 PID 与 LQR 控制算法的对比研究. 广东自动化与信息工程. 2006(2):
- 52 阮晓钢. 机器生命的秘密. 北京邮电大学. 北京邮电大学出版社. 2005: 123-141
- 53 Norbert Wiener. *Cybernetics*. Cambridge, Mass: MIT Press, 1948
- 54 陈石. 基于行为仿真模型的机器生命实体研究. 北京工业大学硕士学位论文. 2006: 1-43
- 55 高富国, 谢少荣. 机器人传感器及其应用. 化学工业出版社. 2005: 4-25 84-85
- 56 Yoshinobu A, shi' ichi Y. Following a wall by an autonomous mobile robot with a sonar-ring.
- 57 孙运旺, 李林功. 传感器技术与应用. 浙江大学出版社, 2006: 117-123
- 58 王俊峰, 孟令启. 现代传感器应用技术. 机械工业出版社, 2006: 49-63
- 59 S Shafer, A Stentz, C Thorpe. An architecture for sensor fusion in a mobile robot. *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, Apr 1986, 3: 2002-2011
- 60 Robin R. Murphy. 人工智能机器人学导论. 杜军平, 吴立成. 电子工业出版社, 2004: 41-155
- 61 J Borenstein, Y Koren. Real-time obstacle avoidance for fast mobile robots in cluttered environments. *Robotics and Automation, 1990. Proceedings. 1990 IEEE International Conference on*, 13-18 May 1990, 10: 572-577

攻读硕士学位期间所发表的学术论文

- [1] 孙亮, 常青, 阮晓钢. 嵌入式移动机器人控制器设计. 计算机工程与应用.
2007.9
- [2] 孙亮, 常青, 阮晓钢, 王巍然. 基于 DSP 的倒立摆控制系统设计. 控制工程.
2008, 15 (2)
- [3] 左国玉, 常青, 孙亮, 阮晓钢. 基于 DSP 和 RTOS 的移动机器人控制软件
V1.0. (软件著作权) 2006SRBJ2556
- [4] 孙亮, 常青, 左国玉, 阮晓钢, 乔俊飞, 李明爱, 于建均, 于乃功, 李建更.
一种小型仿生移动机器人 (实用新型专利). 初审

致 谢

三年的研究生时光就要结束了。在即将告别工大校园之际，心中充满了对辛勤工作在这里的师长和同学们的感激，感谢他们三年来的指导和帮助。祝愿我的母校明天更加辉煌，祝愿我的老师和同学们工作顺利、捷报频传！

感谢我的导师阮晓钢教授三年来对我的悉心指导。阮老师严谨治学孜孜不倦的工作态度、科学高效的工作方法、敏锐的洞察力以及渊博而深广的学识给我留下了深刻的印象，为我树立了学习的榜样。

感谢人工智能与机器人研究所的孙亮副教授和左国玉老师。我在工作中所取得的每一个进展都深深地得益于孙老师和左老师的教诲和帮助。孙老师自 2005 年 5 月开始指导我的工作，至今已有两年。孙老师诲人不倦、风趣幽默，他的谆谆教导多次帮助我在困难面前重新鼓起勇气。左老师作风踏实、工作勤奋、乐于助人，对我的工作和学习给予了热情的关心和帮助。

感谢人工智能与机器人研究所的全体老师为实验室建设所做出的不懈努力。

感谢 1007、1006、1037 实验室的同学们和我的舍友，他们所营造的严肃、活泼、奋进、求实的氛围使我受益匪浅。

感谢王巍然同学在我的实验过程中提供的无私帮助。

最后，感谢我的家人和朋友，他们一如既往的关怀与支持是我不断前进的动力之源。