

分类号: _____

密 级: _____

学 号: 2006077

单位代码: 10407

江西理工大学

硕 士 学 位 论 文

论文题目: 基于改进蚁群算法的迷宫机器人的研究

研 究 方 向 智能仪器与自动化装置

专 业 名 称 控制理论与控制工程

研 究 生 姓 名 吴大舜

导师姓名、职称 王祖麟 教授

2008 年 12 月 12 日

江西·赣州

摘 要

人类科技的不断进步使机器人技术逐渐向智能化发展，智能机器人应运而生。移动机器人路径规划是机器人技术中一个重要研究领域，得到了很多研究者的关注，并取得了一系列重要成果。智能机器人可以通过传感器来识别周围的环境和自身的状态，并能进行分析判断，采取相应的行动。

路径规划问题是智能机器人的关键技术之一，自主移动机器人如何在未知的、复杂的环境中自主规划从起点到终点的路径，并且躲避障碍是智能机器人的最基本、最重要的能力之一，是其它应用的基础。迷宫机器人的路径规划是智能移动机器人路径规划的典型应用，由于迷宫环境的特殊性，迷宫机器人的路径规划有着比一般避障路径规划算法更加复杂的要求。

首先，本文实现了以采用 PHILIPS 公司生产的 32 位机 ARM7LPC2138 为主芯片的迷宫机器人平台。将算法程序下载到机器人本体后，机器人根据算法程序和传感器的障碍物信息自主运行。

其次，在研究了国内外智能机器人路径规划技术的基础上，本文对人工势场法、栅格法、遗传算法的路径规划进行了概述。本文根据机器人路径规划算法的研究现状和向智能化、仿生化发展的趋势，研究了一种基于改进蚁群算法的机器人路径规划方法。由于蚁群算法易陷入停滞、死锁等，本文采用了三点改进策略：1. 算法在执行过程中，当蚂蚁陷入无路可选的境况时，采用回退策略，使得蚂蚁能及时脱离困境，保证算法的健壮性；2. 蚂蚁在搜索过程中，当寻优方向不同的两只蚂蚁相遇时，采取相遇策略，利用两只蚂蚁禁忌表中的信息，生成新路径。这样可充分发挥蚁群的群体协作性，提高生成路径的速度；3. 在生成一定数目的路径后，采用带交叉点的路径交叉策略，提高算法的收敛速度。

接着，本文对实际迷宫进行了大量的研究分析，提出了迷宫的数字化表示方法，为计算机的仿真提供了方便。文章分析了迷宫中存在的八种路况，提出了相应的解决方案。通过仿真实验证实了该改进蚁群算法具有实现简单、效率较高、健壮性强的特点。

最后，对全文工作进行了总结，并对机器人技术和路径规划算法的研究进行了展望。

关键词：迷宫机器人；电脑鼠；路径规划；蚁群算法

ABSTRACT

With the rapid development of the science and technology, robots gradually become more and more intellectualized and intelligent robot is created. The path planning for mobile robots plays an important role in the field of robotics. It is arising many scholars' interesting and it's already got a series of achievements in this field. An intelligent robot acquires information of itself and the environments through the sensors. It then analyzes the information and makes a judgment to take suitable actions.

Path planning is one of the key problems of intelligent robot study. A basic functionality of intelligent robot is its ability to approach the destination autonomously by avoiding the obstacles in its way in an unpredictable, complex environment. This functionality is the basis of its other functionality. The path planning on maze robot is the typically applications of the mobile robot. It is more complex because the special characters of the maze environment.

Firstly, this paper developed a maze robot platform for path planning, which base on the single chip machine ARM7LPC2138 from PHILIPS. On the basis of the obstacles information and the algorithm program downloaded from the computer, the robot can make the best decision for movement.

Secondly, the paper makes a summary of a study up on Artificial Potential Field, Genetic Arithmetic, and Grid Method on path planning of mobile robots. According to the present condition of the studying and the intellectual and bionics tendency of the field's development, this paper presents a new path planning approach based on the improved ant colony algorithm with three new strategies. Firstly, in the process of the algorithm the ants without the next node to chose, employ the retracing strategy to leave the difficult position. So the robust character of the new method is strengthened. Secondly, in the processing when the two ants with the different searching direction meet, it uses the meeting strategy to get the new path by using the records information in the forbidden tables of the two meeting ants. Then the new method could get the enough cooperation character of the ant colony and the speed of getting new path could be improved. Lastly, when getting the fixed number of the paths, it uses path-crossing strategy to improve the converge speed of the new method.

Then, I studied the maze and put forward a method of figure about real maze, analyze and resolve the problems by computer. I analyze eight kinds of maze status and the strategy to solve problems when robots in maze. The results of the experiments show that the new method of this paper has the characteristics of simple, high efficiency and robust.

Finally, the full text is summarized, robot technology and mobile robot path-planning problem are prospected.

Key Words: maze robot; micromouse; path planning; ant colony algorithm

目 录

摘 要.....	I
ABSTRACT.....	II
第一章 绪 论.....	1
1.1 机器人简介	1
1.1.1 机器人的由来和发展.....	1
1.1.2 机器人与机器人学定义.....	2
1.1.3 机器人的基本组成.....	2
1.2 课题研究的背景和意义	3
1.2.1 课题研究的背景.....	3
1.2.2 迷宫机器人研究的学术价值.....	4
1.2.3 迷宫机器人研究的意义.....	4
1.3 本课题的国内外研究现状	5
1.3.1 迷宫机器人的前世今生.....	5
1.3.2 蚁群算法的发展.....	6
1.4 课题研究的主要内容	6
1.4.1 课题研究的主要内容.....	6
1.4.2 课题研究将面临的主要难点.....	7
1.5 论文的章节安排	7
1.6 本章小结	8
第二章 迷宫机器人系统总体设计规划	9
2.1 系统总体设计流程框图	9
2.2 系统硬件机身结构进展流程	9
2.3 系统硬件电路进展流程	9
2.4 基本驱动软件进展流程	12
2.5 搜寻策略软件系统进展流程	13
2.6 本章小结	13
第三章 迷宫机器人硬件系统的具体实现	15

3.1 迷宫机器人的机身设计	15
3.2 电源模块设计	17
3.3 电机驱动模块设计	18
3.4 测速电路模块设计	19
3.5 迷宫挡板检测电路	21
3.6 微控制器模块设计	23
3.7 其它电路	25
3.7.1 界面 LED 电路	25
3.7.2 JTAG 接口电路	26
3.7.3 独立按键电路	26
3.7.4 复位电路	27
3.8 本章小结	27
第四章 迷宫机器人的路径规划	28
4.1 移动机器人路径规划概述	28
4.2 移动机器人的路径规划方法	28
4.2.1 移动机器人路径规划的经典方法	29
4.2.2 移动机器人路径规划的智能方法	31
4.3 基于蚁群算法的机器人路径规划	32
4.4 本章小结	33
第五章 迷宫机器人的控制算法设计	34
5.1 迷宫场地的设计与制作	34
5.1.1 迷宫机器人场地标准	34
5.1.2 迷宫场地的设计与加工	34
5.2 迷宫的表示	35
5.3 迷宫机器人常用搜寻法则和策略	36
5.3.1 迷宫搜寻法则	36
5.3.2 迷宫搜寻策略	36
5.4 求解迷宫问题的经典算法	37
5.4.1 迷宫的广度优先搜索法	37

5.4.2	迷宫的深度优先搜索法.....	37
5.5	蚁群算法的应用	38
5.5.1	蚁群算法的起源与研究进展.....	38
5.5.2	基本蚁群算法的原理和数学模型.....	39
5.5.3	基本蚁群算法的实现步骤及程序结构流程.....	41
5.5.4	基本蚁群算法的软件实现.....	42
5.5.5	参数 α 、 β 、 ρ 组合配置对蚁群算法性能的影响	43
5.5.6	改进蚁群算法解迷宫问题.....	45
5.6	本章小结	48
第六章	迷宫机器人的软件设计与调试	49
6.1	迷宫机器人的控制系统	49
6.1.1	CPU 引脚资源分配.....	49
6.1.2	初始化模块.....	50
6.1.3	运动模块.....	52
6.1.4	迷宫挡板检测.....	54
6.2	迷宫智能算法的实现	57
6.2.1	沿跑道中线前进的判断程序.....	57
6.2.2	有记忆功能的迷宫算法分析.....	58
6.2.3	优化蚁群算法求解迷宫最短路径.....	61
6.3	实验调试及分析	62
6.3.1	蚁群算法用于求解栅格迷宫	62
6.3.2	基于 Visual Basic 开发环境的优化蚁群算法的软件实现.....	63
6.3.3	优化蚁群算法用于解标准迷宫问题的实验结果.....	64
6.4	本章小结	65
第七章	总结和展望.....	66
7.1	课题总结	66
7.2	未来展望	67
参考文献		68

致 谢..... 71

附录 A 迷宫机器人实物图..... 72

附录 B 迷宫求解程序参考源代码 73

个人简历 在学期间发表的学术论文与研究成果 83

第一章 绪 论

1.1 机器人简介

1.1.1 机器人的由来和发展

长久以来，人类都渴望制造一种像人一样能够灵活自由运动的机器，以便将人从繁重的、枯燥的、危险的活动中解脱出来。

早在我国西周时期（公元前 1066—公元前 771 年），就流传有关巧匠偃师献给周穆王一个歌舞机器人的故事。我国东汉时期（25—220 年），张衡发明的指南车是世界上最早的机器人雏形^[1]。

第一次工业革命以来，随着各种自动机器、动力机械^[2]的问世，制造机器人开始由梦想转入现实，许多机械式控制的机器人应运而生。1768—1774 年间，瑞士钟表匠德洛斯父子，设计制造了三个像真人大小的写字偶人、绘图偶人和弹风琴偶人^[3]。1893 年，加拿大人摩尔设计制造了以蒸汽为动力能行走的机器偶人“安德罗丁”。这些事例标志着人类在机器人梦想这一漫长道路上前进了一大步。

然而，直至 19 世纪，在神话、传说、小说和科学幻想作品中并没有机器人（Robot）这个名词。1920 年，捷克剧作家卡雷尔·卡佩克（Karel Capek）发表了科幻剧本《罗萨姆的万能机器人》（Rossum's Universal Robots），首创了 Robot 这个科学术语。在剧本中，卡佩克把机器人描述成与人相似但能不知疲倦地工作的机器，最终机器人背叛它们的创造者而消灭了人类。1950 年，美国著名科幻小说家阿西莫夫在他的小说《我是机器人》中，提出了有名的“机器人三大守则^[4]”：

- (1) 机器人必须不危害人类，也不允许眼看人类将受害而袖手旁观；
- (2) 机器人必须绝对服从于人类，除非这种服从有害于人类；
- (3) 机器人必须保护自身不受伤害，除非为了人类或者是人类命令它做出牺牲。

这三条守则现在仍被机器人研究人员、研制厂家和用户共同遵守。

现代机器人的研究始于 20 世纪中期，其技术背景是计算机和自动化技术的发展，以及核能的开发利用。自 1946 年第一台数字电子计算机问世以来，计算机技术取得了惊人的进步，向着高速度、大容量、低价格的方向发展。大批量生产的迫切需求推动了自动化技术的进展，其结果之一便是 1952 年数控机床的诞生。与数控机床相关的控制、机械零件的研究又为机器人的开发奠定了基础。另一方面，核能实验室的恶劣环境要求用某些机械代替人处理放射性物质。在这一需求背景下，美国原子能委员会的阿尔贡研究所于 1947 年开发了遥控机械手，1948 年又开发了机械式的主从机械手。

1954 年美国戴沃尔最早提出了工业机器人的概念，并申请了专利。该专利的要点是借助伺服技术控制机器人关节，利用人手对机器人进行动作示教，机器人能实现动作的记录和再现。这就是所谓的示教再现机器人。现有的机器人差不多都采用这种控制方式。

最早的实用机型(示教再现)机器人产品是 1962 年美国 AMF 公司推出的“VERSTRAN”和 UNIMATION 公司推出的“UNIMATE”。这些工业机器人的控制方式与数控机床大致相似,但外形特征迥异,主要由类似人的手和臂组成。1965 年,MIT(美国麻省理工学院)的 Roberts 演示了第一个具有视觉传感器的、能识别与定位的、简单积木的机器人系统^[11]。

如今机器人发展的特点可概括为:从横向看,应用面越来越宽,由 95%的工业应用扩展到更多领域的非工业应用,像做手术、采摘水果、剪枝、巷道掘进、排雷,还有空间机器人、潜海机器人;从纵向看,机器人的种类会越来越多,像进入人体的微型机器人,以成为一个新方向,可以小到像一个米粒般大小。最重要的是机器人智能化^[10]得到加强,机器人会更加聪明。具有感知能力和思维能力的机器人是我们追求的目标。归纳起来,机器人的发展除了受到人们想象力的限制外,还受到计算机技术、传感技术、材料的限制。

反过来,机器人的发展促进相关技术的进步。从 1959 年第一台机器人诞生到 20 世纪 80 年代初,机器人技术经历了一个长期缓慢的发展过程。到了 90 年代,随着计算机技术、微电子技术、网络技术等的快速发展,机器人技术也得到了飞速发展。除了工业机器人水平不断提高之外,各种用于非制造业的先进机器人系统也有了长足的进展。

1.1.2 机器人与机器人学定义

机器人的定义处在不断发展变化之中。美国国家标准局于 1981 年下的定义是:一种机械装置,在对其编程之后,可以完成某些依自动控制指令进行的制造和搬运工作。日本的定义更强调机器人的智能和仿人,日本以自动性、智能性、个体性、半机械半人性、作业性、通用性、信息性、柔性、有限性、移动性等 10 个特性来描述机器人的形象。我国科学家对机器人(Robot)的定义是:“机器人是一种自动化的机器,所不同的是这种机器具备一些与人或生物相似的智能能力,如感知能力、规划能力、动作能力和协同能力,是一种具有高度灵活性的自动化机器”。

综上所述,我们认为机器人是一种能自动控制、可重复编程、多功能,可以代替人完成特定任务的一种自动化机电装置。

机器人技术的迅速发展,为机器人学^{[9][12]}的建立奠定了基础。机器人学是综合应用数学、力学、机械、电子、计算机、自动控制、人工智能等多学科知识,对机器人的体系结构、机构、控制、智能、传感、编程语言以及机器人应用等进行研究的一门综合性新兴交叉学科。

1.1.3 机器人的基本组成

机器人系统^[13]一般由下列四个互相作用的部分组成:执行机构、控制器、环境和任务,如图 1.1 所示。

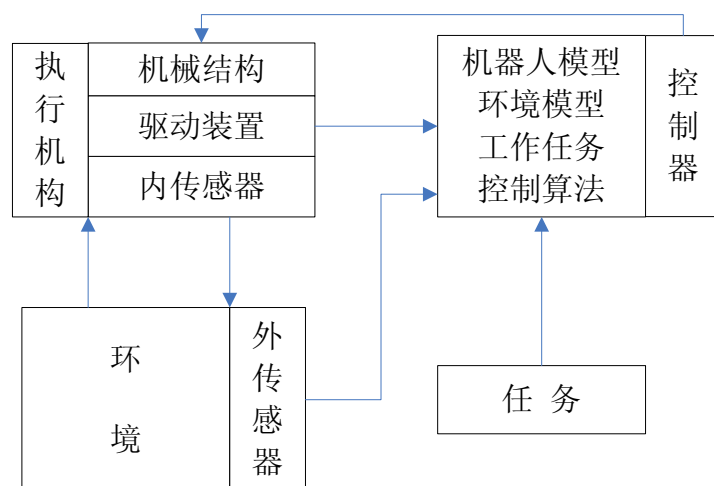


图 1.1 机器人系统的基本组成

执行机构一般是一种多关节式机械结构，由连杆、关节、末端执行器等组成。环境指机器人在执行任务时所能达到的几何空间，且包括该空间中每个事物的全部自然特性所决定的条件。

任务的定义为环境的初始状态和目标状态间的差别。这些任务必须用适当的程序设计语言来描述，并把它们存入机器人系统的控制计算机中。而且，这种描述必须能为计算机所理解。

控制器是机器人系统的指挥中枢，负责信息处理和与人交互，它接收来自传感器的信号，对其进行数据处理，并按照预存信息、机器人的状态及其环境情况等，产生出控制信号去驱动机器人的各个关节。为此，控制器内必须具有保证机器人实现其功能所必需的程序。

1.2 课题研究的背景和意义

1.2.1 课题研究的背景

机器人是一项非常适合科学普及和科技竞赛的主题。机器人赛事^[5]从 20 余年前就已经开始了。现在各类机器人赛事名目繁多，范围遍及世界。国际上比较有名的机器人竞赛有：RoboCup 竞赛^{[22][23]}、FIRACup 竞赛、寻线机器人竞赛、灭火机器人竞赛、舞蹈机器人竞赛、迷宫操作机器人竞赛、迷宫机器人（俗称电脑鼠）竞赛等。

目前，电脑鼠在一些电子设计竞赛中出现的比较多，且电脑鼠走迷宫是机器人比赛中比较重要的一个分支，其主要考察的是电脑鼠对未知世界的自动探索和学习过程。本课题是以“IEEE 标准电脑鼠走迷宫”竞赛为背景，结合模拟电路、数字电路知识以及传感器知识，将平时所学的各种控制理论运用到实践当中，制作出一个性能优良的电子产品——能够走迷宫的电脑鼠。

1.2.2 迷宫机器人研究的学术价值

迷宫机器人俗称电脑鼠(micro mouse)，它实际上是一个由微处理器控制的、集传感与控制于一体的、能够自动穿越迷宫的微型智能移动机器人。它对于学术交流，技术融合都很大的研究价值，主要有：

(1) 迷宫机器人技术的综合性。智能移动机器人集成了数学建模、机械传动、电机驱动、计算机软硬件、智能算法、通信技术等众多领域的科学与技术知识，可以毫不夸张地说，目前没有一种技术平台比智能移动机器人更综合。

(2) 迷宫机器人技术的先进性。智能移动机器人是当今前沿技术之一，集成了许多领域的先进技术。

(3) 迷宫机器人技术的自我更新性。智能移动机器人能自主吸收众多技术领域的新成果，具有良好的技术更新性，从而在本质上保证了智能机器人作为技术教育平台能持续走在时代前列。

(4) 迷宫机器人技术的自主体验性。智能移动机器人便于设计各种科技作品和家庭生活用品，在设计智能移动机器人科技作品的自主体验过程中，能够培养创新能力、综合能力、协作能力和进取精神等。

(5) 机器人技术的前沿带动作用。在机器人中运用的某些先进技术也可以应用于其它领域中，从而为总体科学技术的发展起到推动作用。

1.2.3 迷宫机器人研究的意义

机器人大赛是近年来国际上迅速开展起来的一种高科技对抗活动，它涉及人工智能、自动控制、机械电子、通信、传感机构等多个领域的前沿研究和技术融合。它集高科技、娱乐和比赛于一体，引起了公众的广泛关注和极大兴趣。

迷宫机器人比赛是比较有特色的机器人大赛之一。要求参赛的机器人具备自主寻找路径的功能，以在规定的 $3\text{m} \times 3\text{m}$ 迷宫内到达目的地时间最短者为胜。它提供了一种新开发的科研平台，其研究成果可推广应用于很多领域，并且应用范围越来越广泛。研究迷宫机器人的重大意义如下：

首先，它是素质培训的平台。目前国家所提倡的素质教育中，能力培养是核心。迷宫机器人研究为高校相关专业如计算机、自动控制、通讯、机械等专业的学生提供了能力培养的教学平台，通过它可以使学生将理论和实践紧密结合起来，提高学生的动手能力、创造能力、协作能力和综合能力，是培养高素质人才特别是信息产业人才的重要手段。

其次，它在科学研究方面的意义深远。迷宫机器人是一个极富挑战性的高技术密集型项目，其设计要求科研工作者对人工智能的前沿要有深刻的理解和敏锐的洞察力。在迷宫机器人系统的开发过程中，要融小车机械电子、机器人学、机电一体化、数据融合、精密仪器、实时数字信号处理、图像处理与图像识别、轨迹规划、避障算法等理论和技术于一体，是一个典型的智能机器人系统。迷宫机器人通过提供一个标准任务，使研究人员利用

各种技术来获得更好的解决方案，从而有效地促进各个领域的发展。其中的理论与技术可应用于工业生产、自动化流水线、救援、教育等实践领域，从而有效推动国家科技经济等各方面的发展。

1.3 本课题的国内外研究现状

1.3.1 迷宫机器人的前世今生

迷宫机器人俗称机器鼠、电脑鼠(micro mouse)，它实际上是一个由微处理器控制的、集传感与控制于一体的、能够自动穿越迷宫的微型智能移动机器人^{[25][26]}。

美国是机器人的诞生地，也是迷宫机器人的发源地。1977 年，美国举办了第一届真正的电子式老鼠的比赛，电脑鼠便从此流传至世界各地了！但电脑鼠却在美国销声匿迹了十一年，直至 1988 年美国才再度举办第二届电脑鼠大赛。

在 1977 年美国举办了第一届电脑鼠比赛后，电脑鼠便传至了欧洲。第一届欧洲电脑鼠比赛于 1980 年在英国伦敦举行，结果由主场作战的电脑鼠获得冠军。此后，英国的比赛便由英国电子工程协会（IEE）主办。

接着电脑鼠就传至亚洲来了，1980 年 11 月，日本电脑鼠协会（JMA）在东京举行了第一届全日本电脑鼠比赛，但不幸的是，参赛的十八只电脑鼠全军覆没，但 JMA 并不因此泄气，以后每年都举行电脑鼠比赛。

然后，电脑鼠终于受到国人的注意了！台湾地区在国内率先开展电脑鼠活动。有第一次电脑鼠活动是在一九八六年三月，当时是由第三波邀请日本的一些专家来台作表演赛，接着在同年十月举行了第一届电脑鼠大赛。从此，电脑鼠便在台湾定居，每年都有它的比赛。

我们在谈谈电脑鼠的一匹黑马——新加坡，新加坡的第一届电脑鼠比赛迟至一九八七年才举办，但此次比赛的冠军在隔年参加伦敦国际电脑鼠大赛便获得了第三名。第二届比赛的亚军参加 IEE 比赛，将蝉联两年的世界冠军 MITEE III（麻省理工学院制）给打了下来，勇夺世界冠军，之后，新加坡的电脑鼠声名大噪，在电脑鼠界中拥有一席之地。

我国大陆的电脑鼠起步相对较晚些。但近些年发展很迅猛。比如 2007 年，上海就举办了“2007 第三届上海市嵌入式系统创新设计应用竞赛”——暨 IEEE 标准电脑鼠走迷宫邀请赛（长三角地区）。本次竞赛除了继续进行前二届的创新设计应用外（本届将突出具有创新理念的设计应用），还引进了具有先进理念的“IEEE 标准电脑鼠走迷宫”竞赛。

此次“IEEE 标准电脑鼠走迷宫”竞赛在国内是首次举办。电脑鼠实际上是微型机器人，根据不同迷宫，能自行、智能、快速地走迷宫。该竞赛在国际上已开展了 20 多年，欧美有的高校还开设了相应的选修课。

我们要看到我国与国际水平相比还有很大的差距。比如，2006 年日本第 21 届 micromouse 迷宫机器人初级大赛中部地区预选赛上，由参赛者中村惠也制造的カムカム以

6 秒 541 的成绩夺得冠军。目前我国的电脑鼠还达不到这个成绩，所以我们要加倍努力，改进和提高我们的技术，争取早日赶超国际先进水平。

1.3.2 蚁群算法的发展

20 世纪 90 年代初，意大利学者 Dorigo M, Maniezzao V, Colorni A 等人^[15]从生物进化的机理中受到启发，通过模拟自然界蚂蚁寻食的行为，提出了一种全新的模拟进化算法：蚁群算法 ACA(Ant Colony Algorithm)。

到了 1996 年，Dorigo 等人在 IEEE Transactions on Systems, Man, and Cybernetics-Part B 上发表“Ant system: optimization by a colony of cooperating agents”一文。在这篇文章中，Dorigo 等不仅更加系统地阐述了蚁群算法的基本原理和数学模型还将其与遗传算法、禁忌搜索算法、模拟退火算法、爬山法等进行丁仿真实验比较并把单纯地解决对称 TSP 拓展到解决非对称 TSP、指派问题(quadratic assignment problem , QAP)以及车间作业调度问题(job-shop scheduling problem ,JSP)，且对蚁群算法，初始化参数对其性能的影响作了初步探讨。这是蚁群算法发展史上的又一篇奠基性文章。

自 1996 年之后的五年时间里蚁群算法逐渐引起了世界许多国家研究者的关注，其应用领域得到了迅速拓宽，这期间也有大量有价值的研究成果陆续发表。对蚁群算法不断高涨的研究热情导致了 1998 年 10 月 15 日至 10 月 16 日在比利时布鲁塞尔召开了第一届蚁群算法国际研讨会，会议由创始人 Dorigo 负责组织。第一届就吸引了来自世界各地的 50 多位蚁群算法研究者。随后每隔两年都要在布鲁塞尔召开一次蚁群算法国际研讨会历届会议的论文集均由著名的 Lecture Notes in Computer Science(SCI Index)结集出版。2000 年，Dorigo 和 Bonabeau 等人在国际顶级学术刊物 Nature 发表了蚁群算法的研究综述，从而把这一领域的研究推向了国际学术的最前沿。鉴于 Dorigo 在蚁群算法研究领域的杰出贡献，2003 年 11 月欧盟委员会特别授予他“居里夫人杰出成就奖(Marie Curie Excellence Award)”这一荣誉称号。

进入 21 世纪后，Nature 曾多次对蚁群算法的研究成果进行了报道，Future Generation Computer Systems 和 IEEE Transactions on Evolutionary Computation 都出版了蚁群算法的特刊。如今，在国内外许多学术期刊和会议上蚁群算法已经成为一个备受关注的研究热点和前沿性课题。

1.4 课题研究的主要内容

1.4.1 课题研究的主要内容

(1) 迷宫机器人电脑鼠的硬件结构设计。包括小车车体结构设计，小车控制模块，小车驱动模块，迷宫挡板检测模块，智能移动机器人的运动学原理等。

(2) 迷宫机器人电脑鼠的软件设计。包括初始化模块，运动及其控制模块，记忆模块

等。即要把当前热门的嵌入式系统应用进来。

(3) 迷宫机器人电脑鼠路径规划和避障算法的研究。包括算法分析比较, 蚁群算法解迷宫的具体实现, 路径规划的仿真等。

1.4.2 课题研究将面临的主要难点

现在, 迷宫机器人电脑鼠在硬件上已经没有太多的限制了, 限制其发展的是软件问题。首先是软件壁垒, 当今缺少一套可以适应不同环境和任务的控制算法是制约智能移动机器人发展的最大障碍。目前的算法都仅仅能够适用在被规划好的环境, 在没有规划好的环境中, 机器人几乎是寸步难行。比如: 代表当今移动机器人方面最高水准的美国 NASA 的火星 ROVER 也是由人类操控的。对于移动机器人来说, 能够移动是最基本的要求, 因此发展一套能够让机器人具备像人类一样移动能力的控制软件就变得至关重要。其次, 由于作者在算法分析和模糊控制方面还处于入门级, 所以要把智能算法如蚁群算法应用到迷宫机器人的路径规划当中有一定的难度。第三, 迷宫机器人电脑鼠的速度控制, 要实现电脑鼠准确而快速地做出动作以避免冲撞到障碍物。

还有, 如果在迷宫机器人电脑鼠上安装了摄像头, 那么视觉处理也是其中一个很关键的问题。如果安装了无线通信模块, 那么还要涉及到将小车行走时的数据譬如速度、行走距离、转弯角度等信息传送给 PC 机的通信问题。也许在实际的制作当中, 问题远多于此。那又有什么关系呢! 碰到问题再解决问题这不就是制作研究的乐趣所在吗? 何况, 我不是一个人孤军奋战, 我还有能指点迷津的导师, 还有能一起讨论问题的同学。

1.5 论文的章节安排

本论文的结构安排, 整体上可分为硬件设计和软件设计。

第一章描述了机器人的由来及其发展, 并对机器人的定义、组成等问题进行了叙述。

第二章给读者提供了一套制作迷宫机器人系统的完整设计方案。同时还给出系统总体设计、系统硬件机身结构设计、系统硬件电路设计、系统基本驱动软件设计、系统搜寻策略软件设计的进展流程图。本章指明了全文的工作方向。

第三章是本文的核心章节之一, 是迷宫电脑鼠的硬件设计部分。本章详细叙述了迷宫电脑鼠机身设计、电源模块设计、电机驱动模块设计、测速电路模块设计、迷宫挡板检测电路、微处理器模块设计这六大主要功能模块的具体实现方案, 有的设计甚至提供了多套方案。可以说本章的工作做好了, 我们的工程就成功了一半。

第四章首先介绍了移动机器人的路径规划及其方法。然后介绍基于蚁群算法的机器人路径规划, 为下一章全面介绍蚁群算法及其在迷宫最优路径问题的应用作了铺垫。

第五章是迷宫机器人控制算法设计部分, 是论文中最具鲜明特色的部分, 也是全文的核心章节。本章首先介绍了迷宫场地的设计和迷宫的表示, 把实际迷宫抽象化数学化成一个二维数组。然后又简单介绍迷宫机器人常用搜寻法则和策略。在求解迷宫问题的算法中,

详细阐述基本蚁群算法及其应用；并通过学习和研究，改进蚁群算法来求解迷宫最优路径问题。本章是全文工作目标能否实现的关键。

第六章是迷宫机器人的软件设计和程序实现部分，是本论文的重要组成章节。本章给出了迷宫机器人初始化模块程序、电机调速程序，迷宫挡板检测程序的具体流程框图。本章还给出了详细的迷宫智能算法实现方法和步骤。通过精选三个很有代表性的实验来验证优化蚁群算法求解迷宫最优路径的可行性与优越性。同时从实验过程中发现不足，并把这些不足之处作为将来我们要克服的目标。

第七章总结和展望。对论文进行总结，叙述了在论文中所做的各项工作以及本系统还存在的不足之处，并对本系统提出展望。

1.6 本章小结

在本章中，我先对机器人的由来和发展、机器人的定义、基本组成等问题进行了简要介绍。接着叙述了本课题研究的背景、目的和意义；本课题的国内外研究现状；课题研究的主要内容。最后，对论文的结构进行了安排。通过绪论，读者能对本论文的写作目的与内容有个简要的了解。

第二章 迷宫机器人系统总体规划

“罗马不是一天造成的”，要制作一个迷宫机器人（电脑鼠）也不是一朝一夕就能够完成的，它需要长年累月的积累才能做到。古人说的好：万事开头难。要如何开始，要从何开始，中间过程又如何等，都需要详细的计划，才不致于迷失了方向。在此，我提供了几个流程图，作为制作电脑鼠的参考。我相信大家在看过以下各个流程图之后，对于电脑鼠的制作流程应该有了方向，并且不再迷茫。在以下各节中，我将一步一步，循序渐进地制作一只电脑鼠。

2.1 系统总体设计流程框图

我们要制作一个迷宫机器人，没有一个系统完整的规划是不行的。可以说迷宫电脑鼠的制作是一项工程，所以我们要为这项工程先制定一项总体规划。有了系统总体规划，我们接下来的工作也就有明确的方向，这样有的放矢会大大提高我们的工作效率。系统总体设计流程框图如图 2.1 所示。

从图 2.1 中，读者很容易就能看出本系统主要分两大部分：硬件系统规划和软件系统规划。这两大部分又可以再细分为众多的小模块，如硬件系统可分为机身模块、微处理器模块、电机驱动模块、传感器检测模块等^[3]，软件系统可分为电机驱动模式、搜寻策略模式、传感器检测模式等。当然，我们即要先从大方面知晓要做什么，又要知道具体该干什么。把各个功能模块一件一件地完成，然后再把各个功能模块整合在一起进行调试修正。该删除的就删除，该改进的就改进。这样，一项看似复杂的工程就变得简单了。怎么样，大家看了我的分析后，是不是觉得制作一只灵巧可爱的电脑鼠并不是十分困难的事。下面我们就开始制作迷宫电脑鼠吧。

2.2 系统硬件机身结构进展流程

图 2.2 是迷宫机器人系统硬件机身结构设计的流程框图。本图主要涉及的是迷宫电脑鼠的机械结构设计。本设计方案把电脑鼠机身结构设计大致分成四个部分：底盘结构设计，基板结构设计，主辅轮设计和传感器安放位置设计^{[16][17]}。这四个分模块部分设计好之后，然后将各模块整合在一起进行调试修正。其具体设计的方法及步骤，我将在随后的章节中给与的说明。

2.3 系统硬件电路进展流程

迷宫机器人硬件电路设计规划如图 2.3 所示。本设计方案将电脑鼠硬件电路的设计分成六部分：电源电路、微处理器系统电路、界面电路、电机驱动电路、迷宫挡板检测电路和测速电路。

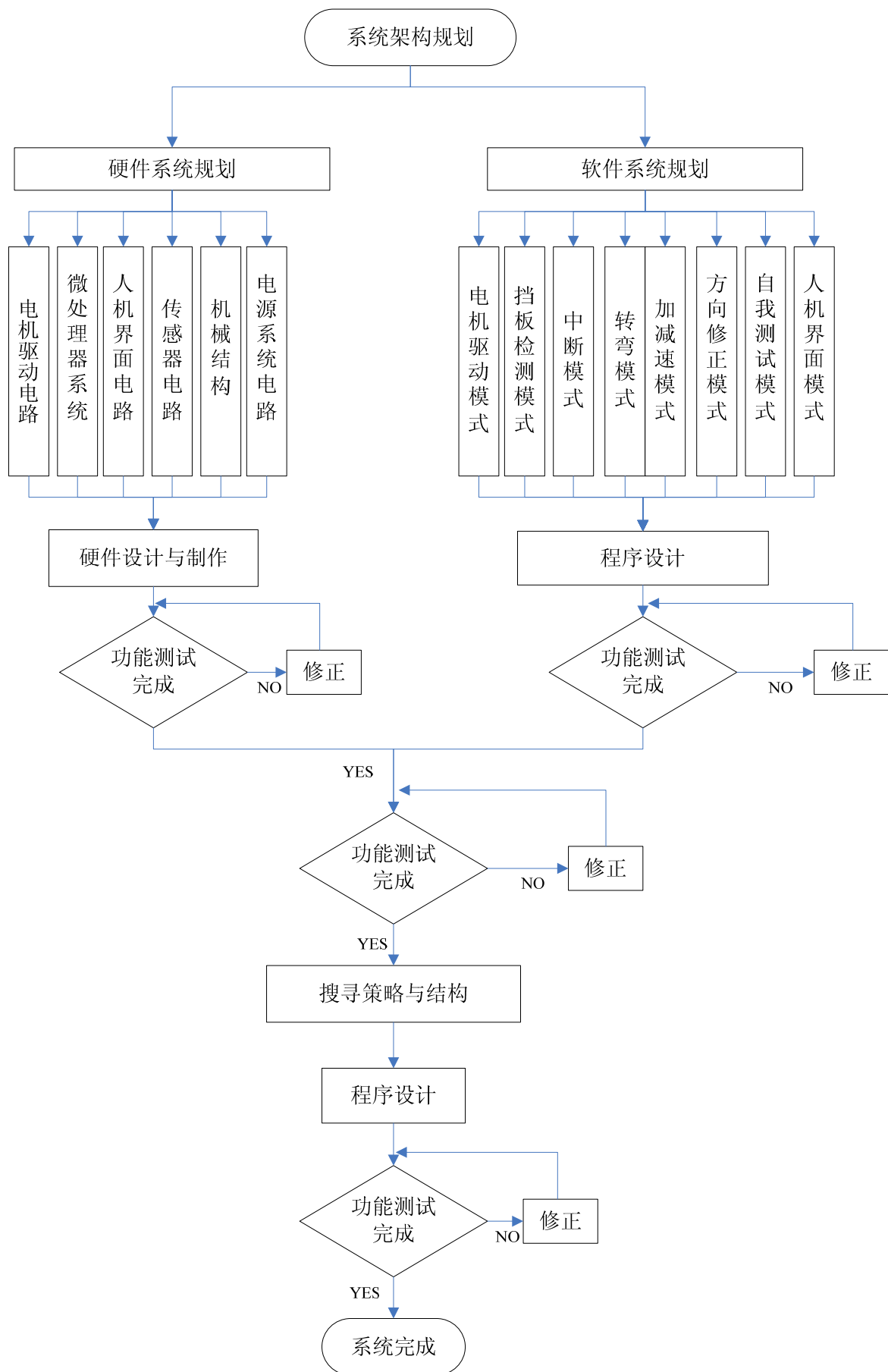


图 2.1 系统总体设计流程框图

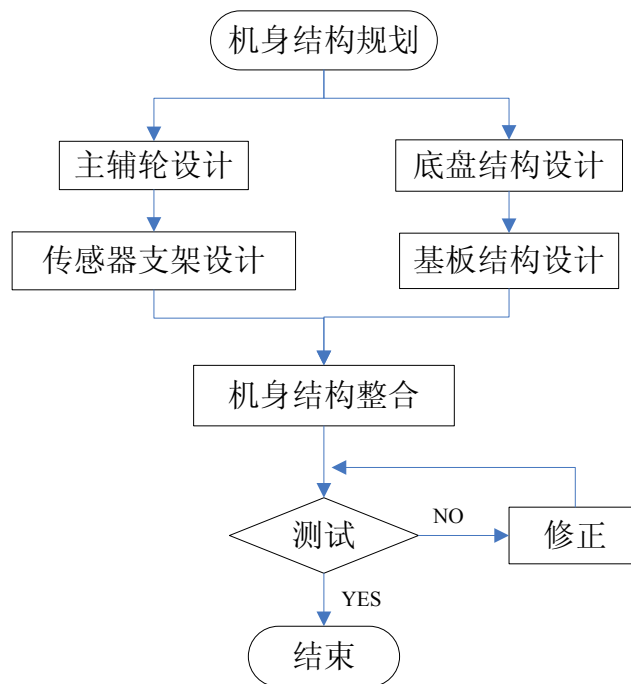


图 2.2 迷宫机器人机身结构设计流程图

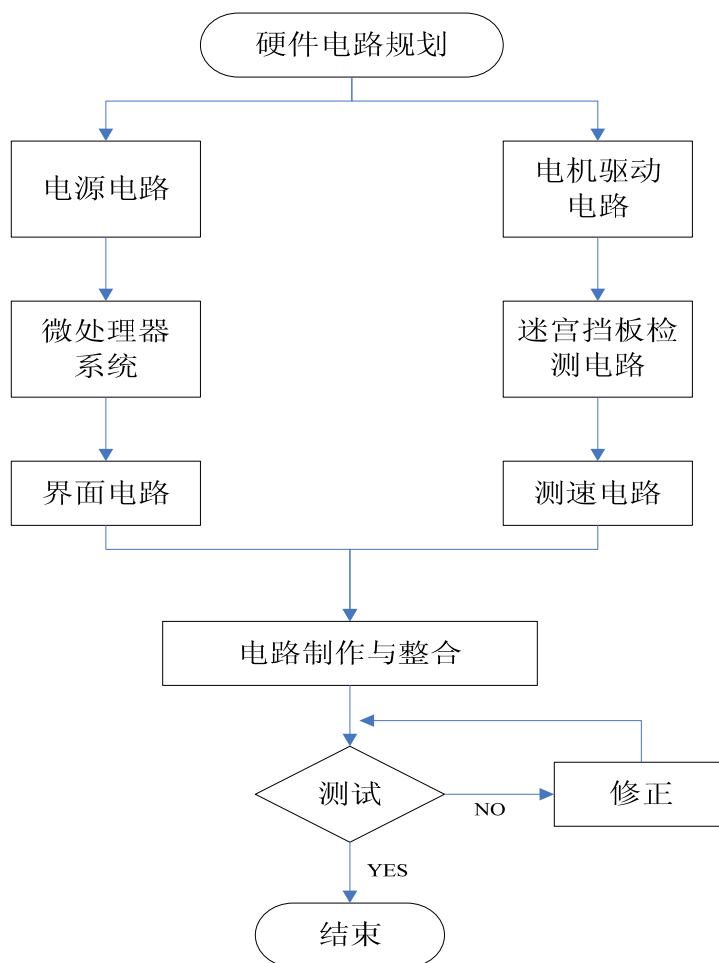


图 2.3 迷宫机器人硬件电路设计流程图

2.4 基本驱动软件进展流程

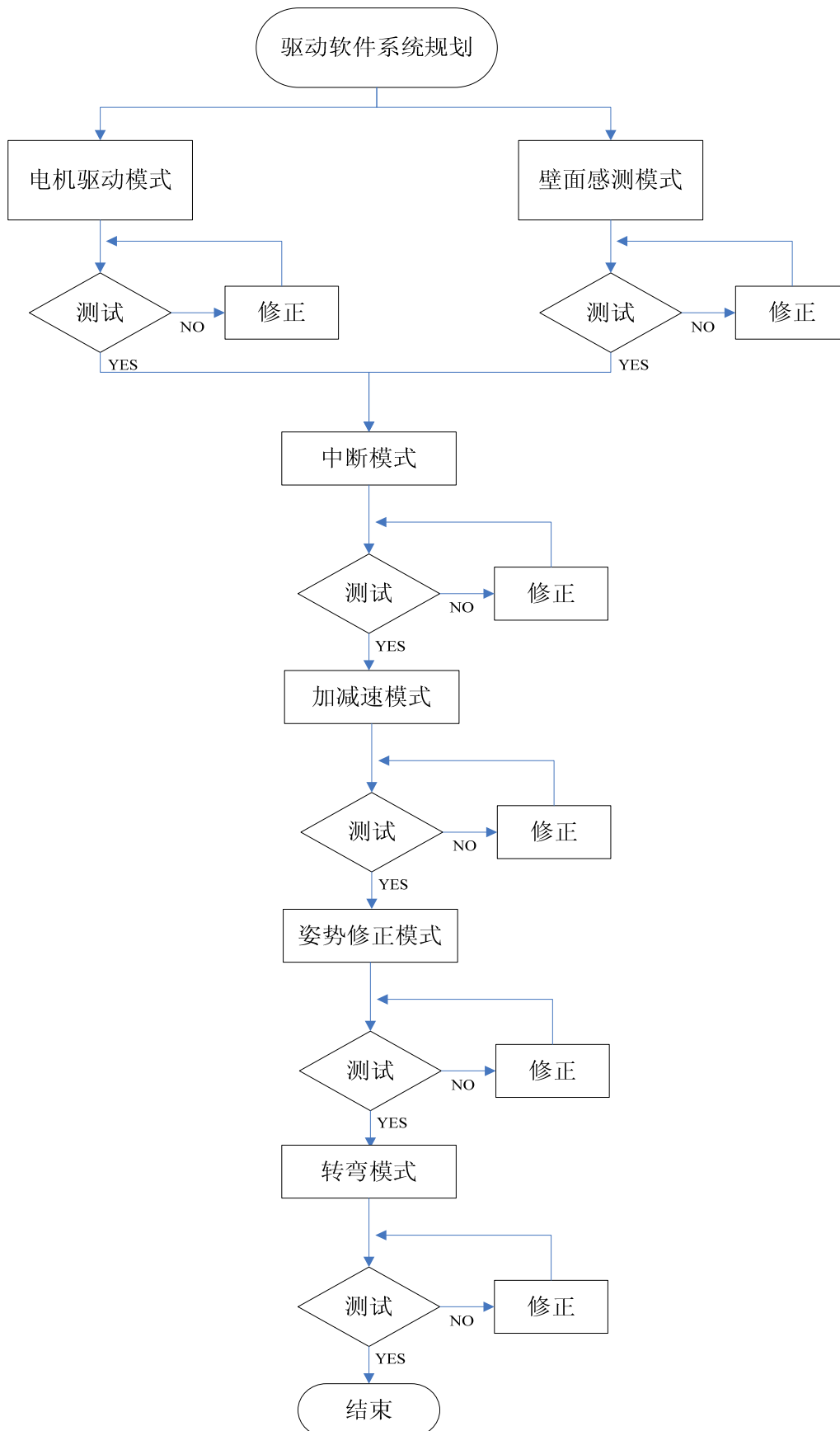


图 2.4 迷宫机器人基本驱动软件进展流程图

2.5 搜寻策略软件系统进展流程

迷宫机器人的搜寻策略软件^{[28][29]}系统规划如图 2.5 所示。此部分设计方案是电脑鼠总体设计中软件部分的核心内容，也是难点内容。在本部分中引入了很多算法，尤其是蚁群算法用于电脑鼠最短路径的计算，这是需要花费很多时间和精力去测试修正的。

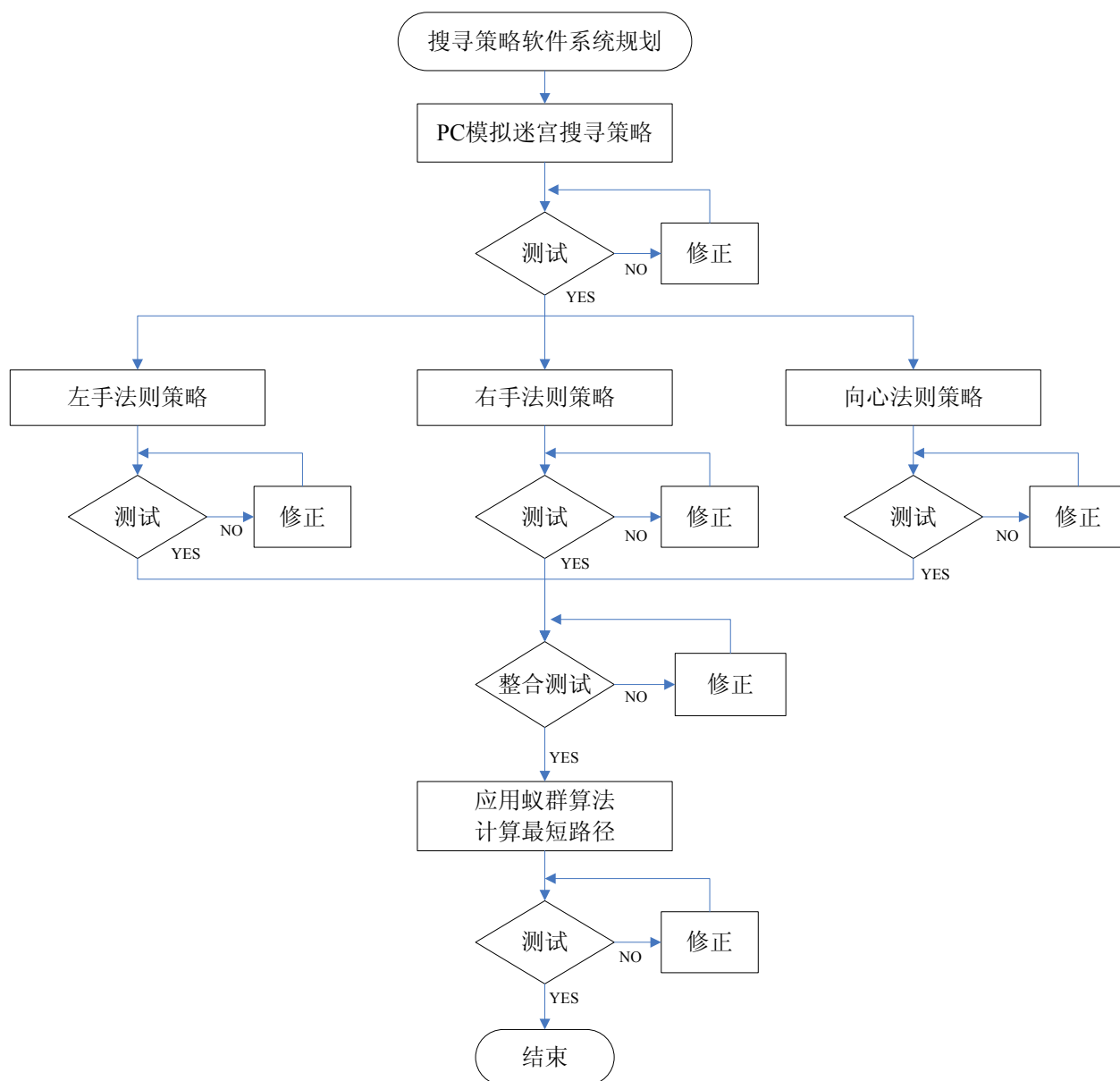


图 2.5 迷宫机器人搜寻策略软件系统流程图

2.6 本章小结

本章给读者提供了一套制作迷宫机器人系统的完整设计方案。方案将系统分为两大部分：硬件系统规划和软件系统规划。接着又将这两大部分再细分为众多的小模块，如硬件系统可分为机身模块、微处理器模块、电机驱动模块、传感器检测模块等，软件系统可分为电机驱动模式、搜寻策略模式、传感器检测模式等。这样做使机器人制作这项看似复杂

的工程变得简单起来。

本章还给出系统总体设计、系统硬件机身结构设计、系统硬件电路设计、系统基本驱动软件设计、系统搜寻策略软件设计的进展流程图。读者通过细读本章提供的设计流程图，对迷宫机器人的制作应该就能产生一套属于自己独有的总体设计规划。而总体设计规划的确会使我们的工作更有方向性、针对性，这样有的放矢就大大提高了我们的工作效率。

第三章 迷宫机器人硬件系统的具体实现

一只完整的迷宫机器人电脑鼠主要由机身、电源、传感器、微处理器、电机及人机界面这六大部分组成，如图 3.1 所示。各部分作用如下：①机身：主要用于支撑电脑鼠，是电脑鼠的骨架。②电源：给整体电路提供一个稳定的电压，是电脑鼠动力来源。③电机：是电脑鼠的驱动机构^[21]，相当于电脑鼠的脚，其动作受微处理器控制。④传感器：相当于电脑鼠的眼睛，用于感知四周的墙壁情况，以及检测电脑鼠是否走偏；⑤人机界面：启动电脑鼠，调节相关参数，并反映电脑鼠实时情况；⑥微处理器：是电脑鼠的大脑，它接受传感器传来的信息，然后经过判断、思考，再控制电机，同时能够记忆迷宫，并计算最短路径，使电脑鼠能在最短时间内到达终点。

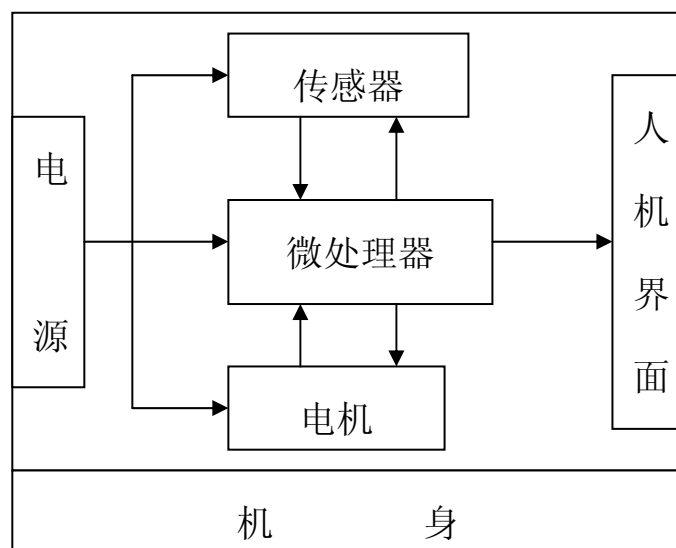


图 3.1 电脑鼠的基本结构

3.1 迷宫机器人的机身设计

机身是迷宫机器人（以下称电脑鼠）最基本的架构，它的设计在电脑鼠的整体规划中占有相当重要的地位。但是，它却常是非机械专业设计人员最感头疼的地方。

有关机身的设计，原则上整体结构要坚固耐用，材质要轻、重心要低。在设计中，即要考虑到符合各零件的规格，又要考虑到配合迷宫的规格。在比赛中，迷宫的每一个单元格都是 18cm*18cm 的正方形，每一个单元格的墙壁都有 1.2cm，也就是说电脑鼠的最大宽度应限定在 $18-1.2=16.8\text{cm}$ 之内。但因为老鼠在迷宫中行走时要有一定的空间裕度，否则稍微有一点不稳定就来不及调整而撞到墙壁上，再者为了便于转弯灵活，所以电脑鼠应该做的尽可能小。实际制作中，电脑鼠机身规格为：长 12cm，宽 9cm，高 5cm。

电脑鼠的底盘采用 3mm 厚的镁铝合金，这样既能满足结构强度和刚度的要求，又节省重量。电脑鼠共有三对调制反射式红外线传感器，如果在发射和接收正对方向的一定距离

范围内存在障碍物，则有相应的 LED 点亮，一共有五个 LED，其在 PCB 上的标号从左至右为 D7，D5，D4，D2，D1。

在测试程序里各个指示灯的具体含义如下：

- D7：点亮表示左边存在挡板，熄灭表示左边没有检测到挡板；
- D5：点亮表示电脑鼠太靠近左边挡板，熄灭则表示左边不会碰到挡板；
- D4：点亮表示前方存在挡板，熄灭表示前方不存在挡板；
- D2：点亮表示电脑鼠太靠近右边挡板，熄灭则表示右边不会碰到挡板；
- D1：点亮表示右边存在挡板，熄灭表示右边没有检测到挡板。

如果要改变传感器的探测距离范围，则需要调节感测电路的可调电阻。他们可分别调节左，前，右的测试距离范围。

另外，车轮在整个机身的设计中也占有相当的地位。一般来说，车轮分主动轮和从动轮，主动轮即电机所带动的轮子。设计中有左右两个主动轮，它们之间的距离必须在 16.8cm 以下电脑鼠才能在迷宫中行走。在本设计中，两轮之间的距离设定为 12cm，安置在机身两侧的中心线上。兼于机身的高度，为了降低重心，保证电脑鼠行使的更稳，我们设计的轮子半径大小约为 24mm。从动轮的选择也要注意配合，首先，从动轮到两主动轮的距离要略小于或等于两主动轮间距离的 1/2。其次，从动轮个数的选择要恰当，若用两个从动轮，设计精度比较高时则行走非常稳定，尤其是在转弯的时候比单个从动轮要稳定的多。但双从动轮对设计精度的要求较高，而且对不平路面的适应性较差，可能出现轮子悬起空转的现象，给控制带来不便。单个从动轮就不存在这个问题，它的灵活性更好，对路面的适应性也较佳。综合考虑，在选择方案时，选用双主动轮，单从动轮的方案，从动轮使用万向轮，转弯时是通过两个主动轮的差速来达到转弯效果，通过读取传感器信息及时调整消除单从动轮转动时造成的误差。电脑鼠单双辅轮设计的优劣对比如表 3.1 所示。

表 3.1 电脑鼠轮子设计的优劣对比

	主轮	单辅轮	双辅轮
设计时注意事项	两主轮间距离要小于 16.8cm(最好在 10~12cm 之间，以免微小的偏移就撞墙)。	从动轮到两主动轮的距离要略小于或等于两主动轮间距离的 1/2。	除了从动轮的距离要考虑之外，轮子的高度也要注意。
优点		设计较容易；行走时主轮较不会空转。	转弯时较稳定；若设计良好，则行走十分稳定。
缺点		转弯时误差较大。	设计较困难；若设计不良，行走时容易造成主轮空转。

电脑鼠机身设计架构如图 3.2 所示。图形仅供参考，它反映了底盘，车轮，LED，传感器等结构的相对位置。

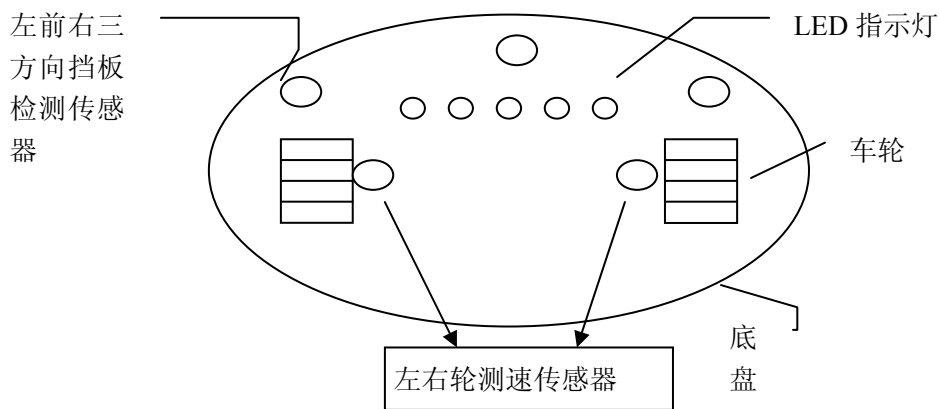


图 3.2 电脑鼠机身架构

3.2 电源模块设计

电脑鼠的电源电路^{[6][7]}如图 3.3 所示。它的结构比较简单，采用 4 节 1.5V 的 5 号电池供电，主要是经过两次降压，分别给 ARM 芯片^[8]的外围电路和 ARM 芯片供电。其中 LM7805 是一个电压调整器，它共有三只引脚，分别是输入端，接地端和输出端。LM7805 的作用是做电压的稳定，只要输入的电压在 6V 以上，负载的变化在允许的范围内，它的输出都能够维持稳定的 5V。另外，由于通过 LM7805 的电流很大，所以最好在其上加装散热片。SPX1117M3-3.3 芯片将 5V 的电源降压成 3.3V 给微处理器及用户外设供电。

在该电路中，5V 电源电路加了工作指示灯，通过发光二极管 D3 指示电源工作状态，R1 为 D3 的限流电阻。发光二极管正常工作电压为 1.5V~1.7V，正常工作电流为 10mA，所以流过 R1 的电流应为 10mA 左右。R1 电阻值的计算公式：

$$R_1 = \frac{5 - 1.5}{10 \times 10^{-3}} = 350\Omega$$

我们实际选用的限流电阻 R1 为 510Ω，其阻值略大于计算值。

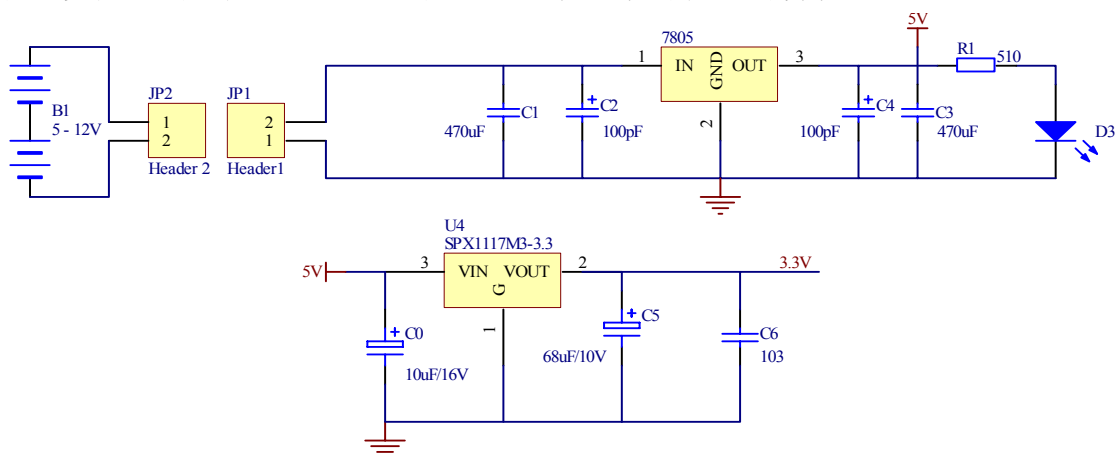


图 3.3 电源电路

3.3 电机驱动模块设计

方案一：采用双 H 桥集成功放电路芯片 L298N 驱动电机。

电机的驱动由双 H 桥集成功放电路芯片 L298N^{[16][17]}来实现的。L298N 是双通道直流电机驱动芯片，它可用来驱动两个直流电机或双极性步进电机。在 6~46V 电压下可提供 2A 的额定电流，过热后会自动关断，并且有电流反馈功能。可以激活这一功能，但需要安装片外续流二极管。电路图如图 3.4 所示：

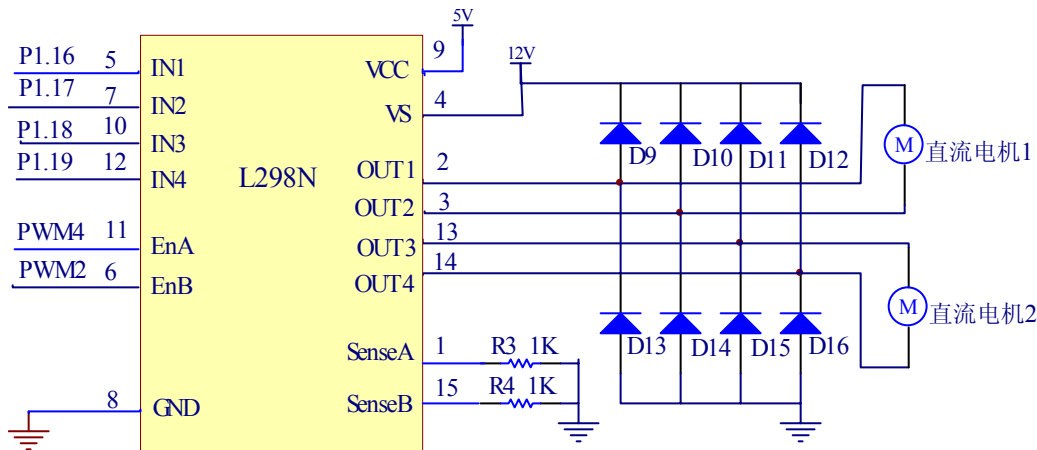


图 3.4 采用 L298N 的电机驱动电路

微控制器产生的 PWM 信号连接到 L298N 芯片的使能端 ENA 和 ENB。从 GPIO 引脚输出的信号送入 L298N 芯片的方向控制端 IN1~IN4，控制直流电动机的转动方向和制动状态。为保证 L298N 驱动芯片正常工作，在其与直流电动机之间加入四对续流二极管用以将电机中反向电动势产生的电流分流到地或电源正极，以免反向电动势对 L298N 产生损害。

L298N 可以控制两个直流电动机，使用两路 PWM 信号对其施加调速控制。OUT1、OUT2 和 OUT3、OUT4 之间分别接 2 个电动机。IN1 与 IN2、IN3 与 IN4 脚接输入控制电平，控制电机的正反转，ENA、ENB 接控制使能端，控制电机的停转。表 3.2 是其使能引脚、输入引脚和输出引脚之间的逻辑关系表。

表 3.2 L298N 的逻辑功能表

EN A(B)	IN1 (IN3)	IN2 (IN4)	电机运行情况
H	H	L	正转
H	L	H	反转
H	同 IN2(IN4)	同 IN1(IN3)	快速停止
L	X	X	停止

方案二：采用单相直流电机桥式驱动芯片 TA7291S 驱动电机。

电机采用直流减速电机，最高输出转速为 1200r/min，工作电压为 DC5V。电机驱动芯片采用单相直流电机桥式驱动芯片 TA7291S，如图 3.5 所示。其中 TA7291S 是 TOSHIBA 公司生产的单相直流电机桥式驱动芯片^{[20][21]}，工作电压为 4~20V，最大输出电流 400mA。电机驱动由输入端 IN1 和 IN2 控制，控制方法如表 3.3 所示。

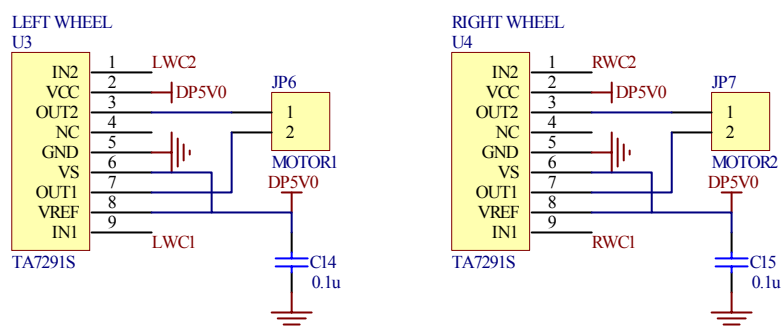


图 3.5 采用单 TA7291S 的电机驱动电路

表 3.3 电机控制

IN1	IN2	OUT1	OUT2	电机工作模式
0	0	∞	∞	停转，待机（STOP）
1	0	H	L	正转（CW）
0	1	L	H	反转（CCW）
1	1	L	L	刹车（BREAK）

我们要注意的是：电机的正转及反转与 OUT1 及 OUT2 和直流电机的接线有关，这里的电位关系是相对的。

对比以上两种方案，L298N 双 H 桥电路实现转速和方向控制，且电子开关的速度很快，稳定性也极强，是一种广泛采用的 PWM 调制技术。但 TA7291S 有电路简单，价格便宜，使用方便等优点。故从程序控制的简易性和成本等方面考虑，在本课题中选用 TA7291S 驱动直流电机。

3.4 测速电路模块设计

方案一：采用反射式光学编码器测速。

我们在电脑鼠主动轮的左右两车轮内侧贴上黑白条纹相间圆盘。它将红外线发射二极管（IR LED）发出的光线照射在码盘上。当码盘旋转时，白色条纹将 IR LED 发出的红外光反射到光电三极管，使其导通；而黑色条纹则将 IR LED 发出的红外光吸收，使光电三极管截止。如此便使光电三极管产生了相应的输出脉冲序列。条纹越多，单位时间产生的脉冲数就越多，码盘的分辨率就越高。如图 3.6 所示。

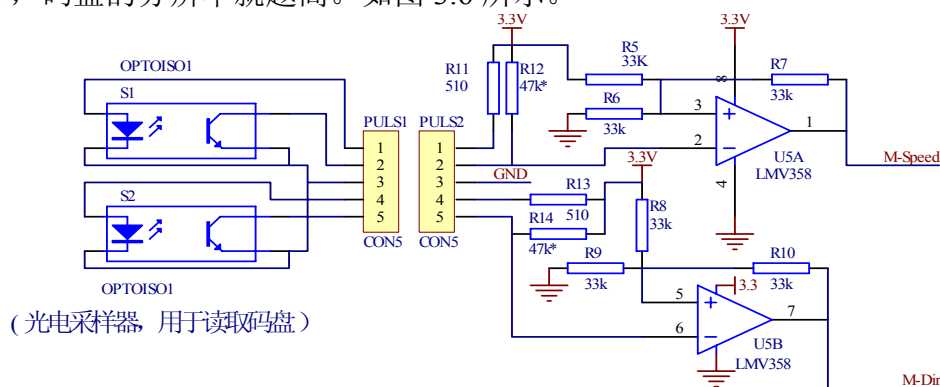


图 3.6 采用反射式光学编码器的测速电路

反射采样器是 Sharp 的 GP330，其特点是反射距离远，受自然光干涉少，作为轨迹采样十分适合，但作为码盘采样分辨率不太高。建议改用其它小体积、近距离的反射采样器，这样分辨率会高一些，可以多印一些黑白相间条纹，以提高转速测量的速度和精度。因为是反射式采样，输入信号不会很好，所以在硬件设计时就做了考虑，在输入电路中设计一个回差较大的比较器，以避免信号不好带来的误脉冲，而且回差可以通过 R7（R10）来调整，还可以通过改变 R12（R14）来改变输入信号幅度，通过改变 R5 和 R6（R8 和 R9）的比值来改变比较的阈值电平，这样设计的目的是：1)弥补反射式采样器安装带来的偏差；2)尽量使输出的脉冲信号接近 50% 的占空比，为倍速测量提供可能。

方案二：采用 ST168 的测速电路。

此方案中，测速模块主要是由 ST168 和轮胎的黑白间条组成。ST168 的外形及内部结构可以查阅相关产品说明。电路图如 3.7 所示。其中 LM324 的作用是一个有双阈值的特殊电压比较器。在信号处理中，它用于过滤掉噪声脉冲干扰信号^{[24][27]}。

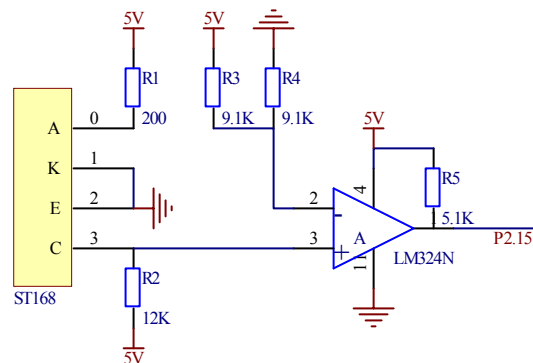


图 3.7 采用 ST168 的测速电路

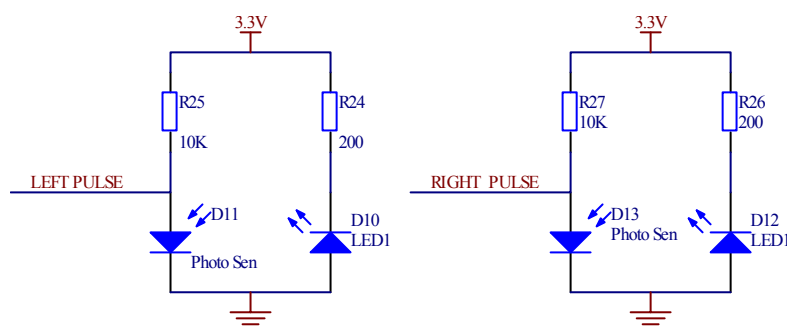
系统要有两个这样的模块，主动轮左右车轮上各有一个。每个模块只需一个 I/O 口来记录脉冲数目。

方案三：采用红外发射接收管的车速检测电路。

采用红外发射接收管的车速检测电路用于检测并记录车体运行的路径，通过车速检测记录车体在迷宫的坐标，同时也起到控制车速和保持左右双轮的速度一致。

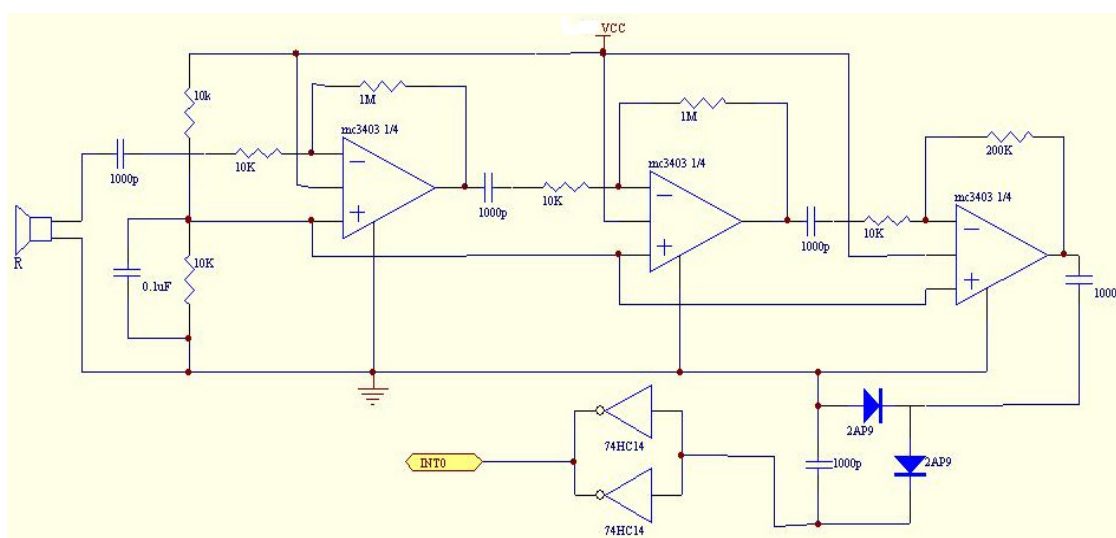
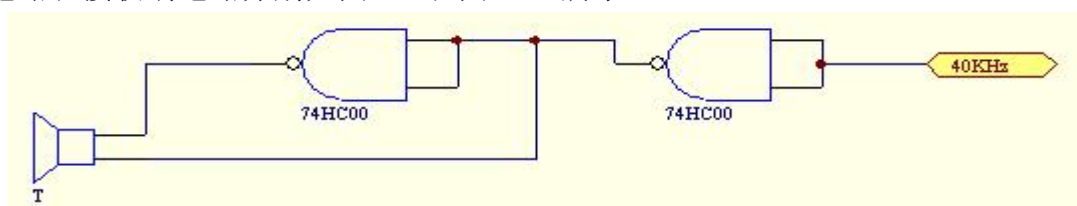
红外发射接收管的检测原理：在左轮和右轮的内侧都贴有光电码盘，码盘由黑白两种颜色组成。红外发射管安装在车轮光电检测码盘的检测区域，当红外发射与接收管正对着黑色边时，红外线没有被反射，接收管的电阻很大；当红外发射与接收管正对着白色边时，红外线被反射，接收管的电阻很小。检测电路如图 3.8 所示。

在图 3.8 的检测电路中，红外发射管与接收管正对着黑色边时，PULSE 输出高电平；正对着白色边时，PULSE 输出低电平；从黑色边到白色边时，PULSE 输出一个下降沿信号；从白色边到黑色边时，PULSE 则输出一个上升沿信号。微处理器可以检测输出脉冲的下降沿信号判断车轮转动的角速度，假如码盘上有 20 道黑白相间的条纹，那么当微处理器检测到 21 个下降沿信号时，说明车轮子转动了一圈。



以上三种方案都在一定程度上受外界环境影响。我们对方案一进行了改进,即使用透射式码盘测速脉冲采集模块来实现测速功能,提高测量的分辨率,并能够实现系统的闭环控制。

方案一：采用超声波模块的挡板检测电路



接收端的 I/O 口是具有外部中断功能的，当接收到信号，MCU 就会触发中断，这时记录时间间隔然后就可以计算出距离障碍物的距离了。我们可以在小车的前、左、右都安装了发射与接收探头，让小车全方位避障。

反射式红外传感器探测障碍物的原理见图 3.11 所示，发射管与接收管平行同向安装，接收管只能接收到发射回来的红外线信号。发射管向外发射红外线后，如果在有效距离范围内存在反射表面，则反射回来的信号将会被检测到。如果不存在反射表面或者反射表面距离车体太远，接收传感器将不能检测到信号。

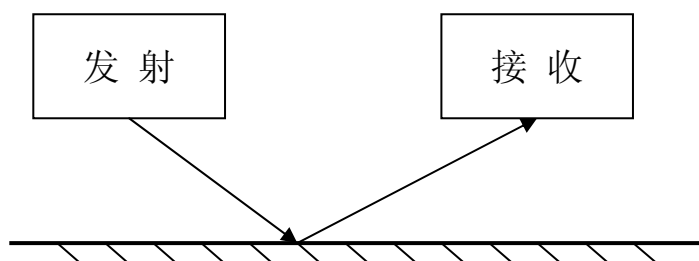


图 3.11 反射式红外线传感器探测原理

方案二：采用不调制的反射式红外发射—接收器。由直流电直接驱动红外管发光，这样电路简单，成本较低。但由于所有物体只要温度高于零摄氏度，就会向外发送红外线，且太阳光和日光灯中最强，所以该电路很容易受到外界干扰。其电路如图 3.12 所示

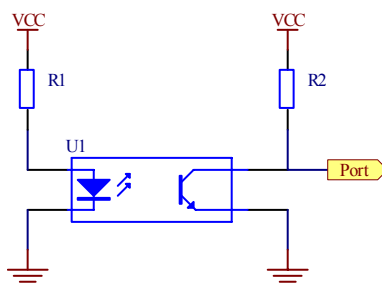


图 3.12 不调制的反射式红外发射—接收检测电路

方案三：采用脉冲调制的反射式红外发射—接收传感器。其电路图如图 3.13 所示。考虑到环境干扰主要是直流分量，如果采用带有交流分量的调制信号，在平均电流不变的情况下，瞬时电流可以很大，这样也大大提高了信噪比，可以有效避免外界环境变化对系统检测精度的影响。

采用脉冲调制的反射式红外发射—接收传感器的检测电路原理图如图 3.13 所示，由可调电阻 RP1，红外线发光管 D1 和三极管 Q1 构成的电路为红外线发射电路。RP1 可以调节红外线发光管的发光张度，Q1 起驱动及保护作用。R1 和 C1 组成的网络为发射端的保护电路。在接收电路中，U1 为体式红外线接收传感器 IRM8601S，它内部集成了自动增益控制电路、带通滤波电路、解码电路及输出驱动电路。当它连续收到 38KHz 的红外线信号时，将产生脉宽 10ms 左右的低电平。如果没有收到信号，便立即输出高电平。还有，由于它是开漏输出，所以输出端需接一个上拉电阻 R3。其中 R2 是限流电阻；C17 滤出电源高频干扰；C20 也是滤波电容；R28 为接收端的限流电阻，防止过大的 I/O 端口电压，起保护微控制器 I/O 口的作用。

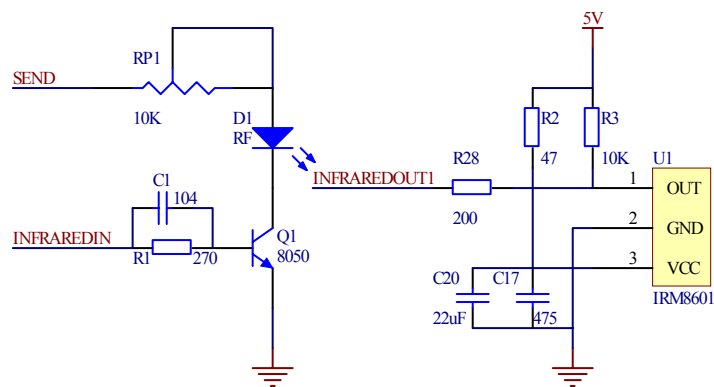


图 3.13 脉冲调制的反射式红外发射——接收检测电路

脉冲调制的反射式红外发射——接收检测电路用于迷宫挡板的检测，我们在电脑鼠车体左侧、右侧和前方三个方向上都装有此电路，三个方向的检测原理相同。在电路图中，INFRAREDIN 接受 38KHz 的红外线信号；SEND 为发射控制端，高电平时发射 38KHz 的红外信号；INFRAREDOUT1 为接收输出端，低电平表示收到信号。

以上三种方案，方案一存在以下几点问题：①存在测量“盲区”。②入射角对测距有较大的影响。超声波换能器发出的超声波波束不是一条直线，而是具有一定指向性的开角波束，所以超声波入射角对测量结果的影响很大，当入射波和障碍物之间的角度太大时就会产生幻影的可能。方案二很容易受到外界干扰。方案三在近距离能实现精确测量，且传感器只接收很小频率范围内的信号，不容易被外界干扰。故我们采用方案三。

3.6 微控制器模块设计

方案一：选用 51 系列单片机 STC12C5410AD。

STC12C5410AD 单片机是宏晶科技的产品。其资源有：10K FlashROM，512 字节 RAM，8 路 10 位 AD，4 个 PCA（可实现 PWM、脉冲捕捉等定时功能），速度也十分快。

我们对单片机 STC12C5410AD 的硬件资源分配如下：

- ◆ P2.4~P2.7：用于驱动 H 桥；
- ◆ PCA3：用于电机的 PWM 输出，工作在定时器模式；
- ◆ PCA2/P2.0：用于主测速脉冲输入，工作在捕获模式，边沿触发；
- ◆ P2.1：主脉冲输入方向辅助输入，用于判断转动的方向；
- ◆ PCA1/P3.5：用于跟踪脉冲输入，工作在捕获模式，边沿触发；
- ◆ P3.4：跟踪脉冲的辅助输入，用于判断方向；
- ◆ PCA0/P3.7：用于 PWM 控制信号输入，工作在捕获模式，边沿触发；
- ◆ P1.4~P1.7：用于 SPI 控制输入；
- ◆ P1.2、P1.3、P2.2、P2.3、INT0、INT1：用于连接无线模块 PTR4000，以便实现无线控制；
- ◆ P1.0、P1.1：保留作为 ISP 的控制口；

- ◆ P3.0~P3.1: 用于 UART 控制输入;
- ◆ T1: 用于产生 UART 的波特率;
- ◆ T0: 系统时基;

采用 51 系列单片机 STC12C5410AD、晶体振荡器和复位电路的原理如图 3.14 所示。

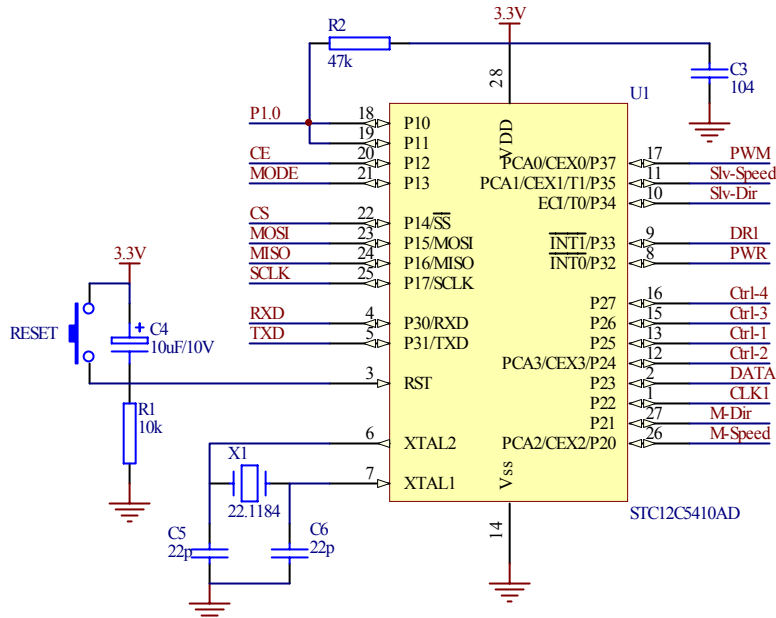


图 3.14 采用单片机 STC12C5410AD 的微处理器控制电路

方案二：选用 PHILIPS 公司的 LPC2138 ARM 微控制器。

选用 PHILIPS 公司基于 ARM7TDMI-S 核、单电源供电的 LPC2138 作为系统的微控制器。对于 LPC2138 芯片，最小系统需要两组电源、复位电路、晶振电路，P0.14 脚接一个上拉电阻禁止 ISP 功能。LPC2138 的 64 脚封装如图 3.15 所示。

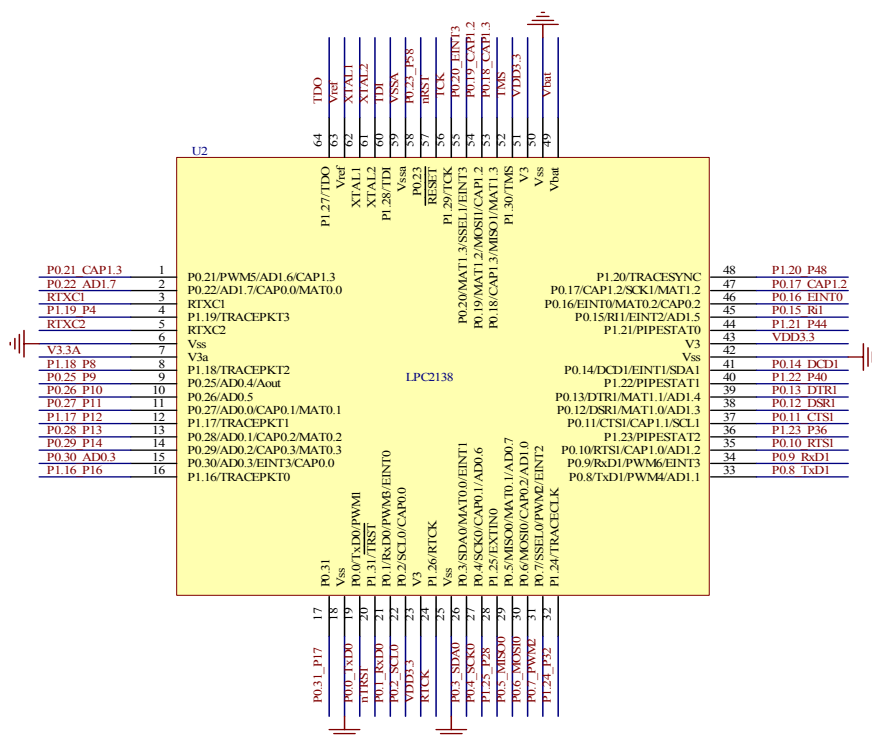


图 3.15 LPC2138 的引脚排列

ARM 内核采用精简指令集计算机 (RISC) 体系结构, 具有大量的通用寄存器, 指令格式使用统一和长度固定, 寻址方式简单, 内含 2 套指令系统 (ARM 指令集和 Thumb 指令集)。极低的功耗, 适合对功耗要求较高的应用, 如便携式产品。能够提供 0.9MIPS/MHz 的三级流水线结构。

LPC2138 一些主要特性:

- ①小型 LQFP64 封装的 16/32 位 ARM7TDMI-S 微控制器;
- ②32KB 片内静态 RAM; 512KB 片内 Flash 程序存储器;
- ③向量中断控制器。可配置优先级和向量地址;
- ④128 位宽度接口/加速器实现高达 60MHz 的操作频率; 片内 Boot 装载程序实现在系统编程 (ISP) 和在应用编程 (IAP)。Flash 编程时间: 1ms 可编程 256 字节, 扇区擦除或整片擦除只需 400ms; EmbeddedICE-RT 和嵌入式跟踪接口, 可实时调试和高速跟踪执行代码;
- ⑤2 个 8 路 10 位 A / D 转换器共包含 8 个模拟输入, 每个通道的转换时间短至 2.44μs; 2 个 32 位定时器 (带 4 路捕获和 4 路比较通道)、PWM 单元 (6 路输出) 和看门狗;
- ⑥多个串行接口, 包括 2 个 16C550 工业标准 UART、2 个高速 I²C 接口 (400Kb/s)、SPI 和具有缓冲作用和数据长度可变功能的 SSP; 多达 47 个可承受 5V 电压的通用 I/O 口;
- ⑦多达 9 个边沿或电平触发的外部中断引脚; 通过片内 PLL 可实现最大为 60MHz 的 CPU 操作频率; 片内晶体振荡电路支持频率: 1~30MHz; 2 个低功耗模式: 空闲和掉电;
- ⑧通过外部中断将处理器从掉电模式中唤醒; 单电源供电, 含有上电复位 (POR) 和掉电检测 (BOD) 电路。CPU 操作电压为 3.0~3.6V。

以上两种方案中, 我们最终选择了方案二。理由如下: ①ARM 的性价比很高, 如今只需要几十元钱就能够买到功能强大的 ARM 微控制器; ②在开始本课题之前, 我们已经对 LPC213X 系列的 ARM 微控制器有过系统的学习。

3.7 其它电路

3.7.1 界面 LED 电路

在 3.1 电脑鼠的机身设计一节中, 我对 LED 电路有过介绍。在本节中, 重复的部分就不再赘述。我设计的迷宫电脑鼠有 5 个独立的 LED, 分别是 D1、D2、D4、D5 和 D7, 各指示灯的具体含义请参见 3.1 节。它们通过微控制器系统的 GPIO 口直接控制, 其电路图如图 3.16 所示。电路采用了 I/O 口灌电流的驱动方式来驱动 LED。GPIO 引脚输出高电平时 LED 熄灭, 低电平时点亮。当 I/O 口不够用时, 可以考虑 JTAG 接口与 LED 共用 I/O 口。参考 3.2 节内容, 限流电阻 R2 电阻值的计算公式:

$$R2 = \frac{3.3 - 1.5}{10 \times 10^{-3}} = 180\Omega$$

我们实际选用的限流电阻 R2 为 220Ω, 其阻值略大于计算值。

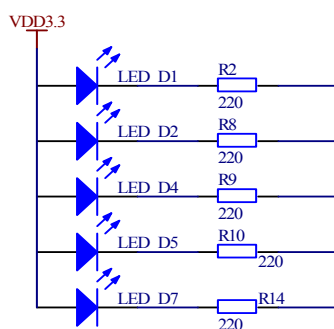


图 3.16 指示灯 LED 电路

3.7.2 JTAG 接口电路

本设计中采用 ARM 公司提出的标准 20 脚 JTAG 仿真调试接口，JTAG 信号的定义及与 LPC2138 的连接如图 3.17 所示。

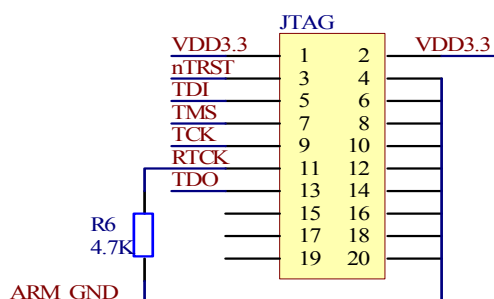


图 3.17 JTAG 接口电路

根据 LPC2138 的应用手册说明，在 RTCK 引脚接一个 4.7K 的下拉电阻，使系统复位后 LPC2138 内部 JTAG 接口使能，这样就可以直接进行 JTAG 仿真调试。如果用户需要使用 P1.26~P1.31 作 I/O 口，不进行 JTAG 仿真调试，则可以在用户程序中通过设置 PINSEL2 寄存器来使 LPC2138 内部 JTAG 接口禁止。

3.7.3 独立按键电路

我们给迷宫机器人电脑鼠设计 2 路独立的输入按键，直接输入到 ARM LPC2138 的 GPIO 引脚，如图 3.18 所示。当按键未按下时，R3、R4 起上拉电阻的作用，LPC2138 检测到引脚为高电平；当按键按下时，LPC2138 检测到引脚为低电平。

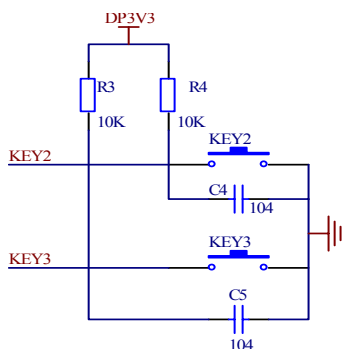


图 3.18 独立按键电路

3.7.4 复位电路

由于 ARM 芯片的高速、低功耗和低工作电压导致其噪声容限低，对电源的纹波、瞬态响应性能、时钟源的稳定性和电源监控可靠性等诸多方面也提出了更高的要求。本系统中的复位电路使用带 I²C 存储器的电源监控芯片 CAT1025JI-30，提高了系统的可靠性。其电路原理如图 3.19 所示。

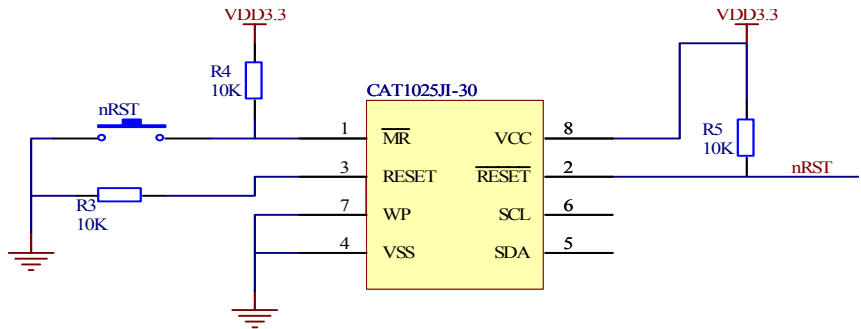


图 3.19 系统复位电路

在图 3.19 中，信号 nRST 连接到 LPC2138 芯片的复位 RESET，当复位按键 RST 按下时，CAT1025JI-30 的 RESET 引脚立即输出复位信号，LPC2138 芯片复位。使用 CAT1025JI-30 芯片时，其 RESET 引脚上的下拉电阻和 RESET 引脚上的上拉电阻都是不能省略。

3.8 本章小结

本章是迷宫电脑鼠的硬件设计部分。从文中可知，一只完整的迷宫电脑鼠主要由机身、电源、传感器、微处理器、电机及人机界面这六大部分组成。因此，我们硬件设计的工作重点就在这六大模块。

本章详细叙述了迷宫电脑鼠机身设计、电源模块设计、电机驱动模块设计、测速电路模块设计、迷宫挡板检测电路、微处理器模块设计这六大主要功能模块的具体实现方案，有的设计甚至提供了多套方案。对于电脑鼠的其它硬件电路，如界面 LED 电路、JTAG 接口电路、独立按键电路、复位电路等也作了介绍。通过阅读本章的内容，相信读者一定能够把迷宫电脑鼠的硬件制作出来。至此，我们的工程成功了一半！

第四章 迷宫机器人的路径规划

机器人路径规划^[32]是根据机器人作业要求，在具有障碍的环境内，按照一定的评价标准，对末端执行器在工作过程中的状态（包括位置、姿态、速度、加速度等）进行设计，寻找一条从起始点到目标点的无碰撞最优路径。

迷宫机器人的路径规划问题是移动机器人路径规划的一个分支。迷宫机器人是种轮式移动机器人，它的运动规划问题主要研究在未知或少知环境中机器人移动路径的确定方法。为了更好地研究机器人的路径规划，我把研究范围扩展到整个移动机器人领域。

4.1 移动机器人路径规划概述

目前，移动机器人路径规划的研究是机器人学领域研究的热点问题之一^{[37][38]}。其最基本的路径规划问题是在完全已知的静态障碍物之间为机器人寻找一条从给定的起始状态到目标状态的满足一定优化指标的无碰撞路径。以 ANSP 表示无碰撞的自由位形空间，路径规划问题可以描述为：给定一个起始状态点 G_{begin} 和目标状态点 G_{end} 在 ANSP 中寻找一条连接这两点的连续曲线，并且要满足某些性能指标，例如：行走路线距离最短、行走时间最短、工作代价最小或者确定不存在这样的连接。

移动机器人路径规划技术是智能机器人领域中的核心问题之一，也是机器人学中研究人工智能问题的一个重要方面^{[40][42]}。我们希望未来的机器人能具有感知、规划和控制等高层能力。它们能从周围的环境中收集知识，构造一个关于环境的符号化的世界模型，并且利用这些模型来规划、执行由使用者下达的高层任务^[43]。其中的规划模块能生成大部分机器人要执行的命令，其目标是实现机器人的使用者在较高层次上给机器人下达一些较宏观的任务，由机器人系统自身来填充那些较低层的细节问题。

还有，路径规划是移动机器人完成任务的安全保障；同时也是移动机器人智能化程度的重要标志。尤其是在机器人硬件系统的精度在短期内不能得到解决的情况下，对路径规划算法的研究显得尤为重要。此研究将从根本上改变移动机器人的导航性能，将提高移动机器人的智力水平，减少移动机器人在移动过程中存在的不确定状态，提高移动机器人移动的速度及灵活性，为开发高智能远距离搬运机器人、探测机器人、服务机器人、汽车自动驾驶系统打下良好的基础^{[36][37]}。下面我将简单叙述下国内外移动机器人路径规划的方法。

4.2 移动机器人的路径规划方法

到本世纪初，关于机器人路径规划的研究成果是非常丰富的。下面我就对其中一些影响较大的关于移动机器人路径规划的研究成果进行介绍。

移动机器人的路径规划方法根据其运用处理算法的不同，可以分为两类：经典方法和

4.2.1 移动机器人路径规划的经典方法

移动机器人路径规划的经典方法大致可以分为以下三类：可视图法、栅格法、势场法。

（一）可视图法

可视图法视机器人为一点，将机器人、目标点和多边形障碍物的各顶点进行组合连接，要求机器人和障碍物各顶点之间、目标点和障碍物各顶点之间以及各障碍物顶点与顶点之间的连线，均不能穿越障碍物，即直线是可视的。搜索最优路径的问题就转化为从起始点到目标点经过这些可视直线的最短距离问题。运用优化算法，可删除一些不必要的连线以简化可视图，缩短搜索时间。常用的优化方法有搜索算法等。该法能够求得最短路径，但假设机器人的尺寸大小忽略不计，使得机器人通过障碍物顶点时离障碍物太近甚至接触，并且搜索时间长。另外的缺点就是此法缺乏灵活性，即一旦机器人的起点和目标点发生改变，就要重新构造可视图，比较麻烦。已经证明用可视图法计算对于 N 条连线的搜索时间为 $O(N^2)$ 。可视图法能求得最短路径，但缺乏灵活性，若障碍物过多，搜索时间会很长。如果采用优化算法删除一些不必要的连线，可以简化可视图，缩短搜索时间。因此，可视图法适用于多边形障碍物，但对于圆形障碍物该法失效。

（二）栅格法

栅格法^[28]是由 W.E.Howden 在 1968 年提出的，他在进行移动机器人路径规划时采用了栅格(grid)表示地图，在处理障碍物的边界时，避免了复杂的计算。这种算法是将机器人的运动空间用固定大小的栅格分成许多小空间，每个栅格代表一个节点，所有节点分为障碍物节点和自由节点两大类。每个自由节点连接相邻的自由节点组成一个自由区，用搜索算法生成从初始节点到目标节点的路径。可以看出栅格粒度的大小与环境描述的精确度以及搜索算法的计算两密切相关，栅格划分越细，障碍物的表示越精确，但会占据大量的存贮空间，算法的搜索范围将成指数增加，如果栅格粒度太大，计算量小，但规划的路径会很不精确。所以栅格粒度大小的确定，是栅格法中的主要问题。

栅格法的基本思想是将一块实际场地划分成等面积的小区，每个小区称为一个栅格，栅格面积的大小一般由机器人车体大小来决定。图 4.1 显示了一块 15×15 规格的迷宫场地栅格划分情况。其中深色小块表示当前栅格有障碍，空格表示当前栅格无障碍。

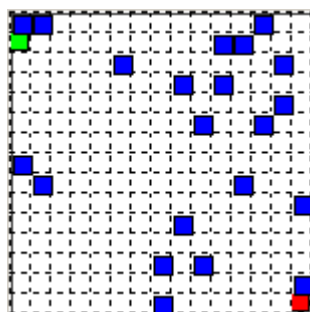


图 4.1 迷宫栅格场地

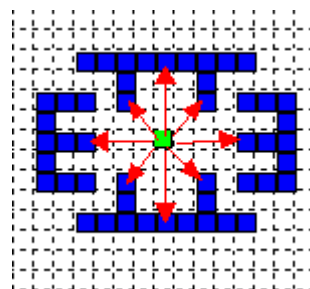


图 4.2 机器人的八种运动方式

我们利用栅格法进行任务规划的目的是建立所谓的数字地图，在地图中存储移动机器人的运行轨迹和障碍等相关信息。栅格法要将连续路径信息离散化，这样在地图中记录的路径信息和实际路径信息会有误差，误差的大小由栅格大小，即地图分辨率决定。机器人的路径信息被离散化后，机器人的运动轨迹被分解为单个的运动，这些单个的运动被记录在每个栅格上。每个栅格上的运动信息规定了机器人在这个栅格上的运动方向(例如，向前，向后，向左，向右等)。理论上，移动机器人在每个栅格上的微小运动有无数种，但考虑到建模的复杂性，一般在实际运用中只定义了 8 种运动，它们分别是：前、后、左、右、右前、右后、左后、左前，如图 4.2 所示。

我们采用栅格法的模型建立起来以后，就应该对模型进行信息的编码。最常见的信息编码格式如下：

①障碍物信息：1—当前栅格有障碍，0—当前栅格无障碍。

②迷宫地图上的栅格采用行列划分的矩阵存储方法，如图 4.1 所示，深色小块代表一个栅格，记为 $G(i, j)$ ，这个栅格的信息将记录在存储器的栅格存储阵列中的第 i 行，第 j 列，采用这种“栅格—存储”映射办法，可以建立起整幅地图。

(三) 势场法

势场法把要达到的目标点当成一个具有吸引力的低谷，而障碍物（包括已知的环境障碍和突然出现的障碍）则当成具有排斥力的高山或壁垒。机器人犹如处在高山与低谷构成的环境中，利用这种自然势能所造成的斜坡产生滑动，使机器人自然地“陷入”低谷。

计算势场的方法有很多，例如可以模拟静电场理论用计算电位差的方法计算势能分布，从而确定某一点机器人应受吸引力的数量和吸引力的方向。由于环境的复杂性，通常的势场法总会存在一些奇异点，在这些点上，各方向力互相抵消，使机器人失去行走的“动力”和“方向”。犹如在复杂地形中，出现一小块平地，使机器人失去“滑动力”。若按照一种方法计算势能，死区是不可避免的，但若加入各种逻辑判断，则可以设法摆脱死区。

下面我们再讲讲目标势场函数的建立。通常，我们可以把移动机器人简化为一个质点，其移动空间是二维的，机器人在移动空间中的位置为 $X = [x, y]^T$ 。目标势场函数如下：

$$U_{at}(X) = \frac{1}{2}k(X - X_{goal})^2 \quad (4.1)$$

其中 k 为位置增益， $(X - X_{goal})$ 为当前与目标位置的相对距离，相应的吸引力为目标势场的负梯度：

$$F_{at}(X) = -grad[U_{at}(X)] = k(X - X_{goal}) \quad (4.2)$$

当机器人接近目标过程这个量收敛到零。

通常的斥力场函数为：

$$U_{re}(X) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)^2; & \rho \leq \rho_0 \\ 0 & \rho > \rho_0 \end{cases} \quad (4.3)$$

其中 η 为位置增益系数, ρ 为机器人在空间的位置 X 与障碍物之间的最短距离, ρ_0 是一个常数, 代表障碍物的影响距离。

相应的斥力为:

$$F_{re}(X) = -grad[U_{re}(X)] = \begin{cases} \eta \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 \frac{1}{\rho^2} \frac{\partial \rho}{\partial X}; \rho \leq \rho_0 \\ 0 \end{cases} \quad (4.4)$$

其中 $\frac{\partial \rho}{\partial X} = [\frac{\partial \rho}{\partial x}, \frac{\partial \rho}{\partial y}]^T$ 。

机器人所受的合力为:

$$F_{sum} = F_{re} + F_{at} \quad (4.5)$$

合力 F_{sum} 决定了机器人的运动。实际上, 当我们用这个力来控制机器人时, 如果目标在障碍物的影响范围之内, 则机器人永远也到不了目标。因为当机器人向目标靠近时, 距离障碍物也越来越近, 这样吸引力减小, 斥力增加, 机器人将受到排斥而不是吸引。这个问题在以前的文献中很少提及, 但它确实存在, 尤其是在像迷宫这样狭窄的环境中, 值得研究。

4.2.2 移动机器人路径规划的智能方法

近 20 多年以来, 智能化方法的深入研究带动了基于智能方法的机器人路径规划^{[44][45]}的研究, 其中比较成熟的智能路径规划方法主要有模糊逻辑方法、神经网络法和遗传算法。

(1) 模糊逻辑法

基于模糊推理^[29]的路径规划方法参考人的驾驶经验, 通过查表的方法, 实现实时局部路径规划。这种方法通过在移动机器人上装配感应器来分辨障碍物, 克服了其它方法的缺点, 在动态变化的未知环境中能够进行实时规划。该方法最大的优点是实时性非常好, 但是模糊隶属函数的设计、模糊控制规则的制定主要靠人的经验, 如何得到最优的隶属函数以及控制规则是该方法最大的问题。近年来一些学者引入神经网络技术, 提出一种模糊神经网络控制的方法, 效果较好, 但复杂度太高。

(2) 神经网络

近年来, 随着人工神经网络研究的深入, 人们已将神经网络引入到智能机器人路径规划和导航研究^{[46][47][48]}中, 通过实例教学使网络收敛, 学习完成后的网络, 除了其固有的并行特性外, 还具有一定的容错能力, 并且对学习未遇到的情况, 也能进行一定的处理。机器人系统是一个实时性要求很高的非线性系统, 人工神经网络提供了解决这方面问题的可能方法。

我们知道神经网络是一个高度并行的分布式系统, 所以可用来完成对视觉系统探测到的图像进行处理^{[49][50][51]}。它不仅处理速度快, 还可以充分利用其非线性处理能力达到环境

及陆标辨识的目的，还可以完成机器人内部坐标和全局坐标的快速转换。另外，基于环境拓扑结构组织的网络，在给出目标后，还可以通过网络能量函数的收敛得到一条最优途径。

神经网络的应用还表现在避障和路径规划方面。由于避障和路径规划工作没有明显的规则和难以进行时间分类，可以让神经网络通过大量的实例学习来掌握。由于不需要迭代，采用前向网络学习算法来学习避障行为时，速度很快；Kohonen 网络的自组织特性也可以用来融合传感器信息，学习从地图上不同位置到目标点的行走路线，一旦学习完成，机器人就能够实现自主导航，并且规划出最短无碰撞路径。

神经网络算法应用于机器人路径规划的最大不足之处在于其过长的运算时间，而且在机器人运动空间数据不够详细完备时，还可能发生不收敛或所规划路径不可行等棘手问题。

(3)遗传算法

遗传算法(Genetic Algorithm)简称 GA，它是以遗传学和自然选择等生物进化理论为理论基础所构造的一种随机的群体搜索算法，并在某种程度上对生物进化过程进行了数学方式的模拟。1962 年 Holland 教授首次提出了遗传算法^[14]的思想，从而吸引了大批的研究者，把遗传算法迅速推广到优化、搜索、机器学习等方面，并奠定了坚实的理论基础。

众所周知，生物的进化是以集团为主体的。与此相对应，遗传算法的运算对象是由 M 个个体所组成的集合，称为群体。与生物一代又一代的自然进化过程相类似，遗传算法的运算过程也是一个反复迭代过程，第 t 代群体记做 P(t)，经过一代遗传和进化后，得到第 t+1 代群体，它们也是由多个个体组成的集合，记做 P(t+1)。这个群体不断地经过遗传和进化操作，并且每次都按照优胜劣汰的规则将适应度较高的个体更多地遗传到下一代，这样最终在群体中将会得到一个优良的个体 X，它所对应的表现型 X 将达到或接近于问题的最优解。

过去 40 多年中，在解决复杂的全局优化问题方面，遗传算法已取得了成功的应用，并受到了人们广泛的关注。与传统优化方法相比，遗传算法具有较好的健壮性，有以下特点：

- (1)遗传算法是对参数的编码进行操作，而不是对参数本身。
- (2)搜索过程是从一组解迭代到另一组解，这样避免陷入局部最优解的可能性，而有较大的可能求得全部最优解。
- (3)使用的是随机搜索过程，而非确定性搜索过程。
- (4)遗传算法对于函数基本上没有任何特殊要求，只利用适应度信息，不需要微分等其它辅助信息，应用范围更广^{[52][53]}。

因此，我们可以用遗传算法来解决移动机器人最优路径选择问题。

4.3 基于蚁群算法的机器人路径规划

20 世纪 90 年代，意大利学者 Dorigo M, Maniezzao V, Colorni A 等人^[15]从生物进化的机理中受到启发，通过模拟自然界蚂蚁寻食的行为，提出了一种全新的模拟进化算法：蚁群优化算法 ACOA(Ant Colony Optimization Algorithm)。

蚁群优化算法^{[58][59][60]}提出后，研究者对其理论以及应用问题作了大量的工作。首先，

在理论领域将蚂蚁视为图论中的结点，建立蚁群算法的图论模型。其次，不断地对基本蚁群算法进行改进，得到集中改进的蚁群优化算法：带精英策略的蚁群系统、基于优化排序的蚁群系统、自适应蚁群算法以及蚁群算法与遗传算法结合的 GAAA 算法。

随着蚁群算法理论研究的深入，其应用研究领域也在不断地拓展。主要的应用领域有：旅行商问题(TSP)、指派问题(QAP)、调度问题(JSP)、车辆路线问题(VRP)、有向连接网络路由、无向连接网络路由、有序排列问题(SOP)、图着色问题(GCP)、频率分配问题(FAP)、一般分配问题(GAP)、多重背包问题(MKP)、光学网络路由、冗余分配问题(RAP)、机器人路径规划(ACA-RPP)等^{[61][62]}。

虽然，蚁群算法在求解复杂优化问题方面很具优势，但是，目前蚁群算法在机器人路径规划领域的文章比较少。而且，基于蚁群算法所建立的模型与方案不是很成熟，仍具有很大的可研究空间。

机器人的路径规划问题属于一种带约束条件的连续函数优化问题，研究解决此类问题的蚁群优化算法，能够扩大 ACOA 算法的应用领域，并能探索与改进一种新的路径优化算法，促进优化理论与实践的发展，并且为经济领域以及工程领域的优化问题(如生产调度、系统控制、物流配送等)提供借鉴。

蚁群算法是基于生物界群体启发行为的一种随机搜索寻优方法，它的正反馈性和协同性使其可用于分布式系统，隐含的并行性使其具有极强的发展潜力，灵活性使其在解决组合优化问题上具有良好的适应性。因此，将蚁群算法应用于智能移动机器人避障的路径规划问题研究，应该说有很好的理论与现实意义。我将在随后的章节中将经过改进的蚁群算法应用于迷宫机器人的路径规划。

4.4 本章小结

本章首先介绍什么是移动机器人的路径规划。然后详述移动机器人的路径规划方法，根据其运用处理算法的不同，可以将移动机器人的路径规划方法分为两类：经典方法和智能方法。其中，移动机器人路径规划的经典方法大致可以分为以下三类：可视图法、栅格法、势场法；而比较成熟的智能路径规划方法主要有模糊逻辑方法、神经网络法和遗传算法。本章的最后介绍了基于蚁群算法的机器人路径规划，为下一章全面介绍蚁群算法及其在迷宫最优路径问题的应用作了铺垫，起到很好的承上启下作用。

第五章 迷宫机器人的控制算法设计

迷宫机器人是竞赛机器人的一种重要形式，它在其核心单元——控制器中有一套“规则”，控制器按照这个“规则”来控制各个部件协调工作，而这套“规则”的核心就是数学算法。

目前，求解迷宫问题涉及了人工智能技术中的搜索问题，常用的搜索策略有：深度优先搜索（DFS）和广度优先搜索（BFS）、遗传算法（GA）、蚁群算法（ACA）等，每种算法都有自己的特点。

在介绍迷宫机器人的数学算法之前，我先介绍一下有关迷宫场地的设计与制作。

5.1 迷宫场地的设计与制作

5.1.1 迷宫机器人场地标准

迷宫机器人（电脑鼠）场地共由 16×16 个网格组成，起点在一角，终点占据中央的四个网格，如图 5.1 所示。每个网格尺寸为 $18\text{cm} \times 18\text{cm}$ （长 \times 宽），隔栅尺寸为 $5\text{cm} \times 1.2\text{cm}$ （高 \times 厚），上红、侧白、地板为黑色，迷宫机器人属于自主型，不允许人为操作。迷宫机器人的尺寸限制为 $25\text{cm} \times 25\text{cm}$ （长 \times 宽），高度及重量不限。



图 5.1 迷宫电脑鼠标准场地外观

5.1.2 迷宫场地的设计与加工

迷宫场地由如下部件组成：小方柱、隔栅、地板。其中隔栅是迷宫的挡板，而小方柱

用于隔栅间的互连。

迷宫的场地由复合木板拼接而成。由于复合木板的标准尺寸是 $19.2\text{cm} \times 121\text{cm}$ ，这里取小方柱和隔栅的宽度为 1.2cm ，则迷宫的跑道恰好是 18cm ，符合国际标准。这样一来，用复合木地板拼接迷宫场地非常方便，工艺简单，价格低廉。加工工具只需要尺子和手钻即可。

5.2 迷宫的表示

1995 年 IEEE 国际迷宫机器人竞赛用场地如图 5.2 所示。图中 S 表示迷宫电脑鼠的出发点，E 是终点，中间的线条表示迷宫的隔栅。我们可以看到，从同样的起点到同一个终点会有多条不同的路径。

那么迷宫在计算机中如何表达呢？我们将迷宫隔栅（挡板）看作 1，迷宫空格看作 0，于是假定了一个迷宫就可以开始解迷宫的路径。经过分析，我们可以把 $12\text{mm} \times 12\text{mm}$ 的小方柱抽象成 $180\text{mm} \times 180\text{mm}$ 的方隔栅，同时再把有一定厚度的 $180\text{mm} \times 12\text{mm}$ 的隔栅也抽象成 $180\text{mm} \times 180\text{mm}$ 的方隔栅。空格还是尺寸为 $180\text{mm} \times 180\text{mm}$ 的空格。

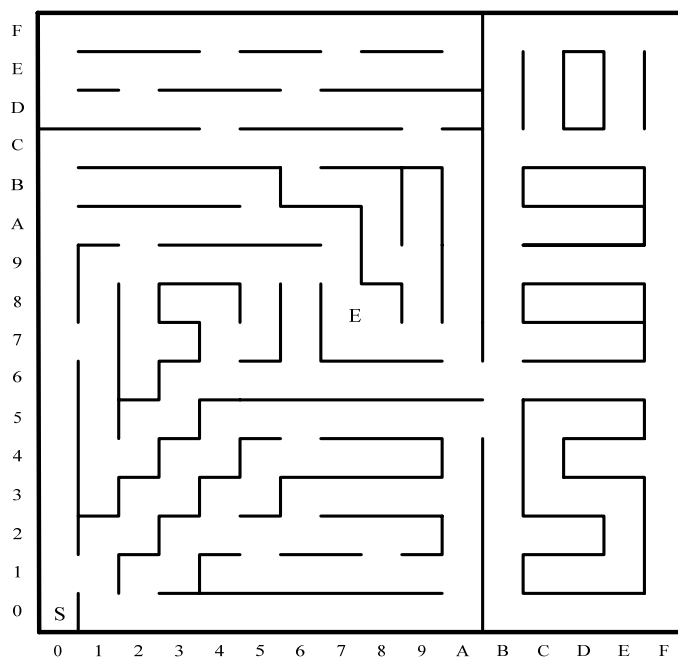


图 5.2 1995 年 IEEE 国际标准电脑鼠迷宫竞赛用场地

我们采用上述的方法把实际迷宫抽象化^[28]，再把小方柱隔栅看作“1”，把空格看作“0”，这时又可以把迷宫数字化。如图 5.3 所示的已经被抽象化和数字化的迷宫可以表示为一个二维数组 $\text{maze}(m,n)$ 。

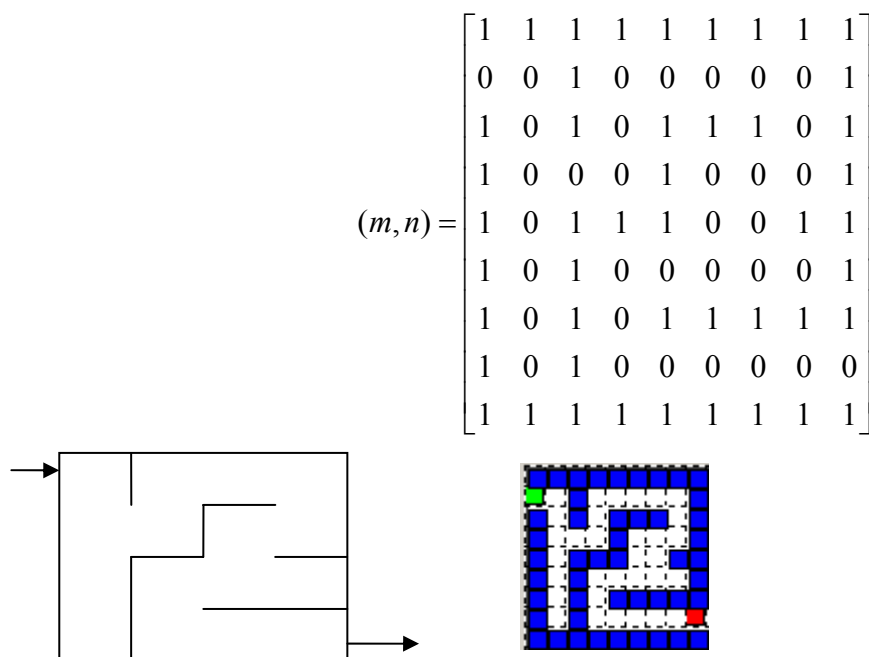


图 5.3 迷宫的数字表示

5.3 迷宫机器人常用搜寻法则和策略

5.3.1 迷宫搜寻法则

设定搜寻法则和策略是为了电脑鼠可以以最快的方式找到终点，到达目标后随即从所走过的路径中找出一条可行路径返回起点，然后再做冲刺，直达目的地；法则的设定很重要，它可以使电脑鼠不多走冤枉路，可节省很多时间而制胜。

每一只电脑鼠到达一方格时它最多有三个方向可前进，最少则因为三面都有墙，没有可以前进的方向；当遇到二个以上的可选择方向时，由于不同场合的需要，就有不同优先搜寻的方向顺序。常见的法则有以下几种：

①右手法则：遇叉路时，以右边为优先前进方向，然后直线方向，左边方向；②左手法则：遇叉路时，以左边为优先前进方向，然后直线方向，右边方向；③中左法则：与右手法则相似，不过方向选择顺序改为直线优先，然后左边，右边；④中右法则：遇叉路时，以直线为优先前进方向，然后右边方向，左边方向；⑤求心法则：遇叉路时，以距中心最短的那个方向优先，然后依次选择。⑥乱数法则：以电脑鼠的随机值作为下一个前进方向。

5.3.2 迷宫搜寻策略

迷宫搜寻模式有全迷宫搜寻策略和部分迷宫搜寻策略两种：①全迷宫搜寻策略：迷宫电脑鼠以任一搜寻法则前进到达终点后，电脑鼠会反身继续前进，然后以原设定的搜寻法则，时时检查未走过的路，直到每一方格都搜寻过后，才回到起点。②部分迷宫搜寻策略：迷宫电脑鼠以任一搜寻法则前进到达终点后，电脑鼠将沿原路线返回起点，不再进行其它搜寻。

如果比赛规则不计算搜寻时间,可采用全迷宫搜寻策略,待地毯式的搜寻过所有方格后,再计算最佳路径,作最后的冲刺,冲刺成绩一定相当不错。由于新制国际比赛规则加入搜寻时间的成绩计量,因此我们必须考虑部分迷宫搜寻策略,甚至还须考虑加入求心法则,截路径功能等更智慧的法则来协助;此时找到的路径可能不是最佳路径,但保证花的时间最短。

5.4 求解迷宫问题的经典算法

迷宫问题是典型的图的搜索问题,常用的搜索策略有:广度优先搜索法(BFS)和深度优先搜索法(DFS)。通常采用栈来实现迷宫的深度搜索,从而找到一条路径,但是却不是从入口到出口的最短路径;搜索最短路径通常用的方法是广度优先搜索,是利用队列来完成的。

5.4.1 迷宫的广度优先搜索法

为了找到最短路径,我们用队列 SQueue 来保存搜索过的位置。搜索的规则是:选择一个可通行的位置而又没有试探过的单元 p ,沿四个方向作搜索,若这四个方向中有可行通路又没有搜索过的单元 q ,则说明从 p 到 q 有通路,将 q 记录在 SQueue 中。SQueue[m, n] 是一个三元数组,用来记录搜索过的单元, SQueue[k].x 记录第 k 次搜索的单元 q 的行坐标, SQueue[k].y 记录 q 的列坐标, SQueue[k].pre 记录产生 q 的单元,即 p 单元在 SQueue 中的下标,记录每个搜索单元的产生轨迹或记录其父节点,目的是为了在找到出口后沿这个下标返回到入口处,方便最后显示这条最短路径。这四个方向都搜索完了(修改 q 单元为已搜索过),接下来从 SQueue 中取出队头,往下继续搜索。在搜索过程中,如果遇上了出口单元,则认为已经找到了一条从入口到出口的最短路径;如果迷宫中所有可通行的道路都搜索完,都没有遇上出口单元,此时认为该迷宫是无法通过的。

5.4.2 迷宫的深度优先搜索法

深度优先搜索所遵循的搜索策略是尽可能“深”地搜索图,这点正如算法名称那样。在深度优先搜索中,对于最新发现的节点,如果它还有以此为起点而未探测到的边,就沿此边继续探下去。当节点 v 的所有边都已被探寻过,搜索将回溯到发现节点 v 有那条边的始节点。这一过程一直进行到已发现从源节点可达的所有节点为止。如果还存在被发现的节点,则选择其中一个作为源节点并重复以上过程,整个过程反复进行,直到所有节点都被发现为止。

为了避免考虑太长的路径,防止搜索过程沿着无益的路径扩展下去,我们往往给出一个节点扩展的最大深度——深度界限。任何节点如果达到了深度界限,都将把它们作为没有后继节点处理。值得说明的是,即使应用了深度界限的规定,所求得的答案路径并不一

定就是最短的路径。含有深度界限的深度优先搜索算法如下：

- (1) 把起始节点 S 放到未扩展节点 OPEN 表中。如果此节点为一目标节点，则得到一个解。
- (2) 如果 OPEN 为一空表，则失败退出。
- (3) 把第一个节点（节点 n）从 OPEN 表移到 CLOSED 表。
- (4) 如果节点 n 的深度等于最大深度，则转向（2）。
- (5) 扩展节点 n，产生其全部后裔，并把它们放入 OPEN 表的前头。如果没有后裔，则转向（2）。
- (6) 如果后继节点中有任意一个节点为目标节点，则求得一个解，成功退出；否则，转向（2）。

5.5 蚁群算法的应用

5.5.1 蚁群算法的起源与研究进展

根据蚂蚁“寻找食物”的群体行为，意大利著名学者 Dorigo M, Maniezzao V, Colorni A 等人于 1991 年在法国巴黎召开的第一届欧洲人工生命会议（European Conference on Artificial Life, ECAL）上最早提出了蚁群算法的基本模型；1992 年，Dorigo M 又在其博士学位论文中进一步阐明了蚁群算法的核心思想^[54]。

蚁群算法(ant colony algorithm)是一种基于种群的启发式仿生进化系统。蚁群算法最早成功应用于解决著名的旅行商问题(traveling salesman problem, TSP)，该算法采用了分布式正反馈并行计算机制，易于与其他方法结合，而且具有较强的鲁棒性。

蚁群算法创立十多年来，无论在算法理论还是在算法应用方面都取得了很多突破性研究进展。它已经从单纯地解决对称 TSP 拓展到解决非对称 TSP、指派问题（quadratic assignment problem, QAP）以及车间作业调度问题（job-shop scheduling problem, JSP）等。尤其是近几年来，作为一个前沿性的热点研究领域，蚁群算法已引起越来越多国内外研究者的关注，其研究人员和研究成果均成几何级数增长^{[61][62][64]}。有统计表明，2000 年蚁群算法的相关学术论文还不足 200 篇；而截至 2005 年 11 月，蚁群算法的相关学术论文已经超过了 1800 篇，其应用范围几乎涉及到各个优化领域，而且还出现了蚁群算法仿生硬件，可见这种新兴的仿生优化算法已经显示出强大的生命力和广阔的发展前景^{[69][70]}。

尽管人们对蚁群算法的研究时间不长，在这一领域还有很多问题需要进一步地研究和解决，但是通过理论研究和实际应用表明，蚁群算法是一种很有前途的仿生优化算法。随着人类认识的进步和社会发展的加速，仿生智能及最优化系统理论将越来越成为科学认识和工程实践^[71]的有力工具。因此，关于蚁群算法理论及其应用的研究必将是一个长期的研究课题。同时，我们有理由相信随着人们对仿生智能系统理论及应用研究的不断深入，蚁群算法这一新兴的仿生优化算法必将展现出前所未有的勃勃生机。

5.5.2 基本蚁群算法的原理和数学模型

1 基本蚁群算法的机制原理

模拟蚂蚁群体觅食行为的蚁群算法是作为一种新的计算智能模式引入的，该算法基于如下基本假设：①蚂蚁之间通过信息素和环境进行通信。每只蚂蚁仅根据其周围的局部环境做出反应，也只对其周围的局部环境产生影响；②蚂蚁对环境的反应由其内部模式决定。因为蚂蚁是基因生物，蚂蚁的行为实际上是其基因的适应性表现，即蚂蚁是反应型适应性主体；③在个体水平上，每只蚂蚁仅根据环境做出独立选择；在群体水平上，单只蚂蚁的行为是随机的，但蚁群可通过自组织过程形成高度有序的群体行为^[57]。

从上述假设和分析可见，基本蚁群算法的寻优机制包括两个基本段：适应阶段和协作阶段。在适应阶段，各候选解根据积累的信息不断调整自身结构，路径上经过的蚂蚁越多，信息量越大，则该路径越容易被选择；时间越长，信息量会越小；在协作阶段，候选解之间通过信息交流，以期望产生性能更好的解，类似于学习自动机的学习机制。

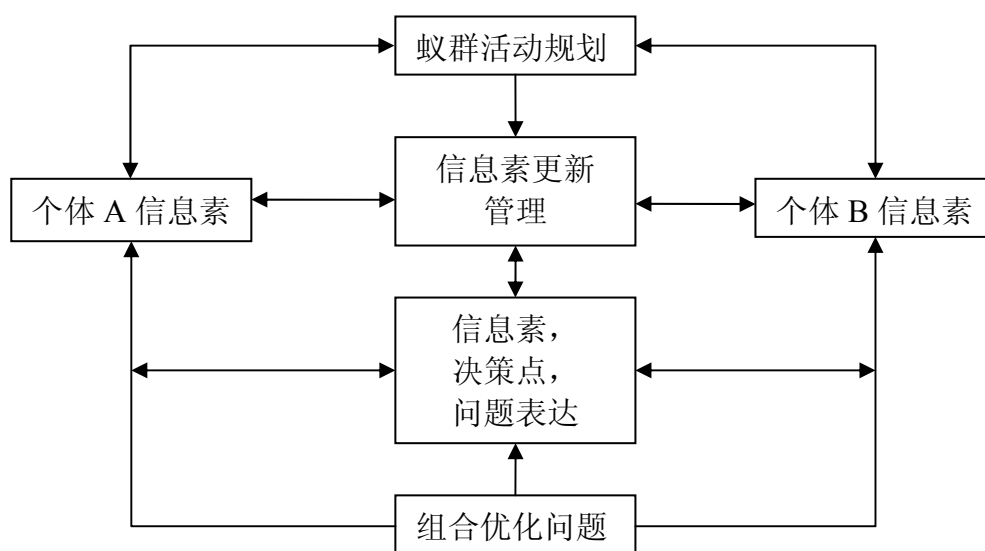


图 5.4 基本蚁群算法的逻辑结构

蚁群算法实际上是一类智能多主体系统，其自组织机制使得蚁群算法不需要对所求问题的每一个方面都有详尽的认识。自组织本质上是蚁群算法机制在没有外界作用下使系统熵增加的动态过程，体现了从无序到有序的动态演化，其逻辑结构如图 5.4 所示。

从图 5.4 可见，先将具体的组合优化^{[55][56]}问题表述成规范的格式，然后利用蚁群算法在“探索(exploration)”和“利用(exploitation)”之间根据信息素这一反馈载体确定决策点，同时按照相应的信息素跟新规则对每只蚂蚁个体的信息素进行增量构建，随后从整体角度规划出蚁群活动的行为方向，周而复始，即可求出组合优化问题的最优解。

2 基本蚁群算法的数学模型

设 $b_i(t)$ 表示 t 时刻位于元素 i 的蚂蚁数目， $\tau_{ij}(t)$ 为 t 时刻路径 (i, j) 上的信息量， n 表示 TSP 的规模， m 为蚁群中蚂蚁的总数目，则有： $m = \sum_{i=1}^n b_i(t)$ ； $\Gamma = \{\tau_{ij}(t) | c_i, c_j \in C\}$ ；其中 $C = \{c_1, c_2, \dots, c_n\}$ 是 t 时刻集合 C 中元素（城市）两两连接 l_{ij} 上残留信息量的集合。在初始时

刻各条路径上信息量^[65]相等，并设 $\tau_{ij}(0) = \text{const}$ ，基本蚁群算法的寻优是通过有向图 $g = (C, L, \Gamma)$ 实现的。

蚂蚁 k ($k = 1, 2, 3, \dots, m$) 在运动过程中，根据各条路径上的信息量决定其转移方向。这里用禁忌表 tabu_k ($k = 1, 2, 3, \dots, m$) 来记录蚂蚁 k 当前所走过的城市，集合随着 tabu_k 进化过程做动态调整。在搜索过程中，蚂蚁根据各路径上的信息量及路径的启发信息来计算状态转移概率。 $p_{ij}^k(t)$ 表示 t 时刻蚂蚁 k 由元素(城市) i 转移到元素(城市) j 的状态转移概率。

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} & , \text{若 } j \in \text{allowed}_k \\ 0 & , \text{否则} \end{cases} \quad (5.1)$$

式中， $\text{allowed}_k \{C - \text{tabu}_k\}$ 表示蚂蚁 k 下一步允许选择的城市； α 为信息启发式因子，表示轨迹的相对重要性，反映了蚂蚁在运动过程中所积累的信息在蚂蚁运动时所起的作用，其值越大，则该蚂蚁越倾向于选择其他蚂蚁经过的路径，蚂蚁之间协作性越强； β 为期望启发式因子，表示能见度的相对重要性，反映了蚂蚁在运动过程中启发信息在蚂蚁选择路径中的受重视的程度，其值越大，则该状态转移概率越接近于贪心规则； $\eta_{ij}(t)$ 为启发函数，其表达式如下：

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (5.2)$$

式 (5.2) 中， d_{ij} 表示相邻两个城市之间的距离。对蚂蚁 k 而言， d_{ij} 越小，则 $\eta_{ij}(t)$ 越大， $p_{ij}^k(t)$ 也越大。显然，该启发函数表示蚂蚁从元素（城市） i 转移到元素（城市） j 的期望程度。

为了避免残留信息素过多引起残留信息淹没启发信息，在每只蚂蚁走完了一步或者完成对所有 n 个城市的遍历(也即一个循环结束)后，要对残留信息进行更新处理。这种更新策略模仿了人类大脑记忆的特点，在新信息不断存入大脑的同时，存储在大脑中的旧信息随着时间的推移逐渐淡化，甚至忘记。由此，第 $t+n$ 时刻路径 (i, j) 上的信息量可按如下规则进行调整。

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (5.3)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5.4)$$

式中， ρ 表示信息素挥发系数，则 $1-\rho$ 表示信息素残留因子，为了防止信息的无限积累， ρ 的取值范围为： $\rho \in [0, 1)$ ； $\Delta\tau_{ij}(t)$ 表示本次循环中路径 (i, j) 上的信息素增量，初始时刻 $\Delta\tau_{ij}(0) = 0$ ， $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在本次循环中留在路径 (i, j) 上的信息量。

根据信息素更新策略的不同，Dorigo M 提出了三种不同的基本蚁群算法模型，分别称之为 Ant-Cycle 模型、Ant-Quantity 模型以及 Ant-Density 模型，其差别在于 $\Delta\tau_{ij}^k(t)$ 求法不同。

在 Ant-Cycle 模型中

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & , \text{若第}k\text{只蚂蚁在本次循环中经过}(i, j) \\ 0 & , \text{否则} \end{cases} \quad (5.5)$$

式中, Q 表示信息素强度, 它在一定程度上影响了算法的收敛速度; L_k 表示第 k 只蚂蚁在本次循环中所走路径的总长度。

在 Ant-Quantity 模型中

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{d_{ij}} & , \text{若第}k\text{只蚂蚁在}t\text{和}t+1\text{之间经过}(i, j) \\ 0 & , \text{否则} \end{cases} \quad (5.6)$$

在 Ant-Density 模型中

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q & , \text{若第}k\text{只蚂蚁在}t\text{和}t+1\text{之间经过}(i, j) \\ 0 & , \text{否则} \end{cases} \quad (5.7)$$

它们的区别: 式(5.6)和式(5.7)中利用的是局部信息, 即蚂蚁走完一步后更新路径上的信息素; 而式(5.5)中利用的是整体信息, 即蚂蚁完成了一个循环后更新所有路径上的信息素, 在求解 TSP^[68]时性能较好, 因此通常采用式(5.5)作为蚁群算法的基本模型。此外, 在 Dorigo M 等人的论文中还对蚁群算法提出了一些讨论, 其中包括不同的蚁群初始分布对求解的影响等, 还提出所谓的精英策略(elitist strategy), 以强化精英蚂蚁(发现迄今最好路径的蚂蚁)的影响。结果发现, 对精英蚂蚁数而言有一个最优的范围: 低于此范围, 增加精英蚂蚁数可较早地发现更好的路径, 高于此范围, 精英蚂蚁会在搜索早期迫使寻优过程始终在次优解附近, 导致性能变差。

5.5.3 基本蚁群算法的实现步骤及程序结构流程

以 TSP 为例, 基本蚁群算法的具体实现步骤如下:

(1)参数初始化。令时间 $t=0$ 和循环次数 $N_c=0$, 设置最大循环次数 $N_{c_{\max}}$, 将 m 只蚂蚁置于 n 个元素(城市)上, 令有向图每条边 (i, j) 的初始化信息量 $\tau_{ij}(t) = \text{const}$, 其中 const 表示常数, 且初始化时刻 $\Delta\tau_{ij}(0) = 0$ 。

(2)循环次数 $N_c = N_c + 1$ 。

(3)蚂蚁的禁忌表索引号 $k=1$ 。

(4)蚂蚁数目 $k=k+1$ 。

(5)蚂蚁个体根据状态转换概率公式5.1计算的概率选择元素(城市) j 并前进, $j \in \{C - \text{tabuk}\}$ 。

(6)修改禁忌表指针, 即选择好之后将蚂蚁移动到新的元素(城市), 并把该元素(城市)移动到该蚂蚁个体的禁忌表中。

(7)若集合 C 中的元素(城市)未遍历完, 即 $k < m$, 则跳转到第(4)步, 否则执行第(8)步。

(8)根据公式 5.3 和式 5.4 更新每条路径上的信息量。

(9)若满足结束条件,即如果循环次数 $N_c \geq N_{c_{\max}}$, 则循环结束并输出程序计算结果,否则清空禁忌表并跳转到第(2)步。

以 TSP 为例,基本蚁群算法的程序结构流程如图 5.5 所示。

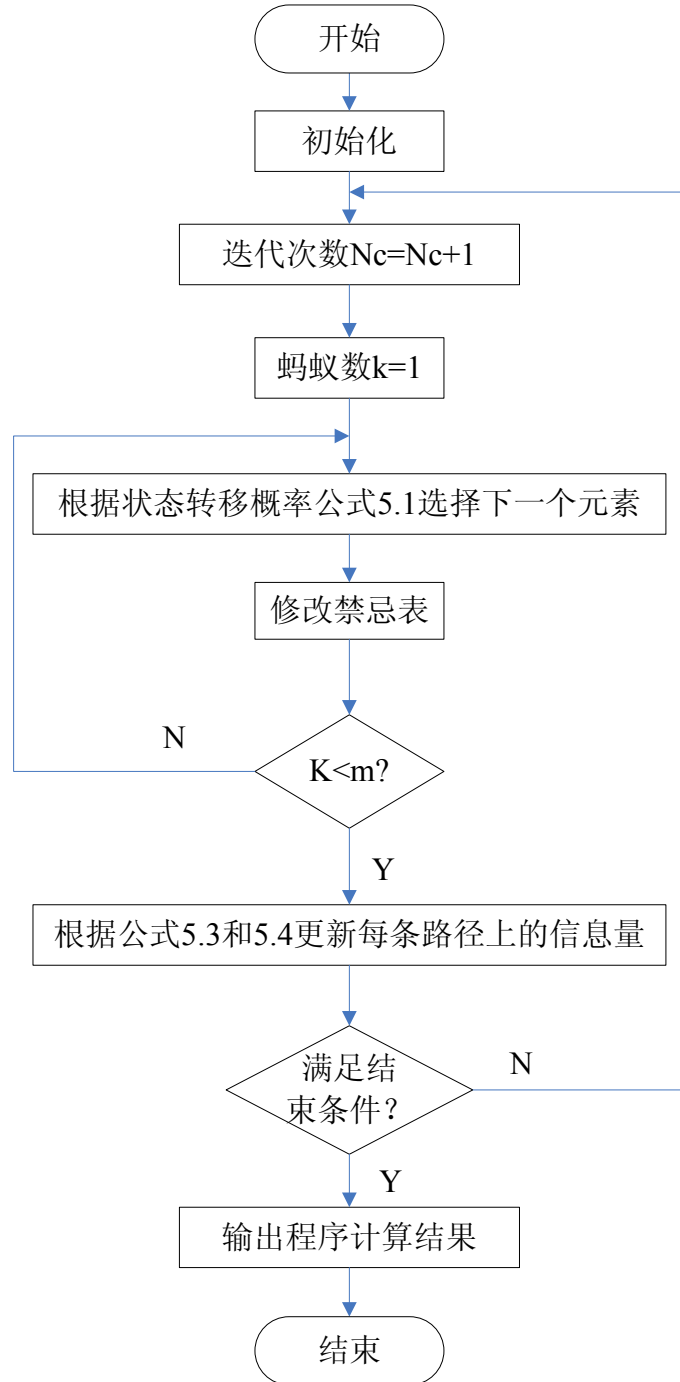


图 5.5 基本蚁群算法的程序结构流程框图

5.5.4 基本蚁群算法的软件实现

以 TSP 为例,基于 Visual Basic 开发环境的基本蚁群算法软件如图 5.6 所示。在图示的软件界面上,用户只要输入要修改的参数即可完成 TSP 的求解,该软件还可以实时显示程

序的当前运行状态，即城市的路径状态、正在运行的迭代周期、到目前所发现最优路径的迭代周期、最优路径上城市排列的序号以及当前最短路径的总长度等等。用户根据需要可在城市状态图和最优路径演化图之间进行切换，并可在程序运行过程中随时停止运行，此外该软件还具有良好、开放的可扩展性，用户在应用中可根据具体问题改写相应的文件，还有平均路径演化图，并在程序中固化了 TSPLIB 中 18 个具有代表性的 TSP，同时还可实时显示当前信息量，并可根据需要进行单步演示。

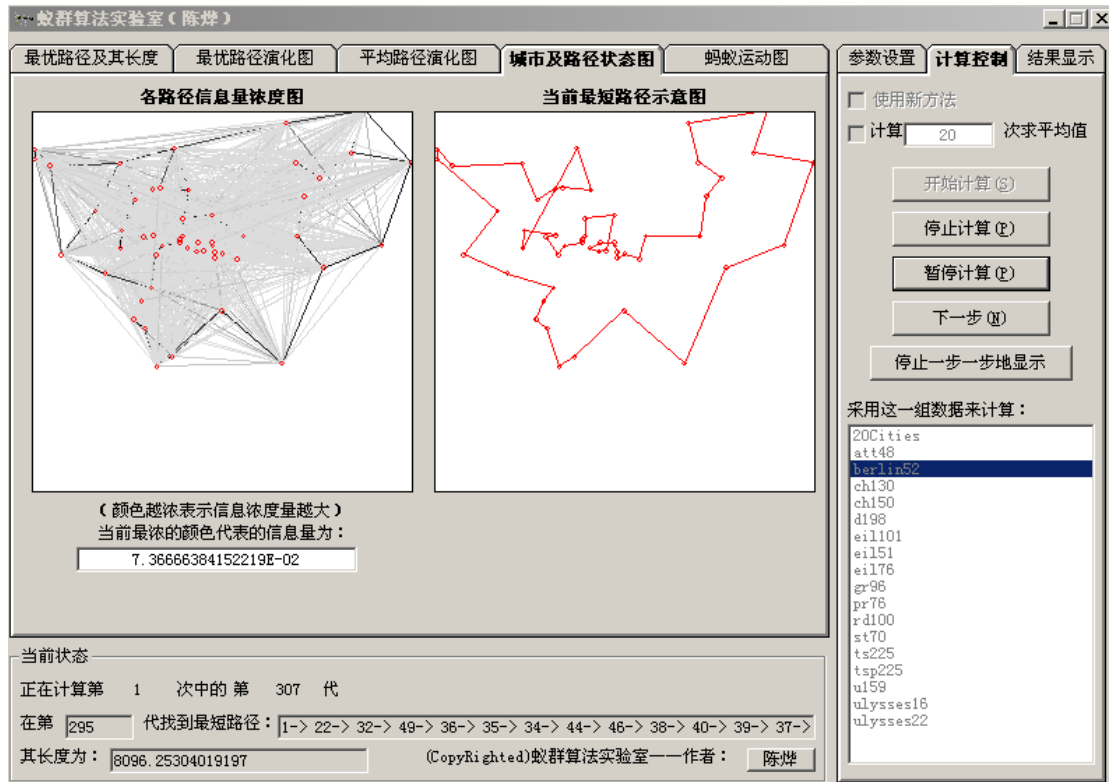


图 5.6 基于 Visual Basic 开发环境的基本蚁群算法软件

5.5.5 参数 α 、 β 、 ρ 组合配置对蚁群算法性能的影响

蚁群算法中各参数的作用是紧密耦合的，其中对算法性能起着最关键作用的应该是信息启发式因子 α ，期望启发式因子 β 和信息素挥发因子 ρ 三个参数。信息素强度 Q 对算法性能的影响则有赖于上述 3 个参数的配置以及对算法模型（Ant-Cycle、Ant-Quantity 和 Ant-Density）的选取， Q 对算法性能的影响情况显然有较大的差异。如果 α 、 β 、 ρ 的组合参数配置不当，会导致求解速度很慢且所得解得质量特别差。因此，我们在研究蚁群算法中主要参数对其性能影响的基础上，研究它的关键参数 α 、 β 、 ρ 的最佳组合配置策略有着非常重要的意义。

我们在这里用 1000 次循环作为算法终止条件，实验仍采用改变一个参数、其他参数不变的策略来探讨参数设置对蚁群算法性能的影响。默认参数设置为 $\alpha=1$ ， $\beta=1$ ， $\rho=0.3$ ， $Q=100$ ，在得到每条备选路径概率的情况下，蚂蚁运用随机选择策略确定下一步要到达的城市。每组数据实验 10 次取平均作比较，实验中所用的 TSPLIB 中的 Oliver30TSP。

Ant-Cycle、Ant-Density 和 Ant-Quantity 模型的实验结果分别如表 5.1、表 5.2 和表 5.3 所示。

表 5.1 Ant-Cycle 模型的实验结果

参数名称	参数值	平均值	最优值	最差值	差值
α	0	682.54	660.22	714.91	54.69
	0.5	538.60	502.82	561.73	58.91
	1	431.05	425.26	436.40	11.14
β	0	872.32	813.04	895.06	82.02
	1	431.05	425.26	436.40	11.14
	5	425.07	423.90	426.53	2.63
ρ	0.1	431.05	428.63	436.01	7.38
	0.3	430.65	424.94	435.57	10.63
	0.5	428.53	424.69	431.31	6.62

表 5.2 Ant-Density 模型的实验结果

参数名称	参数值	平均值	最优值	最差值	差值
α	0	687.45	660.22	714.91	54.69
	0.5	479.44	451.78	503.95	52.17
	2	511.43	489.56	530.70	41.14
β	0	927.06	887.56	957.37	69.81
	1	463.07	443.45	479.78	36.33
	5	428.65	423.90	433.02	9.12
ρ	0.1	433.27	426.60	440.67	14.07
	0.3	463.08	443.45	479.78	36.33
	0.5	537.24	519.83	566.32	46.49

表 5.3 Ant-Quantity 模型的实验结果

参数名称	参数值	平均值	最优值	最差值	差值
α	0	687.62	660.22	714.91	54.69
	0.5	439.19	428.63	452.91	24.28
	1	428.92	423.90	434.93	11.03
β	0	465.07	442.22	478.61	36.39
	1	428.92	423.90	434.23	10.33
	5	427.20	423.90	429.91	6.01
ρ	0.1	426.61	423.90	430.61	6.71
	0.3	428.92	423.90	434.93	11.03
	0.5	455.06	429.91	470.46	40.55

在表 5.1~表 5.3 中，平均值表示将 10 次运行得到的最短路径长的平均值；最优解表示 10 次运行中得到的 10 条最短路径中的最小值；最差解表示 10 次运行中每次得到的最短路径中的最大值；差值表示实验得到的最优解和最差解之间的差值。分析表 5.1~表 5.3 所示的实验结果，可以得到如下结论：

(1)Ant-Cycle 模型中的最佳参数配置为： $\alpha=1$ ， $\beta=5$ ， $\rho=0.5$ ；Ant-Density 模型中的最佳参数配置为： $\alpha=2$ ， $\beta=5$ ， $\rho=0.1$ ；而 Ant-Quantity 模型中的最佳参数配置为： $\alpha=1$ ， $\beta=5$ ， $\rho=0.1$ 。

(2)3 种蚁群算法模型中，若 β 和 ρ 取默认值， $\alpha=1$ 时所得最优解和平均值比 α 其他值时更好，此时其最优解和最差解之差也最小。这说明解的质量和稳定性都是最好的，所以 3 种模型中 α 值的最佳设置应为 1。若 α 和 ρ 取默认值时， β 的最佳设置值应为 5。对于 ρ ，它的最佳设置值应为 0.1，此时解的质量最优。

以上分析是在给定 1000 次迭代的条件下得出的。这里还增加了不是最佳参数配置时算法模型的最大运行次数实验，结果表明，对于不是最佳参数配置的算法模型，即使再增加 1000 次或 2000 次运行，其解的质量与 1000 次时相比没有明显改善；而采用最佳参数配置的模型，可以很快地搜索到质量高的全局优化解^[66]。

5.5.6 改进蚁群算法解迷宫问题

1 迷宫最优路径问题

迷宫机器人标准场地是边长为 16 的正方形区域，对位于点 (i, j) ($0 \leq i \leq 15; 0 \leq j \leq 15$) 的蚂蚁，它每次只能按图 5.7 所示的方向移动一步。那么迷宫的最优路径问题即为：从正方形迷宫区域上的某一点 (i_s, j_s) 出发，沿可行路径到达某一终点 (i_t, j_t) ，使其经过的路径长度最短。

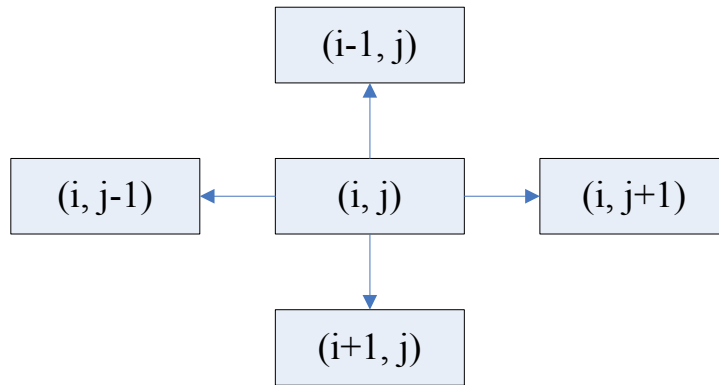


图 5.7 位于点 (i, j) 的蚂蚁可移动方位图

2 求解迷宫问题的蚁群算法

(1) 蚂蚁在点 (i, j) 的下一步可行解点集。

因位于点 (i, j) 的蚂蚁只能沿着图 5.7 所示的 4 个方向（前，后，左，右）之一前进一步，假设可以到达的下一点的坐标为 (g, h) ，则

$$\begin{cases} g = i + di[v] \\ h = j + dj[v] \end{cases} \quad v = 0, 1, 2, 3$$

其中 di, dj 两个一维数组，其值为：

$$di = \{-1, 0, 1, 0\}$$

$$dj = \{0, 1, 0, -1\}$$

为了进一步说明这个数组。我们用数组表示其下标的变化（只记录其增量）如表 5.4 所示。

表 5.4 数组下标变化示意

	0 (上)	1 (右)	2 (下)	3 (左)
行下标变化 Δi	-1	0	1	0
列下标变化 Δj	0	1	0	-1

那么点 (i, j) 下一步的可行点集 $allowed(i, j)$ 可由过程 1 完成。

过程 1: procedure GetAllowedPoints(i, j)

for $p=0:7$ begin

if $(i+di[i] \geq 0) \text{ and } (i+di[i] \leq m-1)$

and $(j+dj[i] \geq 0) \text{ and } (j+dj[i] \leq n-1)$

then 将点 $(i+di[i], j+dj[i])$ 加入到列表 $allowed(i, j)$ 中;

End

(2) 蚂蚁选择路径的规则

考虑到从点 (i, j) 只能到达 $allowed(i, j)$ 中的点集，而 $allowed(i, j)$ 中的元素最多只有 4 个，故定义如下的数据结构来存储与该位置相邻位置间的信息素：

#define M 16 //迷宫的行数

```
#define N 16 //迷宫的列数
```

```
typedef struct
```

```
{ int i, j;
```

```
double pheromone[2][2];
```

```
} Point pheromone;
```

//存储从点 (i, j) 到可达点间的信息素

```
Point pheromone Maze pheromone [M][N]; //存储迷宫中所有点及其相关的信息素。
```

则对位于 (i, j) 处的蚂蚁 k ，按公式(5.8)选择下一个可行点 (u, v) 。

$$P^k_{ij \rightarrow uv} = \begin{cases} \frac{\tau^{\alpha}_{ij \rightarrow uv}}{\sum_{(x,y) \in allowed_k} \tau^{\alpha}_{ij \rightarrow xy}} & \text{if } (u,v) \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (5.8)$$

其中， $allowed_k$ 为蚂蚁 k 当前可到达的点集，其元素为 $allowed(i, j)$ 中的点除去蚂蚁 k 已经经过的点； $\tau^{\alpha}_{ij \rightarrow uv}$ 表示点 (i, j) 与点 (u, v) 之间的信息素，其初始值为一较小的正常数，后随算法的运行将会逐渐改变。 α 表示信息素的重要程度，且 $\alpha > 0$ 。

(3) 迷宫可行解的构造

在利用蚁群算法求解迷宫最短路径问题时，为了使每只蚂蚁能以尽可能高的概率生成可行解，本文采用两组数量相等的蚁群分别从迷宫的起点和终点同时出发，每只蚂蚁按公式(5.8)确定的概率在迷宫中漫游搜索。为尽量避免生成无效路径，为蚂蚁 k 分配一张禁忌表 $tabu(k)$ 。该表记录蚂蚁 k 当前走过的点集，以避免选择已经走过的点。则对任意一只蚂蚁，在移动过程中可以定义如下的生命周期：(i)蚂蚁走进死角，除非沿原路返回一步或多步，不能再朝前移动，则将该蚂蚁从系统中删除；(ii)蚂蚁到达另一组蚁群的出发点，此时该蚂蚁走过的路径为一条可行路径；(iii)蚂蚁碰到另一组的某只蚂蚁。如果这两只蚂蚁所经过的点没有重复(相遇点除外)，则将两只蚂蚁所经过的路径相连以构成迷宫的一条可行路径。因此，从蚁群的产生到生命周期的结束，会有一部分蚂蚁找到问题的可行解，但可行解的数量小于蚁群数的一半。

(4) 信息素更新规则

在每次迭代中，针对当前最好解所属的边按下面的公式进行信息素更新，

$$\tau_{ij \rightarrow uv}(t+1) = (1-\rho) * \tau_{ij \rightarrow uv}(t) + \Delta \tau^{ib}_{ij \rightarrow uv}(t) \quad (5.9)$$

其中， $\rho \in (0,1)$

$$\Delta \tau^{ib}_{ij \rightarrow uv} = \begin{cases} \frac{1}{L^{ib}} & \text{if } ij \rightarrow uv \text{ 在本次迭代最好路径流上} \\ 0 & \text{否则} \end{cases} \quad (5.10)$$

其中 L^{ib} 为本次迭代最好解的长度， ρ 为信息素的蒸发系数。

(5)改进蚁群算法求解迷宫最短路径的具体实现步骤如下：

Step1. 初始化参数 α ， ρ ， $N_{C_{\max}}$ ， m ；

Step2. 生成 $2*m$ 只蚂蚁，将其中 m 只放到起点以 (i_s, j_s) ，另外 m 只放到终点 (i_t, j_t) ；

Step3. while 活蚂蚁数大于 0 do

 for 蚂蚁 $k = 1, 2, \dots, 2m$ do

 if 蚂蚁 k 活着 and $allowed_k$ 非空 then

 (1)按公式计算转移概率 P_k ；

 (2)按概率 P_k 选择下一个点；

 (3)根据蚂蚁生命周期(i)，(ii)，(iii)情况，决定蚂蚁 k 的死、活状态；

 (4)如果蚂蚁 k 没死亡，将其所选择的点加入 $tabu(k)$ 中。

 end if

Step4. 计算本次迭代的最好解，如优于当前最好解，则用其替换当前最好解；

Step5. 按公式(5.9)更新路径上的信息素；

Step6. if N_c 小于 $N_{c_{max}}$ 且未进入停滞状态 then

 (1)清空所有蚂蚁禁忌表中的数据；

 (2) $N_c := N_c + 1$ ；

 (3)转至 Step2

else 输出最优结果

end if

End

5.6 本章小结

本章是迷宫电脑鼠软件设计的数学算法部分，是论文中最具鲜明特色的部分，是全文的核心章节。

本章首先介绍了迷宫场地的设计和迷宫的表示，把实际迷宫抽象化数学化成一个二维数组。然后又简单介绍迷宫机器人常用搜寻法则和策略。在求解迷宫问题的算法中，详细阐述基本蚁群算法及其应用；并通过学习和研究，改进蚁群算法来求解迷宫最优路径问题。本章是全文工作目标能否实现的关键。

第六章 迷宫机器人的软件设计与调试

机器人的核心是控制系统。机器人的先进性和功能的强弱通常都直接与其控制系统的性能有关。机器人的控制是一项跨多学科的综合技术，它涉及自动控制、计算机、传感器、人工智能、电子技术和机械工程等多学科的内容^{[35][36]}。我已经在前面的章节介绍了迷宫机器人的结构和硬件，本章要介绍的是迷宫机器人的软件设计和程序实现。

6.1 迷宫机器人的控制系统

6.1.1 CPU 引脚资源分配

我们选用的微处理器 LPC2138 是 PHILIPS 公司生产的 32 位处理器。它用于控制和检测红外传感器；根据检测到的传感信号，控制电机驱动电路并调整行走路径，直到迷宫机器人到达终点。

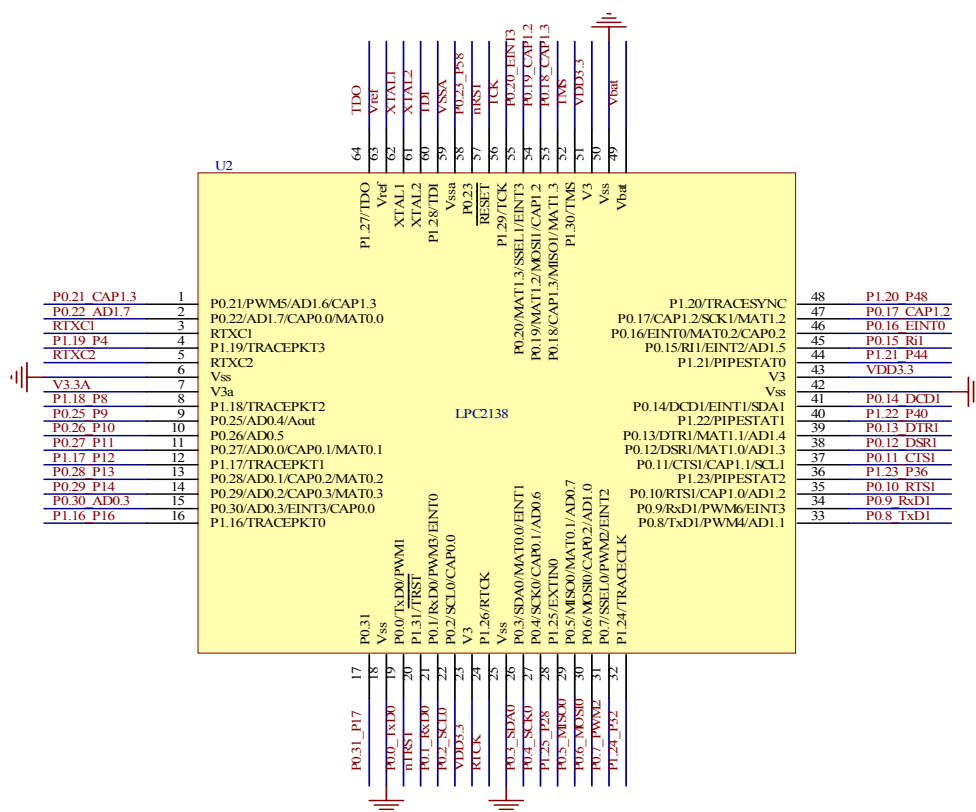


图 6.1 LPC2138 的引脚排列

迷宫机器人控制系统中的输入信号：

- (1) 两个独立按键信号：KEY2、KEY3。
- (2) 复位信号：nRST。
- (3) 石英晶振输入引脚：OSC0、OSC1。
- (4) 3 组电源输入引脚：DP3V3（有 3 组）、GND（有 3 组）。

(5) 车速检测电路的红外发射接收传感器信号：LEFT_PULSE、RIGHT_PULSE。

(6) 迷宫挡板左侧、前方、右侧三方向检测电路红外传感器信号：INFRAREDOUT1、INFRAREDOUT2、INFRAREDOUT3。

输出信号：

(1) 五个控制 LED 指示灯的信号：D7，D5，D4，D2，D1。

(2) 四个控制电机的信号：LWC1、LWC2、RWC1、RWC2。

(3) 38KHz 红外信号：SEND、INFRAREDIN。

(4) JTAG 接口电路：JTAG_nTRST、JTAG_TCK、JTAG_TMS、JTAG_TDI、JTAG_TDO。

本系统部分跳线器选择一览表如表 6.1 所示：

表 6.1 LPC2138 部分硬件资源分配一览表

I/O	功能说明	备注
P0.16	KEY2	独立按键
P0.17	KEY3	独立按键
P0.2	LEFT_PULSE	左轮车速检测电路的红外发射接收信号
P0.10	RIGHT_PULSE	右轮车速检测电路的红外发射接收信号
P0.27	INFRAREDOUT1	迷宫挡板左侧检测电路红外传感器信号
P0.28	INFRAREDOUT2	迷宫挡板前方检测电路红外传感器信号
P0.29	INFRAREDOUT3	迷宫挡板右侧检测电路红外传感器信号
P1[25:18]	D7、D5、D4、D2、D1 等	LED 指示灯信号
P0.18	SEND	发射红外信号
P0.21	INFRAREDIN	接收红外信号
P0.19	LWC2	左轮电机转动方向信号
P0.7	LWC1	左轮电机 PWM 信号
P0.9	RWC1	右轮电机 PWM 信号
P0.20	RWC2	右轮电机转动方向信号
P1[31:27]	nTRST、TCK、TMS、TDI、TDO	JTAG 接口电路

6.1.2 初始化模块

初始化模块完成 LPC2138 各端口定义、变量定义以及赋初值、定时器初始化、中断初始化等内容。

1 系统初始化模块

系统初始化模块的任务是完成 LPC2138 各端口的定义及赋初值、中断初始化。该模块程序如下：

```
void initial_sys( )  
{
```

```

SysCtlClockSet( SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC | SYSCTL_OSC_MAIN |
                SYSCTL_XTAL_11MHZ );           // 设定晶振为时钟源

SysCtlPeripheralEnable( SYSCTL_PERIPH_GPIO );   // 使能 GPIO 口外设

GPIODirModeSet(GPIO_BASE, KEY2, GPIO_DIR_MODE_IN);
                                                // 设定按键为输入
}

```

2 定时器 0 初始化模块

定时器 0 初始化模块的任务是完成与定时器 0 相关的设置。在这里定时器配置为：PWM 初始化有效信号为低电平，设定 PWM 频率为 11.0592MHz。该模块程序如下：

```

void PWMTimer0AIni(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);           // 使能定时器 0 模块
    GPIODirModeSet(GPIO_BASE, LWC1|RWC1 , GPIO_DIR_MODE_OUT);

    /* 控制引脚输出*/
    GPIOPinWrite(GPIO_BASE, LWC1|RWC1 , 0xff);              // GPIO 输出高电平
    GPIODirModeSet(GPIO_BASE, LWC2|RWC2, GPIO_DIR_MODE_OUT);
                                                            // 控制引脚输出
    GPIOPinWrite(GPIO_BASE, LWC2|RWC2 , 0xff);              // GPIO 输出高电平

    /* 定时器配置*/
    TimerConfigure(TIMER0_BASE,TIMER_CFG_32_BIT_PAIR|TIMER_CFG_PWM);
                                                            // 32 位 PWM 输出
    TimerControlLevel(TIMER0_BASE,TIMER0,false);           // PWM 初始化有效信号为低电平
    TimerLoadSet(TIMER0_BASE,TIMER0,110592);               // 设定 PWM 频率
}

```

3 定时器 1 初始化模块

定时器 1 初始化模块的任务是完成与定时器 0 相关的设置。在这里定时器 1 设置为：定时器 1 产生 38KHz 的调制信号，由 P0.18 输出。该端口连接到 LPC2138 的 INFRAREDIN 端口。该模块程序如下：

```

void PULSEIni(void)
{
    GPIODirModeSet(GPIO_BASE, SEND | PULSE, GPIO_DIR_MODE_OUT);
                                                            // 设置 GPIO 为输出口
    GPIOPinWrite( GPIO_BASE,SEND | PULSE,0);              // 红外线初始时停止发射
    SysCtlPeripheralEnable( SYSCTL_PERIPH_TIMER1 );        // 使能定时器 1 外设
    TimerConfigure(TIMER1_BASE, TIMER_CFG_32_BIT_PER);
                                                            // 设置定时器 1 为周期触发模式

    TimerLoadSet(TIMER1_BASE, TIMER_, SysCtlClockGet()/76000);
}

```

```

// 设置定时器装载值
TimerIntEnable(TIMER1_BASE, TIMER_TIMEOUT);
TimerEnable(TIMER1_BASE, TIMER1);
IntEnable(INT_TIMER1);
}
4 定时器 1 中断服务函数
    本设计中定时器 1 产生 38KHz 的信号, 由 P0.18 输出。在中断中翻转 P0.18 输出信号,
    所以要产生频率为 f 的脉冲, 那么定时器中断频率就为 76KHz。该模块程序如下:
void Timer1_ISR(void)
{
    TimerIntClear(TIMER1_BASE, TIMER_TIMEOUT); // 清除定时器 1 中断
    GPIOPinWrite(GPIO_BASE, PULSE, GPIOPinRead(GPIO_BASE, PULSE) ^ PULSE);
    // 翻转 GPIO P0.18 端口
}

```

6.1.3 运动模块

迷宫机器人的运动是通过对其驱动电机的转速进行控制来实现。其运动模块包含了很多子模块。每个子模块执行相应的动作。

- (1) 前进模块: 控制竞赛机器人沿着直线前进。
- (2) 原地右转模块: 控制机器人左轮正转或右轮反转来实现。
- (3) 原地左转模块: 控制机器人右轮正转或左轮反转来实现。
- (4) 左转 90° 模块: 控制机器人按逆时针方向原地左转 90°。
- (5) 右转 90° 模块: 控制机器人按顺时针方向原地左转 90°。
- (6) 加速模块: 传感器检测到前方无障碍时, 机器人加速前进。
- (7) 减速模块: 传感器检测到前方有障碍时, 机器人减速前进。
- (8) 左右轮同转速模块: 保持左右两个电机的转速相同。
- (9) 停车模块: 将左右轮驱动电压置零, 控制机器人准确停车。

(10) 行走模块: 在机器人沿跑道中线前进时, 需要不断根据传感器信号调整行走方向, 使得机器人不会撞到墙上, 这时的左转右转只是在不断地调整机器人的位置。左转功能模块具体由左轮转速小于右轮转速来实现; 右转功能模块具体由右轮转速小于左轮转速来实现。机器人在前进过程中, 要根据其左右侧是否有墙壁和是否太靠近墙壁的情况来执行相应的动作。

以上机器人的各种动作, 我们都是通过控制机器人左右两直流电机的转速来实现的。直流电机的转速控制在本设计中通过 PWM 来控制, 我们用两路 PWM 输出, 非常适合控制两个电机的转速。

两路 PWM 模式用上了两个 32 位的定时器 Timer0 和 Timer1。它们为计数寄存器 (GPTMTnR) 递减计数, 递减到 0 时自动加载预装载值 (GPTMTnILR)。当然预装载值是由用

户设定的，该值决定了定时周期，也即 PWM 的输出周期。

当计数器的值与预装载值相等时，输出 PWM 信号有效，当计数器的值与匹配寄存器（GPTMnMATCHR）的值相等时，输出 PWM 信号失效。通过软件可以设定 PWM 信号有效和信号无效的电平状态。当 GPTMCTL 寄存器的 TnPWML 位值为 0 时，信号有效为高电平，信号无效为低电平；TnPWML 位值为 1 时，则反之。

我们对占空比的约定：占空比为在一个周期内，输出信号有效电平占整个周期时间的比率。在这里为统一软件控制的约定。用户 API 函数输入的占空比值越大，电机转速越快，正向运行和反向运行都一样。

为了简化占空比输出的计算，将计数寄存器与匹配寄存器值相等时，输出的电平信号作为驱动电机的有效信号。例如将 PWM 周期时间设定为 600000 个时钟节拍，需要输出驱动电机的占空比为 75%，则设置匹配寄存器值为 75×6000 。由于电机的转向不一样，所以电机驱动的有效电平也需要调整，通过控制 TnPWML 来实现。

下面我们介绍下迷宫机器人电机调速的程序设计。

左轮的控制函数程序如下。而右轮的控制函数程序与之类似。

该函数的第 1 个参数 sel 为选择轮子的控制方式：0 为停止，1 为轮子向前，2 为轮子向后；percen 参数为占空比，其最大值为 99，最小值为 1，对于轮子的停止控制该参数无效。

```
void LeftWheelRun(int sel,unsigned char percen)
```

```
{
    switch(sel)
    {
        /*轮子停止转动*/
        case 0:
            TimerDisable(TIMER0_BASE,TIMER0);           // 禁止定时器
            GPIOPinWrite(GPIO_BASE,LWC2,0xff);          // 控制引脚输出高电平
            GPIODirModeSet(GPIO_BASE, LWC1, GPIO_DIR_MODE_OUT);
                                                         // 禁止 PWM 引脚输出，改为 GPIO 输出
            GPIOPinWrite(GPIO_BASE,LWC1, 0xff);          // GPIO 输出高电平
            break;
        /*左轮向后*/
        case 1:
            GPIOPinWrite(GPIO_BASE,LWC2, 0xff);          // LWC2 输出高电平
            TimerControlLevel(TIMER0_BASE,TIMER0,false); // PWM 有效电平方向
            GPIODirModeSet(GPIO_BASE, LWC1, GPIO_DIR_MODE_HW);
                                                         // 禁止 GPIO 引脚输出，改为 PWM 输出
            TimerMatchSet(TIMER0_BASE,TIMER0,percen*1100); // 设置占空比
            TimerEnable(TIMER0_BASE,TIMER0);              // 使能定时器
            break;
        /*左轮向前*/
        case 2:
            GPIOPinWrite(GPIO_BASE,LWC2, 0);              // LWC2 输出低电平
```

```

    TimerControlLevel(TIMERO_BASE,TIMERO,true);        // PWM 有效电平方向
    GPIODirModeSet(GPIO_BASE, LWC1, GPIO_DIR_MODE_HW);
                                   // 禁止 GPIO 引脚输出, 改为 PWM 输出
    TimerMatchSet(TIMERO_BASE,TIMERO,percen*1100); // 设置占空比
    TimerEnable(TIMERO_BASE,TIMERO);                // 使能定时器
    break;
}
}

```

我们再用左轮检测脉冲中断处理函数来进行车速检测和控制电机的停止。具体做法是：先读取 P0 口中断状态，判断是否为左轮脉冲中断；计数器进行脉冲计数，当脉冲计数值大于或等于设定值时，停止电机转动。右轮检测脉冲中断处理函数与之类似。程序如下：

```

void GPIO_ISR(void)
{
    unsigned char IntStatus;
    IntStatus = GPIOPinIntStatus(GPIO_BASE,true);        // 读 P0 口中断状态
    if(IntStatus&PULSE_L)                                // 是否为左轮脉冲中断
    {
        PulCount_L++;                                    // 脉冲计数
        if(PulCount_L>= LeftPulse)                        // 判断是否达到设定值
        {
            WheelStop_L= 1;                               // 置电机停止标志
            LeftWheelRun(0, 1);                            // 停止电机
        }
        GPIOPinIntClear(GPIO_BASE,PULSE_L);              // 清中断
    }
}

```

6.1.4 迷宫挡板检测

迷宫电脑鼠在迷宫中是靠侦测路面情况前进的。它的左右传感器不但要检测是否存在支路(没有挡板就是一条支路)还要避免和挡板碰触。因此电脑鼠每一侧在正常情况就需要两组红外传感器，一组检测稍微远一点的距离，判断是否存在支路；另一组检测稍微近一点的距离，判断是否即将碰触挡板。

由此看出，电脑鼠每侧都需要两组传感器，但是如果只用一组传感器来完成两个参数的检测是否可行呢？我们知道，如果使用非调制的普通红外接收头，就可以根据接收到的信号的强弱来计算距离，可是非调制的抗干扰行差。如果使用调制的一体化接收头，检测信号输出的是数字信号，这样通过检测传感器输出信号的强弱来计算距离的方法肯定行不通。那么，还有没有其他方法可以实现距离的测量呢？答案是肯定的。

我们可以把上述的测距原理反过来，通过改变发射出的接收传感器能够识别的信号强度，在接收头刚好能接收到信号时，记录下此时发射的强度，这样也就可以大致测算出距

离。

改变输出接收传感器所能识别信号的强度的方法有两种：

1 改变输出信号的能量。改变输出信号的能量又有两种方法：

- (a) 改变输出信号的幅度。改变 Send 端的输出电压或调节可调电阻 R1 的阻值就可以实现。
- (b) 改变输出信号的占空比。改变输出的 38KHz 信号的占空比就可以实现。

2 改变输出信号的频率。由于一体化接收头是 38KHz 的带通滤波器，所以发射信号的频率偏离 38KHz 越多，能检测到的有效信号就越少。这样也就可以改变有效发射信号的强度。

如果采用改变输出信号的幅度的方式，那么还需要一个 D/A 转换器或手动调节 R1。这样不仅增加了成本，还不适用于自动控制。如果通过改变占空比的方式就需要使用 PWM 功能，但 PWM 功能已经被驱动小车的电机占用，所以只能通过改变输出信号的频率来实现测距，这样只需要一个定时器就可以完成此功能。

在 Micromouse 迷宫电脑鼠中，我们用到了三组（左侧、前方、右侧）反射式红外检测传感器。左边和右边的传感器各自都需要检测两组参数，而前方的传感器只需要探测有无挡板，存在挡板就必须根据策略换行进方向，若不存在就可以继续前进。迷宫挡板检测程序流程框图如图 6.2 所示。

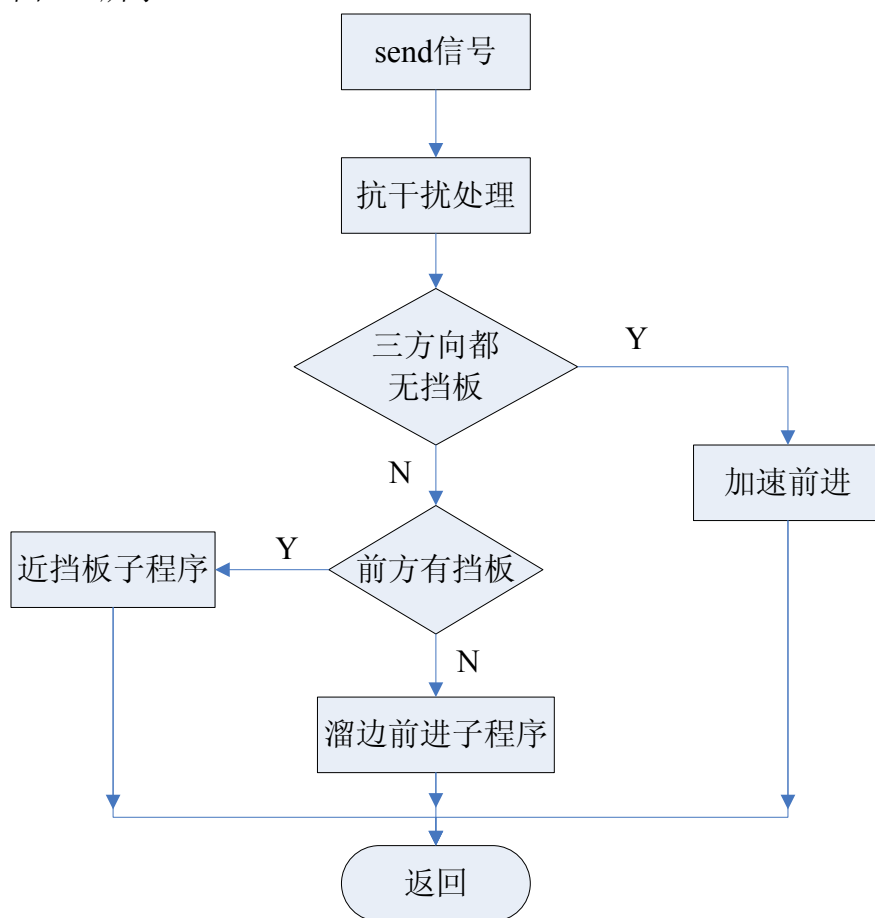


图 6.2 迷宫挡板检测程序流程框图

在迷宫挡板检测程序流程图内，还包含了多个子程序。下面我们对其子程序进行分析。

(1) 抗干扰处理程序

红外线在空气中的传播和反射都受到外界的干扰，如果测量距离刚好处在能够检测到信号的临界状态，保持距离不变，传感器输出信号也可能不确定。这样就需要在软件中进行抗干扰处理。参考程序流程框图如下：

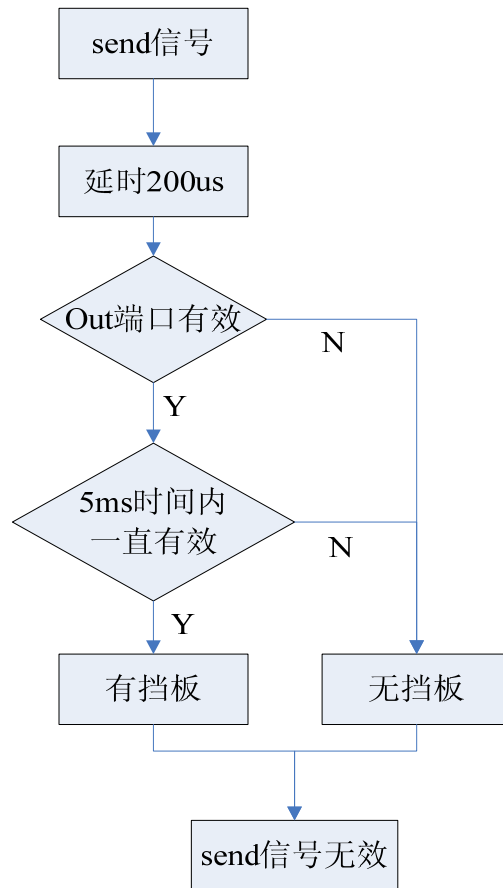


图 6.3 抗干扰检测有无挡板程序流程图

(2) 溜边前进程序

溜边前进程序，我们将在关于迷宫智能算法的 6.2 节中详细叙述，这里先略过。

(3) 近挡板检测和防碰撞程序

当迷宫电脑鼠只需测车体是否太靠近挡板的近距时，程序直接跳到装载定时器为 35KHz（略微偏离 38KHz 的频率均可）的程序处。即迷宫电脑鼠在太靠近迷宫挡板时，调整电脑鼠车体行进的程序。其程序流程图如图 6.4 所示。

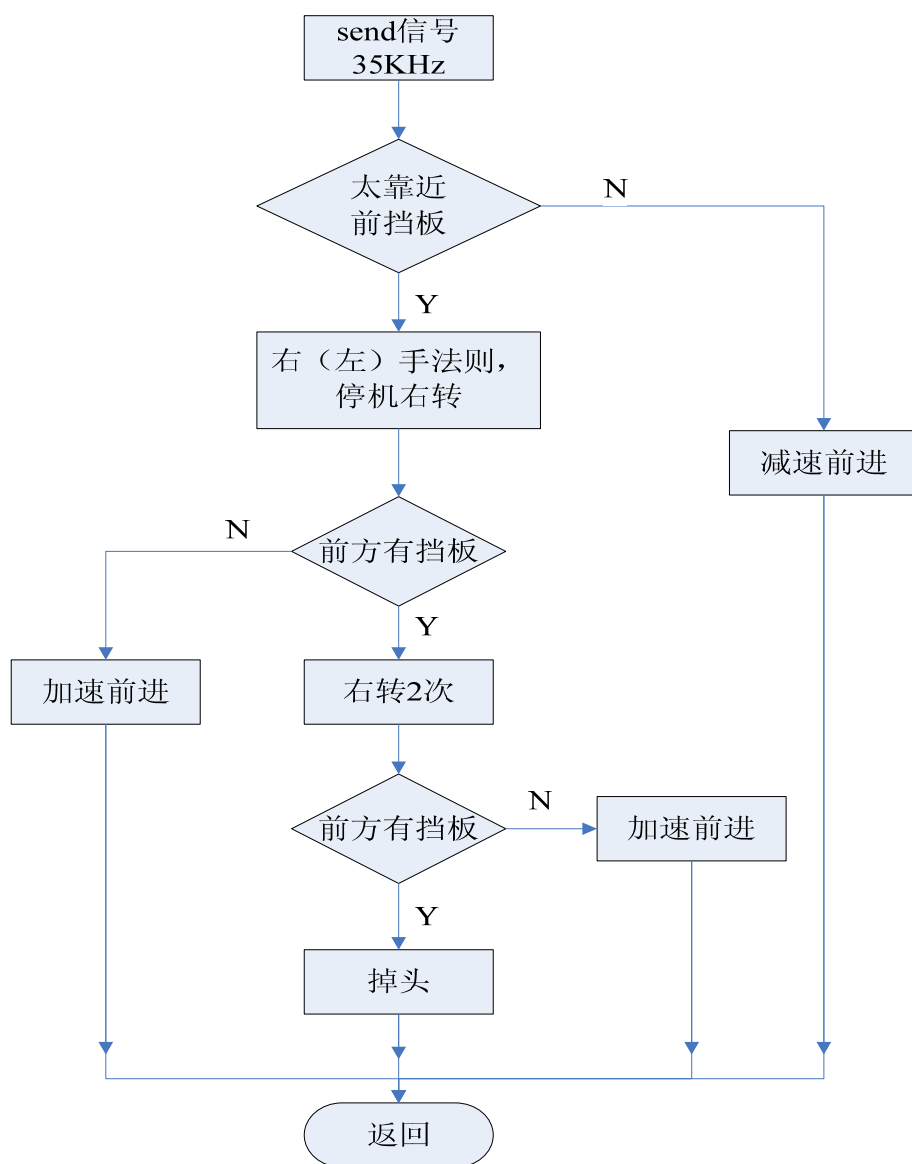


图 6.4 近挡板检测和防碰撞程序

6.2 迷宫智能算法的实现

迷宫智能算法模块^[72]的主要功能是：

- 1) 优化设置模块（如迷宫电脑鼠的最佳加减速速度，控制策略等）；
- 2) 解决迷宫问题算法的实现；
- 3) 经历路径的记忆；
- 4) 迷宫电脑鼠速度的控制（加速、减速的平滑，转弯等）；
- 5) 迷宫策略的优化。

6.2.1 沿跑道中线前进的判断程序

迷宫机器人的场地跑道较窄，而且曲折变化，机器人经常要转 90° 、 180° 等，这就需要迷宫电脑鼠沿着跑道中线前进，否则就会撞到迷宫挡板。

迷宫场地中每个小格的标准长度是 180mm，为了可以方便地搬运、组装迷宫场地。这里采用了标准的木地板组合，木地板的标准宽度是 192mm，迷宫的隔栅宽度是 20mm，故迷宫的跑道宽度是 172mm，与木地板标准宽度几乎一样。若迷宫电脑鼠的宽度是 100mm，故迷宫电脑鼠在跑道中线时，左右边距离各是 36mm。

迷宫电脑鼠的红外传感器输出电压 V_{out} 与检测距离 x 之间有如下关系：当检测距离 $x < 3\text{cm}$ 时，检测电压 $V_{out} > 2.4\text{V}$ 说明电脑鼠偏离中线，向左偏离了。当检测距离 $x > 4\text{cm}$ 时，检测电压 $V_{out} < 1.75\text{V}$ ，说明电脑鼠偏离中线，向右偏了。

我们首先实现电脑鼠溜边前进的程序。当电脑鼠左侧的传感器测得距离 $L_1 < 3\text{cm}$ 时，电脑鼠右拐；当电脑鼠左侧的传感器测得距离 $L_1 > 4\text{cm}$ 时，电脑鼠左拐。路径 S 时电脑鼠的实际前进路线，整体来看，相当于电脑鼠沿着跑道中线前进。溜边前进的程序框图如图 6.5 所示。

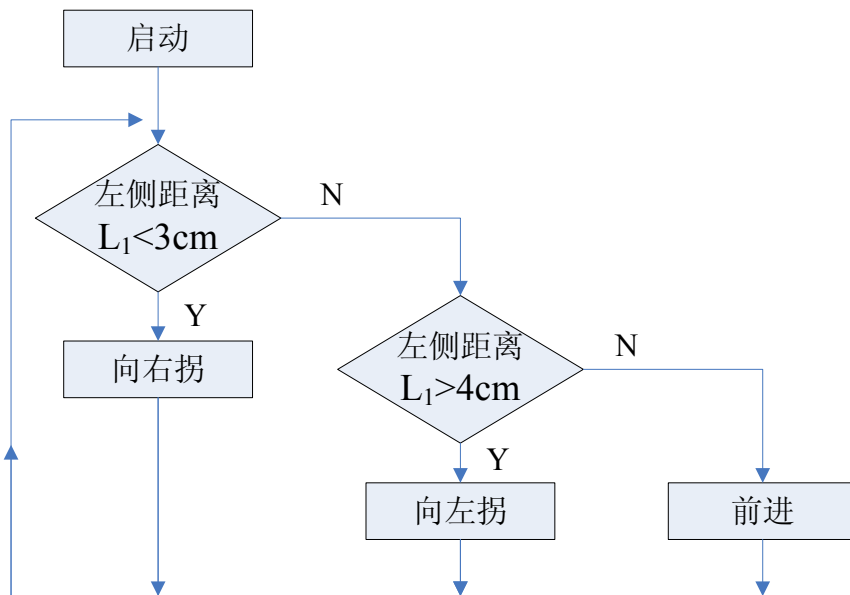


图 6.5 机器人溜边前进程序框图

6.2.2 有记忆功能的迷宫算法分析

我们只有让迷宫机器人有了记忆功能，它才能表现出智能。

(1) 迷宫的状况

迷宫机器人有三个方向的传感器：前方传感器 F、左侧传感器 L、右侧传感器 R。如果前方有挡板，则传感器发送来的信号为 1，无挡板则为 0。迷宫电脑鼠在迷宫中运动时，三个传感器 F、L、R 会传来迷宫的 8 中状况，即死胡同 111、右转弯 110、左转弯 101、直丁字路口 100、直路 011、右丁字路口 010、左丁字路口 001、十字路口 000。如图 6.6 所示。

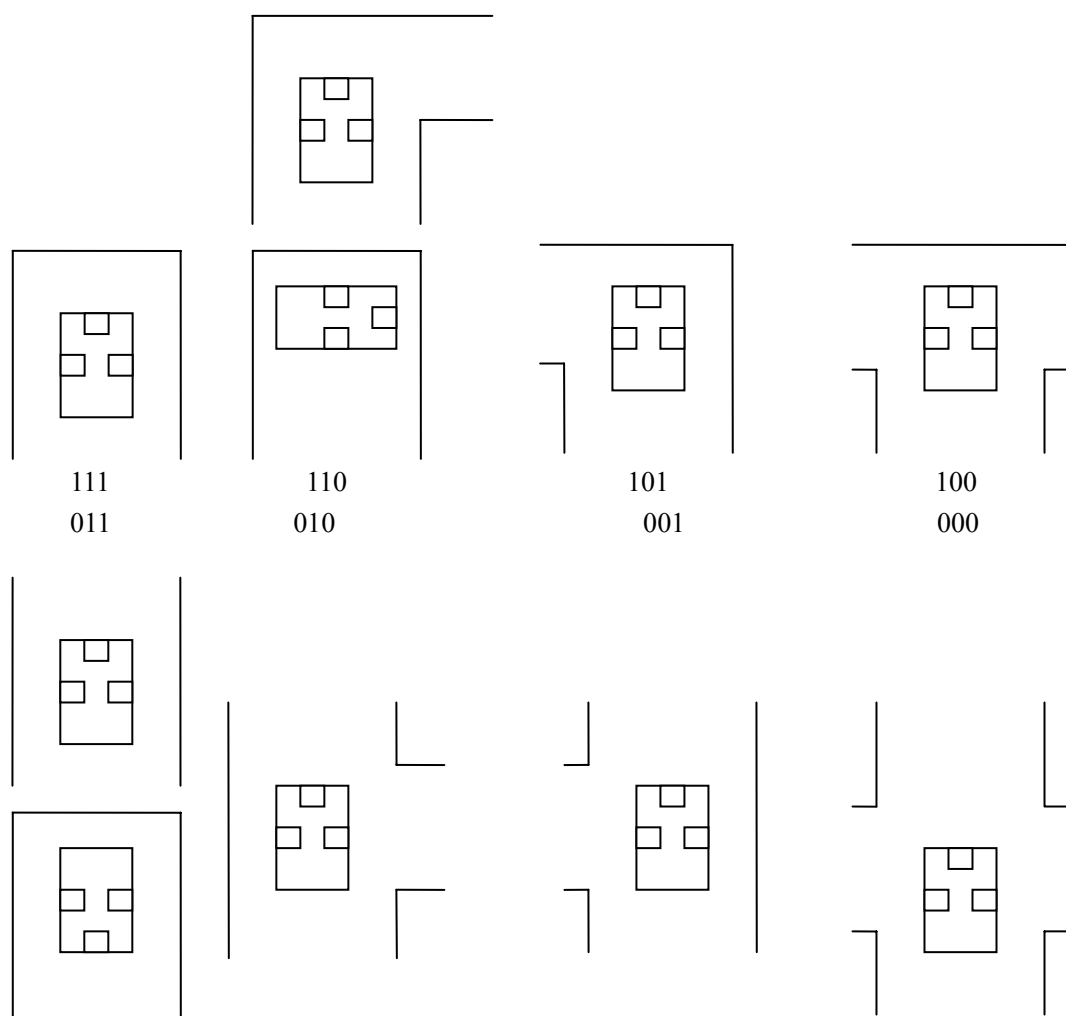


图 6.6 迷宫机器人在迷宫中的 8 种状态

(2) 记忆迷宫的状态寄存器

我们设可位操作的路况寄存器 `road_flag` 用来记忆迷宫的路况, 每遇到变化的分叉路况, 置 `road_flag.0=1` 或 `0`, 且左移一位, 把路况保存起来。“1”表示可前进, “0”表示可右转 (对于十字路口, 右转优先)。设死胡同寄存器 `dead_flag` 用来记忆遇到的死胡同。每遇到一个死胡同, 置 `dead_flag.0=1`, 并且左移一位。然后再设转向寄存器 `turn_flag` 用来记忆迷宫电脑鼠的向左转或向右转的操作状态。

(3) 迷宫问题的分析

一般的迷宫有上面的 8 种状况, 当迷宫电脑鼠遇到这 8 种状况时, 只有做出正确的选择才能到达迷宫终点。下面我们来具体研究下电脑鼠遇到这 8 种状况时应采取的措施。

死胡同 111 状况。当电脑鼠遇到死胡同 111 状况时, 应采取的处理方法是: 原地掉头 180° , 同时置死胡同标志, 可由 `state_111()` 函数来实现。

右转弯 110 状况。当电脑鼠遇到右转弯 110 状况时, 应采取的处理方法是: 原地向右转 90° , 可由 `state_110()` 函数来实现。

左转弯 101 状况。当电脑鼠遇到左转弯 101 状况时, 应采取的处理方法是: 原地向左

转 90°，可由 state_101()函数来实现。

直路 011 状态。当电脑鼠处于直路 011 状态时，应采取的处理方法是：继续保持前进，可由 state_011()函数来实现。

十字路口 000 状态。当电脑鼠遇到十字路口 000 状态时，应采取的处理方法是：向右转 90°前进，可由 state_000()函数来实现。

直丁字路口 100 状态。当电脑鼠遇到直丁字路口 100 状态时，应采取的处理方法的程序框图如图 6.7 所示，由可由 state_100()函数来实现。

右丁字路口 010 状态。当电脑鼠遇到右丁字路口 010 状态时，应采取的处理方法的程序框图如图 6.8 所示，由可由 state_010()函数来实现。

左丁字路口 001 状态。当电脑鼠遇到直丁字路口 001 状态时，应采取的处理方法的程序框图如图 6.9 所示，由可由 state_001()函数来实现。

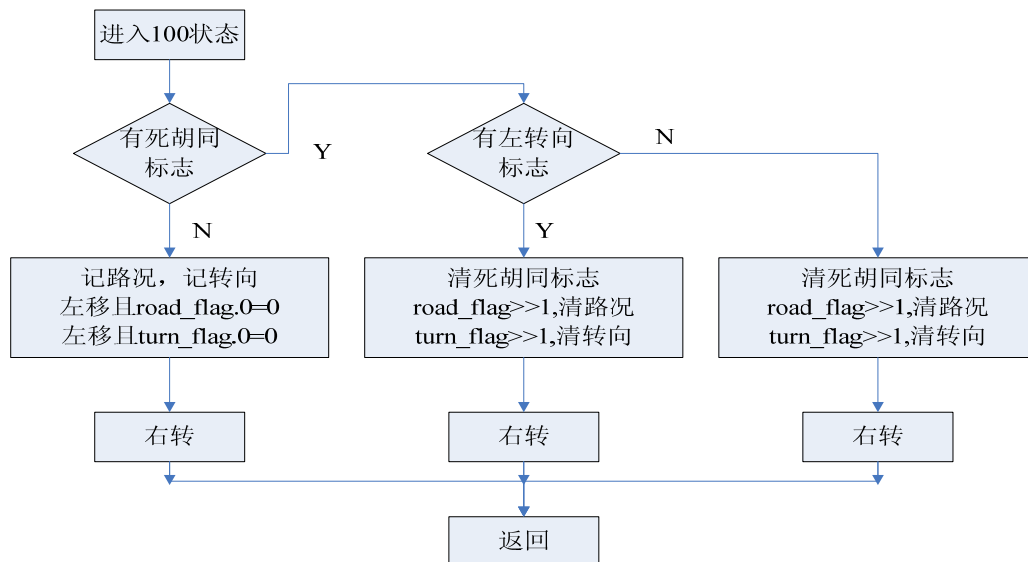


图 6.7 电脑鼠遇到直丁字路口 100 状态时的程序框图

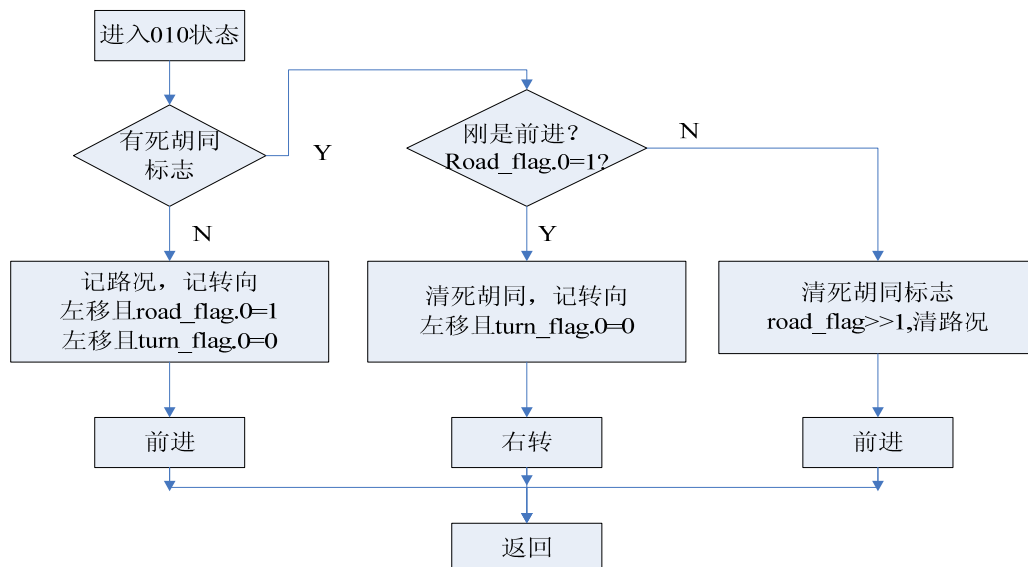


图 6.8 电脑鼠遇到右丁字路口 010 状态时的程序框图

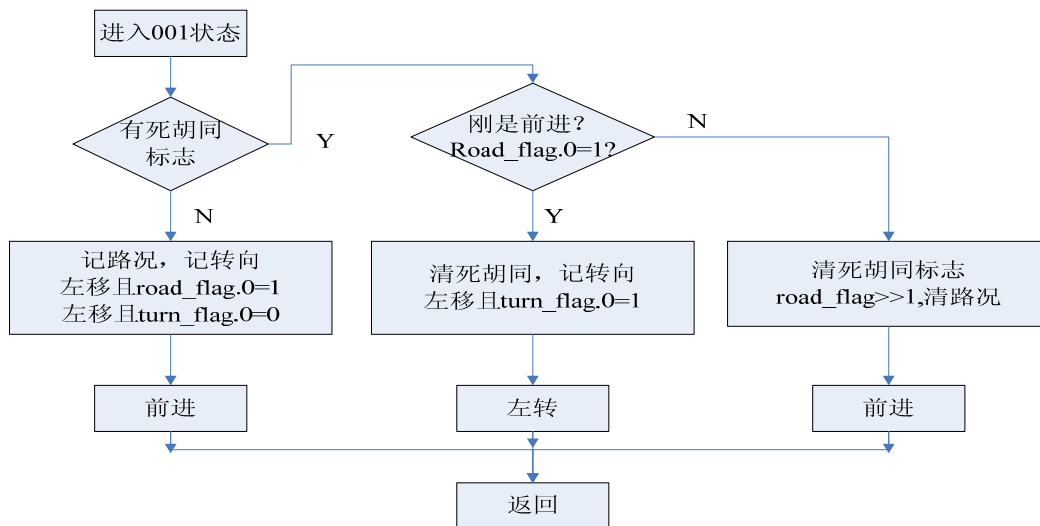


图 6.9 电脑鼠遇到左丁字路口 001 状态时的程序框图

6.2.3 优化蚁群算法求解迷宫最短路径

我们在第五章已经详细讲解了怎样使用蚁群算法求解迷宫最短路径问题，在这里为了保持文章的连贯性，我再次给出改进蚁群算法求解迷宫最短路径的具体实现步骤。同时给出其程序流程框图，如图 6.10 所示。

Step1. 初始化参数 α , ρ , $N_{C_{\max}}$, m ;

Step2. 生成 $2*m$ 只蚂蚁，将其中 m 只放到起点以 (i_s, j_s) ，另外 m 只放到终点 (i_t, j_t) ;

Step3. while 活蚂蚁数大于 0 do

for 蚂蚁 $k = 1, 2, \dots, 2m$ do

if 蚂蚁 k 活着 and $allowed_k$ 非空 then

(1)按公式计算转移概率 P_k ;

(2)按概率 P_k 选择下一个点;

(3)根据蚂蚁生命周期(i), (ii), (iii)情况, 决定蚂蚁 k 的死、活状态;

(4)如果蚂蚁 k 没死亡, 将其所选择的点加入 $tabu(k)$ 中。

end if

Step4. 计算本次迭代的最好解, 如优于当前最好解, 则用其替换当前最好解;

Step5. 按公式(5.9)更新路径上的信息素;

Step6. if N_c 小于 $N_{C_{\max}}$ 且未进入停滞状态 then

(1)清空所有蚂蚁禁忌表中的数据;

(2) $N_c := N_c + 1$;

(3)转至 Step2

else 输出最优结果

end if

End

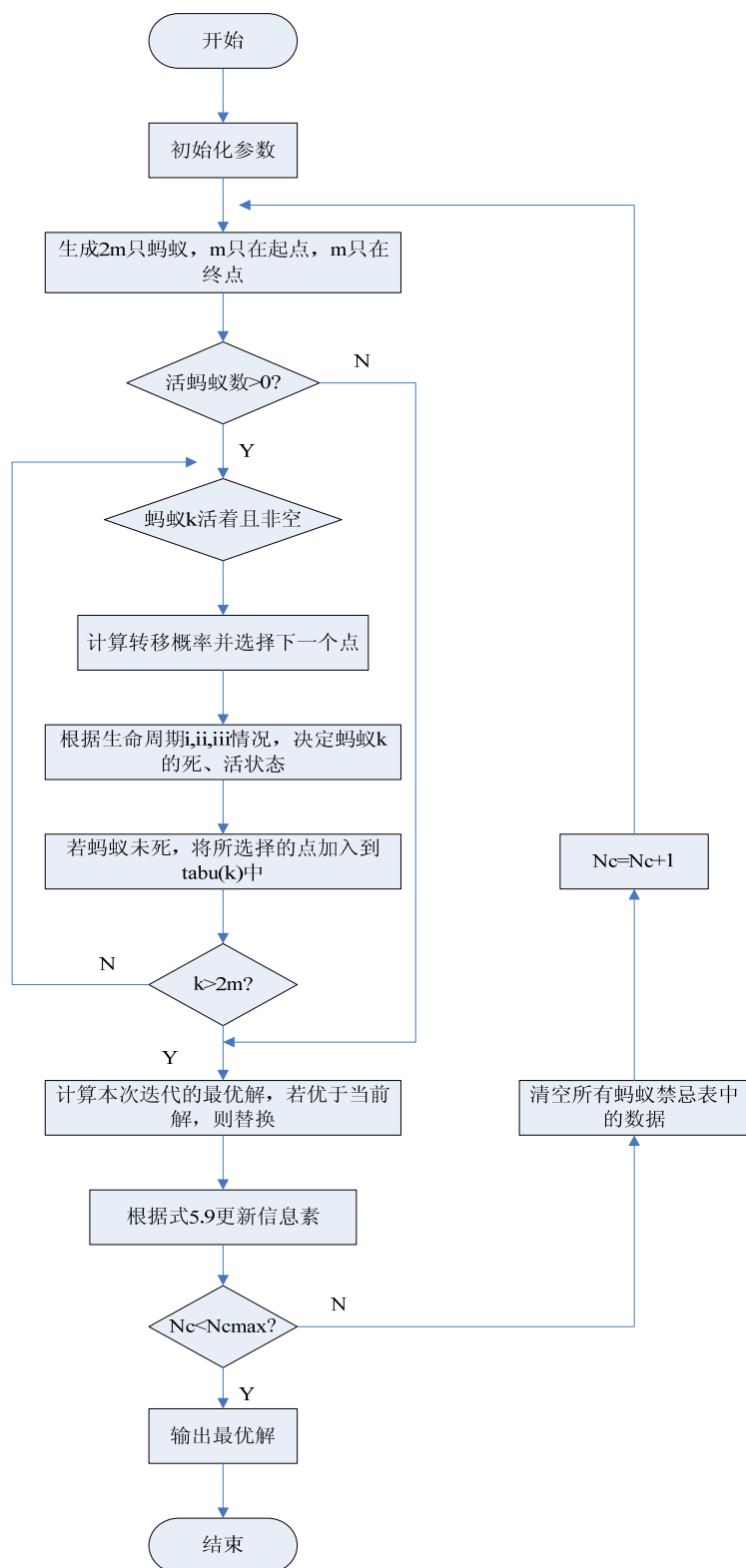


图 6.10 优化蚁群算法求解迷宫最短路径的程序流程

6.3 实验调试及分析

6.3.1 蚁群算法用于求解栅格迷宫

我们使用的软件是 40×40 的栅格迷宫。按照如下步骤进行实验：

首先在下拉菜单“设置”里选“参数设置”选项，修改相应参数当然也可使用默认值，点击确定；选“显示栅格”点击确定；选“自动生成障碍”，点击确定，当然也可以根据需要手动设置障碍。接着对下拉菜单“操作”进行操作。下拉菜单“操作”里共有四个选项：确认障碍设置、开始仿真、显示上一个地图和性能测试。

我们按照上述的实验步骤进行实验，如图 6.11 所示。不论设置障碍多么复杂，只要事实上存在通路，就可以找到最短路径。我在奔 4，1.8GHz 的机子上用软件规划一次路径只需要 0.112ms，可见本算法的效率还是很高的。一般迷宫算法仅仅是找到路径而非最优，而我们的实验，采用了蚁群算法求解迷宫的最短路径。在图 6.11 中，黑色路径为采用蚁群算法规划出来的最短路径，绿色路径为最短且拐弯最少的路径。

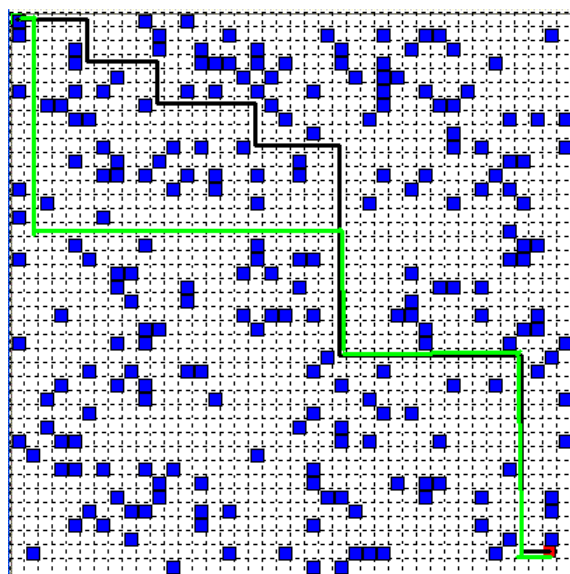


图 6.11 蚁群算法解栅格迷宫问题

从实验中，我们可以看到虽然蚁群算法可以规划出迷宫机器人的最短路径，但是此最短路径不一定是最佳路径。可见，蚁群算法还是有进一步优化的空间的。

6.3.2 基于 Visual Basic 开发环境的优化蚁群算法的软件实现

在第五章的中，对蚁群算法及其应用已经有过详细的介绍，这里不再重复。本部分的内容将简要说明一下如何使用陈烨基于 Visual Basic 开发的“蚁群算法实验室”软件。我们使用默认的参数： $\alpha = 2$ ， $\beta = 2$ ， $\rho = 0.1$ ，其他参数设置见图。如图 6.12 所示，软件计算出了在最大迭代次数为 1000 时，蚂蚁在第 125 代中找到最短路径。还可以在“计算控制”选项中，对蚂蚁的移动进行单步执行。

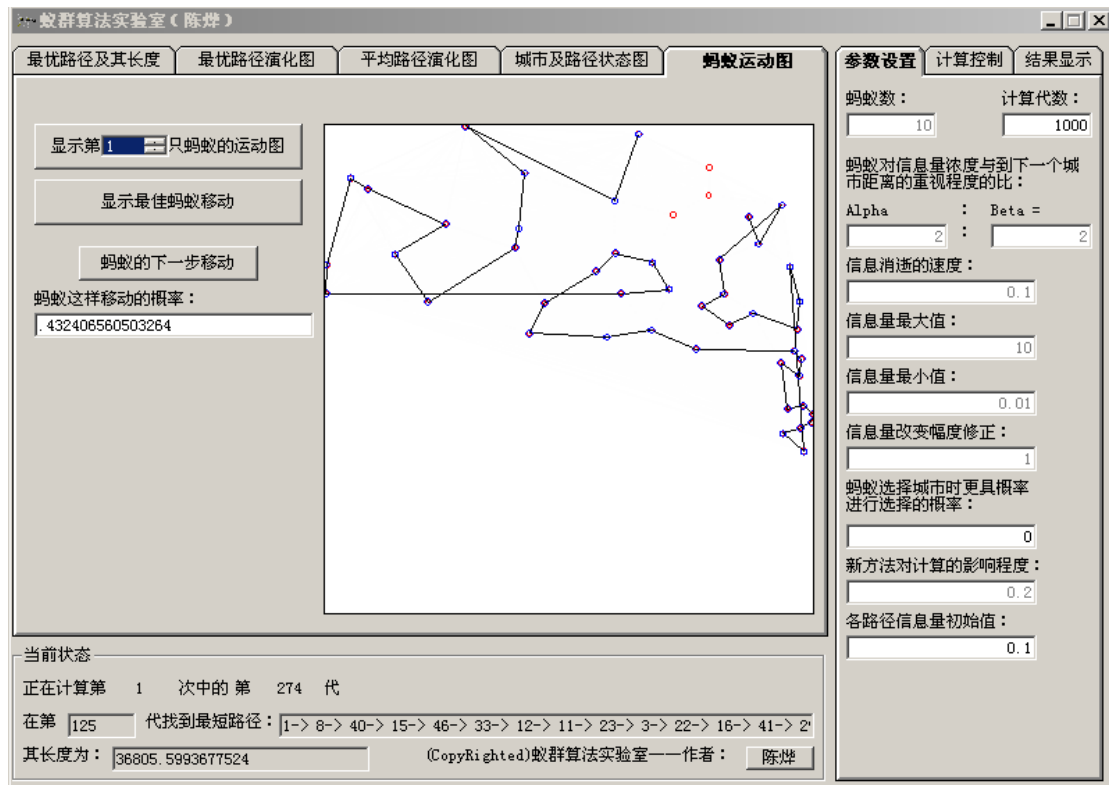


图 6.12 基于 Visual Basic 开发环境的优化蚁群算法的软件

6.3.3 优化蚁群算法用于解标准迷宫问题的实验结果

迷宫机器人比赛的标准场地规格为：16×16 个网格，设迷宫的起点坐标为（0，0），终点坐标为（8，8）。我们的标准迷宫按图 5.1 布局，首先利用其他算法如广度优先搜索算法求出迷宫问题的可行解，然后再用优化蚁群算法^[73]求解该问题，并记录发现最优解时所需的迭代次数，算法初始化参数设置为： $\alpha = 2$ ， $\beta = 5$ ， $\rho = 0.1$ 。路径上初始信息素为 $\tau_{ij \rightarrow uv}(0) = 0.1$ ，蚂蚁数为 $m \times n \times 2$ （其中 m 为迷宫的行数， n 为迷宫的列数），最大迭代次数 NC_{\max} 为 1000。

我们通过应用优化蚁群算法，经过 19 次迭代后，可以找到最优路径为：(0,0)->(0,1)->(0,2)->(0,3)->(0,4)->(0,5)->(0,6)->(1,6)->(1,7)->(1,8)->(1,9)->(0,9)->(0,10)->(0,11)->(0,12)->(0,13)->(0,14)->(1,14)->(2,14)->(3,14)->(3,15)->(4,15)->(5,15)->(6,15)->(6,14)->(7,14)->(8,14)->(8,15)->(9,15)->(10,15)->(11,15)->(12,15)->(12,14)->(12,13)->(12,12)->(12,11)->(12,10)->(12,9)->(11,9)->(10,9)->(9,9)->(8,9)->(7,9)->(7,8)->(8,8)。考虑拐弯对最短路径的影响，迷宫电脑鼠实际走过的网格数为 44 格，拐弯次数为 16。从电脑鼠的拐弯次数看，图 5.1 迷宫机器人的标准比赛场地还是比较复杂的。不过，由于迷宫的规模较小，我们应用优化蚁群算法求解迷宫最短路径的效果很不错。

当然，我们也要认识到应用蚁群算法求解最短路径时，尤其在所求问题的规模比较大时，求解效率并不太理想，故它能有效求解的问题的规模比较小；并且由于蚁群算法本身的结构和特点决定了其求解时间相对比较长，信息存贮量大。同时，在地形非常复杂、路

线非常曲折的类似迷宫的情况下，蚁群算法的搜索效率比较低：每得到一条通路的过程就是一个走迷宫的过程，成功系数较低；而且还会导致大量蚂蚁死亡或生成重复解，以至影响了搜索效率。所以，我们还需要继续努力去研究和完善蚁群算法！

6.4 本章小结

本章是迷宫机器人的软件设计和程序实现部分，是本论文的重要组成章节。本章对迷宫机器人的核心——控制系统进行了较为深入的研究，不仅给出了初始化模块程序、电机调速程序，还给出迷宫挡板检测程序的具体流程框图。其次，为了让读者更清楚地了解迷宫机器人是怎样工作的，本章给出了详细的迷宫智能算法实现方法和步骤。再次，通过精选三个很有代表性的实验来验证优化蚁群算法求解迷宫最优路径的可行性与优越性；同时从实验过程中发现不足，并把这些不足之处作为将来我们要克服的目标。最后发出继续努力研究和完善蚁群算法的倡议！

第七章 总结和展望

7.1 课题总结

人类对科学的探索是永无止境的，这是科学技术不断取得进步的原因所在。同时，阶段性的总结也很有必要。在这里，我们就一起来回顾一下本文的工作。

本文结构清楚，整体上可分为硬件设计和软件设计。

第一章描述了机器人的由来及其发展，并对机器人的定义、组成等问题进行了叙述。

第二章给读者提供了一套制作迷宫机器人系统的完整设计方案。同时还给出系统总体设计、系统硬件机身结构设计、系统硬件电路设计、系统基本驱动软件设计、系统搜寻策略软件设计的进展流程图。本章指明了全文的工作方向。

第三章是本文的核心章节之一，是迷宫电脑鼠的硬件设计部分。本章详细叙述了迷宫电脑鼠机身设计、电源模块设计、电机驱动模块设计、测速电路模块设计、迷宫挡板检测电路、微处理器模块设计这六大主要功能模块的具体实现方案，有的设计甚至提供了多套方案。可以说本章的工作做好了，我们的工程就成功了一半。

第四章首先介绍了移动机器人的路径规划及其方法。然后介绍基于蚁群算法的机器人路径规划，为下一章全面介绍蚁群算法及其在迷宫最优路径问题的应用作了铺垫。

第五章是迷宫电脑鼠软件设计的控制算法部分，是论文中最具鲜明特色的部分，也是全文的核心章节。本章首先介绍了迷宫场地的设计和迷宫的表示，把实际迷宫抽象化数学化成一个二维数组。然后又简单介绍迷宫机器人常用搜寻法则和策略。在求解迷宫问题的算法中，详细阐述基本蚁群算法及其应用；并通过学习和研究，改进蚁群算法来求解迷宫最优路径问题。本章是全文工作目标能否实现的关键。

第六章是迷宫机器人的软件设计和程序实现部分，是本论文的重要组成章节。本章给出了迷宫机器人初始化模块程序、电机调速程序，迷宫挡板检测程序的具体流程框图。本章还给出了详细的迷宫智能算法实现方法和步骤。通过精选三个很有代表性的实验来验证优化蚁群算法求解迷宫最优路径的可行性与优越性。同时从实验过程中发现不足，并把这些不足之处作为将来我们要克服的目标。

本课题在以下方面取得了一些成果：

(1)应用 ARM 控制器对机械子系统进行控制，充分发挥其控制能力强，计算处理能力强，控制精度高等优点。

(2)在车载子系统中加入了红外传感器、测速模块等，应用 ARM 加以控制，提高了迷宫机器人的智能化、处理能力和处理效率。

(3)提出了一种求解迷宫问题的蚁群优化算法。该算法首先将蚁群分成两组，分别从迷宫的起点和终点出发，每只蚂蚁按路径上的信息素独立选择前进的方向。根据蚂蚁在迷宫中的行走状态，定义了三种不同类型的生命周期。根据蚂蚁在每次移动后所处的状态，生

成问题的可行解。仿真实验证实了本算法的有效性。

7.2 未来展望

虽然课题取得一些成果，但仍然有很多方面的工作需要我们在今后的工作进一步探索和研究。

(1)研制各种高性能的传感器，使得机器人对作业任务和环境有正确的感知，这是实现机器人自主功能的前提。

(2)提高机器人本身的智能性，减少机器人对操作者的依赖性，特别是在复杂或非结构化的环境中，这是实现机器人智能的关键。

(3)如何进行智能机器人在各个工作状态下的平滑切换，因为作业任务和环境不同，有时机器人可以完全自动完成，而复杂情况又必须有操作者的实时干预才能有效完成，这就要求不同工作方式的平滑切换，而不至于破坏整个系统的稳定和性能。

(4)进一步研究蚁群优化算法 ACOA(Ant Colony Optimization Algorithm)。首先，ACO 的收敛性证明还很不完善，给出更强的收敛性证明并得出收敛速度的估计将会加速算法的发展。进一步改进 ACO 算法，提高其收敛速度。算法的收敛速度一直是人们关注的问题，虽然改进的 ACO 算法的收敛速度有了很大的提高，但对于求解大规模的优化问题还不是很理想。其次，ACO 算法对复杂组合优化问题的描述能力不强，很多问题比较难以描述为解构造图的形式，提出一种更为合理的解构造方式将能大大扩大该算法的应用领域。再次，研究 ACO 的并行实现。ACO 本质上的并行性为其并行实现提供了坚实的基础，如何实现蚁群之间的通讯以及蚁群的调度是并行实现的关键。最后，与其它算法相结合，实现混合 ACO 算法。实验结果显示，ACO 算法与其它算法的结合有助于提高算法的性能。可利用的算法有局部搜索、遗传算法、免疫算法、模拟退火、禁忌搜索等。

总之，我们对机器人技术和 ACO 算法的研究还有很多工作要做，我们还需要继续努力！

参考文献

- [1] [美]David Cook 著. 机器人制作入门篇. 崔维娜, 高玉苹, 郑静译. 北京: 北京航空航天大学出版社, 2005
- [2] [美]约翰·埃欧文著. PIC 机器人初学指南. 边宇枢, 高志慧译. 北京: 科学出版社, 2005
- [3] [美]丹尼斯·克拉克, 迈克尔·欧文斯著. 机器人设计与控制. 宗光华, 张慧慧译. 北京: 科学出版社, 2004
- [4] 方建军, 何广平. 智能机器人. 北京: 化学工业出版社, 2004
- [5] 王民忠. 缔造传奇——机器人大赛揭秘. 北京: 科学出版社, 2004
- [6] 周立功. ARM 嵌入式系统基础教程. 北京: 北京航空航天大学出版社, 2005
- [7] 周立功. ARM 嵌入式系统实验教程(一). 北京: 北京航空航天大学出版社, 2005
- [8] 魏忠等. 嵌入式开发详解. 北京: 电子工业出版社, 2003
- [9] 张铁, 谢存禧. 机器人学. 广州: 华南理工大学出版社, 2001
- [10] 毛宗源. 机器人的智能控制方法. 北京: 国防工业出版社, 2002
- [11] 张福学. 智能机器人传感技术. 北京: 电子工业出版社, 1996
- [12] 蔡自兴. 机器人学. 北京: 清华大学出版社, 2000
- [13] 徐心和, 郝丽娜, 丛德宏. 机器人原理及应用[M]. 沈阳: 东北大学音像出版社, 2004
- [14] 周明, 孙树栋. 遗传算法原理及应用. 北京: 国防工业出版社, 1996
- [15] Marco Dorigo and Thomas Stützle. Ant Colony Optimization[M]. MIT Press, 2004
- [16] David A Patterson, John L Hennessy, Computer Organization and Design. the hardware/software interface, San Francisco, Morgan Kaufmann. 1998
- [17] David Seal. ARM Architecture Reference Manual. Pearson Education limited, 2001
- [18] Grossberg S. Nonlinear neural networks: Principles, mechanisms, and architecture. Neural Networks: 1998, (1): 17~61
- [19] Aoyama A, Doyle F and Venkatasubramanian V. Control affine neural network approach for nonminimum phase nonlinear process control. Proc. Contr., 1996, 6(1): 17~26
- [20] C. Y. Chow. An Introduction to Computation Fluid Mechanics. New York: Wiley, 1979
- [21] C. I. Connolly. Harmonic function as a basis for motor control and planning, Ph. D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst, MA, USA, 1994
- [22] Bernhard Nebel, Jens-Steffen Gutmann, Wolfgang Hatzack. The CS Freiburg '99 Team. RoboCup-99 Team Descriptions Middle Robots League, Team CS-Freiburg, pages, 113~117
- [23] Bienz Z, Hwang D H and Oh S R. A nonlinear iterative learning method for robot path control. Robotics, 1991, 9(4): 387~392
- [24] Chien Chiang Ju and Lin Jing Sin. A P-type iterative learning controller for robust output tracking of nonlinear time varying systems. Int. J. Control, 1996, 64(2): 319~334
- [25] 李玉榕, 郭智疆, 蒋静坪. 多传感器融合在移动机器人运动控制中的应用, 仪器仪表学报, 2002, 23(1): 106~110
- [26] 曹文智. 自主移动机器人的控制系统开发[D]. 沈阳: 东北大学, 2004
- [27] 王幸之, 王雷, 王闪. 单片机应用系统抗干扰技术[M]. 北京: 北京航空航天大学出版社, 2000
- [28] 段勇, 徐心和. 基于不确定网格地图的移动机器人导航. 控制理论与应用, 2006 年 06 期
- [29] 韩浩, 董砚秋, 张娟. 基于湖面清扫机器人的图像识别技术研究. 科技资讯, 2006 年 33 期
- [30] 赵海文, 岳宏, 张雅丽等. 移动机器人分布式超声探测系统设计. 河北工业大学学报, 2006 年 06 期

- [31] 薛定宇. 反馈控制系统设计与分析——MATLAB 语言应用[M]. 北京: 清华大学出版社, 2000
- [32] 李贻斌, 刘明, 周凤余等. 移动机器人多超声波传感器信息融合方法, 系统工程与电子技术, 1999, 21(9): 55~57
- [33] 段海滨. 蚁群算法原理及其应用. 科学出版社. 2005
- [34] 李士勇, 陈永强, 李研. 蚁群算法及其应用. 哈尔滨工业大学出版社. 2004
- [35] Mark E. Rosheim, Robot Evolution: The Development of Anthrobotics, a wiles-interscience publication JOHN WILEY&SONS, INC. 1994
- [36] A.M. Turing, "Computing Machinery and Intelligence," Mind, Vol. 59, Oct. 1950, pp: 433~460
- [37] Evens. J, et al. "Helpmate: A Service Robot for Health Care.". Industrial Robot, 16: 2 June 1989, Pages: 87~89
- [38] Schodt F. Inside the Robot Kingdom: Japan, Mechatronics, and the Coming Robotopia. New York: Kodansha International, 1988, 256~258
- [39] Yount J and Bennett D. "Design, Construction and Testing of an Anthropomorphic Robot."17th International Symposium on Industrial Robots-Robot II Conference Proceedings, Chicago, IL, April 1987. Dearborn, Michigan: SME. Pages 26~30, and 77~86
- [40] Vladimir J.Lumelsky and Alexander A.Stepanov. "Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape". Algorithmica, 2, 403~430, Springer-Verlag NewYork Inc. 1987
- [41] A.Sankaranarayanan and M.Vidyasagar. "A New Path Planning For Moving A Point Object Amidst Unknown Obstacles In a Plane". Proc. of the 1990 IEEE Int. Conf. on Robotics and Automation, pp 1930~1936, May 1990
- [42] H. Noborio. "A path-planning algorithm for generation of an intuitively reasonable path in an uncertain 2D workspace". Proc.of the Japan-USA Symposium on Flexible Automation, pp: 477~480, July 1990
- [43] H.Noborio. "A sufficient condition for designing a family of sensor-based deadlock-free path-planning algorithms". Journal of Advanced Robots, Vol. 7, No. 5, pp: 413-433, January 1993
- [44] A.Sankaranarayanan and M.Vidyasagar. "Path Planning For Moving A Point Object Amidst Unknown Obstacles In a Plane:a new algorithm and a general theory for algorithm development". Proc.of the 29th Conf. on Decision and Control, pp: 1111~1119, December 1990
- [45] H.Noborio, Y.Maeda and K.Urakawa. "Three or more dimensional sensor-based path planning algorithm HD-I". Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and System, pp: 1699~1706, Octobe 1999
- [46] H.Noborio, L Yamamoto, T.Komaki. "Sensor-based path planning algorithms for a nonholonomic mobile robot". Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, October 2000
- [47] J. Pearl. "Heuristics". Addison Wesley, Oct. 1985
- [48] Vladimir Lumelsky and Tim Skewis. "Incorporating Range Sensing in the Robot Navigation Function", IEEE Transactions on Systems. Man, and Cybernetics, vol. 20, No. 5, Sep/Oct. 1990
- [49] V.Lumelsky and A.Stepanov. "Dynamic path planning for a mobile automaton limited information on the environment," IEEE Trans. Automat. Contr. Vol. AC~31, no. 11, Nov. 1986, pp: 1058~1063
- [50] Ishay Kamon and Ehud Rivlin. "Sensory-Based Motion Planning with Global Proofs". IEEE Transactions on Robotics and Automation, Vol. 13, No. 6, Dec.1997
- [51] Ishay Kamon and Ehud Rivlin, Elon Rimon. "A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots". Proc. of the 1996 IEEE International Conference on Robotics and Automation Minneapolis, Minnesota, April 1996, pp: 429~435
- [52] W. S. Massey. Algebraic Topology. New York: Harcourt, Brace, &World, 1967

- [53] H. Abelson and A. diSessa. Turtle Geometry. MIT Press, Cambridge, MA, 1980, pp: 176~199
- [54] Colorni A, Dorigo M, Maniezzo V. An investigation of some properties of an ant algorithm [A]. Proc. Of the Parallel Problem Solving from Nature Conference (PPSN' 92) [C]. Brussels, Belgium: Elsevier Publishing, 1992, 509~520
- [55] 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240~1245
- [56] 马良, 项培军. 蚂蚁算法在组合优化中的应用[J]. 管理科学学报, 2001, 4 (2): 32~37
- [57] 彭喜元, 彭宇, 戴毓丰. 群智能理论及应用[J]. 电子学报, 2003, 31(12A): 1982~1988
- [58] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[A]. Proceedings of the First European Conference of Artificial Life(ECAL'91) [c]. Elsevier, 1991: 134~142
- [59] M Dorigo, V Maniezzo. A Colorni.The ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on System, Man and Cybernetics PartB, 1996, 26(1): 29~42
- [60] L M Gambardella, M Dorigo. Solving symmetric and asymmetric TSPs by ant colonies [A]. Proceedings of the IEEE International Conference on Evolutionary Computation(ICEC' 96)[C]. Nagoya Japan, 1996: 622~627
- [61] M Dorigo, L M Gambardella. A study of some properties of Ant Q [A]. Proceedings of PPSN IV Fourth International Conference on Parallel Problem Solving From Nature[C]. Springer, Berlin, 1996: 656~665
- [62] Stutzle T, Hhoos H. The MAX MIN ant system and local search for the traveling salesman problem [A]. In Proceedings of the IEEE International Conference on Evolutionary Computation(ICEC'97) [C]. Indianapolis, USA 1997: 309~314
- [63] Bernd Bullnheimer, Richard F Hartl, Christine Straub. A new rank based version of the ant system a computational study [DB/OL]. <http://www.wu.wien.ac.at/am>
- [64] Caro G Di, Dorigo M. AntNet: Distributed stigmergetic control for networks[J]. Journal of Artificial Intelligence Research(JAIR), 1998(9): 317~365
- [65] 孙熹, 王秀坤, 刘业欣等. 一种简单蚂蚁算法及其收敛性分析[J]. 小型微型计算机系统, 2003, 24(8): 1524~1526
- [66] 段海滨, 王道波. 蚁群算法的全局收敛性研究及改进[J]. 系统工程与电子技术, 2004, 26(10): 1506~1509
- [67] 吴启迪, 汪镭. 智能蚁群算法及其应用[M]. 上海: 上海科技教育出版社, 2004
- [68] 李天成. 应用智能蚂蚁算法解决旅行商问题[D]. [硕士论文]. 厦门: 厦门大学, 2002
- [69] 赵强. 蚁群算法在中压城市配电网规划中的应用[D]. [硕士论文]. 成都: 四川大学, 2003
- [70] 沈彬. 改进蚁群算法在物流配送中的应用研究[D]. [硕士论文]. 杭州: 浙江大学, 2004
- [71] 王笑蓉. 蚁群优化的理论模型及在生产调度中的应用研究[D]. [博士论文]. 杭州: 浙江大学, 2003
- [72] 胡小兵. 蚁群优化原理、理论及其应用研究[D]. [博士论文]. 重庆: 重庆大学, 2003
- [73] 闻育. 复杂多阶段动态决策的蚁群优化方法及其在交通系统控制中的应用[D]. [博士论文]. 杭州: 浙江大学, 2004

致 谢

本文的研究是在导师王祖麟教授的悉心指导下完成的。王老师宽广的胸怀、严谨的治学态度、深厚的学术功底、敏锐的洞察力、开拓创新的精神以及对工作和事业的执着莫不深深地影响着我，使我受益终身。在论文完成之际，谨向我的导师王祖麟教授表示由衷的感谢。

同时，还要感谢机电学院的杨杰老师、周克良老师、吴银凤老师、钟萍老师以及教务处钟义红老师的关心和帮助。感谢广州周立功单片机有限公司周立功教授，东华理工学院周航慈教授在学术和就业方面的指导。

感谢李明、刘维、朱顺利、刘洁、王文鑫五位师兄（姐）以及曹兰、卢峰、李洁、赖武军、睢相林、刘凤磊、邓义平等同学在我的学术研究以及生活中给予的支持和帮助。

最后感谢父母、哥哥多年来的关心和爱护，一直以来他们对我的求学都给予坚定的支持！

吴大舜
2008年11月9日

附录 A 迷宫机器人实物图

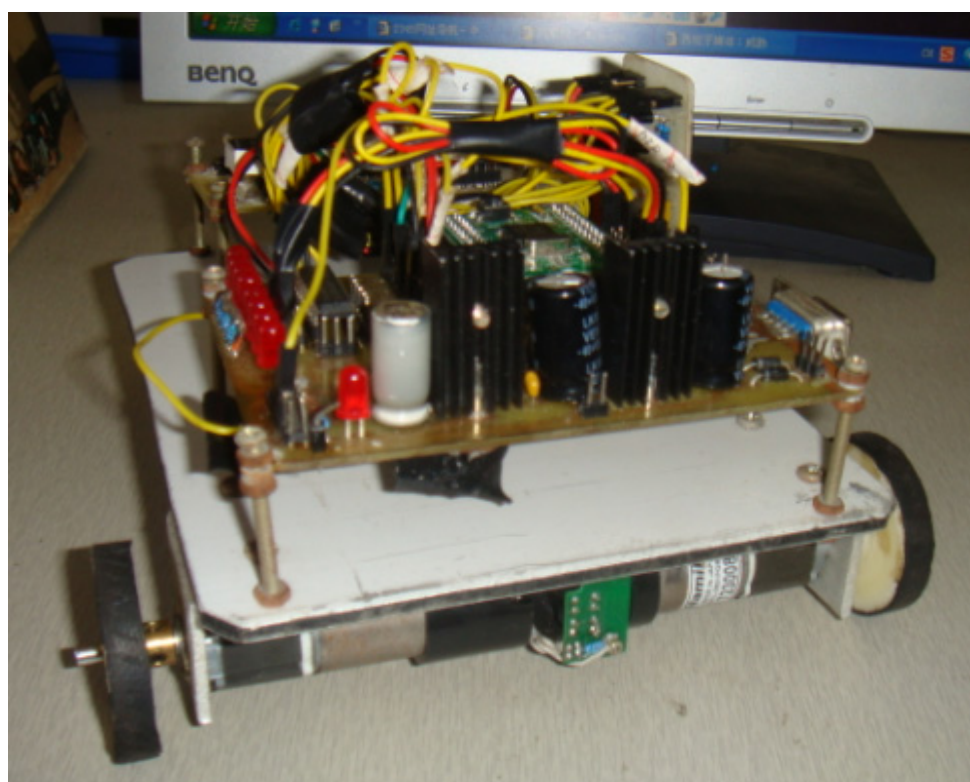
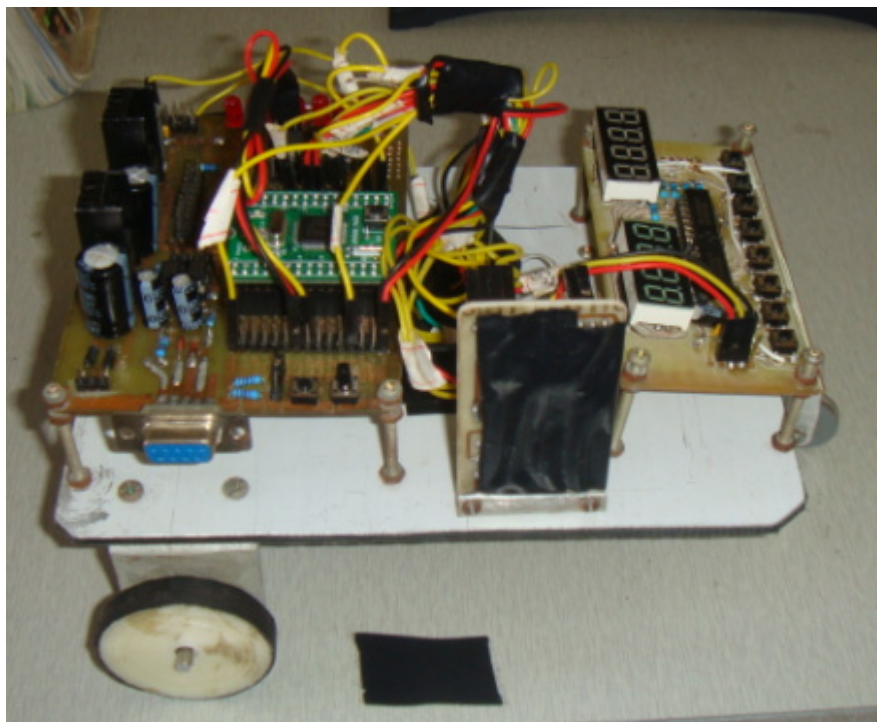


图 A1 迷宫机器人实物图

附录 B 迷宫求解程序参考源代码

```
/*class:
Matrix: 矩阵类
Offsets: 搜索偏移
enum directions: 四个方向
struct item: 搜索节点
Migong: 迷宫类
1.创建一个 Migong 对象
2.使用 Create 方法输入数据
3.使用 Solve 方法进行求解
4.ShowSolve 方法显示解
5.可以重复使用 Create 方法
6.入口只能在左上角
7.默认出口在右下角
ShowAllPath: 穷举所有的路径
备注:
由于算法原因, 这里的所有路径应该是指介于:
a.如果两条路存在某个点不同那么就是不同的路
b.如果在一条路中去掉一个或者一个以上的圈, 那么他们是同一条路之间意义上的路
*/

#include "iostream.h"
#include "stack.h"
#include "vector.h"

using namespace std;

#ifndef MIGONG_H
#define MIGONG_H
//////////
////矩阵类////
//////////
class Matrix{
int* m;
int row, col;
bool iscreate;
public:
Matrix(){m=0;iscreate=false;};
~Matrix() {Release();};
bool Create(int, int);
int& operator () (int, int);
int GetRow(){return row;};
int GetCol(){return col;};
void Release();
```

```

void Show(char, char );
};

bool Matrix::Create(int r, int c)
{
if( r<=0 || c<=0) return false;

Release();

row = r;
col = c;
m = new int[row*col];

for (int i=0;i<row*col;i++)
{
*(m+i) = 0;
}
iscreate = true;
return true;
}

int& Matrix::operator()(int r, int c)
{
return *(m+r*col+c);
}

void Matrix::Release()
{
if (iscreate)
{
row = col = 0;
if (m) delete[] m;
m = 0;
}
iscreate = false;
}

void Matrix::Show(char blk='#', char nblk=' ')
{
int i, j;
for (i=0;i<row;i++)
{
for (j=0;j<col;j++)
{
if (*(m+i*col+j) == 0)

```

```

cout<<nblk;
else
cout<<blk;
}
cout<<endl;
}
}
//////////
////迷宫相关数据结构的定义////
//////////
struct offsets{
int a, b;
};

enum directions{
_S = 0,
_E,
_N,
_W
};

struct item{
int row, col, dir;
};

class Migong{
static offsets move[4];
Matrix maze;
Matrix mark;
int row;
int col;
int desr;
int desc;
stack<item> stk;
bool iscreate;
int pathlength;
bool GetPath();
bool IsInPath(int, int);
public:
Migong(){issolved=false;result=0;pathlength=row=col=0;iscreate=false;};
~Migong(){Release();};
bool Create(int* , int , int , int , int );
void Solve();
void Release();
void OutputMaze();

```

```

void ShowSolve(char, char );
public:
bool issolved;
item* result;
};

offsets Migong::move[4]={ {1, 0}, {0, 1},
{-1, 0}, {0, -1}};

//////////
//////迷宫数据应该是不含边框的////
//////////
bool Migong::Create(int* m, int r, int c, int desrow=-1, int descol=-1)
{
if (r<=0 || c<=0) return false;

Release();

if (desrow==-1 || descol==-1)
{
desr = r;
desc = c;
}
else
{
desr = desrow;
desc = descol;
}

row = r;
col = c;
maze.Create(r+2, c+2);
mark.Create(r+2, c+2);

int i, j;

for (i=0;i<r+2;i++)
{
for (j=0;j<c+2;j++)
{
if (j==0 || j==c+1 || i==0 || i==r+1)
{
mark(i, j) = maze(i, j) = 1;

} else

```

```

{
mark(i, j) = 0;
maze(i, j) = m[((i-1)*col+j-1)];
}
}
}
return iscreate = true;
}

bool Migong::GetPath()
{
mark(1,1) = 1;
item temp;
temp.col = 1;
temp.row = 1;
temp.dir = _S;
stk.push(temp);

while (!stk.empty())
{
temp = stk.top();
stk.pop();

int i = temp.row;
int j = temp.col;
int d = temp.dir;

while (d<4)
{//根据当前点的状态确定下一个搜索点
int g = i + move[d].a;
int h = j + move[d].b;

if (g==desr && h==desc)
{
return true;
}
//如果这个点不是障碍点且没有被搜索过那么可以对这个点进行搜索
if (maze(g, h)==0 && mark(g, h)==0)
{
mark(g, h) = 1;
temp.row = g;
temp.col = h;
temp.dir = d+1;
stk.push(temp);
i = g;

```

```

j = h;
d = _S;//对下一个点进行搜索
}
else d++;
}
}
return false;
}

void Migong::Solve()
{
issolved = GetPath();
if (issolved)
{
pathlength = stk.size();
result = new item[pathlength];
for (int i=0;i<pathlength;i++)
{
*(result+i) = stk.top();
stk.pop();
// cout<<"("<<(*(result+i)).row<<","<<(*(result+i)).col<<")"<<endl;
}
}
while (!stk.empty())
stk.pop();
}

void Migong::Release()
{
if (iscreate)
{
maze.Release();
mark.Release();
row=col=0;
if (result)
delete [] result;
result = 0;
while (!stk.empty())
stk.pop();
}
iscreate = false;
issolved = false;
pathlength = 0;
}

```

```

void Migong::OutputMaze()
{
    if (!iscreate) return;
    maze.Show();
}

bool Migong::IsInPath(int r, int c)
{
    if (!iscreate || !issolved)
        return false;

    item temp;
    for (int i=0;i<pathlength;i++)
    {
        temp = *(result+i);
        if ((temp.row==r) && (temp.col==c))
            return true;
    }

    return false;
}

void Migong::ShowSolve(char blk='#',char s='o')
{
    if (!iscreate) return;
    if (!issolved)
    {
        cout<<"无解"<<endl;
    }
    else
    {
        int i, j;
        for (i=0;i<row+2;i++)
        {
            for (j=0;j<col+2;j++)
            {
                if ((i==1 && j==1) || (i==desr && j==desc))
                {
                    cout<<s;
                }
                else if (maze(i, j) == 1)
                {
                    cout<<blk;
                } else
                {

```

```

if (IsInPath(i, j))
cout<<s;
else
cout<<' ';
}
}
cout<<endl;
}
}
}

//////////
////穷举所有路径////
//////////
offsets move[4]={ {1, 0}, {0, 1},
{-1, 0}, {0, -1}};

struct node
{
int row,col;
};

vector<node> path;
int count;
bool IsReachable( Matrix& maze, Matrix& mark, node beg, node des)
{
if (beg.row==des.row&&beg.col==des.col)
{//如果达到的话那么显示路径
count++;
cout<<"第"<<count<<"条路径:"<<endl;
for (int i=0;i<path.size();i++)
cout<<"("<<path[i].row<<","<<path[i].col<<")";
cout<<"("<<des.row<<","<<des.col<<")";
cout<<endl;
return false;
}
if (maze(beg.row, beg.col)==1 || mark(beg.row, beg.col)==1)
{
return false;
}

path.push_back(beg);
mark(beg.row, beg.col) = 1;

node nextnode;

```



```

for (int i=_S;i<_W+1;i++)
{
nextnode.row = beg.row + move[i].a;
nextnode.col = beg.col + move[i].b;

IsReachable(maze, mark, nextnode, des);
}

path.resize(path.size()-1);
mark(beg.row, beg.col) = 0;

return false;//如果不是穷举的话应该根据 for 循环的结果重新设置返回值
}

/*
参数 maze,mark 为迷宫长宽均加二的矩阵
desr,desc 为出口点
*/
void FindAllPath( Matrix& maze, Matrix& mark, int desr, int desc)
{
node first, last;
first.row = 1;
first.col = 1;
last.row = desr;
last.col = desc;

IsReachable(maze, mark, first, last);

path.clear();
}

/*
m 迷宫矩阵数据
r,c 行和列的大小
desr,desc 目标位置
*/
void ShowAllPath(int* m, int r, int c, int desr=-1, int desc=-1)
{
Matrix maze, mark;
maze.Create(r+2, c+2);
mark.Create(r+2, c+2);

if (desr==-1 || desc==-1)
{

```

```

desr = r;
desc = c;
}

int i, j;

for (i=0;i<r+2;i++)
{
for (j=0;j<c+2;j++)
{
if (j==0 || j==c+1 || i==0 || i==r+1)
{
mark(i, j) = maze(i, j) = 1;

} else {
mark(i, j) = 0;
maze(i, j) = m[((i-1)*c+j-1)];
}
}
}

count = 0;
FindAllPath(maze, mark, desr, desc);

maze.Release();
mark.Release();
}
#endif

```

个人简历 在学期间发表的学术论文与研究成果

个人简历

吴大舜，男，汉族，江西上饶人，1982年8月19日出生。2004年7月，毕业于江西理工大学，获电气工程及其自动化专业工学学士学位；2009年1月，毕业于江西理工大学，获控制理论与控制工程专业工学硕士学位。

在学期间发表的学术论文与研究成果

- [1] 吴大舜，王祖麟. LabVIEW 虚拟仪器技术在大学实验教学中的应用. 科技广场. 2007(11): 185~186
- [2] 吴大舜，眭相林. 基于 DSP 的位置随动系统设计. 江西理工大学学报. (已录用)

江西理工大学学位论文独创性声明及使用授权书

学位论文题目: 基于改进蚁群算法的迷宫机器人的研究				学位级别	工学硕士
				所在学院	机电学院
学科专业	控制理论与 控制工程	导师姓名	王祖麟	学生姓名	吴大舜
学位论文 是否保密		如需保密, 解密时间		年 月 日	

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, 也不包含为获得江西理工大学或其他教育机构的学位或证书而使用过的材料, 指导教师对此进行了审定。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做了明确的说明, 并表示了谢意。

研究生签名: 时间: 年 月 日

学位论文使用授权书

学校有权保留提交论文的印刷版和电子版, 并提供目录检索和阅览服务, 可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人同意江西理工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。

注: 保密学位论文在解密后适用于本授权书。

同意授权

作者签名: 导师签名:
 签名日期: 年 月 日 签名日期: 年 月 日

不同意授权

作者签名: 导师签名:
 签名日期: 年 月 日 签名日期: 年 月 日

说明理由:

注: 本表交一份研究生处保存备案, 另在每本学位论文最后一页还需装订此表