

分类号.....  
UDC .....

密级.....  
编号.....

# 中南大学

CENTRAL SOUTH UNIVERSITY

## 硕士学位论文

论文题目 ..... 移动机器人状态监测与  
..... 故障诊断系统研究

学科、专业 ..... 控制理论与控制工程

研究生姓名 ..... 章慧团

导师姓名及  
专业技术职务 ..... 蔡自兴 教授

日 期 ..... 2006 年 4 月



**MS THESIS**

**Research on Condition Monitoring and Fault  
Diagnose System for Mobile Robot**

**Specialty:** Control Theory and Control Engineering

**Master Degree Candidate:** Zhang Huituan

**Supervisor:** Prof. Cai Zixing

**College of Information Science & Engineering**

**Central South University**

**ChangSha Hunan P.R.C**



## 摘要

移动机器人作为一种高技术手段在工业、交通、太空探索和军事应用领域发挥着十分重要的作用,研究针对移动机器人的状态监测与故障诊断技术已经成为未知环境中移动机器人技术研究的热点。

本论文来源于国家自然科学基金支持的重点项目“未知环境中移动机器人导航控制的理论与方法研究”(项目批准号:60234030),作为该项目研究的一部分,本文以中南大学智能所自行研制的移动机器人“中南移动-1”(MORCS-1)为实验平台,着重于状态监测与故障诊断系统方面的研究,其主要内容如下:

1. 针对 MORCS-1 系统,提出并实现了一个基于无线网络的移动机器人状态参数的远程获取系统,采用多线程方式获取移动机器人车载传感器参数,设计了通信协议帧,对传感器参数进行监测分析,为后续的故障诊断研究提供了基础;对于移动机器人车载的工控机系统,提出并实现了一个基于 C/S 模式的移动机器人车载工控机的监视系统,并对其所引起通讯流量变化进行了分析;同时论文基于 ICMP 和 SNMP 设计和实现了一个针对网络性能监测的子系统,对网络延迟时间、流量、发送错误率等网络性能参数进行监测,为无线网络安全提供保障。

2. 分析了各种故障诊断方法,针对 MORCS-1 系统,采用基于信号处理和知识的诊断方法:研究了基于小波变换算法的传感器诊断方法,利用编码器计算获得移动机器人角速率与光纤陀螺所测角速率的冗余关系,提出了一个针对 MORCS-1 光纤陀螺突发性故障的诊断方法;对于在未知环境中运行的移动机器人,有可能出现的车轮打滑、受阻等情况,提出了一个基于规则的车轮的运动状况诊断方法,为移动机器人摆脱当前状态提供一种策略。

各个子系统通过相互协调,实现了对移动机器人的状态监测。该监测系统在移动机器人平台 MORCS-1 上进行了实验,实验验证了其有效性。

**关键词** 移动机器人, 状态监测, 故障诊断, 小波变换



## ABSTRACT

Mobile robots as a high technology play an important role in industry、traffic、space exploration etc. Research on condition monitoring and fault diagnose system for robot has become a urge task in the field.

The study is supported by the key project of the National Natural Science Foundation of China under grant no.60234030, *Research on Theories and Methods of Navigation Control for Mobile Robots under Unknown Environment*.As one part of the project, this dissertation aims at designing a condition monitoring and fault diagnose system of MORCS-1 which is developed by Central South University Intelligent lab. The main contributions and works are described as follows:

1. Design a remote condition monitoring system for MORCS-1 based on wireless LAN, acquire many kinds of sensor data of mobile robot based on multi-thread method,design the frame format of communication system ,then analyze and storage the sensor data, which is the fundament of the research of diagnose theories; aim at the computer system of MORCS-1,design a industry personal computer condition monitoring system based on C/S model; aim at the delay、bandwidth、loss of wireless LAN, design a network performance monitor system based on ICMP and SNMP.

2. Through analyzing many kinds of fault diagnose theories and methods,aim at MORCS-1,adopt the signal and knowledge methods: research on sensor fault diagnose method based on wavelet algorithm ,according to the redundancy of angle velocity between the fiber optical gyroscope and calculated value based on encoders of robot, design a abrupt fault diagnose method of fiber optical gyroscope; under unknown environment, aim at the problem of the wheel's slip and blocked, design a wheel state diagnose algorithm based on knowledge. the validity of this algorithm was validated by a lot of experimentation on the mobile robot MORCS-1.

The whole system realize the efficiently condition monitoring and fault diagnose of robot by the cooperation of modules of mobile robot , The validity of this system was validated by experimentations on the mobile robot MORCS-1.

**KEY WORDS:** mobile robot, condition monitoring, fault diagnose, wavelet

## 目 录

第一章 绪论.....	1
1.1 研究背景.....	1
1.2 国内外研究现状分析.....	2
1.2.1 状态监测与故障诊断发展.....	2
1.2.2 故障诊断技术的主要方法.....	3
1.2.3 移动机器人故障诊断技术.....	7
1.3 本文的工作.....	8
1.4 论文安排.....	9
第二章 MORCS-1 体系结构.....	10
2.1 MORCS-1 移动机器人.....	10
2.2 MORCS-1 移动机器人硬件平台.....	11
2.2.1 硬件平台总体结构.....	11
2.2.2 MORCS-1 系统组成.....	12
2.3 MORCS-1 移动机器人软件结构.....	14
2.4 小结.....	15
第三章 移动机器人状态监测系统设计.....	16
3.1 状态监测系统设计.....	16
3.1.1 系统设计原则.....	16
3.1.2 功能规划.....	16
3.2 传感器状态参数获取设计.....	17
3.3 车载计算机远程监视设计.....	22
3.4 网络性能参数获取设计.....	25
3.4.1 基于 ICMP 的网络延迟的获取.....	26
3.4.2 基于 SNMP 的网络性能参数获取.....	28
3.5 小结.....	34
第四章 移动机器人故障诊断方法研究.....	35
4.1 小波分析的基本理论.....	35
4.2 连续小波故障诊断的原理.....	36
4.3 传感器故障诊断.....	38
4.3.1 传感器故障诊断仿真分析.....	38
4.3.2 机器人光纤陀螺故障诊断算法设计.....	41
4.4 车轮故障诊断算法设计.....	44
4.5 小结.....	46
第五章 监测与诊断系统实现.....	47
5.1 开发环境.....	47
5.2 系统所采用的关键技术.....	47
5.2.1 HP SNMP++.....	47
5.2.2 多线程及同步技术.....	48
5.3 诊断系统的软件结构.....	49
5.3.1 软件系统构成.....	49

---

5.3.2 软件类清单.....	50
5.4 系统运行.....	51
5.5 小结.....	52
第六章 结论与展望.....	53
6.1 结论.....	53
6.2 展望.....	53
参考文献.....	54
附 录.....	57
致 谢.....	67
攻读学位期间主要的研究成果.....	68

# 第一章 绪论

## 1.1 研究背景

本课题来源于国家自然科学基金重点项目：“未知环境下移动机器人导航控制的理论与方法研究”（项目批准号：60234030）。

21 世纪是人类向太空进军的世纪，太空作为人类尚未开发的宝地，已经成为各国的重要战略目标，机器人作为一种高技术手段在太空探索、军事应用领域发挥着越来越重要的作用，利用机器人进行空间探测和开发，成为世界各主要科技发达国家开发空间资源的主要手段之一<sup>[1]</sup>。

在深空探测中，由于移动机器人具有极大的优势，现已成为各国的研究热点<sup>[2]</sup>。世界各国都在开发、研制移动机器人系统<sup>[3]</sup>：1970 年，苏联月球-17 号探测器把世界第一个无人驾驶的月球车送上月球；后来发射的月球车-2 行驶 37 公里，向地球发回 88 幅月面全景图<sup>[4]</sup>；1996 年，美国国家航空和宇宙航行局(NASA)发射了火星探路者；1997 年索杰纳成功地在火星着陆，索杰纳在 90 余天里共行驶了 90 多米，拍摄了 500 多幅照片<sup>[5,6]</sup>。索杰纳的成功应用，成为移动机器人技术发展的一个崭新的里程碑，向人们展现了移动机器人代替人们从事肮脏、危险、枯燥工作的应用潜力<sup>[7]</sup>。

然而，在深空探测的机器人往往工作在十分恶劣的环境中，如强辐射、大温差、复杂地形等的复杂环境下，一旦发生事故就可能造成巨大的财产损失。在 1971 年，前苏联向火星发射了两辆火星车，其中一辆撞毁了，另一辆由于故障，在发送回一幅尚不完整的图片就失去了联系；在 2003 年发射的“勇气”号出现了驱动右前轮的马达无法正常工作、温度调节装置故障；同年发射的“机遇”号也出现右前轮操舵电动机故障、钻孔机故障、电池能量供应系统故障、机器臂失灵等问题。这引起人们对机器人系统的可靠性、安全性更加的关注，除了在设计与制造阶段，通过改进设计，研究和应用新材料、新工艺及加强生产过程的质量控制等措施来提高系统的可靠性与安全性外，提高系统可靠性与安全性的一个重要途径就是对系统进行实时的状态监测与故障诊断，从而实现对故障进行预警。

移动机器人状态监测和故障诊断技术是保障机器人的生存及成功完成作业任务的关键技术之一<sup>[8]</sup>，本文针对中南大学智能所自行研究设计的移动机器人“中南移动-1”(MORCS-1)，结合“未知环境下移动机器人导航控制的理论与方法研究”课题的研究，介绍了移动机器人状态监测系统的设计以及对移动机器人传感器故障诊断方法进行了研究。

## 1.2 国内外研究现状分析

### 1.2.1 状态监测与故障诊断发展

故障诊断始于(机械)设备故障诊断,其全名是状态监测与故障诊断(Condition Monitoring and Fault Diagnosis)<sup>[9]</sup>,包含两方面内容:一是对设备的运行状态进行监测,主要是指监测零部件是否出现异常现象,通过持续监测,达到在事故发生前给出预警;二是发现异常情况后对设备的故障进行分析、诊断,根据有关信息判断零部件出现故障的原因,为排除故障,恢复零部件功能提供信息。

监测与故障诊断技术基本上有两大用途:其一是用于构成离线式的“计算机辅助决策支持系统”,即利用故障诊断系统快速发现系统的故障,制定更加合理有效的系统维修方案,找出系统的薄弱环节,最终提高系统的可靠性和生产效益;其二是用于构成实时状态监视系统,或者进而实现高性能的自动控制系统。在这类应用中,故障诊断系统通常以在线方式运行,它连续地实时监视整个系统的测量参数,基于适当的在线故障监测与诊断算法给出系统的故障信息,包括故障报警信息、故障位置信息和故障程度信息等。采用先进的控制系统设计方法,可以使控制系统具有自动利用这些故障信息、根据系统运行的不同故障工况自动调整控制方案的能力。

随着检测技术和手段的进步,尤其是计算机技术和网络技术的发展,状态监测模式发生了很大的变化,大致可以分为3个阶段<sup>[10]</sup>:

#### (1) 第一代的单机监测诊断模式

在机器逐步取代手工在生产中的地位后,对机器设备的状态监测和故障诊断仅局限于“点到点”的诊断模式,即对每一台机器由一个监测系统来控制。它是一个封闭的系统,信息只在系统内部交流和处理。它的缺点是每个系统的工作效率低,浪费了大量的人力、物力。

#### (2) 第二代的分布式监测诊断模式

随着大型设备的出现,单机监测诊断模式已不满足需要,针对工厂的大型设备的功能分布和地域分布的特点,人们在网络技术的基础上实现了分布式监测诊断模式。它通过工业局域网将各个监测现场的本地计算机互联起来,实现资源共享,分散监控和集中操作、管理、诊断,提高了系统的工作效率。资源共享减少了人力、物力的浪费。它是一个相对开放的系统,但是,这种模式的缺点在于,它只存在于局域网内。

#### (3) 第三代的远程监测诊断模式

进入20世纪90年代后,随着Internet和万维网技术的普及和完善,机器设

备监测与故障诊断技术和网络技术、通讯技术相结合,形成一种新兴的监测诊断模式——远程监测诊断模式。它将监测诊断现场和诊断中心由网络联系起来。监测现场通过网络向诊断中心发出服务请求,诊断中心根据不同的请求做出响应。

故障诊断技术是现代化生产发展的产物,随着科学技术的进步和社会生产的需要得到了迅速发展,不仅形成了一个研究领域,而且基本形成了一个学科体系。

60年代末,美国国家宇航局创立美国机械故障预防小组,英国成立了机械保健中心。由于诊断技术所产生的巨大的经济效益,从而获得了极大的重视。

70年代初以来,随着计算机科学的发展,自动化装置的规模越来越大,投资也越来越高。某些微小故障若不能及时排除,就有可能造成巨大灾难。因而市场的迫切要求是促进故障诊断技术迅猛发展的主要动力<sup>[11]</sup>。

1971年,美国麻省理工学院 Beard 博士首先提出了用解析冗余代替硬件冗余,并通过系统自组织使系统闭环稳定,通过比较器的输出得到系统故障信息思想,标志着故障诊断技术的开端<sup>[12,13,14]</sup>。

70年代是控制系统故障诊断理论发展的初级阶段,在这一阶段提出了一些简单的故障算法,如检测滤波器、广义似然比、极大似然比等,故障诊断理论还不成熟,理论的应用实例很少。

80年代是故障诊断理论蓬勃发展的一个阶段,理论的实际应用也得到了发展。但是,这一阶段的故障诊断技术应用面比较狭窄,主要集中在航天、船舶、发电厂以及核电站等一些精密的专门领域,普通的工业控制领域涉及并不多。使用的故障诊断方法主要是基于观测器/滤波器方法或是系统辨识和参数估计方法<sup>[15,16,17]</sup>。

90年代以后,由于故障诊断理论的研究更加深入,各种方法相互渗透融合,理论应用领域也有了很大的扩展。尽管参数估计和观测器/滤波器方法仍然是使用最多的方法,但是近年来使用神经网络、模糊逻辑及组合的故障诊断方法在明显地增加。对于非线性系统的故障诊断也有了更多的研究。

### 1.2.2 故障诊断技术的主要方法

故障诊断技术发展至今,已经提出了大量的方法。按照国际故障诊断权威德国 Frank 教授的观点,所有的故障诊断方法可以划分为基于信号处理的方法、基于解析模型的方法和基于知识的方法<sup>[18,19]</sup>。当可以建立比较准确的被控过程的数学模型时,基于解析模型的方法是首选的方法;当可以得到被控对象的输入输出信号,但很难建立被控对象的解析数学模型时,可采用基于信号处理的方法;当很难建立被控对象的定量数学模型时,可以采用基于知识的方法<sup>[20]</sup>,其总体分类如图 1-1 所示。

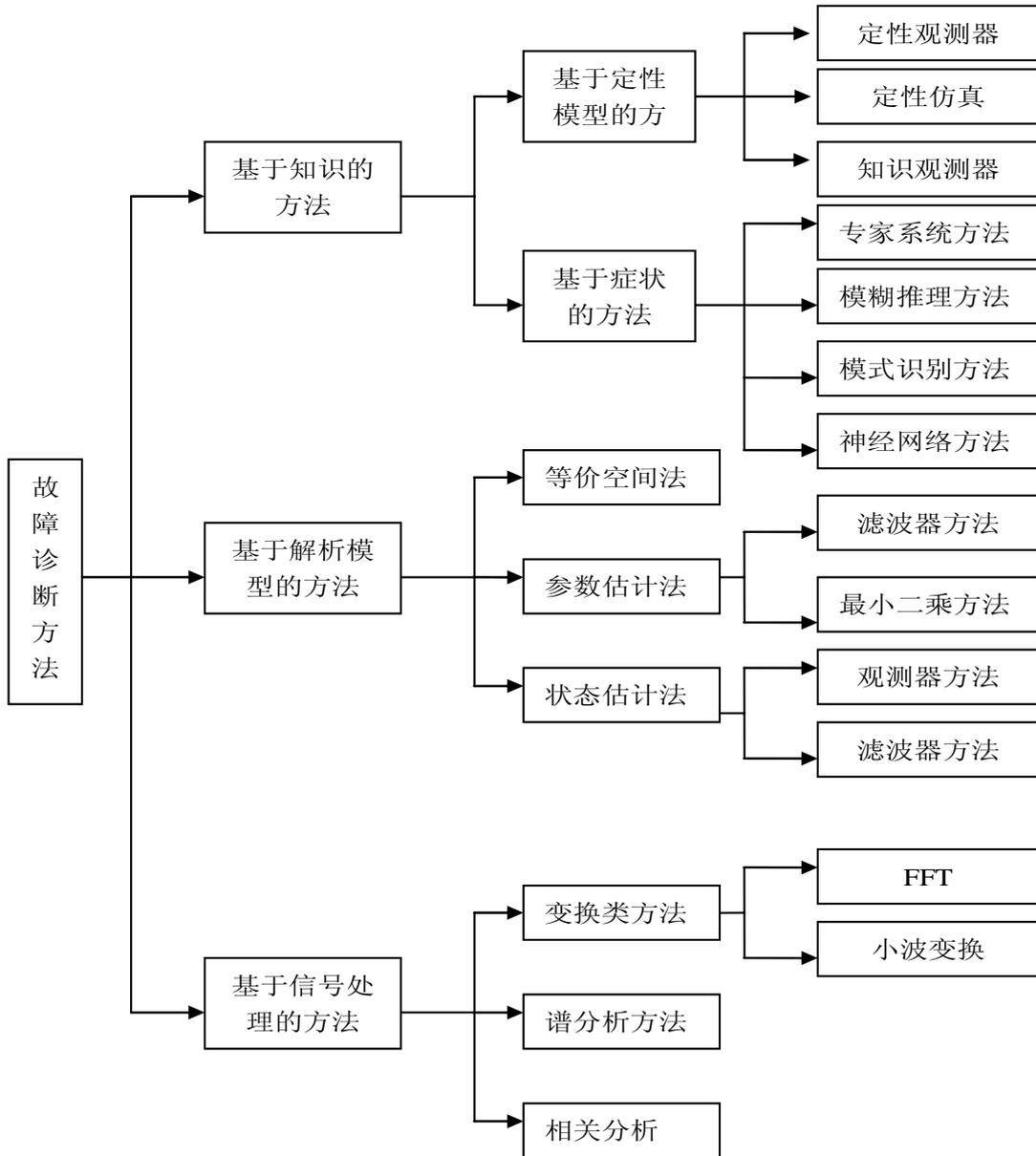


图 1-1 故障诊断方法示意图

(1) 基于信号处理的方法

基于信号处理的方法通常利用信号模型，如相关函数、频谱、自回归滑动平均等，直接分析可测信号，提取诸如方差、幅值、频率等特征值，从而检测故障的发生，近年来出现了一些新的基于信号处理的故障诊断方法，如基于小波变换的故障诊断方法。

小波变换是 80 年代后期发展起来的应用数学分支，最初由法国学者 Daubechies 和 Mallat 引入信号处理领域<sup>[21,22,23,24]</sup>，它具有许多优良特性。文献<sup>[25,26]</sup>给出了三种基于小波变换的故障诊断方法：

### ① 利用观测信号的奇异性进行故障诊断

动态系统的故障通常会导致系统的观测信号发生变化,若能采取一定的措施消除系统状态变化以外的因素的影响,直接利用连续小波变换检测观测信号的奇异点就可以检测出系统故障。

### ② 利用观测信号频率结构的变化进行故障诊断

振动系统的故障通常会导致系统观测信号的频率发生变化。若能采用一定的措施消除系统状态变化以外的因素对观测信号的影响,则利用离散正交小波变换分析观测信号的频率结构随时间的变化情况,就可以检测系统的故障。

### ③ 利用脉冲响应函数的小波变换进行故障诊断

Eykhoff 的连续系统脉冲响应辨识方法的基本思想是将系统脉冲响应函数的辨识转化为脉冲响应函数在一组正交函数基上的投影系数的辨识。若将 Eykhoff 方法中的正交函数基取为离散正交小波基,所得到的脉冲响应辨识方法除了保持原方法的有效性外,而且较基于传统正交函数基的 Eykhoff 方法,具有更强的跟踪参数变化的能力,辨识结果具有明确的频域物理意义。系统脉冲响应函数在最大尺度下的小波变换系数描述了它在大尺度下的概貌情况,完全可以代表其整体特性。而且通常这些小波变换系数中只有 2-3 个元素具有较大的模,其余元素的模都非常小。系统故障导致的系统脉冲响应函数的变化也必然反映在这少数几个小波变换系数的变化中。以系统的状态为参照,根据系统待检状态下辨识得到的这几个元素或其平均值随时间的变化情况就可以判断有无故障。由于这些元素或其平均值和系统的状态相对应,还可以利用它们在突变后的取值并结合系统的先验知识进行故障分离。基于小波变换的故障诊断方法无需对象的数学模型,且对于输入信号的要求较低,计算量不大,灵敏度高,克服噪声能力强,是一种很有前途的故障诊断方法。

## (2) 基于解析模型的方法

基于解析模型的方法又可以分为状态估计方法、等价空间方法和参数估计方法。这三种方法虽然是独立发展起来的,但它们彼此之间并不是孤立的,而是存在一定的关系。

### ① 状态估计方法

状态估计方法一直是研究的热点<sup>[27,28]</sup>,其基本思想是利用系统的定量模型和测量信号重建某一可测变量,将估计值与测量值之差作为残差,以检测和分离系统故障。在能够获得系统的精确数学模型的情况下,状态估计方法是最直接有效的方法。

### ② 等价空间方法

等价空间方法的基本思想就是利用系统的输入输出的实际测量值检验系统

数学模型的等价性（即一致性）以检测和分离故障。主要方法有：基于约束优化的等价方程方法、广义残差产生器方案、具有方向性的残差序列、基于近似扰动解耦的等价空间方法。

### ③参数估计方法

参数估计方法根据模型参数及相应的物理参数的变化来检测和分离故障<sup>[29]</sup>。与状态估计方法相比，参数估计方法更有利于故障的分离。该方法要求找出模型参数和物理参数之间的一一对应关系，且被控过程需充分激励。因此将参数估计方法与其它解析方法结合起来使用，可以获得更好的故障检测和分离性能。主要方法有：强跟踪滤波器方法、参数估计和观测器方法的结合、参数估计和等价空间方法的结合。

### （3）基于知识的方法

由于在实际情况中，常常无法获得对象的精确数学模型，这就大大限制了定量方法的使用范围。而基于知识的方法不需要对象的精确数学模型，因此是很有生命力的方法。基于知识的方法主要可以分为基于症状的方法和基于定性模型的方法。基于症状的方法主要有神经网络方法、专家系统方法、模糊推理方法和模式识别方法等；基于定性模型的方法有知识观测器方法等<sup>[30]</sup>。

#### ①神经网络方法

神经网络方法由于具有模拟任何连续非线性函数的能力和从样本学习的能力，非常适合应用于故障诊断系统，因而在故障诊断中得到了广泛的重视<sup>[31]</sup>。它具有自组织自学习能力，能克服传统专家系统当启发式规则未考虑到时就无法工作的缺陷。

#### ②模糊方法

模糊推理符合人类的自然思维过程，便于处理定性知识，成为故障诊断方法研究的一个热点。其基本原理为：设用一个集合  $Y$  定义系统中所有可能发生各种故障原因，由这些故障原因引起的各种症状，如温度、压力、湿度的波动等定义为一个集合  $X$ 。由于故障征兆是界限不明确的模糊集合，用传统的二值逻辑方法显然不合理，可选用确定隶属函数，用相应的隶属度来描述这些症状存在的倾向性。模糊诊断的方法就是通过某些症状的隶属度来求出各种故障原因的隶属度，以表征各故障存在的倾向性。

#### ③专家系统方法

该方法不依赖于系统的数学模型，而是根据人们长期的实践经验和大量的故障信息知识，设计出一套智能计算机程序，以此来解决复杂控制系统的故障诊断问题。建立专家系统来进行故障检测诊断，是一个比较先进的工程技术，它可汇集各种来源的关于故障的知识，并将可靠性专家和专业技术人员的知识形式化，

使一般工作人员借助于专家系统达到或接近专家的工作水平。

### 1.2.3 移动机器人故障诊断技术

关于机器人系统 FDD 研究的最早报道是美国 Minnesota 大学的计算机科学系。他们认为由于机器人运行在一个不完备的知识空间,因而需要为机器人系统配备一个检测及防止灾难事故的策略,认为要完全做到“自动故障恢复”是非常困难的,因为这要求关于环境、操作、资源等系统的完整信息,其给出了一种自动恢复系统的结构框架,基本思想是广泛使用知识库技术<sup>[32]</sup>。

受故障支配的移动机器人系统是一种典型的随机混合系统,不能由单个连续变化的状态方程来准确描述。不同故障模式以及正常模式之间结构及参数互不相同,其状态既可跳变也可以连续变化<sup>[33]</sup>。多模型方法是处理随机混合系统的一种典型方法,已被应用于结构化环境中移动机器人传感器故障诊断;粒子滤波器克服了多模型方法的某些不足,被 Verma 等选用为漫游车状态检测与故障诊断的技术;Murphy 等提出了一种用于移动机器人传感器信息融合系统的体系结构 SFX-EH(Sensor Fusion Effects Exception Handling);Lamine 提出一种时态模糊逻辑,用于描述监视知识,具有处理不确定性、模糊行为以及含噪声输入信息的能力;Aycard 利用二阶隐马尔可夫模型来辨识漫游车所处的环境状态,模型自动地在有标签的样本集上训练;Umesh 利用故障树分析(Fault Tree Analysis, FFA)技术对无人自主移动机器人故障进行诊断。

移动机器人故障检测与诊断(FDD)研究尚处于起步阶段,一般都局限于概念层,并没有给出实际有效的方法,大多停留在计算机仿真和实验阶段<sup>[1]</sup>。由于移动机器人作为一个复杂的系统,有很多子系统组成,而每一个子系统又由很多个模块组成。每个模块都有各自的故障机理,对其进行故障诊断的方法可能各不相同。所以,要针对每一个子系统,甚至是每一个模块,剖析故障现象和故障原因之间的关系,分别建立主要子系统故障诊断系统。实现对机器人工作状态的自主诊断,主要包括以下三方面<sup>[8]</sup>:

(1) 对传感器状态的自主诊断:在移动机器人的运行过程当中,环境对于传感器的动态影响是不可预测和不可忽略的事实,因此对移动机器人传感器状态的自主诊断对于实现机器人的安全航行、实现作业是十分必要的。由于在机器人的声、光视觉有相对独立的自适应处理系统,因此对机器人传感器的自主诊断主要针对非视觉传感器而言。

(2) 对安全状态的自主诊断:对安全状态的诊断主要包括,电源是否能维持正常工作、机器人系统温度是否正常、通讯是否畅通等。

(3) 对执行器状态的自主诊断:机器人的行为要靠执行器来实现,一旦执

行器发生某种程度的故障，机器人能够自主进行故障诊断，并采取相应的措施，是十分必要的。

研究故障诊断与容错控制技术，提高移动机器人的安全性和可靠性，已经成为未知环境中的移动机器人技术的迫切研究任务。

### 1.3 本文的工作

未知环境中移动机器人是一个强非线性参数不确定系统，且运行在未知的、复杂的环境下，一旦发生事故就可能造成巨大的财产损失。而传感器、机器人车载计算机、通讯网络发挥着感知环境、运动规划、数据传输的功能，如果出现故障将严重影响机器人的正常运行，从而影响其工作能力。采用状态监测与故障诊断技术，可以确保机器人在一种安全的状态下运行，以延长移动机器人的寿命。所以移动机器人状态检测与故障诊断是保障移动机器人的生存及成功完成作业任务的关键技术，具有很大的现实意义，是移动机器人智能技术中重要的一部分。

本文作为国家自然科学基金支持的“未知环境中移动机器人导航控制的理论与方法研究”中的一部分，着重于以下故障诊断系统设计中两个问题的研究：

(1) 状态信息获取：只有充分了解对象，才能对它进行故障诊断。对于一个实际的故障诊断应用系统的开发而言，机器人状态信息的获取是一个关键的问题。本文工作的首要是设计了一个移动机器人远程状态参数的获取子系统，采用基于无线网络的方式获取移动机器人车载的多种传感器参数，并对数据进行分析和存储，为后续的故障诊断研究提供了基础；此子系统同时提供一种可配置诊断模式，能较好的节约计算资源，为了能掌握移动机器人以前状态，还提供了诊断历史纪录查询等功能；对于移动机器人车载的工控机系统，有必要设计一子系统对其进行监视，因此，论文中基于 C/S 模式设计实现了一专门针对移动机器人车载工控机故障的监视系统，它能有效的获取车载的工控机的系统参数，并提供一种远程维护功能；远程客户端和机器人的无线通讯系统对于系统安全运行有着重要作用，因此，论文基于 ICMP 和 SNMP 设计了一网络性能监测子系统，它能获取网络延迟时间、网络流量、发送错误速率等网络性能参数，为无线网络系统安全提供了一定保障。

(2) 故障诊断与异常检测：对机器人工作状态的自主诊断中，传感器和安全状态的自主诊断在机器人安全运行中发挥着极为重要的作用，本论文对各种诊断方法进行了分析，采用了基于信号和知识的诊断方法，研究了基于小波变换算法的传感器诊断方法，利用编码器计算获得移动机器人角速率与光纤陀螺所测角速率的冗余关系，分析设计了一针对 MORCS-1 光纤陀螺突发性故障的诊断方法；对于在未知环境中运行的移动机器人，有可能出现的车轮打滑、受阻等情况，分

析设计了一个基于规则的车轮的运动状况诊断模块，它能有效监测车轮状态，并为移动机器人摆脱当前状态提供一种策略；同时分别设计实现了对车载电源、机器人系统温度、编码器等故障监测模块。

## 1.4 论文安排

立足于课题的研究背景，通过国内外研究现状的分析，结合以上的研究内容，本文将从以下章节展开：

- 第1章 介绍了本课题的来源、意义、应用领域等，综述了状态监测与故障诊断技术发展、主要方法等，并对机器人故障诊断的发展进行了分析，同时对于机器人故障诊断的主要内容进行了介绍，概况了本文的主要内容。
- 第2章 介绍了本课题所研究的智能移动机器人平台，分析了移动机器人的体系结构和软硬件平台，对于内部传感器这些主要的状态监测对象进行深入的分析。
- 第3章 详细地介绍了的状态监测系统设计的相关技术，其中包括了移动机器人的传感器数据的获取、车载计算机监视系统的设计和 network 状态参数获取系统的设计这三部分内容。
- 第4章 首先对连续小波变换在传感器故障检测中的应用作了仿真，然后利用编码器计算获得移动机器人角速率与光纤陀螺所测角速率的冗余关系，对于 MORCS-1 机器人光纤陀螺的故障进行了诊断分析。对于在未知环境中运行的移动机器人，有可能出现的车轮打滑、受阻等情况，设计了一个基于规则的车轮的运动状况诊断方法。在 MORCS-1 上进行了实验分析。
- 第5章 介绍了作者设计的移动机器人状态监测与诊断系统，列举了所采用的关键技术，主要算法和相关实现细节。
- 第6章 对于本论文的研究进行了总结，并对有待进一步研究的问题进行了分析和展望。

## 第二章 MORCS-1 体系结构

智能移动机器人是一类能够通过传感器感知环境和自身状态,实现在有障碍物的环境中面向目标的自主运动,从而完成一定作业功能的机器人系统。本文以中南大学智能系统与智能软件研究所自行设计研制的移动机器人MORCS-1作为研究对象,在本章中,主要介绍MORCS-1的体系结构及其软硬件组成,重点分析了车载内部传感器系统。

### 2.1 MORCS-1 移动机器人

MORCS-1移动机器人系统主要包含了移动机构、传感器系统、控制系统、通讯系统等几个部分<sup>[34]</sup>,采用开放式的分布式控制结构,利用低功耗集成工控机,保障了传感器信息处理与导航控制中复杂计算量的需求,其实物照片如2-1所示。



图 2-1 MORCS-1 实物照片

该移动机器人的总体性能指标如下:

- 车体尺寸: 长 80cm, 高 90cm, 宽 70cm, 重量 72kg。
- 最大直线速度: 0.6m/sec, 最大旋转速度 0.5rad/sec, 绕轴零半径自转。
- 越障性能: 5~6cm 台阶以及小于 25° 斜面。
- 车体驱动电机: 4 个步进电机, 36VDC。
- 传感器系统: 激光雷达与多目视觉系统。
- 车载计算机: 工控机系统 3~4 台, PIII800M~1G。

- 软件操作系统：基于 Win98/2000 环境，采用 Visual C++ 进行程序开发。
- 无线通讯系统：室内障碍环境通讯距离 50m，室外开阔环境 150~200m。
- 电源系统：可携带 2 组镍氢充电电池，每组包含一个 36VDC/13Ah 与 24VDC/13Ah 单元。
- 工作时间：2 小时（1 组电池）；3~4 小时（带扩展电池）。

## 2.2 MORCS-1 移动机器人硬件平台

### 2.2.1 硬件平台总体结构

硬件系统是移动机器人系统的物理基础，承担着环境感知、运动控制、通讯等功能，因此，本机器人系统采用了集成化的工控机底板，可以容纳 4 个独立的计算机系统，每个系统可以安装 1 个 CPU 板、2 个 PCI 插槽功能板和 2 个 ISA 扩展功能板。机器人系统采用镍氢电池供电，考虑到供电能力的限制，系统采用实际配置为 3 个采用低功耗 VIA-C3-800MHz CPU、内存为 256M 的 ROCKY-C800EV 控制主板，分别应用于运动控制、激光雷达信息处理、多目视觉信息处理及决策支持等系统。系统采用 Windows 98/2000 操作系统，采用 Visual C++ 作为软件开发工具。通过一个集线器将车载计算机组织成为一个局域网控制系统。系统的硬件结构见图 2-2 所示。

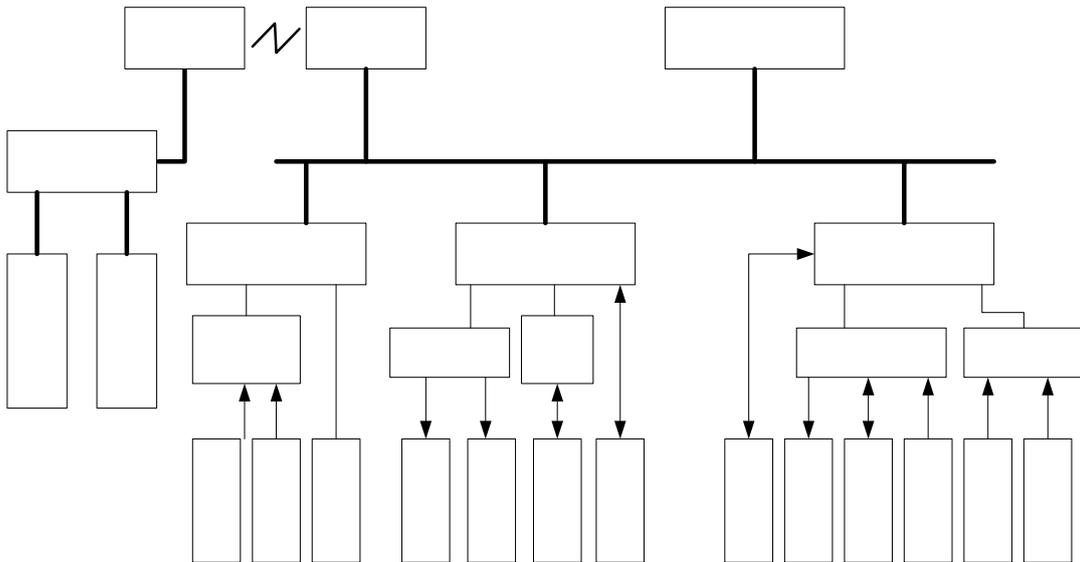


图 2-2 移动机器人体系结构图

### 2.2.2 MORCS-1 系统组成

MORCS-1 包括了几大子系统，各子系统又由不同的传感器组成。

#### (1) MORCS-1 系统介绍

MORCS-1 包括了运动控制系统、激光雷达信息处理系统、视觉处理系统等子系统：

##### ① 运动控制系统

工控机 IPC0 作为运动控制系统，操作系统基于 Windows 98 平台。CPU 主板上集成了 2 个 RS232 串行通讯接口，运动控制系统主要的功能是计算并反馈机器人车体的实时运动姿态，同时实现动态避障与跟踪控制。主要的控制板卡为基于 PCI 总线的 ADT850 步进电机控制卡与 A/D 采集卡 PCI1710。ADT850 运动控制卡是基于 PCI 总线的 32 位四通道步进电机控制卡，对移动机器人的左、右摇杆上的四个驱动电机进行控制，实现车体的运动。PCI1710 采集卡输入移动机器人的模拟量信号并实现 A/D 转换获得数字量，模拟信号包括了左（右）摇杆的转动角度、36VDC 电源信号、24VDC 电源信号、电机电流检测（4 路）等共计 8 个模拟量通道。

##### ② 激光雷达信息处理系统

工控机 IPC1 作为 3-D 环境感知信息处理系统，运行在 Win2000 操作系统下，存储器为 40G 硬盘。安装步进电机控制卡 PCL839，实现对激光雷达转动云台的控制。安装一块基于 PCI 总线的高速 RS-422 接口卡，以 500K 波特率的通讯速度接受激光雷达信息并进行局部 3-D 环境平坦性分析，区分环境中的可行区域与障碍区域，建立局部地图数据库。传感器转动云台为机器人的“头部”，由 2 个精密的电控旋转台实现水平方向  $\pm 150^\circ$ 、俯仰方向  $-50 \sim +15^\circ$  的扫描运动。水平转台最大转速为  $16^\circ / \text{sec}$ ，俯仰最大转速为  $8^\circ / \text{sec}$ ，重复转动误差小于  $\pm 0.01^\circ$ 。

##### ③ 视觉处理系统

工控机 IPC2 作为视觉信息处理系统，工作在 Win2000 操作环境，存储器为 40G。CPU 采用 PIII，1GHZ。安装一块多通道的图像采集卡，对全局视觉信息与前置局部视觉信息进行处理，并建立视觉特征数据库。视觉计算机系统通过 RS-232 接口控制主动视觉的云台。

##### ④ 决策支持系统

该系统主要进行非实时、大计算量的智能计算，例如：全局定位、环境建模与全局路径规划等高层智能任务。该系统没有特别的控制板卡，系统的功能由多个软件 Agent 实现。

### ⑤通讯与监控系统

控制系统的通讯包括移动机器人车载控制系统与实验室监控系统间的通讯以及车载系统中各个控制计算机之间的通讯。车载控制系统配备了一个集线器 (Hub), 把各个计算机系统组织成一个基于局域网的分布式控制系统, 它们之间可以通过 TCP/IP 协议进行通讯。光纤陀螺仪与倾角仪、主动视觉云台等智能仪器、设备均具有 RS232 接口, 与各自的控制主机实现串行通讯。车载局域网与实验室局域网之间通过无线网桥连接成一个网络控制系统。无线网桥在开阔地带的通讯距离为 150~200m, 通讯带宽为 1M~10M 波特率。

#### (2) 内部传感器

移动机器人系统中的内部传感器对移动机器人的定位发挥着非常重要的作用, 在 MORCS-1 中, 主要包括了以下几种内部传感器:

##### ①光电编码器

MORCS-1 中编码器采用 360 脉冲/转的增量式光电编码器, 其具有相差为 90 度的双路脉冲输出, 可以通过双路脉冲的相位变化来判断旋转方向, 编码器安装在驱动电机的轴伸上, 直接测量电机的转动。通过编码器测出的脉冲数, 可以求出左右轮转过的距离, 从而计算出车体的位置坐标和运动方向。

##### ②光纤陀螺仪

MORCS-1 采用的 E.core 1000RD 光纤陀螺仪作为航向测量传感器, 该型陀螺仪是 KVH 公司生产用于惯性导航的单轴光纤角速度传感器, 采用干涉测量方法, 其工作原理是应用 Sagnac 干涉效应, 即利用光纤绕制成环柱型光路, 随光纤的转动产生反向旋转的两路光束, 通过检测他们的相位差既可以算出光纤陀螺的角速度。光纤陀螺由于光路的闭合避免了空气的扰动, 并且不受结构空间限制可以组成千米数量级长度的干涉仪, 因此有利于提高测量灵敏度。

##### ③角度电位器

MORCS-1 中, 利用高精度电位器来采集摇臂相对于车体的姿态信息, 将电位器调节轴固定在摇臂的转轴上, 当摇臂转动时, 带动电位器调节轴转动, 从而改变电位器的输出电压, 电位器输出电压经过电压放大电路放大十倍后, 再通过 PCI1710 数据采集卡 A/D 转换后送入 IPC0。

##### ④倾角传感器

移动机器人在起伏地形下进行航迹推算时, 除了利用光纤陀螺进行定位外, 还需要获得机器人在运行时的俯仰与横滚的姿态参数。本系统采用 Crossbow 公司生产的 CXTILT02E/02EC 传感器, 其具有体积小、重量轻、精度较高、提供标准串行接口等特点, 能够测量俯仰与横滚方向的倾斜角, 精度可以达到 0.1 度, 能较好的满足系统的精度要求。在 MORCS-1 中, 将倾角传感器直接接在工控机



### (3) 通讯中枢/激光雷达模块 (PCL\_Test.exe)

主要功能是 LPC02 即 IPC1 是通讯的中枢, 负责接收和发送数据包。为与之相连的每个模块都设计一个缓冲区 (或称邮箱)。接收每个访问者发送来的数据包, 判断是否发给自己, 是则处理, 否则将其存入相应接收者的邮箱。同时, 把该访问者相应邮箱的内容发送给访问者, 即 IPC1 不主动将各模块的邮箱内容发给各模块。

### (4) 慎思规划模块 (AdvancePlanning.exe)

在这样几种情况下, 将启动慎思模块重新规划新的路径。第一种情况, 运动开始时, 当设定机器人最终目标后启动慎思模块。第二种情况, 当下点目标和机器人的方向角大于某个范围时启动慎思规划。第三种情况, 当底层如反应式规划控制模块找不到可通过的方向时启动慎思规划。在启动慎思模块前, 在协调层中会首先把由激光雷达探测到的环境保存到一个地图中, 在慎思规划前首先读这个地图。具体工作步骤: (a)对地图进行扩张; (b)获得路径; (c)进行优化处理; (d)把路径发给协调层。

### (5) 状态监测与诊断模块 (Robot Doctor.exe)

主要功能是完成对 MORCS-1 系统的状态监测与故障诊断, 该模块提供了对车载的传感器的故障诊断、车轮状态监测、车体温度诊断、网络性能监测等功能, 并提供传感器参数实时保存和移动机器人诊断信息保存等功能, 这部分可以部署在远程任意一台计算机上。

## 2.4 小结

本章首先介绍了中南大学智能系统与智能软件研究所研制的 MORCS-1 平台性能, 分析了移动机器人系统的体系结构, 然后按照机器人中硬件、软件顺序介绍了各系统的组成, 重点介绍了移动机器人的内部传感器系统, 明确了故障诊断的主要对象。

## 第三章 移动机器人状态监测系统设计

移动机器人是一个复杂的系统，由很多子系统组成，而每一个子系统又由很多模块组成。本章针对MORCS-1中不同子系统，介绍了移动机器人状态监测系统的设计。其中包括了移动机器人的传感器数据的获取、车载工控机远程监视和网络状态参数获取系统的设计三部分内容。

### 3.1 状态监测系统设计

#### 3.1.1 系统设计原则

移动机器人状态监测系统由不同的模块组成，除了要能正确地实现系统功能外，还应该具有以下几个特点：

##### (1) 针对性

在充分分析了移动机器人运行特点和常见故障的基础上，有针对性地进行了传感器类型选择、分析方法的选用、软件模块的划分等方面的工作，从而保证了系统在运行时的实用性。

##### (2) 可靠性

信号的可靠性是决定移动机器人状态监测与故障诊断系统的功能能否真正实现的关键因素。因此在系统设计初期，就应重视系统的可靠性，软件上充分考虑数据的存储、传输、特征提取算法，软件系统内存冲突管理方面的可靠性。

##### (3) 可扩展性

系统软件方案上都充分考虑了将来需求的发展，因此系统采用了基于工控机的数据采集系统及上、下位机的网络通信的结构形式。

##### (4) 方便于用户的GUI设计

在系统设计阶段，考虑让方便、简单地使用该系统，因此研制成功的系统直观、友好，使用操作简单。

#### 3.1.2 功能规划

对于移动机器人这种集电子、机械、计算机等多种技术于一体的智能系统，对其全面的监测是十分困难的，因此，根据系统按软硬件的不同划分，有选择性地对机器人的硬件、软件系统建立监测子系统，同时，对于远程监控端与移动机器人系统的通讯系统建立专门的监测子系统，监测系统用例图如3-1所示，通过

对系统中这三部分的监测，就能获系统较多的信息，为后续的移动机器人故障诊断方法研究提供了基础。

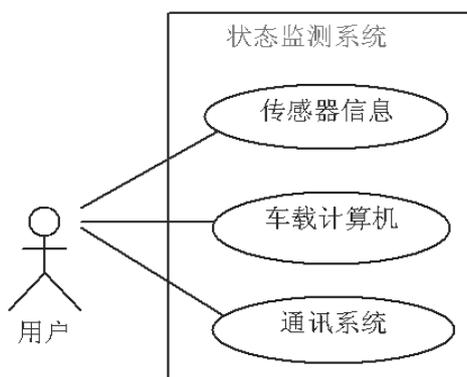


图 3-1 监测系统用例图

在MORCS-1中采用了多种传感器，除了利用这些传感器获取的数据进行机器人定位、避障外，亦可以对系统进行监测，因此，系统应能够获取机器人传感器参数、进行诊断分析、实时显示、数据保存等功能，同时当发现有异常时能够发出报警。

对于车载工控机系统和车载的软件Agent，除了进行大量的软件测试外，对其监视也有一定的必要性，当发现异常时，可以通过其实现远程维护，达到对车载工控机的控制，因此，设计了一针对车载计算机的远程监视子系统。

MORCS-1系统采用基于IEEE802.11b的无线网桥连接机器人与远程监控端，它能带来极大的方便，有利于室内、室外实验的开展，但无线网络存在差错率高、易断线等缺陷，因而设计一子系统对其进行监测，及时获取网络的通断情况和网络性能参数，能更好的维护移动机器人的运行安全。

### 3.2 传感器状态参数获取设计

移动机器人所用的传感器很多，根据使用目的可分为内部传感器和外部传感器两大类。内部传感器用于感知机器人本体的状态、位姿，是实现控制、伺服动作必不可少的装置。外部传感器则是感知外部环境的特征，使机器人能及时了解工作环境，并视其情况来调整自己的决策，以提高机器人的适应性和智能化水平。由于外部环境的复杂性和不确定性，外部传感器所体现的智能水平较高<sup>[35]</sup>。在MORCS-1系统中，内部传感器主要包括了KVH公司的E.Core RD1100光纤陀螺、360 脉冲/转的增量式光电编码器，Crossbow 公司生产的CXTILT02E/02EC 倾角传感器、高精度电位器等。外部传感器则采用了了SICK 公司生产的LMS291 激光雷达测距仪、SONY 公司生产的EVD100 摄像头。用编码器来测量移动机器

人的线速度和位移距离，用光纤陀螺仪测量移动机器人的角速度和航向角，用倾角传感器测量移动机器人车体的俯仰角和横滚角，用高精度电位器测量各摇臂与移动机器人车体底盘的夹角；同时，使用激光雷达和摄像头作为移动机器人的外部环境感知传感器，用于进行外部环境的感知。

在MORCS-1系统采用了一种基于Agent实现智能系统的方式，系统中主要包括了控制Agent、局部规划Agent、激光雷达Agent、动态慎思规划Agent、区域环境知识库、全局建模与规划Agent、人机接口Agent 以及包括视觉系统Agent。在各个Agent之间的通信采用基于进程间通信方式，利用消息队列、管道、邮槽、网络通信等方式实现。因此，在机器人传感器参数获取系统，采用基于Agent方式，通过Agent之间通信方式获取机器人传感器参数。

在MORCS-1系统中，激光雷达Agent中集中了除CCD摄像头外的所有传感器的数据，其提供了一种类似于数据服务器的功能，当其他Agent需要相应的数据时，可以通过发送请求信号给激光雷达Agent，激光雷达获取了请求对象的数据类型后，形成相对应的数据报返回到请求数据的Agent，当Agent获得所需要的数据包后解包就得到所要求的数据，传感器参数获取系统的序列图如3-2所示。

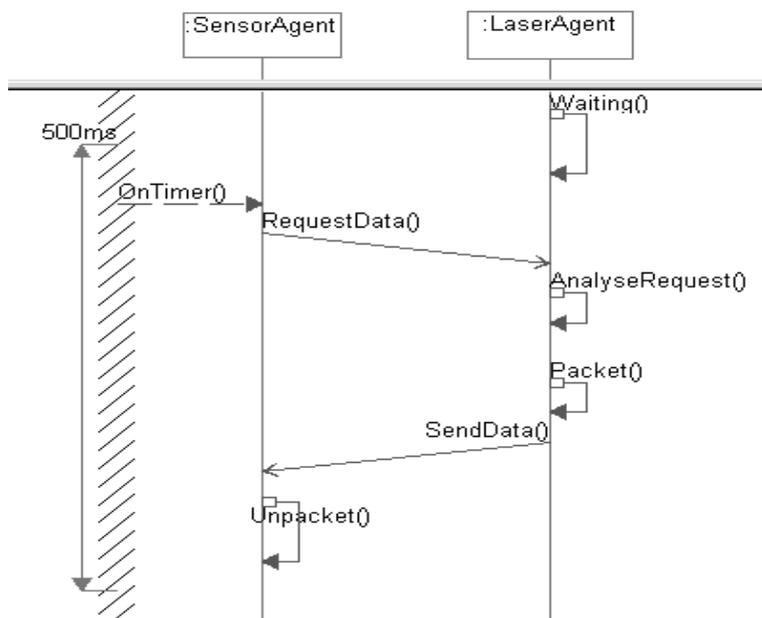


图 3-2 传感器参数获取系统序列图

在MORCS-1系统中，激光雷达Agent部署在机器人的工控机IPC1上，传感器状态监测Agent通过无线网桥与机器人的车载局域网相连，从而实现了一种基于无线网络的机器人远程状态监测系统，系统采用TCP协议，在传感器状态监测Agent和激光雷达Agent之间建立点到点的通信信道，提供面向连接、可靠的数据传输服务。两部分程序都使用C++编写，采用基于Windows Sockets<sup>[36]</sup>的编程方式。

为了保证实验采样数据可靠、有序的被传送到诊断系统Agent，通信程序使用流式Socket，提供了具有流控的双工传输功能，保证了传感器参数数据传输的有序、可靠。

监测系统程序中的Socket工作在同步阻塞机制下<sup>[37,38]</sup>，而在调用Socket的一些方法时，特别当Socket工作在同步阻塞模式时，操作常会阻塞。例如从Socket中读取数据时，其read方法在不同的输入流中容易发生阻塞——即如果没有数据就等待。阻塞现象发生时，主线程将无法响应任何其他消息。所以，为了避免这种情况的发生，主线程在执行含有有可能阻塞的方法的任务时，就动态创建一个新的辅助线程，专门处理该任务。而主线程把任务移交给辅助线程后，继续往下执行其他操作。在本系统中，为了克服传感器状态监测系统Agent和激光雷达Agent的通信阻塞问题，在通信程序中创建了数据获取、保存、分析的读写辅助线程，通信中的所有数据保存任务都交付给辅助线程处理。

在基于TCP/IP的面向连接的网络通信中，传感器状态监测Agent和激光雷达Agent之间传送的是有序可靠的但无实际意义的字节流，所以必要在传输层TCP上定义自己的高层协议<sup>[39,40]</sup>，设计帧结构，将字节流变成有意义的信息。

本系统中的信息帧主要可分为以下两种类型：

#### (1) 数据包请求帧

它是指传感器状态监测系统Agent向激光雷达Agent所发送的数据需求帧，是一种不定长的数据帧，其中包括了发送者、请求对象、指令类型等信息，其具体的结构如图3-3所示：

RecLen	RecName	SenderLen	SenderName	InstruType	InstruNO	...
--------	---------	-----------	------------	------------	----------	-----

图 3-3 数据包请求帧

RecLen: 接收信息包Agent名字长度

RecName: 接收信息包Agent的名字

SenderLen: 发送信息包Agent名字长度

SenderName: 发送信息包Agent名字

InstruType: 获取数据指令类型

InstruNO: 获取数据指令参数的个数

其中RecLen、RecName、SenderLen、SenderName都是不定长度的，而InstruType的长度为5个字节，InstruNO为1个字节，后面的为对应的指令代码。

#### (2) 返回的状态参数帧

它是指激光雷达Agent在接收到数据请求帧后，返回给请求Agent的数据帧，

其中包括了帧头部格式、传感器参数结构体、激光雷达数据结构体等信息，其具体的结构如图 3-4 所示：

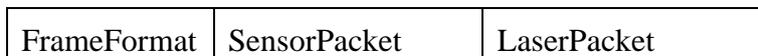


图 3-4 数据返回帧

**FrameFormat:** 整个数据帧的头部

**SensorPacket:** 传感器参数包

**LaserPacket:** 激光雷达数据包

传感器参数包括了车载的除激光雷达和摄像头外的所有数据，按不同的传感器可以分为以下几个部分<sup>[41]</sup>：

①机器人姿态：主要包括了机器人位置、方向角、激光雷达位置、车轮速度等信息，其具体的参数和变量如表 3-1 所示：

表 3-1 移动机器人各种姿态及对应变量

机器人姿态	变量名称
机器人 X 坐标	m_status.x
机器人 Y 坐标	m_status.y
机器人 Z 坐标	m_status.z
机器人方向角（由编码器获得）	m_status.sita1
激光雷达水平转角	m_status.radar_sita
激光雷达俯仰角	m_status.radar_phi
机器人左前轮速度	m_motion_para.left_front_speed
机器人左后轮速度	m_motion_para.left_back_speed
机器人右前轮速度	m_motion_para.right_front_speed
机器人右后轮速度	m_motion_para.right_back_speed
机器人线速度	m_motion_para.speed
机器人角速度（由编码器获得）	m_motion_para.omega1

②光纤陀螺参数：光纤陀螺提供给移动机器人角速度信息，对角速度对时间

的积分而获得机器人的航向角，这部分信息包括了机器人角速度、航向角、陀螺仪温度信息，其结构如表 3-2 所示：

表 3-2 光纤陀螺的相关参数和变量

参数名称	变量名称
机器人角速度	m_motion_para.omiga2
机器人航向角	m_status.sita2
光纤陀螺仪温度	m_status.temperature

③倾角传感器数据：提供了移动机器人车体的俯仰角和横滚角的信息，其参数和变量如表 3-3 所示：

表 3-3 倾角传感器的相关参数和变量

参数名称	变量名称
机器人俯仰角	m_status.arfa
机器人横滚角	m_status.phi

④摇臂姿态及电压数据：其中包括了由高精度电位器传感器所采集的移动机器人三个摇臂姿态信息，还包括了为移动机器人车载工控机和部分传感器供电的 24 伏直流电压和驱动步进电机的 36 伏直流电压信号两部分数据，其参数和变量如表 3-4 所示：

表 3-4 摇臂姿态相关参数和变量

参数名称	变量名称
机器人左摇臂夹角	m_status.beta1
机器人右摇臂夹角	m_status.beta2
机器人后摇臂夹角	m_status.beta3
机器人 24V 电源电压值	m_status.voltage24
机器人 36V 电源电压值	m_status.voltage36

激光雷达 LMS291 的数据由于测量模式的不同，因而有不同的数据量，其数据的个数与水平视场间隔如表 3-5 所示：

表 3-5 水平视场的间隔与数据个数

水平视场	间隔角	数据个数
100	0.25	401
100	0.5	201
100	1	101
180	0.5	361
180	1	181

在实际的使用过程中，采用了水平视场 180，间隔角为 0.5，数据个数为 361 这种工作方式，因此，移动机器人的激光雷达数据个数为 361 个。

### 3.3 车载计算机远程监视设计

在 MORCS-1 机器人系统中有 3 台工控机系统，分别承担着运动控制，环境感知，机器视觉方面的任务，运行 Windows 操作系统，同时各个计算机系统中有多个软件 Agent，通过协调实现系统功能，如在运动控制系统中有 Adt850 Agent、Local\_Planning\_Agent 等，由于操作系统或是应用软件都有可能因为设计或其他原因而产生故障，因此对于车载计算机的远程控制就显示出重要性，利用远程控制软件对车载计算机系统进行维护，能较好的解决被控系统出现的问题，延长系统的使用寿命。

对于远程监视的定义，一般认为远程控制指管理人员在远程通过网络连接到目标计算机，通过本地计算机对远程计算机进行管理和维护的行为<sup>[42,43]</sup>。而远程监视软件实际上是一种客户/服务器程序，服务器程序安装在被控制的计算机端，客户程序安装在控制端。在客户端和服务器端都安装成功后，客户端在网络上搜索到已经安装了服务器的远程计算机，然后客户端就发送指令获得服务器端的连接指令，两台 PC 建立了连接，就可以通过网络的互连协议 TCP/IP 进行远程控制。当操作者使用主控方电脑控制被控端电脑时，可以实现如同对本机一样的操作。

在远程控制系统中，按照 C/S 模式可以分为服务器和客户端两部分<sup>[44]</sup>，客户端程序通过网络连接服务器程序，同时在客户端创建服务器端界面的窗口程序，发送客户端创建好的窗口的大小和需要的位图格式等信息，服务器端在收到客户端的监视窗口大小和格式后获取相应的屏幕窗口，然后通过网络传回到客户

端，而客户端同时截获系统的鼠标运动位置，将其发送到服务器端程序，服务器在接受到鼠标运动指令后对系统产生相应的动作，把服务器端的桌面信息发送回客户端，因此，在客户端就可以获得远端计算机的桌面信息，同时，控制端的鼠标信息也能反映到远端的服务器上，这样就可以远程的维护车载计算机，图 3-5 就是系统的序列图。

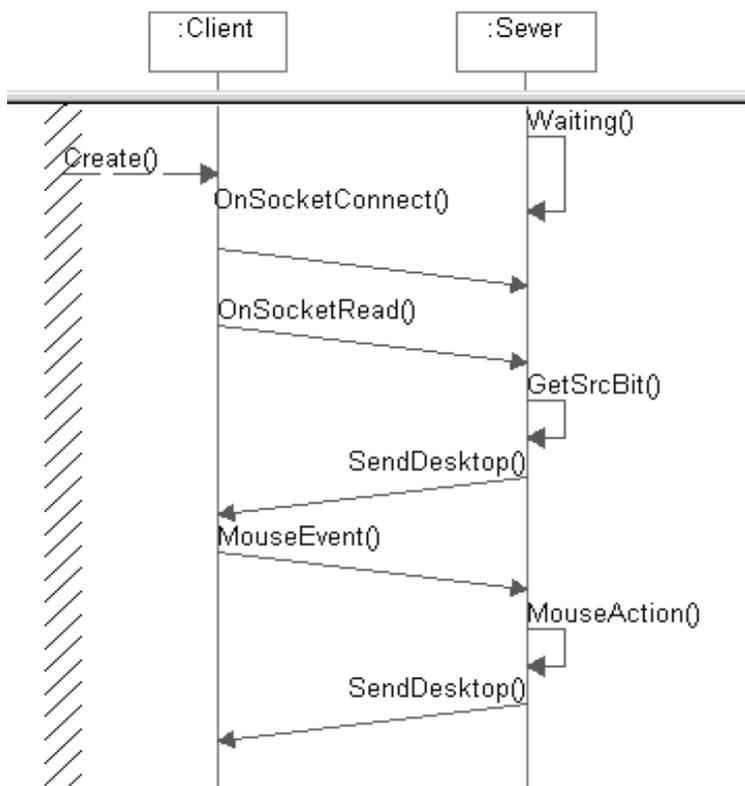


图 3-5 远程工控机监视序列图

在设计中，服务器端采用 WSAAsyncSelect 模型，利用这种模型，应用程序可在一个套接字上接收以 Windows 消息为基础的网络事件通知。该模型的实现方法是通过调用 WSAAsyncSelect 函数自动将套接字设置为非阻塞模式，并向 Winsock DLL 注册一个或多个感兴趣的网络事件，并提供一个通知时使用的窗口句柄，当注册的网络事件发生时，对应的窗口将收到一个基于消息的通知。这种方式有开销小、扩展性好的优点<sup>[45]</sup>，整个工控机远程控制系统中，服务器首先在一个端口上等待客户端的连接请求，当接收客户端连接请求后，再根据客户端的不同请求做出不同的反应，如获取本工控机的桌面并发送到客户端或是响应客户端的鼠标动作并做出相应的反应，为了更好说明系统的流程，把服务器端和客户端程序的流程图列举如图 3-6 和图 3-7 所示。

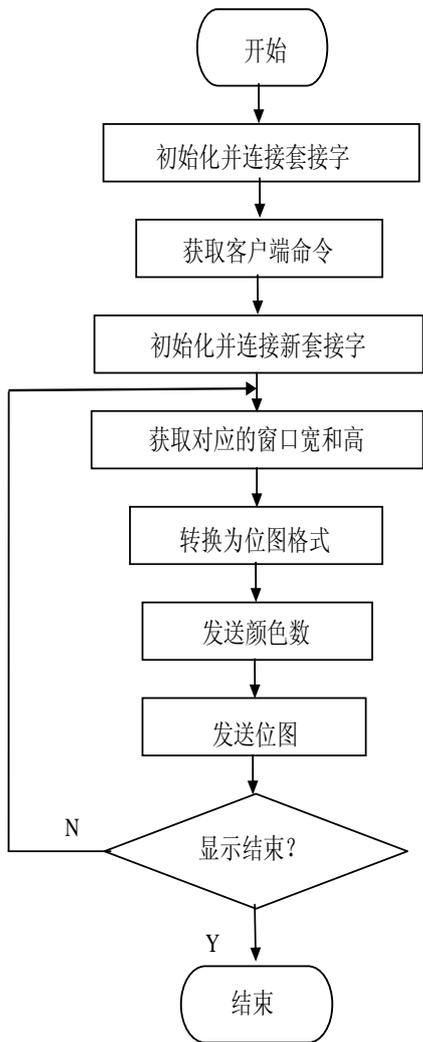


图 3-6 服务器端流程图

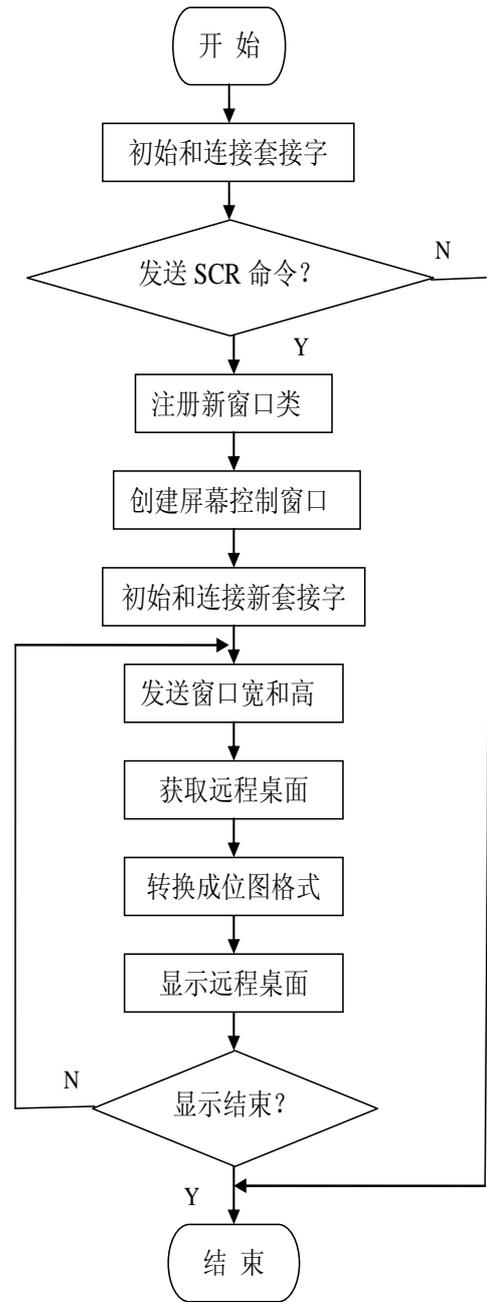


图 3-7 客户端流程图

由于车载计算机远程控制系统有别于机器人视频监控系统，且服务器运行端的工控机还承担着其他任务，所以系统实现时没有对原始的桌面的图像采取压缩，但是可以通过改变服务器端获取桌面的颜色数和远程监视端程序的窗口大小来减少网络传输的数据量，当以 *interval* 秒间隔的采样速度获取桌面时，数据量的大小为：

$$BPS = \frac{Width * Length * ColorNum}{interval} \quad (3-1)$$

其中 *Width*、*Length* 分别为监视端的窗口宽度与长度，*ColorNum* 为颜色数，当

Width=300, Length=200, ColorNum=4, interval=1s 时, 数据量为 0.24 M, 且监视端最小化后数据量可以减少到零, 因此能维护远端系统的前提下, 有效的减少数据量传输, 减轻对网络的压力。图 3-8 为在机器人远程监视端获得的移动机器人车载工控机 IPC1 的一个截图。在机器人野外试验时, 通过这个系统, 可以远程解决车载软件系统出现的一些故障, 达到对系统维护的目的。

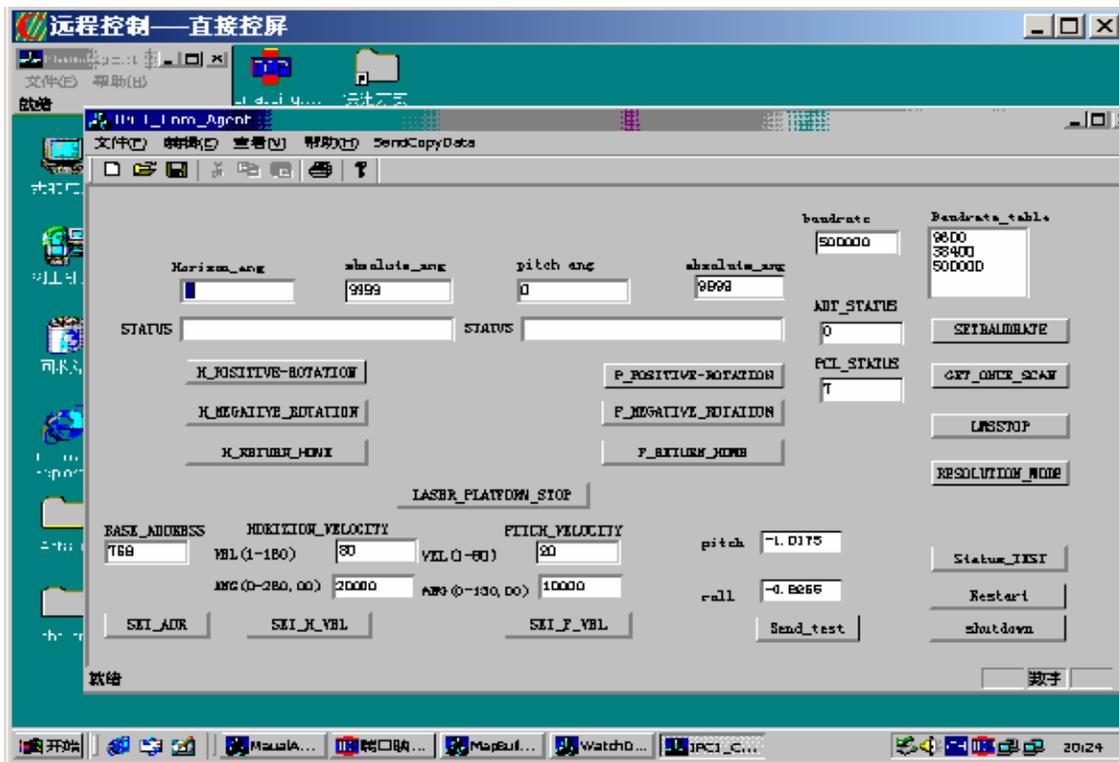


图 3-8 MORCS-1 上 IPC1 远程桌面

### 3.4 网络性能参数获取设计

MORCS-1 机器人的车载局域网与实验室的局域网之间通过无线网桥连接成一个网络控制系统, 系统的拓扑结构如图 3-9 所示, 无线通讯系统基于 UWB-9010 无线网桥方式, 无线网桥能根据实际的信道的情况, 自动的对网络传输速率进行调整 (11Mbps、5.5Mbps、2Mbps、1Mbps), 在开阔地带, UWB-9010 能提供的通讯距离为 150-200m。

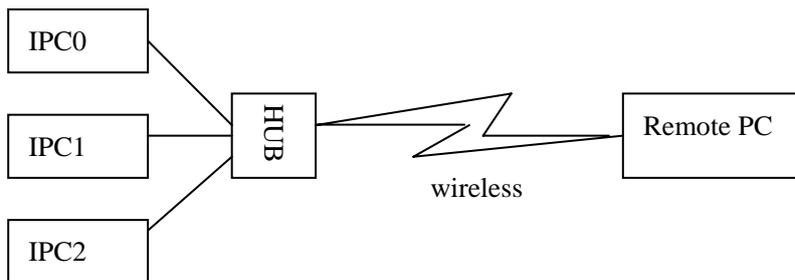


图 3-9 系统拓扑图

无线网络能给广大用户带来了方便,但是无线网络存在差错率高、易断线等缺陷。无线网络的差错率在  $10^{-5}$ - $10^{-1}$  之间,而且随着天气、地形及电磁场等因素变化。相比之下有线网络的差错率低,在  $10^{-8}$ - $10^{-6}$  之间,故有线网络的数据重传一般认为主要是由拥塞引起的数据丢失所致,而无线网络中的数据重传更多是由于传输差错及无线基站处的拥塞所致,所以无线网络中的传输控制比有线网络更为复杂。

在实验室的局域网与移动机器人车载局域网的通信编程中,对网络性能参数(如连通性、延迟、数据包传输速率、网络流量等)的监测成为了网络故障分析的关键<sup>[46]</sup>。在最早的 TCP/IP 网络 ARPANET 的研究实验过程中,人们采用实用的互联网信息控制协议(ICMP) 对其进行简单有效地管理。随着对网络管理工具需求的增长,1988 年,Internet 体系结构委员会开发了 SNMP。1990 年 5 月,Internet 工程任务组发布了 SNMP 系列协议,由于 SNMP 简单实用的优点,SNMP 很快就成为 INTERNET 事实上的网络管理协议标准,后续又发布了 SNMP V2 和 SNMP V3 。

目前网络性能监测按采集流量数据的方法可以分为两种方式<sup>[47]</sup>:

(1) 主动方式:它是指监测者主动发包去探测网络设备的运行情况,如基于 ICMP 协议的 Ping,该命令引发 IP 层发送一个类型是 Echo-request 的 ICMP 包,而目标方收到这个包之后,将源地址与目的地址进行交换,回应一个 Echo-reply 类型的 ICMP 封包给查询端以确定连线的可行性。

(2) 被动方式:它是指监测者被动地采集网络中现有的标志性数据,用以了解网络设备的运行情况,这种方法的监测是在监测点采集真实的网络数据包并进行统计的。

为了能较全面的获得网络的性能,在 MORCS-1 通讯系统监测中,主要基于主动方式与被动方式结合的方法,广泛的采集网络的连通性、延时、丢包、网络流量变化。

### 3.4.1 基于 ICMP 的网络延迟的获取

延迟定义为:输入帧的最后一位到达输入端口和输出帧的第一位出现在输出端口的时间间隔,即 LIFO 延迟。延迟值的大小反映了网络的拥塞等级,延迟有两种定义方式:

(1) 单向延迟:仅仅测量数据包从发送端到接收端的过程。但这种测量有一个问题,就是发送端和接收端的时钟同步问题。解决这个问题一般采用 GPS。

(2) 双向延迟:与单向延迟相比,双向延迟则较简单。由于时钟同步问题,节点间的网络延迟很难精确测量,一般都是测 Round trip Time(RTT)。RTT 指的

是一个包被发送到网络上，它从发送端到接收端后又回到接收端所经历的时间。RTT 参数的影响因素有：点到点的距离、标准队列的长度和拥塞等级。长期的 RTT 数据可用来做趋势分析和数据关联。RTT 的变化通常意味着配置的变化和拥塞等级的变化。

ICMP 属于 TCP/IP 协议中的一个协议，主要用来在网络节点之间发送与网络有关的问题通知，比如可达性、拥塞情况、数据包丢失情况等，因此对 ICMP 数据包的监视在网络故障检测中是最适应的方法。

在 MORCS-1 系统中，由于实验室局域网与机器人之间只经过无线网桥的连接，不存在路由路径不同的问题，所以采用双向延迟测量方法，这使得可以不用 GPS 等设备。直接利用 ICMP 包的方式测试无线网络的双向延迟，该方式利用 IP 层发送一个类型是 Echo-request 的 ICMP 包，而目的方收到这个包之后，将源地址与目的地址进行交换，回应一个 Echo-reply 类型的 ICMP 封包给查询端以确定连线的可行性，此时，可以记录下发送 ICMP 包时的时间和接收到回复包的时间，两者的差值就是所需要的往返网络延时时间。

$$RTT = T_{send} - T_{receive} \quad (3-2)$$

ICMP 报文封装在 IP 数据报的数据部分进行传输，ICMP 消息以基本 IP 头发送，IP 层一旦收到 ICMP 报文，就交给 ICMP 模块进行处理。数据的第一个字节是 ICMP 类型域；此域的值决定了其余数据的格式。任何标记为“未使用”的域都是为以后的扩展保留的，在传送过程中必须全部是 0。

ICMP 消息类型有几种，其中类型 8 为回送请求，类型 0 为回送响应。测量中只需要往返的时间，所以只需要发送回送响应消息的数据包，其格式介绍如下：

IP 域：

地址：回送消息的源地址是回送响应消息的目的地址。若要形成一个回送响应消息，应该将源和目的地址交换，将类型代码更改为 0，重新计算机校验码。

ICMP 域：

• 类型：

8 代表回送消息；

0 代表回送响应消息。

• 代码： 0

• 校验码： 16 位数据（从 ICMP 类型开始）的反码和再取反而得。为计算校验码，校验码域应该为零。这些零在以后会被校验码取代。

• 标识符：如果代码 =0，帮助匹配回送和回送响应的代码可以为 0。

• 序列码：如果代码 =0，帮助匹配回送和回送响应的序列码可以为 0。

- 说明： 回送消息中接收到的消息应该在回送响应消息中返回。标识符和序列码由回送发送者使用帮助匹配回送请求的响应。代码 0 可能会从主机或网关接收到。

按照上面介绍的数据包格式，构造相应回送请求和回应应答的 ICMP 报头格式如图 3-10 所示：

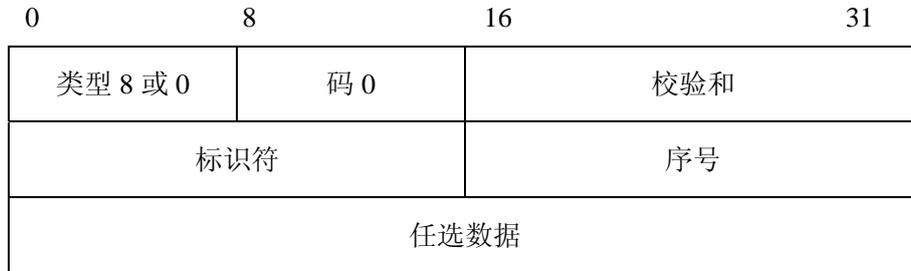


图 3-10 回应请求 (8) 和回应应答 (0)

在 MORCS-1 系统中没有设置过滤 ICMP 包时，操作系统能够自动回复 Echo-request 类型的 ICMP 包，因此在实现时候，只需要编写发送 ICMP 包端的程序即可。采用以 500ms 间隔为单位去采集网络中的 RTT，获取网络延时的序列图 3-11 所示：

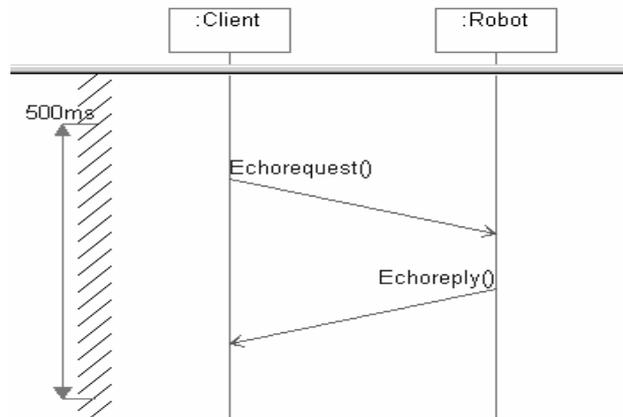


图 3-11 网络延时测量序列图

### 3.4.2 基于 SNMP 的网络性能参数获取

移动机器人系统车载局域网由多台工控机组成，利用集线器连接在一起，三个子系统分别承担着运动控制、路径规划和视觉导航的功能，系统通过 100M 以太网不断交互信息，因而对其网络管理进行管理也是十分重要的。目前网络管理模型主要有两个：ISO/OSI 的公共管理信息模型(CMIP/CMIS)和 TCP/IP 的简单网络管理模型(SNMP)。与其它协议模型相比较，SNMP 以其简单、灵活的特点而

得到广泛应用，且还有以下的优势：

- (1) 基于 TCP/IP 的 SNMP 已作为事实上的协议标准被广泛使用。
- (2) 易于实现。
- (3) 大多数局域网设备如路由器、集线器、PC 计算机都对 SNMP 协议提供了支持。

网络管理有故障管理、计费管理、配置和名称管理、性能管理、安全管理以下五个重要功能，在 MORCS-1 系统中主要是故障管理和性能管理方面的，因此系统设计时主要关注这部分的设计。

SNMP 为应用层协议，是 TCP/IP 协议族的一部分，它通过用户数据报协议 (UDP) 来操作。管理者进程对位于管理站中心的 MIB 的访问进行控制，并提供网络管理员接口。管理者进程通过 SNMP 完成网络管理，SNMP 在 UDP、IP 及有关的特殊网络协议之上实现。

每个代理者必须实现 SNMP、UDP 和 IP，另外，有一个解释 SNMP 的消息和控制代理者 MIB 的代理者进程。

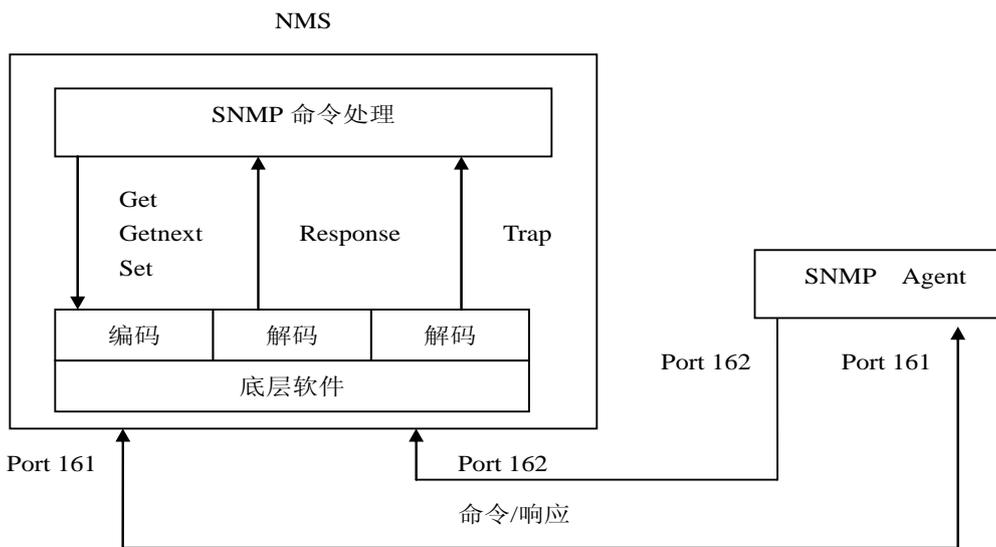


图 3-12 SNMP 的协议环境

图 3-12 描述了 SNMP 的协议环境，关于管理进程和代理进程之间的交互信息，SNMP 定义了 5 种报文<sup>[48]</sup>：

- ① get-request 操作：从代理进程处提取一个或多个参数值。
- ② get-next-request 操作：从代理进程处提取一个或多个参数的下一个参数值
- ③ set-request 操作：设置代理进程的一个或多个参数值。
- ④ get-response 操作：返回的一个或多个参数值。这个操作是由代理进程发

出的，它是前面 3 中操作的响应操作。

⑤ trap 操作：代理进程主动发出的报文，通知管理进程有某些事情发生。

管理进程发出的前面 3 种操作采用 UDP 的 161 端口，代理进程发出的 Trap 操作采用 UDP 的 162 端口，由于收发采用了不同的端口号，所以一个系统可以同时为管理进程和代理进程。

SNMP 管理信息库 MIB 是一个面向对象的数据库，包含着网络管理数据的定义，MIB 由一个个叫 MIB 对象的信息单元组成，MIB 对象记录了管理设备的管理信息，每个对象用来代表一个网络资源。代理通过 MIB 来管理对象的各个方面的参数。MIB 的定义是树形结构。每一分枝被 ISO 指定了唯一整数来标识，分枝的端点即树叶代表设备的某一方面的参数（在 MIB 中称为对象）。管理者通过 SNMP 向代理请求某一对象，对象由从树根开始到该对象为止的一串代表各路经号的整数表示（MIB 中称为 OID）。代表在应答的报文中发送回该对象的值，这样管理者就获知了结点的信息。

MIB 结构采用一种树形结构的数据库，MIB-I 定义了 8 个管理信息类别，MIB 结构如图 3-13 所示。

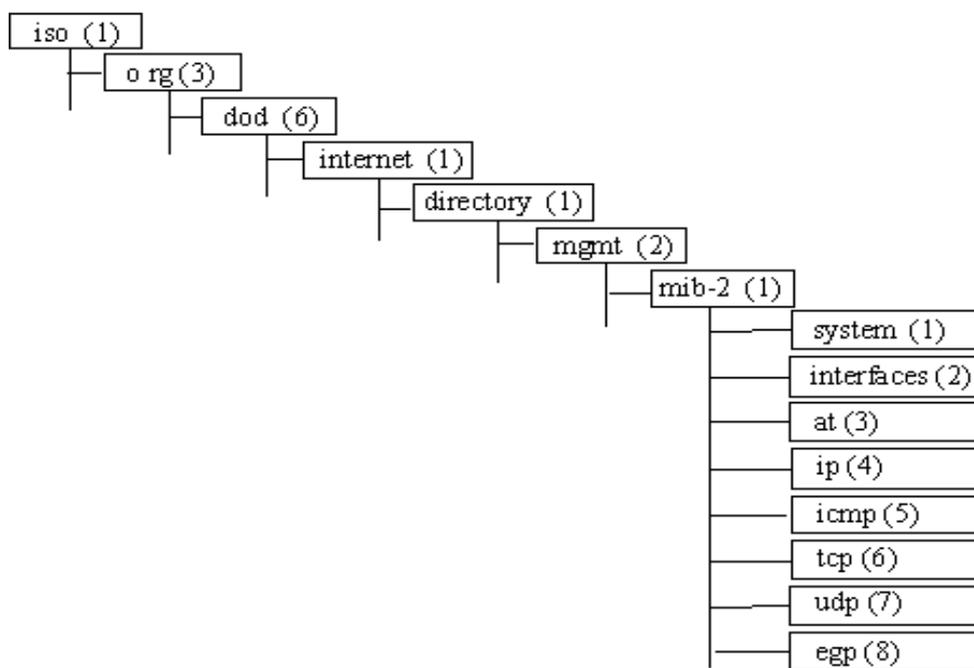


图 3-13 MIB 结构

数据采集是进行管理的基础，采集前要根据管理的需要，确定采集对象，也就是确定使用哪些 MIB 来进行性能管理，下面对图 3-13 中的性能管理对象的作用做一些说明：

① interfaces 组：用于对网络设备的各个网络接口的利用率、错误率和丢

失包率等性能特征进行监测或分析。

② ip 组：对通过 IP 层的网络数据进行统计。

③ icmp 组：对各种 ICMP 消息的接收和发送进行统计。

④ tcp 组：纪录了建立 TCP 连接以及 TCP 传输过程中状态的变化情况，反映了网络的稳定性。

⑤ udp 组：统计 UDP 数据报的输入和输出及发生错误的数量。

性能管理在取得所需要网络设备的 MIB 对象值之后，需要对原始数据作进一步的加工处理，才能得到所需要的信息。下面给出一些本系统用于性能管理的计算公式：

Interface 组：

设备接口接收利用率 =  $\Delta(\text{ifInOctets}) * 8 * 100 / (\text{ifSpeed} * \Delta(\text{seconds}))$

设备接口发送利用率 =  $\Delta(\text{ifOutOctets}) * 8 * 100 / (\text{ifSpeed} * \Delta(\text{seconds}))$

总利用率 = 接收利用率 + 发送利用率

接收丢包速率 =  $\Delta(\text{ifInDiscards}) / \Delta(\text{seconds})$

发送丢包速率 =  $\Delta(\text{ifOutDiscards}) / \Delta(\text{seconds})$

接收错误速率 =  $\Delta(\text{ifInErrors}) / \Delta(\text{seconds})$

发送错误速率 =  $\Delta(\text{ifOutErrors}) / \Delta(\text{seconds})$

Ip 组：

包接收丢失率 =  $\Delta(\text{ipInDiscards}) / \Delta(\text{ipInReceives})$

Ip 输入错误率 =  $(\text{ifInDiscards} + \text{ipInHdrErrors} + \text{ipInAddrErrors}) / \text{ipInReceives}$

ip 输出错误率 =  $(\text{ifOutDiscards} + \text{ipOutHdrErrors} + \text{ipOutAddrErrors}) / \text{ipOutRequests}$

因此必须先对 ifInOctets、ifInDiscards、ipInDiscards 等 MIB 中 OID 数据进行查询。其主要工作是完成 SNMP 的 get 操作，由于 MIB 中的变量存在着简单变量和表变量两种不同的格式，应该采用不同的访问方式，简单变量具有 INTEGER、OCTER STRING、GAUGE 等类型，简单变量的访问流程如图 3-14 所示。

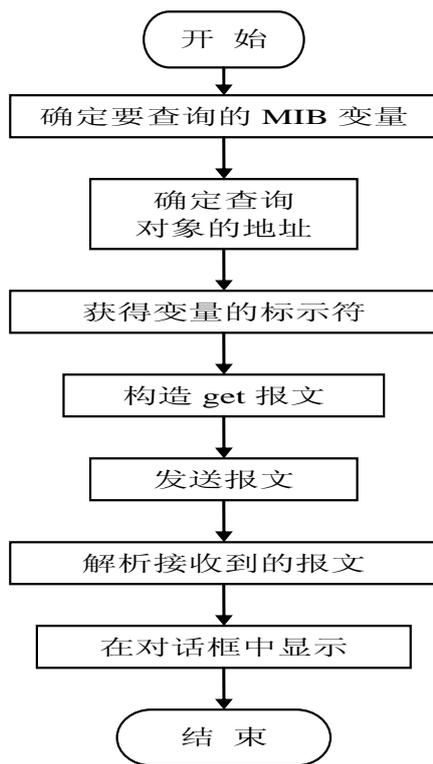


图 3-14 简单变量的访问

在提取表数据时，由于表项的数量和 Oid 都不确定，所以不能通过某个特定的 Oid 直接获得取值使用 GetNext 命令来实现表的遍历，这里主要讨论该算法的原理和如何基于 Snmp++下的设计。

MIB 表是通过行和列来描述的<sup>[49]</sup>，其中列表头是各个表项的原始 Oid，而行表头则是 index。这样以来一个 Oid 和一个 index 就唯一地确定了表中的一项。比如在接口表中，ifDescr (Oid 为 1.3.6.1.2.1.2.2.1.2) 为一列，而具体对于某一个接口则为一行。这样，某一具体表项的 Oid 就表示为：列 Oid+index 的形式，表 3-6 描述了一张表的格式。

表 3-6 表格变量

	ifIndex	ifDescr	ifType	ifMtu	Ifspeed
	1.3.6.1.2.1.2.2 .1.1	1.3.6.1.2.1.2.2 .1.2	1.3.6.1.2.1.2.2. 1.3	1.3.6.1.2.1.2.2. 1.4	1.3.6.1.2.1.2.2 .1.5
Index0	xxx	xxx	xxx	xxx	xxx
Index1	xxx	xxx	xxx	xxx	xxx
Index2	xxx	xxx	xxx	xxx	xxx

按照协议描述，最基本的方法是通过 index 来获取某一表项。但事实上，index

本身也是一个表项，再加之有些表需要多个 index，并且各种 index 的数据类型不同，比如要手工处理 ip 地址类型的 index 就比较困难，所以这种方法具有很难的操作性。因此，在实际编程时，可以采取一些比较技巧化的方法。

从 Snmp 中对于 GetNext 命令的描述可知，如果 GetNext 的参数为一个表中某一列的表头 Oid，比如前面的 ifDescr (1.3.6.1.2.1.2.2.1.2)，则得到的值为该列第一行元素值，并可得到该值的 Oid。再对取得的 Oid 使用 GetNext 就可获得该列第二行的值。如此下去，如果到了该列的最后一行，那么用 GetNext 将得到下一列的第一行。如果到了该表的最后一个元素，那么用 GetNext 将得到按 MIB 树所得的下一个元素值。显然，在越界的情况下，其 Oid 的前部分已不同于本列表头的 Oid，所以，可以通过得到的 Oid 值来判断是否越界，图 3-15 列出了复杂变量访问的流程图，具体代码见附录一。

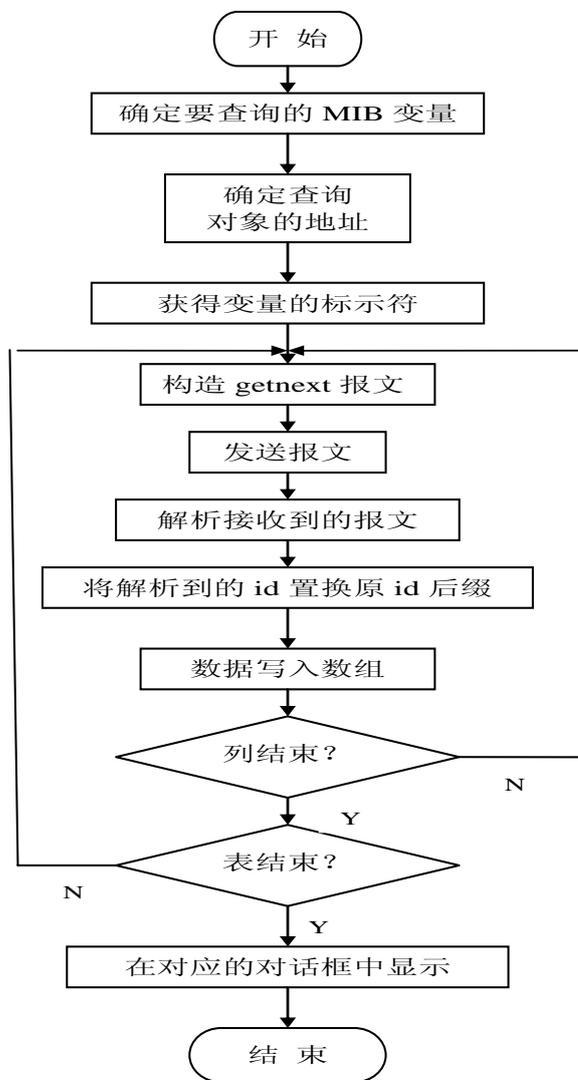


图 3-15 表变量的访问

在 MORCS-1 系统中，运行的是 Windows 系统，SNMP Agent 只需通过系统配置就能完成，因此，系统设计的主要任务是在管理端的设计，当确定了需要访问的对象标识符，根据标识符所属不同的类型，可以分别采用基于简单变量的方式或表格的方式获取需要的网络状态参数，就能实现对网络性能进行分析。

图 3-16 为对实验室与移动机器人 MORCS-1 车载网络之间状态进行监测的结果，左图为基于 SNMP 的网络流量监测界面，右图为基于 ICMP 的网络延时测量界面。他们能对网络状态有效的监测，使系统更加安全有效的运行。

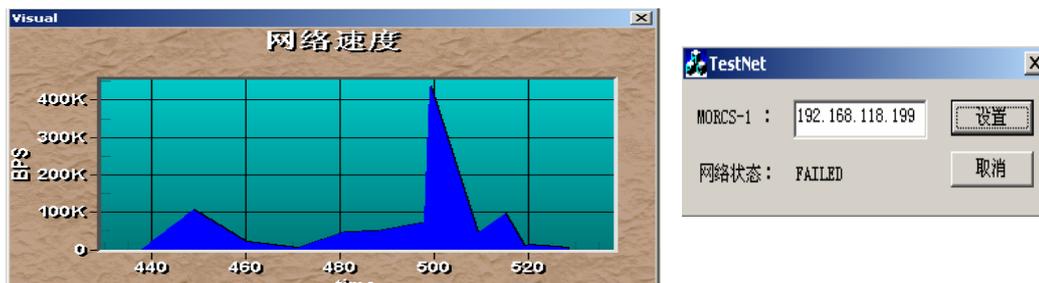


图 3-16 网络状态监测

### 3.5 小结

在本章中，首先分析了系统设计的目标，然后针对移动机器人 MORCS-1 上的传感器，设计了基于多线程方式的移动机器人传感器状态参数获取子系统，根据任务重要程度，设定不同的优先级，使系统能实时的获取移动机器人车载传感器的状态参数；接着对机器人车载计算机的远程监视系统进行了设计，分析了其引起的网络流量的变化，此系统能在远程端维护车载工控机系统，给野外实验中维护车载工控机系统提供了方便；最后基于 ICMP 和 SNMP，设计了 MORCS-1 与远程控制端的网络性能参数监测系统，用于监测网络的延迟、流量等参数。通过这三个子系统的设计，确保了对移动机器人的硬件、软件和通讯系统的全面监测。

## 第四章 移动机器人故障诊断方法研究

移动机器人是一个强非线性参数不确定系统,且运行在未知的、复杂的环境下,一旦发生事故就可能造成巨大的损失。因而对其故障诊断方法的研究就极为重要。本章首先基于小波诊断算法对传感器故障进行了仿真分析,利用移动机器人编码器计算所得角速率与光纤陀螺所测角速率的冗余关系,设计了针对 MORCS-1 机器人光纤陀螺突发性故障的诊断算法。然后对于在未知环境中运行的移动机器人,有可能出现的车轮打滑、受阻等情况,提出了一个基于规则的车轮的运动状况诊断方法,在 MORCS-1 上验证了其有效性。

### 4.1 小波分析的基本理论

小波变换的概念是由法国从事石油信号处理的工程师 J.Morlet 在 1974 年首先提出的,1986 年著名数学家 Y.Meyer 构造出一个真正的小波基,小波分析才开始蓬勃发展起来,其中比利时女数学家 I.Daubechies 撰写的《小波十讲》对小波的普及起了重要的推动作用。它与 Fourier 变换、窗口 Fourier 变换(Gabor 变换)相比,是一个时间和频率的局域变换,因而能有效的从信号中提取信息,通过伸缩和平移等运算功能对函数或信号进行多尺度细化分析,解决了 Fourier 变换不能解决的许多困难问题。

小波(wavelet),即在时(空间)域延续度很小的“波”。如果函数  $\psi(t)$  是平方可积函数,即  $\psi(t) \in L^2(R)$ , 并且其傅立叶变换  $\Psi(\omega)$  满足

$$\int_R \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \quad (4-1)$$

称  $\psi(t)$  为一基本小波或小波基函数。而通常所讲的小波则是由小波基函数经过伸缩和平移而得到的函数族  $\psi_{a,\tau}(t)$ :

$$\psi_{a,\tau}(t) = a^{-\frac{1}{2}} \psi\left(\frac{t-\tau}{a}\right) \quad a > 0, \tau \in R \quad (4-2)$$

其中,  $a$  是尺度(伸缩)因子,  $\tau$  是平移因子。

由定义可知,小波基函数是一类特殊的函数:

- ①通常它们在时(空间)域内是紧支集或近似紧支集的,并且在频域内也具有良好的局部性,可以作为“带通滤波器”或“窗口”使用;
- ②它们具有正负交替波动性,有  $\Psi(0) = 0$ ;
- ③它们经过伸缩和平移变化得到的函数族也同样具有时(空间)域、频域局部

性和正负交替波动性，并且尺度因子  $a$  越小，时(空间)域窗口越小，而对应频域窗口的中心频率和窗口宽度越大。

对于  $L^2(R)$  中的函数  $f(t)$ ，其小波变换可定义为

$$WT_f(a, \tau) = \langle f(t), \psi_{a, \tau}(t) \rangle = \frac{1}{\sqrt{a}} \int_R f(t) \bar{\psi}\left(\frac{t-\tau}{a}\right) dt \quad (4-3)$$

相应的小波逆变换为

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \frac{da}{a^2} \int_{-\infty}^\infty WT_f(a, \tau) \psi_{a, \tau}(t) d\tau \quad (4-4)$$

$$\text{其中, } C_\psi = \int_R \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty。$$

由公式(4-3)可知，小波变换实际上是信号  $f(t)$  与小波函数  $\psi_{a, \tau}(t)$  的内积，即信号  $f(t)$  在  $\psi_{a, \tau}(t)$  上的展开(投影)结果。那么，可以通过小波变换提取信号  $f(t)$  在特定尺度  $a$  下、特定位置  $\tau$  处的信号特征。由于尺度因子  $a$  在一定程度上决定了小波函数  $\psi_{a, \tau}(t)$  的频率特性，可以通过确定尺度因子  $a$  来提取不同频率的信号特征，从这个意义上讲，小波变换具有一定的频率自适应性。

由公式(4-4)可知，信号  $f(t)$  可由小波族  $\psi_{a, \tau}(t)$ ， $a > 0, \tau \in R$  线性拟合而成，而各小波前的系数由相应的小波变换确定，此即为信号小波重建的依据。

若处理离散化的信号，就对  $a$ 、 $\tau$  和  $t$  进行离散化处理，通常，取  $a = 2^m, \tau = 2^m \cdot n$ ，其中， $m, n \in Z$ ，并且对  $t$  进行与  $\tau$  相同的归一化处理，取  $dt = \Delta t = 1$ 。那么，离散化的小波函数可写为

$$\psi_{m, n}(t) = 2^{-\frac{m}{2}} \psi(2^{-m} t - n) \quad (4-5)$$

任意函数  $f(t)$  的离散小波变换为

$$WT_f(m, n) = \sum_{t=-\infty}^{\infty} f(t) \cdot \bar{\psi}_{m, n}(t) \quad (4-6)$$

相应的离散小波逆变换为

$$f(t) = \frac{1}{C_\psi} \sum_m \sum_n WT_f(m, n) \cdot \psi_{m, n}(t) \quad (4-7)$$

## 4.2 连续小波故障诊断的原理

移动机器人对周围环境的感知是通过各种传感器来实现的，因此，对于传感器的故障诊断技术应引起重视，小波分析由于具有时域和变频域特性而倍受关

注。利用小波变换进行传感器故障检测的基本思路是，首先对传感器系统的输入输出信号进行连续小波变换，利用该变换求出输入输出信号的奇异值，然后去除由于输入突变所引起的极值点，则其余的极值点对应于传感器的故障<sup>[25,50]</sup>。

假定所诊断的传感器为一单输入单输出的线性系统，传感器故障表现为系统参数的变化。即：

$$Y(S)=G(S)U(S)+\Delta G(S)U(S)+E(S) \quad (4-8)$$

其中： $*(S)$ 表示相应信号的拉氏变换： $Y(S)$ 、 $U(S)$ 、 $E(S)$ 分别表示输出、输入和噪声信号。假定， $E(S)$ 是零均值平稳随机噪声； $G(S)$ 是系统的传递函数； $\Delta G(S)$ 反映由故障造成的系统参数变化，方便起见，假定：

$u(t)$  是已知的分段平稳随机信号，而且其突变与故障时在不同时间发生的； $G(S)$ 不存在位于原点的极点和零点，而且故障发生时， $\Delta G(S) \neq 0$ 。

对于平稳随机信号  $x(t)$ ，其小波变换  $W_x(s,t)$  的均值为 0，方差随着尺度  $s$  的增大而趋于零。

对于前面描述的传感器系统，当没发生故障时， $W_y(s,t) \approx K * W_u(s,t)$  在较大的尺度  $s$  下成立。其中， $K = G(\omega_{s_0})$ ， $\omega_{s_0}$  是  $\hat{\psi}_s(\omega)$  的中心频率。在实际应用中，可以用  $J = \min_K \sum_i [W_y(s,t) - K W_u(s,t)]^2$  的最小二乘解

$$\hat{k} = \frac{\sum W_y(s,t) W_u(s,t)}{\sum [W_u(s,t)]^2} \quad (4-9)$$

来代替  $K$ 。

下面分别两种不同的情况进行讨论：

(1) 当  $u(t)$  均值不恒为零时

当  $\Delta G(S) \neq 0$ ，系统参数的变化将导致  $y(t)$  均值得变化。当然， $u(t)$  均值的不稳定也可能导致  $y(t)$  均值得突变。故障检测的关键是如何检测出  $y(t)$  的突变，并区分出导致  $y(t)$  突变的两种原因。为此，可以首先利用  $W_u(s,t)$  和  $W_y(s,t)$  分别检测出  $u(t)$  和  $y(t)$  的突变边沿（均值突变），然后比较  $W_u(s,t)$  和  $W_y(s,t)$ ，去除  $W_y(s,t)$  中由于  $u(t)$  突变而造成的极值点。则  $W_y(s,t)$  中剩下的极值点就对应着传感器系统参数发生变化，利用公式

$$r(t) = W_y(s,t) - \hat{k} W_u(s,t) \quad (4-10)$$

其中  $\hat{k}$  由上面的公式给出。它是由系统正常运行时的测量数据计算出来的。

当系统没有发生故障时，由近似关系  $W_y(s,t) \approx \hat{k} * W_u(s,t)$  得知，即使  $u(t)$  发生了突变，残差  $r(t)$  也会随着尺度  $s$  的增大而趋于零。当系统发生故障后， $y(t)$  的均值会发生突变，而  $u(t)$  在一小段时间内使平稳的。因此，随着尺度  $s$  的增大，

$W_y(s,t)$  会出现明显的, 随着尺度增大而增大或缓慢衰减的极值点, 而  $W_u(s,t)$  趋于 0。因此残差  $r(t)$  也会在各尺度上出现明显的均值点, 而且他们不会随着尺度的增加而迅速衰减。

(2)  $u(t)$  均值恒定为零

此时,  $y(t)$  的均值恒为零, 系统参数的变化将反映在  $y(t)$  方差的变化上。当然,  $u(t)$  的方差的突变也会引起  $y(t)$  方差的变化。由于小波变换不能直接检测信号的突变点, 为此, 令:

$$r(t) = |W_y(s,t)| - |\hat{k} W_u(s,t)| \quad (4-11)$$

由近似关系  $W_y(s,t) \approx \hat{k} * W_u(s,t)$  未知, 未发生故障时, 在较大尺度下, 即使  $u(t)$  的方差发生了突变, 也会有:  $r(t) \approx 0$ 。当系统发生了故障后, 由于  $y(t)$  方差的变化,  $W_y(s,t)$  得方差以及  $|W_y(s,t)|$  的均值也会发生变化, 而  $W_u(s,t)$  在一小段时间内是平稳的, 则  $|W_u(s,t)|$  的均值也将保持不变。因此,  $r(t)$  的均值将发生变化。在此基础上, 可以进一步利用  $r(t)$  的小波变换  $W_r(s,t)$  检测其突变点。

### 4.3 传感器故障诊断

#### 4.3.1 传感器故障诊断仿真分析

针对传感器输入信号  $u(t)$  的不同, 下面利用 Matlab 中的 Simulink 仿真工具分别分析不同输入情况下故障诊断情况<sup>[51]</sup>。

(1) 输入信号  $u(t)$  均值不恒为零时

$$\text{假定传感器为单输入单输出系统: } G(s) = \frac{cs + d}{s^2 + as + b}$$

当传感器正常时,  $a=8$ ,  $b=20$ ,  $c=5$ ,  $d=10$ , 以系统参数的突变来模拟传感器的突变故障, 设  $t=100s$  时传感器发生故障,  $c$  突变为 10,  $d$  突变为 25, 其他系数不变, 输入信号  $u(t)$  采用分段平稳的均匀分布随机信号, 其均值在  $t=60s$  时由 0.5 突变为 2, 干扰信号  $e(t)$  采用方差为 0.1 的高斯白噪声。

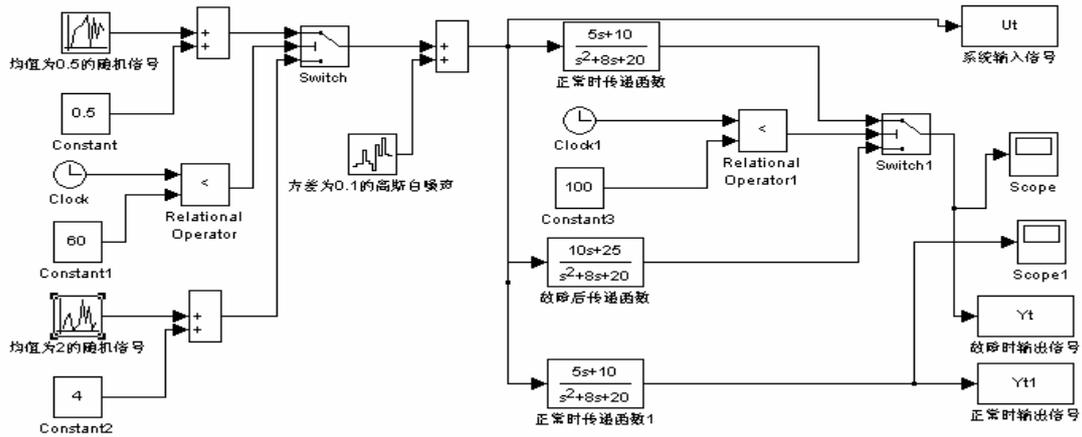


图 4-1 传感器模型

采用上面介绍的连续小波诊断方法进行故障诊断，在 Simulink 仿真环境中建立故障检测系统模型如图 4-1 所示，采样间隔为 0.01s，模拟时间为 200s，运行仿真程序，得到传感器系统的输入输出信号如图 4-2 所示，可以看出在时间 60s 时由于输入变化而引起输出的变化和在由于故障发生时传感器输出的突变。

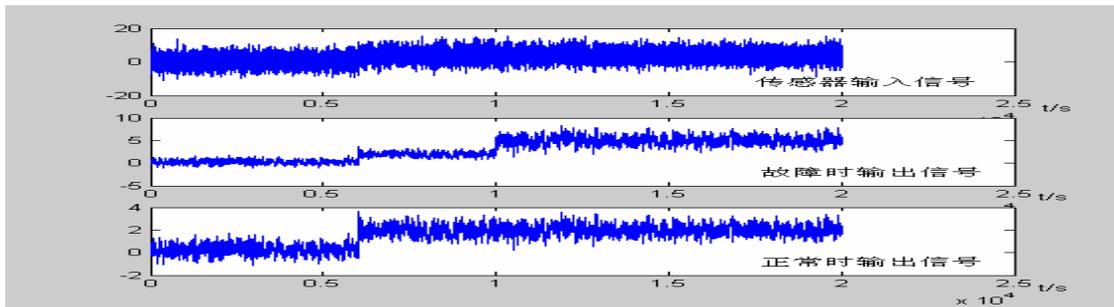


图 4-2 输入输出信号

根据所得输入输出信号，利用上面介绍的连续小波变换方法对传感器进行故障诊断，仿真中采用 db3 小波进行诊断分析，结果如图 4-3 所示。

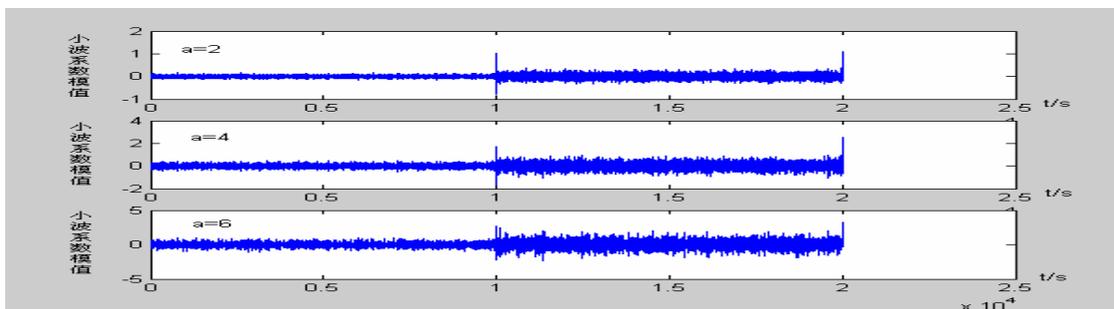


图 4-3 各尺度下的诊断结果

可以看出在  $t=100s$  时，各尺度下的极值图都出现了明显的峰值。由此证明仿真采用的理论是可行的，可以在输入信号发生突变的同时很好的分辨出传感器发生的突变故障。

(2) 当传感器输入信号  $u(t)$  均值为零

取与(1)中相同的系统模型、故障类型、噪声信号。传感器输入  $u(t)$  采用方差为 1 零均值的随机信号，建立故障诊断系统模型，如图 4-4。

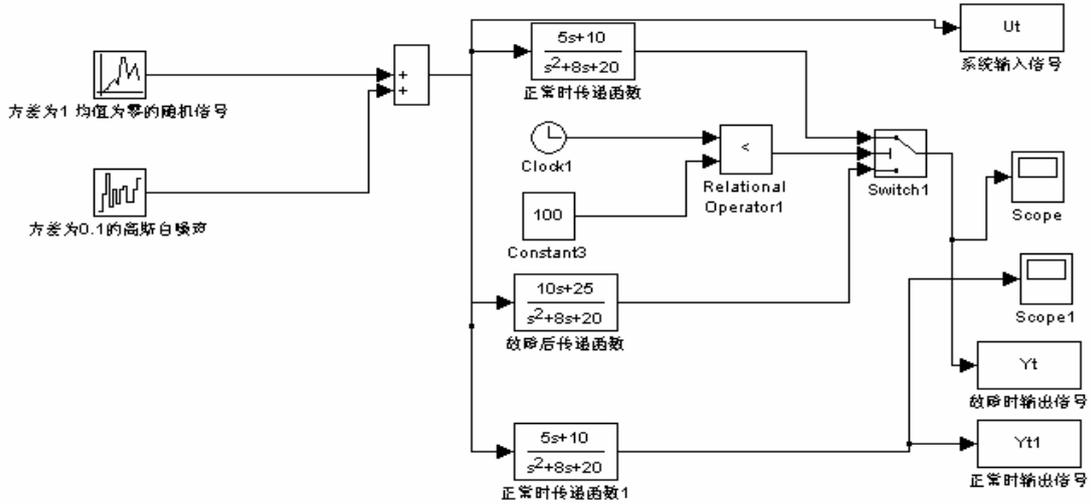


图 4-4 传感器模型

当采样时间为 0.01s，模拟时间为 200s 时候的系统输入输出信号见图 4-5 所示。

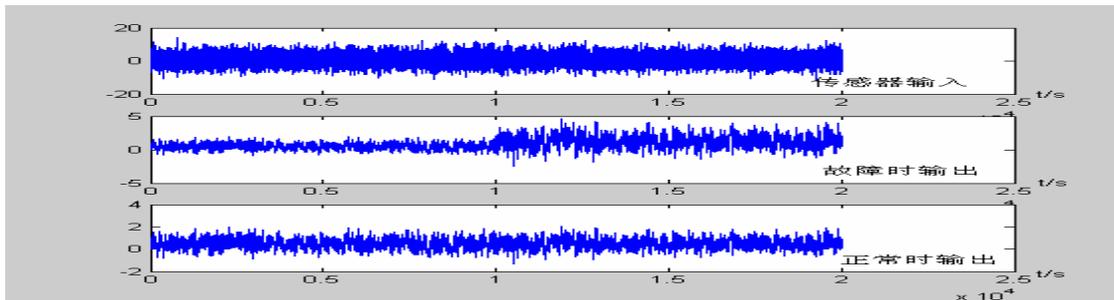


图 4-5 系统输入输出信号

利用所得信号与上节介绍的故障诊断方法，通过 db3 小波得到多尺度下的极值图如图 4-6 所示。

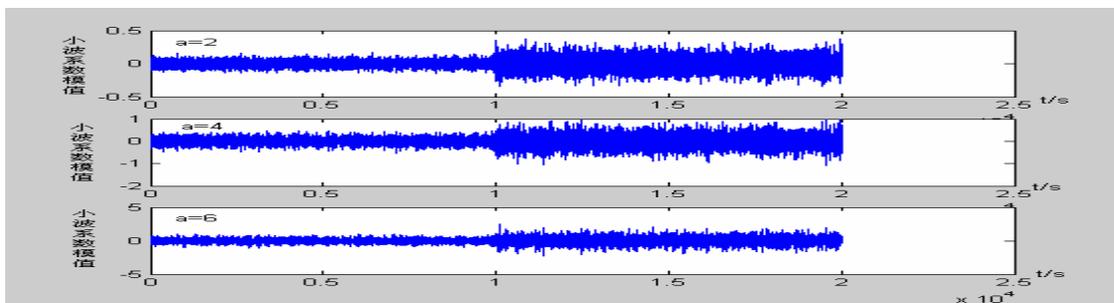


图 4-6 多尺度下的极值图

从图中可以看出在  $t=100s$  时， $r(t)$  的连续小波变换发生了明显的突变，较好的说明了该时刻传感器有故障发生，这说明了利用小波的方法进行故障检测方法

的可行性,可以在输入信号方差突变和传感器突变同时存在的情况下,完成传感器故障检测任务,同时也表明了连续小波变化极值的传感器故障检测方法具有灵敏度高,对输入信号要求低,不需要传感器数学模型的特点。

### 4.3.2 机器人光纤陀螺故障诊断算法设计

对于 MORCS-I 移动机器人系统,精确地进行自定位是一个基本的要求。自定位就是获得移动机器人自身相对于一个固定坐标系的位置和方向角(统称位姿)。因此,移动机器人就必须要有了一定的传感器来获得这些位姿信息(如移动机器人的位置、航向角、线速度、角速度等)。当机器人在水平面内运动,且车轮与地面之间无滑动的条件下,利用编码器就可以给机器人定位,然而当机器人并非在上述条件下运行时,用此方法计算出的航向角误差很大,只能作为参考值。因此,利用光纤陀螺来测量机器人在运行过程中的航向偏角,而编码器主要用来测量机器人的运行距离,其车体的俯视图如图 4-7 所示<sup>[34]</sup>。

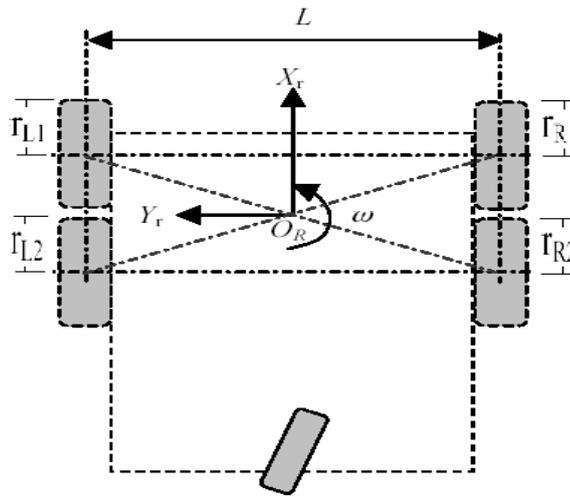


图 4-7 IMORCS-I 的俯视图

相应的运动学模型如图 4-8 及下列方程表示。

$$\begin{cases} \omega = (V_R - V_L) / L \\ V = (V_R + V_L) / 2 \end{cases} \quad (4-12)$$

其中:  $V_L = (V_{L1} + V_{L2}) / 2$ ,  $V_{L1} = r_{L1} \cdot \omega_{L1}$ ,  $V_{L2} = r_{L2} \cdot \omega_{L2}$ ;  $V_R = (V_{R1} + V_{R2}) / 2$ ,  $V_{R1} = r_{R1} \cdot \omega_{R1}$ ,  $V_{R2} = r_{R2} \cdot \omega_{R2}$ ,  $r_{L1}$ ,  $r_{L2}$ ,  $r_{R1}$ ,  $r_{R2}$  和  $L$  分别为左前轮半径、左后轮半径、右前轮半径、右后轮半径和轴长;  $\omega_{L1}$ ,  $\omega_{L2}$ ,  $\omega_{R1}$ ,  $\omega_{R2}$  为编码器输出,分别表示左前轮角速度、左后轮角

速度、右前轮角速度、右后轮角速度； $V_{L1}$ ， $V_{L2}$ ， $V_{R1}$ ， $V_{R2}$ 为线速度，分别表示左前轮线速度、左后轮线速度、右前轮线速度、右后轮线速度； $V_L$ ， $V_R$ 分别为机器人左侧前后轮的平均速度、右侧前后轮的平均速度； $\omega$ 为根据编码器输出和轴长计算出的车体转动角速度， $V$ 为机器人的车体运动线速度。在本系统中，左边两个轮用相同的设定速度，同时，右边两轮也是设定了相同的速度，理论上在水平面运动时，左边两轮的 $V_{L1}$ ， $V_{L2}$ 应该是相同的速度， $V_{R1}$ ， $V_{R2}$ 亦应该相同。由于轮子半径的不同、滑动、噪声干扰等因素，利用其计算出的角速率会有一些的偏差，但其也能体现出车体转动角速率的变化，因此首先采用左右轮的设定速度和编码器的输出进行的诊断分析，在确保编码器输出正常的前提下，利用编码器输出、车轮半径和轴长计算出车体的角速率作为光纤陀螺诊断系统的参考输入信号，然后结合光纤陀螺的实际输出，利用前面介绍的对传感器系统的输入输出信号进行连续小波变换，利用该换求出输入输出信号的奇异值，然后去除由于输入突变所引起的极值点，则其剩余的极值点对应于传感器的故障的方法来对光纤陀螺进行诊断分析。

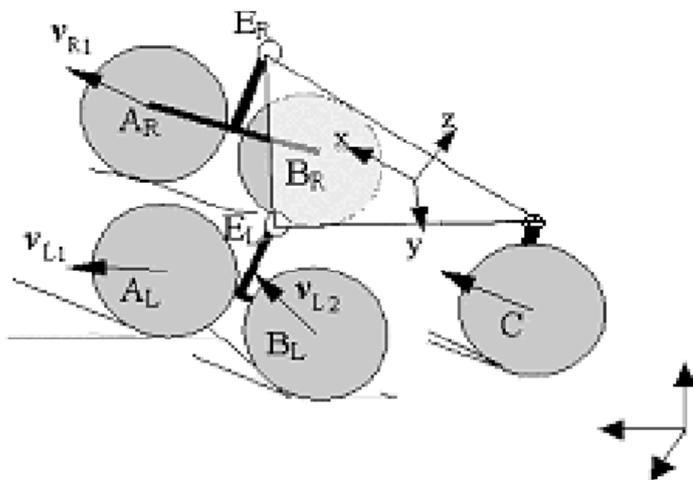


图 4-8 IMORCS-I 机构示意图<sup>[34]</sup>

小波变换中需要选择小波基，小波基是不规则的，不同小波基波形形状差别很大，支撑范围和规则性都有很大的差别。因而，对同一信号选用不同的小波基进行信号处理，往往得到的结果差别较大，这必然影响最终的处理结果。

对于信号的奇异性检测问题，当信号产生奇异点时，在突变点处含有高频成分，并且信号的形状还很不规则，如用 Daubechies 小波族的小波 db2~db10 进行变换可以发现：db2、db3 和 db4 的检测结果要比 db5、db6、db7、db8 和 db10 好。这是因为前者的形状规则性要比后者差，规则性系数越大，规则性越好。并且前面 3 个小波基的支撑要比后面 6 个小。

在这里采用 db3 小波，这种小波的正则性不是很好，但选择正则性比较好的，

即规则系数大的，检测突变点的效果不太理想，在应用中只要知道正交滤波器系数  $h(k)$ ，就可以根据小波分解和重构进行计算。

在实验中，根据系统的运动学模型，假设执行电机无故障，以光纤陀螺为例进行诊断分析，采用第三章设计的系统采集编码器和光纤陀螺的输出数据，间隔为 0.5s。获得的编码器计算输出值和光纤陀螺直接测量所获得的角速率，其如图 4-9 所示。

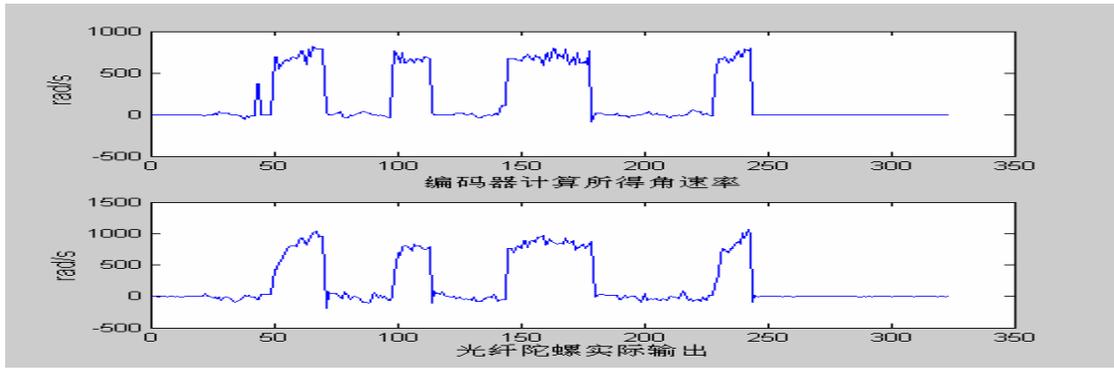


图 4-9 角速率

故障通过对实际测量光纤陀螺所得的数据在 1 分 40 秒时加一个幅度为 1000 的常量来模拟传感器的偏置型故障，如图 4-10 所示。

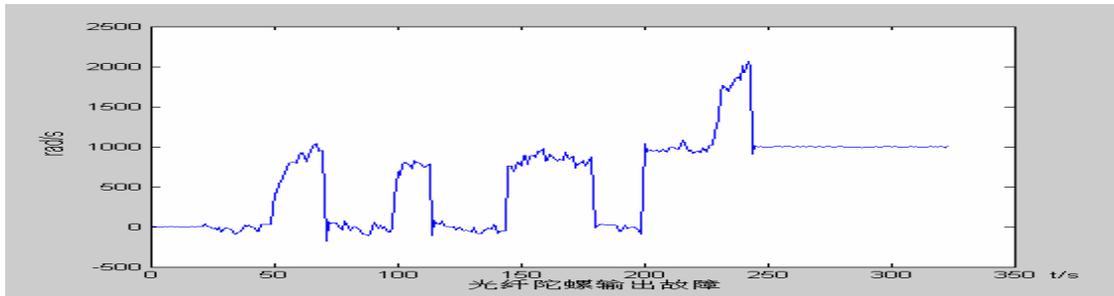


图 4-10 光纤陀螺出现偏置故障

利用偏置故障信号和上节介绍的故障诊断方法，通过 db3 小波得到多尺度下的极值图如图 4-11 所示，从图中可以看出在所有尺度下，在  $WT_a r(t)$  在 1 分 40 秒附近都出现了较大的极值，且随尺度的增大极大值点越来越明显。

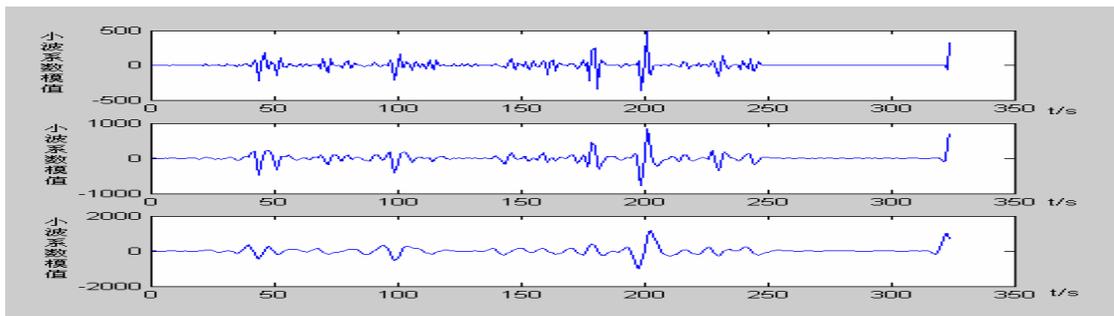


图 4-11 各尺度下的诊断结果

结果表明,该方法灵敏度较高,克服噪声能力强,对输入信号要求低,不需要对象的数学模型,其缺点是在大尺度下,由于滤波器 $\Psi_a(t)$ 的时域宽度较大,检测时会导致一定的延迟。

在 MORCS-1 系统实现对光纤陀螺的诊断中,考虑到计算速度,采用了每 64 个采样数据计算一次,分解为 3 层的方式,在每层中寻找极大值点,同时记录极大值点的位置,并判断极大值是否随着分解层次的加大变大,如果满足则确定为故障点,发出报警信息,算法实现的流程图如 4-12 所示。

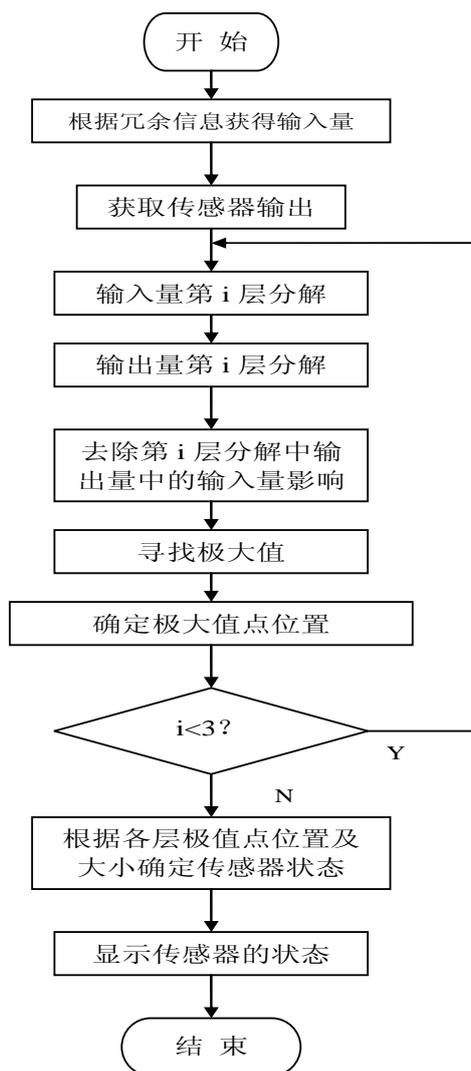


图 4-12 光纤陀螺小波诊断流程图

#### 4.4 车轮故障诊断算法设计

对于在未知环境中运行的移动机器人,有可能出现的车轮打滑、受阻等情况,有必要建立一针对车轮的运动状况诊断方法,为移动机器人摆脱当前状态提供一

种策略。在移动机器人 MORCS-1 中，左边两个轮用相同的设定速度，同时，右边两轮也是设定了相同的速度，根据上一节运动学模型分析，理论上在水平面运动时左侧前后两轮的速度因该相同，右侧两轮的速度亦应相同，当系统的左前、左后、右前、右后中的一个轮子遇到受阻或打滑时候，其编码器的输出就会与其同侧的另一轮编码器输出存在差异，因而，在假设执行电机无故障的前提下，结合系统设定的速度与编码器的输出，根据系统的运动学模型，可以利用轮子的速度差等参数来确定轮子的状态，图 4-13 为对移动机器人 MORCS-1 左侧车轮诊断的流程图。

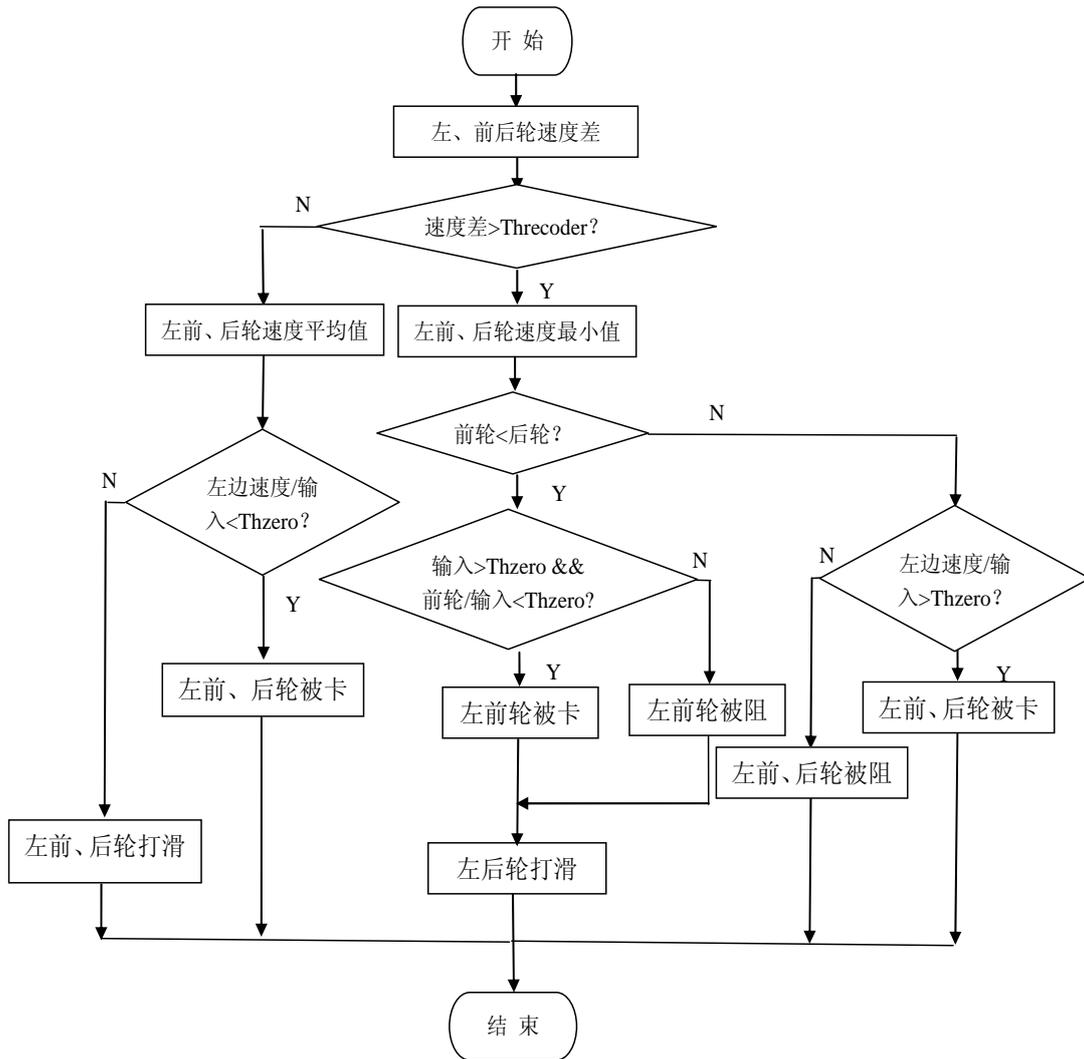


图 4-13 MORCS-1 左侧车轮异常诊断流程图

在移动机器人 MORCS-1 上反复实验的基础上，根据所需要的灵敏性要求，设计时候取  $Threocoder=4, Thzero=0.1$ 。系统实现的具体代码见附录二所示。图 4-14 为对车轮进行诊断的结果，分别为前进、后退、车轮受阻时 MORCS-1 左右侧轮子运动速度及各个轮子的状态。

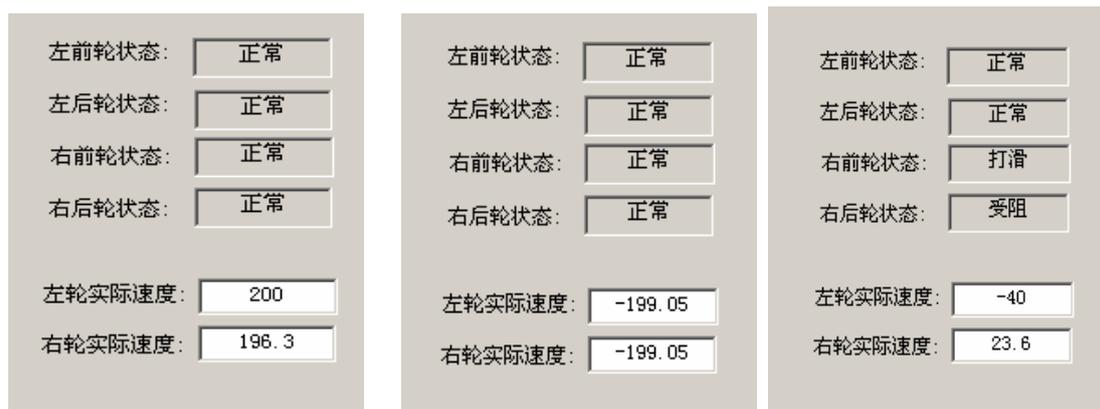


图 4-14 车轮状态

## 4.5 小结

移动机器人的传感器承担着环境感知的任务,对传感器各类故障及时有效的检测与隔离是移动机器人故障诊断的重要组成部分。因此在本章中首先对连续小波变换在传感器故障诊断领域中的应用从原理上做了分析,利用移动机器人编码器计算所得角速率与光纤陀螺所测角速率的冗余关系,设计了针对 MORCS-1 机器人光纤陀螺突发性故障的诊断算法,此方法具有不需要传感器精确数学模型,且对输入信号要求低,克服噪声能力强的优点。对于在未知环境中运行的移动机器人,有可能出现的车轮打滑、受阻等情况,提出了一个基于规则的车轮的运动状况诊断方法,算法能灵敏的检测出车轮状态,并在 MORCS-1 上验证了其有效性。

## 第五章 监测与诊断系统实现

在前面几章对机器人状态监测系统设计和诊断方法研究的基础上,在本章中将研究系统的具体实现,即利用面向对象的编程语言工具,完成代码的编写,介绍了实现中的关键技术、系统模块图、类清单等。

### 5.1 开发环境

C/C++语言兼有高级和低级语言的双重性,功能强大,灵活,执行效率高,Visual C++是由C/C++语言演化而来的可视化编程工具,具有C/C++语言支持面向对象程序设计的所有特性,为软件开发人员提供了完整的编辑、编译和调试工具以及建立于Win32 API基础上的MFC类库,有效的缩短了开发周期。由于Windows操作系统本身大部分都是用C/C++语言编写,而VC++正是基于C/C++语言的Win32应用程序集成开发环境。因此,使用VC++进行Windows应用程序的开发非常具有优势。

状态监测和诊断属实时控制程序,要求实时性强,稳定性好。基于该特点,选用Visual C++ 6.0作为系统开发工具,笔者在编制软件的时候充分利用MFC提供的微软基础类,以提高开发效率。该系统在主流操作系统Windows 2000/XP下进行设计开发。

### 5.2 系统所采用的关键技术

#### 5.2.1 HP SNMP++

SNMP++是HP公司提供的开发基于SNMP管理应用程序的应用程序编程接口,是为网络管理应用开发提供SNMP服务的一系列C++类。SNMP不是现有的SNMP引擎上的附加层或者封装,也不是代替其他现存的SNMP API,如WinSNMP,而是最小限度的使用现存的SNMP库并使开发工作更加的有效。

SNMP++提供成员函数get()、getnext()、getbulk()、set()、trap()和SNMP++类库来完成网络数据的采集。SNMP++的主要目标是提供跨越各种操作系统、网络操作系统和网络管理平台的可移植的应用程序编程接口。通过隐藏SNMP++的内部工作机制,SNMP++在不同的平台上提供统一的API接口。任何使用SNMP++编写的网络管理应用程序,不需要做任何改变就可以移植到其他的平

台。另外一点就是 SNMP++ 可以运行于各种不同的协议上面，当前 SNMP++ 可运行于 IP 协议或者 IPX 协议之上。

SNMP++ 把 WinSNMP 的 API 函数封装成一个个相关的 C++ 类从而大大地简化了 SNMP 网络管理软件开发的复杂性和开发难度，缩短了开发时间。它把 SNMP 编程部分都面向对象化，因此开发人员不必十分了解 SNMP 底层的机制，只需了解 C++ 知识和 SNMP 编程的流程即可。

SNMP++ 封装了如下几个类：SNMP 类、对象标识符 (Oid) 类、字节串 (Octet string) 类、时钟 (Time Ticks) 类、计数器 (Counter32、Counter64) 类、地址 (Address) 类、变量绑定 Vb (Variable Binding) 类等。

在 SNMP++ 的类库中，像 Oid 类、Pdu 类、Snmpp 类、CTarget 类等是对 SNMP 协议中几个重要的概念的封装，对网络管理应用程序的开发都是必须的，因此必须对其有一个很好的理解，才能开发出满意的程序。

### 5.2.2 多线程及同步技术

众所周知，许多传统的应用程序都是单线程的，即在程序运行期间，由单个线程独占 CPU 的控制权，负责执行所有任务。在这种情况下，程序在执行一些比较费时的任务时，就无法及时响应用户的操作，影响了应用程序的实时性能。

然而，在监测系统，特别是在远程监测与诊断系统中，应用程序往往不但要及时把监控对象的最新信息反馈给监视客户（通过图形显示），还要处理本地机与远程机之间的通信以及对机器人系统诊断等任务，这时，仅仅由单个线程来完成所有任务，显然无法满足监控系统的实时性要求。在 Windows 系统下，中断机制对用户是不透明的。为此，可引进多线程机制，主线程专门负责消息的响应，使程序能够响应命令和其他事件。而辅助线程可以用于完成其他比较费时的任务，如通信、图形显示、故障诊断等，这样就不至于影响主线程的运行。

值得注意的是，在系统中使用多线程时，则不可避免会出现多个线程竞争访问数据资源的现象，如果不采取适当的措施，其他线程往往会在线程处理任务结束前就去访问处理结果，这就很有可能得到有关处理结果的错误了解。例如，多个线程同时访问同一个全局变量，如果都是读取操作，则不会出现问题。如果一个线程负责改变此变量的值，而其他线程负责同时读取变量内容，则不能保证读取到的数据是经过写线程修改后的，为了确保读线程读取到的是经过修改的变量，就必须在向变量写入数据时禁止其他线程对其的任何访问，直至赋值过程结束后再解除对其他线程的访问限制。像这种保证线程能了解其他线程任务处理结束后的处理结果而采取的保护措施即为线程同步。

线程同步是一个非常大的话题。从大的方面讲，线程的同步可分用户模式的

线程同步和内核对象的线程同步两大类。用户模式中线程的同步方法主要有原子访问和临界区等方法。其特点是同步速度特别快,适合于对线程运行速度有严格要求的场合。内核对象的线程同步则主要由事件、等待定时器、信号量以及信号灯等内核对象构成。由于这种同步机制使用了内核对象,使用时必须将线程从用户模式切换到内核模式,而这种转换一般要耗费近千个 CPU 周期,因此同步速度较慢,但在适用性上却要远优于用户模式的线程同步方式,系统中提供了临界区、管理事件内核对象、信号量内核对象等方式。

在本系统的实现中,主要采用了基于临界区的方式,因此下面简单的介绍一下,临界区(Critical Section)是一段独占对某些共享资源访问的代码,在任意时刻只允许一个线程对共享资源进行访问。如果有多个线程试图同时访问临界区,那么在有一个线程进入后其他所有试图访问此临界区的线程将被挂起,并一直持续到进入临界区的线程离开。临界区在被释放后,其他线程可以继续抢占,并以此达到用原子方式操作共享资源的目的。

临界区在使用时以 CRITICAL\_SECTION 结构对象保护共享资源,并分别用 EnterCriticalSection() 和 LeaveCriticalSection() 函数去标识和释放一个临界区。所用到的 CRITICAL\_SECTION 结构对象必须经过 InitializeCriticalSection() 的初始化后才能使用,而且必须确保所有线程中的任何试图访问此共享资源的代码都处在此临界区的保护之下。否则临界区将不会起到应有的作用,共享资源依然有被破坏的可能。

## 5.3 诊断系统的软件结构

### 5.3.1 软件系统构成

基于视窗平台的移动机器人状态监测和诊断系统软件的体系结构是建构在 Windows 操作系统平台上的。选择 Windows 作为本系统的软件平台,主要在于:

(1) Windows 平台是基于优先级的抢先式多任务调度,它利用多线程并行处理技术可以构造出强大的、多功能的应用程序。多线程技术简化逻辑结构,降低模块间耦合度。另外,基于优先级的调度能防止程序僵死、保证紧急任务的处理和提高界面响应性能。

(2) 保证系统有较好的保护性和健壮性,进程的地址相互隔离有效地防止病态程序影响操作系统和其它程序。

(3) 强大的、集成的网络通信功能、良好的开放性等。

状态监测和诊断系统软件设计的基本原则是借助各种工具和手段实现规定的技术和性能指标,满足实时、有效等要求。为了实现该目标,采用模块化和多

线程设计思想进行设计，在多任务程序设计中应该遵循 H.Gomma 原则：

- (1) I/O 依赖性 (Dependency on Input/Output Device)
- (2) 时间关键性的功能(Time-Critical Functions-Hard Deadline)
- (3) 计算量大的功能(Heavy Computation Function)
- (4) 功能内聚(Functional Relations)
- (5) 时间内聚(Temporal Relations)
- (6) 周期执行的功能(Cycle Executing Function)

在以上的指导思想下，设计动机器人状态监测和诊断系统其体系结构如图 5-1 所示。

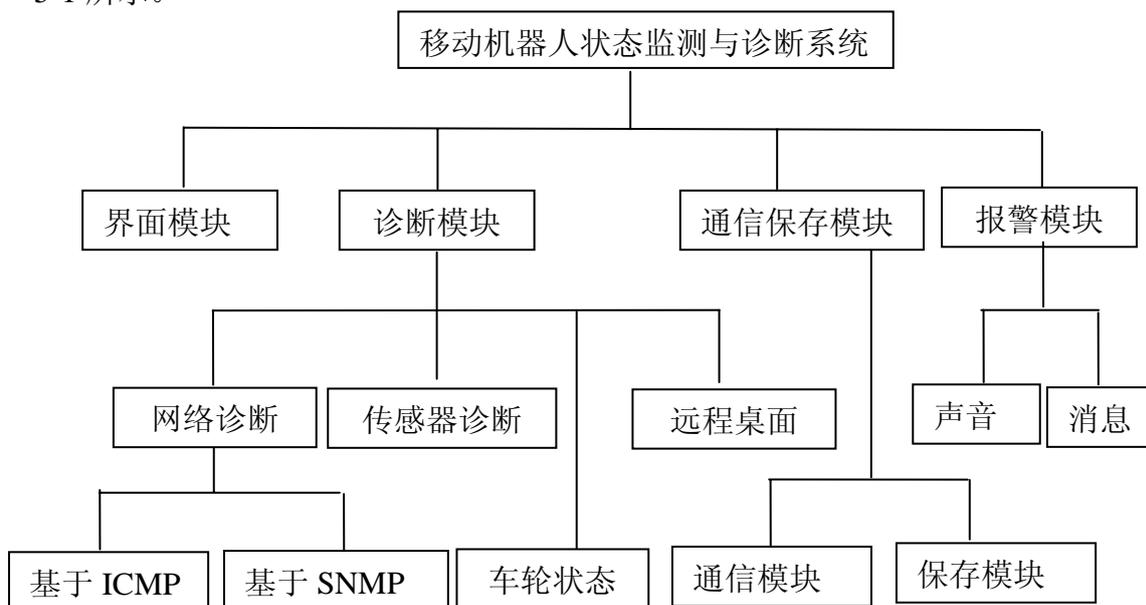


图 5-1 系统整体模块图

实时系统中，考虑到数据包的有效到达，并且还要考虑参数数据的实时性的要求，因此，使用 TCP 协议来传输数据请求包，由于请求的数据包都为小数据包，只有几十个字节，在此系统，为了减少对数据包的断帧处理，尽量减少的网络传输的延迟，提高实时性，关闭 Nagle 算法，因为 TCP 协议中的 Nagle 算法用来减少主机的小包发送数量，它通过缓冲位确认的发送数据直到能够发送一个全长度的包来做到这一点。如果不关闭就会造成约 100-120ms 的延迟，降低了实时性的要求。

### 5.3.2 软件类清单

在软件工程中，提倡低耦合和高内聚，就是要强调模块化，将单一的功能放在单一的模块中，通过接口来通讯。为了实现这目标，系统中将相互之间耦合性小的网络诊断部分、传感器监测部分、远程桌面维护部分以独立的软件 Agent 的形式存在，整个系统充分利用了 Visual C++提供的微软基础类(MFC)，MFC

(Microsoft 基本类)库封装 SDK (软件开发工具包) 结构、功能及应用程序框架内部技术, 该应用程序框架隐藏过去 Windows 程序员不得不处理的许多重复性工作。在 MFC 的基础上进行类的派生, 可以方便灵活的实现后台控制服务器所需的功能。

主要设计和实现的类有下面:

- 1、CDoctorNApp: 派生自 CWinApp, 代表整个应用程序。
- 2、CDoctorNDoc: 派生自 CDocument, 负责数据结构和文档操作。
- 3、CMainFrame: 派生自 CWndFrm, 程序主框架。
- 4、CDoctorNView: 用于显示网络状态及移动机器人的运行状态。
- 5、CSocketCom: 用于数据通信的套接字类。
- 6、CWave: 声音报警类
- 7、RobotData: 机器人传感器状态参数及激光雷达参数
- 8、CVolutage24Dlg: 机器人 24 伏电压显示对话框
- 9、CVolutage36Dlg: 机器人 36 伏电压显示对话框
- 10、 CRingBuffer: 环形缓冲区类
- 11、 CSensorDevice: 传感器设备类
- 12、 CTrayIcon: 派生自 CCmdTarget, 实现程序托盘隐藏功能。
- 13、 CTestNetDlg: 网络诊断对话框类
- 14、 CPingThread: 基于 ICMP 的诊断线程
- 15、 CWatchDServerView: 远程桌面服务器视图类
- 16、 CClientDlg: 远程桌面客户端对话框类
- 17、 CSnmpTestDlg: 基于 SNMP 的网络性能参数获取对话框
- 18、 CStatusDlg: 系统状态显示对话框

## 5.4 系统运行

图 5-2 为状态监测与诊断系统运行主界面, 系统包括了三、四章所介绍的车载计算机远程监视系统、网络性能参数获取、车轮状态监测等功能, 还提供了除外部传感器外的所有传感器的状态监测, 可以按照诊断需要对所需诊断的传感器进行配置, 能有效节约计算资源, 并能实时地显示电压、角速率等信息。系统运行时在系统托盘显示, 当发现设备出现故障时, 能产生声音、消息多种形式的报警, 并在程序的下方显示出对应出现故障的设备信心。同时为了了解系统的历史情况, 提供了对历史诊断信息的保存等功能。

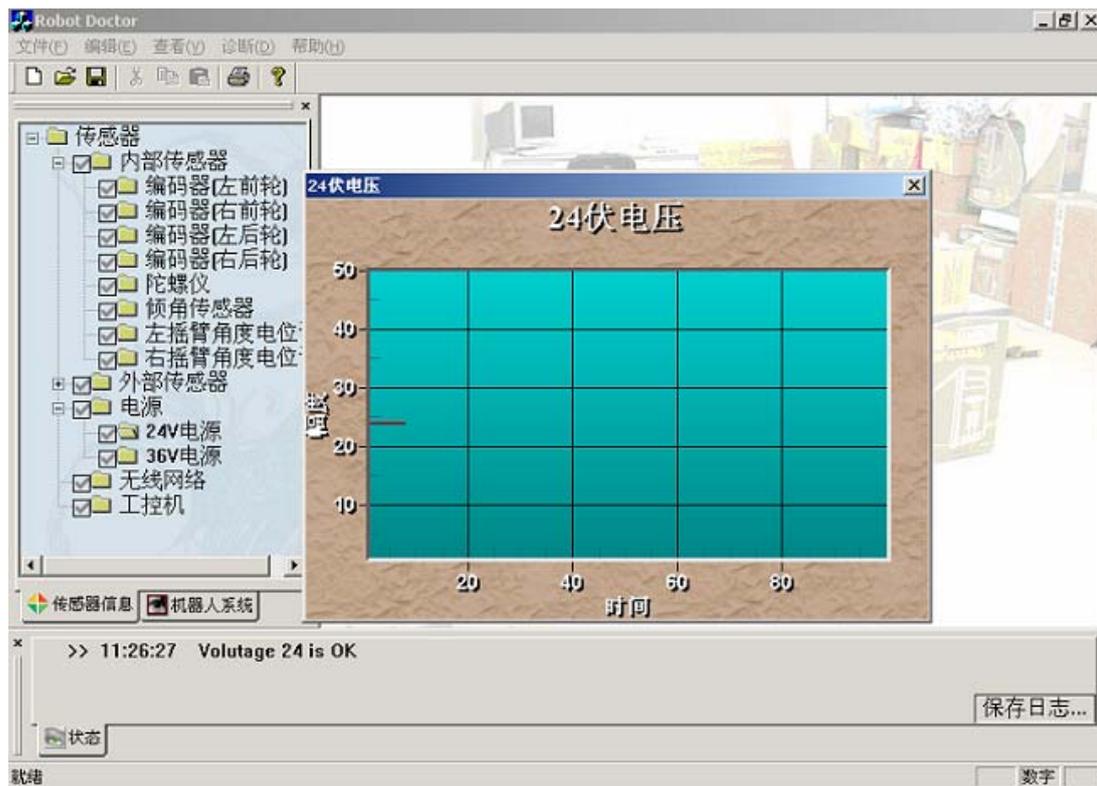


图 5-2 系统运行图

### 5.5 小结

在本章中，首先介绍了系统开发的环境，分两部分介绍了系统开发所采用的关键技术，然后说明了移动机器人状态监测与诊断系统的设计原则，描述了系统整体结构及主要的类，最后给出了状态监测与诊断系统系统运行效果图。

## 第六章 结论与展望

### 6.1 结论

本文的工作是在国家自然科学基金资助项目“未知环境下移动机器人导航控制理论与方法研究”的资助下完成的。移动机器人的状态监测与故障诊断是移动机器人系统的一个重要问题，本文前面各章完成了移动机器人状态监测系统的设计，并对移动机器人传感器的故障诊断方法开展了研究，取得的主要成果如下：

(1) 设计了一个移动机器人远程状态参数的获取子系统，采用基于无线网络的方式获取移动机器人车载的多种传感器参数，并对数据进行分析 and 存储，为后续的故障诊断研究提供了基础；对于移动机器人车载的工控机系统，提出并实现了一基于 C/S 模式的移动机器人车载工控机的监视系统；同时论文基于 ICMP 和 SNMP 设计了一网络性能监测子系统，它能获取网络延迟时间、网络流量、发送错误速率等网络性能参数，为无线网络系统安全提供了一定保障。

(2) 论文对各种诊断方法进行了分析，采用了基于信号和知识的诊断方法：研究了基于小波变换算法的传感器诊断方法，利用编码器计算获得移动机器人角速率与光纤陀螺所测角速率的冗余关系，分析设计了一针对 MORCS-1 光纤陀螺突发性故障的诊断方法；对于在未知环境中运行的移动机器人，有可能出现的车轮打滑、受阻等情况，分析设计了一个基于规则的车轮的运动状况诊断模块，它能有效监测车轮状态，并为移动机器人摆脱当前状态提供一种策略；同时分别设计实现了对车载电源、机器人系统温度、编码器等故障诊断模块。

### 6.2 展望

本论文提出的状态监测与故障诊断系统还有待于进一步的完善，具体而言，在以下几个方面将是移动机器人故障诊断中应当突出的几个方面：

(1) 移动机器人状态监测系统的进一步改进和完善，对于一些诊断方法的理论研究能应用到实际的机器人系统当中去，对于小波基的选取应采用一个更合理的方法。

(2) 故障诊断从发展前景看，当前一个最主要的方向就是各种诊断理论结合进行融合诊断。比如信号处理与神经网络的融合、小波分析与其他诊断方法的结合，融合诊断可以大大提高系统故障诊断的可靠性，更多诊断方法的融合还有待于探讨。

## 参考文献

- [1] 蔡自兴,贺汉根, 陈虹.未知环境下移动机器人导航控制的若干问题[J].控制与决策, 2002, 17(4): 385-390, 464
- [2] 朱森元.月球探测在 21 世纪初将成为航空热点之一[J].世界科技研究与发展,2000(2):9-10
- [3] Rollins E, Luntz J,Foessel,Shamah B and Whittaker W.Nomad: a demonstration of the transforming chassis [A].Proceedings of IEEE International Conference on Robotics and Automation Belgium,1998:611-617
- [4] Woods D. Review of the Soviet Lunar Exploration Programme [J], Spaceflight, 1976,18(7-8):273-290
- [5] Bickler D.Roving over Mars[J].Mechanical Engineering,1998,120(4):74-77
- [6] Laubach S L,Burdick J,Matthies L.Autonomous Path Planner Implemented on the Rocky 7 Prototype Microrover [A],ICRA 1998[C].Leuven,1998,v1:292-297
- [7] Robin R Murphy(著),杜军平,胡立成,胡金春,等译. 人工智能机器人导论[M].北京:电子工业出版社,2004
- [8] 刘建成.水下机器人推进系统故障诊断与容错控制研究[D],哈尔滨:哈尔滨工程大学博士论文,2003
- [9] 侯军虎.基于多参数的风机状态监测与故障诊断的研究[D].保定:华北电力大学博士论文,2003
- [10] 陈仲生,杨拥民.机器状态监测与故障诊断综述.机电工程[J],2000,17(5):1-3
- [11] 周东华,王桂增.故障诊断技术综述.化工自动化及仪表[J],1998,25(1):58-62
- [12] Willsky,A.S. A Survey of Design Methods for Failure Detection in Dynamic System. Automation,1976.12:601-611
- [13] Massoumnia, M. A. A Geometric Approach to Failure Detection and Identification in Linear Systems. Ph.D.Thesis, MIT, Cambridge,MA,1986
- [14] Frank, P.M. Fault Diagnosis in Dynamic System Using Analytical and Knowledge Based Redundancy-A Survey and Some New Results. Automatica, 1990.26(3): 459-474
- [15] Dan,T, Horak.Failure Detection in Dynamic Systems with Modeling Errors. J. Guidance. Novdec,1988:508-516
- [16] Dan, T., Horak. System Failure Isolation in Dynamic Systems.J. Guidance. Novde,1990:1075-1082
- [17] Ding, X. and Frank, P. M. Fault Detection via Factorization Approach. System

- and Control Letters, 1990; 14(5): 431-436
- [18] 周东华,叶银忠.现代故障诊断与容错控制[M].清华大学出版社. 2000:1-10, 60-65
- [19] 张育林,李东旭.动态系统故障诊断理论与应用[M].国防科技大学出版社. 1997:1-2
- [20] 刘晓颖.复杂过程的智能故障诊断技术及其在大型工业窑炉中的应用研究[D].长沙:中南大学博士论文,2003
- [21] Mallat, S. A Theory of Multiresolution Signal Decomposition: The Wavelet Transform. IEEE Trans. PAMI, 1989; 11(7): 674-693
- [22] Mallat, S. Multiresolution Approximation and Wavelet Orthonormal Bases of  $L_2$ . Trans. Amer. Math. Soc., 1989; 315: 69-87
- [23] Daubechies, I. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. IEEE Trans. IT, 1990;36(5): 961-1005
- [24] Daubechies, I. Where Do Wavelets Come From?—A Personal Point of View. Proc. IEEE, 1996;84(5): 510-513
- [25] 叶昊,王桂增,方崇智.小波变换在故障诊断中的应用[J].自动化学报,1997; 23(6):736-741
- [26] 叶昊,王桂增,方崇智等.一种基于小波变换的导弹运输车辆故障诊断方法[J].自动化学报,1998;24(3):301-306
- [27] Ge, W. F. Detection of Faulty Components via Robust Observation. Int. J. Control, 1988; 47(2): 581-599
- [28] Frank, P. M. and Ding, X. Survey of Robust Residual Generation and Evaluation Methods in Observer Based Fault Detection Systems. J. Process. Control, 1997; 7(6): 403-424
- [29] Isermann, R. Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing-Tutorial Paper. Automatica, 1993; 29(4): 815-835
- [30] Frank, P. M. New Developments Using AI in Fault Diagnosis. Proc. of IFAC Workshop on Artificial Intelligence in Real-Time Control, Bled, Slovenia, 1995
- [31] Koppen-Seliger, B. and Frank, P. M. Neural Networks in Model-Based Fault Diagnosis. Proc. of IFAC World Congress, San Francisco, USA, 1996: 67-72
- [32] 朱向彬,曹作良,冯玉伯.基于参数辨识的移动机器人故障检测及诊断系统[J].天津理工大学学报 2005;21(3):4-7
- [33] 段琢华,蔡自兴,于金霞.未知环境中移动机器人故障诊断与容错控制技术综述[J].机器人 2005;21(7):373-379

- [34] 邹小兵.移动机器人原型的控制系统设计与环境建模研究[D].长沙:中南大学  
博士论文,2005
- [35] 叶涛.基于传感器信息的移动机器人导航与控制研究[D].北京:中国中科院自  
动化研究所博士论文,2003
- [36] 陈一民,叶益东.网络 DDE 和 Windows Socket 在机器人远程控制中的应用[J].  
计算机工程,1998,24(9):35-38
- [37] 蒋东兴,林鄂华等. Windows Sockets 网络程序设计大全[M].北京:清华大学  
出版社,2000:1-384
- [38] Shu,LihChyun.On avoiding remote blocking via real-time concurrency control  
protocols.Journal of System and Software,2003,68(2):121-136
- [39] 李彦,卢虎.计算机远程控制技术及其实现[J].空军工程大学学报, 2002, 3(3):  
87-90
- [40] 汤双清,付建科,施仲光,廖道训.基于 Internet 的工业机器人的远程控制[J].计  
算机应用研究,2002,(6):151-153
- [41] 钟新华.移动机器人结构设计与运动方法研究.长沙:中南大学硕士论文,2005
- [42] 朱东辉.基于 WinSock 通信的远程屏幕抓取方法与实现[J].计算机应用研  
究.2005,(8):204-205
- [43] 何志兰,崔杜武,莫毅.一种远程屏幕图像实时传输方法[J].计算机应用研  
究.2005,(8):258-259
- [44] 张友生.远程控制编程技术[M].电子工业出版社,2001:1-189
- [45] 陈 坚,陈 伟. Visual C++网络高级编程[M].人民邮电出版社,2001:33-35
- [46] 凌军,杨桂文,刘羽中,曹阳.Internet 网络性能实时监测关键技术研究[J].武汉  
大学学报(理学版),2001,47(3)
- [47] 唐海娜,李俊.网络性能监测技术综述[J].计算机应用研究,2004:(8),10-13
- [48] W.Richard Stevens (著),范建华,胥光辉,张涛等译.TCP/IP 详解(卷 1: 协议)[M],  
机械工业出版社.1999
- [49] 粟阿方.基于 SNMP 协议的网络管理平台研究.太原:华北工学院硕士论文,  
2001
- [50] 徐胜红,吴进华,胡志强等.组合导航系统故障检测的小波变换方法研究[J].宇  
航学报,2003:24(1),111-114
- [51] 黄治军,雷勇.传感器故障检测的仿真研究[J].计算机仿真. 2005:22(1).110-112,

## 附录

附录一：SNMP++下复杂变量的访问：

```
BOOL CSnmpTestDlg::Mib_Walk(Oid *item_oid, CTarget *target)
{
    int status;
    BOOL tag=TRUE;
    int index=0;
    GenAddress address;
    target->get_address( address);
    Vb vb;
    Pdu pdu;
    Snmp snmp(status);
    if(status!=SNMP_CLASS_SUCCESS)
    {
        AfxMessageBox(snmp.error_msg(status));
        return FALSE;
    }
    vb.set_oid(*item_oid);
    pdu +=vb;
    status =SNMP_CLASS_SUCCESS;
    while((status==SNMP_CLASS_SUCCESS)&&(tag==TRUE))
    {
        if((status=snmp.get_next(pdu,*target))==SNMP_CLASS_SUCCESS )
        {
            pdu.get_vb(vb,0);
            Oid full_oid;
            vb.get_oid(full_oid);
            if(item_oid->nCompare(item_oid->len(),full_oid)==0)
            {
                index++;
                if(index==SELINDEX)
                {
                    m_inumber=atoi(vb.get_printable_value());
                }
            }
        }
    }
}
```

```

        }
        pdu.set_vb(vb,0);
    }
    else
    {
        tag=FALSE;
    }
}
else
{
    AfxMessageBox("SNMP++ Error");
}
}
return TRUE;
}

```

#### 附录二：车轮异常检测：

```

void CDlgWheelDiagnose::WheelState(long Uset[], long Ytest[], long RealV[], char
State_Wheel[], long &State_Avoid, long AvoidStrategy[])
{
    int Threcoeder=4;//前后轮速度差阈值
    double Thzero=0.1;//近似为 0 的阈值
    double TH03=0.7;//被卡与受阻的阈值
    double THavoid=0.5;//是否避让阈值
    int THdistance=10;
    int THMaxdist=2000;//避让距离阈值
    double Vleft;//左轮实际速度 long
    double Vright;//右轮实际速度 long
    double u[2];//左,右轮设定速度
    double y[4];//左前,左后,右前,右后轮测量值
    for(int i=0;i<2;i++)
    {
        u[i]=Uset[i]/10.;//数值转换
    }
    for(int j=0;j<4;j++)

```

```
{
    y[j]=Ytest[j]/10.;//数值转换
}
//-----
//Step1 确定左右轮实际速度（逻辑传感器）
int State_Left=0;
int State_Right=0;
double Vleft_diff=y[0]-y[1];
double Vright_diff=y[2]-y[3];
if (fabs(Vleft_diff) > Threocoder)
{
    State_Left=1;
}
if (fabs(Vright_diff)>Threocoder)
{
    State_Right=1;
}

if(1==State_Left)
{
    Vleft=absmin(y[0],y[1]);
}
else
{
    Vleft=(y[0]+y[1])/2;
}
if(1==State_Right)
{
    Vright=absmin(y[2],y[3]);
}
else
{
    Vright=(y[2]+y[3])/2;
}
```

```
//End Step1
//Step2 确定车轮异常
for(int k=0;k<4;k++)
{
    State_Wheel[k]=char(0);
    //State_Wheel[k]='0';
}
if(1==State_Left) //left:
{
    if(fabs(y[0])<fabs(y[1]))
    {
        if (fabs(u[0])>Thzero && fabs(y[0]/u[0])<=Thzero)
        {
            State_Wheel[0]=char(1); //left front wheel : 被卡(1)
        }
        else
        {
            State_Wheel[0]=char(2); //left front wheel : 受阻(2)
        }
        State_Wheel[1]=char(3); //left rear wheel : 打滑(3)
    }
    else
    {
        if (fabs(u[0])>Thzero && fabs(y[1]/u[0])<=Thzero)
        {
            State_Wheel[1]=char(1); //left rear wheel : 被卡(1)
        }
        else
        {
            State_Wheel[1]=char(2); //left rear wheel : 受阻(2)
        }
        State_Wheel[0]=char(3); //left front wheel : 打滑(3)
    }
}
}
```

```
else//State_Left==0
{
    if( fabs(u[0])>Thzero) // only "set speed >THzero" need to be considered
    {
        if (fabs(Vleft/u[0])<=Thzero)
        {
            State_Wheel[1]=char(1);
            State_Wheel[0]=char(1);
        } //left wheel : 被卡(1)
        else
        {
            if(fabs(Vleft/u[0])<=TH03) //Threshold ,受阻
            {
                State_Wheel[1]=char(2);
                State_Wheel[0]=char(2);
            } //left wheel : 受阻(2)
        }
    }
}
if(1==State_Right) //right:
{
    if(fabs(y[2])<fabs(y[3]))
    {
        if (fabs(u[1])>Thzero && fabs(y[2]/u[1])<=Thzero)
        {
            State_Wheel[2]=char(1); //right front wheel : 被卡(1)
        }
        else
        {
            State_Wheel[2]=char(2); //right front wheel : 受阻(2)
        }
        State_Wheel[4]=3; //right rear wheel : 打滑(3)
    }
}
else
```

```
{
    if(fabs(u[1])>Thzero && fabs(y[3]/u[1])<=Thzero)
    {
        State_Wheel[3]=char(1); //right rear wheel : 被卡(1)
    }
    else
    {
        State_Wheel[3]=char(2); //right rear wheel : 受阻(2)
    }
    State_Wheel[2]=char(3); //right front wheel : 打滑(3)
}
}
else //State_Right==0
{
    if(fabs(u[1])>Thzero) // only "set speed >THzero" need to be considered
    {
        if(fabs(Vright/u[1])<=Thzero)
        {
            State_Wheel[3]=char(1);
            State_Wheel[2]=char(1);
        } //right wheel : 被卡(1)
        else
        {
            if(fabs(Vright/u[1])<=TH03) //Threshold ,受阻
            {
                State_Wheel[3]=char(2);
                State_Wheel[2]=char(2);
            } //right wheel : 受阻(2)
        }
    }
}
}
//end Step2
//Step3 确定避让策略
State_Avoid=0;
```

```

    if(State_Wheel[0]==char(1) || State_Wheel[1]==char(1) || State_Wheel[2]==char(1) ||
State_Wheel[3]==char(1))
    {
        State_Avoid=1;
    }
    else
    {
        if ((State_Wheel[0]==char(2) || State_Wheel[1]==char(2) ) &&
fabs(Vleft/u[0])<THavoid)
        {
            State_Avoid=2;
        }
        if((State_Wheel[2]==char(2) || State_Wheel[3]==char(2) ) &&
fabs(Vright/u[1])<THavoid)
        {
            State_Avoid=2;
        }
    }
    double TempAvoid[4];
    if (State_Avoid>0)
    {
        TempAvoid[1]=-(u[0]-Vleft/2);// left speed
        TempAvoid[2]=-(u[1]-Vright/2);// right speed
        if(TempAvoid[1]>0 && TempAvoid[2]>0)
        {
            TempAvoid[0]=1; //move forward
        }
        if(TempAvoid[1]<0 && TempAvoid[2]<0)
        {
            TempAvoid[0]=2;//move backward
        }
        if(TempAvoid[1]>0 && TempAvoid[2]<0)
        {
            TempAvoid[0]=4;//turn right

```

```
    }
    if(TempAvoid[1]<0 && TempAvoid[2]>0)
    {
        TempAvoid[0]=3;//turn left
    }
    //distance
    double speed_set=(u[0]+u[1])/2;
    double speed_real=(Vleft+Vright)/2;
    double
TempPara=fabs((speed_set-speed_real)*speed_set/(0.5+speed_real+1e-9)*THdistance );
    TempAvoid[3]=min(THMaxdist,TempPara);
}
//end step3
//-----
//数据格式转换
    m_LeftWReal=Vleft;
    m_RightWReal=Vright;
// AfxMessageBox(State_Wheel);//查看状态值
//轮子状态
    switch(State_Wheel[0])//左前轮状态
    {
    case char(0): m_LeftFrontWState="正常";
        break;
    case char(1): m_LeftFrontWState="被卡";
        break;
    case char(2): m_LeftFrontWState="受阻";
        break;
    case char(3): m_LeftFrontWState="打滑";
        break;
    default:    m_LeftFrontWState="异常";
        break;
    }
    switch(State_Wheel[1])//左后轮状态
    {
```

```
case char(0): m_LeftBackWState="正常";
            break;
case char(1): m_LeftBackWState="被卡";
            break;
case char(2): m_LeftBackWState="受阻";
            break;
case char(3): m_LeftBackWState="打滑";
            break;
default:    m_LeftBackWState="异常";
            break;
}
switch(State_Wheel[2])//右前轮状态
{
case char(0): m_RightFrontWState="正常";
            break;
case char(1): m_RightFrontWState="被卡";
            break;
case char(2): m_RightFrontWState="受阻";
            break;
case char(3): m_RightFrontWState="打滑";
            break;
default:    m_RightFrontWState="异常";
            break;
}
switch(State_Wheel[3])//右后轮状态
{
case char(0): m_RightBackWState="正常";
            break;
case char(1): m_RightBackWState="被卡";
            break;
case char(2): m_RightBackWState="受阻";
            break;
case char(3): m_RightBackWState="打滑";
            break;
```

```
default:    m_RightBackWState="异常";
            break;
        }
//end of Wheel State
RealV[0]=(long)(Vleft*10);//Vleft:左轮实际速度
RealV[1]=(long)(Vright*10);//Vright:右轮实际速度
AvoidStrategy[0]=(long)TempAvoid[0];//方向, 1: 向前 2: 向后 3: 向左转 4: 向右转 long
AvoidStrategy[1]=(long)(TempAvoid[1]*10);//左轮速度 long
AvoidStrategy[2]=(long)(TempAvoid[2]*10);//右轮速度 long
AvoidStrategy[3]=(long)(TempAvoid[3]);//移动距离 long
}
```

## 致 谢

在中南大学求学期间，我得到了许多老师、同学和亲人的帮助，在此深表感谢。

首先，深深的感谢我的尊师蔡自兴教授，本文是在蔡老师悉心的指导和热忱的关怀下完成的。在学习上蔡老师给予我悉心的指点，他以认真的工作作风、一丝不苟的求实精神、平易近人的态度，给予我多方面的启迪。在生活上蔡老师对我照顾入微，以他的言传身教使我明白了许多做人的道理，这些都将使我在以后的生活与工作中受益匪浅。在此，谨向辛勤培养我的导师致以崇高的敬意和衷心的感谢。

感谢师母翁环老师，正是她慈母般的关怀，才使我们这些异地求学的学生感到家庭般的温暖。

特别感谢移动机器人研究组段琢华副教授，在本人课题研究和论文撰写期间，是他给予了我最大的指导和帮助，使得课题研究和论文能顺利完成。

在这里我还要感谢于金霞老师、王璐老师和邹小兵博士在工作和生活上给予我的指导，使我能顺利完成研究工作，同时感谢所里唐璘教授、肖晓明副教授、魏世勇老师的帮助。

本文的完成还离不开一个团结奋进的团体——中南大学智能系统与智能软件研究所的老师同学，他们是：陈爱斌副教授、陈白帆老师、高平安老师、刘丽珏老师、潘薇师姐、文志强老师、郑敏捷、戴博、王大勇、金燕、王小勇、谢峰、黄明登、宁火明、肖正、彭梦、胡华梅等，与他们相处的日子给我留下了非常美好的回忆。

另外，我要衷心的感谢我的父母和姐姐们，二十多年来，他们始终如一地支持我的求学生涯，给予了我物质和精神上极大的帮助，本文是献给她们最好的礼物。

最后，感谢国家自然科学基金重点项目：“未知环境下移动机器人导航控制理论与方法研究”（课题编号：60234030）对本研究的资助。

签名：

日期：

## 攻读学位期间主要的研究成果

- [1] 章慧团,蔡自兴.基于数学形态学的突发性故障快速定位方法.计算机测量与控制.2006,14(1):9-11
- [2] CAI Zi-xing, DUAN Zhuo-hua, ZHANG Hui-tuan, YU Jin-xia. Abnormal movement state identification and avoidance strategy for mobile robots. Journal of Central South University of Technology.2006 (Accepted)
- [3] Zhuohua Duan, Zixing Cai, Huituan Zhang, Jinxia Yu. An Adaptive Particle Filter Algorithm for Mobile Robot Fault Diagnosis with Incomplete Models. International Conference on Pattern Recognition. 2006 (Accepted)
- [4] 参与国家自然科学基金重点项目：未知环境下移动机器人导航控制的理论与方法研究（项目批准号：60234030）。
- [5] 参与中南大学“大学生创新创业启航行动”重点资助创新项目：“基于粒子滤波器的移动机器人故障诊断方法研究”（ZA007）。