

1000

1000

1000

1000

1000

1000



摘 要

虚拟仪器概念的出现掀起了测试领域新的浪潮,随着测试系统任务的复杂化,对数据的处理及传输技术提出了更高的要求。在微电子和计算机技术高速发展的背景下,NI公司研制和推出的基于多总线系统的虚拟仪器开发平台LabVIEW,可以灵活地组建不同复杂程度的虚拟仪器自动测试系统。

本文结合目前国内外虚拟仪器技术的研究现状,阐述了基于虚拟仪器的测控系统网络化的研究意义,研究设计了一种基于虚拟仪器的远程数据处理与传输系统。文章首先对数据采集的系统误差特性进行了分析,并采用最小二乘曲线拟合算法进行系统误差补偿,使测量数据满足系统的精度要求。通过分析比较了几种常用的数据压缩算法,重点探讨了适用于测控系统的LZW数据压缩算法,并结合测控数据的特点对该算法作出改进,使其压缩比与原LZW算法相比降低了5%。随后介绍了ARINC429航空数据总线技术,研究了通信差错控制的相关理论,并探讨了以太网应用于测控系统中的关键技术,重点分析了交换式以太网中海量数据传输时间。在此基础上,本文设计了符合实际工程要求的测试系统,并给出了该系统的体系结构。采用模块化设计,对系统中各个功能模块在LabVIEW开发平台下的具体实现进行了详细的阐述,实现了基于虚拟仪器的本地数据采集和处理,实现了基于ARINC429总线和以太网技术的可靠数据传输和远程控制等功能,并应用于某型号综合测控系统中。

关键词: 虚拟仪器 LabVIEW Lzw算法 循环冗余校验

Abstract

The emergence of virtual instrumentation has set off a fresh wave in the testing field. With the complexity of tasks in testing systems, people have put forward higher requirement for the data processing and transmission technology. Along with the rapid development of micro-electronics and computer sciences, NI has developed the virtual instrument platform LabVIEW based on multi-bus system, which can build different complexity of virtual instrument in the automatic testing system

In this paper , combining with the the current research of virtual instrument technology, the research meaning of networked measurement and control system based on virtual instrument is introduced, the design of a remote data processing and transmission system based on virtual instrumentation is discussed. Firstly, the system error which emerge in the course of data acquisition is analyzed, then the least-squares curve fitting algorithm for system error compensation is used, and this approach makes the measured data satisfied to system accuracy. Several common data compression algorithm are analyzed and compared in this article, especially LZW data compression algorithm, then, an improved method combined with the characteristics of measurement data is proposed, which makes the compression ratio reduced by 5% compared with the original LZW algorithm. Then the ARINC429 bus is introduced, which is widely applied to the field of aviation and aerospace at present. The theory of the error control technology is studied, and the network delay in switch ethernet when there are huge volume of data need to transfer is mainly analyzed. On this basis, the paper designs a testing system which corresponds to the actual engineering requirement, and presents the architecture. This paper adopts the idea of modularization design by making use of the functions of the LabVIEW. A set of virtual instrument measurement system has been designed for collecting and analysing the measured data , and each module are detailed. Local-based data acquisition and processing, reliable data transmission and remote control functions are implemented on the basis of ARINC429 bus and Ethernet technology , and finally applied to a type of integrated Test and Control System.

Keywords: Virtual instrument LabVIEW LZW CRC

目 录

第一章 绪论.....	1
1.1 引言.....	1
1.2 课题研究的背景及意义.....	2
1.3 论文研究内容及结构.....	3
1.3.1 论文的研究内容.....	3
1.3.2 论文的结构安排.....	3
第二章 虚拟仪器技术及 LabVIEW 开发环境的研究.....	5
2.1 虚拟仪器的基本概念.....	5
2.1.1 虚拟仪器的概念和与传统仪器的比较.....	5
2.1.2 虚拟仪器的系统结构.....	6
2.2 虚拟仪器开发平台与 LabVIEW 开发环境.....	7
2.2.1 虚拟仪器的开发平台.....	8
2.2.2 LabVIEW 开发环境的特点.....	8
2.2.3 LabVIEW 调用外部程序.....	9
2.3 本章小结.....	10
第三章 数据处理与压缩技术的研究.....	11
3.1 系统误差的分析与处理.....	11
3.1.1 系统误差分析.....	11
3.1.2 最小二乘拟合理论基础.....	12
3.1.3 系统误差处理改进方法.....	14
3.2 数据压缩理论基础.....	17
3.2.1 数据压缩的思想.....	17
3.2.2 数据压缩的性能评价.....	18
3.2.3 数据压缩的分类.....	18
3.3 LZW 算法的研究.....	20
3.3.1 LZW 算法编码与解码.....	20
3.3.2 LZW 算法的优缺点.....	21
3.3.3 LZW 算法在测试系统中改进.....	22
3.4 本章小结.....	25
第四章 基于虚拟仪器的传输系统的研究.....	27
4.1 虚拟仪器网络化的研究与设计.....	27

4.1.1 组建网络化虚拟仪器系统的两种模式	27
4.1.2 多总线分布式网络测控系统的设计	29
4.2 ARINC429 总线通信的研究	31
4.2.1 ARINC429 数字信息传输规范	31
4.2.2 ARINC429 传输格式和类型	31
4.2.3 CRC 编码原理及实现	32
4.2.4 综合检错技术及应用	34
4.3 以太网应用于测控系统的研究	37
4.3.1 以太网应用于测控系统的优势	37
4.3.2 TCP/IP 协议及 Winsock 开发过程	37
4.3.3 基于 TCP 的文件传输设计	40
4.3.4 文件传输时间的测量	42
4.4 本章小结	43
第五章 基于虚拟仪器的某型号综合测控系统的实现	45
5.1 系统方案规划	45
5.1.1 系统总体结构	45
5.1.2 自动测试系统硬件组成	46
5.1.3 测试系统的软件结构	47
5.2 数据处理技术在系统中的实现	48
5.2.1 误差补偿的软件实现	48
5.2.2 数据压缩算法的实现	49
5.2.3 离散量与模拟量数据采集	51
5.3 测控系统通信模块的设计与实现	52
5.3.1 与自动管理系统通信	52
5.3.2 与其他分系统的通信	54
5.4 软件系统界面设计	56
5.5 本章小结	57
第六章 总结和展望	59
致谢	61
参考文献	63
作者在读期间的研究成果	67

第一章 绪论

1.1 引言

随着仪器技术、现代测试技术、计算机通信技术与网络技术的高速发展,传统的测试仪器向一种新概念的测试仪器——虚拟仪器(Virtual Instrument 或 Virtual Instrumentation, 简称 VI) 发展^[1]。最早的仪器可以追溯到中国古代四大发明之一的指南针。从 20 世纪 50 年代至今, 仪器大致经历了四代发展历程, 即传统模拟仪器、数字化仪器、智能仪器和虚拟仪器^{[2][3]}。1986 年, 美国国家仪器(National Instruments, 简称 NI) 公司首次正式提出“虚拟仪器”的概念。1987 年, VXI plug&play(即插即用)系统联盟发布虚拟仪器的各种规范。1998 年, NI 公司又发布了虚拟硬件(Virtual Hardware, VH)和可互换虚拟仪器(Interchangeable Virtual Instruments, IVI)的概念, 提出了软件是虚拟仪器的核心^{[4][5]}。

直到今天, 虚拟仪器技术仍在快速发展, 其本身的内涵不断丰富, 外延不断扩展, 功能和性能不断地提高。由于虚拟仪器充分发挥了通用计算机的功能, 节省了大量的专用硬件设备和软件开发成本, 缩短了开发周期, 而且提供了实现仪器编程和数据采集系统的便捷途径, 提供强大功能的同时保证了系统的灵活性, 使工作效率大大提高^{[6][7]}。虚拟仪器这些特点使其广泛应用于科研开发、工业自动化和控制系统、电力和电子工程、图像的采集和分析处理、远程监控、物矿勘探、医疗、军事、航天等诸多领域^[8], 且应用领域还在不断拓宽, 并对现代测控技术产生了深远不可估量的影响。业界普遍认为, 伴随着 PC、半导体和软件功能的进一步更新, 未来虚拟仪器技术的发展将为测试系统的设计提供一个极佳的模式, 使工程师们在测量和控制方面得到无与伦比的强大功能和灵活性。

由于虚拟仪器具有性价比高、开放性等优势, 从而迅速成为了测控仪器新的经济增长点。发达国家对以软件为核心的虚拟仪器技术蕴含的巨大潜力趋之若鹜, 并在这一领域的研究上投入了巨资。1988 年国际上有五家制造商推出 30 种产品, 到 1994 年底, 虚拟仪器制造厂已达 95 家, 共生产 1000 多种虚拟仪器产品, 销售额达 2.93 亿美元, 占整个仪器销售额 76 亿美元的 4%^[9]。虚拟仪器市场潜力巨大。国外仪器生产厂商在上个世纪 80 年代末开始了虚拟仪器的研究工作。在 90 年代, 以美国 NI 公司、Motorola 公司为代表开始了虚拟仪器的商业化推广。基于虚拟仪器的测控系统的开发是测试领域发展的重要方向, 具有极大的市场潜力和应用前景, 将给社会带来极大的效益^[10]。

1.2 课题研究的背景及意义

从虚拟仪器概念提出至今,有关虚拟仪器技术的研究方兴未艾。研究人员在虚拟仪器硬件接口、虚拟仪器软件及其设计方法等方面做了许多有意义的研究工作,并已开发了许多实用的虚拟仪器系统。各种虚拟仪器的开发平台的研制为虚拟仪器的推广奠定了基础^[9];软件工程领域的新方法新理论在虚拟仪器设计中得到了广泛的应用;虚拟仪器开发向标准方向发展;并采用接口总线组建复杂的虚拟仪器系统。另外随着以 Internet 为代表的网络作为信息交换的工具渗透到工业、科研及日常生活的各个领域,为虚拟仪器的网络化发展提供了契机。使测试人员无论任何地点、任何时间都能进行远程操作、获取测试信息。突破了早期虚拟仪器将仪器的三大功能模块(信号采集、信号分析和显示)都集中在一台机器上的模式,可以将不同计算机上的虚拟仪器、外围设备、被测试点以及数据库等功能模块连入网络,实现资源共享。虚拟仪器向网络化、智能化发展。

国内虚拟仪器研究的起步较晚,最早的研究也是从引进和消化 NI 的产品开始,但经过一段时间的发展,我国在虚拟仪器的研究方面取得了一定的进展^{[11][12]};在科研领域方面,西安交通大学采用面向对象技术研究了可组态生成不同虚拟仪器的可视化虚拟仪器软件开发平台;重庆大学提出了虚拟仪器产品的网络化开发方法以及智能化空间的思想;成都电子科技大学开发出了具有自主知识产权的 VISA 库;哈尔滨工业大学电气工程系开发的虚拟仪器软件开发平台—ATS95 可以实现对 VXI、GPIB 等总线接口的控制;如华中科技大学对分布式网络测控系统的现场应用研究取得了一些重要成果。另外,国家自然科学基金委员会已将虚拟仪器研究作为现代机械工程科学前沿学科之一,曾被列为“十五”期间优先资助领域^[13]。随着计算机及总线技术的发展,虚拟仪器将逐渐取代传统仪器而成为测试仪器的主流。我国专家预测,未来几年我国虚拟仪器行业的产值将达到仪器仪表行业总产值的 50%。我国虚拟仪器技术将走上硬件平台标准化和通用化,软件模块化、标准化、专业化、系列化和网络化的道路。随着社会的发展,测试对象越来越复杂,测控范围也越来越广,需要解决虚拟仪器测控系统中远程数据采集与控制、远程设置故障诊断、远程监控等具体问题。

本文在上述发展背景的基础上对基于虚拟仪器技术的数据处理与传输做了探索性尝试,设计基于 C/S 模式的网络化虚拟仪器系统,对系统中采集的数据误差进行补偿处理,对海量数据进行压缩编码处理,并在对以太网与 ARINC429 总线技术研究的基础上,实现基于虚拟仪器的数据传输与远程控制,最后通过人机界面的形式显示和控制,并应用于某型号的综合测控系统中。

1.3 论文研究内容及结构

1.3.1 论文的研究内容

本文围绕某型机载设备测控系统的建立,以虚拟仪器技术为核心,以 LabVIEW 开发平台为实现手段,设计了基于虚拟仪器的数据处理与传输平台,以实现数据的采集的准确性和多种类型的数据在 ARINC429 总线及网络中传输的有效性 & 可靠性,并以图形化的方法显示。本论文完成的主要工作如下:

1) 实现了以 LabVIEW 为开发平台的现场多通道数据采集系统,主要负责对多路模拟量与离散量的实时采集。为了满足测量系统精度要求,对测量数据中的系统误差进行补偿。

2) 对以虚拟仪器技术为核心的某型号综合测控系统进行总体规划,设计了基于虚拟仪器的数据处理与传输系统,实现了测控系统的数据传输与远程控制等功能。

3) 为优化网络传输效率,对需要上传的大数据量数据进行无损压缩处理,并结合测控数据的特点,对 LZW 算法进行改进,改善压缩性能,从而通过减少网络中传输的数据量减缓网络压力。

4) 为提高数字信息传输的可靠性,系统与被测对象的通信数据采用循环冗余校验与奇偶校验码相结合的编码方式,对 CRC 校验码编码算法进行了研究,并在 LabVIEW 中实现。

5) 利用 LabVIEW 可视化的开发特点,将系统软件设计成人-机交互界面的形式,为测试人员提供关于被测过程和测试系统本身运行情况的全部信息,为测试人员直观地进行操作提供各种手段。

1.3.2 论文的结构安排

本文结构具体安排如下:

第一章 首先介绍本论文的研究背景及意义。在此基础上,介绍国内外虚拟仪器技术的研究状态、发展趋势,以及在工业测控系统中的应用,最后介绍本论文的主要内容和结构安排。

第二章 阐述虚拟仪器的基本概念,介绍了虚拟仪器的软硬件结构。并对比了几种虚拟仪器开发的软件平台,结合本系统特点选择 LabVIEW 作为本系统的开发环境,并介绍了 LabVIEW 的开发环境的特点及调用外部程序的方法,为系统的开发实现奠定了基础。

第三章 对系统误差的处理方法进行研究,并讨论如何将其应用在虚拟仪器系

统中；介绍了三种无损压缩算法并进行对比，重点研究 LZW 压缩算法，分析其编译码原理和实现方法，并对改进后算法压缩比进行统计，为数据传输效率的提高奠定了基础。

第四章 比较了组建网络化虚拟仪器的两种模式，结合本系统需要，设计了基于虚拟仪器的数据传输系统，并对传输系统中的 ARINC429 总线技术和以太网技术进行研究，在此基础上研究了循环冗余校验编译码原理，为系统传输可靠性的实现做了铺垫。搭建海量数据传输时间的测量平台，并对测量结果进行分析。

第五章 实现了基于虚拟仪器的数据采集与传输系统的设计。完成了系统误差补偿方法、压缩算法在 LabVIEW 开发平台上的实现，重点在系统实现的层面上，结合系统需求及测试数据特点构造数据帧格式，并给出详细的数据处理和传输流程，最终给出了 LabVIEW 开发环境下的系统运行界面。

第二章 虚拟仪器技术及 LabVIEW 开发环境的研究

2.1 虚拟仪器的基本概念

2.1.1 虚拟仪器的概念和与传统仪器的比较

虚拟仪器（Virtual Instrument 或 Virtual Instrumentation，简称 VI）的概念是由 NI 公司于 1986 年提出的，是具有虚拟仪器面板的个人计算机测量仪器^{[13][14]}。它将计算机资源与仪器硬件、数字信号处理技术相结合，在系统内共享软硬件资源，既有传统仪器的功能，又有一般仪器所没有的特殊功能^[15]。

与传统仪器一样，它同样划分为数据采集、数据分析处理、显示结果三大模块功能^[16]，其内部功能划分如图 2.1 所示。

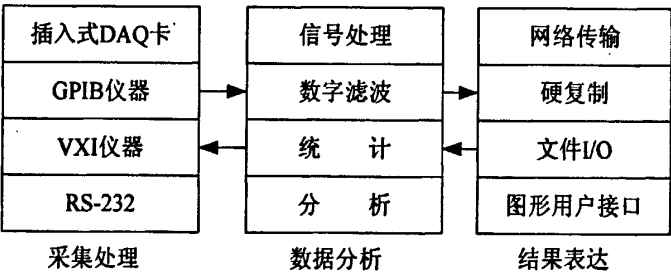


图 2.1 虚拟仪器内部功能划分

考虑到一些传统仪器提供了特定的测量，虚拟仪器软件通常提供了与常用普通仪器总线（如 GPIB、串行总线和以太网）相连接的函数库，因此可与传统仪器完全兼容。测试要求改变时不需重新购买整台仪器，只要增加或更换仪器软硬件模块，就可以构成新的仪器。只需将具有一种或多种功能的通用模块结合，并调用不同的软件模块，就能组成不同的仪器功能。而且，它能充分发挥计算机的作用，便于与计算机通信相结合来建立计算机网络，组建复杂的测试系统。

把虚拟仪器与传统仪器加以比较，有许多不同，比较结果如表 2.1 所示。传统仪器与虚拟仪器最重要的区别在于：虚拟仪器把由厂家定义仪器功能的方式转变为由用户自己定义仪器功能，用户可根据测试功能的需要，自己设计所需要的仪器系统^[7]。决定虚拟仪器具有传统仪器不可能具备的特点的根本原因在于：虚拟仪器的关键是软件。虚拟仪器具有传统仪器无法比拟的优点，从而成为仪器未来的发展趋势。

表 2.1 虚拟仪器与传统仪器比较

传统仪器	虚拟仪器
关键是硬件	关键是软件
功能由仪器厂家定义	用户定义仪器功能
图形界面小、人工读数、信息量小	界面图形化、计算机直接读取数据并分析处理
与其他仪器的设备连接有限	可方便地与网络外设及多种仪器连接
技术更新周期长（5~10 年）	技术更新周期短
系统封闭、功能固定、扩展性差。	资源可重复使用
开发和维护费用高	基于软件体系的结构可大大节省开发费用

2.1.2 虚拟仪器的系统结构

虚拟仪器系统包括硬件与软件两个基本要素^{[17][18]}，硬件的主要功能是获取真实世界中的被测信号，软件的功能定义了虚拟仪器的功能。

1、虚拟仪器的硬件结构

虚拟仪器硬件通常包括通用计算机和硬件接口模块。通用计算机可以是台式计算机、笔记本电脑或工作站等。硬件接口模块可以包括插入式数据采集卡（PC-DAQ）、IEEE488 接口（GPIB）卡、串口、VXI 控制器、PXI 控制器以及其他接口卡。虚拟仪器系统的硬件构成如图 2.2 所示。

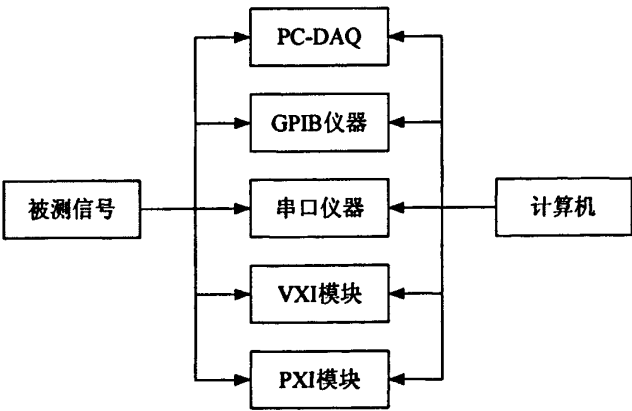


图 2.2 虚拟仪器系统的硬件构成

目前较为常用的虚拟仪器系统是数据采集卡系统、GPIB 仪器控制系统、VXI 仪器系统以及这三者之间的任意组合。其中数据采集卡系统指基于计算机标准总线（如 ISA、PCI、PXI 等）的内置功能插卡。它可更加充分利用计算机的资源，大大增加测试系统的灵活性和扩展性。

2、虚拟仪器的软件结构

在确定的硬件基础条件下, 构造和使用 VI 的关键就是应用不同的软件实现不同的功能。美国国家仪器公司 (NI) 提出的“软件即仪器”(The Software is the Instrument) 形象地描述了软件在 VI 中的重要作用。在虚拟仪器系统中, 硬件是解决信号输入/输出问题的方法和软件赖以生存、运行的物理环境, 软件才是整个仪器的核心^[18]。利用计算机的软件和硬件资源, 使本来需要硬件或电路实现的技术软件化和虚拟化, 最大限度的降低系统成本, 增强系统的功能与灵活性。如图 2.3 所示虚拟仪器的软件结构系统框架包括三个部分: 虚拟仪器软件体系结构 (Virtual Instrumentation Software Architecture, 即 VISA) 库、仪器驱动程序和应用软件。

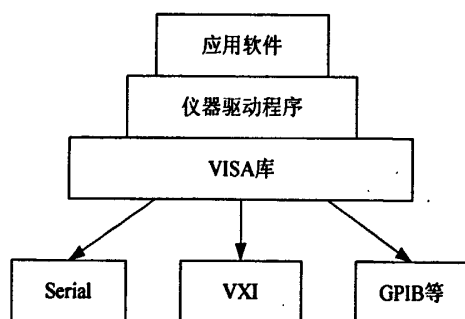


图 2.3 虚拟仪器的软件结构

其中 VISA 库实质就是标准 I/O 函数库及其相关规范的总称, 是计算机与仪器之间的软件层连接, 为仪器驱动程序开发者提供了一个可调用的操作函数库或集合, 以实现对仪器的远程测控; 仪器驱动程序是完成对某一特定仪器的控制与通信的软件程序集合, 每个仪器模块都有自己的仪器驱动程序, 仪器厂商将其以源码的形式提供给用户, 用户在应用程序中调用仪器驱动程序; 应用软件建立在仪器驱动程序之上, 直接面对操作用户, 通过丰富的数据分析与处理功能, 提供直观、友好的操作界面、来完成自动测试任务。应用软件包含两个方面的程序, 一个是实现虚拟面板功能的前面板程序; 另一个是定义测试功能的流程图程序。

2.2 虚拟仪器开发平台与 LabVIEW 开发环境

本文的系统软件开发是基于 LabVIEW 集成开发环境完成的, 采用图形化编程语言——G 语言, G 语言编写的程序称为虚拟仪器, 因为它的界面及功能与实际的仪器十分相像, 为过程监控和测试软件的编写提供了强大的支持。

2.2.1 虚拟仪器的开发平台

虚拟仪器的软件开发平台目前主要有两类^[19]：一是用通用编程软件进行编写，主要有 C、Visual C++、Visual Basic、Delphi 等；二是用专业图形化的编程软件进行开发，如 NI 公司的 LabVIEW 和 HP 公司的 VEE、HEM 公司的 Snap-Master、Iotech 公司的 DasyLab 等。

从实现虚拟仪器功能的角度出发，虚拟仪器开发环境需具备如下特点：编程简单、易于理解和修改；具有强大的人机交互界面设计能力，易于实现各种复杂的仪器面板；具有数据可视化分析能力，提供丰富的仪器和总线接口硬件驱动程序。虽然通用的开发语言具有适用面广、开发灵活的特点，但是这类开发方式对测试人员的要求很高，需要将各种数据处理方法用计算机语言实现，要用于通信的各种连接总线非常熟悉。用这种平台语言开发测控软件难度大、周期长、可扩展性差。而专业的图形化编程组态软件都通过建立和链接图标来构成虚拟仪器工作程序并定义其功能，具有编程效率高、通用性强、交叉平台互换性好的特点，适用于大批量多品种仪器的生产。

因此，基于图形化的编程软件更能满足虚拟仪器开发的要求。在众多的图形化编程软件中，由 NI 开发的虚拟仪器软件 LabVIEW，近年来作为测试工程领域强有力的工具得到了业界的普遍认可，在测试系统分析、设计和研究方面得到广泛应用。LabVIEW 编程语言在测控领域有着明显的优势，它最初就是为测试测量而设计的，同时测试测量也是当前 LabVIEW 被应用的最广的领域^[20]。至今，大多数主流的测试仪器、数据采集设备都拥有专门的 LabVIEW 驱动程序，使用 LabVIEW 可以非常便捷的控制这些硬件设备，同时 LabVIEW 开发了各种适用于测控领域的 LabVIEW 工具包和模块，用户有时甚至于只需简单地调用几个工具包中的函数，就可以组成一个完整的测试测量应用程序，为用户开发测控程序带来方便快捷，大大缩短项目开发时间。根据统计，使用 LabVIEW 开发虚拟仪器比使用基于文本的语言开发效率提高大约 10 倍。通过以上分析，本系统的设计采用 LabVIEW 开发环境。

2.2.2 LabVIEW 开发环境的特点

用 LabVIEW 编程思路与使用其他编程语言是一致的，但是 LabVIEW 的图形化编程语言的特性，使它与文本编程语言还是有所区别。其主要特点归纳如下^[21]：

- 1) 采用“所见即所得”的可视化技术建立人机界面。

- 2) 提供如单步运行、高亮执行、设置断点等调试手段，使程序的调试和开发更为便捷。

3) 采用数据流编程模式, 只要一个节点所有输入都已经准备好了就会执行, 而与其他节点是否执行完毕无关, 会自动把多个需要同时运行的节点放到不同的线程中去运行, 它是自动多线程运行的。

4) 内置 PCI、DAQ、GPIB、PXI、VXI、RS-232 在内的各种仪器通信总线标准的所有功能函数, 支持数据采集卡和 GPIB、串口设备、VXI 仪器、工业现场总线以及用户特殊的硬件板卡。

5) 具有强大的外部接口能力, 可以实现 LabVIEW 对外部应用软件(如 Word、Excel、记事本等)的调用, 可以通过 DLL 调用其他编程语言(如 C 语言、Windows API、Matlab)的编写的模块。

6) 强大的 Internet 功能, 内置了便于应用 TCP/IP、DDE、Active X 等软件标准的库函数。支持常用的网络协议, 方便远程测控的开发。

7) 可直接移植到其他操作系统平台上。

结合 LabVIEW 开发环境的这些特点, 总结本系统采用 LabVIEW 开发平台原因有以下几个方面: 数据采集与传输是测控系统的主要任务, 而 LabVIEW 可以通过加载 DLL 文件直接调用各类采集及通讯卡函数, 为数据采集与传输工作提供支持; 针对本系统涉及到的测控系统网络化的研究, LabVIEW 能提供支持以太网通信的工具包; 测控系统对数据处理和传输的实时性要求较高, 而 LabVIEW 是自动多线程编程语言这一特点能提高程序的执行速度, 节省程序的运行时间, 能很好的满足系统实时性的要求。

2.2.3 LabVIEW 调用外部程序

在 LabVIEW 众多的优势当中, 它在调用外部程序方面的特长对高效开发虚拟仪器软件显得尤为重要, 通过直接调用其他语言或者操作系统已经实现的功能, 能进一步缩短虚拟仪器软件开发周期^[22]。

在 LabVIEW 中, 可以通过 CIN(Call Interface Node)节点和 CLF(Call Library Function Node)节点调用 C 语言编写的函数, 由于 CIN 节点调用的程序模块不通用, 不能调用动态链接库中的函数, 且限制颇多, 所以在拥有相关的动态链接库(DLL)时一般采用 CLF 节点调用外部程序模块。

DLL 是一个包含许多函数的可执行模块, 它只提供一群函数供 Windows 应用程序或其他的动态链接函数库调用。DLL 的加载方式分为“动态”与“静态”两种。常用的方式是静态加载, 即只有在应用程序启动时才与相应的 DLL 链接; 动态加载方式是指应用程序启动时不载入动态链接库, 只有在使用到动态链接库中的某个函数时, 才把动态链接库载入内存。DLL 最大的优势在于代码的共享, 只要某一功能以 DLL 的形式提供出来, 其他应用程序就可以直接使用这一功能, 而

不必再实现一份相同的代码。

在需要调用的程序模块的动态链接库不具备的情况下，首先要在 VC 环境下编写动态链接库格式(*.DLL)的驱动程序。在 VC 环境下新建一个 Win32 Dynamic Link Library 工程文件，添加文件名，并选择建立一个 A simple DLL Project，然后在 DLL 项目中添加输出函数，将需要的函数导出。动态链接库文件生成后，就可被 LabVIEW 调用了。将 CLF 节点放置在程序框图上，并双击图标弹出其配置对话框如图 2.4 所示。

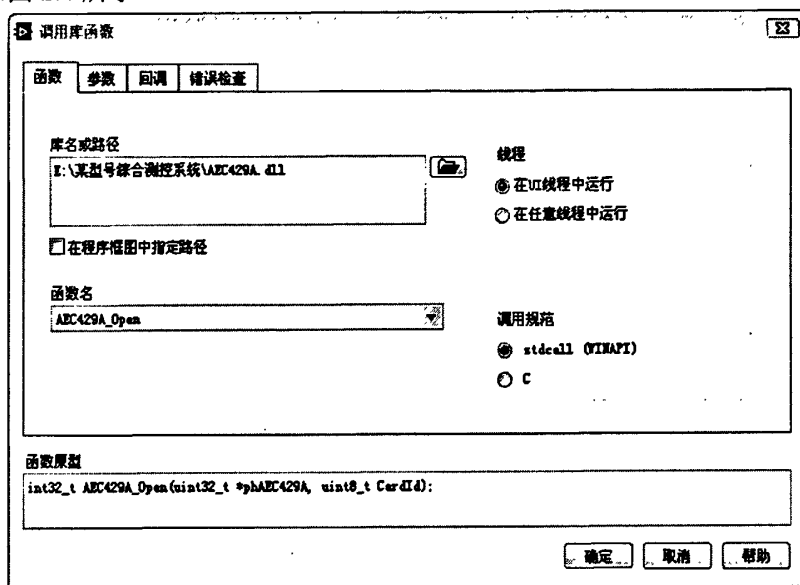


图 2.4 CLF 节点的配置对话框

若“在程序框图中指定路径”没有被选中，则 DLL 是被 LabVIEW 静态加载到程序中的；反之，DLL 是被 LabVIEW 动态加载到程序中的，此时“库名或路径”中指定的路径是无效的。须在程序框图中输入 DLL 路径。

在本文的系统开发中需要使用 DLL 的情况有两种：一是以 DLL 方式提供的第三方驱动，包括多功能数据采集卡和 ARINC429 卡；二是以 C 语言编译成功的数据压缩算法，经过 VC++环境下开发的的 DLL 在 LabVIEW 中的调用。

2.3 本章小结

本章对基于传统语言的开发环境和图形化编程环境进行了对比，结合测控系统的特点，选择了 LabVIEW 作为本文的软件开发环境。并简单介绍了 LabVIEW 开发环境的特点及其外部程序调用的相关内容，为后续系统的实现奠定了基础。

第三章 数据处理与压缩技术的研究

在数据采集系统中总是不可避免地存在各种误差,为了提高测试精度,必须尽可能的减少或消除误差。而且随着对实时变化的数据不断的采集,使所需处理的数据量越来越大,对传输带宽、存储容积和处理速度造成与日俱增的巨大压力。在这种情况下,最好的办法就是通过数据压缩设法减少数据传输所需的带宽、存储所需的容积或处理所需的时间^[22]。本章对误差处理的方法及数据压缩算法进行研究,并在理论研究的基础上作出改进,使其更适用于本文所研究的测控系统。

3.1 系统误差的分析与处理

按照误差出现的规律,测试系统的误差主要由粗大误差、随机误差和系统误差等组成^[23]。其中粗大误差是指测量结果显著偏离实际值的情况,可采用 3σ 准则等方法来剔除坏点^[24]。随机误差是由多种偶然性的误差因素造成的,当观测的数据次数足够多时,随机误差多次出现时具有相互抵消的统计规律^[25]。系统误差的特征是在同一条件下,多次测量同一量值时,误差的绝对值和符号保持不变,或者在条件改变时,误差按一定规律变化。下面将讨论系统误差特性和处理方法。

3.1.1 系统误差分析

实际上,粗大误差可能由于读错、记错或者仪器出错等多方面的原因,而使记录下来的试验数据存在异常值,采用自动测试系统并保证系统正确的连接和设置,就可避免粗大误差的产生。而系统误差存在于测量数据之中不易被发现,而且系统误差不具有抵偿性,它是固定的或服从一定函数规律的误差,多次重复测量也不能减少它对测量结果的影响,这种潜伏性使得系统误差比随机误差具有更大的危险性。

测控系统包括传感器、信号调理电路、数据采集卡以及计算机,每一步测试过程中都会引入不同的误差,这些误差再通过传递而形成系统总误差^[26]。如图 3.1 所示,设传感器、信号调理电路、数据采集卡以及计算机的误差传递系数分别为 a_1, a_2, a_3, a_4 ,由于这几个环节串联在一起,整个系统的误差传递系数 a 就是各误差传递系数的乘积,即

$$a = a_1 \times a_2 \times a_3 \times a_4 \quad \text{式 (3-1)}$$

系统误差传递的过程为

$$\Delta y = a(1 + e\%) \Delta x \quad \text{式 (3-2)}$$

式中, Δx 被测信号在进入测试系统时本身自带的误差; $e\%$ 是测试系统自身误差; a 为测试系统误差传递函数; Δy 为输出信号的误差。

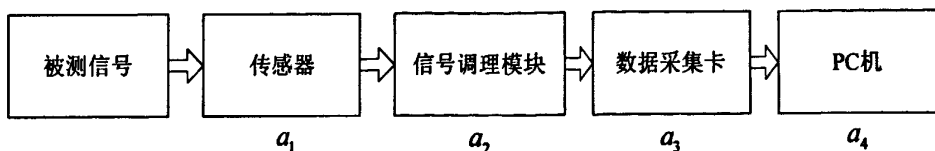


图 3.1 系统误差传递过程

由此可知, 虚拟测试仪器的误差是系统传递函数和本身的误差因素综合作用的结果, 并以误差传递系数的形式出现, 因此在分析确定误差源后, 可通过系统校准的方式来减小其影响。

3.1.2 最小二乘拟合理论基础

在测量系统中, 随机误差是不可避免的, 而系统误差和粗大误差是可以避免的, 为了提高测量精度, 实现尽可能的准确的测量, 应采取相应的误差补偿技术消除或减弱系统误差的影响。目前普遍采用多项式拟合的方法^[27]。

1、最小二乘法的基本原理^[28]

在科学实验或者统计研究中, 常常需要从一组观察数据 (x_i, y_i) ($i=1, 2, \dots, n$), 求出变量 x 与 y 之间某种近似的函数关系 $y = \varphi(x)$ 。从几何上看, 就是根据给定的 n 个点 (x_i, y_i) ($i=1, 2, \dots, n$), 找出一条近似的曲线。这类问题称为曲线拟合问题。由于不同的函数反映不同的近似程度, 因而常常规定在某种确定的函数类中, 寻求一个确定的函数 $\varphi(x)$, 使它拟合已给的数据能逼近得最佳, 从整体上考虑近似函数 $\varphi(x)$ 同所给数据点 (x_i, y_i) ($i=1, 2, \dots, n$) 误差 $r_i = \varphi(x_i) - y_i$ ($i=1, 2, \dots, n$) 的大小, 常用的方法有以下三种: 一是误差 r_i 绝对值的最大值 $\max_{0 \leq i \leq n} |r_i|$, 即误差向量 r 的 ∞ -范数; 二是误差绝对值的和 $\sum_{i=1}^n |r_i|$, 即误差向量 r 的 1-范数; 三是误差平方和 $\sum_{i=1}^n r_i^2$ 的算术平方根, 即误差向量 r 的 2-范数; 前两种方法简单, 但不便于微分运算, 第三种方法采用式

$$\sum_{i=1}^n [y_i - \varphi(x_i)]^2 = \min \quad \text{式 (3-3)}$$

作为最佳逼近的依据, 这种方法就称为数据拟合的最小二乘法。其原理如下:

设由观察得到的数据为 (x_i, y_i) ($i=1, 2, \dots, n$), 要求在函数类 $M = \text{span}\{\varphi_1, \varphi_2, \dots, \varphi_n\}$ 中寻找

$$\sum_{j=1}^m a_j \varphi_j = \varphi^*(x) \quad \text{式 (3-4)}$$

$$\text{使} \quad \sum_{i=1}^n [y_i - \varphi^*(x_i)]^2 = \min \quad \text{式 (3-5)}$$

在实际应用中, 常在最小二乘的式 (3-5) 中加入权因子 ω_i , 即

$$\sum_{i=1}^n \omega_i [y_i - \sum_{j=1}^m a_j \varphi_j(x_i)]^2 = \min \quad \text{式 (3-6)}$$

其中要求

$$\sum_{i=1}^n \omega_i = 1, \omega_i \geq 0 \quad \text{式 (3-7)}$$

2、多项式拟合

假设给定数据点 (x_i, y_i) ($i=1, 2, \dots, n$), ϕ 为所有次数不超过 m ($m \leq n$) 的多项式构

成的函数类, 现求 $\varphi_m(x) = \sum_{k=0}^m a_k x^k \in \phi$, 使得

$$I = \sum_{i=0}^n [\varphi_m(x_i) - y_i]^2 = \sum_{i=0}^n \sum_{k=0}^m [a_k x_i^k - y_i]^2 = \min \quad \text{式 (3-8)}$$

当拟合函数为多项式时, 称为多项式拟合, 满足公式 (3-8) 的 $\varphi_m(x)$ 称为最小二乘拟合多项式。特别地, 当 $n=1$ 时, 称为线性拟合或直线拟合。上述问题即为求 $I = I(a_0, a_1, \dots, a_n)$ 的极值问题。由多元函数求极值的必要条件, 得

$$\frac{\partial I}{\partial a_j} = 2 \sum_{i=0}^n \sum_{k=0}^m [a_k x_i^k - y_i] x_i^j = 0 \quad j=0, \dots, m \quad \text{式 (3-9)}$$

即

$$\sum_{k=0}^m \left(\sum_{i=0}^n [x_i^{j+k}] \right) a_k = \sum_{i=0}^n x_i^j y_i \quad j=0, \dots, m \quad \text{式 (3-10)}$$

式 (3-10) 是关于 $a_0, a_1, \dots, a_n$ 的线性方程组, 用矩阵表示为

$$\begin{bmatrix} n+1 & \sum_{i=0}^n x_i & \dots & \sum_{i=0}^n x_i^m \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \dots & \sum_{i=0}^n x_i^{m+1} \\ \dots & \dots & \dots & \dots \\ \sum_{i=0}^n x_i^m & \sum_{i=0}^n x_i^{m+1} & \dots & \sum_{i=0}^n x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i y_i \\ \dots \\ \sum_{i=0}^n x_i^m y_i \end{bmatrix} \quad \text{式 (3-11)}$$

由于方程组的系数矩阵是一个对称正定矩阵, 故存在唯一解。从式 (3-11) 中

解出 $a_k(k=0,...,m)$ ，从而可得多项式

$$\varphi_n(x)=\sum_{k=0}^ma_kx^k$$

式 (3-12)

最小二乘拟合多项式 $\varphi_n(x)$ 的平方误差，记作：

$$\|r\|_2^2=\sum_{i=0}^m[p_n(x_i)-y_i]^2$$

式 (3-13)

3.1.3 系统误差处理改进方法

在文献[29]中给出了基于最小二乘曲线拟合的误差补偿方法。其误差数据获取的系统连接方式如图 3.2 所示。

为了得到信号调理电路在各个输入电压点下的误差数据，采用 DA 板卡 PXI-6723 输出预设电压到信号调理电路，并应用两块万用表分别测量信号调理电路的输入值 U_1 和输出电压 U_2 。

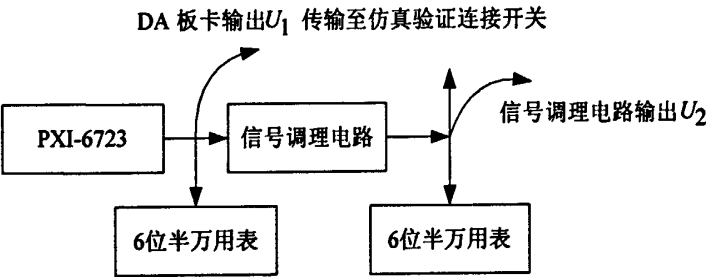


图 3.2 误差数据的获取的系统连接方式

图 3.2 中的这种误差数据获取方式中误差数据的记录仍依赖于测试人员的眼观手记，也正如文献中提到的这种测量方法容易造成粗大误差，影响了数据拟合的效果。本文将结合虚拟仪器的优点，设计基于虚拟仪器的系统误差补偿方法，如图 3.3 所示。改进的系统中包括多路选择器、信号调理电路和多功能数据采集板卡。多功能采集板卡集成 AD 模数转换功能、DA 数模转换功能和 IO 输出功能。由 IO 输出模块控制 a、b 通道的选通，此处将 a 通道称为“误差补偿通道”，b 通道称为“实测通道”。若选通“误差补偿通道”，DA 数模转换模块输出信号将通过多路选择器到达调理电路输入端，并通过调理电路输出，被多功能采集板卡的 AD 功能模块采集到；若选通“实测通道”，采集卡采集到的模拟量是被调理电路处理后的被测系统的信号。

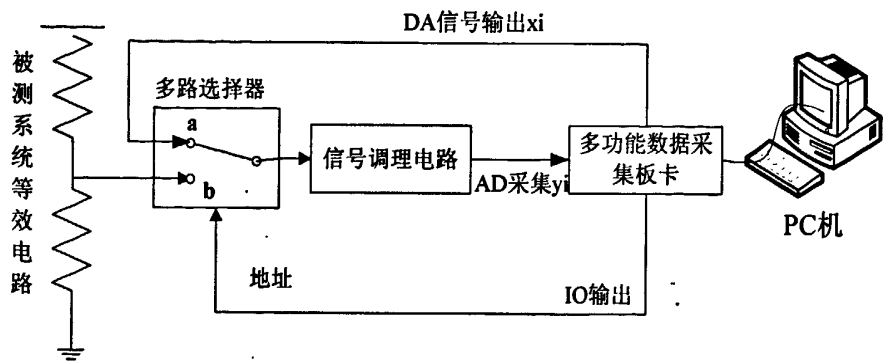


图 3.3 改进的误差数据的获取连接方式

在需要拟合误差曲线时，选通“误差补偿通道”，选择一组训练序列 X ，DA 数模转换模块将 X 以一定时间间隔通过多路选择器输出至信号调理电路的输入端，并采用采集卡上 AD 数模转换功能采集信号调理电路输出端的模拟量信号。

将采集的信号进行算术平均处理后，直接采用最小二乘多项式进行误差曲线拟合，就可知道调理电路的输入-输出特性曲线；也可根据输入-输出特性曲线，根据理论的输出值反推应该输入到调理电路的信号值，这样就可以补偿信号调理模块的误差。

当输入-输出曲线生成后，系统只需选通“实测通道”，被测系统信号值就可经信号调理电路输出端被多功能数据采集板卡获得。系统误差补偿方法流程图如图 3.4 所示。

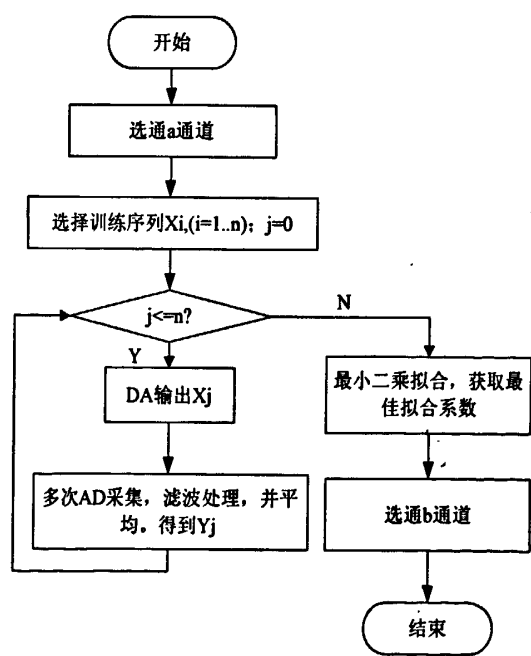


图 3.4 系统误差补偿方法流程图

在系统设计的过程中注意到：误差补偿时多项式阶数并不是越高补偿效果就

越好，高阶多项式在特定区间内的补偿效果可以是最佳的，但在该区间外可能出现极大的误差，所以阶数的选取不仅要考虑补偿的效果，还应当考虑补偿方法在整个量程范围内的可拓展性。而且当阶数选取较大时采用公式法，计算量会比较大，可能会造成采集系统的实时性不高；若将输入输出曲线的采样点存储起来，采用查表法，可能会造成精度不够。采用上述方法，选择了-10v-10v 之间的 20 个采样点，在图 3.5 中分别采用了线性拟合、二次拟合和三次方程拟合，线性方程系数为{0.99, 0.15}，二次方程拟合系数为{0.0043, 0.99, -0.15}，三次方程拟合系数为{-0.00044, 0.0043, 1, -0.015}，在三个方程中一次幂近似为 1，高次幂近似为 0，但拟合的误差随着方程幂次的增加而减少，如下图所示，线性方程的均方误差为：2.1617，拟合后的量化误差信噪比为 22.506dB，同理：二次方程量化误差信噪比 22.695dB，三次方程量化误差信噪比为 22.75dB，可见随着拟合方程的幂次增加，采集精度随着增加，但是方程幂次到一定的程度会出现饱和，如上述数据，二次方程高于线性方程 0.2dB，而三次方程仅高于二次方程 0.06dB。

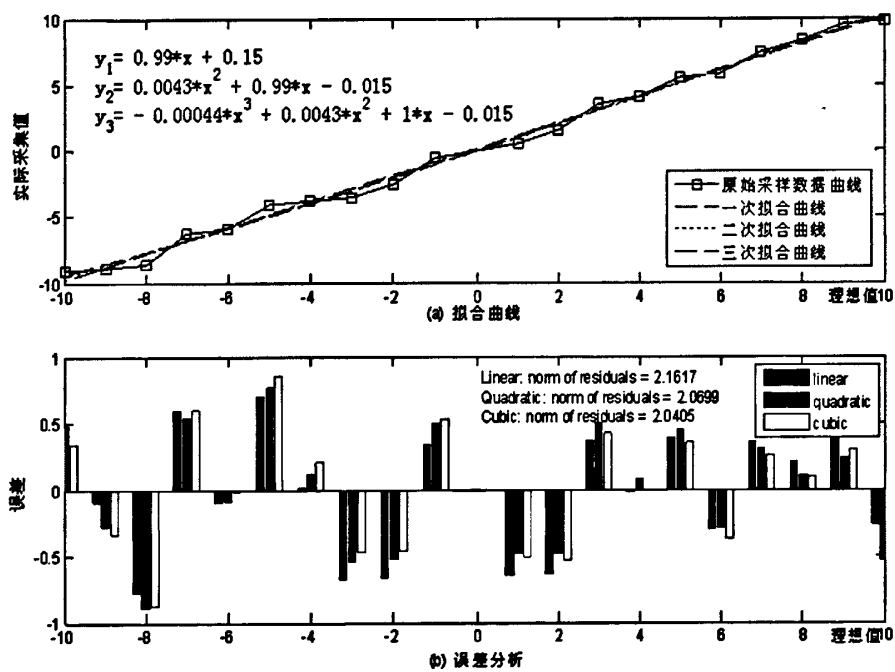


图 3.5 输入输出拟合曲线

如果设备环境改变后（如器件的重新装配等），之前生成的误差曲线不能作为补偿的参考，只需在 PC 机运行相应的误差补偿程序并进行设置，便能生成新的误差曲线。

3.2 数据压缩理论基础

3.2.1 数据压缩的思想

1959 年, 香农定义并推导出了作为最小平均信息速率的率失真函数 $R(D)$, 证明只要 $R(D) \leq C$, 就可以保证接收端的失真不大于规定的失真值。在率失真概念中, 力求找到一个最小信源速率, 使系统接收端的平均失真小于或等于某个规定的失真值。信源速率取决于信源数据的编码方法, 期望得到满足规定失真要求的最小信源速率, 这种编码方法就是数据压缩的内容^[22]。

数据压缩是把输入数据流(源流或原始数据)转变为另一种较小的数据流(输出流或压缩流)的过程。数据压缩有很多有名的方法, 它们基于不同的理念、适合不同的数据类型、产生不同的效果; 但是其原理都是相同的, 即通过去除源文件的原始数据中的冗余度来压缩数据。通过减少冗余度来压缩的思想暗示了数据压缩的通用法则, 即“对常见事件(符号或短语)赋短码, 对稀有事件赋长码”。去除冗余度的压缩原则也解释了已压缩的文件因没有或只有很少的冗余度, 所以不能进一步压缩。

数据压缩之所以流行, 有两个原因^[30]: ①需要存储的数据量越来越大, 不论有多大的存储设备, 迟早都会溢出。数据压缩能延缓这一不可避免的进程; ②需要传输的数据量越来越大, 提高了对传输带宽要求, 延长了传输数据所需的时间。数据压缩通过去除发送数据中的冗余, 减少数据传输所需带宽和所需时间。数据压缩的系统由此也可分作两大类型^[22]: 压缩-存储-回放系统、压缩-传输-解压缩系统。其系统组成原理框图分别如图 3.6(a)、(b)所示。

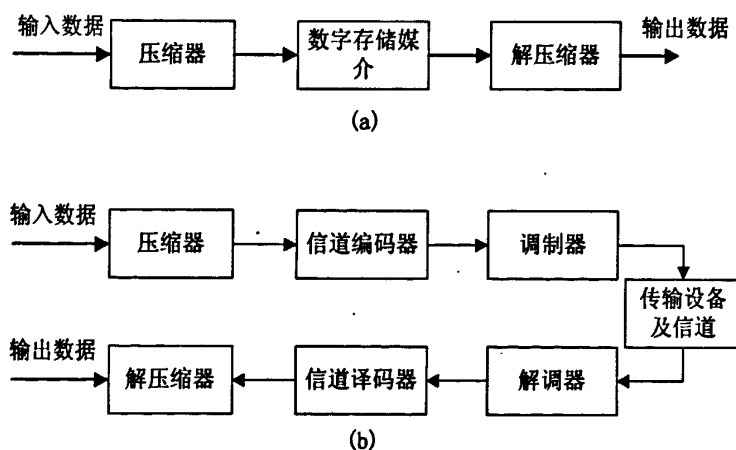


图 3.6 两种类型数据压缩系统组成框图

3.2.2 数据压缩的性能评价

压缩性能是指某些常用于表示压缩算法性能的量,常用的指标如下^[30]:

1) 压缩比 (Compression ratio) 定义为

$$\text{压缩比} = \frac{\text{输出流的大小}}{\text{输入流的大小}} \quad \text{式 (3-14)}$$

该值大于 1 说明输出流大于输入流 (负压缩)。压缩比也可称为 bpb(比特每比特),表示压缩输入流中 1 比特所需的平均比特数。

2) 压缩比的倒数称为压缩因子 (compression factor) 定义为

$$\text{压缩因子} = \frac{\text{输入流的大小}}{\text{输出流的大小}} \quad \text{式 (3-15)}$$

此值大于 1 表示压缩,小于 1 则为扩展。从数据压缩的目的来说压缩因子越大越好。但实际上,压缩因子是有上限的,对基于统计的编码方法而言,这个上限与信息熵有密切的关系。

3) 压缩增益 (compression gain) 定义为

$$\text{压缩增益} = 100 \ln \frac{\text{参考大小}}{\text{压缩后的大小}} \quad \text{式 (3-16)}$$

参考大小一般指输入流的大小,或某种标准无损压缩方法产生的输出流的大小。压缩增益的单位为对数百分比 (percent log ration),用符号%表示。

上述四种表示方式反映了某种压缩方法的一部分性能,另外文件压缩的速度和占用内存大小也是两个重要的因素。

3.2.3 数据压缩的分类

由于数据压缩的历史是近 20 年的事,目前对数据压缩还没有比较统一的分类方法^[31]。采用比较广泛的分类方法是:根据信息论的基本概念,将数据压缩分为冗余度压缩和熵压缩两大类。其中冗余度压缩技术可以从压缩后的数据中无失真地恢复出原始数据,所以也被称为无损压缩;熵压缩技术不仅对输入数据中的冗余度进行压缩,而且还对其所含有的信息进行压缩。由于输入数据中被压缩的信息将无法恢复,解压缩之后会产生失真,因此无损压缩也被称为有损压缩。图 3.7 为大致的数据压缩方法分类。

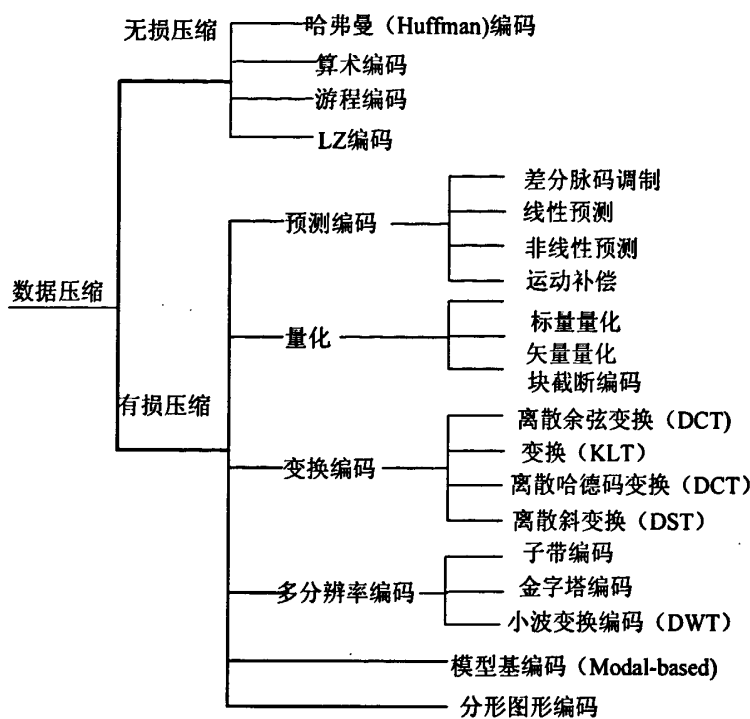


图 3.7 数据压缩方法分类

尽管有损压缩可以获得比无损压缩高得多的压缩比，但由于测控系统保存的模拟量是最原始的采集值，为了保持测量数据的特征以便后续分析及诊断能准确进行，压缩后的数据在解压后应能尽量还原。而且如果系统中离散量数据如果在数据压缩中发生失真，会导致得到的结果与真实状态完全相反。因此在测控系统中不宜采用有损压缩方案。本文主要涉及文本压缩，目前主要的无损文本压缩算法有：Huffman 编码、算术编码、LZ 编码及其变体等^[32]。针对不同的数据特点和压缩要求，如表 3.1 所示，每种压缩方法都有其优缺点和适用场合。

表 3.1 三种压缩算法的比较

压缩方法	优点	缺点	使用场合
霍夫曼编码	理论压缩率高	需要预处理，实时性差，压缩和解压时间长。	适用于较大数据量的压缩
算术编码	压缩率接近压缩的理论极限	需要浮点乘法器，计算比较复杂，实际难于实现	适合相同重复序列组成的文件。
字典压缩	压缩率高，实时性好，运算快。	压缩率受缓存容量、计算复杂度和计算速度等因素的限制。	使用面广，特别是全局或局部相关性好的数据。

在测控系统中,一次监测几百甚至上千个变量,在同一时刻监测的一批数据中相邻数据间的相关性不强,可以认为几乎没有关系,因为同一帧数据的相邻两个数据,所监控的参数可能是完全不相关的;然而时间上相邻的两组数据帧间相关性很强,这是因为测控系统数据采集速度较快,而且本文研究的系统参数一般变化幅度不大,因此不同帧间数据变化不大。结合测控数据的特征,并通过上表比较,在诸多数据压缩方法中,基于字典的无损数据压缩算法比较适合于本课题的要求。

3.3 LZW 算法的研究

3.3.1 LZW 算法编码与解码

基于查字典方法的数据压缩技术不使用统计模型,也不使用变长码而是选择许多字符串,利用字典的方法,将这些字符串编码为标识。其中 LZW 是由改进 LZ78 而提出的,它是面向通用数据、即时、简单、高效、易于实现的一种数据压缩算法。它不依赖于任何数据格式化,具有很高的综合性能并且应用广泛。其基本思想是用简单的代码来代替复杂的字符串以实现压缩,在压缩的过程中自适应建立一个字典,即反映字符串与代码之间的对照关系。

LZW 编码算法^[33]: 首先将字母表中的所有字符初始化到字典中,通常是使用 8 位字符,在输入数据流之前先占用了字典的前 256 项,即 0~255。然后编码器逐个输入字符并累积成一个字符串 I。每输入一个字符串就被串接在 I 后面,然后在字典中查找 I,只要在字典中找到 I,该过程就继续进行。直到在某一点,添加下一个字符 x 导致搜索失败,字符串 I 在字典中,而 Ix (字符 x 串接在 I 后) 却不在字典中,这时编码器就要输出指向字符串 I 的字典指针,并在下一个可用字典词条中存储字符串 Ix,把字符串 I 预置为 x。如此继续,串表将迅速增大,待压缩字符串在表中出现的概率也将增大。只要用于表示该匹配位置的编码码字数小于输入数据流中对应的匹配串的位数,就可以实现压缩的目的。

下面以字符串流 3, 34, 56, 3, 34, 365, 82, 27, 34, 56, 239... 的编码来举例说明,表 3.2 显示该文本字符串的 LZW 算法编码的流程。

表 3.2 LZW 编码例子

前缀 I	后缀 x	串组 Ix	是否在字典中?	输出	新词条 (指针)
	3	(, 3)	是		
3	34	(3,34)	否	3	3,23 (256)
34	56	(34,56)	否	34	34,56 (257)
56	3	(56,3)	否	56	56,3 (258)
3	34	(3, 34)	是		
3,34	365	(3,34,365)	否	256	3,34,365 (259)
365	82	(365,82)	否	365	365,82 (260)
82	27	(82,27)	否	82	82,27 (261)
27	34	(27,34)	否	27	27,34 (262)
34	56	(34,56)	是		
34., 56	239	(34,56,239)	否	257	34,56,239 (263)
239	...	...	...	...	...

由此可见，输入码流经过压缩后输出的码流为 3,34,56,256,365,82,27,257...，而压缩所需字典是在压缩过程中自适应生成的。表 3.3 是例中压缩过程中形成的一个 LZW 字典。

表 3.3 一个 LZW 字典

字典项	0...255	256	257	258	259	260	261
词条	0...255	3,23	34,56	56,3	3,34,365	365,82	82,27
字典项	262	263	...				
词条	27,34	34,56,239	...				

从上述分析中，可以看到 LZW 的一个规律，即字典中的字符串每次仅增加一个字符。因此，在字典中要得到长字符串需要较长的时间，这样才可能压缩得较好。可以说，LZW 是缓慢自适应的压缩编码方式。

3.3.2 LZW 算法的优缺点

1) LZW 算法的优点：LZW 与 LZ78 相比优势在于编码开始时，字典中已经含有各个单字符及其编码的串表，因此在 LZW 编码中下一个输入字符总能在字典中查到。这也是 LZW 标识仅包含匹配字符串的索引值，希望能通过降低码字的长度来提高数据压缩比。解压时这个字典无需与压缩代码同时传送，而是在解压过

程中逐步建立与压缩时完全相同的字典,从而完整、准确地恢复被压缩内容。因此,LZW 算法是一种解码速度与压缩性能较好的压缩算法。

2) LZW 算法的不足:虽然 LZW 是一种自适应的数据压缩方法。但它对输入数据的适应性较慢,因为字典中的符号串每次只增加一个字符,它产生的字典短语编码较多。

从理论角度分析 LZW 算法,当信息源长度趋于无穷时,数据间的冗余度可以趋向于零,因此,我们认为这种压缩算法是渐进最优的。但是由于该算法总是给出不同的代码字分配固定长度的整数,并且不考虑信息源的分布概率,因此存在着不足,留下了改进该算法的空间^{[34][37]}。

3.3.3 LZW 算法在测试系统中改进

1、数据预处理

本系统的采集的数据从数据量上看,没有视频图像大,但是随着时间的延续,测控系统累积的数据量非常的大。在系统中模拟量是缓慢变化的,开关量在某个工作状态下一直持续不变,直到工作状态产生变化,才有部分开关量改变。利用差分去除数据间相关性。设 k 时刻待传送的数据为

$$I(k) = (A(k), D(k), C(k)) \quad \text{式 (3-17)}$$

其中 $A(k)$ 为模拟量多路采集; $D(k)$ 开关量输入; $C(k)$ 为控制量输出。而 $k-1$ 时刻传送的数据为

$$I(k-1) = (A(k-1), D(k-1), C(k-1)) \quad \text{式 (3-18)}$$

则在 k 时刻传送的数据为:

$$\begin{aligned} \delta(k) &= I(k) - I(k-1) \\ &= (A(k) - A(k-1), D(k) - D(k-1), C(k) - C(k-1)) \\ &= (\delta_A(k), \delta_D(k), \delta_C(k)) \end{aligned} \quad \text{式 (3-19)}$$

只要工作状态不变的情况下, $\delta_D(k)$ 、 $\delta_C(k)$ 大多数情况下都为 0。经过这样的数据预处理后,工作日志中的冗余量便增加了。采取同样的算法都能达到比较好的压缩效果。

2、LZW 算法改进

本文针对 LZW 算法的改进主要从字典初始化以及编码长度两个方面着手。并采用字典填满后清除的字典管理方式。

编码时,首先要对字典进行初始化操作,通用的 LZW 编码算法在初始化字典时,字典的前 256 项所对应的字符串就是相应的单字符串,即 ASCII 码表中的前 256 项,编码从第 257 项开始,便于 LZW 算法在图像、文本以及 exe 等各

类文件压缩中广泛应用。然而在本课题所研究的系统中，模拟量与离散量都以十六进制的方式存储，因此需要压缩的文本文件中可能出现的单字符仅有 0~F 和空格。其中模拟量是一个 float 型数据，按 4 个字节处理；离散量仅占字节中的 1 位。所以在进行字典初始化时，不需要对所有 256 项单字符进行初始化，仅需将可能出现在文本中的这 17 项单字符初始化就能满足本系统的要求，并能减少在字典中查找匹配字符串的时间。

在 LZW 算法中，编码位与字典容量是相对应的，而实际应用中的字典容量受一定限制，若无限地增大字典容量，虽然可能获得更好的压缩率，但进行字符串匹配时查找的时间也会增加，而且在传统的 LZW 算法中，直接存储最大编码位，最初加进字典中的短语和后来被加进字典中的短语使用了相同长度的编码，这样做导致非编码数据也需要存储同样大的位数，显然要浪费一些完全没有用处的几个高位。针对原算法给不同的代码字分配固定长度整数的情况，改进的算法给不同的代码字分配变长整数。以 12bits 的 LZW 压缩算法为例，该算法编码表可以容纳 4096 个码字。整个编码表分成 5 个部分，每一部分输出的代码字长度不同，串表图例如图 3.8 所示。在算法中 256、512、1024、2048 为变长临界点，当字典中的表项到达变长临界点时，编码器输出变长标识（在初始化字典时，将 17 定为变长标识），且编码长度加 1，当字典表项达到 4096 时，表示字典已满，输出清除字典标识。

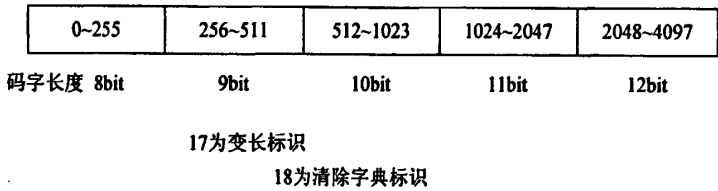


图 3.8 串表图例

考虑到字典编码长度是按位逐渐增大的，因此与定长编码相比，变长编码在存储代码时需要增加一个位处理过程。下面以 9 位编码长度为例，位处理过程如图 3.9 所示。

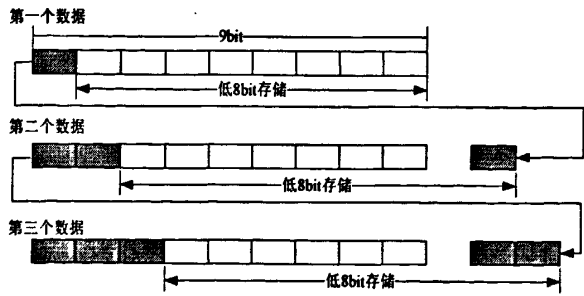


图 3.9 变长编码位处理过程

本系统中的编码长度从 8 位开始的，因此数据存储也从 8 位开始。当字典中的表项达到 256 项时，编码长度加 1，变为 9 位。将第一个数据剩下的高位作为第二个数据的低位，第二个数据剩下的高位作为第三个数据的低位，其他数据依次类推。

LZW 算法的字典是自适应生成的，如果待编码的文件过长，就会引起字典满。目前主要有三种管理字典的方式：①当字典填满时，不再更新字典中的表项，字典转换为静态字典，其后压缩只在表中寻找匹配的字符串。这种方法实现简单，当传送的文件性质、类型变化不大时，压缩效率较高，缺点是通用性不强。②采用 LRU 淘汰原则删除字典中部分表项，借鉴操作系统页面管理的方法。及对每一项进行计数，当清除数据时，根据使用次数的多少来决定是保留还是清除^[35]。然而 LRU 淘汰原则没有真实的反映出串表填满后输入字符概率分布的情况，如果对字典进行部分清除，对于压缩程序来说频繁清除串表影响了压缩速度。③字典填满后，清除字典中添加的表项，将字典恢复至初始化状态。这相当于以字典满为条件，将原文件分为几个独立的压缩数据块，这种方法对长文件的适应性较强。

考虑到第三种方式实现简单且对长文件有着较强的适应性，本文采用第三种方式管理字典。改进的 LZW 算法具体实现流程如图 3.10 所示

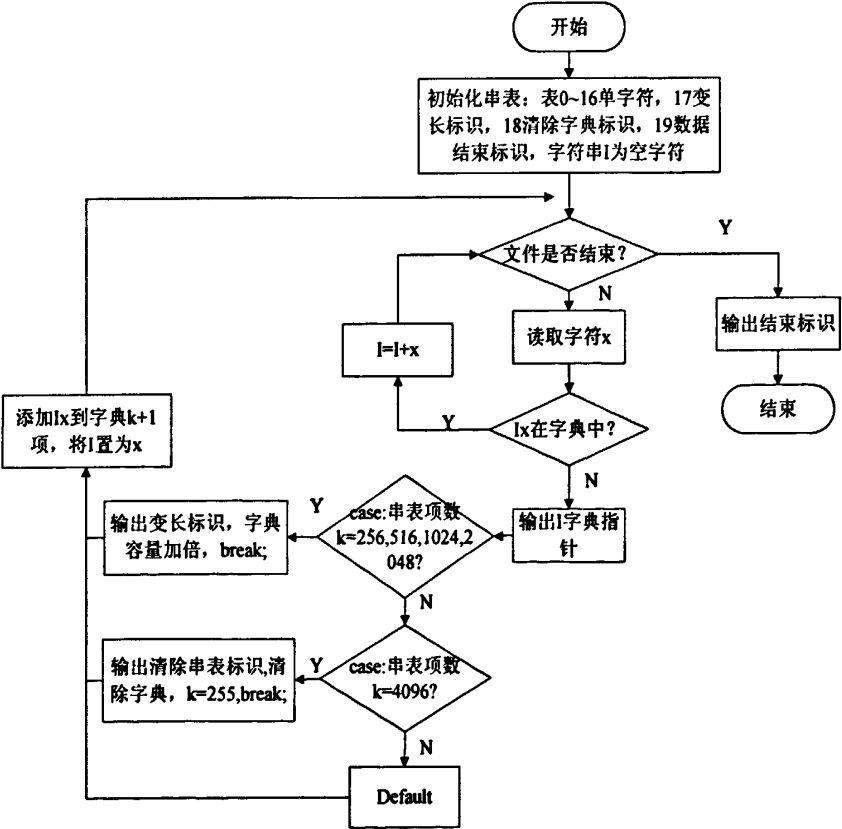


图 3.10 改进的 LZW 算法流程图

3、测试结果及分析

将实际的测控数据中 2 个小时以及 6 个小时的工作日志数据作为原文件，通过采用原 LZW 压缩算法对未进行预处理的数据、和预处理后的数据进行处理；以及采用改进的算法对与预处理后的数据处理后，列出试验数据并进行压缩率的比较。如表 3.4 所示。可以看到对数据的进行预处理后再压缩，能达到更好的压缩效果，改进的 LZW 算法压缩率在原有压缩算法上有所降低，即压缩效果更好，更能适用于测试系统的文件远传中。

表 3.4 数据压缩测试结果

	算法	原文件（文本文件）	压缩完毕文件	压缩比
未预处理	原 LZW 算法	3.16M	1.98M	62.7%
		10.9M	6.6M	60.6%
	改进算法	3.16M	1.83M	57.9%
		10.9M	6M	55%
预处理	原 LZW 算法	3.16M	1.96M	62%
		10.9M	6.52M	59.8%
	改进算法	3.16M	1.8M	57%
		10.9M	5.89M	54%

3.4 本章小结

针对原最小二乘曲线拟合误差补偿方法容易产生粗大误差的缺点，本章将基于最小二乘曲线拟合的误差补偿方法与虚拟仪器相结合，使输出输出曲线的获取不再依赖于人的眼观手记，只需在人机界面上进行设置，便能实现虚拟仪器化的误差补偿方法。在数据压缩理论的基础上，重点分析了无损压缩 LZW 算法原理，及其编码解码流程，结合测试数据的特点，通过对数据进行预处理和对 LZW 算法的改进，经数据实测改进后的算法在原 LZW 算法的基础上压缩比降低了 5%，为测试数据高效传输作出了铺垫。

第四章 基于虚拟仪器的传输系统的研究

随着现代信息技术的高速发展给仪器之间的远距离通信提供了十分有利的条件, 远程监测和控制已经成为测控系统的一项基本功能^[1]。鉴于现代测控系统中将测试设备配置到空间中分布在不同位置的被测对象附近, 连接这些测试设备的通信方式及相关的通信协议就显得至关重要。本章将从虚拟仪器网络化的几种模式开始讨论, 通过 ARINC429 总线、网络传输等实用技术的阐述, 利用 PC 机实现基于虚拟仪器的传输系统的设计。

4.1 虚拟仪器网络化的研究与设计

4.1.1 组建网络化虚拟仪器系统的两种模式

网络化虚拟仪器是指将计算机、外围设备、被测试节点以及数据等软件资源纳入网络进行资源共享, 共同完成测试任务, 实现对现场基于 Internet 的异地或远程测试、控制、数据获取、故障诊断和现场监视功能。

网络化虚拟仪器结构比较灵活, 可将不同的功能模块(如信号获取与处理模块、信号显示模块、数据存取模块等)依托网络技术分布到网络的不同位置, 组成各种网络化虚拟仪器的解决方案。其重要的一个基本特征就是实现信息资源的共享, 使网络化虚拟仪器的功能远远大于系统中单独个体的功能综合。传统的测试(测量)仪器或系统一旦联网, 如虚拟仪器中的计算机接入 Internet, 与某种合适的电子化信息载体结合在一起, 便组成了网络化虚拟仪器。

按客户端程序的形式来分, 网络化虚拟仪器通常有两种模式^[3]: 一种是 Client/Server (简称为 C/S) 模式; 另一种是 Browser/Server (简称为 B/S) 模式。

1、C/S 模式

图 4.1 为采用 C/S 模式构成网络化测控系统的结构图, 它是 20 世纪 80 年代由美国 SYBASE 公司为首的几家计算机公司提出并实现的, 主要由现场测控设备、客户端和远程服务器等组成。由多个客户端来采集数据, 客户端通过通信协议把测试数据写入远程服务器数据库。这一计算机体系结构及网络模式在设备远程状态检测与故障诊断系统的设计方面被认为是较为理想的模式。

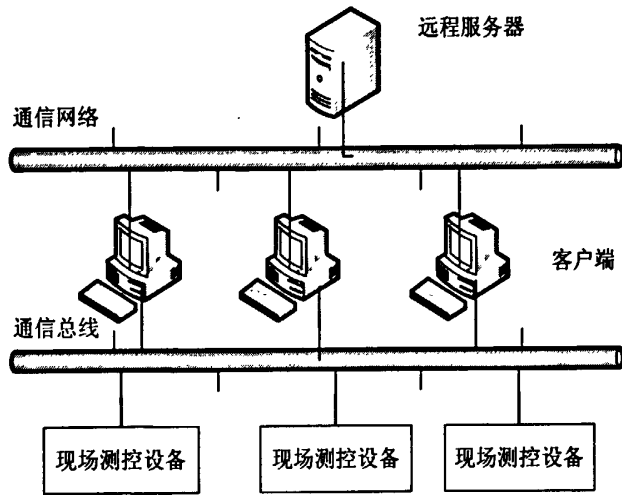


图 4.1 C/S 模式网络化测控系统构成

2、B/S 模式

B/S 模式是一种以 Web 技术为基础的新型网络模式。B/S 模式一般由客户端、服务器以及与服务相连的远程测控设备相连组成。采用 B/S 模式构成的 Web 虚拟仪器将仪器的前面板发布到 Web 页面上，以浏览器/服务器的方式实现网络数据、命令传输，以虚拟仪器实现信号采集、显示与分析，最终通过 Web 服务器与数据采集控制服务器的交互来实现相关的测试与控制需求。B/S 模式网络化虚拟仪器系统构成如图 4.2 所示。

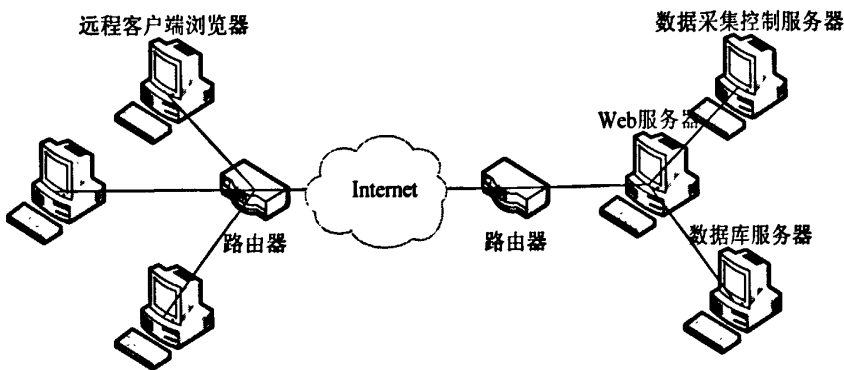


图 4.2 基于 B/S 模式的网络化测控结构

3、C/S 模式与 B/S 模式的比较

C/S 模式是“胖客户/瘦服务器”模式，大量的应用程序在客户端，在不同的客户端需要安装不同的应用软件，而服务器只起到了一个管理作用。许多 LAN 网和大多 DCS（集散控制系统）都采用这种模式。B/S 模式是一种“瘦客户/胖服务器”模式，不再要求不同的客户端安装不同的客户程序，系统的主要工作在于

服务器端程序的开发，客户端只需安装一个浏览器，就可通过直接访问 Web 服务器实现远程监视，并可通过浏览器上界面修改控制参数或指令来实现设备的远程控制。这两种模式的特点^{[3][36]}如表 4.1 所示。

表 4.1 C/S 模式与 B/S 模式特点比较

	C/S 模式	B/S 模式
优点	安全性高、数据传输量大、网络通信负荷小、效率高、数据可靠完整。	系统更新及维护的工作量较小。
缺点	应用程序的更新或维护工作量巨大且繁琐，扩展性和灵活性较差，成本相对较高。	浏览器中加载界面速度较慢，且本地必须启动 Web 发布程序；系统传输大量数据时响应较慢；用户不能与服务器建立长时间的链接。
网络传输	在 LabVIEW 平台下一般采用 TCP/UDP 协议和 DataSocket 技术进行数据传输。	在 LabVIEW 一般采用 HTTP 协议进行数据传输，或通过远程面板技术实现远程监控。

考虑到本文讨论的测控系统中偶有大数据量传输的情况，为满足系统的实时性要求，本文采用 C/S 模式实现虚拟仪器系统的网络化。

4.1.2 多总线分布式网络测控系统的设计

本系统的通信接口主要包括两部分：工业控制计算机与 ARINC429 总线之间的接口以及工业控制计算机和自动管理计算机之间的网络通信接口。ARINC429 总线负责测控系统与被测设备之间、测控系统与控制管理系统之间的通信。网络通信接口一方面接收自动管理系统的指令信息，另一方面回传网络通信状态信息。

在 C/S 模式下，一般采用 TCP/IP 协议进行数据传输。组建网络时，通常将需要共享的任务放在服务器上，不需共享、专用的任务放在客户端，系统应尽可能紧凑以增强系统的安全性；此外还要考虑系统的扩展性和开放性，能够在扩展系统规模时尽量减少重复的工作。依照这些原则，下面组建基于 C/S 模式的测控传输系统。

在测控系统中，要将虚拟仪器网络化，首先要将测量与控制设备网络化。对于虚拟仪器，由于测量与控制设备多连接于计算机上，因此只要计算机具有联网能力，也就等于虚拟仪器具有联网能力。能将计算机连接到网络上的方式很多，如传统的网络接口卡（Network Interface Card, NIC），本身不具备联网能力的的计算机也可通过 GPIB-NET、RS232-NET 等转换卡连接到网络上。

在本传输系统中利用传统的网络接口卡实现虚拟仪器的网络化，这也是实现

虚拟仪器网络化最简单的方法，通过制定 IP 地址和端口，能方便地与网络上的其他任一带有 NIC 的计算机进行信息交换。基于 C/S 模式设计的传输系统结构如图 4.3 所示。

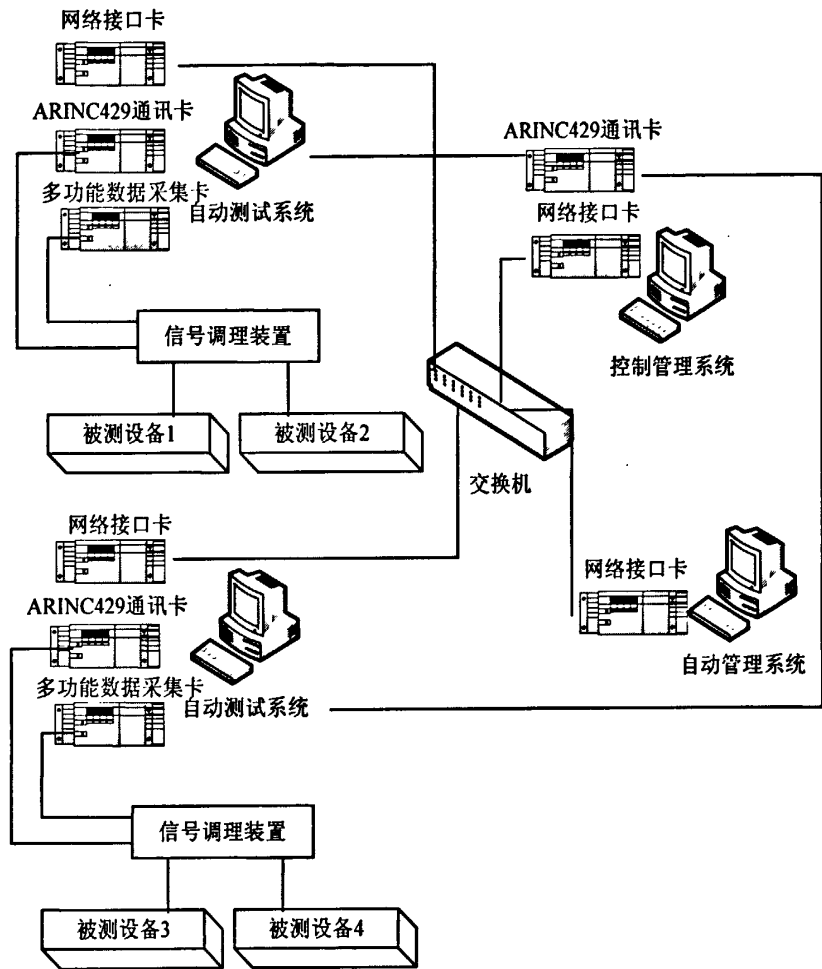


图 4.3 基于 C/S 模式组建的传输系统结构

系统中测试节点的每一个测试仪器都具有唯一的 IP 地址，可以很方便地实现测试仪器乃至多个测试系统之间的网络互连。该系统可根据测试项目后期的需要进行扩充、重构。对于庞大的测试系统而言，可提高系统的稳定性、安全性。下面对本节设计的传输系统中总线技术及关键技术分别进行研究。

4.2 ARINC429 总线通信的研究

4.2.1 ARINC429 数字信息传输规范

ARINC429总线是机载电子设备之间进行数据传输约定的一种标准,具有总线结构简单、性能稳定、抗干扰性强等优点,使测控系统能顺利地扩展到远程操作的领域。ARINC429数字信息传输规范阐述了通过一对屏蔽双绞线(数字数据总线)从一个端口向其他系统和设备以串行方式传输数字数据信息的方法。数据每32位(或25位)为一个数据字;通信采用用双极性归零制的三态码调制方式,即调制信号由“高”、“零”、“低”状态组成的三电平调制状态;数据高速传输速率为100kbps,低速传输速率为12Kb/s~14.5kbps,在同一个总线中,发送速率与接收速率必须一致。协议规定,ARINC-429为单向传输总线,信息只能从通信设备的发送口输出,经传输总线传至与它相连的需要该信息的其他设备的接收口,总线上只允许有一个发送器但最多可达20个接收器。在需要通信两个设备间双向传输时,则在每个方向各用一根独立的传输总线。一般情况下,ARINC429总线系统拓扑结构主要有星型和总线型两种,如图4.4所示,前者每个通道由独立的电缆连接,一条通信线路故障对其他通信线路没有影响,因此可靠性较后者高,但电缆多,重量重。

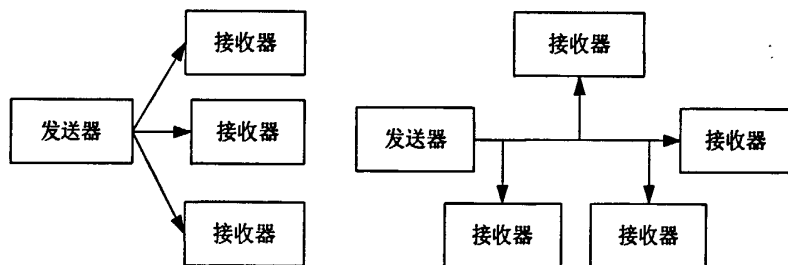


图4.4 ARINC429总线系统拓扑结构

4.2.2 ARINC429 传输格式和类型

在ARINC429传输总线上,数据传输以电压脉冲形式发送,一个电压脉冲代表一位。1个数据字(32位)被分为5段,其中包含8位标识符(LABEL),2位源/目的地址识别码(SDI),19位数据区(DATA),2位符号/状态码(SSM),1位奇偶校验位(PARITY)。码字格式^[38]如表4.2所示:

表 4.2 ARINC429 通用数据字

32	31~30	29~11	10~9	8~1
奇偶校验	符号/状态码	数据区	源/目的地址识别码	标志位

1) 标识符

用于表示信息的类型。实际应用中，用设备识别字来识别不同的源设备，设备识别字按 BCD 数据字格式编码。

2) 源 / 目的地识别码 (SDI)

在测控系统中，需要在被测产品以及各测控分系统间进行数据交换，系统采集以及接收的各种参数需要发送到控制管理系统和被测产品中进行处理，SDI 用于识别该 ARINC429 数据字的源和目的地。

3) 数据区

数据区的作用是将采集到的数据进行封装，以便于传输。

4) 符号状态码 (SSM)

用于标识数据字的特性，如方向、符号等，SSM 也可表明数据发生器硬件的状态，是无效数据还是试验数据。

5) 奇偶校验位 (PARITY)

ARINC429 数字信息传输规范奇偶校验位提供的是奇数奇偶检验。

ARINC429 数据字的发送顺序为：标志码、源 / 目的地识别码、数据区、符号状态码和奇偶校验位。其中标志位与正常发送顺序刚好相反，先发送高位再发送低位。其数据格式转换如图 4.5 所示。

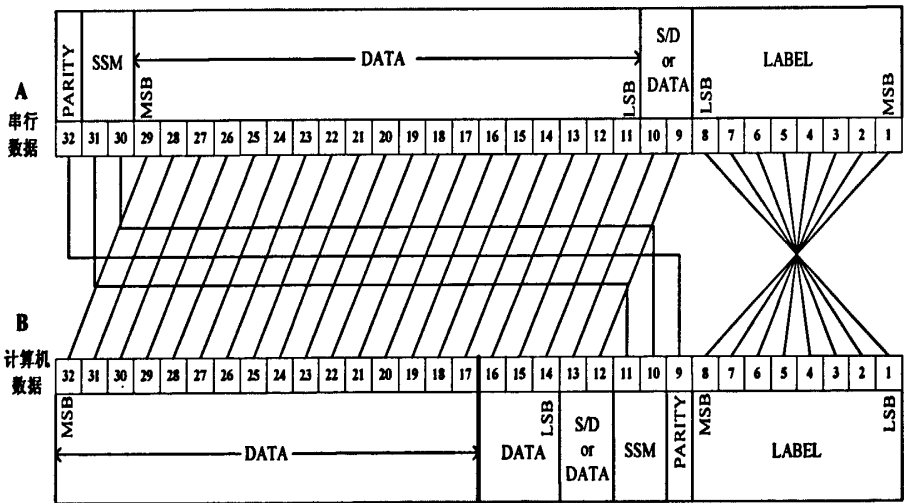


图 4.5 ARINC429 数据格式转换

4.2.3 CRC 编码原理及实现

通信的基本目的是要把接收方不知道的消息及时可靠地传送给接收方。因此，往往要求一个通信系统传输的消息必须可靠、快速^[42]。在第三章中提到的数据压缩就致力于通信传输的快速性，信道编码就是保证通信系统可靠性的有效方法。循环冗余校验 (Cyclic Redundancy Check, CRC) 是一种系统的缩短循环码，是

线性分组码的一个子类, 以其良好的代数结构、编码简单、检错能力强且误判率低, 在通信系统中得到了较为广泛的应用^{[39][40]}。

利用 CRC 进行检错的过程可简单描述为: 在发送端根据要传送的 k 位二进制码序列, 以一定的规则产生一个校验用的 r 位监督码, 即 CRC 码, 附在原始信息后面, 这样构成一个新的二进制码序列数共 $k+r$ 位, 然后再发送出去。在接收端, 根据信息码和 CRC 码之间所遵循的规则进行检验, 以确定传送中是否出错。这个规则, 在差错控制理论中称为“生成多项式”。

本文采用 CRC-16 的生成多项式如式 (4-2) 所示, 来实现 CRC 校验码。

$$G(x) = x^{16} + x^{15} + x^2 + 1 \quad \text{式 (4-1)}$$

目前常用的 CRC 校验方法是查表法和公式法。查表法: 把 6 位二进制 CRC 码 (共 256 个) 全部计算出来, 存储在 256 个单元中, 编码时根据输入字节查找到相应的 CRC 码进行处理输出即可; 公式法与查表法一样也是以字节数据为输入, 采用递推算法, 不同之处在于使用公式可以实时计算 CRC 结果, 从而省去了查找表, 使软件的编写更为简便。因此本文采用公式法实现 16 位 CRC 编解码, 具体算法如下。

先将要发送的二进制信息序列左移 16 位后, 再除以生成多项式, 最后所得到的余数即为 CRC 码, 如式(4-1)所示。其中, $P(x)$ 表示 n 位二进制信息序列, $Q(x)$ 为商 (即整数), $R(x)$ 为余数 (即 CRC 码), $G(x)$ 则为生成多项式。

$$\frac{P(x) \cdot 2^{16}}{G(x)} = Q(x) + \frac{R(x)}{G(x)} \quad \text{式 (4-2)}$$

接收方将接收到的二进制信息序列除以生成多项式, 如果余数为零, 则表示传输过程中无错误发生, 否则说明传输有误。在实际应用中, 可根据接收到的信息码计算 CRC 码, 再和接收到的 CRC 码比较, 通过比较计算出的码字与接收的码字是否相同来判断传输过程中是否无错误发生。

CRC 码采用的是模 2 加减运算法则, 即不带进位和借位的按位加减, 在 LabVIEW 中实现 CRC 校验算法的方法有多种, 本文采用公式节点的方式实现, 并将 CRC 校验封装为子 VI “CRC16.vi”, 流程图如图 4.6 所示。

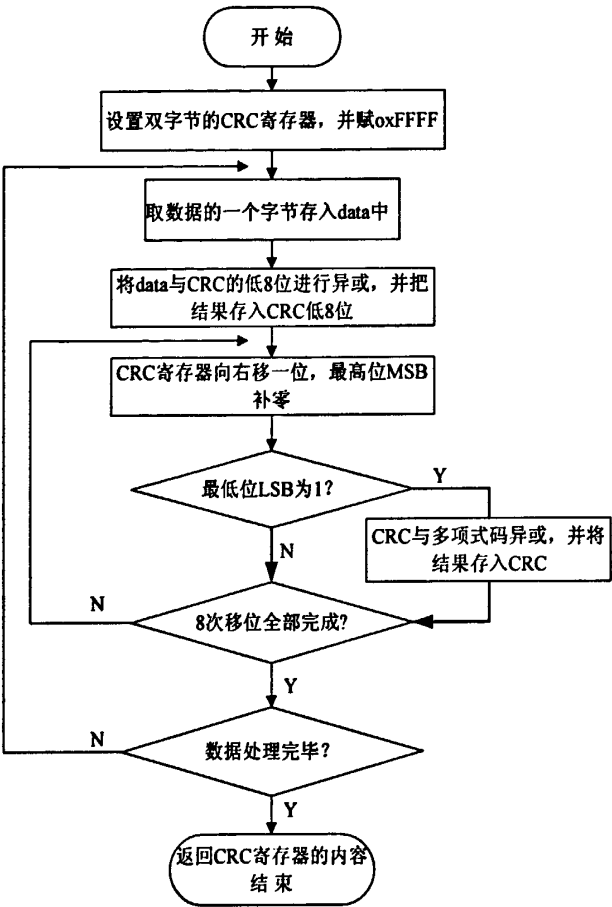


图 4.6 CRC16 校验算法流程图

4.2.4 综合检错技术及应用

前面已对 ARINC429 的数据字格式做了介绍，每个 ARINC429 字的第 32 位为奇偶校验位，此校验位通常由硬件自动生成，但该检错方法仅能检测到奇数个比特位错误的情况，若在传输过程中出现偶数个比特位错误，该校验位无法发现错误。为了解决上述问题，本文采用综合检错技术提高通信的可靠性，即引入 CRC16 校验码与硬件自动生成的奇偶校验结合，并应用在 ARINC429 数据传输中。如图 4.7 所示。

在发送数据之前需按照通信协议对仿真数据进行封装，在此以数据帧长度为 10 的情况为例，共分为两步。首先，将需要发送的信息按照通信协议的规定封装为 32 位 ARINC429 数据字，包括 Label 字段、SDI 字段、数据区 Data 和 SSM 字段，通常 DATA 区中需要发送的信息都以字节为单位存储，因此在实际应用中一般用到 2 个字节，29~27 位补零。通过对通讯板卡的设置可选择奇校验、偶校验或无校验；然后，将 10 个 ARINC429 数据字组成一帧，将前 9 个 ARINC429 数

据的 DATA 区 11~26 位数据经过 CRC 按字节编码,并填充在第 10 个 ARINC429 数据字的 DATA 区的 11~26 位,高位补 0。接收数据时,首先对每个数据字的奇偶校验位进行判断,如果奇偶校验正确,根据接收到的前 9 个数据字的 DATA 区数据计算 CRC 码,与第 10 个数据字的 DATA 区数据比对,若相等则表明数据正确,否则说明数据在传输过程中出现误码,根据系统的实际需求,选择是否请求重传。

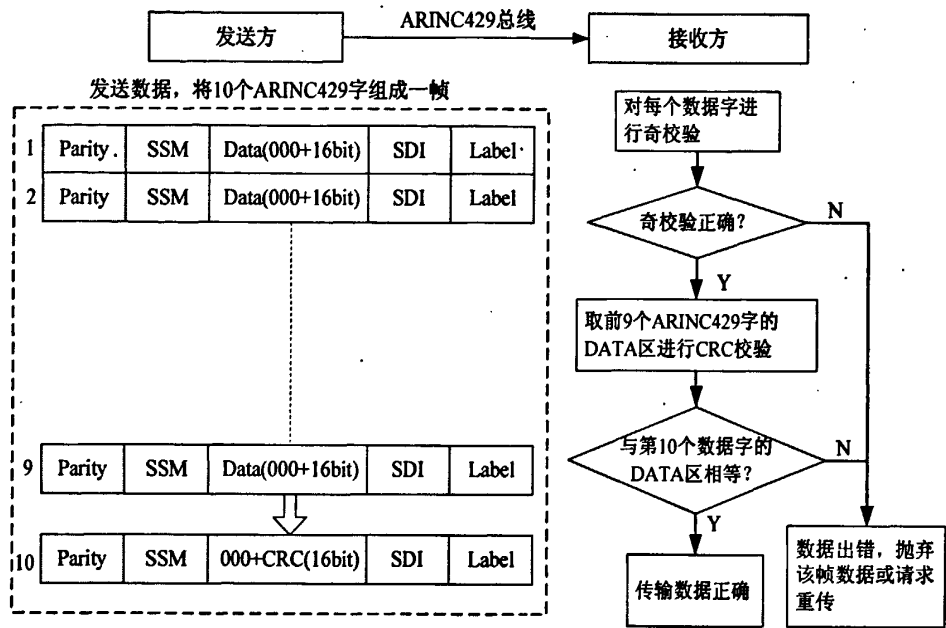


图 4.7 综合检错技术在 ARINC429 通信中的应用

由 CRC 码字的性质可知,所有码字的重量都是偶数,所有重量为奇数的错误都可以被检错出, CRC 漏检概率的上限为 $1/2^{n-k}$, 从性能和开销上考虑,均远远优于奇偶校验及算术和校验等方式。为了检验综合检错技术的性能,在 LabVIEW 开发平台上仿真数据帧的校验过程。将发送数据按图 4.7 所示的方式将 3 个 32bit 数据组成一帧,如图 4.6(a)所示,经高频载波调制后叠加高斯白噪声信号,接收端解调并滤波后的信号,以及采样判决后的信号分别如图 4.8(b)、(c)所示。

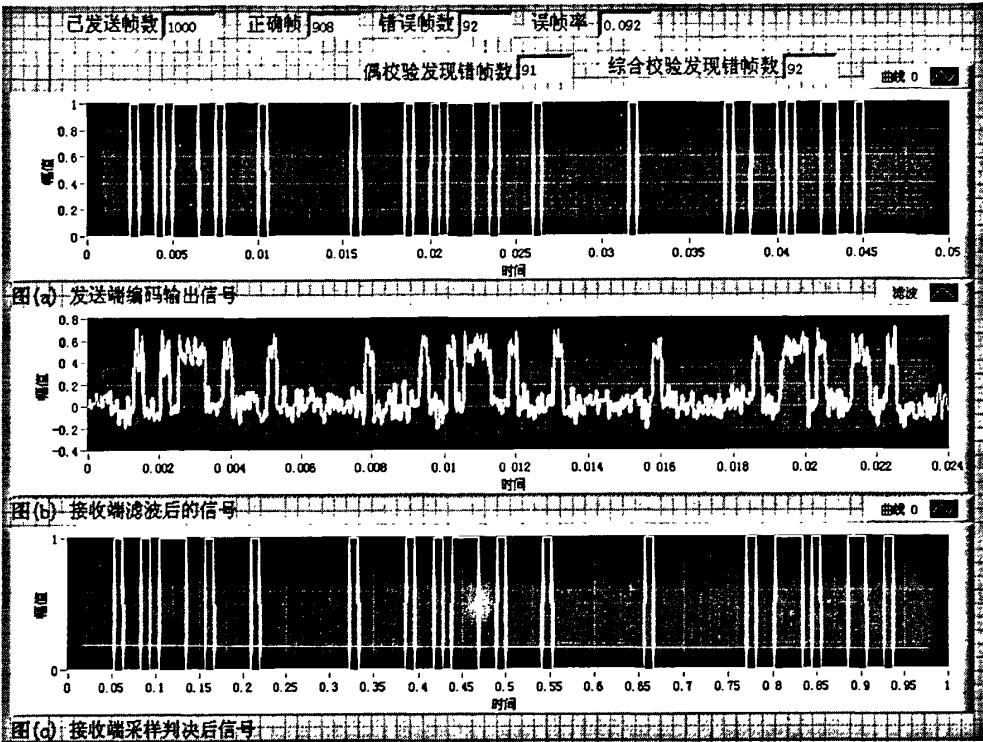


图 4.8 综合校验性能检测试验

为得到综合校验技术的统计性能，在试验中将该帧循环发送 1000 次，在接收端分别统计实际接收的正确帧数、错误帧数，以及偶校验、综合校验发现的错误帧数，并在不同的信噪比条件下进行试验，得到统计值如表 4.3 所示。从表中可看出综合校验比偶校验漏检的错误帧个数少，显然综合校验性能优于偶校验。

表 4.3 偶校验与综合校验统计值

信噪比	1000 帧中发现的错误帧数		
	实际错误帧	偶校验发现错误帧	综合校验发现错误帧
10db	92	91	92
6db	338	314	338
5db	632	572	621
3db	817	725	810

4.3 以太网应用于测控系统的研究

4.3.1 以太网应用于测控系统的优势

传统的共享式以太网采用带碰撞检测的载波侦听多路访问控制协议 (CSMA/CD) 来解决信道竞争, 根据这个协议, 网络中的各个节点具有相同的优先级, 在监听到信道空闲时发送数据, 在发送数据的同时继续监听, 如果发现冲突, 则立刻停止发送, 并等待随机长度的时间后再发送。然而由碰撞而引起的信息传输时间的随机性使其不能满足测控系统实时性的要求, 其不确定性和实时性欠佳是其应用于测控系统中最大的问题。但随着快速以太网以及交换式以太网的发展, 为解决非确定性问题带来了契机: 100Mb/s 的以太网的应用以及千兆以太网的出现, 使冲突几率大大减小; 采取以太网交换技术, 将网络冲突域进一步细化, 来提高以太网系统的实时性^[41]。另外, 以太网的非屏蔽双绞线的抗干扰能力与 4~20mA 模拟传输线路相当, 如果需要更强的抗干扰能力可以采用屏蔽双绞线或光纤通信。以太网解决了实时性、稳定性和抗干扰性的问题, 其应用于测控系统的优越性便凸显出来。

与其它网络协议相比, 以太网具有以下优点^[42]:

- 1) 以太网能便捷地访问远程系统, 能实现办公自动化网络与工业控制网络的无缝连接;
- 2) 基于 TCP/IP 的以太网是一种标准的开放式网络, 不同厂商的设备很容易互联。易于与 Internet 连接, 能够在任何城市、地方通过 Internet 对企业进行监控。
- 3) 具有低成本的特点, 能够降低系统硬件、维护人员培训的费用;
- 4) 以太网受到了广泛的技术支持。几乎所有的编程语言都支持以太网的应用开发, 硬件开发商为以太网系统的设计提供了广泛的硬件产品选择, 人们对以太网的设计及应用有较为丰富的经验;
- 5) 易于升级。对于所有以太网技术, 其帧结构和帧长度是不需作调整, 为以太网技术的升级提供了一个很好的途径。星型集线器的应用和基于 10/100Base-T 的网卡及交换技术, 使升级更为容易。

4.3.2 TCP/IP 协议及 Winsock 开发过程

对于任何类型的网络, 为保证联网的实体之间相互通信, 就必须符合一定的网络协议。TCP/IP 是 20 世纪 70 年代中期美国国防部为其 ARPANET 广域网开发的网络体系结构和协议标准, 是目前远程网络通信中使用最广泛的一种协议, TCP/IP 是一个由不同层次上的多个协议组合而成的协议族, 共分为 4 层^[43]: 网络

接口层、Internet 层、传输层和应用层，如图 4.9 所示。

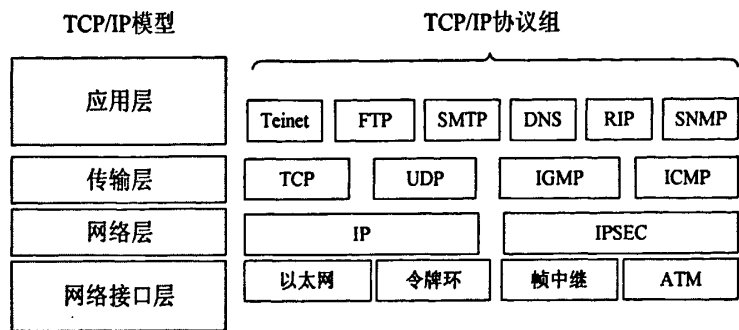


图 4.9 TCP/IP 体系结构

TCP/IP 提供给用户使用的是应用层和传输层。应用层协议直接为用户进程提供服务，它的具体内容就是规定应用进程在通信时所遵循的协议，如实现网页的 HTTP 协议、用于文件传输服务的 FTP 协议。TCP/IP 传输层的主要作用是建立网络中主机应用进程之间的通信，常采用两种协议：TCP(传输控制协议)和 UDP(用户数据报协议)。这两种协议均利用 IP 层提供的服务，提供端到端的服务。TCP 和 UDP 同时在传输层使用的事实说明：计算机网络中的效率与可靠性同样重要^[1]。表 4.4 描述了两者的区别。

表 4.4 TCP 与 UDP 的比较

	UDP	TCP
是否连接	面向非连接的服务	面向连接的服务
传输速度	快	较慢
传输可靠性	不可靠的	可靠的
通信方式	点对点 and 一点对多点	只支持点对点通信
应用场合	少量数据传输	大量数据传输（如文件传输）

考虑到本系统中偶有大量数据传输的情况，且为了保证多总线分布式网络测控系统信息传输的可靠性，采用 TCP 协议进行网络通信。传输控制协议（TCP：Transmission Control Protocol）是一种可靠的、面向连接的、不易阻塞的、具有流量控制功能的协议^[45]。提供全双工的服务，建立连接后，通信双发可以同时进行数据传输。采用字节流的方式，如果字节流太长，将其分段，这种服务给应用带来了便利，使其不必在将数据传给 TCP 之前对数据进行分块，TCP 将它们分组后交给 IP 发送。另外 TCP 对字节流的内容不作任何解释，对于 TCP 来说，传输数据字节流是二进制数据，还是 ASCLL 字符没有任何意义，对字节流的解释由 TCP 连接双方的应用层解释^[44]。TCP 数据段报头格式如图 4.10 所示。

发送端口号		接收端口号	
序列号			
确认应答号			
数据位移量		标志位	窗口大小
校验和		紧急指针	

图 4.10 TCP 数据段报头格式

根据分层原则，应用程序一般不直接涉及 TCP/IP 协议的内部，而是通过 API（Application Programming Interface）进行访问。80 年代初，加利福尼亚大学的伯克利分校在 Unix 操作系统下实现 TCP/IP 协议的 API，即 socket 接口。90 年代初，几家公司联合制定了 Windows 平台上的 Socket 接口标准，即 Winsock，它继承了 Berkeley sockets 的主要特征，并进行了重要扩充以适应 Windows 的消息驱动机制，是一套基于 Windows 的网络编程接口^[45]。本系统采用流式套接字的方式来实现网络数据通讯。

面向连接的通信分为服务器端和客户端两部分：服务器端建立 Socket 与本地的 IP 地址和通信端口进行绑定，并在绑定的端口监听消息，服务器监听到客户端的通信请求后，读取客户端的发送信息，完成相应处理后再写入处理结果，并返回给客户端。客户端需要通信时，创建本地 Socket 并完成与服务器的连接，即可与服务器交换数据。通信结束后关闭套接字，断开连接。面向连接的通信过程如图 4.11 所示。

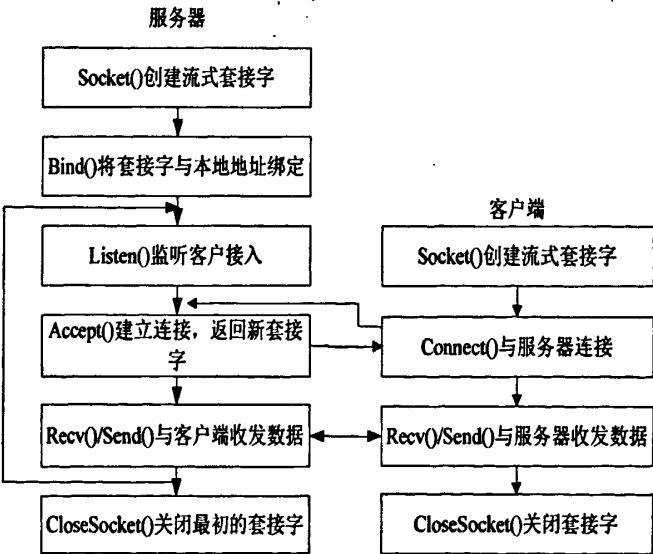


图 4.11 面向连接的通信过程

4.3.3 基于 TCP 的文件传输设计

在测控系统中，有时测试结果数据和参数设定数据并不需要立刻进行传输，如一天的电能测量，需要在虚拟仪器主机本身计算后累计形成数据，再提供给其他用户。而且这样的数据在发送端往往已经以文件形式存放。如果采用 FTP 协议传输文件，需要架设 FTP 服务器，客户端主动请求向服务器下载文件，服务器端无需编程，但无法选取所需传输的文件。若在客户端架设 FTP 服务器，发送端主动连接客户端，但随着系统的扩展，要在每台客户端上架设是不现实的。在此，这里采用 TCP 协议实现文件的传输。

本文将实现的文件传输只有一个发送方和一个接收方，这是最基本的文件传输方式，设计上采用 C/S 架构，功能上分为服务端和客户端，由于 TCP 传输协议提供了报文确认机制，所以发送方发送数据后，接收方无需在接到每个数据报文后发送确认信息，在整个文件传输完成后，接收方将文件发送的结果反馈给发送方，如果有错误则进行补包。整个传输原理为：发送方先获取待传输文件的文件名及文件长度，然后将其发送给接收方，最后将文件以二进制读入缓存的方式交给传输层利用 TCP 协议发送给接收方，与此同时，当接收方接收到文件名和文件长度之后，就为其创建新的文件和数据缓冲区，然后接收传输的文件数据，并将其放在数据缓冲区中，最后依次将数据缓冲区的数据写入新创建的文件中。这样便完成了不同计算机之间的文件传输。

然而当需要传送的文件较大时，不能直接将整个文件放入数据缓冲区，这样会引起内存不足的错误告警，这种情况下，可以将文件按固定大小分成若干数据包，再依次发送。文件分块过程如图 4.12 所示，首先设定数据包的大小 $block_size$ ，将待发送的文件分块，即 $file_size = block_size \times block_num + r_byte$ ，其中 $file_size$ 为文件大小， $block_num$ 是分块大小为 $block_size$ 的个数， r_byte 为剩下的字节数，也就是说，文件被分作 $block_num + 1$ 块。

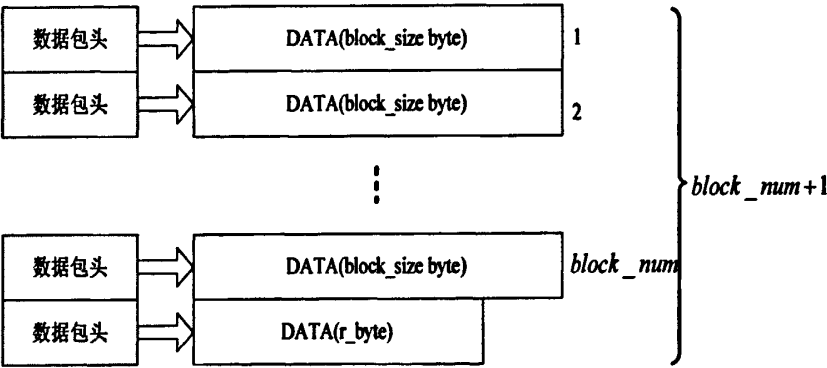


图 4.12 文件分块过程

由于 TCP 采用流协议，会根据网络状况进行 nagle 算法，或者将一个逻辑包

单独发送, 或者将一个逻辑包分成若干次发送, 或者会将若干个逻辑包合在一起发送出去。正因为 TCP 在逻辑包处理方面有这种“粘性”, 因此需要在发送端自定义包格式, 使接收端能对相应的包进行拼包和解包。自定义报文格式如图 4.13 所示。

一级包头				二级包头			数据
类型	长度	终端类型	终端号	文件块号	数据长度	是否为最后一块	
1byte	2byte	4bit	4bit	2byte	4byte	1byte	

图 4.13 自定义报文格式

无论是发送端还是接收端, 在发送数据的时候均需加上数据包头。其中, 一级包头是所有报文的公共首部, 其中长度标明整个完整报的长度, 其值是一级包头、二级包头以及数据的总和; 终端类型, 终端号标识此数据包发送的目的地。根据一级包头中类型的不同, 二级包头也不相同, 图 4.13 中所示的是文件传输时的二级包头, 包括: 文件块号, 标明此数据包为传输文件中的哪一块; 数据长度, 标明数据分块的大小 `block_size`; 是否为最后一块, 标识文件的发送是否结束。文件传输流程如图 4.14 所示。

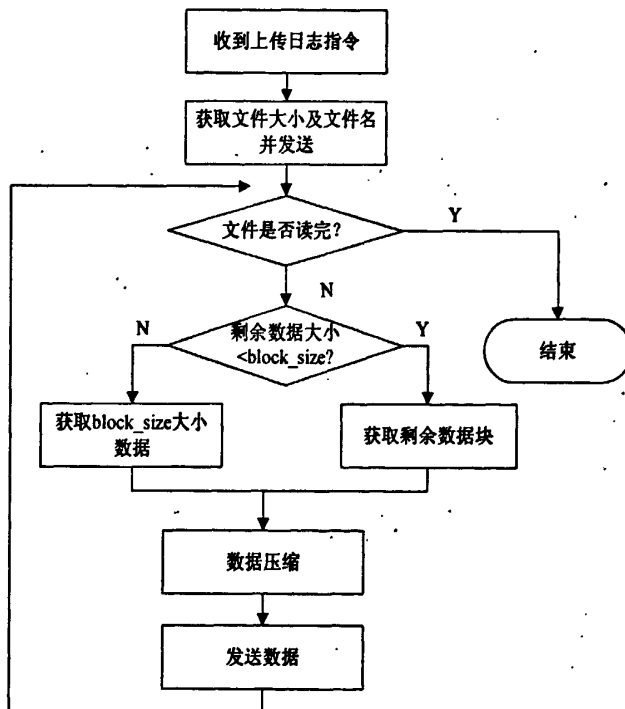


图 4.14 文件传输流程

接收数据后的解析过程如下: 数据先从 socket 缓存拷入应用层缓存, 解析一级包头判断此数据包目的地是否是本机, 数据是否为文件类型, 如果是, 判断包

长度并与缓存里数据长度比较，若缓存数据包长度较短就继续等待，否则逐个提取完整的包并解析二级包头和数据解码，提取完就继续等待下次数据的传输，循环直至整个传输任务结束。

4.3.4 文件传输时间的测量

1、高精度定时器的实现

考虑到 LabVIEW 中定时器控件精度最高只能达到 1ms，为了使测量结果更精确，在实验测量中，采用 CLF 节点调用 system32 目录下的 kernel32.dll 文件中的 API 函数 QueryPerformanceFrequency 和 QueryPerformanceCounter 可获得 μs 级的时间。分别记录在 t_1 和 t_2 时刻调用 QueryPerformanceCounter 获得的定时器的计数值，其差值除以由 QueryPerformanceFrequency 函数获得的计数器频率，乘以 1000，即可得到 $t_2 - t_1$ 的时间(ms)。确切的精度依赖于电脑 CPU 的主频，其中 QueryPerformanceFrequency 函数获得的是系统一个高精度定时器在 1s 内走过的计数值，电脑 CPU 主频不同，该值也不同，在本系统所采用的 PC 机上，该值为 3014530000，也就是说，该定时器的精度为 $0.0003 \mu s$ 。高精度定时器的实现如图 4.15 所示。

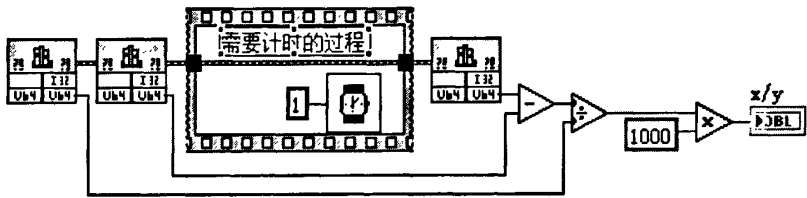


图 4.15 高精度定时器的实现

2、文件传输时间的测量

根据时延的计算方式不同，网络的传输时延可以分为：单程时延和往返时延 [46][47]。单程传输时延是指从源端机器开始发送数据到目的端机器接收到测试报文所需要的时间。由于两台机器的系统时钟很难精确同步，一般网络传输时延是指测试报文的往返时延。网络传输时延的测量平台为局域网内两台计算机。交换机使用的是 24 口普联 TL_SF1024 10/100M 自适应以太网交换机。服务器与远程监控主机为 Windows 操作系统，服务器的 IP 地址为 219.245.65.196，客户端的 IP 地址为 219.245.65.224，通信端口号为 8080。

本节测试的是较大的数据包传输时间，采用服务器客户端请求应答的方式计算传输时间，在客户端运行的程序请求发送数据，服务器端监听到该请求便发送测试数据包，并记录发送时刻 t_1 ，在客户端上运行的程序数据接收完毕后反馈数据接收完成的信息，服务器收到后记下数据返回时刻 t_2 ，由于无论通信量的大小，

交换式以太网的延迟保持在 1ms 以内, 延迟非常小以及波动可以忽略不计^[48], 因此返回时刻 t_2 和发送时刻 t_1 之差即为测试数据传输时间的近似值。

采用上述方法(基于 TCP 协议的网络通信模块 LabVIEW 实现将在 5.3.1 节描述), 发送大小为 3291kB 的数据包, 分别测试未压缩与压缩后的数据包的传输时间, 试验次数为 1000 次, 并将测试的传输时间存在文本中, 并采用 matlab 显示仿真结果如图 4.16 所示。未压缩文件传输时间最大值为 561.9ms, 最小值为 324.8ms, 均值为 349.8ms; 压缩后的文件传输时间最大值为 470.3ms, 最小值为 200.3ms, 均值为 204.6ms。

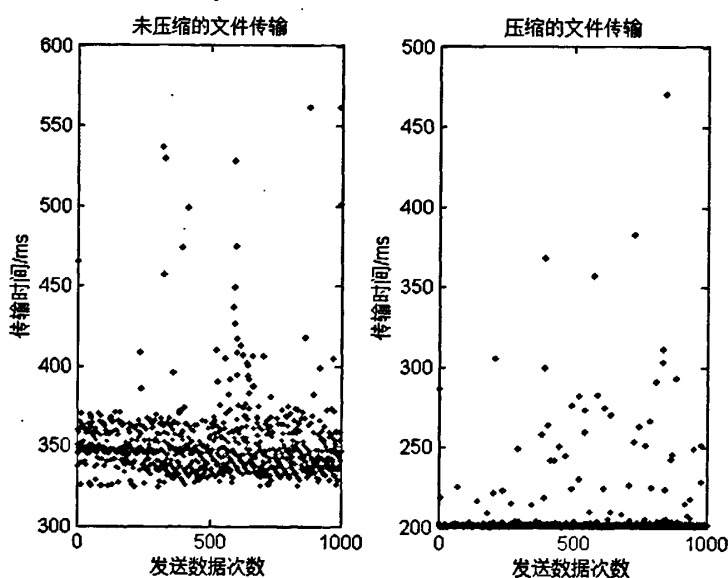


图 4.16 文件传输时间

从图 4.16 中可知, 在多次测量过程中网络传输延迟具有随机性, 因此取平均值更有利于分析。而且从以上实验结果可以看出, 在网络传输中, 传输数据的大小直接影响网络传输时间。由于数据在通过压缩之后数据量都会变小, 这就减少了网络传输量, 节省了通信时间, 而且无损压缩在传输后进行解码也可以取得原文件的全部内容, 这样缩短了中间传输的过程, 加大了传输前和传输后计算机的任务, 减轻中间网络的任务量, 从而达到了任务的平均合理分配。

4.4 本章小结

研究了组建网络化虚拟仪器的两种模式, 结合本系统的实际情况, 设计了基于 C/S 模式的传输系统, 并对传输系统中的 ARINC429 总线技术和以太网技术进行研究, 在此基础上研究了循环冗余校验编译码原理, 并与奇偶校验结合, 并应用于 ARINC429 传输系统中, 提高数据传输的可靠性。搭建海量数据传输时

间的测量平台，并对测量结果进行分析。

第五章 基于虚拟仪器的某型号综合测控系统的实现

5.1 系统方案规划

5.1.1 系统总体结构

随着现代信息技术的高速发展,为测控系统的自动化与网络化提供了技术支持,测试结果早已不再满足于人的眼观手记,远程监测和控制也成为自动测试系统的一项基本功能。针对这一发展趋势,本论文设计的测控系统以数据的处理及高速可靠传输为主要研究内容,利用总线技术、数据压缩与处理、网络传输、纠错码技术和图形化界面软件设计技术等,设法模拟出被测设备的各种典型故障,并向被测设备提供正常运行时所需的激励信号或故障激励信号,同时监测被测在各种工作状态下的控制及保护功能、与系统其它设备之间的通信接口。而且在其缺件的情况下,能替代它与系统其它设备交联。并将测控系统工作过程中的控制和监测信号显示在人机界面上,方便测试人员调试被测产品及评价产品的可靠性。

本系统以高性能的工业控制计算机作为硬件平台,以 LabVIEW 作为软件开发环境,通过 ARINC429 将测控系统与产品连接起来,通过以太网将各分系统连接。测控系统(称为“下位机”)监听自动管理系统(称为“上位机”)指令,或通过测试人员在人机界面上的设置,向产品提供与指令相关的激励信号,将状态信息和通信数据显示在界面上,并自动存储在工作日志中,必要时向上位机上传工作日志。系统组成框图如 5.1 所示,本文所研究的测控系统为图中阴影区。

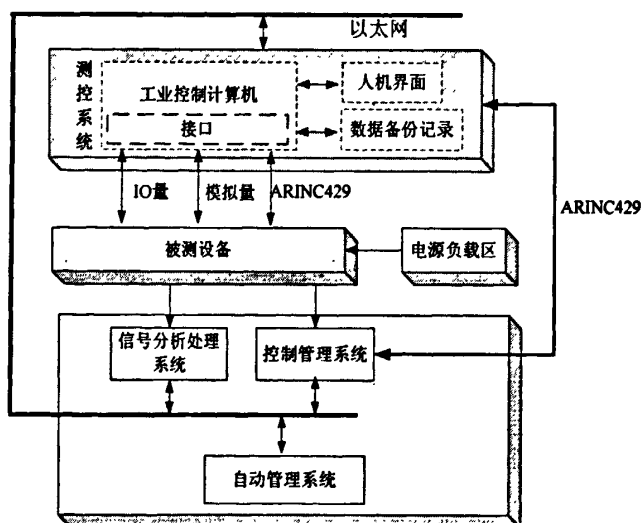


图 5.1 系统组成框图

5.1.2 自动测试系统硬件组成

自动测试系统硬件组成框图如图 5.2 所示。

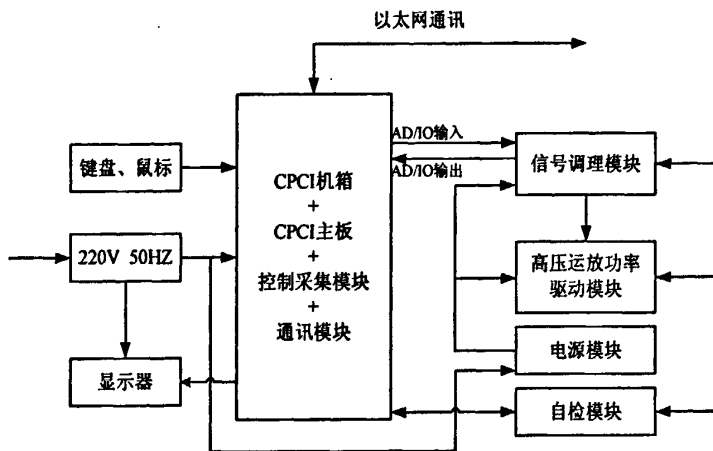


图 5.2 硬件组成框图

其中，

1) CPCI 主板集成多网口；

2) 信号调理模块完成对 AD、IO 输入量的电压转换跟随、电平转换和隔离；完成对 DA、IO 输出量的电压分压转换、高压运放驱动控制、电平转换和信号隔离；

3) 高压运放模块由电源产生 28V 直流电压后，经电压放大模块，产生 90V 的直流高压，再经驱动高压运算放大器，将 DA 模块产生的电压信号放大到产品测试所需的电压值，供产品使用；

4) 功率驱动模块弥补高压运放模块输出功率驱动的不足，避免输出的电压波形由于产品本身的负载而被拉低；

5) 电源模块可以生成系统工作所需要的 +5V、+15V、+28V 等工作电源，为实现被测设备与外部所有设备的完全隔离，使用两套电源模块分别对产品与外部设备进行供电；

6) 自检模块可完成对 AD、IO 输入量，DA、IO 输出量，ARINC429 信号的自检，提高设备可靠性。该模块通过软件控制，通过环绕测试、提供可靠输入源等方式完成对硬件的自检。

7) 多功能采集板卡采用 32 通道单端或 16 通道差分模拟量输入，16 位 AD 分辨率，±10V 输入范围，最高 250KHz 采样率，4 路 16 位 D/A 输出，±10V 输出范围，48 通道 TTL 电平双向 DIO；通讯板卡采用 4 发 4 收通道配置，可选择发送和接收字符格式转换，接收 FIFO 触发深度可设，波特率 12.5Kbps、48Kbps、50Kbps、100Kbps 可设置，2 种数据长度，32 位和 25 位，有自检功能，奇偶校验

软件可设。

5.1.3 测试系统的软件结构

在本系统中涉及到的数据采集及输出操作有：采集开关量及模拟量监测被测设备、输出开关量及模拟量控制被测设备。涉及到的通信方式有两种：ARINC429通信和基于TCP/IP的以太网通信。ARINC429总线负责将本系统与控制管理系统、被测设备的互联；以太网通信负责本系统与自动管理系统的互联。

系统具有激励器及仿真器两种工作模式，在激励器功能模式下能够模拟被测设备实现各种工作状态下控制及保护功能所需要的激励信号特性，并提供给被测设备；在仿真功能模式下，实现被测设备功能及接口仿真，在被测设备缺件时替代它。系统具有本地和远程两种控制模式，在本地模式下，通过界面操作设置激励信号的输出；远程模式下，监听自动管理系统网络指令，根据监听内容控制被测设备（设备不存在则根据监听内容仿真），并回传网络监听状态。系统实时存储被测设备的状态，包括采集到的开关量、模拟量和通信数据，在收到自动管理系统上传日志指令时，压缩工作日志文本并上传。

为实现上述功能要求，本文基于NI公司的虚拟仪器开发平台LabVIEW设计开发了某型机载设备数据处理与传输系统软件。在前几章的基础上，采用模块化的程序设计方法，遵循自顶向下逐步细化的原则设计系统软件，使系统软件易于调试、测试和维护。为了使系统软件满足简单易读、易于维护、结构清晰等特点，在软件设计之前根据总体方案的设计对程序的设计结构进行规划，对测控系统进行软件功能模块的划分，测控网络系统下位机软件结构如图5.3所示。

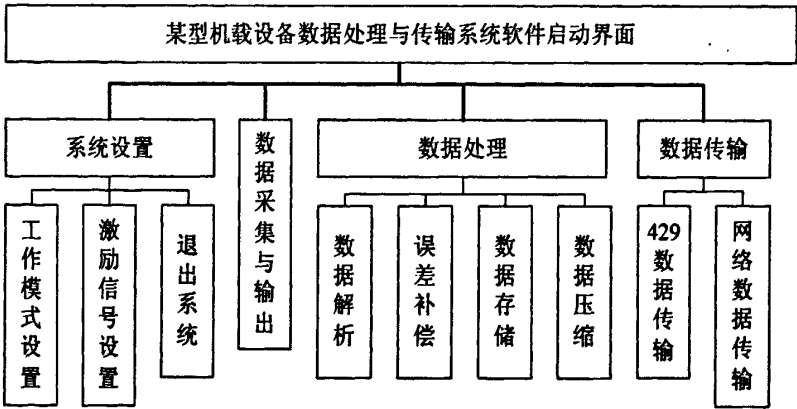


图 5.3 软件功能模块的划分

本系统由系统设置、数据采集及输出、数据处理和数据传输四大模块组成。系统设置主要是通过人机界面对系统的控制；数据处理与传输工作是本文研究的重点。由于各模块之间并不是绝对孤立的，各个模块需相互配合，进行信息交流

才能完成系统任务,为了清晰的表述系统中软件功能模块间的信息交流,软件模块数据传输的方向如图 5.4 所示。

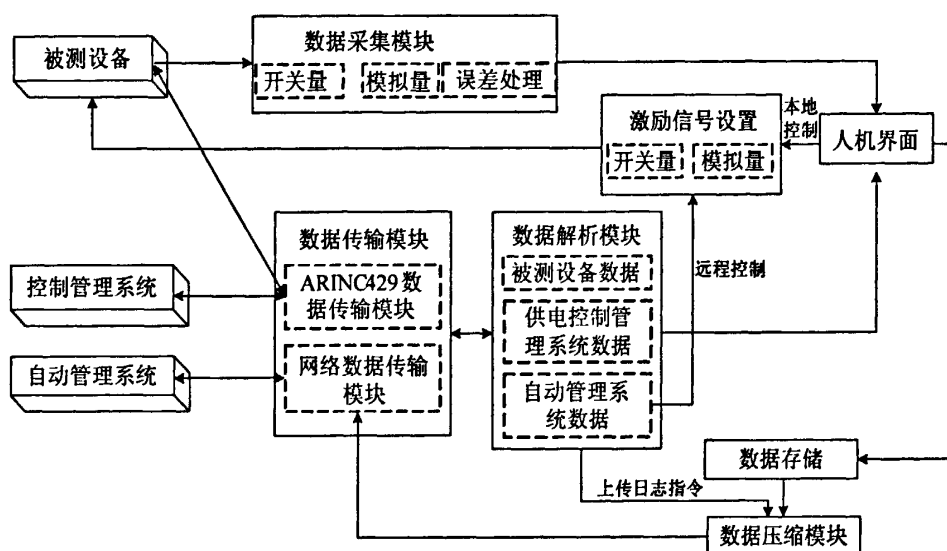


图 5.4 软件模块数据传输方向

5.2 数据处理技术在系统中的实现

5.2.1 误差补偿的软件实现

本系统设计的自适应误差补偿方法中的输入输出曲线是根据离散的 AD 采集和 DA 数据,通过最小二乘曲线拟合算法得出的。LabVIEW 中包含了大量的数据曲线拟合函数以满足不同的拟合需要。譬如线性拟合 (Linear Fit)、指数拟合 (Exp Fit)、幂拟合 (Power Fit)、多项式拟合 (Polynomial Fit)、最小二乘法拟合 (Gen.LS.Lin.Fit) 等。为误差补偿的软件实现提供了便利。自适应误差补偿实现步骤如下:

1) 训练序列的选择: 根据第三章中提到的训练序列的选取规则,考虑到本系统中多功能数据采集卡输入输出范围为 $\pm 10\text{V}$, 选训练序列为 $\pm 10\text{V}$, $\pm 8\text{V}$, $\pm 5\text{V}$, $\pm 4\text{V}$, $\pm 2\text{V}$, $\pm 1\text{V}$ 。

2) 通过 DA_OUT.VI 访问多功能数据采集卡将训练序列从小到大依次输出,每个输出点时间间隔为 300ms。输出训练序列中的某一个点后,调用 AD_IN.VI 以 20ms 为周期循环采集 10 次,对采集到的数据求取平均值,尽量消除或减小随机误差。

3) 将输出的训练序列存放在 X 数组中,采集到的 12 个数据存放在 y 数组中,如此便得到了 Gen.LS.Lin.Fit.vi 的输入参数,即数据的坐标 (x_i, y_i) 。从而得到输

入输出函数的最佳系数。并根据最佳系数生成输入输出曲线公式。

4) 将得到的最佳系数以文本文件的形式存放系统软件所在目录下, 在设备及其环境没有变化或变化很小的情况下, 可调用“Read from text file.vi”读取最佳系数。

5) 若需要重新生成输入输出函数, 点击人机界面上“误差补偿”按钮, 以事件触发的机制重新进行(1)~(4)步骤的过程。

其流程图如图 5.5 所示:

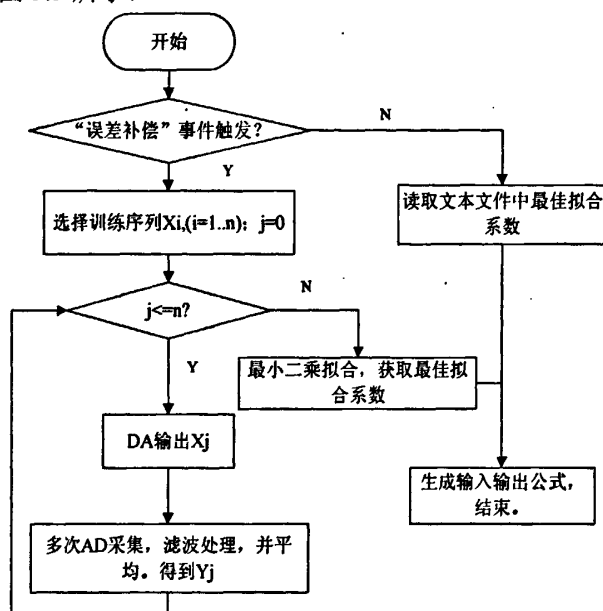


图 5.5 自适应误差补偿流程图

5.2.2 数据压缩算法的实现

对原始数据做预处理, 将处理后的数据以字节为单位读取。因为编码位数与字典容量是相关对应的, 在压缩程序中设置两个变量 `code_bits`、`dict_byte` 分别代表代码位数与字典大小; `next_code` 表示字典中的表项; 以 `Prefix` 表示前缀; `character` 表示当前字符, `code_value[]` 表示编码值数组, `code` 表示编码器需要输出的编码值。根据 LZW 压缩算法的原理, LZW 编码器采用了一种贪婪分析算法, 每一次分析都串行检查来自字符流的字符串, 从中分解出已经在字典中出现的最长的短语, 输出其对应的索引值, 将该短语与下一个输入字符串联形成新的扩展字符串, 加到字典中。本文采用 8~12 位的可变长编码, 串表项最多可达到 4096 个。编码步骤如下:

1) 初始化字典函数 (InitDictionary): 初始化字典, 使其包含可能的单字符。分别表示第 0~15 项为十六进制 0~F, 第 16 项为空格, 第 17 项为变长标识

BIT_CHANGE, 第 18 项为清除字典标识 CLEAR, 第 19 项为结束标识 End_Code, 这 20 项为字典的根。

2) 开始编码 (compress): 由于字典前 20 个索引被占据, 因此初始赋值 $next_code=21$, 此时编码长度为 8 位, 字典可容纳 256 个短语, 因此赋值 $code_bits=8$, $dict_byte=255$ 。从文本文件中读取数据流中的第一个字符 character, 此时 prefix 为空字符串, 调用 find_node 函数在字典中查找 prefix+character 字符串, 若找到返回相应索引值 index, 并将此字符串作为新的前缀 prefix, 继续读取源文件中的下一字符, 并重复进行上述查找过程; 若在字典中没有所查找的字符串, 将 prefix+character 字符串存放在字典的 next_code 的项, 且 $next_code++$, 并将前缀 prefix 设置为当前字符 character, 调用 output 函数输出编码 code, 即 $code_value[index]$ 。

3) 循环执行第二步, 随着压缩过程中字典短语越来越多, next_code 达到 256、512、1024 或 2048 时, 编码长度 $code_bits++$, 此时字典容量 dict_byte 加倍, 为方便解压时识别代码长度的变化, 调用 output 函数输出变长标识 BIT_CHANGE, 当解压时发现该标识, 编码位数增加一位。若 next_code 达到 4096, 说明字典已满, 重新开始执行第一步, 初始化字典。与传统的 LZW 算法相比, 变长编码在存储代码时位处理过程在 output 函数中实现, 编码器变长输出流程图如图 5.6 所示。

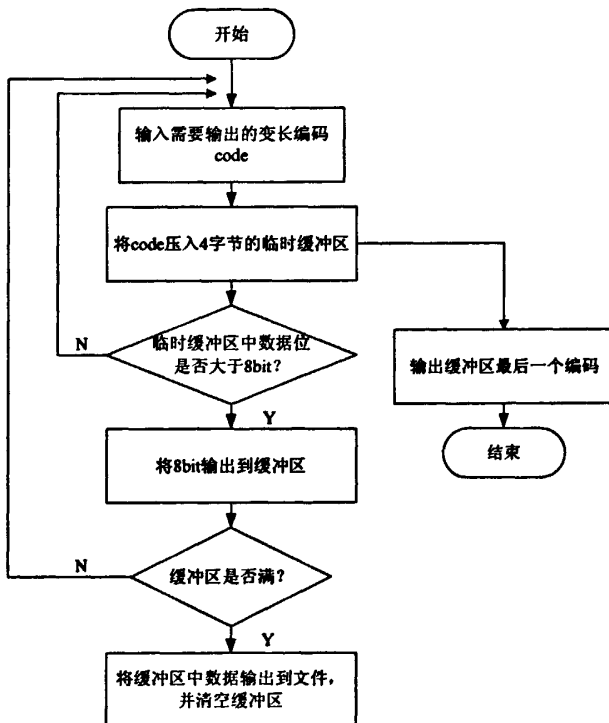


图 5.6 编码器变长输出流程图

5.2.3 离散量与模拟量数据采集

由于本课题中采用的是北京神州飞航 AECAD-CPCI-32M 多功能数据采集卡, 且硬件提供厂商已提供了动态链接库接口, 因此采用 CLF 节点调用板卡的 .dll 库函数完成 LabVIEW 中驱动接口的设计较为简便。

初始化多功能采集板卡的操作包括以下几步: ①调用 AECCP32ADM_Open 函数打开板卡, 并通过板卡打开函数返回板卡句柄, 之后的各种板卡操作均通过句柄唯一标识该板卡; ②若成功打开板卡, 就根据硬件接口状态对 AD 采集增益和数字量的输入/输出模式进行配置, 通过 AECCP32ADM_AD_SetGain 函数设置增益为 1, 通过 AECCP32ADM_IO_Direction 函数, 设置第 0~15 路为数字量采集端口, 16~47 路为数字量输出端口。

本模块主要实现对外部直流电压的采集和地/开离散量的采集工作, 分别在子 VI “AD_IN.VI” 和 “IO_IN.VI” 中实现。采用软件触发的方式对数据采集卡进行编程, 在系统提供的时钟在毫秒级的精确等级上, 通过对寄存器的查询来实现数据采集。通过 AECCP32ADM_AD_Single 函数进行单次采集, 其中寄存器存放数据的基本单位为 16 位, 寄存器的长度等于每个通道采样点数与通道数量的乘积, 数据在寄存器中存放的格式如图 5.7 所示。

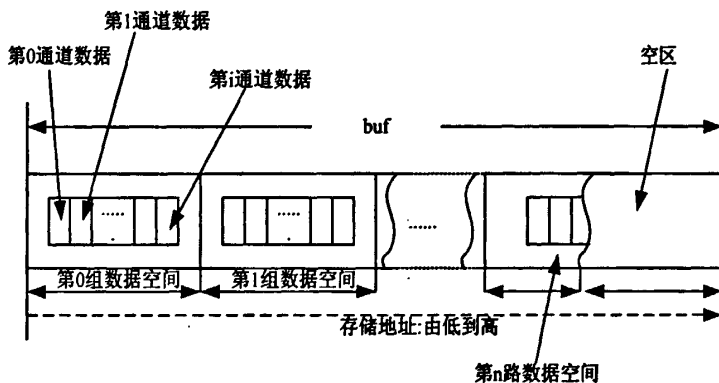


图 5.7 数据在寄存器中存放格式

在 “IO_IN.VI” 中, 需要采集 12 路地/开离散量。根据采集到的值规定: “0” 表示开路, “1” 表示地。最后将采集到的离散量值缓存到名为 “IO_DATA” 的队列中, 等待需要此离散量的子 VI 的读取与处理。最后将采集到的模拟量缓存到名为 “AD_DATA” 的队列中, 等待误差补偿模块的处理。

5.3 测控系统通信模块的设计与实现

5.3.1 与自动管理系统通信

系统与上位机之间的网络通信数据可分为两种：一种是较小的数据包，包括上位机发出的指令数据、系统反馈的状态信息；另一种是文件数据，是测试系统上传至上位机的工作日志文件。这种情况下希望通信双方能够互连数据通道，像计算机测控系统中用总线操作那样完成网络通信任务，采用流式 Socket 的通信机制比较适合。

1、网络通信模块的实现

在 Labview 中用于 TCP 编程的 VI 函数位于程序面板的函数->数据通信->协议->TCP 面板下。从表 5.1 可以看出，Labview 提供了 TCP 编程的通用 VI 函数，且对这些函数操作也较为方便。

表 5.1 TCP 编程通用 VI 函数

VI 函数名称	功能
TCP Listen	在指定端口创建一个监听端，并等待客户端的连接
TCP Create Listener	在指定端口创建一个监听端
TCP Wait On Listener	在指定端口等待客户端的连接
TCP Open Connection	打开与远程服务器端的连接
TCP Read	从指定的 TCP 连接读取数据
TCP Write	从指定的 TCP 连接写入数据
TCP Close Connection	关闭指定的 TCP 连接
IP To String	将 IP 地址转换为计算机名称
Sting To IP	将计算机名转换为 IP 地址
Resolve Machine Alias	通过 VI 服务器或计算机别名返回计算机的 IP 地址

由于 TCP 提供的是全双工通信，服务器端与客户端可同时进行交互式通信。且 TCP 自动管理数据分组、排队等，因此不会造成冲突。为符合程序模块化的设计思想，网络数据的发送及接收过程封装到子 VI “TCP.vi”中，通过主程序对其调用实现测控系统接收来自自动管理系统的网络指令的工作，并反馈状态信息以及上传工作日志至自动管理系统的工作。这样设计系统主程序时就只需要关心系统功能的实现部分，更利于调试时的问题定位和系统功能的维护。

测控系统作为客户端与自动管理系统通信步骤如下：

1)在需要发送网络数据时，用 TCP Open Connection 节点建立一个 TCP 连接。对各参数设置如下：

address 设置为远程服务器地址。该地址可以为 IP 句点符号格式或主机名。

Remote port 设置为用户欲创建的 TCP 连接的端口号, 设置为 3030。相应的远程自动管理系统与本系统通信的接收端口设置为 3030。

返回值 connection ID 是唯一标识 TCP 连接的网络连接引用句柄。

2) TCP Write 通过 data in 端口将需发送的数据写入指定的 TCP 端口。

3) 采用 TCP Read 函数读取数据长度, 经 Type cast 将返回的数据长度转换为 32 位整型。然后根据数据长度, 再采用 TCP Read 函数读回实际数据。

4) 当系统软件退出时, 应及时关闭已建立的 TCP 连接, 以释放资源。利用之前建立 TCP 连接时返回的 connection ID 找到相应的网络连接, 调用 TCP Close Connection 函数, 关闭该连接。

2、与自动管理系统通信的软件实现

“TCP.vi”在定义了图标和端子后, 可供主程序和其他 VI 调用, 此处讨论如何实现与自动管理系统的信息交互。本系统与自动管理系统的信息交互流程如图 5.8 所示。

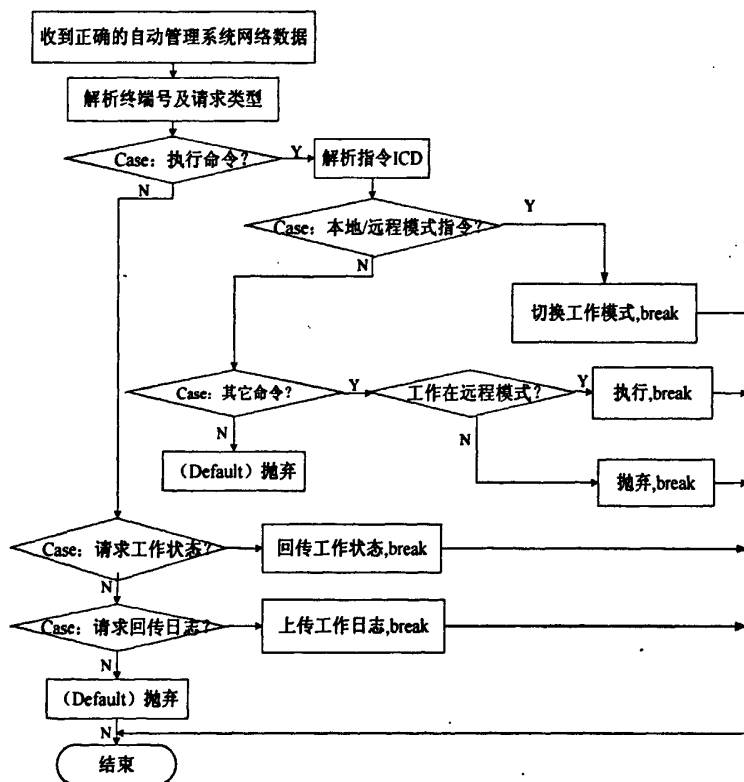


图 5.8 与自动管理系统的信息交互流程

第一类数据的数据帧格式如表 5.2 所示。前 9byte 的数据为信息位, 最后 2byte 数据为校验位。测控系统反馈状态信息至自动管理系统时, 调用上述已封装好的“CRC16.vi”计算出信息位的校验码高、低位, 并分别填充在校验位的第 10、11 字节。按照数据帧的格式, 调用“TCP.vi”即可实现可靠的数据发送。收到自动

管理系统的数 据时，将收到数据的前 9 字节作为“CRC16.vi”的输入端，若输出端的高位与收到数据的第 10 字节相同，且低位与接收数据的第 11 字节相同，表明接收数据正确，否则说明接收数据发生误码，需要向自动管理系统反馈“校验错误”状态信息。

表 5.2 与自动管理系统通信数据帧格式

类型	包长 度	终 端 类型	终 端 号	请求/应答 类型	指 令 ICD	版 本 号	CRC 校验
1bytes	2bytes	4bit	4bit	1bytes	1bytes	2bytes	2bytes

第二类数据属于文件类型，为了减少网络流量，缩短传输时间，在发送文件前采用 LZW 压缩算法对数据进行压缩。上传的工作日志主要包括三个信息：文件大小、文件扩展名和文件内容。本文实现的文件传输只有一个发送方和接收方，文件传输具体实现步骤如下：

- 1) 获取工作日志所在路径，将路径输入到“Open File.vi”获得文件引用句柄，“Get File Size.vi”返回文件大小（以字节为单位），经“Number to Decimal String.vi”转为字符串类型，并根据该长度创建数据缓冲区，并调用“TCP Write.vi”发送至接收方告知文件大小。
- 2) 将文件路径拆分后输入到“Match Pattern.vi”转换为字符串格式，调用“TCP Write.vi”发送文件扩展名至接收方，如此接收方便能根据文件名和文件大小创建新的文件和数据缓冲区。
- 3) 通过“Read from Binary File.vi”读取文件信息，并调用“TCP Write.vi”发送文件内容。发送完毕后，调用“Close File.vi”关闭对文件的引用。

5.3.2 与其他分系统的通信

1、ARINC429 通信软件的实现

由于本课题中采用的是北京神舟飞航科技有限公司生产的 CPCI-44/S1 通讯卡，采用 CLF 图标来调用板卡的.dll 库函数完成 LabVIEW 中驱动接口的设计。

在通讯卡中，设计有 FIFO 数据缓存区，新数据仅仅被添加到 FIFO 数据缓冲区的第二个位置上，而不会覆盖原先的数据，随后的数据会依次排列到缓冲区中。当想从 FIFO 数据缓冲区中搬移数据时，仅仅需要从数据寄存器中读取一个数据即可，这样会把最初的数据取出，FIFO 数据缓存区中下一位数据会取代数据寄存器中的值，可以在任何时候传输来自 FIFO 数据缓冲区中的数据，当在传输旧的数据时，硬件会将新的数据保存在 FIFO 数据缓存区中，从而防止数据丢失。本模块以软件查询的方式，即通过查询状态寄存器的状态来判断 FIFO 数据缓冲区的状态使用 FIFO 数据缓存区。

在程序设计过程中根据实现功能的不同,将访问 ARINC429 通讯卡的操作封装在三个 VI 中:初始化板卡的子 VI “Init_429.vi”、发送数据的 VI “send_429.vi”、接收数据的子 VI “Rev_429.vi” 和关闭板卡释放资源的子 VI “close429.vi”。

1) 初始化板卡 (“Init_429.vi”) 包括以下几步: ①由于驱动程序采用面向对象编程,所以要使用设备的一切功能,首先必须用 AEC429A_Open 函数创建一个设备对象句柄 hDevice,有了这个句柄,用户才拥有对该设备的绝对控制权。将该句柄存储在全局变量中,供操作该板卡的其他函数使用; ②成功获得了 hDevice 设备对象句柄后,便可初始化板卡各通道参数。调用 AEC429A_Reset 函数复位板卡,并清空板卡的接收 FIFO; 若复位板卡成功,便可用 AEC429A_SetRxCfgWord 函数设置各接收通道配置字,关于数据位长度、波特率参数的设置是由这个函数的 CfgWord 参数结构体决定的,对这个 CfgWord 参数结构体各成员赋值使各接收通道数据位长度为 32 位、波特率为 100Kbps; 并用 AEC429A_SetTxCfgWord 函数设置各发送通道配置字,通过对参数结构体各成员赋值,设置发送通道发送速率为 100Kbps、发送字长度为 32 位、发送校验使能奇校验、正常工作状态;最后设置 AEC429A_WordConvertEn 函数中参数 Enable 为 True,使在发送与接收 ARINC429 数字字时自动完成数据位的转换。以上实现了通讯板卡所有硬件参数和状态的初始化。

2) 发送数据 (“send_429.vi”): 使用 AEC429A_TxFIFOSTR 读取发送 FIFO 的状态,若发送缓冲区未满,则调用 AEC429A_TxWrite,设置相应参数发送数据。

3) 接收 ARINC429 数据 (“rev_429.vi”): 采用非空查询的方式实现 ARINC429 数据的接收,调用 AEC429A_RxFIFOSTR 函数判断接收缓冲是否为空,若为空则等到下一个周期再次判断;若不为空使用 AEC429A_RxRead 函数将数据从板卡缓冲区取出并存入名为 “DATA429” 的队列中,等待数据处理模块的循环对数据进行判断和解析。

4) 关闭板卡 (“close429”): 当系统软件关闭时,需要释放通信板资源,有利于对通信板的保护。函数 AEC429A_Close 通过板卡句柄找到相应板卡将其关闭。

上述三个 VI 功能完成后为它们定义连接端子和图标,使它们能作为子 VI 被主程序 VI 或其他 VI 调用,实现 ARINC429 通信的模块化的设计,便于程序的继承及维护。

2、系统中的 ARINC429 通信

在本论文设计的测控系统的工业控制计算机 ARINC429 总线上连接了 3 个接收设备,分别为被测设备 1、被测设备 2、控制管理子系统,而这三个设备又各自拥有一根独立的 ARINC429 发送总线,用来发送状态信息或指令给测控系统。

控制管理系统主要实现对系统各控制装置的维护管理,完成上电 BIT、周期

BIT 和维护 BIT 等自检测功能。在设计与控制管理系统的通信模块时须考虑两种情况：①在两台被测设备同时存在于系统中，即都不缺件的情况下，系统无需代替被测设备与控制管理系统通信；②否则代替缺件的那台设备与控制管理系统通信。

被测产品的功能是检测系统的电压、电流、接触器状态等参数和状态信息，根据检测到的信息自动完成对系统的正常工作和故障隔离控制，并通过 ARINC429 数据总线将系统的工作参数、状态和告警信息上传给控制管理系统和本测控系统。本系统将完成对两台机载设备的测试工作，通过 ARINC429 数据字 SSM 字段进行编码区分，并可在界面中分别查看和控制。根据机载设备是否缺件，本系统功能模式可分为两种：激励器模式和仿真器模式。在某台机载设备缺件的情况下，对应的功能模式为仿真模式；否则为激励器模式。功能模式的切换可通过操作界面上设置完成，也可通过自动管理系统发送工作模式指令实现切换。

5.4 软件系统界面设计

在软件的启动的程序框图中，顺序初始化多功能采集卡和 ARINC429 板卡。本文为实现程序模块化的设计，将多功能采集卡及 ARINC429 板卡的初始化函数分别封装在 “Initdaq.vi”和 “Init429.vi”子 VI 中。在软件启动的程序框图中调用子 VI 即可完成对多功能采集卡和 ARINC429 卡的初始化，并将返回的句柄存放在全局变量“板卡句柄”中。最后将初始化的结果显示在界面中，如果以上初始化函数返回为真，则表示初始化成功，在界面相应的显示框中显示“正确”；否则显示“错误”。初始化成功的软件启动界面如图 5.9 所示。



图 5.9 系统软件启动界面

如果初始化成功则待状态显示完成后延时 1s, 退出启动界面并打开主程序; 如果初始化失败则显示对话框, 用户选择进入主程序或者退出测控系统, 在这种情况下即使选择进入了主程序, 系统也无法完成测控任务中数据采集、输出及传输的工作。系统主程序界面如图 5.10 所示。在主程序界面显示不可与数据采集及接收频率达到一致, 数据采集和 ARINC429 数据的接收周期都为 200ms, 若将每次采集或接收的数据都显示在界面上, 不仅会占用大量的系统资源, 而且还会使界面上模拟量数据因不断刷新而无法读取。因此在本系统中, 采集数据的动态显示设置为以 1s 为周期刷新。并将各种显示数据和控制按钮有序的排列在前面板上, 使测试人员能够直观以及方便的进行设备的监控。

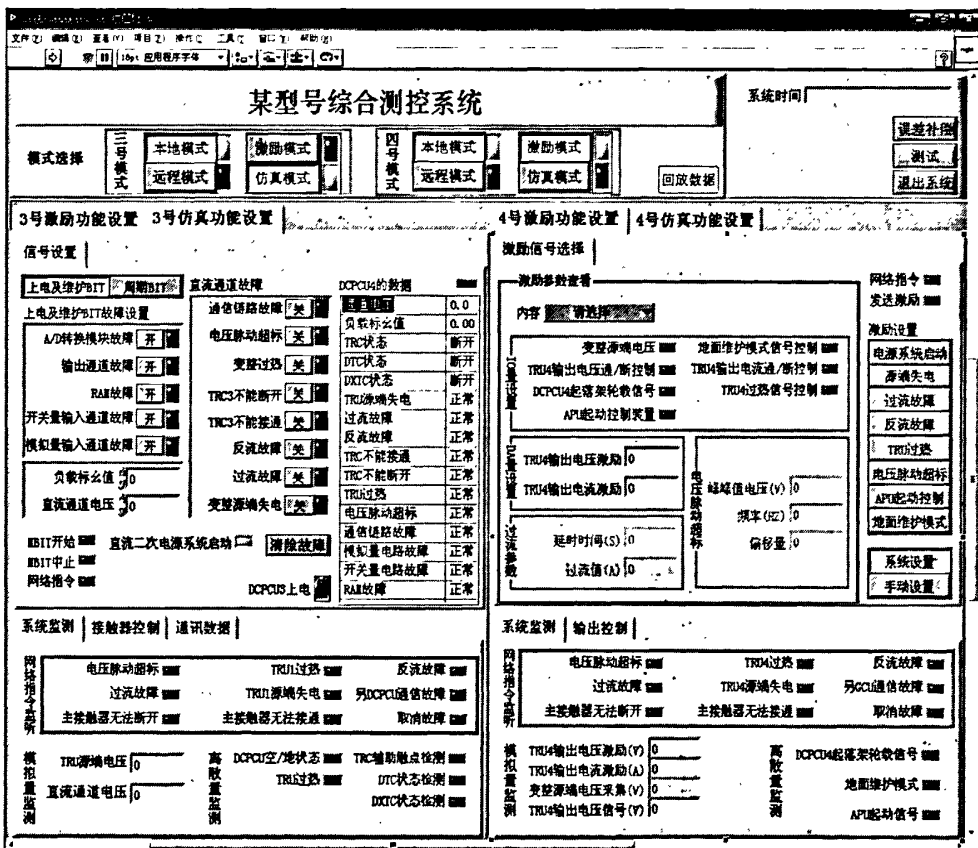


图 5.10 系统软件主界面

5.5 本章小结

本章详细阐述了 LabVIEW 开发环境下某型号综合测控系统的实现, 包括系统总体设计方案的规划和各功能模块的具体软件实现, 最终将航电数据通信系统以人机界面的形式呈现出来, 完成了系统的软件平台搭建。

第六章 总结和展望

基于虚拟仪器的测控系统的开发是测控领域发展的重要方向,具有极大的市场潜力和应用前景,将给社会带来极大的效益。随着社会的发展,测控对象越来越复杂,测控范围也越来越广,需要解决虚拟仪器测控系统中远程数据采集与控制、远程监控等具体问题。

本课题在理论研究的基础上进行工程实践,针对测控数据的处理与通信技术进行研究,探讨了提高数据采集精度的方法、基于 ARINC429 总线和以太网技术的数据传输。

本文主要完成的工作如下:

- 1、为提高数据采集的精度,保证测试数据的准确性,对减小或消除系统误差的方法进行了研究,并将系统误差补偿方式移植到虚拟仪器系统中,改变了测量人员对误差数据的手动读取与测量的方式。

- 2、为提高海量数据传输的效率,对 LZW 压缩算法进行了改进,结合测控数据两次数据采集相隔时间较短,且离散量数据变化次数较少的特征,对数据进行预处理。并从编码输出长度及字典清除规则两个方面对 LZW 压缩算法进行改进,使压缩比在原算法基础上降低了 5%,改善了压缩效果。

- 3、研究了基于 C/S 模式与 B/S 模式的网络化虚拟仪器的构成,考虑到实际系统中数据传输量较大,为保证系统实时性需求,设计了基于 C/S 模式的网络化虚拟仪器系统,并采用 ARINC429 接口总线组建复杂的虚拟仪器系统,使测试人员无论任何地点、任何时间都能进行远程操作、获取测试信息。突破了将仪器的三大功能模块(信号采集、信号分析和显示)都集中在一台机器上的模式,实现资源共享。

- 4、将理论应用到工程实践当中。将通过理论研究得到的数据处理及传输技术应用到实际的测试系统中,并将采集及传输数据以人机界面的方式呈现,并为测试人员直观的操作提供了条件。

本文虽然提出了一些改进,并实现了系统中提出的相关技术,但还需在以下方面做进一步的研究和完善:

- 1、由于综合测控系统中电磁环境非常复杂,采集到的信号除了系统误差外往往还受到随机误差的影响,因此需要在信号去噪方面做深入的研究,并应用在系统中。

- 2、随着基于虚拟仪器的测控系统在军事上的广泛应用,各种数据要在不同的分系统之间进行信息交换,为了保证信息安全,需要在数据传输加密方面做进一

步研究。

3、应用程序可以进一步优化，系统中任务多而复杂，如何实现任务的平台级和用户级二级调度，以及如何充分利用 LabVIEW 自动多线程的特点，需要做进一步研究。

由于作者水平有限，论文中的缺点和不足之处在所难免，恳请各位老师、专家、学者批评指正。

致 谢

首先，衷心感谢我的导师相征教授。本论文的全部工作都是在导师相征教授的悉心指导下完成的。在过去的两年中，相老师一直给予我耐心的指导和鼓励，他的卓越的学术水平、不拘一格的思维方式、严谨的治学态度和勤奋的工作精神，都将使我受益终生。

感谢西安旭彤电子有限公司的王卫斌老师。

感谢实验室的同学琚翔、黎石磊、刘校伟、王加祥、徐连军、齐佩汉、万娟、张一鸣、赵楠、和周一波，感谢他们在学习和生活上对我的帮助，感谢宿舍舍友刘语乔，在我困难时伸出援助之手。在西安电子科技大学相识是我们的缘分，愿大家毕业后的生活能一切顺心。

感谢我的师弟师妹们，希望你们未来一切顺利。

感谢我的父母和弟弟，在多年的学习生涯中对我始终如一的关心和支持，你们是我前进的动力，精神的支柱。

最后，衷心感谢参加论文评审和答辩的各位专家和教授。

参考文献

- [1] 刘思久, 张礼勇. 自动测试系统与虚拟仪器原理. 开发. 应用. 北京: 电子工业出版社. 2009, 8. pp: 11-12.
- [2] 应怀樵. “云”智慧仪器与“云”智慧测试时代——数采 DAQ、虚拟仪器 VI 和“试验室网络云时代”. 电子测量与仪器学报. 2010, 6. 24(6). pp: 507-513.
- [3] 余成波, 谢东坡. 网络化测控技术与实现. 高等教育出版社. 2009, 7. pp: 97-98.
- [4] 应怀樵. 虚拟仪器的研发方向——“六高”的虚拟仪器库时代. 振动与冲击. 2006, 11. 26(6). pp: 174-175.
- [5] 秦树人. 虚拟仪器——测试仪器从硬件到软件. 振动测试与诊断. 2000. 20(1).
- [6] 刘冬冬. 虚拟仪器在测试与测控领域的优势. 仪器仪表用户. 2010. (6).
- [7] James Truchard. 虚拟仪器的未来. 测控技术. 2002. 21(9). pp: 1-2.
- [8] Portillo, E.; Cabanes, I.; Marcos, M.; Orive, D.; Sanchez, J.A. Design of a Virtual-Instrumentation System for a Machining Process. Instrumentation and Measurement, IEEE Transactions on. Dec. 2007 (6). PP: 2616-2617
- [9] 李震, 柯旭贵, 汪云翔. 虚拟仪器的发展历史、研究现状与展望. 2003, 12. 18(4). pp: 1-2.
- [10] 黎梨苗, 陆绮荣, 徐永杰. 基于 VI 技术的测控系统设计. 计算机技术与发展. 2007, 12. 17(12). pp: 231-232.
- [11] 刘金宁, 孟晨, 崔少辉等. 虚拟仪器驱动器综述. 电子技术应用. 2005. 31(7). pp: 1-4.
- [12] 林正盛, 尹爱军, 杨正益等. 虚拟仪器开发中功能组态设计. 振动测试与诊断. 2010. 30(2). pp: 193-196.
- [13] 秦树人, 张思复. 集成测试技术与虚拟式仪器——面向 21 世纪的仪器系统[J]. 中国机械工程. 1999. 19(1). pp: 77-80.
- [14] Rawnsley D.J., Hummels D.M., Segee B.E. A virtual instrument bus using network programming Instrumentation and Measurement Technology Conference, IEEE. 1997. (1). pp: 694 - 697.
- [15] Goldberg, H. What is virtual instrumentation?. Instrumentation & Measurement Magazine, IEEE. Dec 2000. pp: 10-11.
- [16] 林君. 现代科学仪器及其发展趋势. 吉林大学学报. 2002, 3. 20(1). pp: 1-6.
- [17] 郭天太, 周晓军, 朱根兴. 虚拟测试技术概念辨析. 机床与液压. 2003. 5. pp: 3-5.
- [18] 陈尚松, 李智, 雷加等. 虚拟仪器回顾与展望. 中国科技核心期刊. 2009, 12. 28(12). pp: 20-21.

- [19]李江全.LabVIEW 虚拟仪器数据采集与串口通信测控应用实战.人民邮电出版社, 2010,6.pp:25-26.
- [20]阮奇桢.我和 LabVIEW——一个 NI 工程师的十年编程经验.北京航空航天大学出版社.2009,9.pp:3-4.
- [21]S.S. Murthy, Raghu K. Mittal, Avneesh Dwivedi, G. Pavitra, and Sonika Choudhary, "Online performance monitoring and testing of electrical equipment using Virtual Instrumentation", IEEE Transactions on Power Electronics and Drive systems, 27-30 Nov. 2007, pp 1608-1612.
- [22]吴家安.数据压缩技术及应用.科学出版社.2009,1.pp:2-4.
- [23]黄满太.系统误差处理理论的方法综述.西部探矿工程.2008.(9). pp:177-178.
- [24]余剑.高精度智能测量系统中粗大误差的处理技术.测试技术学报.2003.17(3). pp:70-71.
- [25]钱政, 王中宇, 刘桂礼.测试误差分析与数据处理.北京航空航天大学出版社. 2010,9.pp:20-21.
- [26]金涛, 付驰, 秦树人.虚拟测试仪器的误差描述与修正.重庆大学学报.2005,12 28(12).pp:12-14.
- [27]钱政, 梅志刚, 罗承沐.电子式互感器中数据采集系统的实现与误差补偿.高压 电器.2004,2.40(1).pp:37-39.
- [28]宋国乡, 冯有前, 王世儒等.数值分析.西安: 西安电子科技大学出版社.2002.
- [29]王伟斌, 邱长泉.基于最小二乘曲线拟合的信号调理电路误差补偿方法.计算机 测量与控制.2009.17(11). pp:2286-2288.
- [30][美]David Salomon 著, 吴乐南等译.数据压缩原理与应用.北京: 电子工业出版 社.2003.pp:1-2.
- [31]靳鸿, 裴东兴, 祖静.无损数据压缩在数据采集系统中的应用探讨.华北工学院 学报.2003.24(1).pp:459-462.
- [32]王忠效, 姜丹.关于 Lempel-Ziv77 压缩算法及其实现的研究[J].计算机研究与发 展.1996(05).
- [33]王平.LZW 无损压缩算法的实现与研究.计算机工程.2002,7.28(7).pp:98-99.
- [34]Gyoun-Yon Cho, Dong-Ho Cho. A study on the efficient compression algorithm of the voice/data integrated multiplexer IEEE International Conference on Communications. 1995. pp:18-22.
- [35]Jin Hong, Pei Dong xing, Zu Jing. Lossless data compression in data acquisition system. Journal of North China Institute of Technology.2003.24(1). pp:69-72.
- [36]陈国顺, 宋新民, 马峻.网络化测控技术.北京: 电子工业出版社.2006,5.pp:47-60.

- [37] HAYAMA HIROSHI, UEDA KAZUHIRO. An Attempt to Evaluate Users' Adaptation by Applying LZW Compression Algorithm to Operation Logs. 2004. pp:3115.
- [38] 支超有, 唐长红. 机载数据总线技术及其应用. 北京: 国防工业出版社. 2009, 1. pp:11-12.
- [39] 王新梅, 肖国镇. 纠错码——原理与方法. 西安电子科技大学出版社. 1991, 2. pp:145-152.
- [40] 张平安. 16 位循环冗余校验码(CRC)的原理和性能分析. 山西科技. 2005(5). pp:123-125.
- [41] 冯冬芹, 金建祥, 诸建. 工业以太网关键技术初探. 信息与控制. 2003, 6. 32(3). pp: 219-223.
- [42] 郑萍, 潘世永, 立英. 基于 Ethernet 的全开放工业控制网络. 工业仪表与自动化装置. 2001(3). pp:6-9.
- [43] 陈锡辉, 张银鸿. LabVIEW 8.20 程序设计从入门到精通. 北京: 清华大学出版社. 2007. pp:327-328.
- [44] Douglas E, Comer. Internetworking With TCP/IP[M]. Vol 1:Principles, Protocols, and Architecture. Prentice Hall Publish Company. 2005. part6 pp:1-31.
- [45] 张会生, 李军, 刘永文等. 分布式并行仿真系统中的数据通信性能研究. 计算机仿真. 2005, 1. 22(1). pp: 126-127.
- [46] 饶卫进, 李振武, 白英彩. 网络性能参数的测量方法及实现. 计算机应用与软件. 2004, 4. 21(4). pp:55-56.
- [47] 过晓冰, 伍卫国, 刘敏等. 网络传输时延的测量方法. 计算机应用研究. 2002. 19(7). pp:19-21.
- [48] 李凤保, 李凌, 潘泽友. 用于测控网络的交换式 Ethernet 性能分析. 仪器仪表学报. 2005, 8. 26(8). pp:562-565.

作者在读期间的研究成果

参加的科研项目

1 项目名称：新一代飞机航电通信系统关键技术研究；项目编号：GJ0203100102；项目来源：空军重点型号背景预研；参与时间：2009.10-2010.6；主要工作：实现仿真分系统中的驾驶舱模拟接口软件，使用 delphi 开发工具进行软件编程。

2 项目名称：XX 型号飞行模拟控制系统网络技术研究；项目编号 HX0109013707；项目来源：中国第一飞机设计研究院；参与时间：2010.8-2010.11；主要工作：负责使用 LabVIEW 开发工具实现各个模块功能。

