**

## HUNAN UNIVERSITY

毕 业 论 文

|  |  |
| --- | --- |
| **论文题目** | **基于PCI的模拟振动平台** |
|  | **上位机软件系统设计** |
| 学生姓名 | 曾庆宏 |
| 学生学号 | 201407030414 |
| 专业班级 | 自动1404 |
| 学院名称 | 电气与信息工程学院 |
| 指导老师 | 江未来 |
| 学院院长 | 王耀南 |

2018 年 5月 20日

**湖 南 大 学**

**毕业论文原创性声明**

本人郑重声明：所呈交的论文是本人在老师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

学生签名： 日期：2018 年 5月 20日

**毕业论文版权使用授权书**

本毕业论文作者完全了解学校有关保留、使用论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本论文。

本论文属于

1、保密 ，在\_\_\_\_\_\_年解密后适用本授权书。

2、不保密√。

（请在以上相应方框内打“√”）

学 生签名： 日期：2018 年 5月 20日

指导教师签名： 日期：2018 年 5月 20日

基于PCI的模拟振动平台上位机软件系统设计

摘 要

本课题计划并实现一个能够真实高精度再现实际环境的模拟振动平台，通过模拟真实环境中的海浪、地震、滑雪、飞行等,可以促进VR（虚拟现实）技术在各种结构测试中的应用。随着车辆工程、土木工程、海洋工程的迅速发展，模拟振动平台的控制精度和其计算机上位机软件系统也需要达到更高的水平。通过对计算机上位机软件系统的改善，可以很好地控制、监控和调试模拟振动平台。设计一个友好的人机交互界面，可以大大降低模拟振动平台开发难度，并且能保证其能处于安全、平稳的运行状态。目前，分布式组网的自动化控制系统已经被普遍运用，本文采用以太网组网的通信模式，基于TCP/IP协议设计开发了一套用于与模拟振动平台系统下位机（控制器）交互、调试的上位机软件系统。

本文首先根据控制精度、控制需求分析了整个系统的结构，之后简单介绍模拟振动平台系统的机械结构、算法设计、硬件电路、以及下位机的软件系统。对于上位机软件系统，本文采用跨平台开发工具Qt进行开发，使其不仅仅可以在Windows系统下运作，还能在Linux、Mac等系统下工作。

设计过程首先分析了控制精度等需求，根据以上决定系统的总体设计框图，然后对于总体框图的各部分子功能：模拟振动平台系统参数设置、数据传输与处理、以太网及USB通讯、三参量数据可视化以及保存，分析了其设计与实现。其中，重点介绍了动态绘图中的生产者-消费者矛盾以及双缓冲解决办法、以及基于TCP/IP应用层的通信协议制定。最后我们对振动平台上位机软件管理系统各个功能进行了调试和详细测试。

关键词**：**模拟振动平台;用户界面设计;双缓冲机制;生产者-消费者矛盾;通信协议

**Upper computer system of single degree of freedom vibration platform based on PCI**

**ABSTRACT**

This topic plan to design a vibration platform, a high precision actual simulation can be reproduced on this platform.By simulating the real environment of sea waves , earthquakes, ski, flight, etc. With the rapid development of vehicle engineering ,civil engineering and marine engineering , the higher level of the control accuracy and personal computer software system is required .The vibration platform can be controlled、monitored and debugged very well, through improving the upper computer software .Designing a friendly user interface can not only greatly reduce the difficulty of the development of the vibration platform,but can ensure vibration platform in a safe and stable operation state. At present, distributed network automation control system has been widely used, This paper adopt the communication mode of Ethernet networking ,design a upper computer software system for interaction and debugging with the lower computer(controller) of the vibration platform based on TCP/IP protocol.

At first, this paper analysis the whole system structure according to the requirements of control accuracy, after this, the paper briefly introduces the mechanical structure of the simulated vibration platform system, algorithm design, hardware circuit and the software of lower computer system.It can run not only under Window ,but under the Linux and Mac as well.

For upper software system details, first of all ,we introduces the design framework and user interface, then for the functions of upper computer system: Setting parameter of vibration platform system, data transmission and processing, Ethernet and USB communication, three-parameter data visualization and Saving., analyses the design and implementation of that .Among them, the author focuses on the producer-consumer contradiction in dynamic drawing and the double buffering solution, and the communication protocol based on TCP/IP application layer.Finally, the software system modules are tested in detail.

**Keywords**: **Simulated vibration platform;UI design;producer-consumer conflict; double buffering; communicating protocol**

目录

[摘 要 I](#_Toc515382316)

[ABSTRACT II](#_Toc515382317)

[插图索引 V](#_Toc515382318)

[第一章 绪论 1](#_Toc515382319)

[1.1 课题背景及研究意义 1](#_Toc515382320)

[1.2 国内外研究现状 1](#_Toc515382321)

[1.3 本设计的主要内容和章节安排 3](#_Toc515382322)

[第二章 模拟振动平台系统总体介绍 5](#_Toc515382323)

[2.1 总体框架 5](#_Toc515382324)

[2.2 机械设计 8](#_Toc515382325)

[2.3 算法研究 8](#_Toc515382326)

[2.4 硬件系统 9](#_Toc515382327)

[2.5 软件系统 9](#_Toc515382328)

[第三章 上位机软件系统设计 12](#_Toc515382329)

[3.1 软件系统需求分析 12](#_Toc515382330)

[3.2 上位机软件系统整体架构设计 13](#_Toc515382331)

[3.2.1 基本功能框图 13](#_Toc515382332)

[3.2.2 总体架构设计 14](#_Toc515382333)

[3.3 软件开发平台简介 15](#_Toc515382334)

[3.4通信模块总体设计 15](#_Toc515382335)

[3.4.1 通信框架概述 15](#_Toc515382336)

[3.4.2 通信过程 16](#_Toc515382337)

[3.4.3 应用层帧结构 17](#_Toc515382338)

[第四章 系统实现 22](#_Toc515382339)

[4.1 系统管理模块 22](#_Toc515382340)

[4.2 系统通信模块 26](#_Toc515382341)

[4.2.1 以太网TCP/IP通信基本参数的初始化 26](#_Toc515382342)

[4.2.2上位机通信设计 29](#_Toc515382343)

[4.3 数据处理模块 31](#_Toc515382344)

[4.3.1 默认配置信息的保存 31](#_Toc515382345)

[4.3.2 试验参数和控制 34](#_Toc515382346)

[4.3.3 试验数据软件的存储和试验报告 35](#_Toc515382347)

[4.4 数据显示模块 36](#_Toc515382348)

[4.4.1 三参量数据图表分析 36](#_Toc515382349)

[4.4.2 生产者-消费者矛盾 38](#_Toc515382350)

[第五章 系统调试及测试 42](#_Toc515382351)

[5.1 上位机主界面测试 42](#_Toc515382352)

[5.1.1 试验信息界面测试 42](#_Toc515382353)

[5.1.2 试验状态界面测试 43](#_Toc515382354)

[5.1.3状态栏测试 43](#_Toc515382355)

[5.2 上位机子界面测试 44](#_Toc515382356)

[5.2.1新建试验界面 44](#_Toc515382357)

[5.2.2用户登录界面 45](#_Toc515382358)

[5.2.3 通道参数设置界面 46](#_Toc515382359)

[5.2.4 控制参数设置界面 47](#_Toc515382360)

[5.3 上位机实时曲线界面测试 48](#_Toc515382361)

[5.4 上位机通讯功能测试 49](#_Toc515382362)

[第六章 总结与展望 50](#_Toc515382363)

[6.1总结 50](#_Toc515382364)

[6.2 展望 50](#_Toc515382365)

[参考文献 52](#_Toc515382366)

[致谢 55](#_Toc515382367)

插图索引

图 1‑1 电动模拟振动平台 3

图 1‑2 六自由度液动模拟振动平台 3

图 2‑1 模拟振动平台总体框架 5

图 2‑2 控制系统框图 6

图 2‑3迭代控制原理图 8

图2‑4 硬件系统原理图 9

图2‑5 模拟振动平台上位机软件系统总体布局 10

图 3‑1 振动平台上位机软件功能图 13

图 3‑2 振动平台上位机程序流程图 14

图 3‑3 振动平台上位机软件总体框图 14

图 3‑4 TCP/IP组网结构图 16

图 3‑5 TCP三次握手连接示意图 16

图 3‑6 上下位机通信示意图 17

图 4‑1右键浮动菜单 23

图 4‑2系统管理流程图 25

图 4‑3振动台上位机软件系统主界面 26

图 4‑4振动台上位机C-S模式通信概念图 29

图 4‑5并发服务器结构图 29

图 4‑6保存配置菜单 33

图 4‑7配置文件 33

图 4‑8配置文件加载原理框图 33

图 4‑9试验控制按钮介绍 34

图 4‑10实时曲线显示界面 37

图 4‑11生产者-消费者模式图 38

图 4‑12生产者-消费者算法参数 38

图 4‑13 状态 39

图 4‑14双缓冲区机制 39

图 4‑15实时曲线绘制流程图 41

图 5‑1振动平台上位机管理系统主界面 42

图 5‑2试验信息显示界面 42

图 5‑3试验状态显示界面 43

图 5‑4状态栏显示界面 44

图 5‑5试验选择界面 45

图 5‑6用户登录界面 45

图 5‑7用户登录菜单 46

图 5‑8设置菜单 46

图 5‑9通道参数界面 47

图 5‑10控制参数设置界面 47

图 5‑11控制参数设置菜单 48

图 5‑12三参量曲线实时显示界面以及全局状态栏 48

图 5‑13 WIRESHARK网络通信抓包 49

1. 绪论

1.1 课题背景及研究意义

近些年来，VR（虚拟现实）技术不断完善高速发展，其不仅可以用于用户的日常娱乐，而且可以用于在工业生产，基础设施建设中模拟再现一些真实极端环境，从而对各种装置的工程性能进行检验和评判[1]。模拟振动平台通过高精度控制算法，可以真实再现地震、道路状况、海浪、以及一些飞行真实场景，有助于实现房屋结构抗震理论研究和实际设计，亦可以实现机动车零部件和整车可靠性实验，对轮船和海洋工程装备制造也有莫大帮助。模拟振动平台高精度实时再现实际环境技术，目前国内外已有公司开始相关研究，但是成品大多来自国外进口。若能将此技术国产化，不仅可以用于我国虚拟现实中实现更深的浸入感，对于国家的国计民生、工业发展也有着重要的意义，有助于推动我国产业的升级，也能为我国在模拟振动平台的研究积累经验[2]。

1.2 国内外研究现状

模拟振动台是舰船、飞机、海工装备在研发和使用过程中最重要的检验和模拟使用设备，也是汽车底盘、车身、车载电子设备的实验室研究中最重要的一环，同时，在土木工程领域，是建筑抗震性能研究中最重要的设备，但是关键技术一直被美国的MTS公司、英国的Servotest公司垄断，国家每年都要花费十几亿美元外汇从国外购买。由于上述设备集成了机械、电子、液压、传感器、计算机软硬件等多门类的关键技术，使用和维护都比较复杂，从国外购买的设备的使用和售后都非常不便，国防和相关关键部门的设备又不能请美国等西方国家的公司去现场维护，这一切，都极大制约了我国在相关领域的研发进度[3][4]。

如同济大学土木学院国家重点实验室，从美国进口的六自由度模拟振动平台，从验收到开始使用，花了好几年时间，而且购买设备的费用十分昂贵，每年还要抽出额外的经费对振动台进行维护。

目前的美国MTS公司和英国Servotest公司已经垄断了全球的高精度模拟振动平台的技术和市场，据统计，美国MTS公司每年在中国的盈利都超过1亿美元。MTS公司的普通振动平台单价可高于700万美元。相对于美国MTS公司，国内开发的振动平台售价只有其三分之一。然而我国目前在振动平台的研究进度还比较落后，整体开发以仿造国外技术为主，并不能达到高精度的要求[5]。随着科技的发展，各个科研机构、车辆、船舶市场对高精度模拟振动平台需求越来越大。但由于进口费用，维护和使用费用昂贵，制约了这些方向科研发展。

为了推动土木工程、海工装备等领域的研究开发，高精度模拟振动平台作为波形发生的关键设备，国内迫切需要该产品能实现国产化，如模拟振动平台用于道路振动模拟机，国内至少有3个亿的市场[6]。

高精度模拟振动平台的关键技术涉及到机械传动、高精度控制算法研究、计算机控制系统搭建、信号处理、系统工程等等方面，是一个非线性、强耦合的控制系统。目前，我国技术在机械传动等关键方面已经有了突破性进展。

近年来，国内投入到电液伺服模拟平台的专业厂家，研究机构也在逐步增多，中科院力学所、哈尔滨工业大学、华中科技大学、哈尔滨工程大学和浙江大学液压所，都做过类似的科研,发表过相关论文，不过目前市场上只有小型的电机驱动的模拟振动平台出售，一般只用于游戏、科普等领域。

近年来，随着智能控制算法、机械制造精度的不断发展提高，出现了诸如三参量控制、迭代控制、自适应逆控制等先进的控制算法，可以很好地提高模拟振动平台的动态特性，改善系统的频率响应。同时对机械结构上的改进，也可以改善整个系统大惯性、大滞后的问题，从而可以提高控制精度[8]。

通过继续的深入研究，提高振动台控制精度以及增加自由度，可以模拟再现实际场景，这项技术不仅仅可以用于虚拟现实，也可以对国计民生、工业发展有着积极作用[9]，比如应用至地震模拟、道路模拟和海浪模拟中，而目前振动台多由国外进口，关键技术尚未国产化，价格昂贵，一旦研究实现其技术国产化，有较好的市场前景，同时对于社会的经济效益非常大。同时可以为我国高精度模拟振动平台的研究积累经验，推动后续的科研进展[10]。



**图 1‑1 电动模拟振动平台**



**图 1‑2 六自由度模拟振动平台**

1.3 本设计的主要内容和章节安排

本课题研究设计一套采用以太网组网模式，基于TCP/IP协议，以客户端-服务器模式进行连接（下称C-S模式）,有良好人机交互界面的上位机软件系统。本文分析了单自由度模拟振动平台的控制精度，以及控制要求，从而对整个上位机软件系统进行功能模块划分，重点在于三参量实时动态曲线绘制、基于TCP/IP协议的应用层通信协议制定，通过上述步骤实现上位机人机交互界面和后台逻辑处理系统。本文章节安排如下：

第一章介绍了振动平台的研究现状背景，以及到目前为止，国内国外对于振动平台的研究进展，国内对振动平台研究方面的不足，以及本文的章节安排。

第二章分析了振动平台控制系统的控制精度以及要求，概括说明模拟振动平台的整体结构。然后，分别介绍了振动平台的机械结构设计、算法研究、硬件电路、以及软件系统[11]。

第三章是上位机软件系统设计部分，通过对整个系统的控制需求进行分析，根据不同功能，系统由以下四个部分组成——系统参数设置模块、数据传输模块、通讯模块、数据可视化分析界面[4][8]。然后介绍了整个系统的软件开发平台，接下来通过分析振动台系统通信需求，在基于以太网TCP/IP通信协议的基础上，按照规定严格制定应用层的通信协议，并分析其具体实现过程。

第四章为系统实现，首先介绍了系统管理模块的功能，对整个系统主界面的设计，菜单栏设置进行了分析讲解。之后介绍了上位机基于TCP/IP协议的应用层通信制定和详细实现过程。然后介绍了数据处理模块，重点分析了配置文件的存储和导入，以及一些实验参数的控制调节界面。最后通过分析生产者-消费者矛盾，提出双缓冲队列机制的解决方法，解决问题，完成数据显示模块的设计，能实现三参量曲线的动态显示。

第五章是介绍振动平台上位机管理软件系统的人机交互界面，以及完成上位机系统功能的测试以及调试，并分析上位机中各个控件、子界面的具体功能。

第六章为总结展望章节，对全文提出、分析的问题进行总结，并且讨论本文涉及知识点，展望研究内容的未来发展。

1. 模拟振动平台系统总体介绍

2.1 总体框架

本项目研究的主要内容为高精度大型模拟振动平台的随机波形控制，而研究内容主要包括四个部分：第一部分分析了机械系统，探讨如何最大限度减小振动台的非线性，研究不同建模对振动台进行控制的影响；第二部分研究了控制算法，主要分析了传统算法的不足，以及介绍了三种先进的智能控制算法，采用这些算法可以改善系统动态频率响应，提高控制精度；第三部分从硬件电路的角度出发，通过全数字工作方式的控制器，实现多通道模拟振动台的随机波实时波形复现。第四部分从振动平台控制上位机软件出发，通过多功能的上位机控制系统，可以很有效的实现操作指令的下达、数据图像的实时绘制，试验状态的实时监控，从而达到所有数据可视化的效果[12]。

模拟振动平台主要由振动台体、振动台台面机械系统、位移，加速度，速度等各类传感器、电源、和控制系统组成[10]。模拟振动平台利用传感器采集三参量控制信号，并将其反馈回控制系统，通过不同的控制算法，使得振动平台可以高精度跟踪随机输入波形，从而完美复现波形。系统框图如图2-1所示：

**图 2‑1 模拟振动平台总体框架**

参考

信号

控制

系统

驱动

信号

电动

激振系统

台体及台面机械系统

试件夹具

传感器

控制

信号

上位机软件

调节

信号

整个系统控制框图2-2如下所示:

上位机给定信号

前级调理控制电路

电动激振器

振动台台体

控制量反馈

+

输出

**图 2‑2 控制系统框图**

振动台主要技术指标和预期目标如下：

（1）振动台电动激振器  
⚫ 最大激振力：±20kN  
⚫ 最大行程：±100mm  
⚫ 最大速度： 500mm/s  
⚫ 最高频响： 70Hz  
（2）台面机械系统  
⚫ 最大正弦推力：50kN  
⚫ 最大加速度:2g  
⚫ 最大速度:500mm/s  
⚫ 最大位移:200mm  
⚫ 最大负载： 3000kg  
⚫ 台面尺寸： 800\*1200mm  
⚫ 最大力矩:3kN·m  
⚫ 频率范围:0--50Hz

（3）整体系统

⚫ 加速度最大相对误差≤ 10%  
⚫ 位移段波形失真度≤ 5%，加速度段波形失真度≤25%  
⚫ 加速度横向比≤ 25%  
⚫ 时域复现精度：时域相关系数≥ 90%,最大峰值误差≤ 10%

模拟振动平台运动分析如下：

整个振动台最高频率响应为70Hz，在这种情况下，振动平台控制系统高频噪声可以忽略。单精度模拟振动平台动力方程如下所示[11]：

(1)

上述公式中：为系统质量； 为振动台电动激振器的转换系数； 为系统阻尼系数比。为电动激振器产生的动力，为振动平台的水平位移。

对电动伺服激振器进行建模，可以得到其表达式为：

由此，可以得出，振动平台输出位移和其输入电压之间的关系为[12] [14]：

式中：

由此上分析，可以知道,通过三参量反馈、迭代控制等等算法，可以增加系统的阻尼系数，大大提高系统的动态性能，达到更优的控制指标。

2.2 机械设计

机械与动力部分的设计可以参考市场上成熟的进口产品的设计原则，需要解决的是高频电动伺服动作器的国产化，主要的技术难点存在以下三个**[13]**：

* 提高电动伺服动作器的最高响应频率；
* 优化机械结构，减小系统摩擦带来的非线性影响；
* 实现高频响应电动伺服动作器的国产化；

2.3 算法研究

由于模拟振动平台系统存在强耦合，高滞后，采用传统的控制算法效果并不能很好地满足跟踪精度，本项目拟采用三参量控制[14]、迭代学习控制[15]以及自适应你控制算法[16][17]来改善控制效果，下面简单介绍迭代学习控制算法:

**在振动台的控制过程中，由于本系统存在着很大的非线性因素，系统控制过程中的响应和期望有较大稳态误差[12]。因此我们拟在伺服控制系统的外环再增加基于迭代学习控制算法的外闭环构成双闭环控制系统，对输入信号迭代运算进行补偿，从而可以减小系统跟踪误差，使系统的输出能够高精度的跟踪期望波形[15]。**

**其原理如图2-3所示：**

控制存储记忆

被控对象

FFT

FFT

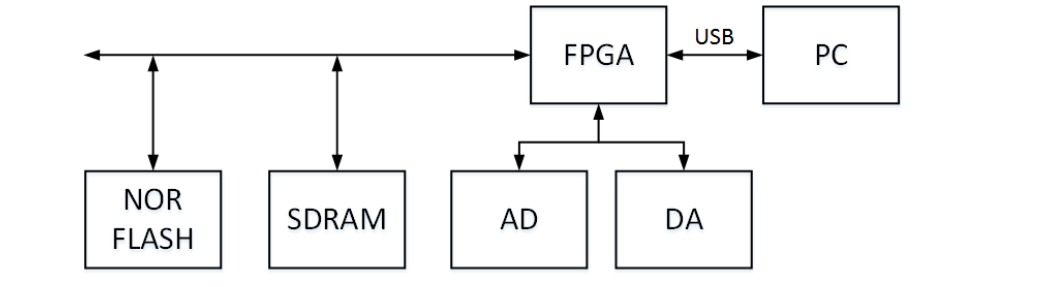
IFT

期望轨迹存储器

**图 2‑3迭代控制原理图**

2.4 硬件系统

模拟振动平台以FPGA作为数字主控芯片，硬件主体结构图如图所示：



**图2‑4** **硬件系统原理图**

1. FPGA

模拟振动平台系统的主控制器为FPGA芯片，其作为可编程逻辑控制芯片，可以实现系统的 AD、 DA 和 USB、以太网转换接口的需要，同时可以高速进行大量并行计算，为数据处理和信号传输提供硬件支持。

1. 信号的输入与输出采集卡

信号的输入输出采集卡采用研华的PCI-1767，它带有一个250KS/s16位A/D转换器.以及多个A/D,D/A转换器.此外PCI-1716可以接入多个传感器，PCI-1716可以对其进行统一管理。通过PCI采集传感器信号，经过调理之后，输入FPGA中进行算法运算，从而达到控制效果。

1. 硬件部分主要工作原理

控制系统中的主控制器实现了整个控制系统的算法，以及各个部件的同步与资源调配，并实现了与以太网口的通讯。系统中具有存储记忆芯片，可以保存下载的代买，FPGA不仅可以从电脑下载代码，从而作为分布式控制器[4]。也能直接从主存储器中读取代码，从而作为Embedded控制器使用。

2.5 软件系统

* 1. 上位机

模拟振动平台上位机管理软件系统，可以极大地方便整个控制系统的试验与调试，其提供简单的UI界面，以实现与下位机控制器的实时交互。 上位机软件具有下面功能，以实现对模拟振动平台的控制管理以及状态监控即：

模拟振动平台系统参数设置、数据传输与处理、以太网及USB通讯、三参量数据可视化以及保存。

模拟振动平台系统参数设置:振动平台的PID等控制参数设置、以及输入/输出通道的硬件校准、控制策略的选择等等。

数据传输与处理：上位机软件系统通过以太网与控制器发送消息，之后进行TCP/IP应用层通讯协议的处理，获得解除封装的数据报之后，再进行数据的分类，错误警告处理等。

以太网通信：采用TCP/IP通讯协议和从机互相收发数据，进行C-S模式通信。

三参量数据可视化以及保存：实现试验中三参量数据的可视化功能，以及数据图表保存功能，以便离线监控分析系统状态。整个上位机软件系统布局如图2-5:

**图 2‑5 模拟振动平台上位机软件系统总体布局**

主窗体

三参量曲线显示界面

菜单栏

工具栏

振动台警告信息显示

振动台状态信息监控

状态栏

按钮事件

子界面

以太网

消息传递

* 1. 下位机

下位机为FPGA全数字控制器的软件系统，实现对模拟振动台平台的控制算法，并且通过与PCI-1716通信，读取、传输信号，实现对整个系统的控制调配。按照功能，下位机可以分为：信号管理模块、算法控制模块、系统调配模块[4]。

系统调配模块搭建起整个系统，负责对其余两个模块进行调配和对系统的总体调理[19]。

信号管理模块负责与PCI-1716通信，通过读取不同的传感器，为算法控制模块提供运算数据，并且其负责与上位机的通信，根据上位机的不同指令做出反应，例如上传控制参数、报告当前状态等。

算法模块是整个下位机软件的核心模块，它通过读取信号管理模块提供的传感器信号，运行预先设计好的核心算法，输出整个控制系统的控制信号，以实现振动台高精度复现期望波形的功能。

1. 上位机软件系统设计

3.1 软件系统需求分析

振动平台需要通过上位机软件系统，对整个控制系统进行控制管理，参数调节，和达到控制状态数据可视化。通过对第二章的振动平台系统各部分的分析，我们得到振动台上位机软件系统需要实现以下功能[23]：

系统与用户的交互，实现试验参数设置、操作命令发送、控制过程监控、数据显示与存储、实验结果产生等功能。

振动平台上位机软件可以根据具体实验精度要求、具体波形选择等，配置不同的实验参数和实验环境，所以其应该具有控制功能和辅助功能。具体功能如下：

1. **控制功能**

* **随机系列**
* **随机试验**
* **随机地震波冲击试验**
* **正弦加随机试验**
* **随机加随机试验**
* **…**
* **冲击系列**
* **典型冲击试验**
* **冲击响应谱试验**
* **瞬态冲击试验**
* **…**
* **正弦系列**
* **正弦试验**
* **谐振搜索与驻留试验**
* **…**

1. **辅助功能**

* 系统校准
* 离线浏览
* 波形编辑器
* 信号报警
* 波形显示
* …

3.2 上位机软件系统整体架构设计

3.2.1 基本功能框图

通过上一节对系统需求的分析，可以知道振动平台的上位机软件管理系统需要实现以下四个基本功能：

* 读取系统的基本初始化文件，设置本次试验参数等等。
* 本系统通过以太网TCP/IP，对控制器发出控制指令，询问指令，调参指令等不同命令。
* 下位机将振动平台的状态实时上传到上位机，并保存至文件中。
* 将振动平台的实时状态以动态曲线绘制成时间表图，根据波形进行试验状态分析。

基于以上的四个功能，可以设计出模拟振动平台的总体框图[24]：模拟振动台基本信息界面、控制参数信息界面、数据图表可视化界面以及系统设置界面。

整体框图3-1如下所示：

**图 3‑1振动平台上位机软件功能图**

整个系统程序流程图如图3-2所示：

**图 3‑2振动平台上位机程序流程图**

开始

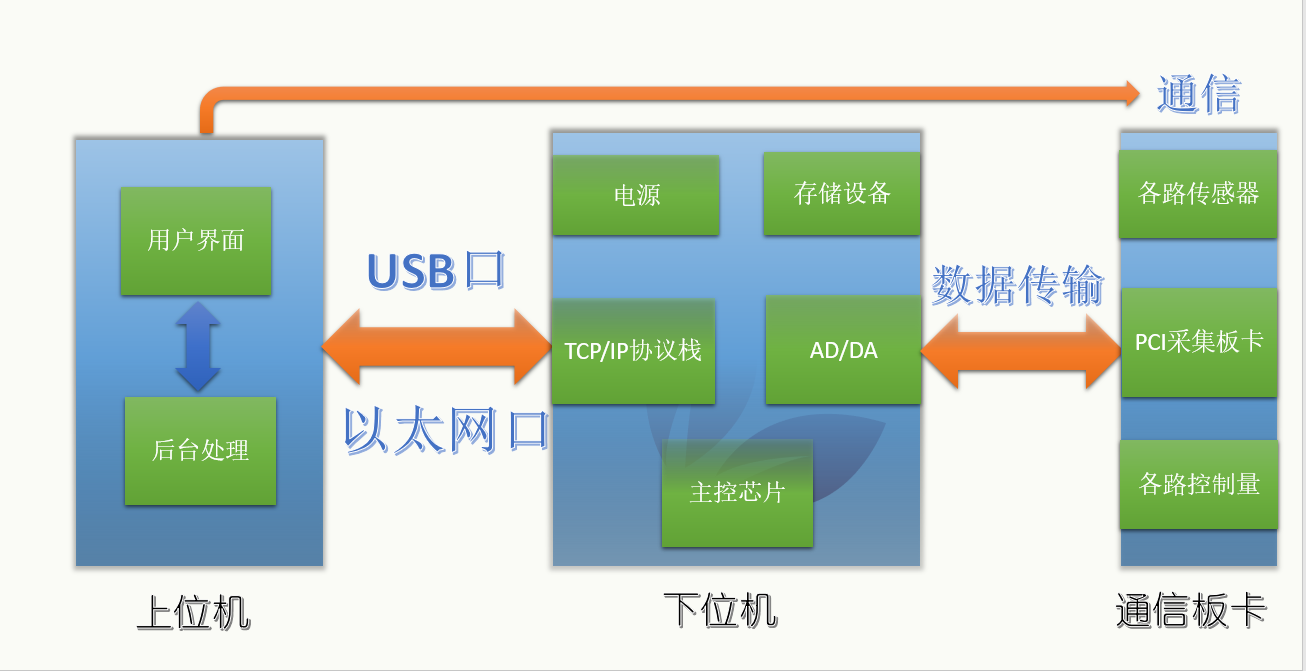
程序环境加载，初始化主窗体

初始化所有控件信号槽函数

初始化TCP服务器，开放端口进行帧听

读入配置文件，加载默认参数，等待事件

3.2.2 总体架构设计

**** 宏观上来看，振动平台软件系统的总体框架由控制器下位机和管理软件上位机组成，框架如图3-3所示:

**图 3‑3振动平台上位机软件总体框图**

由以上软件总体框图可以看出，振动平台上位机和控制器下位机的通信方式为以太网用TCP/IP，上位机软件分为前端用户人机交互界面以及后端的逻辑信息处理，并且可以同时控制下位机控制器以及数据通信板卡PCI-1716。整个系统在Qt5开发平台中采用C++进行开发，具有良好的可移植特性[25]。

3.3 软件开发平台简介

本文在Python, C#, C++等面向对象编程语言中，选择了运用C++语言进行上位机软件系统的开发。基于C++语言的设计在Qt5开发平台上，利用Qt5固有的API函数，可以灵活高效的设计出友好的人机交互界面。对于开发振动平台上位机这种桌面应用程序，C++语言具有以下特点：C++为编译型语言，运行效率高；C++的多态和封装，使得整个系统具有很高的可移植性。

Qt是的桌面应用IDE集成工具，具有很良好的跨平台特性[21]。Qt提供了自动补全，语法高亮，前端、后端设计分离等等方便地特性，而且Qt有着丰富的API函数，拥有250个以上已经封装好的C++的类，还提供了基于模板的类。Qt在网上也有着大量的资料和开发文档，上手容易。Qt不仅仅支持2D图形绘制，也能渲染3D图形，为以后对振动台进行3D建模、实时监测也提供了便利。

3.4通信模块总体设计

3.4.1 通信框架概述

设计模拟振动平台上位机软件系统通信协议，以保证上位机和控制器不仅能无差错的传输数据，而且能分辨数据不同的含义以及请求，是整个软件系统的重点部分。本节将概述上位机和控制器的通信过程，并且分析其原理及其实现，已达到上位机对整个系统实时监控的效果[26]。

模拟振动平台上位机软件系统需要的监控多个控制器状态，同时不同的控制器均需要将数据上传至上位机进行实时显示，这就涉及到多机通讯问题。在工程设计中，利用基于以太网的组网系统已经成为控制系统的标准配置，本文也将采用基于以太网的TCP/IP协议进行通讯[27]。

TCP/IP协议可以实现Client-Server通信模式，其中上位机作为Server，控制机作为Client，对服务器发起连接，从而进行端对端的通信。服务器只需要开放一个端口，客户端就能通过服务器IP地址的端口号，建立起与服务器之间的连接，并不受限于一个主节点多个子节点的通信模式。而且连入以太网的各个客户端与服务器设备各自独立运行，数据收发互不影响，而且通信质量稳定，不容易受到外界干扰。

故本文采用以太网组网方式，组网架构为C-S架构,结构图如图3-4所示：

以太网总线

TCP服务器

TCP服务器

TCP客户端

TCP客户端

TCP服务器

**图 3‑4 TCP/IP组网结构图**

3.4.2 通信过程

采用以太网TCP/IP通讯方式，在上位机和下位机通讯过程中，一个典型的事物请求与处理流程为：

1. 客户端通过服务器的IP地址以及端口号与服务器连接；
2. 客户端构造带有应用层协议的TCP请求，其中包含了应用层的帧头等数据；

**图 3‑5 TCP三次握手连接示意图**

主机1

时间

主机2

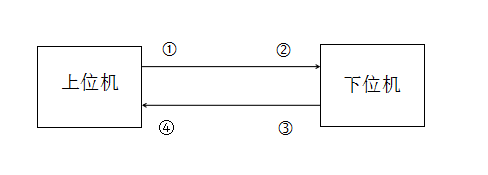
CONNECT REQUEST（SEQ=x）

ACCEPT（SEQ=y , ACK=x+1）

DATA（SEQ=x , ACK=y）

1. 客户端发送TCP请求到服务器，并等待服务器响应；
2. 服务器接收到客户端发送来的TCP数据包，通过制定好的应用层协议解析请求并进行处理，若需要回应，则服务器根据应用层协议构造回复数据报对客户端信息进行回复；
3. 客户端收到服务器回复后，一次事务处理完成。若连接资源有限，此时客户端应主动关闭到服务器的连接，方便服务器为其他客户端提供服务。

**图 3‑6上下位机通信示意图**



具体说明如下：

（1）用户在上位机界面输入控制参数、或者点击按钮等之后。上位机首先判断事件类型，区分该次数据为命令帧、数据帧还是警告帧。之后在数据的头部、尾部加入帧头、帧尾和数据长度、校验码等等字符后，在应用层对整个指令进行一次封装，之后将本次封装好的数据通过以太网TCP/IP传输至下位机中。

（2）下位机采用“双缓冲机制”来接受数据，以解决通讯速率不匹配等问题，提高整个系统的数据通信效率。

（3）下位机接收到数据之后，首先对数据进行解封装，去除数据帧的帧头、帧尾，并对校验码进行检查，一旦发现错误，应立即丢弃本次数据，并构造一个错误帧返回上位机,说明本次通信出现错误。

（4）上位机接收到数据之后，处理过程如同下位机，应首先对数据帧进行校验，若通过数据校验，则通过功能字符判断该数据帧的具体功能（功能，绘图，错误），最后根据不同的功能帧而进入不同的逻辑处理流程，完成规定指令。

3.4.3 应用层帧结构

TCP通信是端对端，面向字节流的通信，所以为了能使上位机和控制器相互确定所发当前数据的含义，需要制定应用层的通信协议。为了保证通信过程的无差错、高效率传输，我们从传输、校验等角度出发，依据TCP/IP应用层协议规范，定义以下严格的数据帧结构以及传输协议[27]，如表3-5所示。

表3-1 帧结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帧头 | LEN | FUN | DATA | CHECK |
| 2字节 | 2字节 | 1字节 | N字节 | 2字节 |

具体说明如下：

* 收码和发码格式均设为十六进制格式。
* 控制器发送自定义数据，格式为：起始数据帧头 + 功能字 + 数据长度 +校验和
* 数据长度为所发送数据的长度(字节数)（不包括起始数据帧头、功能码）数据长度标志还有校验和采用TCP/IP中方式，是整个数据帧的二进制反码和。
* 网络通信中数据采用大端模式传输,传输中的数据类型没有限制，unsigned int ，long，int等均可。
* 下位机传输给上位机显示绘图的帧，（帧格式：帧头+功能字+数据长度+加速度数据+速度数据+位移数据+ 0x0000 + 0x0000+校验和，共32字节，加速度数据/ 速度数据/位移数据为int格式，LEN = 0x10,其中将数据乘以一个系数化成整数后再进行传输（例如位移0.02\*100=2）
* 下位机发送PID数据给上位机的格式为：帧头+功能字+数据长度 + PID数据 + 0x00+0x00 +校验和PID数据为acc\_p,acc\_i,acc\_d,vel\_p..i..d,pos\_p,,i,,d，共9个uint16(unsigned short int)型数据也是将数据乘以系数后传输 (LEN=0x14)。
* 实验参数帧的格式为: 帧头+功能字+数据长度+波形+振幅+频率 +0x00+0x00+校验和,其中数据为波形，振幅，频率3个u16数据，加上两字节填充数据,LEN=0x08
* 限制参数帧的格式为：帧头+功能字+数据长 +限幅+限频+校验和其中数据为 限幅、限频2个u16数据，LEN=0x04
* 所有PID参数扩大之后再进行传输

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 控制器->上位机 | | | | | | |
| 帧功能 | 帧头 | 功能字 | 长度 | 数据 | 校验 | 说明 |
| 状态帧  STATUS | 0xAAAA | 0x01 | LEN | int ACC\_DATA  int VEL\_DATA  int POS\_DATA | SUM | 向上位机汇报当前状态，上位机接受并显示 |
| PID帧 | 0xAAAA | 0x02 | LEN | Short int ACC\_P  Short int ACC\_I  Short int ACC\_D  Short int VEL\_P  Short int VEL\_I  Short int VEL\_D  Short int POS\_P  Short int POS\_I  Short int POD\_d | SUM | 传输当前PID参数，所有PID参数扩大1000倍成为整数后传输 |
| 实验参数帧 | 0xAAAA | 0x03 | LEN | U16 waveform  U16 amplitude  U16 frequency | SUM | 汇报当前实验参数 |
| 限制参数帧 | 0xAAAA | 0x04 | LEN | U16 amp\_cons  U16 fre\_cons | SUM | 汇报当前限制参数 |
| 警告信息帧 | 0xAAAA | 0x05 | LEN | U16 warning | SUM | 向上位机发起警告（例如超过量程、连接断开） |
| 错误帧 | 0xAAAA | 0x06 | LEN | 错误信息 | SUM | 发生错误时，发送当前错误 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 上位机->控制器 | | | | | | |
| 帧功能 | 帧头 | 功能字 | 长度 | 数据 | 校验 | 说明 |
| 控制帧  STATUS | 0xAAAA | 0x01 | LEN | 控制命令   1. 启动停止 2. 传输数据 3. 错误处理 4. 请求返回数据 | SUM | 表明上位机接下来要发送的动作指令，控制器根据相应指令做出反应 |
| PID帧 | 0xAAAA | 0x02 | LEN | Short int ACC\_P  Short int ACC\_I  Short int ACC\_D  Short int VEL\_P  Short int VEL\_I  Short int VEL\_D  Short int POS\_P  Short int POS\_I  Short int POD\_d | SUM | 上位机传输给控制器的三参量PID参数 |
| 实验参数帧 | 0xAAAA | 0x03 | LEN | U16 waveform  U16 amplitude  U16 frequency | SUM | 上位机设定的实验参数传给控制器 |
| 实验数据帧 | 0xAAAA | 0x04 | LEN | U16  position[array] | SUM | 上位机读取波形文件，并将其数据传入至控制器中 |
| 限制参数帧 | 0xAAAA | 0x05 | LEN | U16 amp\_cons  U16 fre\_cons | SUM | 上位机设定的限制参数 |
| 紧急停止帧 | 0xAAAA | 0x06 | LEN | U16 stop | SUM | 接受到错误后，要求振动台立即停止 |

整个应用层的通讯数据包应如下所示：

Struct data\_pack

{

Char head;

Char function;

Int length;

U8 \* data //指向数据的指针

U16 check\_sum;

U16 warning\_no;

U16 error\_no;

}

1. 系统实现

模拟振动平台上位机软件系统是在Qt 5.9平台上，结合以太网组网，以及TCP/IP协议，和C-S模型实现的。在图形用户界面设计中，采用C++编程语言，可以高效地开发交互性强的应用程序,并且具有很好地跨平台特性[21][25]。

4.1 系统管理模块

系统管理模块分为前端设计和后端设计，前端设计主要包括UI人机交互界面，采用Qt中的Qt designer进行前端设计和布局。通过创建一个Qt中MainWindow Widget类，从而创建出整个上位机系统的主界面，在分析出系统管理模块的需求之后，可以在MainWindow中放置如toolbar、statusbar、pushbutton等等控件，从而搭建出满足功能需求的界面。

后端设计则涉及到处理各种按钮的信号逻辑，以及重写他们的功能函数。Qt中使用信号槽机制，通过connect函数，可以检测某个事件发生后，连接的槽函数会自动被回调，例如菜单栏中退出试验的信号槽代码如下，其中QPushButton::trigged为信号，On\_Trigged\_Button\_Exit为用户实现的槽函数：

QObject::connect(&Exit\_Btn,&QPushButton::trigged,On\_Trigged\_Button\_Exit);

C++语言具有面向对象设计的特点，其继承、封装、多态可以很好的实现UI界面中的各种功能。

本上位机软件系统软件进入试验功能主界面后，系统管理模块主要包括七个部分：

* 菜单栏：

列出所有可供用户操作的命令和设置，有文件菜单、设置菜单、视图菜单、命令菜单、窗口菜单、显示菜单、存储菜单、报告菜单、帮助菜单等等。

* 工具栏：

工具栏列出用户将频繁使用的操作、设置和命令的快捷按钮。包括有文件管理工具栏、参数设置工具栏、显示操作工具栏和窗口工具栏等。

* 信号面板

信号面板位于功能主界面的左侧，默认是隐藏的，单击“信号”按钮，信号面板就会显示出来，信号面板中列出本试验的所有信号，包括系统信号和用户定义信号。

* 窗口显示

试验状态显示了试验运行命令、部分试验控制命令，显示本次试验的目标、控制状态信息和运行时间等。每个试验的运行状态项稍有不同。试验信息显示试验报警、中断信息、历史记录以及谐振搜索与驻留记录。

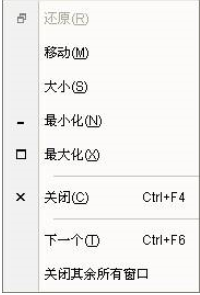
* 试验状态

试验状态显示了试验运行命令、部分试验控制命令，显示本次试验的目标、控制状态信息和运行时间等。每个试验的运行状态项稍有不同。

* 试验信息

试验信息显示试验报警、中断信息、历史记录以及谐振搜索与驻留记录。

* 窗口列表

窗口列表显示当前所有打开的窗口，单击窗口名称按钮，可以使该窗口显示在最上层。在窗口名称按钮上右击，弹出如下浮动菜单：

**图 4‑1右键浮动菜单**

* 状态栏

状态栏显示试验的存储路径和本次试验的运行状态，如初始化、运行预试验、正在执行第1个计划表的第1项等。

系统管理流程图如图4-2所示，系统主界面如图4-3所示,整个系统初始化关键代码如下:

ConfigureParameterAO tmpAO;

tmpAO.deviceName="PCI-1716,BID#1";

tmpAO.channelStart=0;

tmpAO.channelCount=1;

tmpAO.valueRange=*V\_Neg10To10*;

ConfigureParameterAI tmpAI;

tmpAI.deviceName="PCI-1716,BID#1";

tmpAI.channelStart=0;

tmpAI.channelCount=1;

tmpAI.valueRange=*V\_Neg10To10*;

tmpAI.clockRatePerChan=1000;

tmpAI.sectionLength=1024;

ConfigureParameterPCI tmp;

tmp.iniSettingAI=tmpAI;

tmp.iniSettingAO=tmpAO;

IniHelper \*tmpHelper=*new* IniHelper;

bool result=tmpHelper->writeToPCIIni("../IniSetting/PCI1716.ini",tmp);

*delete* tmpHelper;

*//* *Chart* *Viewer*

m\_ChartViewer = *new* QChartViewer(leftFrame);

m\_ChartViewer->setGeometry(10, 25, 640, 480);

connect(m\_ChartViewer, *SIGNAL*(viewPortChanged()), *SLOT*(onViewPortChanged()));

connect(m\_ChartViewer, *SIGNAL*(mouseMovePlotArea(QMouseEvent\*)),

*SLOT*(onMouseMovePlotArea(QMouseEvent\*)));

currentLabel=*new* QLabel;

currentLabel->setMinimumSize(200,25)；

int d7480rtn= Enc7480\_Init();

qDebug()<<"leisai number:"<<d7480rtn;

*if*(d7480rtn < 1)

{

QMessageBox::information(NULL,"提示","初始化ENC7480计数卡失败",QMessageBox::*Ok*|QMessageBox::*Cancel*);

*this*->close();

}

Enc7480\_Set\_Encoder(0,0);

instantAoCtrl = InstantAoCtrl::Create();

startFlag=*false*;

dataScaled[0]=0;

dataScaled[1]=0;

*//将从ini文件中导入的配置初始化*

Initialize();

}

**图 4‑2系统管理流程图**

开始

实例化各个窗口类

初始化运行参数

信号槽函数初始化

是否收到信号

否

主窗口处理其他逻辑事件

是

调用相应的槽函数

打开相应对话框

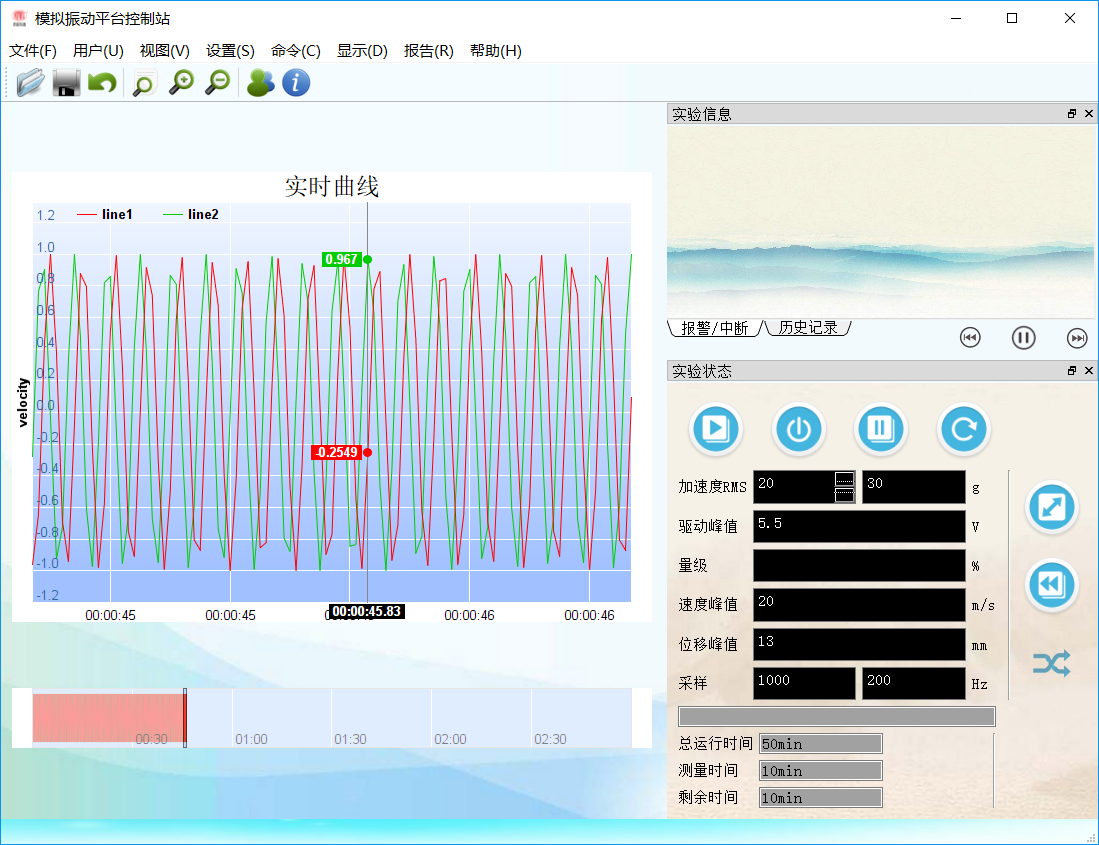
完成相应事件处理

是否线程阻塞

否

是

等待线程退出



**图 4‑3振动台上位机软件系统主界面**

4.2 系统通信模块

4.2.1 以太网TCP/IP通信基本参数的初始化

单精度模拟振动平台采用FPGA作为主控制器[20]。作为高性能的全数字控制器可以在FPGA中移植轻量级TCP/IP协议：LwIP协议栈，从而与上位机进行网络通信。

基本通信参数设置如下：

表4-1 网络局域网通信参数初始值

|  |  |
| --- | --- |
| IP地址 | 192.168.0.X网段 |
| 子网掩码 | 255.255.255.0 |
| 默认网关 | 192.168.0.1 |
| 端口 | 8800 |
| TCP服务器Socket | 上位机 |
| TCP客户端Socket | 控制器 |

在上下位机通信过程中，需要使PC端与控制器端的IP地址在同一局域网内，并且控制器客户端连接上位机服务器打开的有效端口，双方通过socket套接字通信，其通信概念图如图4-4所示，Qt中TCP服务器应用层协议通信关键代码如下：

class ProtocolSet

{

public:

ProtocolSet();

public:

//数据帧类型枚举体,作为公共接口暴露

enum MessageType{

DATA = 0xFF00 ,

COMMAND = 0xFF01 ,

ECHO = 0xFF02 ,

ERR = 0xFF03 ,

TEST = 0xFF04

};

enum ServerPort{

COMMUNICATION\_PORT = 8087,

FILE\_PORT = 40,

};

public:

//构造不同结构的信息帧，返回值为构造的信息帧

QByteArray send\_Msg(ProtocolSet::MessageType msg\_type,QString msg);

QByteArray payload;

qint16 head;

//数据帧头部长度 （字节数） 帧头（qint16） 2bytes + 长度（qint32） 4bytes + 功能码（qint16） 2bytes = 8bytes

qint32 HEAD\_LENGTH;

private:

//内部的数据帧构造

QByteArray data\_msg(QString msg);

QByteArray command\_msg(QString msg);

QByteArray echo\_msg(QString msg);

QByteArray error\_msg(QString msg);

QByteArray test\_msg(const QString &msg);

};

QByteArray ProtocolSet::send\_Msg(MessageType msg\_type, QString msg)

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据不同的msg\_type发送不同的数据包\*\*\*\*\*\*\*\*

qDebug()<<"开始构造数据包: "<<msg\_type;

QByteArray message;

switch (msg\_type) {

case DATA:

message = data\_msg(msg);

break;

case COMMAND:

message = command\_msg(msg);

break;

case ECHO:

message = echo\_msg(msg);

break;

case ERR:

message = error\_msg(msg);

break;

case TEST:

message = test\_msg(msg);

break;

default:

qDebug()<<"错误！发送数据类型不正确";

break;

}

//返回所构造的数据包

return message;

**图 4‑4 振动台上位机C-S模式通信概念图**

Server层

TCP层

负责管理Socket通信，传输数据

Operation层

负责对TCP字节流封装，上传指令等

Application层

负责解包应用层协议，生成纯指令

Logic层

对纯指令进行逻辑处理

4.2.2上位机通信设计

1.并发服务器介绍

在控制过程中，上位机软件系统服务器端可能同时需要连接多个控制器客户端，多个客户端需要同时传输数据，此时因受限其基于打开端口-读取端口数据-写入数据-关闭连接的固定编程模式，服务器同时支持多个客户端的连接变得困难。此时考虑并发服务器的实现，可以为每个客户端连接的TcpSocket分别分配一个专门的线程来处理[26]。从而可以实现同时与多个客户端通信的功能,其结构图如图4-3所示：

已完成连接队列

未完成连接队列

fork

accept

accept

fork

accept

fork

**图 4‑5并发服务器结构图**

图 4‑3并发服务器结构图

图 4‑3并发服务器结构图

图 4‑3并发服务器结构图

图 4‑3并发服务器结构图

图 4‑3并发服务器结构图

图 4‑3并发服务器结构图

图 4‑3并发服务器结构图

2.上位机处理接受数据流程如下：

1. 采用双缓冲机制来接收、发送数据

上位机软件系统和控制器之间存在通讯速率不匹配的问题，若只采用一个缓冲区进行收发数据容易引起内存溢出，读取空地址等通信错误，所以上位机通信采用双缓冲区实现读写分离，保证系统的高效性和鲁棒性。控制器发送数据到上位机后，上位机首先存放在双缓冲区中，之后再进行数据处理。

1. 校验数据

传输协议是基于TCP/IP协议，数据通过链路层、网络层、传输层解封装后到达应用层，此时应首先进行数据的校验，判断数据是否在传输的过程中出现错误、丢失等情况，步骤如下所示：

* 首先判断帧头是否正确，根据表3-1，帧头应为0xAAAA；
* 检查数据帧中FUN是否在规定范围内，根据表3-1，FUN应为0x00-0x10；
* 根据表3-1检查数据帧中LEN字段；
* 计算数据帧帧尾的校验和，判断其是否改变

校验过程中只要发现错误，则应立即丢弃当前数据帧，并且构造一条错误帧，发送至缓冲区中，待处理完毕后发送至错误源。

1. 根据FUN进行不同的逻辑处理

若通过(2)中的校验，则表明本次数据时有效数据帧，此时应根据表3-1判断数据中的FUN，进而转入不同的逻辑处理流程，完成指定的指令。

Qt中用socket编程接口实现并发服务器功能需要用户重写一个类来继承QTcpServer类，以及QTcpSocket类。QTcpServer类中有一个虚函数incomingconnection(qintptr handle),此函数将在TCP服务器监听到客户端连接后回调执行，用户通过重写这个函数，通过该函数获得连接套接字后，实例化不同的QThread类，开启不同的线程，并为每个C-S连接编写相应的QThread::run()函数，从而实现每个线程可以实现不同的处理流程,关键实现代码如下：

void PopWil\_TcpServer::incomingConnection(qintptr socket)

{

Controller\_Client -> setSocketDescriptor(socket);

PopWil\_New\_Thread \*client\_thread = new PopWil\_New\_Thread (socket);

connect(Controller\_Client, &disconnected(), this, & socket\_offline());

client\_thread -> start();

}

4.3 数据处理模块

模拟振动台上位机软件系统提供试验数据浏览和分析处理功能。浏览实验数据不需要硬件的支持，所以在设计过程中需要考虑在试验运行和脱离硬件状态下都可以浏览和处理由正式运行保存下来的实验数据：

4.3.1 默认配置信息的保存

试验配置信息保存了模拟振动台运行过程中的所有试验配置参数，对整个系统的正常、安全运行有着重要作用。在此上位机软件系统的界面设计中，需要实现一种加载和保存默认配置文件的功能，以便保存一些已经调节好的参数，这个问题可以通过将一些重要参数写入一个ini文件，和每次软件系统初始化时，均读入ini配置文件解决，具体实现过程如下:

* 从外部Ini配置文件中读取数据到主窗体的私有属性
* 从主窗体的私有属性传输数据到“通道设置”页面，并渲染到界面上
* 用户的“通道参数”界面可能修改默认的数据，此时需要将修改的数据存储到主窗体的私有属性
* 主程序利用配置的参数，进行数据的采集和控制
* 将修改的通道参数保存为新的默认数据

其实现原理框图如图4-6所示，关键实现代码如下：

void ConfigureDialog::Initailization()

{

InstantAoCtrl \* instantAoCtrl = InstantAoCtrl::Create();

Array<DeviceTreeNode>\*supportedDevices = instantAoCtrl->getSupportedDevices();

*if* (supportedDevices->getCount() == 1)

{

QMessageBox::information(*this*, tr("警告信息"),

tr("没有找到相应设备!"));

QCoreApplication::quit();

}

*else*

{

*for* (int i = 0; i < supportedDevices->getCount(); i++)

{

DeviceTreeNode *const* &node = supportedDevices->getItem(i);

qDebug("%d, %ls\n", node.DeviceNumber, node.Description);

ui.cmbDevice->addItem(QString::fromWCharArray(node.Description));

}

ui.cmbDevice->setCurrentIndex(0);

}

configure.profilePath = L"";

instantAoCtrl->Dispose();

supportedDevices->Dispose();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将配置文件读入到configureAO 和 configureAI中\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

IniHelper \*hh=new IniHelper;

ConfigureParameterPCIiniSettingFile=hh->readFromPciIni("../IniSetting/PCI1716.ini");

configureAO=iniSettingFile.iniSettingAO;

configureAI=iniSettingFile.iniSettingAI;

qDebug()<<"from the iniFile AO is :"<<configureAO.deviceName<<configureAO.channelStart<<configureAO.channelCount<<configureAO.valueRange;

sPIDInfo=hh->readFromCtrlIni("../IniSetting/CtrlIni.ini");

delete hh;

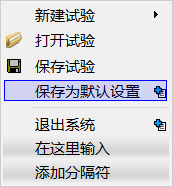
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将配置文件读入到configureAO 和 configureAI中\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IniHelper \*hh=new IniHelper;

ConfigureParameterPCIiniSettingFile= hh->readFromIni("../IniSetting/PCI1716.ini");

configureAO=iniSettingFile.iniSettingAO;

configureAI=iniSettingFile.iniSettingAI;;

功能界面如下所示,以及保存文件如下所示：

**图 4‑8配置文件加载原理框图**

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

主窗体

Ini配置文件

主窗体私有属性

控制过程

通道设置窗体

私有属性

**图 4‑6保存设置菜单栏**

**图 4‑7配置文件**

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

图 4‑5配置文件

4.3.2 试验参数和控制

保存的试验工程文件记录了当前实验的各项系统参数的设置情况，包括通道参数、控制参数、限制参数等等，为数据处理提供参考依据[29]。

这些试验参数仅供分析参考使用，不允许用户修改，所以个参数对话框的“确定”button都处于禁止按下状态。

此外，我们需要浏览一些数据，所以需要一些控制浏览的控件实现，本上位机软件系统中采用了5个PushButton按钮实现功能[30]，并且重写了他们的信号槽函数,功能介绍如下图所示：

* ****显示第一帧数据；
* 显示当前数据的前一帧数据；
* 显示当前数据的后一帧数据；
* 显示最后一帧数据；
* 系统会自动从第一帧数据开始浏览显示，直至用户按下“停止”命令或运行到最后一帧数据，系统才停止浏览
* 停止浏览后面的数据

**图 4‑9试验控制按钮介绍**

图 4‑6试验控制按钮介绍

图 4‑2生产者-消费者模式图图 4‑6试验控制按钮介绍

图 4‑6试验控制按钮介绍

图 4‑3生产者-消费者模式图

图 4‑0‑4生产者-消费者模式图

图 4‑5生产者-消费者算法参数图 4‑6生产者-消费者模式图图 4‑6试验控制按钮介绍

图 4‑6试验控制按钮介绍

图 4‑7生产者-消费者模式图图 4‑6试验控制按钮介绍

图 4‑6试验控制按钮介绍

关键实现代码如下所示:

void MainWindow::on\_Start\_btn\_clicked()

{

mytcpserver->sendData("start",ProtocolSet::COMMAND);

}

//点击菜单栏用户登录按钮

void MainWindow::on\_User\_1\_triggered()

{

loginscreen \*log = new loginscreen ;

log->show();

}

//点击菜单栏控制参数设置

void MainWindow::on\_Setting\_control\_para\_triggered()

{

control\_param \*control\_window = new control\_param;

//ui->Message\_dock->show();

control\_window->show();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*点击菜单栏的退出按钮

void MainWindow::on\_action\_Exit\_triggered()

{

int ret=QMessageBox::information(NULL,"提示信息","请确实是否退出操作界面？", QMessageBox::Ok|QMessageBox::Cancel,QMessageBox::Ok);

switch(ret)

{

case QMessageBox::Ok:

qDebug()<<"closing window";

this->close();

break;

case QMessageBox::Cancel:

qDebug()<<"will not close";

break;

default:

break;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按下菜单栏----新建实验按钮-----

void MainWindow::on\_action\_New\_triggered()

{

new\_experiment \*new\_exp = new new\_experiment;

new\_exp->show();

}

4.3.3 试验数据软件的存储和试验报告

在浏览试验数据过程中和浏览结束后，都可以按照一定的格式和内容产生相应的Word试验报告，用户可以定义报告信息和报告内容。振动台上位机软件系统提供标准的报告形式，可以在线产生报告或历史结果报告等,并且可以保存当前实验数据曲线，以便之后分析，实现代码如下所示：

*//* *The* *Save* *button* *is* *pressed*

void MainWindow::onSave(bool)

{

QString fileName = QFileDialog::getSaveFileName(*this*, "Save", "chartdirector\_demo", "PNG (\*.png);;JPG (\*.jpg);;GIF (\*.gif);;BMP (\*.bmp);;SVG (\*.svg);;PDF (\*.pdf)");

*if* (!fileName.isEmpty())

{

*//* *Save* *the* *chart*

BaseChart \*c = m\_ChartViewer->*getChart*();

*if* (0 != c)

c->makeChart(fileName.toUtf8().constData());

}

}

4.4 数据显示模块

4.4.1 三参量数据图表分析

在试验过程中，模拟振动平台时的实时位移、加速度、速度数据都反馈到上位机中，为了能实时、方便地观察浏览试验情况，需要将其动态曲线绘制出来。浏览试验数据曲线时，可以改变窗格大小、设置窗格中信号的 、添加和删除标记注释、缩放图形等等[32]。实时曲线界面如图4-11所示。

本上位机软件系统采用Qtcharviewer控件进行动态绘图，关键代码如下所示：

*//* *Initialize* *member* *variable*

m\_currentIndex = 0;

*//* *Initially,* *auto-move* *the* *track* *line* *to* *make* *it* *follow* *the* *data* *series*

trackLineEndPos = 0;

trackLineIsAtEnd = *true*;

*//* *Initially* *set* *the* *mouse* *to* *drag* *to* *scroll* *mode.*

pointerPB->click();

*//* *Enable* *mouse* *wheel* *zooming* *by* *setting* *the* *zoom* *ratio* *to* *1.1* *per* *wheel* *event*

m\_ChartViewer->*setMouseWheelZoomRatio*(1.1);

*//* *Configure* *the* *initial* *viewport*

m\_ChartViewer->setViewPortWidth(initialVisibleRange / (double)initialFullRange);

*//* *Start* *the* *random* *data* *generator*

*//dataSource* *=* *new* *RandomWalk(OnData,* *this);*

*//dataSource->start();*

*//* *Set* *up* *the* *chart* *update* *timer*

m\_ChartUpdateTimer = *new* QTimer(*this*);

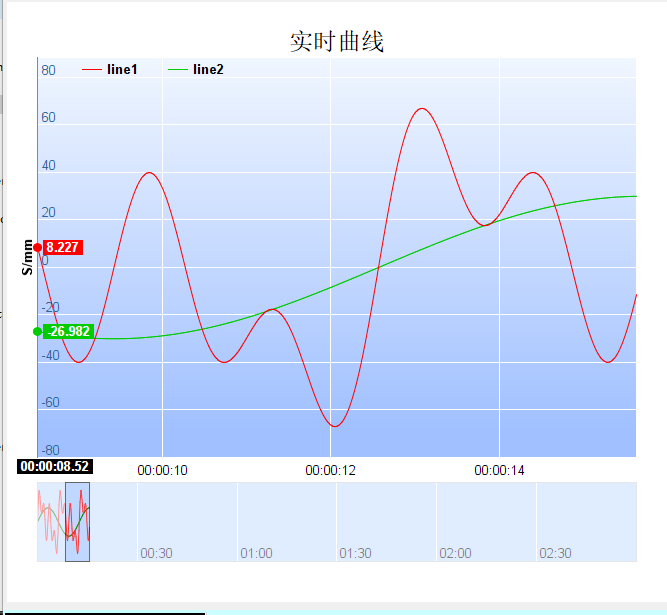
connect(m\_ChartUpdateTimer, *SIGNAL*(timeout()), *SLOT*(onChartUpdateTimer()));

*//* *The* *chart* *update* *rate* *is* *set* *to* *100ms*

m\_ChartUpdateTimer->start(100);

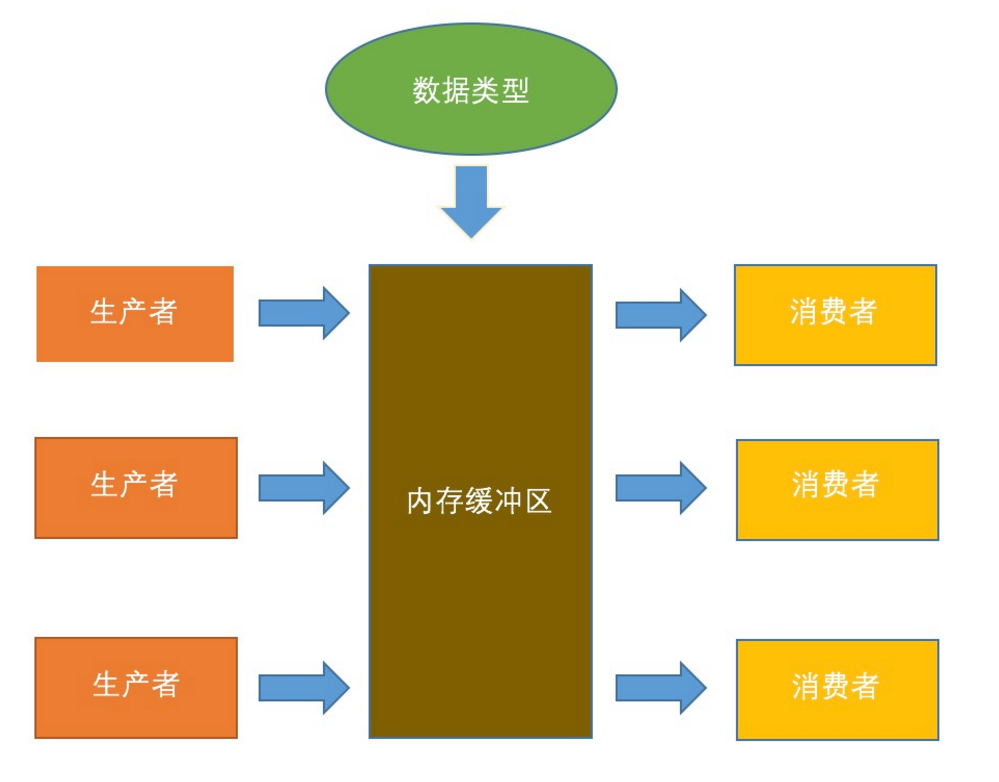
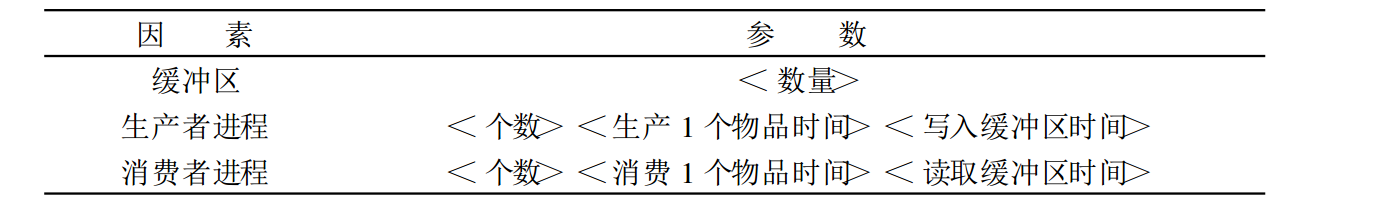
timer=*new* PerformanceTimer(*this*);

connect(timer,timer::timeout()),*this*,timer::update\_data());

timer->start(PERFORMANCEINTERVAL); *//20为毫秒*

**图 4‑10 实时曲线显示界面**

4.4.2 生产者-消费者矛盾

进行动态绘图的过程中，会碰到一个问题，一种情况是下位机在不停产生数据并上传，此时上位机在处理一些其他的任务，那么会导致数据溢出，并在绘图过程中产生闪烁中断等问题。而如果下位机上传数据的速度不够快，则会导致上位机不能正确绘图，并且容易引起系统崩溃。这个问题就是“生产者-消费者矛盾” [33]。而动态绘图过程中必须解决这个生产和消费速度不匹配的矛盾，从而解决这个并发读写的问题，生产消费者模式如下图所示：

**图 4‑12 生产者-消费者算法参数**

**图 4‑11生产者-消费者模式图**

对此生产者-消费者模型进行建模，可以知道分别有：一个消费者Fork、一个生产者Fork，一个数据缓冲区。用下列集合来描述该系统的状态[34]：

其中分别为消费者、生产者、数据缓冲区状态变量。

由于整个系统是异步进行的，当状态=时则会发生线程冲突，根据状态表此时，只能支持整个系统互斥访问。两个线程不能同时访问数据缓冲区，如此将导致程序的运行效率很低，而且整个系统的实时性不能保证。

**图 4‑13 状态**

写

生产

生产者

读

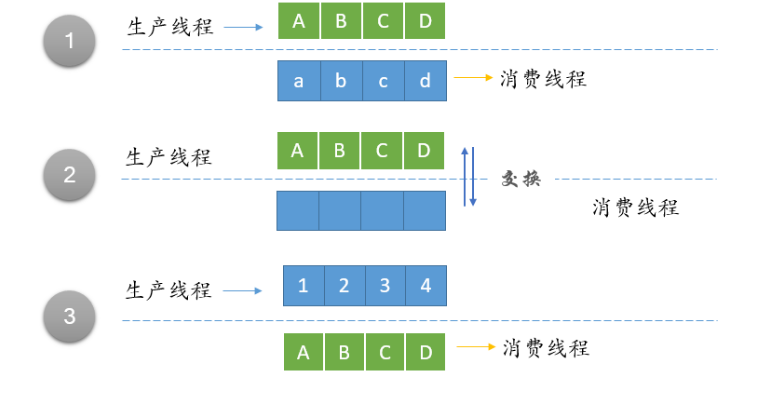
消费

消费者

写

读

缓冲区

解决这个问题可以采用开辟双缓冲区，实现绘图过程中的读写分离。通过添加互斥信号量对缓冲区进行读写保护，当某一个缓冲区作为下位机发送数据的缓冲区时，请求信号量，进行互斥保护[35]，此时上位机读取另外一个缓冲区进行绘图，完成操作后，释放信号量，读写缓冲区交换，由于数据的读写分离，并且有互斥信号量的读写保护，所以可以解决其速度不匹配的问题。实时曲线绘制流程图如图4-15所示。

**图 4‑14双缓冲区机制**

关键实现代码如下：

*class* QDoubleBufferedQueue

{

*public*:

*//构造函数，构造一个双缓冲队列*

QDoubleBufferedQueue(int bufferSize = 10000) :

bufferLen(0), bufferSize(bufferSize)

{

buffer0 = buffer = *new* T[bufferSize];

buffer1 = *new* T[bufferSize];

}

*//* 析构函数

~QDoubleBufferedQueue()

{

*delete*[] buffer0;

*delete*[] buffer1;

}

*//* 添加数据进入queque，成功则返回True，队满则返回False

bool put(*const* T& datum)

{

mutex.lock();*//mutex是互斥量，把buffer已经指针锁起来，不允许其他现成进行操作*

bool canWrite = bufferLen < bufferSize;

*if* (canWrite) buffer[bufferLen++] = datum;

mutex.unlock();

*return* canWrite;

}

*//* 采用二级指针，获得当前消费者队列指针

int get(T\*\* data)

{

mutex.lock();

\*data = buffer;

int ret = bufferLen;

buffer = (buffer == buffer0) ? buffer1 : buffer0;*//交换*

bufferLen = 0;

mutex.unlock();

*return* ret;

}

*private*:

*//* *Disable* *copying* *and* *assignment*

QDoubleBufferedQueue & operator=(*const* QDoubleBufferedQueue&);

QDoubleBufferedQueue(*const* QDoubleBufferedQueue&);

T\* buffer0;

T\* buffer1;

T\* buffer; *//当前指针*

int bufferLen

int bufferSize;*//缓冲区总的长度*

QMutex mutex;

开始

实例化双缓冲队列

初始化绘图数组，以及时间数组

初始化绘图定时器

等待定时器timeout事件

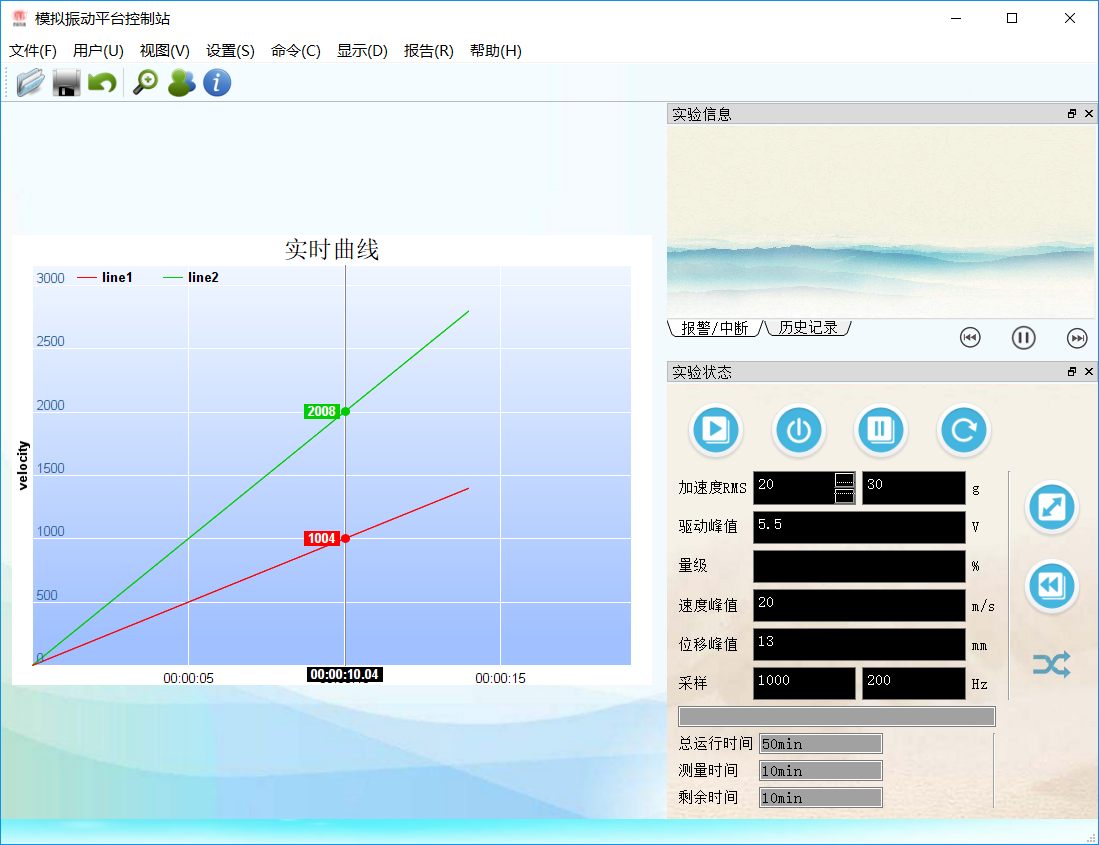
读取双缓冲区中生产者缓冲区数据

生产者消费者缓冲区交换

绘图更新

**图 4‑15 实时曲线绘制流程图**

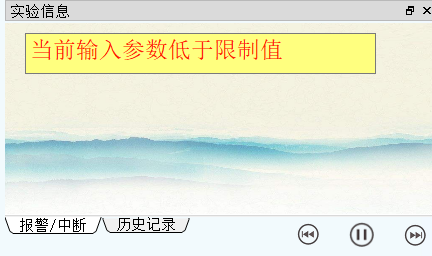
1. 系统调试及测试

模拟振动平台上位机软件系统主要由参数设置模块、数据传输模块、通讯模块、数据可视化分析界面四个模块构成[36]。现在通过对软件的实际使用，测试并评价上述各个模块的实际效果及实现功能，主界面如图5-1所示。

**图 5‑1 模拟振动平台上位机主界面**

5.1 上位机主界面测试

5.1.1 试验信息界面测试

主界面右上侧为试验信息界面，其上显示试验过程中试验报警、中断信息、以及历史记录等模拟振动台的实时状态，实时显示效果如图5-2所示:

**图 5‑2 试验信息显示界面**

5.1.2 试验状态界面测试

试验状态界面位于主界面右下侧，主要功能是监控振动台运行过程中的一些参量的峰值、有效值等等，便于调试过程中的数据分析，其显示效果如图5-3所示:

**图 5‑3 试验状态显示界面**

5.1.3状态栏测试

状态栏显示当前上位机连接情况，由于上位机是作为服务器，其可以连接多个控制器，需要在状态栏中显示当前连接的套接字个数、IP地址等信息，显示效果如图5-4所示:



**图 5‑4 状态栏显示界面**

5.2 上位机子界面测试

5.2.1新建试验界面

在启动模拟振动平台上位机软件系统后，首先初始化PC端与PCI板卡的驱动链接。之后读入之前保存的默认配置文件，加载一些默认配置参数，开启新建实验的方式有以下两种：

* 点击“文件”菜单->“新建实验”按钮
* 直接点击工具栏中的新建实验

在试验选择界面,主要划分为4个功能大类：典型试验、混合试验、波形再现、其他试验。

分别如下所示：

* 典型试验
* 随机试验
* 典型冲击试验
* 正弦试验
* 地震随机波形试验
* 混合试验
* 正弦加随机试验
* 随机加随机试验
* 地震波形加随机波形试验
* 波形再现
* 瞬态冲击试验
* 地震波模拟再现试验
* 其他试验

首先单击标签选择功能大类，通过点击不同的功能按钮，软件会开始随机试验、典型冲击、正弦试验等不同的试验，之后会打开上一步所选择试验的主界面,如下图所示:

**图 5‑5试验选择界面**

图 5‑2试验选择界面

图 5‑2试验选择界面

图 5‑2试验选择界面

图 5‑2试验选择界面

图 5‑2试验选择界面

图 5‑2试验选择界面

图 5‑2试验选择界面

通过点击不同的button控件，从而实现加载不同的实验数据，进行所选择的试验。

5.2.2用户登录界面

模拟振动平台上位机控制管理软件提供了用户管理机制，可以限制用户使用权限，防止软件被擅自使用。

可以在主界面中单击“用户”菜单—‘用户登录’，来进行用户管理,用户登录的界面如下所示:

**图 5‑6用户登录界面**

图 5‑3用户登录菜单

图 5‑3用户登录菜单

图 5‑3用户登录菜单

图 5‑3用户登录菜单

图 5‑3用户登录菜单

图 5‑3用户登录菜单

图 5‑3用户登录菜单

用户登录菜单如下:

**图 5‑7用户登录菜单**

图 5‑4用户登录界面

图 5‑4用户登录界面

图 5‑4用户登录界面

图 5‑4用户登录界面

图 5‑4用户登录界面

图 5‑4用户登录界面

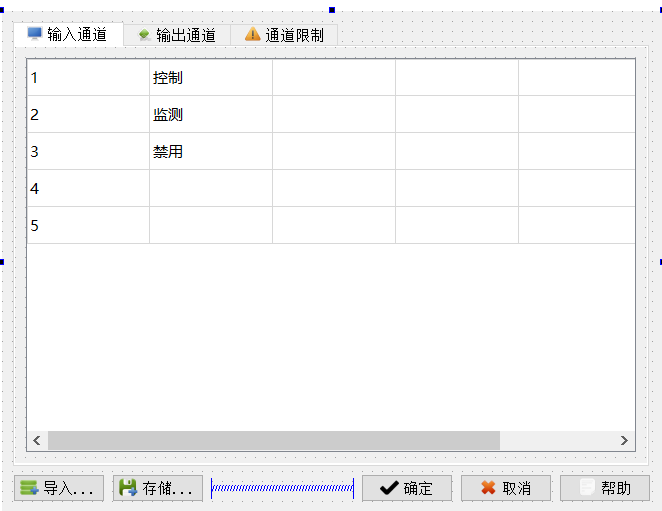
图 5‑4用户登录界面

通过用户管理，可以设置用户的权限，管理员可以修改用户权限：开始试验、通道参数、控制参数等等。管理员通过赋予不同的用户不同的参数修改权限，可以是整个系统运行在稳定、安全的状态。

5.2.3 通道参数设置界面

主界面中点击“设置”---“通道参数”，即可打开通道参数设置界面，通道参数用来设置所有连接到输入端的传感器或者前置调理电路的相关参数，如输入通道的量程、耦合方式、传感器的类型等、还有输出通道的参数，以及通道限制参数等等。如下图所示：

**图 5‑8设置菜单**



**图 5‑9 通道参数界面**

图 5‑6通道参数界面

图 5‑6通道参数界面

图 5‑6通道参数界面

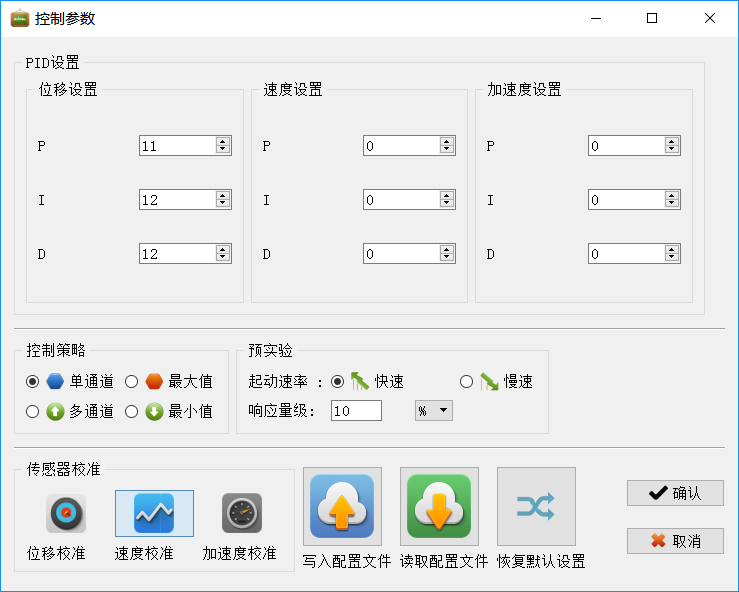
图 5‑6通道参数界面

图 5‑6通道参数界面

图 5‑6通道参数界面

图 5‑6通道参数界面

5.2.4 控制参数设置界面

界面中点击“设置”—-“控制参数”，即可打开控制参数设置界面，控制参数是用来定义与试验运行、控制相关的参数。例如，控制策略、量级、预实验和高级设置等等，如下图所示:

**图 5‑10控制参数设置界面**

主界面中点击“设置”—-“控制参数”，即可打开控制参数设置界面，控制参数是用来定义与试验运行、控制相关的参数。例如，控制策略、量级、预实验和高级设置等等，如下图所示:

**图 5‑11 控制参数设置菜单**

图 5‑7控制参数设置界面

图 5‑7控制参数设置界面

图 5‑7控制参数设置界面

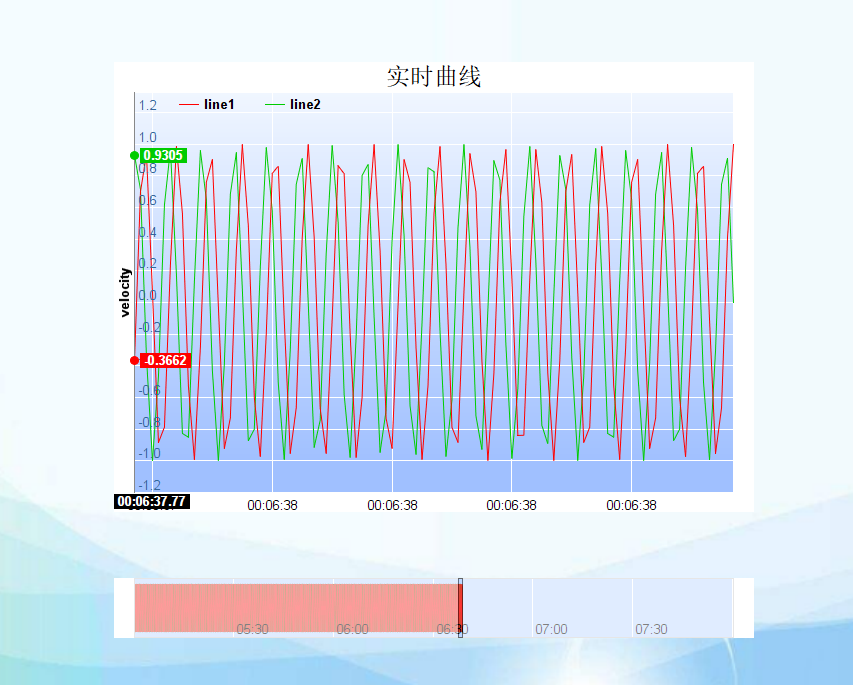
图 5‑7控制参数设置界面

图 5‑7控制参数设置界面

图 5‑7控制参数设置界面

图 5‑7控制参数设置界面

5.3 上位机实时曲线界面测试

在主界面的左侧，为信号实时曲线显示区域，采用Qt charviewer控件开发，利用多媒体定时器实时绘图，并且具有坐标自适应缩放，实时显示数据幅值等功能，在窗口任意处单击并拖动鼠标可以上、下、左、右拖动曲线，双击窗口将弹出数据显示大窗口以供清楚查看数据,并且可以通过点击保存按钮将当前曲线保存为PNG、JPG格式的文件，便于后续离线分析。绘图区域下方有全局绘图状态条，可以全局总览整个试验过程中的数据波形从而进行分析。三参量曲线可视化界面如图所示,

**图 5‑12 三参量曲线实时显示界面以及全局状态栏**

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

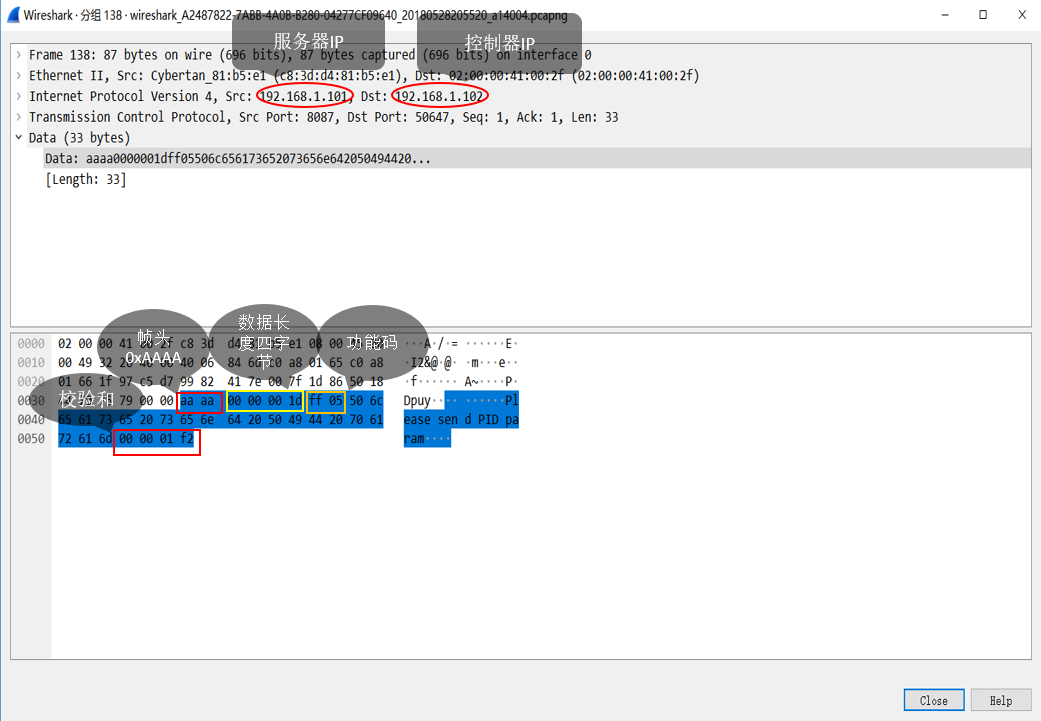
图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

其中line1为设定曲线，line2为实际测量曲线：

5.4 上位机通讯功能测试

上位机通讯过程中采用wireshark网络抓包，可以看出传输数据由帧头+功能码+长度+数据+校验码组成，与之前制定的应用层协议保持一致：

**图 5‑13 WIRESHARK网络通信抓包**

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

图 5‑8三参量曲线实时显示界面以及全局状态栏

1. 总结与展望

6.1总结

本文基于Qt5开发平台设计了一个模拟振动平台的上位机软件管理系统，在本次课题设计中从分析系统需求、建立总体框图，到解决通信、绘图同步等问题，都让我系统的了解到一个大型软件工程的基本开发流程。本文首先介绍了模拟振动平台国内外研究现状，以及其对各个产业发展的重大意义。接着根据了模拟振动平台的控制指标和控制要求，简单分析了振动平台的机械结构、算法设计以及硬件结构，并确立了系统的结构框架图。之后对于振动平台上位机软件系统进行了详细分析，并确立上位机软件总体功能模块图，将系统划分为参数设置模块、数据传输模块、通讯模块、数据可视化分析界面等四个方面。最后是上述四个模块的具体功能以及实现过程中的分析以及问题，重点说明通信协议的设计以及三参量动态绘图过程中生产者-消费者矛盾的解决方案，本文特色和创新点主要有：

* 采用基于TCP/IP以太网的通信协议，通过严格定义应用层数据帧结构，以达到数据的高效、无差错的传输，并能很好的实现各种功能的通信。
* 采用多并发服务器的连接方式，实现上位机对多个控制器的同步控制。
* 振动平台上位机软件采用配置.ini文件导入配置信息，可以方便地配置用户的默认配置，并且可以保存默认数据，更新设置好的各通道的参数配置。
* 通过分析、解决数据显示模块中动态绘图的生产者-消费者矛盾，可以实现三参量曲线的无卡顿动态绘制，并且可以实现绘图窗口的坐标自适应，曲线缩放、截图保存等功能。

本文中涉及到的知识有，TCP/IP协议套接字通信、ChartDirect控件的编程及使用，生产者-消费者矛盾的分析、以及Qt中C++面向对象编程的前端UI设计以及后端的逻辑处理编程等等。

在分析问题、调试系统的整个过程中，积累了许多经验，明白了一个复杂系统的设计工作流程，以及如何发现问题、分析问题、解决问题。

6.2 展望

本文设计了一个模拟振动平台的上位机软件管理系统，该系统可以方便用户调试、管理、监控模拟振动平台，实现与控制器通信功能。但是，由于时间问题，以及学识水平，软件系统很多地方考虑欠缺，许多功能还待完善，整个系统的鲁棒性不够强，整个软件系统可以从以下方面进行优化调整：

* 可以考虑加入OpenGL控件，对模拟振动平台进行3D实时建模，实时监测振动平台在空间中的状态。
* 可以加入数据库，实现对用户权限的管理，更好的保证试验过程的安全性。
* 优化通信过程，提高数据的传输效率。
* 可以加入多线程处理，提高软件整体的运行速度[36]。
* 本课题是电脑端通过以太网与控制器直接连接，未来可以考虑加入无线通信功能。基于Wi-Fi与控制器进行通讯[36]

模拟振动平台上位机管理系统能够很好的配置振动台控制器的使用，非常具有应用价值，通过对整个软件系统的不断优化调整，最终能够最大限度地发挥其振动平台控制器管理、振动平台设备等方面的作用。

参考文献

1. 孙俊磊.基于电液振动台的模型参考自适应控制的研究[D].燕山大学,2009.
2. 贾丽花.电液伺服地震模拟振动台控制算法的研究与实现[D] 北京：北京交通大学,2006
3. 赵勇.液压振动台高精度正弦振动的控制策略研究 [D] 哈尔滨：哈尔滨工业大学 ,2009
4. 田磐.地震模拟振动台的时域复现控制策略研究 [D] 浙江：浙江大学,2015
5. 崔伟清.地震模拟振动台控制方法及动态特性的研究 [D] 河北：河北工业大学,2012
6. 李宁,那文波.一种极高电动振动台低频控制精度的方法 [J],计算机应用与软件 2010, 27(9):178-180
7. 沈刚.三自由度电液振动台时域波形复现控制策略研究[D]. 哈尔滨：哈尔滨工业大学,2011
8. 高春华,纪金豹,闫维明,李娜.地震模拟振动台技术在中国的发展[J].土木工程学报,2014,47(08):9-19.
9. 万凯,王萍,朱冬云.电液振动台控制系统的现状与发展[J].仪器仪表用户,2012,19(04):1-5.
10. 王会民,洪涛,陈家焱,赵佃云.基于模糊PID电动振动台控制系统设计与实现[J].自动化与仪表,2018,33(01):10-14.
11. 王伟. 两自由度液压振动台设计[D].哈尔滨工业大学,2011.
12. 张波. 地震模拟振动台及其控制方法研究[D].兰州理工大学,2013
13. A.M.Karshenas, M.W.Dunnigan and B.W.Williams Adaptive inverse control algorithm for shock testing [J],2000,IEE Pt~oc.-ContrvlThemy Appl,147(3):267-276
14. Gang Shen1,2, Guang-ming LV2, Zheng-mao Ye2,Da-cheng Cong2 and Jun-wei Han2,Feed-forward inverse control for transient waveform replication on electro-hydraulic shaking table [J],,2011,Journal of Vibration and Control,18(10) 1474–1493
15. Sijia Tian. Thermal control of the temperature control furnace and the upper computer design[A]. 计算机科学与电子技术国际学会会议论文集[C]. Atlantis Press,2016:499-502
16. Bin Zhao, Wei Zheng, Li Hui , Ying Yang , Chengji Lu, Li Yang, Zhi Wang, Xiangtao Ran, Realization of Wave Direction Estimation on Upper Computer Software of Acoustics Wave Gauge[A]. IEEE BEIJING SECTION(跨国电气电子工程师学会北京分会)会议论文集[C].IEEE,2017:656-659
17. 范宣华,胡绍全,王东升,张志旭.电动振动台随机振动试验有限元仿真[J].噪声与振动控制,2008(02):41-43.
18. 柯艳,李杰,孔祥雷,刘吉吉.基于USB2.0的多路数据采集系统上位机软件设计[J].测试技术学报,2010,24(04):351-356.
19. 王海霞,颜桂定,李宝辉,刘晓燕,王轶.直线电机运动控制系统的软件设计与实现[J].电子测量与仪器学报,2013,27(03):264-269.
20. 高峰,俞立,王涌,卢尚琼,张文安,于莉洁.无线传感器网络作物水分状况监测系统的上位机软件开发[J].农业工程学报,2010,26(05):175-181
21. 陆文周.Qt5开发及实例 [M] 电子工业出版社,2014.
22. Gang Li School of Information Engineering Chang'an University Xi'an City,Shaanxi Province,China PengChang ZHANG School of Information Engineering Chang'an University Xi'an City,Shaanxi Province,China. Realization of Large-Scale Civil Engineering Wireless Health Monitoring System[A]. Intelligent Information Technology Application Research Association, Hong Kong.Proceedings of 2010 International Conference on Computational Intelligence and Vehicular System (CIVS2010)[C].Intelligent Information Technology Application Research Association, Hong Kong:,2010:3.
23. Chenxi Zheng. On-line Monitoring System Design of Electrical Equipment[A]. 信息化与工程国际学会.Proceedings of 2016 2nd Workshop on Advanced Research and Technology in Industry Applications(WARTIA 2016)[C].信息化与工程国际学会:,2016:4.
24. 朱小超,徐雪春.基于Modbus协议的上位机与单片机通信的实现与仿真[J].仪表技术与传感器,2011(06):65-68.
25. 朱吉佳,蔡家麟.基于Qt的业务监控系统界面设计与实现[J].计算机技术与发展,2008(03):236-239+242.
26. 黄翩,张琼,祝婷.基于Qt的一个服务器多个客户端的TCP通信[J].电子科技,2015,28(03):76-78+82.
27. 钱琳. 嵌入式监控系统上位机软件设计[D].浙江大学,2008.
28. 黄浩华,杨学山,程建伟,董玲,马树林.小型伺服式电动振动台[J].世界地震工程,2002(03):69-72.
29. 胡惠玉.基于工业以太网实现上位机对远程PLC的监控[J].长春工程学院学报(自然科学版),2011,12(02):113-116.
30. 吴晟,苏庆堂,罗斌,赵莉楠,蔡灿民.基于Socket和多线程技术的并发服务器的研究[J].昆明理工大学学报(理工版),2006(04):39-42.
31. 刘新强,曾兵义.用线程池解决服务器并发请求的方案设计[J].现代电子技术,2011,34(15):141-143.
32. 彭均键,史步海,刘洋.基于Qt的嵌入式GUI开发平台的搭建[J].微型电脑应用,2010,26(02):40-42+5-6.
33. 邹伟,杨平,徐德.基于MCGS组态软件的上位机控制系统设计[J].制造业自动化,2008,30(12):103-108.
34. 严兵.生产者/消费者问题的分析和实现[J].西华大学学报(自然科学版),2006(02):13-15+97-98
35. 高升,冯亚丽,林冬梅.基于COM的生产者-消费者问题的解法[J].微型机与应用,2001(04):6-8.
36. 许博雅. 基于足底压力测量的康复训练管理设计[D].苏州大学,2017.

致谢

写到这里,我的本科毕业论文差不多写完了。美好大学时光也即将结束，回想自己大学四年，有太多太多的感慨。

本课题的完成，我首先最要感谢的是湖南大学的江未来老师，在毕业设计期间每当我遇上问题询问江老师，他都会细心回答，给我许多建议，让我能解决困难，同时他也对我的论文提出了许多宝贵的修改意见，他严谨的治学态度，让我受益匪浅，没有他的帮助，我是完不成这篇毕业论文的。还有浙江大学网络传感与控制研究组贺诗波、陈积明、史治国老师，我也要感谢他们的耐心指导。他们言传身教，在我毕设期间他们给予了无数的指导和关心。

在湖南大学电气与信息工程学院学习的四年里，我要感谢电气学院的所有老师，是他们的言传身教，深深影响了我，让我能够在学术的道路越走越远。也正是湖南大学给我提供的这个高水平的平台，让我在本科期间能接触到最前沿的科研信息，也让我能成功保送浙江大学继续深造。

和我一同努力、一起废寝忘食的还有浙江大学教九123实验室的师兄们。他们都给予了我许多许多关键的意见和建议，也正是他们让我体会到了团队的力量， 我也要对他们表达我的谢意。

最后，在离开之前，我多想再看一眼岳麓红枫，再踏进岳麓书院的门栏。但愿下次再与湖大相遇，仿佛不曾走远。