**

## HUNAN UNIVERSITY

毕 业 论 文

|  |  |
| --- | --- |
| **论文题目** | 基于PCI的单自由度振动 |
|  | 平台上位机软件管理系统设计 |
| 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的单自由度振动平台上位机软件管理系统设计

摘 要

VR（虚拟现实）技术，是一种通过搭建一个模拟真实环境的人机交互系统[1]，从而高真实度再现真实环境的技术。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，一方面可以为用户提供身临其境的浸入感，另一方面，可以促进VR（虚拟现实）技术在各种结构测试中的应用，例如道路模拟机，房屋结构抗震设计，海况模拟平台，飞行模拟器等等[30]。本文基于单自由度模拟振动平台系统，主要是开发一套用于与系统下位机（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，首先介绍了其整体设计框架以及用户界面，然后对于上位机软件系统的各个功能：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[3]。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**

In the field of Virtual Reality(VR) , people can experience virtual worlds through a man-machine interactive system [1].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. Through this technology ,on the one hand ,It can provide users with immersion .On the other hand ,It can promote the application which based Virtual Reality(VR) of all kinds of structure test ,such as road simulator ,anti-seismic design of building structure ,sea state simulation platform , the flight simulator , etc[30]. Based on the single-degree-of-freedom simulation vibration platform system ,this paper mainly develops a set of upper computer software system for the interaction and debugging of the system’s lower computer(the controller).

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 .For upper software system, 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[3] .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** 3](#_Toc513752953)

[**ABSTRACT** 4](#_Toc513752955)

[**第一章 绪论** 8](#_Toc513752956)

[1.1 课题背景及研究意义 8](#_Toc513752957)

[1.2 国内外研究现状 8](#_Toc513752958)

[1.3 本设计的主要内容和章节安排 9](#_Toc513752959)

[**第二章 模拟振动平台系统总体介绍** 10](#_Toc513752960)

[2.1 总体框架 10](#_Toc513752961)

[2.2 机械设计 15](#_Toc513752961)

[2.3 硬件系统 14](#_Toc513752962)

[2.4 算法研究 19](#_Toc513752962)

[2.5 软件系统 20](#_Toc513752962)

[**第三章 上位机软件系统设计** 20](#_Toc513752963)

[3.1 软件系统需求分析 20](#_Toc513752964)

[3.2 上位机软件系统整体架构设计 21](#_Toc513752965)

[3.2.1 基本功能框图 21](#_Toc513752966)

[3.2.2 总体架构设计 22](#_Toc513752967)

[3.3 软件开发平台简介 23](#_Toc513752968)

[3.4通信模块总体设计 23](#_Toc513752969)

[3.4.1 通信框架概述 23](#_Toc513752970)

[3.4.2 通信过程 24](#_Toc513752973)

[3.4.3 应用层帧结构 26](#_Toc513752974)

[**第四章 系统实现** 31](#_Toc513752975)

[4.1 系统管理模块 31](#_Toc513752976)

[4.2 系统通信模块 37](#_Toc513752980)

[4.2.1 以太网TCP/IP通信基本参数的初始化 37](#_Toc513752981)

[4.2.2上位机通信设计 41](#_Toc513752982)

[4.3 数据处理模块 43](#_Toc513752983)

[4.3.1 默认配置信息的保存 43](#_Toc513752984)

[4.3.2 试验参数和控制 46](#_Toc513752985)

[4.3.3 试验数据软件的存储和试验报告 51](#_Toc513753090)

[4.4 数据显示模块 52](#_Toc513753091)

[4.4.1 三参量数据图表分析 52](#_Toc513753092)

[4.4.2 生产者-消费者矛盾 53](#_Toc513753093)

[**第五章 系统调试及测试** 58](#_Toc513753094)

[**第六章 总结与展望** 65](#_Toc513753095)

[6.1总结 65](#_Toc513753096)

[6.2 展望 66](#_Toc513753097)

[**参考文献** 67](#_Toc513753098)

[**致谢** 67](#_Toc513753098)

插图索引

图 2‑1 模拟振动平台总体框架 10

图 2‑2 控制系统框图 11

图 2‑3三参量控制框图 14

图 2‑4迭代控制原理图 16

图 2‑5 自适应逆控制原理图 17

图2‑6 硬件系统原理图 17

图 3‑1 振动平台上位机软件功能图 22

图 3‑2 振动平台上位机软件总体框图 22

图 3‑3 TCP/IP组网结构图 24

图 3‑4 TCP三次握手连接示意图 24

图 3‑5 下位机通信示意图 25

图 4‑7生产者-消费者模式图 53

图 4‑8生产者-消费者算法参数 54

图 4‑9 状态 54

图 4‑10双缓冲区机制 55

图 5‑1振动平台上位机管理系统主界面 56

图 5‑2试验选择界面 57

图 5‑3用户登录菜单 58

图 5‑4用户登录界面 59

图 5‑5设置菜单 59

图 5‑6通道参数界面 60

图 5‑7控制参数设置界面 61

图 5‑8三参量曲线实时显示界面以及全局状态栏 62

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[33]。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[7]。

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[7][8]。

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。

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，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。

1.3 本设计的主要内容和章节安排

本课题研究设计一套便于与模拟振动平台下位机通讯，调试，便于用户操作的上位机软件系统。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，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[34]。

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[3]。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。之后介绍了上位机通信设计以及具体实现过程。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：第一部分从机械系统的角度出发，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；第二部分从算法角度出发，在机械角度的基础上，通过算法实现高精度实时复现；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[3]。

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，速度传感器、电源、和控制系统组成。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[5]。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[30]：

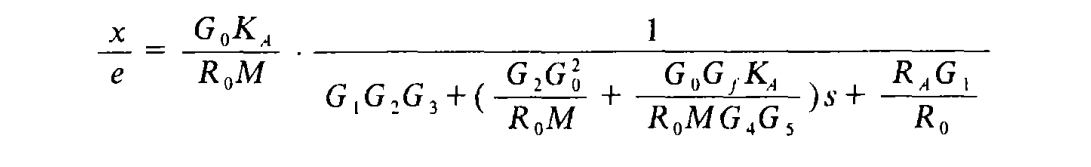
整个振动台最高频率响应为70Hz，在这种情况下，振动平台控制系统高频噪声可以忽略。此时整个系统的运动方程为：

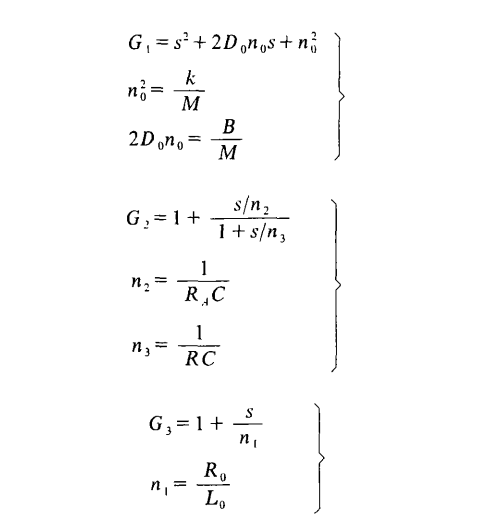
(1)

此式中：为振动平台系统质量； 为振动台电动激振器的转换系数； 为整个系统阻尼系数比。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：



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[11]。

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：

* 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

本部分的技术关键包括：

1. 振动平台的运动学、动力学建模：推导振动台运动学、动力学表达式，具体研究振动台位姿随时间的变化规律以及振动台所受力与其运动规律之间的关系
2. 动力结构设计：

利用振动台的动力学模型，通过分析其功率曲线图，找出振动台的最大功率点，从而选取出模拟振动平台的电动激振元件和驱动元件，推导模拟振动台的动力方程，并通过试验分析其鲁棒性**，在满足工业指标后，运用仿真软件振动台的动力学的数学模型，对模型进行仿真验证。运动学分析是从几何角度去描述和研究物体位置随时间变化规律[35]。**

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，本项目采用下列控制算法来改善控制效果：

1. 三参量控制算法：

**在设计反馈控制器的过程中，为了补偿振动台在单纯的位移控制情况下工作频宽较窄阻尼较小的缺陷，研究采用三参量控制算法来更好的实现不同频段的加速度、速度、位移控制。三参量控制算法的核心思路如下：低频段采用位移控制，可以有效减少系统高频噪声，中频段采用速度控制方式，提高系统响应速度，高频段采用加速度控制，保证系统控制精度，从而能够实现增大系统阻尼，拓宽系统工作频宽的目的[4]。其控制框图2-3如下：**

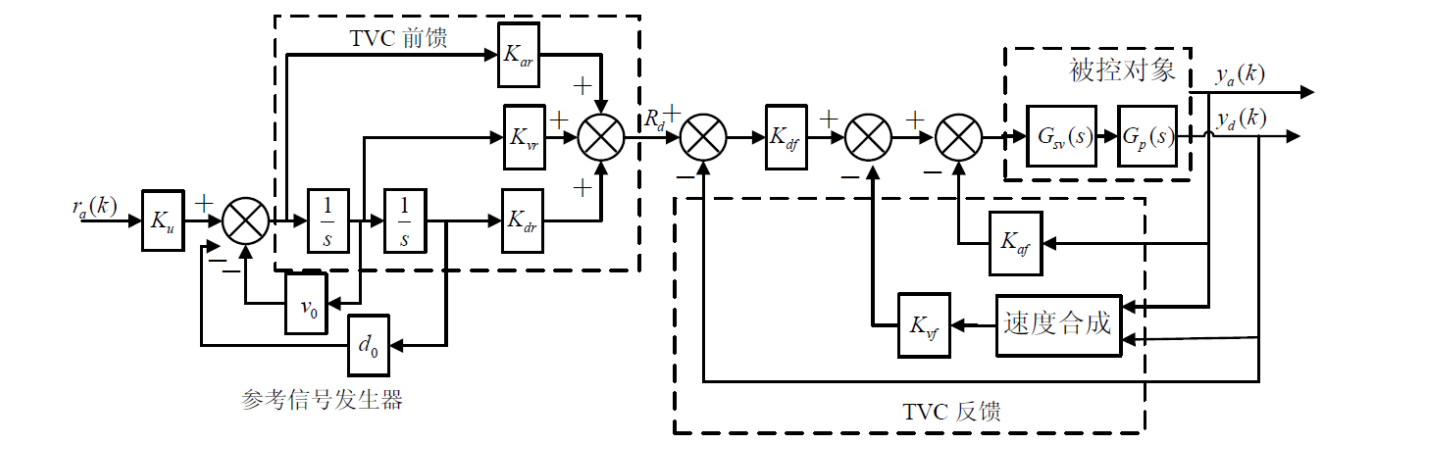


图 2‑3**三参量控制框图**

**本项目根据模拟振动平台控制系统的结构和工作性能，对振动台系统进行一定的线性化处理，搭建系统的数学模型。在位移控制的基础上，增加加速度和速度环节，设计三参量反馈控制器和三参量前馈控制器。三参量前馈控制器适用于系统频宽不足的系统，其原理是可以消除系统闭环传递函数中非常靠近虚轴的极点；三参量反馈控制将在基于位置闭环的基础上，通过加速度反馈和速度反馈来提高系统的阻尼比和固有频率，从而在保证系统稳定性的基础上拓展振动平台系统的工作频宽。**

1. 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[36]。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[36]。**

**迭代学习控制在对具有较高位置的精度要求、较强的非线性耦合和精确轨迹跟踪控制要求的动力学系统研究中有广泛应用[36]。迭代学习算法在应用于期望轨迹跟踪时，通过保存上一次的控制信息和经验数据利，不断迭代补偿下一次的控制量，使得系统能够在短时间内快速、精确的复现期望轨迹波形。**

**迭代学习控制在结构上包括开环学习和闭环学习，闭环学习在收敛条件、收敛速度和抗干扰能力等方面都优于开环，但是运算量要大于开环，而优于伺服控制系统已经能使系统的稳定性得到保障，因此本项目拟采用开环迭代学习控制律，通过迭代学习补偿修正系统的输出信号，从而使振动台控制系统能够高精度复现设定波形。其原理如图2-4所示：**

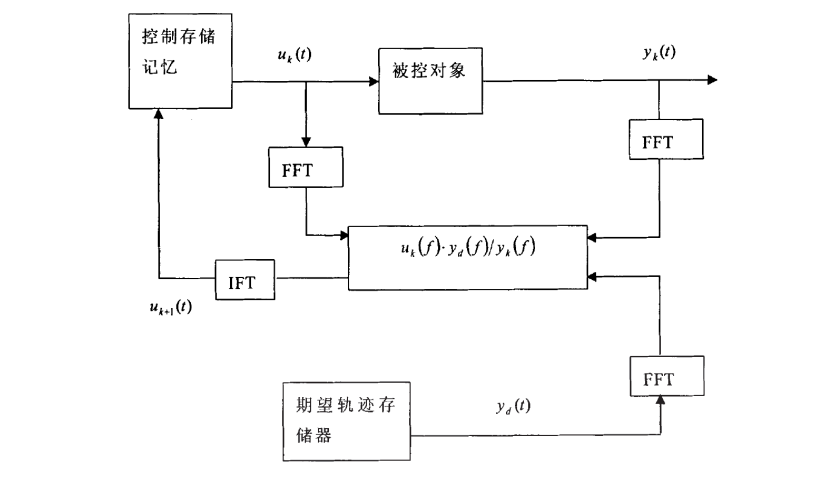


图 2‑4**迭代控制原理图**

1. 自适应逆控制算法[12][15]

**在迭代学习控制算法中，需要对系统的频响函数进行估计，并采用逆频响函数对系统进行输入信号的更新，因此振动台系统的频响函数的测量和辨识的精度对于迭代收敛的速度和波形复现的精度会有很大的影响，使得该控制方法可能存在以下问题：(1） 实际情况中系统的频率响应函数随时间变化，并不能用离线方法进行辨识；（2）迭代学习控制方法需要进行大量的离散傅里叶变化和逆向离散傅里叶变换变换来辨识系统的频响函数并产生修正的输入信号，而时域波形复现是瞬态变化的是瞬态变化的，采用 H1 估计法无法实现实时估计系统的频响函数。**

**对此，本项目拟采用自适应逆控制方法，将期望的输出波形信号通过一个补偿滤波器产生被控系统的输入信号，使系统的输出波形收敛于参考波形，从而使振动台输出期望波形。**

**常用的自适应滤波算法包括最小均方误差（LMS）算法和递推最小二乘（RLS）算法两类。最小均方误差（LMS）算法计算量小，易于实现，但存在着收敛精度和收敛速度之间的矛盾，通常采用变步长的策略来解决这一矛盾。递推最小二乘（RLS）算法，具有快速收敛性，但是在计算复杂度和稳定性方面表现较差。本项目将结合两种算法的优缺点具体分析，对两种算法进行算法实现和比较。自适应逆控制原理图2-5如下：**

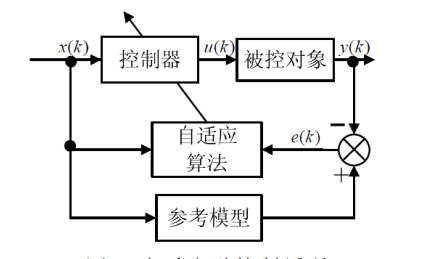
****

图 2‑5 **自适应逆控制原理图**

2.4 硬件系统

模拟振动平台以FPGA作为数字主控芯片，硬件主题结构图如图所示：

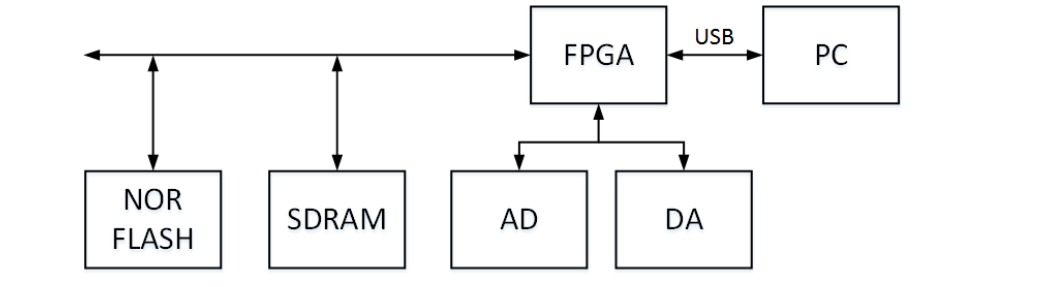


图2‑6 **硬件系统原理图**

1. FPGA

采用高性能的 FPGA 作为整个系统的控制器， FPGA 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.PCI-1716可以提供16路单端模拟量输入或8路差分模拟量输入,也可以组合输入.它带有2个16位D/A输出通道,16位数字量输入/输出通道和1个10HMs16位计数器通道.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。也能直接从主存储器中读取代码，从而作为Embedded控制器使用。

* 1. 软件系统
  2. 上位机[17][13][14]

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：对TCP/IP应用层通讯协议的处理，并通过以太网口与下位机进行消息通讯，数据的分类，错误警告处理等；

以太网及USB通讯：采用TCP/IP通讯协议和从机互相收发数据，进行通讯。

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。

* 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。是全数字控制器的核心部分。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

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。

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：

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。

振动平台软件应具有控制功能和辅助功能，可以根据不同的用户需求配置响应的控制器和软件功能。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：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[34][34]。

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

3.2.2 总体架构设计

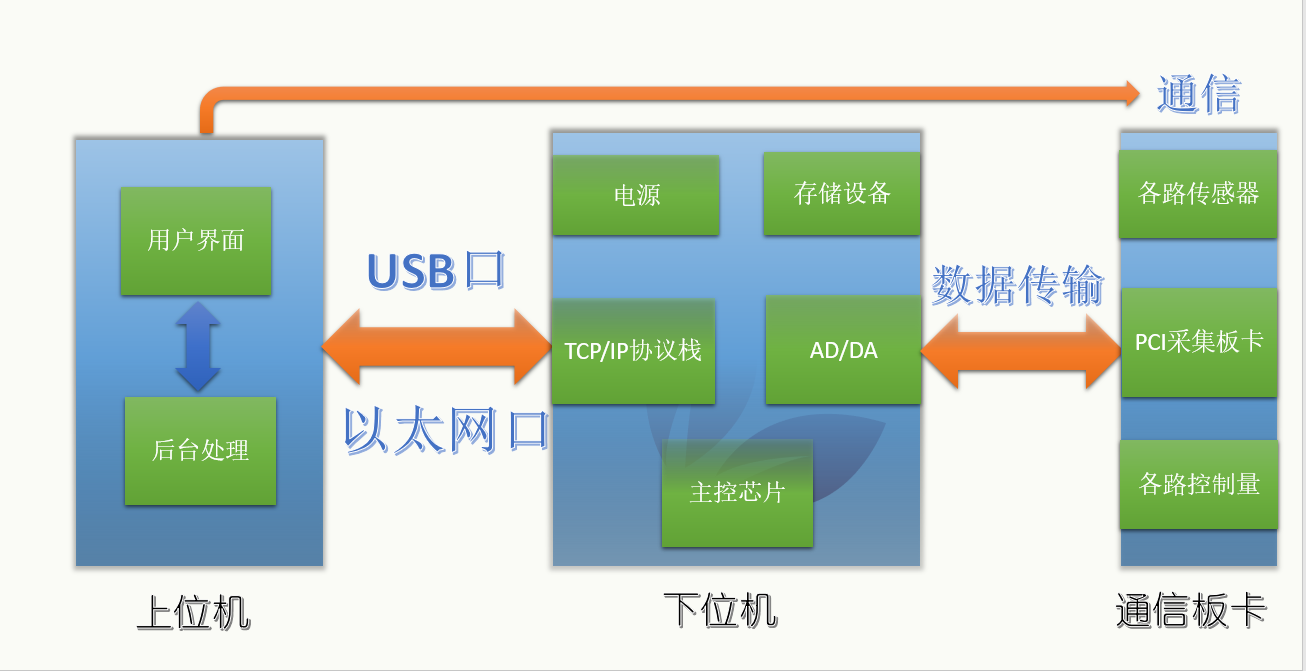
**** 宏观上来看，振动平台软件系统的总体框架由控制器下位机和管理软件上位机组成，框架如图3-2所示:

图 3‑2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。

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[10][34]。

Qt是的桌面应用IDE集成工具，具有很良好的跨平台特。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。

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[28]。

TCP/IP协议可以实现客户端-服务器通讯模式，其中上位机作为服务器，控制机作为客户端，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TCP/IP以太网组网方式，结构图如图3-3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 3‑3 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：

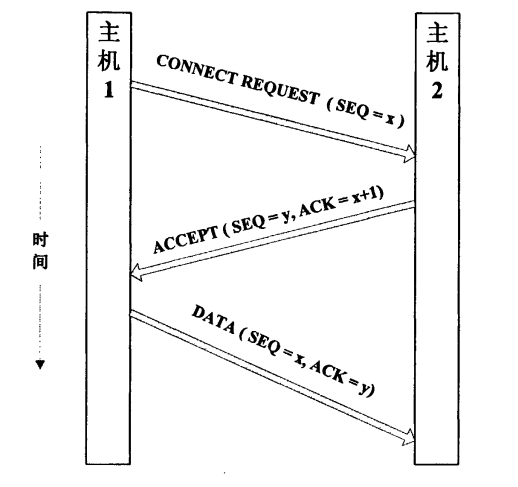
1. 客户端通过服务器的IP地址以及端口号与服务器连接；
2. 客户端构造带有应用层协议的TCP请求，其中包含了应用层的帧头等数据；

图 3‑4 TCP三次握手连接示意图

1. 客户端发送TCP请求到服务器，并等待服务器响应；
2. 服务器接收到请求，解析请求并处理，同时构造响应并发送至客户端；
3. 客户端收到响应后，一次事务处理完成。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。

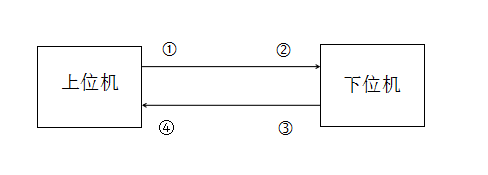


图 3‑5上下位机通信示意图

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）下位机接采用“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[34]，如表3-5所示。

表3-1 帧结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帧头 | LEN | FUN | DATA | CHECK |
| 1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，即多字节数据时，高字节数据在前，低字节数据在后传输，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+ACC DATA（加速度）+VELOCITY DATA（速度）+POS DATA（位移）+ 0x00 0x00 + 0x00 0x00+校验和，共32字节，ACC DATA/ VELOCITY DATA/ POS DATA/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)。
* 上位机发送PID数据给下位机的格式为：

PID数据格式和下位机发送给上位机的格式一样。

* 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+数据长度 +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 | 0x88 | 0x01 | LEN | int ACC\_DATA(加速度)\*100  int VEL\_DATA（速度）\*100  int POS\_DATA(位移）\*100 | SUM | 向上位机汇报当前状态，上位机接受并显示 |
| PID帧 | 0x88 | 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 | 0x88 | 0x03 | LEN | U16 waveform  U16 amplitude  U16 frequency | SUM | 汇报当前实验参数 |
| 限制参数帧 | 0x88 | 0x04 | LEN | U16 amp\_cons  U16 fre\_cons | SUM | 汇报当前限制参数 |
| 警告信息帧 | 0x88 | 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 | 0x88 | 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 | 0x80 | 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做的动作吗，具体格式还未定 |
| PID帧 | 0x80 | 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 | 0x80 | 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 | 0x80 | 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 | 0x80 | 0x05 | LEN | U16 amp\_cons  U16 fre\_cons | SUM | 上位机设定的限制参数 |
| 紧急停止帧 | 0x80 | 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TCP/IP通讯技术实现的。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[24][27]。

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：

QObject::connect(&button,&QPushButton::clicked,&QApplication::quit);

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

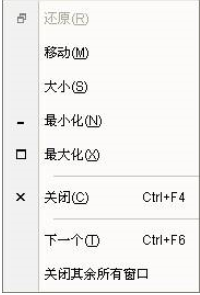
窗口列表显示当前所有打开的窗口，单击窗口名称按钮，可以使该窗口显示在最上层。在窗口名称按钮上右击，弹出如下浮动菜单：

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。

系统界面如图4-2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);

*if* (result)

{

qDebug()<<"PCI1716.ini配置文件写入成功！";

}

*else*

qDebug()<<"PCI1716.ini配置文件写入失败";

*delete* tmpHelper;

leftFrame = *new* QFrame(*this*);

leftFrame->setGeometry(4, 62, 660, 500);

leftFrame->setStyleSheet("background-color:white");

*//* *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\*)));

ui->labeStatusl->setStyleSheet("font:bold;font-size:12px;background-color:#7FFFD4");

ui->labeStatusl\_2->setStyleSheet("font:bold;font-size:12px;background-color:#7FFFD4");

currentLabel=*new* QLabel;

currentLabel->setMinimumSize(200,25);

*//currentLabel->setFrameShadow(QFrame::Sunken);*

currentLabel->setFrameShape(QFrame::*WinPanel*);

ui->statusBar->addWidget(currentLabel);

setWindowIcon(QPixmap(":/Icon/Icon/dashboard.png"));

ui->menuBar->setStyleSheet("background-color:#EEE9E9");

ui->toolBar->setStyleSheet("background-color:#C0C0C0");

ui->statusBar->setStyleSheet("background-color:#CCFFFF");

*//leftFrame->setStyleSheet(QString::fromUtf8("border:1px* *solid* *red"));*

int d7480rtn= Enc7480\_Init();

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

*if*(d7480rtn < 1)

{

qDebug()<<"初始化ENC7480计数卡失败!";

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();

waveformAiCtrl = WaveformAiCtrl::Create();

waveformAiCtrl->addDataReadyHandler(OnDataReadyEvent, *this*);

waveformAiCtrl->addOverrunHandler(OnOverRunEvent, *this*);

waveformAiCtrl->addCacheOverflowHandler(OnCacheOverflowEvent, *this*);

waveformAiCtrl->addStoppedHandler(OnStoppedEvent, *this*);

ConfigureDevice();

*//*

*//* *Initialize* *member* *variables*

*//*

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,*SIGNAL*(timeout()),*this*,*SLOT*(slotFuction()));

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

msCount=0;

}

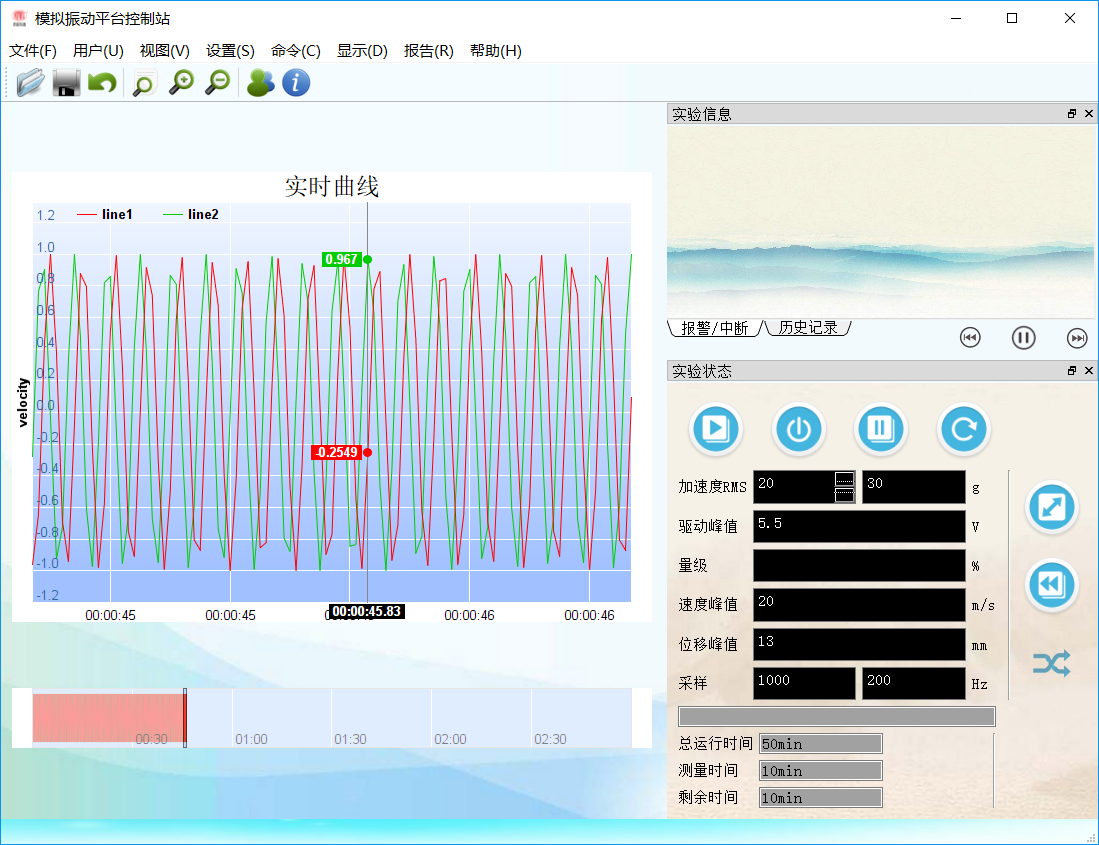


图 4‑2振动台上位机软件系统主界面

图 4‑2振动台上位机软件系统主界面

图 4‑2振动台上位机软件系统主界面

图 4‑2振动台上位机软件系统主界面

图 4‑2振动台上位机软件系统主界面

图 4‑2振动台上位机软件系统主界面

图 4‑2振动台上位机软件系统主界面

图 4‑2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

本次课程设计采用FPGA作为控制器。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，Qt中模拟振动平台上位机端服务器初始化程序如下： MYTcpServer\_v1::MYTcpServer\_v1(QWidget \*parent)

: QDialog(parent)

, ui(*new* Ui:: MYTcpServer)

{

ui->setupUi(this);

uk=uk\_1=ek=ek\_1=ek\_2=0;

sCurrent\_1=0;

mytcpPort = 8800;

static\_S\_Flag=false;

sineWave\_Flag=true;

MytcpServer = *new* QTcpServer(*this*);

IniHelper \*tmpHelper=new IniHelper;

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

if (result)

{

qDebug()<<"PCI1716.ini配置文件写入成功！";

}

else

qDebug()<<"PCI1716.ini配置文件写入失败";

delete tmpHelper;\*/

connect(MytcpServer,*SIGNAL*(newConnection()),*this*,*SLOT*(sendMessage());

initServer();

}

MYTcpServer\_v1::readFromPciIni(QString fileName)

{

ConfigureParameterPCI tmp;

ConfigureParameterAI tmpAI;

ConfigureParameterAO tmpAO;

if (fileName==NULL||fileName=="")

{

QMessageBox::information(NULL,"提示","文件名为空，请重新输入",QMessageBox::Ok|QMessageBox::Cancel);

qDebug()<<"Warning：Ini文件名为空！";

//return false;

}

configIniRead = new QSettings(fileName, QSettings::IniFormat);

//将读取到的ini文件保存在QString中，先取值，然后通过toString()函数转换成QString类型

tmpAO.deviceName = configIniRead->value("/ConfigureParameterAO/deviceName").toString();

tmpAO.channelStart= configIniRead->value("/ConfigureParameterAO/channelStart").toString().toInt();

tmpAO.channelCount= configIniRead->value("/ConfigureParameterAO/channelCount").toString().toInt();

tmpAO.valueRange=ValueRange(configIniRead->value("/ConfigureParameterAO/valueRange").toString().toInt());

tmpAI.deviceName = configIniRead->value("/ConfigureParameterAI/deviceName").toString();

tmpAI.channelStart= configIniRead->value("/ConfigureParameterAI/channelStart").toString().toInt();

tmpAI.channelCount= configIniRead->value("/ConfigureParameterAI/channelCount").toString().toInt();

tmpAI.valueRange=ValueRange(configIniRead->value("/ConfigureParameterAI/valueRange").toString().toInt());

tmpAI.clockRatePerChan= configIniRead->value("/ConfigureParameterAI/clockRatePerChan").toString().toInt();

tmpAI.sectionLength= configIniRead->value("/ConfigureParameterAI/sectionLength").toString().toInt();

tmp.iniSettingAO=tmpAO;

tmp.iniSettingAI=tmpAI;

//打印得到的结果

//qDebug()<<tmp.iniSettingAO.deviceName<<tmp.iniSettingAO.valueRange;

//qDebug() << tmp.iniSettingAI.channelStart<<tmp.iniSettingAI.channelCount<<tmp.iniSettingAI.sectionLength;

//读入入完成后删除指针

delete configIniRead;

return tmp;

}

MYTcpServer\_v1::readFromCtrlIni(QString fileName)

{

TPIDInfo tmp;

if (fileName==NULL||fileName=="")

{

QMessageBox::information(NULL,"提示","文件名为空，请重新输入",QMessageBox::Ok|QMessageBox::Cancel);

qDebug()<<"Warning：Ini文件名为空！";

}

configIniRead = new QSettings(fileName, QSettings::IniFormat);

tmp.SP = configIniRead->value("/SPID/P").toString().toDouble();

tmp.SI = configIniRead->value("/SPID/I").toString().toDouble();

tmp.SD = configIniRead->value("/SPID/D").toString().toDouble();

delete configIniRead;

return tmp;

}

bool IniHelper::writeToPCIIni(QString fileName,ConfigureParameterPCI tmp)

{

if (fileName==NULL||fileName=="")

{

QMessageBox::information(NULL,"提示","文件名为空，请重新输入",QMessageBox::Ok|QMessageBox::Cancel);

qDebug()<<"Warning：Ini文件名为空！";

return false;

}

configIniWrite = new QSettings(fileName, QSettings::IniFormat);

ConfigureParameterAO tmpAO=tmp.iniSettingAO;

configIniWrite->beginGroup("ConfigureParameterAO");

configIniWrite->setValue("deviceName", tmpAO.deviceName);

configIniWrite->setValue("channelStart",tmpAO.channelStart);

configIniWrite->setValue("channelCount",tmpAO.channelCount);

configIniWrite->setValue("valueRange", tmpAO.valueRange);

configIniWrite->endGroup();

ConfigureParameterAI tmpAI=tmp.iniSettingAI;

configIniWrite->beginGroup("ConfigureParameterAI");

configIniWrite->setValue("deviceName", tmpAI.deviceName);

configIniWrite->setValue("channelStart",tmpAI.channelStart);

configIniWrite->setValue("channelCount",tmpAI.channelCount);

configIniWrite->setValue("valueRange", tmpAI.valueRange);

configIniWrite->setValue("clockRatePerChan", tmpAI.clockRatePerChan);

configIniWrite->setValue("sectionLength", tmpAI.sectionLength);

configIniWrite->endGroup();

delete configIniWrite;

return true;

}

4.2.2上位机通信设计

1.并发服务器设置

在进行数据通讯时，模拟振动台上位机端服务器可能同时需要连接多个控制器客户端，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[25]。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[32][31]。从而可以实现同时与多个客户端通信的功能,结构图如图4-3所示：

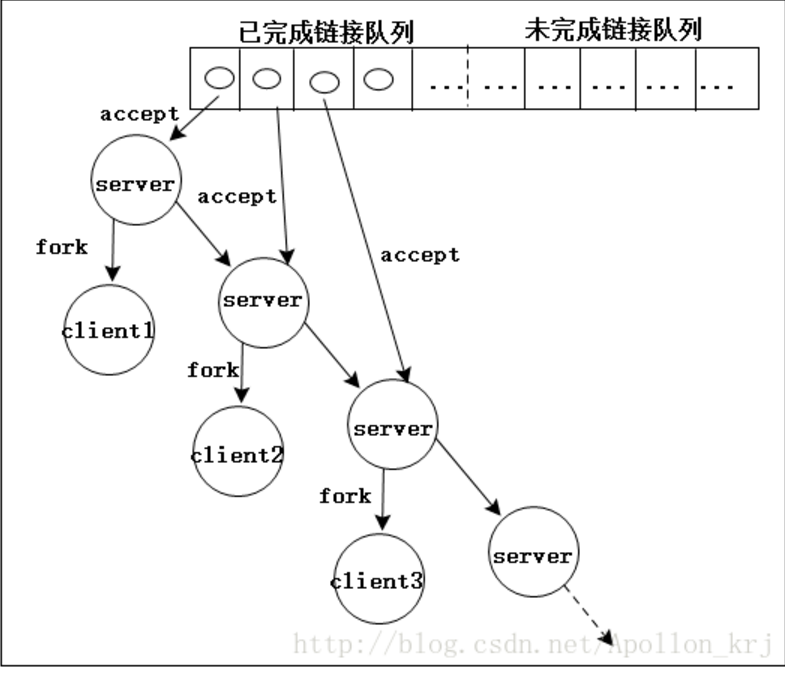


图 4‑3并发服务器结构图

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

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[34]：

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0x88；
  + 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。

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，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("Warning Information"),

tr("No device to support the currently demonstrated function!"));

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;

qDebug()<<"the configureParameterAO is :"<<configureAO.deviceName<<configureAO.channelStart<<configureAO.channelCount<<configureAO.valueRange;

*//将Ini文件中默认的配置显示出来*

void ConfigureDialog::SetConfigureParameterAO(ConfigureParameterAO tmpConfigure,ConfigureParameterAI tmpConfigureAI)

{

configure=tmpConfigure;

ui.cmbDevice->setCurrentIndex(0);

ui.cmbChannelStart->setCurrentIndex(0);

ui.cmbChannelCount->setCurrentIndex(0);

ui.cmbValueRange->setCurrentIndex(2);

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

configureAI=tmpConfigureAI;

ui.cmbChannelStart\_AI->setCurrentIndex(0);

ui.cmbChannelCount\_AI->setCurrentIndex(0);

ui.cmbValueRange\_AI->setCurrentIndex(4);

ui.edtClockRatePerChan\_AI->setText(QString::number(configureAI.clockRatePerChan));

ui.edtSectionLength\_AI->setText(QString::number(configureAI.sectionLength));

}

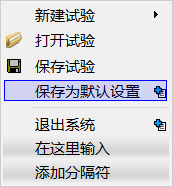
功能界面如下所示,以及保存文件如下所示：

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

图 4‑5配置文件

图 4‑4保存设置菜单栏

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、限制餐宿等等，为数据处理提供参考依据。

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，并且重写了他们的信号槽函数：

****

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 4‑6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

图 4‑生产者-消费者模式图

图 4‑‑生产者-消费者模式图

图 4‑10生产者-消费者算法参数图 4‑11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 4‑12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\_action\_ChannelParameters\_triggered()

{

*//首先进入构造函数*

ConfigureDialog dialog;

dialog.SetConfigureParameterAO(configureAO,configureAI);

int resultDialog = dialog.*exec*();

*//将主程序中的configureAO传入到设置界面*

*//qDebug()<<"the* *configureParameter* *is* *:"<<configureAO.deviceName<<configureAO.channelStart<<configureAO.channelCount;*

*if* (resultDialog == QDialog::*Rejected*)

{

*//exit(0);*

}

*else* *if*(resultDialog == QDialog::*Accepted*)

{

SetConfigureParameterAO(dialog.GetConfigureParameter());\

SetConfigureParameterAI(dialog.GetConfigureParameterAI());

qDebug()<<"the data from configureDialog is :";

qDebug()<<"haha"<<configureAO.deviceName<<configureAO.channelStart<<configureAO.channelCount<<configureAO.valueRange;

qDebug()<<"haha"<<configureAI.deviceName<<configureAI.channelStart<<configureAI.channelCount<<configureAI.valueRange<<configureAI.clockRatePerChan<<configureAI.sectionLength;

Initialize();

}

}

void MainWindow::on\_btnStart\_clicked()

{

double AoValue;

*if* (ui->btnStart->isFlat())

{

ui->btnStart->setFlat(*false*);

ui->btnStart->setStyleSheet("background-color:green");

}

*else*

ui->btnStart->setFlat(*true*);

*//PID静态位移控制开始*

msStartCount=0;

*//AO*

AoValue=ui->lineEdit->text().toDouble();

*//qDebug()<<"输出电压"<<AoValue;*

*if*(AoValue<=-10||AoValue>=10)

{

QMessageBox::information(NULL,"提示","Too large Voltage!",QMessageBox::*Ok*|QMessageBox::*Cancel*);

exit;

}

*for* (int i = 0; i < 2; i++)

dataScaled[i]=AoValue;

*//AI*

ErrorCode errorCode = waveformAiCtrl->Start();

CheckError(errorCode);

startFlag=*true*;

}

void MainWindow::on\_btnStop\_clicked()

{

*if* (ui->btnStop->isFlat())*//没按*

{

ui->btnStop->setFlat(*false*);

ui->btnStop->setStyleSheet("background-color:green");

}

*else*

{

ui->btnStop->setFlat(*true*);

}

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

timer->stop();

dataScaled[0]=0;

dataScaled[1]=0;

ErrorCode errorCode = *Success*;

errorCode = instantAoCtrl->Write(configureAO.channelStart,configureAO.channelCount, dataScaled);

CheckError(errorCode);

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

errorCode = waveformAiCtrl->Stop();

CheckError(errorCode);

ui->lineEdit->setText("0");

}

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

{

CtrlDialog dialog;

dialog.setPIDParameter(sPIDInfo);

int resultDialog = dialog.*exec*();

*if* (resultDialog == QDialog::*Rejected*)

{

*//exit(0);*

}

*else* *if*(resultDialog == QDialog::*Accepted*)

{

sPIDInfo=dialog.getPIDParameter();

qDebug()<<"PID 参数是:"<<sPIDInfo.SP<<sPIDInfo.SI<<sPIDInfo.SD;

}

}

void MainWindow::on\_pushButton\_clicked()

{

*for* (int i = 0; i < 2; i++)

dataScaled[i]=ui->lineEdit->text().toDouble();

ErrorCode errorCode = *Success*;

errorCode = instantAoCtrl->Write(configureAO.channelStart,configureAO.channelCount, dataScaled);

CheckError(errorCode);

}

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。

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* *variables*

*//*

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,*SIGNAL*(timeout()),*this*,*SLOT*(slotFuction()));

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

4.4.2 生产者-消费者矛盾

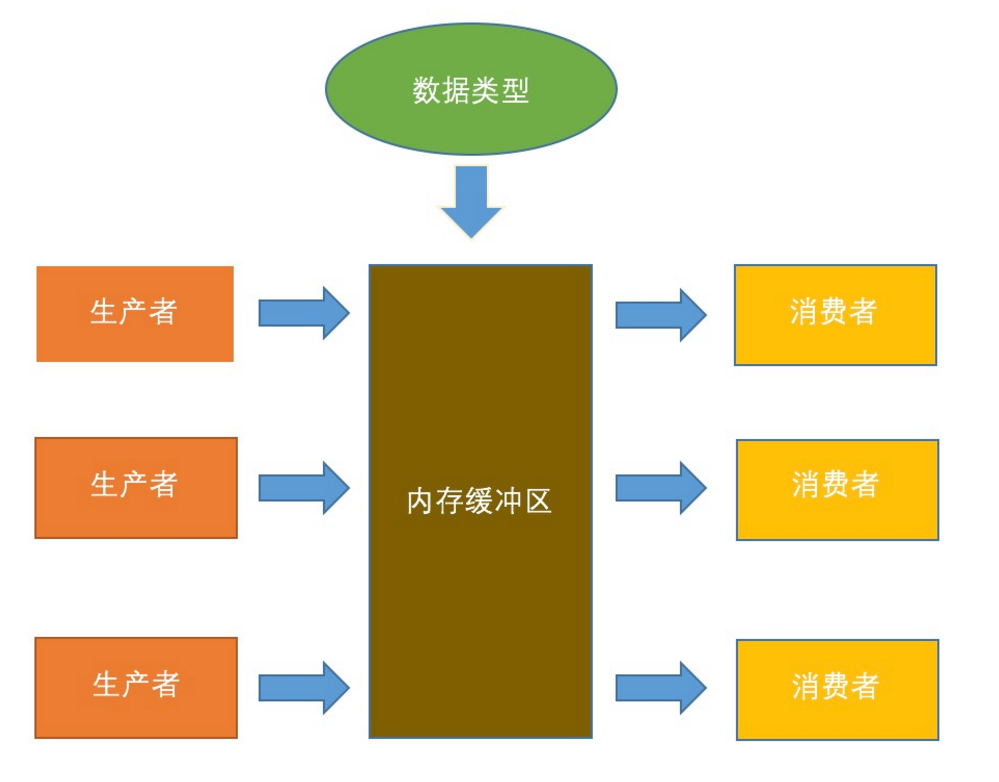
 进行动态绘图的过程中，会碰到一个问题，一种情况是下位机在不停产生数据并上传，此时上位机在处理一些其他的任务，那么会导致数据溢出，并在绘图过程中产生闪烁中断等问题。而如果下位机上传数据的速度不够快，则会导致上位机不能正确绘图，并且容易引起系统崩溃。我们把这个称为“生产者-消费者矛盾” [16]。而动态绘图过程中必须解决这个生产和消费速度不匹配的矛盾，从而解决这个并发读写的问题，生产消费者模式如下图所示：

图 4‑7生产者-消费者模式图

图 4‑‑生产者-消费者模式图

图 4‑生产者-消费者算法参数图 4‑生产者-消费者模式图

图 4‑‑生产者-消费者模式图

图 4‑17生产者-消费者算法参数

图 4‑0‑18生产者-消费者算法参数

图 4‑ 状态图 4‑生产者-消费者算法参数图 4‑生产者-消费者模式图

图 4‑‑生产者-消费者模式图

图 4‑生产者-消费者算法参数图 4‑生产者-消费者模式图

图 4‑‑生产者-消费者模式图

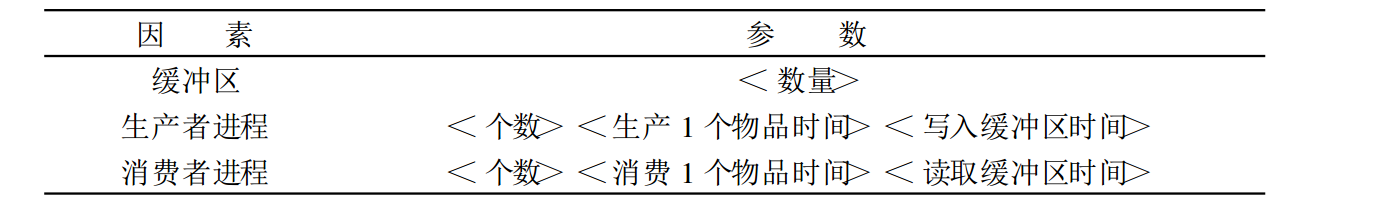


图 4‑8生产者-消费者算法参数

图 4‑0‑26生产者-消费者算法参数

图 4‑27 状态图 4‑28生产者-消费者算法参数

图 4‑0‑29生产者-消费者算法参数

图 4‑ 状态

图 4‑‑ 状态

图 4‑32双缓冲区机制图 4‑33 状态图 4‑34生产者-消费者算法参数

图 4‑0‑35生产者-消费者算法参数

图 4‑36 状态图 4‑37生产者-消费者算法参数

图 4‑0‑38生产者-消费者算法参数

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[20]：

其中表示消费者当前状态，表示生产者当前状态，表示缓冲区当前状态。

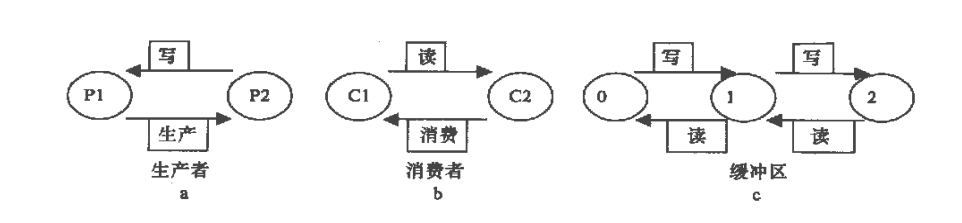
由于整个系统是异步进行的，当状态=时则会发生线程冲突，根据状态表此时，只能支持整个系统互斥访问。当生产者访问缓冲区时，将封锁消费者的线程，当消费者访问缓冲区时，将封锁生产者线程。如此将导致程序的运行效率很低，而且整个系统的实时性不能保证。

图 4‑9 状态

图 4‑‑ 状态

图 4‑双缓冲区机制图 4‑ 状态

图 4‑‑ 状态

图 4‑43双缓冲区机制

图 4‑ 状态

图 4‑‑ 状态

图 4‑双缓冲区机制图 4‑ 状态

图 4‑‑ 状态

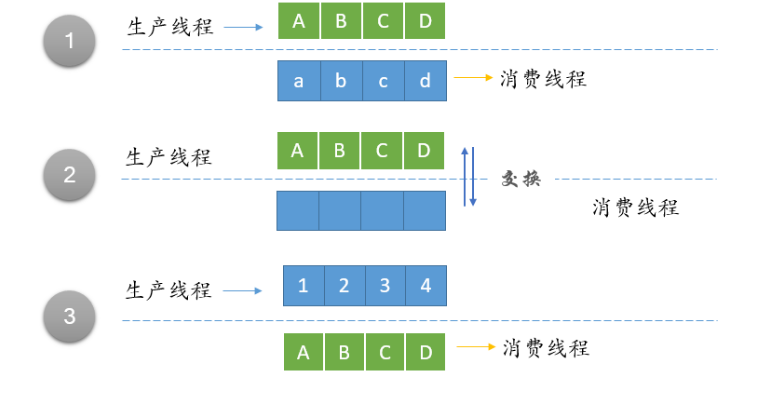
解决这个问题可以采用开辟双缓冲区，实现绘图过程中的读写分离。通过添加互斥信号量对缓冲区进行读写保护，当某一个缓冲区作为下位机发送数据的缓冲区时，请求信号量，进行互斥保护，此时上位机读取另外一个缓冲区进行绘图，完成操作后，释放信号量，读写缓冲区交换，由于数据的读写分离，并且有互斥信号量的读写保护，所以可以解决其速度不匹配的问题

图 4‑10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;

int bufferLen;*//当前指针*

int bufferSize;*//缓冲区总的长度*

QMutex mutex;

};

1. 系统调试及测试

模拟振动平台上位机软件系统主要包括系统管理模块、数据处理模块、通讯模块、数据显示与保存模块等四个模块[3][34]。现从软件的实际使用中，测试本系统各个模块功能的实现情况。

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

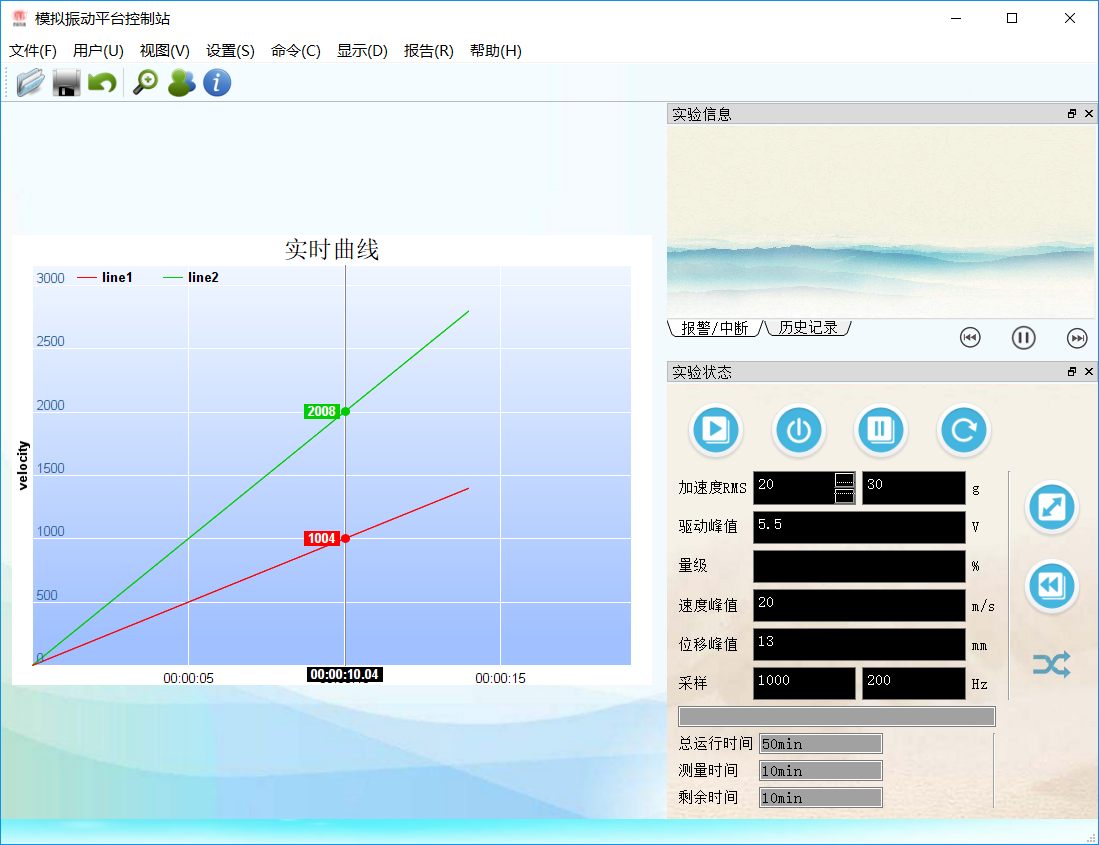


图 5‑1振动平台上位机管理系统主界面

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 5‑2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

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。

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,用户登录的菜单界面如下所示

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

图 5‑3用户登录菜单



用户登录界面如下:

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

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。通过管理员对用户的权限管理，可以提高振动平台实验时的安全性。

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 5‑5设置菜单

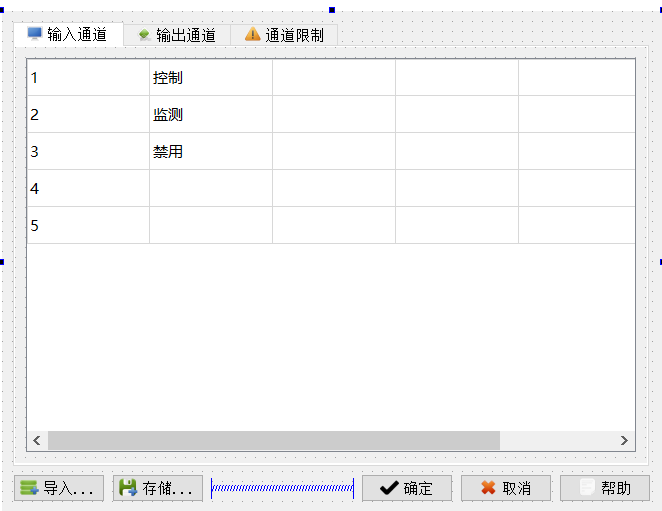


图 5‑6通道参数界面

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

图 5‑6通道参数界面

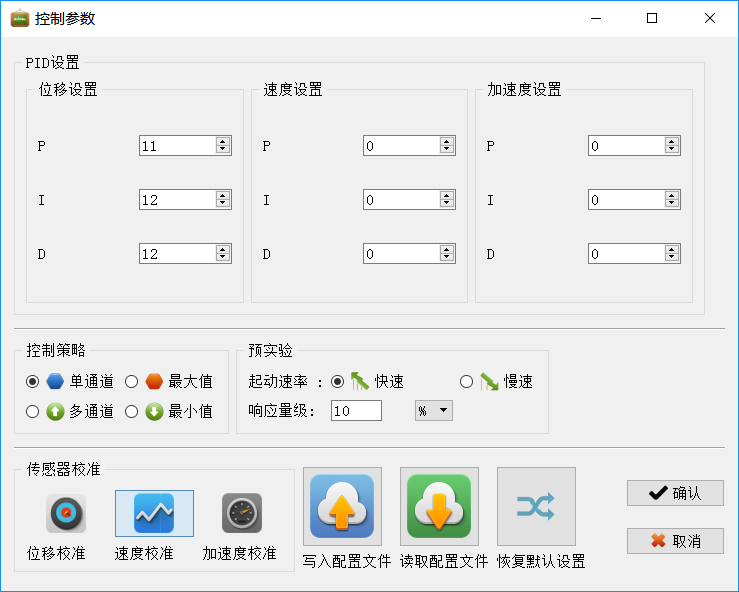
 主界面中点击“设置”—-“控制参数”，即可打开控制参数设置界面，控制参数是用来定义与试验运行、控制相关的参数。例如，控制策略、量级、预实验和高级设置等等，如下图所示:

图 5‑7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

图 5‑7控制参数设置界面

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,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。信号曲线实时绘制界面如图所示：

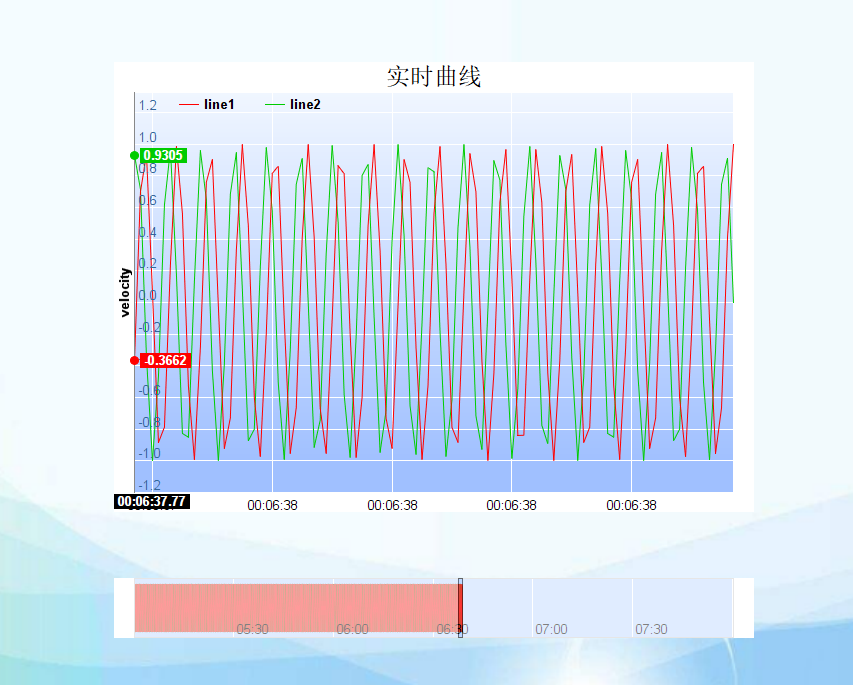


图 5‑8三参量曲线实时显示界面以及全局状态栏

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

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。本文首先分析了该课题研究背景以及意义。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，将系统划分为系统管理模块、系统通信模块、数据处理模块、数据显示模块等四个方面。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文件.ini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[31]。
* 本课题是电脑端通过以太网与下位机直接连接，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[34]

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. 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
2. 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
3. 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
4. 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
5. 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
6. 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
7. 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.
8. 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.
9. 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.
10. 陆文周 Qt5开发及实例 [M] 电子工业出版社 2014
11. 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
12. 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
13. 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
14. 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
15. 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.
16. 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.
17. 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.
18. 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.
19. 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
20. 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.
21. 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.
22. 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.
23. 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.
24. 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.
25. 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.
26. 钱琳. 嵌入式监控系统上位机软件设计[D].浙江大学,2008.
27. 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.
28. 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.
29. 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.
30. 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.
31. 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.
32. 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.
33. 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.
34. 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.
35. 王伟. 两自由度液压振动台设计[D].哈尔滨工业大学,2011.
36. 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

致谢

写到这里差不多结束了，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。

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，使他们的言传身教，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。本课题的完成，同样离不开浙江大学网络传感与控制研究组史治国、贺诗波老师的耐心指导。他们言传身教，我在浙大期间他们给予了无数的指导和关心。

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，再踏进那书院的门栏。但愿下次再与湖大相遇，仿佛不曾走远。