

文章编号: 1006-544X (2007) 01-140-03

基于 C/S 模式的倒立摆远程实验系统设计与分析

张文革, 牛秦洲

(桂林工学院 电子与计算机系, 广西 桂林 541004)

摘要: 为实现资源共享和便于使用, 提出了一种基于 C/S 模式的倒立摆远程实验系统实现方案. 实验系统的整体结构采用 3 层架构模式. 软件主要包括 3 部分: 倒立摆控制模块、网络通信与用户信息管理模块、用户界面模块. 网络通信与用户信息管理模块采用 Socket 和 ADO 开发技术实现网络通信和对用户信息数据库的管理与访问; 用户界面模块采用 MFC 文档视图类结构实现用户与实验系统的交互, 分析和解决了倒立摆控制模块中的中断处理问题.

关键词: C/S 模式; Socket; ADO; 远程实验系统

中图分类号: TP 393.01

文献标志码: A

倒立摆是一种理想的自动控制教学实验仪器, 许多抽象的控制理论概念如系统稳定性、可控性和系统抗干扰能力等, 都可以通过倒立摆系统实验直观地表现出来. 同时, 倒立摆系统也是进行控制理论研究的理想实验平台. 由于其本身所具有的高阶次、不稳定、多变量、非线性和强耦合特性, 许多现代控制理论的研究人员一直将它视为典型的研究对象, 不断从中发掘出新的控制策略和控制方法, 相关的科研成果在航天科技和机器人学方面获得了广泛应用.

为便于教学及控制理论研究, 实现资源共享, 本文提出一种基于 C/S 模式的倒立摆远程实验平台搭建方案. 与 B/S 模式相比, C/S 模式具有技术成熟、开发周期短、软件功能丰富的优点.

1 实验系统设计

1.1 倒立摆装置

倒立摆装置包含倒立摆本体、电控箱及由运动控制卡和普通 PC 机组成的控制平台等三大部分. 各部分间连接方式如下: PC 机通过运动控制卡与电控箱相连, 电控箱与倒立摆本体相连. 其中倒立摆本体包括连杆、摆杆的连接关节. 连杆和摆杆的角度、角速度信号通过光电编码盘反馈

给伺服驱动器和运动控制卡. PC 机从运动控制卡中读取数据, 确定控制决策, 并由运动控制卡来实现该决策, 产生相应的控制量, 驱动电机转动, 带动连杆运动, 保持摆杆的平衡.

1.2 远程实验系统架构

系统采用现场控制端 + 中间服务器 + 客户端的 3 层架构模式. 客户端响应用户请求并处理和显示反馈的数据, 控制端负责对倒立摆设备的控制和采集实时数据, 中间服务器负责在两者之间建立连接.

这种架构模式可将客户端的身份验证和信息管理放在中间服务器来完成, 使系统在功能划分上更为清晰. 同时, 采用中间服务器作为数据通道的方式, 可将控制端和客户端隔开. 通过在中间服务器上安装防火墙, 可有效提高系统的安全与可靠性.

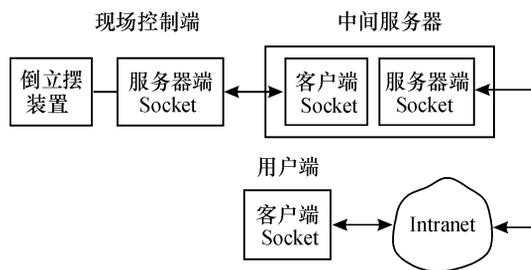


图 1 倒立摆远程实验系统结构

Fig. 1 System structure of inverted pendulum experiment

收稿日期: 2005-11-29

基金项目: 广西区科技攻关项目 (桂科攻 0428002-1)

作者简介: 张文革 (1978-), 男, 硕士研究生, 研究方向: 计算机网络.

通讯作者: 牛秦洲, 男, 博士, 教授, E-mail: dixn@glite.edu.cn.

1.3 软件设计

倒立摆远程实验软件系统主要包括 3 部分: 倒立摆控制模块、网络通信与用户信息管理模块、用户界面模块. 倒立摆装置采用固高公司的 GIP 系列倒立摆. 软件采用 Visual C++ 6.0 设计, 在 Windows 2000 环境下运行.

1.3.1 倒立摆控制模块 倒立摆控制模块工作在现场控制端, 其主要任务是根据控制卡采集的数据对摆进行控制. 程序设计采用固高公司提供的相应动态链接库文件——GT400.h、GT400.lib、GT400.dll, 以及对倒立摆进行控制的函数. 函数主要包括:

```
sv_init():控制卡初始化;
enable_servo():使能伺服,修改控制模式;
first_move():启动倒立摆;
handle_safety():系统异常处理函数;
linear_up():小车运动控制函数;
anti_cran():根据摆状态进行控制输出;
reset_car():小车复位;
handle_pendulum():摆主控函数;
```

其中, handle_pendulum() 是摆主控函数, 它调用其它控制函数, 实现对小车和摆运动的控制. 其主要代码如下:

```
switch( start )
{
case 1:
    enable_servo(); //使能伺服,并修改运动控制模式到速度控制模式
    start = 2; //进入控制状态
    return;
case 2:
    if (.....)根据控制卡采集的摆杆角度数据,判断倒立摆是否处在平衡位置
        anti_cran(); //如果平衡,则计算保持在该平衡位置所需的控制量
        break;
default:
    break;
}
```

anti_cran() 是根据状态反馈计算控制输出的函数. 如果要加入自己的算法到系统中, 可以直接修改该函数.

1.3.2 网络通信与用户信息管理模块 网络通信模块设计包括服务器端和客户端 Socket 的建立与连接、Socket 间数据的收发以及收发数据完毕关闭 Socket 连接. 由于程序中通过网络传输的数据格式较为复杂, 因此采用串行化方法^[2], 定义一个继承于 CObject 的数据类, 将要传输的数据定义

为其成员变量, 然后通过串行化传送和接收数据.

用户信息管理模块工作在实验系统架构中的中间服务器, 其主要任务是对客户端的连接请求进行验证. 模块在中间服务器创建并维护一个用户信息数据库. 当接收到客户端的连接请求时, 程序在数据库查询相应记录, 对用户身份进行验证, 若是合法用户则与现场控制端建立连接, 否则返回错误信息. 为了实现对数据库的管理, 还需要在中间服务器实现对数据库的操作, 包括添加和删除用户、修改用户密码等. 用户信息数据库采用 SQL Server 2000, 数据库系统开发采用 ADO 技术, 它是基于 OLE DB 的访问接口, 具有易于使用、通用性和灵活性好、访问数据源效率高、编程接口丰富的特点^[3].

1.3.3 用户界面模块 用户界面模块实现用户与实验系统的交互. 模块采用 MFC 文档视图类结构. 用户首先在客户端通过用户名和密码进行登陆. 若登陆信息被中间服务器验证通过, 则建立与现场控制端的连接. 然后, 用户在客户端对控制算法的参数进行设置, 参数通过网络传递给现场控制端. 参数设置完成后, 用户可以向现场控制端发送电机开始伺服和起摆的命令. 命令以自定义编码的形式通过网络传递到现场控制端, 现场控制端执行相应的命令. 电机开始伺服后, 摆的实时数据由运动控制卡采集, 通过网络传回用户端. 用户界面模块利用传回的数据, 在窗口中绘制摆的实物图和坐标图, 并设置定时器对窗口定时刷新. 这样, 用户在客户端能直观地看到摆的运动情况, 达到远程实验的目的.

2 软件设计

倒立摆系统中, 主机对倒立摆的控制采用的是中断控制的方式, 即由运动控制器向主机发出中断请求, 主机响应该中断请求, 对摆运动过程中出现的事件做出处理. 在实际的软件控制模块设计中, 采用一种定时中断的机制, 即通过设置定时器, 周期性地产生中断, 然后由主机调用控制模块中的主控函数——handle_pendulum(), 对摆的运动状态进行定时调整.

为了实现中断处理, 需要修改设备中断向量, 并挂接自己的中断服务程序 (interrupt service routine, ISR). 但在 Windows 环境中, 操作系统将系

统内核和应用程序进行隔离,不允许用户层修改设备的ISR,为此采用一种事件同步的机制来实现所需功能.其原理是让设备ISR和用户程序共享同一事件,实现用户程序和设备ISR的同步,从而让用户程序能得知硬件设备的中断请求.具体步骤是在用户程序中创建一个同步事件,利用控制器驱动程序提供的API函数向设备ISR设置同步事件.此后,用户程序和设备ISR就是两个共享同步事件的普通进程.用户程序为了不阻塞自己,应启动一个新的线程,在该新线程中调用WaitForSingleObject()等待事件有效.一旦设备产生中断,设备ISR激活中断同步事件,用户程序就会从WaitForSingleObject()处被激活,开始执行线程的后续部分,这样就实现了Windows环境的中断响应.

程序运行时受系统效率和中断频率的影响,并不能保证每次中断都能激活用户线程,当中断频率很高,而操作系统线程调度较慢时,有可能出现多次中断堆叠,不能保证中断的及时响应.因此在编程中应注意设置合适的中断时间.

下面是在Windows环境下进行中断处理的主要代码:

```
//以下是创建同步事件,并挂接中断服务线程
//创建一同步事件
hSyncEvent = CreateEvent ( NULL, true, false,
NULL);
//将应用程序的事件与底层中断相关连
rtn = GT_SetIntSyncEvent (hSyncEvent);
//中断时间为10个控制周期,应根据实际情况进行
设置
rtn = GT_SetIntrTm(10);
//中断模式设为定时中断
rtn = GT_TmrIntr();
//开设中断服务线程
```

```
DWORD idSubThread;
hSubThread = CreateThread ( NULL, 0, intproc, //
中断处理函数名
NULL, //中断处理函数参数
0, &idSubThread);
//以下是中断处理函数 intproc 中的主要内容
while(1)
{
//等待中断发生
WaitForSingleObject(hSyncEvent, INFINITE);
//添加中断处理代码,handle_pendulum()是主控制
函数
handle_pendulum();
//复位事件状态
ResetEvent(hSyncEvent);
}
//通知设备ISR释放事件
rtn = GT_SetIntSyncEvent( NULL);
//关闭事件句柄
CloseHandle(hSyncEvent);
//退出线程
ExitThread(0);
```

3 结 语

本文介绍了一种基于C/S模式的倒立摆远程实验系统的实现方案.实验者可以通过远程登陆的方式,在倒立摆装置上进行实验.这样既有效地实现了资源共享,又大大方便了教学和研究工作,在实际应用中拥有较为广阔的前景.

参考文献:

- [1] 林光云,王万良. Java技术在基于Web分布式倒立摆远程实验的应用[J]. 中国有线电视, 2004(9/10): 90-93.
- [2] 丁展,刘海英. Visual C++网络通信编程实用案例精选[M]. 北京:人民邮电出版社, 2004.
- [3] 李闯,吴继刚,周学明. Visual C++数据库系统开发实例导航[M]. 北京:人民邮电出版社, 2003.

C/S Mode-based Inverted Pendulum Experimental System

ZHANG Wen-ge, NIU Qin-zhou

(Department of Electronics and Computer Science, Guilin University of Technology, Guilin 541004, China)

Abstract: In order to realize resources sharing and convenient practice, a scheme of the C/S mode-based inverted pendulum experiment system is put forward. The frame of experimental system adopts three layers mode. Software includes three parts: inverted pendulum control module, network communication and user information manage module, user interface module. Network communication and user information manage module adopt Socket and ADO development technique to achieve network communication and database managing and accessing. User interface module adopts MFC document and view structure to complete the interaction between users and experimental system. Finally, the interruption handling problem is settled in inverted pendulum control module.

Key words: C/S mode; socket; ado; long-distance experimental system