

# “基于 Windows 的堆垛机实时控制”软件设计

张爱红

(无锡职业技术学院 机电系, 江苏 无锡 214121)

**摘要:** 介绍了堆垛机微机控制系统的硬件构成, 并从“自动”、“手动”两方面, 根据堆垛机行走、伸缩、升降等控制功能的要求, 开发了基于 Windows 2000 Server 操作系统的堆垛机实时控制软件, 对各模块进行了详细的解析。

**关键词:** 堆垛机; 无线局域网; 多线程

**中图分类号:** TP273+.5

**文献标识码:** A

**文章编号:** 1672-0318 (2006) 02-0017-03

随着计算机网络技术的快速发展, 无线网络得到了越来越多的应用, 与有线网络相比, 无线网络的优势是显而易见的, 主要表现在安装、维修简单方便, 对建筑物的破坏少、无需传输介质等方面。作者将无线局域网技术应用到柔性制造系统(堆垛机)控制软件的开发中, 开发了基于 Windows 2000 操作系统的控制软件, 并从手动、自动两方面实现了对堆垛机的无线控制。

## 1 堆垛机控制系统的硬件组成

堆垛机作为柔性制造系统自动化立体仓库的重要组成部分, 它是实现整个仓库系统“自动”控制功能的关键设备, 它的控制系统可以有多种不同的实现模式, 例如采用 PLC 作为控制器, 编写梯形图控制程序, 实现对堆垛机的行走、升降、伸缩等动作的控制<sup>[1]</sup>。除此以外, 基于 IPC 的堆垛机控制器、数据采集卡、无线网络(无线网卡、无线接入点)等技术的综合运用, 将整个控制系统与有线网络进行了无缝集成, 更加利于分布式控制系统的实现, 系统构成如图 1 所示。

## 2 堆垛机控制软件的开发

本系统采用工业控制微机(IPC)作为主控制器, 安装了 Windows 2000 (Server 版) 操作系统, 控制系统的开发平台为 Visual C++ 6.0, 使用

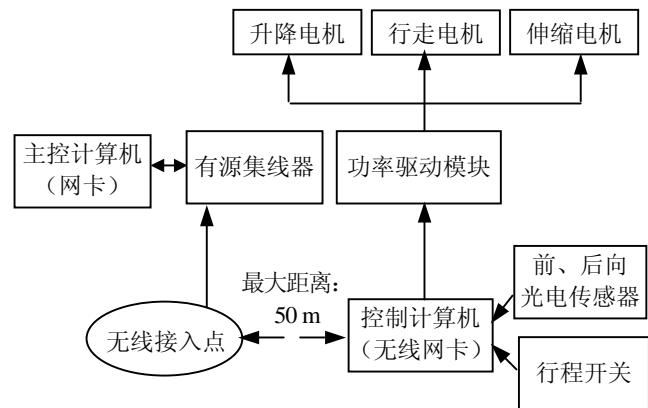


图 1 堆垛机系统原理图

MFC 进行程序开发。

在系统程序的编写过程中, 考虑到 Windows 系统是抢先式的多任务操作系统, 控制程序也就是一个进程, 它有主线程, 还有其他线程, 而线程占用 CPU 的时间由操作系统决定, 由于 CPU 运行速度快, 不同线程曾现出“并行”工作的状态。因此在实时控制系统中, 采用多线程技术编程可以达到简化程序结构、提高程序响应实时操作的能力。

VC++ 6.0 版本中 MFC 应用程序的线程由 CWinThread 对象来表示, 分为 2 种: 用户界面线程和工作者线程。其中用户界面线程能够提供界面与用户交互, 用于处理用户输入和响应各种信息和事件, 工作者线程主要用来处理应用程序的后台任务。基于此, 堆垛机控制系统建立在多线程技术基础上,

**收稿日期:** 2006-03-17

**作者简介:** 张爱红(1971-), 男, 江苏盐城人, 副教授, 硕士, 研究方向: 机器人、数控等机电一体化技术。

除了主线程外,添加一个“实时状态检测”的工作者线程。

## 2.1 程序的总体框架

堆垛机需要控制的动作包括:行走、升降、伸缩,均由直流电机驱动,由于地面没有固定的轨道,需采用寻迹、差动调速方式,因此行走控制难度较大,相对而言,升降、伸缩控制比较简单,可靠性也较高。

堆垛机的动作控制都在工作者线程(下称:循环检测线程)内完成。程序中循环检测线程定义为: `UINT MyControllingFunction( LPVOID pParam )`,由主线程响应“自动运行”菜单后调用 `AfxBeginThread( MyControllingFunction,this)` 起;为了能够调用视图类中的成员函数或变量,在循环函数体中添加以下代码: `CTest_timeView *pView=(CTest_timeView*)pParam`,得到指向视图类的指针 `pView`,然后通过 `pView->ReadState()` 调用视图类中的自定义函数 `ReadState()`,实时读取前、后向光电传感器的位置信号,并根据信号的变化,控制堆垛机的差动行走。

在循环检测体中,除了实时检测光电传感器状态的程序段外,还包括实时检测堆垛机行走启、停控制信号的程序段,以及升降、伸缩叉机构启、停信号,并根据信号的变化实时向主线程发送消息,在消息响应函数中控制堆垛机的升降、伸缩。

控制系统除了能够实现“自动”运行之外,还能够响应“手动”操作。“手动”控制编程的思路与“自动”编程思路不同,无需在其他线程中完成,主要通过定义键盘按键消息响应来实现。为了实现控制过程的实时性,程序中起动了55毫秒定时响应函数。调用的函数包括: `SetTimer(1,55,NULL)`, `void CTest_timeView::OnTimer(UINT nIDEvent)`, `void CTest_timeView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)`。程序起后选择手动操作,如果按下字符‘u’或‘U’时,堆垛机的铲叉将上升,堆垛机铲叉下降、伸缩、行走等动作的实现原理与“上升”过程相同。

## 2.2 编程初始化操作

ADLINK PCI-7296 数据采集卡是 Opto-22 兼

容的96通道DIO卡,提供了在VC,VB环境下的开发编程函数包,这些函数包括: `Register_Card`, `DIO_PortConfig`, `DO_WritePort`, `DI_ReadPort`, `Release_Card`, 可以获取卡的标识、定义端口属性、完成对端口读、写操作以及采集卡释放等。在使用端口读写函数之前必须完成初始化操作,否则程序调试过程中将频频出错。具体操作包括:①复制 `dask.h` 到 VC++软件<include>目录;②复制静态库文件“`pci-dask.lib`”到 VC++软件<lib>目录;③点击“工程”菜单—“设置”子菜单,选择“link”选项后在“对象、库模块”栏添加“`pci-dask.lib`”文件;④在开发的程序中包含<`dask.h`>头文件等。

## 2.3 微妙级定时模块的编写

为了能够控制堆垛机运行平稳、寻迹可靠,必须提高直流电机(行走电机)控制方波的频率,如果用VC++提供的 `Sleep` 函数是无法实现的,因为它的精度局限于毫秒级。为此,必须寻找一个更高精度的延时方法,研究发现,Windows内部有一个精度很高的定时器,能满足控制要求。

对于不同系统、不同硬件配置(CPU)的计算机,其定时器的频率也不同。为了取得这个定时器的频率及定时器的当前值需分别调用应用程序接口(简称:API)函数: `QueryPerformanceFrequency`、`QueryPerformanceCounter`。控制程序根据需延时的时间和定时器频率,算出经过延时时间定时器需要经过的周期数,通过循环调用 `QueryPerformanceCounter` 不停地读出定时器值,指定的周期数一到将结束循环,这样就达到了高精度延时的目的,程序如下:

```
void CTest_timeView::OnBTest(UINT time)
{
    LARGE_INTEGER
    frequency,test,previous,current;
    //首先取得高精度定时器的频率
    if(!QueryPerformanceFrequency( &frequency))
    {
        AfxMessageBox(" 硬 件 不 支 持!");return;
    }
}
```

```

//将微秒数换算成计数值（与 CPU 时钟
有关）
test.QuadPart = frequence.QuadPart *
time/1000000;
QueryPerformanceCounter( &previous );
current = previous;
//循环调用 QueryPerformanceCounter
直到计数值到为止
while(current.QuadPart -
previous.QuadPart < test.QuadPart)
QueryPerformanceCounter( &current );
}

```

函数 void CTest\_timeView::OnBTest(UINT time)的变量 time 单位为微妙，使用方法同 Sleep 函数，通过它可实现微妙级定时，从而控制堆垛机运行平稳。

#### 2.4 行走模块的算法

堆垛机的行走由直流电机驱动，光电传感器在首、尾各安装 16 个，用来检测地面上黑、白相间窄条的反光，并控制直流电机 PWM 差动调速。以首部 16 个传感器为例：传感器高、低 8 位的状态分两次读取，得到的数值需屏蔽掉前（或后）2 位，如果得到的数据是 0x0380, 0x02c0, 0x0340, 0x01c0, 0x01E0 或 0x0780，就认为在中间位置，偏差为 0；如果得到的是其他数据需与 0x03c0 比较，求出偏差值。差动控制就建立在堆垛机运行趋势的变化上，将本次偏差值与前次相比较，如果偏差趋势变大，分左偏和右偏两种情况分别调用函数 leftfor(int i,int j)或函数 rightfor(int i,int j) 进行差动纠偏；如果偏差未变或偏差趋势减小，堆垛机继续沿直线移动直到接受到停止信号为止。

#### 2.5 自定义消息的实现

堆垛机升降机构、伸缩叉机构的运动控制由循环线程向主线程（视类）发消息来实现。在循环线程函数体中实时检测启、停开关量的变化，然后向主线程发送自定义消息，具体操作由消息响应函数完成，从而实现堆垛机的升降与伸缩控制。

消息实现的过程包括 4 个基本步骤：①在视

类头文件中添加用户消息宏定义“#define WM\_MY\_MESSAGE WM\_USER+100”；②在视类实现文件中添加消息处理函数 LRESULT CTest\_timeView::OnMyMessage (WPARAM wParam, LPARAM lParam)，其中 wParam、lParam 是 2 个消息参数，用来完成参数的传送，在消息响应函数体中将根据传送数值的不同而执行不同的分支程序，以完成升降、伸缩等不同控制功能；③在视类头文件的 AFX\_MSG 块中添加语句“afx\_msg LRESULT OnMyMessage (WPARAM wParam, LPARAM lParam)”说明消息处理函数；④在视类的消息块中，使用 ON\_MESSAGE(WM\_MY\_MESSAGE, OnMyMessage)宏指令将消息映射到消息处理函数中。

在循环线程中调用时，首先要取得视类指针 pView，然后调用 PostMessage 或 SendMessage 发送消息，例如：pView-> SendMessage (WM\_MY\_MESSAGE, 0, 1)，结果将参数 wParam=0, lParam=1 传送给视类 OnMyMessage 消息响应函数，函数中 switch 语句根据消息参数的不同而转向对应的分支程序以完成相应的控制功能。

#### 2.6 主控计算机对堆垛机的控制

堆垛机控制机无需显示器、键盘、鼠标等外围设备，它由主控计算机进行操作与控制。主控计算机对堆垛机的控制采用 Windows 2000 Server 自带的远程控制功能，控制主机只需利用键盘、鼠标输出指令，即可实施对远程计算机的完全控制，限于篇幅，本文不作介绍，详见文献[2]。

#### 参考文献：

- [1] 徐菱，劳扬健，王金诺. 基于 PLC 的堆垛机控制系统设计[J]. 组合机床与自动化加工技术, 2005 (1): 72-73.
- [2] 佚名. 终端服务 - Windows 2000 Server 自带的远程控制工具 [EB/OL]. <http://www.txxd.com/art/pc/network/200410/109.html>.

（下转第 39 页）

$$d\lambda(z) = \frac{u_{n_1,0}}{u_{n_1+1,0}} - \frac{u_{n_1-1,0}}{u_{n_1,0}}$$

$$\int_{b_j D^2} |\sqrt{u_{n_1,0}} z_1^{n_1} \bar{z}_2|^2 d\lambda(z) = \frac{u_{n_1,0}}{u_{n_1,1}}$$

$$\int_{b_j D^2} |\sqrt{u_{0,0}} \bar{z}_j|^2 d\lambda(z) = \frac{3}{4} \quad (j=1,2)$$

根据以上结果可得

$$\sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \left\| S_1(\sqrt{u_{n_1, n_2}} z_1^{n_1} z_2^{n_2} d\bar{z}_1 + \sqrt{u_{n_1, n_2}} z_1^{n_1} z_2^{n_2} d\bar{z}_2) \right\|^2 = +\infty$$

所以  $S_1$  不是 Hilbert-Schmidt 算子。

参考文献:

- [1] Bell S. The Cauchy Transform, Potential Theory, and Conformal Mapping[M]. CRC Press, Boca Raton, 1992.
- [2] 郑 红. Hardy 空间的  $\bar{\partial}$  方程的标准解算子[J]. 南昌大学学报理科版, 2005, 29(2): 122-125.
- [3] Friedrich Haslinger. The Canonical Solution Operator to  $\bar{\partial}$  Restricted to Bergman Space[J]. Proc AMS, 2001, 129(11): 3321.

## Canonical Solution Operator to $\bar{\partial}$ Restricted to Hardy Spaces in Polydisc

ZHENG Hong

(Shenzhen Polytechnic, Shenzhen, Guangdong 518055, China)

**Abstract:** The canonical solution operator to  $\bar{\partial}$  restricted to  $(0,1)$ -forms with coefficients  $(f_j \in H^2(b\Omega), j=1, \dots, n)$  can be expressed by an integral operator using the Szegő kernel, which indicates that, in the case of the polydisc in  $\mathbb{C}^2$ , the corresponding operator is not a Hilbert-Schmidt operator.

**Key words:** Hardy spaces; Szegő kernel; Szegő projection

(责任编辑: 王璐)

(上接第 19 页)

## Real-time Control of Stacking Machine Based on Windows System: Its Software Design

ZHANG Aihong

(Department of Mechanical and Electrical Engineering, Wuxi College of Technology, Wuxi, Jiangsu 214121, China)

**Abstracts:** The paper introduces the hardware makeup of stacking machine and the process of developing control software according to the actual requirements such as moving forward and backward, up and down, etc. The design of real-time control software is discussed in detail. As a result, the further development of FMS or CIMS can be achieved conveniently.

**Key words:** stacking machine; wireless LAN; multithread

(责任编辑: 王璐)