基于数据手套的机械手控制技术应用*

张俊杰[1]，李世其[2]，熊友军[2]

(1. 华中科技大学 软件学院，湖北 武汉 430074；2. 华中科技大学 机械学院，湖北 武汉 430074)

摘　要：基于数据手套的机械手控制是一种新的机械手控制模式。与传统的机器人控制方法作比较，分析了各自的优势，介绍了系统框架，讨论了系统实现中的关键技术，最后实现了原型系统。应用表明，这种方法操作简单、交互性好，有广泛的应用前景。

关键词：串口通信；协议设计；同步控制

中图法分类号：TP391.8　　**文献标识码**：A　　**文章编号**：1001-3695(2006)07-0170-02

Application of Robot Control Technology Based on Data Glove

ZHANG Jun-jie[1]，LI Shi-qi[2]，XIONG You-jun[2]

(1. *College of Software, Huazhong University of Science & Technology, Wuhan Hubei* 430074, *China*; 2. *College of Mechanical Science & Engineering, Huazhong University of Science & Technology, Wuhan Hubei* 430074, *China*)

Abstract: Robot control based on data glove is a new module of robot control. Compares with tradition way of robot control and analyses each advantage, and introduces system frame, in what follows discusses the key techno-logy of system, in the end the prototype system has been implemented. Practice indicates that it has simple operation, good interaction and widely application future.

Key words: Serial Port Communication; Protocol Design; Synchronous Control

1　引言

目前，机器人不仅在工业上应用越来越广，而且正在社会服务、海洋开发、宇宙空间、地下矿藏、军事作战、抢险救灾等领域开拓新的应用，以代替人类在无法适应的特殊环境下工作，这将极大地扩展人类的活动范围。

传统的机器人控制模式大多是操作者通过向控制器输入运动指令来控制机器人的运动。这种模式的缺点是操作复杂且很不直观，操作者事先要进行培训，对操作者要求较高，需要他们有一定的计算机编程知识。即使这样，操作者对机器人的控制也很难具备实时性和交互性。这种控制模式在操作环境熟悉、动作相对固定的情况下效果比较好，如 ABB 公司的焊接机器人和中国科学院沈阳自动化研究所的空调通风管道清洁机器人都采用这种模式。本文采用的控制模式是操作者戴上数据手套通过计算机系统实时控制机器人，并且计算机上有生成的虚拟场景辅助操作，在看不到机械手的情况下，操作者也能通过改变虚拟手的姿态对机械手作出相应的调整。与传统控制方法中操作者通过输入运动指令的间接控制相比，这种直接的控制方法更加符合人类的操作习惯，它的操作简单、高效，大大提高了交互性。在卫星维修、深海探索等人类不熟悉的环境下采用这种模式控制机器人则效果更理想。两种控制模式的对比如图 1 所示。目前国内相关人士[1]对这种新型控制方法进行了一些研究，提出了一些实施方案，但还没有应用于实际。

收稿日期：2005-05-24；修返日期：2005-07-25

基金项目：国家民用航天科研专项计划(科工技[2004]1530)

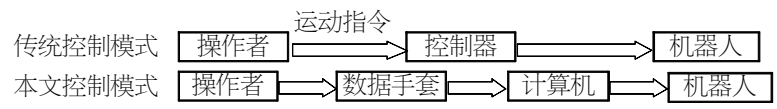


图 1　两种控制模式对比图

2　系统框架

整个系统分为三大部分，即操控系统、计算机软件系统和机械手。这三部分相互作用，构成一个整体，如图 2 所示。

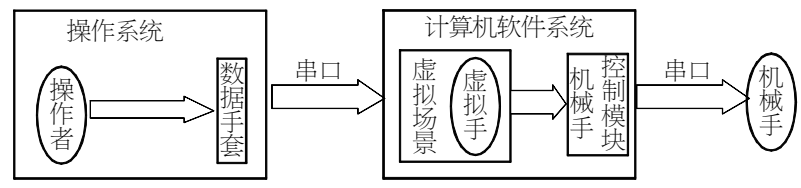


图 2　系统框图

操作者发出操作动作后，数据手套通过自带的传感器捕获操作者手的姿态信息，并将这些信息通过计算机串口传递给计算机系统；计算机系统得到人手的姿态信息后，计算确定人手的运动状态，然后将这些参数传递给虚拟手；虚拟手就会根据这些参数实时地调整姿态，同时把参数传给机械手控制模块；机械手控制模块依据机械手的通信协议，把这些参数转换为机械手对应转轴的运动指令，也通过计算机串口发送出去，最终达到控制机械手的目的。整个过程在我们事先设定好的时间范围内循环进行，因为时间间隔很短，操作者基本感觉不到时延，这就实现了人手对机械手的同步控制。

根据系统要求，我们采用的硬件设备是中鸣 6 自由度机械手和 5DT 数据手套。中鸣 6 自由度机械手臂(图 3(a))由七个精确的伺服马达组成，共有六个自由度，分别对应于转动基座、臂、肘、前腕、后腕、钳六个转轴，每个转轴可在 ±80 °的范围内活动。数据手套(图 3(b))是 5DT 公司生产的不带位置跟

踪器的数据手套, 带有七个传感器, 每个手指有一个, 在手腕部位有两个传感器, 可以测量手套的转动和倾斜两个角度, 分别用于探测手的上下摆动和旋转。

3　系统关键技术

3.1　串口通信

数据手套和机械手都是通过 RS-232 串口与计算机相连的, 在 Windows 环境下实现串口通信有两种方法[2], 即利用 API 通信函数和 ActiveX 控件。Windows API 是所有 Windows 应用程序的根本之所在, 其实质是一系列例程, 应用程序通过调用这些例程来请求操作系统完成一些低级服务。ActiveX 控件是一个动态链接库, 是作为基于 COM 服务器进行操作的, 并且可以嵌入在包容器宿主应用程序中, ActiveX 控件的前身就是 OLE 控件。由于 ActiveX 控件与开发平台无关, 因此, 在一种编程语言上开发的控件可不作修改地应用于另一种平台上。Microsoft Communications Control(MSComm) 是 Microsoft 公司提供的简化 Windows 下串口通信编程的 ActiveX 控件, 为应用程序提供了通过串口收发数据的简便方法。

机械手的控制方法采用的是 MSComm 控件进行串口编程, 在不同应用程序用 MSComm 控件进行串口编程的基本步骤是类似的, 主要包括下面几个步骤:

(1) 在建立的程序工程中插入 MSComm 控件;

(2) 添加 MSComm 控件 ID 的控制变量(或者对象) ;

(3) 对串口进行初始化, 设置 MSComm 控件的属性;

(4) 添加串口事件的消息处理函数 OnComm( ) 函数, 在函数中根据应用需要编写数据处理代码;

(5) 编写串口发送等其他代码;

(6) 关闭串口。

3.2　通信协议设计

机械手的运动是由机械手的通信协议决定的, 要精确控制机械手必须准确设计出机械手的通信协议, 但是机械手没有一个通用的通信协议, 我们要根据通信协议的普遍规则和大量的试验设计出我们这个机械手的通信协议, 从而实现对机械手的精确控制。串口通信协议[3] 分为底层通信协议和用户层协议。底层协议一般由计算机硬件提供商和设备厂家提供, 在一般性的通信编程中很少涉及, 而用户层协议则是面向使用者的, 也就是我们在编程中通常所说的通信协议。编制用户层的通信协议具有很大的随意性, 但通常要遵循以下三个原则:

(1) 数据包必须有包头。包头是供接收方判断一个数据包开始传输的重要标志, 接收方从接收的数据中判断收到了包头, 就认为接收的数据已经开始, 真正的数据信息马上就会到达。包头字符必须有别于数据信息, 这种特征是数据包中其他数据没有的, 否则就会引起混乱。

(2) 非定长数据包必须有包尾。所谓非定长, 是指没有指明数据包的长度。对于非定长的数据包, 接收方只能根据包尾标志判断数据包是否结束。同包头一样, 句尾字符也必须有别于数据信息, 这种特征是数据包中其他数据没有的, 否则也会引起混乱。

(3) 一般应该对数据加入校验。串口通信底层协议(由机器硬件实现) 已经设置了奇偶校验方式, 在用户层加入校验, 可以对数据进一步排错, 更好地保证数据的正确性, 因为在重要的场合, 数据出错可能引发严重的后果。

设计协议时应该按照实际情况具体问题具体分析, 下面是针对本机械手通信协议的设计思想: 运行机械手自带的控制软件 MINI SERVO Explorer 来控制机械手, 然后用串口调试助手软件获取机械手控制软件 MINI SERVO Explorer 传送的数据, 通过大量试验找出数据的规律, 并结合串口通信协议的设计规则设计出机械手的通信协议。

3.3　同步控制

系统要求机械手应该在操作者发出动作后, 根据操作者手的姿态同步运动, 如果不能很好地满足系统所需的实时性要求, 就失去了研究的基础和意义。

众所周知, Windows 是基于消息机制的系统[4], 任何事件的执行都是通过发送和接收消息来完成的。这样就带来了一些问题, 如一旦计算机的 CPU 被某个进程占用, 或系统资源紧张时, 发送到消息队列中的消息就暂时被挂起, 得不到实时处理。因此, 不能简单地通过 Windows 消息引发一个对时延要求严格的事件。另外, 由于在 Windows 中已经封装了计算机底层硬件的访问, 要想通过直接利用访问硬件来完成精确定也比较困难。所以, 绝对的实时控制是不可能的, 我们只能根据实际需要选择最符合要求的方式。

VC 中的 WM_TIMER 消息映射能进行时间控制。首先调用函数 SetTimer( ) 设置定时间隔, 然后在应用程序中增加定时响应函数 OnTimer( ) , 并在该函数中添加响应的处理语句, 把数据手套获取的信息转换为机械手对应转轴的运动指令, 并发送出去。

用这种方法处理相对简单, 并且 CPU 占用率低, 虽然仍有十几毫秒的时延, 但已经完全可以满足系统的要求, 因为对于操作者来说十几毫秒的时延根本感觉不出来。

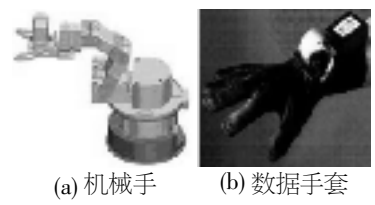
SetTimer 函数的原型是:

UINT SetTimer(UINT nIDEvent, UINT nElapse, void (CALLBACK EXPORT* lpfnTimer) (HWND, UINT, UINT, DWORD))

当使用 SetTimer 函数的时候就会生成一个计时器。函数中 nIDEvent 指的是计时器的标志, 也就是名字。nElapse 指的是时间间隔, 也就是每隔多长时间触发一次事件, 单位是毫秒。第三个参数是一个回调函数, 在这个函数里, 放入想要做的事情的代码, 还可以将它设定为 NULL, 也就是使用系统默认的回调函数 OnTimer( ) 函数。

4　系统介绍及结论

采用上述方法在 Windows XP 操作系统、VC + + 6.0、Open Inventor 的开发环境下建立了一套基于数据手套的机械手控制系统, 如图 4 所示。



(a) 机械手　(b) 数据手套

图 3　中鸣 6 自由度机械手和 5DT 数据手套

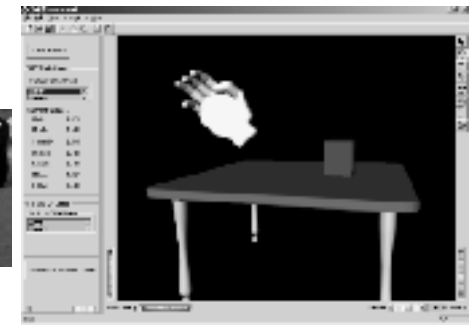


图 4　虚拟场景辅助操作

(下转第 175 页)

目标移动平台,通过目标移动平台通信模块对数据解包,传送到目标平台移动 Agent 管理模块;通过目标平台移动 Agent 管理模块和注册模块对迁移的 Agent 进行安全认证和注册,最后恢复 Agent 的执行,同时通知所有者迁移结束。

移动 Agent 的通信包括移动代理 Agent 间的通信和移动 Agent 与环境间的通信。为了保证本地环境的安全性,移动 Agent 不能直接访问环境,而是通过 ServiceAgent 进行通信。当移动 Agent 访问环境资源或同其他 workerAgent 通信时,它向 ServiceAgent 请求提供服务,ServiceAgent 通过交互和管理模块完成访问和通信功能。移动 Agent 间的通信涉及到移动 Agent 的定位,根据移动代理平台的区域分布,采用树状地址服务体系,分公司代理系统作为根地址,各气矿代理系统作为子树节点,依次类推。同时维护一张地址映射表,存储已经通信的代理 ID 和地址,这样下次再通信时就可以直接在地址映射表中查找,而不必再重新解析地址。

3.2　移动代理的安全性保证

移动代理的安全性是移动代理系统能够实际应用的先决条件,包括网络上数据的安全性、存取资源的安全性、代理自身的安全性。

网络上的安全传输,采用 GPRS + APN + VPN 的方式,由于使用 APN,利用 SIM 卡的唯一性,对用户 SIM 卡手机号码进行鉴别授权,在网络侧对 SIM 卡号和 APN 进行绑定,划定用户可接入某系统的范围,只有属于指定行业的 SIM 卡手机号才能访问专用 APN,普通手机的 SIM 卡号无法呼叫专门的 APN。专用网络采用专用 IP 地址,VPN 方式,且和外网 Internet 无接口,封闭隔离,保护入网 PDA、手机、系统平台和油气田专网安全。

存取资源的安全性,使用 Java 提供的安全控制策略,即 Security Manager。它严格地规定了系列安全规则,包括是否允许程序检查运行堆栈、访问本地文件、访问网络信息、访问系统属性和执行系统命令等。

代理自身安全性的实现,使用 Java 线程组和类装载器,当新代理到达时,根据任务建立若干线程,其组号相同,为这个线程组分配权限,实现代理与主机隔离;Java 虚拟机让每一个类与类装载器关联,类装载器定义它自己的名字空间,每个代理都有一个唯一的类装载器实例,因为一个代理不能以它自己的类版本代替另一个代理类,从而实现了代理与代理的隔离。

4　结束语

移动代理技术应用在数字气田生产运行系统中,减少了主机间数据的交换,降低了系统对网络带宽和可靠性的依赖,提高了系统的实时交互和系统运行的可靠性和稳定性;同时为系统提供了更好的灵活性、可扩充性和适应性。移动代理在数字气田生产运行系统中的应用研究,为数字化气田分布式应用提供了一种新的手段和方法,为数字化气田其他分布式应用提供了一个模板,具有实际的应用前景。

参考文献:

[1]　apastavrou S, Samaras G, Pitoura E. Mobile Agents for World Wide Web Distributed Database Access[J]. IEEE Transactions on Knowledge and Data Engineering, 2000, 12(5): 802-820.

[2]　Venners Bill. Solve Real Problems with Aglets, a Type of Mobile Agent[J]. Java World Magazine, 1997, (5).

[3]　Jou Ahyh-Hong, Kao Shang-Juh. Agent-based Infrastructure and an Application to Internet Information Gathering[J]. Knowledge and Information System, 2002, 4(1): 80-95.

[4]　Guan Jihong, Zhou Shuigeng, Bian Fuling, *et al*. Building Distributed Web GIS: A Mobile-Agent-based Approach[J]. Wuhan University Journal of Natural Sciences, 2001, 6(1-2): 474-481.

[5]　佟强,等.基于 Mobile Agent 的分布式空间数据服务模型[J].计算机工程,2004,30(1):68-70.

[6]　陈剑,周意青,刘振华. Java Applet 的安全性及应用[J].计算机应用研究,2001,18(11):69-70.

[7]　张冠群,陶先平,李新.移动 Agent 的迁移机制研究[J].计算机科学,2001,(9):69-72.

作者简介:

杨力(1975-),男,四川营山人,硕士,主要研究方向为人工智能与分布式计算;张仕强(1973-),男,四川成都人,高工,硕士,主要研究方向为石油工程;邬云龙(1961-),男,四川仪陇人,高工,博士,主要研究方向为天然气测试及开发;曹谢东(1954-),男,四川南充人,教授,主要研究方向为人工智能与分布式计算。

(上接第 171 页)运行程序,单击 Build Scene 按钮,程序窗口中就出现了虚拟场景,包括虚拟手、一张桌子以及放在上面的立方体。在 Build Scene 按钮下方有个选择框,双击 COM1 连接数据手套,如果这时数据手套已经连接在计算机的 COM1 端口上,虚拟手就会根据数据手套捕获的操作者手的姿态信息实时地调整自己的姿态,同时 Current Values 下面的七个值会随数据手套的运动而改变,如果把机械手连接在计算机的 COM2 端口上,机械手就会根据数据手套的姿态实时改变位置。通过这个系统操作者使用数据手套可以精确定位机械手的各个转轴,并且可以形成连续的运动、做出各种复杂动作。例如让机械手臂完成搬运物体、投篮、绘画、木偶表演等动作。

实验结果显示:操作者通过数据手套来控制机械手具有很强的实时性和交互性,这种控制方法相对于传统的控制方法具有操作简单、直观、高效等优点,有广泛的应用前景。

参考文献:

[1]　王家顺,王田苗,魏军,等. 一种面向遥操作的新型数据手套研制[J]. 机器人,2000,22(3):201-206.

[2]　任全,陈西宏,谢鹏. Win32 环境下 Modem 数据传输实现方法及通信协议研究[J].计算机应用研究,2002,19(4):120-124.

[3]　汤双清, Michel Dupas. 串行通信在工业机器人控制中的应用[J]. 计算机应用研究,2000,17(11):79-81.

[4]　施华,薛广涛. 机器人控制系统实时性的研究[J]. 计算机工程,2003,29(1):91-92,125.

作者简介:

张俊杰(1980-),男,湖北武汉人,硕士生,研究方向为基于数据手套的机械手控制技术;李世其(1965-),男,江西人,教授,博导,主要从事计算机辅助设计技术研究和软件开发方面的工作;熊友军(1978-),博士生,研究方向为机器人遥操作技术。