

# Visual C++ 编程实现机器人网络控制

张爱红, 张秋菊

(江南大学机械工程学院, 江苏无锡 214062)

**摘要:** 提出了一种基于虚拟现实技术的微机网络控制方法。该方法基于面向连接的客户/服务器模式, 应用 TCP/IP 网络传输协议, 利用 CSocket 派生类的虚拟函数来完成客户/服务器信息的发送, 通过对不同网络信息的分析判断以执行不同功能的函数代码, 最终完成机器人网络控制。

**关键词:** 机器人; 进程; 通信; 网络控制; 协议

**中图分类号:** TP242 **文献标识码:** A **文章编号:** 1001-3881(2003)6-165-4

## Realization of robot net control using Visual C++ programming

ZHANG Ai-hong, ZHANG Qiu-ju

(Mechanical Engineering College, Southern Yangtze University, Jiangsu Wuxi 214062, China)

**Abstract:** A new method on microcomputer net control based on virtual realization technology was presented. Based on Client/server mode and TCP/IP protocol, by using virtual functions denied from CSocket to accomplish the communication between client and server, robot net control was achieved with different function being executed according to different messages.

**Keywords:** Robot; Process; Communication; Net control; Protocol

## 0 引言

随着计算机、自动化、通信等高新技术的快速发展以及在制造系统中的应用, 制造技术发生了质的飞跃: 从最初的手工制造发展到了如今的柔性制造系统(FMS)、计算机集成制造系统(CIMS)等。在这些先进的制造系统中, 机器人就像数控机床一样已经成为实现柔性制造、智能制造等不可或缺的设备之一。它主要用于物料搬移、零件装配、焊接等操作。在这类系统中一般都配置一台主控计算机, 由计算机对系统中的数控机床、机器人进行集中控制, 可以实现通常的远程控制。为了利用 Internet 网络廉价、便捷的优点以及解决机器人远程操作过程中出现的通讯时延等问题, 本文提出了一种基于虚拟现实技术的微机网络控制方法。

## 1 系统组成

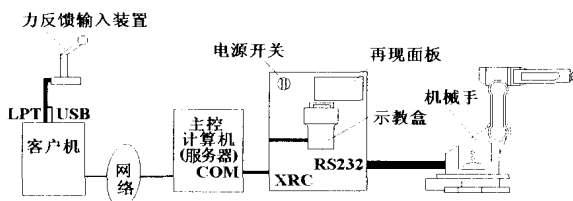


图1 机器人虚拟示教及远程控制系统组成图

本系统主要由六自由度机械手(SV3X)、机器人控制柜(XRC)、示教盒、主控计算机、客户机和具有力反馈功能的示教输入装置等组成。控制柜与机械手、微机、示教盒间均通过电缆连接, 客户机、服务器之间通过网络相连, 输入装置(如游戏操纵杆)连到了客户机的并行端口(或声卡接口)和USB接口(提供力感)上, 如图1。

通过 Visual C++ 调用 OpenGL 图形库建立客户机、服务器上逼真的六自由度机器人模型, 是实现机

器人虚拟示教的基础。限于篇幅, 本文不作介绍。实际操作过程中, 使用者在客户机上利用鼠标以及力反馈游戏操纵杆等输入设备实现机器人的虚拟示教编程, 经仿真动画验证后的作业程序通过网络上传到主控计算机(服务器), 由主控计算机接受客户机的要求实现: 示教、再现、下载作业到控制柜、启动运行等功能, 从而使机器人完成给定的作业序列。

## 2 机器人虚拟示教与作业生成

程序运行后使用者点击右键出现弹出式菜单, 根据菜单提示输入不同的文件名, 完成初始化工作后, 就可以通过客户机完成机器人虚拟示教: 操纵示教输入装置到某一位姿记录示教点数据, 同时通过键盘输入示教指令程序, 所有的示教点输入结束后, 回放再现, 验证整个作业过程, 为生成作业程序作准备。

作业程序是机器人能够识别的语言代码, 为了能够生成机器人语言, 必须按照机器人程序的格式和指令来编写程序。为此, 我们针对 Motoman 机器人编程语言 INFORM II, 剖析了 \*. JBI 程序, 并用 Visual C++ 编程实现了代码的自动生成。限于篇幅, 本文不作详细介绍。而下文将详细阐述如何利用 Visual C++ 编程实现机器人的网络控制。

## 3 网络进程通信原理

进程通信的概念最初源于单机系统, Internet 进程通信不同于单机系统中进程间的通信, 它主要解决不同主机进程间的通信。单机系统中进程标识符可以确定同一主机上的不同进程, 而在网络通信过程中全局地标识一个进程需要一个三地名组: 协议、本地地址、本地端口号, 一个完整的 Internet 进程通信实例由两个进程完成, 需要另外增加两组来标识: 远地地址、远地端口号。而在 TCP/IP Internet 中, 最主要的进程间的作用模型是客户-服务器模式。在机器人远程控制系统中采用主控计算机为服务器、与示教输

入装置直接相连的计算机为客户机，参见图 1。

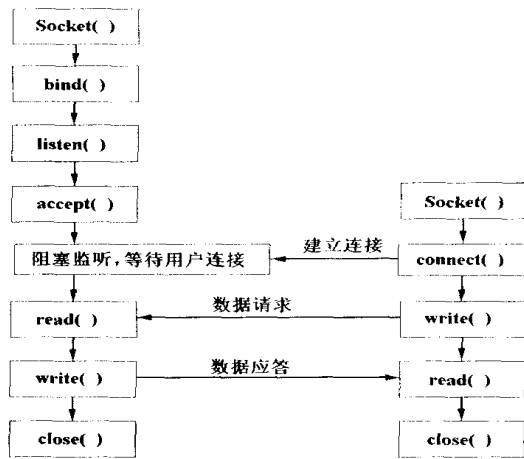


图 2 面向连接的客户/服务器模型

最初程序员在 Unix 系统下使用 Berkeley Socket 编写网络程序，随着 Windows 操作系统的普及，Microsoft、Sun 等公司联合开发了 Winsock 接口 API。它实质上是一种进程间通信，将之从单机环境扩展到网络环境以适合于开发主机/客户机通信程序。网络通信的每一端称为一个端点，而 Windows Socket 为网络端点的抽象表示。编程时，程序在网络对话的每一端建立一个 Socket (套接口)，两个 Socket 之间可以应用面向连接的或者是无连接的网络协议（不作介绍）。面向连接的网络协议不必指定本地协议端口，只需为 Socket 提供远程主机信息：IP 地址和协议端口，Winsock 可自动保存本地 IP 地址和选择本地端口。面向连接的 Winsock 客户机/服务器模型如下图所示，服务器程序首先用 Socket 函数建立一个 Socket，接着服务器调用 bind 函数将此 Socket 与本地协议端口联系起来。Listen 函数将 Socket 置于侦听模式，对于客户端每个请求发送一个确认信息，但在侦听的 Socket 实际上并不接受连接请求，为了真正接受和连接请求，必须调用 Accept 函数。客户端的程序也用 Socket 函数建立一个 Socket，客户端通过调用 connect 函数启动网络对话。在客户和服务器建立连接后，就可以调用 write、read 等函数进行进一步的通信。

#### 4 网络编程的实现

程序分两个应用例程：服务器端程序、客户端程序。我们在编程时采用了 MFC 类库，因为 MFC 类提供了 CSocket 类，它封装了 Windows Sockets API，支持同步操作，简化了 Socket 编程，降低了编程难度。

##### (1) 服务器例程

其中服务器端程序主要实现：监听客户端的连接请求、为客户端的连接请求建立 Socket 队列、处理客户端发送来的数据，并进行分析判断，根据接受到的

不同字符信号转向不同的函数体，完成相应的网络控制，并向相应的客户端发送应答信息。

在应用程序中需要建立两个 CSocket 类的派生类：Clisten 和 CClient。Clisten 类用于建立监听 socket，以随时监测是否有客户端的连接请求。CClient 类用于一旦监测到有客户端的连接请求时，即为每一个客户端分别建立一个 socket，以便通信中不发生混乱。程序代码如下：

```

Clisten::Clisten (CserverDlg *pdlg)
{
    // 需要改写 Clisten 类构造函数
    m_pdlg = pdlg;
}

void CserverDlg::OnNet ()
{
    // 右键菜单 "网络控制" 的响应函数
    m_pListening = new Clisten (this);
    if (m_pListening -> Create (PORT))
    {
        // 建立监听 socket
        if (! m_pListening -> Listen ())
        {
            MessageBox (" port error!", " net message", MB-OK);
            return;
        }
    }
}

```

以下代码利用 ClassWizard 生成的消息响应函数，利用了虚拟函数可以被重载的特点。

```

void Clisten::OnAccept (int nError)
{
    CClient *m_pSocket = new CClient (m_pdlg);
    if (m_pdlg -> m_pListening -> Accept (*m_pSocket))
    {
        // 为每一个客户端分别建立一个 socket
        m_pdlg -> m_pList. AddTail (m_pSocket);
    }
    else
        delete m_pSocket;
}

```

为了处理客户端的请求，还必须重载 CClient 派生类的虚拟函数 OnReceive (int nError)，在函数体中添加代码，对客户发送来的信号进行分析判断，再进一步执行相应的函数，例如：用户发送来的代码为 "1"，执行作业文件传输；代码为 "2"，执行作业下载等。最后，用 Send () 函数向客户端回发信息，表明成功完成网络通信。

##### (2) 客户端例程

客户端的程序主要实现的功能有：建立到服务器

的连接、关闭与服务器的连接、向服务器发送数据(发送控制信息、传送作业文件)、从服务器接受数据等。

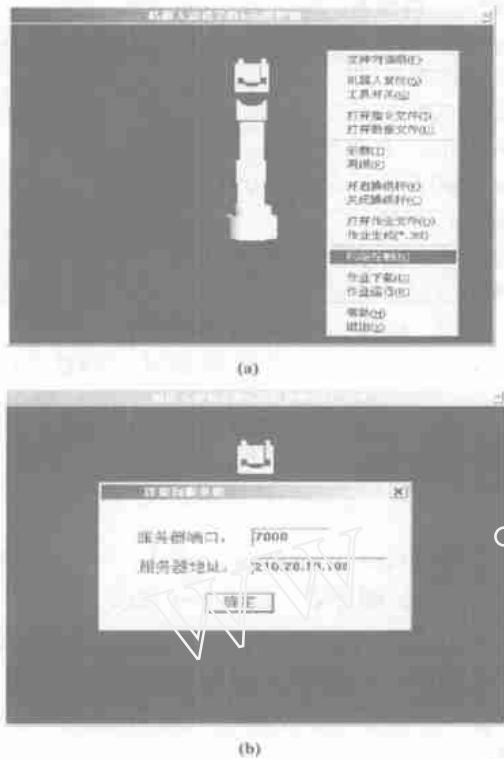


图3 服务器/客户端界面图

在应用程序中建立一个 CSocket 类的派生类: CRequest 类, 用于建立 socket, 实现客户端与服务器的连接通信。

```
CRequest::CRequest(CclientDlg *pdlg)
{
    // 需要改写 CRequest 类构造函数
    m_pdlg = pdlg;
}
void CclientDlg::OnOpen()
{
    // 打开通讯口
    // 以下程序实现打开连接对话框, 参见图 3:
```

服务器/客户端界面图: 图 3 (b)。

```
CConnectDlg dlg1;
if (dlg1.DoModal() == IDOK)
{
    m-port = dlg1.m-Port; // 协议端口
    servername = dlg1.m-servername; // 服务器 IP
    // 服务器 IP 地址
    m-pSocket = new CRequest(this);
    if (! (m-pSocket = ConnectServer ()))
    {
        ...// 代码省略
    }
}
```

```
}
}
CRequest * CclientDlg:: ConnectServer ()
{
    CRequest *pSocket = new CRequest (this);
    if (! (pSocket -> Create ()))
    {
        ...// 代码省略
    }
    if (! pSocket -> Connect (servername , m-port))
    {
        ...// 代码省略
    }
    return pSocket; // 最后连接成功, 返回接口对象指针
}
```

接着利用返回的接口对象指针 pSocket 来完成客户端信息的发送, 编程时先建立发送对话框(参见图 4)。下段代码为响应对话框发送按钮控件程序。

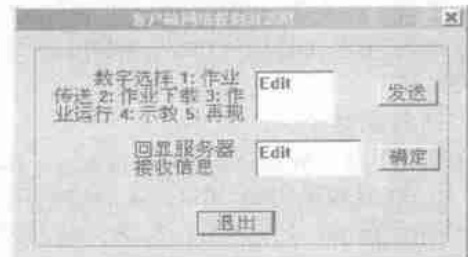


图4 客户端信息发送、回显对话框

```
void CSendDlg:: OnSend ()
{
    UpdateData (); // 完成数据输入更新
    CclientDlg * pdlg = (CclientDlg *)
    AfxGetApp -> m-pMainWnd;
    UpdateData (FALSE);
    If (pdlg -> m-pSocket)
    {
        char pMsg [1000];
        sprintf (pMsg, "%s", m-sendedit. GetBuffer
        (1000));
        pdlg -> SendMsg (pdlg -> m-pSocket ,
        pMsg);
    }
}
```

建立主对话框成员函数 CclientDlg:: SendMsg (CRequest \*pRequest, char \*pMsg) 并在函数体中调用 pRequest -> Send (pMsg, strlen (pMsg)) 完成信息的发送。

### 5 主控计算机与机器人控制器通信编程

主控计算机通过标准 RS232C 串行接口与机器人控制柜通信, MOTOCOM32 通讯软件提供了一套动态连接的库函数, 在服务器端的应用程序中调用这些库函数, 对机器人进行控制。这些库函数包括了机器人的主要操作功能, 例如: 状态读取函数用于读取机器人的位置值、当前程序名、运行方式、操作坐标、控制轴组等状态信息以及文件的上传、下载等; 系统控制函数可对机器人进行各种操作, 包括伺服电源的通断、程序运行的起停、机器人以各种坐标方式按不同的速度进行目标点或者增量运动。通过这些库函数的组合使用, 实现对机器人系统的各种控制与操作, 扩展了机器人的系统集成功能, 利于实现机器人远程控制和网络控制。

#### (1) 动态连接库的调用

为了程序能够调用动态连接库中的库函数, (1) 主对话框的实现文件中必须包含: "direct. h"、"motocom. h" 头文件 (2) 复制 Motocom32. DLL、Mtolk. DLL、Mtolkr. DLL 数据传输动态库到工程文件目录下 (3) 点击工程菜单、设置子菜单, 打开 "link" 项, 在 "Object/Library Module" 里添加 "motocom32. lib"。

#### (2) 通信的建立

首先利用语句: nCid = BscOpen (adlg. jbi-pathname, 1) 取得通讯句柄 nCid, 在后续的语句中调用 BscSetCom (nCid, 1, 9600, 2, 8, 0) 设置通讯口参数, 最后进行通讯线路的连接: BscConnect (nCid)。这样就可以进行文件的上传 (函数 BscUpload (adlg. file-name, nCid))、下载 (函数 BscDownload (nCid, adlg. file-name)) 等操作了。

#### (3) 通讯线路的关闭

在主对话框的 OnDestroy () 函数中调用关闭通讯口的函数 BscClose (nCid), 实现通讯的安全结束。

### 6 结论

本文所介绍的网络控制程序通过了完整和严密的试验, 实践证明它具有可靠、经济实用等优点。为机器人网络控制提供了一种新的思路, 也为解决传统远程控制开辟了一条行之有效的高效、廉价的方法。

### 参考文献

【1】李明柱, 单肃, 赵光启. Visual C++ 最新编程实践与技巧. 北京: 北京航空航天大学出版社, 2000: 107~148  
 【2】雷斌, 杨建华, 黄超等. Visual C++ 6.0 网络编程技术. 北京: 人民邮电出版社, 2000: 60~61  
 【3】官章全, 韩云君, 李罡. Visual C++ 6.0 高级编程范例. 北京: 电子工业出版社, 2001: 666~693  
 【4】庄严, 王伟, 恽为民. 基于网络的机器人控制技术

研究现状与发展. 机器人. 2002. 24 (3): 276~282

【5】耿海霞, 陈启军, 王月娟. 基于 web 的远程控制机器人研究. 机器人, 2002. 24 (4): 375~379  
 【6】Yasnac XRC instructions [M]. Yaskawa electric corporation, 1999.  
 【7】Yasnac XRC operator's manual [M]. Yaskawa Electric corporation, 1999.  
 【8】Motocom32 operation manual [M]. Yaskawa electric corporation, 1999.

作者简介: 张爱红 (1971 - ), 男, 汉族, 江苏盐城人, 工程师, 硕士研究生。研究方向: 机器人及机电一体化技术。电话: 0510 - 5430249 (H)。

收稿时间: 2002 - 11 - 25

(上接第 164 页)

```
// 数据点入堆栈;}
}
CurveToLine (limit, m-curve1, ...); // 嵌套调直线插补函数, 细分第一段 Bézier 曲线
CurveToLine (limit, m-curve2, ...); // 嵌套调直线插补函数, 细分第二段 Bézier 曲线
}
```

### 7 结论

该算法的优点是非常有效地根据曲线曲率控制曲线的插补步长。与其他算法相比, 虽然在存储结构上多了一个存放起始切矢与 X 轴夹角的空间, 同时增加了一些计算量, 但它对曲率变化非常敏感, 可对 Bézier 曲线不同曲率的地方产生疏密不同的插补点。该算法产生 G 代码应用于我们自己设计的玻璃雕花数控加工系统中, 算法稳定, 收到了良好的加工效果, 它对曲线加工质量和速度的提高有着重要的实践意义。

### 参考文献

【1】Choi B K 原著, 柯映林等编译. CAD/CAM 中的曲面造型技术. 浙江大学出版社  
 【2】Aumann, Gunter. Corner cutting curves and a new characterization of Bézier and B-spline curves. Computer Aided Geometric Design, Volume: 14, Issue: 5, June, 1997  
 【3】王晓东. Bézier 曲线树. 计算机辅助设计与图形学学报, 1994 (10)  
 【4】张玉连. Bézier 曲线的一种快速绘制算法及其实现. 燕山大学学报, 2000 (7)  
 【5】李雅青. 一种数控自动编程直线插补的新算法. 现代机械, 2002 (1)

作者简介: 杨海明, 男, 1976 年生。浙江大学机械能源工程学院硕士研究生。

收稿时间: 2002 - 11 - 25