文章编号：1002-0446(2003)02-0113-04

用 JAVA 开发的机器人遥操作系统

白 剑[1,2] 吴镇炜[1] 刘振诗[1]

(1. 中国科学院沈阳自动化所机器人学实验室 沈阳 110016； 2. 中国科学院研究生院)

摘 要：基于 WEB 服务器的机器人控制把传统机器人控制技术和最近数十年兴起的 INTERNET 技术结合到了一起. 由于现在 INTERNET 技术的普及，这种技术结合消除了传统机器人在普通大众前的神秘感，使机器人的应用层次达到了普通百姓的认识层次. 文章着重介绍了如何用 Java 语言开发遥操作系统. 在设计该技术时充分考虑了机器人控制的实用性以及所采用的各种 INTERNET 通讯技术以满足传统机器人的控制要求.

关键词：WEB 服务器；INTERNET 技术；流媒体；多线程通讯

中图分类号： TP24 **文献标识码**： B

THE TELE-ROBOT CONTROL SYSTEM DEVELOPED BY JAVA

BAI Jian[1,2] WU Zhen-wei[1] LIU Zhen-shi[1]

(1. *Robotics Lab, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang* 110016;
2. *Graduate School of the Chinese Academy of Sciences*)

Abstract: The robot control technology based on Web Server System has combined the technology of tradition robot control with Internet technology. By the widespread use of Internet technology, this combinative technology dispelled the mystery of tradition robot in people. This paper introduces how to use Java language to develop tele-robot control system. When we design the technology we fully considered the utility of robot control and communication technologies on Internet and we used to satisfy the quest of tradition robot control.

Keywords: web server, internet technology, media stream, muti-thread communication

1 引言(Introduction)

传统的机器人遥操作技术都是采用特定的操作人员和专用的操作系统，采用的通讯路径有用通信卫星的，也有用 ISDN 的. 而作为现在通用 INTERNET 技术其应用范围已经极为广泛，在西方国家中，网络入户已经得到了普及，可以想象如果把传统机器人控制简单化放到网络中去，将会对机器人的普及产生深远的影响. 基于 WEB 服务器的机器人遥操作技术就实现了这种想法，这种技术采用了客户端/服务器模式，所要实现的基本目标，是允许用户在远程终端上访问服务器，把机器人高层控制命令通过网络服务器传送给机器人控制器，达到间接控制移动机器人的目的，同时机器人现场的图像采集设备把机器人运动的实时图像再通过网络服务器反馈给远端用户，从而实现多用户多服务功能. 所有这些通讯及控制代码都是采用现在 INTERNET 上流行的 Java 跨平台语言所编写的，目的是为了能实现跨平台操作.

2 遥操作技术体系结构(The structure of tele-operate technology)

体系结构中很重要的一个部分就是机器人. 这里所采用的机器人是沈阳自动化研究所机器人开放实验室研制的全方位移动机器人，机器人的控制系统采用的是 QNX 系统. 为了便于远方的互联网用户对机器人的状态有感性的了解，特地给机器人的运动空间里配备了一个摄像头，将摄录的图像实时的传送到用户界面上. 配备服务器一台，用于提供客户端的控制界面下载，传输控制信息和状态信息.

为了支持 Java、Web 服务器和图像服务器运行，需要在上面提到的这台服务器上安装如下软件：

收稿日期：2002－10－25

(1) Java2 开发工具箱 J2SDK(简称 JDK). 它提供了一个完整的 Java 开发环境,用户可以创建任何基于 Java 且利用其核心 API 的解决方案.

(2) 支持 Servlet 的 Web 服务器软件.

(3) 处理音频/视频媒体的 JMF(Java Media Framework)软件包.

服务器和移动机器人的通讯是依靠无线 Ethernet 相联(基于 TCP/IP 协议,传输速率 3Mb/s),服务器可以通过无线网络向移动机器人发布指令、示教、读取运动状态等. 从而可以实现对车体的全无缆监控,这样原有移动机器人的监控部分包括键盘和显示器就并非必要,实际上是把移动机器人的监控终端由本地通过网络延伸到了 Internet 上的任何可能节点.

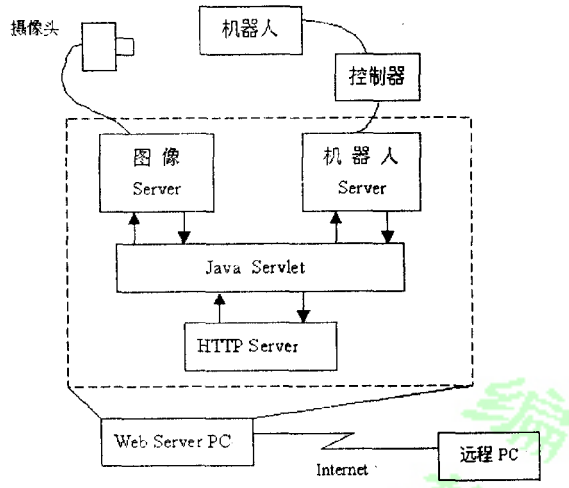

图 1 机器人遥控系统逻辑结构图

Fig. 1 The logistic structure of teleoperate robot

3 遥操作系统的软件设计(The design of tele-operate software)

3.1 程序设计的关键部分

客户端是独立存在的一个由 JavaApplet(Java 小应用程序)构成的页面,该页面向用户提供控制信息交互和流媒体实现两种功能. 程序也主要按照这两项功能的实现来编写.

1)服务器上有专用的 JavaServlet(服务器小程序)响应客户端请求,同时将由另外一个图象采集程序采集的实时图像转发到客户端,图象采集程序可以直接从摄像头连接的 USB 口接收到图像信息,进行实时广播.

2)服务器端的 Server. java 程序主要负责响应客户端的链接请求,分辨控制者以及接收和广播控制信息,响应机器人的连接请求,发送控制信息接收机器人状态信息,以及客户端与机器人之间控制及状态信息的交互.

3)由于采用的是客户机/服务器模式,服务器不但提供客户端与机器人的数据交互,还要向各个客户机提供图像方面的流媒体服务,当同时在线的用户增多时,就会造成服务器负担过重,不利于程序的正常运行. 针对这种情况,服务器将只作为单独的数据库服务器存储数据和执行查询,将结果返回客户机,而将数据服务与其余应用程序逻辑分开,这样其余应用程序逻辑的实现成为客户机的责任,这是所谓的肥客户机(见参考文献 2). 因此客户机上运行的 CCont2. java 小应用程序承担了大部分数据的处理和分析工作.

3.2 程序的要点分析

1)客户端与服务器端之间的通讯,根据不同的需要采用了两种通讯方式,分别是面向连接的 TCP 通讯,以及无连接的 UDP 通讯. 判断客户端是控制者(controller)还是观测者(observer)以及控制指令的传输需要及时送达服务器端,由于 TCP 通讯采用 socket 通道,发送一次必须接收一次,如果不通就发生阻塞,这样就保证了指令的正确有效传送.

客户端:

```
sss=new Socket("210.72.132.63",8888);
in=new DataInputStream(sss.getInputStream());
out = new DataOutputStream(sss.getOutputStream());
out.write(buffer);
int n=in.read();
```

输出的判断标识以及控制指令均被转换成字节包含在 buffer 字节数组中,返回的字节转换成整数用于判断服务器是否把此客户端升级为控制者.

服务器:

```
int port=8888;
serverSocket=new ServerSocket(port);
Socket socket=null;
socket=serverSocket.accept();
UserThread userThread = new UserThread(socket);
userThread.start();
```

每当一个客户端连入以后,服务器都会新开一个 userThread 线程.

```
protected DataInputStream inStream;
```

```
protected DataOutputStream outStream;
k[3]=0x0a;
inStream.read(buffer);
```

在线程里传来的信息读入到 buffer 字节数组中，k[3]作为判断字节被传入的判断标识所改变，用来保证任何状态下最多只有一个控制者存在.

为了让控制者和观测者能够时刻了解机器人的状态，从而对机器人响应控制指令的有效性加以判断，就需要服务器每时每刻都要把状态信息传到各个用户. 由于控制者只有一位，因此传回状态信息的同时还要传回控制指令，使观测者也能了解机器人运行指令的情况. 在这些通讯中，由于最新最快的状态信息反馈是最高要求，因此采用面向无连接的 UDP 通讯，状态信息在传送中丢失一个或者两个并不影响用户的观测，只要保证用户能最快的了解到机器人状态就可以.

客户端：

```
ds=new Socket("210.72.132.63",ServerPort,
false);
InetAddress Caddr = InetAddress.getLocal-
Host(); //客户机地址
String outMessage=Caddr.getHostAddress();
byte[] sendData=outMessage.getBytes();
output = new DataOutputStream(ds.getOut-
putStream());
output.write(sendData);
output.flush();
input=ds.getInputStream();
input.read(receiveData);
```

在客户端窗口刚打开的初始状态客户端向服务器发送一条指令，服务器通过指令了解到新客户端的 IP 地址和 UDP 的端口号码.

服务器端：

```
drs=new DatagramSocket(uPort);
UserUdpThread userUdp = new UserUdp
Thread(drs);
userUdp.start();
```

与 TCP 连接一样，每当一个新用户连入就开一个 userUdp 线程来处理与用户的通讯.

```
drs.receive(inPacketc);
client—address=inPacketc.getAddress();
UDPport=inPacketc.getPort();
outPacketc = new DatagramPacket(buffer2,
buffer2.length,client—address,UDPport);
drs.send(outPacketc);
```

在这个线程中，服务器通过传来的 UDP 包获得客户端的地址和端口，然后将机器人状态信息打包发送到客户端，如果有指令发回来，也将指令变成字节流打包发送到各个客户端.

2）服务器端与机器人的通讯，与服务器与客户端通讯一样其中也包括了指令传输以及状态信息传输，所以也采用了用于控制指令传输的 TCP 和用于状态信息传输的 UDP 通讯. 因为具体方式与前面类似，所以不详加描述.

3）客户端指令传入机器人、机器人状态传回客户端都需要经过服务器端中转，由于前面两种通讯分别在不同的线程中进行，所以信息的传送涉及到了多线程之间的通讯. 为了避免线程之间通讯时对相同变量的同时读写造成数据混乱，采用了同步机制协调通讯. 例如客户端传来控制信息的时候，负责从通道中读取的线程 1 先要把数据放入数组 buffer 中，确定数据全部传入 buffer 后，才能由负责把控制信息传入机器人的线程 2 把数据写入另外一个通道，因此采用对 buffer 数组同步处理，在数据传入 buffer 时，线程 2 挂起，数据写完后，线程 1 中唤醒线程 2，这样就完成了一次数据传递.

线程 2：

```
synchronized(buffer)
{buffer.wait(); }
buffer->control;
outData.write(control);
```

线程 1：

```
inStream.read(buffer);
synchronized(buffer)
{buffer.notify();}
```

4）图像的采集和转发

```
String rtpURL="rtp://" + localhost + ":"
+ Integer.toString(VIDEO_PORT)+"/video/
128";
MediaLocator ml = new MediaLocator(rtp
URL);
```

在服务器上运行的 VideoTransmit.java 程序中将地址直接指向接在 USB 口上的摄像机地址，并且按照 JMF 所订标准创建一个基于 JAVA 的媒体播放器. 每当一个客户端连入后，这个客户端就会发送一个请求，服务器端小程序 servlet 就会响应请求，克隆一个实时的流媒体发送到客户端.

```
ContentDescriptor cd=new ContentDescriptor
```

```
(ContentDescriptor.RAW_RTP);
    processor.setContentDescriptor(cd);
    setJPEGQuality(processor, 0.5f);
    DataSource origDataSource;
    origDataSource=processor.getDataOutput();
    processor.start();
    cloneableDataSource = Manager.createCloneableDataSource(origDataSource);
    cloneableDataSource.start();
```

4 实验结果及结论(Experiment result and conclusions)

只要是互联网上的用户,就可以登陆该主页对机器人进行控制,具体的实施步骤如下:

(1) 用户登陆 http://teleoperate.sia.ac.cn 就可以进入自动化所主页下的一个子目录,该目录将会弹出一个注册界面,提供用户进行注册或者登陆.

(2) 用户登陆成功后进入控制界面,如果事先已经有人登陆成为机器人的控制者,用户就只能成为观察者,无法对机器人发送控制指令.只有在控制者退出以后,根据登陆的排行,下一个用户自动成为控制者.

(3) 用户登陆成功后,无论是观测者还是控制者,都可以享受来自服务器的多种服务:机器人状态信息、机器人运动表现、图像反馈.

该系统客户端的界面如图 2 所示.

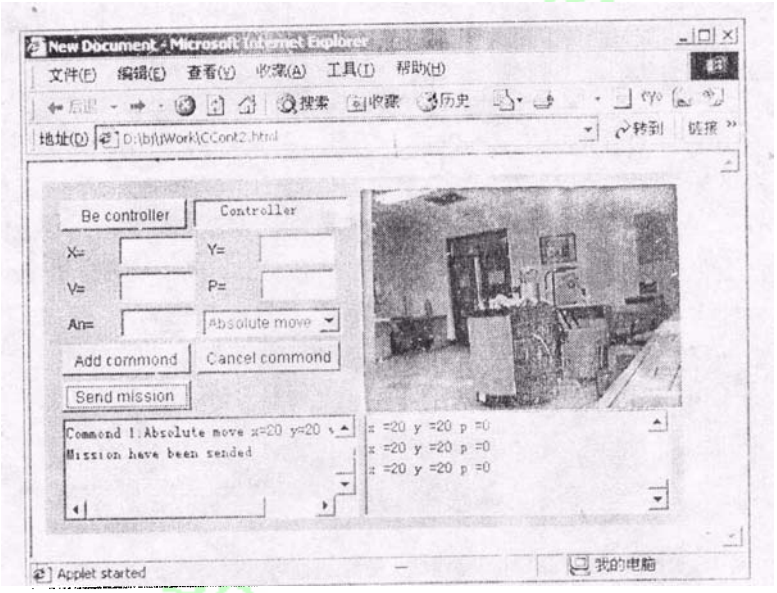

图 2 客户端控制界面

Fig. 2 The interface of client control

现已经实现图像以及指令信息的传送,达到了互联网上多用户控制和观察同时存在的目的,为机器人遥操作又添加了一项新的可供研究的控制方法.这个实验实现了机器人控制技术和互联网技术的初步融合,也为以后提供多操作人员同时控制及其研究机器人打开了方便之门.同时,该系统还可以在原有基础上继续添加实时操控系统,例如控制手柄、虚拟现实等技术的应用,从而大大扩充了传统机器人的控制概念,对传统机器人的应用也提高到一个新的阶段.

参考文献 (References)

[1] Teresa T Ho, Hong Zhang. Internet-Based Tele-Manipulation. Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering

[2] Chad Darby, JohnGriffin, Pascal de Haan. Java 网络编程指南. 电子工业出版社

[3] 金勇华, 曲俊生. Java 网络高级编程

作者简介:

白 剑 (1978-),男,硕士研究生.研究领域:基于 Web 的移动机器人控制开发.