

直升机虚拟座舱中网络通信模块设计

石春虎¹, 姜建芳¹, 曲红星², 陈雷²

(1. 南京理工大学自动化学院, 江苏南京 210094;

2. 总参陆航研究所, 北京 101121)

摘要:为了实现直升机虚拟座舱中多仿真计算机之间的网络通信,在GL Studio中集成开发Windows Sockets,采用C/S模式,运用Socket网络编程原理、方法和技巧,开发了基于UDP/IP协议的服务器和客户机通信程序。将数据封装成结构体进行传输,接收到数据后将其内存拷贝到接收结构体中进行解析并驱动相应虚拟仪表。对每个图形显示更新周期中网络通信收发次数进行试验,给出了在GL Studio中实现实时、可靠通信的策略。研究结果为直升机计算机辅助训练系统由单机模式向多仿真计算机协同训练模式发展提供可借鉴的技术基础,具有一定应用前景。

关键词:计算机辅助训练;GL Studio开发工具;网络套接字;非阻塞;实时性

中图分类号:V275⁺.1

文献标识码:A

文章编号:1673-629X(2011)08-0185-03

Design of Communication Module in Helicopter Cockpit Simulation

SHI Chun-hu¹, JIANG Jian-fang¹, QU Hong-xing², CHEN Lei²

(1. School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China;

2. General Staff Army's Aviation Institute, Beijing 101121, China)

Abstract: In order to realize data communication between multiple simulation computers in helicopter cockpit simulation, Windows Sockets technology is integrated in GL Studio. Utilizing Client/Server mode and Socket network programming method, UDP/IP protocol communication model is developed. Data are packed into struct to transfer. Received data are memory copied to receive struct to drive relative virtual instrument. Experiments are done about receive and send times in communication to ensure data transfer real-time and reliably. The research results provide referenced technical foundation for Computer-based Training System which is concerting from single machine model to multi-machine collaborative training model, and has a certain prospect.

Key words: computer-based training; GL Studio developing tool; Windows Socket; non-blocking; real-time

0 引言

计算机辅助训练系统(CBT, computer-based training)主要采用计算机仿真、计算机控制、图像处理显示等高新技术,实现一个融图形、图像、文字、曲线、图表、声音为一体的多媒体仿真平台^[1]。直升机计算机辅助训练系统主要用于实现直升机飞行员各项模拟操作训练,包括通电检查、空域飞行、悬停起降、特情处理等。

直升机虚拟座舱是直升机计算机辅助训练系统中最重要的仿真模块。直升机座舱中的仪表板和中央操纵台上的分立仪表、控制盒均为触摸屏响应的虚拟仪表,在仿真中用GL Studio进行虚拟仪表开发。座舱中的结构架台、座椅、飞行操作联动机构、驾驶杆、总距

杆等均做成硬件结构,通过网络将硬件机构的控制信号传给仿真计算机进行处理。由三维视景模型和环境音响来实时模拟飞行环境。系统在多种技术综合开发的基础上建立,关键技术包括OpenGL、Vega Prime、DirectX、Winsock以及Windows底层的API和COM对象。

直升机虚拟座舱中的网络通信模块可在GL Studio中集成开发,主要负责仪表板仿真计算机、中央操纵台仿真计算机及其他仿真计算机之间的数据通信,完成各仿真计算机间的网络连接和数据传输。

1 直升机虚拟座舱中通信协议选择

直升机虚拟座舱中使用Windows Socket(套接字)进行网络通信模块设计。使用套接字编写的通讯程序,可以较好实现两个网络进程之间的数据交互。两种典型的网络套接字类型为:

1)流式套接字(SOCK_STREAM)。提供面向连接、可靠的数据传输服务,数据无差错、无重复的发送,

收稿日期:2011-01-02;修回日期:2011-04-07

基金项目:国家大学生创新性实验计划项目(081028815)

作者简介:石春虎(1986-),男,江苏大丰人,硕士研究生,研究方向为计算机仿真技术;姜建芳,教授,硕士,研究方向为计算机仿真技术、工控机与网络。

且按发送顺序接收。流式套接字是基于 TCP 协议实现的。

2) 数据报式套接字 (SOCK_DGRAM)。提供无连接服务。数据报以独立包形式发送, 不提供无错保证, 数据可能丢失或重复。数据报式套接字是基于 UDP 协议实现的^[2,3]。

考虑到仿真座舱各仿真计算机需要不断地传递数据, 数据会周期传输更新, TCP 的重传机制并没有体现出优势。另外, GL Studio 画面还需要实时刷新显示。为了保证数据能高速收发和系统实时性, 虚拟座舱中的仪表板仿真计算机和中央操纵台仿真计算机和其他仿真计算机之间采用 UDP 协议交换数据。

GL Studio 的 calculate 函数每 30ms 执行一次, 仿真计算机刷新画面的频率一般为 30 帧, 作为 100Mbps 的网卡, 一秒钟能传输仿真数据的次数为:

$$100\ 000\ 000 * 0.20 / (160 * 4 * 8) = 3906 \gg 30$$

(1)

其中 0.20 是指 100M 网卡的利用率; 160 是指客户端一次发送和接收共 160 个浮点型数据; 4 是指每个浮点数占据的字节数; 8 是指将字节转化为比特位数^[4,5]。

网络通信采用客户端/服务器模式, 其中仪表板仿真计算机作为 Server 端, 中央操纵台仿真计算机和其他仿真主机作为 Client 端。从公式(1)得出, 在中央操纵台和其他仿真计算机 GL Studio 的 calculate 函数中添加 UDP 客户端代码, 执行一次数据发送和接收, 完全可以保证系统数据可靠传输和更新; 在仪表板仿真计算机 GL Studio 的 calculate 函数中添加 UDP 服务器端代码, 执行 n (n 为客户端数量) 次数据接收和发送, 至多支持 $3906/30 \approx 130$ 个客户端同时运行。

2 直升机虚拟座舱中通信模块的实现

2.1 网络通信流程

通过上一节的讨论, 选择基于 UDP 协议进行直升机虚拟座舱通信模块设计。网络通信的两个进程间采用客户端/服务器模式 (Client/Server)。Socket 用来实现客户端和服务端的连接, 连接完成后, 两个程序之间就建立了一个虚拟的数据通信链路, 可以直接从自己打开的套接字读入和写数据, 而不用关心实际的数据链路^[6,7]。

在 GL Studio 中已经默认添加了 winsock2.h 头文件和 WS2_32.lib 链接, 开发网络程序时可直接使用 Socket 函数。为了编程和通讯的方便, 在系统开发时需要对传递的接口数据的形式进行协议, 保证数据的唯一性和可解释性。由于结构体对象在内存中分配的空间都是连续的, 所以可以将整个结构体直接转化成

字符串发送, 到了接收方再将这个字符串还原成结构体。当然, 这个结构体在发送方与接收方都必须声明。

如果在 GL Studio 的周期函数 calculate 中写一段死循环作 UDP 通信, GL Studio 画面刷新程序将得不到执行, 而变成画面停滞。另外, UDP 通信必须工作在非阻塞方式, 否则网络故障时接收函数 recvfrom 将阻塞 GL Studio 程序的运行同样导致画面停滞。

综合以上分析, 最终确定直升机虚拟座舱周期通信流程如下 (见图 1):

(1) 每个 calculate 周期, 客户端发送 1 次数据到服务器, 并读取 2 次接收缓冲区, 确保接收缓冲区的最新数据被有效取出, 将读到的数据拷贝到结构体中进行处理。

(2) 每个 calculate 周期, 服务器端读取 ($n+1$) 次接收缓冲区, 确保接收缓冲区的最新数据被有效取出, 并根据客户端的 ID, 把接收到的数据拷贝到对应的接收结构体中; 服务器端把客户端需要的数据回发给它。

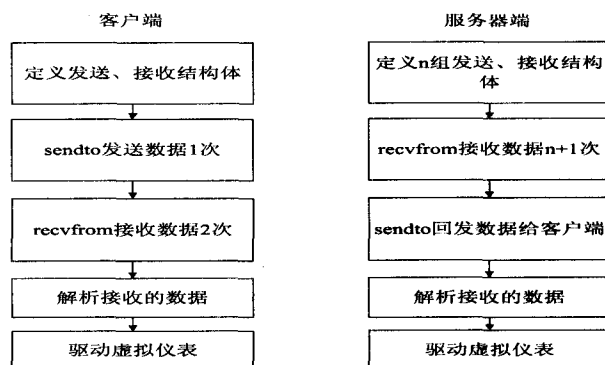


图 1 直升机虚拟座舱周期通信流程图

2.2 客户端程序设计

在 GL Studio 编辑器中创建全局变量, UDPRecvBuf1, UDPSendBuf1, 分别作为客户端 1 的发送结构体和接收结构体。在 GL Studio 的 Initialize 函数中添加初始化代码, 加载套接字库并创建套接字, 并设置套接字工作在非阻塞方式:

```
u_long mode = 1;
ioctlsocket(sock, FIONBIO, &mode); // 默认为阻塞方式
```

在 GL Studio 的 calculate 函数中添加代码, 对客户端发送结构体中的数据进行赋值更新:

```
UDPSendBuf1.bFuelGaugeTest = _newbFuelGaugeTest;
.....
```

并用 sendto 函数将发送结构体强制转化为字符类型发送出去:

```
sendto(sockClient, (char *) &UDPSendBuf1, sizeof(_UDPSendBuf1), 0, (SOCKADDR *) &addrSrv, len);
```

用 recvfrom 函数接收数据, 并将收到的数据拷贝到接收结构体中:

```
char recvBuf[320];
```

```

for(int i=0;i<2;i++)
{
    if ( recvfrom ( sockClient, recvBuf, 320, 0, ( SOCKADDR * )
&addrSrv,&len) !=SOCKET_ERROR)
        memcpy( &UDPrecvBuf1,recvBuf,sizeof( recvBuf) );
}

    处理服务器端发来的数据,驱动相应的虚拟仪表:
    bPower = UDPrecvBuf1. bPower;
    _bJianCha = UDPrecvBuf1. bJianCha;
    .....

```

在客户端 *.h 文件中添加析构函数的声明,在 *.cpp 文件中添加析构函数的实现代码。调用 close_socket 函数关闭套接字,释放为该套接字分配的资源。然后调用 WSACleanup 函数,终止对套接字库的使用。

2.3 服务器端程序设计

在 GL Studio 编辑器中创建 n (n 为客户端数量) 组全局的结构体。UDPrecvBuf1, UDPSendBuf1, UDPre-
cvBuf2, UDPSendBuf2, ..., 在 GL Studio 的 Initialize 函数中添加初始化代码,加载套接字库、创建套接字并进行绑定、设置套接字工作在非阻塞状态:

```

bind( sockSrv, ( SOCKADDR * ) &addrSrv, sizeof( SOCKADDR) );
//绑定套接字

```

```

u_long mode = 1;
ioctlsocket( sockSrv, FIONBIO, &mode); //使能非阻塞模式

```

在 GL Studio 的 calculate 函数中添加代码,对服务器端发送结构体中的数据进行赋值更新:

```

UDPSendBuf1. bPower = _newbPower;
UDPSendBuf1. bJianCha = _newbJianCha;
.....

```

用 recvfrom 函数接收数据,执行 ($n+1$) 次,确保缓冲区的数据被有效读出。并根据 recvbuf[0] 判断是客户端 ID 号,并将收到的数据拷贝到对应的接收结构体中。同时,用 sendto 函数中将发送结构体强制转化为字符类型回发给相应的客户端:

```

if ( recvfrom ( sockSrv, recvBuf, 320, 0, ( SOCKADDR * )
&addrClient,&len) !=SOCKET_ERROR)
{
    switch( recvBuf[0])
    {
        case 1: memcpy( &UDPrecvBuf1,recvBuf,sizeof( recvBuf) );
        sendto( sockSrv, ( char * ) &UDPSendBuf1, sizeof( _UDPSend-
Buf1), 0, ( SOCKADDR * ) &addrClient, len); break;
        case 2: memcpy( &UDPrecvBuf2,recvBuf,sizeof( recvBuf) );
        sendto( sockSrv, ( char * ) &UDPSendBuf2, sizeof( _UDPSend-
Buf2), 0, ( SOCKADDR * ) &addrClient, len); break;
    }
}

```

```

    处理客户端发来的数据,驱动相应的虚拟仪表:
    _bFuelGaugeTest = UDPrecvBuf1. bFuelGaugeTest;

```

.....

在服务器端 *.h 文件中添加析构函数的声明,在 *.cpp 文件中添加析构函数的实现代码。调用 close_socket 函数关闭套接字,释放为该套接字分配的资源。然后调用 WSACleanup 函数,终止对套接字库的使用。

3 直升机虚拟座舱中通信模块实时性研究

在分布式虚拟仪表训练系统中,信号的实时传输是仪表准确响应的前提,只有仪表准确响应,才能保证虚拟训练的真实性。直升机计算机辅助训练系统中虚拟仪表仿真计算机通过数据接口实时接收其他仿真计算机的数据并不断更新当前的画面显示^[8-11]。

理想的状态下,客户端每个 calculate 函数执行周期内发送 1 次 UDP 数据给服务器端,并读取 1 次接收缓冲区;服务器每个 calculate 函数执行周期以非阻塞方式读写 n (n 为客户端数量) 次网口,恰好完成数据通信任务。但是,如果读取接收缓冲区数据不及时,接收缓冲区内的数据会越来越多。Socket 的默认缓冲区大小为 8k,存满后有 25 个结构体。最终,服务器从接收缓冲区取走 1 个最早的结构体数据后,才能接收 1 个新的结构体。服务器从接收缓冲区取出的结构体数据与客户端的最新数据之间的时间差约为:

$$25 * 30\text{ms} = 750\text{ms} \quad (2)$$

其中,30ms 为 calculate 的执行周期,25 为接收缓冲区满时的结构体数。公式(2)得出,数据更新延迟最大约 750ms。

试验证明,在理想状态下,如果断开网络并重新连接,系统更新延时比较严重,约 1s。为保证数据更新的实时性,服务器每个 calculate 函数执行周期内读取缓冲区的次数必须大于客户端发送数据的次数和。即如果有 n 个客户端每个周期内发送 1 次数据,则服务器端每个周期读取缓冲区的次数必须大于等于 ($n+1$),这样就能保证缓冲区内的未读结构体个数是收敛的,最终每次服务器读取到的数据都是客户端最新发来的。同样地,客户端每个 calculate 函数执行周期读取接收缓冲区的数据设为 2,发送数据次数仍为 1 次不变,保证客户端接收缓冲区不会满,从而保证了客户端数据更新的实时性。

4 结束语

在虚拟仪表仿真平台 GL Studio 开发虚拟仪表的基础上,结合 VC++,集成开发 Windows Socket,为计算机辅助训练系统由单机模式向多仿真主机协同训练模式提供可借鉴的技术基础。采用文中提供的网络通信方案进行多仿真主机之间网络通信,数据传输稳定顺

(下转第 191 页)

结构描述转换成以逻辑和意义为基础的语义表示。

(3) 语用分析和信息提取,根据知识库中预先存放的领域知识在上下文中进行推理,并将提取出来的关键内容存入一张信息表。

(4) 文本生成,将信息表中的内容转换为一段完整连贯的文字输出。

用这种方法生成的文摘性能最好,但是它很大程度上依托于自然语言处理技术,自然语言处理中句法分析、语法语义分析技术尚未完全成熟,因此如果想获得高质量的语言分析结果,就必须将待处理的语料限制在某个范围之内。一旦限制在某个领域内,文摘的可移植性就差。随着自然语言处理技术中句法、语法语义分析技术的成熟,自动文摘的性能会有相当大的提高。

2 结束语

综上所述,在自动文摘的三种方法中,最主要的方法还是自动摘录方法,用计算句子权值、计算句子相似度和聚类的方法来抽取句子形成文摘,但是这种文摘往往不连贯,就产生了基于信息抽取的方法,虽然这种方法生成的文摘连贯性好但是它完全依赖于领域知识而且生成的文摘千篇一律,这样的文摘性能也不好。基于理解的方法能生成质量相当高的文摘,它的发展很大程度上依赖于中文信息处理技术,随着中文信息处理技术的发展,生成自动文摘性能会很好。

参考文献:

- [1] Luhn H P. The automatic creation of literature abstract[J]. IBM Journal of Research and Development, 1958, 2(2): 159-165.

(上接第 187 页)

畅,且系统的实时性较高。实践证明,满足直升机计算机辅助训练系统的仿真需求。

参考文献:

- [1] 陈东帆. 航空 CBT 中协同训练系统的设计与实现[J]. 计算机工程与设计, 2007, 28(15): 3727-3730.
- [2] 刘丽娇. 基于 GL-Studio 的飞行模拟机虚拟座舱开发[D]. 哈尔滨: 哈尔滨工业大学机电工程学院, 2009.
- [3] 孙鑫, 余安萍. VC++ 深入详解[M]. 北京: 电子工业出版社, 2008.
- [4] 王大勇. 基于 VAPS 下虚拟仪表开发[D]. 哈尔滨: 哈尔滨工业大学机电工程学院, 2006.
- [5] 朱敏, 陈立奎, 王宏伟, 等. 基于 GL Studio 的分布式虚拟训练系统关键技术[J]. 兵工自动化, 2010, 29(8): 46-48.

- [2] 陶余会, 周水庚, 关信红. 一种基于文本单元关联网络的自动文摘方法[J]. 模式识别与人工智能, 2009, 22(3): 441-443.
- [3] 王志琪. 基于互增强关系的自动文摘句子加权方法[J]. 上海交通大学学报, 2007, 41(8): 1298-1299.
- [4] 杨建林. 一种使用自动聚类思想的自动文摘方法[J]. 情报学报, 2001, 20(5): 534-535.
- [5] 纪文倩, 李舟军, 巢文涵, 等. 一种基于 LexRank 算法的改进的自动文摘系统[J]. 计算机科学, 2010, 37(5): 152-153.
- [6] 王萌, 何婷婷, 姬东鸿, 等. 基于 HowNet 概念获取的中文自动文摘系统[J]. 中文信息学报, 2004, 19(3): 90-91.
- [7] 柴晓丽, 张丽伟, 管玉玲. 基于 HowNet 自动文摘的研究[J]. 电脑编程技巧与维护, 2003, 12(3): 164-165.
- [8] 郭庆琳, 樊孝忠, 柳长安. 文本聚类在自动文摘中的应用研究[J]. 计算机应用, 2005, 25(5): 1037-1038.
- [9] Salton G, Allan J, Buckleym C, et al. Automatic Analysis, Theme Generation, and Summarization of Machine Readable Texts[J]. Science, 1994, 264(3): 1421-1426.
- [10] Salton G, Allan J, Singhal A. Automatic Text Decomposition and Structuring[J]. Information Processing & Management, 1996, 32(2): 127-138.
- [11] Salton G, Singhal A, Mitra M, et al. Automatic Text Structuring and Summarization[J]. Information Processing & Management, 1997, 33(2): 193-207.
- [12] 万敏. 基于统计和语义分析的中英文自动文摘研究[D]. 北京: 清华大学, 2003.
- [13] 傅间莲, 陈群秀. 自动文摘系统中的主题划分问题研究[J]. 中文信息学报, 2005, 19(6): 29-32.
- [14] 刘挺, 吴岩, 王开铸. 基于信息抽取和文本生成的自动文摘系统设计[J]. 情报学报, 1997, 16(增刊): 25-28.
- [15] 刘挺, 王开涛. 自动文摘的四种主要方法[J]. 情报学报, 1999, 18(1): 11-16.

- [6] 罗正卫, 刘建群. Windows(客户端)和 MS-DOS(服务器)网络通信的实现[J]. 计算机技术与发展, 2010, 20(4): 191-194.
- [7] 王恩涛, 李祥. 基于 Socket 的手机与数据库服务器通信的研究[J]. 计算机技术与发展, 2007, 17(2): 81-84.
- [8] 王俊鸣, 张智军, 张安旭. 基于 LabWindows/CVI 的多线程技术的电磁兼容预测试系统设计与实现[J]. 弹箭与制导学报, 2008, 28(2): 311-314.
- [9] 雷振山, 赵晨光. 虚拟仪器系统的网络技术研究与应用[J]. 国外电子测量技术, 2006, 25(5): 59-61.
- [10] 杨志菊, 李洋. GL Studio 在武器系统仿真模拟中的应用[J]. 电子测试, 2010(8): 80-86.
- [11] 邱岳恒, 卢京潮, 刘秉. 直升机视景仿真及座舱仪表显示系统实现[J]. 测控技术, 2010, 29(7): 13-15.