

基于 VC++ 和 Matlab 的数字信号内插处理系统

张秀再

(南京信息工程大学, 江苏 南京 210044)

摘要:利用 VC++ 编程平台的功能强大和灵活性,并结合 Matlab 的运算函数丰富和编程效率高等优点实现混合编程,可以快速开发出功能强大的应用系统。文中以基于 VC++ 和 Matlab 混合编程的数字信号内插处理系统为例,介绍了在 VC++ 开发环境中使用两种 Matlab 引擎调用方法实现混合编程。内插处理系统中读取数据的长度和内插倍数的大小都可以在用户交互界面任意设置,具有较好的灵活性。将 VC++ 和 Matlab 结合使用既发挥了各自的优点又提高了系统开发的效率。

关键词:VC++ ;Matlab;N 倍;一维内插

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2009)01-0109-04

System of Data Signal Interpolation Processing Based on VC++ and Matlab

ZHANG Xiu-zai

(Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract: In order to design the powerful application system quickly, may combine the powerful and flexible programming platform of VC++ and the operational function rich and higher programming efficiency advantages of Matlab. Take the data signal interpolation processing system based on mixed programming with VC++ and Matlab as an example, introduced the mixed programming is realized by using engine with two ways in the environment of VC++. The length of input data and the value of interpolation times all are set on the user contact surface in the system, which has a good flexibility. The mixed programming with VC++ and Matlab not only display those merits but also enhance the efficiency of system development.

Key words: VC++ ;Matlab;n times;one-dimensional interpolation

0 引言

目前,利用并行交替式(Parallal interleaving)技术,超高速数字化示波器的实时采样速率已突破了 10Gsa/s。按照 Nyquist 定理,这种系统的实时带宽可接近 5GHz。在数字化示波器中,由实时采样获得的样本数据原封不动地显示时,因采样理论和显示技术等缘故,可能会产生不同形式的视觉混淆。对于数字化示波器而言,除了要求有高的实时采样速率外,还要求有较高的波形分析细节^[1]。为此需要在采集的数据样本之间进行插值。

Matlab 在矩阵运算,一维、二维和三维图形的绘制,数值拟合等方面均有极强的功能,已被广泛运用于数字信号处理、图像处理等领域,具有极好的可读性和

可移植性。Matlab 的突出特点是简洁,它利用丰富的函数资源,给用户带来最直观、最简洁的程序开发环境,但 Matlab 在用户接口上有着交互性差的缺点。而 VC++ 可以开发基于 MFC 以及基于 Windows API 的 Win32 程序,其特点是代码效率高、执行速度快,可以开发出风格多样的操作界面,其美中不足是开发信号处理算法时代码编写工作量很大。因此,VC++ 开发工具和 Matlab 混合编程的信号处理系统,既利用了 VC++ 的面向对象特性和功能强大的 MFC 来开发专业级的操作界面,又可以方便地使用 Matlab 提供的信号处理函数完成对数字信号的一维插值处理。

因工程需要,文中提到的内插方法是利用 Matlab 的库函数 interp1() 进行三次多项式插值,而不用涉及到复杂的多相滤波器^[2]。该方法简单、有效。

收稿日期:2008-04-30

基金项目:江苏省高新技术产业发展计划项目(JH02-121)

作者简介:张秀再(1979-),男,讲师,研究方向为雷达信号处理、信号与信息处理。

1 系统功能和程序流程

当执行工程程序时启动主对话框(见图 1),由主

对话框设置键启动读取长度和内插倍数设置对话框,在读取长度和内插倍数设置对话框中设置需要读取的数据点长度和内插的倍数(见图 2)。然后启动 Matlab 引擎程序进行后台处理,读取数据文件进行内插处理,并显示相应的数据波形,最后按路径保存处理后的文件。关闭 Matlab 引擎,退出 Matlab 程序。用户交互界面和程序流程图见图 3。

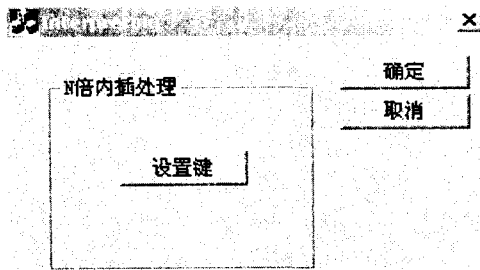


图 1 主对话框

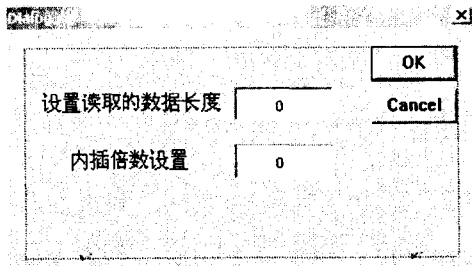


图 2 读取长度和内插倍数设置对话框

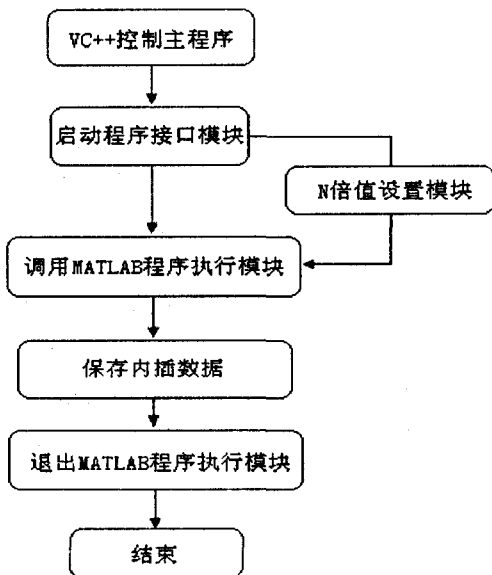


图 3 内插处理程序流程图

2 引擎方法调用 Matlab 的开发环境配置和关键程序

混合编程的开发方式采用在 VC++ 环境下调用 Matlab 引擎(在 VC 6.0 和 Matlab 6.5 环境下)来实现。

所谓 Matlab 引擎(Matlab Engine),是指一组 Matlab 提供的接口函数(Engine API 函数),它采用客户机/服务器计算模式,并且支持 C 语言。具体在 C++ 中使用 Matlab 引擎来编制程序时,C++ 是作为客户端出现的,即负责用户接口和提出数据请求;而 Matlab 则属于服务器端,即向客户端提供数据服务。利用 Matlab 引擎调用工具箱中的函数可以节省大量的系统资源^[3],C++ 客户端通过调用 Matlab 引擎与 Matlab 服务器建立连接,从而实现动态通信^[4]。混合编程编译接口环境的配置主要有以下两步(假设 Matlab 安装在目录 C:\Matlab6p5 下)。

1) 添加 Matlab 引擎库的头文件和库函数的路径。

在 VC++ 6.0 开发环境中,打开菜单“Tools”选取“Options... -> Directories”。在“Show directories for”选项卡选取“Include files”,添加“C:\Matlab6p5\EXTERN\INCLUDE”。这是 Matlab 引擎库的头文件 engine.h 所在的目录。然后在“Show directories for”中选取“Library Files”,添加“C:\Matlab6p5\EXTERN\LIB\WIN32\MICROSOFT\MSVC60”,因为这些引擎库用到的动态链接库都在该目录下。

2) 完成 Matlab 引擎对应的静态链接库的导入。

在 VC++ 6.0 开发环境中,选择“Project”菜单中的“Add to Project”命令,然后选择 File 选项,将“C:\Matlab6p5\EXTERN\LIB\WIN32\MICROSOFT\MSVC60”目录下的 libmx.lib、libmex.lib、libeng.lib 文件添加到工程中^[5]。

在主对话框的 cpp 文件中添加 engine.h 头文件。主对话框的设置键按钮单击响应事件关键代码如下:

```

Csetn dlg;
dlg.DoModal();
int n = dlg.m_ setvalue; //设置内插倍数并赋值给 n
int length = dlg.m_ setlength; //设置读取数据长度并赋值给 length
Engine * ep; //定义 Matlab 引擎变量
MessageBox(“单击确定按钮,Windows 正在准备启动 Matlab 引擎!”,“Matlab 引擎”, MB_OK | MB_ICONINFORMATION);
if (! (ep = engOpen(“\0”))) //打开 Matlab 引擎
{
    fprintf(stderr, “\n Matlab 引擎启动失败! \n”);
    MessageBox(“Matlab 引擎启动失败!”,“Matlab 引擎”, MB_OK | MB_ICONERROR);
    exit(-1);
}
mxArray * N = NULL;
mxArray * L = NULL; //为把数据长度和内插倍数由正型转换成数组定义两个数组
double Nnum[1] = {n};

```

```

double Lnum[1] = {length};
N = mxCreateDoubleMatrix(1, 1, mxREAL);
L = mxCreateDoubleMatrix(1, 1, mxREAL); //实现从 int 到 matrix 的转化生成矩阵
memcpy((void *) mxGetPr(N), (void *) Nnum, sizeof(Nnum));
memcpy((void *) mxGetPr(L), (void *) Lnum, sizeof(Lnum));
//***通过 Matlab 引擎执行 Matlab 命令***
engPutVariable(ep, "N", N);
engPutVariable(ep, "L", L); //将矩阵变量写入 Matlab 引擎的工作区
engEvalString(ep, "[FILENAME, PATHNAME] = uigetfile('*.dat', 'open new format VLR data file');"); //选择路径和文件名
engEvalString(ep, "fid = fopen([PATHNAME FILENAME], 'r');"); //打开文件
engEvalString(ep, "[a, n] = fread(fid, L, 'double = > double');");
//按 double 类型读取 L(为设置长度)个数据
engEvalString(ep, "plot(a);");
engEvalString(ep, "title('原数据');");
engEvalString(ep, "xlabel('n');");
engEvalString(ep, "ylabel('y(n)');");
engEvalString(ep, "T = 1:length(a);");
engEvalString(ep, "NT = 1:1/N:length(a);"); //对信号进行一维 N 倍三次多项式插值
engEvalString(ep, "b = interp1(T, a, NT, 'cubic');");
engEvalString(ep, "figure;"); //重新打开一个绘图窗口
engEvalString(ep, "plot(b);"); //画出内插以后的数据图形
engEvalString(ep, "plot(a);");
engEvalString(ep, "title('内插 4 倍原数据');");
engEvalString(ep, "xlabel('n');");
engEvalString(ep, "ylabel('y(n)');");
engEvalString(ep, "fid1 = fopen('E:\ \ new.dat', 'w');"); //设置保存路径和文件名
engEvalString(ep, "fwrite(fid1, b, 'double');"); //保存插值以后的数据为新文件
MessageBox("关闭 Matlab 引擎,系统将退出 Matlab 应用程序!", "Matlab 引擎", MB_OK | MB_ICONINFORMATION);
engClose(ep); //关闭 Matlab 引擎,退出 Matlab

```

3 引擎类调用 Matlab 的方法和关键程序

为了进一步简化 Matlab 引擎的使用,更加符合基于 MFC 的习惯,这里用了引擎类的方法进行同样的内插处理。通过 CMatlabEng 类使用 Matlab 引擎功能,必须在 Visual C++ 工程进行以下编译环境设置:

(1) 添加 matlab.h 到 stdafx.h 或者需要的接口和执行文件中。

(2) 添加库文件到 Visual C++ 工程中。根据具体情况,可能需要添加的库文件有 libeng.lib, libmx.

lib, libmatlb.lib, libmat.lib 和 libmex.lib。

(3) 添加 CMatlabEng 类的定义文件 MatlabEng.h 和实现文件 MatlabEng.cpp 到工程中^[3]。

在主对话框的 cpp 文件中添加 MatlabEng.h 头文件。主对话框的设置键按钮单击响应事件关键代码如下:

```

Csetn dlg;
dlg.DoModal();
int n = dlg.m_setvalue; //设置内插倍数并赋值给 n
int length = dlg.m_setlength; //设置读取数据长度并赋值给 length
CMatlabEng matlab; //定义 Matlab 引擎类变量
MessageBox("启动 Matlab 引擎,请稍候...");
matlab.Open(NULL);
mxAarray * N = NULL;
mxAarray * L = NULL; //为把数据长度和内插倍数由正型转换成数组定义两个数组
double Nnum[1] = {n};
double Lnum[1] = {length};
N = mxCreateDoubleMatrix(1, 1, mxREAL);
L = mxCreateDoubleMatrix(1, 1, mxREAL); //实现从 int 到 matrix 的转化
memcpy((void *) mxGetPr(N), (void *) Nnum, sizeof(Nnum));
memcpy((void *) mxGetPr(L), (void *) Lnum, sizeof(Lnum));
matlab.PutVariable("N", N);
matlab.PutVariable("L", L); //将矩阵变量写入 Matlab 引擎的工作区
matlab.EvalString("[FILENAME, PATHNAME] = uigetfile('*.dat', 'open new format VLR data file');"); //选择路径和文件名
matlab.EvalString("fid = fopen([PATHNAME FILENAME], 'r');"); //打开文件
matlab.EvalString("[a, n] = fread(fid, L, 'double = > double');");
//按 double 类型读取 L(为设置长度)个数据点
matlab.EvalString("plot(a);"); //画出读取的数据图形
matlab.EvalString("title('原数据');");
matlab.EvalString("xlabel('n');");
matlab.EvalString("ylabel('y(n)');");
matlab.EvalString("T = 1:length(a);");
matlab.EvalString("NT = 1:1/N:length(a);"); //进行一维 N 倍三次多项式插值
matlab.EvalString("b = interp1(T, a, NT, 'cubic');");
matlab.EvalString("figure;"); //重新打开一个绘图窗口
matlab.EvalString("plot(b);"); //画出内插以后的数据图形
matlab.EvalString("title('内插 4 倍后的数据');");
matlab.EvalString("xlabel('n');");
matlab.EvalString("ylabel('y(n)');");

```

```
matlab.EvalString("fidl = fopen('E: \ \ 333. dat', 'w');");//
设置保存成路径
matlab.EvalString("fwrite(fidl,b,'double');");//保存插值数据
为新文件
MessageBox("关闭 Matlab 引擎,系统将退出 Matlab 应用程序!",
"Matlab 引擎", MB_OK | MB_ICONINFORMATION);
matlab.Close();//关闭 Matlab 引擎,退出 Matlab
```

4 内插前后波形实例

两种引擎使用方法不同,但是混合编程的执行结果是相同的。这里仅给出选取 20 个数据点的 4 倍和 40 倍内插处理的结果,如图 4 为内插前波形,图 5 为 4

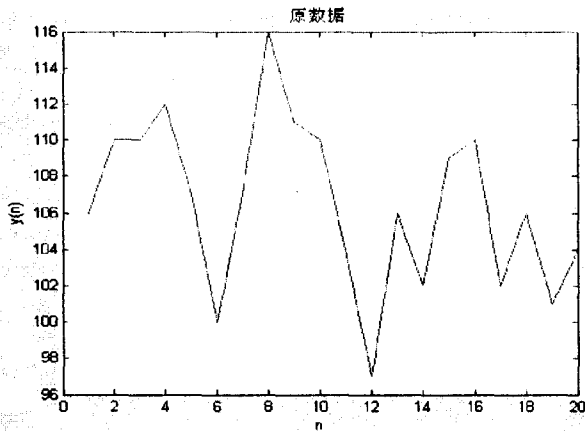


图 4 内插前波形

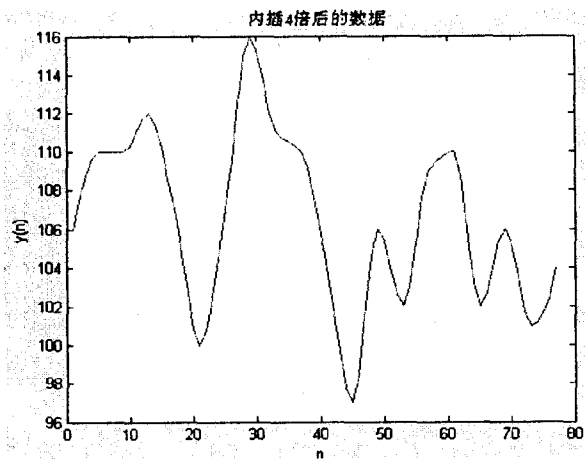


图 5 4 倍内插后波形

倍内插后波形,图 6 为 40 倍内插后波形。

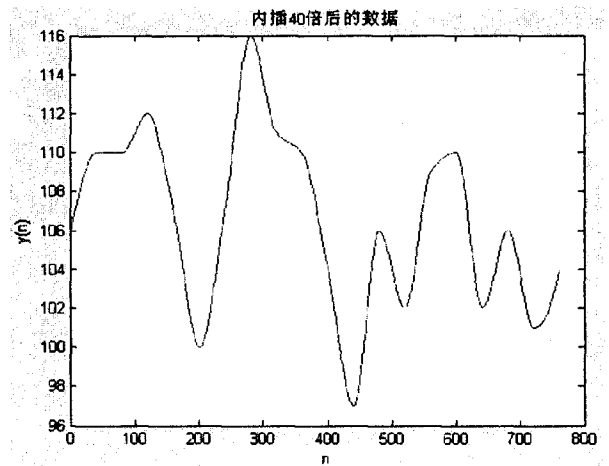


图 6 40 倍内插后波形

5 结束语

将 Matlab 与 VC++ 结合起来,充分发挥它们各自的优点,可以建立一个开放式的、编程效率极高的开发环境[6]。从而可以降低开发难度,缩短编程时间。该系统具有良好的用户界面,较好的处理效果,并且在实际应用中验证了系统的可行性和优越性。

参考文献:

- [1] 林茂六,尹宝智,刘治宇. 高速采样信号数字内插理论与正弦内插算法研究[J]. 电子学报,2002(12):8-10.
- [2] Gochiere R E, Rrabiner L. Interpolation and decimation of digital signals J. ATutorial Review, Proc, IEEE, March, 1981, 69(3):300-320.
- [3] 罗 迅,王 虹,周天庆,等. Visual C++ 与 Matlab 相结合在编程中的应用[J],2003(3):25-26.
- [4] 谢佩军,计时鸣,张 利. VC++ 与 Matlab 混合编程的探讨[J]. 计算机应用与软件,2006,23(2):128-130.
- [5] 杨高波. 精通 Matlab7.0 混合编程[M]. 北京:电子工业出版社,2006.
- [6] 王周益,刘继兴,柳长安. VC++ 与 Matlab 混合编程研究及开发实例[J]. 计算机应用研究,2006(5):154-156.

(上接第 108 页)

ization Functional in Regularized Image Restoration[J]. IEEE Trans on Image Processing,1995,4(5):594-602.

- [6] 肖庭延,于慎根,王延飞. 反问题的数值解法[M]. 北京:科学出版社,2003.
- [7] 邹学文,王宝娥,寇 婷,等. 基于遗传算法获取正则参数的一种新方法[J]. 科学导报,2006,24(2):46-49.
- [8] Kennedy J, Eberhart R. Particle swarm optimization[C]//

Proc of IEEE International Conference on Neural networks. Piscataway, NJ: IEEE Service Center,1995:1942-1948.

- [9] 杨 维,李歧强. 粒子群优化算法综述[J]. 中国工程科学. 2004,6(5):87-94.
- [10] Morozov V A. Methods for solving incorrectly posed problems [M]. New York: Springer-Verlag,1984.