

【基础·应用】

VC++环境下调用 MATLAB 程序实现数字信号处理的方法

王炳和 梁根池

(武警工程学院 通信工程系, 陕西 西安 710086)

【摘要】阐述了在 VC++ 环境下如何调用 Matlab 程序进行数字信号处理, 以 Matlab 工具箱中功率谱密度分析函数为例, 介绍了通过 Matlab 自带的引擎和 Matlab 自身的编译器对工具箱函数的调用。

【关键词】Matlab; M-文件; 引擎; 编译器; VC++

MATLAB 是美国 MathWorks 公司自 80 年代中期推出的数学软件, 它在矩阵运算、二维和三维图形的绘制、数值拟合等方面均有强大的功能, 已被应用于数字信号处理各个领域。MATLAB 的突出优点是简洁, 它利用其丰富的函数资源, 给用户带来最直观、最简洁的程序开发环境^[1]。Visual C++ 系列产品是开发 Windows 应用程序的有力工具, 而 Visual C++ 6.0 则是 Microsoft 公司发布的成熟版本。它是目前综合性高、功能最强大、也是最复杂的软件开发工具之一, 应用极为广泛^[2]。

但是由于 Matlab 在程序编写方面的一些局限性, 使得它在处理许多问题时缺乏灵活性, 因此 VC++ 与 Matlab 的混合编程将是一种极其强大的编程手段。Matlab 的信号处理工具箱是信号算法文件的集合, 它处理的基本对象是信号与系统, 信号处理工具箱位于: 目录 \ Toolbox \ Signal 下, 利用工具箱中的文件可以实现信号的变换、滤波、谱估计、滤波器设计等。在其他的环境如 VC++ 下, 如果能调用 Matlab 工具箱中的文件, 会大大地加快一些算法的实现, 同时其可靠性也很高。

1 利用 Matlab 引擎

1.1 Matlab engine 的介绍

Matlab 计算引擎是一系列允许用户在别的程序中与 Matlab 交互的函数库和程序库。在调用的过程中, Matlab 引擎函数库工作在后台, 应用程序进行通信, Matlab 引擎采用客户和服务器计算方式。在运用中, VC++ 的 C++ 语言的程序作为前端客户机, 它向 Matlab 引擎传递命令和数据信息, 并从 Matlab 引擎接收数据信息, 它提供了下列几个函数: engOpen, engGetArray, engPutArray, engEvalString, engOutputBuffer, engClose 与客户机进行交互。

1.2 Matlab engine 库函数

Matlab engine 库中主要使用函数如下:

- (1) engOpen: 启动 Matlab engine
- (2) engGetArray: 从 Matlab 引擎的工作区中获取一个矩阵
- (3) engPutArray: 将矩阵送到 Matlab 计算引擎的工作区中去
- (4) engEvalString: 执行一个字符串表示的 Matlab 表达式
- (5) engOutputBuffer: 确定存放 Matlab 输出结果的缓冲区
- (6) engClose: 关闭 Matlab 引擎

1.3 VC++ 调用 Matlab 的实现方案

在 Windows 系统中, DLL 是一种很特别的可执行文件, 可以被多个 Windows 应用程序同时访问, 具有

收稿日期: 2007-04-25

作者简介: 王炳和(1964.09-), 男, 陕西岐山人, 1999年毕业于西北工业大学获工学博士学位, 现为武警工程学院通信工程系教授; 梁根池(1983.04-), 广东江门人, 现为武警工程学院通信工程系通信与信息系统专业硕士研究生。

固定的共享数据段。该数据段的数据在 DLL 被 Windows 下载前一直保留在内存中,因此可以通过 DLL 实现用户程序与 Matlab 之间的数据传输和调用。

下面例程是在 VC++ 下建一个基于对话框的应用程序,在对话框中设置一个 Button 控件 OnMatlabEngine,在对话框 .cpp 文件中加入“engine.h”和“math.h”头文件,下面给出部分程序清单。

```
Void CtestmatlabDlg::OnMatlabEngine()
{
    engine * ep;
    mxArray * T=NULL, * result=NULL, * mFs=NULL, * mnfft=NULL;
    double datax[1024];
    char buffer[1024];
    for(int j=0;j<1024;j++) // 注:如通过采集卡采集数据可将采集的数据放在 datax[]数组中,
    { // 此循环就不需要
        double samt=(double)(1.0/1024);
        datax[j]=sin(2.0*630.0*saMt*3.1415926+2.0*1017.0*saMt*3.1415926);
    }
    double * pPxx, * pFxx;
    if(! (ep=engOpen("/0"))){ // 打开 Matlab 引擎,建立与本地 Matlab 的连接
        fprintf(stderr,"/n Can't start MATLAB engine/n");
        exit(-1);
    }
    double Fs[1]={1024}; // 因为 Matlab 所有参与运算的参数都是矩阵的形式,因而下列
    double nfft[1]={1024}; // 几行将参数转变成 Matlab 可接受的矩阵形式。
    T=mxCreateDoubleMatrix(1,1024,mxREAL);
    mnfft=mxCreateDoubleMatrix(1,1,mxREAL);
    mFs=mxCreateDoubleMatrix(1,1,mxREAL);
    mxSetName(T,"T");
    mxSetName(mnfft,"mnfft");
    mxSetName(mFs,"mFs");
    memcpy((char *)mxGetPr(T),(char *)datax, 1024*sizeof(double));
    memcpy((char *)mxGetPr(mnfft),(char *)nfft, sizeof(double));
    memcpy((char *)mxGetPr(mFs),(char *)Fs,1*sizeof(double));
    engPutArray(ep,T); // 将转化的参数放入引擎中,此时可在 Matlab command 窗口下查看此参数
    engPutArray(ep,mnfft);
    engPutArray(ep,mFs);
    engEvalString(ep,"[pXX,fo]=psd(T,mnfft,mFs);"); // 利用引擎执行工具箱中文件
    engOutputBuffer(ep,buffer,512); // 如只想看显示图形,可将返回参数去掉,
    // psd 无返回参数缺省情况下会自动画图形
    result=engGetArray(ep,"pXX"); // 取出引擎中的数据放在所指的区域中供后续处理
    pPxx=mxGetPr(result);
    result=engGetArray(ep,"fo");
    pFxx=mxGetPr(result);
    engEvalString(ep,"plot(fo,10*log10(pXX));"); // 利用引擎画图
    engEvalString(ep,"title('功率谱分析');");
}
```

```

engEvalString(ep,"xlabel('Hz');");
engEvalString(ep,"ylabel('db');");
mxDestroyArray(T);          //释放内存
mxDestroyArray(mFs);
mxDestroyArray(mnfft);
mxDestroyArray(result);
engEvalString(ep,"close;");
engClose(ep);
}

```

上述程序在 VC++ 下编译需要将 libeng.dll 和 libmx.dll 两个动态库用 VC 自带的 lib.exe 转换成 ibeng.lib 和 libmx.lib。方式为:lib /def:[def 文件绝对地址及其文件名]/machine:ix86 /out:* lib。再将其复制到[Matlab 目录]\extern\lib 中,最后在 project\link\object library modules 下加载即可。

2 利用 Matlab 自身的编译器调用工具箱中的函数

2.1 工具箱的介绍

工具箱是由 Matlab 的一系列扩展函数(称为 M 文件)构成的,它可用来求解各个特定学科的问题,包括信号处理、图像处理、控制系统、神经网络、模糊逻辑、小波和通信等。

2.2 编译 M 文件原理

Matlab 的编译器可将 Matlab 的 M 文件转换为 C 或 C++ 的源代码以产生完全脱离 Matlab 运行环境的独立的运用程序,但 Matlab 本身的资料说明编译器如用来建立独立的运用程序,不能调用 Matlab 工具箱中的函数,这非常不利于搞一些特殊的算法。本人研究了一段时间发现,工具箱中的函数既然是 M 文件就一定可以用编译器来编译,以提供如 VC++ 调用,但是编译器只能编译一个独立的 M 文件,即这个 M 文件不依赖于其他的 M 文件。如果 M 文件中又调用了其他的 M 文件,可将被调用的 M 文件复制到调用 M 文件的相应位置,作适当的改动就可以用于编译器编译。编译器不支持图形函数,所以 M 文件中如有图形函数则需要注释掉。

2.3 编译的实现方法

当 Matlab 的编译器 mcc 加入适当的参数 -e(mcc -e *.*) 或 -p(mcc -p *.*) 就可生成将输入的 M 文件转换为适用于特定运用的 C 或 C++ 源代码。如果要通过 VC++ 编译,还需连入以下几个库 libmmfile.dll, libmatlb.dll, libmcc.dll, libmat.dll, libmx.dll, mibut.dll 以及 Matlab C math 库,建议采用前述的方法将动态连接改为静态连接。对于 C/C++ 编译环境的设置,在 Matlab command 窗口下运行 mex -setup 然后依提示操作,而对于 C/C++ 连接环境的设置,运行 mbuild -setup 依提示操作即可。

下面给出利用编译器将 Matlab 工具箱中 psd.m 文件生成可供 VC++ 调用的函数。

将 psd.m 文件复制一份至 Matlab\bin 目录下,改写相应调用的 M 文件如 nargchk.m, hanning.m 等。为了生成的代码简洁,对于输入参数很明了的情况下,可作大量删减,最终使 psd.m 成为一个不依赖于其他 M 文件的独立的 M 文件,注意千万要注释掉作图代码,最终改成如下形式,限于篇幅给出关键的几步:

```

function [Pxx,f]=psd(Fs,nfft,noverlap,x)
window=0.5*(1-cos(2*pi*(1:nfft)'/(nfft+1)));    //hanning 窗
dflag='none';
window=window(:)
.....

```

以上只要稍懂 Matlab 语言和信号处理知识就可完成这项工作。假设上述代码重新存为 testwin.m,在 Matlab command 窗口下设置好环境参数运行 mcc -e testwin,则可在 Matlab\bin 下生成 testwin.c,如运行 mcc -p testwin 则生成 testwin.cpp。

VC++ 下建立一个基于对话框的文件,然后在对话框里加一个 Button 控件 OnButtonPsd,将上述生成的.c 文件的头文件加入到工程的.cpp 中,且将

```
#ifdef -cplusplus extern "c"
{ #end if
```

代码声明加入 VC++ 的包含文件和生成的.c 的包含文件之间;将

```
#ifdef -cplusplus
}
#end if
```

加入.cpp 文件末尾。

为了简洁且便于处理将生成的 c 函数稍改动,给出部分代码如下:

```
void CTestpsdwinDlg::OnButtonPsd()
{
    mxArray * x-rhs;          //指向采集数据存放区
    Fs=23510;                  //数据采集的频率 nfft=1024; //1024 点的 fft
    double datax[1024];       //采集的数据
    x-rhs=mxCreatDoubleMatrix(1,1024,mxReal);
    memcpy(mxGetPr(x-rhs-),datax,1024*sizeof(double));
    noverlap=512;
    .....
    mccCopy(&Pxx,&Spec);
    mccCopy(&f,&frevvg-vector);
    for(int j=0;j<(int)(nfft/2+1);j++){
        datap[j]=mccGetRealVectorElement(&Pxx,(j+1));
        dataf[j]=mccGetRealVectorElement(&f,(j+1));
    }
    mccFreeMatrix(&Pxx);
    .....
    sendMessageBox(WM-PAINT,0,0); //利用 Vc 下的图形函数画图
    return;
}
```

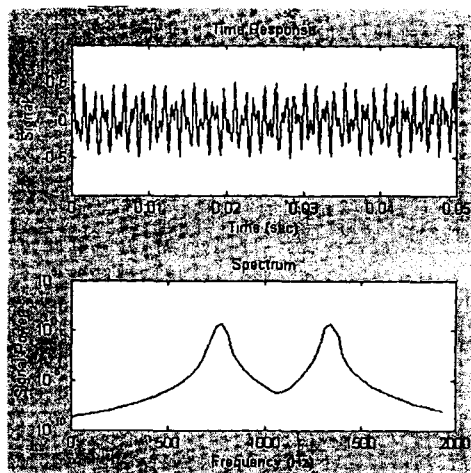


图 1 两个例子的 PSD 图

如上生成的程序可读性不太好,而生成的 c++ 代码则可读性较好,但千万注意只能用 Matlab 的 Math 库,不可用 C++ 的 Math 库,否则编译会出错,限于篇幅在此不述。

3 结论

利用 Mtlab 引擎调用工具箱中的函数可节省大量的系统资源,应用程序整体性能较好,但不可脱离 Matlab 的环境运行。用 Matlab 编译器进行工具箱函数的调用,须转换相应的 M 文件使其成为独立的 M 文件,且不支持图形函数,转换的代码可读性不太好。

以上方法在工程实践中已得到很好的运用。

参考文献:

- [1] 楼顺天,陈生潭,雷虎民. MATLAB 5.x 程序设计语言[M]. 西安:西安电子科技大学出版社,2000.
- [2] 刘庆红. Visual C++ 6.0 程序开发宝典[M]. 北京:冶金工业出版社,1999.
- [3] 董长虹,余啸海. Matlab 接口技术与应用[M]. 北京:国防工业出版社,2003.

(责任编辑:郭 捷)