

文章编号: 1671-6906(2004)02-0039-04

## 应用 Visual C++ 开发小波变换应用程序

王星海<sup>1</sup>, 许 珉<sup>1</sup>, 谢志棠<sup>2</sup>

(1. 郑州大学 电气工程学院, 河南 郑州 450002; 2. 香港理工大学 电机系, 香港)

**摘 要:** 介绍了 Visual C++ 调用 MATLAB 引擎函数的方法, 利用这种方法在应用软件开发过程中既能发挥 Visual C++ 强大的界面开发和图形绘制功能, 又充分利用了 MATLAB 处理复杂运算的能力, 并把这一技术应用于小波变换的程序设计之中, 取得了较好的效果。

**关键词:** Visual C++; MATLAB; 引擎函数; 小波工具箱  
**中图分类号:** TP311 **文献标识码:** A

小波变换是目前国际上公认的最新时-频分析工具, 由于其具有自适应性和数学显微镜性质已经成为众多学科共同关注的焦点, 近年来在许多工程领域中被广泛应用, 成为科技工作者常使用的工具之一。

目前能够进行小波变换的工具软件有很多, 其中应用最广泛的是 Math Works 公司推出的包含于 MATLAB 之中的小波工具箱 Wavelet Toolbox, 其最新版本为 2.2. 它几乎包含了小波变换的各个领域, 涉及连续小波变换、离散小波变换、小波包变换、信号消噪、密度估计、小波系数选取等方面, 适用于一维信号和二维信号的分析。然而, 由于实际应用中可能对小波变换的不同领域有不同的要求, 所以直接应用小波工具箱不能满足用户的特殊要求, 用户需要开发自己的专用软件。在开发过程中可以利用小波工具箱所提供的丰富的函数来处理信号, 以减少用户开发小波变换程序的工作量。此专用程序还可与虚拟仪器技术构成的仪器相结合, 在信号检测方面有较强的实用价值。

在程序的开发方面, Visual C++、Visual Basic、Delphi 等编程工具各有其特点。其中 Visual C++ 是 Microsoft 公司推出的一种基于 C++ 语言的集成开发工具。由于程序基于微软基础类库(MFC), 只需要编写少量的代码就可以编辑出界面非常友好的 Windows 风格的程序。此外用 Visual C++ 开发的程序还具有易于维护升级、编程灵活、代码效率高、执行速度快等一系

列优点, 因此 Visual C++ 已成为目前应用软件开发所广泛采用的主要工具之一。可见 Visual C++ 是开发小波变换程序的理想工具, 如果能将 Visual C++ 和 MATLAB 联合使用以编写程序, 必将起到事半功倍的效果。

### 1 Visual C++ 调用 MATLAB 的应用程序接口

MATLAB 的应用程序接口(API)主要由三部分组成, 它们分别是: (1)MEX 文件, 它是 MATLAB 系统在其运行环境中调用外部程序的接口; (2)MAT 文件应用程序, 它是 MATLAB 程序与外部程序进行数据输入输出交换的接口; (3)MATLAB 计算引擎函数, 它是外部程序调用 MATLAB 数学运算工具函数进行有关数学运算的接口。

在这三种接口中, MEX 文件是作为一种动态链接函数被调用和执行的。它不能脱离 MATLAB 的工作环境而执行, 必须在 MATLAB 的工作环境中通过 MATLAB 调用才能运行, 它与 MATLAB 的内建函数的调用方式完全相同。

MAT 文件应用程序是一种可脱离 MATLAB 工作环境而独立运行的应用程序, 但它所具有的功能非常有限, 只能用于程序间的数据交换, 而不能利用 MAT-

LAB 所提供的丰富的数学运算功能函数来完成计算等任务。

MATLAB 计算引擎函数也是一种可脱离 MATLAB 工作环境而独立运行的应用程序, 它允许用户在用其它软件编写的应用程序中对 MATLAB 的数学运算工具函数进行调用. 在应用程序运行时, MATLAB 作为计算引擎在后台运行, 完成复杂的数学计算, 应用程序可以与 MATLAB 引擎函数之间进行数据交换.

可见通过 MATLAB 引擎函数, 才能最大限度地充分发挥 MATLAB 的功能. 沿着这一思路, 利用 MATLAB 引擎函数, 在 Visual C++ 中调用 MATLAB 小波工具箱中的函数, 并将分析后得到的信号数据传递给 Visual C++, 利用 Visual C++ 强大而灵活的绘图工具将信号绘制在屏幕上, 从而开发出有工程实用性的小波变换程序.

## 2 Visual C++ 中 MATLAB 引擎程序的建立和调试

第一步, 为了使 Visual C++ 的编译、调试环境能够对 MATLAB 引擎函数应用程序进行编译和调试, 需要对 MATLAB 的编译器进行设置. 首先在 MATLAB 命令窗口中运行 mex-setup, 然后根据屏幕提示选择 Visual C++ 的编译器. 然后在 MATLAB 命令窗口中运行 mbuild-setup, 设置方法与前面基本相同. 当 Visual C++ 的版本或路径发生变化, 可随时根据需要再次进行配置.

第二步, 设置 Visual C++ 环境. 在 Visual C++ 开发窗口下, 打开工具 (Tools) 菜单, 点击选项 (Options), 在弹出的对话框中选择目录 (Directories) 标签, 在显示目录下拉列表中选中 “Include files” 选项, 把目录 %MATLAB%\extern\include 加入到路径中, 然后在显示目录下拉列表中选中 “Library files” 选项, 把目录 %MATLAB%\extern\lib 加入到路径中, 其中 %MATLAB% 代表本地机器上 MATLAB 所安装的目录.

第三步, 因为 MATLAB 引擎函数都在文件 engine.h 中予以说明, 所以在所创建的工程文件源文件中应包含 engine.h. 如果在程序中涉及到对 MAT 文件的操作, 还应对文件 mat.h 进行包含. 第四步, 链接 MATLAB 所必需的库文件. 为了能够调用 MATLAB 引擎函数, 所编程序必须包含几个 MATLAB 库文件. 为了编程方便, 建议采用静态链接的方法. 打开工程 (Project) 菜单, 点击设置 (Settings), 在弹出的对话框中选择连接 (Link) 标签, 在 Object/library modules 栏中的库文件队列尾部 (最右端) 再添加 libeng.lib, libmx.lib 这 2 个

文件. 如果在程序中涉及到对 MAT 文件的操作, 还应添加 libmat.lib 库文件.

完成以上四步, 即可以在 Visual C++ 集成环境中开发 MATLAB 引擎应用程序, 经过编译、连接, 最终生成可执行文件.

## 3 小波变换程序的设计

该程序基于微软基础类库 (MFC) 的对话框风格. 对话框的左侧为图形输出, 右侧分为数据分析、图形显示、坐标输出、数据清除几部分. 图 1 显示出该程序的主界面.

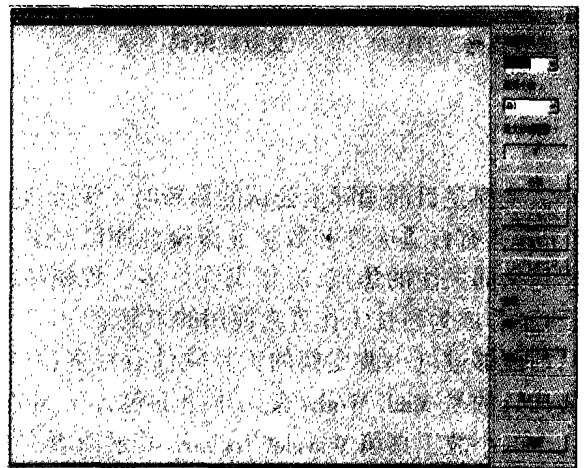


图 1 小波变换程序的主界面

通过组合框分别选择进行此次分析的信号和小波. 按下分析按钮, 程序将根据载入信号的长度和小波类型计算出所能分解的最大尺度, 并将信号以最大尺度分解, 重构得到各尺度下的低频信号和高频信号. 根据需要可以分别显示原始信号、各尺度下高频信号和低频信号. 同时可以在图形上任意处单击鼠标左键, 程序会在该处绘制一红色光标, 坐标输出区会显示出该点坐标值, 如图 2 所示. 分析完后可以通过重新设置按钮清除参数, 进行下一次的信号分析工作.

下面给出分解、重构信号这个函数的部分程序清单, 具体说明如何在 Visual C++ 中利用 MATLAB 引擎函数调用小波工具箱中的函数.

```
// m_SignalName 为所选信号
// m_WaveletName 为所选小波
// m_MaxLevel 为最大分解尺度
// *signal, *signalmax, *signalmin 分别存储原始信号
// 及其最大值、最小值
// *app[ 10]、*appmax[ 10]、*appmin[ 10] 指针数组分
```

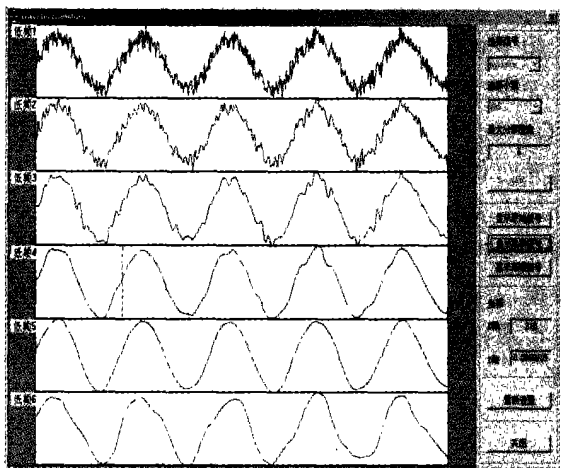


图2 各尺度下低频信号和光标处坐标值

别存储//各尺度下重构的低频信号及其最大值、最小值

// \*det[ 10]、\*detmax[ 10]、\*detmin[ 10] 指针数组分别存储//各尺度下重构的低频信号及其最大值、最小值

//以上变量皆为全局变量

void CWaveletTransformDlg::OnAnalyze()

{

...

//定义变量

double \*level1, level2[ 1], \*length;

mxArray \*Level1, \*Level2, \*N, \*Length;

mxArray \*Signal, \*SignalMax, \*SignalMin;

mxArray \*AppSignal, \*AppSignalMax, \*AppSignalMin;

mxArray \*DetSignal, \*DetSignalMin, \*DetSignalMax;

Engine \*ep; //定义 ep 为 MATLAB 引擎的指针

//开启本地 MATLAB 引擎,成功则执行以下指令

if(ep = engOpen(NULL))

{ //载入指定的信号

engEvalString(ep, "load"+ m \_SignalName);

engEvalString(ep, "S="+ m \_SignalName);

//计算信号的长度

engEvalString(ep, "Length=length(S);");

Length=engGetVariable(ep, "Length");

length=mxGetPr(Length);

m \_SignalLength=length[ 0];

//从引擎中取出原始信号

Signal=engGetVariable(ep, "S");

engEvalString(ep, "SignalMax=max(S);");

SignalMax=engGetVariable(ep, "SignalMax");

engEvalString(ep, "SignalMin=min(S)");

SignalMax=engGetVariable(ep, "SignalMax");

signal=mxGetPr(Signal);

signalmax=mxGetPr(SignalMax);

signalmin=mxGetPr(SignalMin);

//在最大尺度下将信号进行小波变换

//即调用多尺度一维分解函数(wavedec)

engEvalString(ep, "[ C, L]=wavedec(S, MaxLevel, ' "+ m \_WaveletName+" ');");

//从引擎中分别取出各尺度下重构的低频信号和低频信号

for(int i=1; i<=m \_MaxLevel; i++)

{ //进行第 i 尺度的重构

double buffer[ 1];

buffer[ 0]=i;

N=mxCreateDoubleMatrix(1, 1, mxREAL);

memcpy((void \*)mxGetPr(N), (void \*)buffer, sizeof(buffer));

engPutVariable(ep, "N", N);

//从引擎中取出第 i 尺度下的低频信号

//即调用重构函数(wrcoef)

engEvalString(ep, "AppSignal=wrcoef(' a', C, L, ' "+ m \_WaveletName+" ', N);");

AppSignal=engGetVariable(ep, "AppSignal");

engEvalString(ep, "AppSignalMax=max(AppSignal);");

AppSignalMax=engGetVariable(ep, "AppSignalMax");

engEvalString(ep, "AppSignalMin=min(AppSignal);");

AppSignalMin=engGetVariable(ep, "AppSignalMin");

app[ i]=mxGetPr(AppSignal);

appmax[ i]=mxGetPr(AppSignalMax);

appmin[ i]=mxGetPr(AppSignalMin);

//从引擎中取出第 i 尺度下的高频信号

//即调用重构函数(wrcoef)

engEvalString(ep, "DetSignal=wrcoef(' d', C, L, ' "+ m \_WaveletName+" ', N);");

DetSignal=engGetVariable(ep, "DetSignal");

engEvalString(ep, "DetSignalMax=max(DetSignal);");

DetSignalMax=engGetVariable(ep, "DetSignalMax");

engEvalString(ep, "DetSignalMin=min(DetSignal);");

DetSignalMin=engGetVariable(ep, "DetSignalMin");

det[ i]=mxGetPr(DetSignal);

detmax[ i]=mxGetPr(DetSignalMax);

detmin[ i]=mxGetPr(DetSignalMin); }

```
engClose(ep); //关闭计算引擎
...
}
```

## 4 结 语

利用 MATLAB 引擎函数可以简单有效地实现在 Visual C++ 中调用 MATLAB 小波工具箱中功能强大的函数. 虽然在 Windows 环境下其内部实现使用的是较为复杂的基于组件对象模型(Component Object Model)的 ActiveX 技术, 但用户可以不考虑这些技术细节,

直接调用 MATLAB 引擎提供的函数, 充分发挥 Visual C++ 和 MATLAB 二者各自的优势, 编写出功能强大的应用程序. 利用本文所述的方法, 作者开发了基于 Visual C++ 6.0 和 MATLAB 6.5 的小波变换程序. 其中程序的界面和信号的绘制依靠 Visual C++ 编制, 所有的计算均为 MATLAB 小波工具箱中的函数承担, 从而大大提高编程效率. 这种方法为用户自己开发所需的小波变换程序提供了一种新思路. 该程序也可用于虚拟仪器技术构成的仪器中, 在工程应用中有一定的实用价值.

## 参考文献:

- [1] 杨福生. 小波变换的工程分析与应用[M]. 北京: 科学出版社, 1999.
- [2] 飞思科技产品研发中心. MATLAB 6.5 辅助小波分析与应用[M]. 北京: 电子工业出版社, 2003.
- [3] 王超龙. Visual C++ 6.0 入门与提高[M]. 北京: 人民邮电出版社, 2002.
- [4] 张志涌. 精通 MATLAB 6.5[M]. 北京: 北京航空航天大学出版社, 2003.

# Development of Wavelet Transform Based on Visual C++

WANG Xing-hai<sup>1</sup>, XU Min<sup>1</sup>, C.T. Tse<sup>2</sup>

(1. School of Electrical Engineering, Zhengzhou University, Zhengzhou 450002;

2. Department of Electric Engineering, The Hong Kong Polytechnic University, China)

**Abstract:** A method of calling the engine function of MATLAB from Visual C++ is introduced in this paper. In the development of application software products, this method makes it possible to fully explore the potential of Visual C++ in generating user graphic interfaces and drawing graphics and that of MATLAB in performing numerical computations. The technique has also been used in the design of wavelet transform with a satisfactory effect obtained.

**Key words:** Visual C++; MATLAB; engine function; wavelet toolbox

(上接第 38 页)

# Analysis of the Structure of BOM with Directed Graph

LIU Zong-dou<sup>1, 2</sup>, DENG Si-er<sup>1</sup>

(1. Henan University of Science and Technology, Luoyang 471039;

2. Xuchang Tobacco Machinery Co., Ltd., Xuchang 461000, China)

**Abstract:** This paper describes the level and the tree shape structure of BOM and simply analyzes the structure model of BOM in a digraph law, by analysing and explaining the relevant traditional concept of product structure with the basic directed graph. We have put forward a new idea to use the critical path method in the production plan of enterprise.

**Key words:** BOM; directed graph; critical path method