

LabWindows/CVI 与 Matlab 接口实现的 几种方法

张 燕 陈 华

(新疆大学电气工程学院, 乌鲁木齐 830008)

摘 要: 根据现代测控领域研究的需求, 结合LabWindows/CVI和Matlab两种软件的优势, 介绍了几种实现LabWindows/CVI与Matlab软接口的方法, 并加以比较。

关键词: LabWindows/CVI; Matlab; 混合编程

Several Methods of Implementing Interfaces between Labwindows/CVI and Matlab

ZHANG Yan, CHEN Hua

(College of Electric Engineering, Xinjiang University, Urumqi 830008, China)

Abstract: According to research demand of modern measurement and control area and with the advantages of LabWindows/CVI and Matlab software, several methods of implementing interfaces between LabWindows/CVI and Matlab software are introduced and compared.

Keywords: LabWindows/CVI; Matlab; hybrid programming

0 引言

美国NI公司的LabWindows/CVI软件和Mathwork公司的Matlab软件各有侧重和优势。前者是基于ANSI C规范的虚拟仪器软件平台, 采用事件驱动与回调函数的面向对象的编程方式, 凭借较丰富的库函数以实现复杂的数据采集和处理。而后者拥有大量的函数、科学计算功能和友好易用的开放式可扩展环境, 使之成为计算机辅助设计与分析、算法研究与开发的基本工具和首选平台。

近年来, 信号分析与处理技术在多方面取得了飞跃发展, 如神经网络、模糊理论、小波变换、混沌技术等新技术正在逐步展现它们在测控领域应用的潜力。这些新技术均可通过调用Matlab中的功能函数来实现, 从而使得编程环境既有强大的数值计算能力又有高的执行效率。因此, 两者有机结合、优势互补, 成为新的方案。

实现LabWindows/CVI和Matlab结合的关键是其数据和命令的交换, 即软件接口。LabWindows/CVI和Matlab都有提供与外部程序接口的组件。因此, 技术的核心是采用正确的方法建立起它们的联系。目前将LabWindows/CVI与Matlab有机结合的方法有3种, 可根

据自身掌握语言情况和习惯, 采用不同的方法来实现, 从而使工程应用更加方便、快捷。

1 LabWindows/CVI对Matlab的引擎调用技术(方法1)

本技术的实质是将Matlab当作计算引擎, 通过LabWindows/CVI编写引擎程序向Matlab传送数据并调用, 使之在后台完成运算并返回结果。环境的建立须有3个静态链接库文件, 分别为Libeng.lib、Libmx.lib、Libmat.lib, 引擎程序中需要的engine和mx-文件均包括在这些静态库中。此外, engine.h头文件中含有对引擎函数及相关数据类型定义并包含matrix.h, 如果缺少, 将无法使用引擎^[1]。

程序的关键代码如下:

```
#include "engine.h"
.....
int main()
{
.....
Engine*ep;
mxArray *A = NULL, *B = NULL;
.....
.....
if (! (ep = engOpen (NULL) ) )
.....
```

作者简介 张燕(1983-) 硕士 研究方向为智能控制与系统开发。

收稿日期 2010-09-12

```
return EXIT_FAILURE;}
A=mxCreateDoubleMatrix(a,b,mxREAL);
mxSetName(A,"A");
.....
engPutArray(ep,A);
.....
B=engGetArray(ep,"Y");
engClose(ep);
.....
```

引擎调用技术简单实用,充分发挥了2种软件的优点,可成为开发智能虚拟仪器的有效途径,具有广泛的应用价值。但在混合编程过程中,由于在LabWindows/CVI运行的同时,又启动了Matlab进行后台数据处理,需要有较高的计算机硬件配置。

2 用Matlab编译器实现软接口(方法2)

编译器(Matlab Compiler)是随着Matlab自动安装的,其版本向下兼容。它可以通过设置mcc命令的选项,将m文件转化成C/C++或p等各种类型的源代码,根据需要生成可执行文件*.lib文件,*.dll文件等。最后,由其C/C++代码完成调用。将m文件编译成C/C++源代码或mex文件的工作由编译器命令自动完成,但是初始环境的配置却是很重要的。安装环境配置完毕,即可利用mcc或mex命令并根据需要选择相关命令参数,实现对目标m文件的编译,其编译文件类型是由命令和参数决定的^[2]。

2.1 环境建立

整个配置过程分为如下步骤:

(1) 安装Matlab编译器,随Matlab自动安装。

(2) 安装ANSI C编译器或者C++ 编译器,通常选择VC++ 或BC++。并利用mbuild - setup 和mex-setup 完成编译器定位。

(3) 验证mbuild file. c命令是否能正确将示例中的C/C++ 程序编译生成可执行程序。

(4) 验证mcc-m file.m命令是否能正确编译成C代码和mex 包裹文件,并最终生成独立可执行程序。

2.2 使用Matlab 编译器

安装环境配置完毕,就可以利用mcc或mex命令并根据需要选择相关命令参数对目标m文件实现编译了,其编译文件类型是由命令和参数决定的。使用mcc - m*.m命令后会生成4个文件:*.c(源文件),*.h(头文件)、*main.c(包裹文件,用来调用库函数及初始化)和*.exe(可执行档)。只要采取一定的方法,上述4个文件均可脱离matlab环境被使用。用户甚至可以不必关心这些文件的内容是什么。此外使用mcc - *.m还可以生成动态链接库

文件,包括:*.c (mex文件的源代码)、*.h(头文件,数据结构说明)、*mex.c(MEX包裹文件,包含了MEX函数接口)、*.dll(最终生成的dll(mex)文件)。对于其它代码生成控制参数,可参见mcc 命令帮助。在LabWindows/CVI 环境下调用已编译的C代码或动态链接库(dll)则需要包含相关的头文件和库文件。

3 利用ActiveX 服务控件实现(方法3)

此方法和引擎调用技术的实现方法类似,同样是通过创建功能函数,并利用这些数进行混合编程,实现与Matlab的数据交换和命令传输。但不同的是,这些功能函数是由LabWindows/CVI中Active X服务控件提供的。利用ActiveX服务控件实现在LabWindows/CVI中Matlab的调用,得到ActiveX服务函数通常有3种方法,可以根据具体情况选择使用。

3.1 Matlab 驱动器函数库的生成(方法3.1)

LabWindows/ CVI开发环境提供了使用ActiveX控件的标准函数库和开发工具。在标准函数库中, ActiveX Automation函数库提供了使用ActiveX控件的各种功能函数,可以通过这些函数来调用ActiveX服务器。

首先应该建立一个Matlab 驱动器函数库。在LabWindows/CVI开发环境下选择Tools Create ActiveX Automation Controller,然后在ActiveX服务控件选择框中选择Matlab Automation Server Type Library,将会生成函数对话框,并将其保存为matlabsrvr. fp。同时在选择的目录中还会生成matlabsrvr. c、matlabsrvr. h、matlabsrvr. obj、matlabsrvr. sub等4个文件。将matlabsrvr. fp、matlabsrvr. c、matlabsrvr. h 文件加入工程后,就可以调用生成的函数库,从而实现LabWindows/ CVI 与MATLAB 的混合编程^[4]。

生成了MATLAB 驱动函数库以后, 就可以在LabWindows/CVI开发环境下调用一般函数。为方便开发者实现LabWindows/CVI环境下与MATLAB的混合编程,NI 公司还专门配置了一个更加友好的接口函数文件,该文件在LabWindows/CVI的Sample中给出,名称为matlabutil. c。通过这个接口文件,以上的函数可以转型为其它更容易操作的函数。例如SendMatrix将矩阵送入Matlab工作空间,GetMatrix从Matlab工作空间获得矩阵,RunMatlabCommand运行Matlab命令,RunMatlabScript运行Matlab的M函数。

3.2 修改Matlab 的注册码(方法3.2)

在5.0 版本以上的LabWindows/ CVI 中Samples\ActiveX\Matlab直接提供有LabWindows/CVI和Matlab的ActiveX服务函数。但得到的ActiveX服务函数

由于用户的Matlab版本不同而不能直接应用,需要修改Matlab的注册码^[5]。

找到当前所用Matlab版本的注册码,按照ActiveX服务函数要求的格式改写原来函数自带的注册码。生成的ActiveX服务函数matlabssvr.c包含了LabWindows/CVI环境下实现同Matlab混合编程所用的基本功能函数。但对这些基本函数调用并不方便,为此,必须对这些函数进行封装。在进行ActiveX服务函数封装时需要注意:由于Matlab不支持BSTRs,在发送和接受字符串的过程中需要进行字符串和双精度浮点数的相互转换,通过Fmt函数实现。在发送和接受矩阵的过程中需要进行安全阵列和C语言数据型的相互转换,通过CA_Array2DToSafeArray和CA_Array2DToSafeArray和CA_SafeArrayTo2DArray实现。这些函数都是LabWindows/CVI的内建函数,可以通过函数面板得到调用指南。接着需要创建DLL文件。建立一个matlabutil.prj,在其中加入matlabssvr.fp,matlabssvr.h,matlabssvr.sub,matlabutil.h和matlabutil.c,在CVI工程窗口中创建DLL。在文件夹中会出现matlabutil_dbg.cdb,matlabutil_dbg.dll,matlabutil_dbg.lib。

这种混合编程方法得到的DLL文件可移植性强,在需要进行LabWindows/CVI和Matlab混合编程时只要在工程文件中添加matlabutil.h,matlabssvr.h,matlabutil_dbg.lib即可。

3.3 重新获得ActiveX服务函数(方法3.3)

此方法得到ActiveX服务函数是建立在3.1方法的基础之上。利用该方法得到ActiveX服务函数的具体步骤为:

(1) 打开安装目录...\National Instruments\CVI\samples\activex\matlab\下的NI例程。

(2) 从工程中删除“matlabssvr.h”、“matlabssvr.fp”、“matlabssvr.c”3个文件。

(3) 打开面板文件“matlabdemo.uir”,点击菜单“tools>CreateActiveXController...”,按照3.1方法重新生成Matlab的驱动文件Matlabssvr.fp。

(4) 把重新生成的“matlabssvr.c”和“matlabssvr.h”文件重新加入到工程中。

(5) 此时在文件“matlabdemo.c”中,语句“stat=MLApp_NewD IMLApp(NULL,&hMatlab);”被修改成“stat=MLApp_NewD IMLApp(NULL,1,LOCALE_NEUTRAL,0,&hMatlab);”,此时在LabWindows/CVI中调用Matlab的相应命令也应该改为“stat=MLApp_NewD IMLApp(NULL,1,LOCALE_NEUTRAL,0,&hMatlab);”,然后在“matlabssvr.c”中将所有的MLApp_IID_DIMLApp都用IID_IDispatch

代替(在Edit replace中选replace all即可,其中case sensitive,regular expression和wrap这3项需要打勾)。

(6) 在matlabssvr.c中,将const IID IID_IDispatch={.....},const IID MLApp_IID_ IMLApp={.....},const IID MLApp_IID_DIMLEval={.....}这几行删除,然后在此处加上变量定义:const IID MLApp_IID_D IMLEval, const IID MLApp_IID_ IMLApp。

按照上述步骤,就得到了新的ActiveX服务函数,接着按照3.1的方法,把目录下的matlabutil.c,matlabutil.h,matlabssvr.c,matlabssvr.h,matlabssvr.fp这5个文件添加到工程文件中,并在程序的开始部分做相应的头文件说明^[6]。

4 结论

通过对以上2种软接口技术的比较可以知道,在易用性方面,LabWindows/CVI对Matlab的引擎调用技术(方法1)最佳,此方法要求计算机内存足够且配置高。在数据处理的灵活性方面,利用ActiveX服务控件来实现(方法3)最佳。此方法又分为3种解决方案。Matlab驱动器函数库的生成(方法3.1),得到的ActiveX服务函数仍为底层函数,调用不方便,而且要求用户机先装Matlab,后装LabWindows/CVI;修改Matlab的注册码(方法3.2),得到的ActiveX服务函数却由于用户机的Matlab版本各不相同而不能直接应用,需要在注册表中修改Matlab的注册码,这对注册表不熟悉的用户来说也很不方便,重新获得ActiveX服务函数(方法3.3),此方法在方法3.1的基础上作以修改,简单易操作,但成熟度不够,还有待于改进。

通过对LabWindows/CVI和Matlab这2种软件的研究,充分发挥两者的优点,提高了编程的效率和水平,促进了软件在新技术领域的应用。

参考文献

- [1] 郑力新,周凯汀,龚勇,等. LabWindows/CVI对Matlab的引擎调用技术[J]. 华侨大学学报(自然科学版), 2007, 28(4)
- [2] 郑力新,周凯汀,龚勇. LabWindows/CVI与Matlab软接口技术及应用[J]. 自动化仪表, 2004, 25(4): 14-16
- [3] 张福波,张志伟,刘光军. LabWindows/CVI环境下调用MATLAB函数的实现方法[J]. 机床与液压, 2004, (8): 100-101
- [4] 刘君华. 基于LabWindows/CVI的虚拟仪器设计[M]. 北京: 电子工业出版社, 2005
- [5] 陈希有. 一种LabWindows/CVI与Matlab的混合编程的实现方法[J]. 微计算机信息, 2005, 21(1)
- [6] 刘洪昌,李洪儒,郭利,等. 基于LabWindows/CVI和Matlab平台的小波消噪仪的设计与实现[J]. 仪表技术, 2009, (1)