doi:10.16055/j.issn.1672-058X.2015.0010.017

电力电子技术仿真实验平台的开发*

王君洪, 陈跃东**
(安徽工程大学 安徽省电气传动与控制重点实验室,安徽 芜湖 241000)

摘　要:针对《电力电子技术》实践教学中教学资源有限,平均教学资源严重不足的问题,为了提高教学质量,适应多种教学方式并存的需要,利用 Visual C++和 MATLAB 开发了电力电子仿真实验平台;仿真实验平台能够在不安装其它播放软件的情况下方便的使用,并通过反复调试电路的各参数,观察输出波形的变化,以提高学生使用知识的能力和创新能力。

关键词:电力电子技术;Matlab;VC++;仿真实验平台
中图分类号:G642.23　　**文献标志码**:A　　**文章编号**:1672-058X(2015)10-0082-05

电力电子技术是一门新兴而且后期发展非常迅速的跨学科的科学技术,它出现在 20 世纪中后期,它不仅仅是各种创新性的高新技术产业发展十分重要的基础,而且还是传统产业的深刻变化的关键和手段[1]。电力电子技术十分重视各种电路的工作过程是它最为明显的特点,电路的分析显得非常重要,然而由于教学条件和实验设备的匮乏,课程的实践教学并不理想,针对以上问题,在 MATLAB 和 VC++的混合编程下,设计和开发电力电子仿真实验平台。Matlab 是一款应用于工程计算的数学软件,Matlab 中的电力电子模块库以 Simulink 为运行环境,包含电路、电力电子、电力系统等电学中常用的基本元件和系统仿真模型,可以实现电力电子的仿真设计和复杂的控制[2]。Microsoft Visual Studio 2008 是面向 Windows 7 的开发工具,使用 Visual Studio 2008 可以高效开发 Windows 应用程序,它还支持项目模板、调试器和部署程序。因此,选用 Matlab 做电力电子电路仿真,用 VS2008 下的 MFC 开发平台的用户交互界面。

1　电力电子实验平台的开发

试验平台基于 Matlab 的 Simulink 和 Power System 工具箱以及 VS2008 的 MFC 开发,平台拥有简单明了的用户界面[3],具有交互功能,用户友好的模拟输出显示,操作方便等特点。

1.1　Visual C++与 Simulink 接口

Visual C++ 2008 应用程序经过 Matlab 引擎执行 Simulink 仿真指令行对 Simulink 进行控制。事实上,利用 Matlab 引擎创建起的对话,是将 Matlab 以 ActiveX 控件方式开动的。

MATLAB 引擎,指的是 MATLAB 支持 C/C++语言的接口函数,用户利用这些接口函数可在其他的编程环境中完成对 Matlab 的控制[5]。通过 Matlab 引擎库函数提供的交流和应用指令之间的数据传输的任务与 Matlab 引擎完成。MATLAB 引擎函数库共包含许多 C 语言的引擎功能,头文件 engine.h 中均有详细的说明,因此在使用时,就要包含该头文件。Matlab 引擎函数库里的主要引擎函数如表 1 所示。

收稿日期:2015-01-18;**修回日期**:2015-03-12.
* **基金项目**:安徽高校省级自然科学研究项目(KJ2013A041).
作者简介:王君洪(1991-),男,安徽宿州人,硕士研究生,从事电气传动与控制研究.
** **通讯作者**:陈跃东(1956-),男,湖北宜昌人,教授,硕士生导师,从事电气传动与控制研究.E-mail: ydchen@ ahpu.edu.cn.

表 1　引擎函数

引擎函数	功能描述
engOpen	启动 Matlab 计算引擎
engGetArray	从 Matlab 引擎获得一个 Matlab 矩阵,用于数据交换
engPutArray	从应用程序向 Matlab 引擎发送一个 Matlab 矩阵,用于数据交换
engEvalString	执行一个 Matlab 命令
engOutputBuffer	创建字符缓冲区,以获得 Matlab 文本输出
engOpenSingleUse	打开一个单独的非共享的 Matlab 计算引擎
engClose	关闭 Matlab 引擎
engSetVisible	设置 Matlab 引擎
engGetVisible	返回 Matlab 显示状体
engGetVariable	从 Matlab 工作区(Workspace)获得一个变量
engPutVariable	将指定名称的 Matlab 变量存入 Matlab 工作区

使用到的主要有:EngOpen-打开 Matlab engine,EngClose-关闭 Matlab 引擎,engEvalString-传递指令让 Matlab 执行,Engine * ep;定义引擎指针,int engSetVisible(Engine * ep, bool value)——显示和隐藏 Matlab 主窗口。

1.2　开发环境的设置

只有包含 MATLAB 的引擎头文件 engine.h 而且导入 Matlab 对应的库文件 libmx.lib、libmat.lib、libeng.lib,才能在 VC 中成功编译 Matlab 引擎程序。

(1) 在菜单上的工具/选项,找到并在 VC++目录此选项打开的项目和解决方案,在“显示下列目录:”下面的下拉列表框中,选择“文件”下面的下拉列表框里选择“包含文件”,在下面增添路径:“D:\Program Files\MATLAB\extern\include”(计算机中 MATLAB 安装目录下的 extern 里的 include 文件夹)。

(2) 在显示以下内容的目录的列表框里选择“库文件”,然后添加以下路径:“D:\Program Files\MATLAB\extern\lib\win32\microsoft”。若计算机操作系统和 MATLAB 安装的都是 x64 位,要在该页面里的“平台”下面选择 x64,然后添加一个路径中的“库文件”:“D:\Program Files\MATLAB\extern\lib\win64\microsoft”(计算机里 MATLAB 安装的路径下找上述这些文件夹),若安装 32 位操作系统的计算机只要把 x64 改成 x32 即可。

(3) 通过菜单项目/属性,在“配置”里选择“活动(Debug)”,在平台里设置自己程序运行的环境是 win32 还是 x64 平台,然后在“属性”菜单中的“链接”的“输入”选项下的“附加依赖项”添加 libmx.lib libmat.lib libeng.lib。

以上步骤(1)、(2)设置一次就即可,而步骤(3)则需要对每个工程都设定一次。

1.3　平台主程序的设计

程序使用的是对话框的形式,在主对话框里选择要进行的仿真,进入 Simulink 仿真界面进行仿真运行。若设置一个对话框,将仿真的结果输出到对话框里显示,并且通过对话框将元器件参数传递到 Simulink 仿真模型当中,这样将会大大增长响应时间,使效率降低,故实验平台没有采用方案去做,而是直接在 MATLAB 下进行 Simulink 仿真,这样更加节省资源、时间,效率更高,也有助于学生了解、研究、学习使用 MATLAB,Simulink[3]。

(1) 双击 LISTBOX 控件,直接进入“List Box”定义的变量函数所在位置,按这样格式添加代码 lbox.AddString(_T("xxxxx"));其中的 xxxxx 为仿真电路的名称,将以下代码加入到其中,代码如图 1 所示。

右键点击 List Box,再点击属性,然后把 Visible 设置为 Ture,这样列表里的选项不会产生乱序,而是按所排序的顺序显示。

(2) 双击“进入实验”按钮,开始对该按钮功能进行编程,使其实现点击该按钮就可以进入在列表里所选的实验。实现代码如图 2 所示。

在代码中的 engEvalString(ep,"uiopen('xxx',1)");的 xxx 处填写所做的电路仿真.mdl 文件的路径(路径中最好不要出现中文),值得注意的是,仿真电路要与前面所提到的“List Box”中所写的电路相对应,这样,选中列列表里的电路,再点击“进入实验”才可以进入所对应的电路仿真,运行仿真[4]。

(3) 双击“新建模型”按钮,对该按钮进行编程,使其可以实现点击就可以进入 Simulink 新建 Model,制作一个新的仿真的功能。代码实现如图 3 所示。

CtestxxyDlg

```
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。当应用程序主窗口不是对话框时，框架将自动
    //  执行此操作
    SetIcon(m_hIcon, TRUE);         // 设置大图标
    SetIcon(m_hIcon, FALSE);        // 设置小图标

    // TODO: 在此添加额外的初始化代码

    lbox.AddString(_T("单相桥式全控整流电路电阻负载"));
    lbox.AddString(_T("三相桥式全控整流电路电阻负载"));
    lbox.AddString(_T("直流降压斩波电路"));
    lbox.AddString(_T("直流升压斩波电路"));
    lbox.AddString(_T("电压型单相全桥逆变电路"));
    lbox.AddString(_T("电流型三相逆变电路"));
    lbox.AddString(_T("单相交流调压电路电阻性负载"));
    lbox.AddString(_T("单相交流调压电路电阻电感负载"));

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}
```

图1 具体的代码实现

```
void CtestxxyDlg::OnBnClickedButton11()
{
    // TODO: 在此添加控件通知处理程序代码
    Engine *ep; //定义Matlab引擎指针。
    if (!(ep=engOpen(NULL))) //测试是否启动Matlab引擎成功。
    {
        MessageBox("未能启动MATLAB引擎!" );
        exit(1);
    }
    int i=lbox.GetCurSel();
    /*CString str;
    lbox.GetText(i,str);
    MessageBox(str);*/
    switch(i)
    {
    case 0:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\single_phase_bridge_r.mdl',1)");

    break;
    case 1:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\three_phase_bridge_lr1.mdl',1)");
    break;
    case 2:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\Buck.mdl',1)");
    break;
    case 3:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\Boost.mdl',1)");
    break;
    case 4:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\fb_inverter1.mdl',1)");
    break;
    case 5:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\three_phase_csi.mdl',1)");
    break;
    case 6:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\acregulatorr.mdl',1)");
    break;
    case 7:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\singlephaseacregulatorrl.mdl',1)");
    break;
    case 8:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\single_phase_hw_lr.mdl',1)");
    break;
    case 9:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\single_phase_bridge_er.mdl',1)");
    break;
    case 10:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\three_phase_hw_lr.mdl',1)");
    break;
    case 11:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\three_phase_fb_inverter.mdl',1)");
    break;
    case 12:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\buck_boost.mdl',1)");
    break;
    case 13:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\three_phase_ac_regulator.mdl',1)");
    break;
    case 14:
    engEvalString(ep," uiopen('D:\\Program Files (x86)\\Microsoft Visual Studio\\MyProjects\\testxxy\\single_phase_cycloconverter.mdl',1)");
    break;
    }
```

图2 列表实现代码

(4) 双击“进入MATLAB”按钮,对该按钮进行编程,使其可以实现点击进入MATLAB程序的功能。代码实现如下:

```
void CtestxxyDlg::OnBnClickedButton4()
{
WinExec("D:\Program Files\MATLAB\bin\matlab.exe",SW_SHOW);
}
```

```
void CtestxxyDlg::OnBnClickedButton2()
{
    Engine *ep; //定义Matlab引擎指针。
    if (!(ep=engOpen(NULL))) //测试是否启动Matlab引擎成功。
    {
        MessageBox("未能启动MATLAB引擎!" );
        exit(1);
    }
    engEvalString(ep,"simulink");
    //WinExec("D:\\Program Files\\MATLAB\\bin\\matlab.exe",
    // TODO: 在此添加控件通知处理程序代码
}
```

图3 功能代码

该段代码使用了 WinExec 指令,代码当中加下划线的为 MATLAB 程序的安装目录下的 matlab.exe 程序。

在仿真结束后,可以点击此按钮打开 MATLAB 查看仿真数据。

平台的使用很简单,在“实验选择”下方的选择框里选中要进行的仿真实验,然后点击“进入实验”按钮,即可打开该电路的 Simulink 仿真实验,运行仿真,查看输出波形。

若要新建一个仿真,只要点击“新建模型”按钮即可打开 Simulink 窗口,新建一个 Model,然后即可向里面拖元器件做仿真了。

打开做好的仿真之前,平台会先运行 MATLAB 引擎,打开 MATLAB 指令窗口,然后在 MATLAB 指令窗口里输入指令,让 MATLAB 执行。若有必要打开 MATLAB 主程序,点击“进入 MATLAB”按钮进入便可,方便快捷。

2　仿真实例

仿真实验平台可对多种电力电子典型电路进行仿真,以电压型单相全桥逆变电路为实例进行说明[6]。在主程序界面列表里选中电压型单相全桥逆变电路,点击“进入实验”,打开实验模型,如图 4 所示。

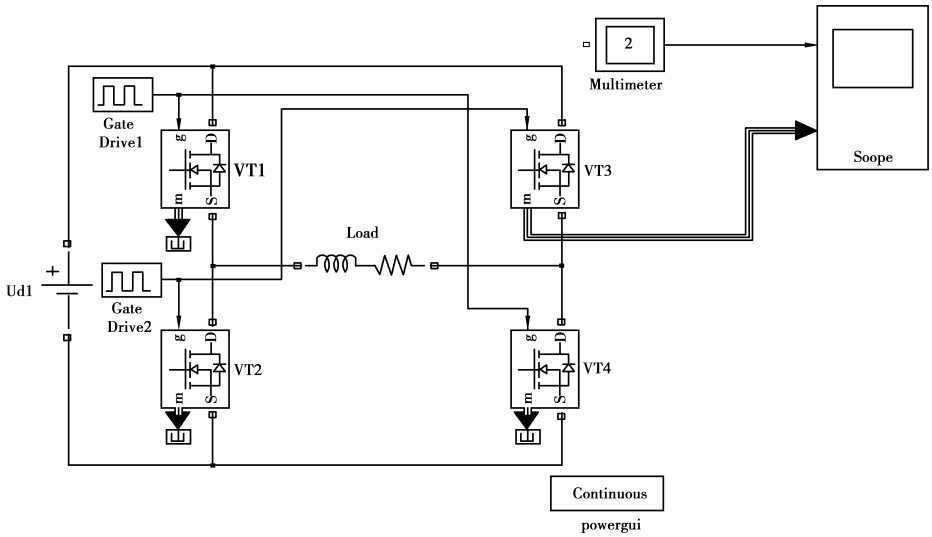

图 4　电路仿真

IGBT 栅极驱动 1 的参数设置和栅极驱动 2 的参数设置[7] 如图 5、6 所示。设置完成后,运行仿真,察看示波器里输出的波形图[7],如图 7 所示。

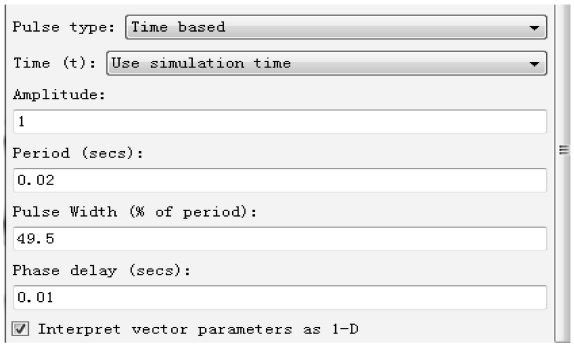

图 5　Gate Drive1 脉冲信号

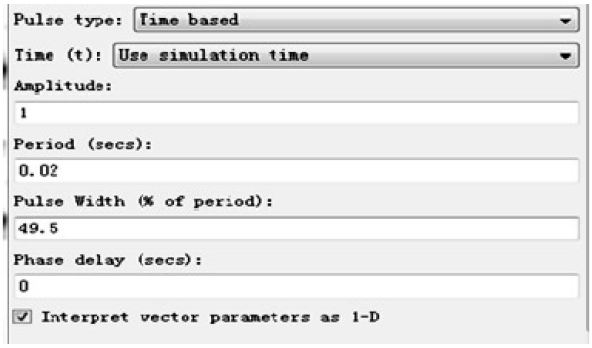

图 6　Gate Drive2 脉冲信号

在直流电源电压 V_d 一定时,输出电压的基波大小不可控,且输出电压中谐波频率低、数值大,直流电源电流 I_d 脉动频率低且脉动数值大。因此为了使负载获得良好的输出电压波形和减小直流电源电流的脉动,必须采用较大的 LC 输出滤波器和 LdCd 输入滤波器[1]。

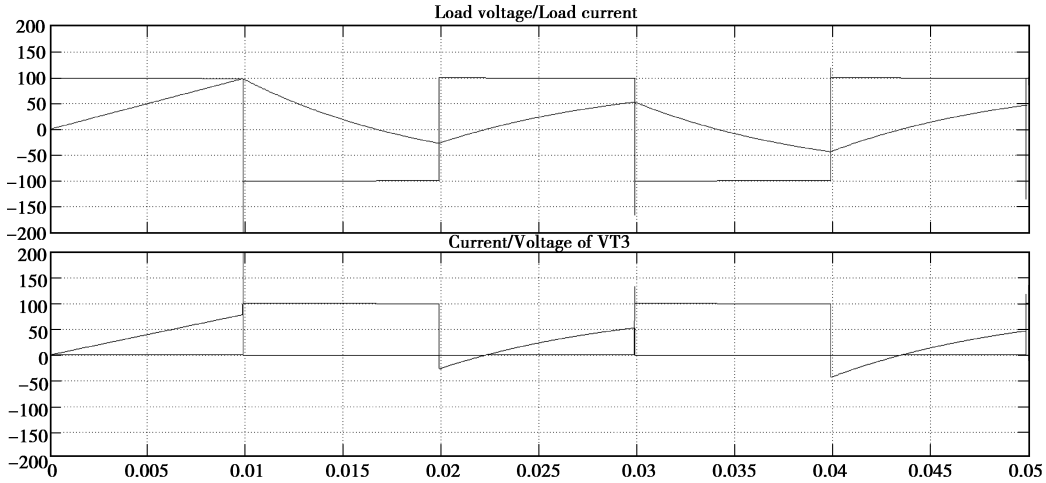

图7 电路仿真输出结果

3 总 结

经实践表明,采用VC与Matlab的混合编程,利用MFC对SIMULINK仿真程序进行二次开发,有效地搭建起电力电子电路仿真实验平台,实现多个电力电子典型电路的仿真,这种开发方案特点有成本低,开发周期短,扩展容易,有良好的用户交互界面等[8],充分地发挥了Matlab强大的计算能力,Simulink出色的仿真功能还有MFC的构建平台接口的能力,比单独使用C/C++语言开发更为简单,更加优越。

参考文献:

[1] 敖文刚.基于MATALB的可视化理论力学虚拟实验[J].重庆工商大学学报:自然科学版,2012,29(9):101-105
[2] 刘桂英,粟时平.《电力电子技术》的Matlab/Simulink教学仿真实践[J].电气电子教学学报,2011,33(1):87-90
[3] 吴文进,李彦梅,江善和,等.MATLAB仿真在《电力电子技术》课程教学中的应用[J].安庆师范学院学报:自然科学版,2013,02:124-127
[4] 刘艳.基于MATLAB的《电力电子技术》虚拟实验仿真平台[J].大连大学学报,2004(4):71-74
[5] 党智乾.基于MATLAB的电力电子典型电路虚拟实验平台的开发[J].电子测试,2013,19(8):81-82
[6] 田红霞.基于VC与Matlab的电力电子技术软件实验平台设计[J].实验室科学,2011(1):6-11
[7] 王春民,安海忠,王丰贵.基于VC和Simulink的电力电子仿真实验平台设计[J].吉林大学学报:信息科学版,2008(3):264-269
[8] 李雯.基于MATLAB的电力电子技术仿真分析[J].硅谷,2009,17(9):10-16

Simulation Experiment Platform for Power Electronics Technology

WANG Jun-hong, CHEN Yue-dong

(Anhui Key Laboratory of Electric Drive and Control, Anhui Polytechnic University, Wuhu 241000, China)

Abstract: For the shortage of teaching resources on average in power electronic technology practice teaching, in order to improve teaching quality and meet the needs of various teaching means, Visual C++ and Matlab is used to develop simulation experimental platform for power electronics. Simulation experiment platform can be used without other player software installation. Through repeatedly debugging parameters of the circuit and observing the effects of parameter changes on the circuit simulation results, students' ability to use knowledge and to innovate is improved.

Key words: power electronic technology; Matlab; VC++; the simulation experiment platform