

文章编号: 1005-8451 (2006) 11-0024-04

## 基于 ActiveX 机制的 VBMatlab 在故障诊断中的应用

杨 凡, 米根锁

(兰州交通大学 信电学院, 兰州 730070)

**摘 要:** 介绍 VB 调用 Matlab 的方法, 着重介绍 ActiveX 技术, 详述在 VB 中如何将输入的数据转化为 Matlab 能识别的二维数组, 给出具体实例—分析总结模拟电路的故障, 得到训练样本, 给出接口程序, 通过此应用, 充分证明: 二者混合编程能有效优化系统性能, 提高效率, 降低编程难度, 缩短软件开发周期。

**关键词:** ActiveX; VB; Matlab; 故障诊断; 应用

**中图分类号:** U279.2 : TP39 **文献标识码:** A

### Application of Matlab by VB based on ActiveX principle to fault diagnosis

YANG Fan, MI Gen-suo

(College of Information and Electrical Lanzhou Jiaotong University, Lanzhou 730070, China)

**Abstract:** It was described the methods of calling Matlab by means of VB, and paid attention to introduce the ActiveX principle, given the ways to transfer the data inputed in the TextBox of VB into two dimensional array identified by Matlab, given specific examples to analyse and sum up fault diagnosis of Analog Circui, got learning sample, given interface procedures. By this application, it could be approved to optimize the System characteristics, improve efficiency, decrease programming difficulty and reduce the period of software development.

**Key words:** ActiveX; VB; Matlab; fault diagnosis; application

Matlab 是一种面向科学和工程计算的解释性语言, 它可根据不同处理对象的需要, 通过简单编程, 方便地完成相应的功能。但 Matlab 也存在自身的缺陷, 如只提供 C 和 Fortran 语言使用的编程接口, 无法在 VB 和 VC 中直接调用, 编写可视化界面的能力弱等。

目前, 利用 VB 和 Matlab 相互连接的常用方法

有: 借助 ActiveX 部件、借助 DDE 技术、借助编译 M 文件, 引入 MatrixVB 等。本文主要介绍如何利用 ActiveX 技术实现在 VB 中调用 Matlab。

### 1 ActiveX 技术

ActiveX 是一种支持组件集成 MicroSoft Windows 协议, 通过 ActiveX 技术可以将不同环境下

收稿日期: 2006-06-28

作者简介: 杨 凡, 在读硕士研究生; 米根锁, 副教授。

实现代码参数化管理, 利于移植, 对系统主页面采用模块化程序管理, 使用单位可在参数库添加新开发的扩展功能模块。

### 4 系统特点

(1) 方便客户使用: 由于使用了此查询系统, 用户可以更好地了解现阶段国家实行的各项铁路规章制度, 真正做到了“手指一点、信息全知”。另外货主还可以在网和专业人士进行动态交流。从而真正服务了大众。

(2) 提高工作效率: 采用本查询系统后, 只要对各种规章制度进行一次录入, 就可以满足各个阶

段各个层次不同用户的要求, 由于只要派专人定期对系统进行维护和升级就能保证系统高速稳定的运行下去。因此, 既降低了劳动强度, 又减少了重复作业和差错, 从而提高了工作效率。

### 5 结束语

系统使用大型关系数据库系统和 ASP 编程语言, 从而使以静态为主的 Web 带来全新的动态效果, 使站点具有更加灵活和方便的交互性, 在 Internet 中实现信息的传递。此外, 系统对维护人员分层授权管理, 使维护人员各自负责相应的数据维护, 以增强系统的可靠性、安全性和可扩充性。

开发的组件集成到一个应用程序中, ActiveX 同时又是面向对象技术的一种, 属于组件对象模型 (COM) 的子类。COM 为所有的 ActiveX 对象定义了对象模型, 每个 ActiveX 对象支持一定的接口, 也就是不同的方法、属性和事件。利用这些方法执行相应的动作, 帮助客户端实现一定的功能。

Matlab 支持 ActiveX 技术: ActiveX 控制容器和 ActiveX Automation。ActiveX 控制容器可以将不同的 ActiveX 控件集成在一个应用中, 而 ActiveX Automation 是一种允许一个应用程序 (客户端) 去控制另一个应用程序 (服务器端) 的协议。在本文中, Matlab 是服务器端, 而 VB 是客户端, 通过 Automation, 客户端可以执行 Matlab 命令, 并与 Matlab 的工作空间进行数据交换, 以达到调用 Matlab 来训练神经网络的目的。

## 2 VB 通过 ActiveX 自动化 Matlab 的方式

Matlab ActiveX 自动化服务的功能主要包括在 Matlab 工作空间执行 Matlab 命令, 直接从工作空间存取矩阵等。

Matlab 提供了一个自动化对象, 其外部名为 Matlab.Application, 它支持 COM 技术。其它程序 (如 VB) 通过 COM 技术提供的函数, 得到自动化服务器对象支持的指针, 通过调用接口函数控制和使用自动化对象。在 VB 应用程序中创建了 Matlab 的 ActiveX 对象后, 就可以使用这个对象所包含的各种方法来实现对 Matlab 的调用。Matlab.Application 对象含有 3 种方法, 它们是 Execute、GetFullMatrix 和 PutFullMatrix, 具体如下:

### (1) BSTR Execute ([in]BSTR Command);

BSTR 表示宽字符串类型, 它与 VB 存储字符串所采用的数据格式相同。该方法接收字符串命令并在 Matlab 中执行, 将结果以字符串形式返回。

### (2) void GetFullMatrix ([in]BSTR Name

[in]BSTRWorkspace,

[in, out]SAFEARRAY (double) \*pr,

[in,\*out]SAFEARRAY (double) \*pi);

GetFullMatrix 方法从指定的工作空间检索一个完整的实型或虚型 mxArray, 其实部和虚部被分别存放到两个单独的 Double 型数组中。可以利用 GetFullMatrix 方法将 Matlab 中的一个矩阵变量传送到 VB 程序中的一个一维或二维数组中, 其中 Name

指定了 Matlab 中的矩阵变量名, Workspace 指定了该矩阵所在的 Matlab 工作区, pr 是指 VB 程序中的实部数组, pi 是指 VB 程序中虚部数组。

(3) void PutFullMatrix ([in]BSTR Name, [in]BSTRWorkspace,

[in, out]SAFEARRAY (double) \*pr, [in, out]SAFEARRAY (double) \*pi);

PutFullMatrix 方法将一个 mxArray 放入指定的工作空间。各参数的含义及调用方法与 GetFullMatrix 方法类似。

## 3 VB 中获得二维数组的方法

如上介绍的 PutFullMatrix 方法需要将 VB 程序中的一个一维或二维 Double 型数组传送到 Matlab 的一个矩阵变量中, 而 VB 接收的输入数据是 String 型的, 且所有的输入数据是做为一个整体出现的。

转化方法: 通过多行文本框进行输入, 输入数据时, 每行数据对应数组中的一行。一行的数据由逗号分隔, 一行输入完毕后按回车键表示一行的结束。

```
oldstr = Text1
```

```
i=0
```

```
Do While oldstr <> ""
```

```
    n1 = InStr(oldstr, Chr(10))
```

```
    If n1 > 0 Then
```

```
        newstr = Left$(oldstr, n1 - 1)
```

```
        oldstr = Right$(oldstr, Len(oldstr) - n1)
```

```
        Do While InStr(newstr, ",") <> 0
```

```
            i = i + 1
```

```
            n2 = InStr(newstr, ",")
```

```
            If n2 <> 0 Then
```

```
                linestr(i) = Left$(newstr, n2 - 1)
```

```
                newstr = Right$(newstr, Len(newstr) - n2)
```

```
            End If
```

```
        Loop
```

```
    End If
```

```
Loop
```

最后将 linestr(i) 中的数据赋值给二维数组。

以上方法经过试验, 是可行的。

## 4 实例中用到的模拟电路的故障及数据

模拟电路如图 1。

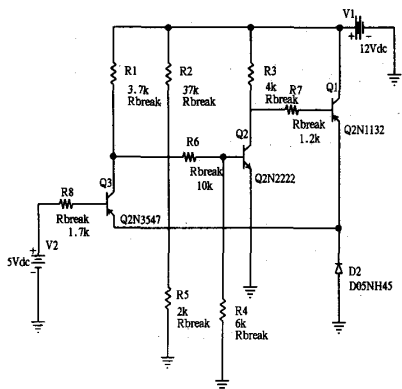


图1 模拟电路图

图中包含3个三极管，8个电阻，1个二极管。对二极管与三极管进行故障诊断，故障定义如表1。

表1 故障定义表

| F0 | NORMAL | 正常         |
|----|--------|------------|
| F1 | Q3BO   | Q3 基极开路    |
| F2 | Q2BES  | Q2 基极发射极短路 |
| F3 | Q2BO   | Q2 基极开路    |
| F4 | Q1BO   | Q1 基极开路    |
| F5 | DS     | 二极管短路      |

通过用PSPICE进行模拟各种故障态，可以得到各种故障状态下及正常状态下，各测试节点的电压的理论值，建立故障状态表，如表2。

表2 故障状态表

| V1     | V3      | V5      | 故障 |
|--------|---------|---------|----|
| 5.481v | 73.36mv | 720.6mv | F0 |
| 8.892v | 71.62mv | 726.6mv | F1 |
| 5.453v | 1.094mv | 0mv     | F2 |
| 5.703v | 1.7mv   | 165.6mv | F3 |
| 5.480v | 1.16mv  | 695.9mv | F4 |
| 5.481v | 0mv     | 720.6mv | F5 |

对V1、V3和V5点电压进行数据归一化处理后，得到输入神经网络进行训练的数据：

$P=[0.616, 1, 0.613, 0.641, 0.616, 0.616,$   
 $1, 0.976, 0.015, 0.023, 0.158, 0,$   
 $0.992, 1, 0, 0.228, 0.958, 0.992];$   
预期的输出数据为：  
 $T=[100000, 010000, 001000,$   
 $000100, 000010, 000001];$

5 基于实例具体设计的流程

基于前面介绍的模拟电路的故障诊断，将采用BP神经网络中的traingdm方法。这样，在设计可视

化界面时，在Form表单上需要8个单行文本框，分别对应：输入数据个数、隐层节点数、输出数据个数、训练时的参数Epochs、Show、Goal、Lr、Mc；3个多行文本框对应的是：输入训练数据P、预期输出数据T、实际输出数据；4个按钮则对应：获得输入数据、获得预期输出数据、网络训练、退出。在将多文本框的String型数据（如P、T）转换为Double型二维数组时，采用了前面第3部分介绍的方法，效果不错。此代码段分为6部分：

(1) 定义初始化变量；(2) 将文本框输入的故障诊断的数据转化为matlab能接受的二维数组；(3) 取得BP训练时所需的其它参数；(4) 通过ActiveX技术VB调用matlab进行网络训练，并输出训练后的值在文本框中；(5) 将文本框输入的故障诊断的预定输出值转化为二维数组；(6) 退出表单。

下面仅给出第4部分的代码：

```
Private Sub Command2_Click()  
Set MatLab = CreateObject  
("MatLab.Application")//VB调用Matlab  
Call MatLab.PutFullMatrix("P", "base",  
Sample_I, MImag1)//将Sample_I的数据放入P中  
Call MatLab.PutFullMatrix("T", "base",  
Sample_O, MImag2)//意义同上  
//从VB界面获得进行traingdm训练所需的参数  
包括num_s、Epochs、Shw、Goal、Lr、Mc等。  
MReal(0) = num_s  
Call MatLab.PutFullMatrix("R1", "base",  
MReal, MImag)  
MReal(0) = Epochs  
Call MatLab.PutFullMatrix("E", "base",  
MReal, MImag)  
MReal(0) = Shw  
Call MatLab.PutFullMatrix("S", "base",  
MReal, MImag)  
MReal(0) = Goal  
Call MatLab.PutFullMatrix("G", "base",  
MReal, MImag)  
MReal(0) = Lr  
Call MatLab.PutFullMatrix("L", "base",  
MReal, MImag)  
MReal(0) = Mc  
Call MatLab.PutFullMatrix("M", "base",  
MReal, MImag)
```

文章编号: 1005-8451 (2006) 11-0027-04

## 差分全球定位系统在车站调车作业中应用的研究

张大鹏, 戴 钢, 朱国辉, 徐 悦

(北京交通大学 网络管理中心, 北京 100044)

**摘 要:** 针对目前编组站的实际工作特点, 提出车站 DGPS 应用系统的总体设计方案, 说明系统各个模块的功能, 并且详细介绍系统涉及的主要技术、原理、算法以及其它一些需要注意的问题。

**关键词:** GPS; 编组站; 铁路车站应用系统; 调车机

**中图分类号:** U297: TP39 **文献标识码:** A

### Research on application of DGPS technology to shunting operation in stations

ZHANG Da-peng, DAI Gang, ZHU Guo-hui, XU Yue

(Network Management Research Center, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** According to the work characteristics of the marshalling stations at present, it was given a design plan of the Railway Station Application System, explained the function of each module in this System, introduced the main technology, principle, algorithm and other needed problems of this System.

**Key words:** GPS; marshalling station; Railway Station Application System; shunting locomotive

目前,《车站行车工作细则》中具体规定的车站技术作业组织和管理工作中有很大一部分还需要人工计算、人工管理,并且对调车机的工作还不能实

现实时进行有效地监测,调车机工作数据不能及时准确采集和上报,导致调车现场有人和无人监管的情况下调车机工作有很大的不同,自动化程度还比较低,这不能达到铁路信息化的要求,不能满足优质、高效的运输生产管理的需要。所以,迫切需要一种新的管理工具来改变这种落后状况。采用铁路

收稿日期: 2006-05-28

作者简介: 张大鹏,在读硕士研究生;戴 钢,高级工程师。

MatLab.Execute ("R2=size(T,1);")//获得输出层神经元的个数

MatLab.Execute ("net=newff(minmax(P),[R1,R2],{'tansig','pur-

elin'},'traingdm');")//调用执行 MatLab 语句

MatLab.Execute ("net.trainParam.epochs=E;net.trainParam.goal=G;net.trainParam.show=S;net.trainParam.lr=L;net.trainParam.mc=M;")//同上

MatLab.Execute ("[net,tr]=train(net,P,T);")

Text9.Text = MatLab.Execute

("A=sim(net,P);disp(A);")// MatLab 语句执行完后,将结果输出到 VB 界面

MatLab.Execute ("E=T-A;MSE=mse(E);")

End Sub

当隐层数取 7 时,训练的步数最少,得到的输出数据如下:

1.0022,-0.0046,-0.0157,0.0186,0.0278,-0.0306  
0.0017,0.9984,-0.0110,0.0114,-0.0018,-0.0008

0.0016,-0.0051,0.9870,0.0162,0.0389,-0.0403  
-0.0016,0.0058,0.0137,0.9830,-0.0465,0.0475  
0.0003,0.0237,0.0202,-0.0354,0.7276,0.2662  
-0.0009,-0.0271,-0.0094,0.0266,0.3213,0.6890  
从输出结果看,训练是成功的。

## 6 结束语

通过示例模拟电路故障诊断这个实例,可以看到 VB 基于 ActiveX 技术能成功调用 Matlab 软件,既达到了界面可视化效果,又使 Matlab 的神经网络训练数据的功能得到了充分的发挥,更加利于操作。

**参考文献:**

- [1] 张宏利,陈 华.实现 VB 与 MATLAB 数据交换的新方法[J].计算机应用与软件,2004,21(12):51—52.
- [2] 汤红梅,张 军.基于 ActiveX 的 Matlab 与 VB 接口技术在仿真软件设计中的应用[J].煤矿机械,2004(12):86—87.