

MATLAB 语言与 ActiveX 技术

ActiveX 部件是 Microsoft 公司提供的一种用于模块集成的新协议，它是 Visual Basic 工具箱的扩充部分。ActiveX 部件是一些遵循 ActiveX 规范编写的可执行代码，比如一个 .exe, dll 或 .ocx 文件。在程序中加入 ActiveX 部件后，它将成为开发和运行环境的一部分，并为应用程序提供新的功能。ActiveX 部件保留了一些普通 VB 控件的属性、事件和方法。ActiveX 部件特有的方法和属性大大地增强了程序设计者的能力和灵活性。

当前的新软件一般都支持 ActiveX 部件的嵌入，所以令现代的程序设计不再是一个个孤立的程序，而可以是一些在各个方面有优势的软件集成。这确实是软件业的重大革命。10 年前英国的一个大型 控制系统计算机辅助设计环境 ECSTASY 因为在 MATLAB 和其他软件集成中遇到困难而搁浅。在现代技术条件下，做那样的集成算不得什么难事。

MATLAB 支持的 ActiveX 有两个方面的涵义：

- (1) 在其他程序下运行 MATLAB 的 ActiveX 部件；
- (2) 在 MATLAB 下运行其他 ActiveX 部件。

从某种意义上来说，ActiveX 技术的应用有些像 DDE 链接，但比 DDE 链接更强大，所以 ActiveX 出现后，链接 DDE 的方法就不再那么吸引人了。与 ActiveX 的链接将成为主要的链接方式。ActiveX 实际上是在调用一种部件，而无需要求该部件对应的程序正在运行，这会使得其功能更加强大。

我们可以由一个例子来演示在 Microsoft Visual Basic 下，通过 ActiveX 自动化接口 将 MATLAB 作为 Visual Basic 语言的一个 ActiveX 部件调用的方法与应用。

[例 7-15] 在 VB 下建立一个新窗口，其上有 3 个控件，其中一个是按钮，其 VB 名称为 Command1，且其标注为 Run MATLAB；一个是编辑框，用来接收 MATLAB 命令，其 VB 名称为 Text1；另一个为文本控件，其 VB 名称为 Label1。我们的目的是在用户按下 Run MATLAB 按钮时，该程序中运行编辑框中给出的 MATLAB 语句，并将其结果在 Label1 控件中显示出来。

[解答] 建立一个 VB 窗口，在其变量声明部分申明要调用 MATLAB 的 ActiveX

```
Dim Matlab As Object
```

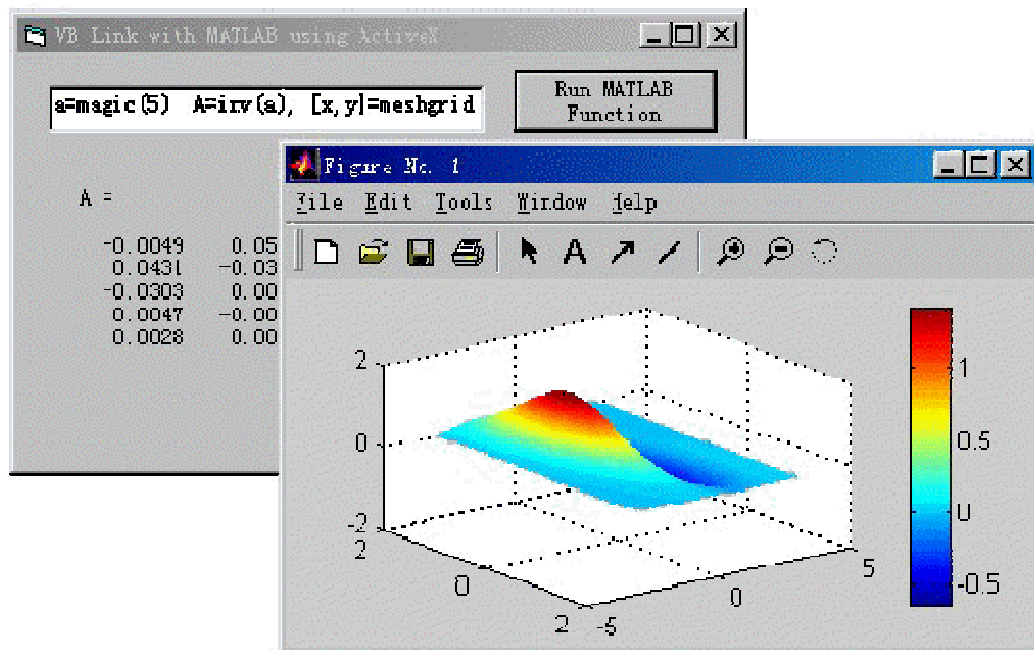
并在其 Form_Load 中给出下面命令来链接 MATLAB 的 ActiveX 部件。给按钮给出下面的响应命令，整个问题就坚决了：

```
Private Sub Command1_Click()  
Set Matlab = CreateObject("Matlab.Application")  
Label1.Caption=Matlab.Execute(Text1.Text)  
End Sub
```

值得指出的时，这个程序段不只可以解决计算与文字显示的 MATLAB 语句，还可以解决图形绘制等问题。例如在编辑框中给出下面带有绘图命令，

```
a=magic(5); A=inv(a), [x,y]=meshgrid(-3:0.1:3,-2:0.1:2);  
z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y);surf(x,y,z); colorbar;  
shading interp
```

则将得出如图所示的效果（窗口中框原来应该有白色背景，作透明处理后变成了这样）。



可见，调用 ActiveX 还是比较简单的。

[例 7-16] 我们还可以将其中生成的 MATLAB 变量调至 VB 环境中，由 VB 程序直接使用，或将 VB 中的变量写入 MATLAB 的工作空间。能解决这样问题的 MATLAB 函数为 `GetFullMatrix()` 和 `PutFullMatrix()`。例如我们可以在变量声明程序段中给出

```
Dim MatLab As Object  
Dim Result As String  
Dim MReal(10, 10) As Double  
Dim MImag() As Double  
Dim MReal1(5, 5) As Double  
Dim i, j, i0, j0 As Integer  
Dim b0 As Double
```

并在按钮响应区内写出

```
Private Sub Command1_Click()  
Set MatLab = CreateObject("Matlab.Application")  
MatLab.Execute ("a = hilb(11);")
```

```

Call MatLab.GetFullMatrix("a", "base", MReal, MImag)
b0 = 0
For i = 0 To 10
b0 = b0 + MReal(i, i)
Next i
Label1.Caption = b0
i0 = 0: j0 = 0
For i = 0 To 10 Step 2
For j = 0 To 10 Step 2
MReal1(i0, j0) = MReal(i, j)
j0 = j0 + 1
Next j
i0 = i0 + 1: j0 = 0
Next i
Call MatLab.PutFullMatrix("BMat", "base", MReal1, MImag)
End Sub

```

则可以生成一个 11x11 Hilbert 矩阵，调入 VB 空间，由 VB 提取其子矩阵，显示迹并将子矩阵返回到 MATLAB 工作空间。

[例 7-17] MATLAB 还可以调用其他 ActiveX 部件，这可以通过 actxcontrol() 函数来设置，该函数的调用格式为：

```
h=actxcontrol(ProgID, Position, Parent, EventCall)
```

其中，ProgID 为 ActiveX 部件的标识；Position 为部件的位置；Parent 为父窗口句柄；而 EventCall 为事件与回调函数对，它应该由单元字符串数据表示。在一般应用中，ProgID 应该由开发者提供。

Global Majic Software 公司为 Simulink 提供了一些用于表盘显示的 ActiveX 部件，已知其中一个 ActiveX 部件标识为 mwagauge.agaugectl.1,这样我们可以使用它演示 MATLAB 对 ActiveX 部件的调用。

假设我们想建立一个窗口界面，在窗口上画出一个 ActiveX 表盘，同时在它的旁边画一个垂直滚动杆，在滚动杆的位置发生变化时，在表盘上自动显示滚动杆的当前位置。

```

hf=figure('Position',[120 370 316 140]);
h=actxcontrol('mwagauge.agaugectl.1',[50,10,130,110],gcf);
h1=uicontrol('Position',[250 10 20 120],'Style','slider','Min',0,'Max',10,...
'Tag','mySld','CallBack',set(h,"_NeedleValue",get(gco,"Value")));

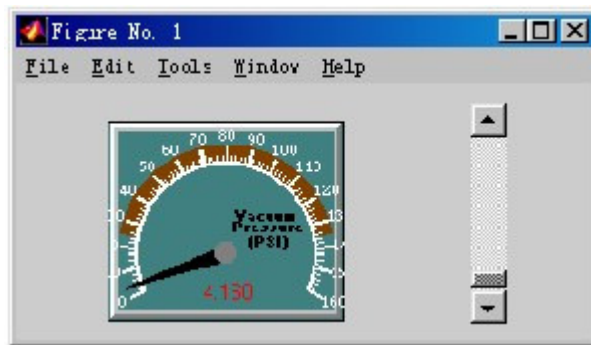
```

还可以用下面的命令选择一个新的，更美观的表盘

```

set(h,'Configuration','\Angular Gauge\General Meters\Vacuum Pressure');
set(h1,'Max',get(h,'ScaleMaxValue'),'Min',get(h,'ScaleMinValue'));

```



[例 7-18] Prog_ID 获取有时是个难点。有的程序不直接提供其 Prog_ID, 而用户需要从 Windows 注册表中获得。例如我们想得出 Windows Media Player 的 ActiveX 部件 Prog_ID, 则需要首先运行 Windows 下的 regedit 程序, 从其下庞大的注册表中找出 HKEY_LOCAL_MACHINE\Software\CLASSES 栏目, 并从中找出该程序的代号为 MediaPlayer.MediaPlayer.1。我们可以给出下面的语句

```
hf=figure('Position',[120 370 316 100]);
h=activexcontrol('MediaPlayer.MediaPlayer.1',[20,10,260,80],gcf);
set(h,'FileName','dadao.au');
```

在 MATLAB 窗口下建立一个如图所示的窗口界面



其中 dadao.au 是存放声音的文件, 更一般地它可以是任意的多媒体文件。例子中的文件存放“大刀向鬼子们头上砍去”歌曲的 au 文件, 可以从[这里下载 \(346K\)](#)并将其放在 MATLAB 的 work 目录下测试。如果放在其他路径下, 则应在引用文件时给出绝对路径。