

Faq-060111: 将 CST 软件与 Matlab 连接

该 FAQ 条目的说明了如何将 CST 微波工作室 (CST MWS) 与 MATLAB 一起使用。MATLAB 是以线性代数和矩阵数学为基础的科学计算程序。COM (组件对象模型) 和 ActiveX 用来通过 CST MWS 的 VBA 宏语言与 MATLAB 连接。CST MWS 与 MATLAB 连接的主要目的在于想利用 MATLAB 的广泛的数据处理、信号处理和图形处理能力。

COM 和 ActiveX 接口允许两个程序在工作后台的数据传递和紧密结合。这里所描述的连接方法对 CST 工作室的其他产品都适用, 同样也适用于将 CST 工作室与任意其他以 Windows 为基础的第三方软件相连接, 比如: MATHCAD, MATHEMATICA, MAPLE 等。

1. 背景

该 FAQ 条目需要用户熟悉与 CST MWS VBA 兼容的宏语言。如果用户对宏语言不熟悉, 可以参考 CST 微波工作室的 Advanced Topics manual (高级用户手册)。

CST MWS 可以设置成控制 MATLAB 或者是受 MATLAB 控制。当 CST MWS 控制 MATLAB 时, CST MWS 是客户系统, MATLAB 是服务系统, 反之亦然。该用户说明主要是关于 CST MWS 作为客户系统, MATLAB 作为服务系统。

在 MATLAB 的主页 www.mathworks.com 上可以得到更多的使用和编程方面的相关信息。MATLAB 中 COM 对象技术的相关说明也可以通过在 MATLAB 命令窗口中输入以下命令来获得。

```
h=actxserver('Matlab.Application');  
invoke(h);  
命令窗口中会出现以下语句:  
Execute=String Execute(String)  
GetCharArray=String GetChatArray(String,String)  
GetFullMatrix=Void GetFullMatric(String,String,Variant(Pointer),Variant(Pointer))  
MaximizeCommandWindow=Void MaximizeCommandWindow()  
MinimizeCommandWindow=Void MinimizeCommandWindow()  
PutCharArray=Void PutCharArray(String,String,String)  
PutFullMatrix=Void PutFullMatrix(String,String,SafeArray,SafeArray)  
Quit=Void Quit()
```

以上指令可以用于在 MATLAB 中发送、获取和处理数据。该执行方法可以用于 MATLAB 中任意语句, 让用户感觉到与直接在 MATLAB 的命令窗口中编程一样。

2. CST 与 MATLAB 连接例子说明

以下将通过一个具体的例子来说明 CST 与 MATLAB 是如何连接的。该例子通过一个 CST VBA 宏 MATLAB Data Exchange 来实现。

该宏主要是关于在 CST MWS 和 MATLAB 之间交换数据。数据交换将 CST 项目中的仿真结果 S11 的数据线性地传递给 MATLAB, 由 MATLAB 计算 S11 的 dB 值, 再将计算后的值送回到 CST MWS, 并在 1D Result 中画出。

3. 具体处理过程

这个宏用来证明 COM 对象方法的正确使用, 同样它们仅仅局限于一个端口/一种模式的结构。很容易扩展到任意端口、任意模式。你使用 CST MWS 的电脑上必须安装 MATLAB。这些宏都可以集成到任何一个可仿真获得 S11 的 CST MWS 的项目中。

将该宏文件放置在项目目录下的 "Model\3D" 文件夹中, 这样仿真完成后选择菜单中的 Macros, 最下方列出的局部宏就是 "MATLAB Data Exchange"。

选择 MATLAB Data Exchange, CST MWS 将会启动 MATLAB, 在出现的 MATLAB 窗口中将会以 dB 的形式显示 S11 的值。确保 MATLAB 命令窗口依然开着, 点击 MATLAB 的命令窗口, 在命令行中输入 whos 命令显示这些内部变量。注意到这些变量在 MATLAB 中以 cst_data.mat 文件存储, 该文件存在 C:\Program Files\CST Matlab data 目录下。返回 CST MWS 窗口中, 在 1DResults-> Matlab-> S11dB 中可以找到在 MATLAB 中计算得到的 S11dB 值。

点击 CST MWS 的 Macros 菜单, 在下拉菜单中选择 Edit/Move/Delete VBA Macro 子菜单。从结果列表中选中 MATLAB Data Exchange, 并且选择 edit。MATLAB Data Exchange 的 VBA 脚本将会在 VBA editor 中出现。你可以滚动察看应用细节。脚本的第一部分标志为 user defined settings (用户定义的设置)。Matlab_visible 可以被设置为 True 或者 False, 使得 MATLAB 命令窗口在计算结束时保持开启或者关闭状态。

User_path 中包含了所有文件存储的完整路径和 MATLAB 的设置路径。Save_file_name 文件中指定了 MATLAB 数据存储的 .mat 文件的名称。独立的 MATLAB 脚本文件 (.m 文件) 则能够通过 COM 对象方法的执行而运行。

4. 宏 VBA 代码:

Option Explicit

Sub Main

```
'User defined settings
Dim matlab_visible As Boolean
Dim user_path, save_file_name As String
matlab_visible = True           'True to keep Matlab command window open, False to not display it
user_path = "C:\Program Files\CST Matlab data" 'path that Matlab opens to and any data is saved to
save_file_name = "cst_data"      'name of .mat file matlab saves all data to
```

'variable defn

```
'-----
'Matlab variables
  Dim matlab As Object
  Dim Result As String
  Dim sendA(2) As Double

'set up variables
  Dim NumPorts, NumModes, NumFreq, NumEng, NumTime As Integer
  Dim factorA, factorB As Double
  Dim j, i As Integer
```

'Result variables

```
  Dim eng_obj, sig_in_obj, sig_out_obj As Object
  Dim file_nameA As String
  Dim freqs() As Double
  Dim times() As Double
  Dim eng_time() As Double
  Dim eng_A() As Double
  Dim T_stop As Double
  Dim sig_in() As Double
  Dim sig_out() As Double
  Dim MImag() As Double
  Dim NumSamp As Integer
  Dim Fmin, Fmax As Double
  Dim S11dB() As Double
  Dim freqD() As Double
```

'MWS problem information

```
  NumPorts = Solver.GetNumberOfPorts
  NumModes = Solver.GetNumberOfPorts
  factorA = Units.GetFrequencyUnitToSI
  factorB = Units.GetTimeUnitToSI
  NumSamp = Solver.GetNFsteps
  Fmin = Solver.GetFmin
  Fmax = Solver.GetFmax
```

'Retrieve MWS data

```
'-----
  Set S11A = Result1D("a1(1)1(1)")
  NumFreq = Solver.GetNFsteps
  ReDim S11B(NumFreq-1)
  ReDim freq(NumFreq-1)
  ReDim S11dB(NumFreq-1)
  ReDim MImag(NumFreq-1)
  For j = 0 To NumFreq-1
    freq(j) = S11A.GetX(j)/factorA
    S11B(j) = S11A.GetY(j)
  Next j
```

'Matlab COM/ActiveX interaction

```
'-----
  '1) create COM object and initiate Matlab as an activeX server
  Set matlab = CreateObject("Matlab.Application")
  If matlab_visible Then
    Result = matlab.Execute("h=actxserver('Matlab.Application');set(h,'visible',1);")
  End If
  '2) send all MWS data to Matlab using Automation methods (see app note for listing)
  Call matlab.PutCharArray("user_path", "base", user_path)
```

```

Call matlab.PutCharArray("save_file_name","base",save_file_name)

Call matlab.PutFullMatrix("S11","base",S11B,MImag)
Call matlab.PutFullMatrix("freq","base",freq,MImag)
'3) use execute command to control Matlab engine
Result = matlab.Execute("cd(user_path)")
Result = matlab.Execute("S11dB=20.*log10(abs(S11));")
Result = matlab.Execute("plot(freq,S11dB);grid on;")
Result = matlab.Execute("title('S-parameters from CST Microwave Studio');")
Result = matlab.Execute("xlabel('Freq, GHz');")
Result = matlab.Execute("ylabel('S11, dB');")
Result = matlab.Execute("eval(['save ',save_file_name,'];");")
'4) retrieve all data from Matlab to display in MWS
Call matlab.GetFullMatrix("S11dB","base",S11dB,MImag)

'display or store Matlab data in MWS form
'-----
'1) write any data you wish to display to a file
fileA = "matlab_output.txt"
Open fileA For Output As #1
For j = 0 To NumFreq-1
    Print #1,CStr(freq(j)) & " " & CStr(S11dB(j))
Next j
Close #1
'2) insert Matlab data into result tree
    With ResultTree
        .Reset
        .Name "1D Results\Matlab\S11 dB"
        .Type "XYSignal"
        .Subtype "Linear"
        .Title "S11 in dB computed in Matlab"
        .Xlabel "Frequency, Hz"
        .Ylabel "dB"
        .File fileA
        .Add
    End With
End Sub

```