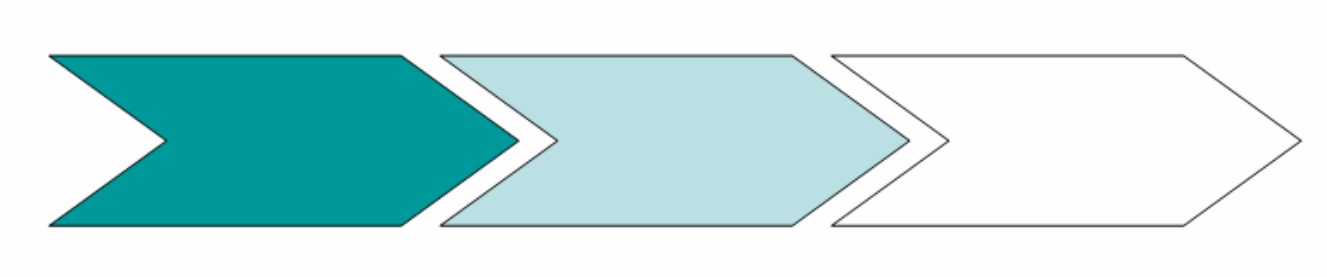


Linking MATLAB® and CST STUDIO

USER NOTE

This user note is centered on the use of CST MICROWAVE STUDIO® (CST MWS) with MATLAB®. MATLAB is a scientific computing program based on linear algebra and matrix mathematics. Component Object Models (COM) and ActiveX were used to interface with MATLAB via CST MWS's VBA macro language. The motivation for CST MWS <->MATLAB integration is to take advantage of the extensive data manipulation, signal processing, and graphics capability in MATLAB.

COM and ActiveX interfaces allow behind the scenes data transfer and tight integration between the two programs. The described approach can be applied on other members of the CST STUDIO product line as well. Linking CST STUDIO with any third parties Windows based tools, e.g. MATHCAD®, MATHEMATICA® , MAPLE® can be achieved using the technique described, too.

Keywords: COM, ActiveX, MATLAB®, MATHCAD®, MAPLE®, MATHEMATICA®

Originated by sws 08.11.2002
Last modification: btr v1.1 29.11.2004

Table of contents

Background

This user note assumes some familiarity with scripting in CST MWS's VBA compatible macro language. User's unfamiliar with the macro language can refer to CST Microwave Studio's Advanced Topics manual. This manual can be found in PDF form under C:\Program Files\CST Microwave Studio 5.1\Documentation. There is also extensive help and examples in the HTML based help pages. The HTML help pages are easily accessed via the drop down menu under Help->VBA Macro Language.

CST MWS can be configured to either control or be controlled by MATLAB. When CST MWS controls MATLAB, CST MWS is the client, and MATLAB is the server. When CST MWS is controlled by MATLAB, it is acting as the server and MATLAB is the client. This user note centers on using CST MWS as the client and MATLAB as a server.

There is extensive information on using and programming in MATLAB at www.mathworks.com. Attached to this user note is an excerpt from a Mathworks PDF file focusing on COM and DDE Support for use with MATLAB, entitled COM&DDE.pdf. One can also view available COM object methods by typing the following at a MATLAB command prompt:

```
h=actxserver('Matlab.Application');
invoke(h)
```
The following will be displayed in the command window:
```
Execute = String Execute (String)
GetCharArray = String GetCharArray (String, String)
GetFullMatrix = Void GetFullMatrix (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 ()
```

This is a summary of the available methods allowing a user to send, retrieve, and manipulate data in MATLAB. The Execute method can be used to implement any MATLAB command, allowing a user to program as if he/she was in the MATLAB command window.

Macro Overview

Macro 1 - Data Exchange
The first macro entitled “MATLAB / Data Exchange” focuses on exchanging data between CST MWS and MATLAB. Data Exchange passes S11 linear to MATLAB, computes S11dB, and returns this value to CST MWS to be plotted in the 1DResults tree.

Macro 2 - Signal Processing
The second macro entitled “MATLAB / Time Window” relies on data exchange methods, a FFT routine, and an interactive plotting window. Time Window allows a user to change the simulation end time and recalculates S11 based on this new stop time. The new stop time is chosen graphically from the energy plot. A CST MWS simulation is completed when the total system energy reaches the dB level set in Accuracy under Solver Settings. This setting is found in the transient solver dialogue box. Time Window then allows a user to investigate the effect of the Accuracy setting on the resulting S-Parameters.

Handling

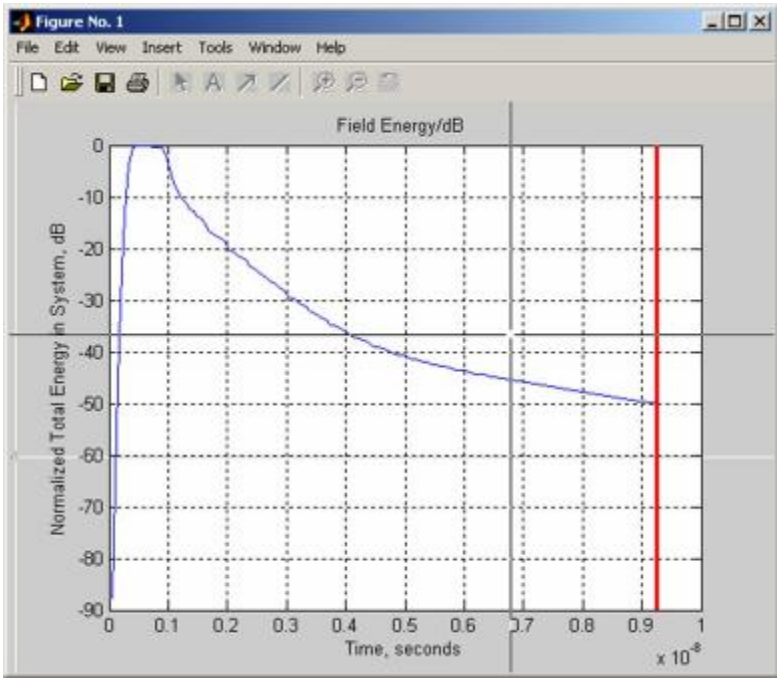
These macros are meant to demonstrate the correct implementation of the COM object methods, as such they are limited to one port/one mode structures. They can be easily expanded to include any number of ports and modes. You must have MATLAB installed on the PC where you are using CST MWS. These macro’s were successfully tested using MATLAB Release 12.1. Each macro has been incorporated into the CST MWS model file “antenna patch.mod”. To execute either macro open antenna patch.mod in CST MWS, and run the transient solver. After the simulation has completed select the Macro drop down menu. The last 2 macro’s listed should read “MATLAB / Data Exchange” and “MATLAB / Time Window”. The execution of “MATLAB / Time Window” requires the storage of four accompanying .m files as specified in the below details.

Macro 1 - Data Exchange
Select the Data Exchange macro. CST MWS will initiate MATLAB, and a MATLAB window will appear displaying S11 in dB. Also note the MATLAB command window has remained open. Now click in the MATLAB command window and type whos at the command prompt to display the internal MATLAB variables. You can now perform various signal processing and data analysis on these internal variables. Also note that these variables have been stored as a MATLAB data file as cst_data.mat. This file has been stored under C:\Program Files\CST Microwave Studio 5.1\Macros, which can be verified by typing cd at the MATLAB command prompt. Returning to the CST MWS command window the MATLAB computed S11dB can be found under 1DResults->Matlab->S11dB.
Now click Macros from the drop down menu in CST MWS, and select edit / Move / Delete VBA Macro. Highlight MATLAB / Data Exchange from the resulting list and choose edit. The VBA script for the MATLAB / Data Exchange macro will now appear in the VBA editor. You can now scroll down through the script to view the implementation details. Note that the first section of script is labeled user defined settings. Matlab_visible can be set to True or False, allowing the MATLAB command window to stay open or close after the calculation is completed.

User_path contains the complete path where all files are stored and the directory MATLAB is set to. Save_file_name specifies the name of the .mat file where the MATLAB data is stored to. Note that independent MATLAB scripts (.m files) can be run via the COM object method execute. This is will demonstrated in the Time Window macro.

Macro 2 - Time Window

The Time Window Macro requires the use of several independent .m files. These files are matlab_fft.m, nearest.m, calc_sparm.m, and energy_plot.m. These files must be saved in the directory specified by user_path. Click on the Macro drop down menu, and select MATLAB / Time Window. CST MWS will initiate MATLAB, and a MATLAB window will appear displaying the 1DResults energy plot. A red vertical line will appear at the end of the energy plot, indicating the current simulation stop time.



Bring the cursor over the MATLAB plot window, where a set of cross hairs should appear. Click anywhere on the graph to indicate the new simulation stop time. The vertical red line will be reset to the new simulation stop time, and S11 will be recomputed. The new S11 is then passed back to CST MWS and displayed in the 1DResults->Matlab->S11dB.

As in the Data Exchange macro one can then inspect the VBA script and conduct further analysis of the time signals in the MATLAB command window.

Conclusion

This user note discussed the use of CST MICROWAVE STUDIO® (CST MWS) with MATLAB®, the scientific computing program by MathWorks. Component Object Models (COM) and ActiveX were used to interact with MATLAB through CST MWS's VBA macro language. The use of CST MWS with MATLAB allows for greater flexibility in signal processing and data manipulation. A Data Exchange macro was presented using COM object methods to transfer data between CST MWS and MATLAB. A Time Windowing macro was also presented expanding on the transfer of data to include a computation of S parameters based on a new user defined simulation stop time.