
A Software Lab Notebook Integrating SigLab, MATLAB, and Word 6.0

Measurement instrumentation, such as SigLab™, which is tightly integrated with the MATLAB™ environment, allows educators, researchers, and practicing engineers to easily post-process and display the measurement data. The MathWorks® MATLAB Notebook document template allows "Point and click" control of SigLab's measurement applications from within a Microsoft® Word™ document. The measurement results appear in either graphical or tabular form directly in the Word document. An example is included which demonstrates the construction of a student lab experiment using this software technology.

A Historical Review: Getting Measurement Data Into Documents

Polaroid screen photos

MIT Lab Notebooks used to be stuffed with curled up Polaroid photos of a (burnt) phosphor screen. Many times, a whole package of film was wasted to get a decent shot. Applying the "fixer" and taping it into a notebook was art by itself. The last step was adding the scale settings of the instrument and what the measurement was about in the first place, but this step was often omitted in the rush to complete the process. It was a time-consuming task that did not result in optimal results.

Pen plotters

A step up from the Polaroid came in the form of a pen plotter connected to the measurement instrument. Of course, the instrument had to be capable of driving the plotter, and this usually meant the instrument was a high priced one. Unless the instrument drew its own axis, the graph paper had to be aligned in the plotter. The axis had to be manually labeled, and the pens often skipped and ran out of ink. The final sizing was done with the scissors, and Scotch Tape mounted the final work of art into a notebook.

Printers

Driving a dot-matrix or laser printer was another step in the evolution of hard copy technology. At this stage of the game, the relevant annotation and axis labels would usually automatically make it onto the paper. Post processing was also a possibility by recording front panel keystrokes or some other equally archaic method of "programming". The integration of the results into a document was still "cut & paste" and the font and color selections were sparse at best. Only the highest priced instruments properly supported this feature.

File transfer

Upon entering the Personal Computing age, many instruments used an internal micro-processor for human interface, storage and display functions. Price notwithstanding, it was by no means a technical breakthrough to add an optional floppy drive to the instrument! Measurement results could then be written to PC compatible files. Transferring the floppy to the PC, allowed the file to be read by a more general purpose computing platform. Format conversion routines either supplied by the instrument manufacturer or home grown could be applied to the data in the file to get the results into MATLAB. Once in the MATLAB environment, the data could be easily post processed and plotted or tabularized. Then, the results could be

imported by a word processing program. Less than seamless, but a real step up from those slimy Polaroid photos!

Today's Solution: Seamless Integration

PC integrated measurement instruments

Well on the way to the second millennium, the era of tight integration between measurement results and the ubiquitous PC is here. This integration facilitates a seamless transfer of measurement results into a document. The important measurement issues are:

- a) measurement type
- b) measurement quality
- c) ease of use
- d) software environment.

SigLab is a professional quality, audio bandwidth, dynamic signal and system analysis tool. An astounding variety of measurements in the audio frequency range can be made with SigLab. SigLab comes complete with turn-key measurement application software implementing numerous "virtual instruments".

The software allows a comprehensive array of measurements to be implemented with no programming required. The MathWorks' MATLAB technical computing environment is second to none for the analysis and visualization of engineering-oriented measurements.

Consequently, for measurements on signals and systems in the audio frequency range, the choice is simple—SigLab. Please consult the data sheets and application notes for more details on SigLab's measurement capabilities.

Questions and Answers Regarding The MATLAB NOTEBOOK

What is the "MATLAB Notebook" ?

In August of 1994, the MathWorks released a tool called "The MATLAB Notebook". This tool consisted of a Word 6 document template which allows MATLAB commands to be executed from within a Microsoft Word document. The graphical results are automatically imported and embedded in the Word document. Tabular results can also be embedded.

What technology does it use?

The Notebook uses a set of Word Basic macros and Dynamic Data Exchange routines to implement the communication with MATLAB. The underlying template and macro code are available for modifications by the user, if desired.

Does it work with SigLab?

SigLab runs under Windows 3.1 or Windows '95 and is controlled by Word 6. The following small enhancements to the underlying Notebook macros have been made. First, a variable called "EmbedTest" was added to control graphic embedding. This variable is tested by a Word macro. Second, a button in the toolbar was added to allow the execution of the MATLAB command script to be initiated with a mouse click rather than requiring a keystroke. Finally, modified virtual instruments are supplied with SigLab to optionally lock out file operations that could result in overwriting measurement setup files. The Notebook template (file: *dsptlab1.dot*) is available from Spectral Dynamics, Inc. Consult the factory about Word 7 support.

How does it work?

The act of opening the document invokes MATLAB automatically and from then on it is a simple matter of point and click.

First, create a document using the *dsplab1.dot* template. Enter the desired MATLAB commands. Select the command text by clicking and dragging with the mouse until the commands are highlighted. Then, use the Notebook/Define Input Cell menu to specify the selected text as commands which will subsequently be executed by MATLAB. To try the script, click anywhere in the previously entered MATLAB commands (now displayed in green) and press the *MatlabEval* button in the toolbar shown below.



Are the documents hard to create?

It is no harder to create than any other document... simply write the text, and the desired MATLAB scripts and/or functions. The functions and scripts are written in the usual way with only the addition of a single variable to control the graphical embedding process.

How about debugging?

Scripts can be actually be debugged from within the document due to the fact that even error messages are returned to the document.

An Example

A student lab notebook

One obvious application of this technology is in university student lab courses. Since SigLab's virtual instruments have their states saved and restored from a file, the time spent setting up measurement equipment is drastically reduced. Most will recollect how long it can take a student to set up a function generator and get the 'scope to trigger!

With a properly constructed software lab notebook, the student can focus on the learning objectives of the lab, rather than the details of setting up the measurement instrumentation.

However, if the lab involves gaining measurement expertise, the instruments will always start with the "knobs and switches" in a known state. This is a considerable benefit over traditional bench instrumentation.

The lab sessions can be issued from an individualized floppy disk or from a network server. Each lab session is a Word document based on the *dsplab1.dot* template. The session contains lab background material and tasks for the student to perform. Initially, the document has only the background material and (possibly) the MATLAB scripts. When the lab is completed the student has entered the required answers and measurement data.

An example of a completed lab session follows this document. See file lab1a.doc for the stand-alone lab session document.

Conclusion

The once tedious process of getting measurement data into documentation has been reduced to a point and click operation. The MATLAB/SigLab/Word combination provides professional quality measurement capability with unparalleled post processing and data display flexibility.

For more information contact:

*Spectral Dynamics
1010 Timothy Drive
San Jose, CA 95133-1042
Phone: (408) 918-2577
Fax: (408) 918-2580
Email: siglabsupport@sd-star.com
www.spectraldynamics.com*

Two-Port Network Analysis

The theoretical analysis of a simple two-port passive "Twin T" RC network will be performed. The student constructs the network and compares the analytical results with actual measurements made on the network. This Lab features an interactive software notebook using the Word 6, MATLAB, and SigLab.

Using Z Parameters to Describe a Two-Port Network.

There are numerous methods available for analytically describing linear time invariant electrical networks. These include parameterization by Z, Y, H, G, ABCD (chain matrix), and S parameters. The network to be analyzed and constructed will be in the audio frequency range hence the classical Z parameters are easy to measure and will be used. Any of the other representations can be obtained by appropriate linear transformations, including the S (scattering) parameters.

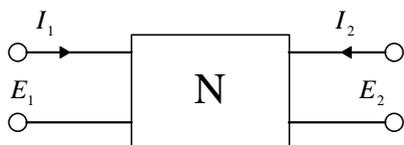


Figure 1 - A General Two-Port Network

The matrix equation for the Z parameters is

$$\begin{bmatrix} E_1 \\ E_2 \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \quad (1)$$

The z 's are called the open-circuit impedances, since typically the output or input port current is forced to zero (open circuited) to evaluate the parameter as shown in equations 2-5.

$$z_{11} = \frac{E_1}{I_1} \Big|_{I_2=0} \quad z_{21} = \frac{E_2}{I_1} \Big|_{I_2=0} \quad (2,3)$$

$$z_{12} = \frac{E_1}{I_2} \Big|_{I_1=0} \quad z_{22} = \frac{E_2}{I_2} \Big|_{I_1=0} \quad (4,5)$$

For a passive network like the Twin T, at most 3 of the four parameters are independent. The reciprocity theorem states that the transfer function of a passive network with the dimensions of impedance (or admittance) remains unchanged if the points of excitation and

response are interchanged. The theorem therefore demands that the transfer impedances z_{12} and z_{21} must be equal.

Furthermore, if the network is symmetrical with respect to the input and output ports, as is the Twin T, z_{22} must equal z_{11} .

Twin T Analysis

The circuit for the Twin T is shown in Figure 2.

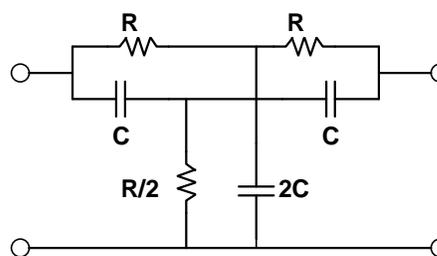


Figure 2 - Twin T network.

Analysis of the Twin T yields:

$$z_{11} = \frac{R^2 C^2 s^2 + 4RCs + 1}{4Cs(RCs + 1)} \quad (6)$$

$$z_{21} = \frac{R^2 C^2 s^2 + 1}{4Cs(RCs + 1)} \quad (7)$$

$$X_{21} = \frac{E_2}{E_1} \Big|_{I_2=0} = \frac{z_{21}}{z_{11}} = \frac{R^2 C^2 s^2 + 1}{R^2 C^2 s^2 + 4RCs + 1} \quad (8)$$

Task 1: Derive equations 6 and 7 using elementary circuit theory. Insert your derivation into this document. Use the equation editor and the drawing objects as needed. This task can be accomplished off line.

Solution to Task 1 (provided by student)

The indefinite admittance matrix for the Twin T is:

$$\begin{bmatrix} I_1 \\ I_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} g + cs & 0 & -g & -cs \\ 0 & g + cs & -g & -cs \\ -g & -g & 2(g + cs) & 0 \\ -cs & -cs & 0 & 2(g + cs) \end{bmatrix} \cdot \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix}$$

where $g = 1/R$ and $c = C$

This can be rewritten as:

$$\begin{bmatrix} I_A \\ 0 \end{bmatrix} = \begin{bmatrix} Y_{AA} & Y_{AB} \\ Y_{BA} & Y_{BB} \end{bmatrix} \cdot \begin{bmatrix} E_A \\ E_B \end{bmatrix} \quad I_A = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \quad E_A = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \quad E_B = \begin{bmatrix} E_3 \\ E_4 \end{bmatrix}$$

Rewriting the lower row gives:

$$0 = Y_{BA}E_A + Y_{BB}E_B \quad \text{therefore} \quad E_B = -Y_{BB}^{-1}Y_{BA}E_A$$

Rewriting the top row using the solution for E_B

$$\text{gives: } I_A = (Y_{AA} - Y_{AB}Y_{BB}^{-1}Y_{BA}) \cdot E_A \quad \text{which are then,}$$

after expansion, the Y parameters:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} \frac{c^2s^2 + 4gcs + g^2}{2(g + cs)} & \frac{-(g^2 + c^2s^2)}{2(g + cs)} \\ \frac{-(g^2 + c^2s^2)}{2(g + cs)} & \frac{c^2s^2 + 4gcs + g^2}{2(g + cs)} \end{bmatrix} \cdot \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

The inverse of the Y parameters are the Z parameters. The inverse of a 2x2 matrix is:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

For this example $ad - bc = 2gcs$. therefore

$$\frac{c^2s^2 + 4gcs + g^2}{2(g + cs) \cdot 2gcs} = \frac{R^2C^2s^2 + 4RCs + 1}{4Cs(RCs + 1)} = z_{11}$$

A similar calculation holds for z_{21}

Constructing a test circuit

This circuit can be constructed using readily available components. The above analysis assumes ideal zero tolerance components and clearly the agreement between the analysis and measurements will be determined by the tolerance of the components. For best results, use good quality mylar or polystyrene capacitors. Ceramic capacitors with a high dielectric constant are often non-linear and should be avoided.

Task 2: Construct the circuit shown in Figure 3.

Parts List:

Qty	Description	Use
5	1000 ohm Resistors	2 for R 2 in parallel for R/2 1 for z_{11} meas. (Rx)
4	0.047uF Mylar Cap	2 for C 2 in parallel for 2C
3	RCA Connectors	J1,2,3

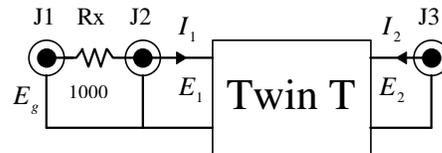


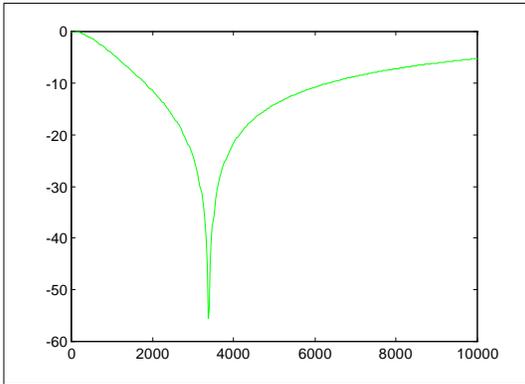
Figure 3 - Twin T Test Circuit.

The following MATLAB script (Script #1, embedded in this document) evaluates equations 6, 7, and 8 along the $j\omega$ axis over the dc to 10000 Hz region. As can be seen from the script, the analysis uses the component values given in the above parts list. The script computes z_{11} and z_{12} for subsequent use and a plot is made of the voltage transfer function X_{21} .

To run the script simply point and click anywhere in the following (green) MATLAB script, and the press the MatlabEval button on the toolbar below. The script will execute and a plot of the magnitude of X_{21} vs. frequency will be made.

Script #1

```
close all;
global EmbedTest;
EmbedTest='Embed_On';
f=0:25:10000;
jw=2*pi*f*j;
R = 1000;
C = 0.047e-6;
a=polyval([R^2*C^2,4*R*C,1],jw);
b=polyval([R^2*C^2,0,1],jw);
c=eps+polyval(4*C*[R*C,1,0],jw);
z11=a./c;
z21=b./c;
X21=b./a;
plot(f,20*log10(abs(X21)), 'g');
title('Twin T Voltage Transfer Function E2/E1 with I2=0.0');
xlabel('Hertz');
ylabel('dB');
```



Plot 1 - Computed Twin T voltage Transfer Function

Measuring the Circuits Parameters

To measure the actual Twin T circuit, an FFT based network analyzer will be used. This class of analyzer is capable of directly measuring the voltage transfer function (E_2/E_1) of any two-port system. The Z parameters can be obtained by simply inserting a resistor (R_x in Figure 3) to facilitate the measurement of impedance. With the aid of some simple MATLAB manipulations, z_{12} and z_{11} can be easily computed from the available measurements. Begin by setting up the test circuit and analyzer to measure the voltage transfer function X.

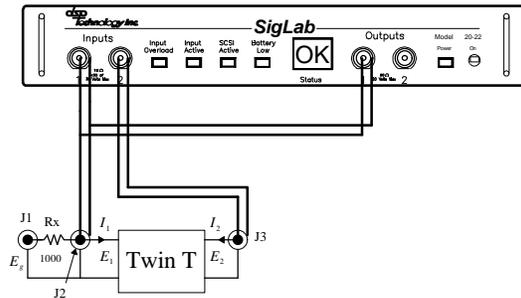


Figure 4 - Connecting the Analyzer to the Twin T Circuit for a Voltage Transfer Function Measurement.

Connect the analyzer to the network as shown in Figure 4. Using the RCA Y, and RCA patch cables provided with the analyzer. Notice channel 1 of the analyzer measures the two-port input voltage E_1 , while channel 2 measures the two-port output voltage E_2 . Since the analyzer input impedance is 1M ohm, and the cable capacitance has negligible reactance at the frequencies of interest, I_2 is assumed to be zero.

When the network has been connected to the analyzer, invoke the measurement with the cursor and MatlabEval button as before. The next Script (#2) does the following:

invokes the network analyzer
loads measurement setup from a file
starts the measurement

Note: Do not try to close the analyzer, just return to this document when the measurement display stops updating.

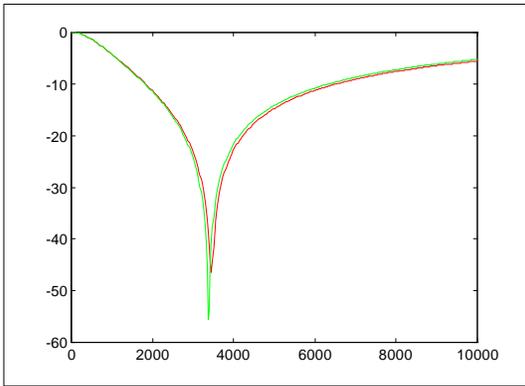
Script #2 `meas_xf1;`

Script 3 does the following:
extracts measurement data from analyzer
terminates the analyzer application
assuming Script 1 has been run, plots the measured data (in green) over the analytical data (in red).
returns a frequency vector Fvec, and measured transfer function X21meas to MATLAB.

Script #3

```

global EmbedTest;
[Fvec,X21meas]=get_xf1a;
% work around label problem
close(gcf);
EmbedTest='Embed_On';
plot(Fvec,...
     20*log10( abs(X21meas)),...
     'r'
     f,...
     20*log10( abs(X21)),...
     'g');
title('Voltage Xfer Function X21,
Computed=green, measured=red');
```



Plot 2 - Computed Twin T voltage Transfer Function Overlayed With Actual Measurement.

Experimenting with different display formats or measurement parameters is encouraged. If errors occur—simply re-run Scripts 2 and 3. These scripts are designed to work as a team, so *always* run them as a pair.

The z_{11} parameter will be measured next.

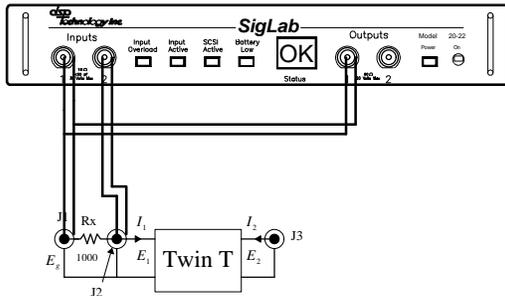


Figure 5 - Connecting the Analyzer to the Twin T Circuit for Input Impedance Measurement.

Since the analyzer is capable of measuring voltage transfer functions, it is easy to measure impedance by simply inserting a resistor R_x into the circuit as shown in Figure 3 and 5. Let the voltage transfer function X_m be defined as:

$$X_m = \frac{E_1}{E_g} \quad (9)$$

where E_g is the generator voltage measured with channel 1 of the analyzer and E_1 is the input voltage to the Twin T network. It then follows that

$$z_{11} = R_x \cdot \frac{X_m}{1 - X_m} \quad (10)$$

Connect the analyzer to the circuit as shown in Figure 5. Execute Script #4 to make the measurement of X_m .

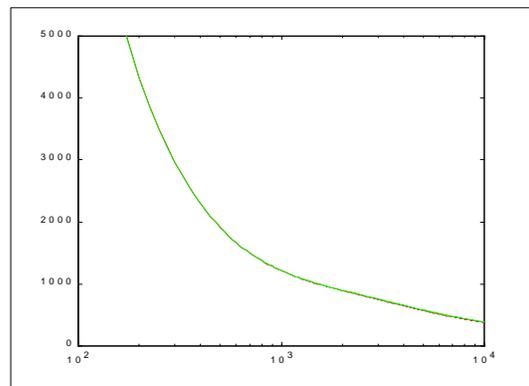
Script #4

```
meas_xf1;
```

Now, execute Script 5 to retrieve the measurement and perform the transformation defined in equation 10.

Script #5

```
[Fvec,Xm]=get_xf1a;
close(gcf);
EmbedTest='Embed_On';
Rx = 1000;
z11meas = Rx*Xm./(1-Xm);
plot(f,...
     abs(z11),'r',...
     Fvec,...
     abs(z11meas),'g');
set(gca,...
     'xlim',[100 10000],...
     'ylim',[0 5000],...
     'xscale','log');
title('Magnitude of z11,
computed=red measured = green');
ylabel('Ohms');
xlabel('Hertz');
```



Plot 3 - Twin-T Computed and Measured Magnitude of z_{11} in Ohms

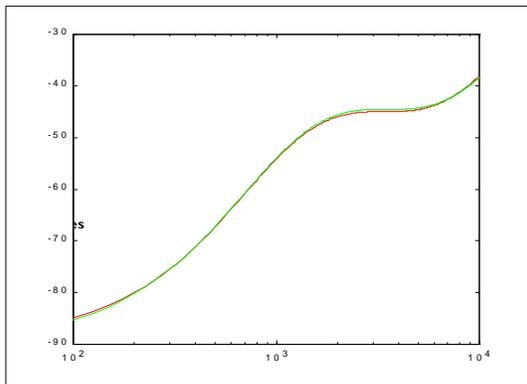
A comparison of impedance angles can also be made. Script 6, plots the angle in degrees for the theoretical and measured values of z_{11} .

Script #6

```

close(gcf)
EmbedTest='Embed_On';
plot(f,...
(180/pi)*atan2(imag(z11),...
                real(z11)),...
      'r',...
      Fvec,...
(180/pi)*atan2(imag(z11meas),...
                real(z11meas)),...
      'g');
set(gca,...
      'xlim',[100 10000],...
      'ylim',[-90 -30],...
      'xscale','log');
title('Phase of z11, computed=red
measured = green ');
ylabel('Degrees');
xlabel('Hertz');

```



Plot 4 - Twin-T Computed and Measured Phase of z_{11} in Degrees

Rearranging equation 8 gives

$$z_{21} = X_{21} \cdot z_{11} \quad (11)$$

so it is a simple matter of multiplication to produce z_{21} . Script 7 computes z_{21} using equation 11 and plots the magnitude of the trans-impedance in ohms for both the theoretical and measured values.

Script #7

```

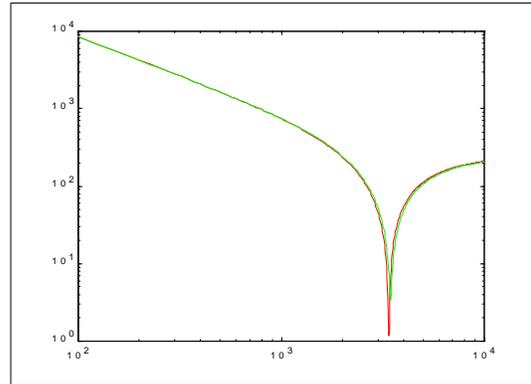
close(gcf)
EmbedTest = 'Embed_On';
z21meas=X21meas.*z11meas;
plot(f,...
      abs(z21),'r',...
      Fvec,...
      abs(z21meas),'g');
set(gca,...
      'xlim',[100 10000],...

```

```

      'ylim',[1 10000],...
      'xscale','log',...
      'yscale','log');
title('Magnitude of z21,
computed=red, measured = green');
ylabel('Ohms');
xlabel('Hertz');

```



Plot 5 - Twin-T Computed and Measured Magnitude of z_{21} in Ohms

Conclusion

The computation and measurement of the two-port Z parameters on a Twin T network have been made. The results indicate excellent agreement between the analytical and measured parameters. The Z parameters were measured by using an FFT based network analyzer and manipulating the measurements with some simple MATLAB scripts.

For more information contact:
Professor Foo Ling
Information Systems Department
Stratford University
Framingham, MA 01761
Phone: (508) 657-7555
E-Mail: fooling@eecs.stratford.edu

ATLAB