Developing a Matlab Interface to the Software Sonnet

Fernando Salas Barajas
ITESO

Introduction.

Sonnet Software provides commercial EDA software solutions for high-frequency electromagnetic (EM) analysis. Sonnet's suite of EM Software is aimed at today's demanding design challenges involving predominantly planar (3D planar) circuits and antennas. Predominantly planar circuits include microstrip, stripline, coplanar waveguide, PCB (single and multiple layers) and similar structures with vias, vertical metal sheets (z-directed strips) and any number of layers of metal traces embedded in stratified dielectric material. Electromagnetic analysis assumes the circuits are in a six-sided metal box to simulate a true metal enclosure, and accurately includes the effects of package resonances. Changing boundary conditions enables analysis under open environment conditions.

Viewing all this qualities that any other software don't offer like Sonnet does, converts it to Sonnet to an appetizing Software to explore and to manipulate.

This job pretends to begin working with this the Sonnet Software manipulating it from Matlab that is an excellent software that helps to make calculus and optimizations and a lot more. This can give to Sonnet another versatility y to wide its expectatives.

Objetives

- The main objective is to find a way to create a Sonnet file since Matlab and to execute it from this.
- To obtain response data and to manipulate to graphic it. (This will give the initialization to a wide way of analysis using a different and multifunctional tool called Matlab). *Note:* For this example it only will be manipulate the type of analysis and the band to be analyzed.

Overview of Sonnet

Sonnet hands 10 blocks for a geometry project –in which we will focus. Adding some more blocks for Netlists projects.

1. **Header Block.**
   The header block provide license and date information about the Sonnet Project.

```
FTYP SONPROJ 1 ! Sonnet Project File
VER 8.51.lite
HEADER
LIC SL220791.31804
DAT 04/26/2003 17:15:47
MDATE 04/20/2003 17:35:08
HDATE 04/20/2003 17:35:08
ANN This circuit is used for……………
END HEADER
```

The first two lines even thought they no belongs to the specifically to the Header block, are the begging of the file and mention if it is a Geometry or a Netlist project and the version of the software that is in use.

.

2. Dimensions Block.

Provides all the units to be used in the project file. There is an entry for each type of unit.

DIM
FREQ GHZ (HZ, KHZ, MHZ,GHZ,THZ,PHZ)
IND NH (H, MH, UH, NH, PH, FH)
LNG MIL (MIL, UM, MM, CM, IN, M) Default MIL
ANG DEG (Only DEG)
CON /OH (Conductivity siemens/meter)
CAP PF (Capacitance used for Netlists)
RES OH (Resistance only for Netlists)
END DIM

3. Frequency Block.

Contains all the frequency sweeps which have been input in a project.

FREQ
SIMPLE 4.0 8.0 0.25 (Linear frequency sweep)
ABS 4.0 8.0 (Analysis band synthesis –ABS.)
END FREQ

4. Control Block.

Specifies the type of analysis sweep presently defined for the project.

CONTROL
SIMPLE
OPTIONS -d (options depend of the data required for analysis)
SPEED 0 (How precise will be the analysis)
END CONTROL.

Note: For this example we are going to manipulate the word SIMPLE that is shown in this block. We are going to be switching from the SIMPLE word to the ABS word. This is to make an Analysis in a simple frequency or in an Adaptive mode (ABS).

5. Geo Block.

Specifies the circuit geometry in a geometry project.

```
GEO
TMET "Lossless" 0 SUP 0 0 0 0
BMET "Lossless" 0 SUP 0 0 0 0
MET "Aluminum" 1 SEN 0
BOX 1 330 200 66 40 20 0
   100 1 1 0 0 0 2 "Air"
   20 9.8 1 0 0 0 2 "Alumina"
POR1 STD
POLY 9 1
3
1 50 0 0 0 0 100
POR1 STD
```

POLY 9 1
1
2 50 0 0 0 330 100
NUM 3
0 5 -1 N 9 1 1 100 100 0 0 0 Y
0 90
330 90
330 110
0 110
0 90
END
.
.
.

END GEO

6. Opt.

Specifies the sweep, parameter data range and optimization goals for an optimization.

OPT
MAX 100 (Maximum number of interactions)
VARS (optimitation Parameters settings)
END OPT

Note. When some of the blocks or utilities have not been used in an analysis, Sonnet anyways declares the block without completing all the structure that must be declared to run the option.

7. Varswp. (Parameter sweep block)

Specifies a frequency sweep and data range for each parameter sweep. It is declared only if varswp is used in the control block

VARSWP
END VARSWP

Note. This is another example of the blocks or utilities that have not been used in an analysis.

8. FILEOUT.

Specifies an output file which allows you to store response data from your analysis to outside your project file.

FILEOUT
CSV D Y $BASENAME.csv 8 S MA R 50.00000
END FILEOUT

Note. This block will be pretty important to obtain the resulting data from the analysis of the Sonnet file which will be manipulated from Matlab.

9. QSG

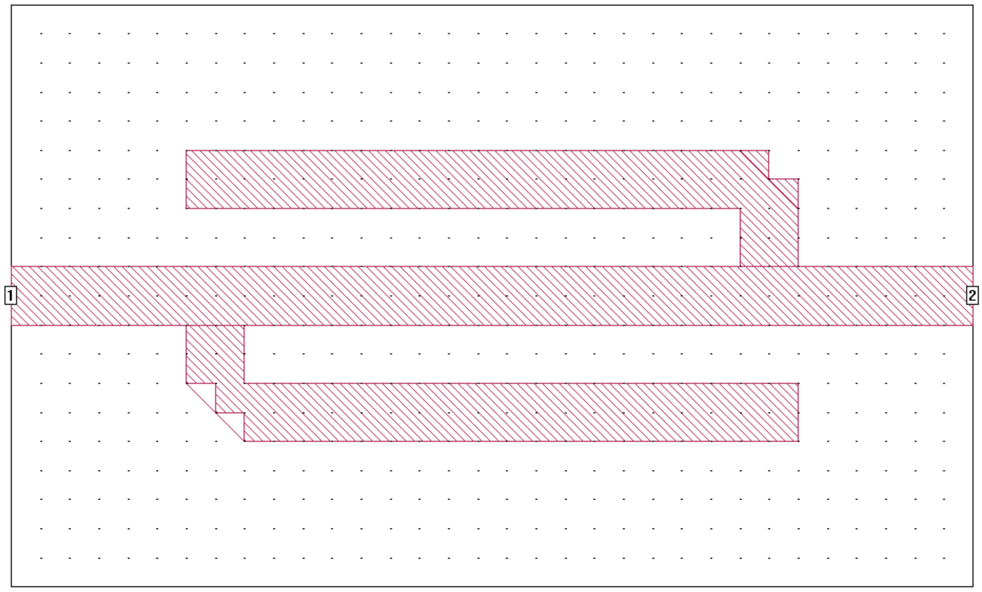
Is used to specify the settings used by the user from the Quick Start guide
QSG
IMPORT NO
EXTRA_METAL NO
UNITS NO
ALIGN NO
REF NO
VIEW_RES YES
METALS YES
USED YES
END QSG

10. SUBDIV.
Specifies the subdividers and resulting filenames for a geometry project.

Subdiv
EndSubdiv

Beginning to Work with Matlab̃

The next figure shows the Sonnet Geometry that we will use for this example.



This geometry represents a Notch Filter.

.
When you create a geometry in Sonnet, it produces a file with *.son extension. For our propose we will create the geometry in Sonnet, we will open it with the Notepad and we will transcribe it to Matlab.
The name of our geometry is *dstub.son* and the resulting file we'll transcribe it to Matlab, pretty similar to what we did with the file of Spice.

The resulting is the next.

```
%Name of the file is dstub.m
%This is the function used to create the Sonnet file
%which will be used to obtain data to process it in Matlab.
%Inside function A is the type of analysis to perform.
%Initfreq is the initial frequency for the analysis desired. The number
%must be given according to the units declared below in the DIM Block.
%Finalfreq is the final frequency for the analysis desired. The units must be
% declared as same that Initfreq.
%Step is used for the SIMPLE analysis and corresponds to the STEP desired
%for the SIMPLE analysis.

function dstub (A, Initfreq, Finalfreq, Step);
Title = '!dstub.son';
% Netlist Block
nb{1} = '!Netlist';
nb{2} = 'FTYP SONPROJ  1 ! Sonnet Project File ';
```

```
nb{3} = 'VER 8.51.lite';
nb{4} = 'HEADER';
nb{5} = 'LIC SL220791.31804' ;
nb{6} = 'DAT 04/25/2003 12:34:07 ';
nb{7} = 'MDATE 04/20/2003 17:35:08 ';
nb{8} = 'HDATE 04/20/2003 17:35:08 ';
nb{9} = 'END HEADER';
nb{10} = 'DIM ';
nb{11} = 'FREQ GHZ ';
nb{12} = 'IND NH';
nb{13} = 'LNG MIL';
nb{14} = 'ANG DEG ';
nb{15} = 'CON /OH';
nb{16} = 'CAP PF ';
nb{17} = 'RES OH ';
nb{18} = 'END DIM';
nb{19} = 'FREQ';
nb{20} = ['SIMPLE  ' num2str(Initfreq) ' ' num2str(Finalfreq) ' ' num2str(Step)];
nb{21} = ['ABS ' num2str(Initfreq) ' ' num2str(Finalfreq)]; %'ABS 4.0 8.0 '
nb{22} = 'END FREQ';
nb{23} = 'CONTROL';
nb{24} = [char(A)];
nb{25} = 'OPTIONS  -d';
nb{26} = 'SPEED 0';
nb{27} = 'END CONTROL';
nb{28} = 'GEO';
nb{29} = 'TMET "Lossless" 0 SUP 0 0 0 0';
nb{30} = 'BMET "Lossless" 0 SUP 0 0 0 0';
nb{31} = 'MET "Aluminum" 1 SEN 0';
nb{32} = 'BOX 1 330 200 66 40 20 0 ';
nb{33} = '      100 1 1 0 0 0 2 "Air" ';
nb{34} = '      20 9.8 1 0 0 0 2 "Alumina" ';
nb{35} = 'POR1 STD';
nb{36} = 'POLY 9 1';
nb{37} = '3';
nb{38} = '1 50 0 0 0 0 100';
nb{39} = 'POR1 STD';
nb{40} = 'POLY 9 1';
nb{41} = '1';
nb{42} = '2 50 0 0 0 330 100';
nb{43} = 'NUM 3';
nb{44} = '0 5 -1 N 9 1 1 100 100 0 0 0 Y';
nb{45} = '0 90';
nb{46} = '330 90';
nb{47} = '330 110';
nb{48} = '0 110';
nb{49} = '0 90';
nb{50} = 'END';
nb{51} = '0 8 -1 N 12 1 1 100 100 0 0 0 Y';
nb{52} = '60 110';
nb{53} = '60 130';
nb{54} = '80 150';
nb{55} = '270 150';
nb{56} = '270 130';
nb{57} = '80 130';
nb{58} = '80 110';
nb{59} = '60 110';
nb{60} = 'END';
nb{61} = '0 8 -1 N 13 1 1 100 100 0 0 0 Y';
nb{62} = '270 90';
```

```
nb{63} = '270 70';
nb{64} = '250 50';
nb{65} = '60 50';
nb{66} = '60 70';
nb{67} = '250 70';
nb{68} = '250 90';
nb{69} = '270 90';
nb{70} = 'END';
nb{71} = 'END GEO';
nb{72} = 'OPT';
nb{73} = 'MAX 100';
nb{74} = 'VARS';
nb{75} = 'END OPT';
nb{76} = 'VARSWP';
nb{77} = 'END VARSWP';
nb{78} = 'FILEOUT';
nb{79} = 'CSV D Y $BASENAME.csv 8 S MA R 50.00000';
nb{80} = 'END FILEOUT';
nb{81} = 'QSG';
nb{82} = 'IMPORT NO';
nb{83} = 'EXTRA_METAL NO';
nb{84} = 'UNITS NO';
nb{85} = 'ALIGN NO';
nb{86} = 'REF NO';
nb{87} = 'VIEW_RES YES';
nb{88} = 'METALS YES';
nb{89} = 'USED NO';
nb{90} = 'END QSG';
nbLines = length(nb);
NetlistBlock = '';
for i = 1:nbLines
   NetlistBlock = str2mat(NetlistBlock, nb{i});
end


% Save Sonnet circuit file
ckt_file = str2mat(Title,NetlistBlock);
[rows,columns]=size(ckt_file);
fp = fopen('dstub1.son','w+');
for i = 1:rows
   for j = 1:columns
      fprintf(fp, '%s', ckt_file(i,j) );
   end
   fprintf(fp,'%s\n','');
end
fclose(fp);

%Time to run Sonnet
%Note: The Sonnet working directory must be where em.exe  and the *.son files are located

!em dstub1.son


%Here is read the Sonnet output file .csv to convert it to .txt using the function that is called
 %with the next line.

   FormatcvsFile('dstub1.csv', 'dstub1.txt')
```

Explanation:

The first part to be considered in this file is that inside of the *FREQ BLOCK* we define variables that will be changed according to the user's decision. This means that we will manipulate the band of analysis.
The first comments in the last file explain what the variables mean.

The second part to consider is that inside of the *CONTROL BLOCK* There is a "*Char (A)*" statement, this helps us to add the **ABS** or **SIMPLE** word to this part, depending of what input is written in the *main file "mainstub.m"* that we will explain later and that is the one that manipulates this statement.

nb{79} = 'CSV D Y $BASENAME.csv 8 S MA R 50.00000' This line will be pretty important to obtain the *.csv file which will contains the data obtained from the Sonnet analysis.

CSV means that we want a data file with .csv extension.
D is that we want to use de-embedding.
Y mention if we want all the discrete data obtained or only with *N* only the points calculated for the analysis mentioned in the *Freq Block.*
$BASENAME.csv Here we say that the file resulting will have the same name that the Sonnet file but with .csv extension.
8 Is the exactitude of the data. This is data with 8 digits or you can change only for **15** digits.
S Means that like result we want S parameters. We can change it to **Y** or **Z** parameters.
MA The response will be given in Magnitude and Angle. We could change to DB (DB and angle) or RI
(Real and Imaginary)
R 50.00000 Means that we have ports of 50 ohms resistance.

*Function that converts the * .csv file to a *.txt file*

We used this function before, in the Spice exercises

```
%This is the Matlab function to convert a *.csv file to an *.txt file
% Format WinSpice Data File
% This function formats an output file generated with Sonnet (using the
% CSV structure in the fileout block), so that it can be read directly by Matlab.
% Usage: FormatcvsFile(file1,file2)
% file1: name of the ASCII file generated by Sonnet using the
% fileout structure with extension .csv (coma separated values)
% file2: name for the ASCII file that can be assigned to a matrix by
% Matlab using the command load.
function FormatcvsFile(file1,file2)
fid1 = fopen(file1);
fid2 = fopen(file2,'w+');
line = fgetl(fid1);        % Drop first line.
line = fgetl(fid1);        % Drop second line.
line = fgetl(fid1);        % Drop third line.
line = fgetl(fid1);        % Drop fourth line.
line = fgetl(fid1);        % Drop fifth line.
line = fgetl(fid1);        % Drop sixth line.
line = fgetl(fid1);        % Read seventh line.
while line~=-1
   line_length = length(line);
   for i = 1:line_length
      if line(i) == ','     % Replace comas by blankspaces.
         line(i) = ' ';
      end
      fprintf(fid2, '%s', line(i));
```

```
  end
  fprintf(fid2,'%s\n','');
  line = fgetl(fid1);          % Read next line.
end
fclose(fid1);
fclose(fid2);
```

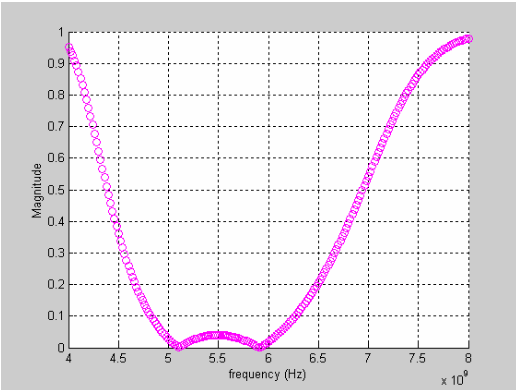
The only difference is that at the beginning we have to eliminate 6 lines before reading the first of our interest.

Main file

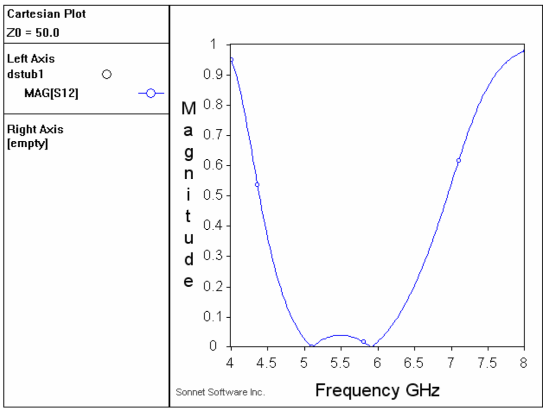
```
%################################################################################
% Fernando Salas
% Industrial Electronics Student
% ITESO
%################################################################################
% This is the main File. Here is where are given the values
% that will be manipulated from the Sonnet file.
% A = is the type of analysis required. For the this exercise  we contemplate
%  SIMPLE or ABS analysis . The word required must be written in between the quotes
% Step is the step size we want to manage in a Simple Analysis.
% Initfreq mentions the initial frequency for the sweep (ABS or SIMPLE).
% And the Finalfreq refers to the final frequency of the sweep (ABS or SIMPLE).
A=['SIMPLE'];
Step = 0.15;
Initfreq = 4.0;
Finalfreq = 8.0;
dstub(A,Initfreq,Finalfreq,Step);  % Here is called the function that creates Sonnet File
load dstub1.txt;
f = dstub1(:,1)';                    % Always the fist column of the .csv file belongs to frequency
Av = dstub1(:,2)';                   %  Here will be taken the S parameter required
plot(f,Av);
xlabel('frequency (Hz)');
ylabel('impedance (Zin)');
grid on
!emgraph dstub1.son                  % Check Note down
 end
```

Note: Here I call the graphic utility of Sonnet to compare the responses plotted in Matlab.

Examples with an ABS Sweep:

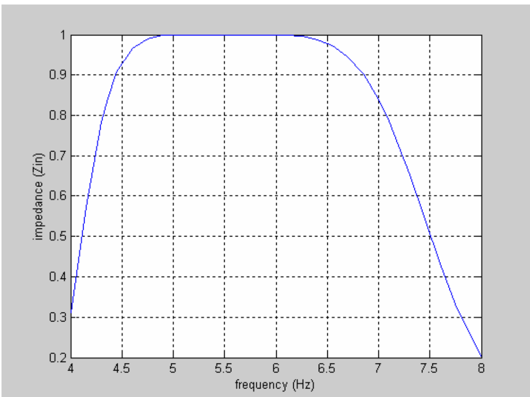


Matlab graphic (S12 Response ABS Sweep)

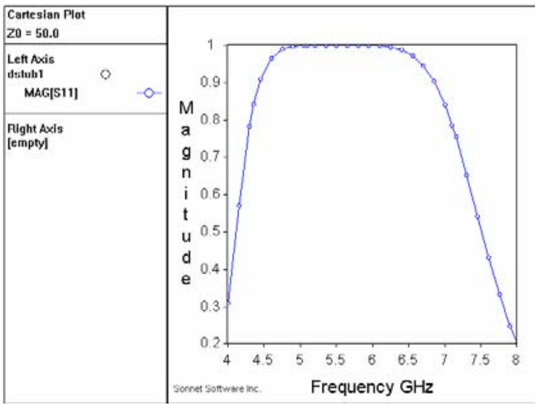


Sonnet Graphic (S12 Response ABS Sweep)

Comparing Results with a SIMPLE Sweep



Matlab graphic (S11 Response SIMPLE Sweep)



Sonnet Graphic (S11 Response SIMPLE Sweep)

Conclusions:

- This is a first step to continue finding the way of manipulating Sonnet from Matlab. A propose is to get until the point of changing geometry sizes and implanting optimizations and more.
- A fast way and pretty good way to obtain fast responses from Matlab, is to work with ABS Sweeps.
- Even though Sonnet is a Software that could take a lot to obtain a response, is an excellent an accurate Software for Electromagnetic simulations.

Bibliography:

- http://www.sonnetusa.com/
- Sonnet Tutorial.
- Sonnet users guide.
- The 8.0 Sonnet Project Format .
- Dr. Ernesto Rayas notes