Focus Microwaves Inc.
277 Lakeshore Road
Pointe-Claire
Quebec, H9S-4L2, Canada
Tel 514-630-6067 Fax 514-630-7466

Application Note 26

Create Your Own Load Pull Tests using MATLAB-TUNE®

MATLAB-TUNE® is a library of functions callable from MATLAB® that permits to create test routines using microwave tuners as fully precalibrated programmable components together with any other GPIB instrument. This note describes how to create a Load Pull and a Power Saturation test procedure..

Introduction

The main difficulty, when creating load pull test procedures using tuners, is not, as some may think, to control the tuners. It is the fact that we must know instantaneously both the impedance presented by the tuner to the DUT **and** the actual loss of tuner, test fixture and setup, up to the power meters. Otherwise we blindly search in the P(Γ,L) space, whereas the tuner impedance Γ and loss L are **both changing** for each tuner position [7].
At this point almost all commercially available load pull (and noise) measu-rement systems provide "closed" software with access to different test routines via a menu. Some softwares permit to include new instrument drivers, or even make small modifications to the procedures, as long as they are written in a simple language, such as BASIC.
Really powerful softwares, however, are written in C language and are practically impossible to understand and modify by non-experts.

Still, there is a need for a software tool that would permit simple operation through a commonly available interface, such as MATLAB [1] or VEE [2], but with full control of both tuner and GPIB operation; "tuner" meaning full knowledge of s-parameter of the tuner box at any impedance setting and full tuning, ie. impedance synthesis capability.

Focus Microwaves supplies such software, both for MATLAB® and HP-VEE®.
This note describes the steps to follow and gives examples for test routines using MATLAB-TUNE.

Summary of MATLAB-TUNE® operation

MATLAB-TUNE is a group of compiled functions that allow control of precalibrated computer controlled microwave tuners and full GPIB communication with programmable instruments inside the MATLAB® DOS or WINDOWS® environment.
These functions can be used inside the MATLAB M-file in order to make full Load Pull type of measurements.
MATLAB® is a mathematical environment which offers easy computing using complex numbers, matrices, signal processing and graphic plots. MATLAB uses a high level programming language thus simplifying the operator's work.
The programmable microwave tuners permit to synthesize impedances. The tuners are pre-calibrated on a microwave network analyzer and the calibration data can be used to generate any required microwave impedance and to compute the actual tuner losses. Such tuners are available from 0.4 to 50GHz in coaxial and 50 to 110GHz in waveguide bands.
The GPIB communication capability permits to send and receive character strings ("mnemonics") to all programmable instruments. The instrument response data can be manipulated (de-embedded) by MATLAB in order to correct them to the DUT reference plane and to generate table, plots and contours.

MATLAB-TUNE includes the following functions in form of DOS executable programmes (..EXE).

- MOVEXY Moves tuner motors absolute or relative
- TUNINI Initializes (returns to zero position) a tuner
- TUNE Synthesizes an impedance (tunes)
- CASCADE Inserts a Twoport between tuner and DUT (ex: Test Fixture)
- WAIT Interrupts programm execution for a given time
- SGPIB Send a character string to a GPIB instrument
- RGPIB Read the GPIB instrument response

Using the MATLAB capability and the above functions in addition to the independent tuner calibration package a variety of automatic test procedures can be developed tailored to each operator's specific needs.
All functions described above can be used in the M-file format of MATLAB to create the final test routines.

In the following sections we will describe a saturation measurement and a load pull measurement procedure developed using MATLAB-TUNE.

Load Pull setup

The measurement routines described in this note have been developed using a conventional load pull setup as shown in figure 1. It uses one programmable signal source, one dual channel power meter, isolator, coupler, bias tee and two programmable tuners, model CCMT-1808 (0.8 to 18 GHz, [3]). The transistor under test (DUT) is mounted in the modular power transistor test fixture PTJ-0 [4]. S-parameter of the test fixture have been determined using TRL [5] and saved in an ASCII file in S2P format.
The complete hardware and software of the setup is controlled by an IBM PC. We used a model 80486-60MHz, but normally any other 386, 486 or Pentium computer can be used. There are no particular memory requirements, except if the Windows version of MATLAB is used, in which case the recommended RAM is 4 to 8MB.

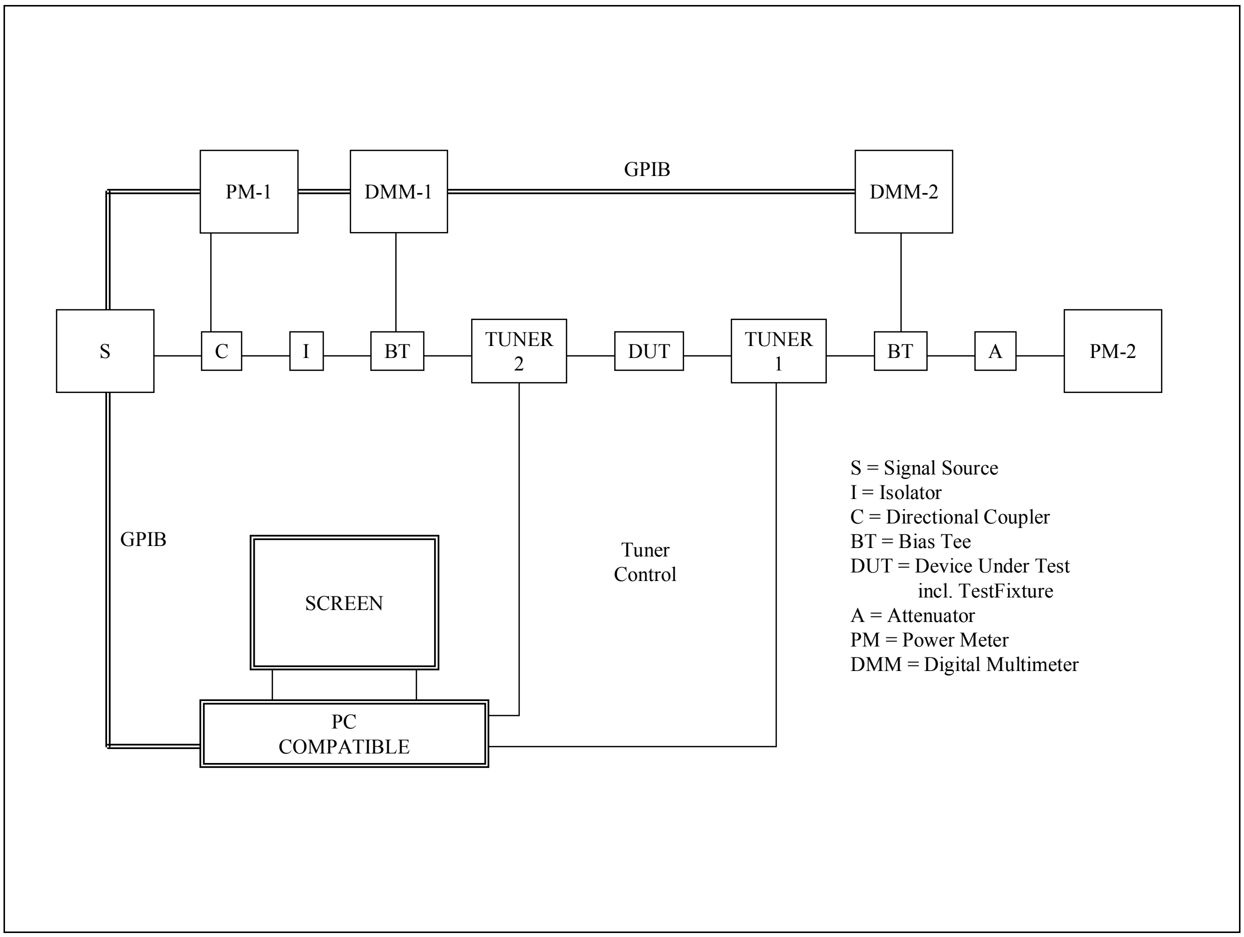
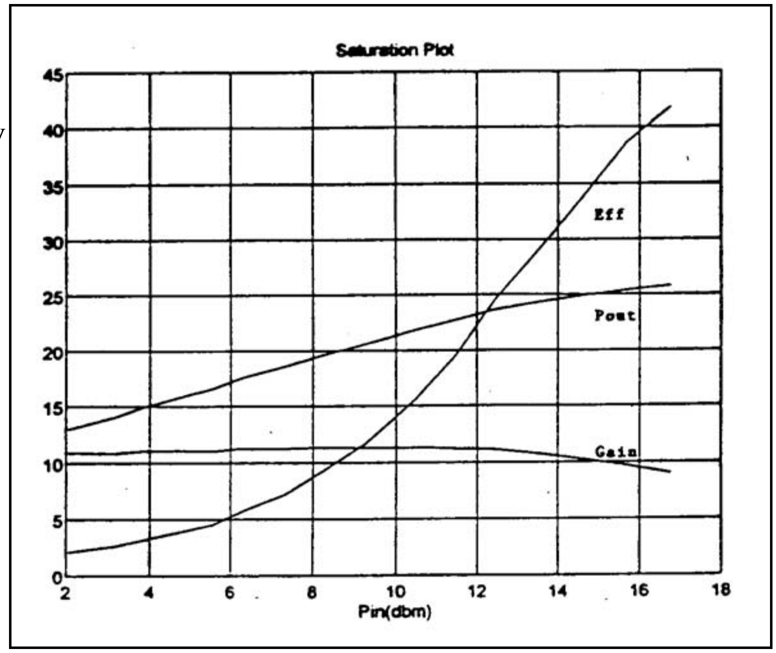


Figure 1: Load Pull setup used in the development of MATLAB-TUNE® measurement routines.

Power Saturation Measurement

In power saturation measurement the tuner is set to some selected impedances, in general such as to generate a maximum gain and output power and the source power is swept from a minimum to a maximum value. During this sweep the input and output power meter are triggered and read. The data are corrected for input and output tuner and setup losses and the final data are plotted over Pin. In addition DC parameters can be measured using another MATLAB compatible routine and Power added Efficiency data can be collected.
The following code includes two additional routines which have also been developed using MATLAB and will be presented in the appendix;



Routine PIPO
This routine measures output power and efficiency as function of input power for a given gamma load.

```
% this routine measures Pout(Pin)
%
echo off
[loss,gamma,phi]=tuneout(0.5,90)          %
                                          tune to gamma load = 0.5/90deg, get loss
setfreq(2)                                % set frequency to 2 GHz [appendix]
for i=1:5
        pis(i)=-30+i*2;                   % increase power from -30 by step of 2dBm
        setpow(pis(i));
        pin(i)=powermeter(1)+30;          % measure input power in dBm, 30dB coupler
        pout(i)=powermeter(2) + loss + 20 % measure output power, include 20dB
                                                  % attenuator [appendix]
        dcp=dcpower;                      % routine measures DC power [appendix]
%
        eff(i)=100*(10^(pout(i)/10 - 10^(pin(i)/10))/dcp;
        disp(sprintf('pi=%4.1f   po=%4.1f   eff=%4.1f', pin(i),pout(i),eff(i))
end
setpow(-50);                              % reset signal source to -50dBm
plot(pi,po,'r'pi,eff,'g')
grid
```

Load Pull Measurements

In load pull measurements the input power and the source tuner are both set to fixed values such as to generate, in general, a maximum output power at a given, mostly 1dB, gain compression. The input power required can be determined from the plot in figure 2. Once these conditions are set then the output tuner is tuned to a number of impedance states (reflection factors) such as to cover either the entire Smith Chart or a section of particular interest. An important reason for selecting a limited section of the Smith Chart to tune to is to avoid parasitic oscillations of the DUT.

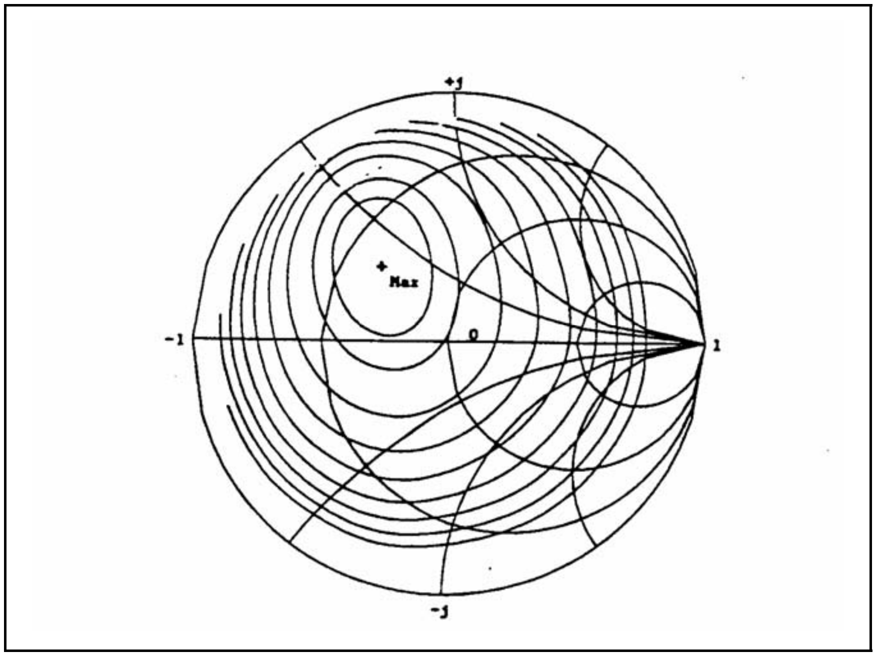
Routine LOADPULL

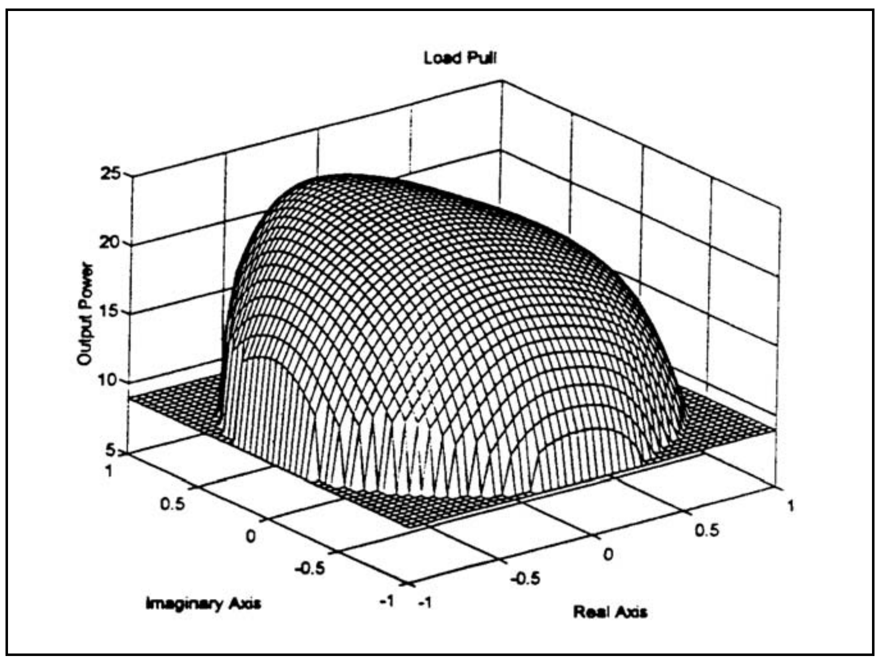
This routine tunes the load impedance to a grid of points on the Smith Chart and measures the output power of the transistor corrected to its reference plane.

```
smith                                   % this MATLAB routine draws a Smith Chart
k=0;
for gamma=0.1: 0.8: 0.1                 % step through Gamma Load
   for phi=0: 315: 45            % step through phase in steps of 45 degr.
        k=k+1;
        polar(phi*pi/180,gamma,'r+');       % target gamma load
        [loss,rga,rph]=tunout(gamma,phi);   % tune
        polar(rph*pi/180,rga,'g+');         % real gamma
        pout=powermeter(2)+loss             % read output power meter, correct loss
        outdata(k,:)=([rga rph pout]);
   end
end
save lp.mat outdata /ascii
```

Figures 3 and 4 show contour and 3D plots made using MATLAB's plot capability from data collected using this measurement routine.

Remark: MATLAB's contouring capability requires data points to be distributed in an equidistant square matrix. This is not necessarily compatible with real load pull, since the tuner does not always synthesize exactly the reflection factor it is sent to, because of limited resolution in the tuner mechanics and in the interpolation routines [6]. It is therefore required to convert a measured file into a MATLAB compatible structure. For this purpose the MATLAB-TUNE library includes software which reads load pull files in ASCII format and generates a square data structure with equidistant points using a simple interpolation technique.This data are used to generate plots as figures 3 and 4.


+j
Max
-1
0
1
-j


Load Pull
Output Power
25
20
15
10
5
1
0.5
0
-0.5
-1
Imaginary Axis
-1
-0.5
0
0.5
1
Real Axis

Conclusion

MATLAB-TUNE® is a selection of executable routines that permit to use precalibrated automatic microwave tuners as fully integrated measurement components in a user defined setup. They also permit to control and read any GPIB programmable instrument from an IBM-PC. MATLAB's® mathematical and graphics capability permits to de-embed to DUT reference plane and to generate saturation and contour plots. MATLAB-TUNE® routines run under DOS 6 and Windows 3.1.

Appendix

This note includes programm examples which use standard MATLAB® functions but also some other functions we developed in order to make the measurement operation possible. Some of these functions use GPIB communication and some just manipulate data and generate plots within MATLAB®. The above examples include the following routines:

- **setfreq(x)** controls via GPIB the signal source and tunes to a frequency
- **setpower(y)** controls via GPIB the source and sets the power
- **powermeter(n)** reads via GPIB the input or output power meter
- **dcpower** measures via GPIB the DC power using the HP-34401 DMM
- **smith** plots a Smith Chart
- **tunout(g,p)** tunes the output tuner to a reflection factor

Detailed syntax of these routines is included in the MATLAB-TUNE® operation manual and can be supplied on request.

Related topics

[1] "MATLAB®, Interactive Scientific and Engineering Software...A preview". The Math Works Inc.

[2] "HP-VEE®, An Iconic Programming Language", Robert Hessel, Hewlett Packard Professional Books, Prentice Hall Inc, 1994.

[3] "0.8 - 18 GHz Programmable Tuner", Data Sheet, Focus Microwaves, 1990.

[4] "Test Fixture for Medium and High Power RF Transistors", Product Note 7, Focus Microwaves, 1994.

[5] "Applying the HP 8510 TRL Calibration for Non-Coaxial Measurements", Product Note 8510-8, Hewlett Packard Co.

[6] "High Resolution Tuners Eliminate Load Pull Performance Errors", Application Note 15, Focus Microwaves, 1995.

[7] "Load Pull Measurements and Amplifier Design", MTTS 1993, Workshop on Design of MMIC Power Amplifiers.