From October 2005 *High Frequency Electronics*

# Using a Signal Analyzer and MATLAB® to Measure ACPR Versus Power

**By Justin Magers**
**Anritsu Company**

Third-party software is a valuable tool for instrument programming and analysis of measurements such as adjacent channel power ratio (ACPR) in wireless transmissiion systems

Using a mathematics software package like MATLAB® to control benchtop lab instrumentation is an easy way to automate otherwise tedious measurements and present the results in a useful format. This article will demonstrate the use of MATLAB on the Anritsu Signature™ (MS2781A) signal analyzer to automate an ACPR (adjacent channel power ratio) versus power measurement of an amplifier. Simple MATLAB examples will be given to show how to control the Anritsu MG3700A Vector Signal Generator, via GPIB and how to control Signature using Web Services.

ACPR (or ACLR—adjacent channel level ratio as it is called in UMTS WCDMA specification [1]) measures the amount of power in an adjacent channel due to intermodulation distortion generated by a digitally modulated signal in the main channel passing through a non-linear device such as an amplifier. Additionally, noise and phase noise also contribute to ACPR.

Unfortunately, the ACPR of an amplifier is a function of the amplifier's output power. This is important since many modern communication systems utilize adaptive power control. In the UMTS system, for example, the base station transmit power in each channel is varied "in order to minimize interference with other channels as well as provide sufficient power to mobiles that are far from the base station" [2]. Therefore, being able to easily characterize the ACPR versus power of an amplifier is necessary to insure that both ACPR and output power specifications are met.

Often, engineers in an R&D environment will not automate measurements because of the many obstacles, such as installing a GPIB card into a computer, installing Visual Basic (VB), and writing VB code to communicate over GPIB with two or more instruments. Finally, even if this is done, the engineer must still find a nice way to display the measured data, something which most programming languages, such as VB, are not well-suited for without significant programming effort.

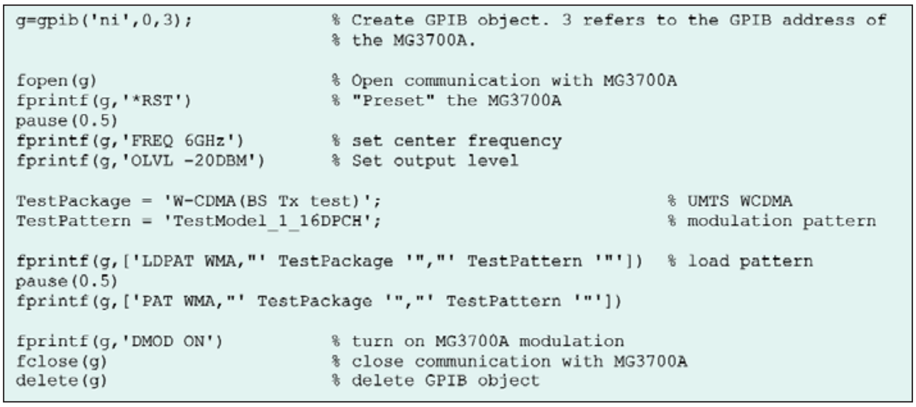
Signature simplifies these tasks considerably by seamlessly integrating MATLAB (option 40) into the instrument. With only a few short commands, MATLAB can control external instruments via GPIB, and it can control the spectrum analyzer via Web Services. Furthermore, MATLAB excels at making nice graphs for data output without sophisticated programming. Finally, distributing the data is much easier since Signature uses Windows XP. Essentially, any printer can be connected via the parallel port or the USB port. Or, if Signature is plugged into the Internet, the results can be emailed or shared on the file system.

## Using MATLAB to Control Instruments via GPIB & Web Services

It should be emphasized that, in this example, MATLAB is installed on Signature and that the Anritsu MG3700A is connected to Signature via a standard GPIB cable. This configuration uses minimal hardware as opposed to other configurations using an external computer.

```
url = 'http://localhost/SignatureSpectrum/SignatureSpectrum.asmx?wsdl';
createClassFromWsdl(url);
spa = SignatureSpectrum;
SetFrequencySpan(spa,30,'MHz');
f = str2double(GetCenterFrequencyInHz(spa))
```

Example 1 · Using MATLAB to communicate with MS2781A using Web Services.



```
g=gpib('ni',0,3);                 % Create GPIB object. 3 refers to the GPIB address of
                                  % the MG3700A.

fopen(g)                          % Open communication with MG3700A
fprintf(g,'*RST')                 % "Preset" the MG3700A
pause(0.5)
fprintf(g,'FREQ 6GHz')            % set center frequency
fprintf(g,'OLVL -20DBM')          % Set output level

TestPackage = 'W-CDMA(BS Tx test)';                              % UMTS WCDMA
TestPattern = 'TestModel_1_16DPCH';                              % modulation pattern

fprintf(g,['LDPAT WMA,"' TestPackage '","' TestPattern '"'])  % load pattern
pause(0.5)
fprintf(g,['PAT WMA,"' TestPackage '","' TestPattern '"'])

fprintf(g,'DMOD ON')              % turn on MG3700A modulation
fclose(g)                         % close communication with MG3700A
delete(g)                         % delete GPIB object
```

Example 2 · Using MATLAB to communicate with an instrument using GPIB.

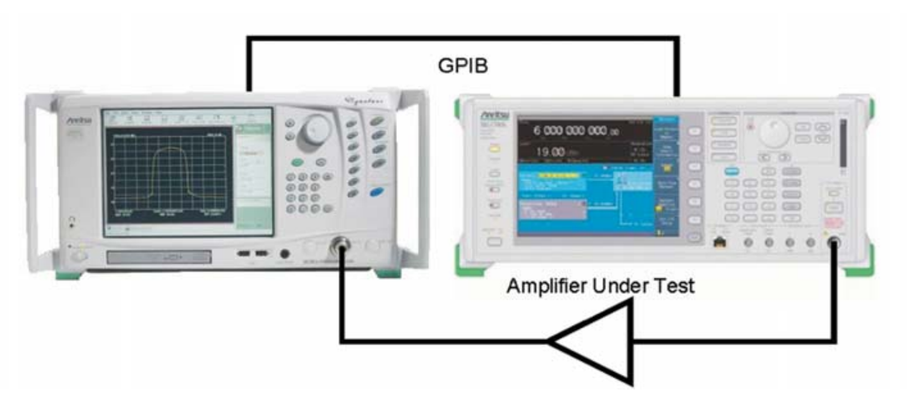


Figure 1 · ACPR Measurement Setup.

MATLAB can be started from Signature's GUI by going to the Tools pull-down menu and selecting MATLAB. In the MATLAB Setup dialogue, next to "Connect to MATLAB," click "On" and close the dialogue. Before using Signature to control other instruments via GPIB, it is also necessary to make Signature the GPIB system controller. To do this, hit the System button on the front panel. From the System menu, select the Configuration sub-menu and then hit the IO Button. Selecting GPIB will start the Measurement & Automation Explorer (MAE). In the MAE Configuration menu under My System/Devices & Interfaces, right-click on GPIB0 (PMC-GPIB) and select Properties. Check the "System Controller" checkbox, select OK, and hit Exit from the File pull-down menu. Signature will indicate that it is the system controller in the bottom right hand corner of the screen.

To become familiar with controlling Signature using Web Services and controlling other instruments using GPIB, consider the following examples. Example 1 sets up the Web Services interface with the instrument and then uses Web Services to set the frequency span to 30 MHz, query the center frequency in Hz, and store the value in the variable *f*. The MATLAB function str2double() is required because the GetCenterFrequencyInHz() function returns the frequency as a character array.

Example 2 presets the MG3700A vector signal generator, sets the frequency to 6 GHz, sets the power to –20 dBm, loads the modulation signal, and turns on the modulation. Test Model 1 is the modulation signal described by the 3GPP committee for ACLR measurements [1].

Of course, there are hundreds of other Web Services and GPIB commands which let the MATLAB user control nearly all aspects of the instruments. These commands can be found in the manuals for each instrument [3,4].

The previous two examples demonstrated the use of Web Services and GPIB in MATLAB. In Example 3, both will be used at the same time. Specifically, a for loop will be used to sweep the power of the MG3700A (via GPIB) from –24 dBm to –10 dBm in 1 dB steps. Within the for loop, Web Services will be used to query Signature and capture the adjacent channel power. Finally, a plot of the left and right ACPR in dB versus input iower in dBm will be generated.

### Measurements

Figure 2 shows the results of the measurement setup in Figure 1. The DUT is the Mini-Circuits ZEL-1724LN packaged amplifier, which has the following specifications: gain = 20 dB, $P_{1dB}$ = 10 dBm, and Output IP3 = +22 dBm. The MATLAB script ACPRvsP.m used to automate this measurement is slightly more sophisticated when compared to Example 3 in the following ways:

1. It uses several Web Services commands to set up the spectrum ana-

lyzer's channel bandwidth, channel spacing, symbol rate, RBW, etc.
2. It adds more annotation to the output graph
3. It records the main channel output power and uses it as the x-axis.

The rest of the code is nearly identical to the examples given above. The ACPRvsP.m script is included with the Signature MATLAB Connectivity option (option 40), release 2.0 or later.

MATLAB with Web Services and GPIB in conjunction with Anritsu Signature provide a convenient environment for automating measurements. MATLAB's data processing and graphics presentation features enable engineers to easily manipulate and present measured data. The synergy of these elements allow unlimited measurement possibilities without the usual complexity of modern automated measurement systems.

## References

1. 3GPP TS 25.141 version 6.6.0 Release 6, Technical Specification, 3rd Generation Partnership Project, Technical Specification Group Radio Access Network, Base Station (BS) conformance testing (FDD) (Release 6).

2. Measuring UMTS ACLR with the Anritsu Signature, Anritsu Company.

3. Series MS278XA High Performance Signal Analyzer Programming Manual, Anritsu Company.

4. MG3700A Vector Signal Generator Operation Manual (Mainframe), First Edition, Nov 2004, Anritsu Corporation.

Note: MATLAB is a registered trademark of The Mathworks, Inc.

## Author Information

Justin Magers received his BSEE from the University of California at Davis in 2000 and his MSEE from Santa Clara University in 2003. He presently designs mixers as a microwave design engineer at Anritsu Company in Morgan Hill, CA. Additional product and applications information is available online at: www.us.anritsu.com.

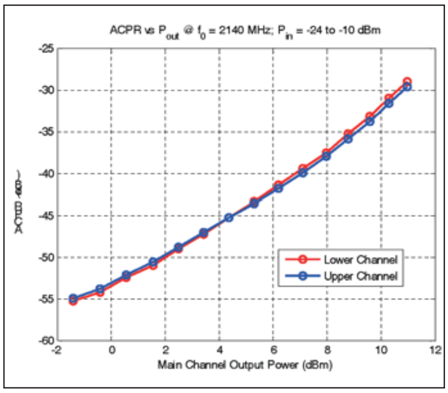


Figure 2 · ACPR vs. output power plot of a Mini-Circuits ZEL-1724LN generated by ACPRvsP.m.

```
% setup a web services object (named sys) in MATLAB
url = 'http://localhost/SignatureSystemControl/SignatureSystemControl.asmx?wsdl';
createClassFromWsdl(url);
sys = SignatureSystem;
Preset(sys);                          % "preset" Signature

url = 'http://localhost/SignatureSpectrum/SignatureSpectrum.asmx?wsdl';
createClassFromWsdl(url);
spa = SignatureSpectrum;
SetSweepMode(spa,'Single');           % put Signature into single sweep mode
SetCenterFrequency(spa,2140,'MHz')    % set Signature center frequency
SetFrequencySpan(spa,30,'MHz')        % set Signature frequency span

g=gpib('ni',0,3);
fopen(g);

TestPackage = 'W-CDMA(BS Tx test)';   % UMTS WCDMA modulation pattern
TestPattern = 'TestModel_1_16DPCH';

fprintf(g,['LDPAT WMA,"' TestPackage '","' TestPattern '"'])  % load pattern
pause(0.5)
fprintf(g,['PAT WMA,"' TestPackage '","' TestPattern '"'])
fprintf(g,'FREQ 2140MHz')
fprintf(g,'DMOD ON')                  % turn on MG3700A modulation

pwr = -24:1:-10;                      % sweep power from -24 to -10 dBm in 1 dB steps

for i = 1:size(pwr,2),

    % Set MG3700A output level using GPIB
    fprintf(g,[ 'OLVL ' num2str(pwr(i)) ' DBM'])

    pause(1)          % pause to allow MG3700A to change power level
    StartSweep(spa)   % start a single sweep
    pause(1)          % pause 1 second for sweep to finish before collecting data
                      % this can be shorter or longer depending on the sweep time

    % query MS2781A using Web Services for the left and right ACPRs.
    % Store results in the vectors acp_r and acp_l
    [valid,acpright,units,acpleft,units] = GetACPRAdjacentChannelResults(spa);

    acp_r(i) = str2double(acpright);
    acp_l(i) = str2double(acpleft);

end

fclose(g)
delete(g)

% plot the data, add title, label axes, and add a legend
plot(pwr,acp_l,pwr,acp_r)
title('ACPR vs DUT Input Power');
xlabel('DUT Input Power (dBm)')
ylabel('ACPR (dB)')
legend('Left ACPR','Right ACPR');
```

Example 3 · A simple MATLAB script to automated ACPR versus Power.