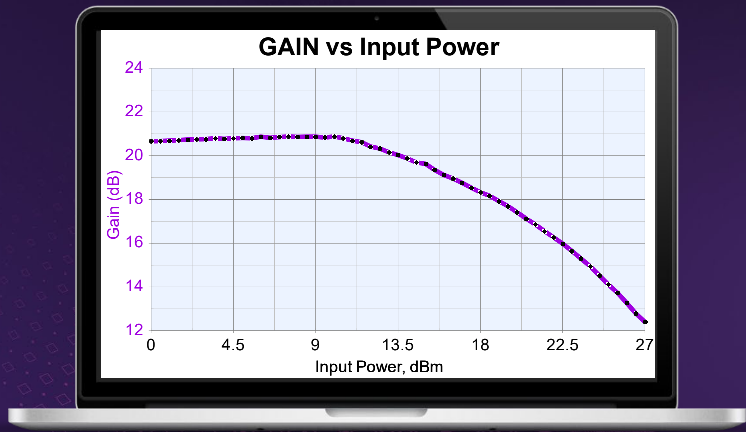


TECHNICAL OVERVIEW

PATHWAVE

Modeling AM-to-AM Behavior

PathWave RF Synthesis (Genesys) & PathWave System Design (SystemVue)

A significant enhancement has come to the RFAMP model in RF System— you can now specify a **custom gain compression curve.** The model senses its input power no matter where it is in a chain and applies the correct gain and compression to itself, according to the user-defined curve.

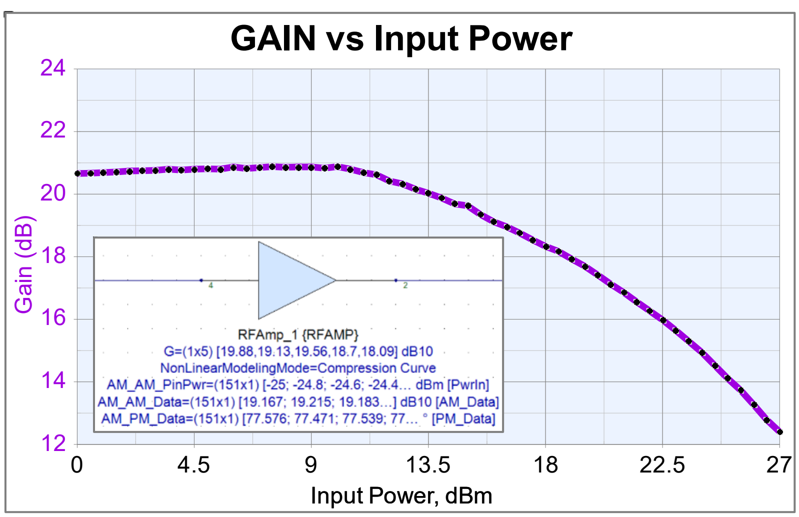


Figure 1. Any custom gain compression curve can be modeled.

This is all made possible by new RFAMP parameters **AM_AM_Data** and **AM_AM_PinPwr,** by which you supply a vector of gain vs. input power.
Any curve can be defined: Gain expansion, glitches, ripple, non-standard rolloffs, or anything else are possible.

KEYSIGHT TECHNOLOGIES

As you acquire AM-to-AM (and corresponding AM-to-PM) data for your device model, keep the following guidelines in mind:

- Data can be entered either as absolute or relative (delta) gain values. Absolute gain is converted (silently) to relative gain inside the model, using the first point as the nominal small-signal gain.
- Include data points from about 30 dB below to about 3 dB above P1dB.
- Use small power steps: 0.2 dB steps give higher accuracy than 0.5 dB steps.
- Employ techniques (e.g., IF bandwidth, averaging) to reduce noise in the data.
- Absolute gain data is preferred over relative because then both OP1dB and OPSat can be extracted automatically from the curve. If relative data are used, only IP1dB and IPSat are extracted, which may not provide all the behavior you need.
- In any case, 200 points are more than enough for good accuracy.

A breakthrough feature of the new model is the ability to extract P1dB and the odd-order intercepts like IP3, IP5, IP7, etc., from the gain compression curve. This technology was developed for Keysight vector network analyzers measuring device gain compression.

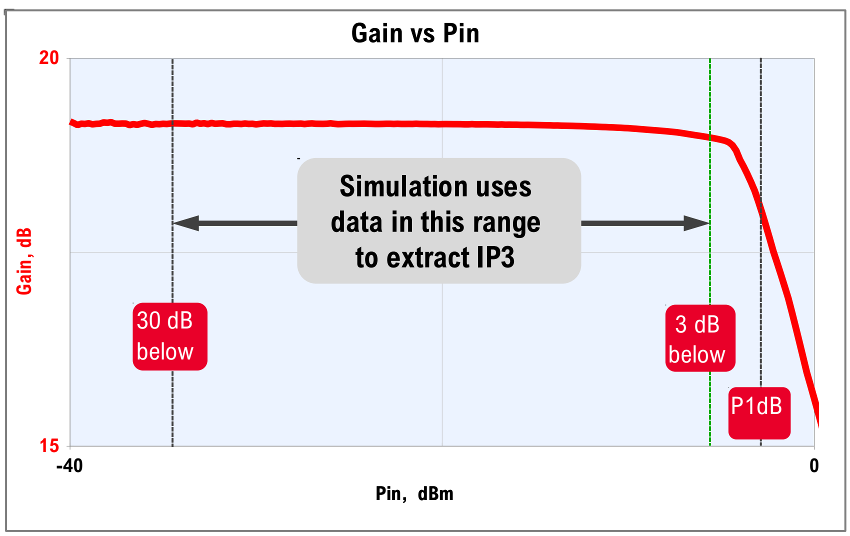


Figure 2. For best IP3 extraction accuracy, a portion of the entire curve is used.

There is a “sweet spot” of the gain compression curve that results in the highest accuracy of extracted IP3. The device at left was swept from a low level to a few dB above P1dB. Even though the full curve is available for simulations, only a subset is used to extract IP3— from about 30 dB below to about 3 dB below P1dB. Any noise at the lowest levels is avoided, yet enough of the curve is retained for the extraction algorithm to perform well.

A Modeling Example

```
!MDF A.01.01
!Keysight Technologies
!Mini-Circuits ZJL-6G Amplifier, 1 GHz Power Sweep,
!Gain = 13 dB, OP1dB = +15 dBm, CW Freq: 1000000000 Hz
BEGIN CH1_DATA
% Power(real) S21LogMag(real) S21Phase(real)
-25  14.274675  72.339058
-24  14.284138  71.801453
-23  14.275489  72.086693
.
.
.
+3   10.170382  66.214767
+4   9.3716393  64.539444
+5   8.4133139  62.742283
END
```

File → Import → ADS Files → System MDIF File…

Figure 3. Import an MDIF file containing gain and phase vs. input power.

Begin by measuring a device with a vector network analyzer to obtain an MDIF file containing swept gain (and phase) vs. Pin. Absolute gain/phase format, in dB/degrees, is shown here (real/imaginary format is also accepted). For brevity, short vector lengths are shown above but you should use more points as previously described to improve the accuracy of the extract parameters: P1dB, Psat, & odd-order intercepts. Since Pathwave already understands this file format, import the file and then prepare the MDIF data for use with 3 simple lines of MATLAB Script on an Equation page:

```
using('ImportedFileName.mdf');
% For data in dB / degrees format
PwrIn = Power;              % This is swept input power which is the independent
                            % variable for the AM and PM data. The 'AM_AM_PinPwr'
                            % parameter unit is set to 'dBm'.
AM_Data = S21LogMag;        % The 'AM_AM_Data' unit is set to 'dB' to match the
                            % S21LogMag format.
PM_Data = S21Phase;         % The 'AM_PM_Data' unit is set to 'deg' as S21Phase is in
                            % degrees.
Gain = AM_Data(1);          % Extract the first AM_Data entry to be used for the Gain
                            % if desired.

% OR, if the data are in real / imaginary format
PwrIn = Power;
AM_Data = 20*log10( abs( S2_1 ) );       % Convert to dB.
PM_Data = angle( S2_1 );                 % Radians or degrees.
```

Figure 4. Prepare the data with just three lines of MATLAB Script on an Equation page.

Instead of measuring a device, you can paste data from other sources, such as a datasheet, directly onto an Equation page. Here is an example showing AM-to-AM data in relative gain format. The numbers indicate delta gain in dB relative to the first point. Absolute gain format is also accepted.

```
AM_Data = [ 0.0000, -0.0572, -0.0220, -0.0295, -0.0018,...
-0.0239, -0.0330, -0.0522, -0.0204, -0.0128, -0.0337,...
-0.0387, -0.0511, -0.0596, -0.0658, -0.0873, -0.0997,...
-0.1161, -0.1519, -0.1954, -0.2516, -0.3285, -0.4769,...
-0.7063, -1.1158, -1.6663, -2.2838, -2.9593, -3.6728,...
-4.4993, -5.3515 ];
```

No matter which method is used, the model setup is seen in the RFAMP's Properties dialog below.

- First, set **NonLinearModelingMode** to **1:Compression Curve.**
- Second, supply vectors (arrays) for **AM_AM_PinPwr** and **AM_AM_Data,** either as absolute or relative values. **AM_PM_Data** can also be entered, in either degrees or radians.
- Third, enter a value for **OIP2** as desired. And note that with **AM_AM_OverrideOddIPn = OFF,** the model will self-extract its IP3.

Finally, since **AM_AM_Data** is always converted to relative gain internally, you must enter **G.** The model always looks to **G** for its small-signal gain. This also allows **G** to be specified as a Sys-parameter (versus frequency) if desired. However, since the model's OP1dB and OPSat are derived from **AM_AM_Data,** only absolute gain will give accurate values for these parameters. Relative gain will not. This is why the absolute gain format is recommended for all devices.
If your device only uses input-referenced parameters like IP1dB and IPSat, then the relative format is acceptable.

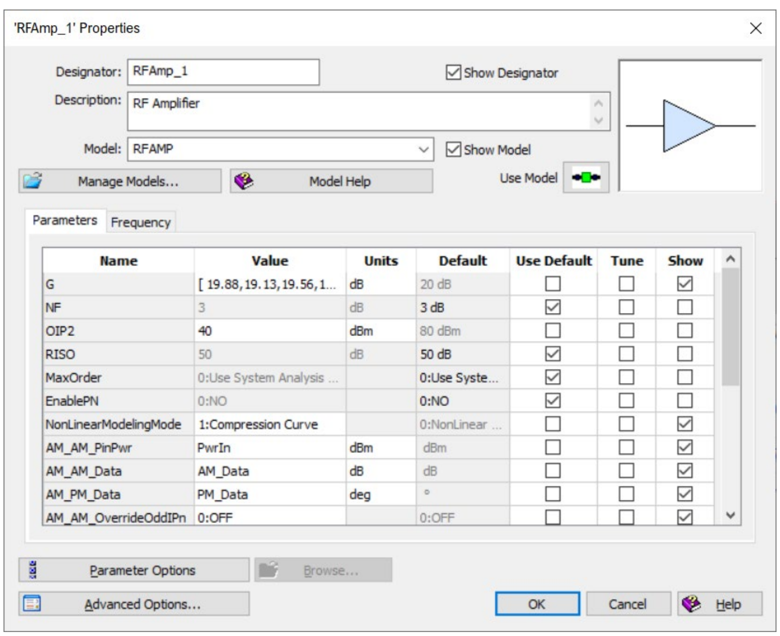
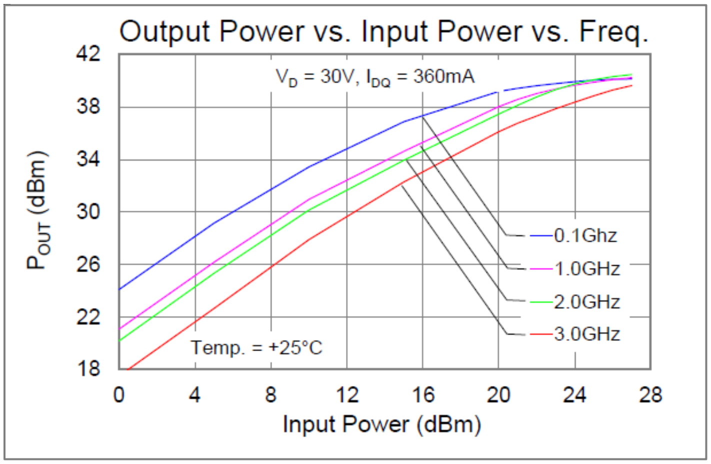


Figure 5. RFAMP Properties.

If the parameter **AM_AM_OverrideOddIPn** is set to **ON**, the **OIP3** parameter appears so you can set it as desired, and IP3 will no longer be auto-extracted from the gain curve. Do this when the accuracy of the extracted IP3 is not acceptable, either because of too few points, insufficient data range, or any other reason. Notice in this screenshot the gain **G** is specified as a Sys-parameter. This is permitted.

If the device has a different gain compression characteristic at different frequencies, as seen in the following plot, the MATLAB Script technique shown below would put the behavior described by this graph into the model, and only requires a few lines of code:



TGA2237 datasheet ©2020 Qorvo US, Inc.

```
% ************************************************
%      TECHNIQUE FOR INTERPOLATING A NEW GAIN
%      COMPRESSION CURVE AT ANY RF FREQUENCY.
% ************************************************

% First, combine your measured curves into one large 2-D array. Each row is a
% different frequency; each column is a different input power.
Gains = [ GC_100_gains; GC_1000_gains; GC_2000_gains; GC_3000_gains ];

% Attach an indep, F, so we know the frequency of each curve. Here F is in MHz.
F = [ 100, 1000, 2000, 3000 ];
setindep('Gains', 'F');

% Last, use interp1() to create a whole new curve at the simulation frequency,
% RFfreq. This result is entered into the AM_AM_Data parameter.
GC_curve = interp1(F, Gains, RFfreq, 'linear');
```

Figure 6. Technique for implementing multiple gain compression curves.

Complete documentation of this new AM-to-AM capability can be found in the Help at: Users Guide → Simulation → RF Design Kit – Spectrasys and WhatIF → Spectrasys → Advanced Spectrasys → AM to AM and AM to PM → **AM to AM.**

With this new functionality, many different types of devices with non-standard gain vs. input level characteristics can now be modeled by RFAMP in its new AM-to-AM mode. The ability to set-and-forget a custom transfer function brings lots of flexibility— and realism— to simulations.

Glossary

AM-to-AM – the nonlinearity described by a gain compression curve, or Pout vs. Pin curve, where the device amplitude transfer function at higher input levels deviates from the corresponding function at low levels.

AM-to-PM – the nonlinearity in the phase transfer function at higher input levels deviates from the corresponding function at low levels.

Gain Expansion – a characteristic of some amplifiers whose gain rises before falling in compression.

IP3 – third-order intercept.

MATLAB Script – the language used by Pathwave System for calculations and post-processing.

MDIF – Measurement Data Interchange Format, an industry-wide format frequently used for simulation model data and that most test equipment (network analyzer, vector signal analyzer, etc.) uses to store and share measured data.

P1dB – 1 dB compression point.

Psat – Saturation power, typically the maximum power an amplifier can produce.

Sys-parameter – a parameter of a behavioral model that allows a user to specify a device characteristic (gain, noise figure, third-order intercept, impedance, etc.) versus frequency.