ADS Fundamentals – 2009

LAB 7: Harmonic Balance Simulations

Overview - This exercise continues the amp_1900 design and shows the fundamentals of using the Harmonic Balance simulator to look at the spectrum, analyze compression, calculate TOI, and perform other non-linear measurements.

OBJECTIVES

- Set up and perform a 1 tone HB simulation.
- Set up and perform a 2 tone HB simulation.
- Use variables for simulation and source control.
- Test Gain, Compression, Available Power, Noise Figure, IP3, and other specifications.
- Use the *ts* transform on HB data.
- Work with equations, plots, and the Mix table.

Agilent Technologies

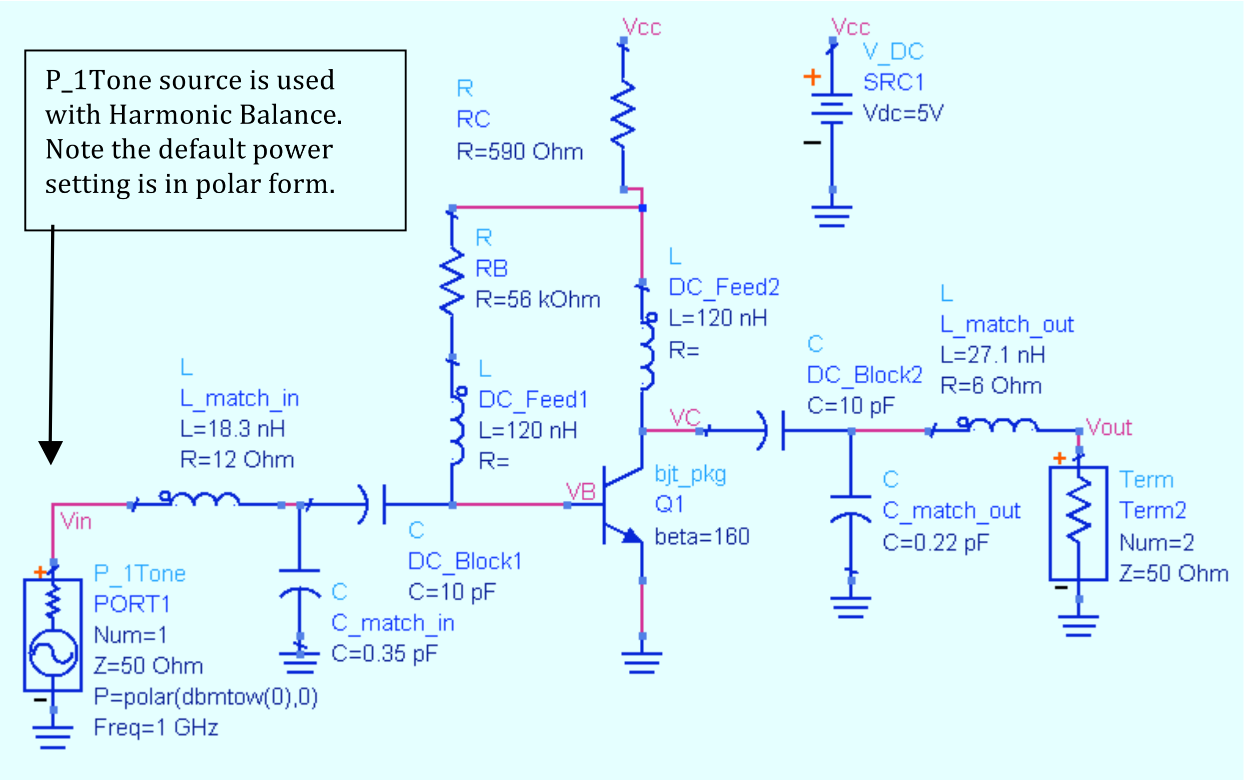
TABLE OF CONTENTS

PROCEDURE

1. Set up the circuit with a P_1Tone source.

a. Close the system_prj if it is still opened. Then open the **amp_1900** project and schematic: **s_final**.

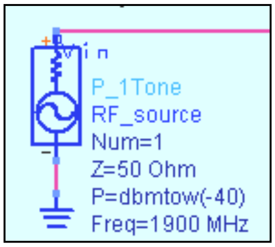
b. Save the s_final schematic with a new name: **hb_basic**. Then delete all the simulation and measurement components and the input Term. Begin building the setup shown here.


P_1Tone source is used with Harmonic Balance. Note the default power setting is in polar form.

c. Insert a **P_1Tone** (Sources-Freq Domain palette) for the RF input.

d. Insert 4 pin labels (node names) **Vin**, **Vout**. **VC** and **VB** as shown so that the voltages will be available in the dataset.

e. Set the RF source as shown: Freq=1900 MH. Also, remove remove the polar function so that only the dbm-to-watts watts function remains: **P=dbmtow (-40)**. Also, rename rename the source **RF_source**. The port number is defined defined by Num=1.



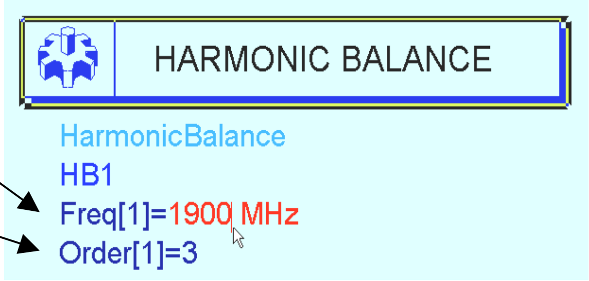
2. Set up a one-tone Harmonic Balance simulation.

a. Go to the **Simulation_HB** palette and insert a **Harmonic Balance** simulation controller as shown here.

b. Edit the Freq setting on the screen: change it to **Freq [1] = 1900 MHz** so that it matches the Freq setting in the P_1Tone source.

Freq[1]must be set to match the source frequency.

Order is the number of harmonics.

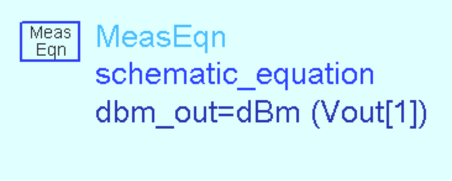


HarmonicBalance
HB1
Freq[1]=1900 MHz
Order[1]=3

3. Write a measurement equation for <u>dBm of Vout</u> and simulate.

a. From the simulation palette, insert a **measurement equation**.

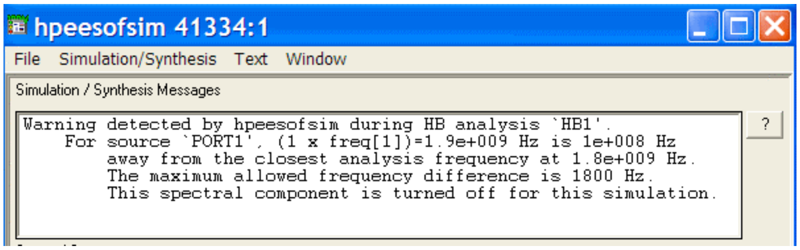
b. Write an equation to calculate the output power at Vout in dBm: **dbm_out dbm_out = dBm (Vout[1])**. The number in braces [1] refers to the index value of the calculated frequencies in the analysis. With Order = 3, the index values are: index [0] is the DC component, index [1] is 1900 MHz, index [2] is the second harmonic or 3800 MHz, and index [3] is the third harmonic. Therefore, the equation should produce the output power in dBm for 1900 MHz only.



MeasEqn
schematic_equation
dbm_out=dBm (Vout[1])

c. **Simulate** – you should have no warnings or error messages.

d. Change the HB controller to: **Freq[1]= 1800 MHz**. Now, **simulate** again and read the error message - the source is 100 MHz away from the HB frequency of 1800 MHz. This is a common error when the source and controller do not agree.



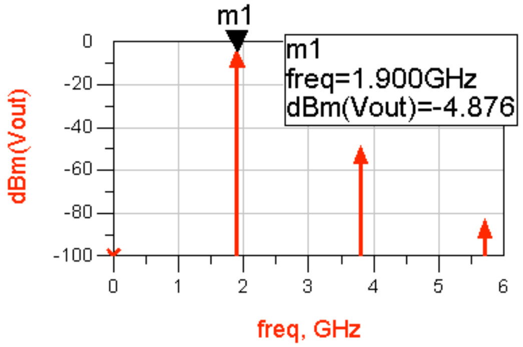
```
Warning detected by hpeesofsim during HB analysis `HB1'.
    For source `PORT1', (1 x freq[1])=1.9e+009 Hz is 1e+008 Hz
        away from the closest analysis frequency at 1.8e+009 Hz.
        The maximum allowed frequency difference is 1800 Hz.
        This spectral component is turned off for this simulation.
```

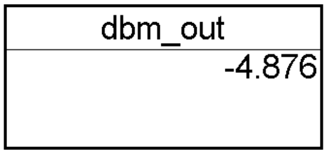
e. Reset the HB controller **Freq[1] = 1900 MHz** and **simulate** again.

4. Plot the spectrum, equation, and *ts* of the node voltages.

a. In the data display, plot **dBm** of **Vout**. Also, insert a list of **dbm_out**. Whenever you write a measurement equation, it will appear in the dataset. The two values should be the same as shown here.

b. Put a **marker** on the fundamental and verify that your amplifier has about 35 dB of Gain with output power in dBm = -4.876 at 1900 MHz.

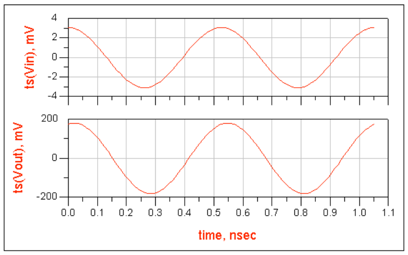




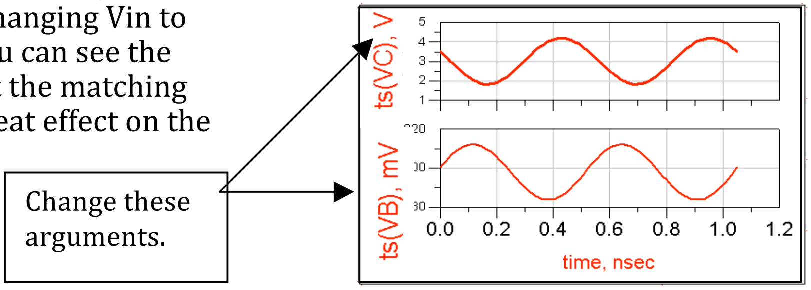
dbm_out
-4.876

NOTE on results: With Order set to 3 in the simulation controller, you get 3 tones: fundamental plus two harmonics. The DC component also shows up on the plot because Harmonic Balance always computes DC for convergence.

c. Insert a **stacked rectangular plot** and insert two data traces as **time domain signals**: **Vin** and **Vout**. The *ts* (time series) function operates on HB and transforms it into the time domain. In this case, you can see that the amplifier does not invert the signal as you might expect. These will be two separate plots in one frame. Put markers on the same time point as shown.



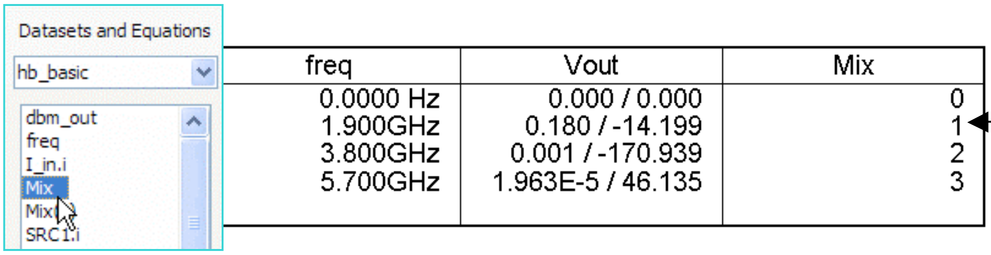
d. Edit the Y-axis label on the trace by changing Vout to **VC** and changing Vin to **VB** as shown here. Now you can see the inversion. This means that the matching network probably has a great effect on the phase.



Change these arguments.

5. Operate on Vout and Mix using functions and indexing.

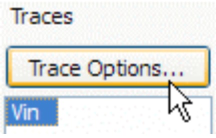
a. Insert a list of **Mix** and **Vout** as shown here. Whenever a HB simulation is performed, a Mix table (index values) is created in the dataset. Notice that Vout is always complex (mag and angle), unless you operate on it using dB, dBm, etc. In the next steps, you will learn how to write equations to display or operate on specific tones in the Mix table. This is especially useful for multiple tones or mixers.

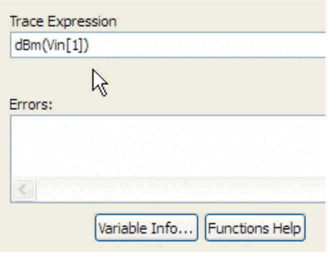


freq	Vout	Mix
0.0000 Hz	0.000 / 0.000	0
1.900GHz	0.180 / -14.199	1
3.800GHz	0.001 / -170.939	2
5.700GHz	1.963E-5 / 46.135	3

Index value of 1900 MHz = 1. The DC index = 0. .HB.

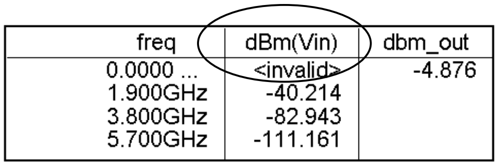
b. Edit the first list and **add Vin**. Then select **Trace Options** and and edit **Vin** by typing in the **dBm** function: **dBm(Vin)** and click and click **OK**. Notice that whenever you edit a trace or insert an insert an equation the buttons appear for Variable Info (dependencies) or Function Help (manuals).



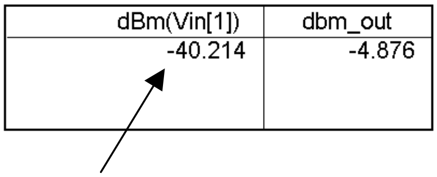

Trace Expression: dBm(Vin[1])

c. Your list should now contain the schematic equation **dbm_out** and the expression **dBm (Vin)** for all frequencies. frequencies. Now, edit the dBm(Vin) data by inserting the inserting the index value **[1]** in the Vin argument as shown – shown – now you get the value of Vin at the index value or value or 1900 MHz.

d. Insert the **[1]** in the dbm_out equation - it becomes invalid because it was indexed as {1} on the schematic.

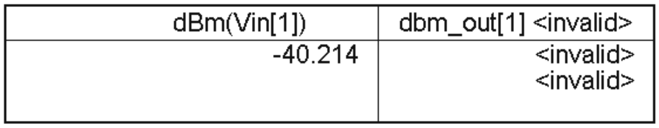


freq	dBm(Vin)	dbm_out
0.0000 ...	<invalid>	-4.876
1.900GHz	-40.214	
3.800GHz	-82.943	
5.700GHz	-111.161	



dBm(Vin[1])	dbm_out
-40.214	-4.876

Adding [1] to the Vin data returns the 2nd index value = 1900 MHz.

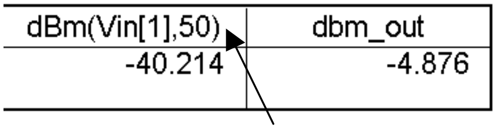


dBm(Vin[1])	dbm_out[1] <invalid>
-40.214	<invalid>
	<invalid>

Adding [1] to the measurement equation makes it invalid.

e. Remove **[1]** from the invalid dbm_out equation to make it valid again.

f. Insert the cursor in the dBm (Vin[1]) expression and add a **comma** and **50** as shown here. The second argument in the dBm function is Zin. If no argument is given, the default is 50 ohms. Therefore, no change should occur. **Undo** the comma fifty (,50) so that it reads **dBm(Vin[1]** again.

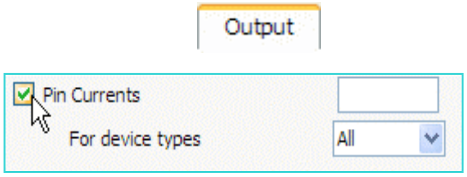


dBm(Vin[1],50)	dbm_out
-40.214	-4.876

Insert Zin: 50 as the 2nd argument separated by a comma.

NOTE on dBm function and Zin of your designs - The dBm function converts a voltage into dBm assuming an exact 50 ohm impedance. However, if Zin is not exactly 50 ohms +/- j0, then the power at Vin may be incorrect. Therefore, you may want to use the correct value of Zin as you will see in the next step.

6. Calculate Delivered Power and Zin using Pin Current

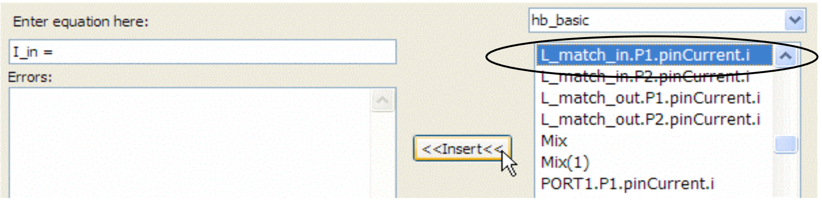


☑ Pin Currents

For device types All

a. Edit the Harmonic Balance controller and select the select the Output tab and check the box for Pin Pin Currents (shown here) and click OK. This will This will add all the values of current to the data set data set – you will be using the input pin current instead of a current probe.

b. **Simulate** again. When finished, go to the same Data Display and write an **equation** for the input current which uses the total current through the input inductor. To do this, use the editor: write **I_in =** and then insert the inductor current as shown here:



Enter equation here:

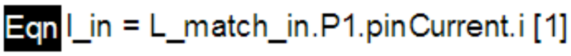
I_in =

Errors:

hb_basic

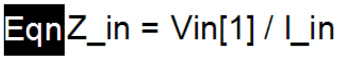
L_match_in.P1.pinCurrent.i
L_match_in.P2.pinCurrent.i
L_match_out.P1.pinCurrent.i
L_match_out.P2.pinCurrent.i
Mix
Mix(1)
PORT1.P1.pinCurrent.i

Then add the bracketed [1] so that it's the current at 1900 MHz. The result equation should be as shown here – you will use this to calculate impedance and power.

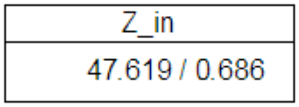


Eqn I_in = L_match_in.P1.pinCurrent.i [1]

c. Write another equation to calculate **Zin** using Vin Vin and I_in at 1900 MHz as shown here. Then insert insert a list of the Z_in equation. Notice the complex complex impedance is not 50 ohms!

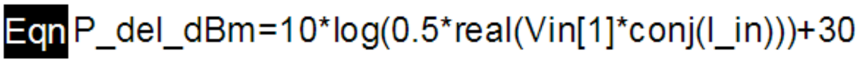


Eqn Z_in = Vin[1] / I_in



Z_in
47.619 / 0.686

d. Write an equation to calculate average delivered power using the node voltage Vin and the input current equation I_in:

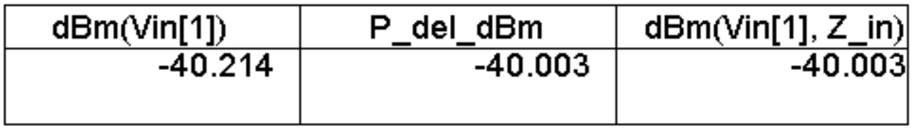


Eqn P_del_dBm=10*log(0.5*real(Vin[1]*conj(I_in)))+30

Note that 0.5 gives the average of the peak value, the **conj** function converts the complex current to its conjugate because V&I must be in phase to dissipate power and + 30 converts the value to dBm (same as dividing by 0. 001).

e. Edit your earlier list of dBm(Vin[1]) and **delete dbm_out** and add the equation **P_del_dBm**. Also, add another Vin trace and edit the trace expression to read: **dBm(Vin[1],Z_in)**. Now you have three ways of computing input power to compare. Notice that two of the values are the same:

dBm using defaults dBm using V and I dBm using Z_in eqn



dBm(Vin[1])	P_del_dBm	dBm(Vin[1], Z_in)
-40.214	-40.003	-40.003

NOTE on pin currents vs current probes: You could have used a current probe at the input instead of the pin current through the inductor. Either way is OK. However, the pin currents can make the datasets large if the circuit is also large.

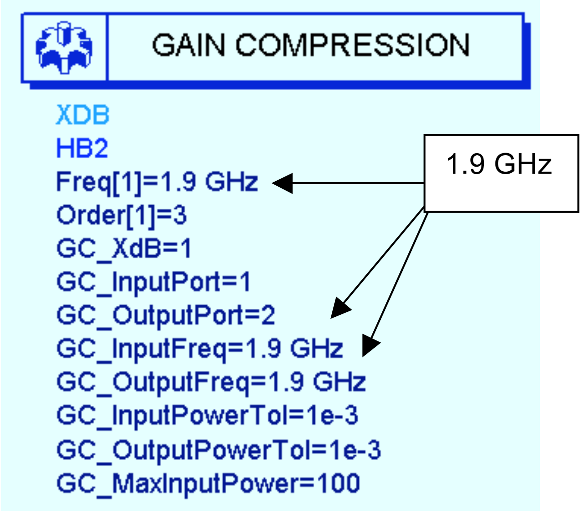
7. Test for Gain Compression using the XDB simulator.

The XDB simulation controller is a special use Harmonic Balance simulation for gain compression.

a. Save all you current work: schematic and data display. Then save the schematic with a new name: **hb_compression**. Afterward, close the hb_basic data display.

b. In the new schematic, **deactivate** the HB1 controller.

c. Go to the **Simulation-XDB** palette and insert the **XDB** controller. Edit the controller on screen so that Freq[1] and GC input and output frequencies are all **1.9 GHz** as shown. The parameter GC_XdB = 1 means that the test will be for 1 dB compression. Later, if you wanted 3 or 6 dB compression, simply change the value.

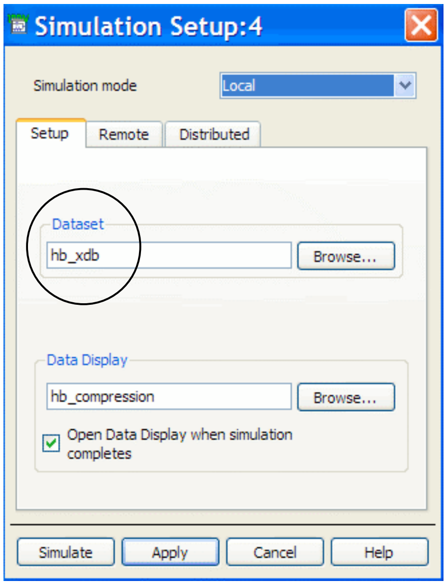


GAIN COMPRESSION

XDB
HB2
Freq[1]=1.9 GHz
Order[1]=3
GC_XdB=1
GC_InputPort=1
GC_OutputPort=2
GC_InputFreq=1.9 GHz
GC_OutputFreq=1.9 GHz
GC_InputPowerTol=1e-3
GC_OutputPowerTol=1e-3
GC_MaxInputPower=100

1.9 GHz

d. In the **Simulation Setup**, change the Dataset name to **hb_xdb** and then **Simulate**.



Simulation Setup:4

Simulation mode: Local

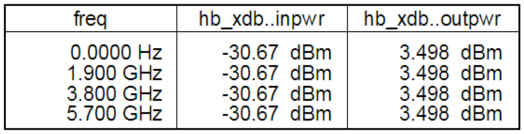
Setup | Remote | Distributed

Dataset: hb_xdb

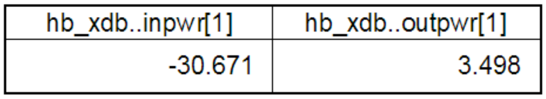
Data Display: hb_compression

☑ Open Data Display when simulation completes

e. When the data display opens, insert a list of of **inpwr** and **outpwr**. Then edit directly on the on the list by inserting a bracketed one **[1]** after **[1]** after each data item as shown here. If If desired, title the plot as shown. You just just performed a 1 dB gain compression test in test in only a few seconds! Because this amplifier is biased quite high, the 1dB compression point occurs when the input input power is about –30 dBm as shown here. here. In the next steps, you will modify the the schematic and set up a power sweep with with harmonic balance – another way to test test compression!

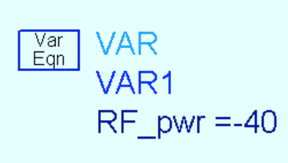


freq	hb_xdb..inpwr	hb_xdb..outpwr
0.0000 Hz	-30.67 dBm	3.498 dBm
1.900 GHz	-30.67 dBm	3.498 dBm
3.800 GHz	-30.67 dBm	3.498 dBm
5.700 GHz	-30.67 dBm	3.498 dBm



hb_xdb..inpwr[1]	hb_xdb..outpwr[1]
-30.671	3.498

8. Simulate compression with a power sweep.

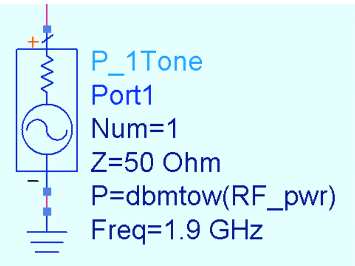


Var Eqn

VAR
VAR1
RF_pwr =-40

a. Deactivate the XDB and activate the HB controller.

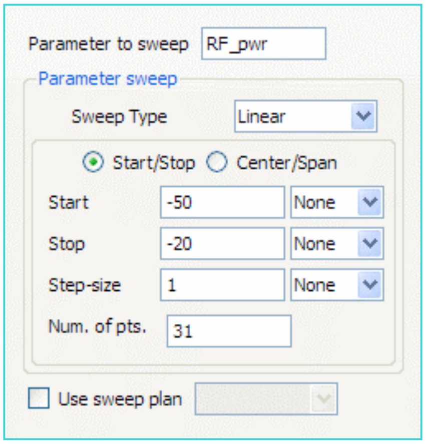
b. Insert a variable equation **VAR** for **RF_pwr = -40**.



P_1Tone
Port1
Num=1
Z=50 Ohm
P=dbmtow(RF_pwr)
Freq=1.9 GHz

c. Set the RF source power to the variable**: P=dbmtow (RF_pwr)**.

d. Edit the HB controller. In the sweep tab, set the **RF_pwr** sweep as shown from –50 to –20, step 1.



Parameter to sweep: RF_pwr

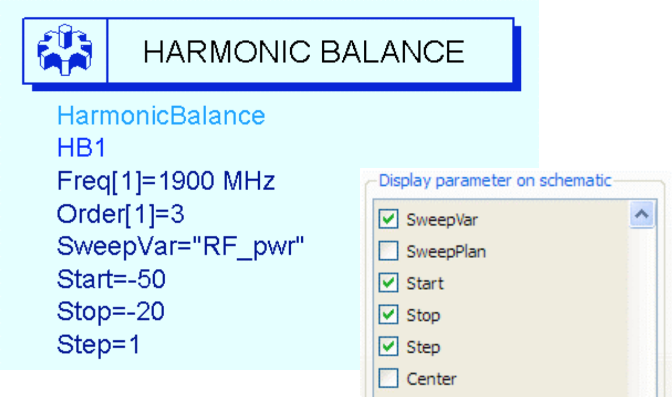
Parameter sweep

Sweep Type: Linear

● Start/Stop ○ Center/Span

Start	-50	None
Stop	-20	None
Step-size	1	None
Num. of pts.	31	

□ Use sweep plan



HARMONIC BALANCE

HarmonicBalance
HB1
Freq[1]=1900 MHz
Order[1]=3
SweepVar="RF_pwr"
Start=-50
Stop=-20
Step=1

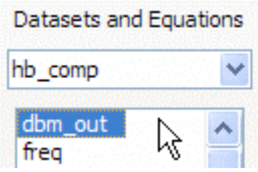
Display parameter on schematic

☑ SweepVar
□ SweepPlan
☑ Start
☑ Stop
☑ Step
□ Center

e. Go to the **Display** tab and set the SweepVar and its values to be displayed on the HB controller component as shown here.

f. Change the dataset name to: **hb_comp** and **simulate**. When the data display window opens, answer **No** to changing the dataset - this will keep the XDB data valid as the default dataset. Now, you will have to explicitly plot the hb_comp data – this is common practice.

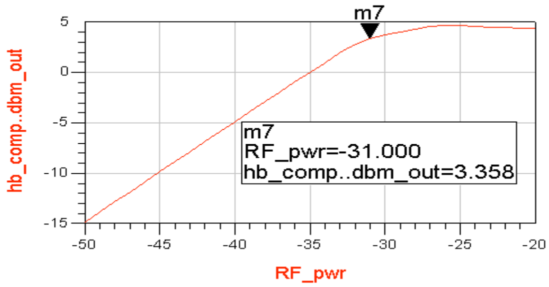
g. Insert a plot and select the **hb_comp** dataset. Then plot the schematic measurement equation **dbm_out**. Insert a marker on the trace where the value of RF_pwr is near the XDB inpwr value: -31. As you can see, the two values are close but they differ because the sweep resolutions resolutions are different – the XDB simulation (different DDS DDS and dataset) used many closely spaced sweep values.



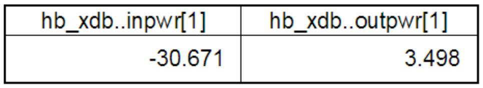
Datasets and Equations

hb_comp

dbm_out
freq



1 dB compression point from XDB

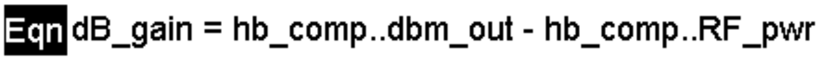


hb_xdb..inpwr[1]	hb_xdb..outpwr[1]
-30.671	3.498

values.

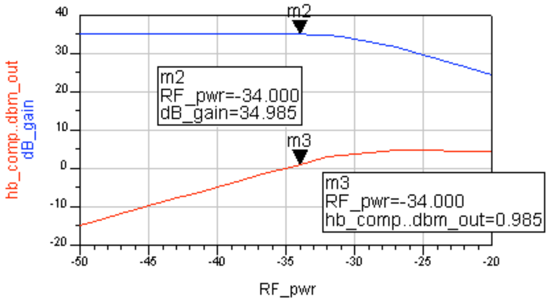
9. **Plot various gain, power, and line equations.**

a. Write an equation, **dB_gain** that uses the *dbm_out* measurement equation. By subtracting the linear input RF_pwr from dbm_out, the result is the gain at all values of RF input power:



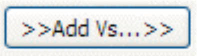
Eqn dB_gain = hb_comp..dbm_out - hb_comp..RF_pwr

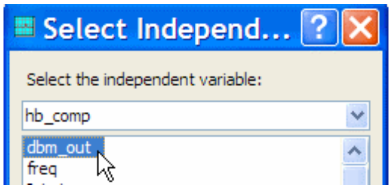
b. Edit the plot of dbm_out and add the dB_gain equation - the Y axis scale will automatically adjust. You can add markers to see both values at one RF power level as shown.

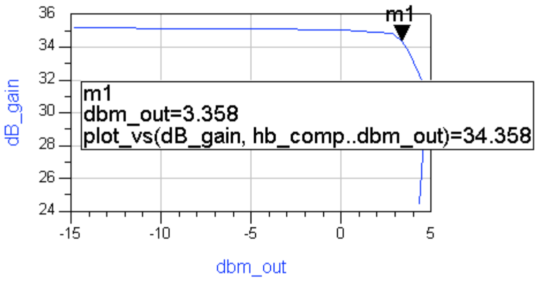


c. To plot dB_gain against output power, insert a new plot, add the **dB_gain** equation and then click **AddVs**. Next, select the **hb_comp** dataset and the independent variable for the X axis: **dbm_out**. Click OK and the sharp fall of gain will be plotted as shown. Use markers to read the values.

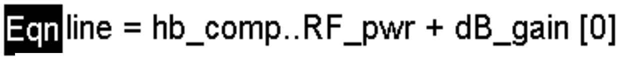
d. Write one more equation, **line**, to create a linear line (extrapolated data)







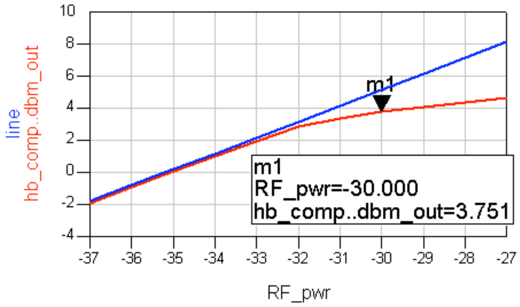
that represents the ideal output power with no compression. By adding the uncompressed gain at the first data point [0] to the RF power at every point, you get the ideal gain or line.



Eqn line = hb_comp..RF_pwr + dB_gain [0]

e. Insert a new plot of **dbm_out** (using hb_comp data) and add **line** also. This visually shows the amplifier's deviation from linear output power.

f. Save all your work.

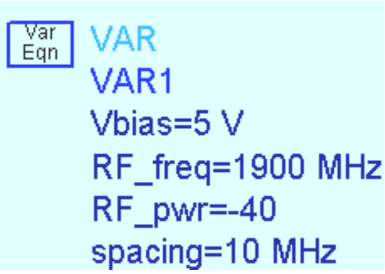


10. Two-tone HB simulation with variables.

The next few steps show more use of variables in simulation control. This is important for more complex circuit refinement, calculations in the remaining labs, and working with ADS examples which use this method of simulation control.

a. Save the last schematic design with a new name: **hb_2Tone**.

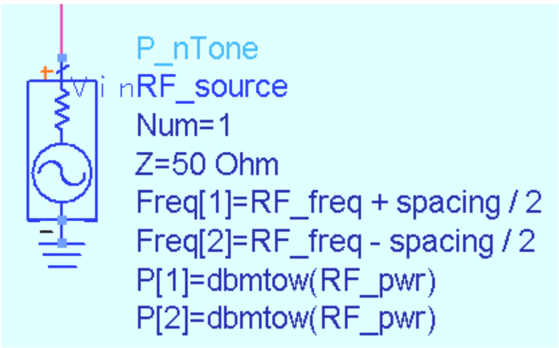
b. **Edit** the **VAR** and add variables for **RF_freq** and **spacing** as shown here. Vbias is not required - you may or may not have Vbias if you did an optional step earlier.



VAR
VAR1
Vbias=5 V
RF_freq=1900 MHz
RF_pwr=-40
spacing=10 MHz

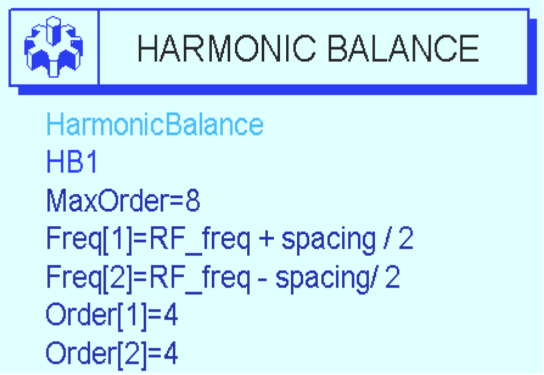
NOTE on units in VARs – If you set units here **do not set them anywhere else** or they may multiply in the simulation.

c. **Change the source to a P_nTone.** Edit the source so that it has two tones: Freq [1] and [2] with RF_pwr for each as shown here.



P_nTone
RF_source
Num=1
Z=50 Ohm
Freq[1]=RF_freq + spacing / 2
Freq[2]=RF_freq - spacing / 2
P[1]=dbmtow(RF_pwr)
P[2]=dbmtow(RF_pwr)

d. **Edit the Harmonic Balance controller** as shown here by adding another frequency, Freq[2], and the values as shown, using the spacing variable / 2. Also, set **Order = 4** for both and set **MaxOrder = 8.** In this case, the two RF tones are spaced 5 MHz apart (channel spacing).



HARMONIC BALANCE

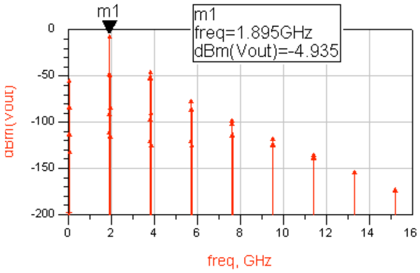
HarmonicBalance
HB1
MaxOrder=8
Freq[1]=RF_freq + spacing / 2
Freq[2]=RF_freq - spacing/ 2
Order[1]=4
Order[2]=4

MaxOrder = number of mixing products.

Freq[1] is a variable or a number. Order [1]= 4 means Freq[1] will be calculated with 4 harmonics.

e. **Remove the RF_pwr sweep** from the controller by erasing it on-screen or in the dialog and display. Also, remove any other controllers or unwanted components and save the design again.

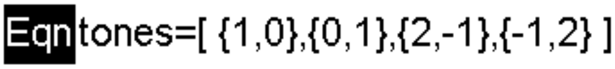
f. **Simulate** and **plot** the spectrum of Vout in dBm. Put a marker on a tone near 1900 MHz. Notice that you cannot clearly see the adjacent tones. To see the inter-modulation tones, you can either zoom in on the plot or try changing the X axis scaling. Try both of these methods quickly because the next step shows a technique using equations.



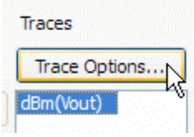
11. Use equations to access and control HB data.

a. Create a matrix with vectors (index values) to the desired tones. To do this, write the tones equation shown here. This equation creates a matrix using the square brackets. Within the brackets are curly braces with index values for the *mix* table. In this case, the number 1 represents the RF tone with spacing. Zero means that no other tone is desired (same as DC), and 2 represents two times the RF simulation tone.

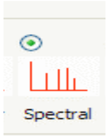
b. Insert a rectangular plot of **Vout** – spectrum in dBm. Then **use Trace Options** to edit **the Trace Expression** as shown here, using parenthesis – type in: **dBm(mix(Vout,tones))**. Also, set the **Trace Trace Type** to **Spectral**.

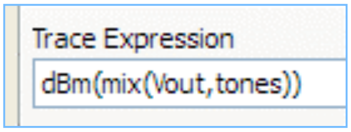

```
Eqn tones=[ {1,0},{0,1},{2,-1},{-1,2} ]
```

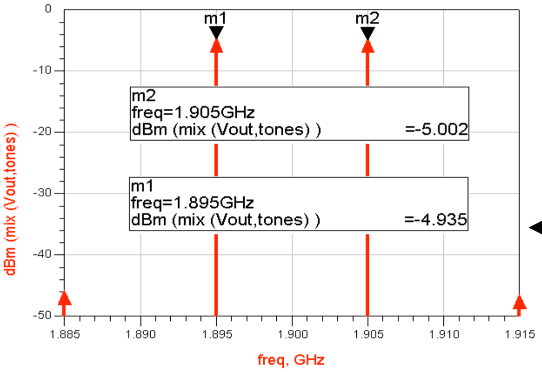
Use curly braces within brackets.

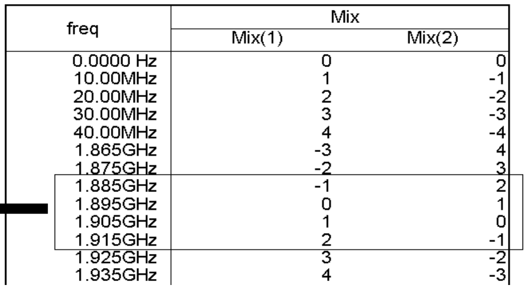


c. The plot should now show only the four tones tones you specified (10 MHz apart). To verify verify this, insert a **list** of **Mix** (Mix table). The The index values from the Mix table are the tones the tones that you specified with the ***tones*** equation. This is how Harmonic Balance data can be accessed and controlled using equations.







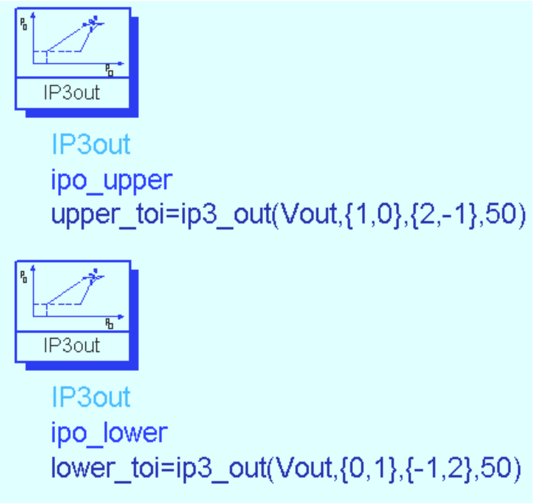


freq	Mix	
	Mix(1)	Mix(2)
0.0000 Hz	0	0
10.00MHz	1	-1
20.00MHz	2	-2
30.00MHz	3	-3
40.00MHz	4	-4
1.865GHz	-3	4
1.875GHz	-2	3
1.885GHz	-1	2
1.895GHz	0	1
1.905GHz	1	0
1.915GHz	2	-1
1.925GHz	3	-2
1.935GHz	4	-3

12. Simulate IP3 or TOI (Third Order Intercept)

a. On the hb_2Tone schematic, insert two Harmonic Balance **IP3out** measurement equations: one for the upper and one for the lower spaced tone. Many measurements require two-tones so name the instances upper and lower as shown here.

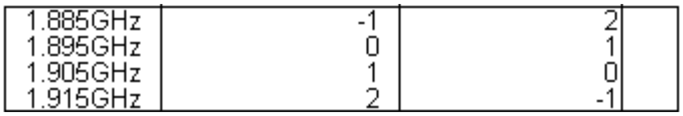
b. Note the default node label (vout), vectors {1,0}, and impedance 50. To match these values to your circuit, change vout to **Vout** (uppercase V). Then set the index values to correspond to your Mix table shown here from the last simulation (only lower_toi needs to change).



IP3out
ipo_upper
upper_toi=ip3_out(Vout,{1,0},{2,-1},50)

IP3out
ipo_lower
lower_toi=ip3_out(Vout,{0,1},{-1,2},50)

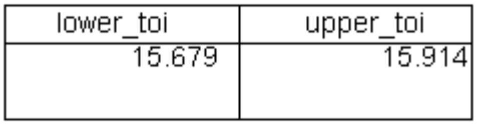
2-tone Mix HB data



1.885GHz	-1	2
1.895GHz	0	1
1.905GHz	1	0
1.915GHz	2	-1

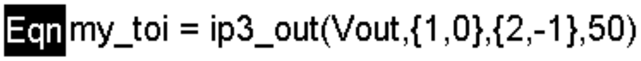
c. Check the equations to be sure they are correct and then **Simulate**.

d. In the Data Display, list the two measurement equation values as shown here.
Remove the independent variable using Plot Options. Here the amplifier TOI values appear reasonable and almost symmetrical.

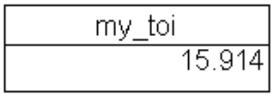


lower_toi	upper_toi
15.679	15.914

e. As an exercise in controlling data with ADS functions, write an equation in the Data Display for the same measurement as shown here. Then list it (my_toi) as shown here. You get the same results because you use the same function: ip3_out. The only difference is that this is after the simulation. Also, this equation is used in the optional step at the end of this lab.

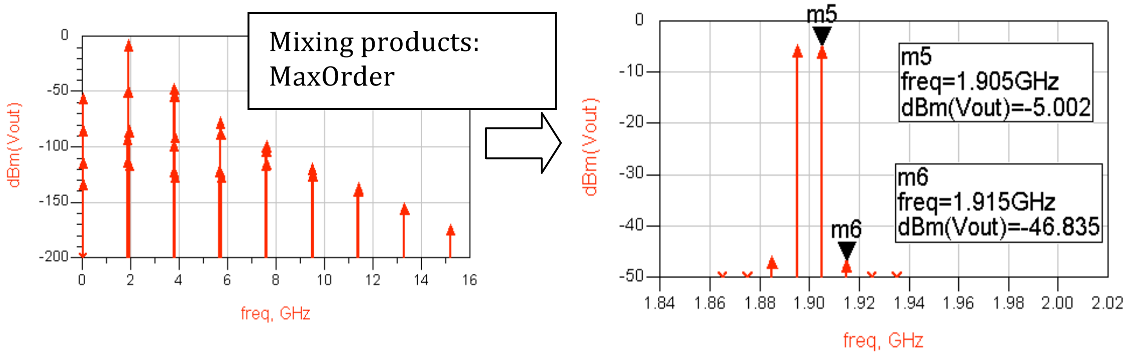


Eqn my_toi = ip3_out(Vout,{1,0},{2,-1},50)



my_toi
15.914

f. Plot the spectrum of Vout in dBm and then zoom in on the plot to see the two tones you just simulated. Put markers on the upper fundamental and the 3rd order tone – these should match the frequency values in the Mix table.



NOTE – You could easily go back to the schematic, change the spacing VAR value and simulate again. All the equations, plots and tables would simply fill up with the new data. This is the value of using variables for simulation and data displays.

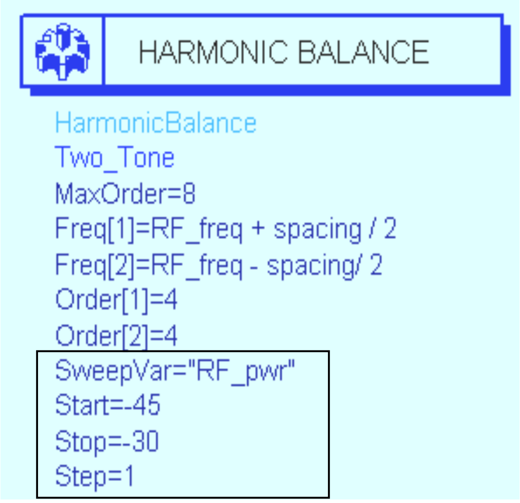
g. **Save** the schematic and data display.

NOTE for Mixer measurements – If you design mixers, the LO should be Freq[1] in the simulation controller because it has the most power. Also, in measurement equations, you will have to treat 2-tone data as if it were 3-tone: LO, RF1 and RF2 for upper and lower tones. For example, the upper IP3 equations for a down-converter would have the following index values: {-1,1,0},{-1,2,1} where –1 in the first represents the LO tone.

13. OPTIONAL - Sweep RF power against the TOI measurement

This step shows the effects on TOI when the input power begins to drive the device toward compression. In general, many measurements can be refined to get a better measure of circuit performance, beyond the required specifications. To do this, you must have a powerful non-linear simulator and data display tool such as ADS.

a. Using the same hb_2Tone design, set up the HB simulation controller to sweep the RF power as shown here from –45 to –30 dBm. You already tested 1 dB compression (about –31 dBm RF input power) and you just finished measuring TOI which (about 15 dBm).

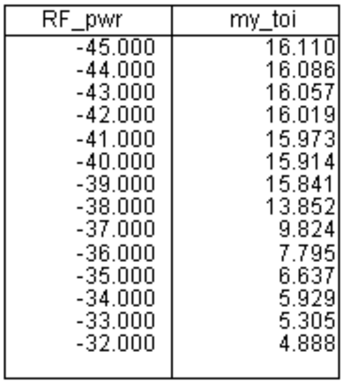


HARMONIC BALANCE

HarmonicBalance
Two_Tone
MaxOrder=8
Freq[1]=RF_freq + spacing / 2
Freq[2]=RF_freq - spacing/ 2
Order[1]=4
Order[2]=4
SweepVar="RF_pwr"
Start=-45
Stop=-30
Step=1

b. **Simulate** and watch the changes in the data display.

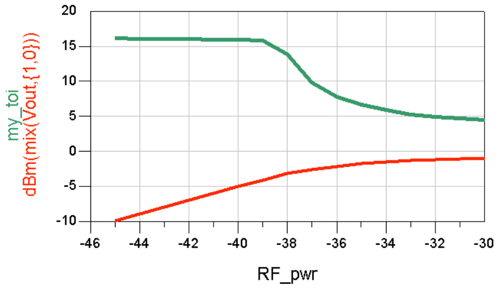
c. Edit the **my_toi** list to include the independent data (RF_pwr). Then increase the list size so that all the values appear. As you can see, TOI begins to change greatly as RF_pwr moves higher. However, the change is not linear. The next step will show this with more refinement.



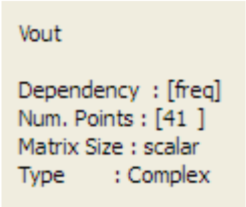
RF_pwr	my_toi
-45.000	16.110
-44.000	16.086
-43.000	16.057
-42.000	16.019
-41.000	15.973
-40.000	15.914
-39.000	15.841
-38.000	13.852
-37.000	9.824
-36.000	7.795
-35.000	6.637
-34.000	5.929
-33.000	5.305
-32.000	4.888

d. Change the list of **my_toi** to a **rectangular plot** (Plot Options - click the plot type icon). Then, on the same plot, insert **Vout** in **dBm** and edit the trace expression to return the upper RF tone as shown here. Now you can see how the TOI measurement tracks with that tone:

dBm(mix (Vout, {1,0}))



2 traces: my_toi and dBm of upper RF tone.



Vout

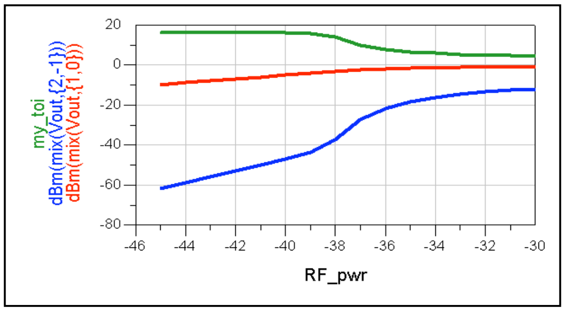
Dependency : [freq]
Num. Points : [41]
Matrix Size : scalar
Type : Complex

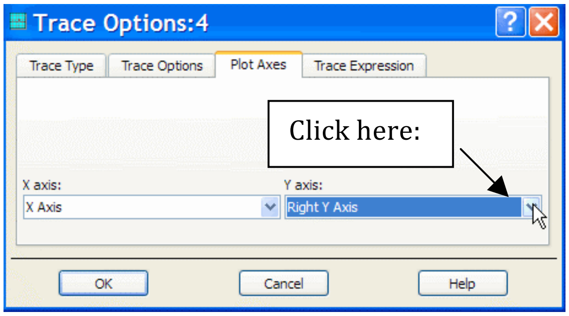
NOTE on Vout data – You must use the *mix* function because Vout contains 41 total frequency tones: 2 spaced fundamentals with 4 harmonics (this means 8 tones), with 8 max_order (this means 32 more intermod tones), plus the dc component. These 41 tones are present at each of the 16 values of RF power.

e. Add one more **Vout** trace to the plot. Again, edit the trace (Trace Expression) so that is becomes the upper 3rd order product:

dBm (mix (Vout, {2,-1}))

f. Your plot should now look like the one shown here. It should contain the upper RF_freq, the upper 3rd order product, and the equation my_toi (upper toi). Now, edit the **my_toi** trace and select **Plot Axes** as shown here. Then select **Right Y axis** for this trace and watch the change.

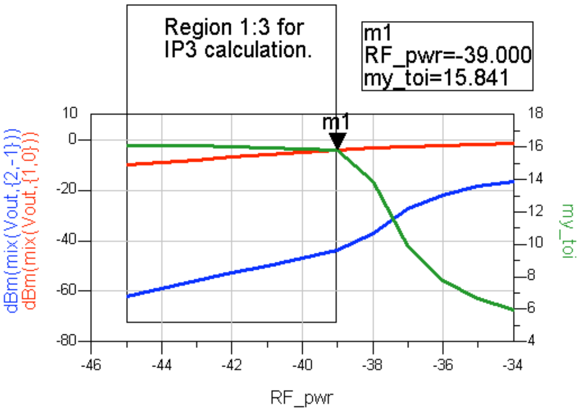




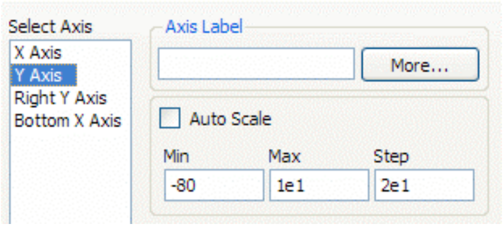
X axis: X Axis

Y axis: Right Y Axis

g. Your plot should now have the TOI value from your equation on the Right Y axis and the two tones used to calculate TOI on the left. Now, use **Plot Options**, select **Y Axis**, and remove the Auto Scale (uncheck the box). Then increase the **Max** to **10** and click **OK**. Finally, place a marker on the point where the slope of the two tones is no longer 3:1. As you can see, IP3 was calculated in the correct region. However, after the marker, the 3rd order product begins to rise at a sharper rate. This is a good example of using ADS to learn more about the performance of your design, beyond the specification.



Make your plot look similar using the Data Display text and drawing features.



□ Auto Scale

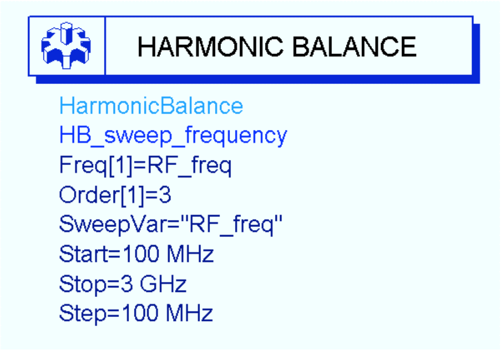
Min: -80

Max: 1e1

Step: 2e1

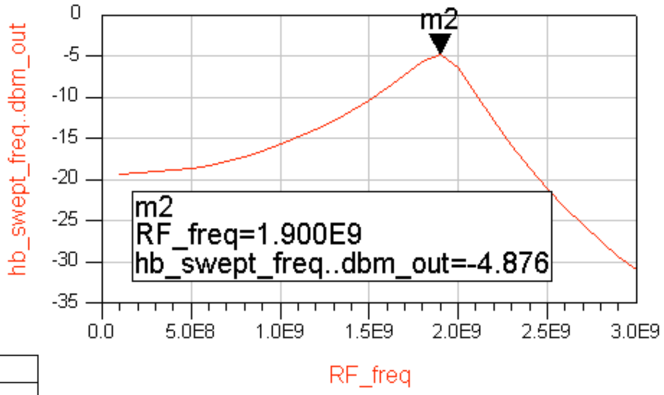
EXTRA EXERCISES:

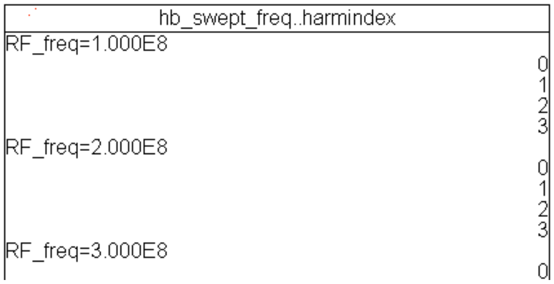
1. **Swept RF frequency** - Copy the schematic and then change the swept variable from RF power to one tone RF freq. To do this, set up the VAR for RF_freq in both the controller and the source. Sweep RF_freq from 100 MHz to 3GHz in 100 MHz steps. Be sure to change the dataset name, then simulate and plot the output power equation against the swept frequency as shown. Also, note that the dataset will contain a list of the harmonic index as shown.



HARMONIC BALANCE

HarmonicBalance
HB_sweep_frequency
Freq[1]=RF_freq
Order[1]=3
SweepVar="RF_freq"
Start=100 MHz
Stop=3 GHz
Step=100 MHz





	hb_swept_freq..harmindex
RF_freq=1.000E8	0 1 2 3
RF_freq=2.000E8	0 1 2 3
RF_freq=3.000E8	0

2. Try writing an equation to pass all the 5th order products to a spectral plot.

3. Use the **pspec** function to calculate power gain to the load. To do this, first look at the Help for pspec. Then insert a current probe at the Vout node.