# Agilent EEsof EDA

## Overview on Mixer Simulation with Agilent's ADS

**Agilent Technologies**

# Mixer Simulation with HP Advanced Design System

## Technical Note

### Introduction

Mixers are nonlinear circuits used to translate signals from one frequency to another. They are characterized by how much gain or loss they introduce in this conversion, as well as how much distortion and noise they contribute. For this reason, accurate nonlinear device models are essential to getting good results, but obtaining these models is not discussed here. This technical note covers some of the details required to simulate mixers with HP Advanced Design System. Unless otherwise noted, all of the simulation set-ups and data displays are included in the HP ADS 1.0 example project **examples/RFIC/ Mixers_prj**. A designer could easily copy and modify parts of this project in order to run the same simulations on his or her own design. Note that some of these simulations are applicable to receivers as well as mixers. Simulations discussed include:

DC analysis and back-annotation to the schematic (p. 2)

Conversion gain (p. 3)

Noise figure (p. 3, 11)

Third-order intercept (TOI) point (p. 5)

RF-to-IF gain compression (p. 7)

Conversion gain and noise figure versus RF frequency (p. 9)

Total noise power and noise power spectral density at the mixer output (p. 12)

Spurious-free dynamic range (p. 13)

Intermodulation distortion, input- and output-referred TOI, conversion gain, and leakage versus RF input power (p. 14)

Intermodulation distortion, input- and output-referred TOI, conversion gain, and leakage versus LO input power (p. 16)

Third-order intercept point and conversion gain of a two-stage down-converting receiver (p. 17)

Image rejection of a two-stage down-converting receiver (p. 19)

HEWLETT®
PACKARD

## Gilbert Cell Mixer Simulation

The simulations described are applied to the Gilbert cell mixer [1] shown in Figure 1. Note that the simulations are done on a finished mixer, but the analyses could be used to determine how the design could be improved. This Gilbert cell mixer is in both MDS and Series IV. It is the IAM-81018 mixer, originally designed by two engineers at Avantek [2]. This part is in the HP Communications Components catalog [3]. Note that package parasitics are not included in this example.
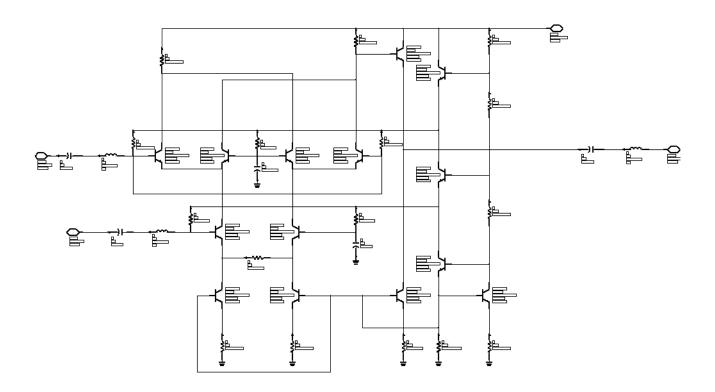


**Figure 1. Gilbert cell mixer used in the simulations described in this document.**

## DC Simulation and Back Annotation

The schematic in the example file, **DCTests,** shows the DC simulation of the mixer. Note that the new ADS back annotation feature allows the designer to see the DC voltages at each node, as well as the DC currents in each device terminal. This feature is accessed by first running a DC simulation, then opening the mixer subcircuit, and then selecting **Simulate/ Annotate DC Solution**. The DC voltage at each node as well as the current in each branch, as shown in Figure 2, indicates whether each device is biased correctly or not.
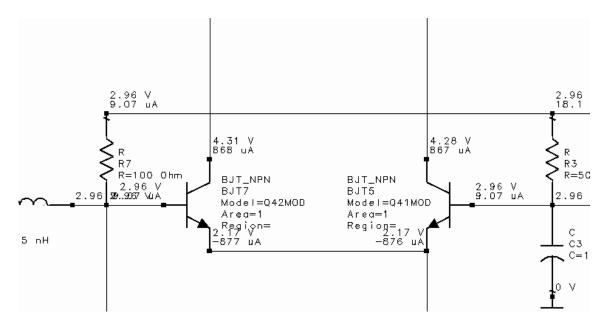
**Figure 2. Back annotation of the DC solution to a schematic.**

## Conversion Gain and Noise Figure

The schematic, **ConvGain** shown in Figure 3**,** shows how to set up and simulate conversion gain and noise figure at a single frequency. The conversion gain is calculated on the schematic, so this parameter could be optimized, if desired. The RF, LO, and IF frequencies as well as the RF power are all defined as variables, making it easier to change values and rerun the simulation if necessary. The two harmonic balance fundamental frequencies are defined as the LO and RF frequencies, respectively. The RF signal is small, so its order is just set to 1. The harmonic balance controller has been configured to compute noise figure. In order for this computation to be done, it is necessary to specify the input and output ports as well as the input frequency (InputFreq) and output frequency (FreqForNoise). With these settings, the simulator computes the correct conversion gain, which is necessary to compute the noise figure. It is not necessary to specify a noise node, unless the user wants to see the noise voltage computed at a particular node. For noise simulation, only nodes specified here will be output to the dataset, not all named nodes. The Krylov solver is used in this simulation (on the Harmonic Balance controller, set UseKrylov=yes) in order to speed it up, and it is quite fast, requiring only about 15 seconds on an HP 9000 Series C180 workstation.

The noise figure to each port is computed automatically. In this case only the noise figure to port 2 (nf(2)) is of interest. The conversion gain is computed as the power at the IF port at the IF frequency minus the power available from the RF source. This can be computed using the measurement equation shown, "ConvGain2." Power_RF can be used in this equation as long as it is included as an OutVar in the harmonic balance controller block. The mix() function is used to specify which spectral tone at the mixer output is the desired one. In this case, the mix(Vif,{-1,1}) function specifies the tone at -1*(Freq[1] in the HB analysis controller) +1*(Freq[2] in the HB analysis controller), or -1*LOfreq +1*RFfreq. The simulation results are shown in Figure 4.

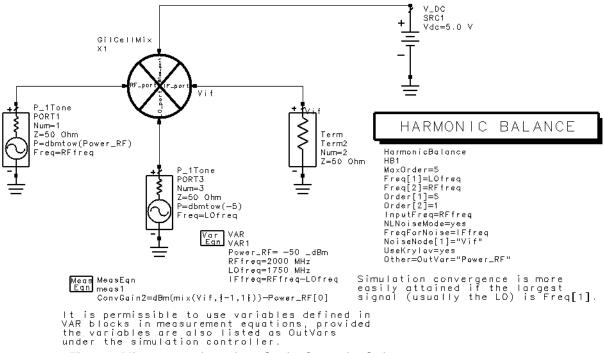SIMULATION OF NOISE FIGURE AND CONVERSION GAIN.
RF = 2000 MHz, IF = 250 MHz.

GilCellMix
X1

V_DC
SRC1
Vdc=5.0 V

P_1Tone
PORT1
Num=1
Z=50 Ohm
P=dbmtow(Power_RF)
Freq=RFfreq

Vif

Term
Term2
Num=2
Z=50 Ohm

HARMONIC BALANCE

P_1Tone
PORT3
Num=3
Z=50 Ohm
P=dbmtow(-5)
Freq=LOfreq

HarmonicBalance
HB1
MaxOrder=5
Freq[1]=LOfreq
Freq[2]=RFfreq
Order[1]=5
Order[2]=1
InputFreq=RFfreq
NLNoiseMode=yes
FreqForNoise=IFfreq
NoiseNode[1]="Vif"
UseKrylov=yes
Other=OutVar="Power_RF"

Var
Eqn
VAR
VAR1
Power_RF= -50 _dBm
RFfreq=2000 MHz
LOfreq=1750 MHz
IFfreq=RFfreq-LOfreq

Simulation convergence is more
easily attained if the largest
signal (usually the LO) is Freq[1].

Meas
Eqn
MeasEqn
meas1
ConvGain2=dBm(mix(Vif,{-1,1}))-Power_RF[0]

It is permissible to use variables defined in
VAR blocks in measurement equations, provided
the variables are also listed as OutVars
under the simulation controller.

**Figure 3. Mixer conversion gain and noise figure simulation set-up**

Eqn IF_spectrum=dBm(HB.Vif)

Spectrum at IF port



freq, GHz

Eqn ConversionGain=dBm(HB.Vif[1])-Power_RF[0]

| ConversionGain |
| --- |
| 11.044 |

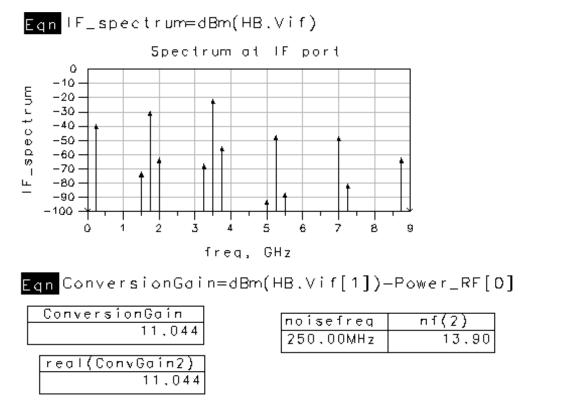| noisefreq | nf(2) |
| --- | --- |
| 250.00MHz | 13.90 |

| real(ConvGain2) |
| --- |
| 11.044 |

**Figure 4. Mixer output spectrum, conversion gain, and noise figure results. The conversion gain can be computed on the schematic page or here on a data display page, yielding identical results.**

4

Note that the input signals could be specified as voltages rather than powers, and the conversion gain could also be computed using voltages rather than powers. Also, small-signal mixer mode could be used for this type of simulation, which allows a reduction by one of the number of large-signal tones simulated in the circuit, when the tone that is being eliminated can be assumed to be small relative to the other tones. There is an example showing how to set up such a simulation in the examples/RFIC/MixerDiffMode_prj example project.

The schematic, **ConvGain_wFilt,** is the same set-up as ConvGain, except that an image-rejection filter has been added at the input, and a bandpass filter at the IF has been added.

## Third-Order Intercept Point

The schematic, **MixerTOI** shown in Figure 5, shows how to set-up a third-order intermodulation distortion simulation. In this simulation, two tones are supplied at the RF port. Nonlinearities in the mixer generate third-order intermodulation distortion, and the third-order intercept point is a measure of how much distortion is generated. This simulation is very fast, with UseKrylov set to yes.
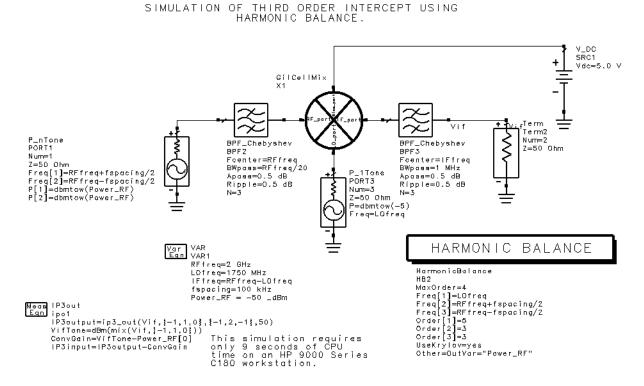
SIMULATION OF THIRD ORDER INTERCEPT USING HARMONIC BALANCE.



**Figure 5. Mixer Third-Order Intercept (TOI) simulation set-up.**

This simulation set-up is similar to the one for conversion gain, except that now the input signal has two large-signal tones separated by fspacing. The measurement equation calculates the output-referred third-order intercept point, conversion gain, and input-referred third-order intercept point. A built-in function, Ip3_out() uses notation similar to the mix() function to specify the desired output signal as well as the undesired

intermodulation tone.  In a more realistic simulation, the input and output filters should be replaced by less ideal ones, or measured S-parameters could be used.

Figure 6 shows the output spectrum with intermodulation distortion products at the IF frequency.  Markers have been added, and the output-referred third-order intercept point has been calculated based on simple geometry and the assumption that the intermodulation products will rise three times as fast as the desired IF tones, for each dB increase in RF power. The TOI calculations are identical, whether done on the schematic page or using markers here on the data display page. The output-referred third-order intercept point is calculated here using the marker readouts and the display shows that the measurement function on the schematic calculates the identical result.   The input-referred third-order intercept point is just the output-referred third-order intercept point minus the conversion gain. (The calculations are shown graphically in Figure 7).
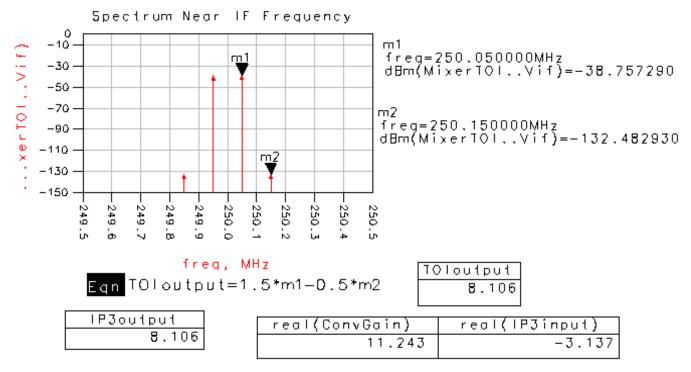


**Figure 6. Mixer TOI Simulation Results**

Note that when using this approach for computing TOI, the user should verify that the upper and lower intermodulation distortion terms in Figure 6 are the same amplitude.  If they are not, the number of harmonics of each input tone (in particular the LO) and/or the MaxOrder setting in the Harmonic Balance controller should be increased.  Also, this extrapolation assumes that the mixer nonlinearities generate intermodulation distortion terms that track the ideal slope of Figure 7, which may not always be the case.  It may be necessary to do a more thorough simulation, in which the RF power is swept.  This is shown in the **Intermodulation Distortion, Input- and Output-Referred TOI, Conversion Gain, and Leakage versus RF Input Power** section below.

Output Power (dBm)

Output-Referred TOI

Desired output signal
slope=1:1

☆ m1

Output-Referred TOI= m1+ 0.5(m1-m2)

3rd-order
intermod signal
slope=3:1

☆ m2

Input Power (dBm)
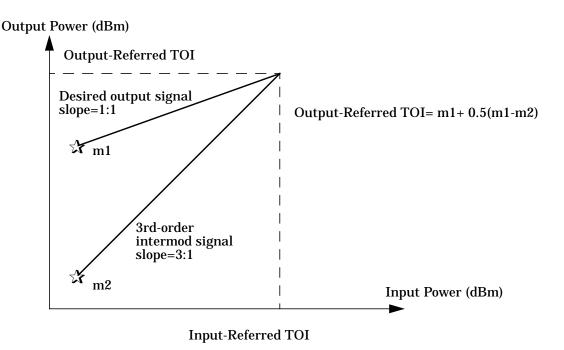
Input-Referred TOI

**Figure 7. Output-referred TOI is calculated by extrapolating the results of a low input power simulation.**

## RF-to-IF Gain Compression

The schematic, **RFIFcompression** in Figure 8, shows an easy way to determine the 1 dB gain compression of the mixer, using the new Gain Compression analysis component. The Gain Compression component uses harmonic balance to automatically sweep the input signal power level until the conversion gain from the specified input port and frequency to the specified output port and frequency is reduced from its small-signal value by the specified number of dB. The simulation results are shown in Figure 9.
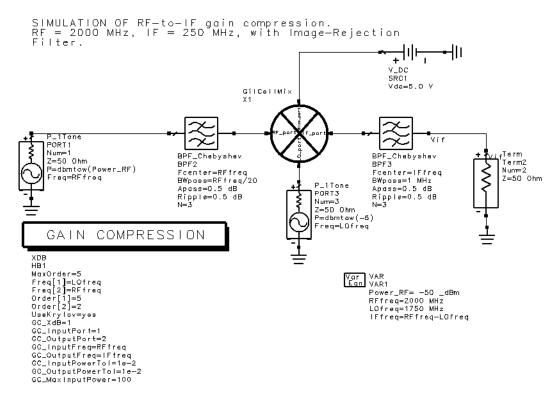
SIMULATION OF RF-to-IF gain compression.
RF = 2000 MHz, IF = 250 MHz, with Image-Rejection
Filter.

V_DC
SRC1
Vdc=5.0 V

GilCellMix
X1

P_1Tone
PORT1
Num=1
Z=50 Ohm
P=dbmtow(Power_RF)
Freq=RFfreq

BPF_Chebyshev
BPF2
Fcenter=RFfreq
BWpass=RFfreq/20
Apass=0.5 dB
Ripple=0.5 dB
N=3

RF_port   IF_port
O-port

P_1Tone
PORT3
Num=3
Z=50 Ohm
P=dbmtow(-5)
Freq=LOfreq

BPF_Chebyshev
BPF3
Fcenter=IFfreq
BWpass=1 MHz
Apass=0.5 dB
Ripple=0.5 dB
N=3

Vif

Term
Term2
Num=2
Z=50 Ohm

GAIN COMPRESSION

XDB
HB1
MaxOrder=5
Freq[1]=LOfreq
Freq[2]=RFfreq
Order[1]=5
Order[2]=2
UseKrylov=yes
GC_XdB=1
GC_InputPort=1
GC_OutputPort=2
GC_InputFreq=RFfreq
GC_OutputFreq=IFfreq
GC_InputPowerTol=1e-2
GC_OutputPowerTol=1e-2
GC_MaxInputPower=100

Var  VAR
Eqn  VAR1
Power_RF= -50 _dBm
RFfreq=2000 MHz
LOfreq=1750 MHz
IFfreq=RFfreq-LOfreq

**Figure 8. RF-to-IF gain compression simulation set-up.**

inpwr_lin is the lowest input power in the
simulation, in this case, -80 dBm.  outpwr_lin
is the output power when this input power is
applied.  ConvGain_linear is the conversion
gain in this "linear" region.

| inpwr_lin[1] | outpwr_lin[1] |
|---|---|
| -80.000 | -69.266 |

Eqn ConvGain_linear=outpwr_lin[1]-inpwr_lin[1]

| ConvGain_linear |
|---|
| 10.734 |

inpwr is the input power at which the conversion
gain drops by 1 dB from its low power value.
outpwr is the corresponding output power at
1 dB compression.  ConvGain_at_1dB_comp is the
conversion gain at the 1 dB compression point.

| inpwr[1] | outpwr[1] |
|---|---|
| -12.117 | -2.381 |

Eqn ConvGain_at_1dB_comp=outpwr[1]-inpwr[1]

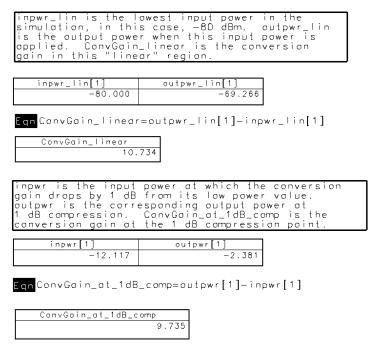| ConvGain_at_1dB_comp |
|---|
| 9.735 |

**Figure 9. RFIFcompression.dds window showing results of RF-to-IF conversion gain simulation. The conversion gain when the RF power is small (ConvGain_linear) is different from in Figure 4 because RF and IF filters are included here, and the order (number of harmonics) of the RF signal is 2 instead of 1.**

## Conversion Gain versus Swept RF Frequency

The schematic, **SweptRF** in Figure 10, simulates the conversion gain of the mixer as the RF frequency is swept from 200 MHz to 7 GHz.

In this case, the RF frequency is defined as a variable, Frf, and it is swept in the harmonic balance controller. The IF frequency, Fif, is also defined to be a constant at 70 MHz, and the LO frequency is set to one IF below the RF frequency.  Two sweep plans are used, to attain a finer-resolution sweep when the RF is below 1 GHz, and a coarser sweep when the RF is above 7 GHz.  The conversion gain is computed using the measurement equations shown, and it is output to the dataset. The conversion gain versus RF frequency is shown in Figure 11.

Figure 12 shows correct and incorrect ways to compute the power of the IF tone.  HP ADS outputs a spectrum at each named node on the schematic.  However, to compute the conversion gain, we need to know the power at the IF frequency, 70 MHz.  This is always the RF frequency minus the LO frequency, and the correct way to specify this spectral component is to use the mix function, as shown in the measurement equation for VifTone in Figure 10, and in the equation for IF_tone in Figure 12.



**Figure 10. Simulation set-up for swept RF, constant IF conversion gain simulation.**

**Eqn** ConvGain=IF_tone-Power_RF[0]

Conversion Gain versus RF frequency
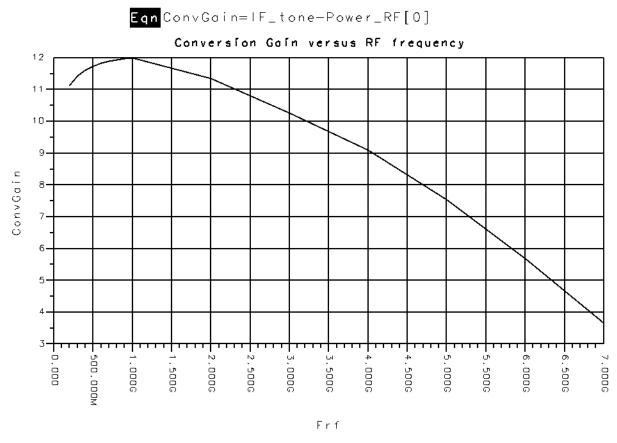


**Figure 11. Conversion gain versus RF input frequency.**

**Eqn** IF_toneOld=dBm(Vif[1])

**Eqn** IF_tone=dBm(mix(Vif,{-1,1}))

| Frf | IF_toneOld | IF_tone |
|---|---|---|
| 2.000E8 | -213.095 | -8.869 |
| 3.000E8 | -8.565 | -8.565 |
| 4.000E8 | -8.388 | -8.388 |
| 5.000E8 | -8.266 | -8.266 |
| 6.000E8 | -8.177 | -8.177 |
| 7.000E8 | -8.112 | -8.112 |
| 8.000E8 | -8.065 | -8.065 |
| 9.000E8 | -8.032 | -8.032 |
| 1.000E9 | -8.012 | -8.012 |
| 2.000E9 | -8.655 | -8.655 |
| 3.000E9 | -9.740 | -9.740 |
| 4.000E9 | -10.904 | -10.904 |
| 5.000E9 | -12.451 | -12.451 |
| 6.000E9 | -14.302 | -14.302 |
| 7.000E9 | -16.357 | -16.357 |

**Figure 12. Two ways of specifying spectral components. The mix() function, while more complicated, avoids selecting the wrong spectral component, which can occur if explicit indexing is used, as in the equation for IF_toneOld.**

## Noise Figure versus RF Frequency

The schematic, **SweptRF_NF** shown in Figure 13 below, simulates the noise figure of the mixer as a function of RF frequency, with the IF fixed at 70 MHz. The simulation set-up is very close to that shown in Figure 10 above, except that the large-signal RF tone has been replaced by a Term component, and the harmonic balance controller has been configured to calculate noise figure. Removing the large-signal tone from the RF input speeds up the simulation and should have very little effect on the computed result provided the amplitude of the RF input signal that is removed is at least 10 dB below the amplitude of the LO. Alternatively, noise figure can be computed using the Figure 10 simulation set-up, just by configuring the harmonic balance controller to calculate noise. Noise figure is calculated using the following equation:

$$NF = \frac{Noise\ Power\ at\ output\ due\ to\ noise\ at\ RF\ input\ at\ RF\ frequency}{Total\ Noise\ Power\ at\ output\ port\ at\ IF\ frequency}$$

$$NF = \frac{Na + kTBG}{kTBG}$$

Where Na is the noise added by the mixer

kTB is the noise power available from the input noise source, and
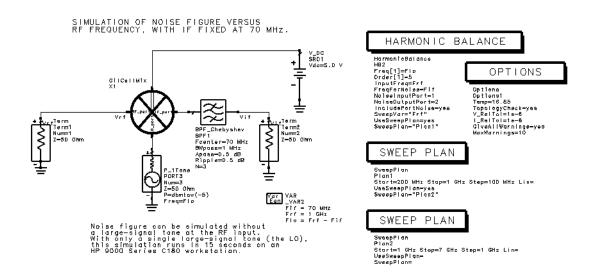
G is the conversion gain from RF to IF.



**Figure 13. Set-up for simulating noise figure versus RF frequency. Without a large-signal tone at the RF input, this simulation is faster than a set-up that includes the RF input tone.**

The simulator computes Na, which includes the noise that is frequency-translated to the IF by the large-signal LO tone (and any other large-signal tones.) The conversion gain, G, is also computed, and kTB is computed from Boltzmann's constant, the analysis temperature in degrees Kelvin, and the user-settable noise bandwidth (usually 1 Hz.) Note that this computed noise figure assumes that there is only one input frequency that

translates to the IF, unlike in an actual measurement, where it would be necessary to include an image-rejection filter at the input.

Figure 14 shows the results of the simulation.  Note that nf(2) is a response with two independent variables, Frf and noisefreq.  Adding the [0] syntax eliminates the noisefreq independent variable (it has only one value anyway), enabling the noise figure to be plotted and listed more conveniently.
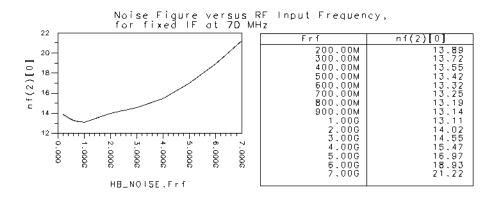


Noise Figure versus RF Input Frequency, for fixed IF at 70 MHz

| Frf | nf(2)[0] |
|---|---|
| 200.00M | 13.89 |
| 300.00M | 13.72 |
| 400.00M | 13.55 |
| 500.00M | 13.42 |
| 600.00M | 13.32 |
| 700.00M | 13.25 |
| 800.00M | 13.19 |
| 900.00M | 13.14 |
| 1.00G | 13.11 |
| 2.00G | 14.02 |
| 3.00G | 14.55 |
| 4.00G | 15.47 |
| 5.00G | 16.97 |
| 6.00G | 18.93 |
| 7.00G | 21.22 |

**Figure 14. Noise Figure versus RF input frequency.**

## Output Noise Power and Spurious-Free Dynamic Range

The schematic, **NoiseFloor,** simulates the total noise power at the output of  the mixer.  This result is combined with the MixerTOI simulation result to calculate the spurious-free dynamic range.  Note that it is possible to compute the spurious-free dynamic range from just a single simulation, but by running the noise simulation separately, only a single large signal tone (for the LO) is required, leading to a much faster noise simulation.  Figure 15 shows the simulation set-up.
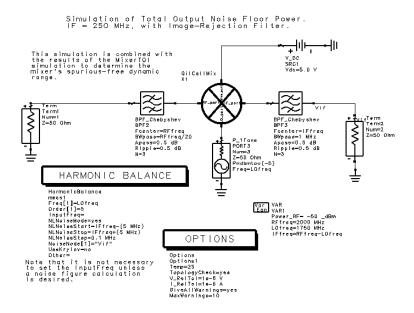


**Figure 15. Simulation set-up for calculating output noise power.**

The noise voltage spectral density will be computed in a 10 MHz bandwidth centered on the IF frequency, in 0.1 MHz steps. The total noise power at the output in this bandwidth will be computed, and this result along with a previous TOI simulation will be used to compute the spurious-free dynamic range. Figure 16 shows the noise spectral densities and noise output power calculations.
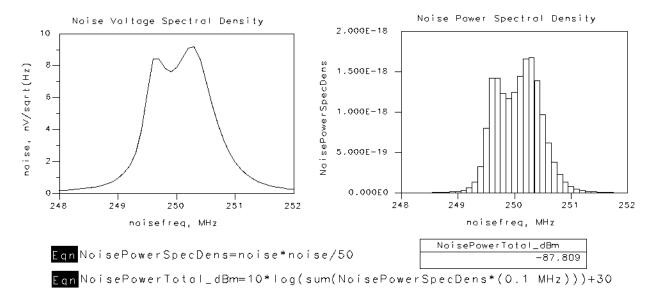


Eqn NoisePowerSpecDens=noise*noise/50

Eqn NoisePowerTotal_dBm=10*log(sum(NoisePowerSpecDens*(0.1 MHz)))+30

**Figure 16. Noise spectral density plots and output power calculations.**

There are several things to point out concerning Figure 16. First, if there is only one specified noise node in the simulation, the dataset will just have an entry called "noise," which is the noise voltage spectral density at this node. This value is plotted on the left in Figure 16. In order to calculate the total noise output power, we must first convert this to a power spectral density. Since ADS outputs rms noise voltages (unlike results from harmonic balance analyses which are peak voltages), and the output resistance is 50 ohms, the noise power spectral density is computed by squaring the noise voltage and dividing by 50. The result is plotted on the right in Figure 16, where the trace type has been set to histogram. The total noise output power is computed by summing the areas under each of the rectangles, since there is no built-in integration function in ADS 1.0. The NoisePowerTotal_dBm equation carries out this summation and converts the power into dBm. Note that if the simulation is rerun with a different frequency step size, the new step size will have to be entered into this equation. (It would be better to make the step size a variable and pass it into the dataset via the Other=OutVar= syntax on the Harmonic Balance controller, as shown in Figure 5.)

## Calculating Spurious-Free Dynamic Range

One definition of spurious-free dynamic range is the difference between a desired IF signal at the output and the noise floor at the output, when the spurious intermodulation distortion signals generated by the mixer or receiver have the same power as the noise floor. So it is necessary to combine results from both a noise floor simulation and an intermodulation distortion simulation to compute the spurious-free dynamic range. Figure 17 shows the calculation. The dynamic range calculation would be just the difference between the two markers if marker two where at the same power level as the noise floor. In general , this is not the case. So we assume that for each dB change in the input signals to the mixer, the signal at marker m1 increases 1 dB and the signal at

13

marker m2 increases 3 dB.  Based on this we can determine where marker m1 will be when the spurious signal at marker m2 has the same power level as the noise power.  This calculation is carried out in Figure 17.



**Figure 17. Spurious-free dynamic range calculation.**

## Intermodulation Distortion, Input- and Output-Referred TOI, Conversion Gain, and Leakage versus RF Input Power

The schematic, **IMDRFSwpHB,** simulates the mixer's intermodulation distortion, conversion gain, RF-to-IF leakage, LO-to-RF leakage, and LO-to-IF leakage versus RF input power, using harmonic balance.  The simulation set-up is shown in Figure 18.  This set-up is very similar to the TOI simulation set-up in Figure 5, except that Norton equivalent sources are used for the RF and LO sources.  This reduces the number of nodes in the circuit by one per source, which leads to a slightly faster simulation time.  Also, the power of each of the RF input tones is swept.
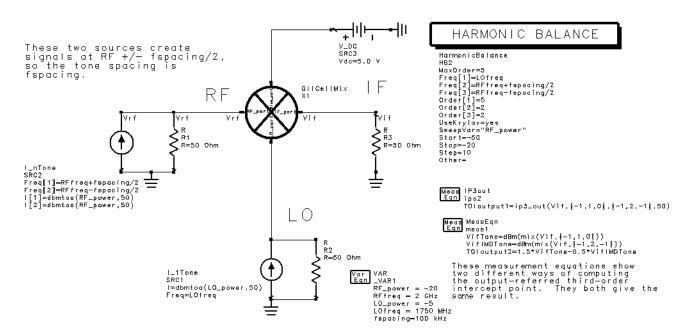


**Figure 18. Figure 18: Simulation set-up for intermodulation distortion, input- and output-referred TOI, conversion gain, and leakage versus RF input power**

Figure 19 shows the TOI and conversion gain simulation results.  The lower left plot shows one of the desired signals and one of the undesired intermodulation distortion signals at the output, versus RF input power.  The input- and output-referred TOI points

are computed via simple geometry as described in the **Third-Order Intercept Point** section above.  The TOI points can be computed either on the schematic or on the data display.  As shown here, the results are the same.
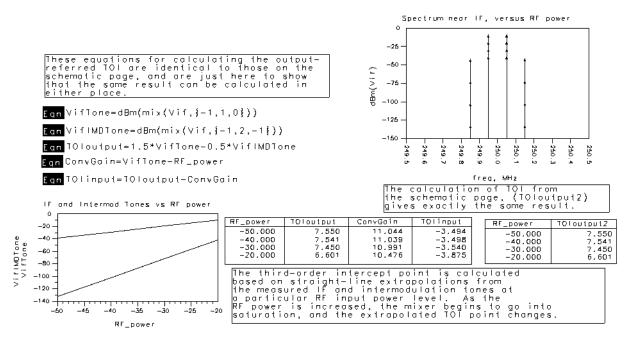


**Figure 19. TOI and conversion gain simulation results.**

Figure 20 shows the RF-to-IF leakage, LO-to-IF leakage, and LO-to-RF leakage as a function of RF input power.  As can be seen, these leakage terms vary only slightly with RF power.  Again, the mix() function is used to conveniently specify particular spectral components for these calculations.
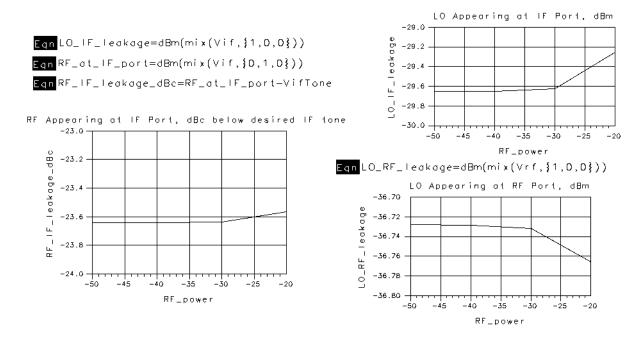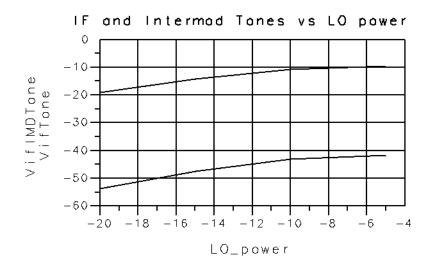


**Figure 20. Various leakage terms versus RF input power.**

## Intermodulation Distortion, Input- and Output-Referred TOI, Conversion Gain, and Leakage versus LO Input Power

The schematic, **IMDLOSwpHB,** simulates the mixer's intermodulation distortion, conversion gain, RF-to-IF leakage, LO-to-RF leakage, and LO-to-IF leakage versus LO input power, using harmonic balance.  It is identical to Figure 18, except that the LO power is swept instead of the RF power.  Figure 18 could be modified to sweep any other parameter in the circuit, instead.  Figure 21 shows that the TOI points and conversion gain do depend on the LO power. Figure 22 shows that the leakage terms do vary substantially with LO power.

The schematic, **IMDRFSwpEnv** simulates the mixer's intermodulation distortion versus RF input power, using the Circuit Envelope simulator.  While this was a good approach when using HP MDS, this type of simulation is generally faster with the New Harmonic Balance (Krylov) simulator, using the set-up shown in Figure 18.

IF and Intermod Tones vs LO power

| LO_power | TOIoutput | ConvGain | TOIinput |
|---|---|---|---|
| −20.000 | −1.669 | 0.955 | −2.624 |
| −15.000 | 2.285 | 5.709 | −3.425 |
| −10.000 | 5.447 | 9.246 | −3.800 |
| −5.000 | 6.601 | 10.476 | −3.875 |

**Figure 21. TOI points and conversion gain versus LO power.**

16

Eqn LO_IF_leakage=dBm(mix(Vif,{1,0,0}))

Eqn RF_at_IF_port=dBm(mix(Vif,{0,1,0}))

Eqn RF_IF_leakage_dBc=RF_at_IF_port-VifTone

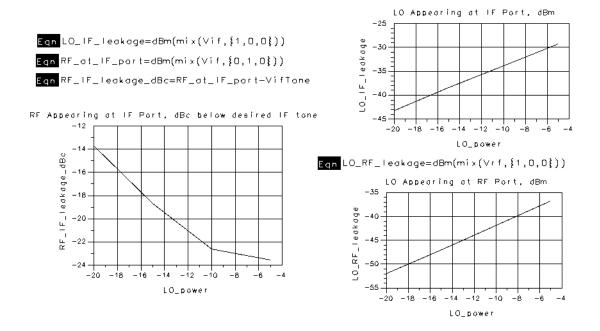Eqn LO_RF_leakage=dBm(mix(Vrf,{1,0,0}))

**Figure 22. Various leakage terms versus LO power.**

## Third-order intercept point and conversion gain of a two-stage down-converting receiver

The schematic, **DoubleConvHB,** simulates a double downconversion receiver's third-order intercept point and conversion gain. The simulation set-up is shown in Figure 23. There are two RF input signals, centered at 2 GHz and spaced 100 kHz apart, and two LOs, one at 1800 MHz and a second at 180 MHz. With the New Harmonic Balance (Krylov) simulator, this 4-tone simulation takes only 100 seconds to run.

Figure 24 shows the full output spectrum, the spectrum near the IF, the output-referred TOI point, and the conversion gain.
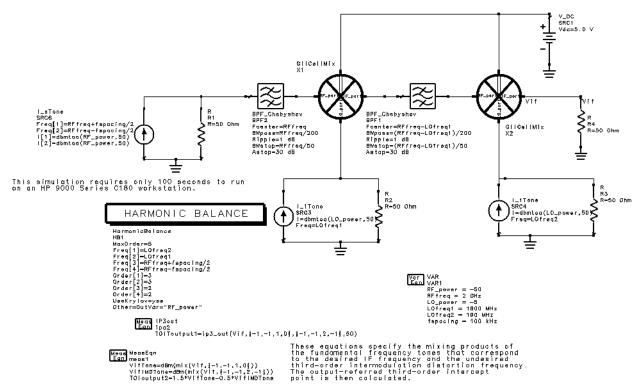
17

This simulation requires only 100 seconds to run
on an HP 9000 Series C180 workstation.

**Figure 23. Double-conversion receiver simulation set-up.**



m1
freq=10.050000MHz
Spectrum=-28.251666

m2
freq=10.150000MHz
Spectrum=-101.051489

| freq | TOIoutput | | TOIoutput2 |
|---|---|---|---|
| <inval... | 8.148 | | 8.148 |

Eqn TOIoutput=1.5*m1-0.5*m2

The TOI calculation using the marker readouts
gives the same result as the calculation from
the schematic.
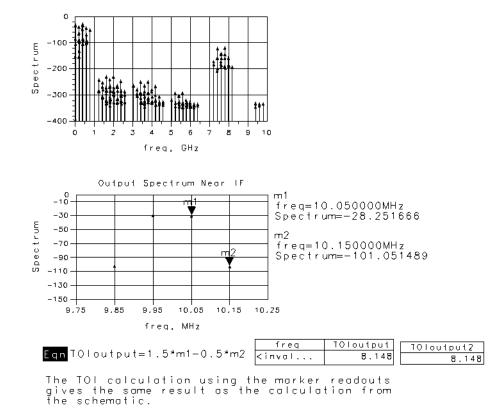
**Figure 24. Double-downconversion receiver simulation results.**

18

## Image Rejection of a Two-Stage Down-Converting Receiver

The schematic, **DblConvImag,** simulates the image rejection and conversion gain of the two-stage downconverter.  The image rejection is determined almost entirely by the rejection of the input filter at the image frequency.  The simulation set-up is identical to Figure 23, except that one RF input frequency is set to RFfreq=2 GHz and the other is set to 10 kHz below the image frequency (2*LOfreq1-RFfreq-(10 kHz)).  The 10 kHz offset term makes it easier to distinguish between the desired signal and the image signal appearing at the IF.  Figure 25 shows these two signals as well as a calculation of image rejection and conversion gain.
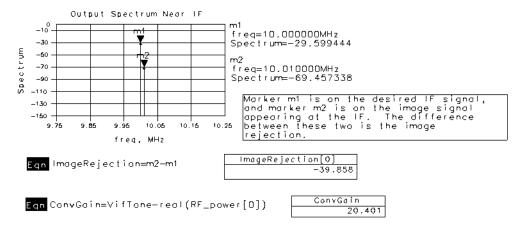
```
        Output Spectrum Near IF
    0                                    m1
 -10                                     freq=10.000000MHz
 -30          m1                         Spectrum=-29.599444
 -50          ▼
 -70          m2                         m2
 -90          ▼                          freq=10.010000MHz
-110                                     Spectrum=-69.457338
-130
-150
    9.75  9.85  9.95  10.05  10.15  10.25
            freq, MHz
```

Marker m1 is on the desired IF signal, and marker m2 is on the image signal appearing at the IF.  The difference between these two is the image rejection.

Eqn  ImageRejection=m2-m1

| ImageRejection[0] |
|---|
| -39.858 |

Eqn  ConvGain=VifTone-real(RF_power[0])

| ConvGain |
|---|
| 20.401 |

**Figure 25. Image rejection simulation results.**

## Filter Frequency Responses

The schematic, **RFBandFiltTest,** simulates the RF filter's frequency response (S21 and group delay), and the schematic, **FirstIFFiltTest,** simulates the first IF filter's frequency response.

## Summary

There are numerous simulations that can be applied to a mixer or receiver. These schematics and data displays show how to use HP Advanced Design System to set up various simulations and display results. With the speed improvements with the New Harmonic Balance (Krylov) simulator, none of the simulations shown here required more than a minute or two to run on an HP 9000 Series C180 Unix workstation. Users should be able to copy and modify this example project to easily simulate the same results on their mixers or receivers.

## References:

[1] B.Gilbert, "A Precise Four-Quadrant Multiplier with Subnanosecond Response," *IEEE Journal of Solid-State Circuits*, Vol. SC-3, pp. 365-373, December 1968.

[2] James Wholey and Issy Kipnis, "Silicon Bipolar Active Mixers", *Applied Microwave*, Spring 1990, pp. 287-293.

[3] Hewlett-Packard Co., *Communications Components GaAs & Silicon Products Designer's Catalog*, HP literature number 5091-4574E, 1993.

[4] Paul R. Gray and Robert G. Meyer, *Analysis and Design of Analog Integrated Circuits, Third Edition*. Wiley, New York, 1993, Chapter 10.

[5] Hewlett-Packard Co., HP EEsof Division, "Low-Power Mixer Design Example using HP Advanced Design System," 1998.

[6] Hewlett-Packard Co., HP EEsof Division, "Designer's Applications Guide," May, 1998. This is a description of each example in HP Advanced Design System, Release 1.0, and is available from your Hewlett-Packard field representative.

For more information about
Agilent EEsof EDA, visit:

**www.agilent.com/find/eesof**

**www.agilent.com**

For more information on Agilent Technologies'
products, applications or services, please
contact your local Agilent office. The
complete list is available at:

**www.agilent.com/find/contactus**

Product specifications and descriptions
in this document subject to change
without notice.

**Agilent Technologies**