



Agilent Technologies

Circuit Envelope Simulation

September 2004

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Copyright © 1998-2004, Agilent Technologies. All Rights Reserved.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries.

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

Pentium® is a U.S. registered trademark of Intel Corporation.

PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated.

UNIX® is a registered trademark of the Open Group.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Contents

1

Circuit Envelope Simulation

Performing a Circuit Envelope Simulation

Example (ADS only)

More Examples

Circuit Envelope Simulation Description

Circuit Envelope and Frequency-Domain-Defined Devices

Circuit Envelope and Components

Using Functions and Equations

Automatic Verification Modeling (Fast Cosimulation) Overview

Limitations

Envelope Simulation Controller

Env Setup

Env Params

Oscillator Analysis

Defining HB Simulation Parameters

Small-Signal Frequencies

Noise(1) Tab in ADS

Noise(2) Tab in ADS

Nonlinear Noise Setup in RFDE

Solver

TAHB

Cosim (in ADS)

Automatic Verification Modeling (in RFDE)

Troubleshooting a Simulation

Increasing Accuracy Easily

Improving Mixer IMD Measurement Accuracy and Speed

Using the Circuit Envelope Simulator to Analyze an Oscillator

Using the Circuit Envelope Simulator to Analyze Noise

Convolution Techniques Used in Circuit Envelope

1-4

1-5

1-9

1-10

1-17

1-17

1-18

1-19

1-22

1-23

1-24

1-26

1-29

1-31

1-35

1-35

1-36

1-36

1-36

1-36

1-36

1-37

1-41

1-41

1-41

1-41

1-42

1-43

Index

Chapter 1: Circuit Envelope Simulation

The Envelope simulator component, in the Simulation-Circuit Envelope palette, simulates high-frequency amplifiers, mixers, oscillators, and subsystems that involve transient or modulated RF signals. You can simulate:

- Amplifier spectral regrowth and adjacent channel power leakage with digitally modulated RF signals at the input
- Oscillator turn-on transients and frequency output versus time in response to a transient control voltage
- PLL transient responses
- AGC and ALC transient responses
- Circuit effects on signals having transient amplitude, phase, or frequency modulation
- Amplifier harmonics in the time domain
- Subsystem analyses using modulation signals such as multilevel FSK, CDMA, or TDMA
- Efficient third-order-intercept (TOI) and higher-order intercept analyses of amplifiers and mixers
- Time-domain optimization of transient responses
- Intermodulation distortion (although the Harmonic Balance simulator, with the new Krylov option selected, may provide a faster solution in most cases)

Typical applications for the Circuit Envelope simulation include:

- Time Domain Data Extraction

Selecting the desired harmonic spectral line it is possible to analyze:

- Amplitude vs. Time
 - Oscillator start up
 - Pulsed RF response
 - AGC transients
- Phase vs. Time
 - VCO instantaneous frequency, PLL lock time

- Amplitude & phase vs. time
Constellation plots
EVM, BER
- Frequency Domain Data Extraction

By applying FFT to the selected time-varying spectral line it is possible to analyze:

- Adjacent channel power ratio (ACPR)
- Noise power ratio (NPR)
- Power added efficiency (PAE)
- Reference frequency feedthrough in PLL
- Higher order intermods (3rd, 5th, 7th, 9th)

Circuit Envelope is highly efficient in analyzing circuits with digitally modulated signals, because the transient simulation takes place only around the carrier and its harmonics. In addition, its calculations are not made where the spectrum is empty.

- It is faster than Harmonic Balance, for a given complex signal Spice, assuming most of the frequency spectrum is empty.
- It does not compromise in Signal complexity, unlike time-varying HB or Shooting Method Component accuracy, unlike Spice, Shooting Method, or DSP.
- It adds physical analog/RF performance to DSP/system simulation with real-time co-simulation with ADS Ptolemy.
- It is integrated in same design environment as RF, Spice, DSP, electromagnetic, instrument links, and physical design tools.

Circuit Envelope provides these advantages over Harmonic Balance:

- In Harmonic Balance, if you add nodes or more spectral frequencies, the RAM and CPU requirements increase geometrically. Krylov improved this, but it's still a limitation of Harmonic Balance because the signals are inherently periodic.
- Conversely the penalty for more spectral density in Circuit Envelope is linear: just add more time points by increasing TSTOP. The longer you simulate, the finer your resolution bandwidth.
- Doing a large number of simple 1-tone HB simulations is effectively faster and less RAM intensive than one huge HB simulation.

- With a circuit envelope simulation the amplitude and phase at each spectral frequency can vary with time, so the signal representing the harmonic is no longer limited to a constant, as it is with harmonic balance.

However, Circuit Envelope does contain these limitations:

- More occupied spectrum than unoccupied spectrum. You're carrying more overhead with frequency-domain assumptions and harmonics than necessary. Use SPICE.
- Everything baseband; depends. If everything linear, use AC/S-parameter (for noise or budget). If everything nonlinear or digital, use SPICE. If everything logic/behavioral, use ADS Ptolemy.
- Occupied spectrum is relatively sparse. If you can do what you want using Harmonic Balance, you should. Post-processing, optimization, and yield are simpler and faster.

Refer to the following topics for details on Circuit Envelope simulation:

- [“Performing a Circuit Envelope Simulation” on page 1-4](#) describes the minimum setup requirements for a Circuit Envelope simulation.
- [“Example \(ADS only\)” on page 1-5](#) is a detailed setup for calculating intermodulation distortion, using a Gilbert Cell mixer as the example. The location of the mixer example is also given.
- [“Circuit Envelope Simulation Description” on page 1-10](#) is a brief outline of the simulation process, including specific details of the Circuit Envelope simulator.
- [“Automatic Verification Modeling \(Fast Cosimulation\) Overview” on page 1-19](#) describes a user-selected mode that can speed up lengthy cosimulations of Analog/RF circuits.
- [“Envelope Simulation Controller” on page 1-23](#) provides details on the tabs and fields in the Envelope simulation controller.
- [“Troubleshooting a Simulation” on page 1-41](#) offers suggestions on how to improve a simulation.

Performing a Circuit Envelope Simulation

Note You must have the Circuit Envelope simulator license (included with the RFIC Pro and Premier suites, the RF Board Premier suite, or the Microwave Circuits Premier suite) to use the simulator.

Start by creating your design, then add current probes and identify the nodes from which you want to collect data.

For a successful analysis, be sure to:

- Use either time domain or frequency domain sources in your circuit. In a circuit employing a mixer, provide a source for the LO.
- Add the Circuit Envelope controller to the schematic. (From the Component palette, choose Simulation-Envelope. Add the ENV component to the schematic.) Double-click to edit it. Fill in the fields under the Env Setup tab:
 - A Circuit Envelope simulation runs in the both the time and frequency domain. Set the stop time and time step (start time is 0). Time step defines the maximum allowed bandwidth ($\pm 0.5/\text{Time step}$) of the modulation envelope. The analysis bandwidth ($1/\text{Time step}$) should be at least twice as large as the modulation bandwidth to ensure accurate results at the maximum modulation frequencies.
 - Enter fundamental frequencies and order.
- You can perform small signal analysis. Enable Small-signal and fill in the fields under the Small-Sig tab.
- You can perform nonlinear noise analysis. Enable Nonlinear noise and fill in the fields under the Noise(1) and Noise(2) tabs.
- If your design includes an OscPort component, select the Env Params tab and fill in the Oscillator options.
- You can use previous simulation solutions to speed the simulation process. For more information, refer to the topic “Reusing Simulation Solutions” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.

Note Unless there are convergence problems, it is recommended that you leave the other parameters under the Env Params and HB Params at their default settings.

- After the simulation is complete, in the Data Display, Envelope data variables are identified by the prefix *ENV*.

Example (ADS only)

[Figure 1-1](#) illustrates an example setup for using the Envelope simulator to find mixer intermodulation distortion (IMD).

Note You must have the Circuit Envelope simulator license to simulate examples. You may build the Circuit Envelope example without this license, but will be unable to run the simulations.

Note This design, *IMDRFSwpEnv.dsn*, is in the *Examples* directory under *RFIC/Mixers_prj*. The results are in *IMDRFSwpEnv.dds*.

These two sources create signals at $RF \pm f_{spacing}/2$, so the tone spacing is $f_{spacing}$.

I_nTone

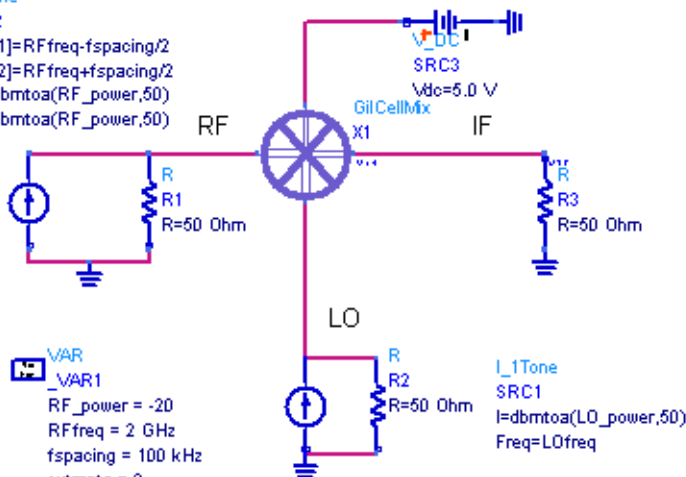
SRC2

Freq[1]=RFfreq-fspacing/2

Freq[2]=RFfreq+fspacing/2

I[1]=dbmtoa(RF_power,50)

I[2]=dbmtoa(RF_power,50)



VAR
_VAR1

RF_power = -20

RFfreq = 2 GHz

fspacing = 100 kHz

extrpts = 3

numpts = 20

LO_power = -5

LOfreq = 1750 MHz

tstop = (numpts+extrpts)*tstep

tstep = 1/(fspacing/2)*numpts)



PARAMETER SWEEP

ParamSweep

Sweep2

SweepVar="RF_power"

Sim Instance Name[1]="Env1"

Sim Instance Name[2]=

Sim Instance Name[3]=

Sim Instance Name[4]=

Sim Instance Name[5]=

Sim Instance Name[6]=

Start=-50

Stop=-20

Step=10



ENVELOPE

Envelope

Env1

Max Order=4

Freq[1]=LOfreq

Freq[2]=RFfreq

Order[1]=5

Order[2]=2

Sweep Offset=extrpts*tstep

Stop=tstop

Step=tstep

Figure 1-1. Example Setup for Using the Envelope Simulator to Find Mixer IMD

In this example:

- An RF center frequency of 2000 MHz and an LO frequency of 1750 MHz have been established by a VarEqn component. The spacing between tones has been established by the equation $fspacing=100\text{ kHz}$.
- An I_nTone source establishes two intermodulating RF frequencies by means of the following equations:
$$\text{Freq}[1]=\text{RFfreq}-fspacing/2 \text{ and } \text{Freq}[2]=\text{RFfreq}+fspacing/2$$
- An I_1Tone source establishes the LO frequency by means of $\text{Freq}=\text{LOfreq}$.

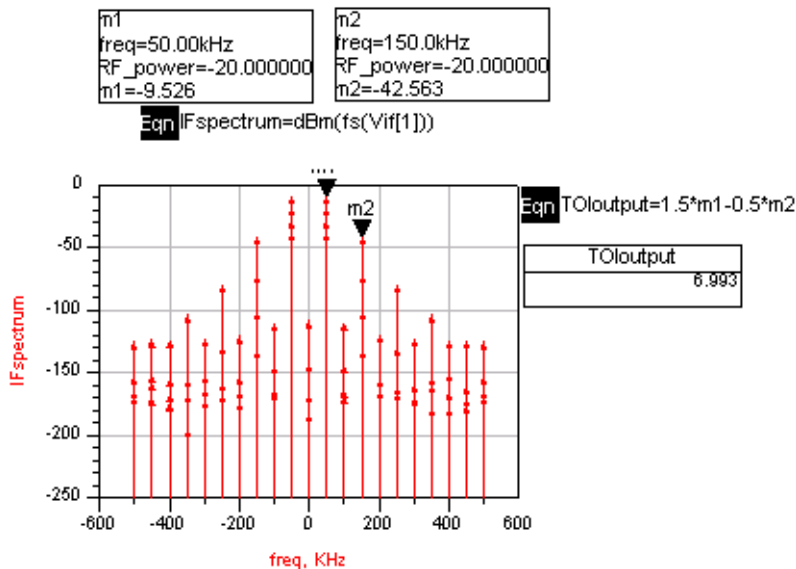
Hint Using current sources instead of voltage sources leads to faster simulations, because one fewer equation per source is generated. The function *dbmtoa* converts power to current at a default reference impedance of 50 ohms. P_1Tone and P_nTone components can also be used.

- A ParamSweep component establishes RF_power as the parameter to be swept. This component also establishes the Start, Stop, and Step values for the power sweep.
- In the Envelope Simulation component, LOfreq and RFfreq have been assigned to Freq[1] and Freq[2], respectively.
- *Stop time* has been determined by *tstop*, which in turn is defined by an equation in the VarEqn component. Similarly, *Time step* has been determined by *tstep*.

Note Because this example will later use the *fs()* function, the number of time points (determined by numpts=20 in the VarEqn component) must be even. *numpts* is the number of timepoints that are simulated per period of the modulation frequency. Modulation frequency is determined by $fspacing/2$, and *fspacing* has been established as 100 kHz.

- Transient responses are discarded by *extrapts*, the number of extra points to simulate at the start. This is the same as the *Sweep offset* parameter (under the *Env Params* tab).

The following figure shows the results of the simulation.



The x-axis frequency here is as an offset from the IF frequency.

IF spectral power is plotted against frequency in kHz, by means of the equation

$$\text{IFspectrum}=\text{dBm}(\text{fs}(\text{Vif}[1]))$$

Note The function *fs* performs a time-to-frequency transform, transforming the IF (Vif[1]) into the frequency domain.

The value zero on the *x*-axis represents the IF, with values to the left and right representing the mixing products that are offset from the IF. The marker M2, at +150 kHz, indicates one of the third-order IMD products. The equation *TOLoutput* uses simple geometry.

More Examples

For more Circuit Envelope simulations, refer to these example projects:

- For ways of generating sources for use in envelope simulations (such as $\pi/4$ -DQPSK, FSK, QAM, and CDMA), see *Tutorial/ModSources_prj*.
- To simulate amplifier spectral regrowth and adjacent channel power leakage with digitally modulated RF signals at the input, see *RF_Board/NADC_PA_prj*.
- To simulate PLL transient responses, see *RF_Board/PLL_5th_Order_prj* and *DECT_LO_Synth_prj*.

Circuit Envelope Simulation Description

The Envelope simulator combines features of time- and frequency-domain representation, offering a fast and complete analysis of complex signals such as digitally modulated RF signals.

Briefly, this simulator permits input waveforms to be represented in the frequency domain as RF carriers, with modulation “envelopes” that are represented in the time domain (Figure 1-2).

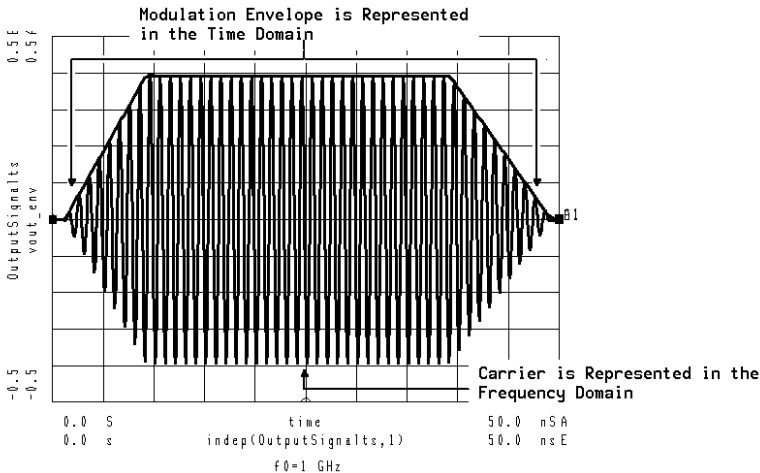
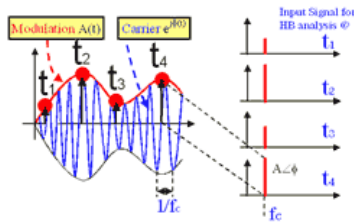


Figure 1-2. Modulated Signal in the Time Domain

The following five concepts present a basic overview of the circuit envelope simulation process.

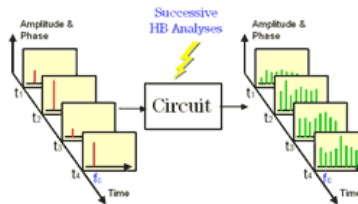
- Transform input signal

Each modulated signal can be represented as a carrier modulated by an envelope - $A(t) \cdot e^{j\phi(t)}$. The values of amplitude and phase of the sampled envelope are used as input signals for Harmonic Balance analyses.



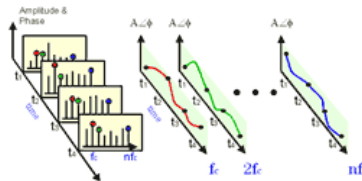
- Frequency domain analysis

Harmonic Balance analysis is performed at each time step. This process creates a succession of spectra that characterize the response of the circuit at the different time steps.



- Time domain analysis

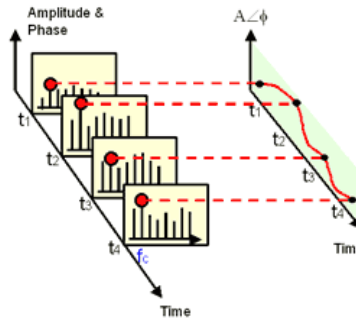
Circuit Envelope provides a complete non steady-state solution of the circuit through a Fourier series with time-varying coefficients.



- Extract data from time domain

Selecting the desired harmonic spectral line (f_c in this case), it is possible to analyze:

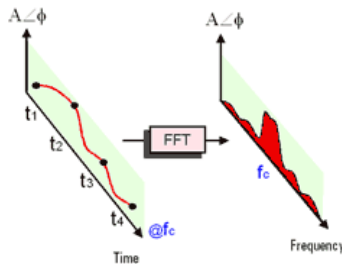
- Amplitude vs. time (oscillator start up, pulsed RF response, AGC transients)
- Phase (ϕ) vs. time (t) (VCO instantaneous frequency (df/dt), PLL lock time)
- Amplitude and phase vs. time (constellation plots, EVM, BER)



- Extract data from frequency domain

By applying FFT to the selected time-varying spectral line it is possible to analyze:

- Adjacent channel power ratio (ACPR)
- Noise power ratio (NPR)
- Power added efficiency (PAE)
- Reference frequency feedthrough in PLL
- Higher order intermods (3rd, 5th, 7th, 9th)



To describe the circuit envelope simulation process more specifically, in an envelope simulation each node voltage is represented by a discrete spectrum having time-varying Fourier coefficients. The set of spectral frequencies is user-defined; the amplitude and phase at each spectral frequency can vary with time, so the signal representing the harmonic is no longer limited to a constant, as it is with harmonic balance. Each spectral frequency can be thought of as the center frequency of a spectrum; the width of each spectrum is $\pm 0.5 / \text{Time step}$. [Figure 1-3](#) illustrates this, where the minimum envelope bandwidth is equal to the bandwidth of the modulation signal. In most cases the bandwidth of the modulation signal is much smaller than the lowest user-defined spectral frequency (which corresponds to the “carrier” frequency), unlike what is shown in the figure.

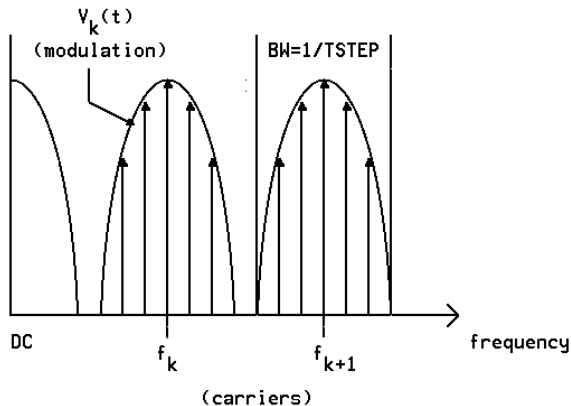


Figure 1-3. Spectra in the Frequency Domain

The bandlimited signal within each spectrum can contain periodic, transient, or random tones. The actual time-domain waveform is represented as a sum of carriers (with harmonics and intermodulation products), where each envelope can vary with time:

$$v(t) = \text{real} \left[\sum_{k=0}^N V_k(t) e^{j2\pi f_k t} \right]$$

Here, $v(t)$ is a voltage at any node in the circuit, including the input. The Fourier coefficients, $V_k(t)$, are allowed to vary with time and may represent an arbitrary modulation of each carrier. Since each time-varying spectrum $V_k(t)$ can be thought of as a modulation waveform with a center frequency f_k , these are often referred to as “envelopes.” This spectrum may represent transient signals with continuous spectra, such as a digital modulation envelope over an RF carrier, or periodic signals with discrete spectral lines, such as the two RF tones required for intermodulation distortion analysis.

Figure 1-4 illustrates a modulated signal and the time-varying spectrum that results from the simulation. Any spectral component obtained from the simulation can be processed and displayed in amplitude, phase, I (the in-phase modulation component), or Q (the quadrature modulation component). By computing the Fourier transform of the spectral component, the simulator can present the spectrum around the component, as in a spectrum analyzer display.

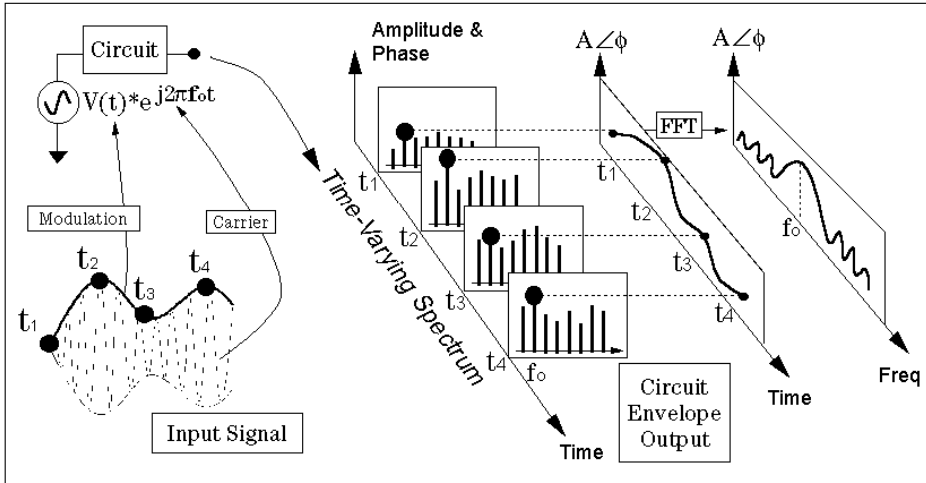


Figure 1-4. A Modulated Signal and Its Simulated Time-Varying Spectrum

This simulator does not require that a spectral component be present at the center frequency. [Table 1-1](#) gives examples of how the spectrum can be defined. You can use a `V_1Tone` source component to generate the various signals depicted in the table, assigning to the source parameter `V` the various expressions for V_k . In a given envelope simulation, there are $N + 1$ possible spectral components. The one at DC (also referred to as the baseband component) is limited to a bandwidth of $0.5/\text{Time step}$, and only the real portion of V_k is used. The other N spectra have a double-sided bandwidth of $1/\text{Time step}$ and are usually complex, as shown in [Figure 1-3](#).

The envelope waveform $V_k(t)$ has many useful properties. For example, to find the instantaneous amplitude of the signal around f_k at time t_s , simply compute the magnitude of the complex number $V_k(t_s)$. Similarly, the phase, real, and imaginary values of instantaneous modulation can be extracted by simply computing the phase, real, and imaginary values of $V_k(t_s)$.

Note This process only extracts the magnitude of the modulation around f_k . It does not include any of the spectral components of adjacent f_{k-1} or f_{k+1} spectra, even if these spectra actually overlap. If this f_k spectrum has multiple tones inside of it, then this demodulation does include their effects.

Table 1-1. Examples of Defining a Spectrum around f_k

| # | Formula that specifies the envelope | Description of the signal in the time domain |
|----|--|--|
| 1 | $V_k=1$ | Pure cosine: $\cos(2\pi f_k \text{time})$ |
| 2 | $V_k=\exp(j\pi/2)$ or $\text{polar}(1,90)$ or $1*j$ | Pure sine: $\sin(2\pi f_k \text{time})$ |
| 3 | $V_k=A*\exp(j*2\pi f_m \text{time}+B)$ | One tone (SSB) $A*\cos(2\pi f_m (f_k+f_m)\text{time}+B)$ |
| 4 | $V_k=A*\exp(j*B)$; $\text{freq}=1.1 \text{ GHz}$ | Same as (3) (assuming $f_k + f_m = 1.1 \text{ GHz}$) |
| 5 | $V_k=2*\cos(2\pi f_m \text{time})$ | Two tones (AM suppressed carrier) |
| 6 | $V_k=\exp(j*2\pi f_m \text{time}) + \exp(-j*2\pi f_m \text{time})$ | Same as (5) |
| 7 | $V_k=\text{pulse}(\text{time},...)$; $\text{freq}=f_k+f_m$ | Pulsed RF at a frequency of $f_k + f_m$ |
| 8 | $V_k=-\text{step}(\text{time} - \text{delay})$ | A negative cosine wave, gated on at $t=\text{delay}$ |
| 9 | $V_k=(\text{vreal}(\text{time})+j*\text{vimag}(\text{time}))*\exp(j*2\pi f_m \text{time})$ | I/Q modulated source centered at f_k+f_m . ($\text{vreal}()$, $\text{vimag}()$ user-defined functions) |
| 10 | $V_k=(1 + \text{vr1}) * \exp(j*2\pi \text{vr2})$ | Amplitude- and noise-modulated source at f_k . (vr1 , vr2 are user-defined randtime functions, not available in Release 1.0) |
| 11 | $V_k=\exp(j*2\pi*(-f_0 + a_0*\text{time}/2)*\text{time})$ | Chirped FM signal starting at $f_k - f_0$, rate = a_0 |

This simple technique does not allow the demodulation of only one of the tones inside this f_k spectrum and the exclusion of the others. To demodulate only one tone, first select the desired tone by using an appropriate filter in the circuit to be simulated. Also, note that because the baseband (DC) spectrum represents a real signal and not a complex envelope, its magnitude corresponds to taking the absolute value of that signal, and its phase is either 0 or 180 degrees.

Circuit Envelope and Frequency-Domain-Defined Devices

As the applications for wireless communications continue to grow, it is no longer possible to satisfy all modeling needs with standard, preconfigured models. Users need methods to define their own nonlinear models in either the time or the frequency domain. The frequency-domain-defined device (FDD) has been developed to allow a user to describe current and voltage spectral values directly, in terms of the algebraic relationships of other voltage and current spectral values.

Another trend in digital communication systems is the issue of timing, since it is increasingly common to encounter subsystems that behave in ways that cannot be modeled as time-invariant. Clocked systems, sampled systems, TDMA pulsed systems, and digitally controlled systems are all becoming more common, even in the RF and microwave area, and behavioral models must be able to include these effects. So, in addition to its frequency-domain modeling attributes, the FDD also allows the modeler to define trigger events, to sample voltages and currents at trigger events, and to generate outputs that are arbitrary functions of either the time of the trigger or of the complex spectral voltage and current values at these trigger events.

Circuit Envelope and Components

In general, any of the components available in Advanced Design System can be used in circuits where envelope simulations are performed. Some components lend themselves especially well to such circuit designs, such as n -state modulators and demodulators or spectral waveform sources.

Using Functions and Equations

Functions and equations can be placed in a schematic in the same manner as ADS components, and some are especially well-suited to envelope simulations and behavioral model designs. There is also a set of functions that can be used only with FDDs, and a set of variables that can be used only with SDDs; they, too, can facilitate digital communication designs where the envelope simulator is applied.

To use the Variables and equations component:

1. From the **Data Items** palette, select **Var eqn** (Variables and equations) and place it in the Schematic window.
2. Edit it and select the **File Based** option from the *Variable or Equation Entry Mode* drop-down list.
3. In the Data Access Component field, type the Instance Name—or select it from the drop-down list—of the DAC that points to the file containing the data of interest.

This makes it possible to use lists of data as component parameter values. Such data can also be used with n -state components to define the phase and amplitude of the various states. These components are especially efficient in envelope simulations, and dataset and list arrays simplify the entering of strings of data as parameter values.

Automatic Verification Modeling (Fast Cosimulation)

Overview

Automatic Verification Modeling (AVM) is a user-selected mode of operation that can significantly speed up formerly lengthy cosimulations of Analog/RF circuits. When this mode is enabled, the analog subcircuit is first characterized using a variety of Harmonic Balance simulations at the start of every Ptolemy simulation. Then during the actual Ptolemy simulation, this characterization data is used to predict the response of the subcircuit instead of performing the full circuit simulation at each time point.

Since this characterization is normally done at the start of every Ptolemy sweep based on the full circuit level schematic, the overall capability is basically the same as if the actual circuit level representation is used throughout the cosimulation. For example, optimization of circuit level parameters, or swept parameters including bias, temperature or swept carrier frequency will continue to operate as expected. These capabilities do not exist when the circuit is manually replaced with behavioral models.

This ability to predict the modulated response based on the Harmonic Balance characterization relies on the fact that many circuits, when used in relatively narrow band modulated applications, can have their nonlinearity represented as a static nonlinearity that is strictly a function of the instantaneous amplitude of the carrier. Many of these circuits, such as amplifiers and mixers, have little, if any, frequency response over the modulation bandwidth of interest. Any frequency response effects that do exist can then often be represented as either a linear filter on either the input or the output of the nonlinearity.

Each output of the Analog/RF subcircuit is then characterized by the following equation:

$$Out_k = (P_k(InputMag) - P_k(0)) \times e^{jInputPhase \times HarmGain} + P_k(0)$$

The $P_k(mag)$ functions are determined by the swept amplitude Harmonic Balance simulation. The HarmGain is the harmonic gain determined from the harmonic indices of the input and output frequencies.

If phase characterization has been enabled, by setting the “Num. of phase pts” parameter to a non-zero value, then each output is characterized by this modified equation:

$$Out_k = P_k(InputMag, InputPhase)$$

In this case, $P_k(mag, phase)$ functions are determined by a two dimensional swept amplitude and swept phase Harmonic Balance simulation.

If the subcircuit nonlinearities are a function the input phase, as in a nonlinear IQ demodulator, then the amplitude only characterization is not accurate and the two dimensional amplitude and phase characterization must be used. However, if the IQ Modem phase characteristics are linear, then the IQ input or output pair can be identified in the Node Names section, and the Magnitude only characterization can still be used, as the HarmGain value is then set to 1. This requires that the I/Q pair be properly identified such that there are no phase inversions introduced, since this would require a harmonic gain of -1.

Note that the magnitude only characterization assumes the output phase can be determined from the harmonic indices of the input and output frequencies. In certain rare cases, this can be ambiguous. For example, if the input frequency is *fund1* and the output frequency is $2*fund1$, then the simulator assumes the output signal is generated by a doubling the input frequency and so the input phase is doubled. However, if this $2*fund1$ output frequency is actually generated by mixing with another LO source at the *fund1* frequency and so the phase relationship is supposed to be linear, then the AVM (Fast Cosim) results will be incorrect. If the mixer LO is operating at an independent *fund2* frequency, with a mixer output at $fund1 + fund2$, then the HarmGain of 1.0 is correctly determined. So, as with the IQ demodulator, if there are circuit sources operating at the same frequency as the input signal, then caution should be used when enabling this AVM (Fast Cosim) mode, and the two dimensional amplitude and phase characterization may be required.

In addition to the swept amplitude characterization, the AVM characterization also includes a small signal Harmonic Balance frequency sweep. In this case, the input amplitude is set to 0, and the small signal frequency is swept between $\pm 0.5/TimeStep$. Note that even though the input amplitude is set to 0, a nonlinear analysis is still being done so any frequency translation effects due to internal mixers will be fully captured. An impulse response representing this frequency response can then be generated, and then, as the user's choice, placed on either the input or the output of the nonlinear block. An additional delay can also be added to the frequency response, so that the impulse can be made a more accurate representation of the frequency response. The user should set the number of frequency points high enough that any frequency response deviations are sufficiently sampled (a minimum of 4 samples every 360 degrees). The maximum duration of the impulse response will be about $0.75/FreqSpacing$, where

$$FreqSpacing = \frac{1}{2^N \times TimeStep}$$

N is determined so that 2^N is just larger than the user-specified number of frequency points. So, if frequency compensation is needed, the number of frequency points should generally be greater than the maximum impulse response time of the circuit around the carrier frequency plus any additional *Delay* specified in the Cosim Implementation block, both normalized by the Circuit Envelope *TimeStep* value.

In addition to the amplitude and frequency response characterization, nonlinear noise characterization is also done. A single value for the equivalent input noise for each output is determined and then added to the input signal prior to the above nonlinear and frequency response effects. In the case of multiple outputs, these equivalent noise sources are uncorrelated. So any correlation of the Analog/RF subcircuit added noise between multiple outputs is lost during this AVM (Fast Cosim) mode. Whether or not this noise characterization is done and implemented is determined by the standard EnvNoise parameter in the Envelope Controller.

In certain cases, the time spent doing this characterization can be eliminated if the user requests that the simulator use previous characterization. Once this mode is selected, then it is the responsibility of the user to make sure that the previous characterization is still valid, and that circuit parameters have not been changed, perhaps by optimization, biases have not changed, carrier frequencies have not changed, etc. The circuit should not have changed its connectivity within the Ptolemy environment and names of the OutputSelectors cannot have changed either. Also, the format and data in the dataset must be in the same expected configuration as when it was written by simulator.

In addition to the characterization and implementation portion of the AVM (Fast Cosim) mode, there is also a user selectable verification step. If the user specifies a non-zero verification Stop Time, then the normal Circuit Envelope simulation is performed in parallel with the fast cosimulation predictions. The error over this verification is then computed and output to determine how well these predictions are performing. If the behavior is unacceptable, as determined by the Accept Tolerance, then the AVM (Fast Cosim) will be disabled and only the normal Circuit Envelope results will be used. Clearly, if used, this verification time should be set long enough to include a representative portion of the input signal. This may need to take into account the fact that many sources, due to channel filtering, take a while to generate their full amplitude outputs.

Limitations

Though you may have selected the AVM (Fast Cosim) mode, it may be disabled for the following reasons:

- Verification was enabled but performance did not meet acceptance level.
- There was more than one input to the Analog/RF subcircuit from Ptolemy and the user did not specify the active input node name.
- The Envelope analysis includes an oscillator analysis.
- The input frequency was a mixing term and not harmonically related to a single fundamental.
- One of the Envelope OutSelectors was in AllPass mode.
- The input frequency is baseband and does not have a non-zero carrier frequency and no IQ input pairs were identified.
- The input frequency does not exactly match an Envelope analysis frequency.
- There was a problem reading the previously generated dataset.

Additional limitations of the AVM (Fast Cosim) mode, that may not be automatically detected by the simulator unless verification is enabled, include:

- Envelope analyses parameters such as the Freq parameters should not be swept or optimized.
- Do not use the AllPass mode in the EnvOutSelector component when using AVM (Fast Cosim) mode.

Envelope Simulation Controller

The Envelope Simulation controller enables you to define these aspects of the simulation:

- **Env Setup** sets parameters related to time and frequency.
- **Env Params** selects an integration mode and sweep offset, turns on source and resistor noise, and sets device-fitting parameters.
- **Oscillator Analysis** sets parameters for analyzing oscillators.
- **HB Params** sets parameters related to convergence, status and device operating point levels, and FFT oversampling.
- **Small-Sig** sets parameters related to small-signal/large-signal simulation.
- **Noise (1)** sets parameters related to noise simulation, including sweeps. Input and output ports can be defined here.
- **Noise (2)** selects nodes at which to calculate noise data, and sorts the noise contributors. Port noise options are provided here also.
- **Solver**—Choose between a Direct or Krylov solver or an automatic selection. For details about choosing solvers, see the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.
- **TAHB** enables and sets up parameters for automated transient assisted harmonic balance to provide a transient initial guess for the underlying harmonic balance simulation. For details, see the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.
- **Cosim (in ADS) / Automatic Verification Modeling (in RFDE)**—Sets parameters related to the Automatic Verification Modeling (Fast Cosimulation) mode.
- **Output**—In ADS, selectively save simulation data to a dataset. For details, refer to the topic “Selectively Saving and Controlling Simulation Data” in the chapter “Simulation Basics” in the *Using Circuit Simulators* documentation.

In RFDE, use *Outputs > Save Options* in the Analog Design Environment window.
- **Display (in ADS)**—Control the visibility of simulation parameters on the Schematic. For details, refer to the topic “Displaying Simulation Parameters on the Schematic” in the chapter “Simulation Basics” in the *Using Circuit Simulators* documentation.

Env Setup

Following are details on setting up the parameters related to time and frequency. Simulator parameter names, as they appear in netlists and ADS schematics, are in parentheses.

Table 1-2. Env Simulation Freq Options

| Env Setup | |
|--|--|
| Times (in ADS) Time Setup (in RFDE) | |
| Stop time (Stop) | The time the analysis stops. |
| Time step (Step) | Sets the fixed time step that the simulator uses to calculate the time-varying envelopes. |
| Note: The parameter Time step defines the maximum allowed bandwidth (0.5 /Time step) of the modulation envelope. Because of the nature of the time-domain integration algorithms, the analysis bandwidth (1/Time step) should be at least twice as large as the modulation bandwidth to achieve accurate simulations at the maximum modulation frequencies. Stop time simply defines the maximum duration of the swept time simulation. An analysis starts at time = 0, so the total number of simulation time points that are stored is equal to (Stop time)/Time step. At each time point, the envelope values of all of the analysis frequencies, including DC, are saved. | |
| Fundamental Frequencies (in ADS) Fundamental Tones (in RFDE) | |
| Maximum order (MaxOrder) | <p>The maximum order of the intermodulation terms in the simulation. The combined order is the sum of the individual frequency orders that are added or subtracted to make up the frequency list. For example, assume there are two fundamentals and Order (see below) is 3.</p> <p>If Maximum order is 0 or 1, no mixing products are simulated. The frequency list consists of the fundamental and the first, second, and third harmonics of each source.</p> <p>If Maximum order is 2, the sum and difference frequencies are added to the list.</p> <p>If Maximum order is 3, the second harmonic of one source can mix with the fundamental of the others, and so on.</p> |
| Edit (in ADS) | Edit the Frequency and Order fields, then use the buttons to Add the frequency to the list displayed under Select. |
| Frequency (Freq[n]) | The frequency of the fundamental(s). Change by typing over the entry in the field. Select the units (None, Hz, kHz, MHz, GHz) from the drop-down list. |

Table 1-2. Env Simulation Freq Options (continued)

| Env Setup | |
|---|--|
| Order (Order[n]) | The maximum order (harmonic number) of the fundamental(s) that will be considered. Change by typing over the entry in the field. |
| Select (in ADS) | <p>Contains the list of fundamental frequencies. Use the Edit field to add fundamental frequencies to this window.</p> <p>Add -- Allows you to add an item. Cut -- Allows you to delete an item. Paste -- Allows you to take an item that has been cut and place it in a different order.</p> |
| Fundamental Tones (in RFDE) | <p>Add and edit fundamental tones by specifying these parameters:</p> <p>Frequency-The frequency of the fundamental(s).</p> <p>Order-The maximum order (harmonic number) of the fundamental(s) that will be considered.</p> <p>Oversample-Sets the FFT oversampling ratio. Higher levels increase the accuracy of the solution by reducing the FFT aliasing error and improving convergence. Memory and speed are affected less when the direct harmonic balance method is used than when the Krylov option is used.</p> <p>Add enters the values from the input fields into the table. Delete removes selected tone from the table. Update changes tone selected in table with contents of the input fields. Clear removes contents of the input fields.</p> |
| Small-signal (SS_MixerMode) (in ADS)—Choose this if you want a small-signal analysis. Used only with the Small-Sig tab. | |
| Nonlinear noise (NLNoiseMode) (in ADS)—Choose this if you want a nonlinear noise analysis. Used only with the Noise(1) and Noise(2) tabs. | |

Env Params

Following are details on setting up the Env Params portion of the simulation. Simulator parameter names, as they appear in netlists and ADS schematics, are in parentheses.

Table 1-3. Env Simulation Params Options

| Env Params | |
|---|---|
| Env Params (in ADS) Envelope (in RFDE) | |
| Integration (EnvIntegOrder) | Displays the integration options. |
| Backward Euler (1) | Invokes the backward-Euler integration algorithm. |
| Trapezoidal (2) | Invokes the trapezoidal integration algorithm. Integrates between time points by assuming they are connected by line segments |
| Gear's (UseGear) | Invokes Gear's method. Integrates by assuming that the time points are connected by a polynomial curve. Lower-order polynomials tend to create greater truncation error, while higher-order polynomials can become unstable |
| Sweep offset (SweepOffset) | Disables data output until the time elapsed exceeds the value entered in this field. Then it subtracts the value in the field from the output time. Enter appropriate values in this field to eliminate undesired responses resulting from turn-on transients. |
| Turn on all noise (EnvNoise) | <p>Includes in the simulation the white noise in devices such as resistors, lossy transmission lines, diodes, transistors, etc. This adds independent, white, Gaussian noise at all of the envelope frequencies. Explicit noise sources, such as V_Noise, I_NoiseBD, OSCwPhNoise Amplifier, etc., also add their noise contribution. Full nonlinear circuit equations are applied to the resulting composite signal, so that no small-signal assumptions have to be made about the relative size of the noise, and random times and voltages are added to the simulation. The noise will be complex for non-baseband envelope frequencies, generating both amplitude- and phase-equivalent noise.</p> <p>The noise is generated by a random number generator. It will produce a different sequence of random numbers each time the simulation is run. If a repeatable sequence is required, it can be obtained by setting the simulator variable <code>__randseed</code> to an integer value with a schematic equation. For example, <code>__randseed=12345</code> (two underscores precede randseed)</p> |

Table 1-3. Env Simulation Params Options (continued)

| Env Params | |
|--|---|
| Device Fitting | <p>There are several ways to control the linear device, time-domain modeling required by the circuit envelope simulator when analyzing a modulation envelope. Most built-in elements now have an Laplace or a transmission line approximation. This parameter is used only with respect to dataset devices or generic linear devices whose frequency response cannot be represented as a rational polynomial of the form</p> $e^{-sT} \left(\frac{P(s)}{Q(s)} \right)$ <p>where s is the Laplace variable, T is time delay, and P and Q are the numerator and denominator polynomials, respectively.</p> <p>For linear elements a model must be generated that reflects the envelope frequency response around each of the analysis frequencies. The first three parameters in this area are used in a pole/zero fit of the frequency response around each carrier frequency. The remaining options are used when a valid or sufficiently accurate pole/zero fit cannot be obtained.</p> |
| Bandwidth fraction (EnvBandwidth) | <p>Determines what fraction of the envelope bandwidth to use to determine the fit.</p> <p>The initial value provided for Bandwidth fraction is 1.0. The default value for Bandwidth fraction when the value is left blank is 0.1, so that only the frequency values that lie between $0.5 \times \text{BandwidthFraction} / \text{Timestep}$ around each carrier frequency are used to determine the fit.</p> <p>If greater accuracy is required at the edges of the envelope bandwidth, this number can be increased. However, the simulator will then typically require a higher order and a more time-consuming fit to be generated and then used during the simulation. Also, the integration algorithms cannot maintain 100% percent accuracy out to the edges of the envelope bandwidth.</p> <p>A Bandwidth fraction value of 0.0 will effectively disable this pole/zero fitting, and just the constant value will be used. This will result in the fastest simulation, but any transient effects from these models will not be included. The Relative tolerance and Absolute tolerance parameters (see below) can also be set to help determine how accurate a fit is desired.</p> |
| Relative tolerance (EnvRelTrunc) | Sets a relative truncation factor for envelope fitting. |
| Absolute tolerance (EnvAbsTrunc) | Sets an absolute truncation factor for envelope fitting. |
| Warn when poor fit (EnvWarnPoorFit) | Causes a warning message to appear when an envelope fit is poor. |

Table 1-3. Env Simulation Params Options (continued)

| Env Params | |
|---|---|
| Use fit when poor (EnvUsePoorFit) | Instructs the simulator to use poor fits instead of constant values. |
| Skip fit at baseband (EnvSkipDC_Fit) | <p>Instructs the simulator not to use pole/zero fitting at the baseband (DC) envelope.</p> <p>Note: If an external frequency-domain-device is supplied (such as an n-port data device using a dataset of S-parameter measurements read from an instrument), and that device does not accurately represent the low-frequency or DC response, then a good pole/zero fit may not be obtained. Three of the above parameters determine what to do in these cases. Skip fit at baseband can be used to disable the fitting process at just the DC (baseband) frequencies. Warn when poor fit can be used to disable the output of these warnings. Use fit when poor then determines whether to use these poor fits in the simulation or to replace them with the constant, center frequency value. However, there is a potential risk associated with using poor fits, in that the simulation may generate incorrect, possibly unstable results.</p> |
| Small-signal (SS_MixerMode) (in ADS)—Choose this if you want a small-signal analysis. Used only with the Small-Sig tab. | |
| Nonlinear noise (NLNoiseMode) (in ADS)—Choose this if you want a nonlinear noise analysis. Used only with the Noise(1) and Noise(2) tabs. | |

Oscillator Analysis

Following are details on setting up the Oscillator Analysis portion of the simulation. Simulator parameter names, as they appear in netlists and on ADS schematics, are in parentheses.

Table 1-4. Env Simulation Oscillator Analysis Options

| Oscillator Analysis | |
|---|--|
| Enable Oscillator Analysis (in ADS) Oscillator (in RFDE) (OscPortName) | This causes a normal harmonic balance simulation to be performed prior to the first time step. This is used to determine and set the analysis frequency to the steady-state oscillator frequency. Select this option to simulate a circuit containing an oscillator. |
| Method | The <i>Use Oscport</i> method should be selected if the circuit contains an OscPort or OscPort2. The <i>Specify Nodes (ADS)</i> / <i>Specify Oscillator Nodes (RFDE)</i> method should be selected if the circuit is an oscillator and does not contain an OscPort or OscPort2. |
| Specify Oscillator Nodes The following parameters are available only when selected Method is <i>Specify Nodes (ADS)</i> / <i>Specify Oscillator Nodes (RFDE)</i> . | |
| Node Plus (OscNodePlus) in ADS (Node[1]) in RFDE | This is the required name of a named node in the oscillator. Recommended nodes are those at the input or output of the active device, or in the resonator. Hierarchical node names are permitted. |
| Node Minus (OscNodeMinus) in ADS (Node[2]) in RFDE | This second node name should only be specified for a differential (balanced) oscillator. Leave it blank for single-ended oscillators. <i>Node Plus</i> and <i>Node Minus</i> should be chosen symmetrically. Hierarchical node names are permitted. |
| Fundamental Index (OscFundIndex) in ADS (FundIndex) in RFDE | Specifies which of the fundamental frequencies is to be treated as the unknown oscillator frequency which the simulator will solve for. The default value of 1 means that Freq[1] is the unknown. |
| Harmonic Number (OscHarmNum) in ADS (Harm) in RFDE | Specifies which harmonic of the fundamental frequency is to be used for the oscillator. Normally this parameter stays at its default setting of 1. If an oscillator followed by a frequency divider is to be analyzed, this parameter should be set to the frequency divider ratio. |
| Octaves to Search (OscOctSrch) in ADS (NumOctaves) in RFDE | Is used in the initial frequency search during oscillator analysis. This many octaves are searched, centered around the frequency specified by the user on the Freq tab. To skip the initial frequency search, provide a good initial guess of the frequency on the Freq tab and set this parameter to zero. |

Table 1-4. Env Simulation Oscillator Analysis Options (continued)

| Oscillator Analysis | |
|---|---|
| Steps per Octave (OscOctStep) in ADS (Steps) in RFDE | Specifies the number of steps per octave used in the initial frequency search. A high-Q oscillator may require a much larger value, such as 1000, in order for the search to find the phase shift at resonance. |
| Calculate oscillator startup transient (ResetOsc) | <p>This option resets the oscillator voltage solution to zero so that the transient buildup can be simulated. If this option is not selected, the time-domain solution begins at the steady-state solution, the transient buildup time is skipped, and the oscillator can immediately start responding to any external modulation.</p> <p>Note: The OscPort, if present, is used only for this initial simulation. It is disabled once the actual envelope simulation starts.</p> |

Defining HB Simulation Parameters

Defining the HB simulation parameters consists of the following basic parts:

- Specifying the desired level of detail in the simulation status summary
- Specifying the amount of device operating-point information to save

Simulator parameter names, as they appear in netlists and ADS schematics, are in parentheses.

Table 1-5. Envelope Simulation HB Parameter Options

| HB Params | |
|-------------------------------|---|
| Convergence | |
| Max. iterations (MaxIters) | The maximum number of iterations to be performed. The simulation will iterate until it converges, an error occurs, or this limit is reached. |
| Pack FFT (PackFFT) | Controls the frequency map packing for multitone Harmonic Balance. By default, when it is not explicitly set to <i>yes</i> or <i>no</i> by the user, the simulator disables it (sets it to <i>no</i>) when there are three or fewer tones (fundamental frequencies), and enables it when there are four or more tones. Setting PackFFT to <i>yes</i> may improve the simulation speed and reduce memory consumption by using a smaller number of time samples (smaller FFTs), but at the potential loss of dynamic range and accuracy due to the aliased harmonics of the first fundamental now possibly landing on various mixing tones. For mixers and other applications with a single large dominant (LO) tone, that frequency should be assigned to the first fundamental and PackFFT should be left at the default setting of <i>no</i> so that any aliased harmonics of this large signal will just land on its own harmonics and not on mixing terms. If you are simulating mixer intermodulation or third-order intercept, it is recommended that this parameter be set to <i>no</i> to achieve the most accurate results. (To set this parameter in ADS, check it on the Display tab, then set it to <i>yes</i> or <i>no</i> directly on the schematic.) |
| Guard Threshold (GuardThresh) | The Jacobian matrix from the direct solver within the Newton solver is a block matrix. A block matrix is a matrix whose elements are matrices and vectors. The blocks of the Jacobian are truncated to a specified threshold by default. The default threshold (bandwidth) is set by Guard Threshold, and its default value is 10 ⁻⁴ . This bandwidth truncation speeds up the Jacobian factorization and saves memory, but can lead to convergence problems due to an inaccurate Newton direction. In order to get the full bandwidth of the Jacobian blocks and improve the convergence, set Guard Threshold=0. (In ADS, display and set this parameter directly on the schematic.) |

Table 1-5. Envelope Simulation HB Parameter Options (continued)

| HB Params | |
|---|--|
| Samanskii Constant (SamanskiiConstant) | Controls how frequently the Jacobian is constructed and factored rather than being reused. The default SamanskiiConstant is 2, and it will yield faster computation times due to fewer Jacobian factorizations. If the Direct solver fails to converge, then set SamanskiiConstant=0 so that the Jacobian will be computed at each iteration and will not get reused for future iterations. A * next to an iteration number in the status window output indicates a re-computation of the Jacobian for that iteration. (In ADS, display and set this parameter directly on the schematic.) |
| Arc Max Step (ArcMaxStep) | Limits the maximum size of the arc-length step during arc-length continuation. In the arc-length continuation, the arc-length is increased in steps. The step size is calculated automatically for each problem. However if the ArcMaxStep is specified and is nonzero, it will define an upper-limit for the size of the arc-length step. The default is 0 which means there is no upper limit for the ArcMaxStep. (In ADS, display and set this parameter directly on the schematic.) |
| Arc Level Max Step (ArcLevelMaxStep) | Limits the maximum arc-length step size for source-level continuation. The default is 0 which means there is no limit for the ArcLevelMaxStep. (In ADS, display and set this parameter directly on the schematic.) |
| Arc Min Value (ArcMinValue) | Set relative to ArcMaxValue. ArcMinValue determines the lower limit that is allowed for the continuation parameter p during the simulation. In the arc-length continuation, p can trace a complicated manifold and its value can vary non-monotonically. ArcMinValue specifies a lower bound for p such that if during the arc-length continuation, p becomes smaller than ArcMinValue, the simulation is considered to have failed to converge. The default is $p_{min} - \text{delta}$, where delta is $p_{max} - p_{min}$. p_{min} is the lower end of the parameter sweep, and p_{max} is the upper end of the parameter sweep. (In ADS, display and set this parameter directly on the schematic.) |
| Arc Max Value (ArcMaxValue) | Set relative to ArcMinValue. ArcMaxValue determines the allowed upper limit of the continuation parameter p during the simulation. In the arc-length continuation, p can trace a complicated manifold and its value can vary non-monotonically. ArcMaxValue specifies an upper bound for p such that if during the arc-length continuation, p becomes greater than ArcMaxValue, the simulation is considered to have failed to converge. The default is $p_{max} + \text{delta}$, where delta is $p_{max} - p_{min}$. p_{min} is the lower end of the parameter sweep, and p_{max} is the upper end of the parameter sweep. (In ADS, display and set this parameter directly on the schematic.) |

Table 1-5. Envelope Simulation HB Parameter Options (continued)

| HB Params | |
|--|---|
| Max Step Ratio (MaxStepRatio) | Controls the maximum number of continuation steps (default is 100). (In ADS, display and set this parameter directly on the schematic.) |
| Max Shrinkage (MaxShrinkage) | Controls the minimum size of the arc-length step (default is 1e-5). (In ADS, display and set this parameter directly on the schematic.) |
| Levels (in ADS) Annotation (in RFDE) Enables you to set the level of detail in the simulation status report. | |
| Status level (StatusLevel) | <p>Prints information about the simulation in the Status/Summary part of the Message Window. A value of 0 causes no or minimal information to be reported, depending on the simulation engine. Higher values print more detail.</p> <p>The type of information printed may include the sum of the current errors at each circuit node, whether convergence is achieved, resource usage, and where the dataset is saved. The amount and type of information depends on the status level value and the type of simulation.</p> <p>Note: To view a report of the simulator's progress in the Status/Summary window while the simulation is running, set Status level to 3.</p> |
| Device operating point level (DevOpPtLevel)—Enables you to save all the device operating-point information to the dataset. In ADS, if this simulation performs more than one Env analysis (from multiple Env controllers), the device operating point data for all Env analyses will be saved, not just the last one. Default setting is None. | |
| None (None) | No information is saved. |
| Brief (Brief) | Saves device currents, power, and some linearized device parameters. |
| Detailed (Detailed) | Saves the operating point values which include the device's currents, power, voltages, and linearized device parameters. |
| FFT (in ADS) | |
| Oversample (FundOversample) | Sets the FFT oversampling ratio. Higher levels increase the accuracy of the solution by reducing the FFT aliasing error and improving convergence. Memory and speed are affected less when the direct harmonic balance method is used than when the Krylov option is used. |

Table 1-5. Envelope Simulation HB Parameter Options (continued)

| HB Params | |
|----------------------------------|---|
| More... (Oversample[n]) | Displays a small dialog box. To increase simulation accuracy, enter in the field an integer representing a ratio by which the simulator will oversample each fundamental. |
| Initial Guess | |
| Use Initial Guess (UseInFile) | <p>Check this box to enter a file name for a solution to be used as initial guesses. This file is typically generated from a previous simulation by enabling <i>Write Final Solution</i>. If no initial guess file name is supplied, a default name is generated internally, using the design name and appending the suffix <i>.hbs</i>. A suffix is neither required nor added to any user-supplied file name. For example, if you have saved the Harmonic Balance solution from a previous simulation, you can later do a nonlinear noise simulation and use this saved solution as the initial guess, removing the time required to recompute the nonlinear Harmonic Balance solution. Or you could quickly get to the initial Harmonic Balance solution, then sweep a parameter to see the changes. In this latter case, you will probably either want to disable the <i>Write Final Solution</i> option or use a different file name for the final solution to avoid overwriting the initial guess solution. (See “Write Final Solution (UseOutFile)” on page 1-35.)</p> <p>The Annotate value specified in the DC Solutions tab in the Options block is also used to control the amount of annotation generated when there are topology changes detected during the reading of the initial guess file. Since HB simulations also utilize the DC solution, to get optimum speed-up, both the DC solution and the HB solution should be saved and re-used as initial guesses. See <i>DC Simulation</i>.</p> <p>The initial guess file does not need to contain all the HB frequencies. For example, one could do a one-tone simulation with just a very nonlinear LO, save that solution away and then use it as an initial guess in a two tone simulation. The exact frequencies do not have to match between the present analysis and the initial guess solution. However, the fundamental indexes should match. For example, a solution saved from a two tone analysis with Freq[1] = 1GHz and Freq[2] = 1kHz would not be a good match for a simulation with Freq[1] = 1kHz and Freq[2] = 1 GHz.</p> <p>If the simulator cannot converge with the supplied initial guess, it then attempts to a global node-setting by connecting every node through a small resistor to an equivalent source. It then attempts to sweep this resistor value to a very large value and eventually tries to remove it.</p> |
| File (InFile) | Specify a filename to save results. |

Table 1-5. Envelope Simulation HB Parameter Options (continued)

| HB Params | |
|---|---|
| Final Solution | |
| Write Final Solution (UseOutFile) | <p>Check this box to save your final HB solution to the output file. If a filename is not supplied, a file name is internally generate using the design name, followed by an .hbs suffix. If a file name is supplied, the suffix is neither appended nor required. If this box is checked, then the last HB solution is put out to the specified file. If this is the same file as that used for the Initial Guess, this file is updated with the latest solution.</p> <p>Transient simulations can also be programmed to generate a harmonic balance solution that can then be used as an initial guess for an HB simulation. Refer to <i>Harmonic Balance Simulation</i>.</p> |
| File (OutFile) | Specify a filename to save results. |
| Small-signal (SS_MixerMode) (in ADS)—Choose this if you want a small-signal analysis. Used only with the Small-Sig tab. | |
| Nonlinear noise (NLNoiseMode) (in ADS)—Choose this if you want a nonlinear noise analysis. Used only with the Noise(1) and Noise(2) tabs. | |
| Other | <p>Check this box to enable access to hidden parameters. The text field is enabled to allow assigning values to the parameters. The format is Other=HiddenParameter1=value HiddenParameter2=value... Hidden parameters are used typically when troubleshooting convergence problems. (In ADS, display and set parameter directly on the schematic.)</p> |

Small-Signal Frequencies

Refer to the topic “Setting up Small-Signal Simulations in ADS” or “Setting up Small-Signal Simulations in RFDE” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.

Noise(1) Tab in ADS

Refer to the topic “Setting up Parameters for Noise(1) Tab in ADS” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.

Noise(2) Tab in ADS

Refer to the topic “Setting up Parameters for Noise(2) Tab in ADS” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.

Nonlinear Noise Setup in RFDE

Refer to the topic “Setting up Nonlinear Noise Parameters in RFDE” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.

Solver

Refer to the section “Selecting a Harmonic Balance Technique” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation.

TAHB

This enables automated transient assisted harmonic balance. TAHB provides a transient initial guess for the underlying harmonic balance simulation at the first time point of a circuit envelope simulation. Refer to “Enabling Transient Assisted Harmonic Balance” in the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation. For additional information about using TAHB, see the chapter “Transient Assisted Harmonic Balance”, in the *Harmonic Balance Simulation* documentation.

Cosim (in ADS)

Automatic Verification Modeling (in RFDE)

These parameters enable and control the Automatic Verification Modeling (Fast Cosimulation) mode and are only applicable when the Envelope Simulation Controller is being used in a Ptolemy cosimulation. Simulator parameter names, as they appear in netlists and ADS schematics, are in parentheses.

Table 1-6. Envelope Cosim Parameter and WTB AVM Options

| Cosim Params | |
|---|---|
| Mode | |
| Enable AVM (Fast Cosim) (in ADS) (ABM_Mode) Automatic Verification Modeling (AVM) checkbox (in RFDE) | This enables the Automatic Verification Modeling (Fast Cosimulation) mode to be used for the Analog/RF subcircuit. If AVM (Fast Cosim) is not possible for this subcircuit, then a warning will be output and regular Circuit Envelope Cosimulation will be performed. |
| Characterization | |
| Max Input Power (ABM_MaxPower) | This specifies the maximum input power to this Analog/RF subcircuit that will be used during the AVM (Fast Cosim) characterization phase. Excessively high values will take longer to characterize due to potentially more difficult circuit convergence. If the input power during the cosimulation exceeds this value, a warning will be generated since the AVM (Fast Cosim) results will no longer be accurate. |
| Num. of amp. pts. (ABM_AmpPts) | This sets the number of linear amplitude points between 0 and the full scale value defined by the Max Input Power. Depending on how much variation there is in the output vs. input amplitude characterization, more amplitude points may be needed to achieve optimum accuracy at a cost of additional characterization time. Due to the continuation nature of the swept amplitude harmonic balance characterization when not using Krylov modes, the cost of additional amplitude points is usually small. In addition to these linear spaced points, the characterization adds an additional power point every 6 dB down to a value 100 dB below the Max Input Power. |

Table 1-6. Envelope Cosim Parameter and WTB AVM Options

| Cosim Params | |
|---|---|
| Num. of phase pts. (ABM_PhasePts) | Any value greater than 0 will enable the characterization to be done as a function of both amplitude and phase. This specifies the number of phase points to be used at each amplitude point. Since this will now be a two dimensional sweep and so will be slower, it should only be used when required, such as with IQ demodulators where the output is a nonlinear function of the input phase. IQ Modems that are linear with phase, but nonlinear with amplitude, do not require phase characterization. Just identify the I/Q pair with the correct polarity in the Node Names section. |
| Num. of freq pts. (ABM_FreqPts) | This sets the minimum number of small signal frequency points that are used to characterize the Analog/RF subcircuit. The actual number of points is increased to the next highest power of 2 value. These points are spaced between +/- 0.5/TimeStep, where TimeStep is the Step time defined in the Envelope controller. The maximum impulse duration for this frequency response characterization is determined by this frequency spacing. So the number of frequency points should be greater than the maximum impulse response time of the circuit around the carrier frequency plus any additional Delay specified in the Implementation block, both normalized by the Circuit Envelope TimeStep value. |
| Use previous data (in ADS) (ABM_ReUseData) Use Previous Characterization (in RFDE) | Checking this tell the simulator to re-use any previous characterization that was done for this Analog/RF subcircuit. This characterization is saved in a dataset named after the subcircuit name. This eliminates any overhead time associated with the characterization, but it is then the responsibility of the user to make sure that nothing significant enough has changed (including carrier frequency, time step, bias voltages, temperature, optimization variables, etc.) since the last characterization. |

Table 1-6. Envelope Cosim Parameter and WTB AVM Options

| Cosim Params | |
|--|---|
| Implementation | |
| Freq. compensation (ABM_FreqComp) | This specifies whether or not a frequency compensation filter is to be created for use in the AVM (Fast Cosim) mode. In addition, the user can specify whether this filter is best placed on the input or the output of the nonlinear block. If the modulation is sufficiently narrow that there is not significant frequency response over the envelope bandwidth, then None should be selected. If the frequency response is primarily due to input filtering or transistor bandwidth limitations, then an Input frequency compensation should perform the best. Similarly, if the dominant filtering is at the output of Analog/RF subcircuit, such as the channel filter, then an Output frequency compensation should be used. |
| Delay (ABM_Delay) | This adds additional transit delay to all the outputs of the Analog/RF subcircuit. In cases where this absolute delay is not critical to the overall system simulation, adding additional delay permits more accurate impulse implementation of the frequency response. This delay should not exceed half the impulse length, as determined by the frequency response characterization. |
| Verification | |
| Stop Time (in ADS) (ABM_VTime) | If this verification stop time is not zero, then both the normal Envelope cosimulation and the AVM (Fast Cosim) results are computed. The RMS error between these two results is computed and output after this verification time has ended. This gives an indication as to how well the AVM (Fast Cosim) is matching the Circuit Envelope results. |
| Verification Time (in RFDE) | |
| Accept Tolerance (in ADS) (ABM_VTol) | If the Verification Stop Time has been set, then the resultant RMS error must be less than this value or else the AVM (Fast Cosim) will be turned off and just the normal Envelope cosimulation results will be used for the remainder of the Ptolemy simulation. The stop time must be large enough to account for turn-on delays of filters and to give a sufficiently representative sample of the normal input signal. |
| Verification Acceptance Tolerance (in RFDE) | |

Table 1-6. Envelope Cosim Parameter and WTB AVM Options

| Cosim Params | |
|---|--|
| Node Names | |
| Active Input (in ADS) (ABM_ActiveInputNode) | When multiple cosimulation inputs exists, only one (or one I/Q pair) can be active. Enter the node name of the active input here. Do not use any node name in the subcircuit input, but use the node name defined at the higher circuit level. If this is an I/Q pair input, then just use either the I or Q node name. Any non-active inputs will be monitored for activity and a warning generated if they are not truly static during a Ptolemy sweep. |
| Dynamic Input Node Name for Characterization (in RFDE) | |
| IQ Pair (in ADS) (ABM_IQ_Nodes[n]) | If multiple inputs or outputs correspond to an IQ pair, one pair can be defined here. Enter the I node name and the Q node name, as defined in the higher circuit level, separated by a space. If more than one IQ pair exists, use the Other = parameter in the Display tab, and use ABM_IQ_Nodes="<I node> <Q node>". Note that for IQ Modems that are linear with respect to phase, phase characterization is not required if the I/Q pair is properly identified here. |
| Node Names of Baseband IQ Pairs (in RFDE) | |

Troubleshooting a Simulation

This section presents suggestions for using this simulator and improving the accuracy of results.

Increasing Accuracy Easily

In general, accuracy is increased and aliasing reduced when the signal of greatest amplitude is entered as the Freq[1] parameter, and lower-amplitude signals are entered as Freq[2], Freq[3], and so on.

Improving Mixer IMD Measurement Accuracy and Speed

Prior to the invention of the Krylov subspace solver (available in harmonic balance simulation), Circuit Envelope was the recommended choice for mixer IMD (intermodulation distortion) measurements. Now the Krylov option appears to provide faster solutions for this application. Nevertheless, when the envelope simulator is used, there are several ways to improve accuracy and speed. Also, *Maximum order* can have a profound effect on simulation results; in general, set *Maximum order* to a low value and increase it until the simulation produces marginal differences in results.

Using the Circuit Envelope Simulator to Analyze an Oscillator

This simulator can support the simulation of oscillator circuits and subsystems, through the *Oscillator* analysis options. The oscillator options work as follows:

- If *Oscillator* analysis is disabled, the envelope simulation proceeds as normal and any oscillations, if present, will build up from some initial value. The analysis frequencies and time step must be set to ensure that the oscillator's instantaneous frequency is well within one of the envelope bandwidths being simulated. This can be either the baseband (DC) envelope or any of the fundamental, harmonic, or mixing-term analysis-frequency envelopes. Multiple oscillations can be simulated as well, as long as each one is within an analysis envelope.

- When the transient buildup is being simulated, a tickler voltage is usually needed in the same envelope as the actual oscillator frequency, to define an initial start-up value. This value can be set by a low-level transient source that is disabled after the first few time samples. Alternatively, a noise source or resistor noise can also be used. For the oscillator control elements, the option *Turn on all noise* is enabled. This will cause devices with noise in the oscillator loop to generate the white noise that will provide the required tickler stimulus.
- If *Oscillator* analysis is enabled, then a normal harmonic-balance oscillator simulation is automatically performed prior to the first time step (refer to the chapter “Harmonic Balance Basics” in the *Harmonic Balance Simulation* documentation). The results of this simulation determine and set the analysis frequency value to the steady-state oscillator frequency. If the oscillation frequency does not change significantly during its buildup, this often allows larger time steps to be used. This is because the envelope bandwidth does not have to include any uncertainty as to the oscillator’s final frequency value.
- If *Oscillator* analysis is enabled, you may reset the oscillator voltage solution to zero by selecting *Calculate oscillator startup transient*, so that the transient buildup can be simulated. If this option is not enabled, the time-domain solution begins at the steady-state solution, the transient buildup time is skipped, and the oscillator can immediately start responding properly to any external modulation. Note that *Oscillator* analysis affects only the initial simulation, and is disabled once the envelope simulation begins.

Using the Circuit Envelope Simulator to Analyze Noise

Noise in envelope simulations can be added as random time voltages and currents. Noise can originate from circuit components such as resistors or noise sources. The full, nonlinear circuit equations are applied to this composite signal of random voltages and currents, so no small-signal assumptions about the relative size of the noise are required.

Setting the option *Turn on all noise* (under the *Env Params* tab) enables white noise from many devices in the circuit to be included in the simulation. The noise sources that are included in the circuit are also included in the simulation. They add white, Gaussian noise at all frequencies in the analysis envelope. The noise voltages will be complex for non-baseband envelope frequencies, generating both amplitude and phase equivalent noise.

Convolution Techniques Used in Circuit Envelope

In Circuit Envelope, the response within the envelope bandwidth ($1/\text{timestep}$) around each of the Harmonic Balance frequencies (including DC) needs to be represented in the time domain. The advantage here over a transient simulation is that the entire frequency response need not be represented in the time domain, and if the envelope bandwidth is relatively small, then a simple fit or a constant is sufficient to represent the response. And since a different constant could be used for each carrier, it maintains good accuracy at the exact Harmonic Balance frequencies.

A consequence of this method is that any components which are characterized in the frequency domain need to have their response converted to the time domain for the portions of their frequency response which fall within the envelope bandwidth.

By default, the Circuit Envelope simulator uses recursive convolution for this task, which first tests to see if the frequency response is described by a rational polynomial. If it is not, the simulator then tries to fit the frequency response to a rational polynomial (with no delay) around each carrier. If this fails, a poor fit warning is generated and the simulation will either proceed with the best fit or just use the center frequency constant, depending on the setting in the analysis control.

Impulse convolution can be used, but it is currently only available for certain components such as S1P_Eqn, Y1P_Eqn, or Z1P_Eqn. These components have an *ImpDeltaFreq* parameter, and an appropriate value must be entered for the parameter to use impulse convolution. However, recursive convolution has priority over impulse convolution, even for such components, if the frequency response is described by a rational polynomial.

Index

A

- advantages, 1-2
 - over Harmonic Balance, 1-2
- applications, 1-1
- Automatic Verification Modeling (fast cosimulation)
 - limitations, 1-22
 - overview, 1-19
 - parameters, 1-37

B

- bandwidth fraction
 - in circuit envelope simulations, 1-27

C

- Circuit Envelope simulations
 - and FDDs, 1-17
 - envelope formulas, 1-16
 - in noise analysis, 1-42
 - of oscillators, 1-41
 - overview, 1-10
 - performing, 1-4
 - troubleshooting, 1-41
 - using V_1Tone source components, 1-15
 - versus harmonic balance, 1-13
- Circuit Envelope simulator
 - bandwidth fraction, 1-27
 - device fitting, 1-27
 - simulation parameters, 1-23
- cosimulation, fast. *See* Automatic Verification Modeling

D

- device fitting
 - in circuit envelope simulations, 1-27

E

- envelope simulations. *See* Circuit Envelope simulator

F

- fast cosimulation. *See* Automatic Verification Modeling
- FDDs
 - using with Circuit Envelope simulations, 1-17

I

- I (in-phase modulation component), 1-14
- IMD (intermodulation distortion), 1-5
 - using circuit envelope simulations, 1-1
- intermodulation distortion (IMD), 1-5

L

- limitations, 1-3

M

- maximum order, 1-24
- mixers
 - finding intermodulation distortion (IMD), 1-5

N

- noise
 - in circuit envelope simulations, 1-26
 - produced by random number generator, 1-26

O

- oscillator simulations
 - and Circuit Envelope simulations, 1-41
 - with circuit envelope simulator, 1-29
- oscillators
 - calculating startup transient, 1-30

Q

- Q (quadrature modulation component), 1-14

R

- random number generator, 1-26

S

- simulation parameters
 - circuit envelope, 1-23

T

- TAHB, 1-36
- third-order intercept (TOI)
 - using circuit envelope simulations, 1-1

V

- V_1Tone source components

in Circuit Envelope simulations, 1-15
verification. *See Automatic Verification*
Modeling