NOTICE: This document contains references to Agilent Technologies. Agilent's former Test and Measurement business has become Keysight Technologies. For more information, go to **www.keysight.com.**

KEYSIGHT
TECHNOLOGIES

Advanced Design System 2011

September 2011
Circuit Envelope Simulation

Acknowledgments

About Circuit Envelope Simulation

This is a description of Circuit Envelope simulation, including when to use it, how to set it up, and the data it generates. Examples are provided to show how to use this simulation. Detailed information describes the parameters, theory of operation, and troubleshooting information.

Circuit Envelope simulation, 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)

In ADS, the *Envelope* simulation controller is available in the Simulation-Envelope palette.

See the following topics for details on Circuit Envelope simulation:

- *Using Circuit Envelope Simulation* (cktsimenv) explains when to use Circuit Envelope simulation, describes the minimum setup requirements, and gives a brief explanation of the Circuit Envelope simulation process.
- *Examples of Circuit Envelope Simulation* (cktsimenv) 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.
- *Limitation of Circuit Envelope Simulation* (cktsimenv) explains the limitations of using Circuit Envelope simulation.
- *Envelope Simulation Parameters* (cktsimenv) provides details about the parameters available in ADS for the Envelope simulation controller.
- *Theory of Operation for Circuit Envelope Simulation* (cktsimenv) is an outline of the simulation process, with specific details of the Circuit Envelope simulator including a user-selected mode that can speed up lengthy cosimulations of Analog/RF circuits.
- *Troubleshooting a Circuit Envelope Simulation* (cktsimenv) offers suggestions on how to improve a simulation.

Using Circuit Envelope Simulation

This section describes when to use Circuit Envelope simulation, how to set it up, and the basic simulation process used to collect data.

License Requirements

The Circuit Envelope simulation uses the Circuit Envelope Simulator license (sim_envelope). You must have this license to run Circuit Envelope simulations. You can work with examples described here and installed with the software without the license, but you will not be able to simulate them.

When to Use Circuit Envelope Simulation

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, assuming most of the frequency spectrum is empty.
- It compromises neither in signal complexity, unlike Harmonic Balance or Shooting Method, nor in 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. The Krylov solver improves this, but it is 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 one-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.

How to Use Circuit Envelope Simulation

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:

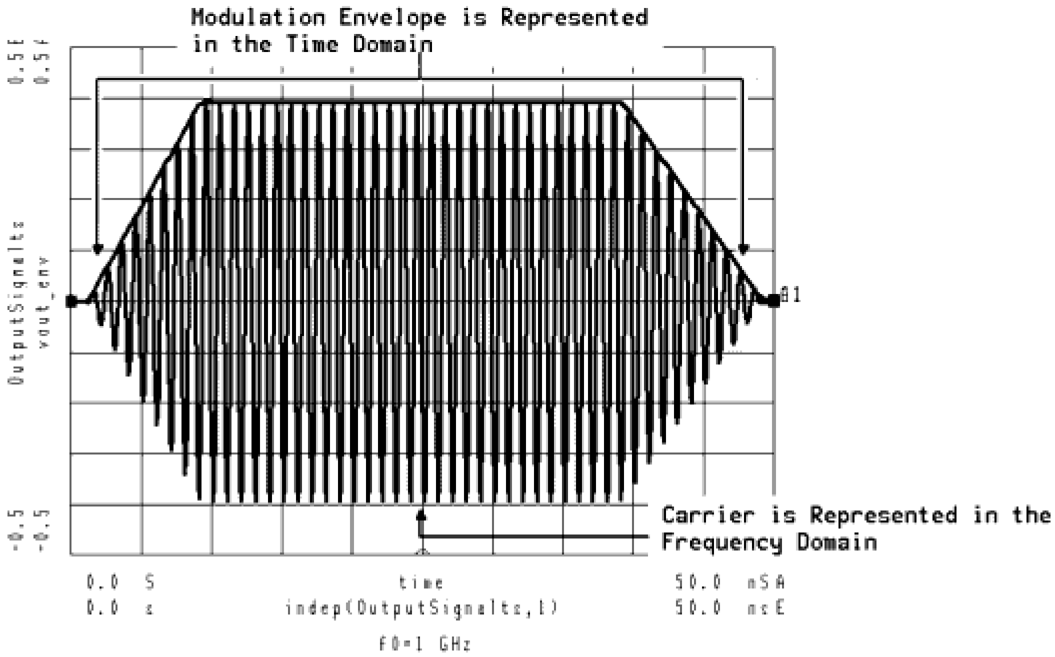
- Use either time domain or frequency domain sources in your circuit. In a circuit employing a mixer, provide a source for the LO.
- Add the Circuit Envelope controller to the schematic. (From the Component palette, choose Simulation-Envelope. Add the ENV component to the schematic.) Double-click to 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 (± 0.5/Time step) of the modulation envelope. The analysis bandwidth (1/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.
- If your design includes an OscPort component, select the Env Oscillator tab and fill in the Oscillator options.
- You can use previous simulation solutions to speed the simulation process. For more information, see *Reusing Simulation Solutions* (cktsimhb).

> **Note**
> Unless there are convergence problems, Agilent EEsof recommends that you leave the other parameters under the Env Params and HB Params set to their default values.

- After the simulation is complete, results appear in the data display window. Envelope data variables are identified by the prefix *ENV*.

What Happens During Envelope Simulation

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. 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 (Modulated signal in the time domain).



Modulated signal in the time domain

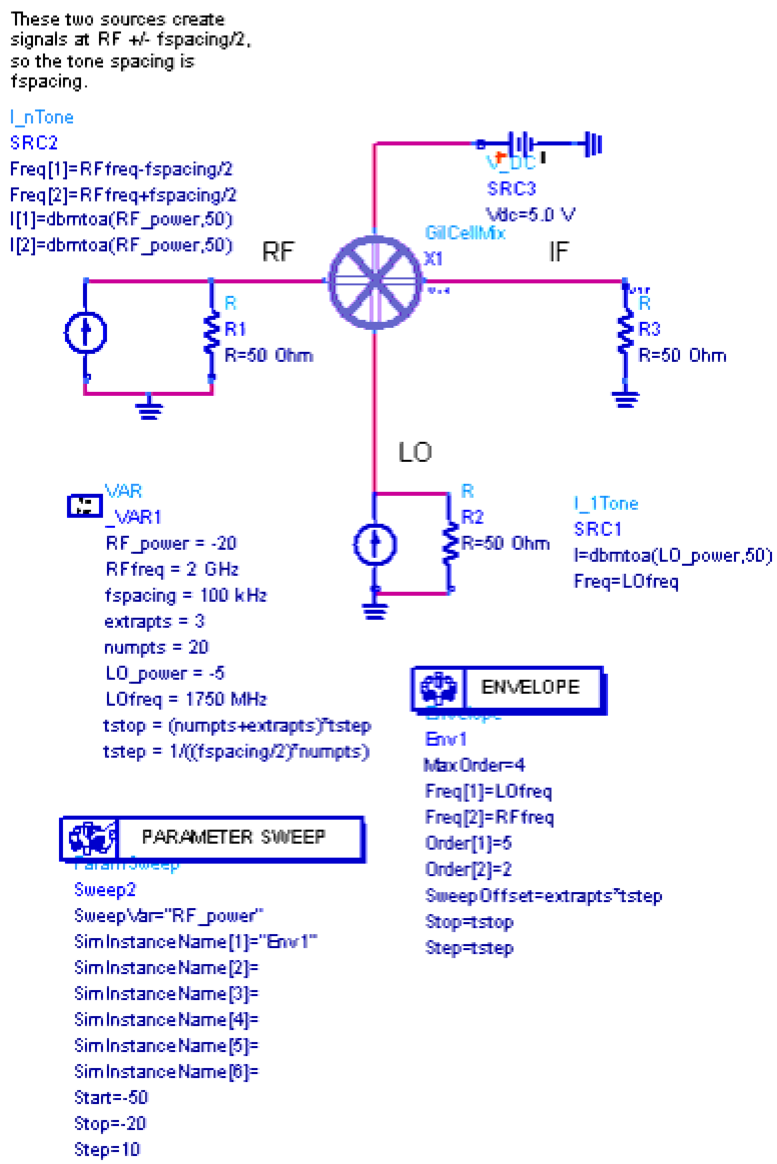
For details about the Envelope simulation process, see *Theory of Operation for Circuit Envelope Simulation* (cktsimenv).

Examples of Circuit Envelope Simulation

The following figure 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.

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



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 kHz*.
- An I_nTone source establishes two intermodulating RF frequencies by means of the following equations:
  Freq[1]=RFfreq- fspacing/2 and Freq[2]=RFfreq+fspacing/2
- An I_1Tone source establishes the LO frequency by means of *Freq=LOfreq*.

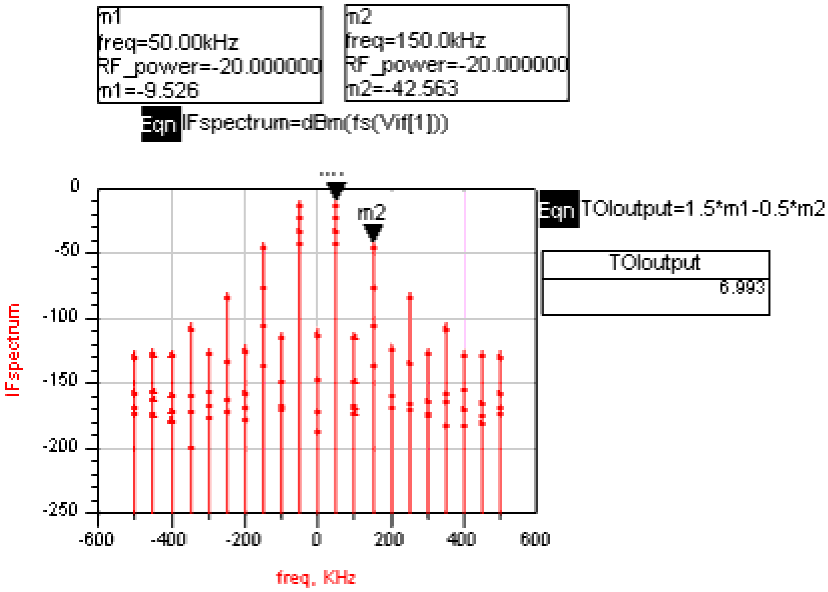
> **Hint**
> Using current sources instead of voltage sources leads to faster simulations, because one fewer equation per source is generated. The function *dbmtoa* converts power to current at a default reference impedance of 50 ohms. P_1Tone and P_nTone components can also be used.

- 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

IFspectrum=dBm(fs(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 *TOIoutput* uses simple geometry.

More Examples

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

- For ways of generating sources for use in envelope simulations (such as Π/4-DQPSK, FSK, QAM, and CDMA), see *Tutorial/ModSources_wrk.*
- To simulate amplifier spectral regrowth and adjacent channel power leakage with digitally modulated RF signals at the input, see *RF_Board/NADC_PA_wrk*.
- To simulate PLL transient responses, see *RF_Board/PLL_Examples/PLL_5th_Order_wrk* and *DECT_LO_Synth_wrk*.

Limitation of Circuit Envelope Simulation

Circuit Envelope contains the following limitation:

Circuit Envelope assumes that the signal can be expressed in time domain as the product of an envelope and a carrier. In frequency domain, the spectrum of the carrier is a discrete grid of frequency components. The spectrum of the envelope is continuous in a limited bandwidth around each frequency component of the carrier. Normally, Circuit Envelope is more efficient than a broadband Transient when the envelope spectra at adjacent carrier frequency components do not overlap. Otherwise, the broadband Transient or SPICE would be a better alternative. Although there are sporadic cases with overlapping spectra where Circuit Envelope still works better than a Transient, Circuit Envelope is not generally recommended for overlapping spectra. Particularly, when there is an oscillator involved, overlapping spectra might cause convergence problems. Also, envelope noise is not rigorously correct when envelope spectra overlap, because noise in the overlapping spectra is double counted.

Envelope Simulation Parameters

ADS provides access to envelope simulation parameters enabling you to define aspects of the simulation listed in the following table:

Tab Name	Description	For details, see...
Env Setup	Sets parameters related to time and frequency, and status level.	Setting Frequencies
Env Params	Selects an integration mode and sweep offset, turns on all model noise, and sets device-fitting parameters.	Defining Envelope Simulation Parameters
Initial Guess	Sets parameters related to initial guess, including automated transient assisted harmonic balance (TAHB), harmonic balance assisted harmonic balance (HBAHB), initial guess from a data file, and initial guess for parameter sweep. It also allows the user to save the final solution in a data file. TAHB provides a transient initial guess for the underlying harmonic balance simulation at the first time point of a circuit envelope simulation.	In *Harmonic Balance Simulation* (cktsimhb), see *Setting Up the Initial Guess* (cktsimhb). For additional information about using TAHB and HBAHB, see *Transient Assisted Harmonic Balance* (cktsimhb) and *Harmonic Balance Assisted Harmonic Balance* (cktsimhb).
Oscillator	Sets parameters for analyzing oscillators.	Enabling Oscillator Analysis
Fast Cosim	Enables the Fast Cosimulation mode and sets related parameters.	Enabling Fast Cosim
Params	Sets device operating point levels and FFT oversampling.	Defining HB Simulation Parameters
Solver	Choose between an automatic selection, or a Direct or Krylov solver. The Auto Select mode is the default and recommended choice.	In *Harmonic Balance Simulation* (cktsimhb), see *Selecting a Harmonic Balance Solver Technique* (cktsimhb).
Noise †	Parameters related to noise simulation, including sweeps, input and output ports, and the nonlinear noise controllers to be simulated.	In *Harmonic Balance Simulation* (cktsimhb), see *Selecting Nonlinear Noise Analysis* (cktsimhb).
Small-Sig †	Sets parameters related to small-signal/large-signal simulation to achieve faster simulations when some signal sources are much smaller than others, and are assumed not to exercise circuit nonlinearities.	In *Harmonic Balance Simulation* (cktsimhb), see *Setting Up Small-Signal Simulations* (cktsimhb).
Output	Selectively save simulation data to a dataset.	*Selectively Saving and Controlling Simulation Data* (cktsim)
Display	Control the visibility of simulation parameters on the Schematic.	*Displaying Simulation Parameters on the Schematic* (cktsim)

† Small-Signal and Noise analysis are performed only after the last Envelope time points, so that the Envelope sweep is allowed to get to a desired operating point, and then perform the standard small signal or noise characterization at that point.

Setting Frequencies

The Env Setup tab involves parameters related to time and frequency, and status levels.

The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Envelope Simulation Env Setup Parameters

Setup Dialog Name	Parameter Name	Description
Times		
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 1+(Stop time/Time step). At each time point, the envelope values of all of the analysis frequencies, including DC, are saved.		
Fundamental Frequencies		
Edit		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.
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		Contains the list of fundamental frequencies. Use the Edit field to add fundamental frequencies to this window. - Add - Enables you to add an item. - Cut - Enables you to delete an item. - Paste - Enables you to take an item that has been cut and place it in a different order.
Maximum mixing order	MaxOrder	The maximum mixing 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. If Maximum mixing 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. If Maximum mixing order is 2, the sum and difference frequencies are added to the list. If Maximum mixing order is 3, the second harmonic of one source can mix with the fundamental of the others, and so on.
Levels		Enables you to set the level of detail in the simulation status report.
Status level	StatusLevel	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. 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. 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.

Defining Envelope Simulation Parameters

The Env Params tab involves selecting an integration mode and sweep offset, turns on all model noise, and sets device-fitting parameters. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Envelope Simulation Env Params

Setup Dialog Name	Parameter Name	Description
Env Params		
Integration	EnvIntegOrder	Displays the integration options.
Backward Euler	EnvIntegOrder=1	Invokes the backward-Euler integration algorithm.
Trapezoidal	EnvIntegOrder=2	Invokes the trapezoidal integration algorithm. Integrates between time points by assuming they are connected by line segments
Gear's	UseGear	Invokes second-order Gear's method.
Sweep offset	SweepOffset	Delays the output of the swept data until the SweepOffset value is reached. It also offsets that value to 0. For example, a sweep to 1 msec with a SweepOffset of 0.6 msec will result in output data with a time axis of 0 to 0.4 msec. This is one reason why this parameter is not called a TimeStart value, as in Transient. The SweepOffset value does not change the start time of transient simulation. Transient simulation begins at time = 0 regardless.
Turn on all noise	EnvNoise	Includes in the simulation the 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 voltages are added to the simulation. The noise will be complex for non-baseband envelope frequencies, generating both amplitude- and phase-equivalent noise. 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 *_randseed_* to an integer value with a schematic equation. For example, *__randseed=12345* (two underscores precede randseed).
Turn on nonlinear noise at every time point	EnvNoiseAtEveryTimePoint	When this parameter is set to yes, nonlinear noise specified in the Noise tab is computed at every envelope time point. When it's set to no, nonlinear noise is only computed at the last time point. Default is no. This parameter is accessible only by using the Other parameter (Other=EnvNoiseAtEveryTimePoint).
Device Fitting		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 $e^{-sT(P(s)/Q(s))}$

		where *s* is the Laplace variable, *T* is time delay, and *P* and *Q* are the numerator and denominator polynomials, respectively. 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.
Bandwidth fraction	EnvBandwidth	Determines what fraction of the envelope bandwidth to use to determine the fit. 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 x BandwidthFraction/Timestep around each carrier frequency are used to determine the fit. 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% accuracy out to the edges of the envelope bandwidth. 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.
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.
Use fit when poor	EnvUsePoorFit	Instructs the simulator to use poor fits instead of constant values.
Skip fit at baseband	EnvSkipDC_Fit	Instructs the simulator not to use pole/zero fitting at the baseband (DC) envelope. 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.

Setting Up the Initial Guess

This enables automated transient assisted harmonic balance (TAHB) and harmonic balance assisted harmonic balance (HBAHB). TAHB provides a transient initial guess for the underlying harmonic balance simulation at the first time point of a circuit envelope simulation.

In the *Harmonic Balance Simulation* (cktsimhb) documentation, see *Setting Up the Initial Guess* (cktsimhb). For additional information about using TAHB and HBAHB, see *Transient Assisted Harmonic Balance* (cktsimhb) and *Harmonic Balance Assisted Harmonic Balance* (cktsimhb).

Enabling Oscillator Analysis

The Oscillator tab involves setting up parameters to analyze oscillators. The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Envelope Simulation Oscillator Analysis Parameters

Setup Dialog Name	Parameter Name	Description
Enable Oscillator Analysis	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 *Use Oscport* method should be selected if the circuit contains an OscPort or OscPort2. The *Specify Nodes* method (OscProbe) 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 *Specify Nodes*.		
Node Plus	OscNodePlus	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	This second node name should only be specified for a differential (balanced) oscillator. Leave it blank for single-ended oscillators. *Node Plus* and *Node Minus* should be chosen symmetrically. Hierarchical node names are permitted.
Fundamental Index	OscFundIndex	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	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	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.
Steps per Octave	OscOctStep	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	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. Note: The OscPort, if present, is used only for this initial simulation. It is disabled once the actual envelope simulation starts.

Enabling Fast Cosim

These parameters enable and control the Fast Cosimulation mode and are only applicable when the Envelope controller is being used in a Ptolemy cosimulation. Fast Cosim parameters are used with Wireless Test Bench (WTB) cosimulation. This mode is also known as Automatic Verification Modeling (AVM). The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Envelope Fast Cosim Parameters and WTB AVM Parameters

Setup Dialog Name	Parameter Name	Description
Enable Fast Cosim	ABM_Mode	This enables the Fast Cosimulation mode to be used for the Analog/RF subcircuit. If Fast Cosim is not possible for this subcircuit, then a warning will be output and regular Circuit Envelope Cosimulation will be performed.
Characterization		
Build Model	ABM_ReUseData= ABM_ReUseData=no	Selecting this activates the *Set Characterization parameters* button, and tells the simulator to use characterization parameter values to build a new model for this Analog/RF subcircuit. This new characterization is saved in a dataset named after the subcircuit name. To open the Characterization Options dialog and change the characterization parameter values, click the *Set Characterization parameters* button.
Use previous data	ABM_ReUseData=yes	Selecting this tells 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.
Characterization Options - Click *Set Characterization parameters* to access the dialog box with these options.		
Max Input Power	ABM_MaxPower	This specifies the maximum input power to this Analog/RF subcircuit that will be used during the 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 Fast Cosim results will no longer be accurate.
Min Mumber of Amplitude Points	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.
Perform Phase Sweep	ABM_PerformPhaseSweep	Select to enable phase sweep using value set for ABM_PhasePts. Enabling this may significantly increase the number of simulations required for characterization and may impact performance. Default is off. When selected ABM_PerformPhaseSweep=yes.
Number 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. If number of points entered is greater than 0 and less than 4, the simulator will change the value to 4 during the simulation.
Number of Frequency 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.
Noise Characterization		
Use the same frequencies as for Small-signal Frequency Response	ABM_NoiseLogScale=no	This is the default mode for noise characterization. Noise simulation is performed at the same frequencies as used for small-signal response around the frequency carrier. Default is selected. When selected ABM_NoiseLogScale=no.
Use independent log sweep	ABM_NoiseLogScale=yes	Select to enable independent log sweep and set values for parameters ABM_NoiseLogStartFreq and ABM_NoiseLogPtsPerDec. Default is unselected (ABM_NoiseLogScale=no). This feature is particularly beneficial in the characterization of 1/f noise, speeding up the characterization phase significantly.
Log Sweep Start Frequency	ABM_NoiseLogStartFreq	If *Use independent log sweep* is selected this parameter establishes the beginning of the frequency sweep. It must be greater than 0.
Number of Points per Decade	ABM_NoiseLogPtsPerDec	If *Use independent log sweep* is selected this parameter establishes the number of points per decade in the logarithmic frequency sweep. The highest frequency in the sweep is determined automatically from the envelope bandwidth.
Model Simulation		
Apply frequency compensation	ABM_FreqComp	This specifies whether or not a frequency compensation filter is to be created for use in the 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. Default is off (ABM_FreqComp=None). When selected, ABM_FreqComp=Input|Output depending on value set for *Place filter at*.
Add delay	ABM_AddDelay	Enables the ABM_Delay parameter and uses the value set for it. Default is off. When selected ABM_AddDelay=yes.
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	ABM_VTime	If this verification stop time is not zero, then both the normal Envelope cosimulation and the 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 Fast Cosim is matching the Circuit Envelope results.
Accept Tolerance	ABM_VTol	If the Verification Stop Time has been set, then the resultant RMS error must be less than this value or else the 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.
Node Names		
Active Input	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.
IQ Pair	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=" *". 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.*

Defining HB Simulation Parameters

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

- Specifying the amount of device operating-point information to save.
- Specifying the FFT oversampling ratio.

The following table describes the parameter details. Names listed in the *Parameter Name* column are used in netlists and on schematics.

Envelope Simulation Params

Setup Dialog Name	Parameter Name	Description
Device operating point level	DevOpPtLevel	Enables you to save all the device operating-point information to the dataset. 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		
Fundamental 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.
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.

Selecting a Solver

Use the Solver parameters to select a convergence mode and solver type. These are the same parameters used to set up the solver for harmonic balance simulations. In the *Harmonic Balance Simulation* (cktsimhb) documentation, see *Selecting a Harmonic Balance Solver Technique* (cktsimhb).

Selecting Noise Analysis

Use the Noise parameters to set up noise analysis including sweeps, input and output ports, and the nonlinear noise controllers to be simulated. These are the same parameters used to set up noise controllers for harmonic balance simulations. In the *Harmonic Balance Simulation* (cktsimhb), see *Selecting Nonlinear Noise Analysis* (cktsimhb).

Setting Up Small-Signal Simulations

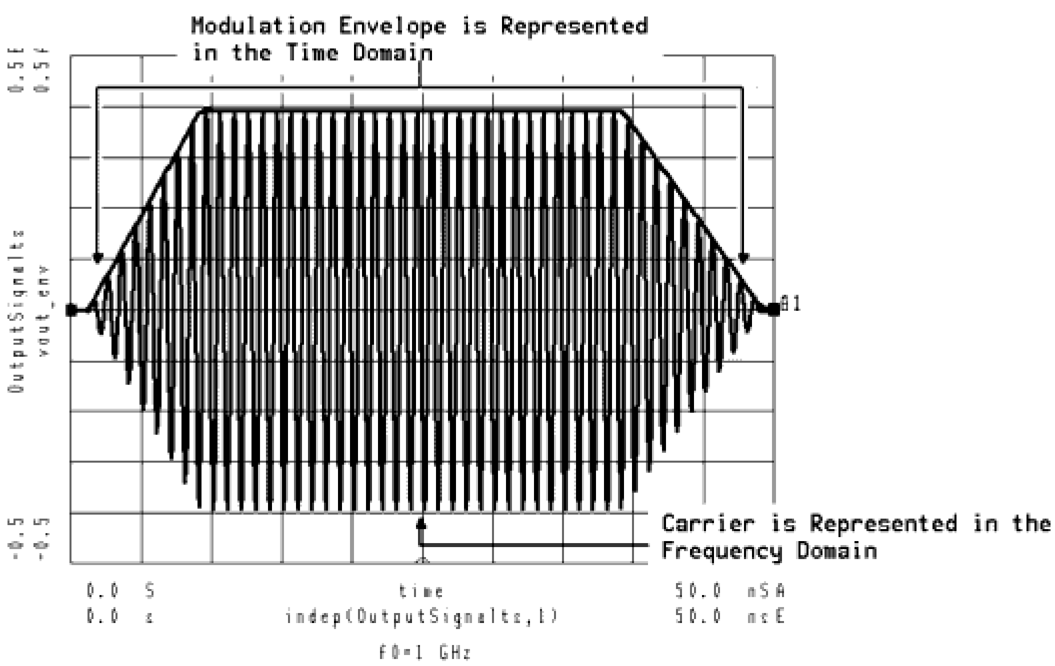
Use the Small-Signal parameters to use a large-signal/small-signal method to achieve faster simulations when some signal sources are much smaller than others, and are assumed not to exercise circuit nonlinearities.
Small-Signal and Noise analysis are performed only after the last Envelope time points, so that the Envelope sweep is allowed to get to a desired operating point, and then perform the standard small signal or noise characterization at that point.

These are the same parameters used to set up small-signal simulations for harmonic balance. In the *Harmonic Balance Simulation* (cktsimhb) documentation, see *Setting Up Small-Signal Simulations* (cktsimhb).

Theory of Operation for Circuit Envelope Simulation

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.

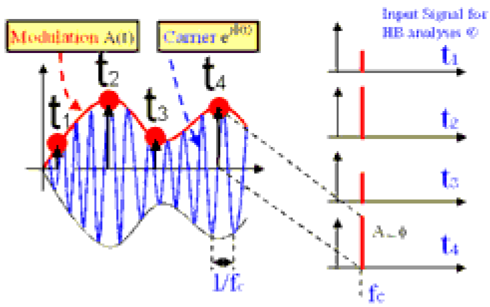
Briefly, this simulator permits input waveforms to be represented in the frequency domain as RF carriers, with modulation "envelopes" that are represented in the time domain as shown in the following figure.



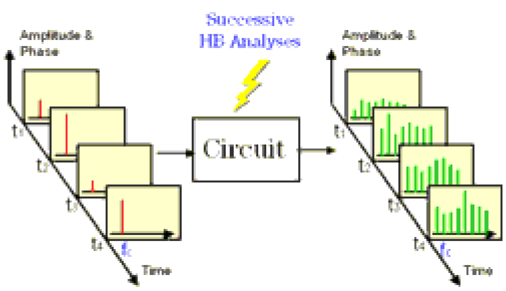
Modulated signal in the time domain

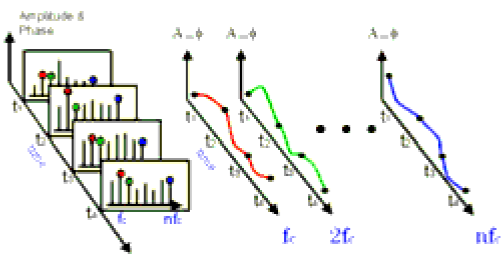
The following concepts present a basic overview of the circuit envelope simulation process.

- Transform input signal
  Each modulated signal can be represented as a carrier modulated by an envelope - A(t)*ejf(t). The values of amplitude and phase of the sampled envelope are used as input signals for Harmonic Balance analyses.

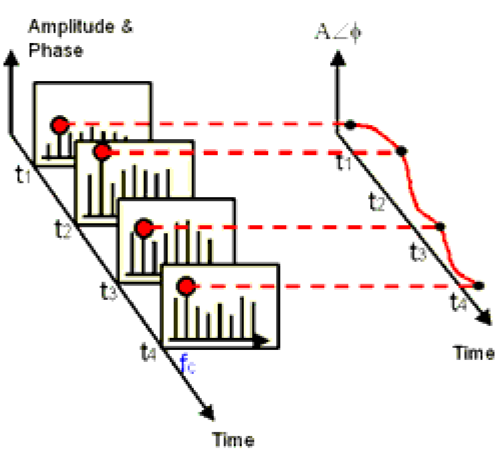


- Harmonic Balance analysis with time-varying envelopes
  Harmonic Balance analysis is performed at each time step, which includes both the basic HB equations as well as the effects due to time-varying envelopes. This process creates a succession of spectra that characterize the response of the circuit at the different time steps. Circuit Envelope provides a complete nonsteady-state solution of the circuit through a Fourier series with time-varying coefficients.





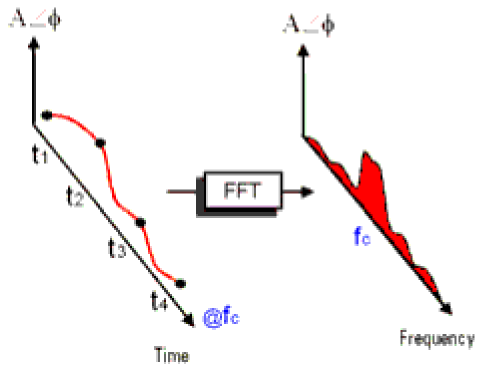
- Extract data from time domain
  Selecting the desired harmonic spectral line (fc in this case), it is possible to analyze:
  - Amplitude vs. time (oscillator start up, pulsed RF response, AGC transients)
  - Phase (f) vs. time (t) (VCO instantaneous frequency (df/dt), PLL lock time)
  - Amplitude and phase vs. time (constellation plots, EVM, BER)



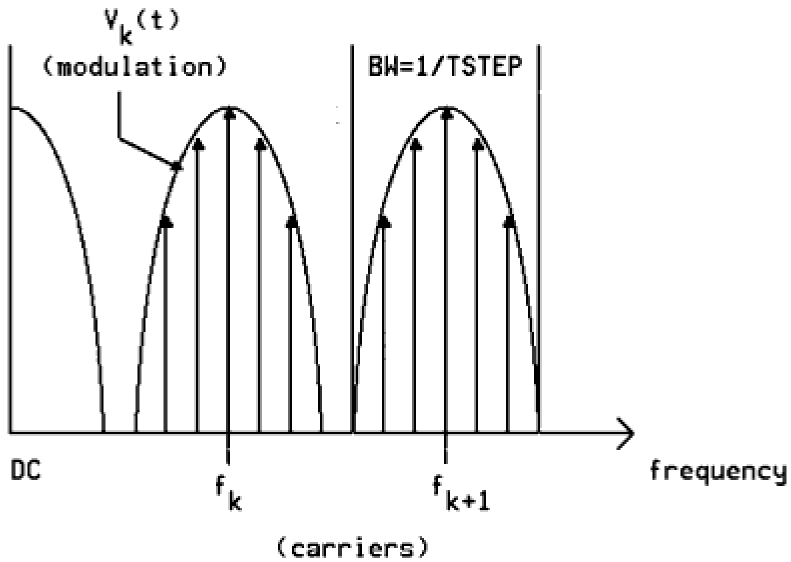
- Extract data from frequency domain

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 ±0.5/*Time step*. The following figure 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.



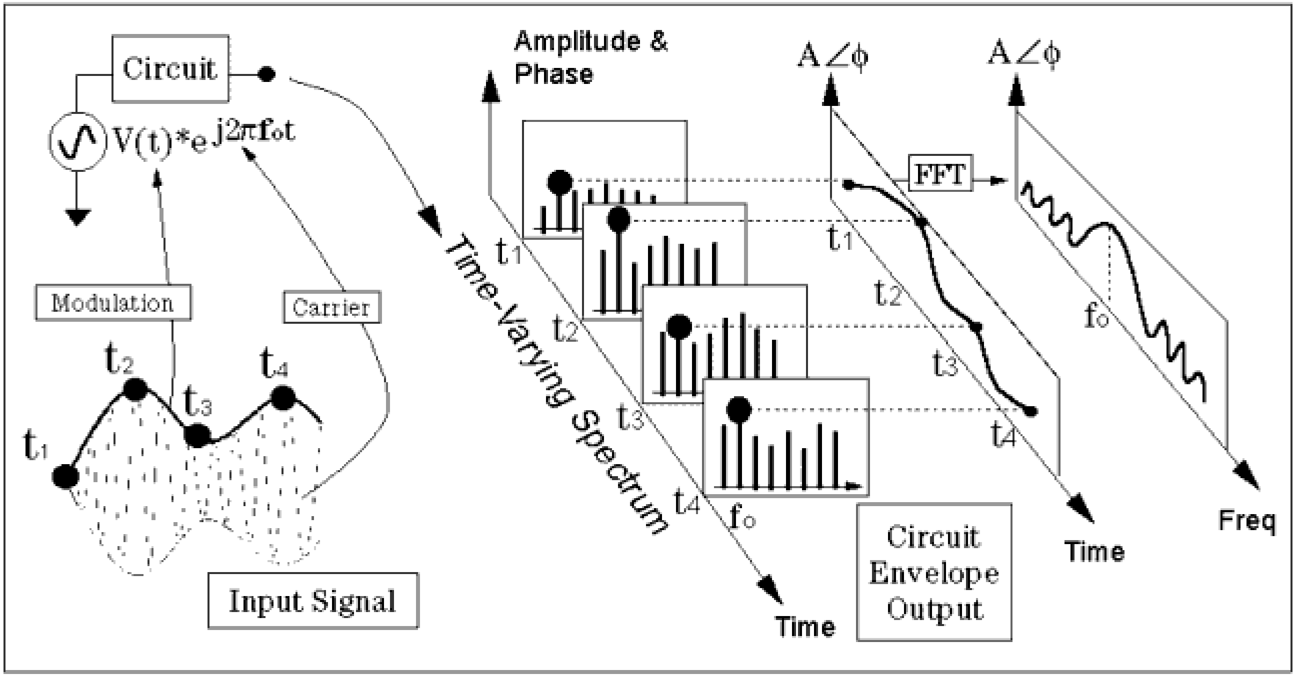
Spectra in the frequency domain

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.

The following figure 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.



A Modulated signal and its simulated time-varying spectrum

This simulator does not require that a spectral component be present at the center frequency. The following table 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/*Time step* , and only the real portion of V_k is used. The other N spectra have a double-sided bandwidth of 1/ *Time step* and are usually complex, as shown in the figure Spectra in the frequency domain.

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 $V_k(t)$. 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.

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*time)
2	V_k =exp(j*pi/2) or polar(1,90) or 1*j	Pure sine: sin(2*pi*f_k*time)
3	V_k =A*exp(j*2*pi*f_m *time+B)	One tone (SSB) A*cos(2*pi*(f_k+ f_m)*time+B)
4	V_k =A*exp(j*B); freq=1.1 GHz	Same as (3) (assuming f_k+ f_m = 1.1 GHz)
5	V_k =2*cos(2*pi*f_m *time)	Two tones (AM suppressed carrier)
6	V_k =exp(j*2*pi*f_m *time) + exp(-j*2*pi*f_m *time)	Same as (5)
7	V_k =pulse(time,...); freq=fk+fm	Pulsed RF at a frequency of f_k + f_m
8	V_k = −step(time − delay)	A negative cosine wave, gated on at t=delay
9	V_k = (vreal(time)+j*vimag(time))*exp(j*2*pi*f_m *time)	I/Q modulated source centered at f f_k + f_m. (vreal(), vimag() user-defined functions)
10	V_k =(1 + vr1) * exp(j*2*pi*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 *time/2)*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

Au tomatic Verification Modeling (AVM) is a user-selected mode of operation that can significantly speed up formerly lengthy cosimulations of Analog/RF circuits. This mode is also known as Fast Cosimulation. 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 ±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 set in the Envelope simulation. Normally, the frequency points used for noise characterization are the same as for small-signal response characterization in which a linear frequency sweep is used. Such a linear frequency sweep may require too many frequency points to properly capture narrow band noise such as 1/f noise. To handle 1/f noise characterization efficiently you can choose to set an independent log sweep.

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.

AVM 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.

Troubleshooting a Circuit Envelope 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 mixing order* can have a profound effect on simulation results; in general, set *Maximum mixing 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 (see *Harmonic Balance Basics* (cktsimhb)). 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/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.