



---

# A User's Guide to Envelope Following Analysis

*The procedures described in this application note are deliberately broad and generic. Requirements for your specific design may dictate procedures slightly different from those described here.*

## **Purpose**

This application note describes how to use Envelope Following Analysis with SpectreRF.

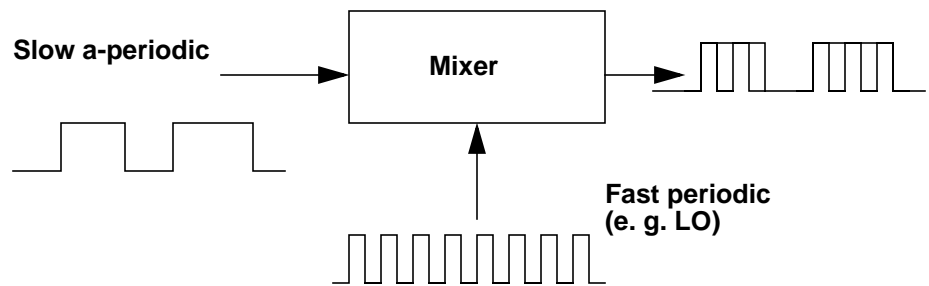
## **Audience**

This document is intended for SpectreRF users who need to simulate the envelope transient response of RF circuits. Knowledge of the basic properties of SpectreRF, Analog Artist environments is assumed. The analysis can be applied to efficiently and accurately analyze modulation signals in large communication circuits. Important applications include prediction of spectral regrowth of amplifiers or mixers, design of feedback loops such phase-locked loops or AGC loops, and transient behavior of switching power converters or switched capacitor filters.

# Overview

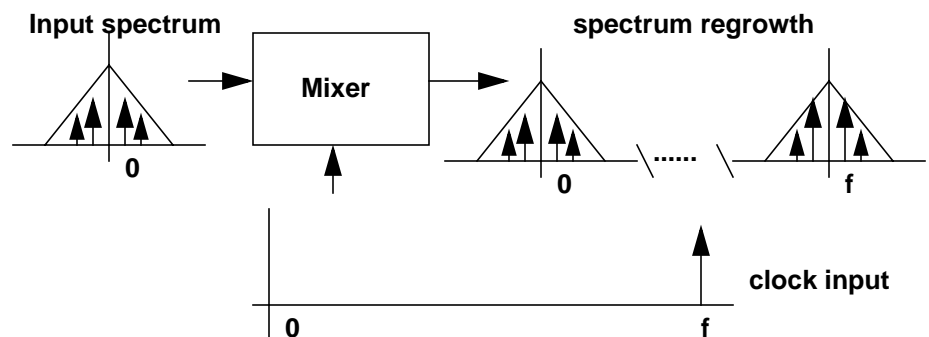
Efficient and accurate prediction of envelope transient response of RF circuits is useful for RF circuit designers who are interested in simulation of communications systems. For example, one may be interested in predicting the spectral regrowth a mixer. As shown in Figure 2-1, the input can be a complete digital RF modulation.

Figure 2-1 Time-domain modulation



The resulting spectral regrowth shown in Figure 2-2 can be simulated using the Envelope Following Analysis.

Figure 2-2 Spectrum-domain modulation



Typically, these modulation schemes are difficult to simulate because they are clocked at a frequency whose period is orders of

magnitude smaller than the time interval of interest to the designer. Classical transient simulation is extraordinarily expensive. New RF analyses such as PSS and PDISTO may not work directly because the modulation signal is neither periodic nor quasi-periodic.

Envelope Following Analysis is a response to the need for efficient simulation of these modulation schemes. It reduces the simulation time without compromising accuracy by exploiting the property that the behavior of the circuits in a given high frequency clock cycle is similar, but not identical, to the behavior in the preceding and following cycles. In particular, the “envelope” of the high-frequency clock can be followed by accurately computing the circuit behavior over occasional cycles, which accurately capture the fast transient behavior. The slow varying modulation is accurately followed by a piecewise polynomial. As a result, the spectrum of the circuit response can be obtained by combining the piecewise polynomial and the integration of occasional clock cycles.

This document is organized as follows. We first give a brief introduction to how the envelope following analysis works in the “Envelope Following Algorithm” section. This section is intended for users who are interested in some mathematical background of the computational algorithms. Afterwards, we discuss how to use the analysis in the “Envelope Following Analysis” section. We discuss application scenarios and interpretation of its output data. In the “Analysis Parameters” section, we give the definition of each analysis parameter. We discuss similarities to transient analysis parameters and discuss differences between the two analyses. In the “User Interface” section, the set-up of this analysis at the UI level is discussed. In particular, an example in amplitude modulation is given to show how to compute the power spectral density. Finally, application examples are given. In the “ACPR calculation” section, we show how a transmitter circuit may be simulated to compute estimate to adjacent channel power ratio. In the “AGC example” section, we show how to simulate the automatic gain control loop dynamics.

# Envelope-Following Algorithm

The SpectreRF Envelope Following analysis uses a multi-stage multi-past-point integration algorithm, which is an extension to a method introduced by Kundert, White and Sangiovanni-Vincentelli [1]. The method is based on a time domain shooting scheme, in which the clock nonlinearity is resolved by the time domain integration, rather than explicitly being expressed as harmonics of fundamental frequencies.

Most RF circuits used in communication systems are clocked at a high frequency. The clock (such as LO) is usually causing the most nonlinearity in the circuit response. Time-domain integration is more efficient and accurate in resolving sharp transitions and transient details than representing them in term of harmonics. SpectreRF Envelope Following Analysis samples the circuit waveforms at the clock frequency, and assumes the resulting envelope can be accurately represented by a piecewise polynomial. The nonlinearity caused by the clock signal is resolved by occasional integrations of a period of circuit responses.

Most circuits can be described by a system of differential equations of the form

$$\frac{d}{dt}p(v(t), u(t)) + f(v(t), u(t)) = 0 \quad (1)$$

where  $v(t) \in \mathfrak{R}^N$ , the state, is the vector of capacitor voltages and inductor currents,  $u(t) \in \mathfrak{R}^M$  is the vector of input sources,  $p(v(t), u(t)) \in \mathfrak{R}^N$  is the vector of capacitor charges and inductor fluxes, and  $f(v(t), u(t)) \in \mathfrak{R}^N$  is the vector of resistance currents and inductor voltages. If the state  $v$  is known at some time  $t_0$ , it is possible to solve (1) and compute the state at some late time  $t_1$ . In general, one can write

$$v(t_1) = \phi(v(t_0), t_0, t_1) \quad (2)$$

where  $\phi: \mathfrak{R}^N \times \mathfrak{R}^N \times \mathfrak{R}^N \rightarrow \mathfrak{R}^N$  is a state transition function for the differential equation.

Consider that the circuit to be simulated has as an input a clock with a known period  $T$  that is much smaller than the simulation interval. Assume the sequence formed by sampling the state at the beginning of each clock cycle,  $v(0), v(T), v(2T), \dots, v(mT), \dots$ , changes slowly as a function of  $m$ , the clock cycle number. A

smooth continuous function can be defined to interpolate the sequence formed by sampling the state at every time interval  $T$ .

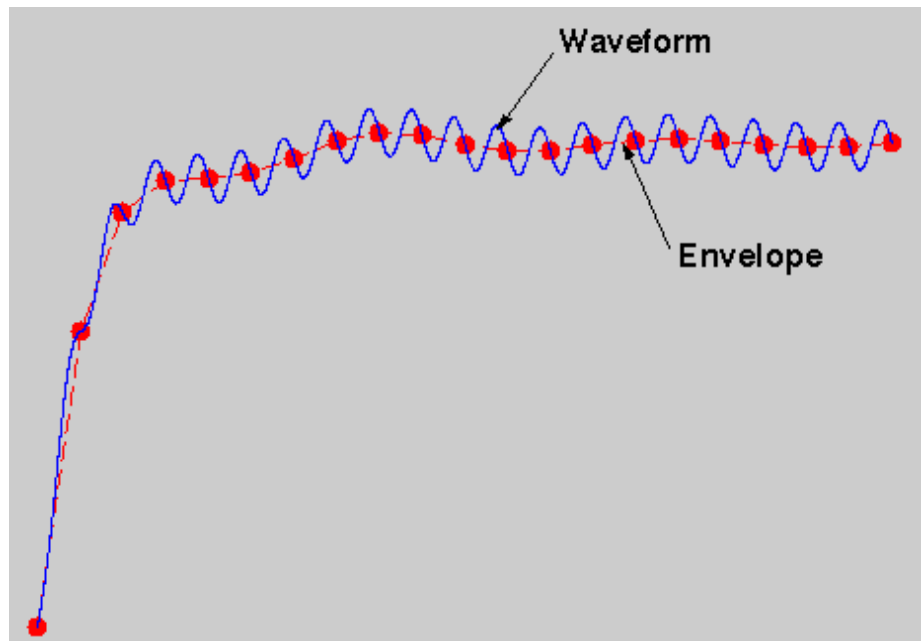
In Figure 2-3, the waveform is sampled at the clock period, resulting in an circuit envelope that reveals the slow varying modulation.

Figure 2-4 illustrates how the envelope following algorithm works. It is similar to transient integration. In this example, two past stages are used to compute a new stage. Assume the state values are known at  $t_{(n-1)}$  and  $t_n$ , which are a number of clock cycles apart from each other, and are at the beginning of the clock cycle. The state values at  $t_{(n-1)}$  and  $t_{(n-1)} + T$  are related by integration of one clock cycle, i.e.,

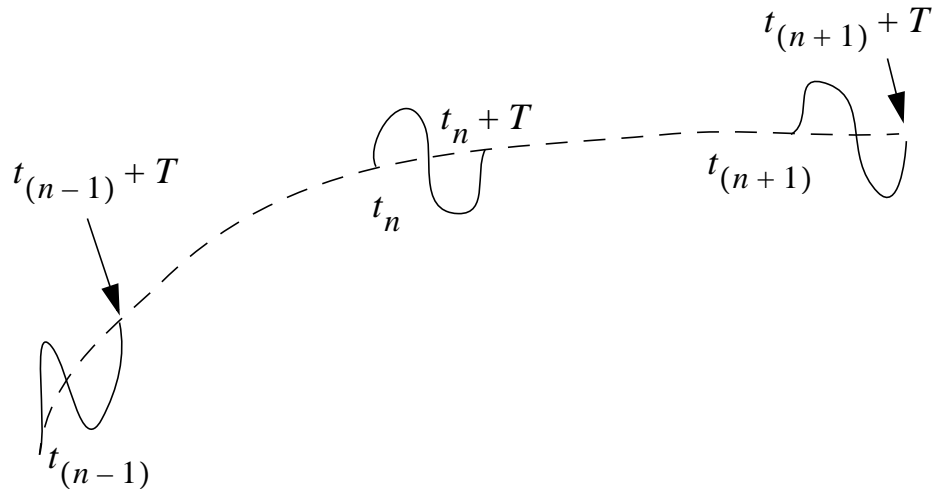
$$v(t_{(n-1)} + T) = \phi(v(t_{(n-1)}), t_{(n-1)}, t_{(n-1)} + T), \quad (3)$$

where  $\phi$  is the state transition function. Likewise, the states at  $t_n$  and  $t_n + T$  are related by  $v(t_n + T) = \phi(v(t_n), t_n, t_n + T)$ . The job of the algorithm is to find the state value at a new time point  $t_{(n+1)}$  many cycles from  $t_n$ , such that the pair  $v(t_{(n+1)}), t_{(n+1)}$  interpolates a quadratic polynomial defined by the three state values  $v(t_{(n-1)})$ ,  $v(t_n)$ ,  $v(t_{(n+1)} + T)$  at  $t_{(n-1)}$ ,  $t_n$ ,  $t_{(n+1)} + T$ , respectively. On the one hand, the state value  $v(t_{(n+1)} + T)$  is given by

**Figure 2-3** Sampled Envelop



**Figure 2-4 Envelope Following Integration**



$$v(t_{n+1} + T) = \phi(v(t_{n+1}), t_{n+1}, t_{n+1} + T). \quad (4)$$

On the other hand, from the interpolation condition,  $v(t_{(n+1)} + T)$  can be written as a linear combination of  $v(t_{(n-1)})$ ,

$v(t_n)$ ,  $v(t_{n+1})$ , i.e., there exist scalars  $\alpha_1, \alpha_2, \alpha_3$  such that

$$v(t_{n+1} + T) = \alpha_1 v(t_{n-1}) + \alpha_2 v(t_n) + \alpha_3 v(t_{n+1}). \quad (5)$$

The scalars  $\alpha_1, \alpha_2, \alpha_3$  can be obtained by standard techniques such as first defining a quadratic function using three state values and then asking the fourth value to be interpolated by the quadratic function. Combining the two equations gives

$$\alpha_1 v(t_{(n-1)}) + \alpha_2 v(t_n) + \alpha_3 v(t_{n+1}) - \phi(v(t_{n+1}), t_{n+1}, t_{n+1} + T) = 0 \quad (6)$$

which can be used to solve for  $v(t_{n+1})$  by shooting Newton method. At each iteration an adjustment of  $v(t_{n+1})$ ,  $\Delta v(t_{n+1})$ , is solved from the Newton equation

$$\left[ \alpha_3 I - \frac{\partial \phi}{\partial v(t_{n+1})} \right] \Delta v(t_{n+1}) = RHS \quad (7)$$

This process is repeated until the interpolation equation is satisfied to certain tolerance. If the tolerance can not be satisfied, we make  $t_{(n+1)}$  closer to  $t_n$  and repeat the shooting process.

The three stage process is repeated until the desired stop time is reached.

Similar to PSS (for details see [3]), the Newton equation is solved efficiently by a matrix-implicit iterative method, which allows circuits of tens of thousands of nodes to be simulated using standard workstations. Compared to harmonic balance based envelope following algorithms, it has advantages and weaknesses similar to SpectreRF Periodic Steady-State (PSS) versus analogous harmonic balance based steady-state computation approaches.

## Envelope Following Analysis

This analysis computes the envelope response of a circuit. The user specifies the analysis **clockname**. The simulator automatically determines the clock period by looking through all the sources with the specified name. The envelope response is computed over the interval from **start** to **stop**. If the interval is not a multiple of the clock period, it is rounded off to the nearest multiple before the stop time. The initial condition is taken to be the DC steady-state solution if not otherwise given.

Envelope following analysis is most efficient for circuits where the modulation bandwidth is orders of magnitude lower than the clock frequency. This is typically the case, for example, in circuits where the clock is the only fast varying signal and other input signals have a spectrum whose frequency range is orders of magnitude lower than the clock frequency. An example is given by Figure 2-1 in the situation of calculating spectrum regrowth of a mixer.

In general, Envelope Following Analysis is not intended for circuits working with multiple high-frequency fundamentals. However, similar to Periodic Steady-State (PSS) Analysis, it can be stretched to work for particular classes of circuits operating with multiple high-frequency fundamentals.

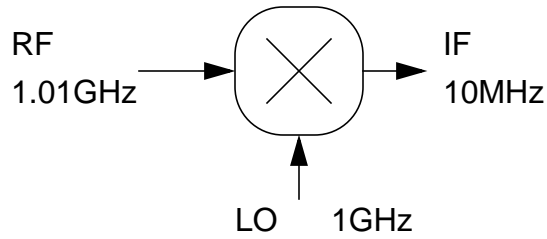
The first case is when the multiple fundamentals are commensurate. In this situation, the greatest common denominator of all fundamental frequencies can be used as the clock frequency. This is similar to the process of figuring out the beat frequency for PSS analysis. In term of efficiency, it is also analogous to PSS analysis. At each integration of the clock period, many fast cycles could be involved. The efficiency of the method depends on how many fast cycles are involved in the beat period, the more the fast cycles simulated, the less efficient. One difference is that one can give the fundamentals the same name and specify it using **clockname**, and the simulator will figure out the beat frequency automatically. While, for PSS analysis, in similar situation, the beat frequency is figured out either at the UI level or by the user.

For another example, see Figure 2-5, the down conversion of two closely placed frequencies can also generate a slow-varying modulation envelope whose frequency is orders of magnitude lower than the input frequencies. In this situation, Envelope Following Analysis can be used to trace out the modulation envelope by



choosing either of the fast varying signal as the clock. However, normally the one that causes the most nonlinearity should always be chosen.

Figure 2-5 Down conversion of two closely placed frequencies



Envelope Following Analysis generates two types of output files, a voltage versus time (**td**) file, and an amplitude/phase versus time (**fd**) file for each of specified harmonic of the clock fundamental.

The **td** file contains real time-domain waveforms. It is similar to waveforms generated by transient analysis. The difference is that the integration of a clock cycle is only done occasionally. You will normally see big gaps between integrated clock cycles.

The **fd** file contains time varying Fourier coefficients (complex) of circuit response at clock harmonics. The time varying Fourier coefficients are discussed in more detail in the section on ACPR calculation.

The spectrum of each harmonic response is calculated from Artist, which is useful for applications such as ACPR calculation.

The **fd** result is also useful for other forms of data analyses. See the section on ACPR calculation for details. For example, by plotting amplitude versus phase, constellation diagrams may be obtained.

In some sense, Envelope Following Analysis can be considered as a fast version of transient analysis. Its efficiency comes from skipping simulation cycles. However, if the analysis can not find cycles to skip, it reduces to transient analysis. Therefore it is always as accurate as transient analysis. It may not always be more efficient than transient analysis due to more computational

overhead when not enough cycles can be found to skip, but it is always as accurate.

# Analysis Parameters

Setup of envelope following analysis is similar to setup of transient analysis. Important parameters are **stop**, **clockname**, **harms**, **modulationbw**, **method**, **swapfile**. There are a few important differences from transient analysis:

1. You must specify **clockname**. This is similar to PDisto analysis. However, in envelope following analysis you only need to give a name to the clock source.
2. Specify **harms/harmsvec**. This tells the simulator which harmonics of the clock frequency are of interest. **harms** is defaulted to 1, which covers most of the applications. The number of harmonics effects both output and computation. Hence, one should avoid specifying unnecessary harmonics.
3. Stop time **stop** should be long enough for slow signals to behave some (say to complete at least a few clock cycle).
4. It is helpful to set **modulationbw** to a frequency that reflects how the envelope is varying.
5. Similar to other RF analysis, **method=gear2only** might be preferable.
6. It is recommended to use **swapfile** (similar to PSS, PDisto), for simulation of large circuits.

Most of the rest of the parameters are inherited from Spectre **tran** analysis and their meanings are consistent.

## Envelope fundamental parameters

**clockname** Name of the clock fundamental.

**modulationbw** (Hz) Modulation bandwidth.

## Simulation interval parameters

**stop** (s) Stop time.

**start=0** s Start time.

**outputstart=start** s Output is saved only after this time is reached.

## Time-step parameters

**maxstep** (s) Maximum time step for inner transient integration. Default derived from **errpreset**.

**maxenvstep** (s) Maximum outer envelope step size. Default derived from **errpreset**.

### Initial-condition parameters

**ic=***all* What should be used to set initial condition. Possible values are *dc*, *node*, *dev*, or *all*.

**skipdc=***no* If yes, there will be no dc analysis for initial transient. Possible values are *no*, *yes*, *waveless*, *rampup*, or *autodc*.

**readic** File that contains initial transient condition.

### Convergence parameters

**readns** File that contains estimate of initial DC solution.

**cmin=**0 F Minimum capacitance from each node to ground.

### State-file parameters

**write** File to which initial transient solution is to be written.

**writefinal** File to which final transient solution is to be written.

**swapfile** Temporary file that holds the matrix information used by Newton's method. Tells Spectre to use a regular file rather than virtual memory to hold the matrix information. Use this option if Spectre complains about not having enough memory to complete this analysis.

### Integration method parameters

**method** Integration method. Default derived from ``errpreset'`. Possible values are *euler*, *trap*, *traponly*, *gear2*, or *gear2only*.

### Accuracy parameters

**errpreset=***moderate* Selects a reasonable collection of parameter settings. Possible values are *conservative*, *moderate* or *liberal*.

**relref** Reference used for the relative convergence criteria. Default derived from ``errpreset'`. Possible values are *pointlocal*, *alllocal*, *sigglobal*, or *allglobal*.

**iteratio** Ratio used to compute LTE tolerances from Newton tolerance. Default derived from *errpreset*.

**steadyratio** Ratio used to compute steady state tolerances from LTE tolerance. Default derived from `errpreset`.

## Annotation parameters

**stats=no** Analysis statistics. Possible values are *no* or *yes*.

**annotate=sweep** Degree of annotation. Possible values are *no*, *title*, *sweep*, *status*, or *steps*.

**title** Analysis title.

## Output parameters

**harms=1** Number of clock harmonics to output.

**harmsvec=[...]** Array of desired clock harmonics. Alternate form of **harms** that allows selection of specific harmonics.

**outputtype=both** Output type. Possible values are *both*, *envelope* or *spectrum*.

**save** Signals to output. Possible values are *all*, *lave*, *playboy*, *lovably*, *selected*, or *none*.

**nestlvl** Levels of subcircuits to output.

**compression=no** Do data compression on output. Possible values are *no* or *yes*.

## Newton parameters

**maxiters=5** Maximum number of Newton iterations per transient integration time step.

**envmaxiters=3** Maximum number of Newton iterations per envelope step.

**restart=yes** Do not use previous DC solution as initial guess. Possible values are *no* or *yes*.

## Circuit age

**circuitage** (Years) Stress Time. Age of the circuit used to simulate hot-electron degradation of MOSFET and BSIM circuits.

We further discuss some analysis parameters. The simulator examines all the sources whose name matches the clock name specified by the **clockname** parameter to determine the clock frequency. If more than one frequency is found, the greatest common factor of these frequencies is used as the clock frequency.

The maximum envelope step size is affected by many parameters. It can be directly limited by **envmaxstep**. It is also limited by **modulationbw**. The user gives an estimate of the modulation bandwidth. The simulator will put at least eight sample points within the modulation period.

The **harms** and **harmsvec** parameters affect the simulation time in a significant way. The spectrum is calculated for all the specified harmonics for all sampled integration cycles as the envelope following analysis marches on. For each harmonic, a file is generated. The user should refrain from specifying unnecessary harmonics. Typically, **harms** is set to 1 or 2.

Most parameters of this analysis are inherited from either transient or PSS analysis and their meanings are consistent. However, a few of them need to be clarified. The effect of **errpreset** on some particular envelope following analysis parameters is shown in the following table.

**Figure 2-6** Parameter defaults as a function of 'errpreset'

errpreset	envmaxstep	steadyratio	envlteratio
liberal	Interval/10	1.0	10.0
moderate	Interval/50	0.1	1.0
conservative	Interval/100	0.01	0.1

Its effect on parameters such as **reltol**, **relref**, **method**, **maxstep**, and **lteratio** are the same as defined for transient analysis, except for that the transient simulation interval is always a clock period.

# User Interface

Basically the set-up of the Envelope Following choose analysis form is similar to the set-up of the transient analysis choose analysis form. Important parameters are **clockname**, **stop time**, **output harmonics**, **accuracy defaults**, **modulationbw**, **integration method** and **swapfile**.

A few important differences from the transient analysis:

- (1) Need to specify a clockname. This is similar to PDisto; however, here you only need to give the name of the clock source. A “select clockname” button is available which will show a list of all the valid clock names. Often, the fastest source in the circuit is used as the clock. All sources must be with a small relative bandwidth of a clock harmonic.
- (2) The output harmonics section allows the user to specify the **harms** or **harmsvec** parameters. This tells the simulator which harmonics of the clock frequency to compute and to output.

The Envelope Following direct plot form allows the user to plot three different waveform types. If voltage is selected and then time is selected as the sweep type then this will plot the voltage envelope versus time. Using an AM example with a 1GHz carrier and a 1 MHz baseband tone then a sampled version of the Amplitude Modulated waveform would be plotted. Instead of taking a continuous set of points like tran does, the Envelope analysis takes intelligently spaced samples of this waveform in order to save simulation time. If the user mentally connects the tops of all the samples, then they will clearly see the envelope that “tran” will give.

If voltage is selected and “harmonic time” is selected as the sweep type then the user can plot the complex time-varying Fourier coefficient for each harmonic. Using the AM example from above, if the 1st harmonic is selected then the user would see the 1 MHz baseband signal (so the Envelope analysis basically strips off the 1GHz carrier).

If voltage is selected and spectrum is selected then several Power Spectral Density fields will show up. Once these fields are entered then the direct plot form will calculate the “total number of samples”, window size, and number of bins and then it calls the **psdbb** function. Here is a description of the Power Spectral Density Parameters:

Start time and stop time - Starting and ending times, respectively, for the interval the spectral analysis is to be performed over. Normally, these will be set to the simulator start and stop times, respectively.

Nyquist half-bandwidth - maximum frequency at which there are signals of interest. Normally this will be set to 3 to 5 times the maximum baseband frequency.

bin-width - is the frequency resolution (i.e. the width of the frequency bins).

Max and Min plotting frequency - These set the x-axis on the waveform to be plotted.

windowing name - This is a preset list of available windowing functions used during the spectrum calculation.

detrending - allows one to remove certain trends from the data before the spectral analysis.

Here are some typical settings for the AM example mentioned above (these numbers assume that the stop time was set 60 us):

---

Start time	0
Stop time	60us
Nyquist half-bandwidth	5e6
Bin-width	20e3
Max plotting freq	5e6
Min plotting freq	-5e6
Windowing name	hamming
Detrending	none

---

For the AM example mentioned above, the user should see a waveform that is in  $V^2/(Hz)$  versus frequency and there should see some signal strength at 1 MHz. The calculated “total number of samples”, window size, and bin-width are printed to the CIW in case the user would like to use the psddb function on the calculator. Here are the calculations that are used to generate these numbers:

1. Get  $L = T_o - T_f$  - From (where  $T_o$  and  $T_f$  times are user inputs).



2.  $f_{\max}$  (Nyquist half-bandwidth) and binwidth are user inputs.
3. compute  $\#bins = \text{floor}(L * \text{binwidth})$ , with  $\#bins \geq 1$ .
4. compute the smallest  $m$  such that  $2^m * (\#bins) > 2 * L * f_{\max}$ .
5.  $\text{window size} = 2^m$ .
6. number of samples =  $\#bins * \text{window size}$

## ACPR calculation

Transmitter ACPR (Adjacent Channel Power Ratio) cannot be computed directly with PSS and PDISto because the input baseband signals carry information and are therefore not periodic. The J-model (see [2] for details) is a very fast indirect method for computing ACPR, but like all behavioral models, has limitations. Envelope Following is a way to check J-model ACPR calculations that is much faster than brute force Spectre-transient simulation. This section describes the various Envelope Following outputs and how to compute ACPR. Figure 2-7 shows the test circuit. The behavioral blocks model an ideal I/Q modulation. The carrier frequency is 1GHz. The amplifiers are transistor-level models.

Figure 2-7 ACPR test circuit

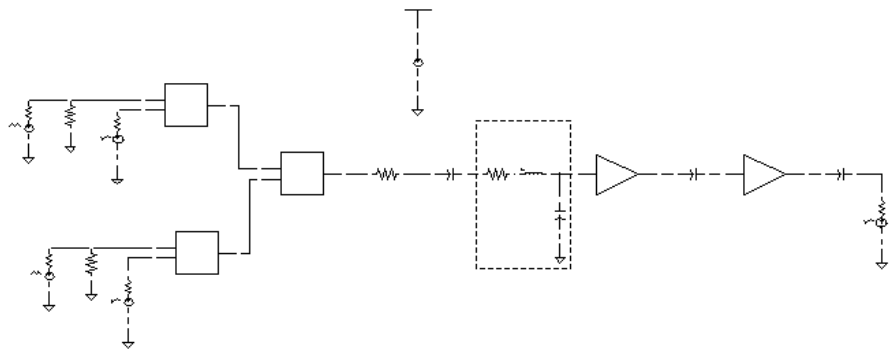
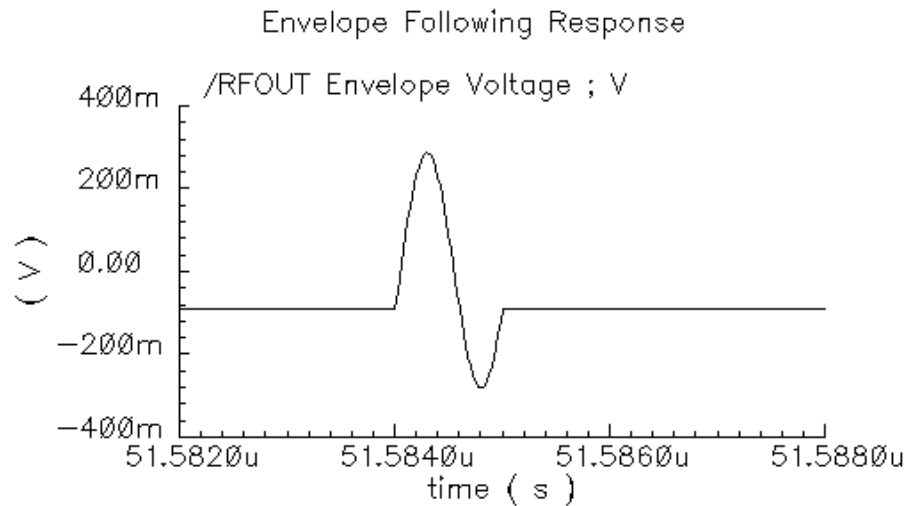
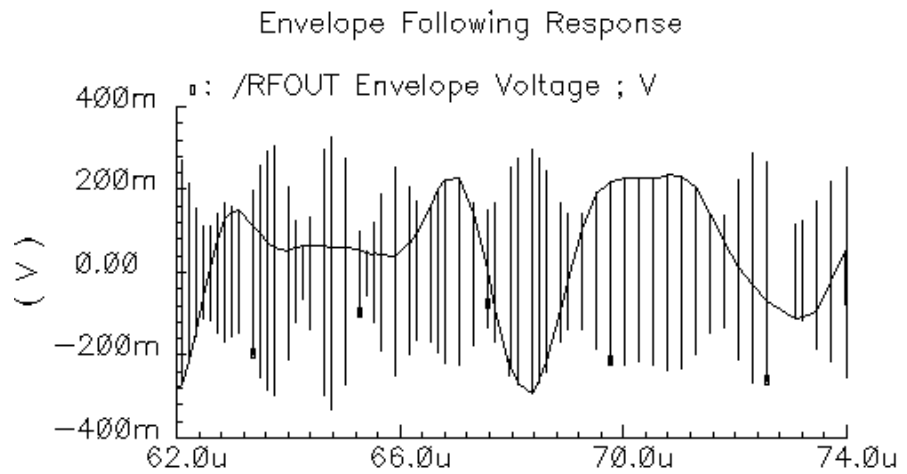


Figure 2-8 shows the “time” response of the RF output. The top picture shows some of the cycles. Note that lots of cycles are missing. That is why Envelope following is faster than transient analysis. The bottom picture focuses on one of those cycles.

Although one can see something resembling an envelope in the time response, the detailed cycles can obscure it on larger time scales and it is not guaranteed to represent any particular phase of the envelope. The “harmonic time” response is designed to extract the baseband components or any other harmonic of interest. Most of the time the fundamental, or first harmonic, is the most

Figure 2-8 Envelope Following time response.

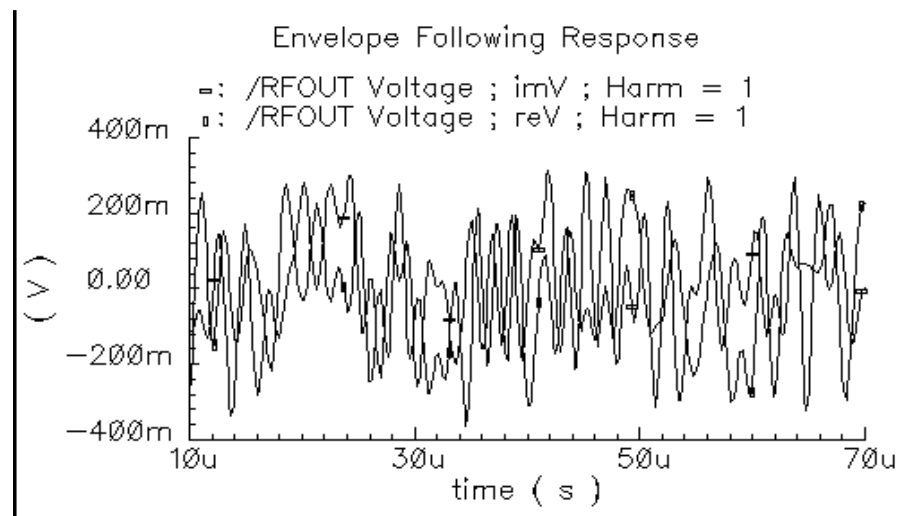
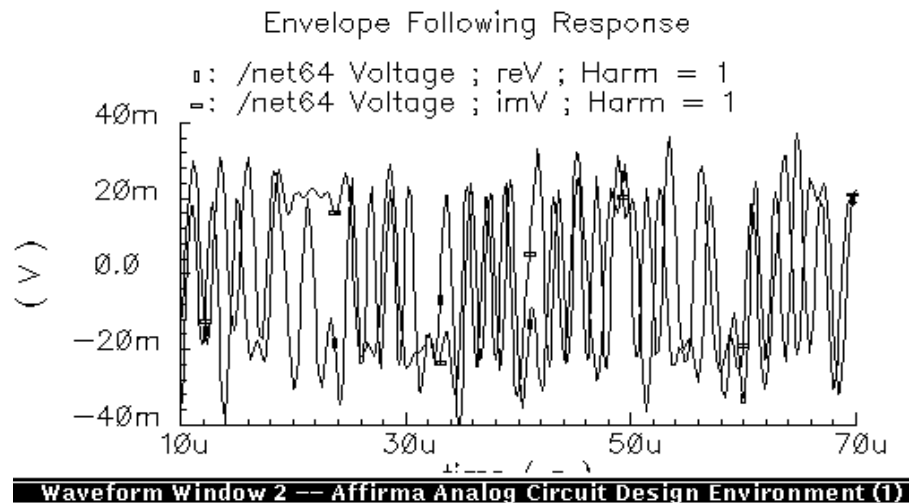


interesting. The harmonic analysis plots the real and imaginary parts of the specified Fourier component of each cycle as a function of its location in time. Figure 2-9 shows how the real and imaginary parts of the fundamental components of the RF input and output signals evolve with time. These signals are the baseband representations of the RF signals. Not much can be ascertained directly from time-domain baseband waveforms. However, x-y plots of real and imaginary waveforms show the trajectory the baseband

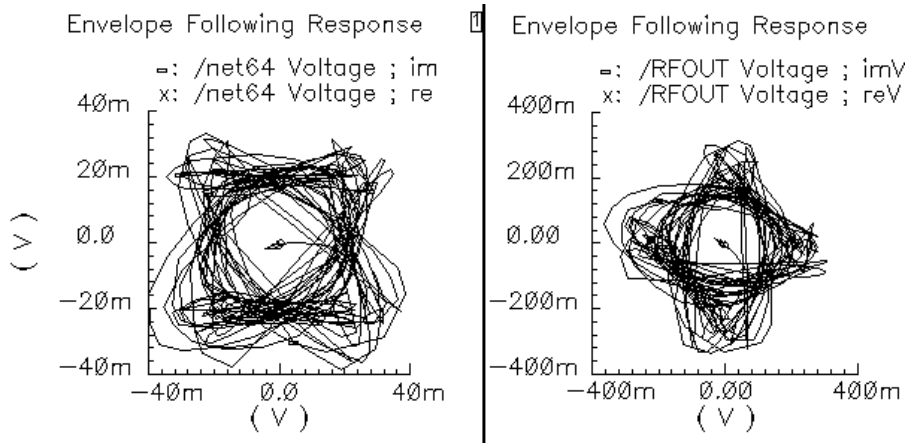
signal traces out in the symbol constellation space. Figure 2-10 shows the input and output baseband trajectories. A trajectory is displayed by changing the x-axis to be the real waveform. The output trajectory is a scaled and rotated version of the input trajectory. The scaling factor is evident in the time domain waveforms but the trajectories also make the phase shift obvious. The transmitter introduces about 45 degrees of phase shift.

**Figure 2-9**  
(harmonic time analysis).

**top=input baseband waveforms, bottom=output baseband waveforms.**

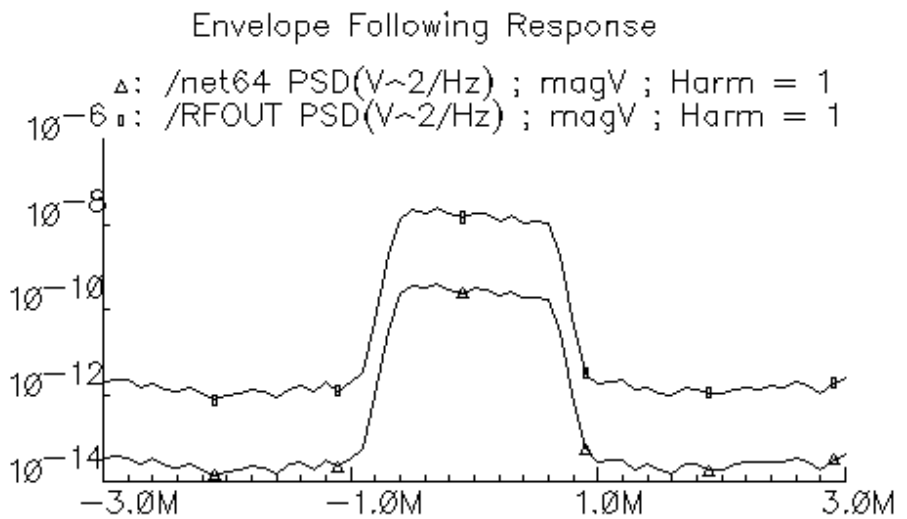


**Figure 2-10 Baseband Input and Output Trajectories**



ACPR estimation requires power spectral densities. The “spectrum” button on the Envelope Following results window is for estimating the power spectral density of a baseband signal. The baseband signal is the time-varying fundamental Fourier component described above. Figure 2-11 compares input and output power spectral densities. This example does not show much spectral regrowth. The input and output ACPRs are comparable.

**Figure 2-11 Power Spectral Densities of the Baseband representations of the Input and Output RF Signals.**



The power spectral densities computed in the last figure were done with the following options:

From	0
To	100u
Nyquist half-bandwidth	3M
Frequency bin width	100k
Max. plotting frequency	3M
Min plotting frequency	-3M
Window	Hanning
Detrending	None

The “From” and “To” parameters define the time interval over which the spectrum will be computed. You may want to make the “From” time different than zero to exclude start-up transient from the analysis. The “Nyquist half-bandwidth” indirectly determines the spacing of the time points used in the FFT calculations. If the original time domain data points are too far apart to support the Nyquist bandwidth, you may see strange results such as unexpected spikes in the spectrum. The “Frequency bin width” determines the frequency resolution. A smaller frequency resolution usually produces a “noisier” spectrum. The power spectral density is a frequency-by-frequency average of the FFT taken over several time windows within the main time interval. A smaller bin width produces fewer samples to average together at each frequency. The min and max plotting frequencies are self-explanatory.

The “window” selection determines how data at the edges of a time window will be attenuated to control spectral leakage. Spectral leakage occurs because an FFT is always taken over a finite time interval. That is like multiplying the original waveform by a pulse of amplitude one and duration coincident with the duration of the data. Multiplication in the time domain corresponds to convolution in the frequency domain. The transform of the long pulse is a high and narrow sinc function. As the pulse length goes to infinity, the main lobe of the sinc function approaches a Dirac delta function. When estimating the spectrum at a given frequency, the sinc function’s side lobes cause spectral components of the untruncated signal to “leak” into the estimation of the spectrum at the main lobe.

## AGC Example

One application of Envelope Following is in assessing AGC (Automatic Gain Control) loop dynamics. PSS and PDISTO can not compute the AGC response because it is not periodic. The loop might have a periodic response if it were unstable but PSS and PDISTO would be difficult to apply because the period of the instability would not be known *a priori*. Transient analysis is usually out of the question because the frequencies of the amplitude oscillation and carrier are too far apart. Envelope Following is an efficient and accurate alternative for assessing stability of AGC loops as the example below shows.

Consider the AGC loop in Figure 2-12. The behavioral blocks are simple multipliers from the ahdl library. The upper left multiplier is the variable gain amplifier (VGA). After filtering, the right-most multiplier produces the square of the amplitude of the VGA output. The AGC loop input is a fixed-amplitude 100Mhz carrier. A stable version of this circuit would generate an output signal with the same phase as the input signal but with the amplitude determined by the feedback loop. This design has an extra integration in the feedback loop that makes the loop unstable. The simulator's challenge here is to detect the design error by quickly and accurately simulating the unstable behavior.

Figure 2-13 compares Spectre transient and SpectreRF Envelope Following results. Both show the instability in the output amplitude but the Envelope Following analysis runs 14 times faster because it avoids redundant carrier cycles. The time savings would be even greater with a 1Ghz carrier. Figure 2-14 compares the transient and Envelope Following for a single cycle. The Envelope Following simulation was not only fast, it was accurate

Figure 2-12 AGC Loop

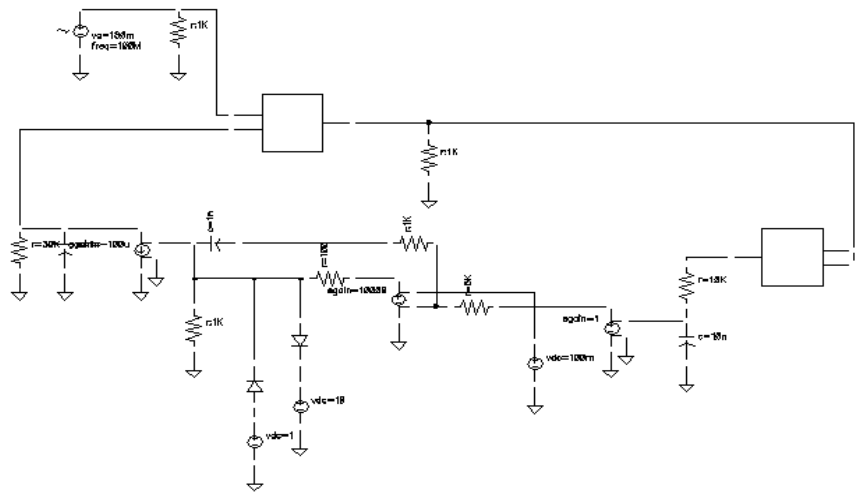
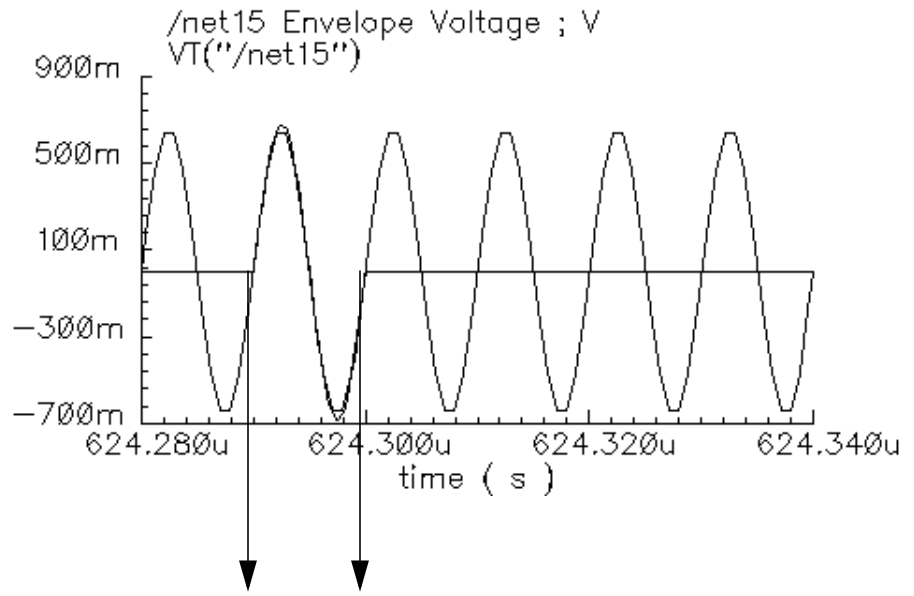


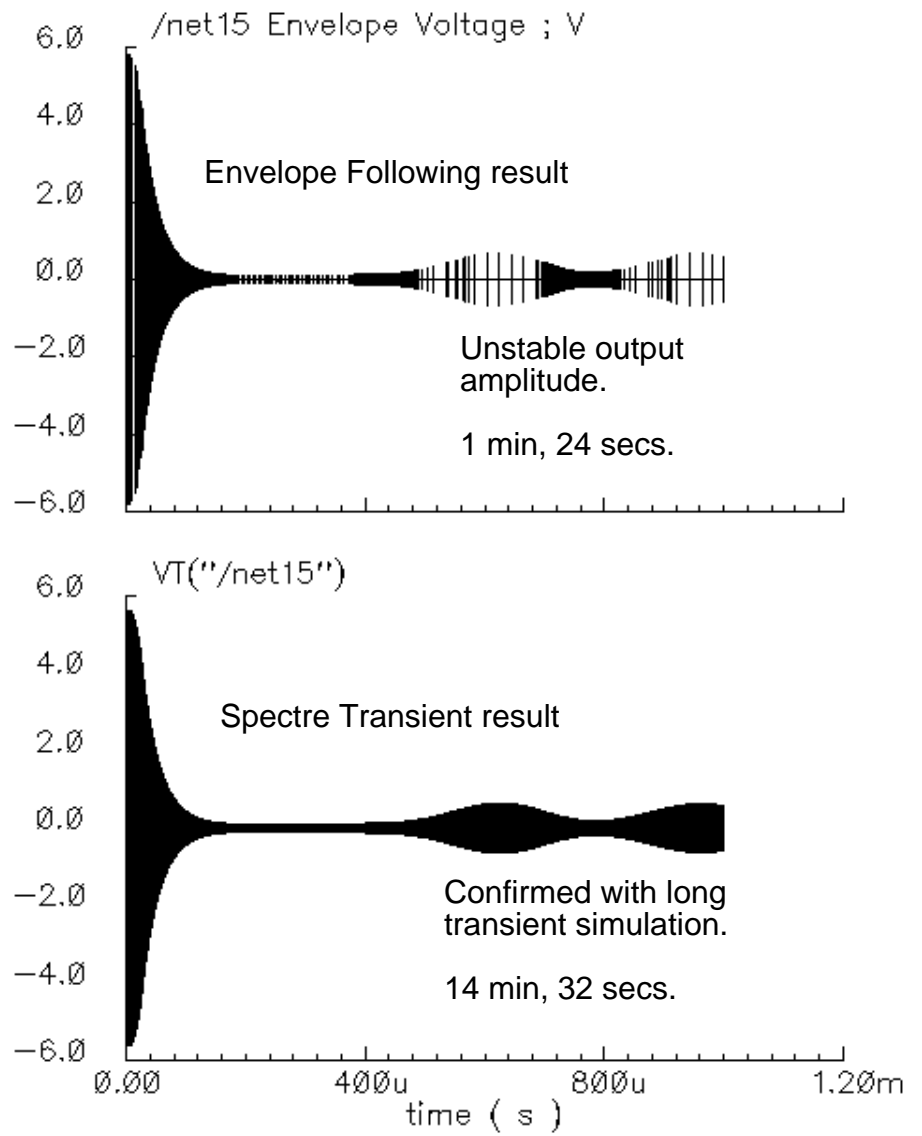
Figure 2-13 Blow-up of One Cycle.



Cycle simulated by Envelope Following, others skipped.



Figure 2-14 Comparison Between Transient and Envelop Runs



## References

- [1] K. Kundert, J. White and A. Sangiovanni-Vincentelli. An envelope-following method for the efficient transient simulation of switching power and filter circuits. *IEEE International Conference on Computer-Aided Design: Digest of Technical Papers*, November 1988.
- [2] J.Chen, D.Feng, J.Phillips, and K.Kundert, Simulation and modeling of intermodulation distortion in communications circuits. *Proceedings of the Custom Integrated Circuit Conference, San Diego, CA*, May 1999.
- [3] R. Telichevesky, K. Kundert and J. White. Efficient steady-state analysis based on matrix-free Krylov-subspace methods. *Proceedings of the 32<sup>nd</sup> Design Automation Conference*, June 1995.