RF Analysis with Virtuoso Spectre Simulator

Engineer Explorer Series

Version MMSIM 7.2

Lecture Manual **August 4, 2010**

cādence®

Cadence Design Systems, Inc., 2655 Seely Avenue, San Jose, CA 95134, USA

Cadence Trademarks

Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence trademarks, contact the corporate legal department at the address above or call 800.862.4522.

Allegro®
Accelerating Mixed Signal Design®
Assura®
BuildGates®
Cadence®
CeltIC®
Conformal®
Connections®
Diva®
Dracula®
ElectronStorm®
Encounter®
EU CAD®
Fire & Ice®
First Encounter®
HDL-ICE®
Incisive®
InstallScape™
IP Gallery™
NanoRoute®
NC-Verilog®
NeoCell®
NeoCircuit®
OpenBook® online documentation library
OrCAD®
Palladium®
Pearl®
PowerSuite®
PSpice®
SignalStorm®
Silicon Design Chain™
Silicon Ensemble®
Silicon Express™
SKILL®
SoC Encounter™
SourceLink® online customer support
Specman®
Spectre®
Speed Bridge®
UltraSim®
Verifault-XL®
Verification Advisor®
Verilog®
Virtuoso®
VoltageStorm®
Xtreme®

Other Trademarks

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

All other trademarks are the property of their respective holders.

Confidentiality Notice

No part of this publication may be reproduced in whole or in part by any means (including photocopying or storage in an information storage/retrieval system) or transmitted in any form or by any means without prior written permission from Cadence Design Systems, Inc. (Cadence).

Information in this document is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence customers in accordance with a written agreement between Cadence and its customers. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

RESTRICTED RIGHTS LEGEND Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

UNPUBLISHED This document contains unpublished confidential information and is not to be disclosed or used except as authorized by written contract with Cadence. Rights reserved under the copyright laws of the United States.

Table of Contents

RF Analysis with Virtuoso Spectre Simulator

RF Analysis with Virtuoso Spectre Simulator

Engineer Explorer Series

Version MMSIM 7.2

August 4, 2010

Blank Page

About This Course

Module 1

August 4, 2010

Course Objectives

This course explores applications of the Shooting Newton and Harmonic Balance engines used for RF analysis in the Virtuoso® Spectre® Circuit Simulator environment.

In this course, you

- Use and apply the *periodic steady state* (PSS) analysis algorithm, its options, and properties
- Use and apply the *quasi-periodic steady state* (QPSS) analysis algorithm, its options, and properties
- Apply accelerators and convergence aids to maximize efficiency
- Use the *linear periodic time varying* algorithms
- Use and apply *periodic noise* analysis to driven and autonomous circuits
- Apply strategies for autonomous circuits
- Run and verify full receiver chains efficiently

Course Agenda

Day 1

Module 1 – About This Course

Lab 1-1 Locating Cadence Online Support Solutions

Lab 1-2 Customizing Notification and Search Preferences

Module 2 – RF in the Spectre Environment

- Shooting and Harmonic Balance (HB) Engines
- Using Sufficient Harmonics with HB
- LNA PSS Multi-Tone Example

Lab 2-1 Running an LNA PSS Simulation

Lab 2-2 Running an LNA IP3 Simulation

Module 3 – Quasi-Periodic Steady State (QPSS) Analysis

- QPSS for multitone simulation
- Harmonic Trimming, Diamond and Funnel Cut

Module 3 – Quasi-Periodic Steady State (QPSS) Analysis (continued)

- MaxImOrder and MaxPeriods
- Swept QPSS

Lab 3-1 Running a QPSS Analysis

Lab 3-2 Measuring Composite Triple Beat

Module 4 – The Time Domain Solver

- Overview of the DC Engine
- Understanding a Time Domain Engine for TRAN and RF
- Convergence aids

Module 5 – RF Accelerators

- Transient Assist – TSTAB
- SaveFile – Recover
- TSTAB and Dividers
- Turbo and Parasitic Reduction

Course Agenda (continued)

Day 2

Module 6 – Linear Periodically Time Varying Circuits

- Understanding Linear Analysis over PSS/QPSS
- Periodic AC Analysis
- PAC Mixer Example
- Periodic Transfer Analysis

Module 7 – Noise Analysis

- How to set up PNOISE
- Understanding Reference Sideband
- Noise Summary
- Jitter from PNOISE
- Time Domain PNOISE
- Transient Noise

Module 7 –Noise Analysis (continued)

Lab 7-1 Simulating Conversion Gain and Noise in Mixers

Lab 7-2 Running Mixer Compression and Desensitization Measurements

Lab 7-3 Multiple Pnoise Using APS

Lab 7-4 Running a Mixer IP3 Simulation

Lab 7-5 Calculating Jitter

Lab 7-6 Analyzing Transient Noise with PSD and DFT

Course Agenda (continued)

Day 3

Module 8 – Voltage Controlled Oscillators (VCOs)

- Setting up Autonomous Simulations
- Convergence aids
- Pnoise for Oscillators

Lab 8-1 Running Oscillator Simulations in the Turbo and APS Mode

Lab 8-2 Analyzing Linear and Periodic Stability

Lab 8-3 Analyzing Semi-Autonomous Harmonic Balance with HB Noise

Lab 8-4 Applying Parasitic Reduction

Lab 8-5 Putting It All Together: Simulating a Full Receiver Chain

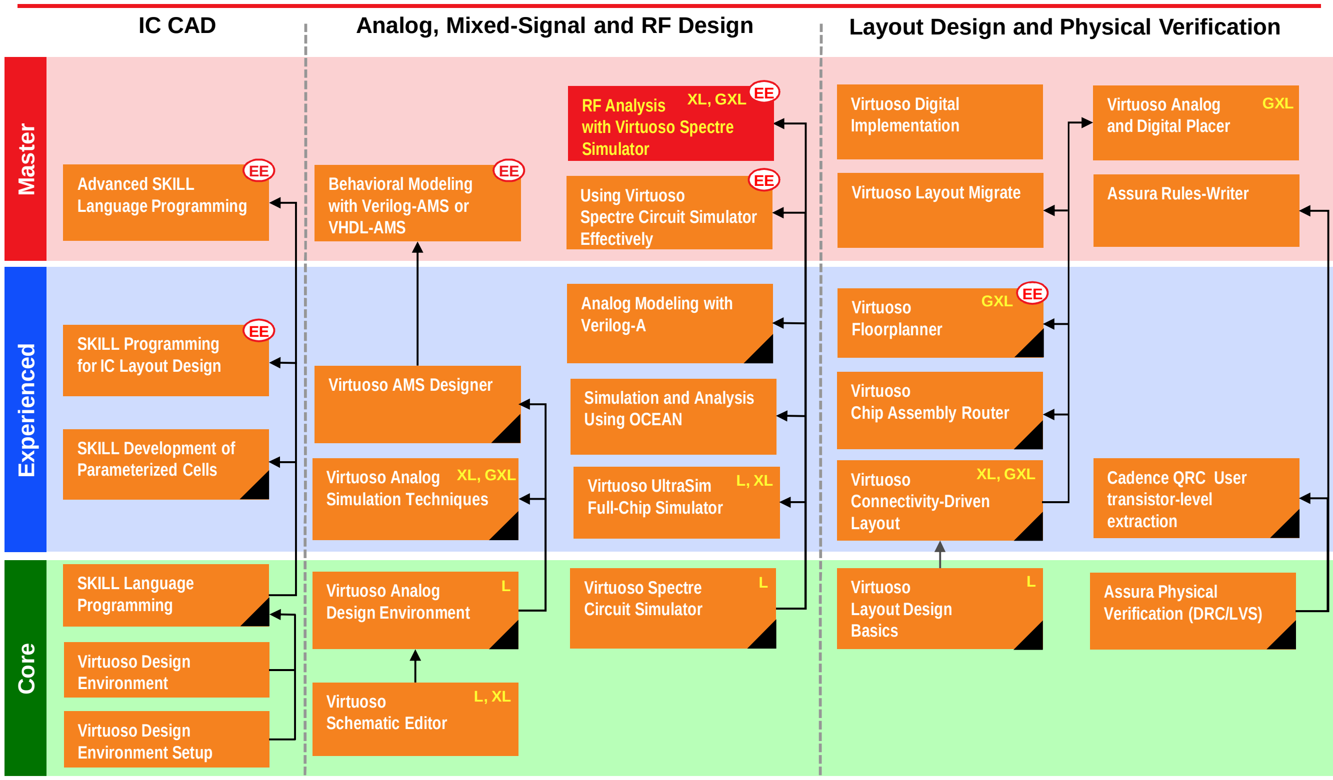
Module 9 – Envelope Analysis

- Shooting and Harmonic Balance
- ACPR Wizard
- Options

Lab 9-1 Running Envelope Analysis

Learning Map for

Custom Design with Virtuoso® and Assura® Technology



Some course titles may vary. Please refer to your regional catalog for exact titles and course datasheets.

L, XL, and GXL denote the tiers of Cadence products.

Also available as an Internet Learning Series course (online)

EE This is an Engineer Explorer Course.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 8

For more information about Cadence® courses:

1. Go to www.cadence.com.
2. Click **Support & Training**.
3. Choose one of the catalogs to explore.
   - Europe, Middle East, and Africa
   - India
   - North America
   - China
   - Japan
   - Korea
   - Taiwan

Getting Help

There are three ways to get help:

- Cadence Help: Accessing tool-specific help
  - cdnshelp
- Cadence Online Support: Accessing documents and support
  - http://support.cadence.com
- Cadence Community: Staying current with tips and tricks
  - http://www.cadence.com/community/forums
  - http://www.cadence.com/community/blogs
  - http://www.cadence.com/cdnlive

Cadence Help

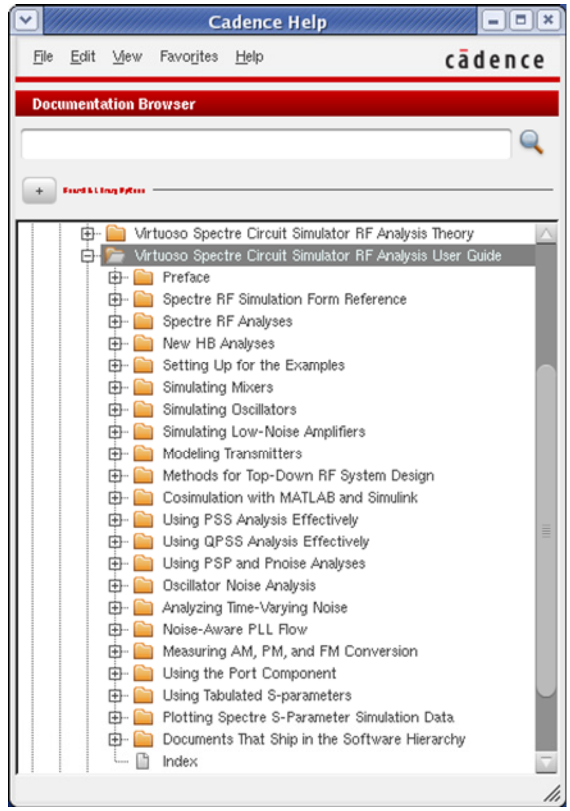
Cadence Help gives you access to the Cadence online product documentation system.

Documentation for each product is included automatically when you install the product. Documents are available in both HTML and PDF format.

The Library window lets you access documents by product family, product name, or type of document.

You can access Cadence Help from

- The graphical user interface
  - The Help menu in windows
  - The Help button on forms
- The command line: *cdnshelp*
- The Cadence Online Support system (if you have a license agreement)



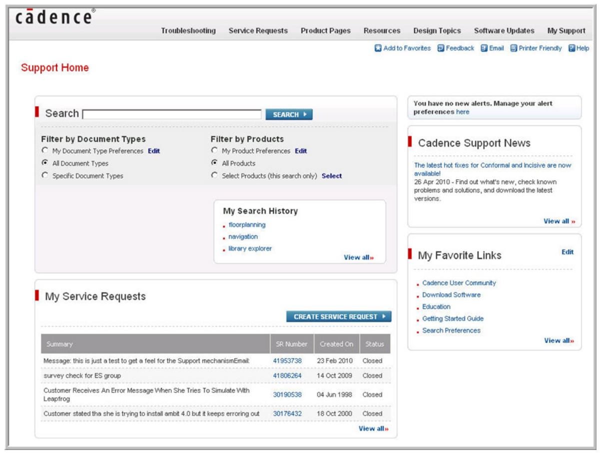
- To access Cadence Help from a Cadence software application, from the menu, choose **Help—Cadence Documentation**. The browser displays the document page. After the document page opens, click the **Library** button to open the Library window.
- To access Cadence Help from a command line, enter *cdnshelp &* in a terminal window. When the Library window appears, navigate to the manual you want by selecting the specific product, then double-click to open the manual in your browser.

Cadence Online Support (COS)

Cadence Online Support (http://support.cadence.com) is a website that gives you access to support resources, including

- An extensive knowledge base with
  - User guides
  - Reference manuals
  - Design topics
  - Frequently asked questions
  - Known problems and solutions
  - White papers
  - Application notes
- Software updates for Cadence products
- Access to Cadence customer support engineers

Register now and take advantage of the many benefits of Cadence online support.



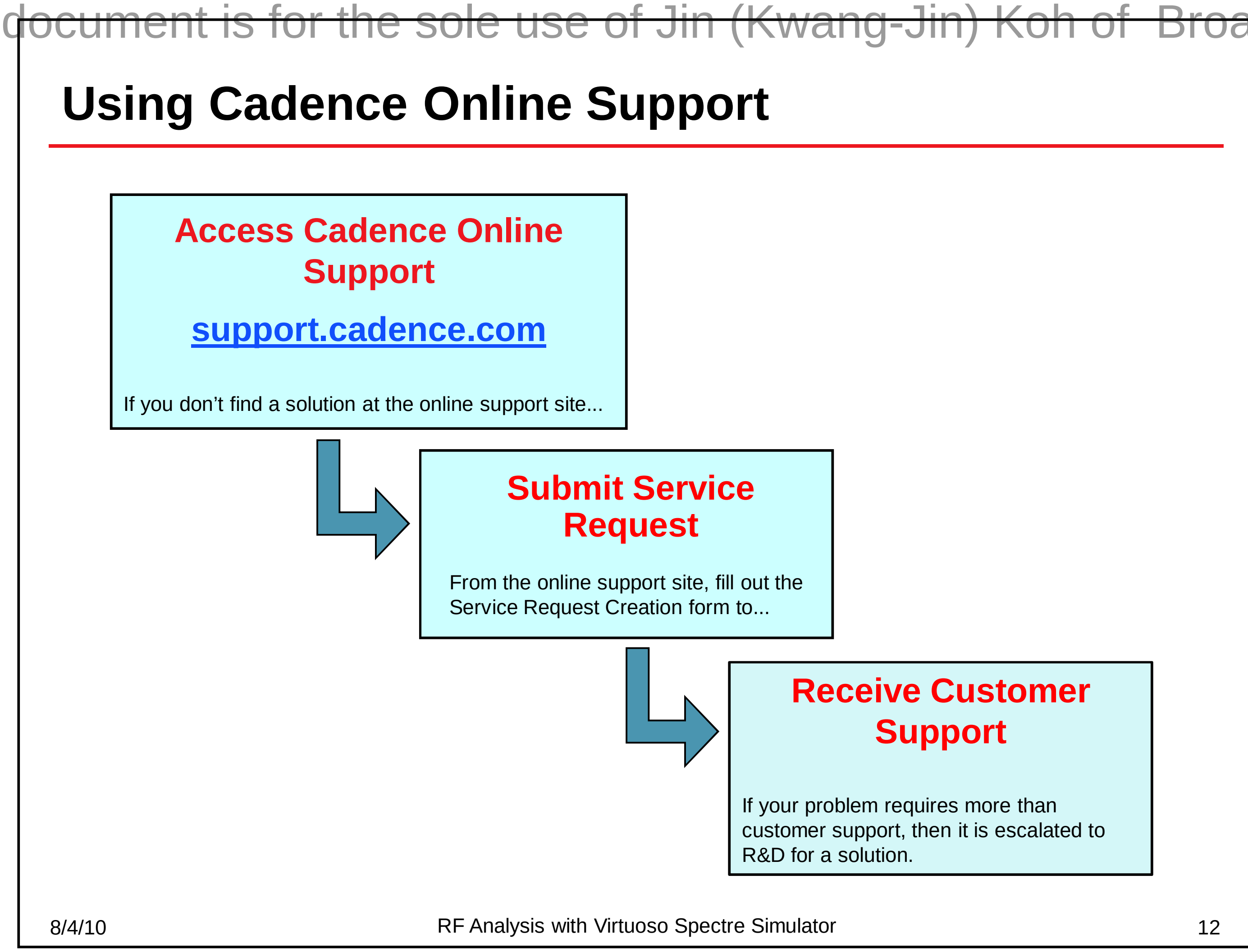
Searching the Knowledgebase

You can personalize the document types and products by editing your preferences.

To search the knowledge base for different document types:

1. Point your web browser to **support.cadence.com**.
2. Log in to the online support site.
3. Enter search criteria.
4. Specify document type for current search.
5. Select products for this search only.
6. Click Resources menu to access:
   - Application Notes
   - Computing Platforms
   - Installation Information
   - Product & Release Lifecycle
   - Product Manuals
   - SKILL Information
   - Tech Info and White Papers
   - Links to Other Resources
   - Troubleshooting Information
   - Video Library

When search results are displayed, you can use the *within* option to refine your search. You can also filter your results based on the year under *Date Filters*.



You can use the Cadence Online Support (COS) site for service requests. Fill out the Service Request Creation form and submit it. The request goes first to Customer Support, and if necessary is escalated to Cadence R&D for a solution.

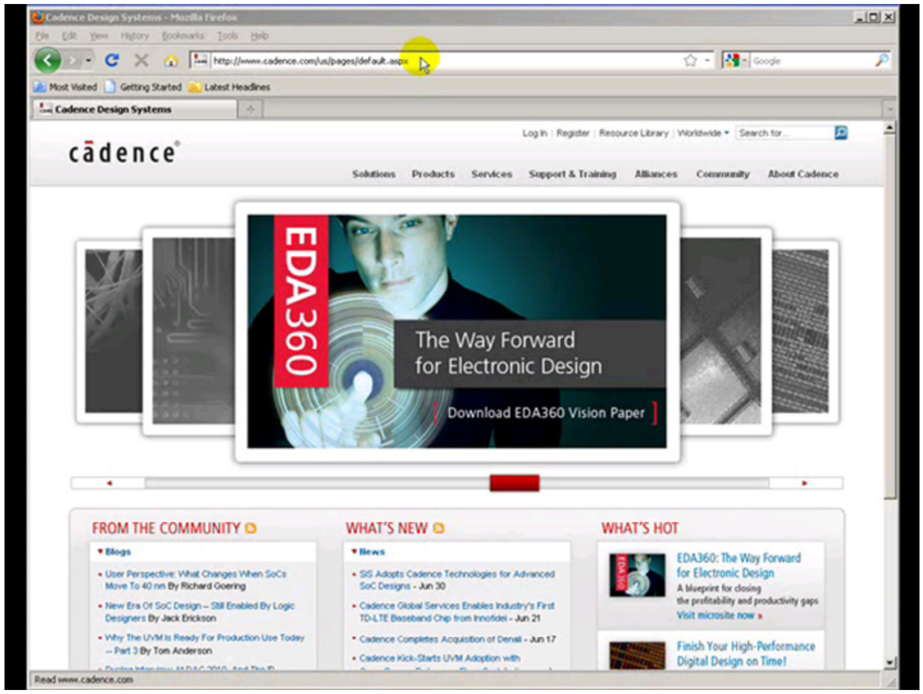
Submitting a Service Request

1. Point your web browser to **support.cadence.com**.
2. Log in to the online support site.
3. Click **Create Service Request** to interact directly with Cadence Customer Support.
4. Select the product from the list of products.
5. Describe the problem.
6. Click **Continue**.
7. View documents that might solve your stated problem.
8. Set additional request attributes.
9. Click **Submit SR**.

Demo: Cadence Online Support

The demo at right shows you how to:

- Register for Cadence Online Support
- Search the knowledge base
- Submit a service request
- View your service request



Click the graphic to play the demo.

This section outlines how to register for COS.

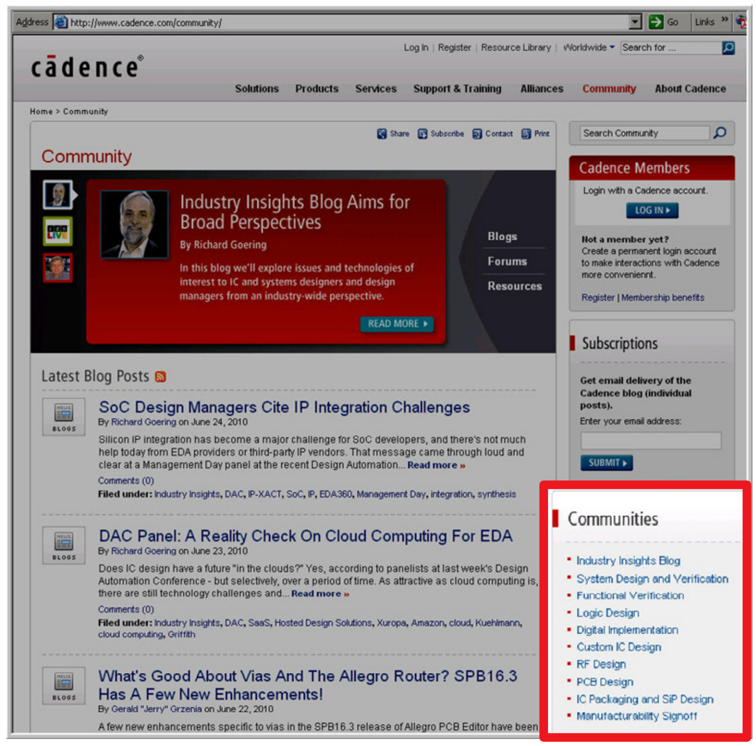
Registering for Cadence Online Support

1. Go to http://support.cadence.com.
2. Click the “Register Now” link.
3. You need:
   - Email address you use at work
   - You might also be asked for your license server Host ID or Reference Key.
4. After registering, check your email.
5. Use your email ID and the one time password provided in the email to log in.
6. Then change your password, security question, and personal data.

Cadence Online Communities

Stay connected by visiting resources, such as blogs and forums.

1. Go to http://www.cadence.com/community
2. Select your area of interest.



RF Analysis with Virtuoso Spectre Simulator 14

Lab Exercises

Lab 1-1 Locating Cadence Online Support Solutions

Lab 1-2 Customizing Notification and Search Preferences

Blank Page

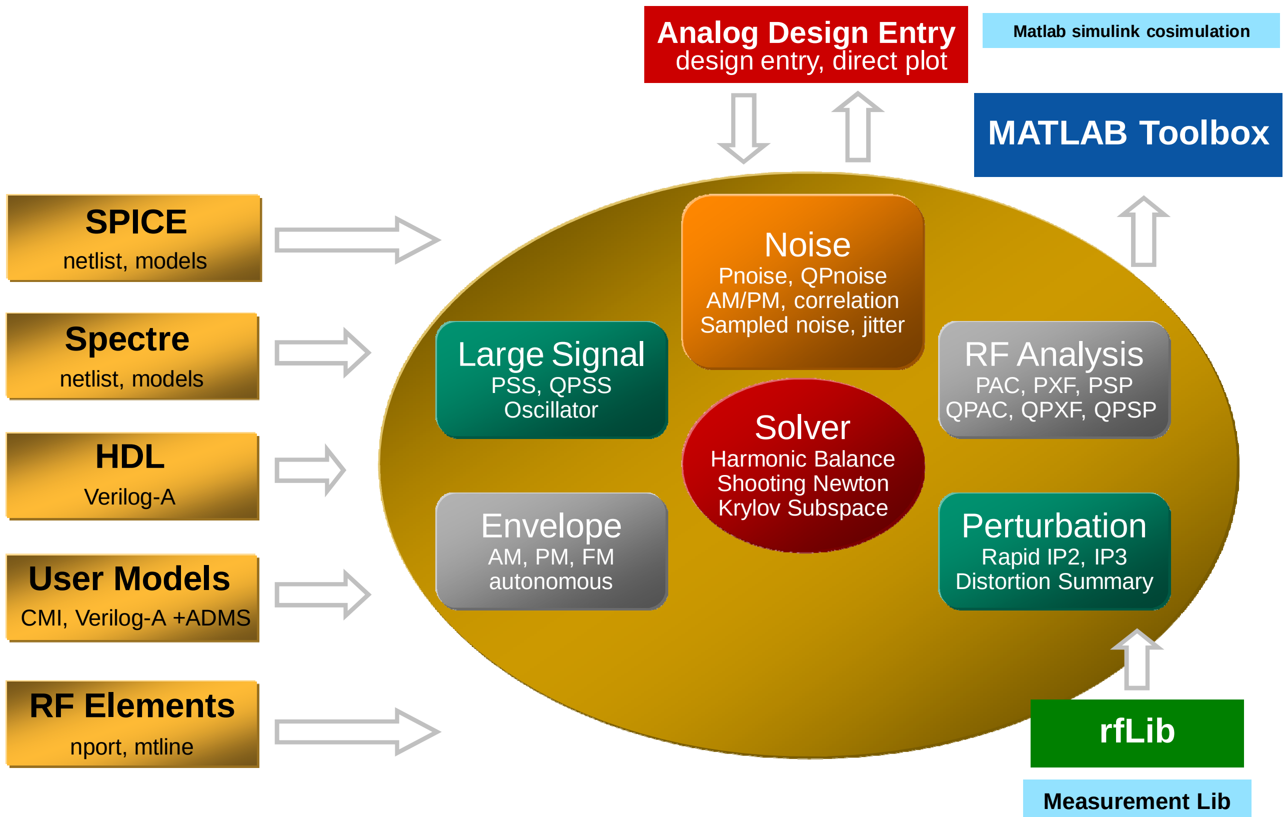
cādence

RF in the Spectre Environment

Module 2

August 4, 2010

Virtuoso Spectre RF: Supporting Technology



8/4/10 RF Analysis with Virtuoso Spectre Simulator 18

This page shows key components in Spectre® RF simulation engine and how the engine is related with the supporting environment.

Spectre RF shares the same interface and extension components as the Virtuoso® Spectre Circuit Simulator engine, including netlist, device models, CMI, Verilog®-A, nport, mtline, etc.

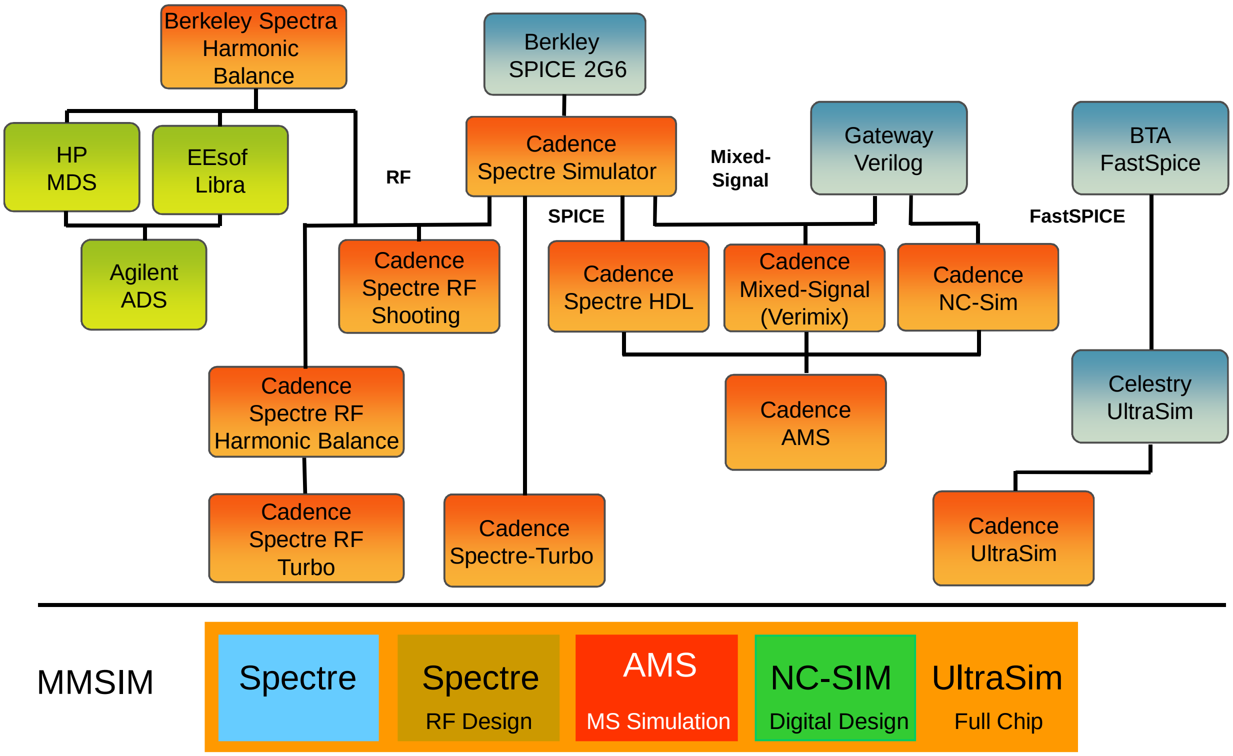
Spectre RF works seamlessly with the Virtuoso Analog Design Environment (ADE). In addition to schematic entry and modification, RF simulation testbench components such as driving signals and postprocessing plots are conveniently set up through ADE. ADE provides a rich offering in direct plot. RF measurements such as phase noise, gain circle, IP3 are just a click away.

rfLib provides behavioral blocks for top-down design support. RF simulations can be run with Monte Carlo and parametric analyses through ADE.

RF simulation engine provides a rich selection for RF measurements. It supports a wide variety of applications through suite of large signal analyses, small signal analyses, perturbation analyses. It provides full set of noise analyses for all noise measurement needs including AM/PM, sampled noise, jitter, noise correlation.

The core solver of the engine is built on patented time domain shooting technology and Krylov subspace technology which enables large simulation capacity. Simulation efficiency is greatly enhanced by recycling, preconditioning and inexact-Newton techniques.

Simulation Lineage (A Bit of History)



8/4/10 RF Analysis with Virtuoso Spectre Simulator 19

Dr. Ken Kundert is responsible for creating many important software tools in the simulation field.

He is known for creating two highly successful circuit simulators: Cadence® Virtuoso Spectre Circuit Simulator and the harmonic balance simulator from Agilent. From 1989 to 2005, Ken worked at Cadence Design Systems as a Fellow. He created the Spectre simulator and was the principal architect of the Spectre circuit simulation family. As such, he led the development of Spectre, Spectre HDL, and Spectre RF. He also played a key role in the development of the Cadence® AMS Designer and made substantial contributions to both the Verilog®-AMS and VHDL-AMS languages.

While in school he authored Sparse, an industry standard sparse linear equation solver and created the Agilent harmonic balance simulator.

Before that Ken was a circuit designer at Tektronix and Hewlett-Packard, and contributed to the design of the HP 8510 microwave network analyzer. He has written three books on circuit simulation: The Designer's Guide to Verilog-AMS in 2004, The Designer's Guide to SPICE and Spectre in 1995, and Steady-State Methods for Simulating Analog and Microwave Circuits in 1990, and created The Designer's Guide Community website. He also holds 11 patents and has authored three dozen papers published in refereed conferences and journals.

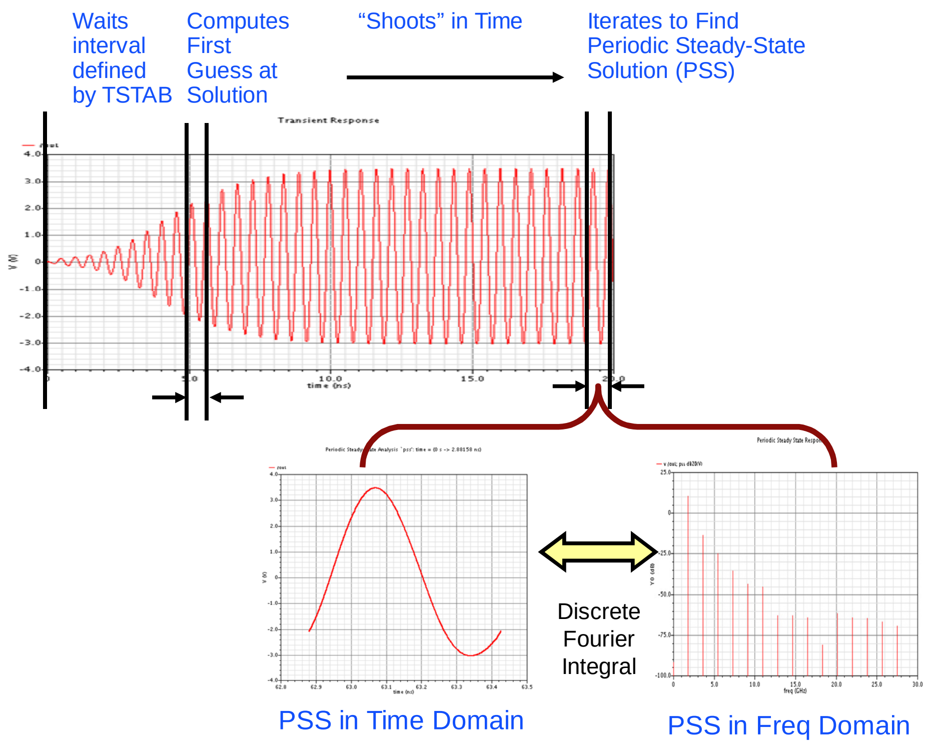
For more information pertaining to Spectre RF, go to his website, *www.designers-guide.org*.

The RF in Spectre: PSS (Periodic Steady-State)

- In RF circuit simulations you often encounter nonlinear (translation from one frequency to another) circuits and notice harmonics due to the carrier being a periodic signal.
- PSS Analysis is used to calculate the response of a circuit to large signals. It can be used for both, autonomous (such as oscillators) and nonautonomous (driven) circuits.
- All the harmonics are calculated in PSS, thus when there are 2 closely spaced input frequencies, QPSS (or HB) should be used because only the mixing harmonics are calculated.
- PSS Analysis utilizes the *Shooting Method* or *Harmonic Balance* algorithms in the time domain and frequency domain respectively. As the voltages and admittances are saved during simulation at each time point, this gives the analysis access to the waveshapes in the circuit.

The algorithm methods are discussed on other pages.

The RF in Spectre: PSS (Periodic Steady-State)



There are two phases to the PSS analysis:

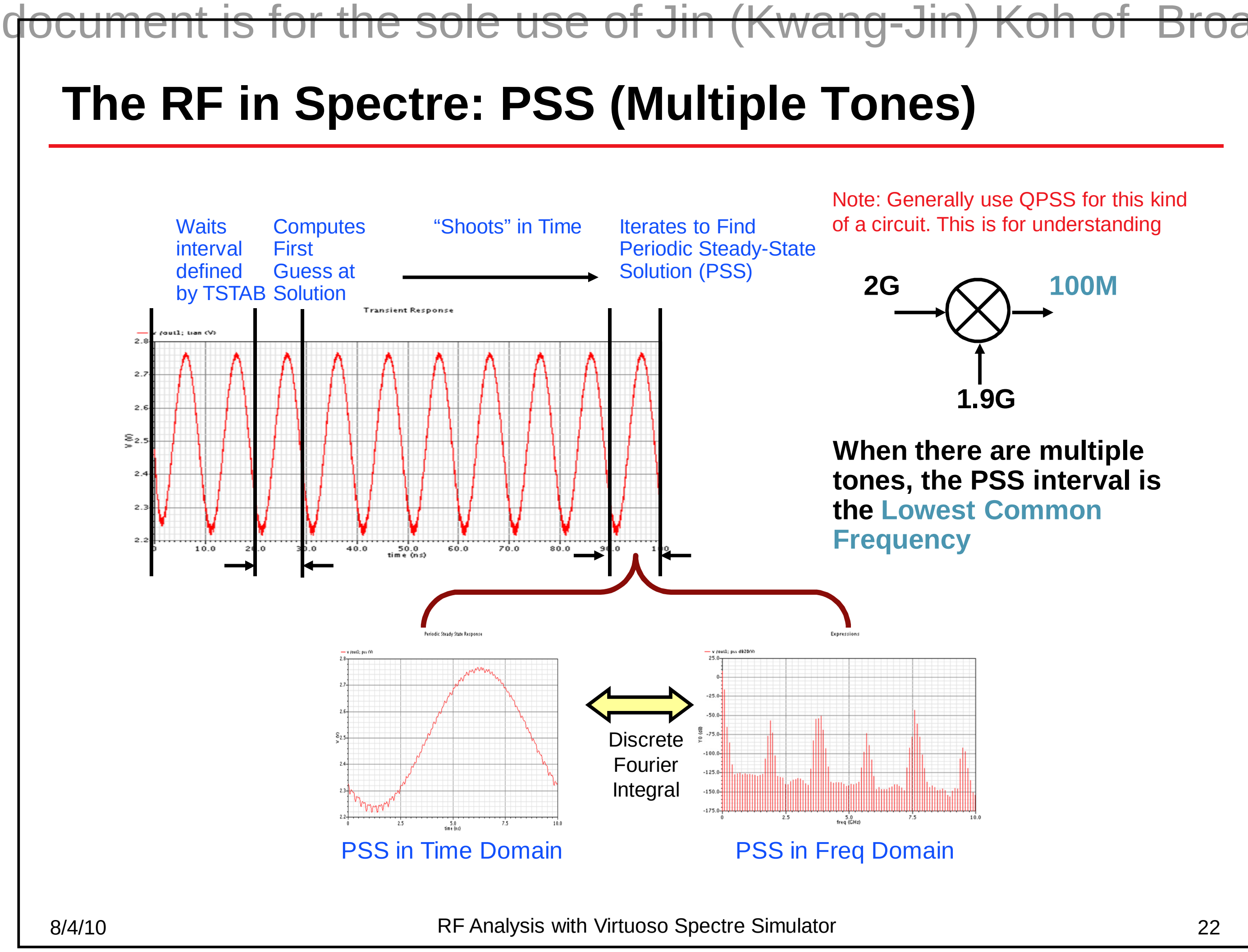
- In the figure above, you see an initial transient phase that initializes the circuit, followed by a *shooting* phase that computes the periodic steady state solution.

The first transient phase of simulation is divided into three or four intervals:

- The simulation starts at *tstart* (normally 0) and continues through the minimum time for which all sources are periodic.
- After this as a user you can define an optional stabilization interval using *tstab*.
- For autonomous circuits, 6.5 periods of the specified frequency is run. This is to see if there is a digital frequency divider is in the circuit.
- The third final interval is the *period* for driven circuits.

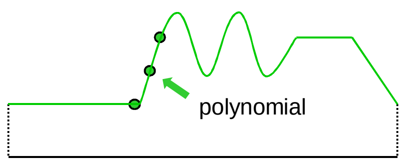
In the second shooting interval phase, the circuit is repeatedly simulated over one period while adjusting the initial condition (period applied in case of autonomous circuits) to obtain the periodic steady state solution.

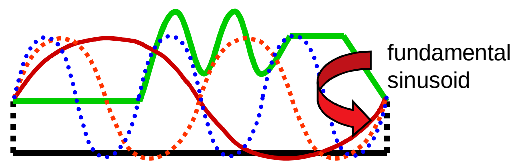
The Spectre simulator internally computes the response of the circuit in the time domain and converts the results into the frequency domain using a discrete Fourier integral approach.



- Above you see an example of circuits having multiple tone frequencies. A different fundamental frequency can be selected as long as it is an integral multiple of the period of each built-in, time-varying independent source in the circuit.
- QPSS is normally used for such a circuit. The example provided is to illustrate how for multiple tones in a circuit the lowest common frequency is chosen as the PSS interval in the second phase of the analysis.
- The algorithm simulates one period of the common frequency. The voltages and slopes at the start of the PSS shooting interval are adjusted and another period is simulated.
- For an accurate estimate of the initial condition for another iteration, the final state of the circuit must closely match its initial state.

Two Engines: Both Solving PSS





Shooting Newton

- Low-Order time-domain (Gear, Trap)
- Solutions represented by piecewise low-order polynomials in time domain
- Derivatives approximated by difference formula

$$\left.\frac{dv}{dt}\right|_{t=t_i} = c_1 v(t_{i-2}) + c_2 v(t_{i-1}) + c_3 v(t_i)$$

Harmonic Balance

- Represents solution with sum of sines, cosines (Fourier series)
- Solutions represented by global sinusoidal basis functions in frequency domain
- Derivatives also sine waves

$$\frac{de^{j\omega t}}{dt} = j\omega e^{j\omega t}$$

8/4/10 RF Analysis with Virtuoso Spectre Simulator 23

The **Shooting method** is an iterative time domain algorithm that is akin to solving a boundary value problem by reducing it to the solution of an initial value problem.

- The algorithm works on a circuit by using an initial condition that is calculated by the tstab interval. It then uses iterative methods to result in a final steady state solution.
- On comparison after the first iteration, if the final state is a not equal to the initial state, then the non-periodicities and the sensitivities are used to compute a new initial condition and additional iterations are needed.
- If the final state is the same as the initial state (with a small error band), then the periodic response of the system has been calculated and iterating stops.

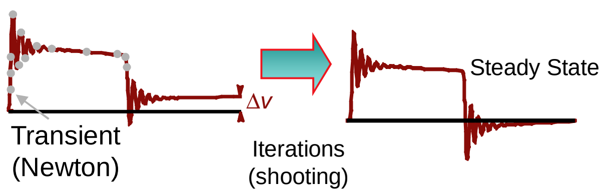
Harmonic balance is an algorithm that is used to find the steady state response in the frequency domain.

- Harmonic balance can be transient-assited if tstab is set. In this case, a transient is run for the specified time, and a fourier transform is computed for the signal. This becomes the starting point of the frequency domain iterations.
- If tstab is left blank, the starting condition is the DC solution, or the solution computed by the linear *oscic* if it's an oscillator. (*oscic* stands for oscillator initial condition.)
- Iterations are made in the frequency domain until the sum of all the harmonics at all the nodes is near zero for all the nodes.
- When the solution is calculated, the time domain waveform is calculated using an *ifft*.

Two Engines: How They Work

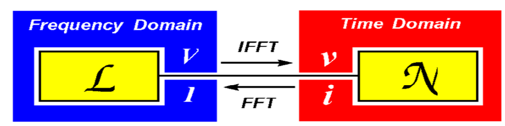
Shooting Newton Procedure

- Estimate an initial condition
- Run transient analysis for one period of the beat (T)
- Compare the final state to the initial state *ΔV = vf - vi*
- Estimate settling behavior based on the solution matrices during time interval
- Iterate until converged *ΔV = 0* and *ΔI = 0* at the beginning and end of the shooting window.



Harmonic Balance Algorithm Summary

- Formulates circuit equations in frequency domain
- Evaluates linear devices in the frequency domain
- Evaluates nonlinear devices in the time domain (intrinsic nonlinearities).
- Converts between domains with FFT & IFFT
- Iterates until the sum of the currents for the harmonics we are solving for at every node is near 0 in the frequency domain.



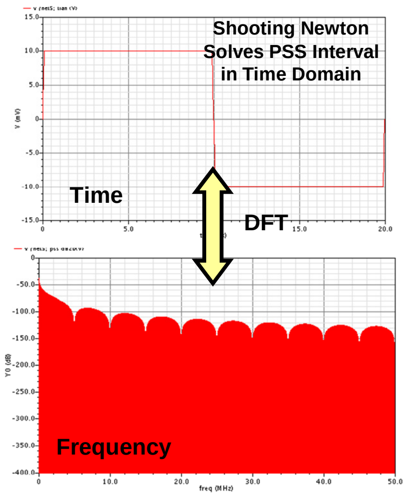
8/4/10 RF Analysis with Virtuoso Spectre Simulator 24

- Shooting Newton should be used for very nonlinear circuits like sampling circuits or switched-capacitor circuits.
- The algorithm runs up to 20 iterations by default. Typically, shooting Newton requires about 5 to 15 iterations on a circuit in practice.
- Harmonic balance is faster for simulation of linear and weakly nonlinear circuits. When multiple tones are present, either harmonic balance QPSS or HB should be used for the simulation, because it requires less time and memory to complete.

Comparing the Spectrums of the Two Engines

Shooting Newton

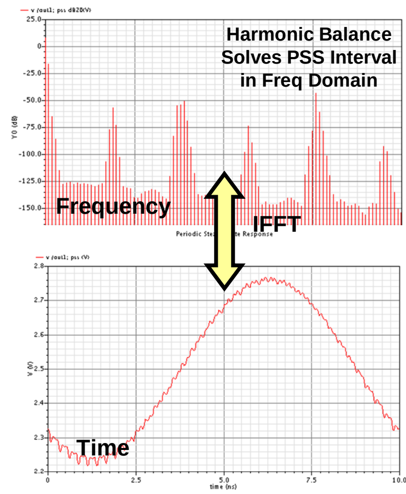
Analysis of strongly nonlinear circuits



- Shooting is more efficient when there are very sharp edges in the system being simulated..
- Generally, we recommend Shooting to be used with PSS only.

Harmonic Balance

Fast simulation of RF transceivers



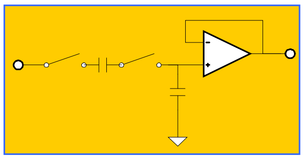
- Harmonic Balance (HB) is very efficient when the spectral density is sparse or nearly linear..
- Harmonic Balance can be used in PSS, QPSS, or HB.

All large signal (PSS, QPSS), small signal, noise and envelope analysis can be performed using either algorithm in a similar use model.

Which Engine Should I Use?

Harmonic balance and shooting Newton are complementary.

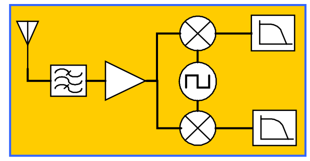
Shooting Newton
Analysis of strongly nonlinear circuits



Generally, **shooting Newton** works well for

- Strongly nonlinear circuits
  - Ring oscillators
  - Sampling Circuits
  - Switched Capacitor Circuits
  - Frequency Dividers
- Input signals have sharp transitions

Harmonic Balance
Fast simulation of RF transceivers

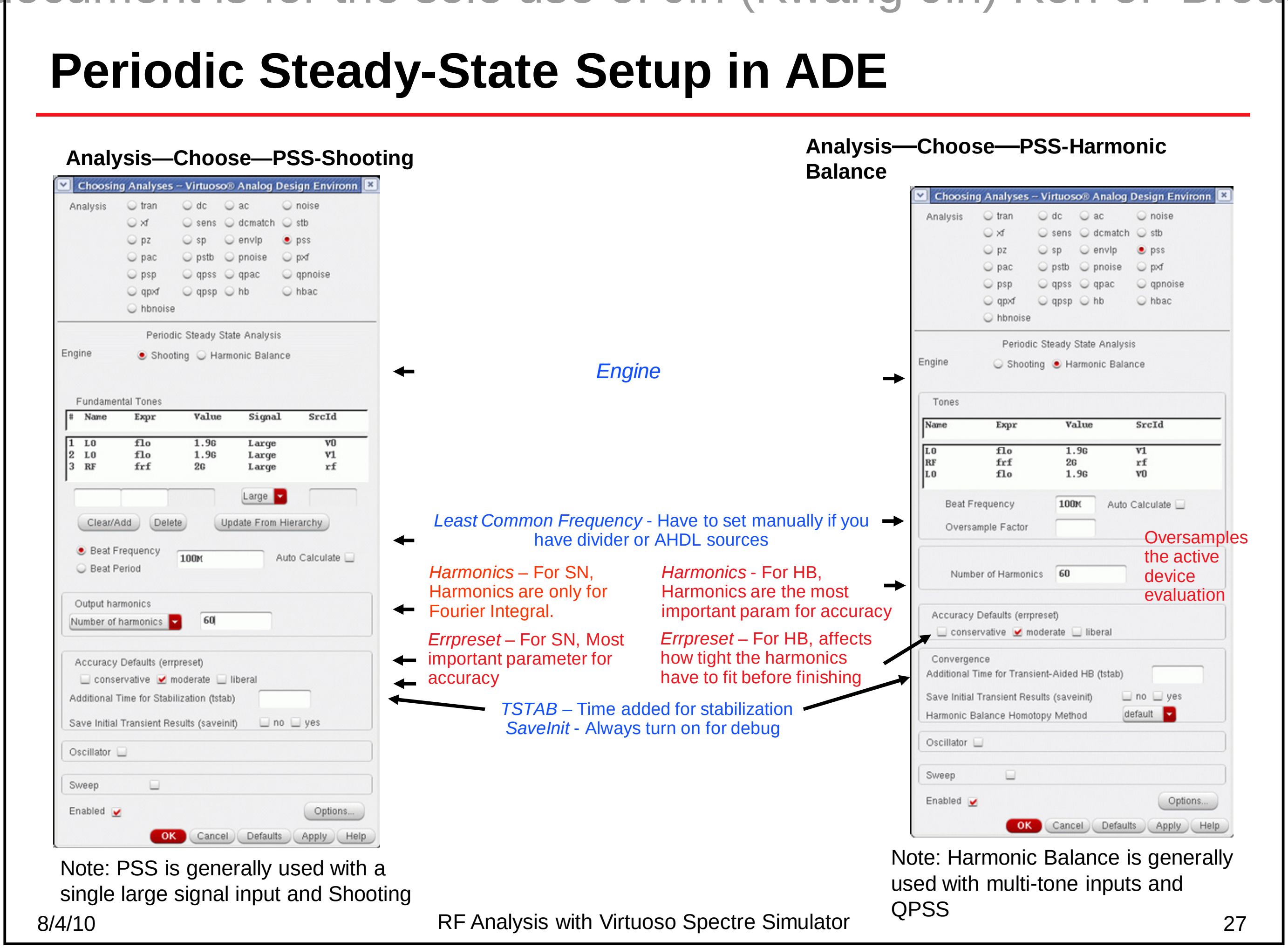


Generally, **harmonic balance** works well for

- Weakly nonlinear systems with a high dynamic range
  - RF front-ends (LNA, Mixer)
  - IQ modulators
  - LC and crystal oscillators
- Circuits with distributed components (transmission lines, S-parameter models)

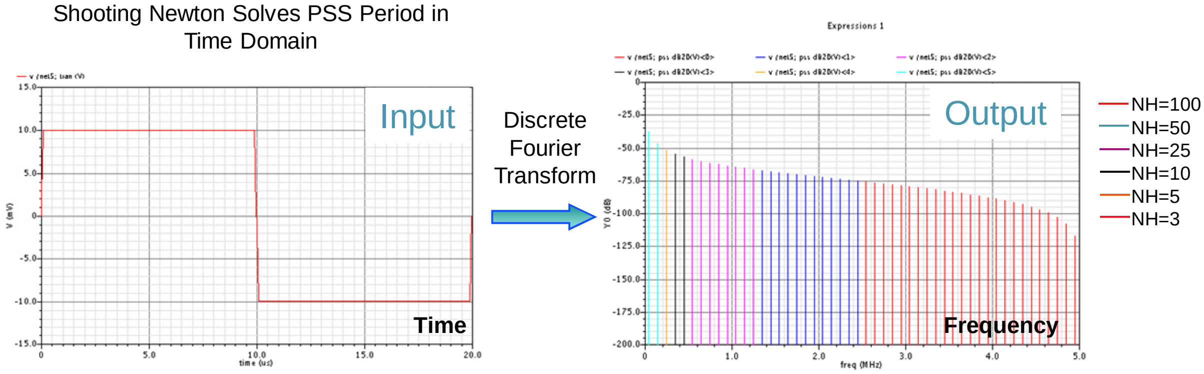
8/4/10 RF Analysis with Virtuoso Spectre Simulator 26

- Depending on the linearity and type of circuit, you can choose the algorithm suitable for PSS simulation also bearing in mind the accuracy and simulation time.
- In harmonic balance, choosing a larger number of harmonics increases the simulation time, but gives good expected results.



- The above page shows the setup form of the PSS Analysis in ADE for both the *shooting* and *harmonic balance* algorithms.
- There is an option provided to *Auto Calculate* the *Beat Frequency* from the sources in the circuit.
- The *tstab* option is not always required but in cases where convergence problems exist because the algorithm is converging at a peak or valley, the tstab value can be adjusted to move the start point off the peak or valley thus aiding in convergence of PSS Analysis.
- The *Oversample Factor* option in the harmonic balance setting is used for circuits that have relatively sharp edges. Setting Oversample Factor to a positive integer makes the frequency domain solution more accurate by forcing more timepoints in the device evaluations.

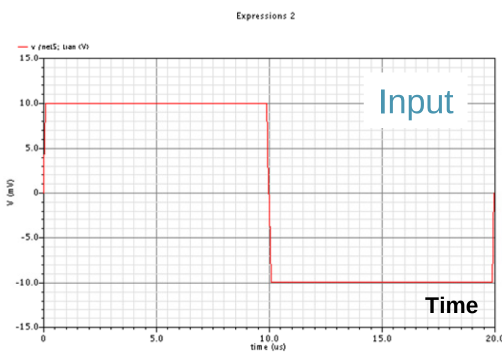
Displaying Spectrums: PSS – Shooting Newton

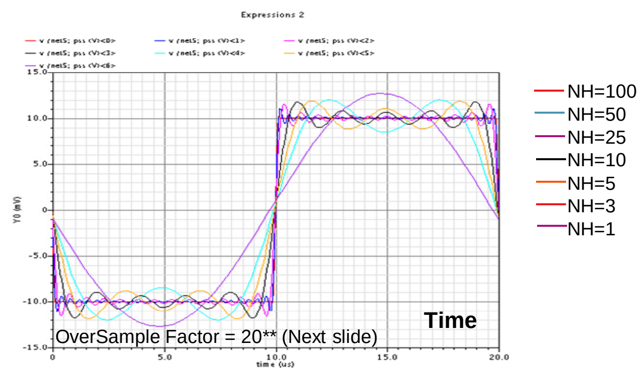


- The number of harmonics is for the display of the *discrete Fourier integral* (DFI) and does not affect the accuracy of the solution. (To the first order)
- The resulting spectrum of PSS analysis is Fourier integral based rather than FFT based.
- The Fourier integral accepts unequally spaced time points, but will be more accurate if evenly spaced.
- For example, when you specify 3 harms you get the right answer for those 3 harms because all of the time points are used to calculate the Fourier transform. This is different than an FFT where you get the triangular sample.

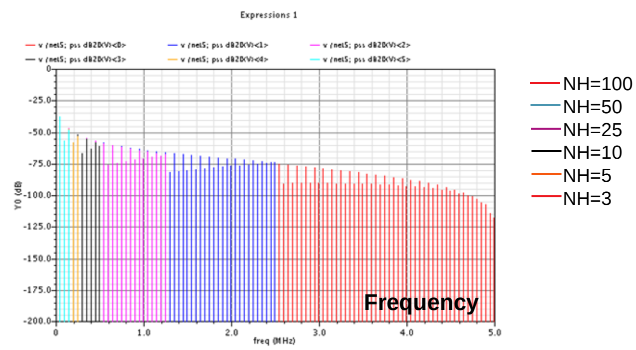
- Above you can see the PSS solution using Shooting Newton after the first initial phase of transient analysis for a 20 uS period in the time domain.
- The spectral content is correct (unlike an FFT) even if the number of harmonics is underspecified. This is a result of using a Fourier integral-based approach.
- The different colors of the spectral components on the right are the superimposed waveforms from different numbers of harmonics being set in the number of harmonics field.

Using Sufficient Harmonics: PSS – Harmonic Balance



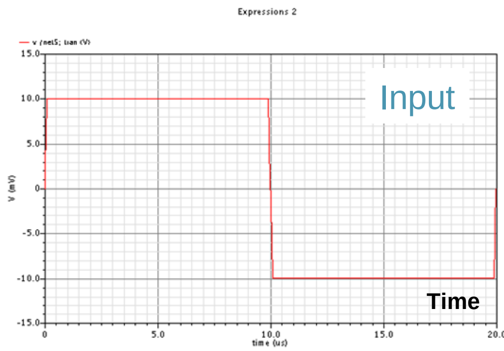


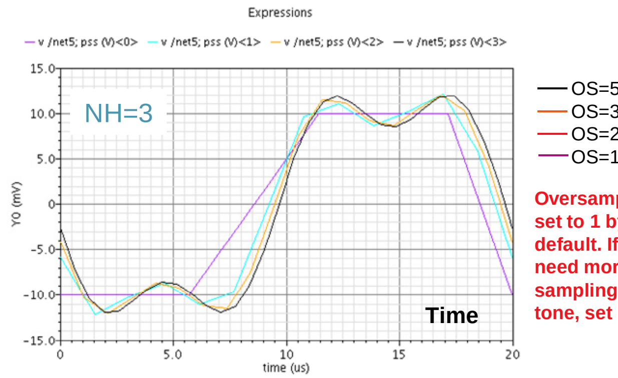
- If not enough harmonics are specified, then the results will be inaccurate.
- Harmonics should be set to the period of the signal divided by the fastest risetime in the circuit.
- The HB solution is energy conserving. If you do not have the right amount of harmonics defined, energy from the upper part of the spectrum will “fold-back” into the number of harmonics you defined.



In this graphic, you can see the PSS solution using harmonic balance. Increasing the number of harmonics gives rise to a more accurate solution.

Oversample: PSS – Harmonic Balance



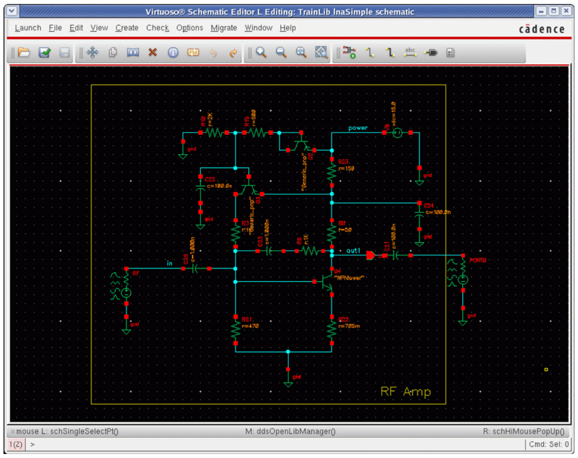


Oversample is set to 1 by default. If you need more sampling on a tone, set it to 2

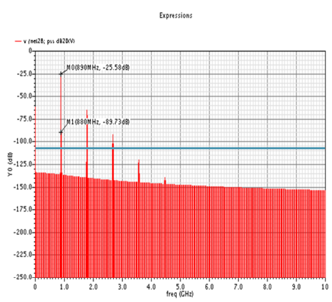
- Harmonic balance algorithm contains Fourier Transformation. If oversample=1 (which is the default) and harmonics number is K, then there will be (2K+1) samples in one period of the waveform; while if the oversample=S, there are (2K+1)*S samples. Increasing the number of oversamples will results in more sample points in time-domain, and thus HB spectrum and the following noise calculation will be more accurate.
- Different from the harmonic number, oversample only affects the number of samples in time-domain, and won't increase the number of unknowns (which is determined by equation and harmonic number). Thus increasing oversample is an effective way to improve accuracy without using too much memory as increasing harmonic number.
- Typically we only set Oversample =2 on highly nonlinear waveforms. Oversample > 2 can seriously increase simulation time.

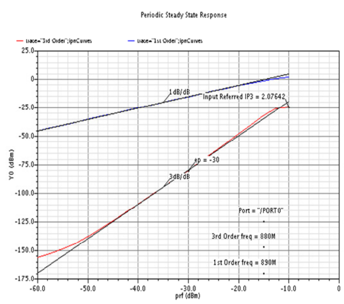
Oversampling with harmonic balance is an effective way of improving accuracy using a lesser number of harmonics.

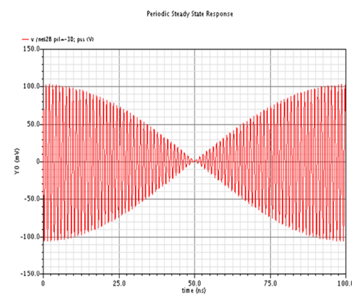
Simple LNA Example



- A Figure of Merit for the linearity or more importantly the Spurious Free Dynamic Range (SFDR) of the Receiver is a test called the IP3 (Third Order Intercept Point). This example is used as a spur generator that has a very low noise floor.
- Using PSS on this example is not the most efficient, but it allows you to see how the different settings affect the solution.





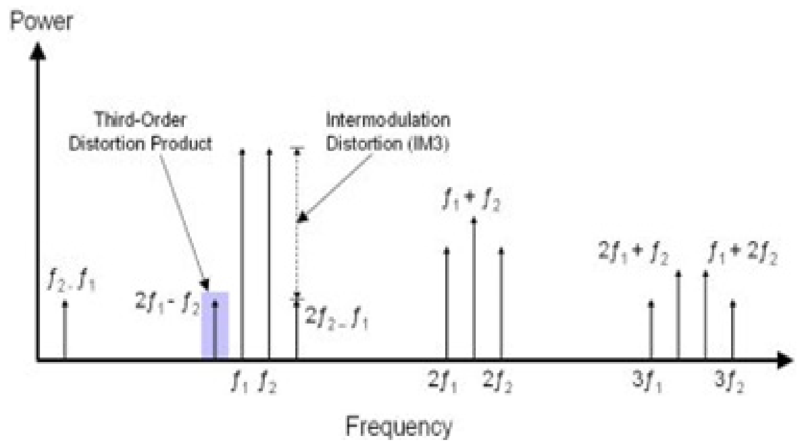


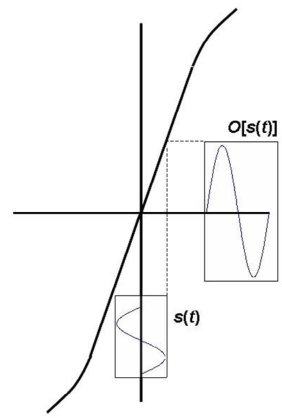
This shows a PSS shooting method for calculating IP3, which is generally quite slow. A better way is to use rapid IP3, which is much faster and just as accurate.

The third order intercept point (IP3) relates the third order term of this polynomial to the linearly amplified signal, as such it is a purely mathematical concept.

IP3 Example: Background

- A simple way to specify third order distortion products is with a two-tone intermodulation test.
- This test generates two tones of the same power level but with individual frequencies (typically within a few hundred kilohertz). Since there is non-linearity within the instrument, the two tones will be visible at the output along with their respective distortion products.
- These distortion products are closely related to the frequencies of the signal of interest and can be visualized in the figure below:





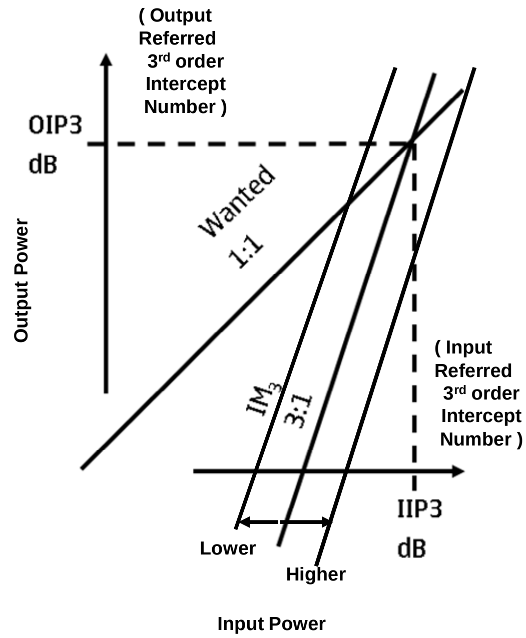
"Linear" device transfer function, with input and output signals

- Y-axis signal s(t) is the input
- X-axis signal O[s(t)] is the output
- O is the device transfer function that relates the output voltage level with the input signal level. It is usually an odd function of the input signal giving rise to a third-order harmonic.
- The IP3 is found at the input voltage level where the first and third-order terms have equal magnitudes.

- The frequencies of interest in the signal in the output power spectrum above are *f1* and *f2*.
- The figure on the right depicts a linear amplifier transfer function.
- The concept of intercept point is generally used for signals in the small-signal range for a receiver LNA or Mixer.
- For transmitters, there might be an IP3 spec at a signal in the large-signal range. This will not be a small-signal projection, and rapid IP3 cannot be used for this measurement. QPSS or multitone HB would be appropriate.

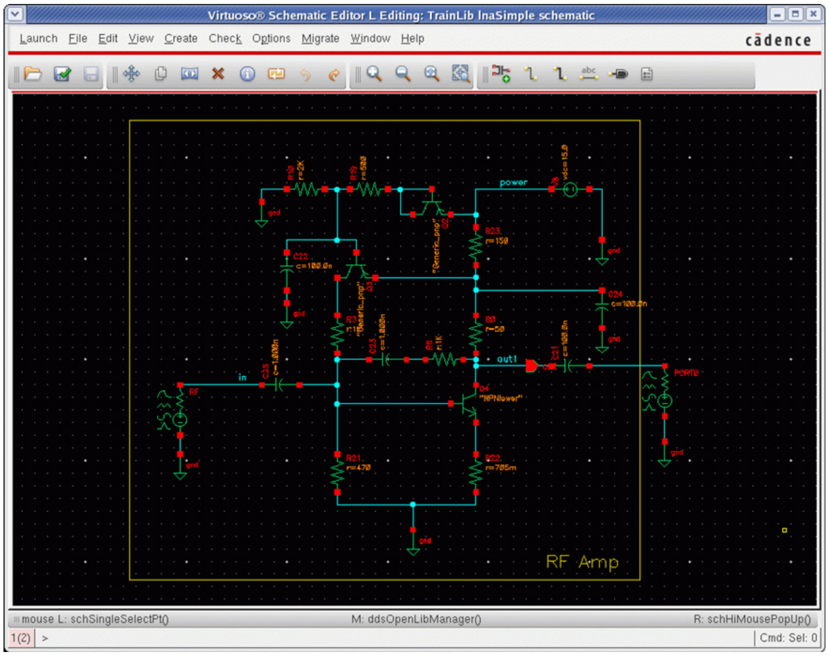
Small-Signal IP3 Example: Background (continued)

- By combining the intermodulation distortion with output power we can design a second specification, *third order intercept* (TOI).
- This theoretical specification is that the output power at which the distortion products will be equal in amplitude to the two-tone signal of interest.
- For every 1 dB increase in the fundamental tones, there is a corresponding 2 dB and 3 dB increase for the second- and third-order distortion products respectively.



- The graphical third-order intercept point is obtained by plotting the logarithmic (in dB) output power versus the input power.
- The *Wanted* signal is the input tone that is linearly amplified with a slope of 1 because on a logarithmic scale a function x^n becomes a straight line with a slope of n.
- So, when the input power is raised by 1 dB, the second- and third-order distortion products increase by 2 dB and 3 dB respectively.
- The IM curves are the *intermodulation distortion* curves for the nonlinear products.
- These curves are extended as straight lines with their corresponding slopes, the point where they intersect is the intercept point.
- This intercept point can be read from either the output or input power axes corresponding to OIP3 or IIP3.
- The small-signal gain of the device gives rise to the difference between the IIP3 and the OIP3.

LNA Simple IP3 Example: Schematic Setup

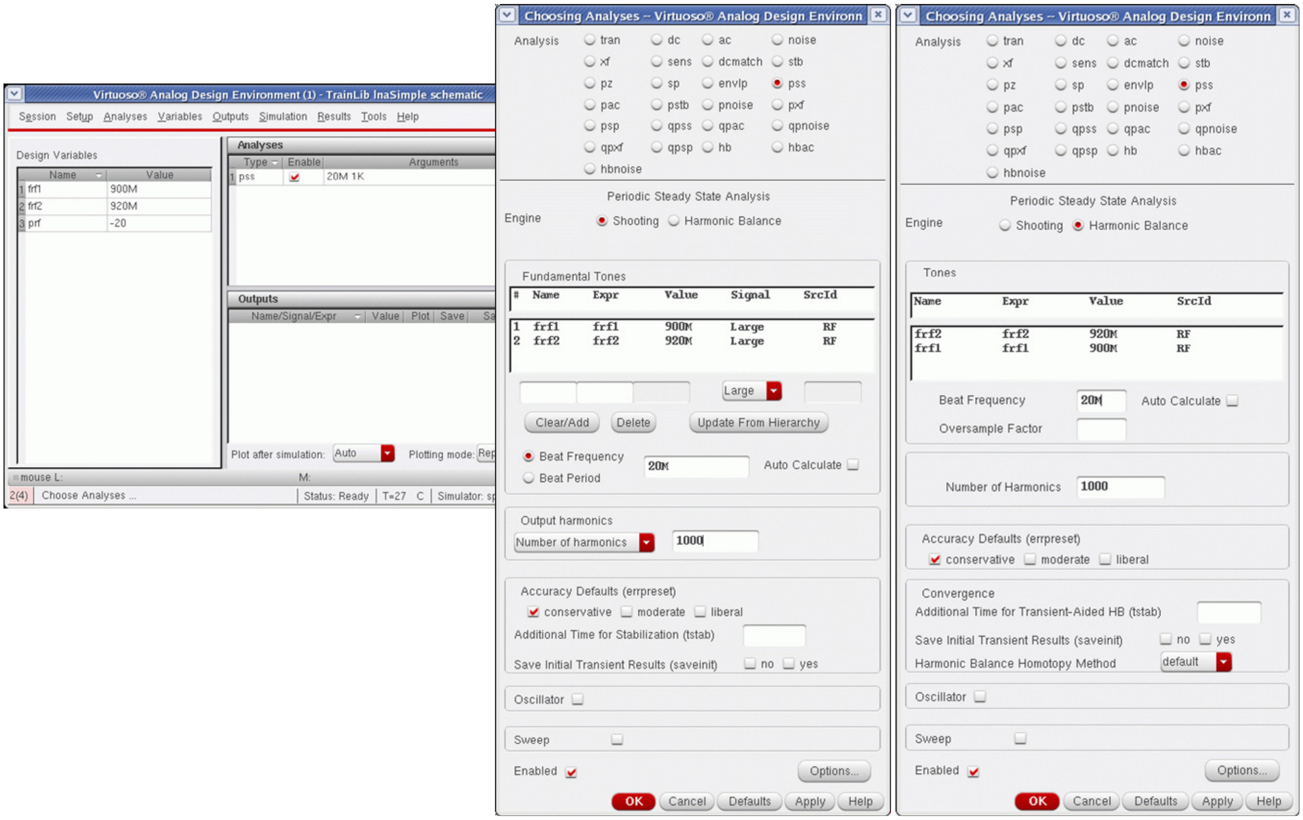


8/4/10 RF Analysis with Virtuoso Spectre Simulator 34

The following couple of pages illustrate an example to calculate the IP3 of a DUT.

- You will see this schematic in the labs. It is a BJT based Low Noise Amplifier.
- The corresponding *Edit Object Properties* window in the figure above shows the setup of the dual-tone sine wave input RF source with a 50 ohm resistance.

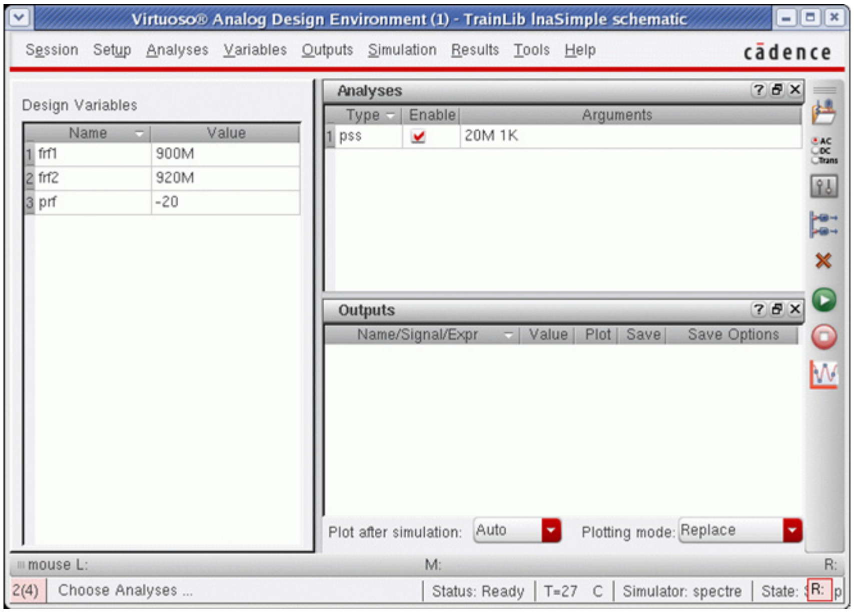
LNA Simple IP3 Example: ADE Setup

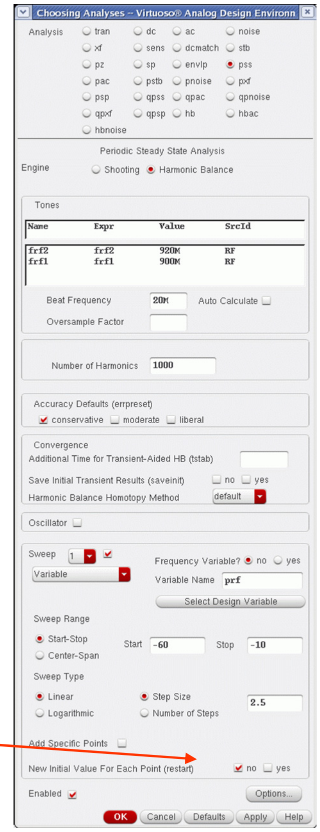


8/4/10 RF Analysis with Virtuoso Spectre Simulator 35

- The analysis setup in the Analog Design Environment (ADE) is shown in the figures above for both the shooting (left) and harmonic balance (right) algorithms using PSS analysis.
- The two-tone frequencies are set at 900 MHz and 920 MHz respectively.
- 1000 harmonics are used for computation and the *Beat Frequency* calculates out to 20 MHz.

LNA Simple IP3 Example: Swept Power Setup





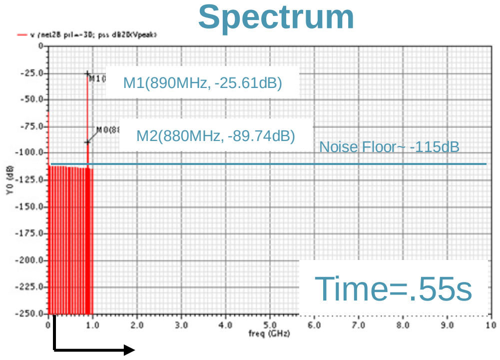
Set Restart to No for most sweeps

- Enable the *Sweep* option in the *Choosing analyses* form and fill in the required sweep frequency values for the frequency amplitude variable.
- When using *Harmonic Balance*, setting **restart** to **no** saves simulation time in recalculating a new starting point and the previous simulation point is used as the initial value for the next sweep.
- If you use *Shooting*, then *always* set the **restart** value to **yes**.

For Understanding Only

PSS Shooting

Spectrum



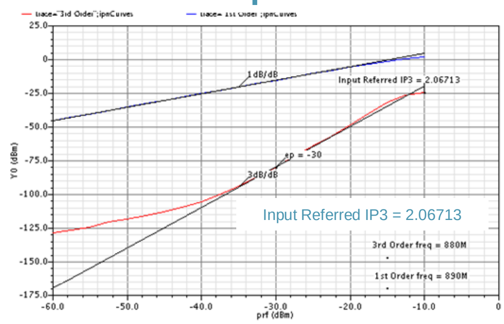
PSS starts from the left and counts # of harmonics to the right in intervals of the LOWEST COMMON FREQUENCY

PSS interval is 10M, so you need 100 harmonics to see the output tones.

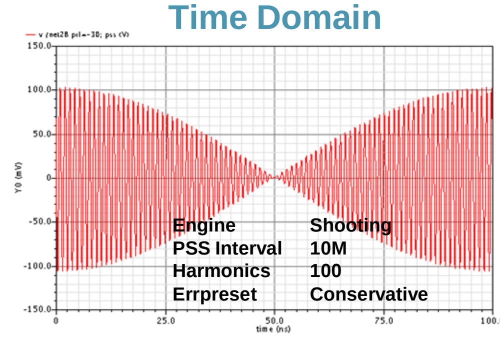
Noise floor can usually be improved to about -130 dBV– About 0.3 microvolts RMS.

This information is provided for your understanding only. Please do not try to do the actions described. Pages with this kind of information will have a “For Understanding Only” note in the upper right.

Swept IP3



Time Domain

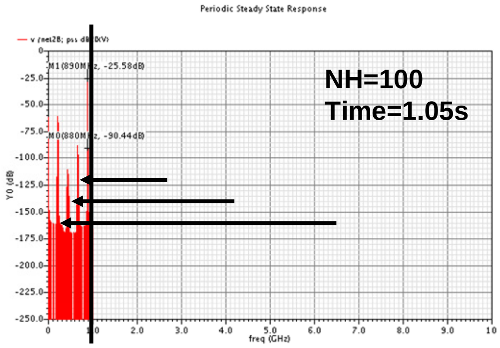


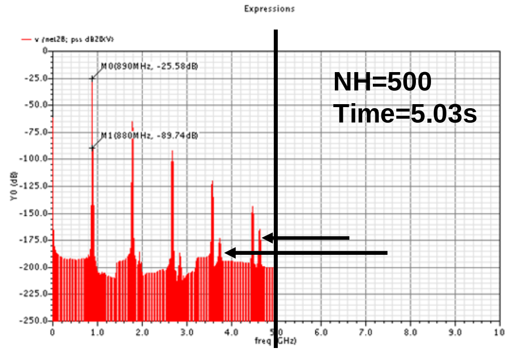
8/4/10 RF Analysis with Virtuoso Spectre Simulator 37

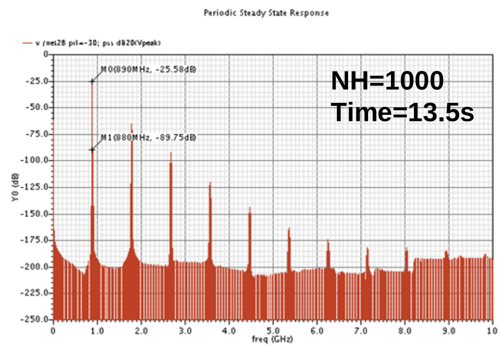
- These are the results from using PSS shooting for calculating the IP3.
- The two harmonics defined are at 890 MHz (1st harmonic) and 880 MHz (third harmonic). The 10 MHz PSS interval warrants 100 harmonics to see the output tones.
- The third-order input-referred intercept point is at 2.06713 dBm.

For Understanding Only

PSS Harmonic Balance







- Due to conservation of energy, you have to have enough harmonics to characterize the waveform. If you do not have enough harmonics, energy in the higher bands will fold back.
- The IP3 tones are still at the correct levels, but there are extra harmonics in the spectrum

8/4/10 RF Analysis with Virtuoso Spectre Simulator 38

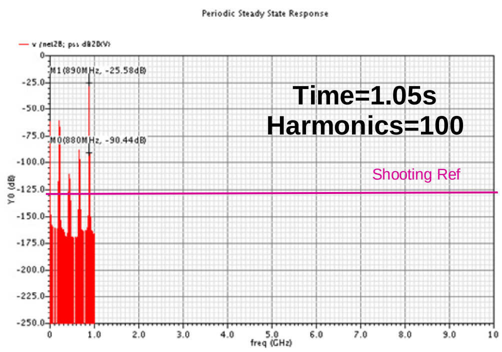
The following pages analyze the results obtained for IP3 using Harmonic Balance for the LNA.

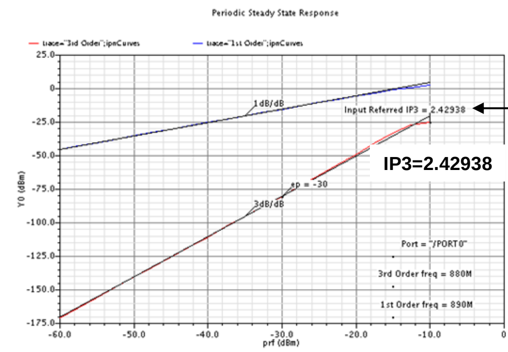
The energy of all the harmonics is in the solution for the number of harmonics specified. This results in power appearing at frequencies that are not expected in the solution. This is called **aliasing**.

- You need to take enough harmonics so that aliasing is not a problem for the measurement that you want.
- Start with a reasonable number of harmonics, simulate, and make your measurement. Then increase the number of harmonics by about 50%.
- If the measurement changed, you need more harmonics.
- If the measurement didn't change, you might be able to reduce the number of harmonics.

For Understanding Only

PSS Harmonic Balance (continued)

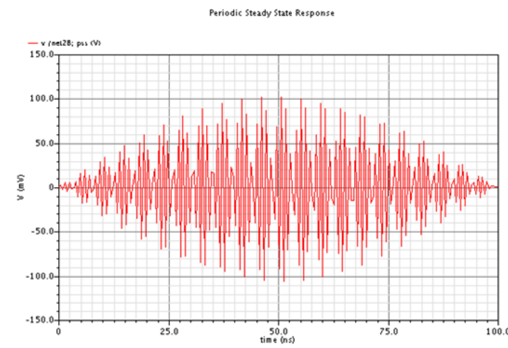




** IP3 value is not Accurate.

More harmonics are needed

- When you have multiple tones, you have to be sure to have sufficient number of harmonics to resolve the waveform.
- The time domain waveform is not accurately represented even though solution converged.

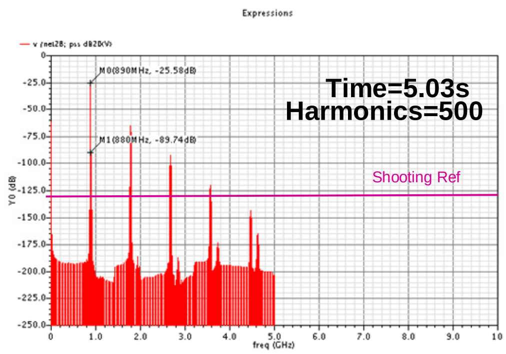


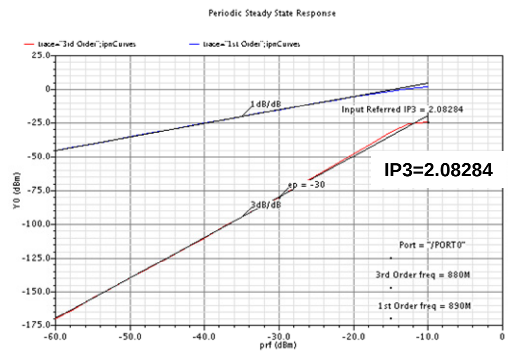
In the case of using just 100 harmonics as shown above, though time taken in less, the energy folds back significantly giving rise to a higher IP3. This is caused by aliasing.

The simulation converges but the IP3 value is not accurate. (Remember the value when using the Shooting algorithm.) This gives rise to the notion that more harmonics are required to get the correct IP3.

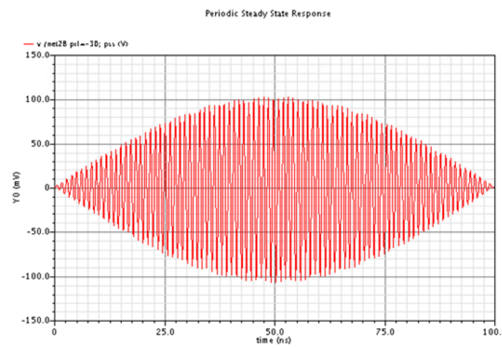
For Understanding Only

PSS Harmonic Balance (continued)





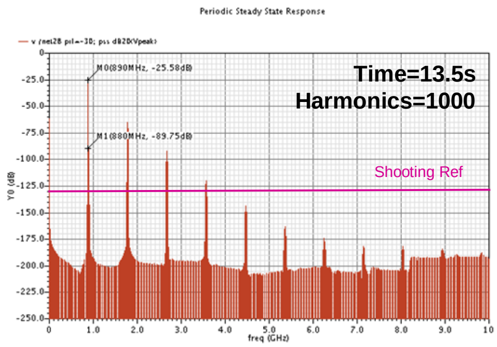
- There is still a little fold back, but the tones of interest are well resolved.

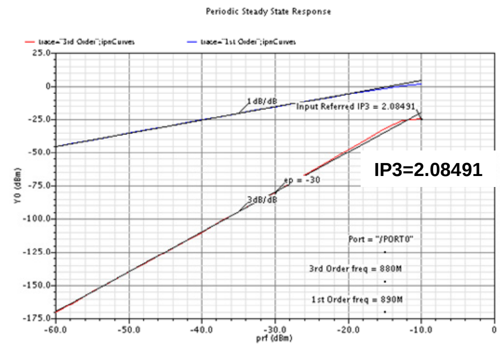


In the case of using 500 (a higher number) harmonics as shown here, the IP3 value is accurate, but the spectrum still shows a little energy foldback at the higher harmonics.

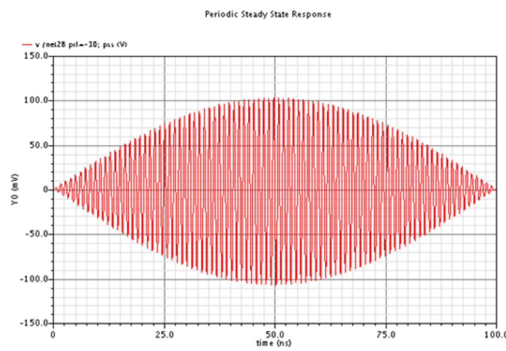
For Understanding Only

PSS Harmonic Balance (continued)





- Fully resolved with a good noise floor
- Simulation Time and Accuracy becomes comparable to Shooting Engine.



In the case of using 1000 harmonics (at the cost of significantly higher simulation time), as shown here, the waveforms are well resolved.

The IP3 value is almost identical as that at 500 harmonics. It might be possible to use less than 500 harmonics to get a good IP3.

Convergence Aids for Shooting

- Add a **tstab**. This allows the waveform to become more settled before the steady-state waveform is computed
- Try setting a tstab of about ¼ cycle longer or shorter. This puts the starting point of the waveform in a different place.
- Make sure that **reltol** isn't too small.
- Relax **steadyratio** to 0.01. This allows a bit more startup behavior to be present in the solution.
- Except for very high Q oscillators, tstabmethod and method should be set to gear2only.
- If the time step collapses to a small value, set **relref** to *sigglobal*.
- Set **maxperiods** greater than 20 to allow more iterations.

Oscillators

- Set tstab long enough for the oscillations to reach steady-state.
- Set **steadyratio** = 0.01.
- Try setting the *linear oscic* if there is no divider in the circuit.

oscic stands for oscillator initial condition.

Convergence Aids for Harmonic Balance

- For systems with sharp transitions, set *oversamplefacto*r to 2 or larger.
- Raise the number of harmonics.
- Specify a *tstab* to allow a better frequency domain starting point.
- For systems that are very near linear, set iters to 1e-2.
- Set *maxperiods* greater than 100 to allow more iterations.

Oscillators

- Set tstab long enough for the oscillations to reach steady-state.
- Try setting the *linear oscic* if there is no divider in the circuit.
- Try using the *two tier* method stated below.
  - In this method, choose a single pinnode inside the oscillator feedback system for single ended, and both nodes for differential.
  - Specify 1 for the harmonic index.
  - Specify an estimate for the peak-to-peak amplitude. If in doubt, specify a smaller value than you anticipate.

Lab Exercises

Lab 2-1 Running an LNA PSS Simulation

Lab 2-2 Running an LNA IP3 Simulation

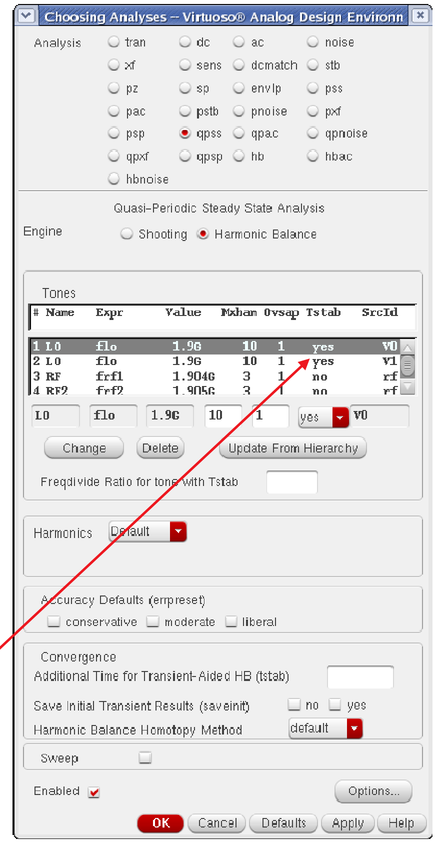
Quasi-Periodic Steady-State Analysis (QPSS)

Module 3

August 4, 2010

What Is Quasi-Periodic Steady State (QPSS)?

- Quasi-periodic steady-state (QPSS) analysis computes circuit response with multiple fundamental frequencies using harmonic balance (in frequency domain) or shooting algorithms.
- QPSS can compute circuit responses having closely spaced or nonharmonically related fundamentals, which cannot be resolved by PSS efficiently.
- QPSS analysis sets the circuit quasi-periodic operating point, which can be used during a quasi-periodic time-varying small-signal analysis, such as QPAC, QPXF, QPSP and QPNOISE similar to the varied small-signal PSS analysis.
- QPSS requires at least two inputs to run successfully.
  Important: Set tstab to **yes** for the largest tone.



- The QPSS analysis calculates only the mixing products that are produced by the input frequencies to the system. This usually results in fewer harmonics that need to be calculated compared to PSS where all the harmonics are calculated, thus QPSS runs faster than PSS for multiple input circuits.
- QPSS harmonic balance (or the hb option on the Choosing Analysis form) always runs much faster than shooting, even if pulse inputs are applied to the circuit.

QPSS Analysis

- QPSS computes the large signal, periodic steady-state response of the circuit.
- QPSS models distortion effects from multiple large signal inputs.
  - Internal to the simulator, one input is treated as the **large** signal, which causes the most nonlinearity or the largest response in the circuit. For shooting, set the signal to large. For HB, enable tstab for this signal.
  - Other signals are treated as **moderate** and do not need to be harmonically related to the large signal or integer multiples of each other. Moderate is just terminology. QPSS calculates the full large-signal response with all the signals applied.
  - The **moderate** signals can be large enough to create distortion (near the 1 dB Gain Compression Point in amplitude)
- Applications of QPSS:
  - Measuring the intermodulation distortion in transmitter mixers.
  - Measuring the distortion in amplifiers.
  - Multiple input frequency circuits.

- Use QPSS to measure any multiple input frequency system.
- The QPSS runtime doesn't depend on how closely spaced the input frequencies are. It depends on the number of harmonics that are created by the system.
- Choose QPSS to compute the steady-state responses of a circuit driven by two or more signals at unrelated frequencies.
- QPSS can measure distortion and frequency translation effects created by multiple moderate-signal inputs, including all third order products. Note that for IP3, rapid IP3 using the AC form, or the PSS/PAC forms is the fastest way to get IP3.

General Setup Notes

- Each input signal can be a composition of more than one source. However, these sources must have the same fundamental name. If you have multiple sources at the same frequency, give them the same *Frequency Name 1* designation.
- Harmonic Balance is much faster than shooting and requires less memory.
- Shooting should be used for switched-capacitor filters and sampling circuits or other very nonlinear circuits.
- Sweeping is done by enabling the *Sweep* button in the *Choose Analyses* form for QPSS setup.
- The ability to sweep QPSS provides a way to perform intermodulation distortion calculations with multiple input signals, considered as moderately large signals.
- QPSS also has the ability to plot IPn (IP2, IP3, IP5,etc.). Like PSS, QPSS calculates IPn and plots the outputs using Direct Plot capabilities.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 48

For each fundamental name, the fundamental frequency is the greatest common factor of all frequencies associated with the name. If you have for example 1 GHz and 2 GHz sources, give them the same name, because they are harmonically related. This naming greatly reduces the overall simulation time.

In QPSS, select one signal to be the **large** Fundamental frequency. This is accomplished in shooting by setting the signal to *large*. This is accomplished in HB by enabling *tstab*. This signal can be a sinusoid, square wave, triangle wave, or pulse. In other words, it can be essentially any periodic waveform. The other applied signals need to be sinusoids, with the exception that a second tone can be a pulse tone.

Select your Large signal to be the signal that

- Causes the largest response in the circuit
- Is the least sinusoidal signal in the circuit
- Causes the most nonlinearity in the circuit

Designate a non-zero harmonic range (Harms) value for each tone.

Two Types of QPSS

Shooting QPSS

Shooting QPSS solves in the time domain and consists of four phases.

1) Set harmonics for individual tones
2) TSTAB is run
3) A number of (at least 2) stabilizing iterations with all signals activated is run.
4) Last, Newton method is followed.

QPSS employs the Mixed Frequency Time (MFT) algorithm extended to multiple fundamental frequencies. (slices)

Shooting QPSS should be used for sampling circuits and switched capacitor circuits only.

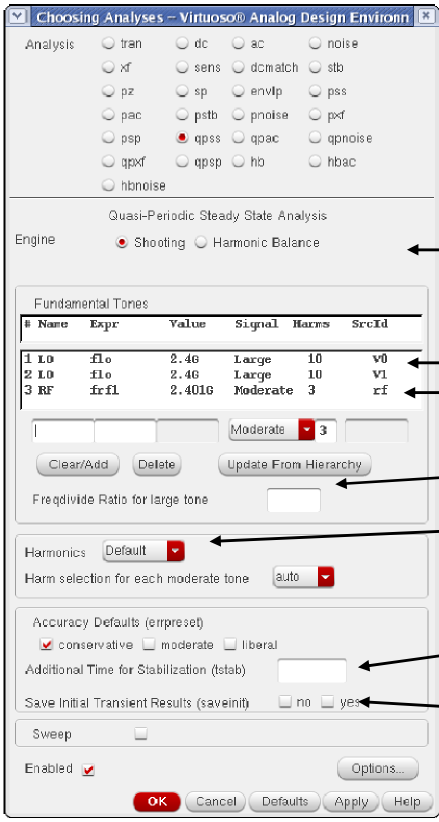
Harmonic Balance QPSS

Harmonic Balance QPSS solves in the Frequency Domain.

1) Set harmonics for individual tones. Only the actual mixing products are calculated.
2) TSTAB (Can set for 1 of the signals applied, usually LO or CLK)
  - MMSIM 7.0 and before, Default is transient assisted, but you can get direct HB by setting TSTAB=0
  - After MMSIM 7.0.1, Default is Transient Assist = Off
3) Begins iterating the frequency domain

- **TSTAB** is the extra stabilization time after the onset of periodicity for independent sources. If HB is selected and tstab is not specified, the DC solution becomes the starting point for the frequency domain iterations.
- The *tstab* parameter controls both the length of the initial transient. For HB, at the end of the tstab interval, a single period of the input tone is run, and an FFT is calculated. This FFT solution is the starting point for the frequency domain iterations.
- The stabilizing iterations run before the Newton iterations begin for shooting only. It is a very simple-minded optimizer.
- The MFT algorithm for shooting is a method where only a few pieces of the waveform is simulated. The more harmonics are set on each tone, the more pieces of waveform are simulated. Thus more time and memory is needed.

QPSS Using Shooting



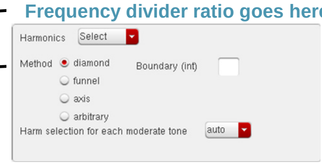
Can run QPSS with either the Shooting or Harmonic Balance engines

Assign one tone to be a Large Tone. This is the main LO or CLK tone
Usually this tone needs more harmonics

Assign the other tones to be Moderate.

Highlighting a tone above allows you to edit it.

Frequency divider ratio goes here.



Can trim the higher order harmonics for a more efficient harmonic set to be simulated (more to come)

Shooting always runs at least one cycle of the large tone. More time can be added by specifying a time in the tstab field.

Puts initial waveform in Direct Plot

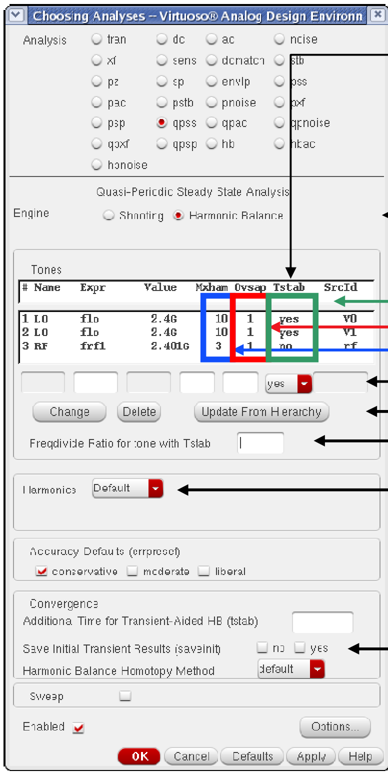
The QPSS Analysis setup is shown here with a analysis that uses the Shooting engine. Note that the solution is a full large-signal solution with all the signals applied. Large and moderate are just terminology.

Definitions

- **Large signal**: Dominates the nonlinearity of the periodic operating point.
- **Moderate signal**: All the rest of the signals

More on the Harmonic Trimming options is explained later.

QPSS Using Harmonic Balance



Important: Set tstab to yes for the signal that causes the largest amount of distortion in the system.

You can run QPSS with either the Shooting or Harmonic Balance engines.

Can apply Tran Assist to one Tone only.

Can set oversample on each tone.

Set the number of harms on each tone. Higher for more nonlinear tone.

Edit the tones here.

Click **Change** when re-entering data.

Enter frequency divide ratio here.

Harmonic Trimming is set here. (more to come)

Transient-Aided is off by default for HB. Uses DC solution.

Puts initial waveform in Direct Plot.

**** Usually oversample is applied only to a strongly nonlinear signal in need of more samples in the time-domain. This results in a more accurate Fourier Transformation.**

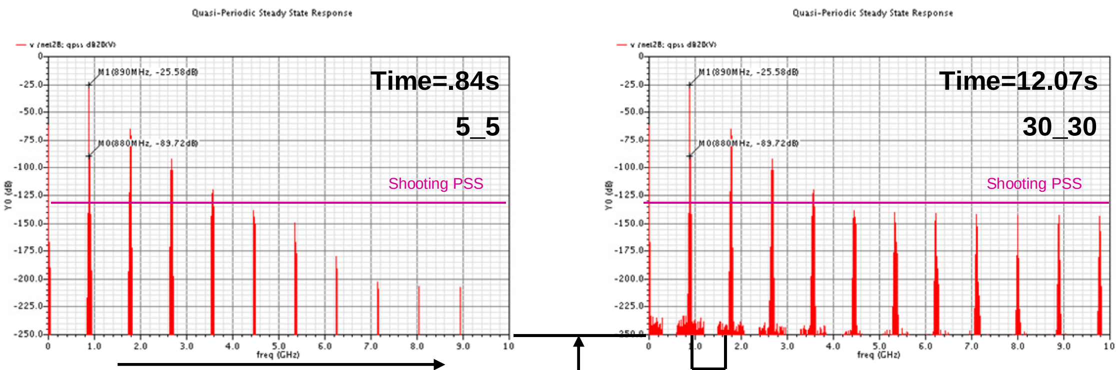
8/4/10 RF Analysis with Virtuoso Spectre Simulator 51

The QPSS Analysis setup is shown above. The analysis uses the harmonic balance engine.

- Always identify the signal that causes the most amount of nonlinearity by setting tstab to yes for this signal.
- Set *oversample* on the tones with very sharp edges. Note that if there are two signals, this will dramatically extend the run time. This causes more timepoints to be run in the nonlinear devices, thus improving the accuracy of the DFT.
- *Mxham* is the maximum number of harmonics set on each tone. It accepts a list of numbers of harmonics needed for each fundamental. If a larger number is used for a highly nonlinear tone, the results are more accurate.
  However, if you do not specify *Mxham*, a warning message is issued, and the number of harmonics defaults to 1 for each fundamental.
- Set *tstab* just long enough for the signal with tstab enabled to reach steady-state. This can improve convergence, or to speed up the overall simulation time. Many circuits run faster if a small tstab is specified.

For Understanding Only

LNA Example with Shooting QPSS



Its important to include the energy of the primary driving tone far enough out in frequency or the energy will fold back into the desired spectrum

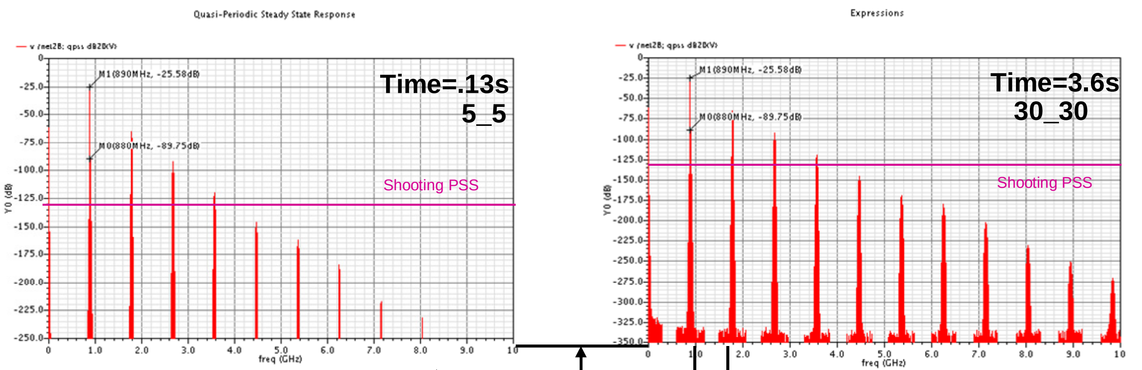
Noise Floor is very good with QPSS.

Really not necessary to simulate this part of the spectrum. The energy is very low.

- QPSS builds harmonic content out from around a strong, nonlinear tone (LARGE Tone).
  - The rest of the tones are called MODERATE tones.
- Too many harmonics have diminishing returns.
- Most multitone circuits are more efficient with harmonic balance.

- Above are the results of the QPSS analysis using Shooting carried out on an LNA. The noise floor you see is very good. It is not necessary to use a large number of harmonics to see the lower part of the spectrum, because it does not convey significant information.
- How do you set the number of harmonics?
  1. First choose a number of harmonics and run the simulation.
  2. Make the measurement.
  3. Increase the number of harmonics by about 50% and re-run.
  4. If the measurement didn't change, the first guess is enough, and might be able to be reduced.
  5. Set the minimum number of harmonics to ensure an accurate measurement.

LNA with HB QPSS



It is important to include the energy of the primary driving tone far enough out in frequency or the energy will fold back into the desired spectrum.

Noise Floor is very good with Harmonic Balance.

It is really not necessary to simulate this part of the spectrum. The energy is very low.

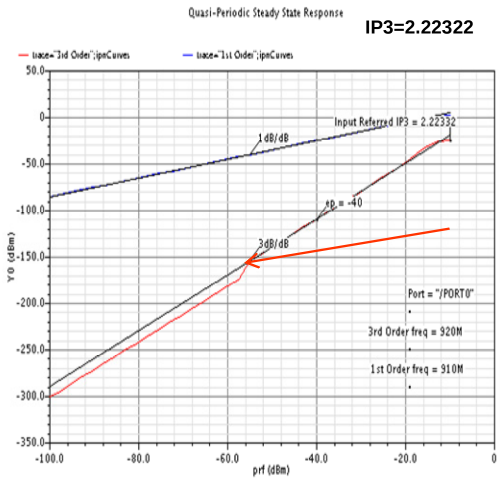
- QPSS and Harmonic Balance are very well suited to each other and constitute the most efficient way to simulate large, multitone problems.
- Most of the energy of the spectrum is accounted for with as little as 5 harmonics of each tone.
- This is the mode in which most commercial harmonic balance simulators operate.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 53

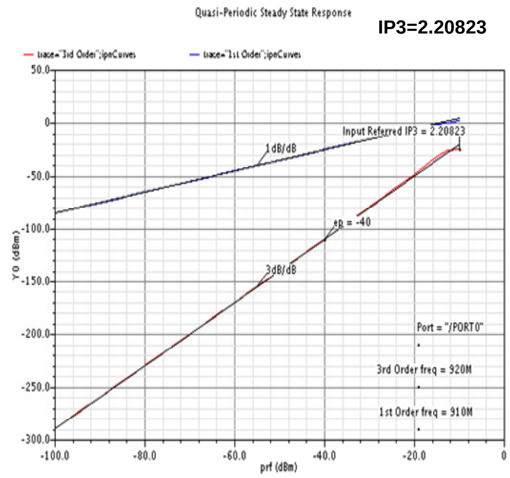
- Above are the results of the QPSS analysis using HB carried out on an LNA. The noise floor you see is again very good. It is not necessary to use a large number of harmonics to see the lower part of the spectrum, because it does not convey significant information.
- Although the noise floor is typically lower for HB than for Shooting, they are both good enough.
- How do you set the number of harmonics?

1. First choose a number of harmonics and run the simulation.
2. Make the measurement.
3. Increase the number of harmonics by about 50% and re-run.
4. If the measurement didn't change, the first guess is enough, and might be able to be reduced.
5. Set the minimum number of harmonics to ensure an accurate measurement.

How Does errpreset Affect Harmonic Balance?



Errpreset = Conservative
Not enough accuracy for signals below -150db as indicated by the arrow in the figure above.

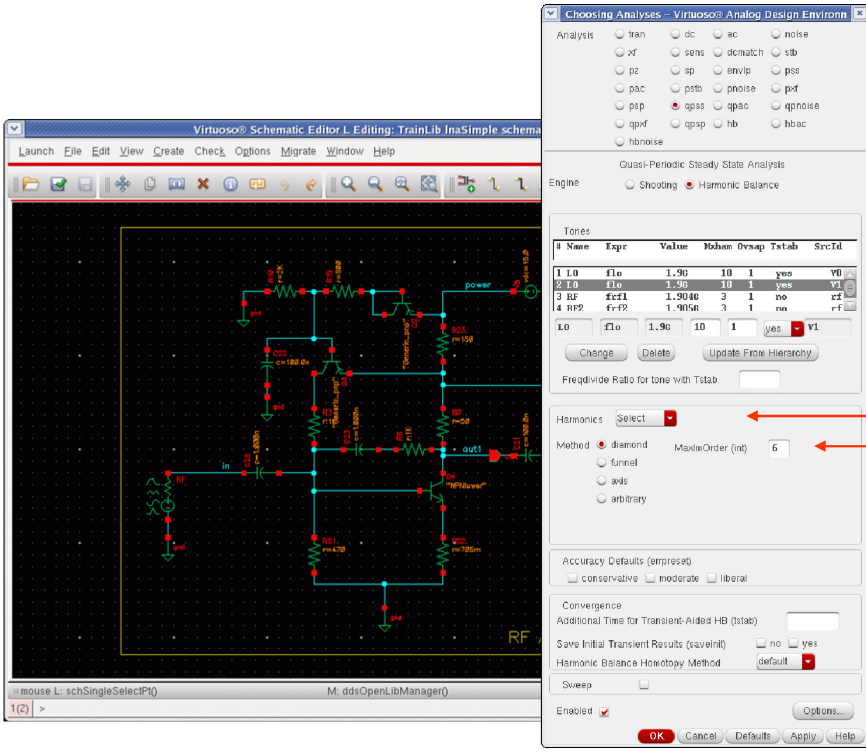


Errpreset = Moderate
Reltol = 1e-5
Vabstol = 3e-8

- For very small signals, such as IP3 or Phase Noise, you can select **conservative** or tighten some of the tolerances manually.
- This affects when the Harmonic Balance algorithm considers that the harmonics are balanced enough.

- The *errpreset* parameter quickly adjusts several simulation parameters. In most cases, *errpreset* is the only parameter you need to adjust.
- The graphs show a comparison of the IP3 calculation of the LNA using QPSS. Note that when the *errpreset* is set to *conservative*, the accuracy is not enough for signals below -150dB.
- When the tolerances are tweaked by changing the values of *reltol* and *vabstol* manually, it leads to better accuracy in the calculation even with a *moderate errpreset* setting.

Trimming Harmonics



Can trim the # of harmonics

8/4/10 RF Analysis with Virtuoso Spectre Simulator 55

- The Harmonics area enables you to select the important moderate tone harmonics for QPSS multitone simulation.
- Choices for the *Harmonics* are: **Default, View All** and **Select**.
- In *Default* – You can select harmonics by choosing *auto*, *even*, *odd*, or *all* in the *Harm selection for each moderate tone field*. The Harm selection for each moderate tone field only appears when Engine is set to Shooting. The default selection gives a rectangular cut for the harmonics to be calculated. More on this later.
- In *View All* – It lets you first enter a frequency range and then the form shows all the harmonic frequencies that will be calculated.
- In *Select* – You have a **Method** option that specifies how the harmonics are selected: **diamond, funnel, axis** or **arbitrary**. The harmonics selected by each method depend on the *maxharms* and **MaxImOrder** (Harmonic Balance) or **Boundary** values (Shooting) where **Boundary (int)** specifies the maximum order of the harmonics to be calculated..
- If the Harmonics value is something other than *Select*, the *Method* field does not appear in the form and the box selection method is the *rectangular cut.*

Understanding MaxImOrder

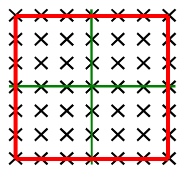
- MaxImOrder is the maximum intermodulation order used with harmonic balance (same parameter as *boundary* used in Shooting).
- MaxImOrder reduces the number of harmonics simulated in an intelligent way by trimming the high order terms that have very little energy in them, while still keeping accuracy in the primary or “on-axis” tones.
- You can target important harmonics with most of the energy by using MaxImOrder. This targeting reduces simulation time and memory used.
- However, anytime you reduce the number of harmonics simulated, there is some inaccuracy in the highest order terms specified due to conservation of energy. (This phenomenon is also called aliasing.)

8/4/10 RF Analysis with Virtuoso Spectre Simulator 56

If *maxImOrder* (boundary in shooting) is set to six, then the sixth order mixing products and below are calculated.

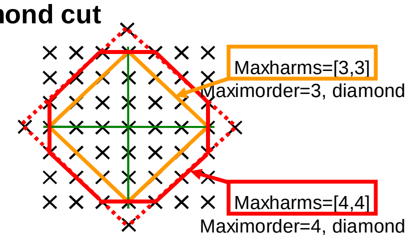
Trimming Harmonics with MaxImOrder/Selectharm

default



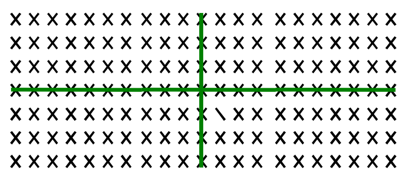
box: the default. A QPSS simulation with M tones will solve for all frequencies ω =*k*1ω1 + *k*2ω2 … where *ki* ≤ *Hi* for each tone and *Hi* is the maximum harmonic for tone i.

Diamond cut



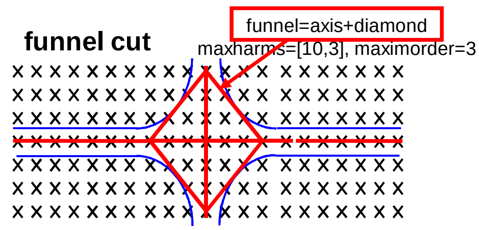
diamond: selects a diamond cut with ∑ *ki* ≤ *P*, P is the maximum order set by the parameter **MaxImOrder**

Axis cut



axis: selects harmonics along an axis, that is, there are no intermodulation harmonics

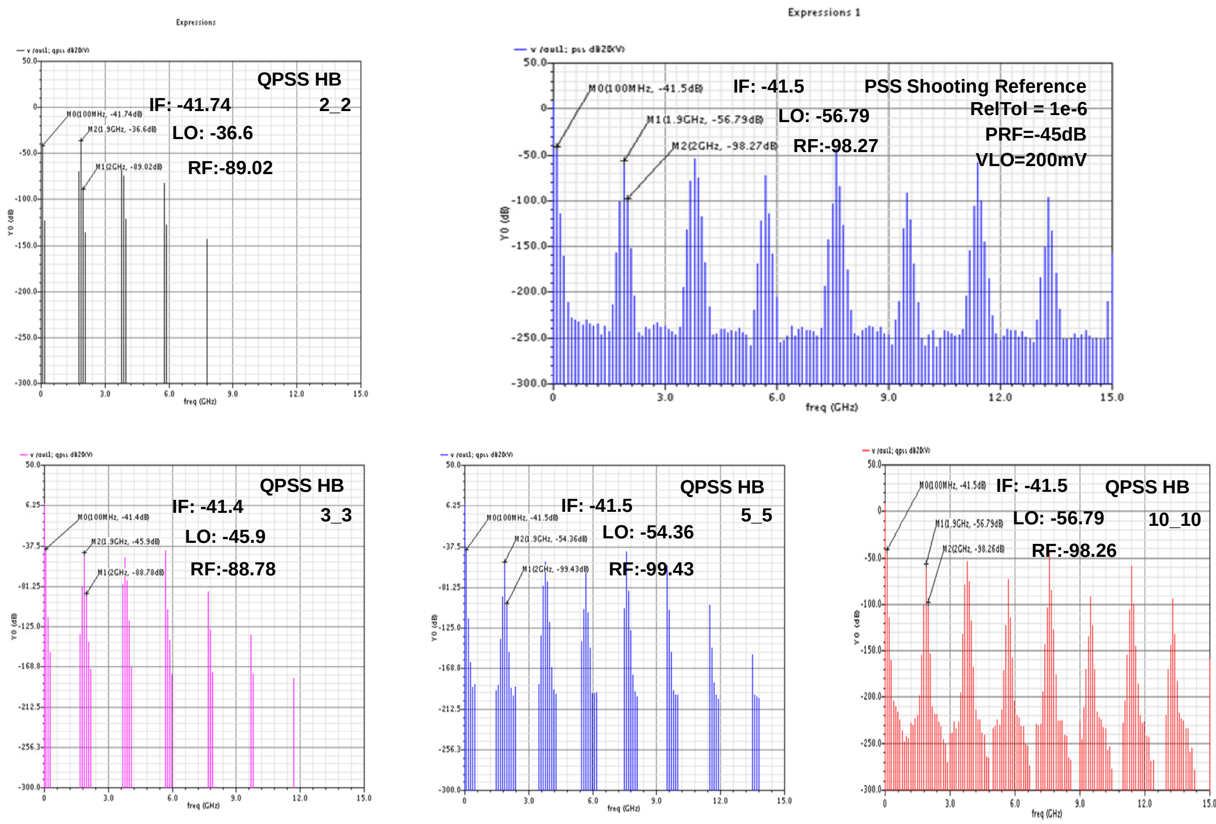
funnel cut



funnel: a combination of the axis cut and diamond cut. The funnel cut selects harmonics along an axis and intermodulation harmonics of order P or less.

- The X Axis is the first frequency applied to the system.
- The Y axis is the second frequency applied to the system.
- The Xs indicate harmonics that are calculated. The first X to the right of the origin is 1*first frequency plus 0*second frequency. The second X to the right of the origin is 2*first frequency plus 0* second frequency. The first X at 45 degrees to the origin is 18 first frequency plus 1* second frequency. The second X at -45 degrees is 2*first frequency plus -1* second frequency.
- The **Diamond** cut is designed to keep the important on-axis harmonics but to reduce the number of off-axis (intermodulation) harmonics. It is the same as **Default** but with higher order harmonic trimming applied.
- The evaluated harmonics consist of the intersection of the universe of harmonics (specified by the maxharms values) and of the superimposed diamond (whose order or size is specified by either the MaxImOrder or Boundary values).
- For example, in this illustration, the evaluated harmonics lie on or within either the solid red line when *maxImOrder* is set to 4 or the solid orange line when *maxImOrder* is set to three.
- In the above illustration for the **Axis** method, the evaluated harmonics include only those indicated by the green lines. This calculates only the harmonics of both tones and none of the mixing products are calculated.
- The **Funnel** method is a combination of the diamond method and axis methods. It selects harmonics along the axes and intermodulation harmonics of a specified order or less. It is unique to Spectre® RF and includes energy of the on-axis, higher order harmonics.
- The **Arbitrary** method lets you specify indices of interest for each of the moderate fundamental tones. The ADE provides a guidance on the format to use in a message when you select this method. *This method is not usually recommended.*

Mixer Spectrums (Using Enough Harmonics)

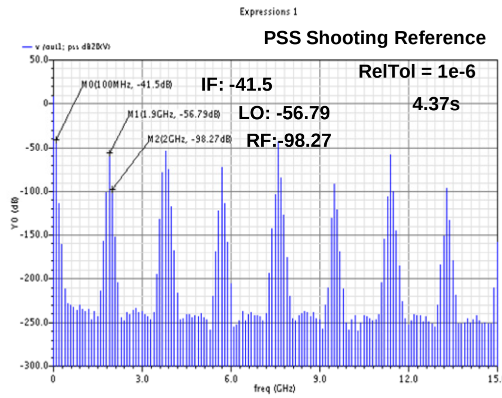


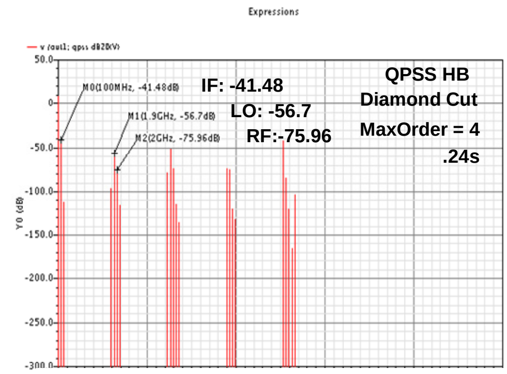
8/4/10 RF Analysis with Virtuoso Spectre Simulator 58

The different graphical output waveforms shown here try to stress the point that you need to take enough harmonics to get the correct answer.

- With 2 and 2, all of the tone amplitudes are high.
- 3 and 3 causes the levels to come down, so that isn’t enough harmonics.
- 5 and 5 still isn’t enough for this system.
- 10 and 10 are required for full accuracy.

Mixer Example: Trimming Harmonics with MaxImOrder



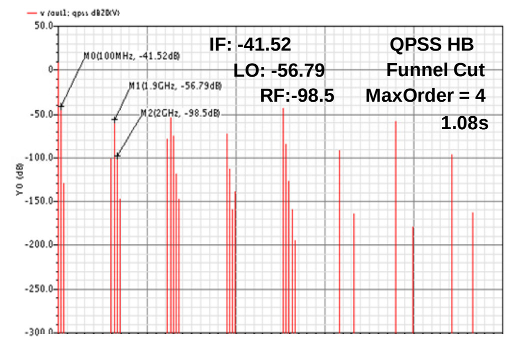


Diamond Cut is the default for many HB simulators

Spur Order (Partial List)

1,0 1st order
1,1 2nd order
2,1 3rd order
3,1 4th order
2,2
4,0 MaxOrder = 4

3,2 5th order
4,1
2,4 6th order
5,2 7th order
etc



Funnel Cut keeps the harmonics on the major axis.
Most accurate and fast.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 59

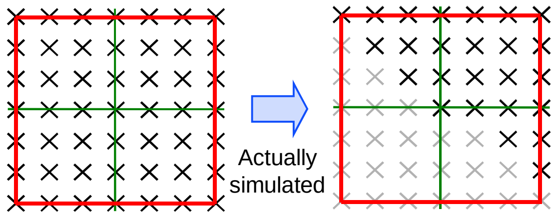
These waveforms are outputs of the PSS (with shooting) and QPSS HB analyses carried out on a mixer.

- The waveform on the left is for the output spectrum using the shooting algorithm with a tighter tolerance.
- The waveforms on the right are obtained by using Harmonic Balance with a MaxImOrder of 4 on a Diamond Cut and a Funnel Cut to trim the harmonics.
- On comparison with the shooting results, the funnel cut gives a more accurate result on a trade-off with simulation time.
- The Funnel cut is more accurate than diamond cut because higher order harmonics of the LO and RF are calculated.

Note: There might be bugs in the funnel cut. Always check the spectrum of funnel results with the default cut.

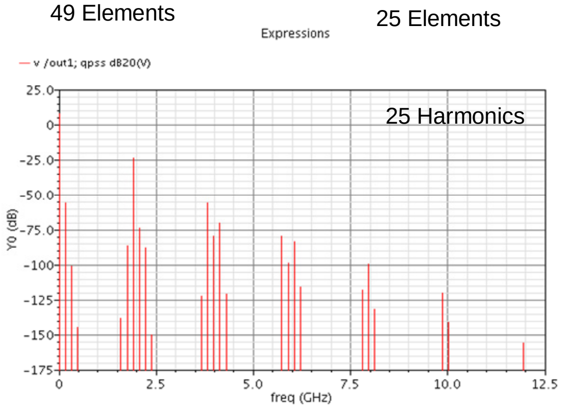
Harmonic Balance: Default

RF x 3 harmonics
LO x 3 harmonics



49 Elements

25 Elements



Default
RF=2.07G x 3
LO=1.91G x 3

Mixer Spectrum

Order = /LO/ + /RF/

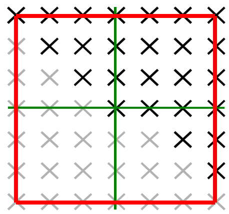
	Frequency	LO	RF
1.	0	0	0
2.	160M	-1	1
3.	320M	-2	2
4.	480M	-3	3
5.	1.59G	3	-2
6.	1.75G	2	-1
7.	1.91G	1	0
8.	2.07G	0	1
9.	2.23G	-1	2
10.	2.39G	-2	3
11.	3.66G	3	-1
12.	3.82G	2	0
13.	3.98G	1	1
14.	4.14G	0	2
15.	4.3G	-1	3
16.	5.73G	3	0
17.	5.89G	2	1
18.	6.05G	1	2
19.	6.21G	0	3
20.	7.8G	3	1
21.	7.96G	2	2
22.	8.12G	1	3
23.	9.87G	3	2
24.	10.03G	2	3
25.	11.94G	3	3

The default is a rectangular cut.

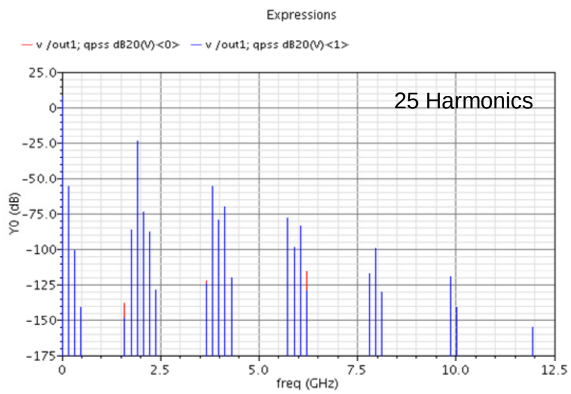
The corner members are 3*first frequency mixing with 3*second frequency. This is a 6th order term, and probably doesn't have much power in it. The diamond and funnel cuts can trim the number of harmonics to be calculated, thus speeding up the simulation.

Harmonic Balance: Diamond – MaxImOrder=6

MaxImOrder=6
or greater gives
same spectrum as
Default



25 Elements



Default
RF=2.07G x 3
LO=1.91G x 3

Diamond
RF=2.07G x 3
LO=1.91G x 3
MaxImOrder=6

Mixer Spectrum

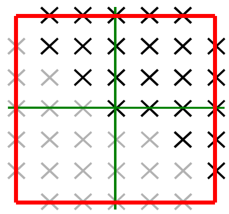
Order =
/LO/ + /RF/

Frequency		LO	RF
1.	0	0	0
2.	160M	-1	1
3.	320M	-2	2
4.	480M	-3	3
5.	1.59G	3	-2
6.	1.75G	2	-1
7.	1.91G	1	0
8.	2.07G	0	1
9.	2.23G	-1	2
10.	2.39G	-2	3
11.	3.66G	3	-1
12.	3.82G	2	0
13.	3.98G	1	1
14.	4.14G	0	2
15.	4.3G	-1	3
16.	5.73G	3	0
17.	5.89G	2	1
18.	6.05G	1	2
19.	6.21G	0	3
20.	7.8G	3	1
21.	7.96G	2	2
22.	8.12G	1	3
23.	9.87G	3	2
24.	10.03G	2	3
25.	11.94G	3	3

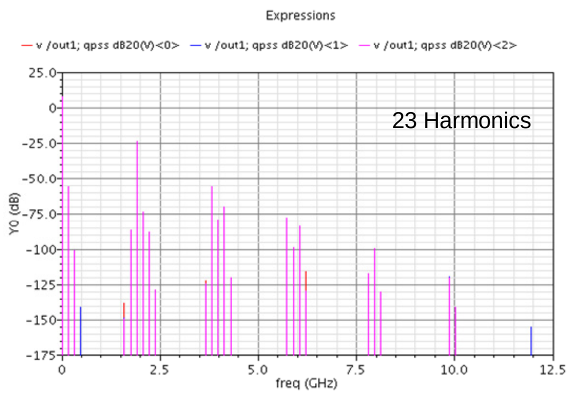
Note that this is the same as the rectangular cut.

Harmonic Balance: Diamond – MaxOrder=5

MaxImOrder=5



23 Elements



Default
RF=2.07G x 3
LO=1.91G x 3

Diamond
MaxImOrder=6

Diamond
MaxImOrder=5

Mixer Spectrum

Order = /LO/ + /RF/

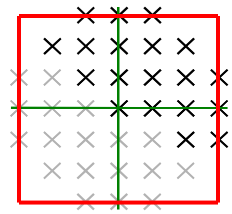
Frequency		LO	RF
1.	0	0	0
2.	160M	-1	1
3.	320M	-2	2
4.			
5.	1.59G	3	-2
6.	1.75G	2	-1
7.	1.91G	1	0
8.	2.07G	0	1
9.	2.23G	-1	2
10.	2.39G	-2	3
11.	3.66G	3	-1
12.	3.82G	2	0
13.	3.98G	1	1
14.	4.14G	0	2
15.	4.3G	-1	3
16.	5.73G	3	0
17.	5.89G	2	1
18.	6.05G	1	2
19.	6.21G	0	3
20.	7.8G	3	1
21.	7.96G	2	2
22.	8.12G	1	3
23.	9.87G	3	2
24.	10.03G	2	3
25.			

8/4/10 RF Analysis with Virtuoso Spectre Simulator 62

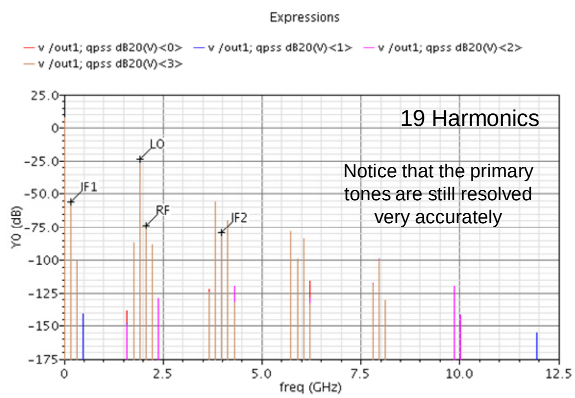
Only the corners have been cut out.

Harmonic Balance: Diamond – MaxOrder=4

MaxImOrder=4



19 Elements



Default
RF=2.07G x 3
LO=1.91G x 3

Diamond
MaxImOrder=6

Diamond
MaxImOrder=5

Diamond
MaxImOrder=4

Mixer Spectrum

Order = /LO/ + /RF/

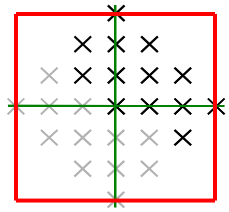
Frequency		LO	RF
1.	0	0	0
2.	160M	-1	1
3.	320M	-2	2
4.			
5.			
6.	1.75G	2	-1
7.	1.91G	1	0
8.	2.07G	0	1
9.	2.23G	-1	2
10.			
11.	3.66G	3	-1
12.	3.82G	2	0
13.	3.98G	1	1
14.	4.14G	0	2
15.	4.3G	-1	3
16.	5.73G	3	0
17.	5.89G	2	1
18.	6.05G	1	2
19.	6.21G	0	3
20.	7.8G	3	1
21.	7.96G	2	2
22.	8.12G	1	3
23.			
24.			
25.			

This cuts out harmonics more aggressively.

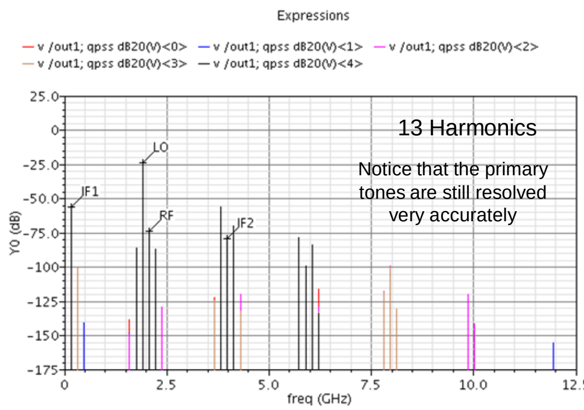
While this might be good for an IP3 simulation, it is likely **not** good for a mixer with an LO and an RF tone. The LO amplitude is usually large and causes quite a few harmonics. If these harmonics are not calculated, there is likely to be a lot of aliasing in the simulation result.

Harmonic Balance: Diamond – MaxOrder=3

MaxImOrder=3
Note: Diamond Cut only makes a Diamond if the MaxImOrder equals the order on each tone.



13 Elements



Default
RF=2.07G x 3
LO=1.91G x 3
Diamond
MaxImOrder=6
Diamond
MaxImOrder=5
Diamond
MaxImOrder=4
Diamond
MaxImOrder=3

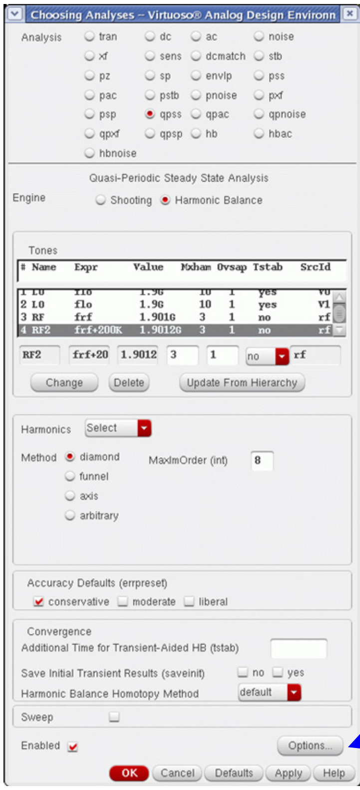
25 Harmonics

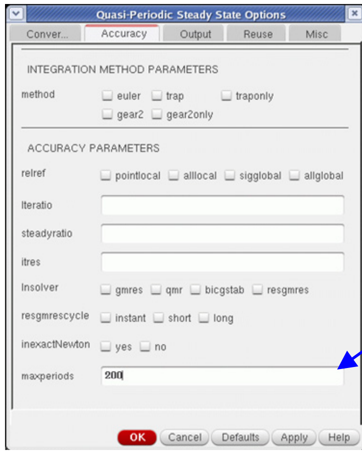
Order = /LO/ + /RF/

	Frequency	LO	RF
1.	0	0	0
2.	160M	-1	1
3.			
4.			
5.			
6.	1.75G	2	-1
7.	1.91G	1	0
8.	2.07G	0	1
9.	2.23G	-1	2
10.			
11.			
12.	3.82G	2	0
13.	3.98G	1	1
14.	4.14G	0	2
15.			
16.	5.73G	3	0
17.	5.89G	2	1
18.	6.05G	1	2
19.	6.21G	0	3
20.			
21.			
22.			
23.			
24.			
25.			

This is a very aggressive cut. Do **not** use it for most simulations. However, it reduces the run time a lot.

Maximum Iterations Using Maxperiods





Maxperiods

Use this field to increase the number of maximum iterations that shooting or HB uses before it quits. The default is 20 for shooting and 100 for HB. If necessary, you can increase it to more than the default for more nonlinear circuits.

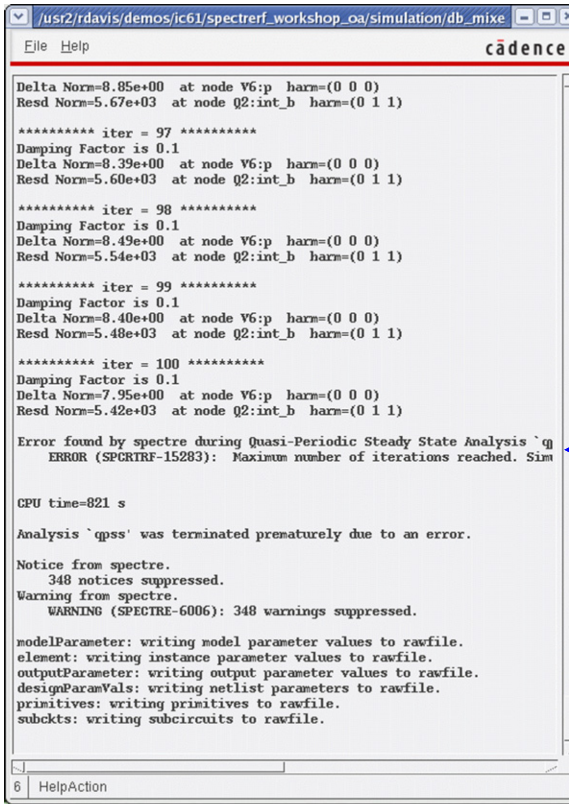
Access the field from ADE by choosing **Choose Analyses**, select **qpss** in the Analysis field, and click **Options**.
A form appears. Click the **Accuracy** tab to access the **maxperiods** field.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 65

Use the *maxperiods* field to specify the maximum number of iterations in the QPSS or HB analysis.

- The default for shooting is 20.
- The default for HB is 100.

Maximum Iterations: Simulator Exit



When running a simulation in harmonic balance, the Spectre RF simulator tries for 100 iterations but cannot find a solution so it closes.

Although it provides an error, the results are available to be viewed from the direct plot form.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 66

Because the result is available for plotting even if the solution doesn't converge, be careful when viewing a result.

Harmonic Balance Steady State Analysis

- Harmonic Balance is used to compute the response of circuits that have either one fundamental frequency (PSS) or that have multiple fundamental frequencies (QPSS) and also their corresponding operating points, which can then be used during a periodic or quasi-periodic time-varying small-signal analysis, such as HBAC or HBnoise.
- Usually, harmonic balance (HB) analysis is a very efficient way to simulate weakly nonlinear circuits. Also, HB analysis works better than shooting analysis (in the time domain) for frequency dependent components, such as delay, transmission line, and S-parameter data.
- An HB analysis consists of two phases.
  - The first phase calculates an initial solution.
  - The second phase uses this initial solution to compute the periodic or quasi-periodic steady-state solution, using the Newton method.
- The three most important parameters for HB analysis are *funds, maxharms, and oversamplefactor*.

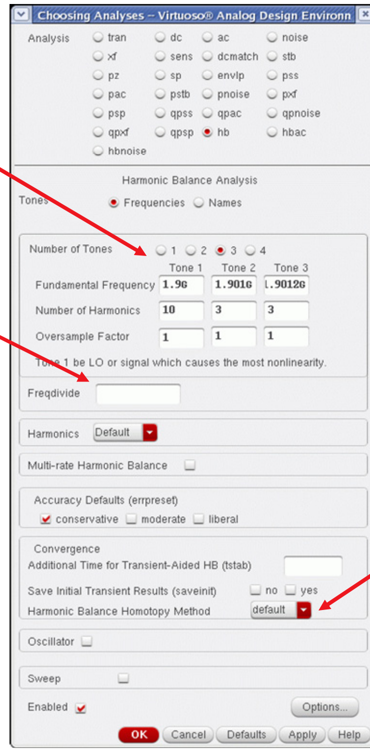
- The *funds* parameter accepts a list of names of fundamentals that are present in the sources. These names are specified in the sources by the *fundname* parameter. When only one name appears, the analysis is an HB PSS analysis. When more than one name appears, the analysis is an HB QPSS analysis.
- The *maxharms* parameter sets the number of harmonics that are to be calculated for that tone.
- Oversamplefactor should be set to 2, 3, or 4 for waveforms that have very sharp transitions. The sharper the transition, the higher the oversamplefactor.

Harmonic Balance Setup

If you want only harmonic balance, you can select the hb option directly from the *Choose Analyses* form.

Important: Set the largest tone as the first tone. If you select Names, enable tstab for that tone.

The freqdivide parameter is used to set divider ratios for the first tone in the list. When this is set, the actual frequency of the first harmonic that is solved for is the frequency in the choose analysis form divided by freqdivide.



Hbhomotopy = source is useful for very nonlinear circuits or high Q circuits.

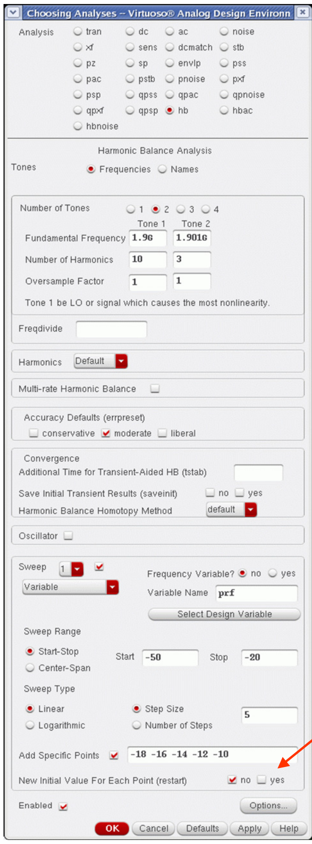
Hbhomotopy=tone is recommended for circuits with multiple inputs and one of the inputs is quite large.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 68

This illustration shows how to set up an HB analysis.

- The *Tones* field has two options; *Frequencies* and *Names* to define the fundamental frequencies used by the HB analysis.
  If Frequencies is selected, you can directly input the fundamental frequencies to be used by HB simulation. The Frequencies selection is recommended for composite triple beat (CTB) simulation.
- The *Harmonic Balance Homotopy Method* cyclic field determines the method used to pursue convergence in a circuit. The choices are *tstab*, *source*, *tone* and *gsweep*.

Setting Initial Value for Each Point (Restart)



Access the field from ADE in the Choosing Analyses form. Select hb in the Analysis field.

- Setting Restart=NO allows the simulator to use the answer from the previous point in a sweep as the starting point for the next point of the sweep.
- If the simulator is having trouble converging at a high-power level, create a sweep that starts at a low-power level and gradually build up to the higher power level. It works even better with fine steps in the sweep.
- Since the MMSIM7.0 version, the restart option is in the *Sweep* section in the Choosing Analyses form.

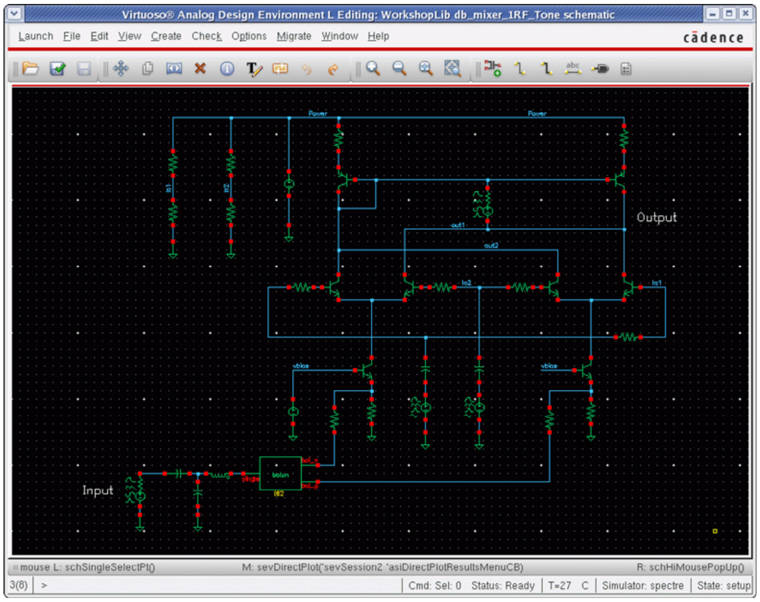
8/4/10 RF Analysis with Virtuoso Spectre Simulator 69

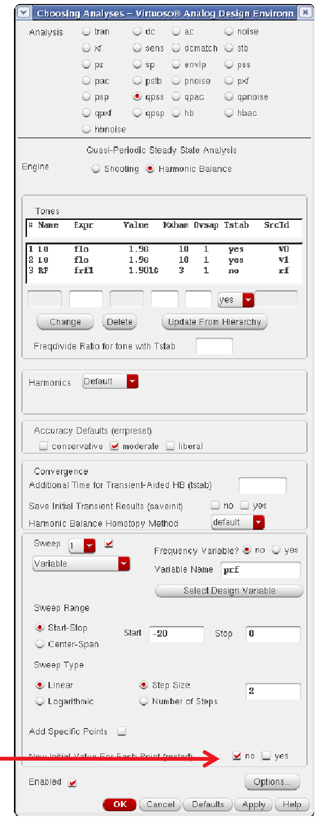
Setting Restart=yes means restart from scratch for each simulation.

Restart=no means take the value from the previous member as the starting point for the current simulation. For sweeps, this usually saves a lot of time.

For Shooting, always use *restart=yes*.

Swept HB QPSS Mixer Restart Example





Restart

This page shows the schematic and ADE analysis setup to illustrate the initial value restart option using QPSS with HB.

Swept HB QPSS Mixer Restart Example Results

Hard Compression Example
Power Sweep from -20 to 0 in 1db increments

Shooting had good results, but was slow.

Harmonic Balance with Restart=Yes (default) had trouble at higher power levels and began to reach the maximum number of iterations at each power level. Warnings were issued in the Spectre.Out file.

Harmonic Balance with Restart=No allowed the simulator to use the answer from the smaller power levels as good starting points for the higher power levels.

Significant gains in sweep simulation time can be realized with Restart = No.

Note that setting *restart = yes* dramatically improves the accuracy of the solution in the high power region.

Convergence for Shooting

- Add a small tstab.
- Set *stabcycles* to the 10 to 15 range.
- Set method to gear2only.
- Set relref to allglobal.

- Setting a small *tstab* makes a better starting point for the simulation, thus improving convergence.
- Setting stab cycles to the 10 to 15 range causes more simple optimization steps before the Newton iterations begin. This gives a better starting point for the Newton iterations, improving convergence.
- Setting method to *gear2only* removes trapezoidal ringing from the time-domain waveforms. The ringing is unpredictable and depends on the accumulated integration error. This can cause different amounts of ringing at the beginning and end of the waveforms, thus causing convergence difficulties.
- *Relref allglobal* prevents the timestep from collapsing to near zero.

Convergence for Harmonic Balance

- Set tstab to yes for the largest tone. If you are using the HB interface and Tones is set to Frequencies, set the largest tone as the first in the list.
- Set tstab to about 10 to 100 periods of the largest tone.
- Set hbhomotopy to source or tone. Source is useful for very nonlinear signals, or high Q circuits. Tone is useful for circuits with one large tone.
- For nearly linear systems, set itres to 1e-2.
- Set Maxperiods greater than 100 to allow more iterations.

Oscillators (Using Semi-Autonomous in the hb choose analysis form)

- Set tstab long enough for the oscillations to reach steady state.
- Try setting the linear oscic if there is no divider in the circuit.
- Try using the two tier method.
  - In this method, choose a single pin node inside the oscillator feedback system for single ended, and both nodes for differential.
  - Specify 1 for the harmonic index.
  - Specify an estimate for the peak-to-peak amplitude. If in doubt, specify a smaller value than you anticipate.

- Setting *tstab* to yes and specifying a number in the tstab field causes a transient to be run for tstab seconds. At the end, an FFT is performed and used as the starting point for the frequency domain iterations. This is closer to the real solution than the DC solution, so it converges better.
- Enabling *hbhomotopy* can be very effective for convergence. Source does a power sweep on the tone with tstab enabled, then applies all the other tones in the circuit. Tone does a PSS analysis for the signal with tstab set to yes, then does the qpss/hb analysis.
- Setting *itres* to 1e-2 forces a first solution in HB to have more precision. The precision is always increased in HB with more iterations. For very linear systems, it reduces the total number of iterations, because the first solution is more accurate and it runs faster.
- Allowing more iterations than the default might allow the system to converge.
- For oscillators, allowing tstab to be long enough to capture the steady-state behavior gives a very good first solution, thus improving the convergence.
- *Linear oscic* estimates the starting frequency and amplitude, again resulting in a better first solution.
- The two-tier method can improve the convergence by intelligently allowing more iterations. Make sure you set one (single ended) or 2 (differential) nodes inside the oscillator feedback system. Set the harmonic index to one, and set the magnitude to an estimate of the peak-to-peak amplitude. If you are unsure of the amplitude, specify a value you know is smaller than the amplitude the system actually produces.

Lab Exercises

Lab 3-1 Running a QPSS Analysis

Lab 3-2 Measuring Composite Triple Beat

cadence

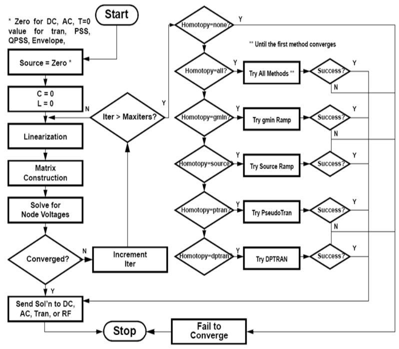
The Time Domain Solver

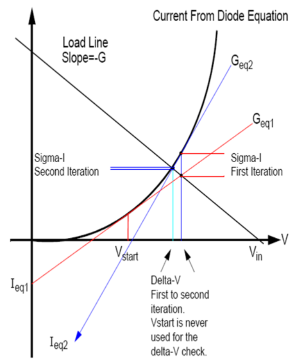
Module 4

August 4, 2010

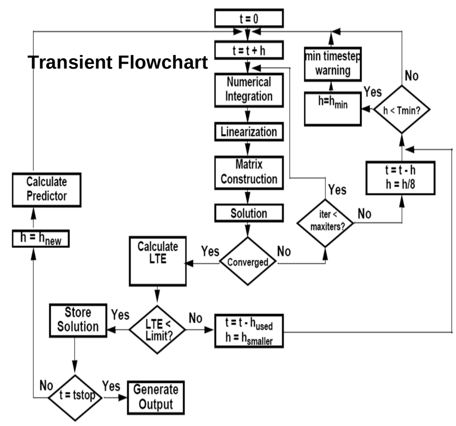
Shooting Engine Based on Time Domain Engine

DC Algorithm Flowchart for the Spectre Simulator





Newton–Raphson Iteration



The diagrams on this page are discussed in upcoming slides.

DC Algorithm in Spectre Simulator

The Spectre® simulator calculates a DC operating point separately for each analysis that requires a DC solution. Thus, the DC analysis is run separately and independently for the AC, Transient, PSS, QPSS, and Envelope analyses. The control of the DC algorithm is with the respective analysis that needs it.

There are two general classes of DC analysis.

- The first is the DC operating point (DCop) analysis. This analysis is performed when the DC analysis is run, and it is also needed for the AC analysis. By default, the time-dependent sources are set to zero and the initial conditions are ignored.
- The second is the Initial Transient Solution (ITS) analysis. Each transient-based simulation needs a time-zero time point, which usually comes from the DC algorithm for each specific analysis. The time-dependent sources are set to their time-zero value and the initial conditions are applied.

The normal time-zero transient point is the DC solution with the time zero source values applied. Because of this, we need to discuss the DC algorithm a little.

DC Conditions for Devices at Analysis Start

- All DC analyses:
  - Caps = open (by setting value=0)
  - Inductors = short (by setting value=0)
  - DC Sources = assigned values
- DC Operating Point Solution (OPS or DCop):
  - Signal sources = time-varying part set to zero volts
  - Initial Conditions ignored by default
- AC Analysis Operating Point Solution (ACop):
  - Signal sources = time-varying part set to zero volts
  - Initial Conditions ignored by default
- Transient analysis, PSS, QPSS, and Initial Transient Solution (ITS) for Envelope (which is the time-zero DC solution).

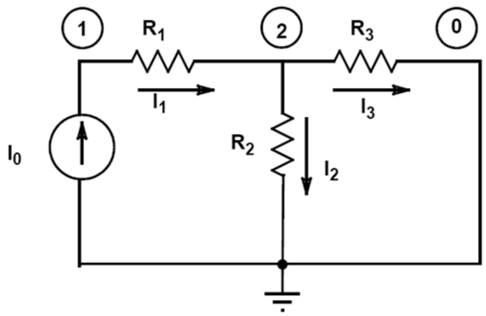
  Controlled by the Transient, PSS, QPSS, and Envelope analyses:

  Signal Sources = *Time-zero value*

  Initial Conditions observed by default.

This page identifies the DC condition for circuit components at commencement of the analysis listed.

Kirchoff's Laws



Kirchoff's Current Law

Sum of currents = 0 at every node

$I_0 = I_1$

$-I_1 + I_2 + I_3 = 0$

Ohm's Law

$G * V = I$

$G_1 * (V_1 - V_2) = I_0$

$-G_1 * (V_1 - V_2) + G_2 * (V_2 - V_0)$

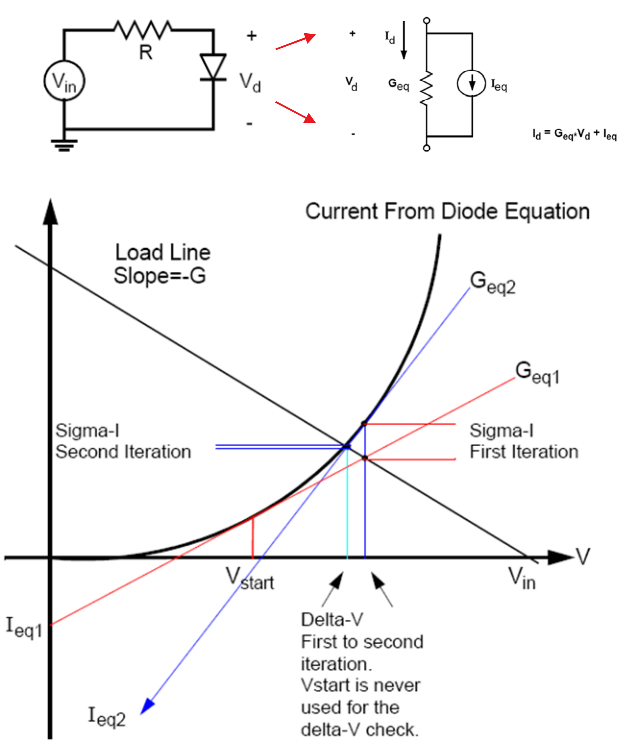
$+ G_3 * (V_2 - V_0) = 0$

- If all we have are linear elements, we don't need a simulator.
- Because the conductance of a semiconductor is not constant, we need a technique that lets us solve the simultaneous equations using only arithmetic.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 79

Assume that KCL is observed, and then do an Ohm's law substitution for the currents.

Newton–Raphson Iteration



- To accurately represent the tangent, both the slope (Geq) and the intercept (Ieq) are needed. This preserves the diode terminal characteristics at only the point where the tangent is taken. The device current and the tangent are supplied by the device model.
- On the third iteration, delta-V and sigma-I are very small, and the linear solution are very close to the nonlinear solution.
- The transistor models are more complicated than the diode model, but the idea is the same.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 80

- The Newton-Raphson method starts with a first guess V_{start}. This guess depends on the estimated operating region that is set on the device in the schematic. For the ON state, the point of maximum curvature in the diode I-V curve is chosen. For the OFF state, 0 volts across the junction is used for the first iteration.
- The diode equations are used to calculate the tangent (G_{eq1}) and the y-axis intercept of the tangent (I_{eq1}) at the first guess V_{start}. These values are used along with the resistor and battery values to construct and solve the matrix. *Remember that matrix algebra gives linear solutions*. This gives a new voltage at the intercept of the G_{eq} line and the load line from the battery-resistor combination. This is the first Newton-Raphson iteration.
- From this first solution voltage, the tangent to the curve is calculated again. The first derivative of the diode curve equation supplies the slope of the tangent G_{eq2}. The y axis intercept is calculated also. This is I_{eq2}. Note that both values are different than those used in the first iteration.
- The new matrix derived from the circuit can now be constructed and solved. Its results supply our new solution for the operating point. This is the end of the second iteration.

Each time in this example, the solution gets closer, but the exact solution can never be achieved. The question is, **when do we stop iterating?** The answer lies in **convergence**.

DC Convergence Criteria

After the matrix is inverted and the solution is calculated, then the circuit converges if;

- For every nonlinear branch current: (Update Criteria)

  $$| I_{dn} - I_{ln} | < RELTOL^* \max(I_{dn}, I_{ln}) + IABSTOL$$

  - where I_{dn} is the nonlinear diode current from the equation at the last iteration and I_{in} is the linear diode current from the solution at the present iteration. The difference in current is **delta-I**.
- For every node voltage: (Update Criteria)

  $$| V_n - V_{n-1} | < RELTOL^* \max(V_n, V_{n-1}) + VABSTOL$$

  - Where **n** is the latest iteration number, **delta-V** is the voltage difference
- At each node: (Residue Criteria)
  - The sum of all currents into each node is less than the absolute value of

  $$RELTOL * |Imax| + IABSTOL$$

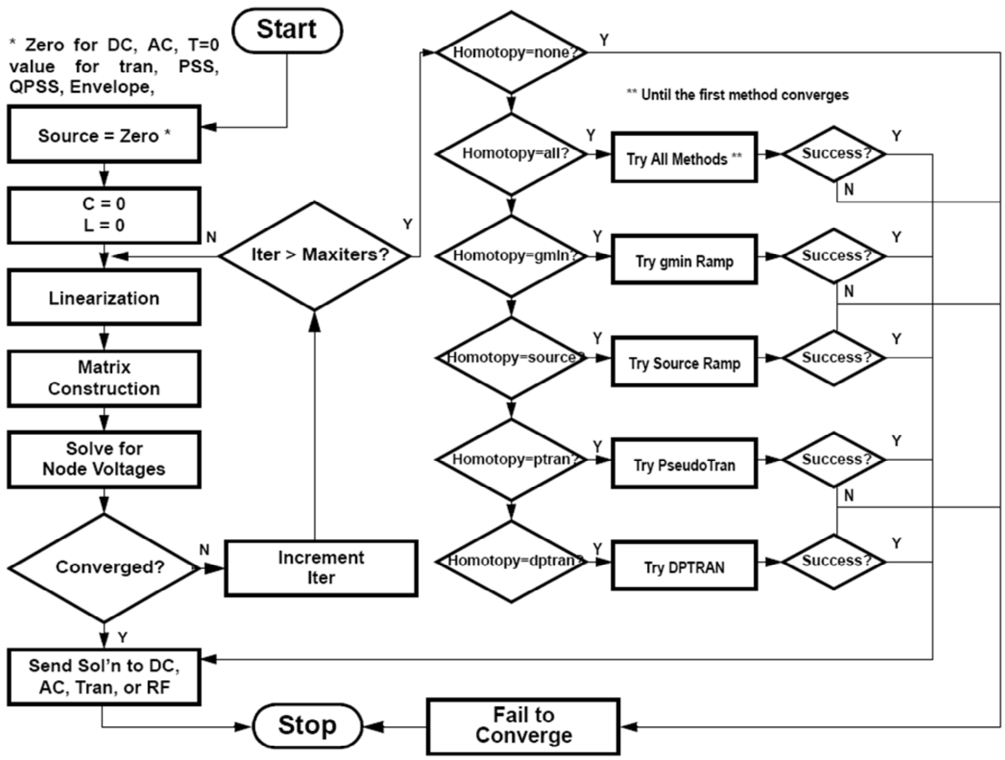
  - Where **Imax** is the largest current entering into the node from any one branch
- For signals larger than 1 millivolt and 1 nanoamp (by default) the convergence criteria error term is dominated by the relative tolerance (reltol).

8/4/10 RF Analysis with Virtuoso Spectre Simulator 81

- If the first *update* (nonlinear branch current) convergence criteria is not met, then the remaining two criterion are not checked for and solution proceeds to the next iteration.
- When the first update criteria is met then the delta-V (second update criteria) from the last iteration and KCL (residue criteria) check are performed.
- When all three convergence criteria are met, then the solution stops iterating. The exact solution is *never* achieved because there is another iteration that gets a closer solution.
- Only *nonlinear* components can cause the circuit *not* to converge because they get different Geq and Ieq values at each iteration.

- The Spectre simulator substitutes a KCL check at each iteration for the delta-I term that other SPICE-based simulators use. Because of the KCL check, the Spectre simulator is immune from false convergence at gigavolts or teravolts.
- The simulator itself doesn't conserve charge. Current is created or destroyed at each nonlinear node in the circuit. This is true of ALL analog simulators.
- The analysis options *reltol*, *vabstol*, and *iabstol* can be found in the Virtuoso Analog Design Environment window under **Simulation—Options— Analog**. These are global to all simulations run in ADE.

DC Algorithm Chart for the Spectre Simulator



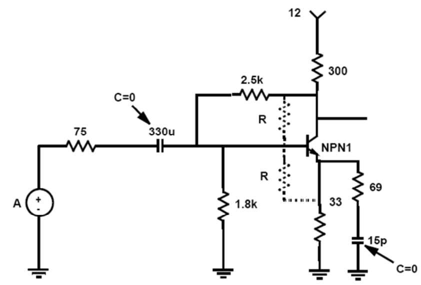
8/4/10 RF Analysis with Virtuoso Spectre Simulator 82

- The default for *maxiters* is 150.
- If the Spectre simulator uses more than one homotopy method before converging, set the *homotopy* option to the method that works in order to save time.
- You might also want to use the *write* and *readns* options to speed up convergence.
- The following few slides talk about different automatic DC convergence methods.

gmin Stepping

gmin resistors start small and are stepped up to *gmin*.

gmin [1e-12] {Global}

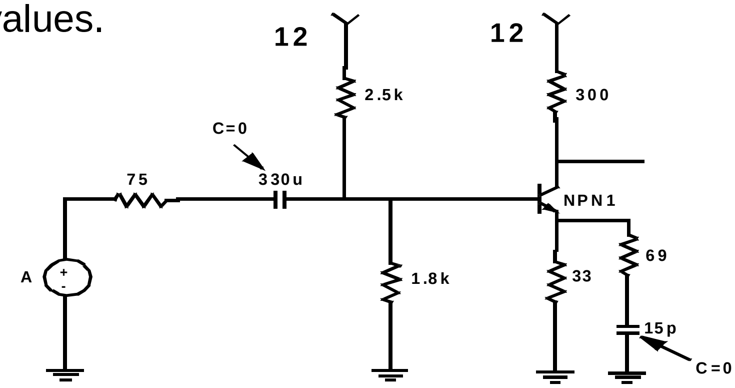


- *gmin* specifies the conductance across semiconductor junctions in diodes and BJTs, or S-D connections on MOSFETs and JFETs. Raising *gmin* may produce DC convergence but it will move the DC solution away from the real solution. Applies to all analyses.
- For diodes, a *gmin* resistor is across the diode.
- For BJTs, there is a *gmin* resistor across the B-C and B-E junctions.
- For FETs, there is a *gmin* resistor across the S-D connection. There is NO gmin resistor to the gate because it could dramatically effect the gate current.
- *gmin* resistors are always present in the circuit to prevent numerical problems when devices are off. The very small device conductance is paralleled by *gmin* so there is more leakage than the real device, especially when the temperature is set very low.
- For AC-coupled high-gain amplifiers, gmin may cause errors because of leakage currents that aren't there in the physical design.
- *gmin* is allowed to be set to zero, which means a perfect open, but numerical problems with the matrix can occur because of zero-valued entries on diagonal elements in the matrix.

- This is an automatic DC convergence method and can be accessed through the **Simulation—Options—Analog** form in the Analog Design Environment (ADE).
- The first simulation is performed with every resistor set to 1 ohm. Because the circuit is very linear, the simulator will likely have no problems calculating the voltages. If it doesn't converge, the *gmin* resistors are lowered till 1 milliohm is reached. If the circuit still doesn't converge at 1 mohm, this method fails.
- When convergence is reached at a resistor value, the simulator slowly increases the value of the resistors while using the current solution as the first guess for the next solution. The algorithm is adaptive with respect to resistor values depending on the number of iterations it took to converge at each step. A large number of iterations causes the *gmin* value to be stepped up slowly.
- The Spectre simulator continues to increase the value of the *gmin* resistors until they reach the value set by *gmin* or until the smallest delta-R value is tried and fails. Note that the devices will still have *gmin* resistors. These can be completely removed by setting *gmin* option to zero, but this is definitely not recommended.

Source Stepping

- **Source stepping**: The Spectre simulator executes this method (by default) if *gmin* stepping fails.
- All independent sources are ramped proportionally from near 0 to their specified values.

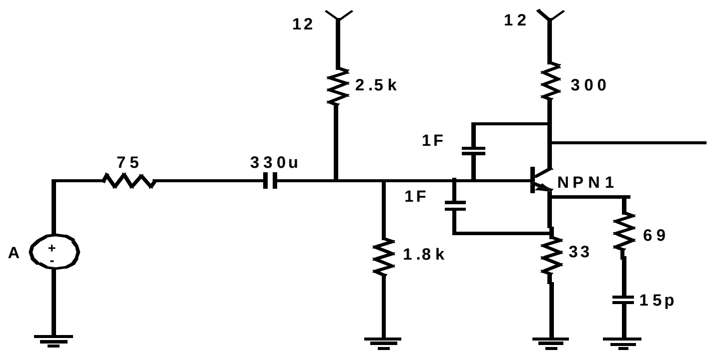


- Source A starts at V/1000, and is stepped up. When V is increased, the solution at the previous V value is used as the new starting solution. This continues until the independent sources are at the full voltage value. *maxsteps* controls the maximum number of intermediate values that are allowed. Default is 10,000.

- The source stepping method is similar to the *gmin* stepping method. Here the independent sources are ramped from 1/1000th of their nominal value to their nominal value instead of stepping the resistors. At each voltage point, a DC analysis is performed, with the solution being the starting point for the analysis at the next voltage. As in *gmin* stepping, the amount of change in the DC sources depends on the number of iterations at the last voltage. Because the supplies are near zero, convergence should occur near zero, and by stepping the value of the supplies, convergence should be obtained at the full supply value.
- Note that until the supplies are large enough to cause some current flow in the devices, the resistance of the devices can be quite high. This can cause a situation where it is difficult to converge to a single DC value, especially at the junction of two semiconductors. This can cause non convergence or singular matrix problems.
- The high resistor values combined with normal circuit values can cause numerical problems with inverting the matrix. This is one of the few times that reducing the *pivrel* value can be beneficial.

Pseudotransient (ptran) and Damped Pseudotransient (dptran)

- **Pseudotransient ramping**: The Spectre simulator uses this method (by default) after *gmin* stepping and *source* stepping fail.



- 1 farad capacitors with 0 volt initial conditions are added in parallel with the *gmin* resistors. Pseudo-Time is swept to near infinity. When time reaches near infinity, the last time point is used as the starting point for the first iteration in a normal DC analysis. Note that if the added caps cause oscillation, the method can fail because the ending time point isn't close enough to the real DC solution.

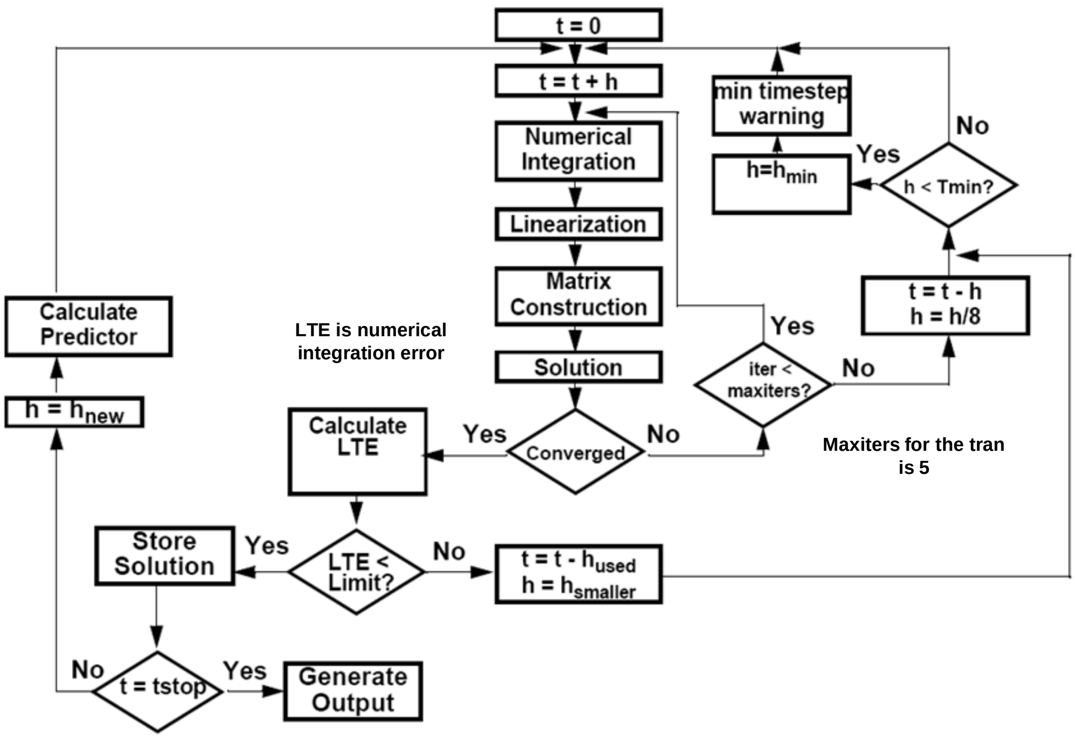
- The pseudo-time is swept to *50 * 1F * largest resistor value*.
- Simulation takes place normally with a LARGE numerical integration error allowed in the Transient analysis. The waveform to the final state doesn't matter. Only the final solution matters.
- The maximum number of time points is specified by the *maxsteps* option, which defaults to 10,000.

Transient Analysis Overview

- The simulation of the system is broken into discrete time points.
- The following steps are carried out by the simulator:
  - Capacitors and Inductors are added
  - Integral voltage and current relationships are represented with linear approximations at each time point.
  - The nonlinear elements are linearized
  - The Newton-Raphson iterative method is used at each time point.
  - The DC problem is solved at each time point.
  - Accuracy and time point spacing is controlled by establishing limits on iteration error and on numerical integration error .
  - The Spectre simulator reports the actual time points calculated. There is no printout interval property.
  - The time points are connected together in the waveform tool to present the waveform.

- A transient analysis can be seen as a series of DC analyses over time, except that capacitors and inductors are part of the analysis at each DC solution.
- Notice that the waveform obtained is *not* continuous. It is a series of time points, rather like a sample data system. Because we calculate the next time point assuming the current time point is correct, if there are unacceptable errors, the wave shape will be wrong.
- The same basic method as in the DC analysis is used for the linearization of the semiconductors. When the matrix is constructed at each time point, the capacitor and inductor models are added in along with the semiconductor models.
- The Spectre simulator now lets you launch small-signal analyses as well as info analysis at specific time points during transient analysis.

Transient Analysis Flowchart



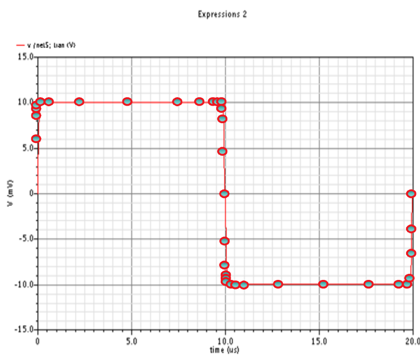
- At t=0 the Spectre simulator arbitrarily takes a **timestep h1 (t=t+h).** The simulator first calculates the **Geq** and **Ieq** values for all capacitors and inductors in the circuit by using the numerical integration method (*trap* or *gear*).
- Next, the simulator calculates the **Geq** and **Ieq** for all the semiconductors (linearization) in the circuit.
- Finally, the simulator builds the matrix and solves it. From the solution the simulator calculates **delta-V** and **delta-I** to determine if the circuit converges during the **Newton-Raphson** method. If the solution doesn't converge, the time step is reduced by a factor of eight and another attempt is made.

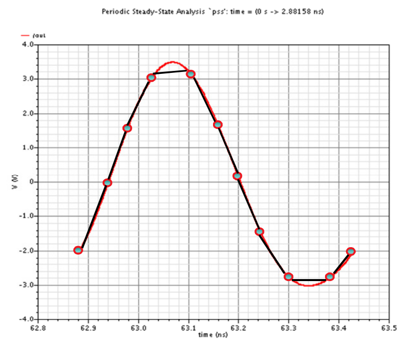
Note: For the first iteration the simulator has a predictor, and the predictor counts. Thus, it can converge on the first iteration.

- After this, the simulator calculates the **LTE (Numerical integration error)** and compares it with the limit. If the error is less than the limit, the solution is kept. Otherwise the simulator recalculates the entire solution for this time point using a smaller time step based on the LTE that was actually produced. By checking LTE, the simulator verifies the validity of the Geq and Ieq for the capacitors and inductors.
- When the solution passes checks for both the convergence and numerical integration methods, the simulator calculates the next time step based on the LTE that just occurred and the trend, and calculates the predicted voltage for each node.
- Then the simulator restarts the entire process for the next time point calculation.

Variable Time Step Algorithm

- The transient waveform is NOT continuous. It is a series of time points, rather like a sampled data system. You connect the time points together in the waveform tool to present the waveform. Because you calculate the next time point assuming the current time point is correct, if there are unacceptable errors, the wave shape will have inaccuracies.





- The time-step algorithm in the Spectre simulator was designed to conserve the number of time points simulated by putting more points during regions of sharp transition and to spread out the time steps during “well-behaved” sections of the waveform.

> Sinusoidal signals are smooth, so there are not enough time points to resolve the waveform perfectly. The piece-wise behavior raises the noise floor.

Maximum time step during analysis is set by the *maxstep* option and the default value depends on *errpreset*.

- For *liberal* it is (Tstop - Tstart)/50
- For *moderate* it is (Tstop - Tstart)/50
- For *conservative* it is (Tstop - Tstart)/100

The time step is reduced in areas of curvature. Areas of curvature are the areas where numerical integration error is produced. The time step is reduced in these areas to improve the accuracy of the wave shape where there is curvature.

reltol, vabstol and iabstol

After the matrix is inverted and the solution is calculated, the circuit converges if:

- For every nonlinear branch current: (Update Criteria)

  $$| I_{dn} - I_{ln} | < RELTOL * \max(I_{dn}, I_{ln}) + IABSTOL$$

  - where I_{dn} is the nonlinear diode current from the equation at the last iteration and I_{ln} is the linear diode current from the solution at the present iteration. The difference in current is **delta-I**.
- For every node voltage: (Update Criteria)

  $$| V_n - V_{n-1} | < RELTOL * \max(V_n, V_{n-1}) + VABSTOL$$

  - Where **n** is the latest iteration number, **delta-V** is the voltage difference
- At each node: (Residue Criteria)
  - The sum of all currents into each node is less than the absolute value of

    $$RELTOL * |Imax| + IABSTOL$$

  - Where **Imax** is the largest current entering into the node from any one branch

8/4/10 RF Analysis with Virtuoso Spectre Simulator 89

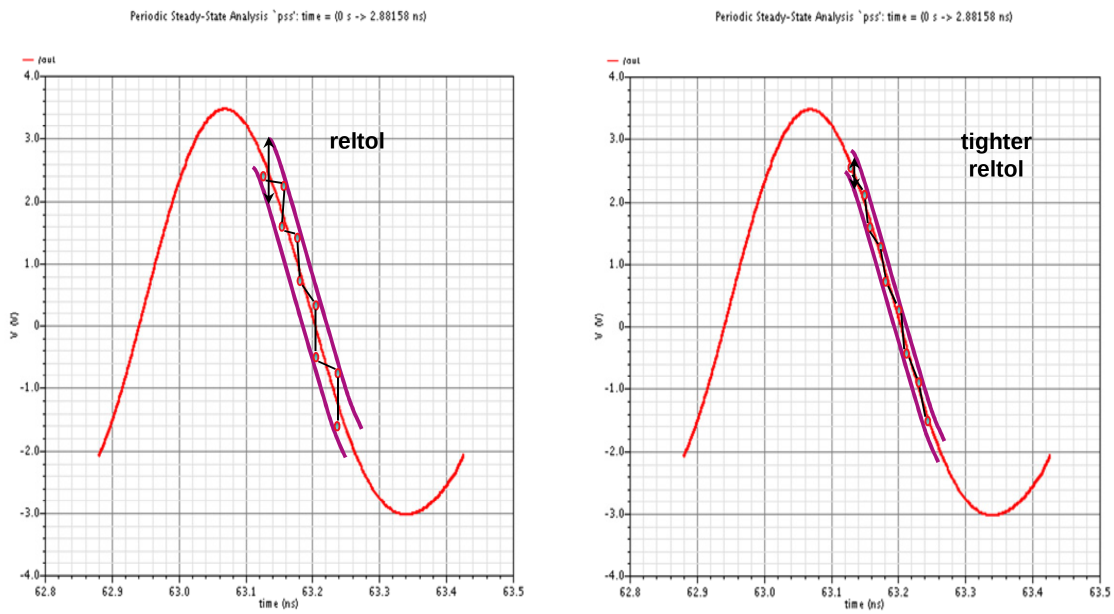
These **three tolerance options** set the fundamental accuracy of each time point by controlling the convergence criteria and local truncation error (LTE). The LTE ratio (*lteratio*) is used to set integration error tolerances relative to *reltol* and *abstol* to compute the local truncation error from Newton tolerances. The ratio cannot be less than 1.0.

Smaller tolerance values produce more accurate waveforms and smaller time steps in areas of transitions.

reltol has the largest effect in most circuits.

Note: The convergence criteria creates a fundamental uncertainty in the accuracy of each time point. This looks like noise in a Fourier transform where usually a log scale is used.

reltol



- When the solution is iterated to a point where Kirchoff's laws are met within a given tolerance, the simulator moves on to the next time point.
- This scatter between time points relates directly to the numerical noise floor when the time points are converted to a spectrum.

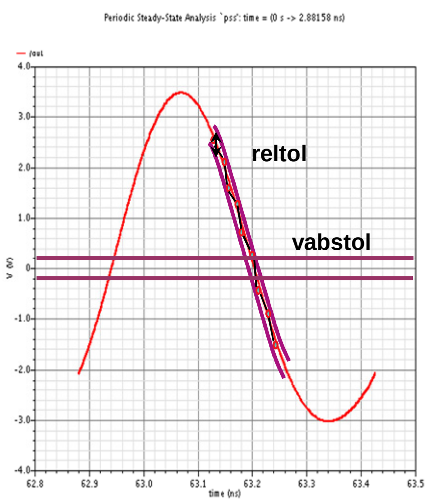
For convergence to be met, a very small difference between solutions in consecutive (last two) iterations is to be met. The *reltol* helps in setting this value as it sets the maximum relative tolerance for values computed in these iterations. The default value is 1e-03.

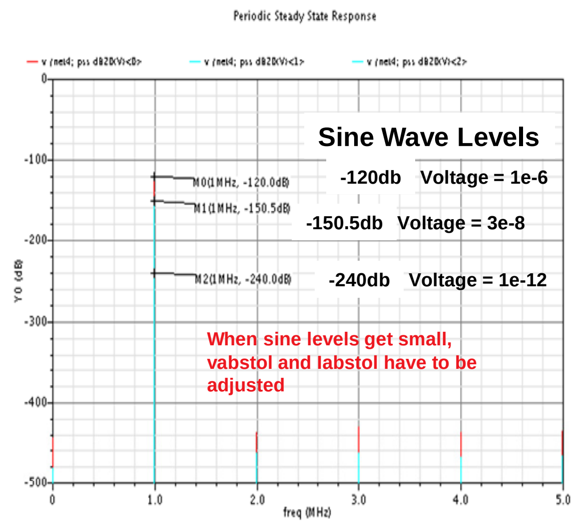
Decreasing the *reltol* value by a certain amount increases the number of time points considered by the shooting algorithm affecting convergence and simulation speed.

Raising *reltol* results in fewer, more inaccurate points. Except for a quick idea of the system performance, raising *reltol* is not normally suggested.

vabstol and iabstol

$$| V_n - V_{n-1} | < RELTOL^* \max(V_n, V_{n-1}) + VABSTOL$$





As the function crosses the axis,
| RELTOL * Largest (Vn, Vn-1) |
can become very small and the simulator will lose convergence because of the small deltas involved.

Adding the VABSTOL and IABSTOL constant gets it through this crossover region. However, if you are looking at very small voltages and currents, VABSTOL and IABSTOL can dominate the error function and you have to tighten them.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 91

The *vabstol* and *iabstol* parameters set absolute tolerances for differences between solutions in consecutive (last two) iterations as opposed to *reltol*.

These values are considers in conjunction with *reltol* to define the simulation convergence criteria.

With the defaults for reltol, vabstol, and iabstol, reltol dominates the error once the solution is larger than 1 mV and 1 nA. This dominance is because the convergence criteria is reltol plus vabstol for voltages and reltol plus iabstol for currents. When the solution is at 1 mV, the term from reltol is 1e-6 and the solution from vabstol is 1e-6. The terms are equal from *reltol* and *vabstol*. As the voltage gets larger than 1 mV, the relative term gets larger than the absolute term, thus it dominates the allowed error.

Suggested Settings for errpreset

Use errpreset *moderate* for most simulations.

Use errpreset *conservative* for cases where you want more accuracy at the expense of run time.

Errpreset *liberal* is very liberal. Use it carefully.

The *errpreset* Effect for Spectre Transient Analysis

Setting	Liberal	Moderate	Conservative
reltol	Times 10	Times 1	Times 0.1
lteratio	3.5	3.5	10.0
relref	sigglobal	sigglobal	alllocal
method	trapgear2	traponly	gear2only
maxstep *	SimTime/50	SimTime/50	SimTime/100

*If a sinusoidal source is present in the circuit, *maxstep* can never exceed the sinusoidal period divided by N where N is 10 for *liberal*, 10 for *moderate*, and 20 for *conservative*.

errpreset is a global way of "tuning" the accuracy of a simulation.

errpreset has three settings:

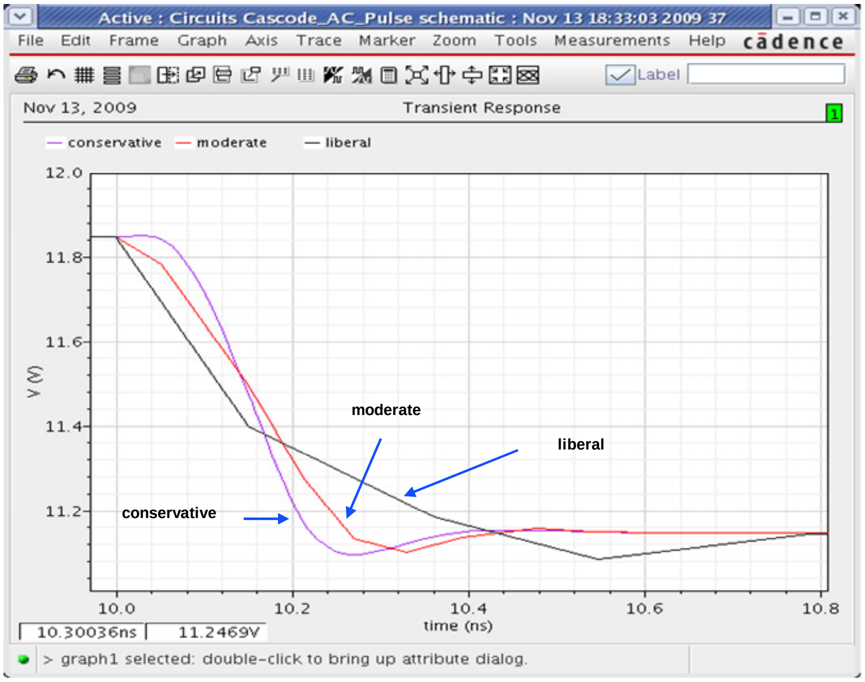
- liberal
- moderate
- conservative

If a sinusoidal source is present in the circuit, *maxstep* can never exceed the sinusoidal period divided by N where N is 10 for *liberal*, 10 for *moderate*, and 20 for *conservative*.

errpreset sets a number of defaults for the simulation. If you set these options manually, the Spectre simulator uses the setting you specified.

errpreset TRAN Example

This graph shows the transient simulation result of the Cascode circuit, using defaults for every other option and varying only the *errpreset* setting.



How do you judge which waveform is accurate?

Waveforms with obvious piecewise linear behavior are incorrect.

You can find the table in *spectre -h* that indicates what happens to all the options. This is the best way to find out, because we occasionally change the defaults.

Note that the values specified in ADE might not be the values used in the simulator. For example, if reltol is set to 1e-3 in ADE and *liberal* is chosen for *errpreset*, the actual value of reltol is multiplied by 10.

errpreset = Liberal was designed for square waveforms with well-defined breakpoints on the transitions. On smooth waveforms there CAN be quite a bit of inaccuracy.

The *errpreset* Effect for Spectre Autonomous PSS

Setting	Liberal	Moderate	Conservative
reltol****	1e-3	1e-4	1e-5*
lteratio	3.5	3.5	10.0**
relref	sigglobal	alllocal	alllocal
method	traponly	gear2only	gear2only
Maxstep***	Period/50	Period/200	Period/400
steadyratio	.001	.01	.1

* In the shooting window, conservative always sets a maximum of 1e-5. TSTAB uses the setting for *reltol*.

** For *errpreset*=conservative, *lteratio*=10.0 by default. *lteratio*=3.5 *only* when you set *reltol*<=1e-4 $*$ (10.0/3.5).

*** Spectre sets the value of *maxstep* so that it cannot be larger than the value given in the Table.

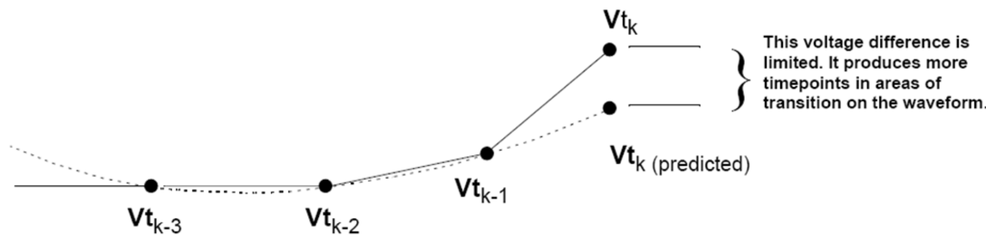
**** The value of *reltol* can be decreased from the default value in the options statement. The only way to increase *reltol* is to relax *errpreset.*

Except for the *reltol* and *maxstep* parameters, *errpreset* does not change the value of any parameters you have explicitly set.

The actual values used for the PSS analysis are given in the log file after simulation.

If *errpreset* is not specified in the netlist, liberal settings are used.

Iteratio (Local Truncation Error Ratio)



1. Imagine that an edge is input at Tk-2. Because of the capacitors in the circuit, the circuit begins to respond after some delay. The solid line represents the circuit waveform.
2. To calculate Vtk (predicted) a 2nd order curve (parabola) is drawn through the last three solution points (trap and gear2) [dashed line] or a straight line between the last two time points (Euler).

   This is the Spectre simulator's PRIMARY method of time step control. The algorithms for LTE time step control have received a tremendous amount of tuning to produce the fastest possible execution while letting you control the resolution and accuracy in areas of transitions.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 96

Local Truncation error is caused by unpredicted curvature. It's in areas of curvature that the time step must be reduced.

Local Truncation Error (LTE)

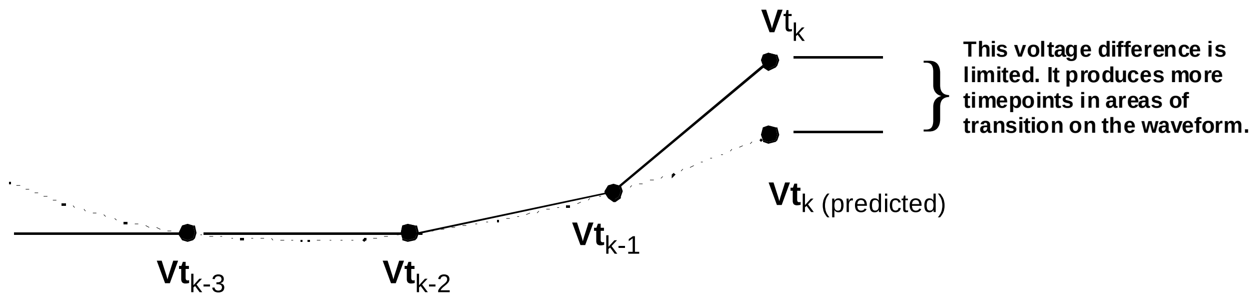
Before we decide to keep a particular time step after it converges, we must ensure that the error term from the numerical integration method is held to a *reasonable* value.

- A *reasonable* value is defined as
  - **| Vtk – $Vtk_{(predicted)}$ | < | Iteratio * [(RELTOL * $Vtk_{\{max\}}$) + VABSTOL] |**

    Note that this is the *lteratio * convergence* criteria.
  - The predicted value is a parabolic curve fit for *trap* and *gear2*, and a straight line between the last two data points for *Euler*.
  - Vtk is the voltage at this time point.
  - $Vtk_{\{max\}}$ is controlled by *relref*.
- This check must be done at each time point after the convergence criteria is met.

This is the Spectre simulator's *primary* method of time-step control. The algorithms for LTE time-step control have received a tremendous amount of tuning to produce the fastest possible execution while letting you control the resolution and accuracy in areas of transitions.

Local Truncation Error (LTE) (continued)



- Imagine that an edge is input at t_{k-2}. Because of the capacitors in the circuit, the circuit begins to respond after some delay. The solid line represents the circuit waveform.
- To calculate Vt_k (predicted) a 2nd order curve (parabola) is drawn through the last three solution points (trap and gear2) [dashed line] or a straight line between the last two time points (Euler).
- The Spectre simulator sets a limit on this voltage difference. The limit is set by

  {lteratio * ([RELTOL * Vtk$_{\{max\}}$] + VABSTOL)}

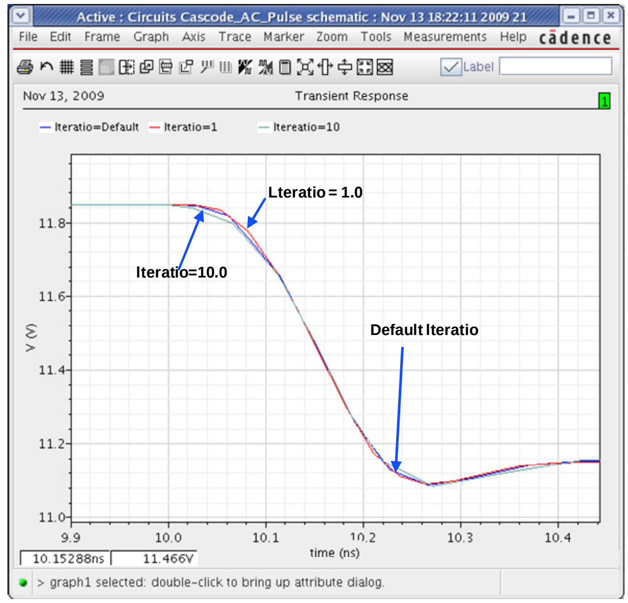
The time step is constantly controlled by the amount of LTE that actually occurred.

- Too much forces smaller time steps.
- Too little forces longer time steps.

The time step cannot more than double in the next time step.

lteratio (Local Truncation Error Ratio)

- The next time step that is calculated after a node has converged is a function of the time step used for this solution, the amount of LTE that occurred at this time step, and the LTE trend.
- If the LTE is low and decreasing, a longer time step is chosen. If the LTE is large (near the limit) and increasing, a smaller time step is chosen.
- Considerable work has been done to make the selection of the next time step result in the minimum runtime possible without sacrificing accuracy. The Spectre simulator almost never has to throw away a time point because of LTE being too high.



Transient response of the Cascode circuit with variations in the *lteratio* with

reltol=1e-3, relref=alllocal, method=traponly

The default value for *lteratio* is 3.5. It corresponds to moderate *errpreset* setting.

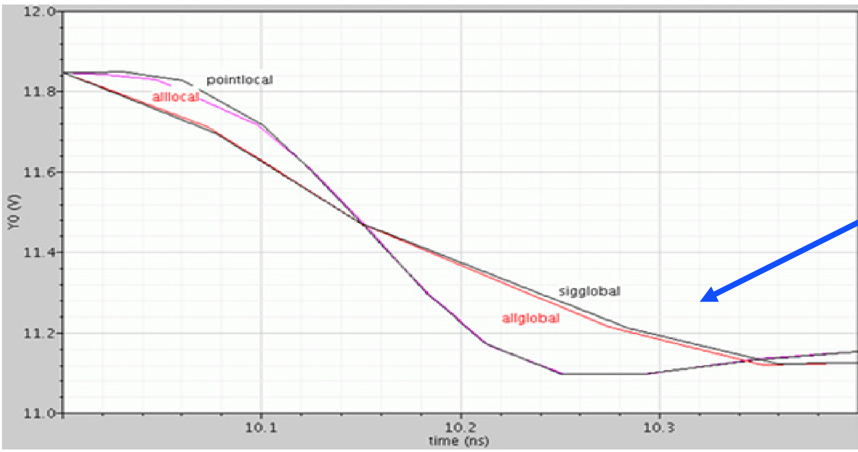
As *lteratio* gets smaller, the waveform gets more accurate.

The smallest value allowed for *lteratio* is 1.0.

What Is relref?

The quantity *relref* is used for the convergence criteria (at all transient-based time points, not DC), LTE (Local Truncation Error) calculations, PSS (Periodic Steady-State) beginning-to-end match, and Envelope beginning-to-end match. Values used are

- *allglobal*
- *sigglobal*: Default for both *liberal* and *moderate* error preset
- *alllocal*: Default for the *conservative* error preset
- *pointlocal*: Default for SPICE-based simulators



relref **example simulation with all other options at DEFAULT**

8/4/10 RF Analysis with Virtuoso Spectre Simulator 100

The normal default for Spectre RF is *alllocal*. In some cases, the time step can collapse to a very small value. If this happens, set *relref* to *sigglobal*. The solution will be a bit less accurate, but the time step problem is greatly reduced.

The relref Criteria: Applies to Transient-based Algorithms

There are three main convergence control criteria for *relref*.

- *pointlocal* specifies the largest branch current or voltage of the present or previous iteration at THIS node only. This applies to integration error also.
- *local* specifies the largest voltage or branch current solution that has occurred at THIS node or branch in the circuit *at any point* up to now. This also applies to integration error.
- *global* specifies the largest voltage or branch current that has occurred at ANY node or branch in the circuit at any solution point up to now. Usually, this is the power supply voltage and current at their respective maximums. This can cause large errors on nodes with small voltages or branches with small currents. This applies to integration error also.

relref	KCL Check	Delta-V	Integration (LTE)
allglobal	global	global	global
sigglobal	local	global	global
allglobal	local	local	local
pointlocal	pointlocal	pointlocal	pointlocal

Allocal is a good choice unless the time step gets too small. If this happens use *sigglobal*.

The method Option for Integration

The *method* option controls the integration method used by the Transient analysis to approximate the current in capacitors and voltage across inductors.

Linearizing caps and inductors with numerical integration:

- The relationship between current and voltage for an inductor is:

  $\Phi = LI$

  $\Phi = \int V\,dt$

  $LI = \int V\,dt$

- The relationship between current and voltage for a capacitor is:

  $CV = Q$

  $Q = \int I\,dt$

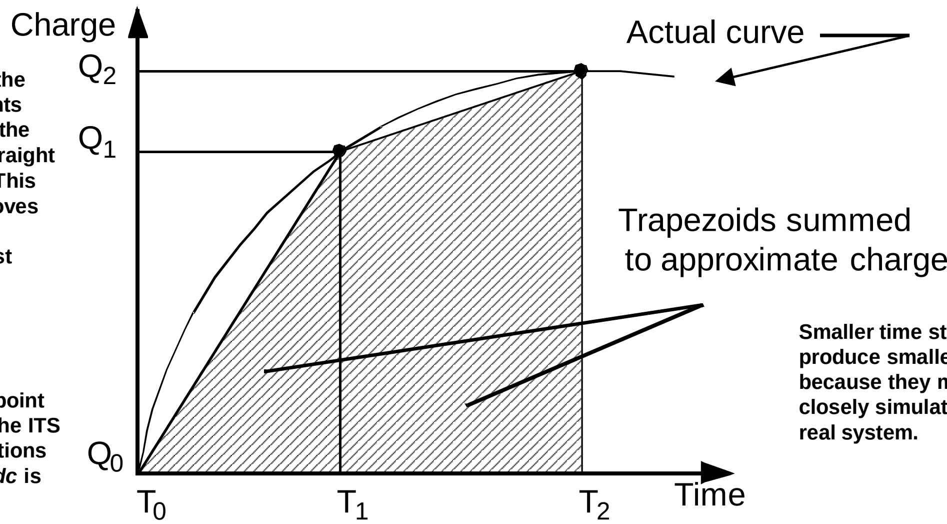
  $CV = \int I\,dt$

- By ***approximating*** this integral, we can calculate a linear slope and intercept with which to model the cap or inductor at each time point.

The Trapezoidal (Trap) Integration Method

Calculate the charge Q at a given time point n by approximating the area under the charge curve as the sum of trapezoidal areas:

Note that doubling the number of time points drastically reduces the area between the straight line and the curve. This tremendously improves the accuracy of the simulation at modest runtime cost.



Smaller time steps produce smaller errors because they more closely simulate the real system.

The time-zero time point comes either from the ITS or from initial conditions in the circuit if *skipdc* is Set.

In this example, the time-zero solution Is assumed to be zero.

$$Q_1 \text{ at } T_1 = (T_1 - T_0) * \frac{(I_0 + I_1)}{2}$$

$$Q_2 \text{ at } T_2 = Q_1 \text{ at } T_1 + (T_2 - T_1) * \frac{(I_2 + I_1)}{2}$$

The assumption is that the current is constant at the average current during the time interval.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 103

At each time point, the algorithm calculates the area under the curve (the charge Q) by using the trapezoid formula.

This method produces an error in calculating Q by replacing the curve associated with the real charge on the capacitor by a series of straight lines.

The current is assumed to be constant at the average current during the time step.

The Spectre simulator builds forward from each time point solution, thus the error produced by the integration method must be small at each time point to insure the accuracy of the overall wave shape.

The Different *method* Options Used

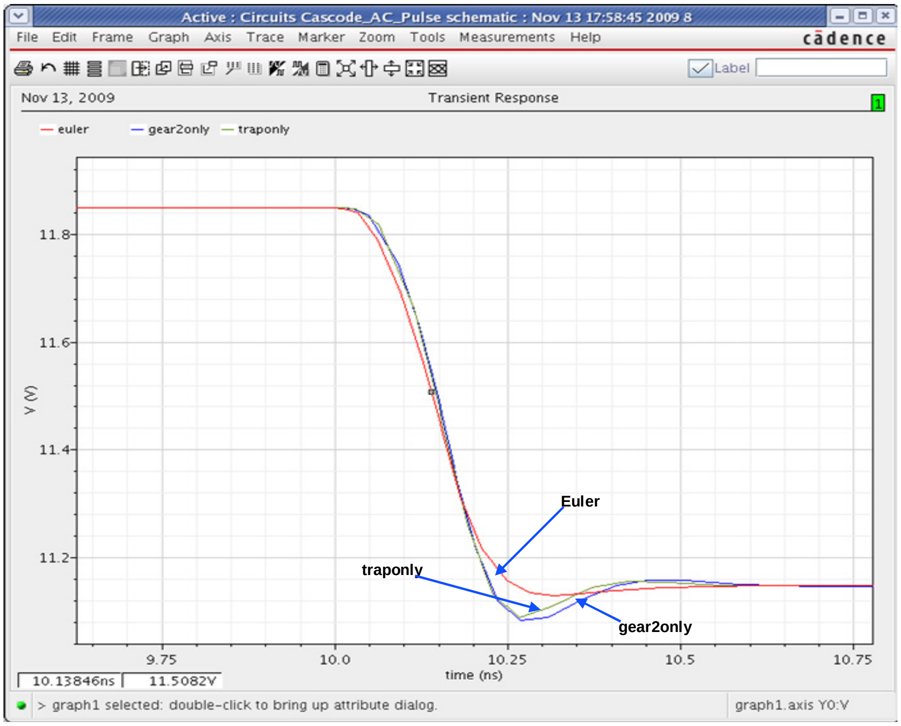
- *Euler*
  - It is a gear1 method. It numerically damps the circuit quite heavily.
- *trap*
  - Uses a mixture of *trapezoidal* and *Euler*. *Euler* is used at the first time point after each breakpoint and also when a longer timestep would result if *Euler* is used as the integration method.
- *traponly*
  - Uses a mixture of trapezoidal and *Euler*. *Euler* is used only at the first time point after each breakpoint. Otherwise, *trap* is used.
- *gear2*
  - Uses a mixture of *gear2* and *Euler*. *Euler* is used at the first time point after a breakpoint and also when a longer timestep would result when Euler is used as the integration method. Otherwise, *gear2* is used.
- *gear2only*
  - Default for *conservative*. Uses a mixture of *gear2* and *Euler*. *Euler* is used only at the first time point after each breakpoint. Otherwise, *gear2* is used.
- *trapgear2*
  - This method alternates between *trap* and *gear2* at each time point. *Euler* is used at the first time point after a breakpoint and also when a longer timestep would result when *Euler* is used as the integration method.

All Methods: The first time point after a breakpoint is a Euler integration.

- **trap**: This method is electrically neutral. Occasionally, an Euler step is used if it can take a longer time step.
- **traponly**: This method is the default for *moderate* errpreset setting. It is electrically neutral.
- **gear2**: This method is the default for *liberal* errpreset setting. *gear2* slightly numerically damps. This effect nearly disappears if the time step is kept small in areas of curvature. With gear2, occasionally, a Euler step is used if a longer time step can be used with Euler.
- **gear2only**: This method is the default for *conservative* errpreset setting. Damping effects are similar to gear2.
- **trapgear2**: Alternates between trap and gear2.

Illustration of Various Integration Methods

Transient response of a Cascode Circuit by varying the integration methods with all other default settings used except for *relref* = alllocal



8/4/10 RF Analysis with Virtuoso Spectre Simulator 105

Notice the difference in the waveforms. *Euler* visibly damps, whereas *traponly* and *gear2only* are similar.

Trapezoidal rule ringing in PSS analysis can cause the shooting iteration to stall before convergence is achieved. You can remedy this problem by changing the PSS options method parameter from *traponly* to *gear2only*. The only time to use *traponly* is for very high Q circuits.

Controlling Accuracy with maxstep

maxstep

- Setting a very small *maxstep* produces the most accurate simulation possible. If this is done, many time points are created, so the simulation takes a long time to run.
- The transient analysis builds the waveform assuming that the “current” solution is correct. Longer time steps tend to produce more numerical integration errors at each time point, and so longer time steps distort the shape of the waveform in transitions.
- Setting *maxstep* large has the potential of producing very large time steps, especially if the *local truncation error* (LTE) options have been relaxed. Setting *maxstep* small is a brute-force way of controlling accuracy.
- In most circuits, you WANT the inherently variable time step to reduce the run time. Use this setting carefully keeping the tradeoff in mind.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 106

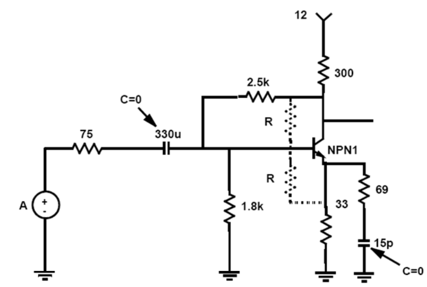
Setting *maxstep* small is a brute force way of making sure the correct answer is arrived at.

Forcing small constant time steps (using either maxstep or strobeperiod) gives the greatest accuracy.

Setting *lteratio* and/or *reltol* smaller will make the timestep smaller only where it is needed.

Convergence Aid (Not Recommended)

gmin [1e-12] {Global}



- *gmin* resistors are always present in the circuit to prevent numerical problems when devices are off. The very small device conductance is paralleled by *gmin* so there is more leakage than the real device, especially when the temperature is set very low.
  - For diodes, a *gmin* resistor is across the diode.
  - For BJTs, there is a gmin resistor across the B-C and B-E junctions.
  - For FETs, there is a *gmin* resistor across the S-D connection. There is NO gmin resistor to the gate, because it could dramatically effect the gate current.
- *For FET circuits and AC-coupled high-gain amplifiers, gmin may cause errors because of leakage currents that aren't there in the physical design.*
- *gmin* is allowed to be set to zero, which means a perfect open, but numerical problems with the matrix may occur because of zero-valued entries on diagonal elements in the matrix.

Convergence aid is not recommended because the circuit that is simulated has much more leakage than the physical circuit.

Wherever there is a gmin resistor, it gets the value specified by gmin.

The Spectre simulator places a single resistor across the source to drain connections for the JFETs and MOSFETs.

Standard SPICE puts a resistor from source to gate and from drain to gate. In general, don't change *gmin*.

Convergence Aid (continued)

cmin

- Controls the value of an added capacitor from each node to ground. Setting this can remove time step problems by preventing instantaneous transitions.
- To the extent that the real circuit DIFFERS from this setting, errors are introduced. These capacitors are not physical. If you have time step problems and you allow it, setting *cmin* from 10f to 50f might fix the problem.
- Setting *cmin* can eliminate time step issues by preventing instantaneous transitions.

cmin in the 10f to 50f range can produce convergence. However, beware that small capacitors are being added from every node in the circuit to ground. Exercise caution when using this option.

Summary: Controlling Accuracy and Resolution

reltol, vabstol, iabstol: Set the fundamental accuracy of each time point by controlling the convergence criteria and local truncation error. Smaller values produce more accurate waveforms and smaller time steps in areas of transitions. *reltol* has the largest effect in most circuits.

relref: Controls which voltages and currents are used to calculate the Delta-V and Delta-I terms in the convergence criteria, and in the LTE calculation. From least accurate to most accurate are *allglobal*, *sigglobal*, *alllocal*, and *pointlocal*. Try *alllocal* and if you have timestep problems, use *sigglobal*.

method: Controls the numerical integration method. The methods all have different amounts of numerical damping, and thus the shape of the transitions in the waveform. From least damping to most damping are *traponly* (electrically neutral, but has ringing), *gear2only* (damps, but negligible with small *reltol*), *Euler* (heavily numerically damps).

errpreset: Controls the default setting of *reltol*, *lteratio*, *relref*, *method*, and *maxstep*. All of these control the accuracy of each time point that is computed. Make sure that *errpreset* is set appropriately, or set the individual options as appropriate for the circuit and the test conditions.

maxstep: The Transient analysis builds the waveform assuming that the “current” solution is correct. Longer time steps tend to produce more numerical integration error at each time point, and so longer time steps distort the shape of the waveform in transitions. *maxstep* is a brute-force way of setting small time step. Setting *lteratio* and/or *reltol* smaller will make the time step smaller only where it is needed.

The normal method of controlling accuracy is to adjust *reltol* and/or *lteratio* to get a reasonable number of steps. To judge whether there are enough steps, zoom in to the area of fastest change. Look at the waveform. If there are visible pieces of piecewise linear behavior, you need more points, which can be accomplished by making reltol or lteratio smaller.

relref is preferred to be *alllocal*, but if time step problems arise, set it to *sigglobal*.

For shooting, it is best to set *method* and *tstab* to *gear2only* except for very high Q circuits. For harmonic balance, only tstab method is supplied, and any method except Euler can be chosen.

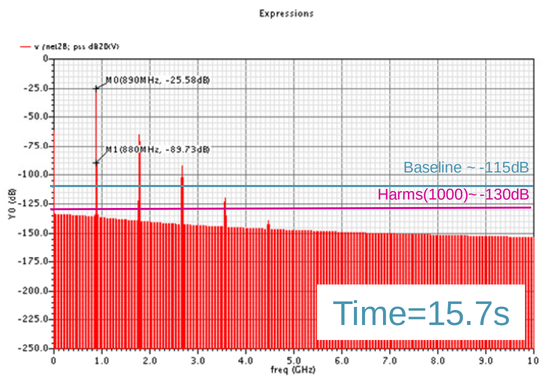
Errpreset is best set to *moderate* for most simulations, and *conservative* for those who want more accuracy.

Maxstep (or strobeperiod) is best reserved for those circuits where the utmost accuracy is desired.

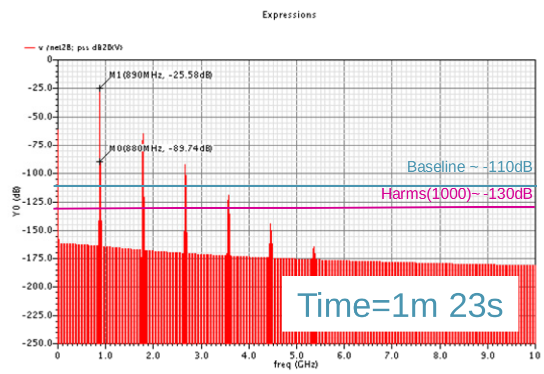
LNA_Simple: Controlling Time Steps

For Understanding Only

- The *maxstep* parameter is the longest time interval the simulator is allowed to go before adding an additional time step.
- Adding a small “Maxstep” time will give you the most accurate answer, but it adds a LOT of points and can make the simulation run VERY long with no added benefit in accuracy.



LNA		
Shooting		
PSS Interval	**10M**	
Harmonics	**1000**	
Convergence	**Cons**	
MaxStep		**Default**

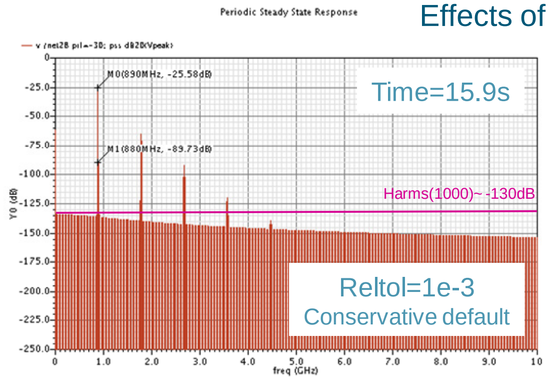


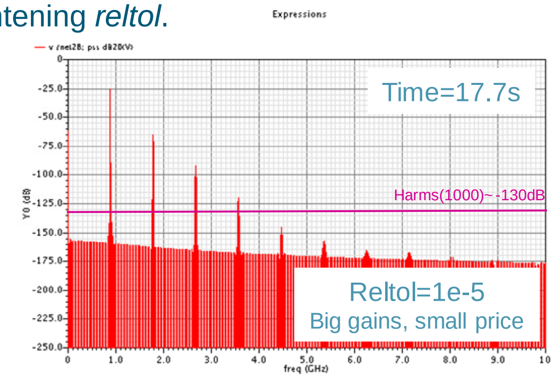
LNA		
Shooting		
PSS Interval	**10M**	
Harmonics	**1000**	
Convergence	**Cons**	
MaxStep		**1ps**
or		
MaxACFreq	**200G**	

The spectrum on the right has a better noise floor, because many more time points are forced by setting *maxstep* small or *maxACFreq* large.

LNA_Simple Convergence Criteria

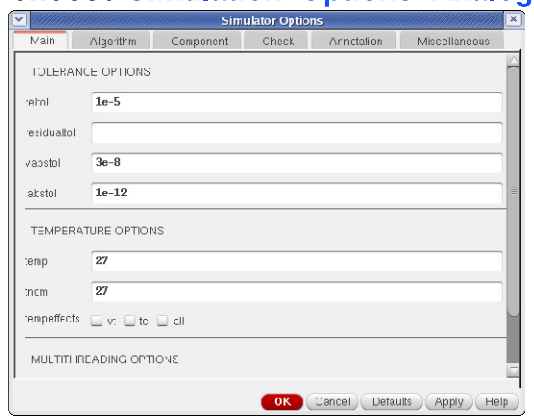
Effects of tightening *reltol*.





To change the *reltol* value, from the ADE window choose Simulation–Options–Analog

Be careful when tightening *reltol* because over-tightening this parameter can cause the simulator to run a very long time.



8/4/10 RF Analysis with Virtuoso Spectre Simulator 111

Setting *reltol* smaller caused more accurate time points. Thus the numerical noise floor goes down. (Shooting only)

Lab Exercises

There are no labs for this module.

RF Accelerators

Module 5

August 4, 2010

Module Objectives

This module explains some of the parameters that you can use to accelerate your RF simulation, keeping in mind:

- Accuracy

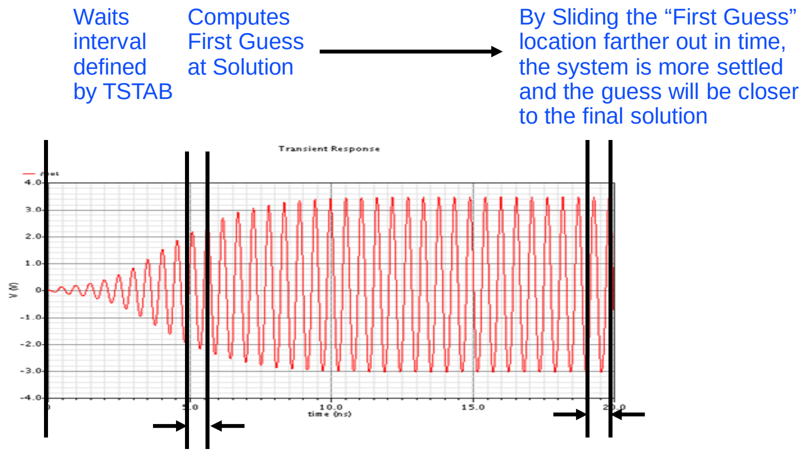
  and

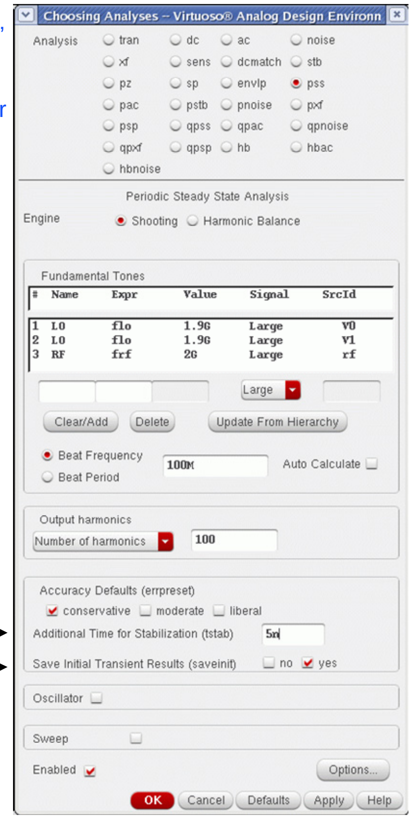
- Convergence

In this module, you

- See results from examples of RF amplifiers, mixers, dividers, and oscillators, illustrating the effects of the parameters.
- Understand basic Spectre RF turbo usage
- Become familiar with RF Accelerated Parallel Simulator (APS)

The Stabilization Parameter (TSTAB)





- Spectre RF makes use of an initial transient "first guess" at the solution matrix for the shooting interval to use as a starting point. The better the first guess, the faster the simulator can find the final steady-state solution.
- *tstab* is an important aid in convergence problems.

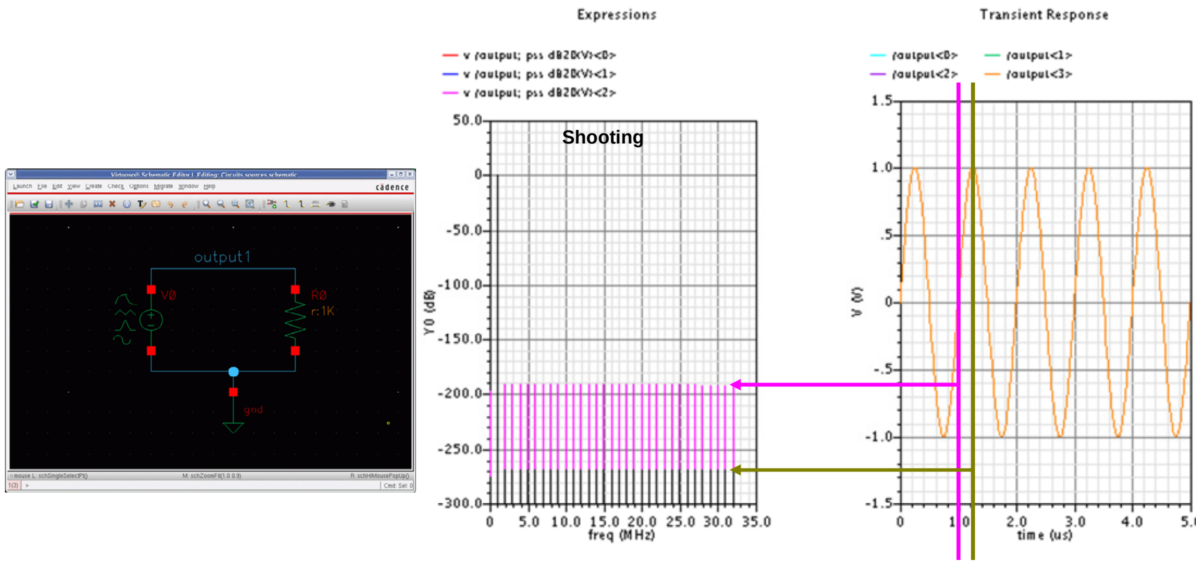
8/4/10 RF Analysis with Virtuoso Spectre Simulator 115

Providing a larger value for *tstab* usually improves convergence.

Occasionally, you must set *tstab* to a value equal to or greater than the time needed for the circuit to reach approximate steady state.

tstab can be accessed from the Accuracy section in the *Choosing Analyses* form.

TSTAB Starting Location: Partial Periods



Even for something as simple as a sine wave, moving TSTAB by ¼ cycle creates a much better starting point for the DFT.
Function is changing slower compared to Time at the GREEN starting point.

This is the simulation example of a simple sine wave source connected to a resistor.

You can estimate the required value of *tstab* by running a transient analysis.

Note: Both noise floors are very low.

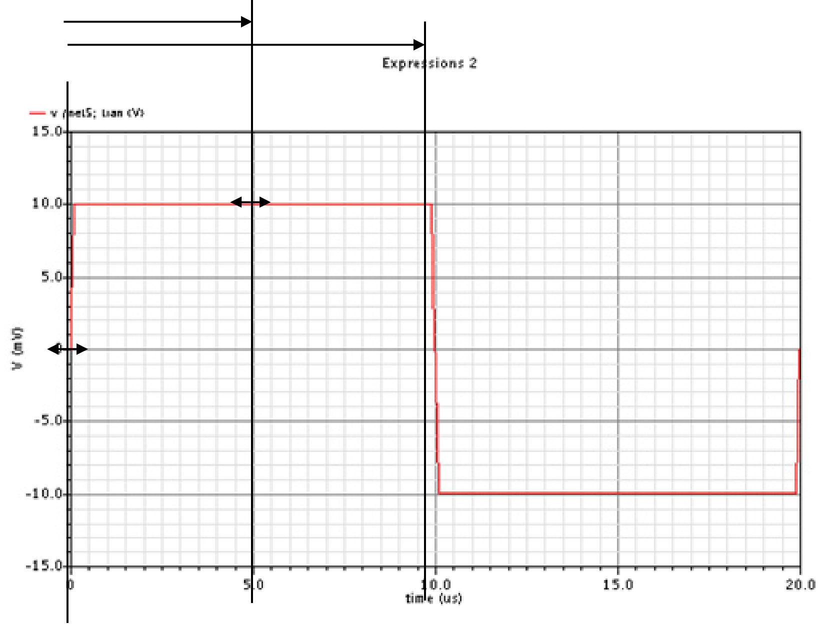
TSTAB for Square Waves

For pulses, TSTAB =blank is the right place.

Good Place

Best Place

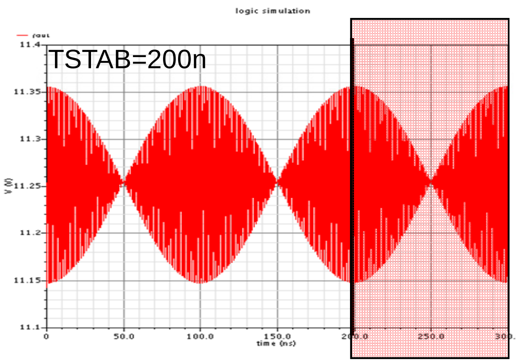
By moving to the end of the flat part (for settling), start and stop points of the PSS interval are well-behaved.

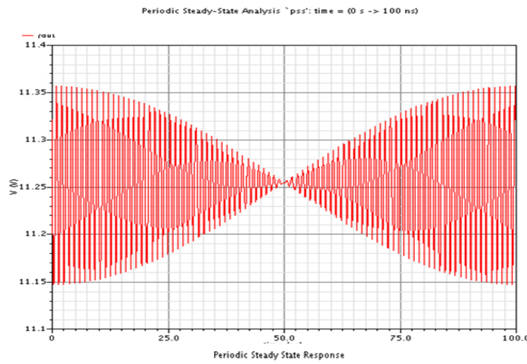


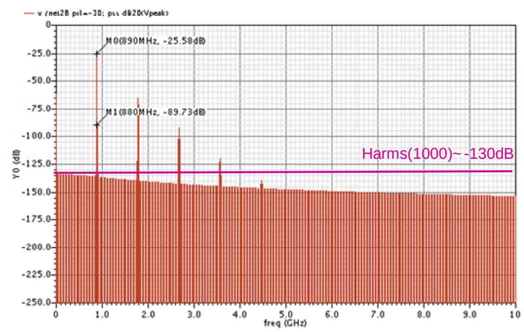
A small change in time 't' creates a large change in V. The signal is changing rapidly through this section, therefore convergence is more difficult.

The best place to have *tstab* set for a pulse is at the end of one of the flat parts of the waveform. This is the point that is most settled, and thus has the best convergence.

LNA: TSTAB=200n







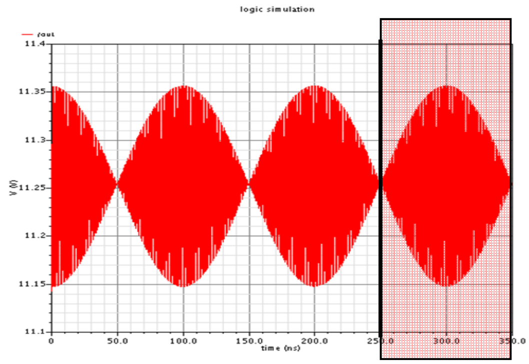
Adding one full cycle to the default produces no change in the dynamic range.

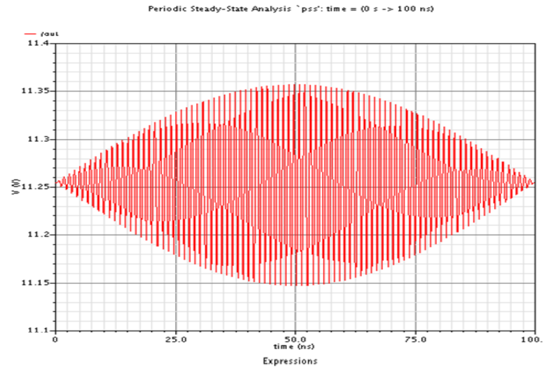
This means you are fully settled when you take the DFT.

tstab is set at 200 ns in this LNA simulation. The waveform response on the right is a zoomed-in version of the PSS analysis.

Note the noise floor in the spectral output waveform. This is about 0.3 microvolts RMS.

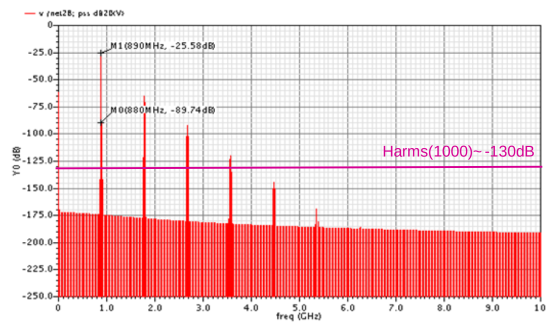
LNA: TSTAB=250n





By moving TSTAB ½ a cycle, the solution noise floor was improved by 50 db. The better the first guess is, the better the solution and the chances for convergence.

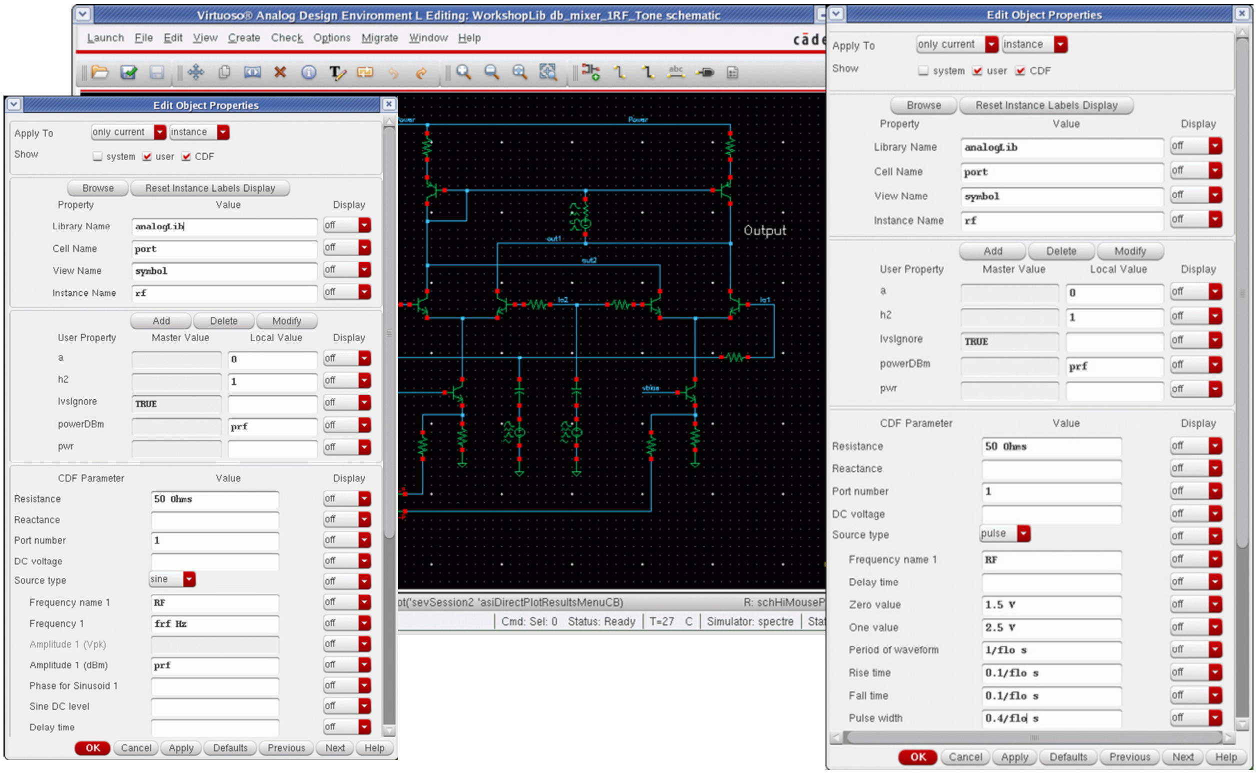
Guideline: Pick a place on the waveform where a small change in time creates a small change in voltage.



tstab is moved by 50 ns to the right at the start of the cycle. The waveform response on the right is a zoomed in version of the PSS analysis.

The noise floor is improved in the spectral output. The noise floor now is about 3 nanoVolts.

TSTAB Mixer Example: Adding Chopped LO



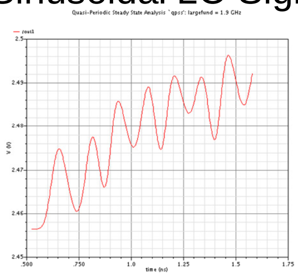
Sinusoidal LO Signal **Square Wave LO Signal**

Here is an RF Mixer schematic. The property windows are for the input RF source that has a 50 ohms internal resistance. The setup is shown for sine wave and pulse LO signals.

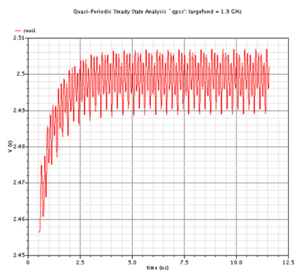
The setup is used to explain the effect of *tstab* when using QPSS Analysis.

QPSS: Mixer Example

Sinusoidal LO Signal



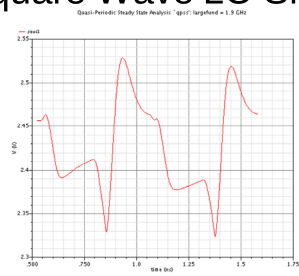
TSTAB=Default
Shooting 1m 7s*
HB 6s*



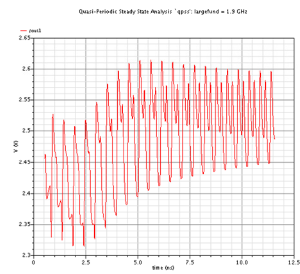
TSTAB=10ns
Shooting 48.9s*
HB 15.3s*

* Denotes total simulation time

Square Wave LO Signal



TSTAB=Default
Shooting: **No Convergence**
HB 22.24s*



TSTAB=10ns
Shooting 1m 11s*
HB 17.44s*

Extending TSTAB to 10 ns helped convergence and reduced the total simulation time.

This example also illustrates how multitone simulation with QPSS is often better suited for the Harmonic Balance Engine.

Harmonic balance is suggested for QPSS, because even for square wave input, it's faster.

Small *tstab* values can improve convergence by providing a better starting waveform to the QPSS algorithm.

Small *tstab* values can sometimes improve the overall run time as well.

Using TSTAB with Dividers

- Always use TSTAB
  - For circuits with a divider, always use a *tstab* of at least 1x the period (after divider).
  - Use *saveinit=yes* to check if the output signal is dividing properly.
- The number of harmonics chosen for the nonlinear tone needs to be at least 5 times the divide ratio.
  - This is regardless of whether you are using shooting or HB engine.
- You can use *Oversample=2* (or larger) to help add accuracy to the highly nonlinear divider signal.
  - This might let you get away with using fewer harmonics, lowering the number of unknowns, and saving memory.
- Only save a few key nodes, such as divider or mixer outputs, to save disk space.

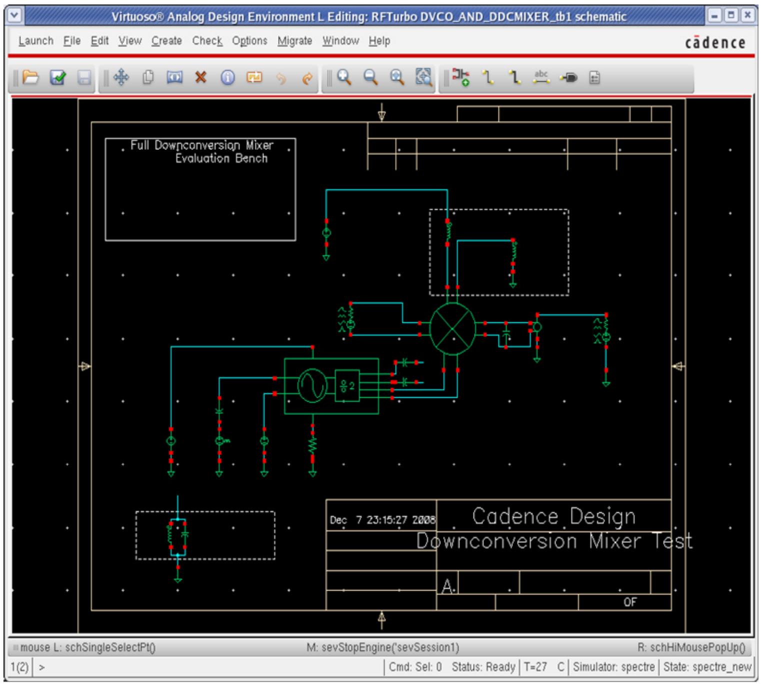
For a divider, always add at least the time it takes for the divider to divide one cycle to tstab.

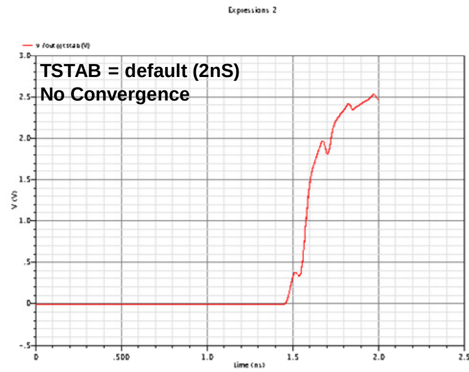
Remember that there is a square wave in the circuit. Use an odd number of harmonics.

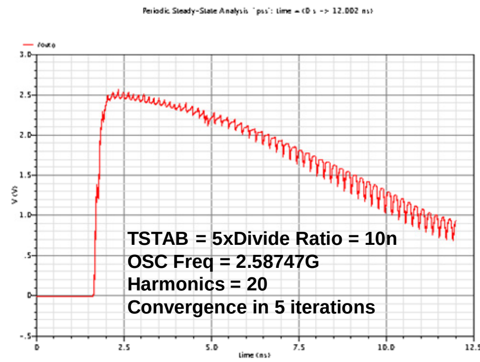
Use at least 5 harmonics on the input to the divider.

VCO+Divider with Mixer Example

PSS Shooting Oscillator Analysis







This example shows the schematic and PSS Analysis results of an RF down-conversion mixer with a VCO and Divider. The solutions are for the shooting method. The number of harmonics chosen is 20.

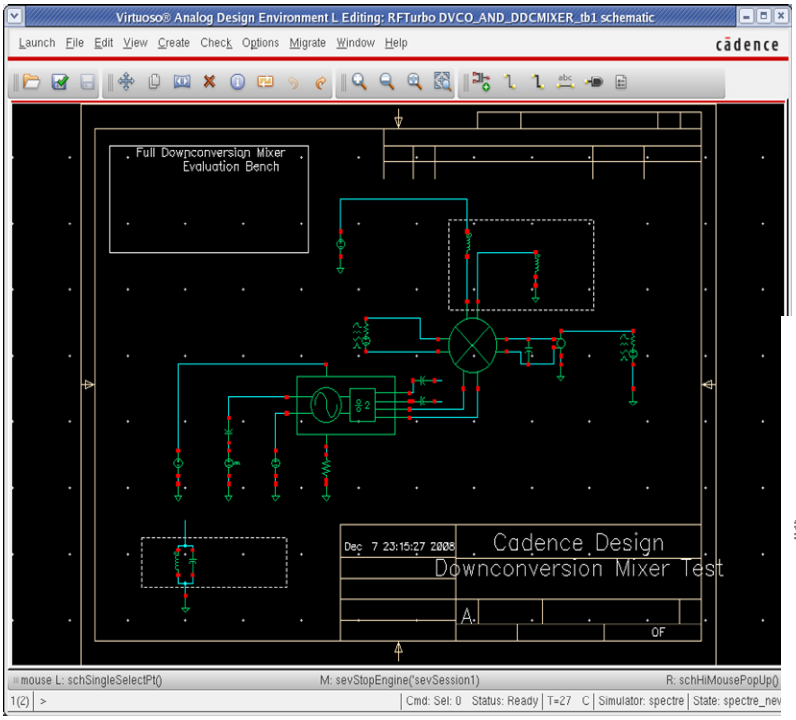
In the upper waveform, when a default TSTAB of 2 ns is used, note that the waveform has still not stabilized and hence the solution does not converge.

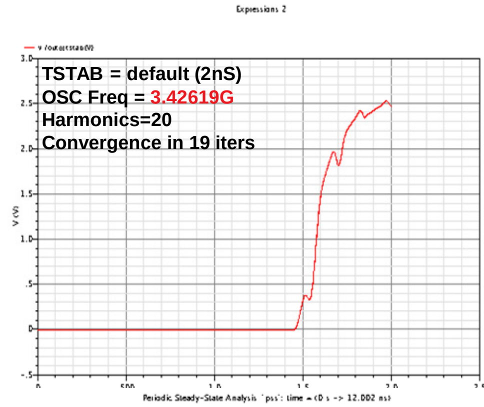
In the lower waveform, when the TSTAB value is raised to 5 times the period after the divider (five times the divide ratio), the solution converges.

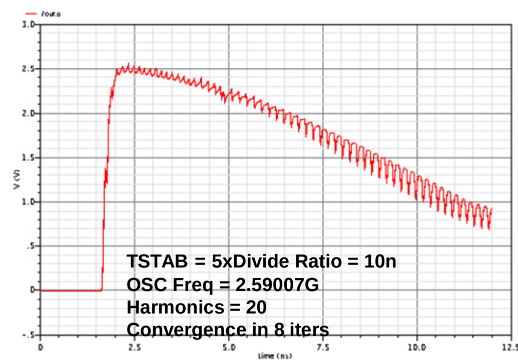
Usually, *tstab* just needs to be the time to divide once.

VCO+Divider with Mixer Example

PSS Harmonic Balance Oscillator Analysis







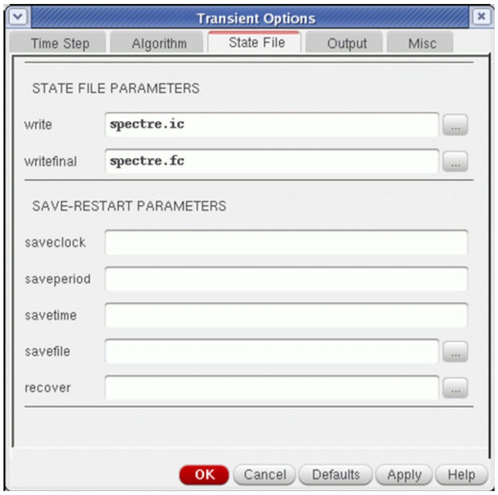
This pages shows the schematic and PSS Analysis results of a RF down-conversion mixer with a VCO and Divider. The solutions are for the Harmonic Balance Method and 20 harmonics are chosen.

In the upper waveform, when a default TSTAB of 2 ns is used, note that the waveform has still not stabilized and the solution converges after many iterations due to the harmonics chosen but the oscillator frequency obtained is incorrect.

In the lower waveform, when the TSTAB value is raised to 5 times the divide ratio, the solution converges though in more number of iterations than that for shooting.

Saving Restart Files for Tran: Dangerous Way

Using **readic, write** and **writefinal** options



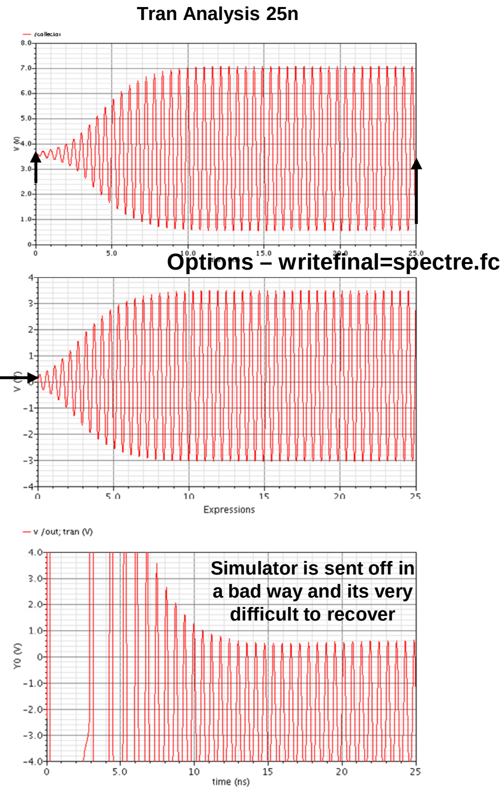
Options – State File
write=spectre.ic

Options – Algorithm
readic=spectre.ic

Notice oscillator starts up sooner with initial conditions saved.

Options – Algorithm
readic=spectre.fc

Simulator can go off in a bad direction, because only the node voltages are saved and there is no history to compute a derivative.



When *writefinal* is specified in the transient and is used as an initial condition in a tran or for the tstab interval of PSS/QPSS/HB, bad things can happen on startup.

It is much better to use Save/Recover.

Better Way to Save Restart Files for Tran

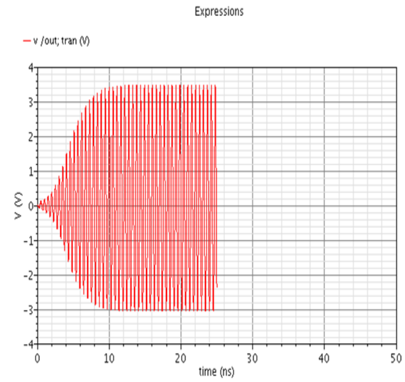
Use the **SaveFile – Recover** option.

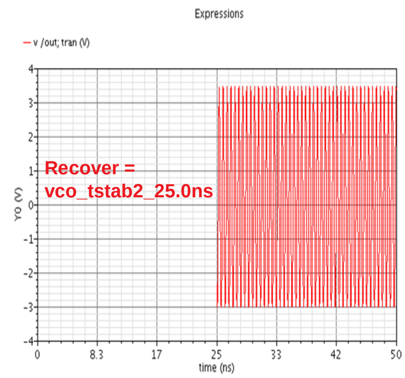
Savefile – Recover does a better job because the trend or history is stored which the simulator can compute a derivative from.

Format: "savefile" + "int counter" + "savetime"

```
[dneilson@pc-dneilson08 netlist]$ ls
amap          input.scs      raw            spectre3.ocn  spectre.sim
artSimEnvLog  map            runSimulation  spectre6.ocn  vco_tstab0_10.0ns
control       netlist        si.env         spectre.fc    vco_tstab1_20.0ns
ihnl          netlistFooter  si.foregnd.log spectre.ic    vco_tstab2_25.0ns
input.scllog  netlistHeader  spectre0.ocn   spectre.inp
[dneilson@pc-dneilson08 netlist]$ pwd
/export/home/dneilson/Workshops/spectrerf_workshop_new/simulation/oscillator_ckt
/spectre/schematic/netlist
[dneilson@pc-dneilson08 netlist]$
```

Recover = vco_tstab2_25.0ns





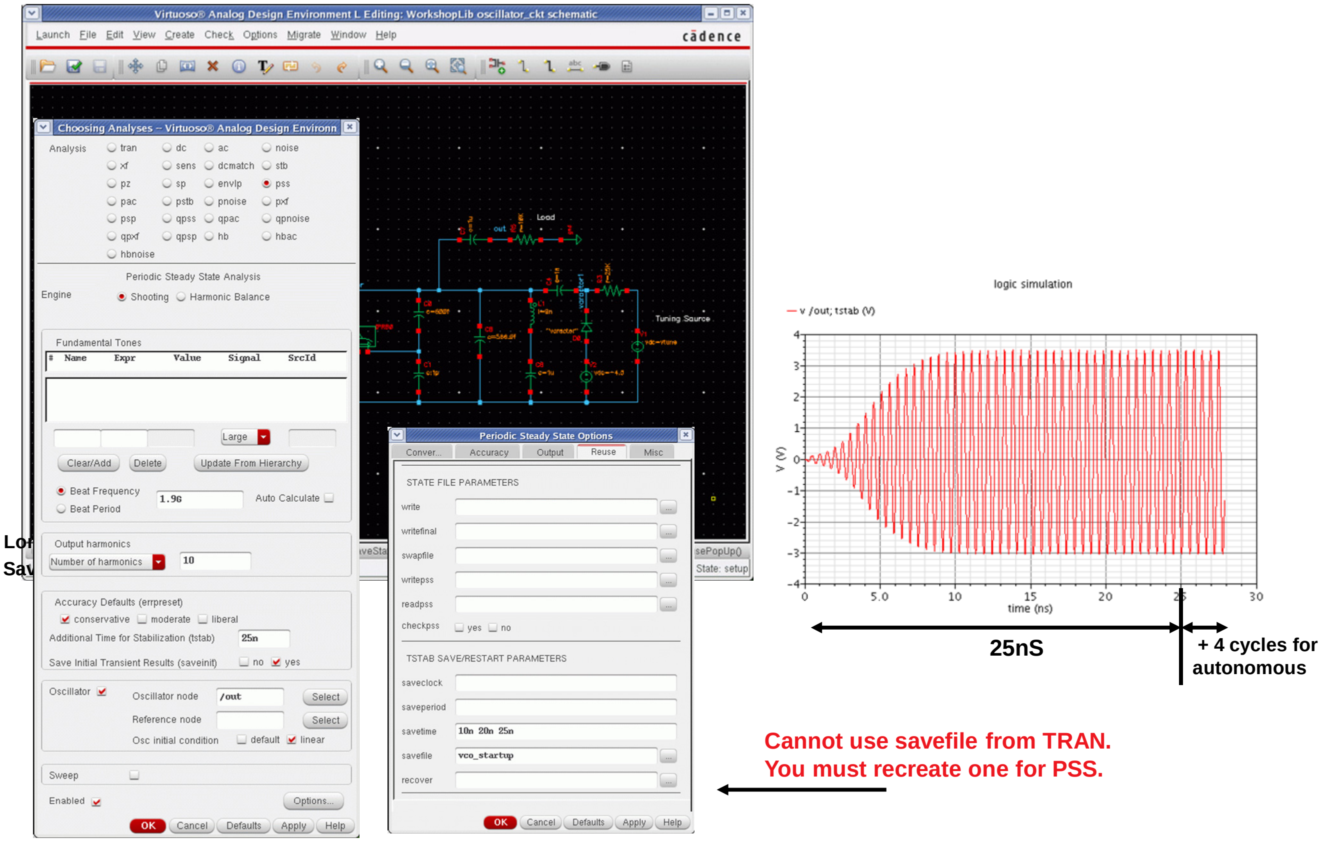
Save recover is provided for the transient and for the tstab interval in Spectre RF analysis.

The files are NOT interchangeable. Spectre RF files cannot be used in the transient and vice versa.

For RF, specify *savefile* and either *saveperiod* or give a list of times in savetime. Saveclock is not so useful.

When a list of times is given, look in the netlist directory to see the actual filenames. This is required for the restart option.

VCO: Saving TSTAB for PSS – Step 1



8/4/10 RF Analysis with Virtuoso Spectre Simulator 127

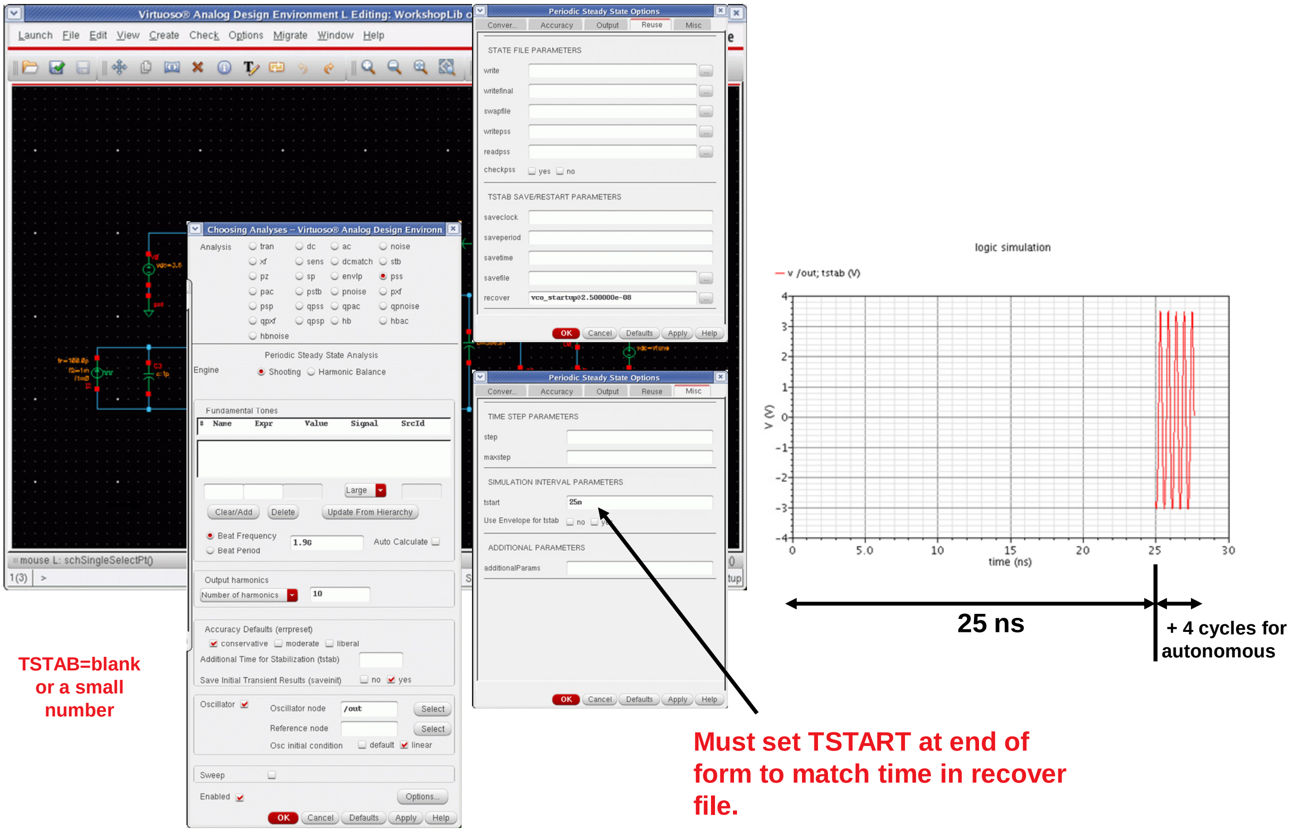
The same parameters are discussed earlier are available for the PSS analysis.

Click the **Options** button in the PSS Analysis form to access the PSS Options and you find the *write*, *writefinal*, *savetime*, *savefile* and *recover* options here under the *Reuse* tab.

Doing this lets you save the initial data that is simulated until the waveform stabilizes and it also gives you a range for *tstab*.

In this waveform, 25 ns (+4 additional cycle times for autonomous circuit simulation) are chosen for the *tstab* value. You can save this value and can recover the simulation from this point at a later time.

VCO: Saving TSTAB for PSS – Step 2

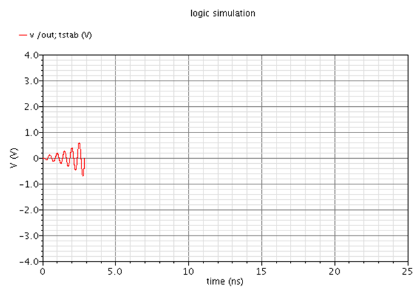


8/4/10 RF Analysis with Virtuoso Spectre Simulator 128

TSTAB is set at 25 ns as shown in the previous slide and the file is saved (using *savefile*).

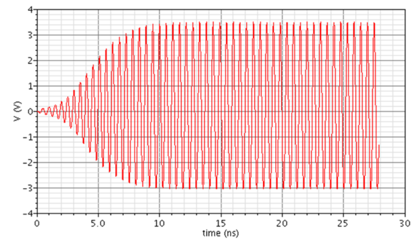
- You can now specify the filename at the particular time point in the *recover* field under the *Reuse* tab, which will invoke the data in the saved file.
- You also need to specify the Simulation Interval parameter TSTART under the *Misc* tab. Set this time to be the last time point specified in the *recover* file.

VCO TSTAB Variation and Simulation Summary

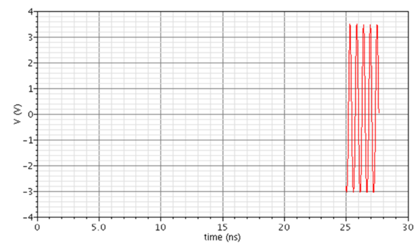


Actual Osc Freq = 1.83314GHz

PSS Iterations = 23 Iters
TSTAB = Blank
PSS Sim Time = 1.86S
TSTAB Sim Time = 100mS
Estimated OSC Freq = 1.98745 GHz



PSS Iterations = 5 Iters
TSTAB = 25nS
PSS Sim Time = .52 S
TSTAB Sim Time = 340 mS
Estimated OSC Freq = 1.83216 GHz



PSS Iterations = 4 Iters
TSTAB = Blank
PSS Sim Time = .42 S
TSTAB Sim Time = 150 mS
Estimated OSC Freq = 1.83398 GHz

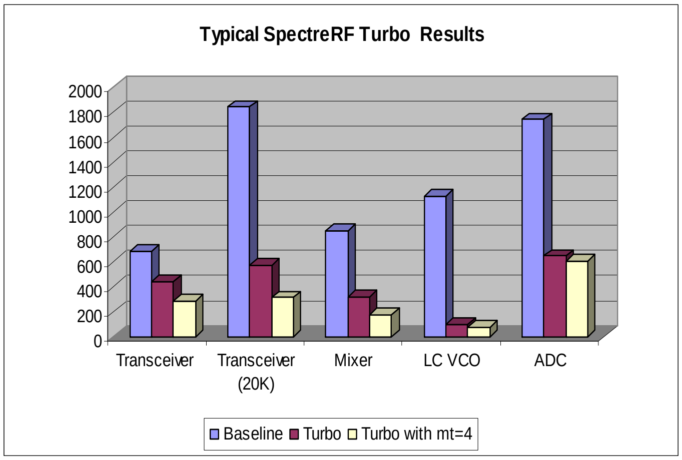
The results in the waveforms above are for an oscillator simulation.

- When *tstab* is shorter than the time needed for settling, the estimate for the frequency and amplitude is not good, so either many iterations are required or the system will not converge.
- Note that when tstab is set to a time long enough for the waveform to reach steady state, the total time is smaller and the number of iterations goes down.
- The best time occurs when recover is used with a short tstab.

Spectre RF Turbo

Spectre RF Turbo is a Cadence® technology that significantly reduces the run times of your Spectre RF simulations. Some of its features include:

- 2 to 5 times faster simulation run time (or more).
- Maintains golden accuracy.
- No "fast-spice" type shortcuts.
- Leverages multicore computing resources.



Spectre RF Turbo was introduced in mmsim7.1.1

- Turbo works by compacting the CMOS device models and by multithreading CMOS device current evaluation. This does not compromise accuracy of the model.
- Typical speedups are 1.5 to 5 times the speed of standard Spectre simulation.
- More than 4 cores is unlikely to provide much speedup.

Spectre RF Turbo Performance Enhancements

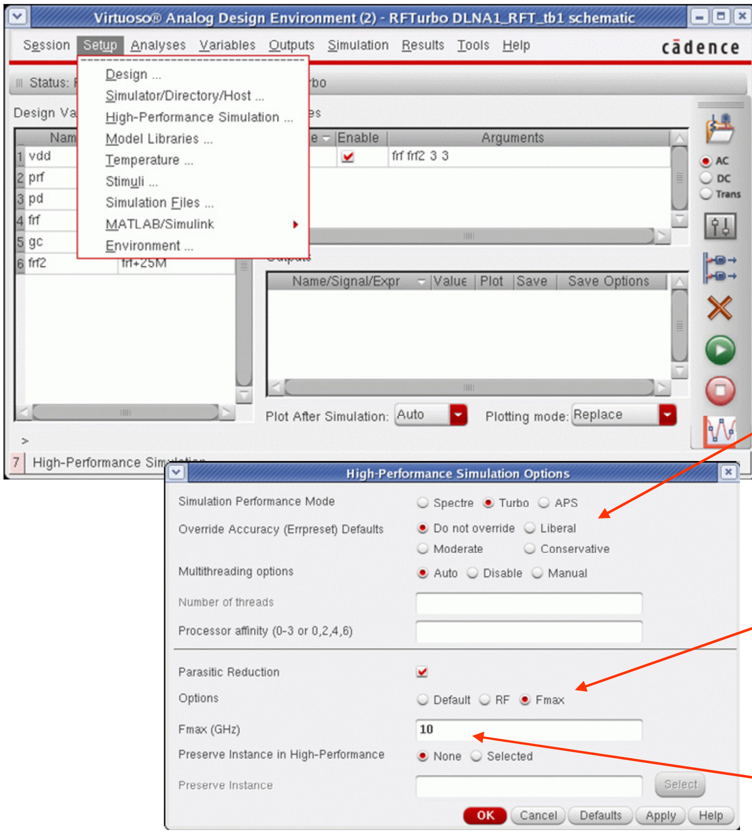
The Turbo technology increases the performance of the Spectre RF simulator in five main categories:

- Overall improvements in Core simulation technology.
- More efficient evaluation of models.
- Leverages multi-core computing resources.
- Includes advanced parasitic reduction algorithms.

Spectre RF Turbo Background

- Spectre RF Turbo is currently optimized for designs with large numbers of CMOS device models. Performance gains for designs with other device models, or for designs with a low transistor count, might not be significant.
- Turbo works by parallelizing device current evaluation and by making the CMOS device models more efficient
- Spectre RF Turbo provides higher performance gains for advanced device technology (90 nm and below).
- Large postlayout designs that contain large amounts of parasitic R and C elements can see significant performance gains from parasitic reduction, but gains from other Turbo technologies might be limited due to the dominance of the parasitic elements.
- To run Spectre RF Turbo, you need a **Spectre GXL license or 4 tokens**.

Spectre RF Turbo in the ADE Interface



- Turbo can take on of the following values: **liberal, moderate** or **conservative**.
- As it does for non-RF analyses, the value you specify for the turbo mode overwrites the *errpreset* value specified in the netlist for the RF PSS, QPSS and envelope analyses. This is not recommended.
- If you do not specify a value for +turbo, the *errpreset* value specified for the RF analysis is used. This is recommended.

- The +parasitics option can take either one of the following values: **Default**, **RF**, or **Fmax**. Default has a 1 GHz minimum pole frequency.
- **RF** has a 30 GHz minimum pole frequency.
- **Fmax** lets you set the minimum pole frequency.

Don't enter G after the pole freq for specifying GHz because it is the default unit.

You can access the Spectre RF Turbo form in ADE by choosing **Setup – High Performance Simulation**. It opens the **High Performance Simulation Options** form where you can choose Turbo as the Simulation Performance Mode.

- To access this option your simulator needs to be set to Spectre and the GXL license needs to be available.
- Select *Do not override* if you want the accuracy level specified for transient analyses runs in the Choosing Analysis form to be used.
- Select a default accuracy level (*Liberal*, *Moderate* or *Conservative*) if you want to override the accuracy level specified for transient analyses runs in the Choosing Analyses form.

Multithreading Options

- Select *Auto* to use the maximum number of available threads to run Spectre Turbo.
- Select *Manual* to specify the number of threads to use to run Spectre Turbo. Then specify the number of threads in the *Number of Threads* field.
- Select *Disable* to disable multithreading.

Parasitic Reduction

- Select **Default** to run parasitic reduction in default mode.
- Select **RF** to preserve the level of accuracy needed for RF analysis when running parasitic reduction.
- Select **Fmax** to specify an Fmax value for parasitic reduction.

Known Issues and Limitations of Spectre RF Turbo

- The presence of assertions (for device checking) slows down simulation and might degrade performance gains from Spectre RF Turbo.
- If device checking is not important in your simulation session, you can turn it off using the *-docl* or *-dochecklimit* command line argument

  spectre +turbo -docl …

- When running multiple multithreading simulation sessions on the same machine, Cadence recommends that you fix the simulation sessions to particular cores. Otherwise, different sessions can race against each other to get the available cores, thus degrading the simulation performance.
  - For example, when running two 4-thread simulation sessions on a 8-core machine, you might start the simulation sessions as follows.

    % taskset –c 0-3 spectre +turbo +mt=4 … &

    % taskset –c 4-7 spectre +turbo +mt=4 …

  - Note that *taskset* is a Linux command; the equivalent Solaris command is *psrset*.

Spectre RF APS Background

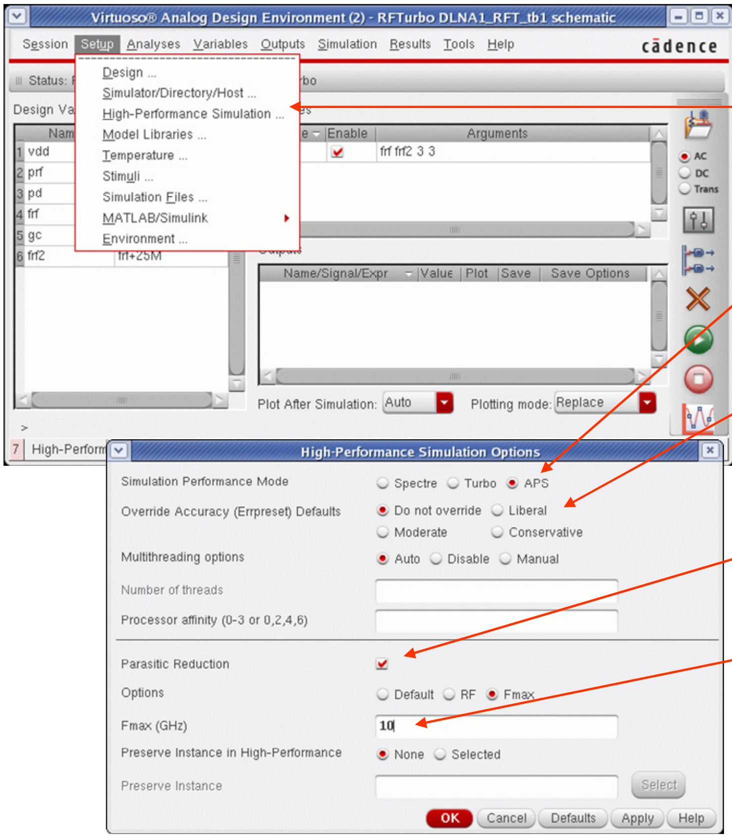
APS is the Virtuoso *Accelerated Parallel S*imulator.

- Spectre RF APS is effective on large CMOS designs.
- Spectre RF APS works by parallelizing all operations in the solution and by using the APS algorithm for the DC and tstab intervals.
- Spectre RF APS is usually three to ten times faster than standard Spectre.
- Spectre RF APS requires an APS-XL license or six 9003 tokens.

*Accelerated Parallel S*imulator (APS) is a next generation SPICE simulator that provides high performance, high capacity circuit simulation with full Spectre accuracy.

- APS achieves maximum simulation performance by enabling multithreading of all operations on multicore and multi-CPU shared memory systems. This feature lets you quickly simulate large pre- and postlayout designs.
- APS combines an advanced simulation engine with existing Spectre and Spectre Turbo technologies.
- It is primarily targeted at speeding up DC and Transient analyses in standard Spectre, and DC, tstab, and harmonic balance in the Spectre RF simulation.
- The APS use model is identical to the Spectre model, with the same netlist syntax, device model, analyses, features, and output format support.

APS in the ADE Interface



To select APS, choose Setup – High Performance Options in ADE.

Select APS.

Accuracy defaults can be overridden if desired. This is not suggested.

Parasitic reduction can be used for extracted simulations.

If you use Fmax, don't put a G after the frequency.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 136

You can access APS by from ADE by choosing **Setup – High Performance Simulation**. It opens the **High Performance Simulation Options** form where you can choose APS as the Simulation Performance Mode.

The options for accuracy, multithreading, and parasitic reduction are similar to those used for Spectre RF Turbo.

In the *Processor affinity* field, specify the processors using which simulation sessions are to run.

Lab Exercises

There are no labs for this module.

Blank Page

8/4/10 RF Analysis with Virtuoso Spectre Simulator 138

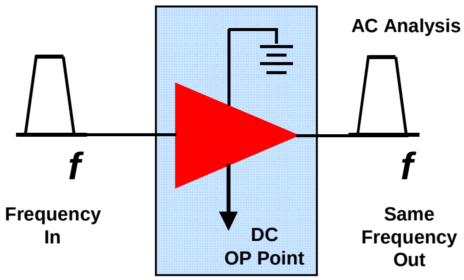
cādence

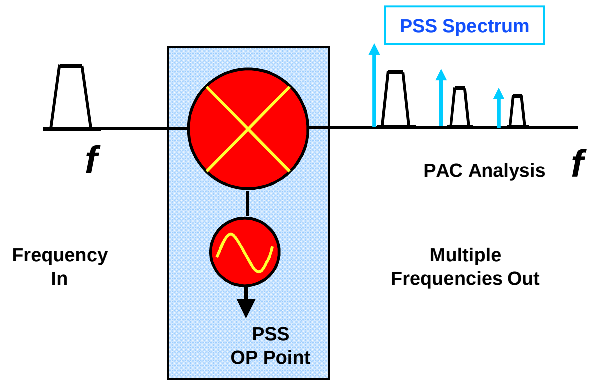
Linear Periodically Time Varying Circuits

Module 6

August 4, 2010

Linear Periodically Time Varying Circuits





DC analysis

Nonlinear large signal periodic state analyses
- **Shooting Method (time-domain)**
- **Harmonic Balance (frequency-domain)**

Linear Time Invariant (LTI) analyses
- **AC**
- **XF**
- **SP**
- **Noise**

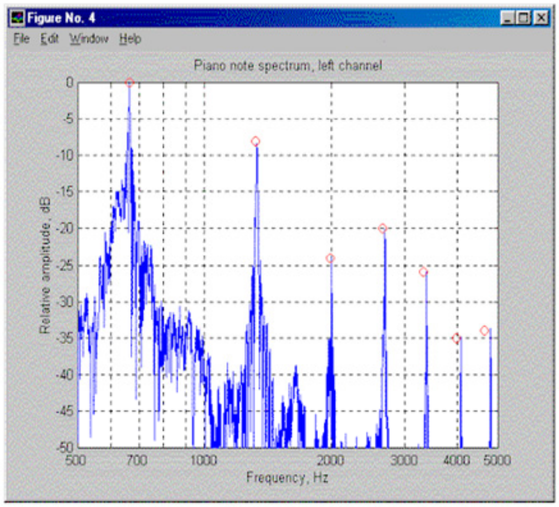
Linear Periodic Time Varying (LPTV) analyses
- **Periodic AC**
- **Periodic XF**
- **Periodic SP**
- **Periodic Noise**

When you apply a small sinusoid to a linear time-invariant (LTI) circuit, the steady-state response is a sinusoid at the same frequency. However, when you apply a small sinusoid to a linear circuit that is periodically time-varying (LPTV), the circuit responds with sinusoids at many frequencies.

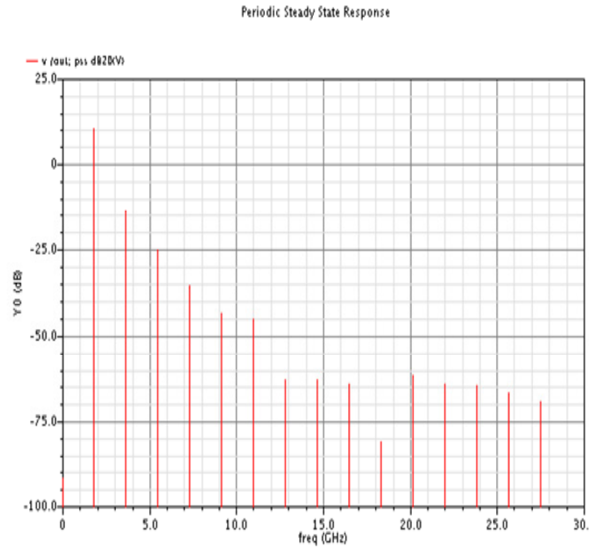
Linear Periodic time varying circuits, where you can obtain multiple frequencies (frequency translation) at the output to a single frequency input, are analyzed by using Periodic AC (PAC), Periodic XF (PXF), Periodic SP (PSP) and Periodic Noise analyses.

LPTV analyses are carried out on mixers, switched-capacitor filters, samplers, low noise amplifiers, sample-and-holds, and similar circuits.

Example of Linear Analysis Over PSS



Spectrum of 662 Hertz Piano Note



A 662 Hertz note played on a piano sounds different from the same note played on a trumpet or guitar due to timbre or harmonic content of the note.

The PSS simulation captures the nonlinearity of the circuit due to a nonlinear drive signal, which is similar to capturing the “note” with all of its timbre.

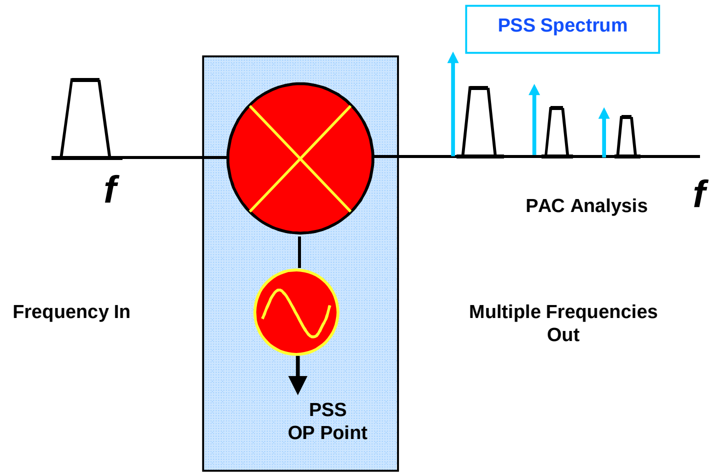
A note is a “thing” you can capture just like a PSS state is a “thing” you can capture as an operating point, and then you can run linear analysis over the top of it.

The timbre is the sound quality of a note.

PAC: Periodic AC Analysis

Small-signal operational flow:

- Virtuoso Spectre RF runs a PSS simulation and computes the periodically time-varying operating point.
- After PSS finishes, the PAC analysis runs.
- The PSS Direct Plot form displays the small-signal analyses results.



- The circuit responds with sinusoids at many frequencies (i.e., the circuit produces mixing products).
- With PAC analysis, there are many transfer functions between any single input and output.
- There are in fact as many transfer functions as harmonics in the periodic operating point (zero, one, or an infinite number).
- The PAC analysis calculates the signals (sidebands) in multiples of the PSS Fundamental.

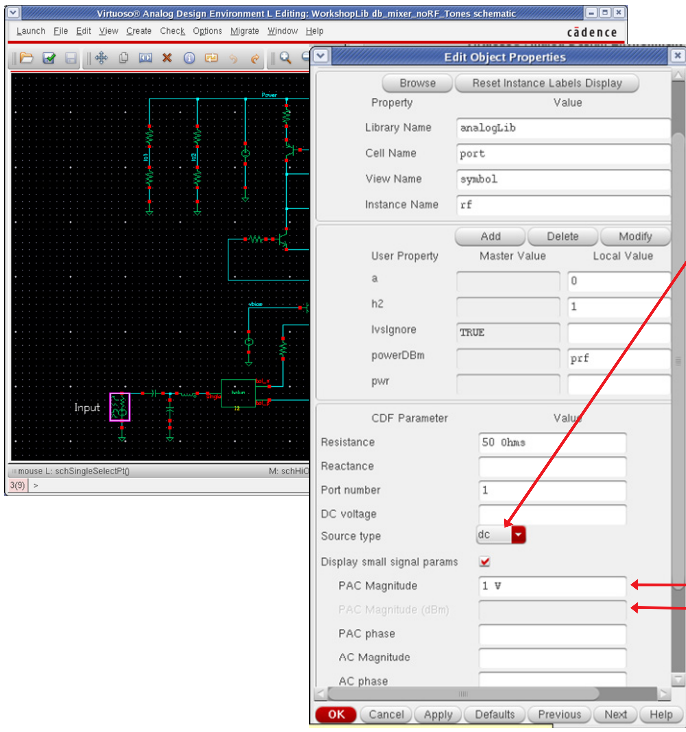
PAC analysis is a small-signal analysis like AC analysis, except the circuit is first small-signalized about a periodically varying operating point as opposed to a simple DC operating point.

Small-signal analysis about a periodically time-varying operating point allows transfer-functions that include frequency translation created by the nonlinearities of the largest signal in the circuit, whereas simply linearizing about a DC operating point does not because linear time-invariant circuits do not exhibit frequency translation.

A PAC analysis cannot be used alone. It must follow a large-signal PSS analysis.

PAC can be used on mixers, switched-capacitor filters, samplers, lower noise amplifier, sample-and-holds and similar circuits.

Mixer PAC Conversion Gain Example



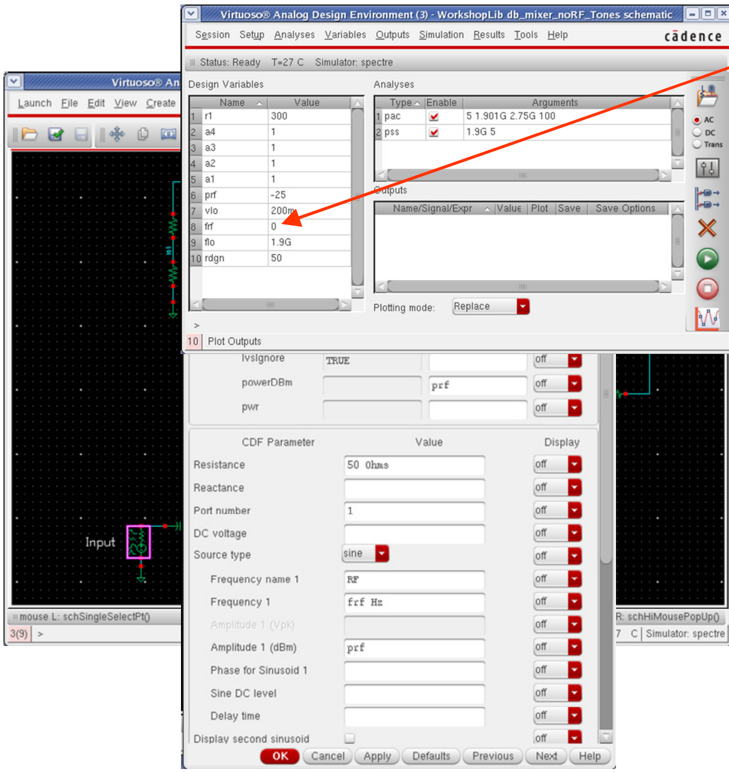
- Setting the Input Port “Source Type” to dcremoves the source from being calculated in the Least Common Frequency.
- The PAC power is still applied at the source with the PAC Magnitude property.
- The PAC Frequency is set in the PAC simulation form when you Choose Analysis and select PAC.

You can set PAC Magnitude in Volts like XF, or you can set PAC Magnitude in dBm. You have to subtract output dBm from input dBm to get the gain.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 143

- To run a PAC analysis to determine small-signal gain, you need to set the PAC magnitude in the input source.
- For conversion gain measurements, set the PAC magnitude to 1 volt. For measuring IP3, set the magnitude to dBm using a variable that can be swept in amplitude in ADE.
- You can combine a PSS analysis with a PAC analysis to determine the conversion gain of any up/down converter.

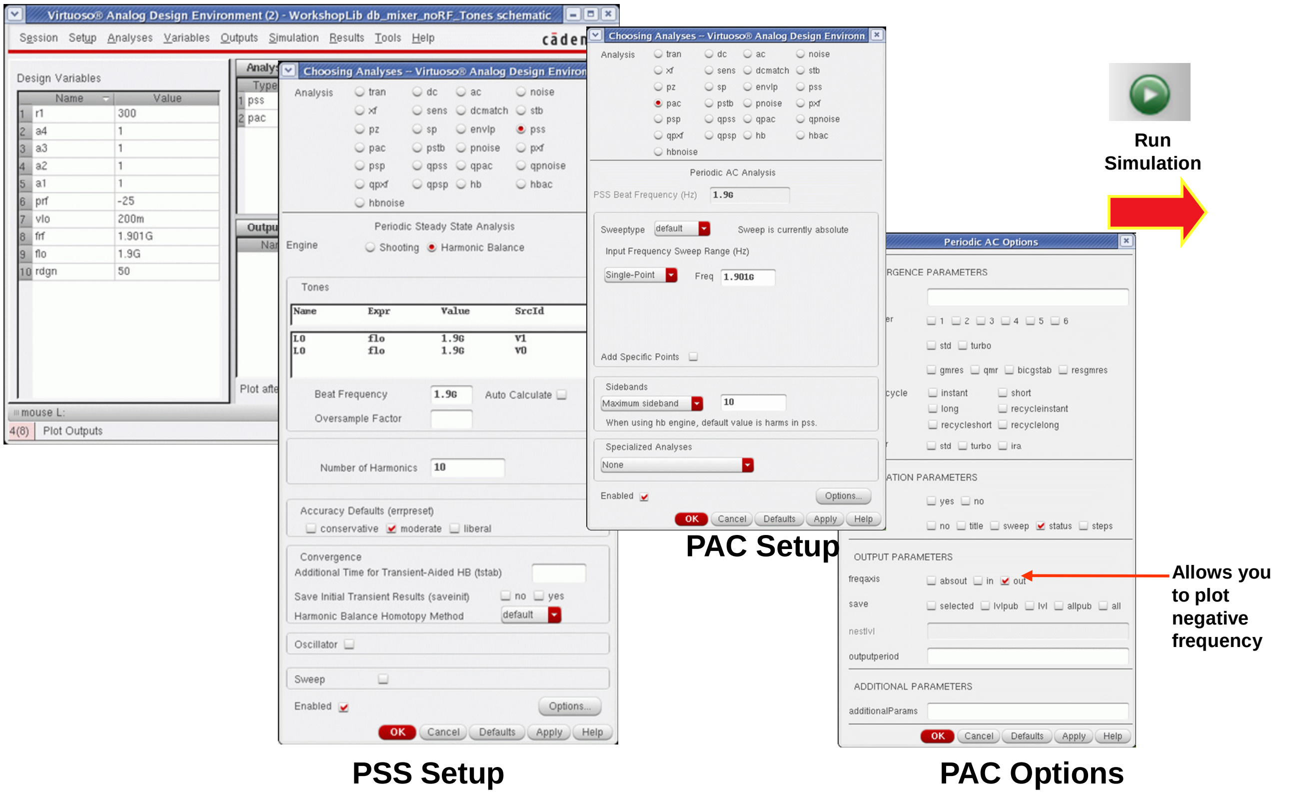
Mixer PAC Conversion Gain Example (Trick)



- Set the Input Port “Source Type” to sine and set **Frequency 1** in the **Edit Object Properties** form to a variable (*frf* as shown in the figure).
- The PAC power is still applied at the source with PAC Magnitude.
- Setting the *frf* variable to “0” in ADE is the same as setting the source to DC, but now you *don’t have to edit the schematic and check-n-save.*
- The PAC Frequency is still set in the PAC simulation form when you Choose Analysis and select PAC.

Instead of setting the Input Source type to *dc*, you can leave the *Source type* set to *sine*, and disable the waveform by setting the frequency to zero.

Mixer PAC Conversion Gain Setup



8/4/10 RF Analysis with Virtuoso Spectre Simulator 145

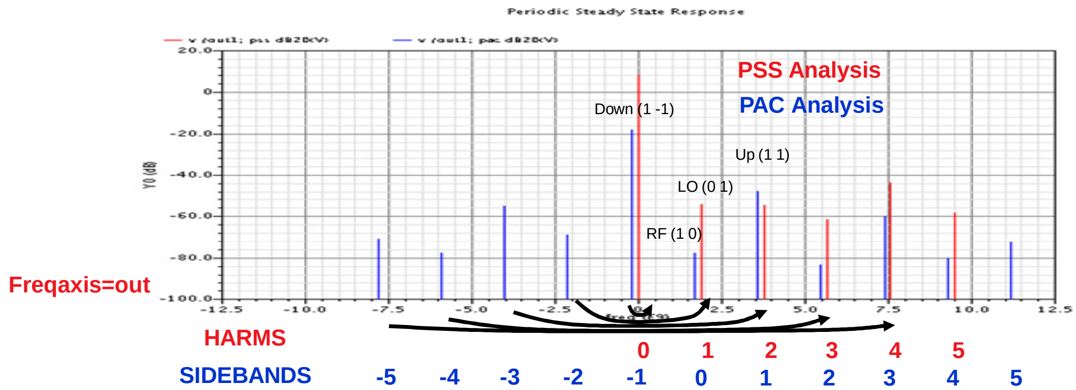
Computing the small-signal response of a periodically varying circuit is a two-step process:

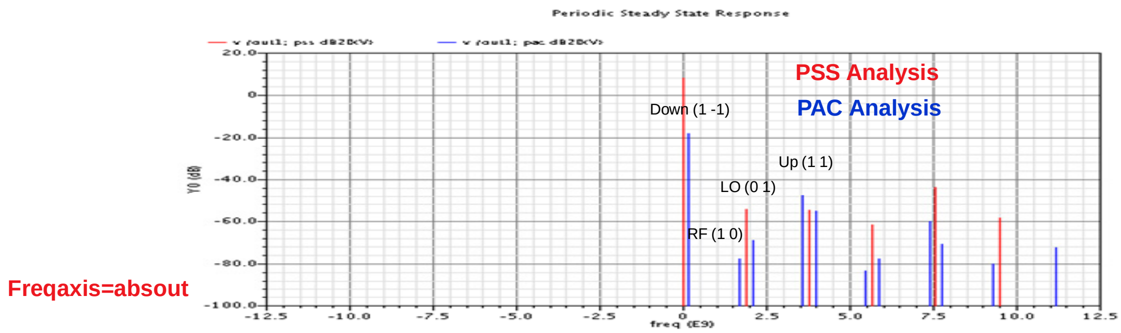
- First, the small stimulus is ignored and the periodic steady-state response of the circuit to possibly large periodic stimulus is computed using PSS analysis. As a normal part of the PSS analysis, the periodically time-varying representation of the circuit (Shooting) or the harmonic content (HB) is computed and saved for later use.
- The second step is carried out using the PSS Analysis where the small stimulus is applied to the periodically varying small-signal representation to compute the small signal response.

When PAC analysis is chosen, you can click the **Options** button and set additional output parameters. You can set the *freqaxis* to plot just the absolute frequency or negative sidebands also if required. The *freqaxis* parameter determines whether the results are output versus the input frequency, the output frequency, or the absolute value of the output frequency.

Once the setup is finished, click the **Netlist and Run** button to start the simulation. The results of the simulation are on the next page.

Mixer_PAC_Conversion Gain (Single Tone)





Setting the PAC option *freqaxis* to *out* shows the negative frequency terms that are produced.

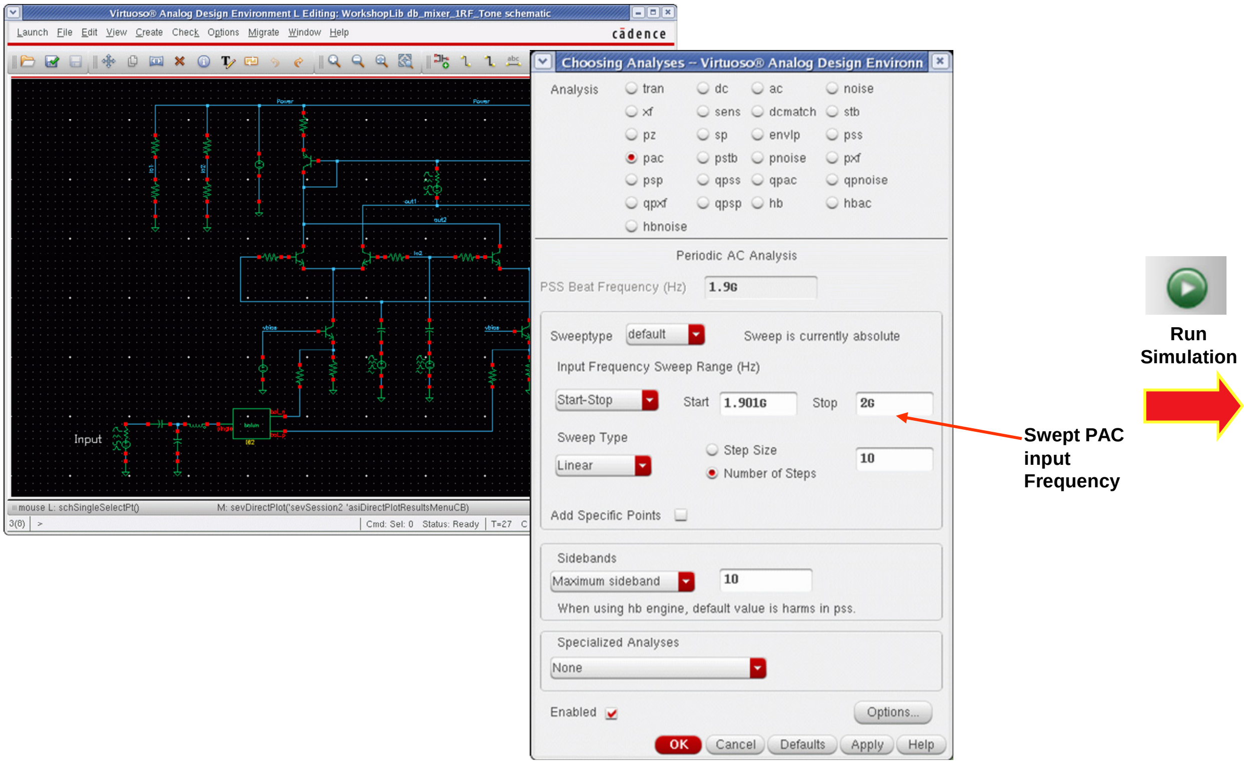
Usually, we think of the frequencies being positive, and when *freqaxis* is set to *absolut*, the simulator takes all the negative frequency terms and shows them on the positive frequency axis.

This page shows the PSS and PAC results for the mixer displayed with an absolute set of frequency values and also with the negative frequency sidebands. A single frequency tone of 1.901 GHz and 10 maximum sidebands for display are chosen for this simulation.

The Spectre RF simulation labels the transfer functions with the offsets from the input signal in multiples of the PSS beat frequency. These same labels identify the corresponding sidebands of the output signals. The labels are used as follows:

- 0 For circuits in which the input and output are at the same frequency, such as switched-capacitor filters, and sampling circuits the transfer function or sideband is set to 0. (zero)
- -1 For down-conversion mixers, the transfer function or sideband is labeled -1 because the output frequency is offset from the input frequency by -1 times the LO frequency.
- +1 For up-conversion mixers, the transfer function or sideband is labeled +1.

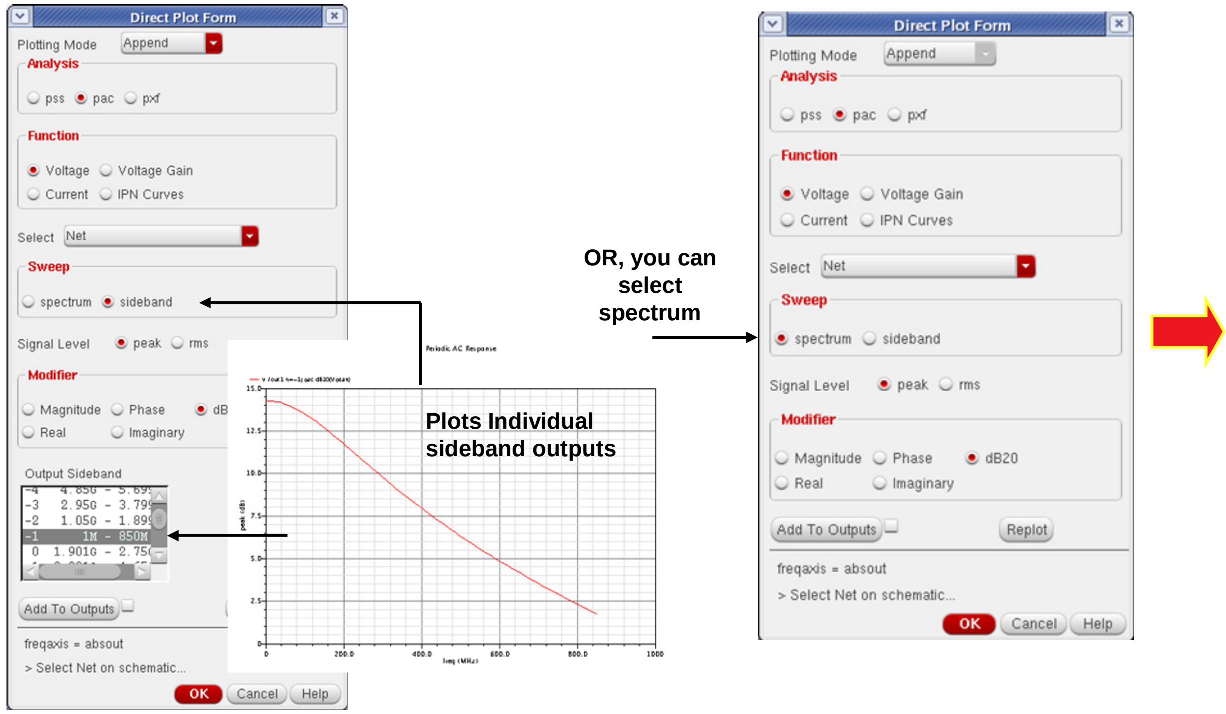
Mixer_PAC_Conversion Gain (Swept-Tone)



PAC Setup

- PAC computes a series of transfer functions, one for mixing with each PSS harmonic.
- Instead of using a single tone as in the previous slide, you can specify sweep limits by providing either the end points or the center value and the span of the sweep. Remember to type the exact frequency magnitude as the setup default unit is in Hz.
- Maximum sideband controls how many PAC mixing products calculated. Remember these mixing products are created by mixing with the harmonics in the PSS.

Mixer PAC Direct Plot (Swept-Tone)



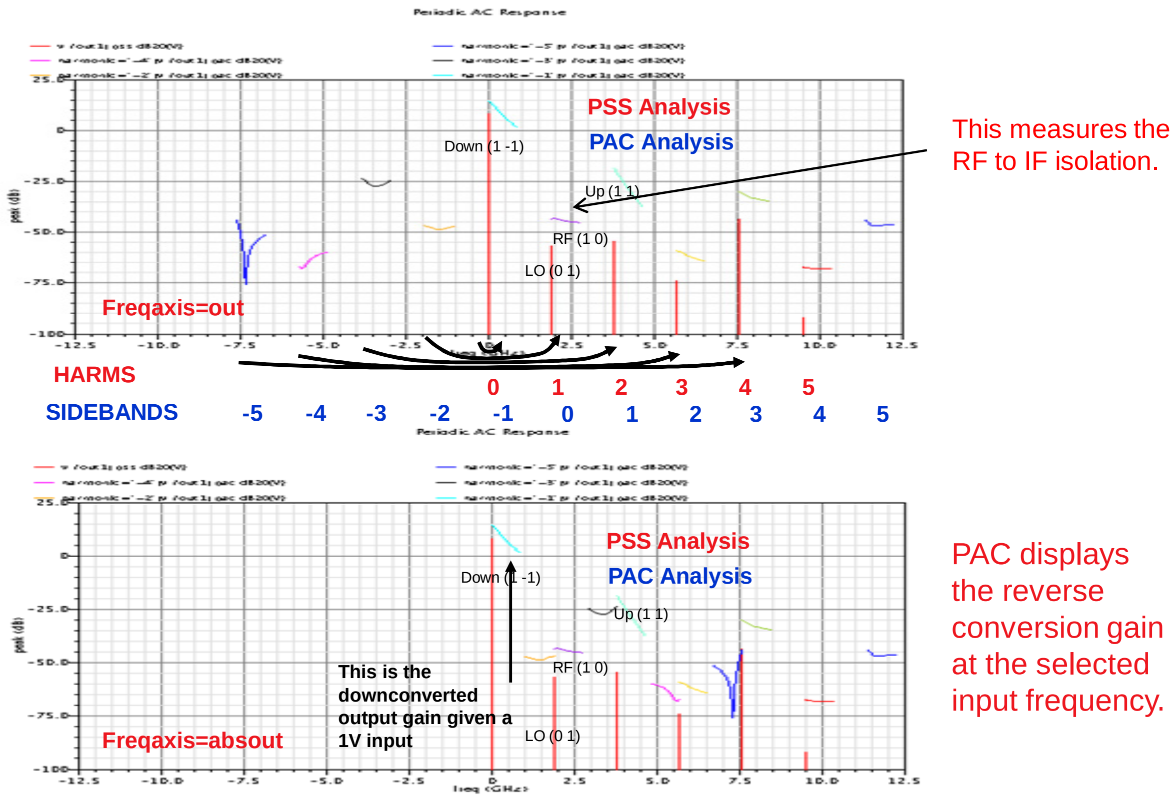
When the sweep simulation is carried out by pressing the *Netlist and Run* button, you can use the *Results–Direct Plot–Main Form* from ADE to view the waveform(spectrum).

This shows a single output transfer function by plotting one sideband.

Remember that the input frequency is swept, so the mixing product also sweeps in frequency.

Choose the desired output frequency range in the *direct plot form*.

Mixer_PAC_Conversion Outputs (Swept-Tone)

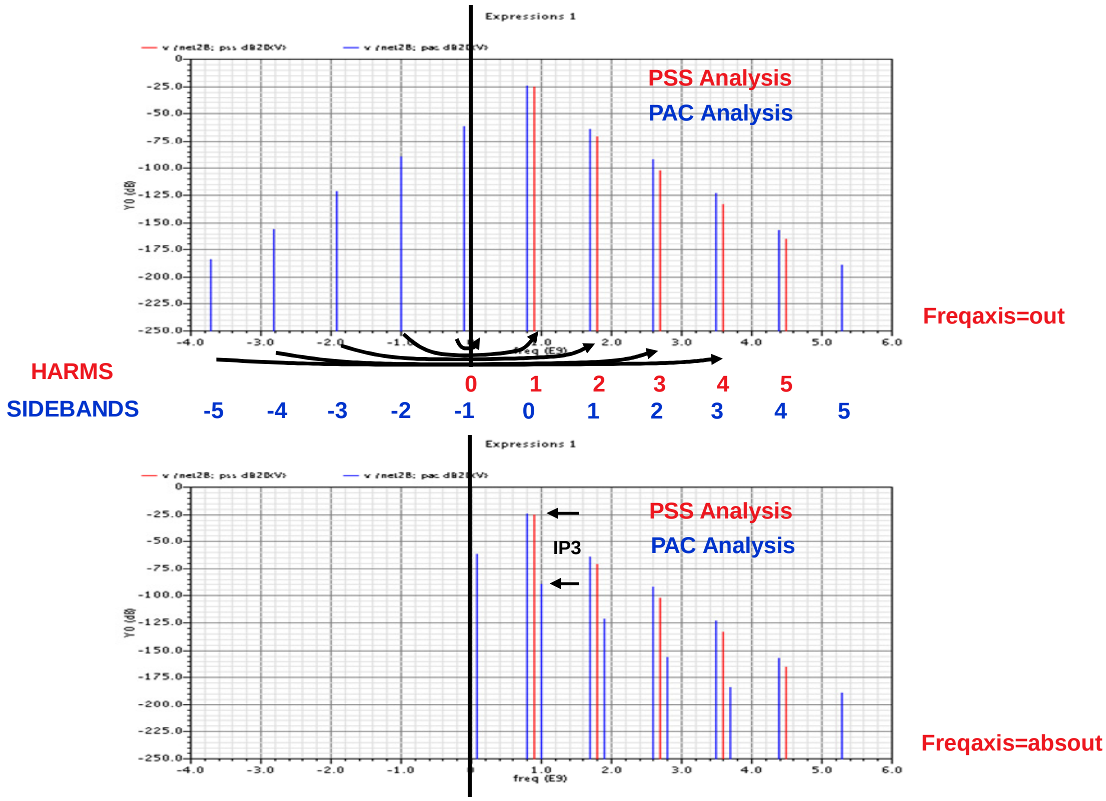


8/4/10 RF Analysis with Virtuoso Spectre Simulator 149

When there are multiple harmonics in the PSS to mix with, there are multiple output mixing products that are created.

- The input frequency is specified in the PAC choose analysis form.
- To measure the conversion gain, go to the desired output frequency and place a marker or use the tracking cursor.
- Because the input was 1 volt, the gain is measured directly.
- When *freqaxis* is set to *out*, the negative frequency axis is shown. When *freqaxis* is set to *absolut*, the negative frequency terms are reflected up into the positive frequency axis.

LNA: Using PAC to Calculate IP3



8/4/10 RF Analysis with Virtuoso Spectre Simulator 150

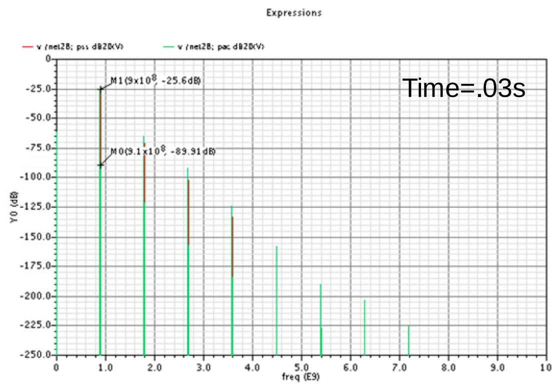
The PSS analysis followed by a PAC analysis can measure the intermodulation distortion of amplifiers. You can combine a swept PSS analysis with a PAC analysis to produce data for an IP3 plot. For amplifiers, rapid IP3 is faster and just as accurate. Accordingly, it is suggested.

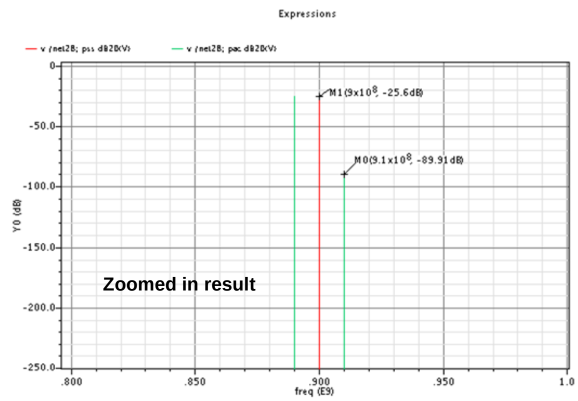
This page shows the simulation result with differing *freqaxis* settings to display the output spectrum of a LNA.

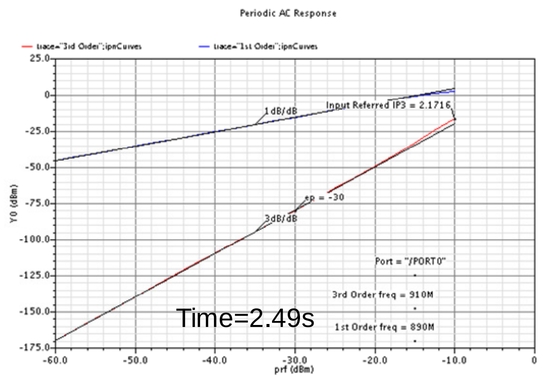
After the PSS analysis computes the circuit response to one large tone, then the PAC analysis applies the second tone close to the first. If you consider the small input signal to be one sideband of the large input signal, then the response at the other sideband is the third-order intermodulation distortion, as shown in the absolute frequency axis spectrum. This result obtained in dB at the output is used to mathematically determine the IP3 of the LNA.

Note that only the third-order distortion term on the other side of the PSS frequency from the input tone in PAC is calculated. The other third-order term is not calculated with PAC. As a strong suggestion, use the AC rapid IP3 form where any of the frequencies can be chosen.

PAC/PSS LNA IP3 Example







8/4/10 RF Analysis with Virtuoso Spectre Simulator 151

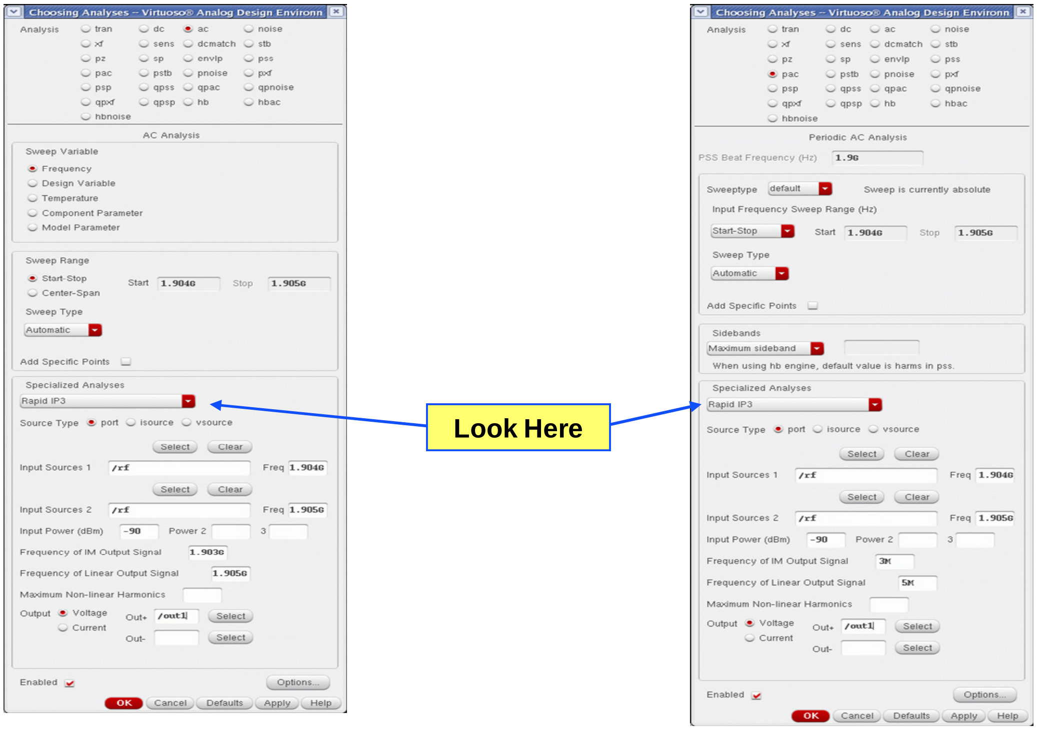
You can use the *Results–Direct Plot–Main form*– **Ipn Curves** option to plot the Input or output Referred IP3 using the 1st and 3rd harmonic.

PAC/PSS is used to normally calculate IP in amplifiers. Again remember rapid IP3 in the AC choose analysis form.

Rapid IP3 Analysis Forms

The option is under Specialized Analyses Section of AC analysis, which also has options for IP3, IP2, and distortion summary for amplifiers.

The option is under Specialized Analyses Section of PAC analysis also has options for IP3, IP2 and distortion summary for Mixers



The *rapid IP3* measurement computes the nonlinear response to RF signals around the periodically time-varying operating point. This IP3 measurement is much faster than the IP3 measurement procedure described with swept PSS and PAC.

- This option is available under the *Specialized Analyses* section.
- For amplifiers, use the AC choose analysis form. For mixers, apply the LO only in PSS, and select rapid IP3 in the PAC form.

Perturbation-Based PAC Method to Calculate Rapid IP3

The perturbation-based method solves weakly nonlinear circuit based on the mathematical **Born approximation**.

- It directly computes the third order intercept point without power sweep
- The nonlinear circuit equation can be expressed as:

$$L \cdot v + F_{NL}(v) = \varepsilon \cdot s$$

- Under weakly nonlinear condition, nonlinear part is small compared to the linear part, so equation can be solved by using Born approximation iteratively:

$$v^n = v^1 - L^{-1} \cdot F_{NL}(v^{n-1})$$

- The method does not require explicit high order device derivatives, it is equivalent to successive small signal analysis.

Rapid IP2 and IP3 measurements are calculated with 2nd and 3rd order *Born approximation* under weakly nonlinear conditions.

All equations are formulated as RF harmonics and they can be implemented in both the time and frequency domains.

Spectre RF supplies four specialized PAC and AC analyses based on perturbation methods that provide a basic set of rapid distortion measurements.

- Compression Distortion Summary
- Rapid IP3
- IM2 Distortion Summary
- Rapid IP2

These four perturbation-based analyses characterize intermodulation distortion and compression distortion for RF circuits such as mixers and amplifiers.

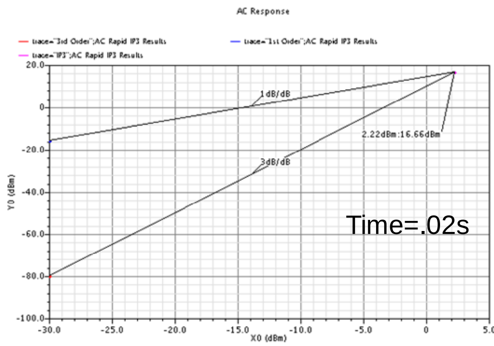
Comparison of IP3 Methods

- Mixer (periodic operating point):
  - Three tone large signal QPSS
  - Two tone large signal QPSS + QPAC
  - One tone large signal PSS + PAC Rapid IP3

Faster ↓

- LNA (stationary operating point):
  - Two tone large signal QPSS
  - One tone large signal PSS + PAC
  - AC Rapid IP3

Faster ↓



Rapid IP3 of LNA

The IP3 evaluation methods for both the mixer and LNA circuits are summarized above.

The speed of evaluation is listed in a bottom-up order with the Perturbation methods being the fastest.

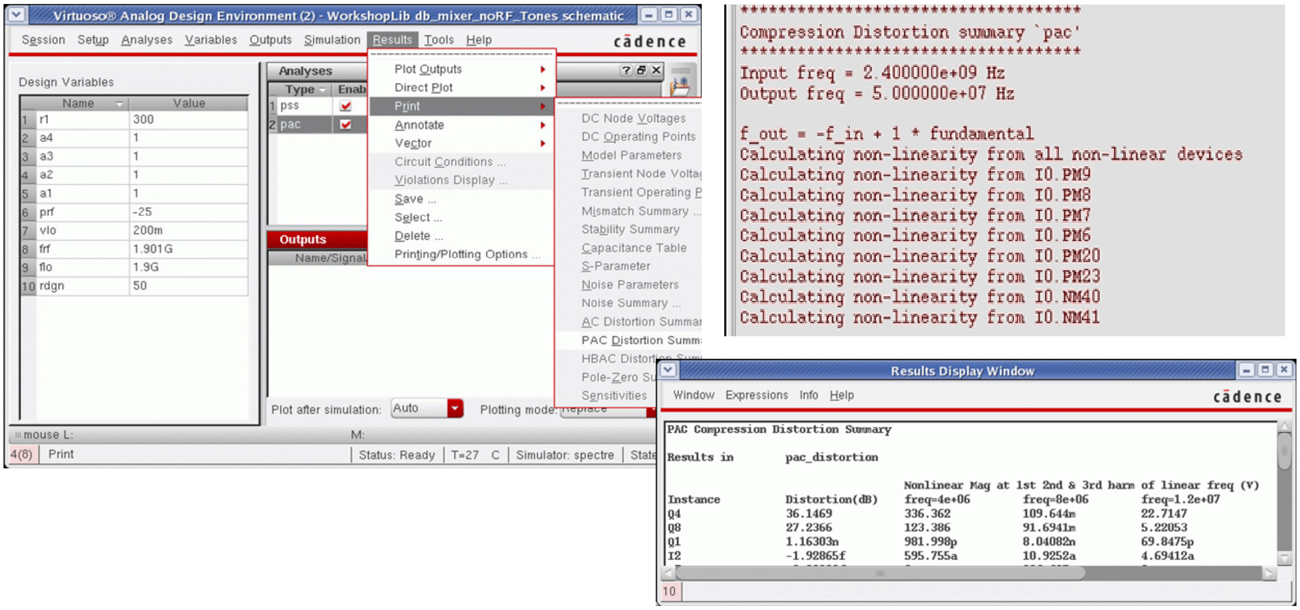
Tightening Tolerances to Measure Rapid IP3

Cannot get Rapid IPx to agree with QPSS/QPAC.

- May have to do with not setting tolerance tight enough.
- The dynamic range of signals of interests may give a clue on how the tolerance should be set.
- For example, typically, *reltol = 1e-4* will give numerical dynamic range of 80dB (about *20*log(reltol*))
- If IM products are very small (for example -100dBm), tight tolerance such as *reltol=1e-5* may be needed to give enough numerical dynamic range for accurate IPx calculation.
- HB usually gives larger dynamic range than shooting for the same tolerance.

To get a match between large-signal IP3 measurement (QPSS, or hb multitone) tighter convergence options might be needed. Especially if shooting is selected. If shooting is selected, try reltol=1e-5 and vabstol=3e-8.

Rapid IP3 or IP2 Distortion Summary



- Calculates the IM Distortion from each device and gives you the results in table form.
- Very similar to Noise Summary Tables.

Make sure the *pac* magnitude property in the source is set in the small-signal range.

The gain reading is the gain with the nonlinearity of the device divided by ideal PAC gain.

The Compression Distortion Summary is available as a part of the *Specialized Analyses* when you choose PAC.

You can combine a PSS and PAC analyses to measure and print the compression distortion summary for selected components.

PAC analysis calculates the distortion contributed by each selected component in the schematic. The distortion summary is a quantitative measurement of how much distortion a device adds to the output signal.

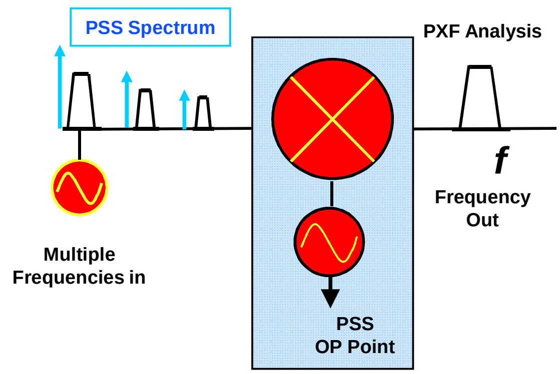
In ADE, choose **Results – Print – PAC Distortion Summary** to view the results in a tabular format. The summary lists the various distortion contributors and how much distortion each contributes to the output.

When no devices are selected, all the nonlinear devices are calculated. Because PAC needs to be run for each device specified, computation is time-consuming if the device list is large. Select only the most important devices to the output.

PXF: Periodic XF Analysis

Periodic Transfer Function (PXF) Analysis:

- Is a frequency-translated transfer function analysis.
- Is a small-signal analysis that can be swept in frequency.
- Simulates circuits based on a periodically varying operating point.
- Calculates multiple input frequencies for a given, single (or swept) output frequency.
- Computes the transfer function from any source at any frequency to one output at one frequency (or range of frequencies).
- Is run after the PSS analysis.



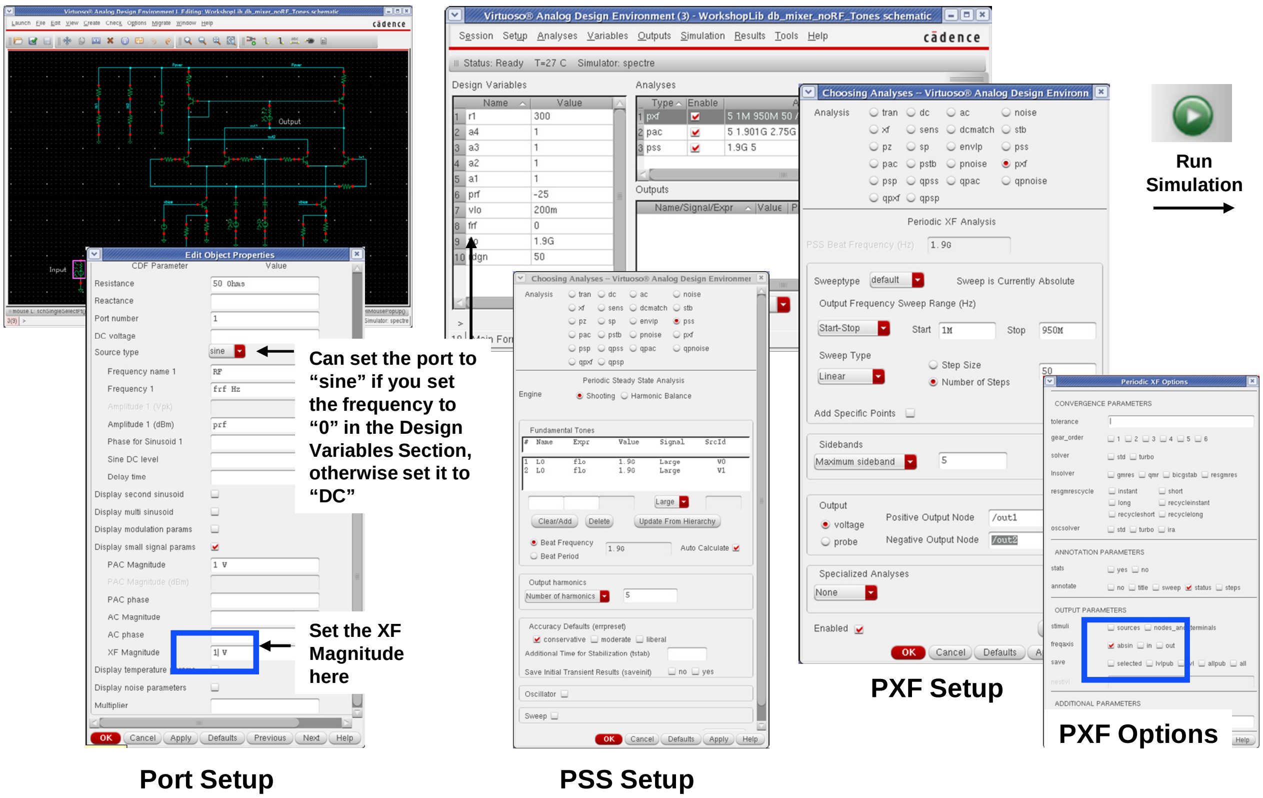
8/4/10 RF Analysis with Virtuoso Spectre Simulator 157

A PXF analysis cannot be used alone, it must follow a PSS analysis.

PXF Applications

- Determines the conversion gain (efficiency) of a circuit that is the transfer function from input to output at a desired frequency.
- Determines the image and sideband rejection (spurs in a receiver design at an undesired frequency).
- Determines the power supply rejection and LO feed-through (measure of frequency translated gain from the power supply to the output at all frequencies).

PXF: Simulation Setup



8/4/10 RF Analysis with Virtuoso Spectre Simulator 158

Computing transfer functions for a periodically varying circuit is a two-step process.

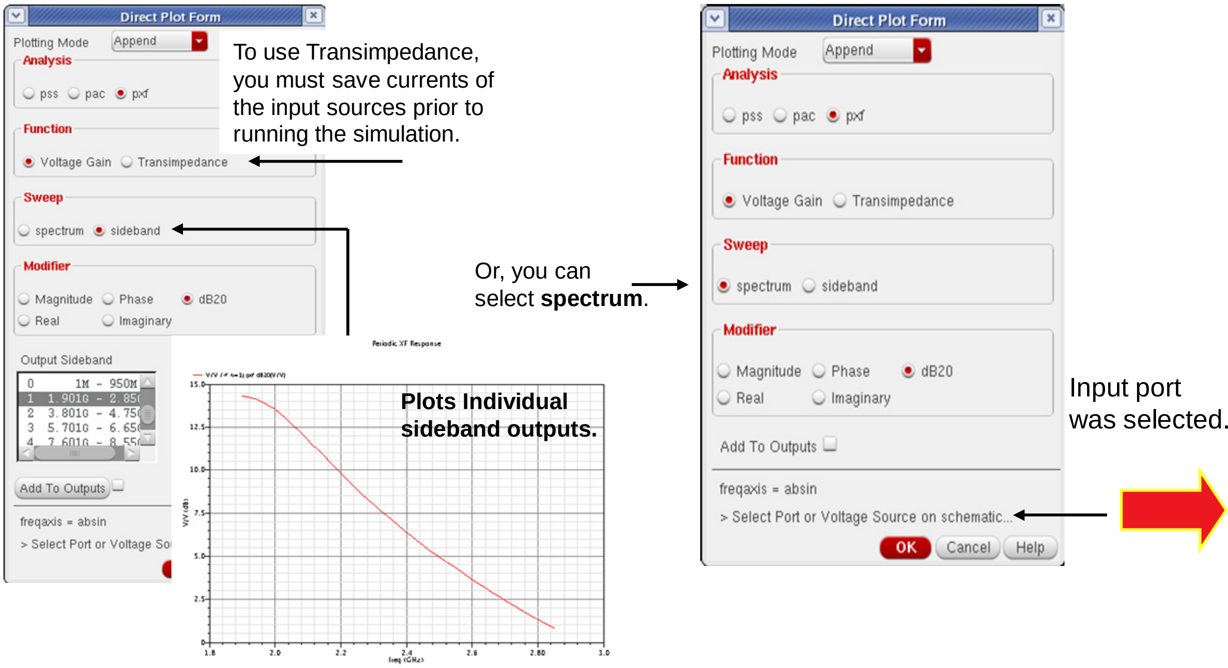
- First, the small stimulus is ignored and the PSS response of the circuit to possibly large periodic stimulus is computed using PSS analysis. As a normal part of the PSS analysis, the periodically time-varying representation of the circuit is computed and saved for later use.
- Second, using the PXF analysis, small-signal excitations are applied to compute the transfer functions.

Here you see the input port setup made in the *Edit Object Properties* form. If using Design Variables in ADE, you can use a sine wave source if you set its frequency to zero. If not, use a DC source. You can set the transfer function magnitude in this form.

After this, you have to complete the PSS setup and then set up the PXF analysis. You can select the output to be measured. The output variable you measure can be voltage or current, and the variable frequency is not limited by the period of the large-periodic solution.

Clicking the *Options* button in the PXF analysis form gives you access to set the convergence and output parameters that specify whether or not the negative frequency axis is shown.

PXF: Direct Plot

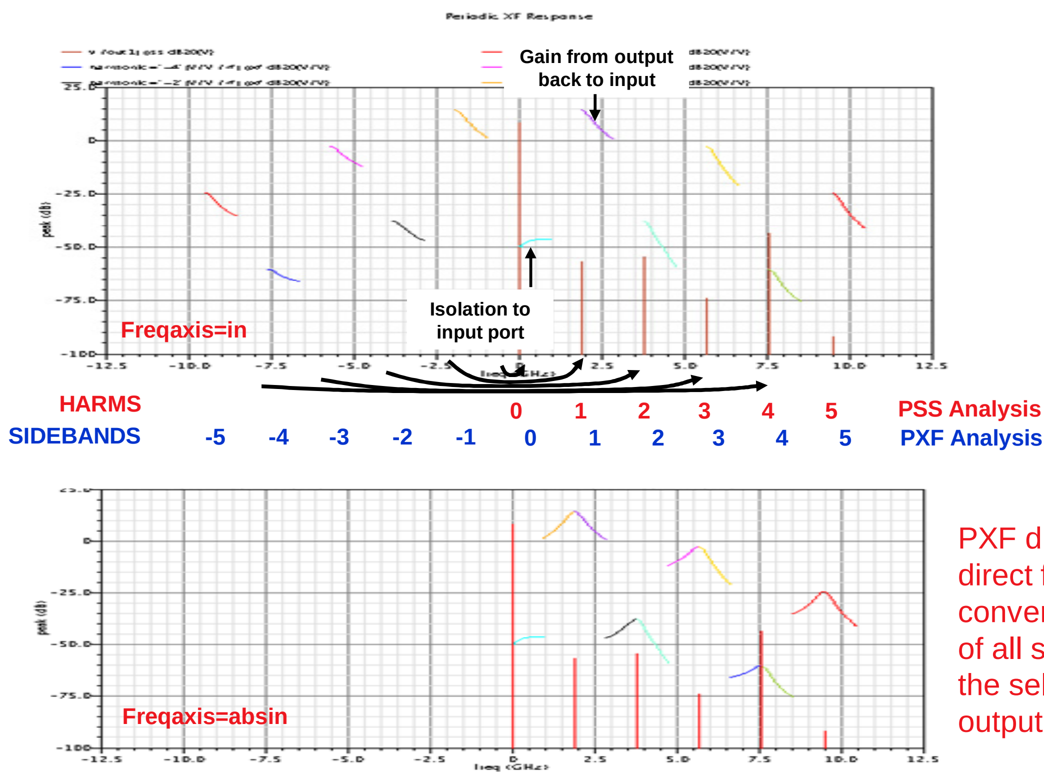


This form prompts for the selection of a port or a voltage source on the schematic, so do not attempt to select a net in the schematic — it will not work.

When the PXF Analysis is set up using the *Choose Analyses* form in ADE and the simulation is run, you can plot the PXF using the *Direct Plot Form*.

- Based on your setup in the Direct Plot Form, you can choose to plot individual sidebands or the whole spectrum. You are prompted to choose the Port or Voltage source on the schematic for which to plot the chosen option.
- The gain from the selected source to the output is plotted.
- To measure conversion gain, go to the desired input frequency and place a marker or use the tracking cursor.

PXF: Periodic XF Analysis Results



PXF displays the direct forward conversion gain of all sources at the selected output frequency.

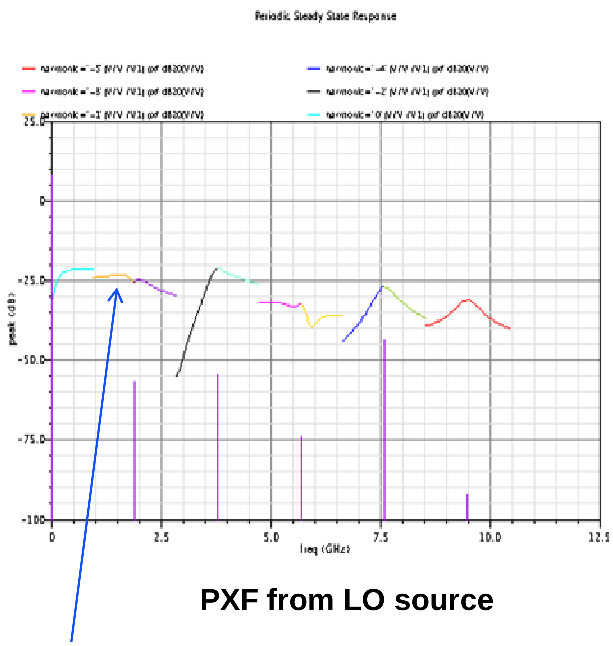
PXF analysis measures conversion gains, especially those from the input source to the output.

The PXF analysis also computes the conversion gain of the specified sideband as well as various unwanted images including the baseband feed through.

These are the sideband and spectrum sweep outputs of the PXF analysis with respect to the input port.

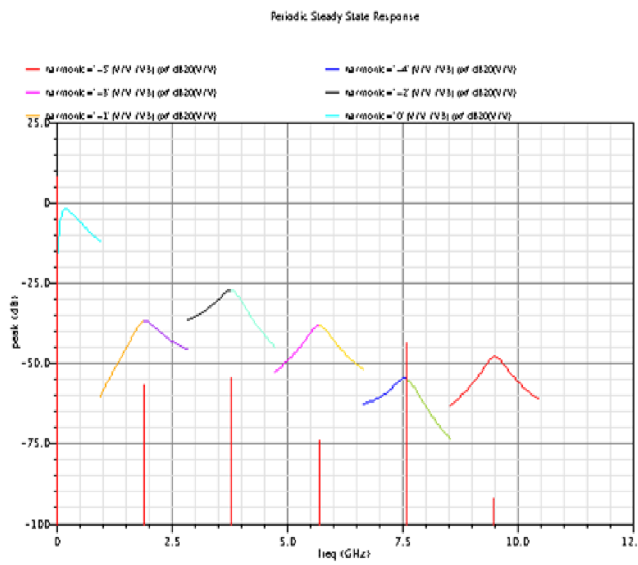
Choosing *freqaxis* = *in* plots the negative sidebands as a function of output versus the input frequency.

PXF: Periodic XF Analysis Results (continued)



PXF from LO source

Small-signal rejection is about 20dB



PXF from VSS source

No gains above unity. Pretty safe.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 161

PXF analysis calculates the gain from all the sources in the circuit to the output specified in the PXF form.

There is a real question about the use of PXF to calculate LO to IF gain.

- PXF is a small-signal and the LO is generally a large signal. Most designers use PSS to measure the LO leakage to the IF output of the mixer.
- If there is a path from the power supply to the output with greater than unity gain, there is a potential cause of oscillation.

PSP: Periodic S-Parameter Analysis

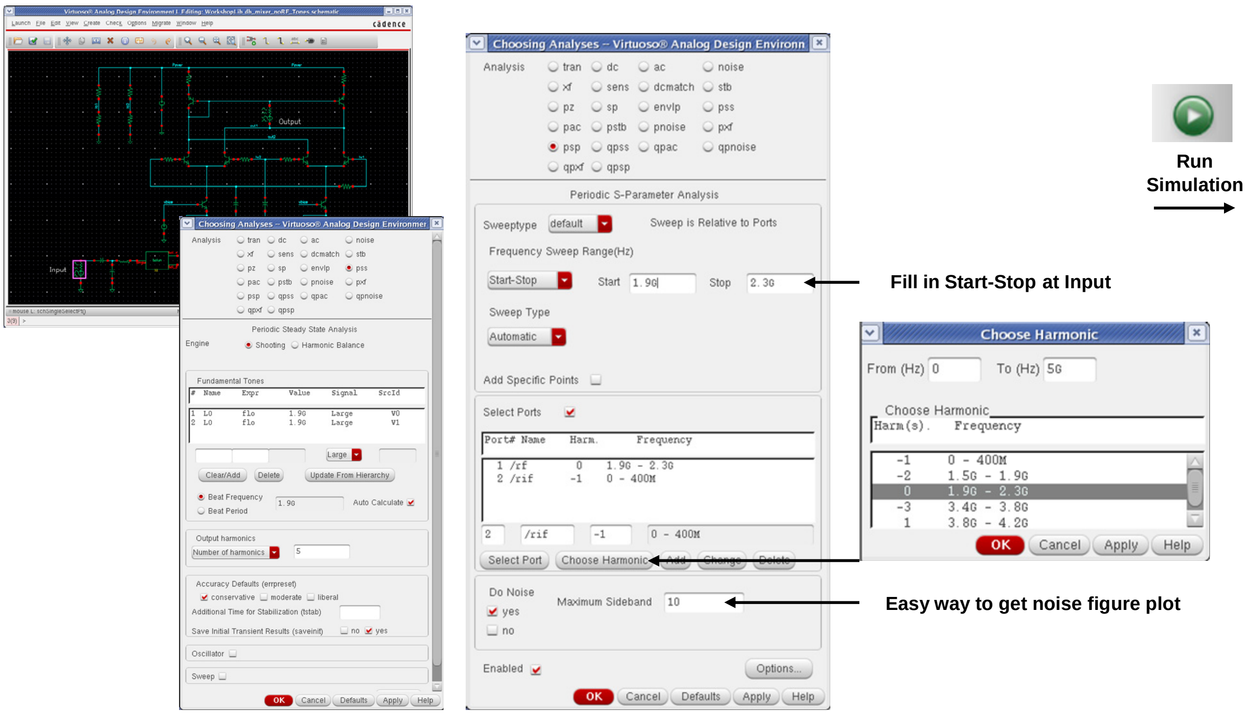
The Periodic S-Parameter (PSP) analysis is used to compute scattering and noise parameters for n-port circuits that exhibit frequency translation.

- The circuit is first linearized about a periodically time-varying operating point.
  - This allows the computation of S-parameters between circuit ports that convert signals from one frequency band to another.
- PSP analysis is used to calculate noise parameters in frequency-converting circuits.
  - The noise features of PSP analysis include noise folding effects due to the periodically time-varying nature of the circuit.
  - PSP can compute noise figure (both single-sideband and double-sideband), input referred noise, equivalent noise parameters, and noise correlation matrices.
- PSP analysis can be carried out on mixers, switched-capacitor filters, samplers and other similar circuits.

The frequency in the Choose Analysis form can be either input or output frequency. The actual frequencies used are determined by the setting on the harmonic field in the Choosing Analysis form.

- If the frequency is negative, the Smith Chart will be upside down. The input and output frequencies are listed in the Spectre output window.
- PSP measures power gain, not voltage gain
- PSP provides the same frequency on the plots for the input and the output. Using PSP is sometimes confusing and is recommended for experienced customers

PSP: Simulation Setup

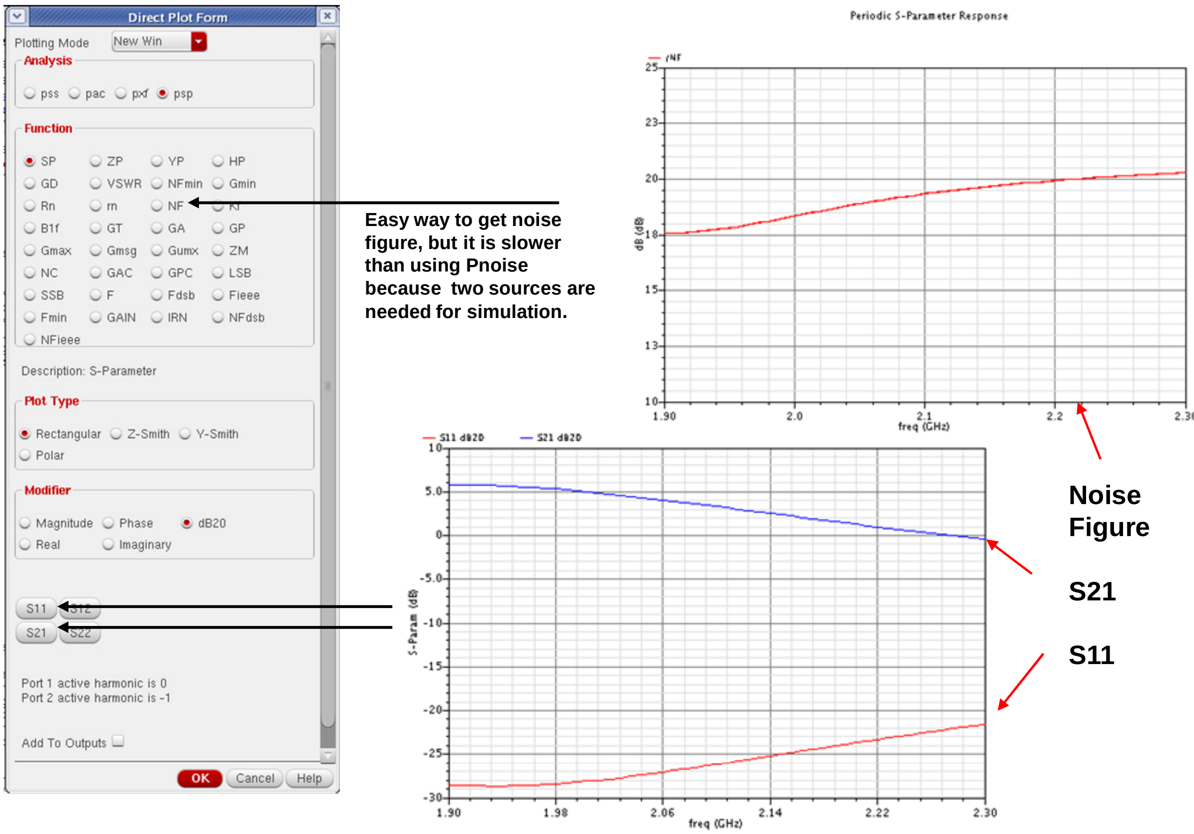


PSS Setup **PSP Setup**

PSP requires a port on the input and output.

- This is an example where the input frequency range is specified in the *Choose Analysis* form.
- In the Ports listing, always use the **Choose Harmonic** button.

PSP: Simulation Results



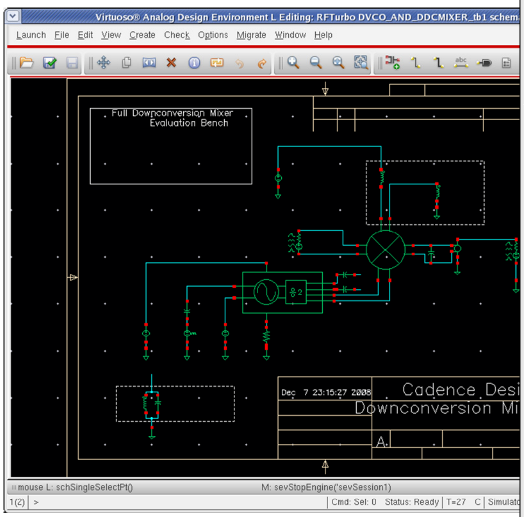
8/4/10 RF Analysis with Virtuoso Spectre Simulator 164

Here are the results of the PSP simulation. The results can be plotted by choosing the **Results – Direct Plot – Main Form**.

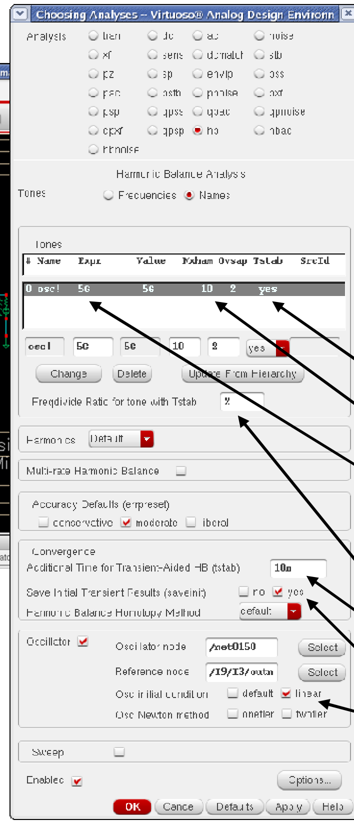
- You can see the rectangular plot (SP function) of the PSP response modified by a dB20 scale. You can plot any of the four s-parameters with the available buttons.
- You can also use the NF Function to plot the noise figure.

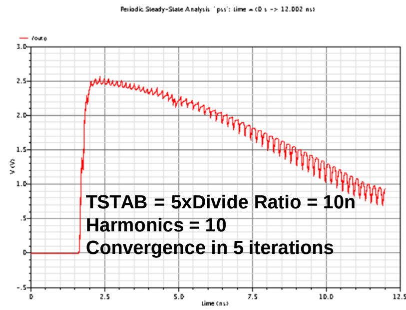
VCO+Divider with Mixer Example

PSS HB Oscillator Analysis



Setting up PSS for LPTV analysis.
At the end of the PSS, note the oscillation frequency.





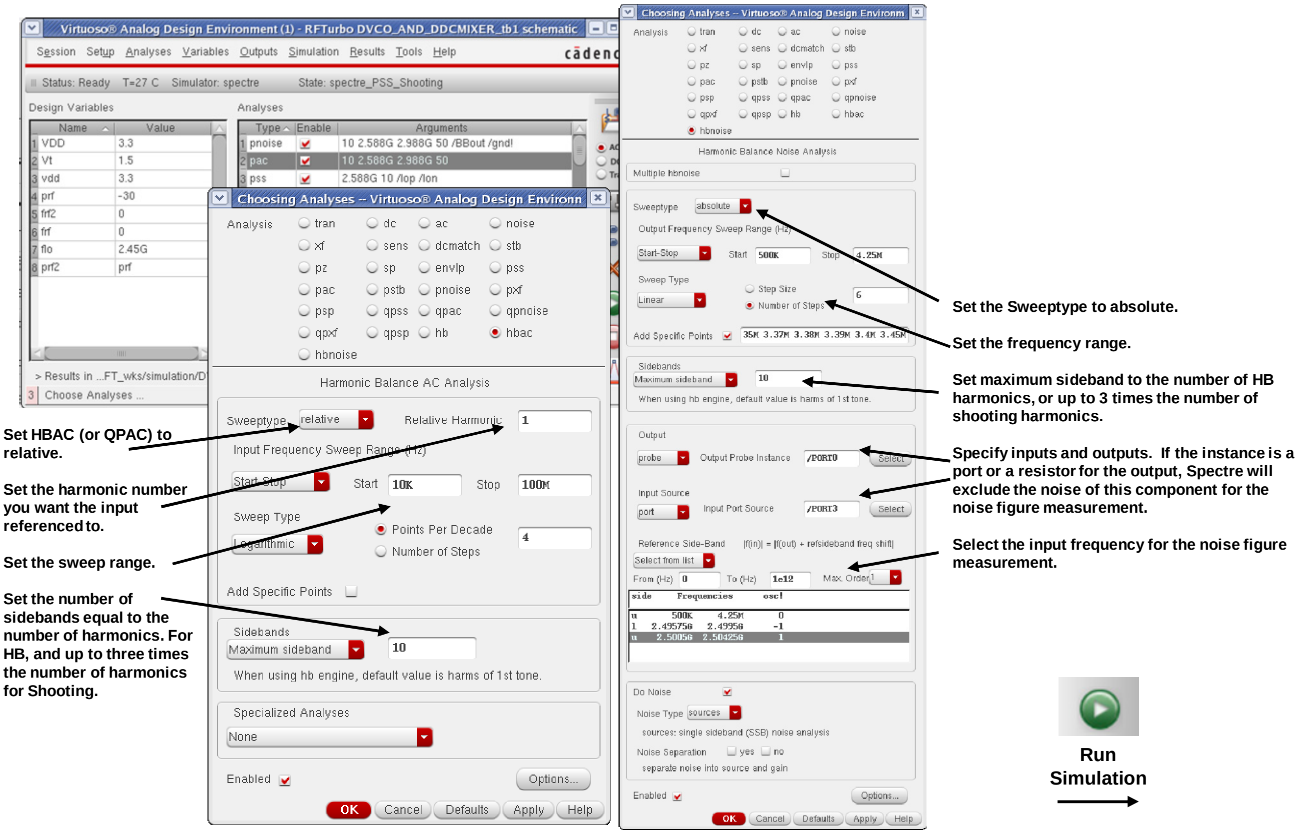
- Set Tstab to yes for the oscillator (Largest) tone.
- Set a reasonable number of harmonics.
- Set the oscillator frequency from 0.67 to 1.5 * actual oscillation frequency. The PSS algorithm will find frequency dividers if the divide ratio is 4 or less.
- If using hb, set the divider ratio in freqdivide
- Set tstab to at least one divider cycle.
- If you want to see the startup waveform, select yes for saveinit.
- Select the linear oscic to minimize oscillator startup time.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 165

The following few pages show how to set up and run the small signal PAC and PNOISE analysis after an initial PSS operating point has been calculated for the LPTV system. The DUT consists of a Mixer working in conjunction with a VCO and Divider.

- Set the oscillation frequency to a reasonable estimate of the actual oscillation frequency.
- The PSS algorithm finds frequency dividers as long as the divide ratio is 4 or less.
- If using HB, set the divider ratio in the *freqdivide* property.
- Set a reasonable number of harmonics. At least 5 harmonics of the input frequency need to be calculated.
- Setting the *linear oscic* causes the oscillator to start at the beginning of tstab.

VCO+Divider with Mixer Example (continued)



8/4/10 RF Analysis with Virtuoso Spectre Simulator 166

After the PSS operating point is determined, PAC and PNOISE analyses is now run on the DUT.

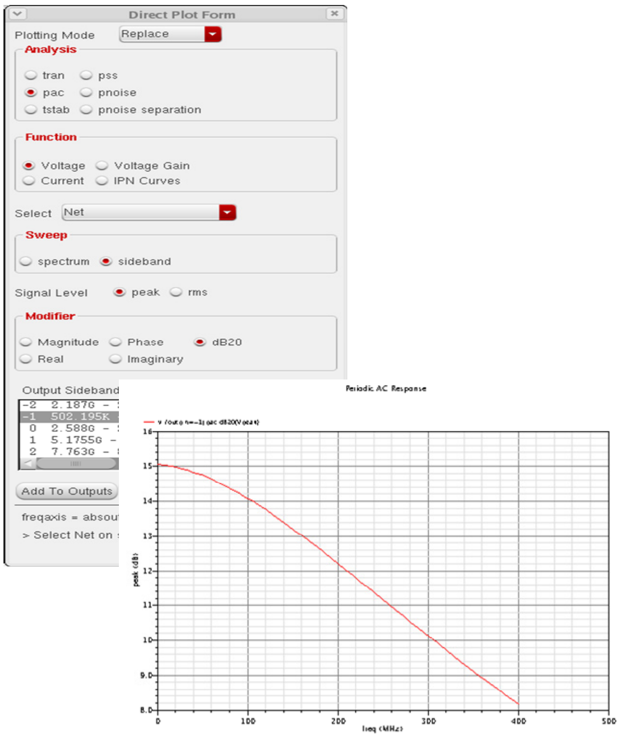
A swept PAC analysis is carried out as a relative sweep. This way you don't need to know the LO frequency beforehand.

Now to run Pnoise set up as usual. Use an absolute sweep.

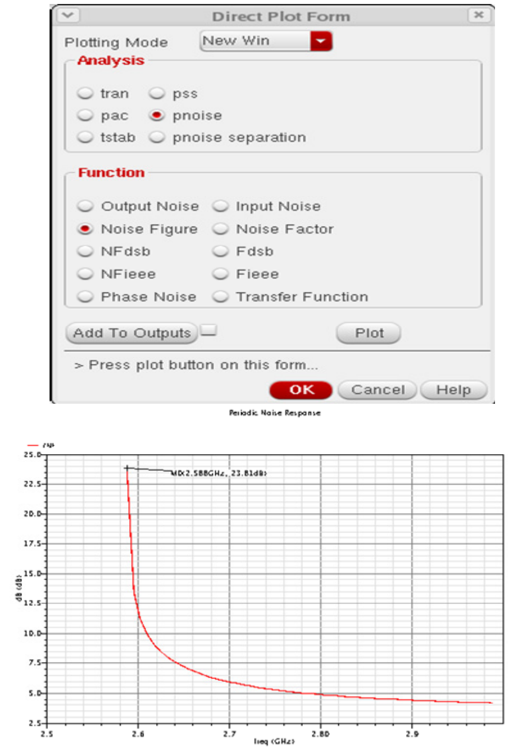
Setting the output instance to a port or a resistor causes the Spectre simulator to exclude the noise of the instance for the noise figure measurement.

The *Reference Side-Band* field specifies the desired input frequency.

VCO+Divider with Mixer Example



PAC Output



PNOISE Output

When the simulation is finished, you can view the results using the Direct Plot Form. The Conversion Gain and SSB Noise Figure is plotted.

PSTB: Periodic Stability Analysis

The periodic stability (PSTB) analysis evaluates the local stability of a periodically time-varying feedback circuit.

- First set up a PSS or HB analysis to characterize the system with the large signal present.
- The loop-based algorithm requires that you place the probe on the feedback loop to identify and characterize the particular loop of interest.
  - To perform the analysis place an *iprobe* (if the circuit is single-ended) or a cmdmprobe (if the circuit is differential) from analoglib in series with the feedback loop.
  - The summing junction is in the circuit and can't be separated out.
- The oscillation condition in feedback is unity gain and the phase condition is 0 or multiples of 360 degrees. This is because the summing junction is included with the circuit.

The loop gain of the feedback circuit can change when a large signal is applied to the circuit because of large-signal effects.

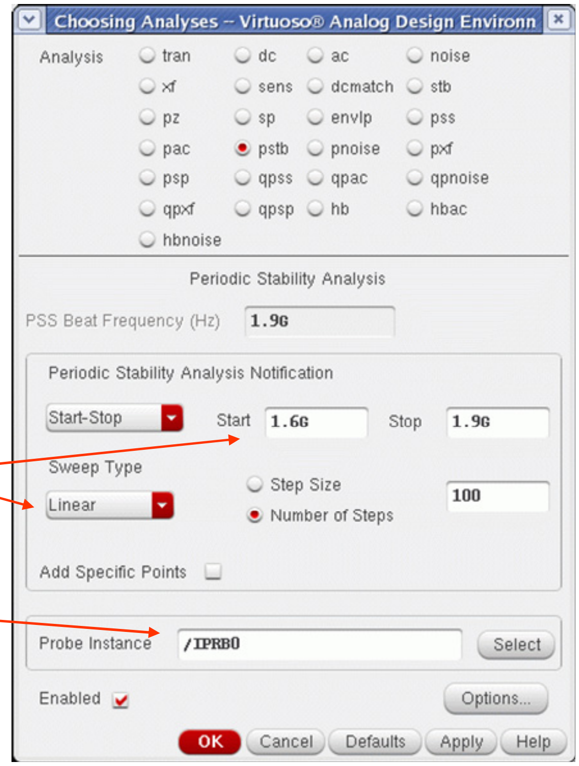
Use only the loop-based algorithm.

It provides stability information for both single loop circuits and for multiloop circuits in which you can place an *iprobe* component or a *cmdmprobe* component on a critical wire to break all loops.

PSTB: Simulation Setup

Enter the frequency range and spacing here.

Enter the iprobe or cmdmprobe here.



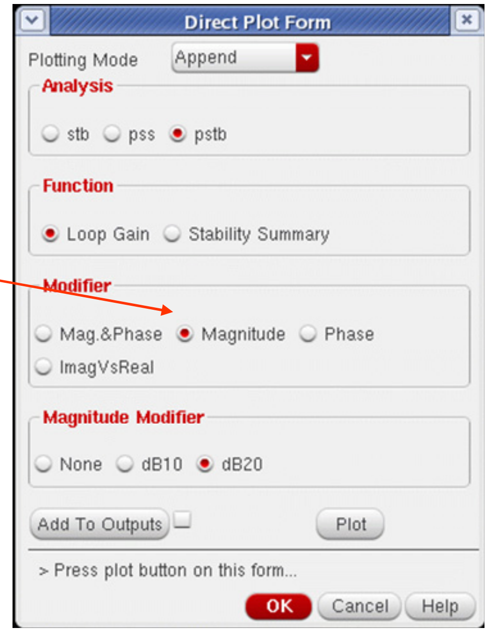
8/4/10 RF Analysis with Virtuoso Spectre Simulator 169

This is a variation of the AC algorithm.

- Set the frequency range to be the area near unity loop gain as determined by the *stb* analysis.
- Choose the *iprobe* instance or the *cmdmprobe* instance for the probe.

Direct Plot the PSTB Results

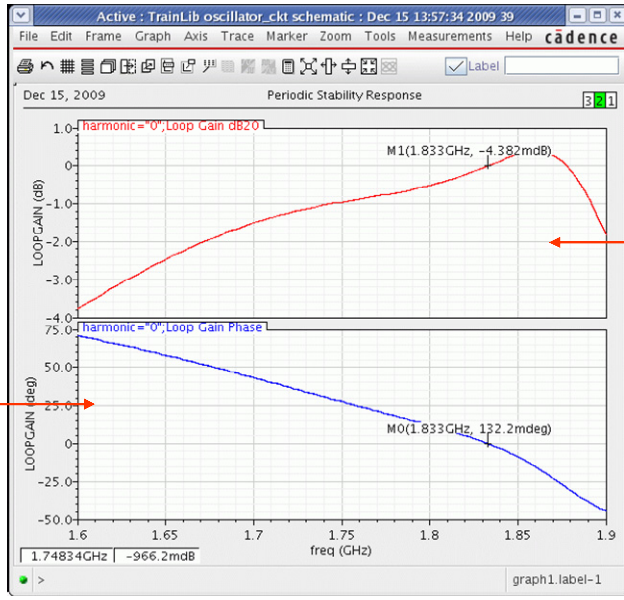
Generally, plot the magnitude and phase separately because the Magnitude and Phase selection doesn't have dB20



When the simulation is finished, you can plot the results from the Direct Plot Form.

PSTB Results for an Oscillator

Position a marker at zero degrees (or multiples of 360 degrees) on the phase curve.



Position a marker on the amplitude curve at the same frequency point.

The *iprobe/cmdmprobe* maintains the loading on the circuit, so the loop gain is correct.

The waveforms shown here fulfill the criteria for conditions of oscillation.

Lab Exercises

There are no labs for this module.

Noise Analysis

Module 7

August 4, 2010

Periodic Noise

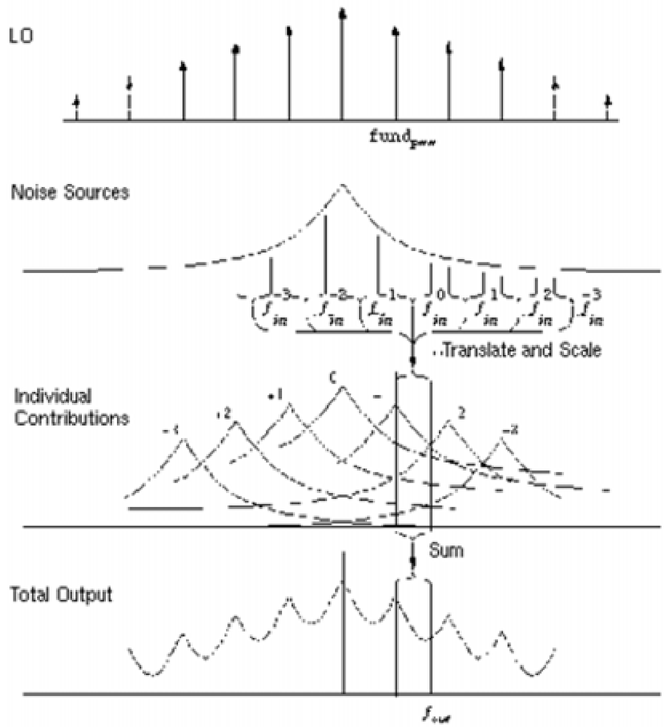
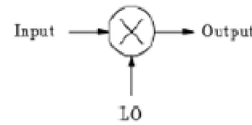


Illustration of noise moving around in a mixer



How noise is translated in frequency?

- For noise sources that are bias dependent, the time-varying operating point modulates the noise sources.
- The transfer function from the noise source to the output modulates the noise source contribution to the output.
- The final result is the sum of the noise contributions that are up-converted and/or down-converted to the desired output frequency.

Periodic noise gets into account the frequency conversion effects in periodically driven circuits.

In this illustration where PSS analysis computes the response to a large periodic signal, you can see that the time average of the noise at the output is computed as a spectral density versus the output frequency.

Two Effects that Translate Noise in Frequency

Bias-dependent noise sources	These sources are modulated by the time varying operating point (shot noise).
Noise source transfer function	This function is time-varying and modulates the contribution of the noise source to the output

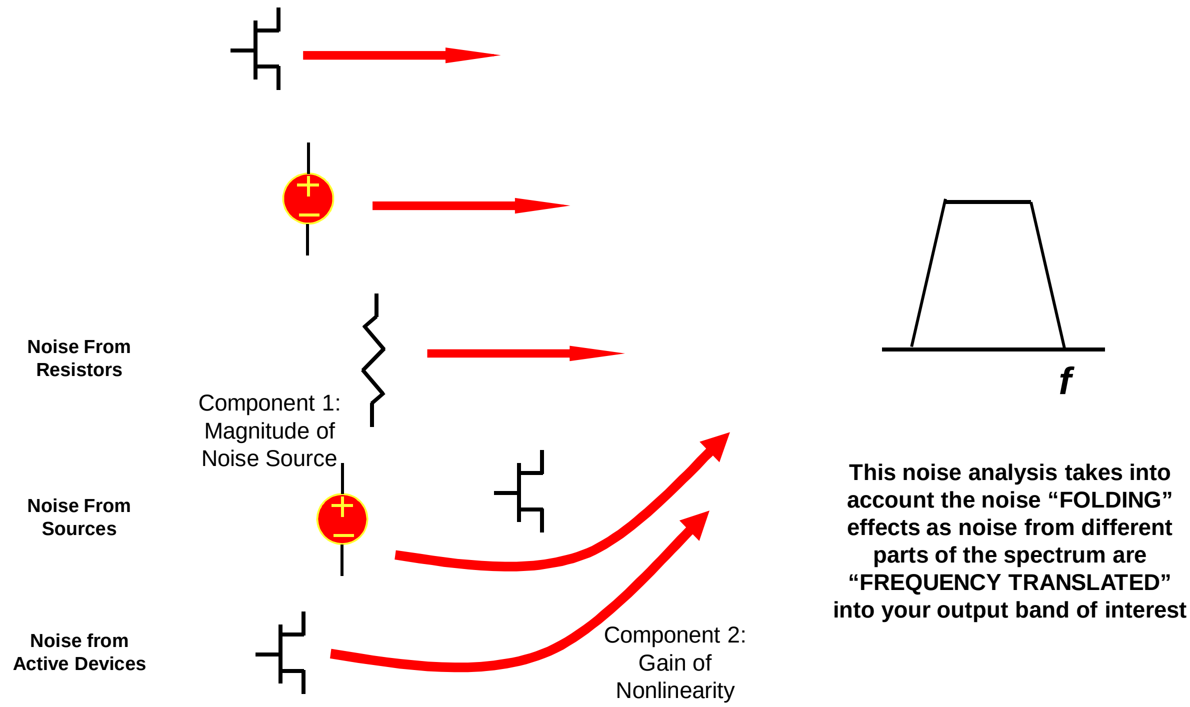
Result of Pnoise Analysis

- The result is the sum of noise contributions, both up- and down-converted, to the output frequency.
- The reference sideband specifies the desired input frequency for calculating Input Referred Noise (IRN), Noise Factor, and Noise Figure.
- The Noise Summary Table displays the following data:
  - Noise contribution (value and %) for each component in the circuit
  - Total output noise
  - Total input referred noise

This Pnoise analysis computes the total noise at the output, which includes contributions from the input source.

A noise summary table is also available that shows which devices and parameters are causing the noise.

Small-signal Analysis Over PSS/HB (PNOISE Example)



8/4/10 RF Analysis with Virtuoso Spectre Simulator 176

Due to the nonlinearities of the circuit, noise folding can occur. The Pnoise analysis calculates the up- and down-converted noise terms.

Fundamental Assumptions of PNOISE/HBNOISE Analysis

The periodically time-varying operating point is the basis for calculating frequency shifted terms.

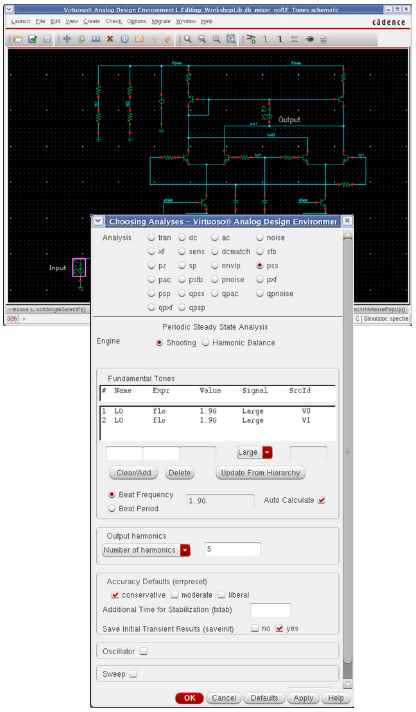
The number of sidebands (harmonics) in the simulation can affect performance.

- When *maximum sideband* is set to 50 or less, the run time is largely unaffected. When *maximum sideband* is set larger than 50, the run time increases linearly with the number of sidebands calculated.
- When Maximum sidebands is set to more than 20, and shooting PSS is used, set the PSS option maxacfreq to the PSS beat frequency times the maximum sideband setting in order to maintain accuracy of the pnoise.
- For Harmonic balance, the number of sidebands should be set to the number of harmonics in the PSS choose analysis form.
- You must find a balance between accuracy, speed, and aliasing when determining the number of sidebands to use.

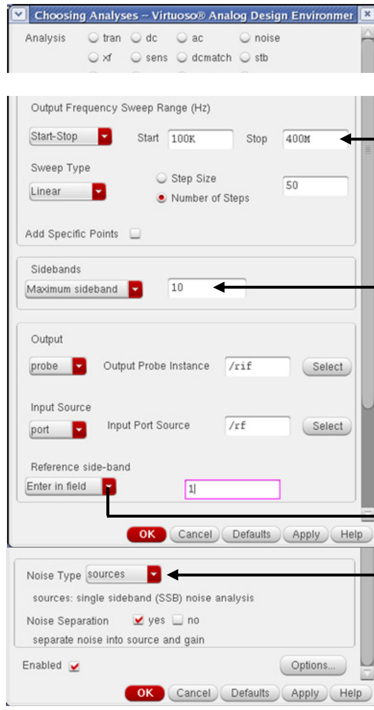
When using *pnoise/hbnoise*, the upper frequency to be considered for noise folding is set by the maximum sideband term.

- PSS shooting has enough timepoints to insure accuracy for up to 20 sidebands. If more than 20 sidebands are needed, set the PSS option *maxACFreq* to the PSS frequency times the number of sidebands.
- PSS harmonic balance and HB only have the harmonics that are specified to be calculated. The pnoise maximum sideband needs to be set to the same number as in the HB analysis. In MMSIM 7.2 and IC614isr2 or later, this is the default.

Pnoise: Mixer Example



PSS Setup

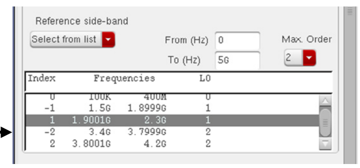


Run Simulation

Enter OUTPUT freq range. PNOISE analysis always adds up the noise at the output.

Number of sidebands affects the accuracy of the answer. Start with the same number as PSS harmonics and then add more sidebands to see if the answer changes significantly.

Reference sideband is required so the output noise can be referenced back to the input through frequency translation.



Noise Type = sources means the noise is going to be averaged across the period of the PSS interval. This is the most common type of noise.

PNOISE Setup

Pnoise analysis is useful for predicting the noise behavior of mixers, switched-capacitor filters and other periodically driven circuits.

- You have to perform the PSS operating point analysis prior to the PNOISE analysis.
- You can access the Pnoise analysis from the *Choosing Analyses* form.
- The number of Sidebands you specify corresponds to the set of periodic small signal output frequencies of interest. For instance setting the Maximum sideband lets you specify the number of frequency conversion terms to take into account.

The *Noise Type* option lets you specify the type of noise you want to compute.

- The *sources* options means to average the noise over the PSS/HB interval.
- *Modulated* allows AM and PM components of noise to be calculated.
- *Jitter* allows the time-domain jitter to be calculated.

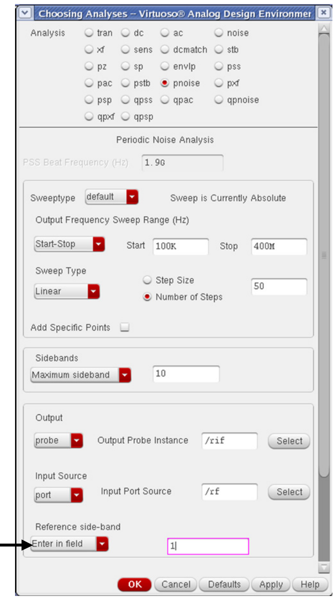
Periodic Noise (Reference Sideband): Example 1

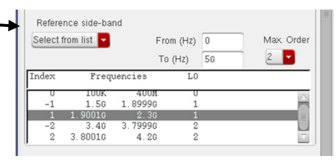
Customer Example: *My fundamental LO is 800M with a tripler, and my RF is 2401. The receiver is direct conversion (DSB). What should I use for reference side band?*

- Both SSB and DSB results will be available in the Direct Plot Form.
- *Reference side-band* specifies the input frequency relative to the output frequency with

  |f(input)| = |f (output) + relsideband*fund(pss)|

- So it should be (1901M-1M)/1900M=1.
- Alternatively, you can select the correct reference sideband from the list.



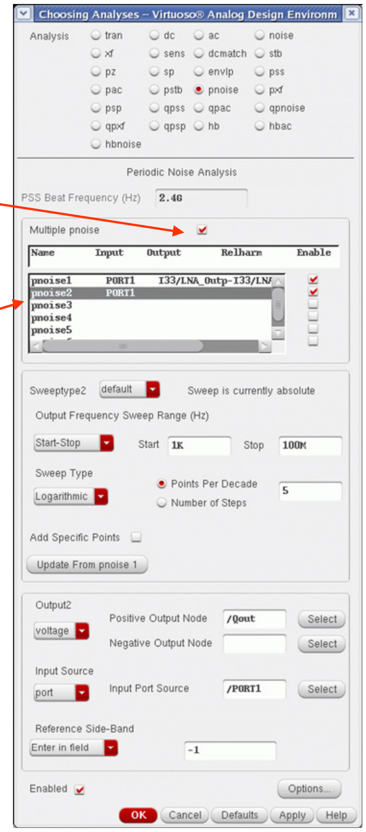


Reference sideband specifies the correct input frequency for the input-referred noise and noise figure calculations.

The easiest way is to use select from list.

The Multiple Pnoise Option

- You can carry out a multiple pnoise analysis at different ports of your design. The pnoise *Choose Analyses* form enables you with a checkbox to access this option.
- To enable **Multiple pnoise**, check the box at the top of the form.
- Fill out the form as needed for each output point in the circuit.
- You can switch between the analyses by selecting the entry.



Multi-pnoise is useful for when you want to measure the noise at more than one point in a cascaded circuit.

Set up each pnoise as normal for that point in the circuit.

Pnoise (Reference Sideband) (Example 2)

Customer Question: *How do you set Reference Sideband for a double conversion mixer?*

For the double conversion receiver, which has two mixers (say, LO1 and LO2), you need to use QPSS/QPnoise.

- 1. Set RF port as DC if there is no blocker signal present, so it will be a 2 tone qpss simulation.
- 2. Since both LO1 and LO2 have large amplitudes, set higher *maxharms* for both on the LO tones
  - For example, RF is 2.4G+10K, LO1 is 1.9G with an IF of 500M and LO2 is 500M with a 10K output frequency, the reference sideband is -1 -1. Both LOs down convert using the first harmonic.
- 3. When you set up QPnoise, pay attention to the Sidebands option. You can use the following equation to put the sideband vector in the field, for example -1 -1,...

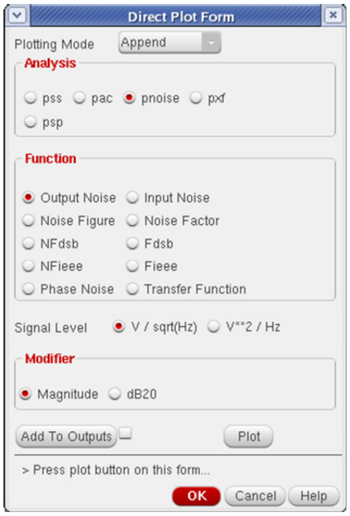
  |Fin| = |Fout + K1 X FLO1 + K2 X FLO2|

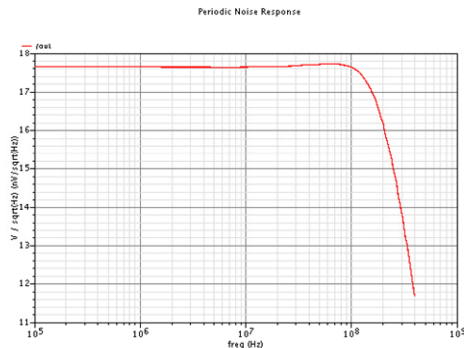
  - **Or you can simply choose "Select From Range" to choose the correct input frequency.**

8/4/10 RF Analysis with Virtuoso Spectre Simulator 181

- The QPnoise analysis is similar to the conventional noise analysis except that with QPnoise the circuit is first linearized about a quasi-periodically time-varying operating point as opposed to a simple DC operating point. This linearization includes frequency conversion and intermodulation effects.
- Once the QPSS analysis completes, the small stimuli representing both individual noise sources (FLO1 and FLO2) in the circuit as well as the input noise are applied to the periodically-varying linear representation to compute the small signal response.
- Use the refsideband parameter to specify which conversion gain to use when computing input-referred noise, noise factor, and noise figure.

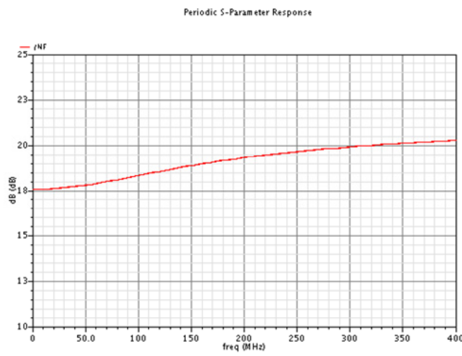
Pnoise: Mixer Example





Output Noise

Noise is always added up at the output



Noise Figure

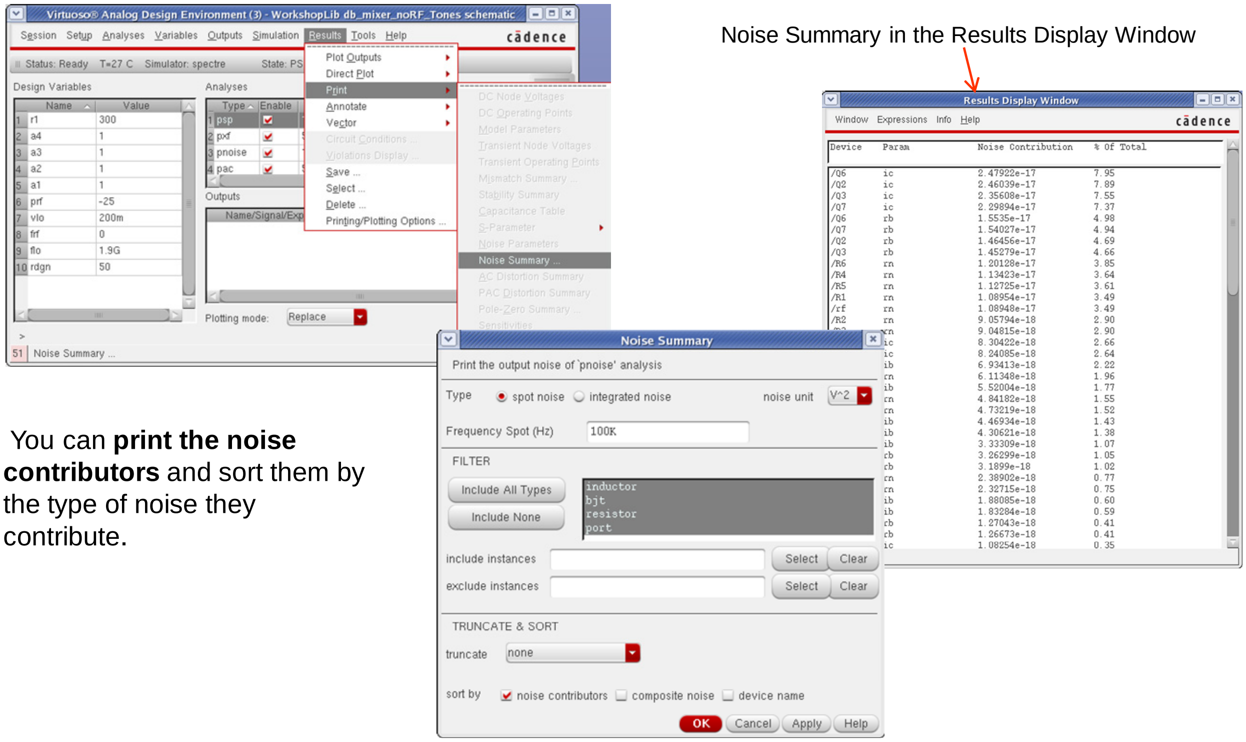
Noise figure takes the output noise and references it back to the input through the reference sideband

Notice this is the same curve as from the NF button under the PSP analysis in Chap 5, but the X-axis is the output freq range, not the input freq range.

- You can use the *Results–Direct Plot–Main Form* after the Pnoise analysis to plot the results.
- There are available functions for plotting the Output Noise, Noise Figure, Noise Factor, Input Noise, etc

Noise Summary Table of Pnoise Analysis

Noise Summary in the Results Display Window



You can **print the noise contributors** and sort them by the type of noise they contribute.

- The *Results–Print–Noise Summary* option available in the ADE has been a popular feature of ADE for many years.
- The spot noise option produces a noise summary at a single frequency.
- The integrated noise option produces a noise summary integrated over a frequency range.
- The FILTER section provides a method of limiting the summary report to include only some of the device types. In the list box, you can select those devices that you want included in the report.

Noise Summary Table With Multiple Pnoise

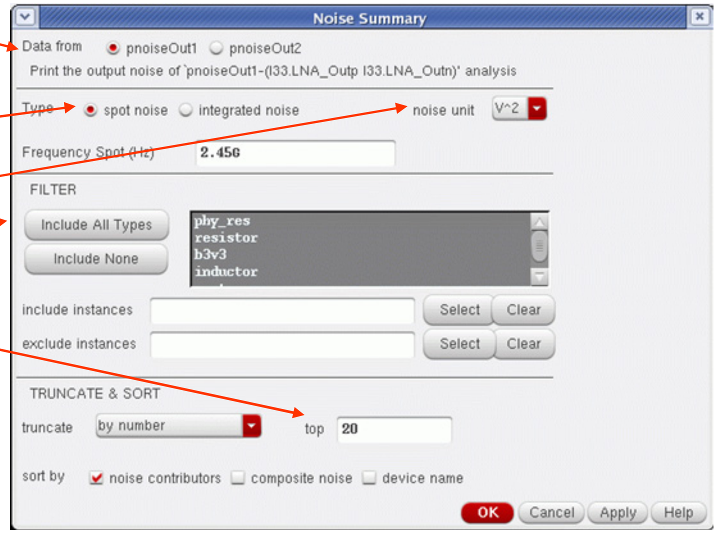
Select the pnoise whose result you want to view from the multiple options available at the top of the form.

Select Spot or integrated noise.

Select the noise Unit. (V or V^2)

Select Include All Types.

Set the number of noise contributors here.



When you do a multiple pnoise analysis, the *Noise Summary* form lets you choose from the evaluated results of the different noise contributors that you had mentioned in the Pnoise setup form and print their noise summary.

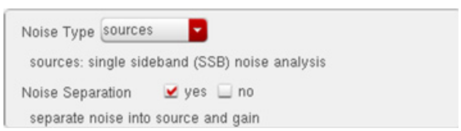
Noise Separation in Pnoise/Qpnoise Analysis

In a previous version of Spectre® RF, Noise Summary could not differentiate how much noise was coming from the noise source and how much was coming through the periodic transfer function.

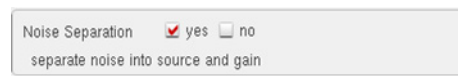
Now, you can decrease the output noise:

- By decreasing the noise sources with different device dimensions
- By decreasing transfer functions with alternative circuit architectures.

If Pnoise Separation is selected, there will be an extra option in the *Direct Plot Form*.



Setting of Noise Separation in Pnoise analysis form

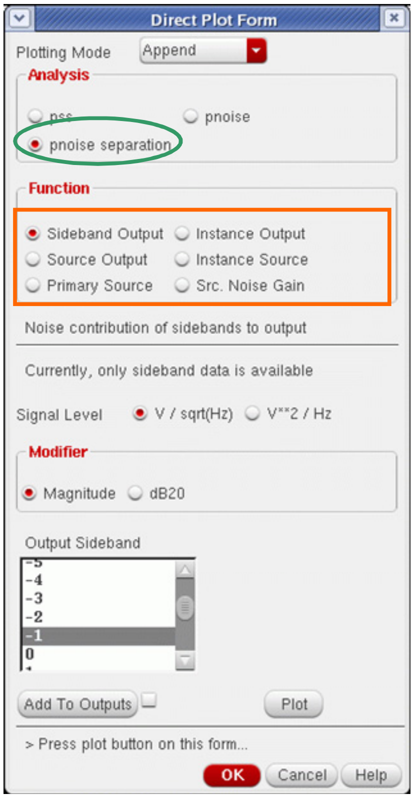


Setting of Noise Separation in QPnoise analysis form

For a Pnoise analysis, after setting **Noise Separation = yes**, run the simulation, the Pnoise separation feature is included during the simulation and the corresponding results are saved.

For a QPnoise analysis, after setting **Noise Separation = yes**, run the simulation, the QPnoise separation feature is included during the simulation and the corresponding results are saved.

Noise Separation Direct Plot Form



Sideband Output plots the noise contribution of selected sidebands to the output.

Instance Output plots the noise contribution of some instances such as MOS, BJT etc to the output at one selected sideband.

Instance Source plots the noise sources of some instances at one selected sideband.

Source Output plots the noise contribution of primary noise source such as re, rb in a BJT to the output at one selected sideband.

Primary Source plots the primary noise sources such as re, rb in a BJT at one selected sideband.

Src. Noise Gain plots the noise gains of primary noise sources such as re, rb in a BJT from source to output at one selected sideband.

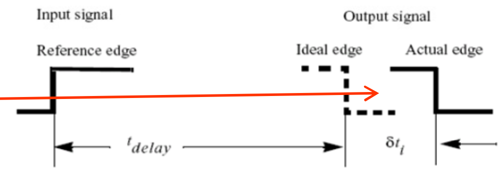
If the *Noise Separation* option is selected to run during pnoise simulation, after the simulation ends and you access the Direct Plot Form to see the output, there is an additional option for pnoise separation.

When you select the pnoise separation option, you can choose from different available functions that let you plot the different noise contributions.

Introduction to Jitter

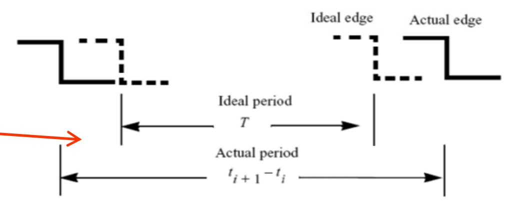
Jitter (Jee) (Driven digital Circuits)

- The deviation of a signal event from its ideal position.
- This is often observed at the edge, when signal is crossing a preset threshold.



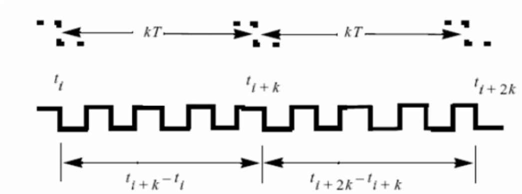
Period jitter (Jc) (Oscillators)

- The variation in the period from the nominal period.
- This is also called *cycle* jitter.
- This jitter reduces the time when reliable data can be observed.



Adjacent period jitter (Jc-c) (Oscillators)

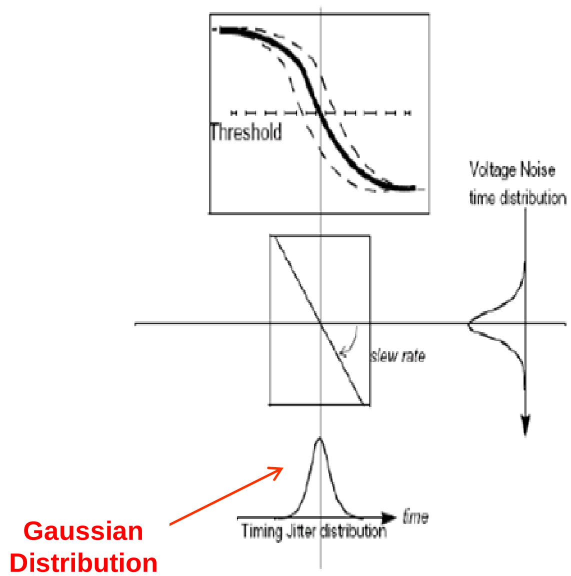
- Local variations in the period, from one cycle to the next.
- This is also called *cycle to cycle* jitter because it gives the time difference between successive periods of a signal.



- Jitter is normally an undesirable factor that one has to deal with when designing RF systems. Jitter gives rise to randomness between signal events. It can be observed in the amplitude, frequency or phase of periodic signals.
- You can quantify jitter in clock signals in the ways shown on this slide.
- You have the option of evaluating and plotting the absolute jitter, period jitter, and adjacent period jitter (cycle-to-cycle) in a signal. You can access these functions with the Direct Plot form after the simulation finishes.

Integrated Jitter: Main Features

- The Actual Phase Noise curve is used in the integration. Flicker noise is included, no assumption about the shape of noise PSD.
- Adaptive integration for the linear and log frequency sweep. Calculator integration can not be used for comparison.
- You can change the limits of the integration if needed.
- Long term jitter is accurate as long as the small signal approximation is valid.



8/4/10 RF Analysis with Virtuoso Spectre Simulator 188

Integrated jitter over a specific bandwidth can be obtained by integrating the measured phase noise over it. This is integrated across all the cycles in the bandwidth.

Jitter Applications: Autonomous Circuits

Output noise is dominated by PM component.

- The perturbation from noise in components, substrates, and sources is translated into mostly phase noise at the output.
- Phase deviation accumulates with time in the absence of a restoring force or a feedback.

Spectre RF approach

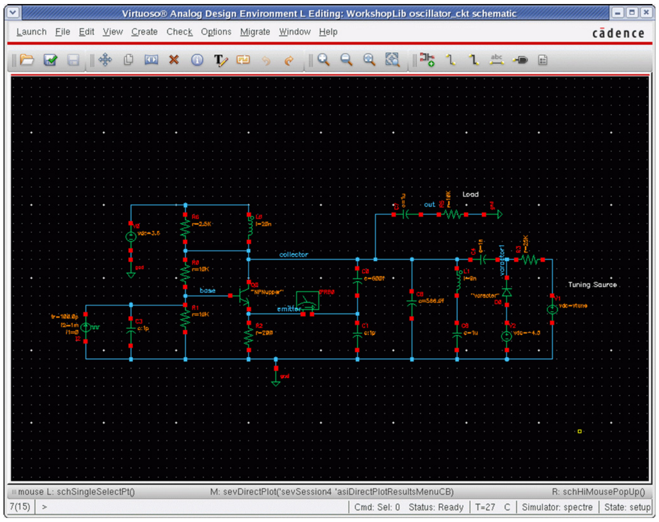
- Find PM component of the cyclostationary noise using modulated Pnoise analyses.
- Convert the Phase Noise into the jitter.

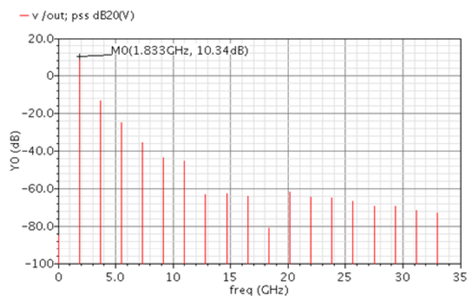
$$J_c^2(kT_c) = \frac{1}{(\pi f_c)^2}\int_0^{\infty} S_\phi(f)\sin^2(\pi k f T_c)df$$

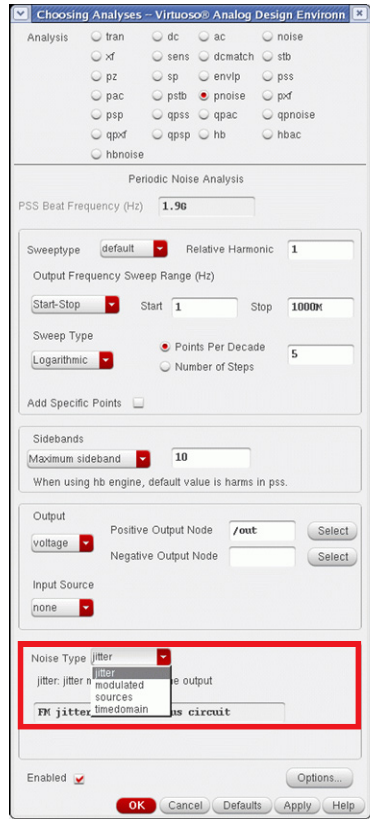
8/4/10 RF Analysis with Virtuoso Spectre Simulator 189

PM jitter is the *phase modulation* jitter component at the output where the phase of the signal gets randomly modulated.

VCO Phase Noise: Jitter







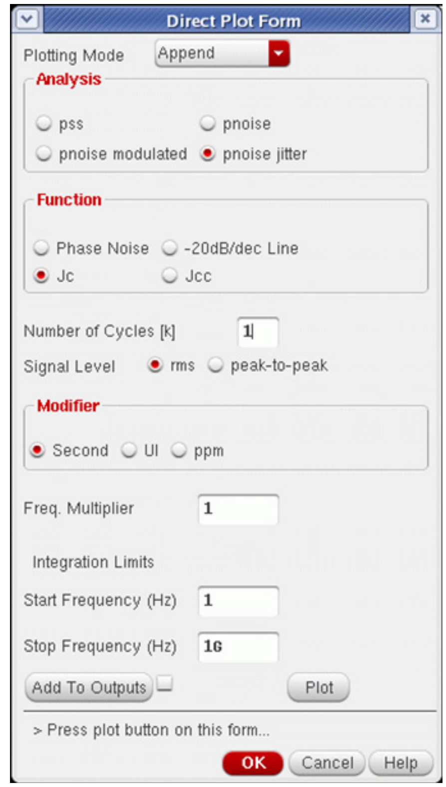
Here you see the *pnoise* simulation setup to calculate the jitter at the output of a VCO. The **jitter** option is selected in the *Noise Type* cyclic field.

The spectrum result shown is for an oscillator, using PSS analysis.

Integrated Jitter: Autonomous Case

Use the Direct Plot Form.

- Select period jitter (Jc) or adjacent period jitter (Jcc).
- Choose single-period or long-term jitter.
- Select the units and the signal level (RMS or peak-to-peak).
- Select the frequency range or BW for integration.
- Compute the value of the timing jitter by selecting Plot.

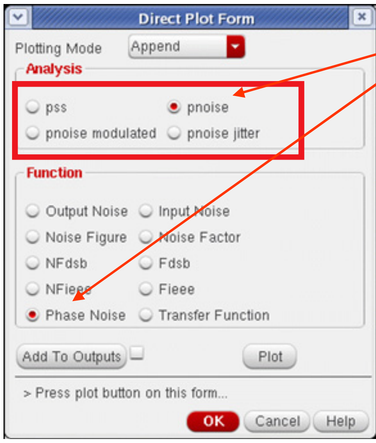


Here you see the **Direct Plot Form** that you can access from the ADE Results menu to plot the results at the end of pnoise simulation.

The *period jitter* (Jc) and *adjacent period jitter* (Jcc) functions are available as functions with the *pnoise jitter* analysis to plot the output.

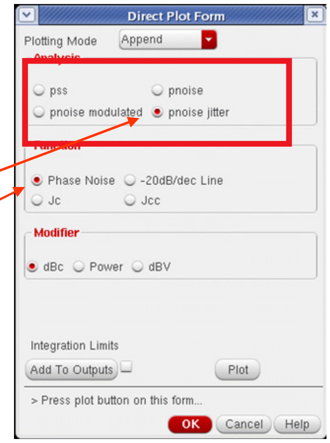
Note: Something must already be plotted in the subwindow you are using in order to see the jitter calculation.

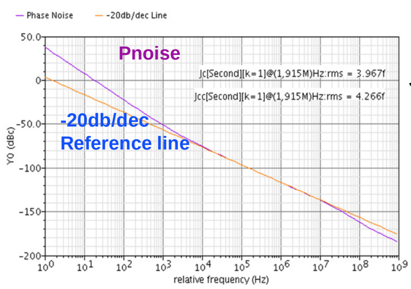
VCO Outputs Using the Direct Plot Form



Plotting Phase Noise from the pnoise selection is SSB phase noise

Plotting Phase Noise from pnoise jitter is DSB phase noise





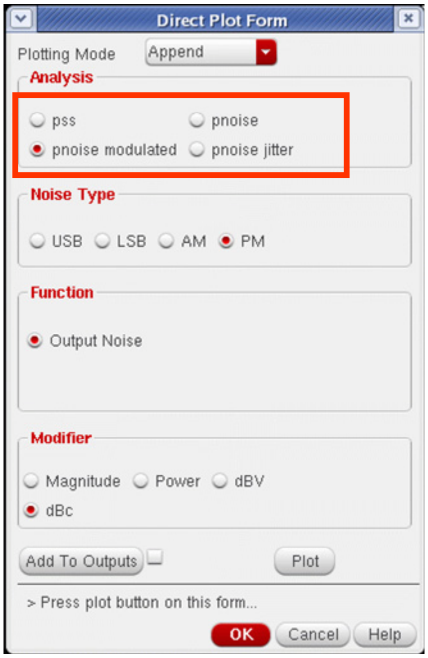
Outputs for Noise Type =Jitter in PNOISE form

8/4/10 RF Analysis with Virtuoso Spectre Simulator 192

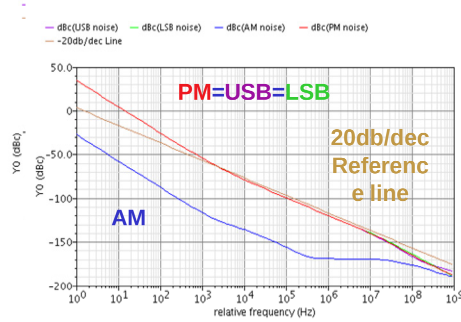
The *phase noise* is numerically integrated during the jitter computations. It accounts for all noise contributors that exist in the schematic.

- Depending on the analysis chosen in the *Direct Plot Form* you can now plot the phase noise calculated either as a *single-side band* (SSB) phase noise using the pnoise analysis **or** plot the phase noise as a *double-side band* (DSB) phase noise using the *pnoise jitter* analysis.
- In the waveform above, you can see the combined plots for the phase noise using the *pnoise jitter* analysis with the *–20dB/dec Line* function.
- The -*20dB/decade* function provides a way to find the frequency range where white FM noise is significant. The function also makes it possible to see the presence of the flicker noise in the system.

VCO Outputs Using the Direct Plot Form (continued)



Outputs for Noise Type = Modulated in PNOISE form

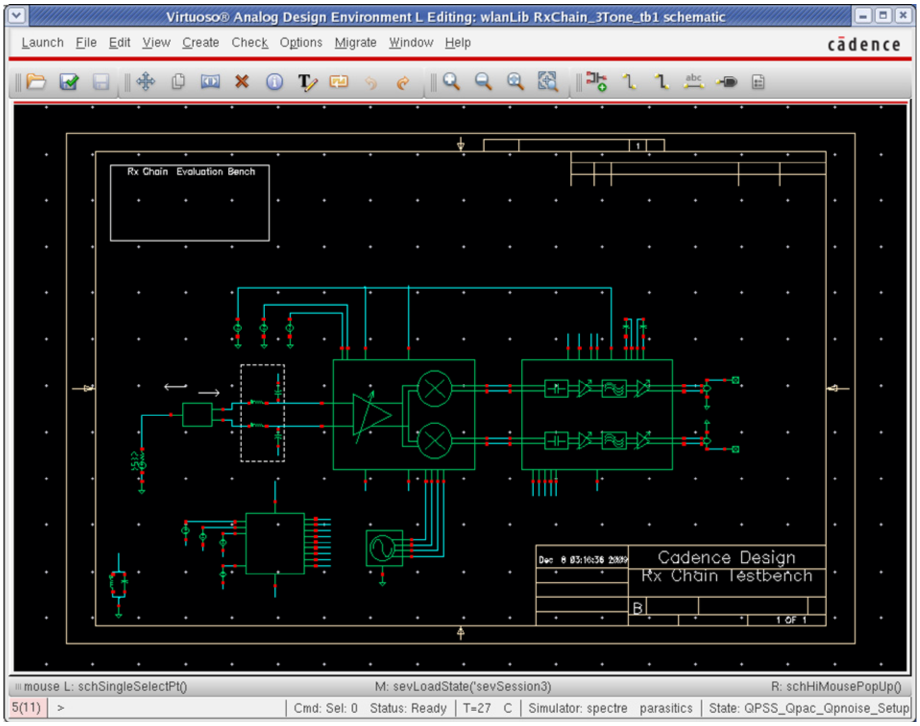


8/4/10 RF Analysis with Virtuoso Spectre Simulator 193

The Direct Plot Form also gives you the option to plot the output modulated pnoise, depending on your choice of separation of the Noise Type from PM and AM components to noise calculated from the LSB and USB.

Here you can see the noise from all the four types of noise modulations.

Noise in the Presence of an In-Band Interferer

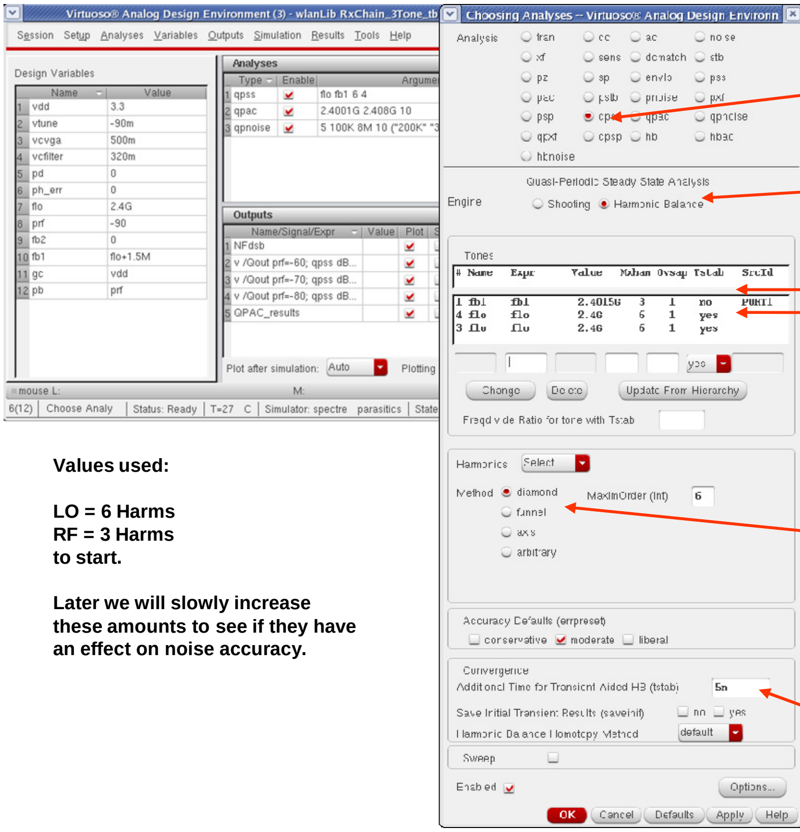


Circuit inventory:

nodes	**1841**
balun	**1**
bsim3v3	**989**
capacitor	**87**
inductor	**27**
mutual_inductor	**12**
phy_res	**341**
port	**3**
resistor	**1034**
transformer	**4**
vcvs	**8**
vsource	**10**

The next few pages show you how to set up and run a QPnoise analysis in the case of multitone circuits. The receiver chain shown in the above schematic is used as an example.

Setting Up QPSS (HB can also be used)



Because the problem is multitone, we will use QPSS. (HB can also be used)

Harmonic Balance is more efficient for large, multi-tone problems

You have an RF bandwidth of 8M. You have an LO of 2.4G and in in-band interferer at 2.4015G (1.5M IF) Set TSTAB on the LO port to get past initial transients and use more harms on this tone.

Values used:

LO = 6 Harms
RF = 3 Harms
to start.

Later we will slowly increase these amounts to see if they have an effect on noise accuracy.

Because the problem is large, the best approach is to use Diamond cut to trim some of the higher order harmonics that do not have very much power. Works even better because the circuit is balanced and filtered.

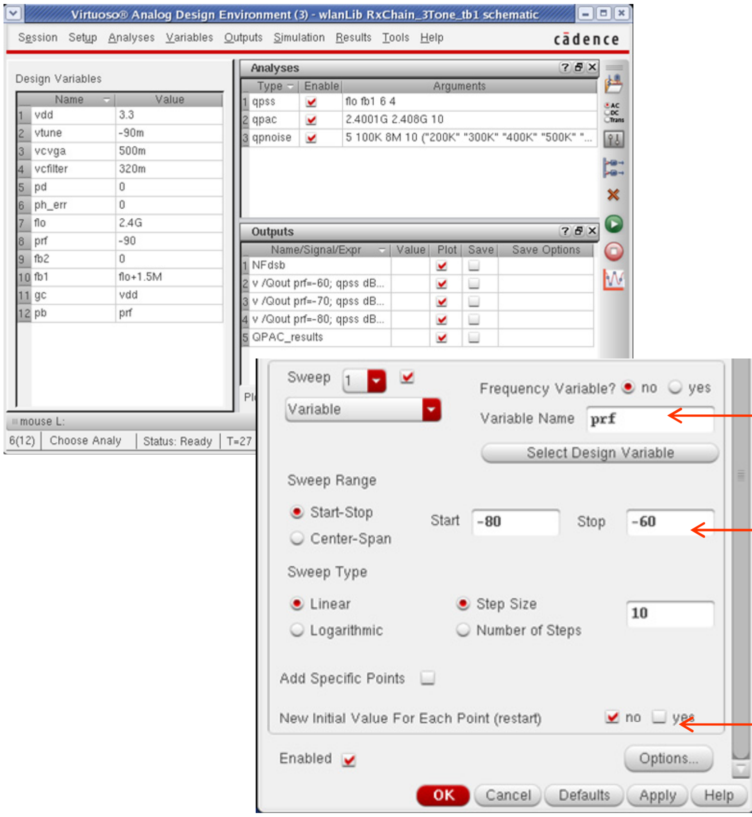
TSTAB is set to 5n. If we had a divider, TSTAB might be longer. Use saveinit, so you can see if the waveform is settled.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 195

Setting Up a QPSS Analysis

- *Always use harmonic balance.*
- Set a reasonable number of harmonics for the tones.
- If some speedup is desired, use diamond cut and set the MaxImOrder to the number of harmonics for the largest amplitude tone.
- Use *tstab* to reduce the number of frequency domain iterations and/or to get the circuit to converge.
- If the circuit is not converging, try hbhomotopy=source for systems with large amplitude LOs.
- hbhomotopy = tone may also be effective.

Setting Up QPSS (continued)



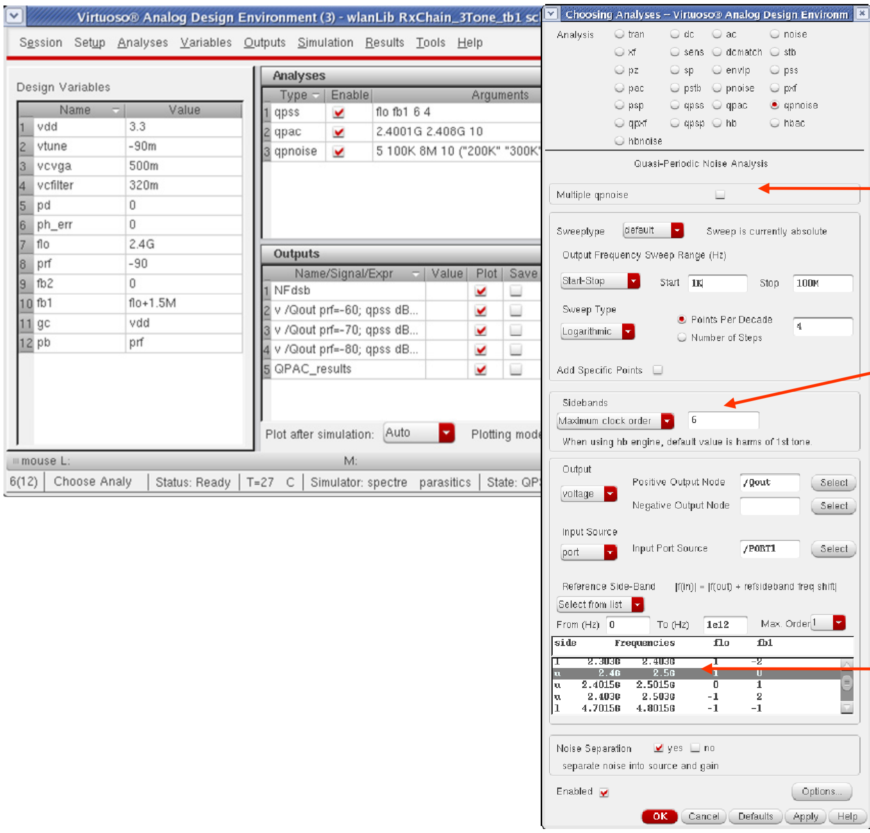
We are stepping the power of the RF tone and creating 3 different QPSS solutions at -80, -70 and -60. Notice that the receiver is being pushed very far into compression to show the abilities of the tool.

When doing sweeps with harmonic balance, always set restart=NO. This will use the last solution as a first guess for the next. If you use Shooting, always set this to yes.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 196

The extension of the QPSS form is shown here where the sweep section is enabled. This elaborates the QPSS form so you can fill in the sweep ranges and other required parameters.

Setting Up QPnoise



Netlist and Run Simulation

We are sweeping the QPnoise analysis across the IF band of 100K to 8M.

Set **Maximum clock order** to the number of harmonics in the tone with tstab enabled.

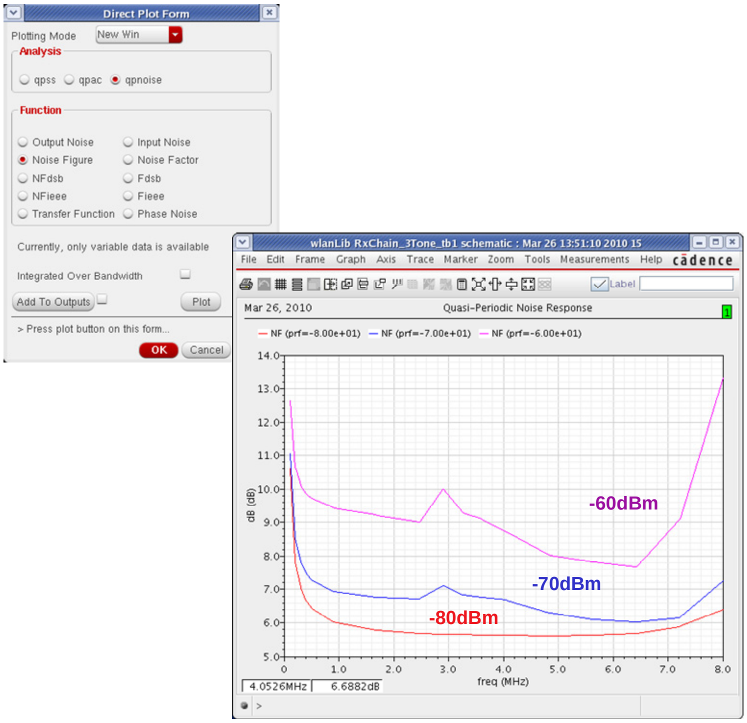
Reference Side-band has to be set to take the output noise and reference it back to the input for Noise Figure. Always use **Select from list** and choose the Index that covers the input frequency.

QPnoise includes all the noise frequency translations based on the nonlinearity calculated by the QPSS.

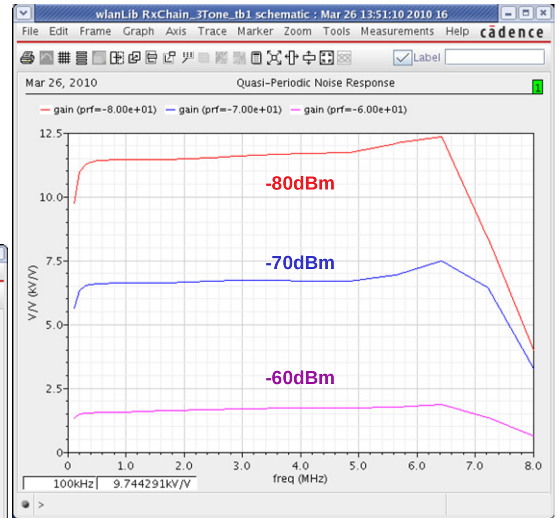
QPnoise has maximum clock order, which sets the highest harmonic of the tone with tstab enabled to calculate all the mixing products for. It agrees with the setting of the number of harmonics for the tone with tstab enabled.

Especially for QPnoise, *Select from list* is recommended for the reference sideband. Select the desired input frequency range in *Select from list*.

Direct Plot



Plot of Noise Figure

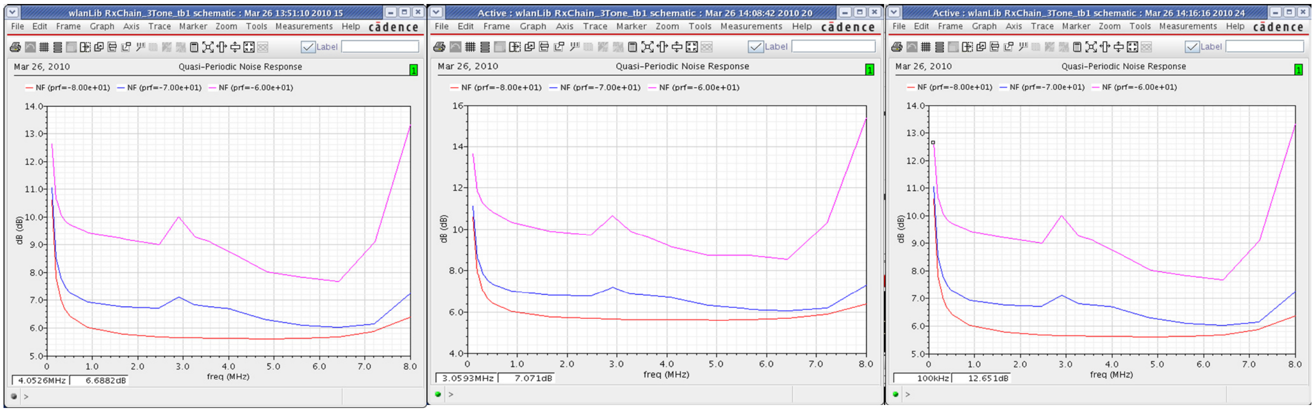


Plot of Transfer Function

This simulation takes about 4 ½ minutes on an IBM laptop with:
- **LO=6 harms**
- **RF=4 harms**
- **Diamond cut at 6th order**
- **Pnoise Sidebands = 5**

- The chosen signals in the Outputs section in the ADE plot automatically. You can access the *Direct Plot Form*, choose *qpnoise* here, and plot the noise figure and transfer function.
- The results above are at three different sweep values that were set up in the pss analysis.

Direct Plot: Adding More Detail



SimTime = 4 ½ mins **LO=6 harms** **RF=4 harms** **Diamond cut at 6th order** **Pnoise Sidebands = 5**	**SimTime = 8 mins** **LO=10 harms** **RF=6 harms** **Diamond cut at 10th order** **Pnoise Sidebands = 5**	**SimTime = 4 ½ mins** **LO=6 harms** **RF=4 harms** **Diamond cut at 6th order** **Pnoise Sidebands = 10**

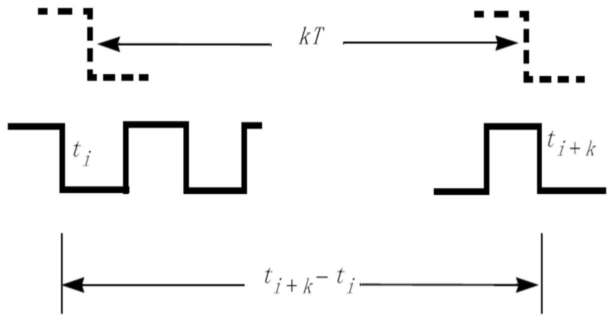
Little or no change in solution as more harmonics and sidebands are added. Therefore, the answer is well converged.

The number of sidebands chosen, the trimming methods used, and the harmonics chosen for the RF or LO affect the simulation results.

The plots above draw a comparison of different setups to calculate noise figure.

Long-Term Jitter

- Long term jitter characterizes the variation in the accumulated width of a large number of clock periods.
- If the observation time is long and involves many periods, k-cycle jitter represents the long term jitter.



- An open-loop VCO accumulates jitter with time and increases as the square root of time for the white noise dominated oscillator.
- In PLLs, the variation in the clock edges over the long time interval are partially tracked out by the negative feedback of the loop.
- The accumulation of the jitter over many periods is limited by the loop bandwidth.
- In general, noise frequency content, system dynamics, type of the measurement, and the observation time defines the variation of the long term jitter with time.

Long term jitter is also called *accumulated* jitter because it measures the accumulated variation between the measured and initial signals at a given location after a very large number of clock cycles.

How large or “long-term” depends on the type of RF circuit/system, application, and frequency involved.

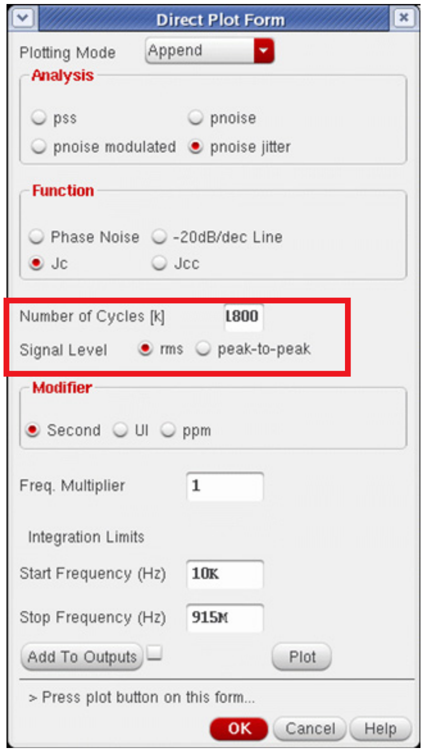
Setting Number of Cycles and Start Frequency

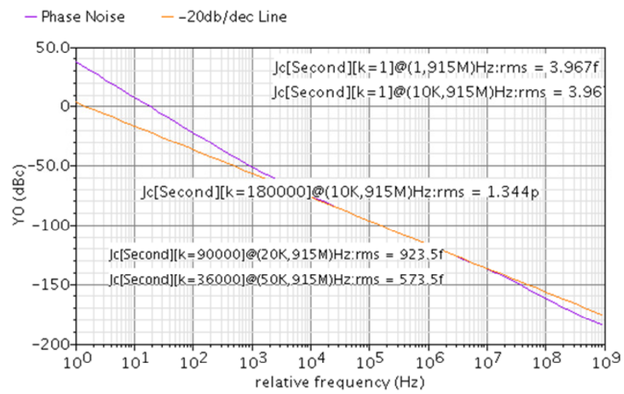
$$f_{min} = \frac{1}{T_0} \qquad k = \frac{f_{osc}}{f_{min}}.$$

- The time of observation or the loop bandwidth of PLL are the two typical factors that are used to select the “Number of Cycles” k and the “Start Frequency” (Fmin).
- For the lowest frequency of phase noise that can be observed, we have to use either PLL loop bandwidth frequency or the jitter measurement (observation) time (To)
- The maximum frequency (½ Fosc) does not affect the long-term jitter computation because the high frequency content of phase noise PSD will be much lower than at small frequency, especially in the presence of the flicker noise.

VCO Test: Long-Term Jitter

Assume PLL loop bandwidth of 10K, k=1.8G/10K





Cycle jitter (k=1) does not change with lower integration limit.

Loop Bandwidth of 20K and 50K for comparison.

1) For low *k, it* will act as a high pass filter and the contributions of the low frequency noise are negligible for a jitter observed over a small number of periods.
2) Increase *k* and the slow varying noise begins to play a more and more important part in the jitter.
3) That makes long term jitter very sensitive to the lower frequency limit that is used in integration of the phase noise.

This page shows the long-term jitter for a VCO. Long-term jitter measures the standard deviation.

Jitter Applications: Driven Circuits

Logic and sampling circuits are sensitive to noise only during the signal transition through the threshold.

- These circuits are sensitive to input noise during transition.
- They Produce the highest output noise when transitioning.
- Typical measurements (NF, IRN) are not helpful.

Spectre RF Approach

- Find transition event for the noiseless signal in PSS. Save slew rate.
- Compute an additive noise at the time of event and convert it into jitter.

$$J_c = \sqrt{2var(j_{PM})(t_X)} = \frac{\sqrt{2var(n_v, t_X)}}{\frac{d}{dt}v(t_X)}$$

The noise in the low and high state doesn't matter, because the circuit can't clock another digital circuit then.

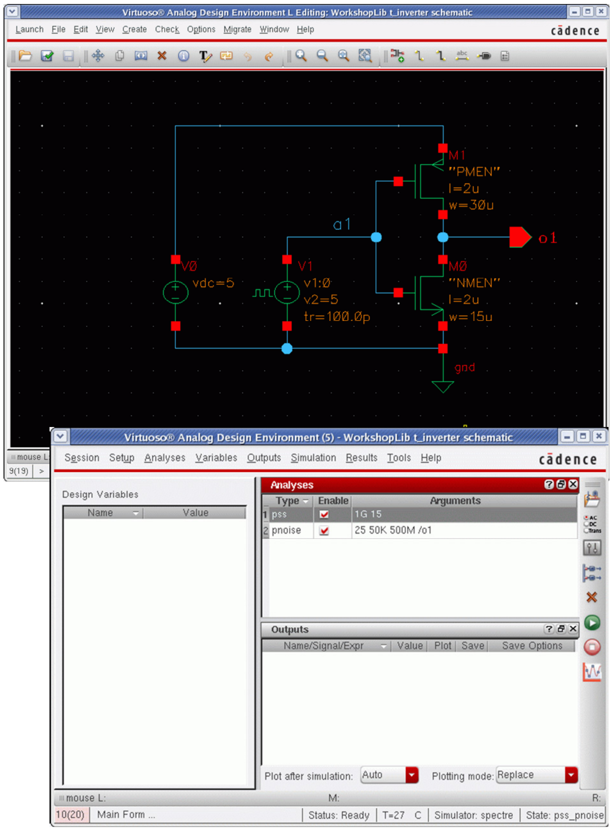
The only time the noise matters is during the transition when a threshold for switching of succeeding stages is crossed.

Integrated Jitter: Sampled Jitter for Driven Circuits

- Pnoise setup lets you select the threshold and the direction of the signal during the transition.
- PSS automatically finds the threshold point (plus or minus the convergence criteria) and puts a timepoint there..
- Sampled Pnoise (*tdnoise* type) computes the value of the sampled noise at the transition.
- The *Direct Plot Form* accessible through ADE lets you plot the PSD of the absolute/edge jitter and compute the timing jitter by using different units (seconds, unit intervals, part per million).
- Small signal approximation is used. It is also assumed that the slew rate is not affected by the noise.

The next few pages explain and provide an example for sampled pnoise (time-domain jitter).

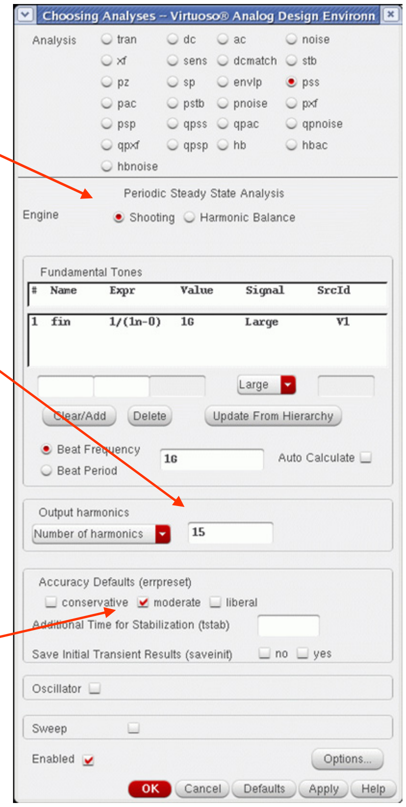
Time Domain Noise Example PSS Setup



Always use Shooting.

Set harmonics to one quarter to 1/3 the number of sidebands in pnoise, with a minimum of 15.

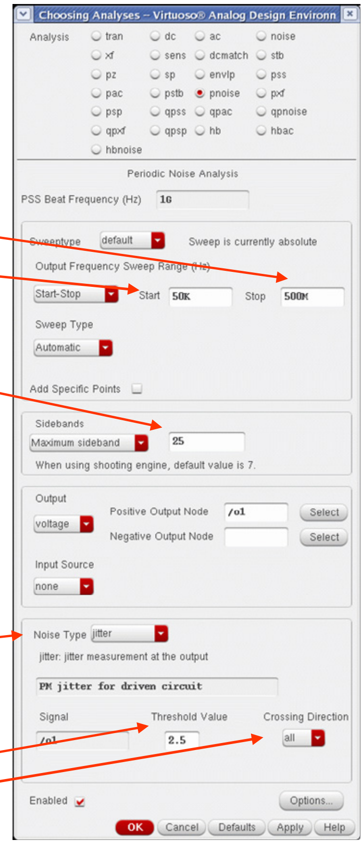
Moderate accuracy is good enough.



This page shows you how to set up the PSS analysis as the first step in evaluating time-domain jitter. The driven circuit under consideration is an inverter.

Note that the shooting engine is used for this evaluation.

Time Domain Noise Example Pnoise Setup



Set the Stop frequency to half the PSS frequency.

Set the start frequency three of 4 decades smaller.

Set sidebands high enough to capture the noise adequately. In many cases, this is Ft divided by the PSS frequency.

If max sidebands is greater than 20 and you are using shooting, set the PSS option maxacfreq to the PSS beat frequency * number of sidebands.

Set the Noise Type to jitter.

Set the threshold voltage and crossing direction.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 206

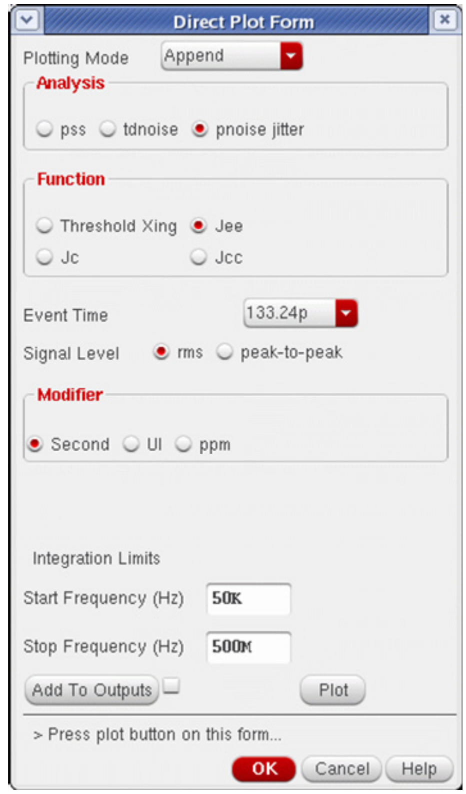
You now have to set up the *pnoise* analysis next. Note how the *Output Frequency Sweep Range* is set up.

Note: The main concern for driven circuits is in choosing a reasonable amount of sidebands. For sharp transitions and highly nonlinear situations, it might require inclusion of a large number of sidebands into the noise computation. The question is how much is enough. Because adding twice as many sidebands in *pnoise* slows you down while only increasing the final result by 2%, is it worth it? If you are well aware by design which sidebands contribute to the frequency translation, set max sidebands to that frequency divided by the PSS frequency.

Integrated Jitter: Direct Plot for Driven Circuits

Use the Direct Plot Form.

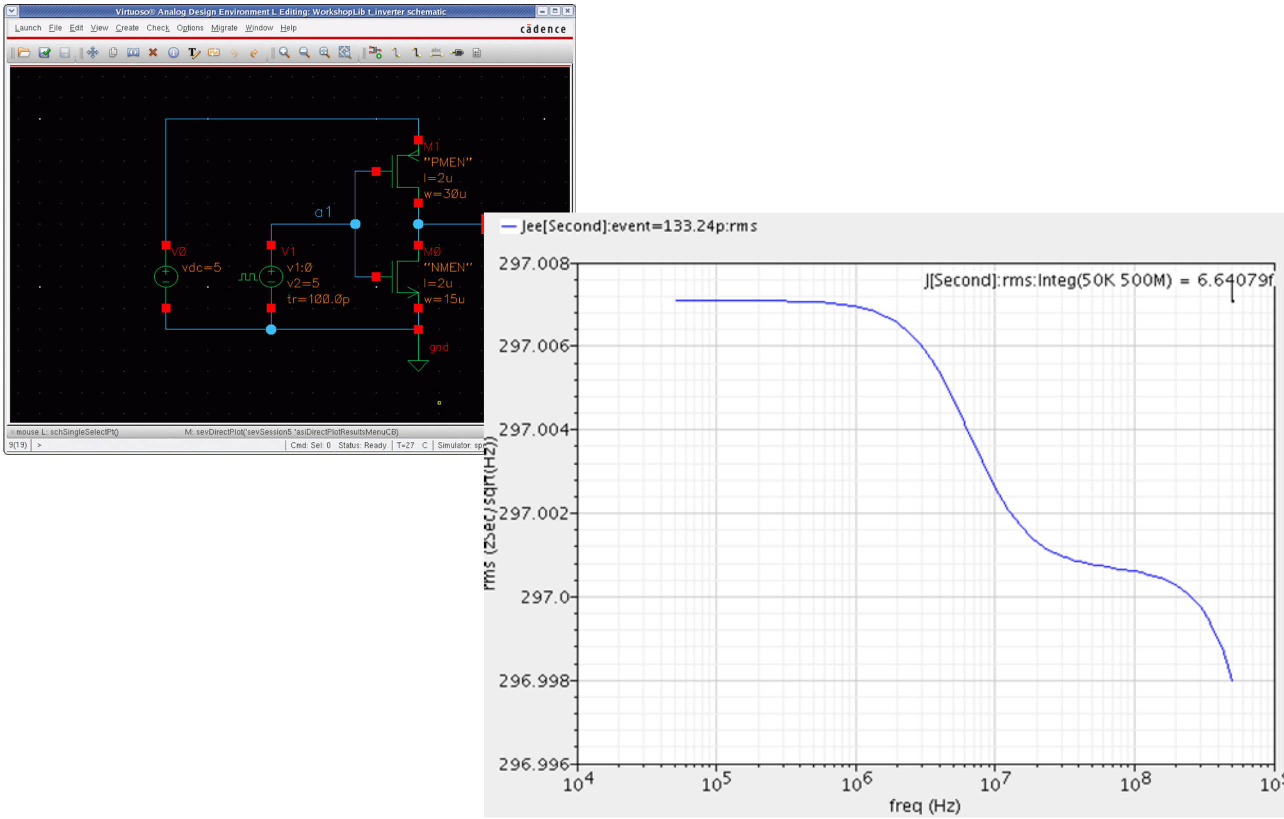
- Select absolute jitter (Jee).
- Select the units and the signal level (RMS or peak-to-peak).
- Select the frequency range or BW for integration to the frequency range set in the pnoise form.
- Click **Plot**.



When the netlisting and simulation run has finished, you can plot the results with the Direct Plot Form. You have the option of plotting the pnoise jitter in the frequency domain as well as the time-domain noise (jitter).

Note: You must have a waveform plotted in the subwindow you are sending the jitter calculation to so you can see the jitter number.

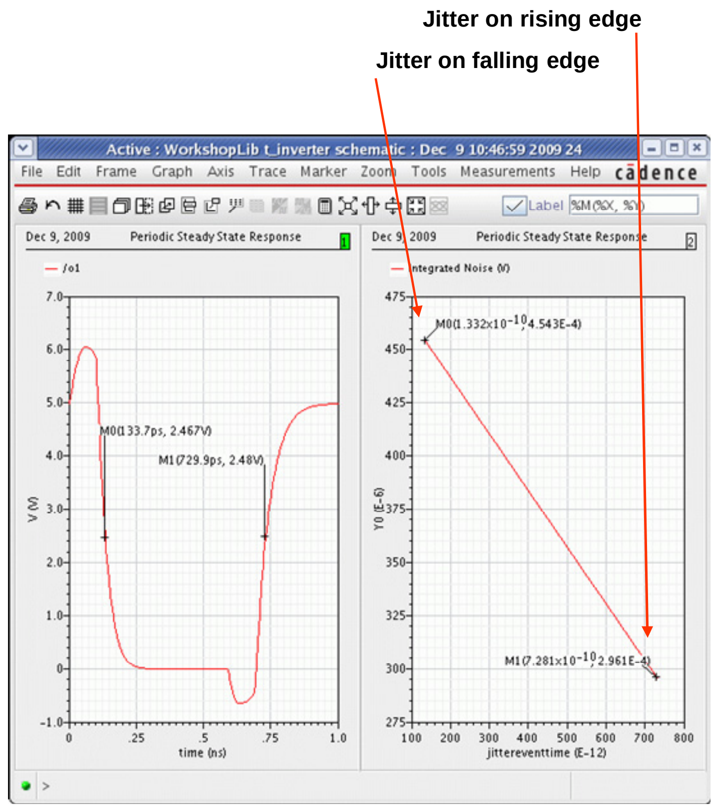
Time Domain Noise Example



8/4/10 RF Analysis with Virtuoso Spectre Simulator 208

This is the absolute jitter result over the range that you specified, shown in the frequency domain.

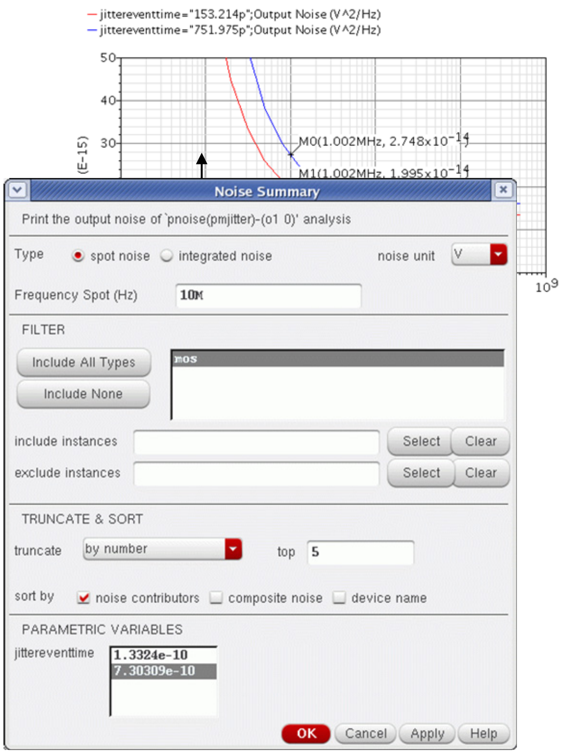
Time Domain Noise Example (continued)

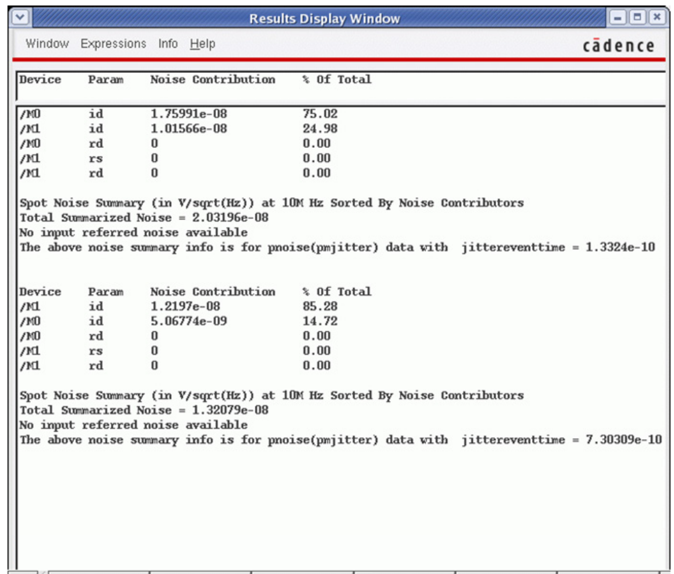


Here you can see the plot for the integrated output noise in the time domain. You can get the jitter values at the rising and falling edges.

Time Domain Noise Example (continued)

Spot Noise Summary Table



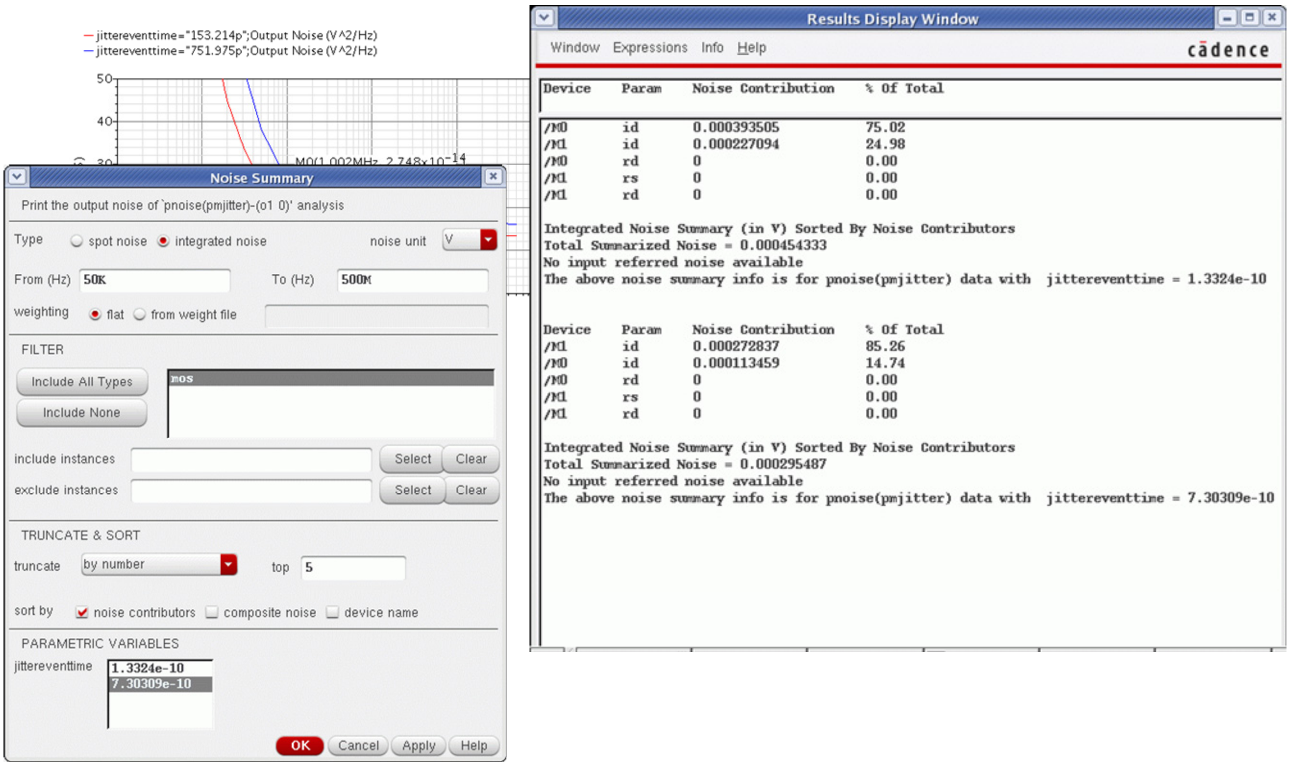


8/4/10 RF Analysis with Virtuoso Spectre Simulator 210

The Noise Summary for spot noise can be displayed in a window by choosing **Results** – **Noise Summary** from the ADE. Here you can select the Spot Frequency and select *jittereventtime* to print out the summary at the selected time.

Time Domain Noise Example (continued)

Integrated Noise Summary Table



You can access the Noise Summary for *integrated noise* from the same form. You specify the range to determine the integrated noise. This is mostly derived from your output frequency range.

Transient Noise

- Transient noise is a large-signal approach.
- At each noise time point, noise is calculated for every noise source in the circuit, and is applied to the circuit.
- Noise points occur at one half the period of *noiseFmax*.
  - For example, if *noisefmax* is 1 GHz, every half nanosecond the noise is updated.
- Set *noisefmax* high enough to capture the noise in the circuit.
- If you simulate long enough to have flicker noise present, set *noisefmin* to a low frequency.
  - For example, 1 Hz.
- Always use *conservative* accuracy default.
  - If you need a lower noise floor in the transient, set *reltol* and *vabstol* smaller.
  - Remember that conservative multiplies *reltol* by 0.1.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 212

Transient Noise analysis provides the benefit of examining the effects of large-signal noise on many types of systems. It gives you the opportunity to examine the impact of noise in the time domain on various circuit types without requiring access to the Spectre RF analyses.

- Transient noise is useful mainly for systems that don't have a periodic operating point.
- It can take a very long time to run the analysis.
- Set noisefmax high enough so that all the noise is captured.
- If you want flicker noise, set *noisefmin* to a very low number (like 1 Hz) and simulate to a stop time that is long enough to see the flicker noise.
- Always use *conservative*. This is necessary to get the correct answer.
- In some cases, it might be necessary to reduce *reltol* to 1e-4. Note that conservative multiplies reltol by 0.1, so 1e-4 becomes 1e-5.

Transient Noise Options with DFT or PSD

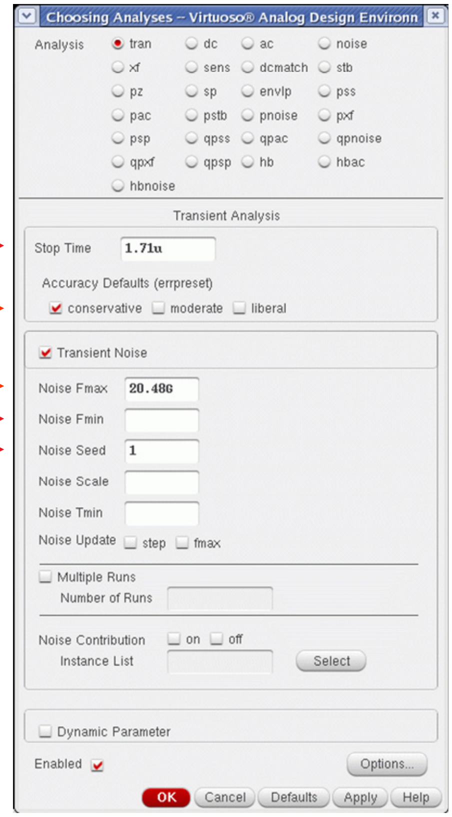
- Pick a reference frequency with an easy to remember period.
  - For example, 10 MHz.
- Set the *Stop Time* to a power of 2 periods of the reference frequency.
  - Add 1 period for startup and add a little bit after to insure there is data at the end of the sample window.
  - For example, 1.71 microseconds. (One period of 10MHz for startup, 16 periods of 10MHz, and 0.01u to make sure there is data at 1.7usec).
- Set *noisefmax* to a power of two times the reference frequency.
  - For example 20.48GHz. This is 2048*reference frequency
- Set *strobeperiod* to force time points at the noise points.
  - For example 0.1u/4096. Remember that there are 2 noise points at noisefmax.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 213

The transient noise setup form illustrating the discussed options is shown on the next page.

Transient Noise Setup

- If you are using a DFT or PSD to postprocess the data, select the stop time by allowing some time for startup and then add a power of 2 periods, and add a little bit more time to ensure that data is present.
- Always select *conservative.*
- If you are using a DFT or PSD, make *Noise Fmax* a power of 2 times the reference frequency.
- Set *Noise Fmin* only if you want flicker noise.
- Specify a Noise Seed.
- Remember to select **Options** and set strobeperiod.

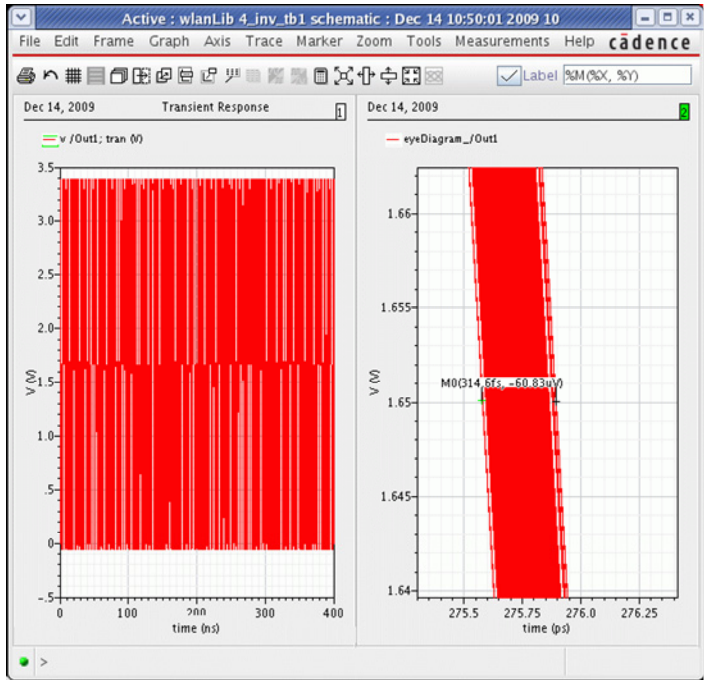


Measure Digital Jitter with Eye Diagram

Plot the time-domain waveform.

- Select the trace.
- In the waveform window, choose **Measurements – Eye Diagram**.
- Zoom in
- Position a marker at the start of the jitter.
- Select that marker.
- In the waveform window, choose **Marker – Add Delta** and select the other side of the jitter.

Remember that you are making a peak-to-peak jitter measurement. This is 6 standard deviations for 99.7% of the responses.



Jitter in the time domain is analogous to phase noise in the frequency domain. Jitter is the time-domain uncertainty, that is, small variations in the period of a waveform.

To measure the jitter in the result, use the eye diagram plot available in the Virtuoso Visualization & Analysis software from the waveform window in which the waveform signal is divided into fixed time periods, which are then superimposed on each other.

- The result is a plot that has many overlapping lines enclosing an empty space known as the eye. The quality of the receiver circuit is characterized by the dimension of the eye.
- An open eye means that the detector will be able to distinguish between 1s and 0s in its input.
- A closed eye means that a detector placed on Vout is likely to give errors for certain input bit sequences.

Power Spectral Density Function

The *power spectral density* (PSD) describes how the power of a signal is distributed with frequency.

- The PSD is computed by first breaking up the time interval into overlapping segments.
  - Each segment is multiplied, time point by time point, by the specified windowing function.
  - The DFT is performed on each windowed segment of the baseband waveform.
  - At each frequency, the DFTs from all segments are averaged together and the squared modulus of these averages gives the PSD.
- The PSD function returns volts-squared versus frequency on the waveform window. If you want dB, use the dB10 function.
- The PSD is normalized to a 1 Hertz bandwidth, so no scaling is required.

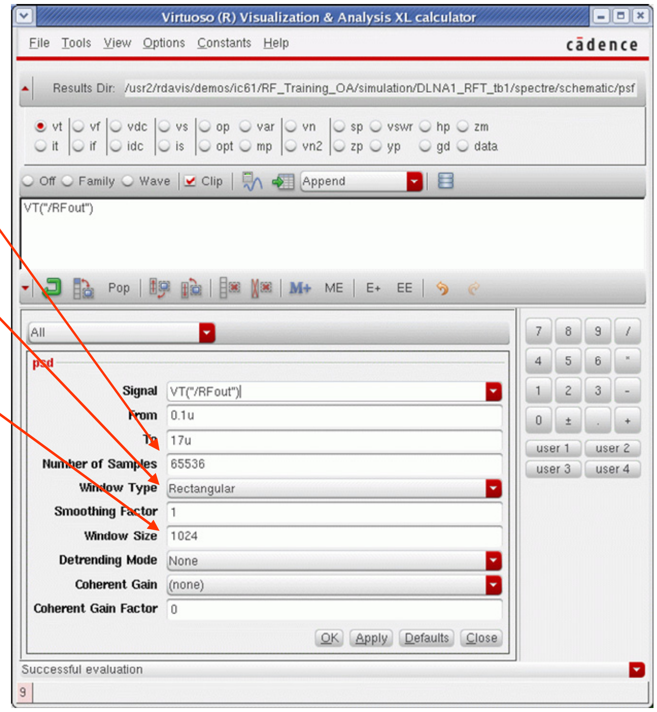
8/4/10 RF Analysis with Virtuoso Spectre Simulator 216

Mathematically, PSD is defined as the Fourier Transform (calculates DFT) of the auto correlation sequence of the time series of the signal.

PSD Function in the Calculator

PSD calculates the DFT for multiple overlapping periods of the transient waveform and then averages the result.

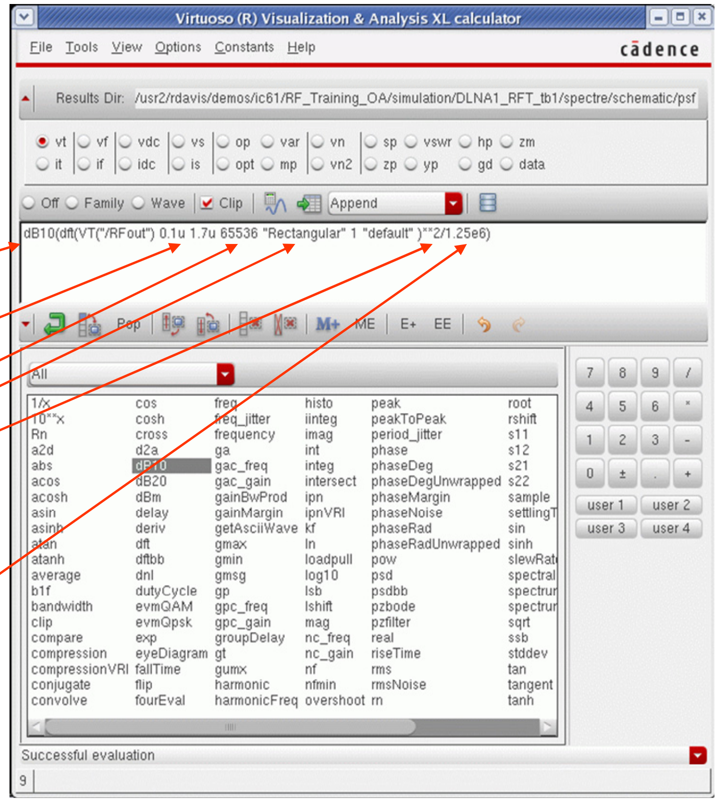
- **Number of Samples** must be a power of 2.
- Use the **Rectangular** window function.
- **Window Size** is a compromise between noise reduction and lowest frequency in the PSD result.
  - Start at about 1024 and look at the PSD waveform.
  - If it's too noisy, set Window Size smaller. If it has too much averaging, set it larger.
  - Window Size must be a power of 2.



You can calculate PSD from the waveform window by using the Calculator tool after simulation ends. You can display the calculator in ADE or from the waveform window. Select the signal whose PSD you want to calculate and then choose the PSD function and fill in the required fields.

DFT Function in the Calculator

- If you want to compare directly with the PSD or the noise result, you need to scale by the bandwidth in each frequency bin of the DFT.
- Also remember that the DFT reports peak, not rms values.



- dB10
- Sample Window
- Number of samples
- Window Function
- Squared
- Divided by 2 * bandwidth of the sample window

The Fourier Transform converts a time domain signal to the frequency domain in terms of its frequency components. The DFT is the discrete formulation of the Fourier Transform.

- The time taken to compute the DFT is proportional to the square of the number of samples in the series.
- The DFT can be calculated from the waveform window using the Calculator tool once simulation completes. You can display the calculator in ADE or from the waveform window. Select the signal whose DFT you want to calculate and then choose the DFT function and fill in the required fields.

Lab Exercises

cadence

Voltage Controlled Oscillators (VCOs)

Module 8

August 4, 2010

VCO: Shooting or Harmonic Balance?

Depending on the type of oscillators, either of the two simulation engines can be used.

- Harmonic balance (HB) is more efficient for near-sinusoidal oscillators.
  - It generates more ‘like’ sine waves
  - Recommended tuned oscillators including LC and crystal oscillators
- Shooting is more efficient for highly nonlinear oscillators.
  - It generates more ‘like’ square waves
  - Recommended for untuned oscillators including ring and relaxation oscillators
- HB gives much lower numerical noise floor.

VCO: Transient Initialization

Remember that when running a TRAN simulation, **tstab** is your best option.

- Run a transient analysis and use the waveform to estimate oscillating frequency and initial guess of the solution.
- Length of transient analysis is specified by *tstab* parameter.
- It should be long enough so that signals are mostly settled. This is usually about 100 periods for a moderate Q oscillator.
- The tstab interval can be shortened to about 10 periods by using the linear oscic.
- Use *saveinit* to save waveforms for examination. If it is long, try only to save signals that are of interest.

In HB if *tstab* is left blank or set to 0 explicitly (*tstab=0*), then no transient is performed, DC solution and input frequency is used as initial guess

- *tstab* tran might have problems with S-param data, unless the causality Correction parameter is set to Fmax.

Adjust *tstab* to avoid shooting starting near sharp transition region (some auto adjusting is done in the code).

When performing PSS analyses, remember that oscillators have no drive signal and you do not know the actual period of oscillation precisely in advance. So, you need to specify an estimate of the oscillation period and the PSS analysis computes the precise period along with the periodic solution waveforms.

A PSS analysis consists of two phases:

- An initial transient phase, which allows the circuit to be initialized.
- The shooting or harmonic balance phase, which is where the periodic steady-state solution is computed.

The transient phase further consists of three intervals.

- The first starts at tstart, which is normally 0, and continues through the onset of periodicity for the independent sources.
- The second is an optional stabilization interval, that you specify, whose length is tstab.
- In the final interval of the transient phase, the length of which is five and a half times the specified estimate of the oscillation period, the PSS analysis monitors the waveforms in the circuit and develops a better estimate of the oscillation period.

In some cases, when an autonomous PSS analysis does not converge after a few iterations, increasing the tstab interval makes convergence faster and easier.

VCO: Linear Initialization

- Oscillation frequency and amplitude are estimated based on a variation of the linear stability analysis at the DC solution.
- No trigger or initial condition value is needed and ICs should not be used.
- Effective for linear type of oscillators such as LC and Crystal.
- Turned on by the *Osc initial condition* option in the Oscillator section in PSS or HB analyses
  - By *default* regular *tstab* method is used
  - If *lin* is chosen then, linear method is used, transient analysis controlled by *tstab* is still performed using the estimated linear solution as a starting point.
    - Shooting: linear solution used as starting point
    - HB with non-zero: same as in shooting
    - HB with *tstab=0 or default tstab*, frequency and solution from linear method used directly as initial guess
- Significantly shortens initial ramp up time for high-Q oscillators

It is important to allow the oscillator to run for a while before the PSS is applied to compute the steady-state result.

- To do so, specify an additional stabilization interval using the tstab parameter. In practice, an additional stabilization interval often improves convergence, especially when simulating high-Q oscillators.
- For LC and high Q oscillators, the *linear oscic* can be very beneficial.

VCO: Not Oscillating

The following methods can be adopted to kick-start the oscillation when the circuit is not oscillating.

- Make sure *maxstep* is set somewhere close to the period divided by 10 to 20.
- Set an initial condition for a particular node or use the linear oscic.
- Set an initial condition for the inductor/capacitor in the resonator.
- Add a damped current source parallel to the resonator.
- Add a current pulse source that puts a single pulse into the circuit.

Assure that whatever method you choose to start your oscillator is effective. It should "kick" the oscillator hard enough to start the oscillation and have it respond with a signal level that is between 25% and 100% of the expected final level.

- Avoid kicking the oscillator so hard that it responds in an unnecessarily nonlinear fashion.
- Also, try to avoid exciting response modes in the circuit that are unrelated to the oscillation, especially those associated with long time constants.

Two-Tier Method

The two-tier method can be very useful for convergence when the one-tier method fails.

It's supported in HB and PSS where Harmonic Balance is selected only.

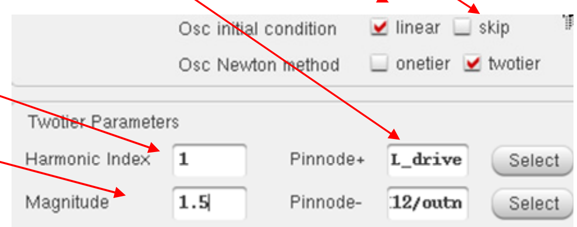
If the oscillator is a ring oscillator, select **skip**.

If the oscillator is LC, select **linear**.

If the oscillator is single ended, specify a single node inside the feedback system for Pinnode+. If it is differential, select the differential node as Pinnode-.

Set the Harmonic Index to 1.

Specify an estimate of the peak-to-peak amplitude in the magnitude. If you are unsure of the amplitude it is much better to specify a low number than a high number.

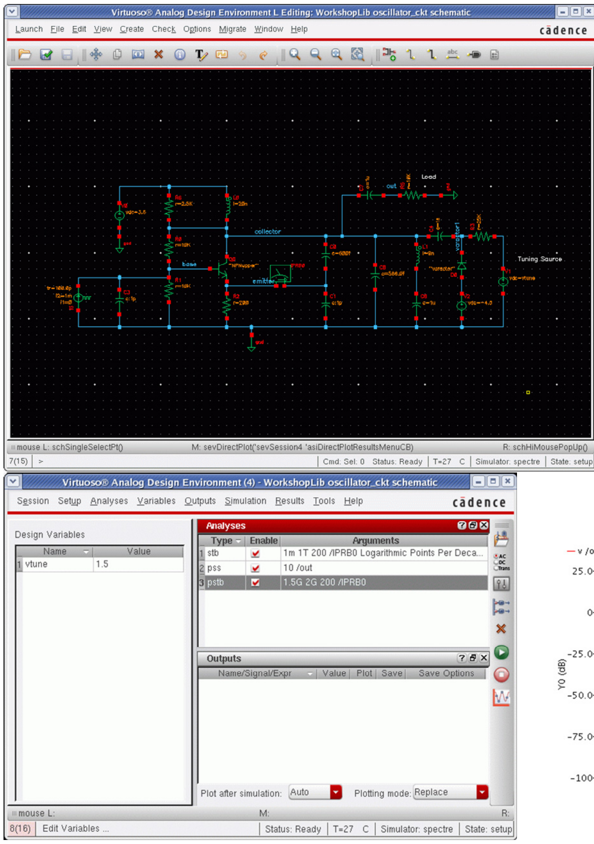


8/4/10 RF Analysis with Virtuoso Spectre Simulator 226

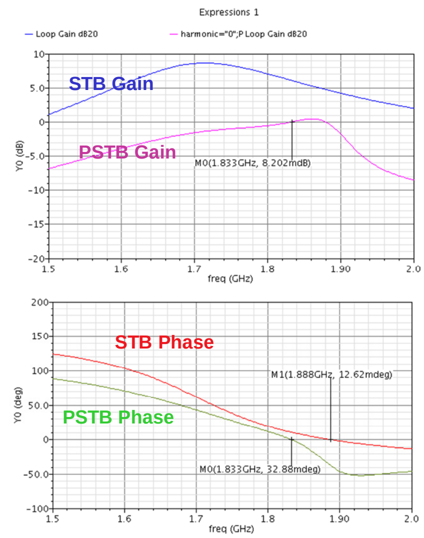
The settings for the two-tier method are shown on this page.

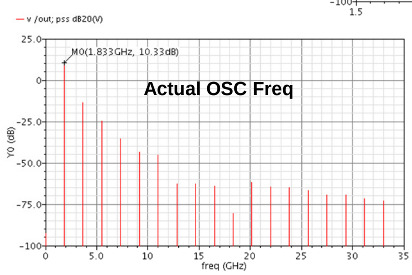
Two-tier can work very well for an oscillator that doesn't converge.

VCO PSTB: Nonlinear Periodic Stability Analysis



Gain Margin is much less than expected





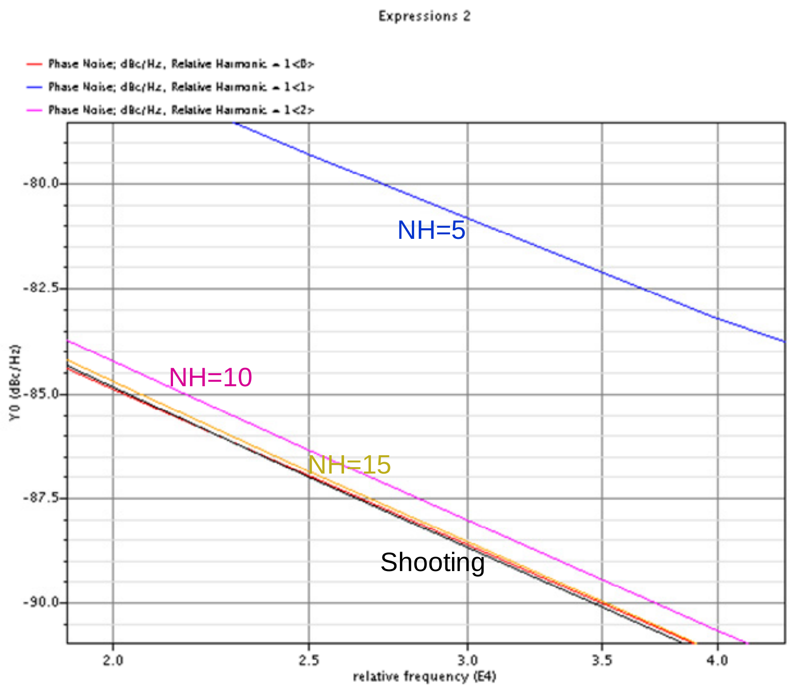
STB does not predict OSC Freq well due to nonlinearities.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 227

The Barkhausen condition for oscillation states that the effective loop gain equals unity and the phase equals 360 degrees at the oscillation frequency.

- Here you see the stability analysis results of an oscillator. Both the STB and PSTB analyses are performed.
- The two waveform graph sets are for the magnitude and phase results in dB and degrees respectively. These can be plotted using the *Direct Plot Form*.
- The spectrum at the bottom obtained from the PSS analysis shows the actual oscillation frequency to be 1.833 GHz.
- Ideally for the oscillators, the loop gain at the oscillating frequency M0 is1, so the dB20 (loopgain) and the Phase (loopgain) are both zero at M0 and the phase changes abruptly at M0. In practice, both dB20 (loopgain) and the Phase (loopgain) are close to zero, which is the case when using a PSTB analysis as shown here but the conditions don’t comply with an STB analysis.
- The results show that a PSTB analysis gives more accurate stability information for nonlinear circuits than does a STB analysis.
- The STB analysis fails to predict the behavior of periodic steady state regimes of nonlinear circuits due to the nonlinear effects these circuits produce. Periodic stability analysis (PSTB) performs stability analysis for circuits with a periodically time-varying operating point, which must first be obtained using a PSS analysis. The small-signal PSTB analysis calculates the loop gain, gain margin, and phase margin for circuits with a periodically time-varying operating point.

VCO: Phase Noise



Be sure HB has enough harmonics to settle on the solution.

Start with an estimate of the number of harmonics that are needed.

- Set max sideband equal to the number of HB harmonics.
- Run the simulation.
- Increase harms and *maxsideband* and run again. If the answer significantly changed, you need more harms. If it didn't change much, try reducing harms and *maxsideband*.

Shooting isn't subject to aliasing because it has a minimum of 200 time points in the PSS solution. This means that inherent in the shooting data is at least the 100th harmonic of the system. Because the shooting data has such high frequencies, the small-signal analyses (all of them) are not subject to aliasing.

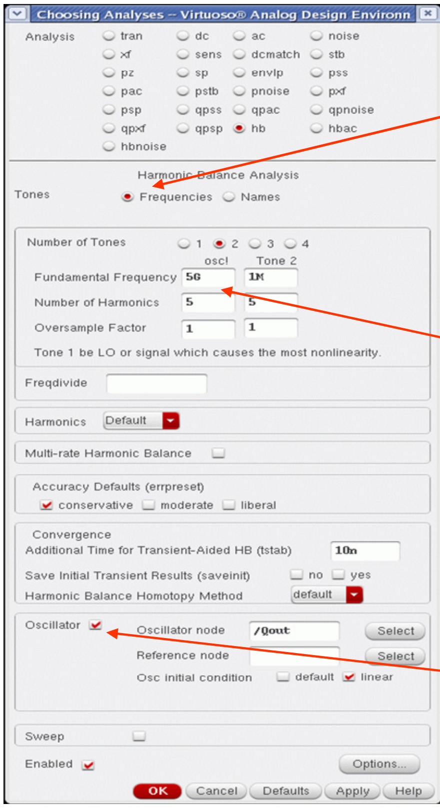
Semi-Autonomous HB Setup

- Semi-Autonomous setup allows the simulation of driven oscillators with one or more periodic inputs. Now the algorithms can handle two typical driven oscillator scenarios namely
  - Oscillator-Mixer with RF tone simulation
  - Oscillator with power supply ripple simulation
- You get a full large-signal solution for all the tones.
- It is a frequency domain simulation technique and is supported in the **hb** *(*select from *Choosing Analyses* form*) analysis* only.
- You can follow the hb analysis with various small-signal analyses (hbac, hbnoise) after the hb runs.

The semi-autonomous simulation is for oscillators that also have driven inputs.

You can use the method to simulate oscillators and mixers together, or simulate oscillators perturbed by periodic external interferences, such as power supply interference.

Semi-Autonomous Choose Analysis Form

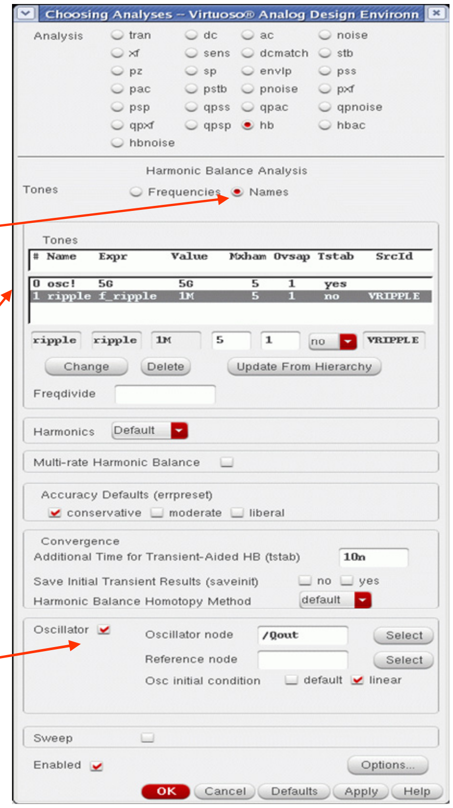


Set Tones to Frequencies.

Set Tones to Names.

Set Osc! To an estimate of the oscillation frequency.

Push the oscillator button to enable.



8/4/10 RF Analysis with Virtuoso Spectre Simulator 230

If the oscillator doesn't oscillate in the semi-autonomous simulation, try to increase the tstab time.

You can use the *saveinit=yes* parameter to save the tstab waveforms.

If the oscillator doesn't oscillate in the tstab transient simulation, give it a “kickstart” by using the IC method, or by using the linear initial condition (*oscic=lin*).

Lab Exercises

Lab 8-1 Running Oscillator Simulations in the Turbo Mode and APS Mode

Lab 8-2 Analyzing Linear and Periodic Stability

Lab 8-3 Analyzing Semi-Autonomous Harmonic Balance with HB Noise

Lab 8-4 Applying Parasitic Reduction

Lab 8-5 Putting It All Together: Simulating a Full Receiver Chain

Blank Page

cadence

Envelope Analysis

Module 9

August 4, 2010

The RF Envelope Analysis: ENVLP

- Efficient and accurate analysis and prediction of the envelope transient response of modulated signals in RF circuits is important when simulating communications systems.
- Envelope analysis reduces simulation time without compromising accuracy by exploiting the property that the behavior of a circuit in a given high frequency clock cycle is similar, but not identical, to its behavior in the preceding and following cycles.
  - The envelope of the high-frequency clock can be followed by accurately computing the circuit behavior over occasional cycles, which accurately captures the fast transient behavior.
  - The slow varying modulation is accurately followed by a smooth curve.
  - The spectrum of the circuit response can be obtained by combining the spectrum of the smooth curve and the spectrum of occasional clock cycles.

Traditional transient analysis is inefficient for the resolution of low modulation frequencies in the presence of a high carrier frequency because the high-frequency carrier forces a small time step while the low-frequency modulation forces a long simulation interval.

The ratio between the lowest frequency in the modulation and the frequency of the carrier is a measure of the relative frequency resolution required of the simulation.

Applications of Envelope Analysis

The applications of envelope analysis include:

- Predicting the spectral regrowth of amplifiers and mixers
- Designing feedback loops such as *automatic gain control* (AGC) loops
- Predicting the transient behavior of switched capacitor filters
- Simulating large transients in phase lock loops
- Helping the oscillator designer identify the load pull effect for the communication systems with VCO and power amplifier

The clock is normally the most rapidly changing signal in the circuit and thus causes the most nonlinearity. It is referred to differently in different applications.

- For mixers, the clock is called the LO.
- For detectors, the clock is called the carrier.
- For switched-capacitor filters, the clock is called the clock.

8/4/10 RF Analysis with Virtuoso Spectre Simulator 235

Remember that envelope analysis is not designed to simulate circuits having a filter with nodes that have higher frequencies than clock. A transient analysis is usually faster for these circuits.

The Time Domain Shooting Envelope Algorithm

- The high frequency clock (such as LO) in RF circuits usually causes the most nonlinearity in the circuit response.
  - Envelope analysis samples the circuit waveforms at the clock frequency, and assumes the resulting *envelope* can be accurately represented by a piecewise polynomial.
- Shooting envelope is a variation of transient as such that it skips transient cycles, so the simulation runs faster.
- Shooting starts with several cycles of transient, then when the circuit becomes predictable with a low order polynomial, it begins to skip cycles of the input signal.
- The number of cycles skipped depends on the curvature of the waveform.
  - In places where the curvature is high, relatively few cycles are skipped.
- Shooting can be used for startup waveforms where long time constants are present, such as crystal oscillators.

Time Domain Shooting ENVLP integration is similar to transient integration.

The Harmonic Balance Envelope Algorithm

- Harmonic balance is usually much faster than shooting, but it requires a driven circuit. You can't see the startup waveform using harmonic balance because it is a steady-state analysis at each timepoint in the envelope.
- Harmonic balance has a variable or fixed timestep. Fixed timestep is faster and just as accurate.
- When using the fixed timestep algorithm set the step period to the interval between samples in the modulation file.
  - It performs a harmonic balance simulation at each timestep, and at the end you can see the overall spectrum.
- HB envelope analysis samples the modulation envelope in time and outputs a time-varying spectrum for different time steps. The spectrum is converted to time domain waveforms in each clock cycle.
  - In the first several clock cycles, a transient analysis is used in single carrier HB Envelope, as it is in Shooting Envelope.
  - For multi-carrier HB Envelope, HB QPSS is adopted to get the steady state at the beginning. Then the time domain data is converted to frequency domain

For linear and nearly linear circuits, HB Envelope is more efficient than shooting Envelope analysis.

In shooting and single carrier HB Envelope analysis, the clock is pointed directly. In multi-carrier HB Envelope analysis, the first one of the fundamentals is regarded as the clock.

General Setup Notes

- For computing the ENVLP response of a circuit, you need to specify the analysis *clockname*. This should be the largest amplitude signal in the circuit.
- The simulator automatically determines the clock period by looking through all the sources with the specified *clockname*.
  - 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 or determined by *tstab* if not otherwise given.
  - If *flexbalance* is yes, HB ENVLP is used, otherwise shooting ENVLP will be used. The default value is *no*.
- If you want to see the phase of the signal in the time-domain result, take the frequency represented by the delta-T in the I and Q modulating piecewise linear waveform file and multiply it by an integer to set the carrier frequency.
  - This is not required for harmonic balance, but this is required if you use shooting because shooting can only skip an integer number of cycles.

The procedure for setting up an envelope analysis is similar to the set-up procedure for transient analysis.

As is true for *periodic steady-state* (PSS) analysis, you can use envelope analysis for particular classes of circuits operating with multiple clock fundamentals.

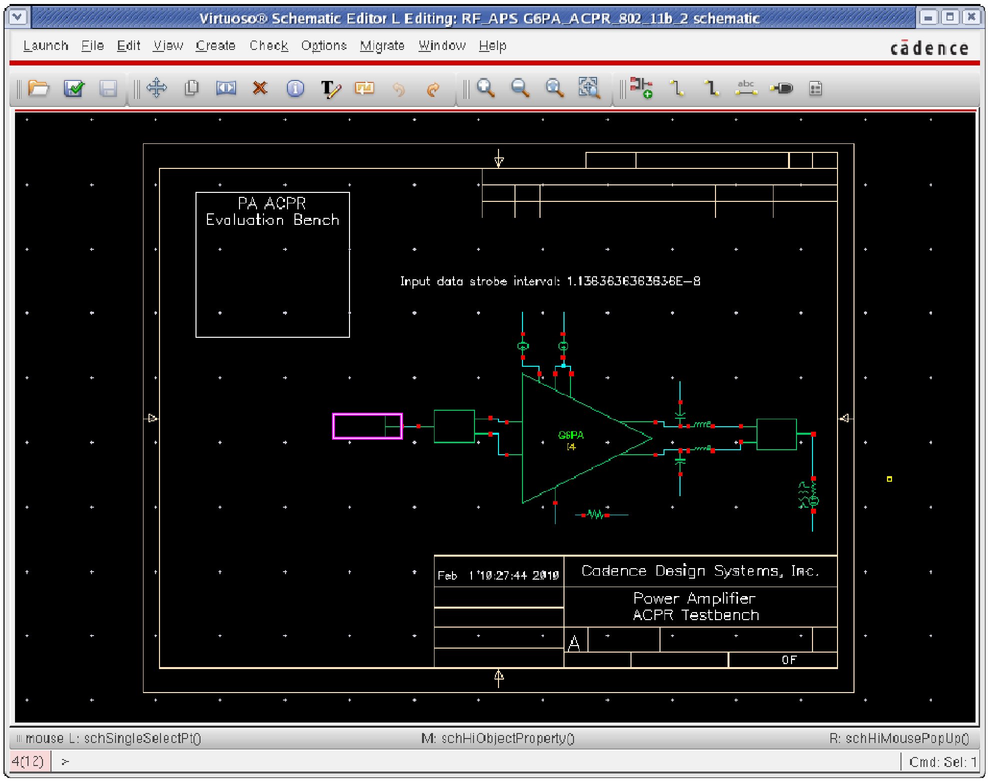
- For PSS analysis, the multiple fundamentals are commensurate.
- For envelope analysis , you can use the greatest common denominator of all fundamental clock frequencies as the clock beat frequency.

The process of selecting the clock beat frequency for envelope analysis is similar to the process of figuring out the beat frequency for PSS analysis.

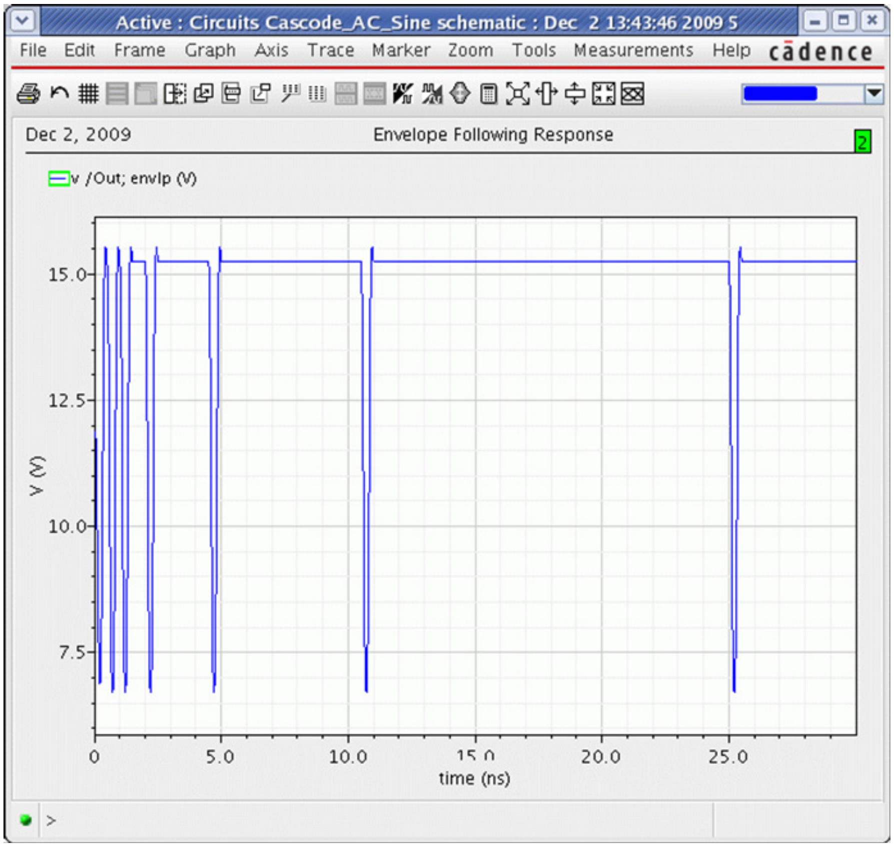
ACPR_Source in RF_Examples Library

ACPR_Source has been added to the rfExamples library.

The ACPR_Source is a behavioral modulator that you just need to supply with an I and Q PWL file.



Shooting Envelope Simulation At Startup

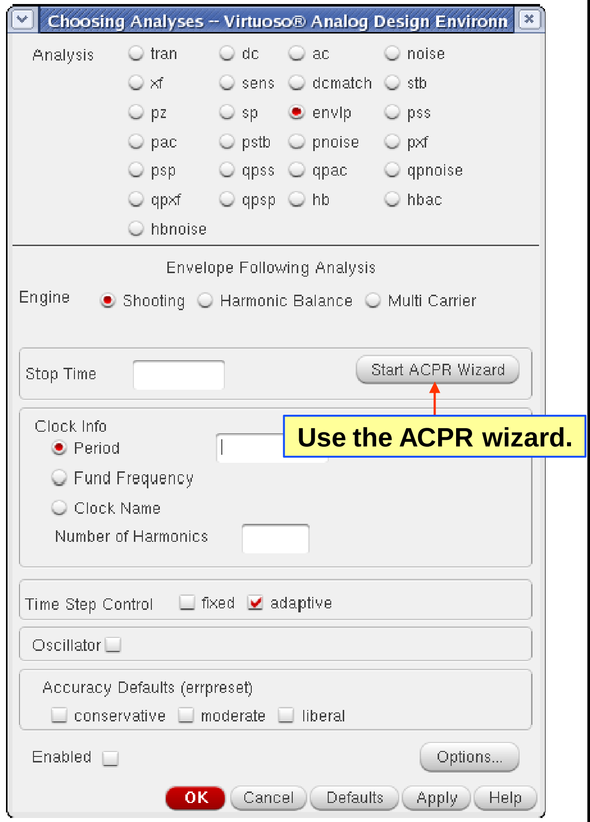


Note the cycles being skipped as the simulation progresses.

Remember that an envelope analysis might be considered as a fast transient analysis whose efficiency comes from skipping simulation cycles. However, when an envelope analysis cannot find cycles to skip, the analysis effectively reduces to a transient analysis.

The ACPR Simulation

- ACPR (Adjacent Channel Power Ratio) is a common measure of how much power a transmitter emits outside its allotted frequency band.
- It is the ratio of the power in an adjacent band divided by the power in the allotted band. In measuring ACPR it is crucial to drive the transmitter with the proper baseband signals.
  - Regardless of exactly how you chose the frequencies and bands for the ACPR measurement, it is always extracted from the power spectral density (PSD) of the transmitted signal.
  - The PSD is a frequency-by-frequency average of a set of DFTs (discrete Fourier transforms) of the baseband signal. Here, the baseband signal is the harmonic-time result of an ENVLP analysis.

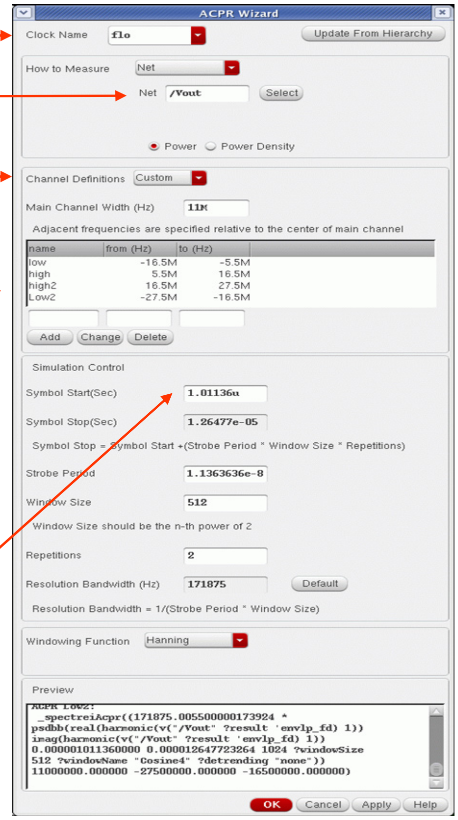


The ACPR Wizard helps you through the complex process of measuring *adjacent channel power ratio* (ACPR) and *power spectral density* (PSD). It is available when you select the **envlp** option in the Choosing Analyses form.

- The Envelope Choose Analysis form is similar to the transient Choosing Analyses form. Most ENVLP analysis parameters are inherited from either transient or PSS analysis and their meanings are consistent.
- For more information on envelope analysis parameters, refer to the *Virtuoso Spectre Simulator RF Analysis User Guide*.

Using the ACPR Wizard

- Select the name of the RF source.
- Select the output net(s).
- If you have standard CDMA, select IS-95. If you have wideband CDMA, select W-CDMA. Otherwise, select **Custom**.
- If you select custom, define the frequency ranges for the ACPR measurement. This is referenced to the center frequency of the channel.
- Many I and Q signals are digital filtered and need some time to start. Set **Symbol Start** after the modulation has started and make it agree with one of the times in the modulation file.



The following couple of slides explain the setup of the *ACPR Wizard* form.

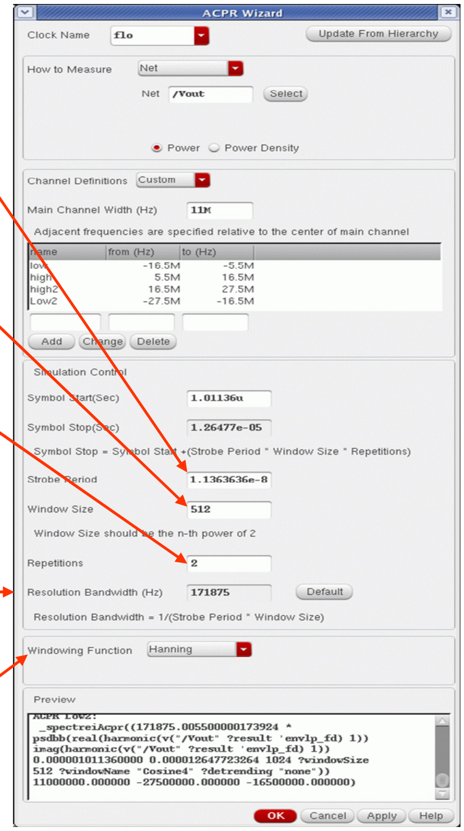
In the channel definitions section when you select IS-95,

- The Main Channel Width (Hz) field is calculated. This is the width of the main channel in Hz.
- The adjacent frequencies are determined and displayed in the list box. Note that adjacent frequencies are specified relative to the center of the main channel.

Symbol Start (sec) and *Symbol Stop* (sec) specify the starting and ending points of the time domain waveform to be used for DFT analysis. Symbol stop is calculated based on the strobe period, window size, and repetitions.

Using the ACPR Wizard (continued)

- Strobe period is the time interval between the points in the modulation file.
- Window size specifies how many points in the modulation file to include in the PSD. It should be a power of 2.
- The PSD is inherently noisy. Setting a repetition factor reduces the noise at the cost of simulation time.
- The stop time and the resolution bandwidth are calculated based on the symbol start, strobe period, window size, and repetitions.
- Because the data is random, you need to select a Windowing Function.

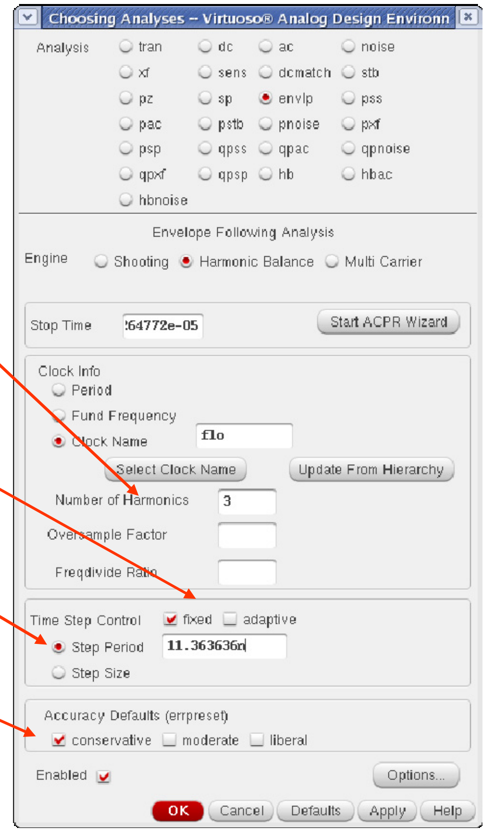


The *Repetitions* field specifies the number of times to repeat the DFT for averaging. When you increase the number of repetitions, the PSD curve is smoother but the simulation time is longer and the data file is larger.

A chosen *Windowing Function* tapers the signal before performing the DFT to reduce the effect of any edge discontinuities.

Settings in the *envlp* Choosing Analyses Form

- Set a reasonable number of harmonics.
- For HB, choose Fixed time Step Control. For SN, choose adaptive.
- If you use the ACPR wizard, the Step Period is set automatically.
- Choose a desired accuracy.

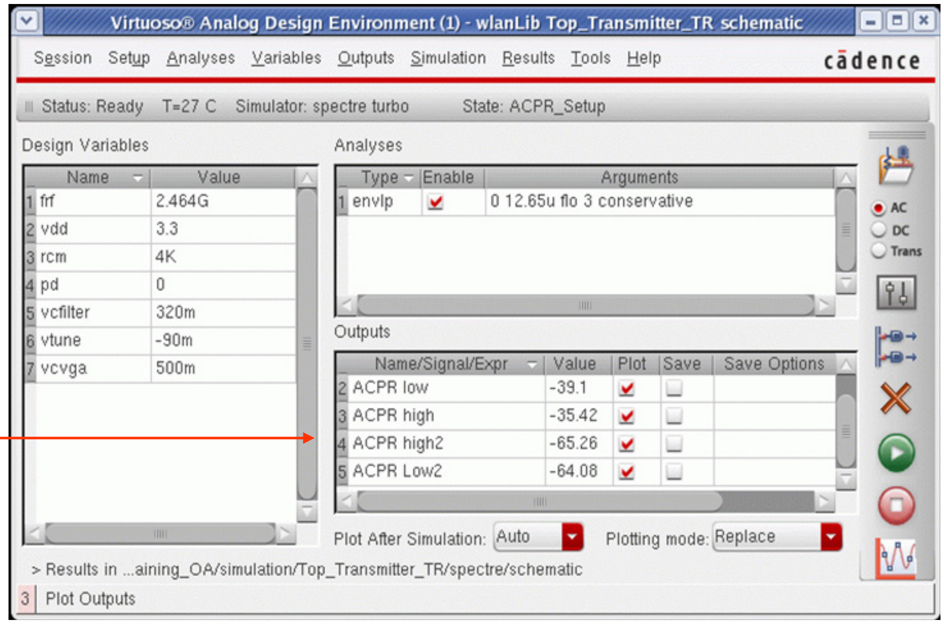


You can select either the Harmonic Balance or Shooting engine. Harmonic balance is usually faster.

- The default value of Number of Harmonics chosen is 3. The best value to use depends on the linearity in one cycle of the fast signal (LO or clock). For linear cases, a value of 1 is enough for accurate results. However, for nonlinear cases, such as a circuit with a square clock, even a value of 15 might not be large enough.
- Here you can select the *Clock Name* and the simulator automatically calculates the frequency from the clock or you can specify the fundamental frequency or period.
- A *conservative* setting produces a larger simulation time.

ACPR Is Shown in ADE After Simulation

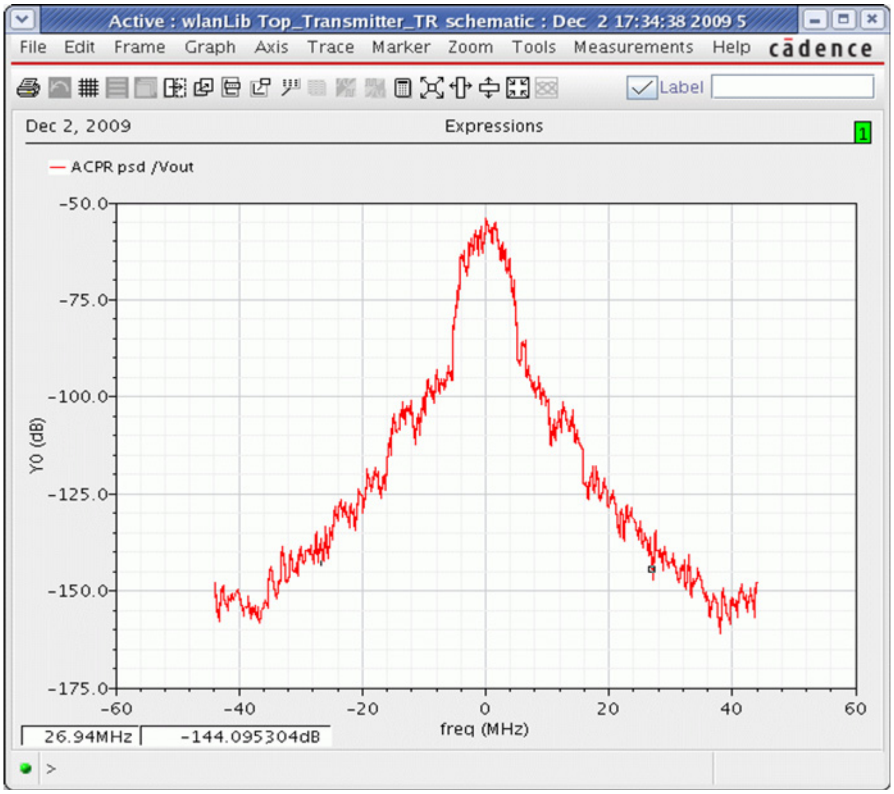
The ACPR is in the Outputs section in ADE after the simulation finishes.



8/4/10 RF Analysis with Virtuoso Spectre Simulator 245

When the envelope analysis finishes, the ACPR values display in the Outputs area in the ADE window. The PSD is displayed in the waveform tool.

PSD After the Simulation

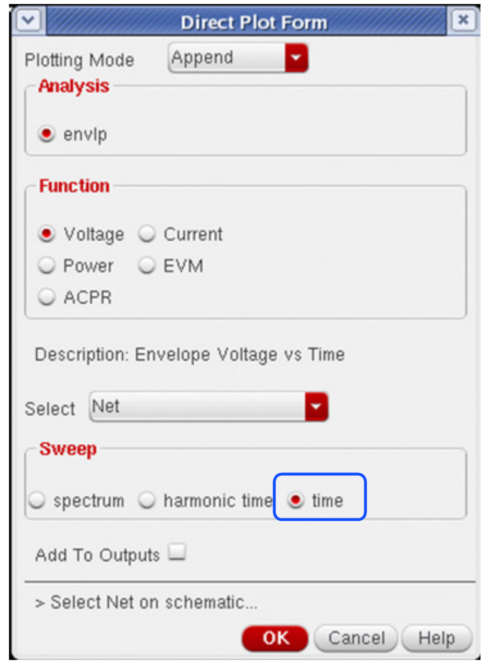


When the simulation is run with the ACPR wizard, the PSD is calculated and displayed.

Direct Plot Form for the Time-Domain Result

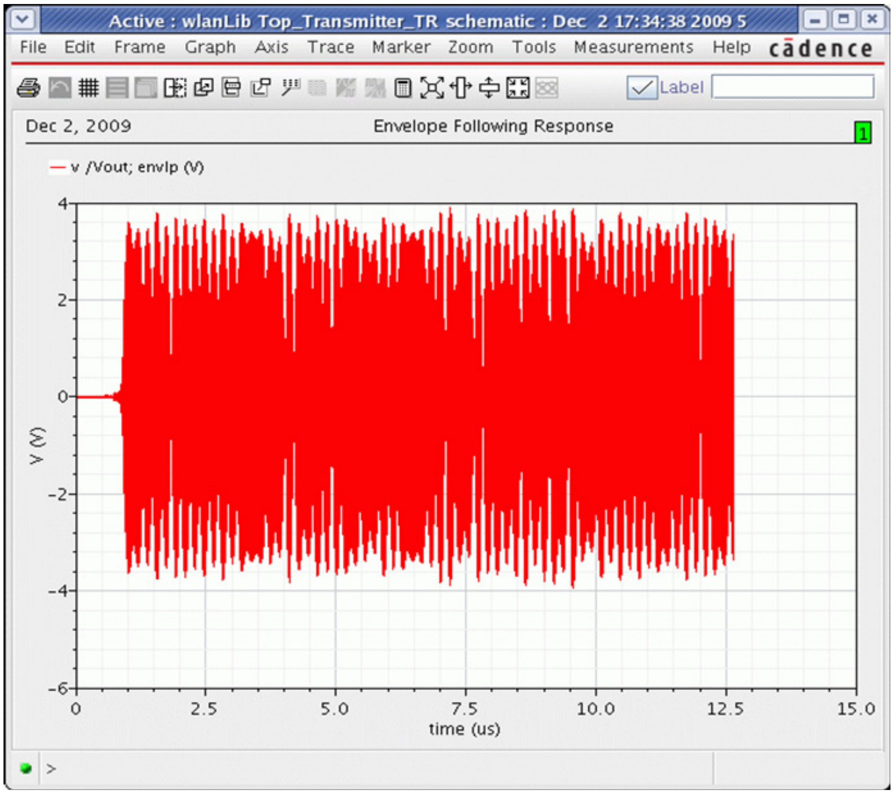
Choose **Results–Direct Plot–Main form** from the ADE window.

Select *time* as the sweep parameter.



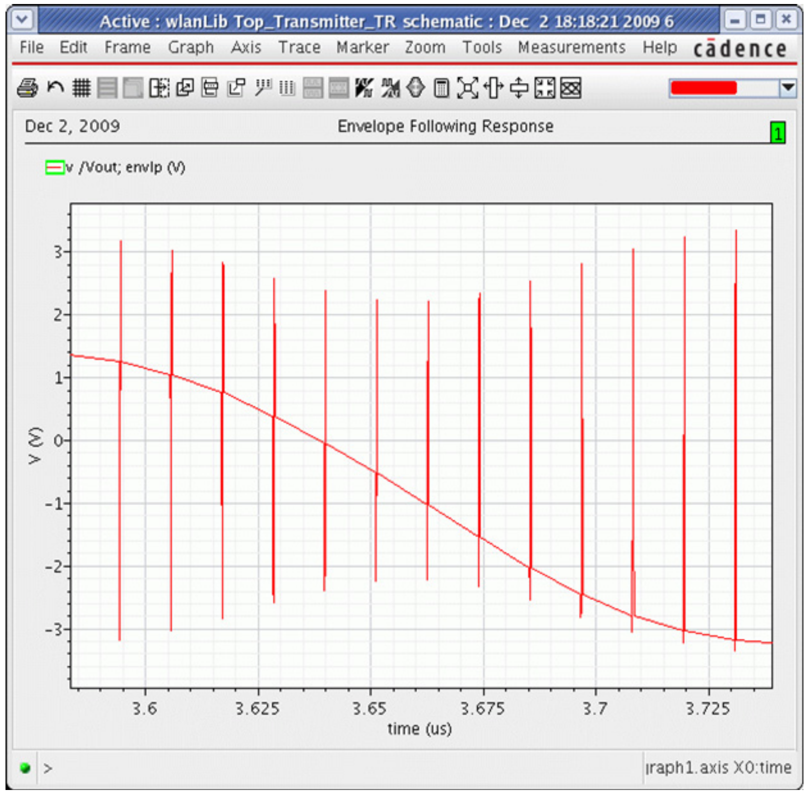
You can also plot the voltage envelope results in the time domain using the *Direct Plot Form*.

Time-Domain Result: Zoomed Out



Time-Domain Result: Zoomed In

You can see the phase of the signal by looking at the line between the sinusoidal sections that look like spikes in this view.



Lab Exercises

Lab 9-1 Running Envelope Analysis