SpectreRF Workshop

VCO Design Using SpectreRF

MMSIM 7.1

November 2008

Contents

VCO Design Using SpectreRF

The procedures described in this workshop are deliberately broad and generic. Your specific design might require procedures that are different from the ones described in this application note.

Purpose

This workshop illustrates how to use SpectreRF in the Virtuoso Analog Design Environment (ADE) to measure parameters that are important in the design verification of voltage controlled oscillators (VCOs).

Audience

Users of SpectreRF in the Analog Design Environment.

Overview

This application note provides you with a basic set of common measurements for VCOs.

Introduction to VCOs

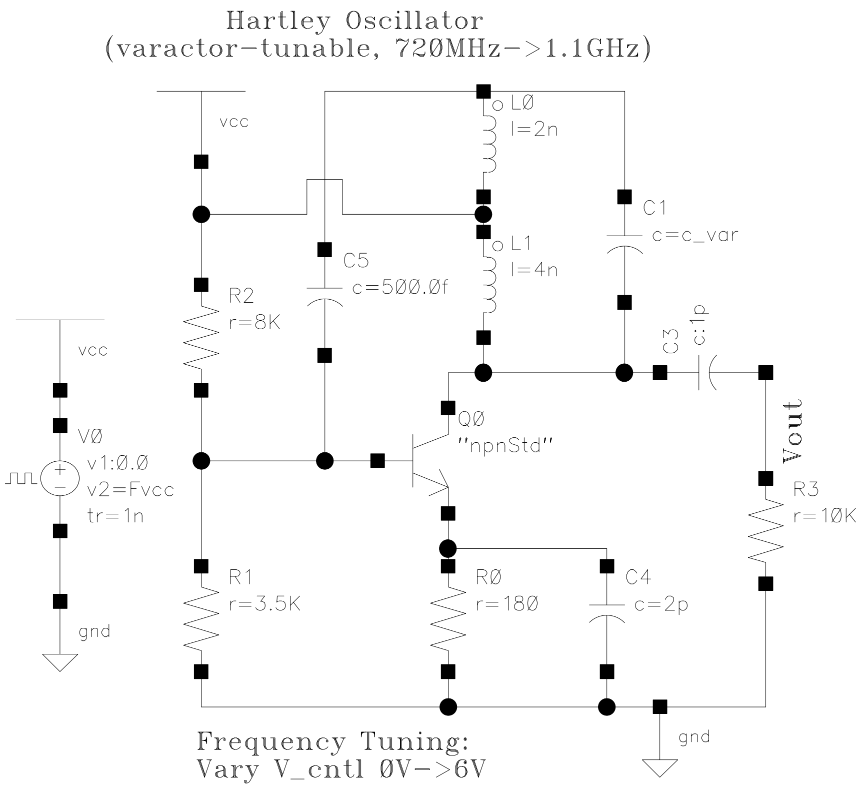
Oscillators generate a reference signal at a particular frequency. In voltage controlled oscillators, or VCOs, the frequency of the output varies in proportion to some control signal. Oscillators are generally used in RF circuits to generate the local oscillator, or LO, signal for mixers. VCOs are used in both receivers and transmitters.

The noise performance of a mixer is strongly affected by noise on the LO signal. The LO signal is always passed through a limiter, which is generally built into the mixer, to make the mixer less sensitive to small variations in the amplitude of the LO signal. Oscillators, except for reference oscillators, are embedded in phase-locked loops to control PLLs frequency and reduce their phase noise. Reference oscillators are generally fixed-frequency crystal oscillators, and as such have well controlled frequency and noise.

However, oscillators still produce enough variation in the phase of their output to affect the performance of the transceiver. Thus, it is important to minimize the phase noise produced by the oscillator.

The Design Example: oscHartley

The VCO measurements described in this workshop are calculated using SpectreRF in the Analog Design Environment. The design investigated is the Hartley oscillator shown below:



The oscHartley VCO uses the basic Hartley topology and is tunable between 720 MHz and 1.1 GHz. The oscillation frequency (Fo) is determined by the resonant circuit made up of inductors (L0, L1) and the C1 capacitor. In this particular VCO, the values of L1 and L2 are fixed whereas the value of C1 is variable.

In this example, the resonant circuit's capacitor C1 serves as a varactor diode. As a result, the varactor diode's junction capacitance, C_{var} , is a function of the applied voltage as shown in the following equation.

$$C_{var} = \left(\frac{C_{j0}}{\left(1 + \frac{V}{\phi}\right)} \right) \gamma$$

Where

- V = applied junction voltage (V)
- C_{j0} = junction capacitance (F) for V = 0 V
- f = barrier potential (V)
- γ = junction gradient coefficient

The varactor diode for this VCO has the following values

- C_{j0}=8 pF
- f = 0.75 V
- γ = 0.4

Because C_{var} is inversely proportional to V, and Fo is inversely proportional to C_{var}, the oscillation frequency is proportional to V. In other words, as you increase V, C_{var} decreases and Fo increases.

Example Measurements Using SpectreRF

To achieve optimal circuit performance, you should measure and evaluate several VCO characteristics or parameters under varying conditions. As an example, one fundamental measurement is the plot of VCO output frequency versus tuning voltage. An extension of this parameter is tuning sensitivity (expressed in Hz/V), which is the differential of the output frequency versus tuning voltage curve. The slope change as a function of frequency is a critical design parameter.

In practice, both of these parameters should be evaluated under different supply (Vcc) conditions because the output frequency may shift with Vcc changes. This DC power sensitivity is called *frequency pushing*.

The RF power output is a function of both Vcc and output frequency. You should evaluate the RF power output because an output power level that is too low results in excessive noise and an output power level that is too high creates distortion and consumes excess DC power. Moreover, the DC power has the chance to translate Vcc noise into oscillator output modulation and noise. Many of the parameters you must consider for a complete evaluation are not covered in this workshop.

In this workshop, you begin your examination of the flow by bringing up the Cadence Design Framework II environment for a full view of the reference design:

Action P-1: Move into the **./rfworkshop** directory.

Action P-2: Run the **icfb&** tool.

Action P-3: In the CIW window, select **Tools — Library Manager…**.

Lab 1: Output Frequency, Output Power, Phase Noise, and Jitter Measurement (Pnoise with Shooting or Harmonic Balance Engine)

Usually you cannot specify an analysis period for an autonomous circuit because you do not know the precise oscillation period in advance. To that end, you can estimate the oscillation period and have SpectreRF compute the correct period. The output power of a VCO is typically expressed in dBm.

Phase noise is random phase variation in the output oscillating signal of the VCO. Close to the carrier, phase noise is mainly composed of flicker noise. The flicker noise measured in a VCO is generated only by the active devices, such as the transistor and the tuning diode. The phase noise is measured at distances from 1 KHz off the carrier to several megahertz (MHz) off the carrier in a 1-Hz bandwidth. Phase noise is the ratio of the output power divided by the noise power at a specified value and is expressed in dBc/Hz. Phase noise is the most significant source of noise in oscillators, which makes it a crucial measurement.

Jitter is the measure of an uncertainty in the output of the oscillator or fluctuations in the timing of events. In oscillators and frequency synthesizers, jitter affects sensitivity and selectivity. In RF systems, jitter causes an increase in the channel separation. SpectreRF determines autonomous jitter by first determining noise versus frequency with correlations (modulated Pnoise). Then the integration is done using a user-defined sample frequency point as an input. As a result, SpectreRF runs two Pnoise analyses (modulated Pnoise analyses) to extract jitter information.

Action 1-1: In the Library Manager window, open the *schematic* view of the design *oscHartley* in the library *RFworkshop*.

To start some oscillator circuits, you apply initial conditions or kick start the oscillator. This is not necessary for the *oscHartley* circuit used in this lab, because circuit contains a pulse source that kick starts the oscillator. Therefore, no real initial conditions are necessary.

Action 1-2: In the Virtuoso Schematic Editing window, select **Tools — Analog Environment**.

Action 1-3: (Optional) Choose **Session — Load State** and in the Virtuoso Analog Design Environment window, select **Cellview** in **Load State Option** and load the state “**Lab1_Pnoise_shooting**”, then skip to Action 1-12.

Action 1-4: In the Analog Design Environment window, select **Analyses — Choose…**.

Action 1-5: In the Choosing Analyses window, select **pss** in the **Analysis** field of the window. Set **Fundamental Frequency** to **600 MHz** based on estimation. Set the **Number of Harmonics** to **3** for this oscillator.

Set the **tstab** value to add additional time for the oscillator to converge on its operating frequency. Click **yes** to save the initial transient results. Enable **Oscillator.** When activated, the **Oscillator** option tells the simulator to run an autonomous circuit and to treat the specified Fundamental Frequency of 600 MHz as an estimate. To set the Oscillator node and reference node, click the select buttons and choose the nodes indicated in the oscHartley schematic.

Action 1-6: Click **Options** at the bottom of the Choosing Analyses form.

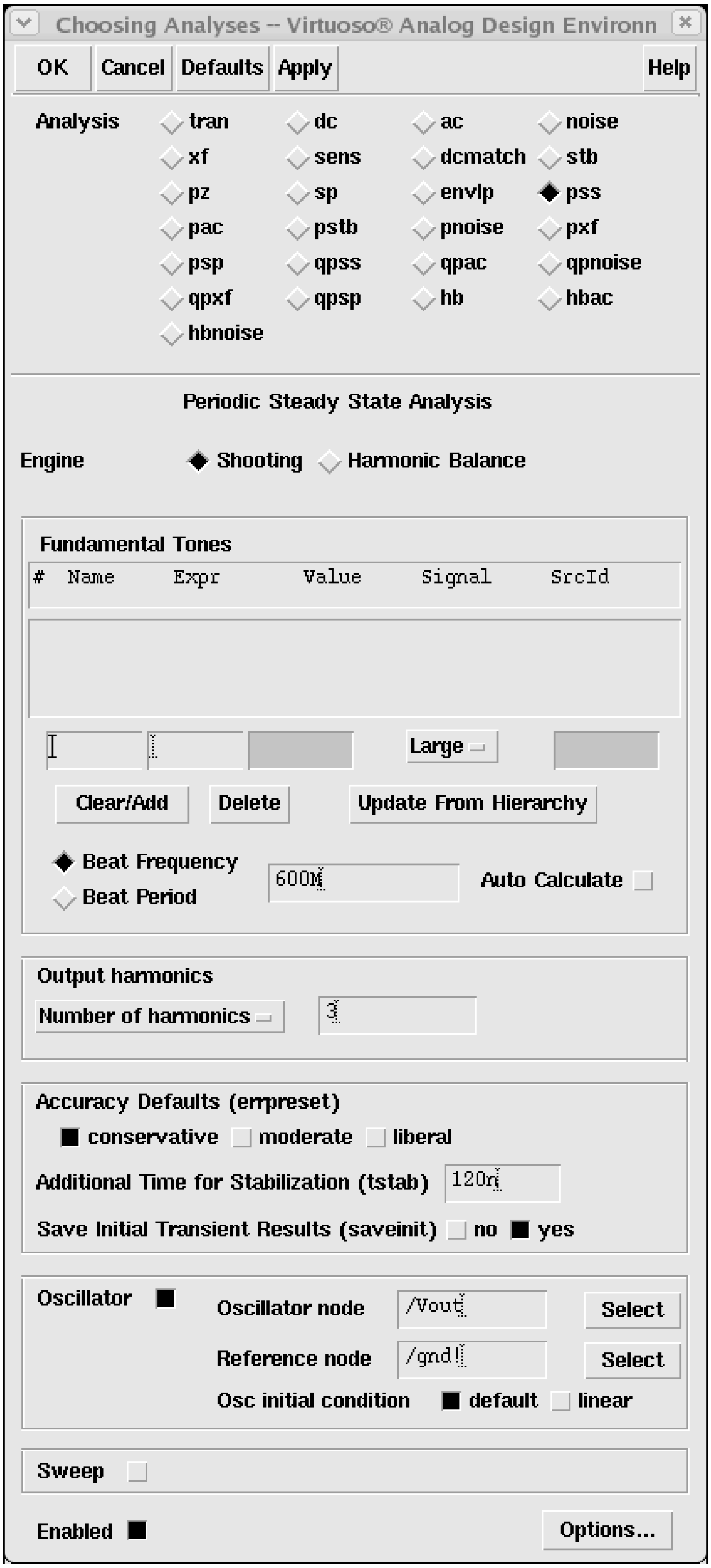
The Periodic Steady State Options form appears.

Action 1-7: In the INTEGRATION METHOD PARAMETERS section, verify that the method is set to **gear2only**.

Action 1-8: Click **OK** in the Periodic Steady State Options form.

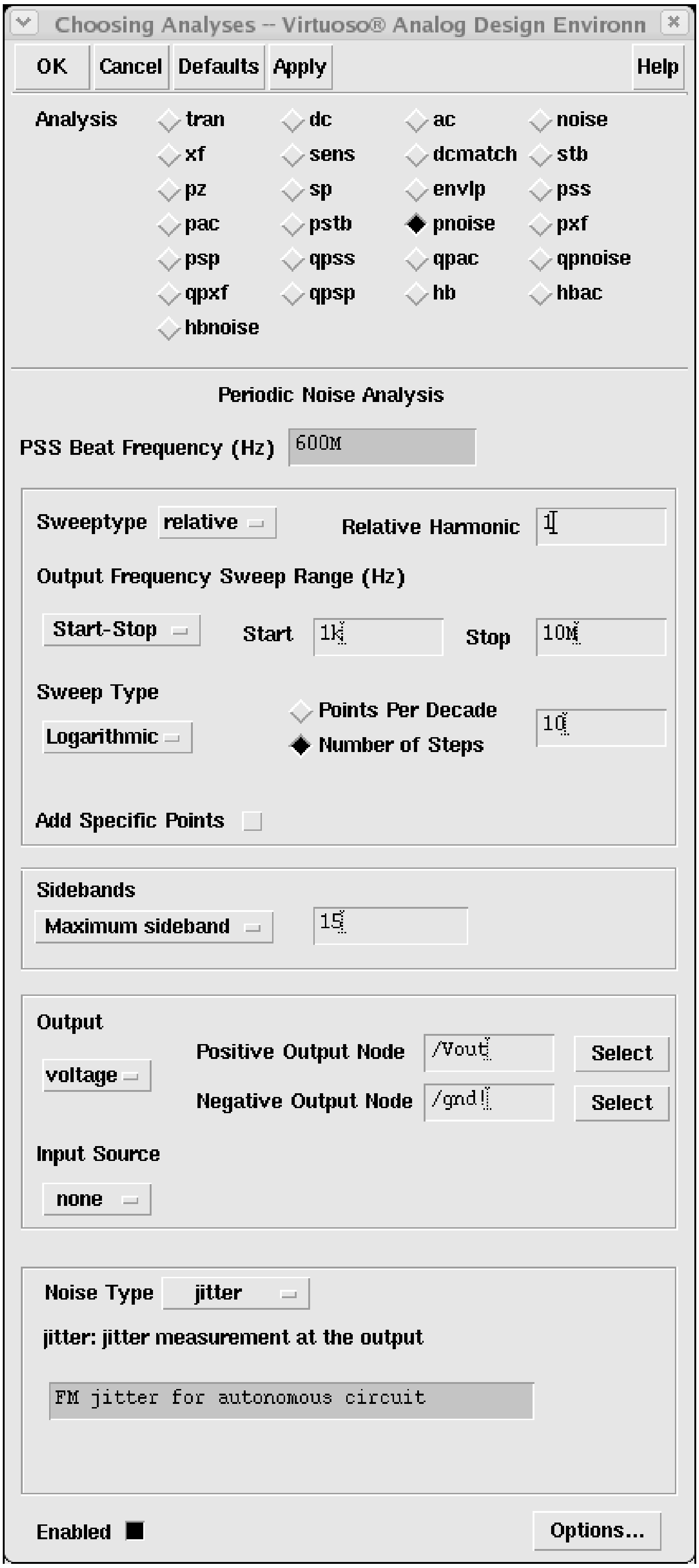
Action 1-9 Click **Apply** in the Choosing Analyses form.

The Periodic Steady State Analysis form looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environn
OK
Cancel
Defaults
Apply
Help
Analysis
tran
dc
ac
noise
xf
sens
dcmatch
stb
pz
sp
envlp
pss
pac
pstb
pnoise
pxf
psp
qpss
qpac
qpnoise
qpxf
qpsp
hb
hbac
hbnoise
Periodic Steady State Analysis
Engine
Shooting
Harmonic Balance
Fundamental Tones
Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Hierarchy
Beat Frequency
Beat Period
600M
Auto Calculate
Output harmonics
Number of harmonics
3
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Additional Time for Stabilization (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Oscillator
Oscillator node
/Vout
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Sweep
Enabled
Options...

Action 1-10: Click **pnoise**. In the Periodic Noise Analysis form, set **Sweeptype** to **relative**, and **Relative Harmonic** to **1**. Set **Start** to **1K** and **Stop** to **10M**. Set the **Sweep Type** field to **logarithmic** and **Number of Steps** to **10** (You may want to have more steps for accuracy). Set the **Maximum sideband** to **15**. Set **Output** to **voltage**. Select the Positive and Negative nodes in the oscillator schematic. Set **Input Source** to **none**.

The form looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environn
OK
Cancel
Defaults
Apply
Help
Analysis
tran
dc
ac
noise
xf
sens
dcmatch
stb
pz
sp
envlp
pss
pac
pstb
pnoise
pxf
psp
qpss
qpac
qpnoise
qpxf
qpsp
hb
hbac
hbnoise
Periodic Noise Analysis
PSS Beat Frequency (Hz)
600M
Sweeptype
relative
Relative Harmonic
1
Output Frequency Sweep Range (Hz)
Start-Stop
Start
1k
Stop
10M
Sweep Type
Logarithmic
Points Per Decade
Number of Steps
10
Add Specific Points
Sidebands
Maximum sideband
15
Output
voltage
Positive Output Node
/Vout
Select
Negative Output Node
/gnd!
Select
Input Source
none
Noise Type
jitter
jitter: jitter measurement at the output
FM jitter for autonomous circuit
Enabled
Options...

A Pnoise Analysis is set up to run after PSS has calculated the steady-state oscillation frequency. The phase noise from 100 Hz to 10 MHz, *relative to the derived oscillation frequency*, is calculated. Because this analysis is for an autonomous circuit, *Sweep type* defaults to **relative**. For driven circuits, *Sweep type* defaults to **absolute**.

For a typical bipolar oscillator, the phase noise is specified at 10 kHz off the carrier. The sweep limits should include the lowest offset frequency of interest, but at the frequencies close to the LO the small signal approximation breaks down and the information at those frequencies is not valid. That depends on the Q of the oscillator and on the presence of the flicker noise. Another factor could be the bandwidth of the circuit itself. The highest limit is usually on the order of f(LO)/2 to f(LO) because we are only interested in the phase noise around the output harmonic. The bandwidth of the circuit is often the factor here too. More frequency points are always helpful to increase an accuracy of the jitter computations. You have to trade off the frequency range, number of points per decade and the maximum sidebands that are used for folding in PNoise.

The Sidebands field is set to a **Maximum sideband** of **15**. In this case, you are interested in the up-converted 1/f device noise to the oscillation frequency, which is manifested as phase noise. To account for higher harmonics of the oscillator that also contribute noise, change this value.

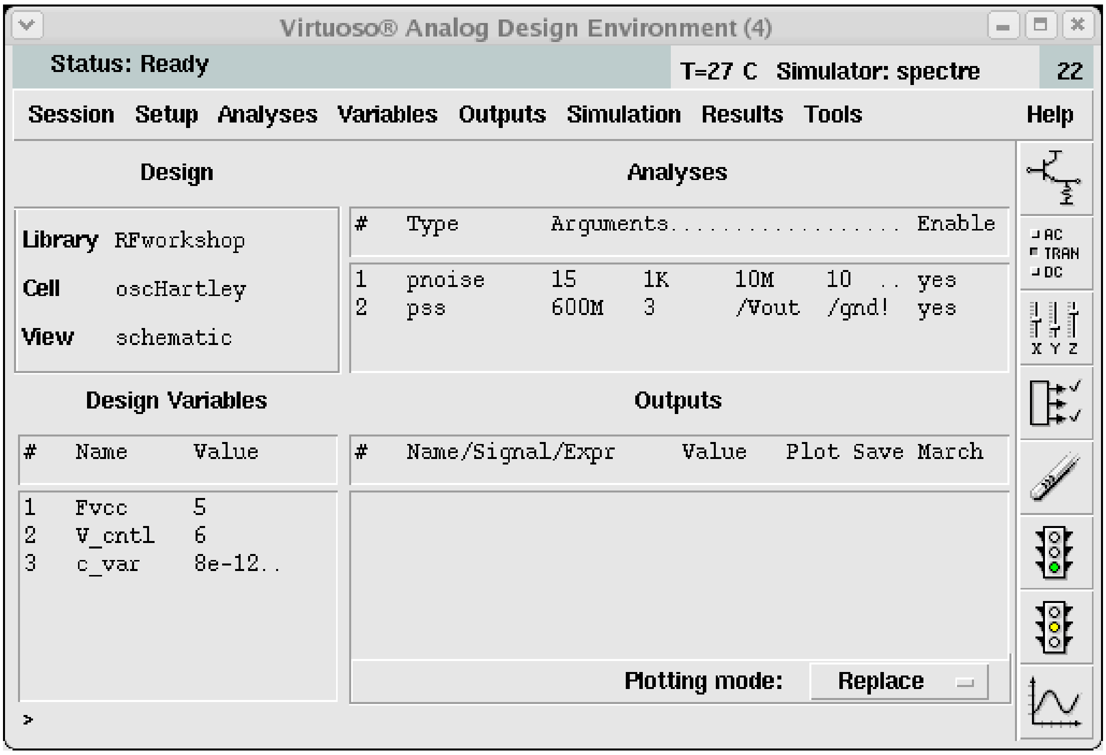
Because the up-converted noise appears at the oscillator output, the *Vout* node was selected as the **Positive Output Node** in this analysis. The noise power of each noise contributor in the circuit is stored, but the “output node of interest” needs to be specified to tell the Virtuoso® Spectre® RF software where to sum the noise powers.

No **Input Source** is specified. For an oscillator, the noise comes from the autonomous circuit itself, not from a driven source such as *port* from *analogLib*.

The Pnoise output is used to compute the jitter. Because we are using the phase noise to calculate the jitter, the netlist has two pnoise analyses in it. For pnoise jitter for autonomous, the pnoise modulated analysis runs in the background. Consequently, pnoise modulated results are available in the direct plot form.

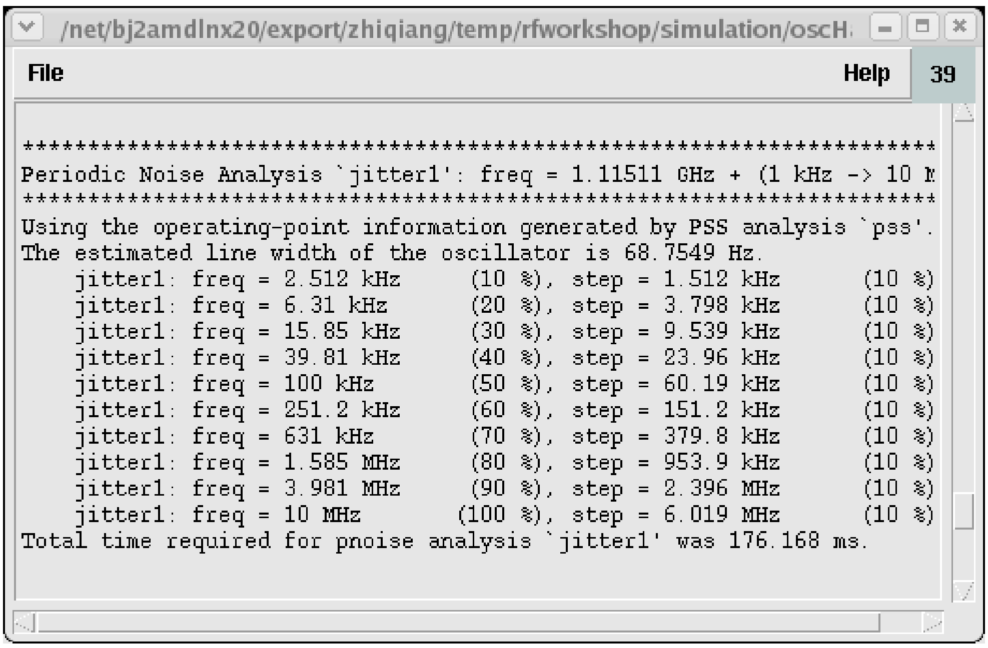
Action 1-11: Turn on **Enabled**. Click **OK** in the Choosing Analyses form.

The Virtuoso Analog Design Environment window looks like this:



Action 1-12: Choose **Simulation — Netlist and Run** to start the simulation or click the **Netlist and Run** icon in the Virtuoso Analog Design Environment window.

The output log file displays the following message:

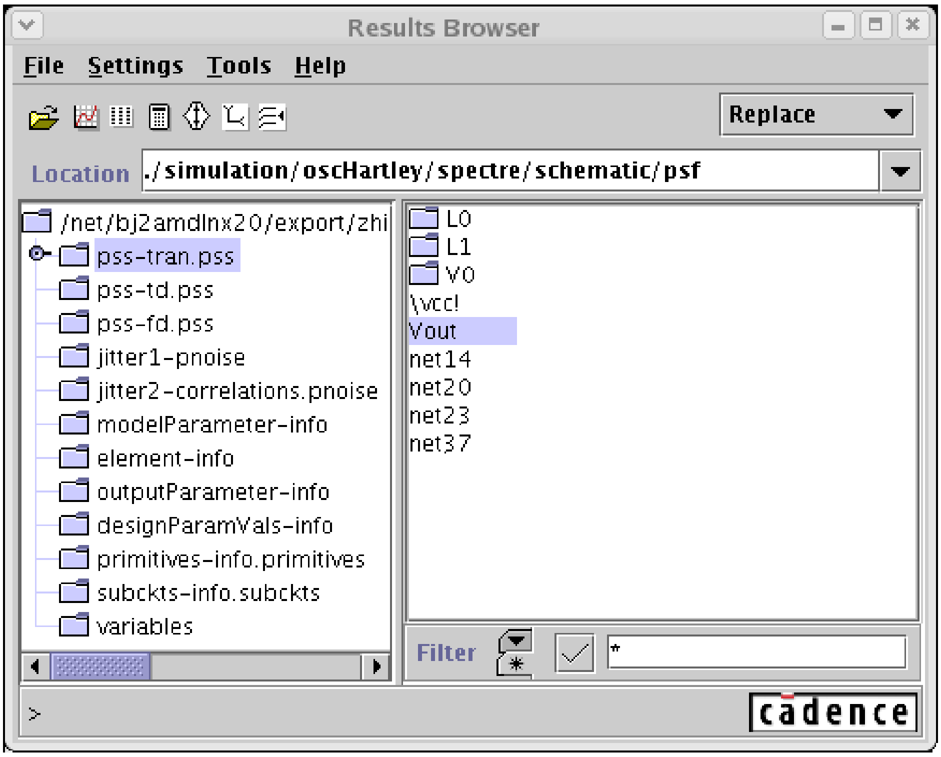


SpectreRF now enhances noise computation to compute the corner offset-frequency for oscillators. In this case, the corner frequency 68.75Hz is calculated.

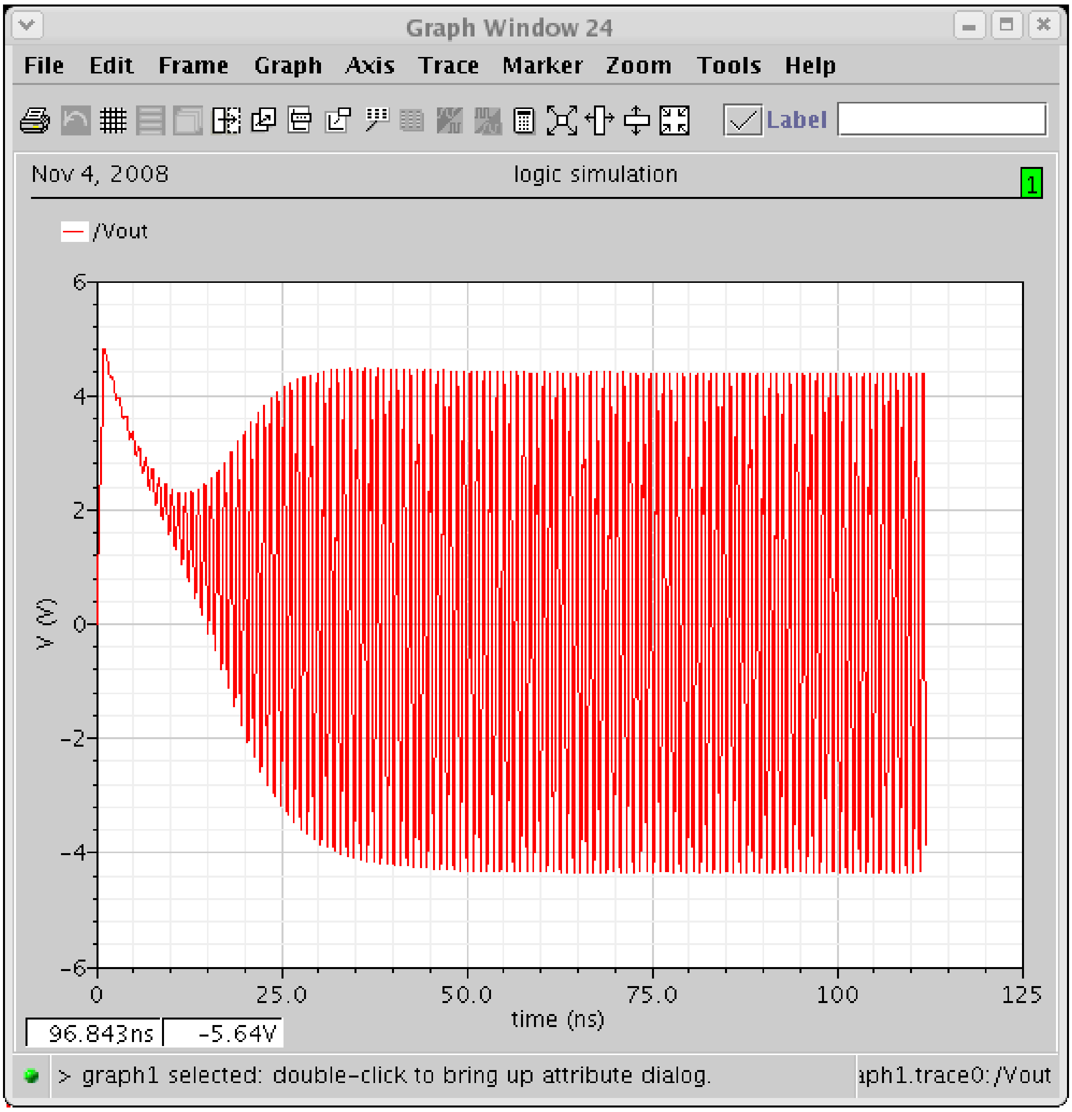
After the simulation finishes, the following actions plot the simulation results.

Action 1-13: In the Virtuoso Analog Design Environment window, choose **Tools — Results Browser**.

Action 1-14: In the Results Browser form, expand **pss-tran.pss.**



Action 1-15: Double click Vout. The *Vout* transient node voltage appears in the waveform window.

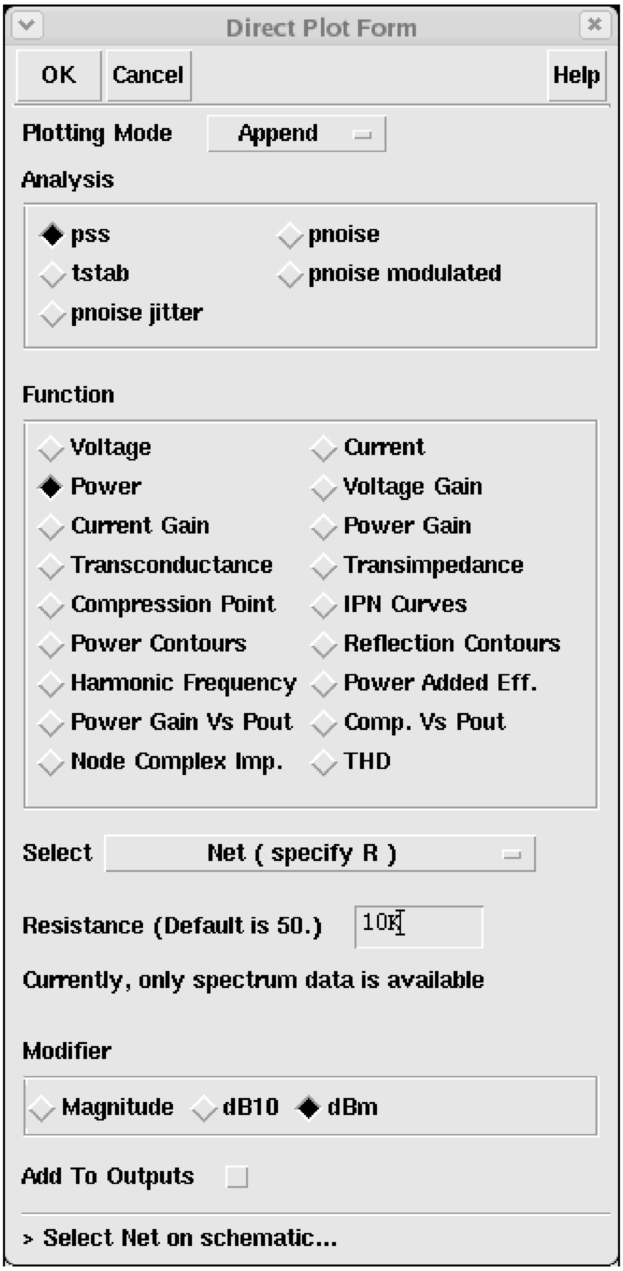


Action 1-16: Close the waveform window.

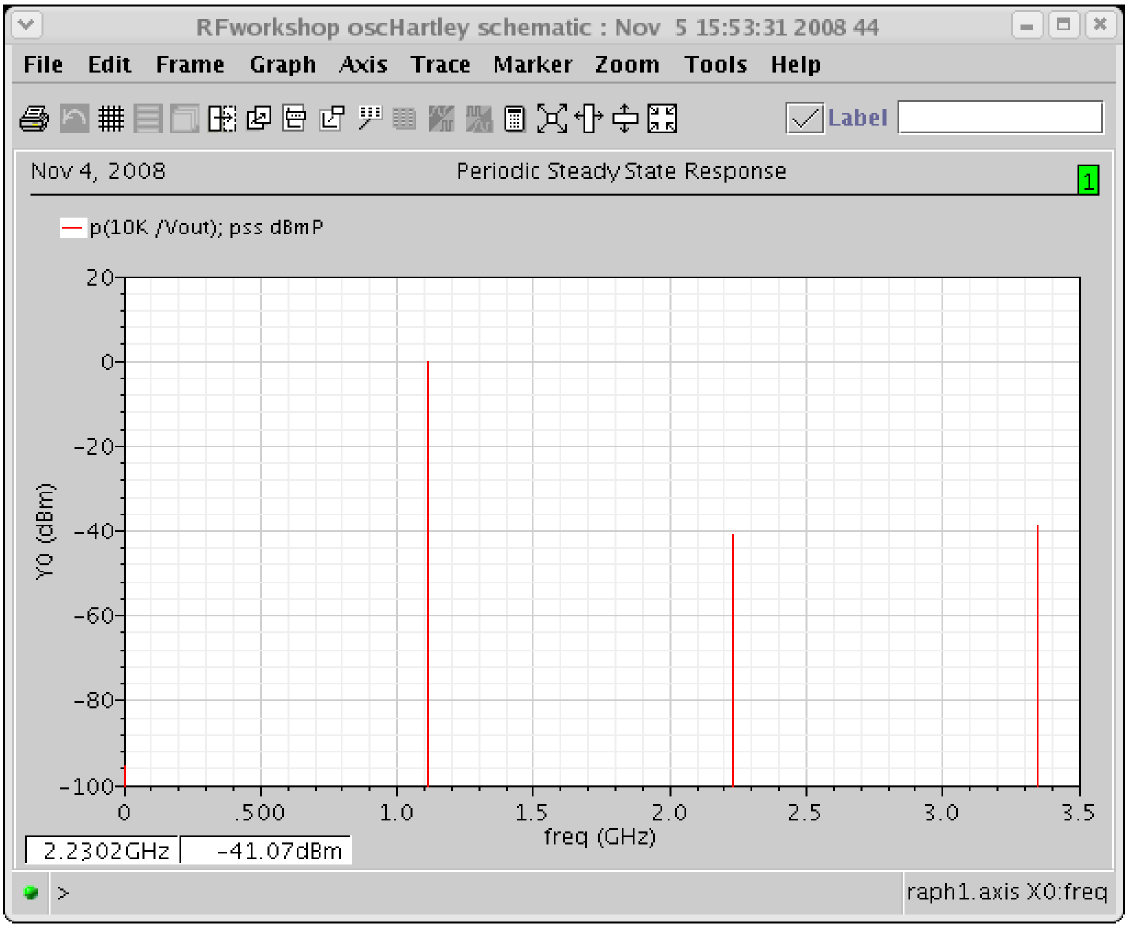
Action 1-17: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form** to plot the calculated oscillation frequency, output power and output noise.

The Direct Plot form appears.

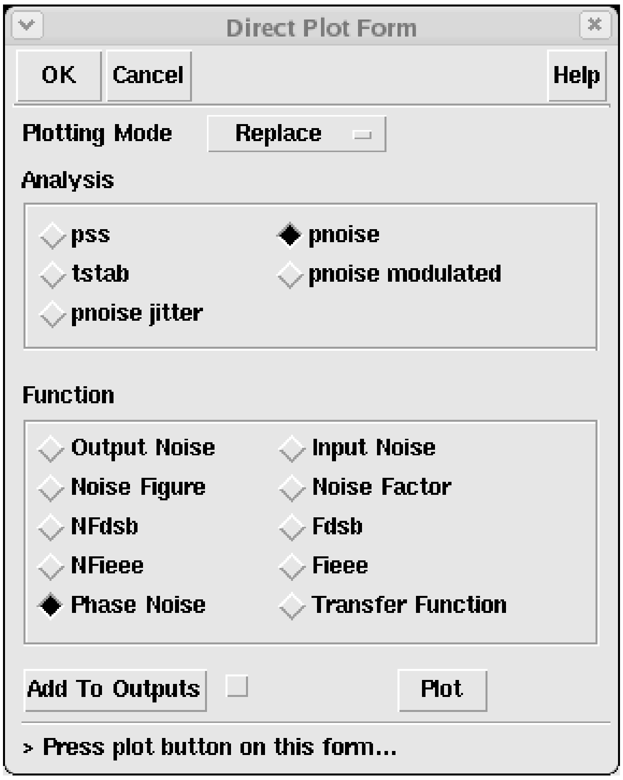
Action 1-18: In the Direct Plot Form window, choose **pss** as the **Analysis** type and configure the form as follows:



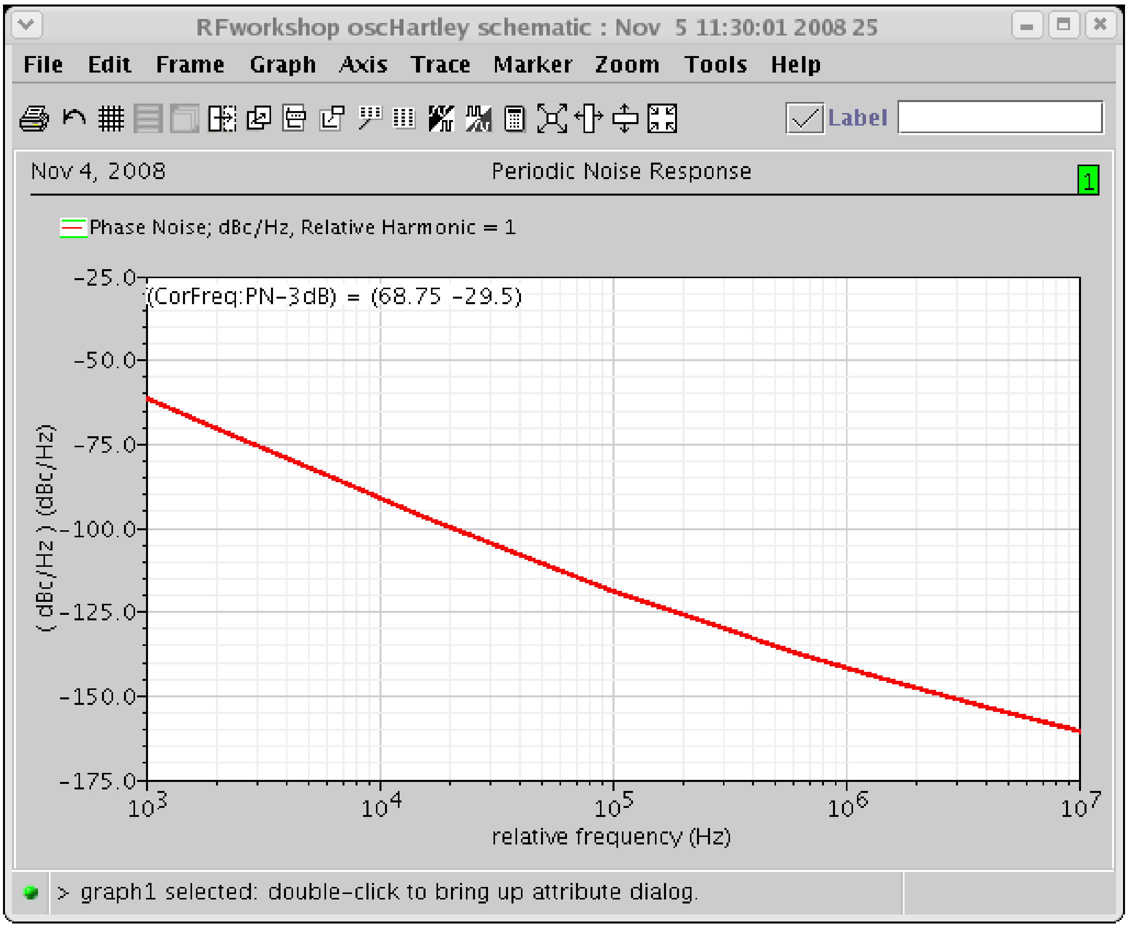
Action 1-19: Select net **Vout** on the schematic. The results show the oscillation frequency is around 1.1G, and output power is around 0 dBm.



Action 1-20: In the Direct Plot form, change the **Plotting Mode** to **Replace** and the Analysis type to **pnoise**. Change the **Function** to **Phase Noise**, and click **Plot**.



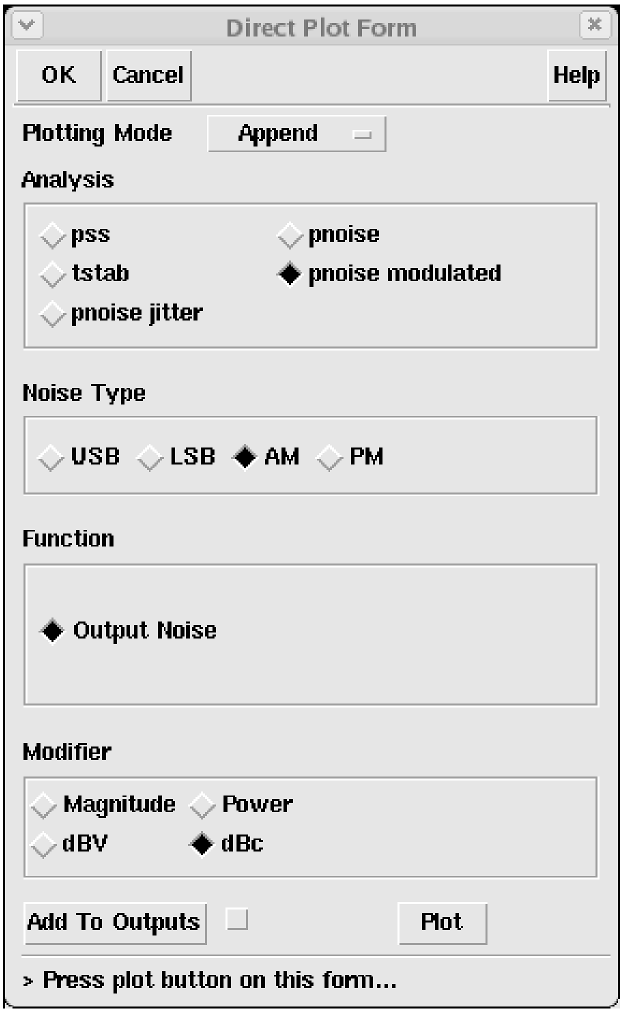
Now you get the following waveforms:



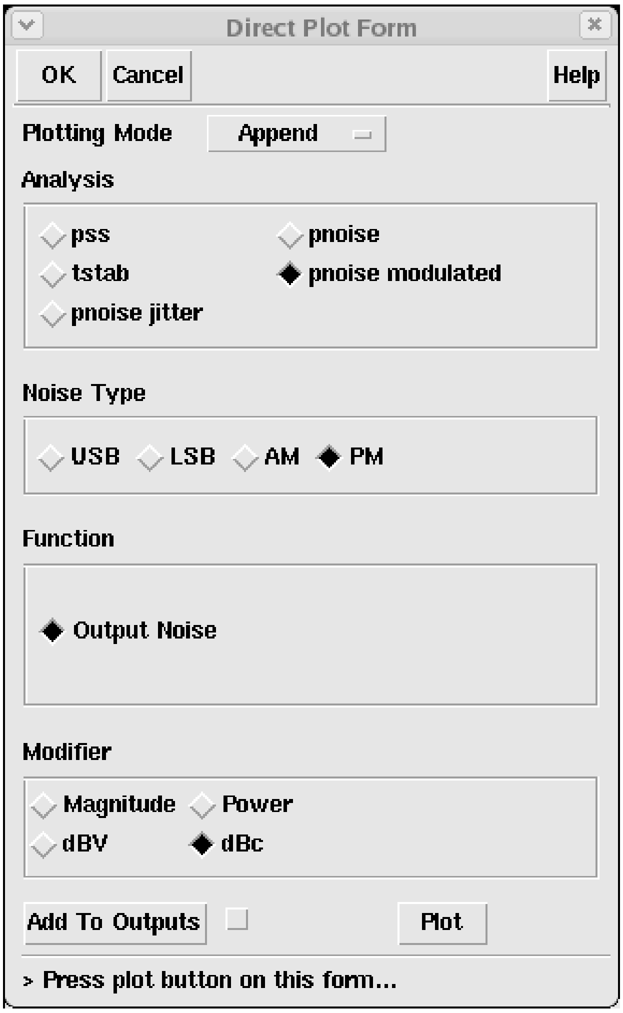
SpectreRF determines autonomous jitter by first determining noise versus frequency with correlations (modulated Pnoise). When running pnoise modulated, SpectreRF computes the correlation between sidebands by using pnoise=correlation in the background. Using SpectreRF Pnoise modulated analysis, you can fully characterize the AM, PM, USB, and LSB components of the noise.

Action 1-21: In the waveform window, click the **Add Subwindow** icon.

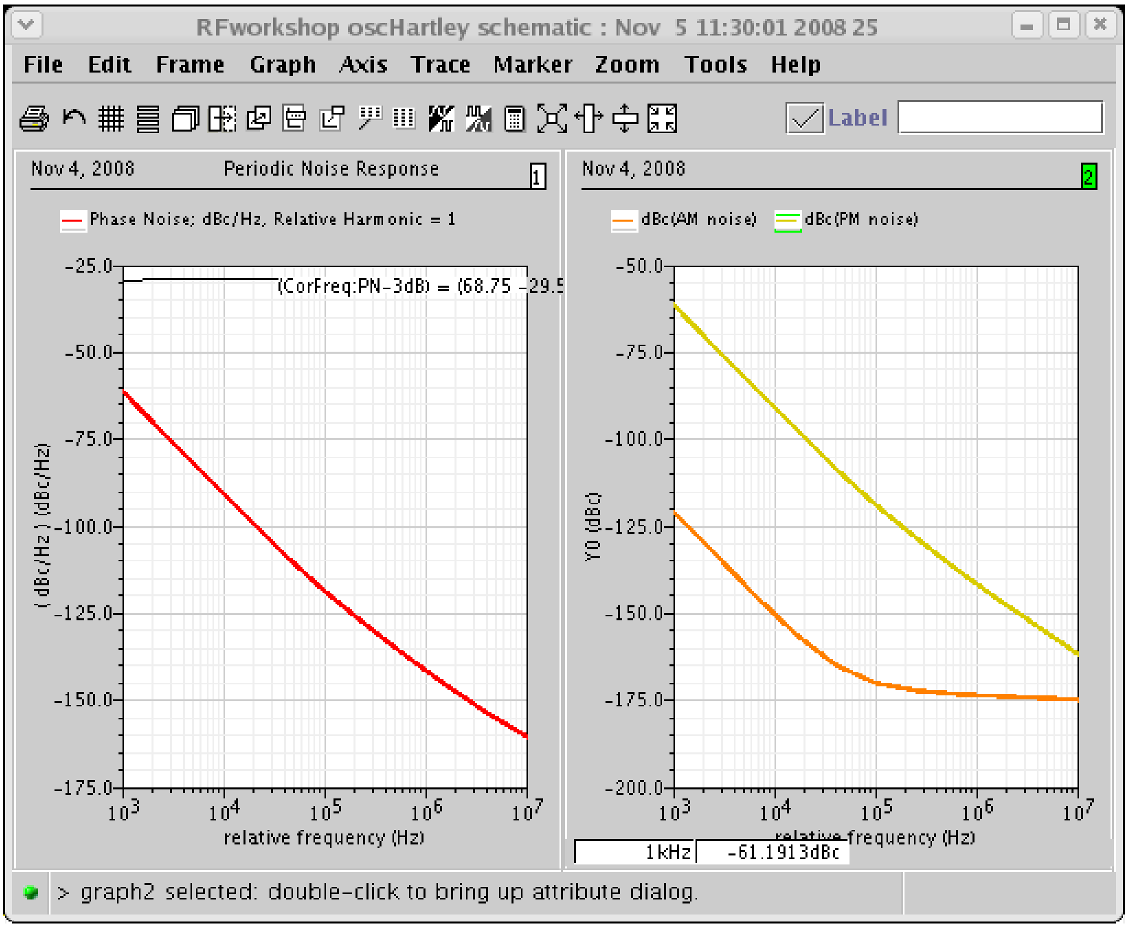
Action 1-22: In the Direct Plot form, set the **Plotting Mode** to **Append**. Select **pnoise modulated**. Select **AM**. Select **dBc**. Click **Plot**.



Action 1-23: In the Direct Plot form, select **PM** and click **Plot**.



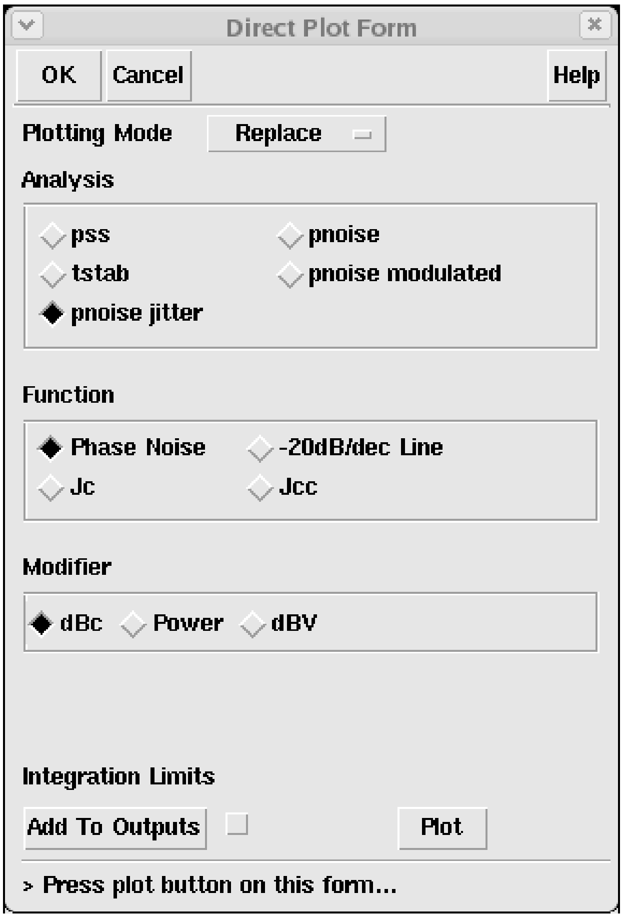
The waveform window updates.



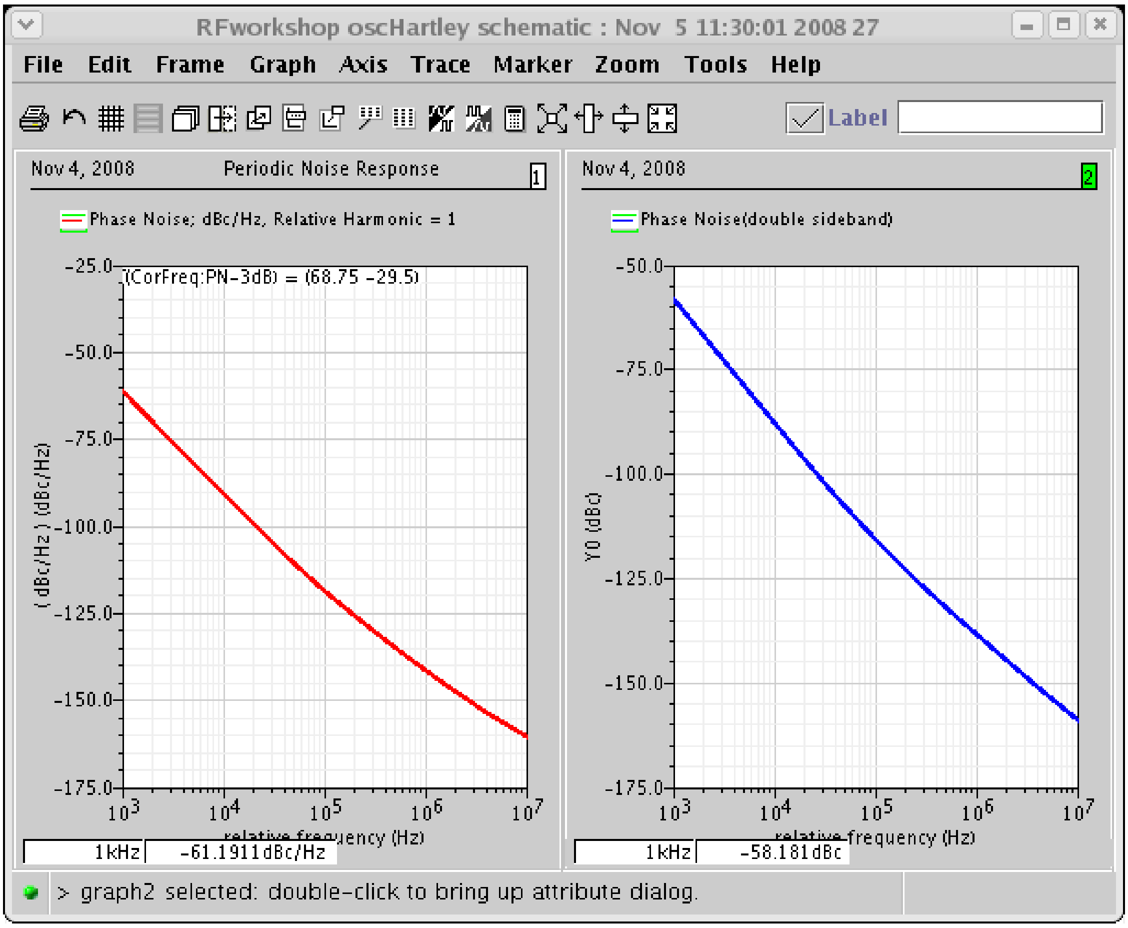
Action 1-24: Compare the total phase noise to the PM noise. Move your mouse cursor in the waveform window and read the phase noise plot and the PM noise at 10K Hz. They are the same.

Action 1-25: In the waveform window, select window 2.

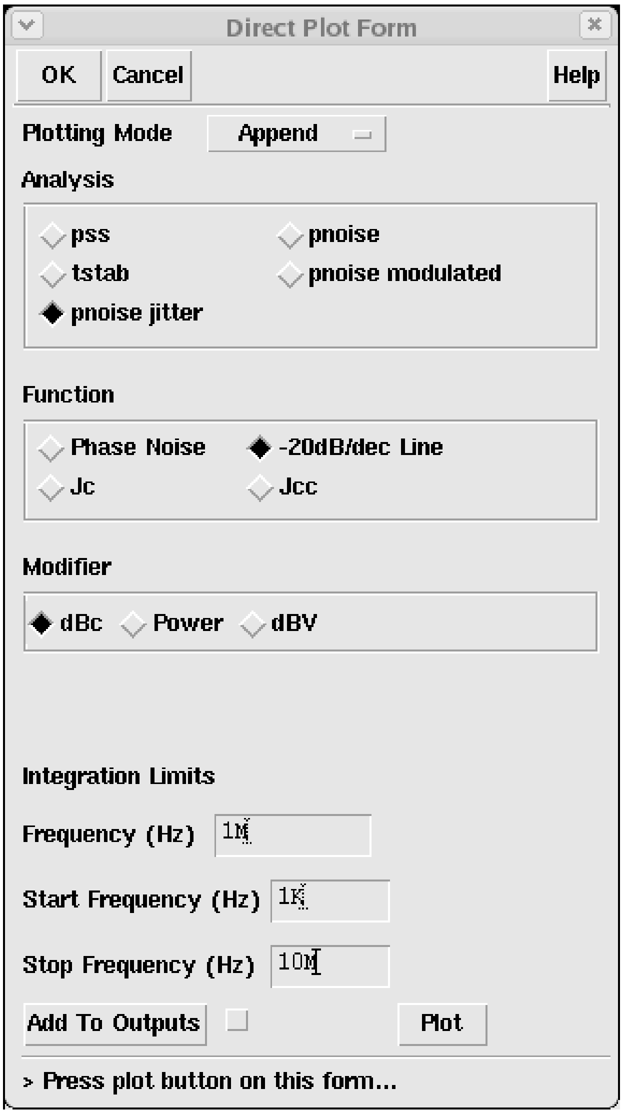
Action 1-26: In the Direct Plot form, select **pnoise jitter**. Select **phase noise**. Change the **Plotting Mode** to **Replace**. Select **Plot**.



The following plot shows the phase noise. Notice that the result from **pnoise jitter** is around 3 dB higher than that from **pnoise** because the **pnoise jitter** simulation calculates double sideband noise rather than single sideband.

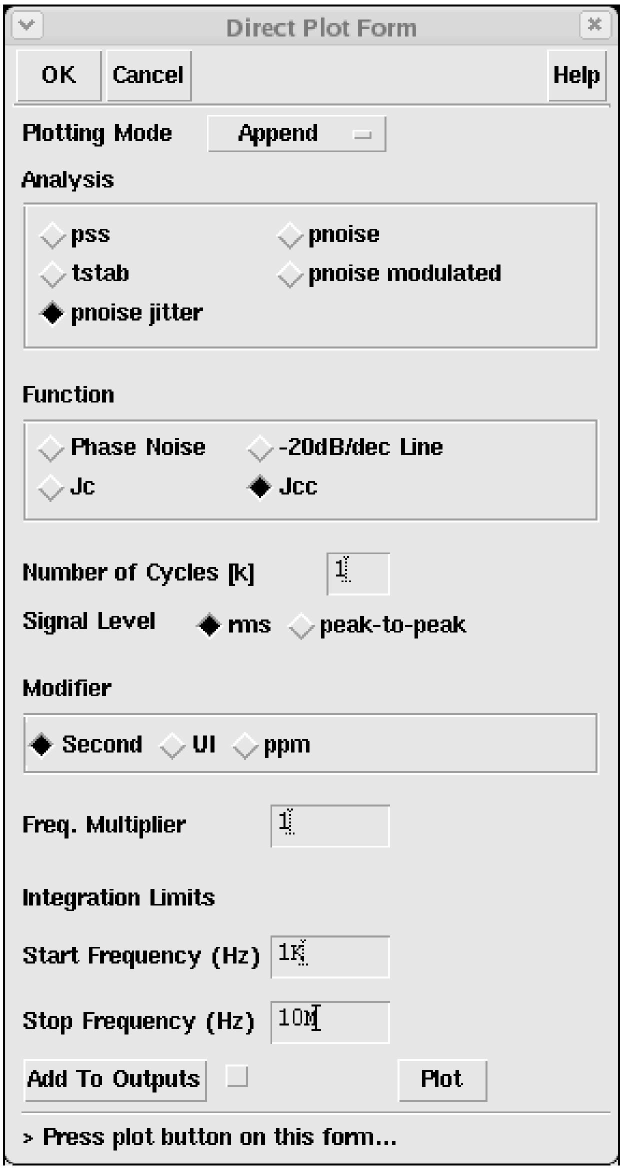


Action 1-27: In the Direct Plot form, select **pnoise jitter**. Select **-20dB/dec Line** and configure the form as follows:

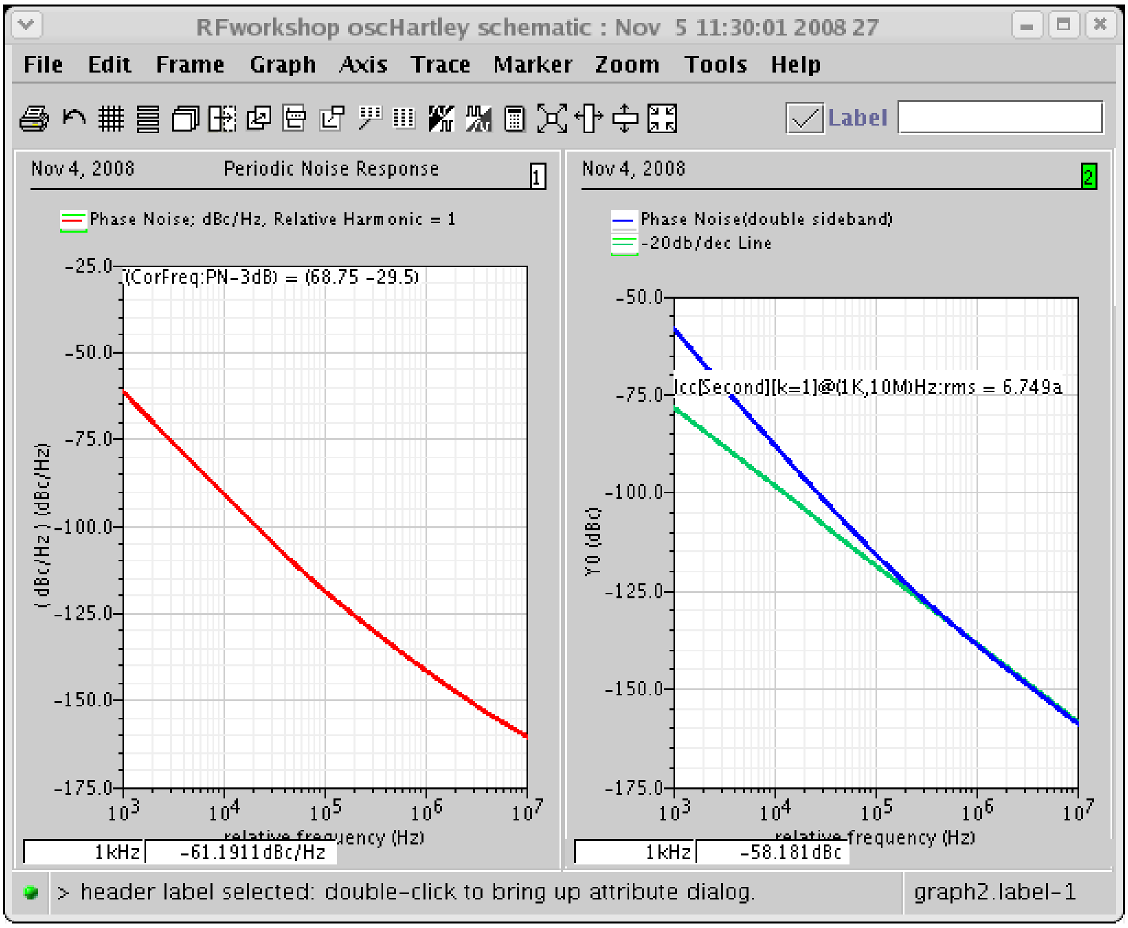


The “-20dB/dec” curve could be placed on the phase noise curve at any frequency. The slope of -20dB/dec “assistant” lets us distinguish the regions of the 1/f^3, 1/f^2 and 1/f phase noise PSD. If you prefer, quick manual calculations using simple white noise jitter approximations could be used after the proper region of 1/f^2 slope is determined [3] The white noise approximation was used in the first release of the jitter measurements. It required users to select the point for the approximation of the slope. In later releases of Direct Plot, the integration is numerical and the selection of the point is no longer required. Therefore, the slope assistant is for informative purposes only now.

Action 1-28: In the Direct Plot form, select **pnoise jitter**. Select **JCC** (Cycle-to-cycle jitter). Select **Plot**. The **Plotting Mode** is **Append**. Note that the integration limits have been set to the entire frequency range specified in the pnoise Choosing Analyses form.



The waveform window updates with the jitter calculation.



MMSIM60 USR2 introduces Harmonic Balance for autonomous circuit application. In this release Harmonic Balance is supported for oscillators and phase-noise simulation. Harmonic Balance is a frequency domain technique similar to Harmonic balance which solves the spectra of each node voltage unlike the time-domain technique (such as shooting PSS) which solves the waveform of each node voltage over the period.

Action 1-29: (Optional) Choose **Session — Load State** in the Virtuoso Analog Design Environment window, select **Cellview** in **Load State Option** and load state “**Lab1_Pnoise_FB**”, then skip to Action 1-35.

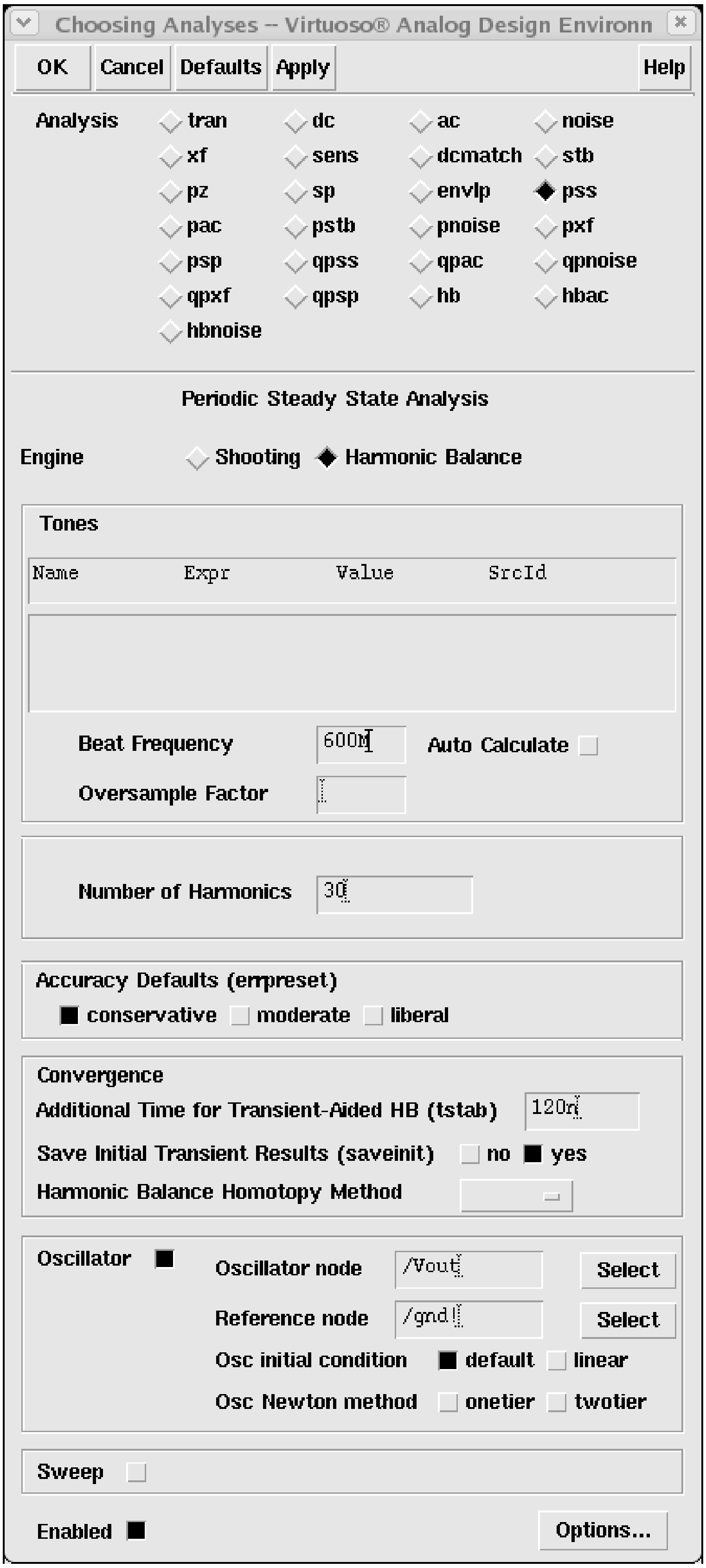
Action 1-30: In the Analog Design Environment window, select **Analyses — Choose…**.

Action 1-31: In the Choosing Analyses window, select **pss** in the **Analysis** field of the window.

Action 1-32: In the Periodic Steady State Analysis form, choose **Harmonic Balance** engine. Set **Fundamental Frequency** of **600 MHz**.

Set the **Number of Harmonics** to **30**. Set tstab to add additional time for the oscillator to converge on its operating frequency. Click **yes** to save the initial transient results. Turn on **Oscillator**. Keep the other settings as they were before.

The Periodic Steady State Analysis form looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environn
OK
Cancel
Defaults
Apply
Help
Analysis
tran
dc
ac
noise
xf
sens
dcmatch
stb
pz
sp
envlp
pss
pac
pstb
pnoise
pxf
psp
qpss
qpac
qpnoise
qpxf
qpsp
hb
hbac
hbnoise
Periodic Steady State Analysis
Engine
Shooting
Harmonic Balance
Tones
Name
Expr
Value
SrcId
Beat Frequency
600M
Auto Calculate
Oversample Factor
Number of Harmonics
30
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Convergence
Additional Time for Transient-Aided HB (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Harmonic Balance Homotopy Method
Oscillator
Oscillator node
/Vout
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Osc Newton method
onetier
twotier
Sweep
Enabled
Options...

Currently, there are two Harmonic Balance methods implemented: onetier or twotier. They are set by parameter "oscmethod". In the onetier method, the frequency and voltage spectrum are solved simultaneously in one single set of nonlinear equations. In the twotier method, the nonlinear equations are split into two sets: the inner set of nonlinear equations solves the spectrum of node voltage equations; the outer set of nonlinear equations solves the oscillation frequency. Physically it is equivalent to add a sinusoidal voltage probe to a pinning node and adjust its frequency and amplitude until it has no effect on the oscillator. SpectreRF automatically chooses the pinning node. By default, SpectreRF runs the onetier method first for n iterations; and then, if necessary, it runs the twotier method for n iterations (n is set by "maxperiods"). You can also run only the onetier method or only the twotier method by specifying "oscmethod". The twotier method has a larger convergence zone as its convergence is slightly more robust. However, this method is slower than the onetier method.

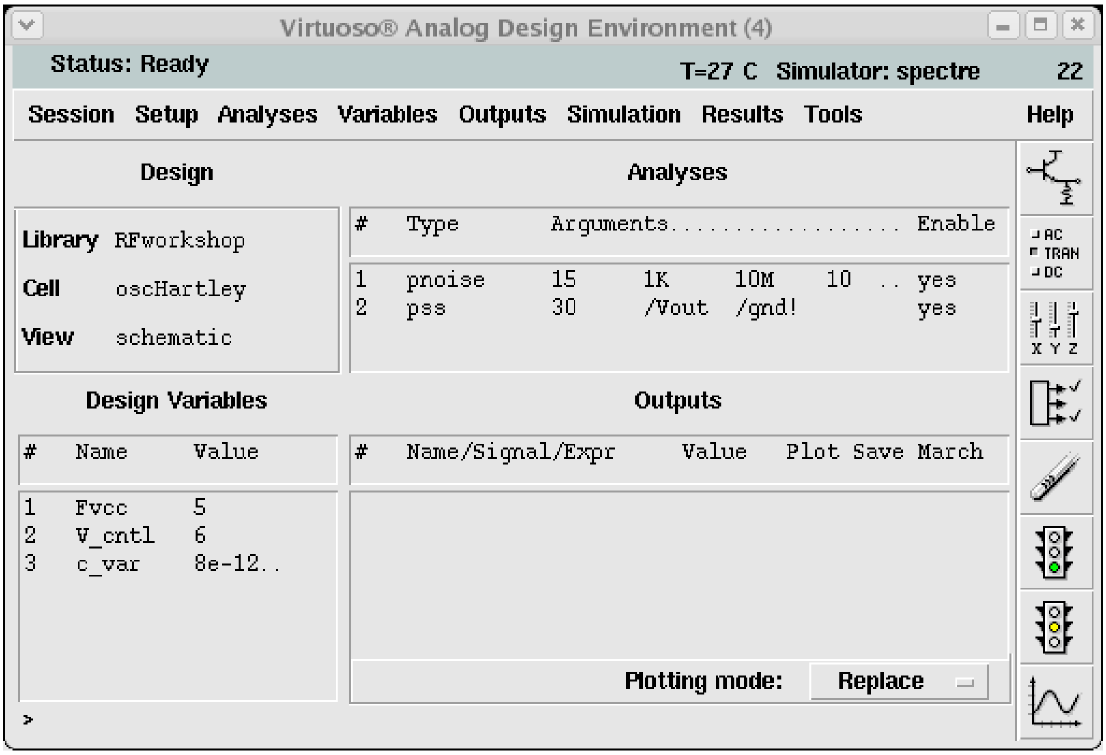
SpectreRF Harmonic Balance is transient-assisted. SpectreRF runs transient analysis (length is specified by "tstab") and then switches to Harmonic Balance. Stable oscillation in the transient analysis can help Harmonic Balance converge. Stable oscillation can be achieved by:

- Setting an initial condition for a particular node (for example: ic net01=5.0).
- Setting an initial condition for the inductor/capacitor in the resonator in OSC (for example: L14 (Xtal02 net01) inductor l=6.m ic=0.5m).
- Adding a damped current source in parallel to the resonator in OSC (for example: Ikicker (net01 net02) isource type=sine freq=1.0G ampl=1m damp =1.0G.
- Adding a voltage pulse (for example: vdd (vdd 0) vsource type=pulse val0=0.0 val1=3.3 rise=1n).
- Specifying "oscic=default" or "oscic=lin".

Action 1-33: Click **pnoise**. Set the analysis form as you did in Action 1-10.

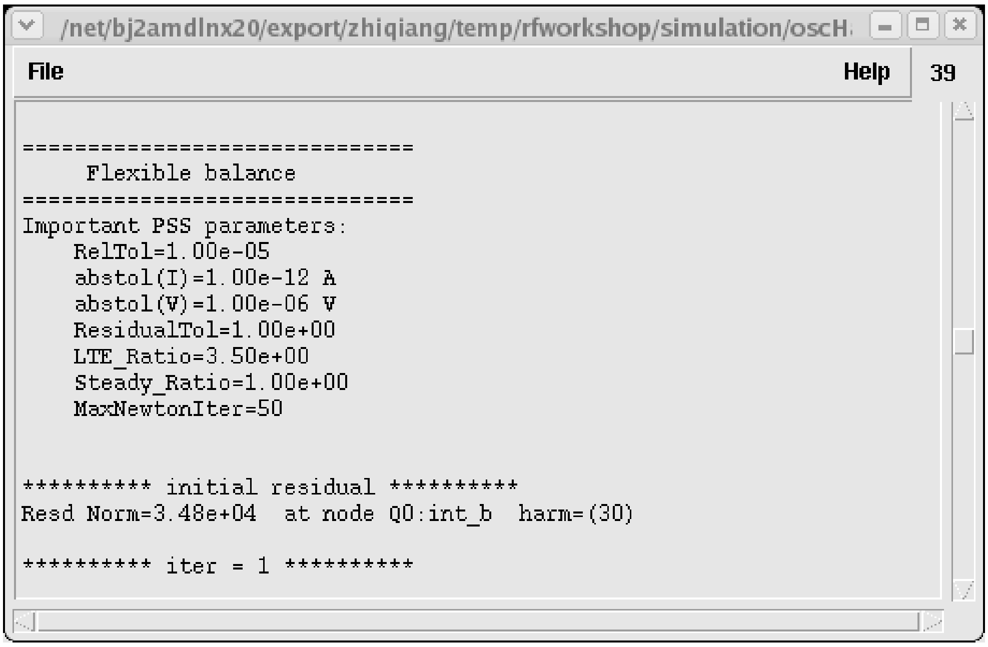
Action 1-34: Turn on **Enabled**. Click **OK** in the Choosing Analyses form.

The Virtuoso Analog Design Environment window looks like this:

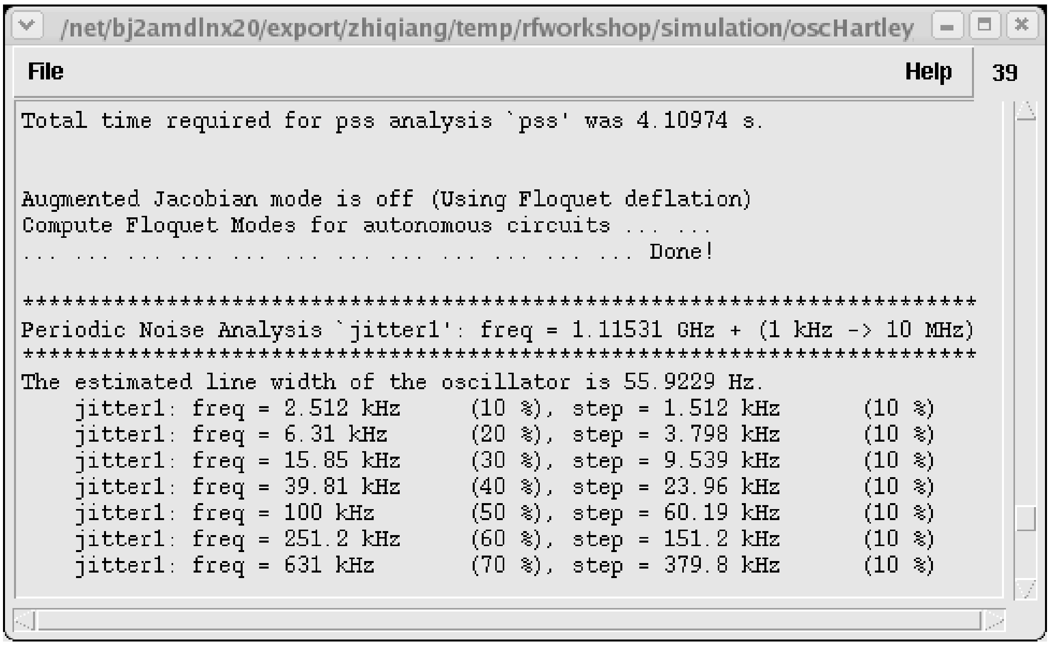


Action 1-35: In the Analog Design Environment window, choose **Simulation — Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

As the simulation progresses, messages similar to the following appear in the simulation output log window. The messages differ from those produced during time domain pss.



The output log file prints out the following message:

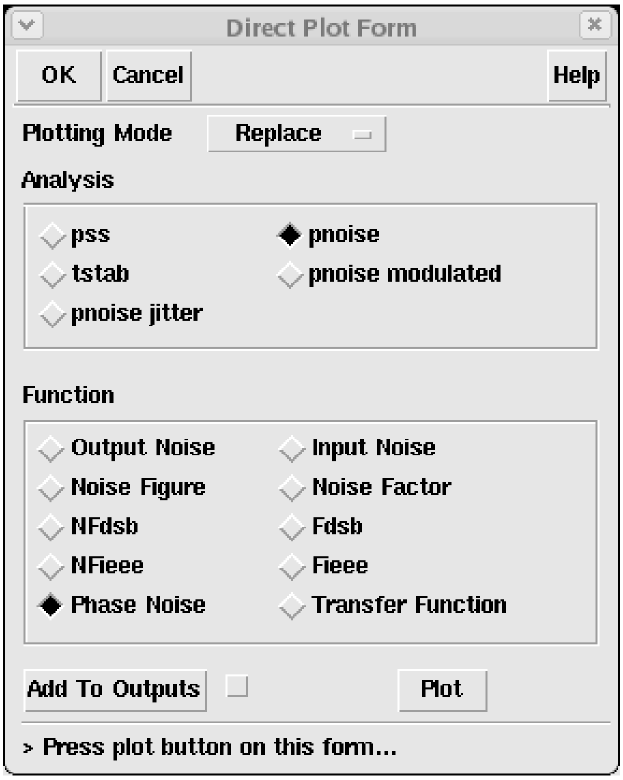



```
Total time required for pss analysis `pss' was 4.10974 s.


Augmented Jacobian mode is off (Using Floquet deflation)
Compute Floquet Modes for autonomous circuits ... ...
... ... ... ... ... ... ... ... ... ... ... ... Done!

************************************************************************
Periodic Noise Analysis `jitter1': freq = 1.11531 GHz + (1 kHz -> 10 MHz)
************************************************************************
The estimated line width of the oscillator is 55.9229 Hz.
    jitter1: freq = 2.512 kHz      (10 %), step = 1.512 kHz      (10 %)
    jitter1: freq = 6.31 kHz       (20 %), step = 3.798 kHz      (10 %)
    jitter1: freq = 15.85 kHz      (30 %), step = 9.539 kHz      (10 %)
    jitter1: freq = 39.81 kHz      (40 %), step = 23.96 kHz      (10 %)
    jitter1: freq = 100 kHz        (50 %), step = 60.19 kHz      (10 %)
    jitter1: freq = 251.2 kHz      (60 %), step = 151.2 kHz      (10 %)
    jitter1: freq = 631 kHz        (70 %), step = 379.8 kHz      (10 %)
```

Notice that the corner frequency is also calculated with the HB engine.

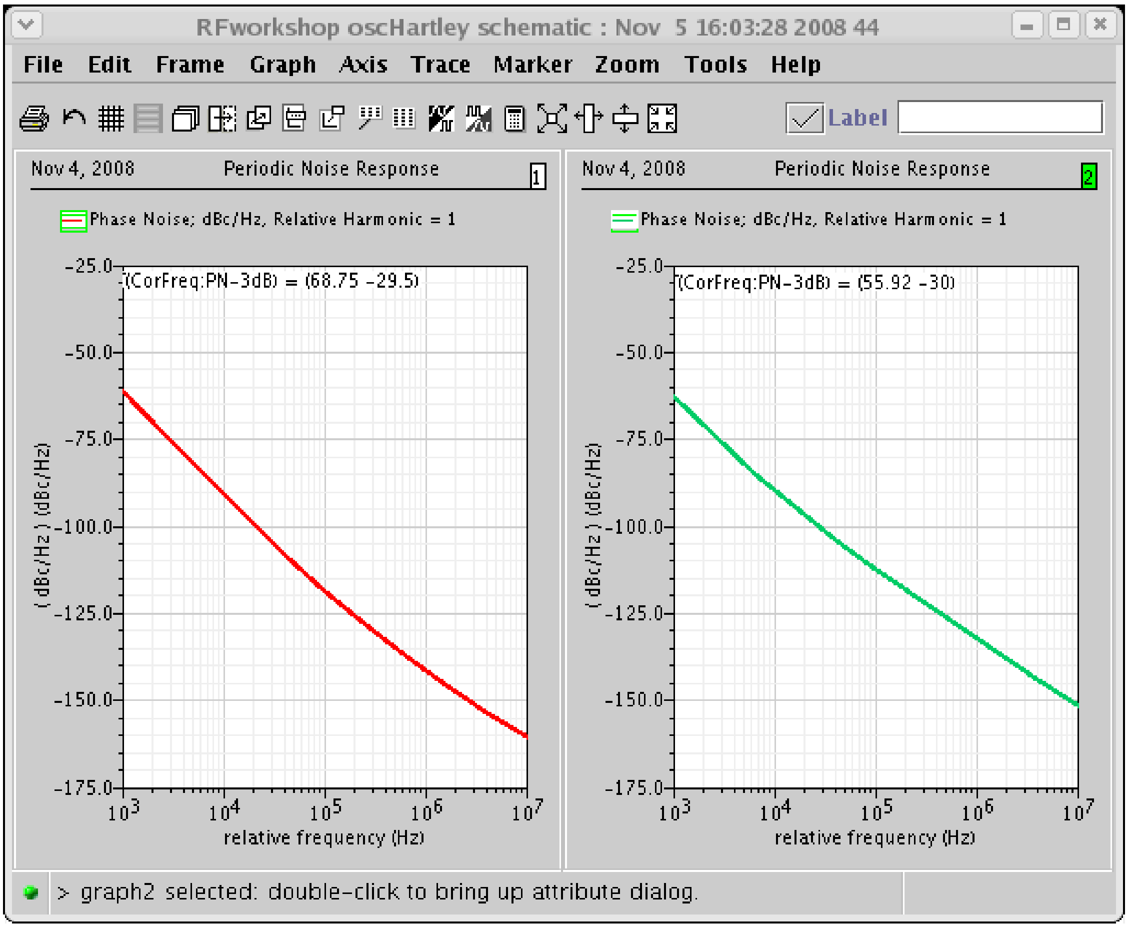
Action 1-36: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form** to plot the phase noise.

Action 1-37: In the Direct Plot Form window, choose **pss** as the **Analysis** type and configure the form as follows:



Action 1-38: In the Direct Plot form, click **Plot**.

Now you get the following waveforms:



The Harmonic Balance method is a good candidate for simulating mildly-nonlinear oscillators with resonators, such as LC oscillators, negative-gain oscillators, and crystal oscillators. The Shooting PSS method is a good candidate for simulating strongly-nonlinear non-resonator oscillators, such as ring oscillators, relaxation oscillators, or oscillators containing digital control components. Harmonic Balance simulation is slower in this case, so it might not be a good choice for this circuit.

Action 1-39: Close the waveform window, the Direct Plot form, and Virtuoso Analog Design Environment window.

Lab 2: Frequency Pushing (Swept PSS)

Frequency pushing is the variation of the VCO output frequency due to a change in the power supply (Vcc). One way to measure frequency pushing is as follows.

1. Set the supply voltage (Vcc) at its nominal setting and compute the VCO frequency for different tune voltages.

2. Increase the supply voltage by a specific amount, and measure the VCO frequency for different tune voltages as before.

3. Decrease the supply voltage by the same amount, from the nominal value, and measure the frequency for different tune voltages as before.

At a given tuning voltage, the frequency change due to a 1 volt supply voltage change yields the frequency pushing. Frequency pushing may be different at different tuning voltages.

Action 2-1: If it is not already open, open the *schematic* view of the design *oscHartley* in the library *RFworkshop.*

Action 2-2: From the oscHartley schematic, start the Virtuoso Analog Design Environment with the **Tools — Analog Environment** command.

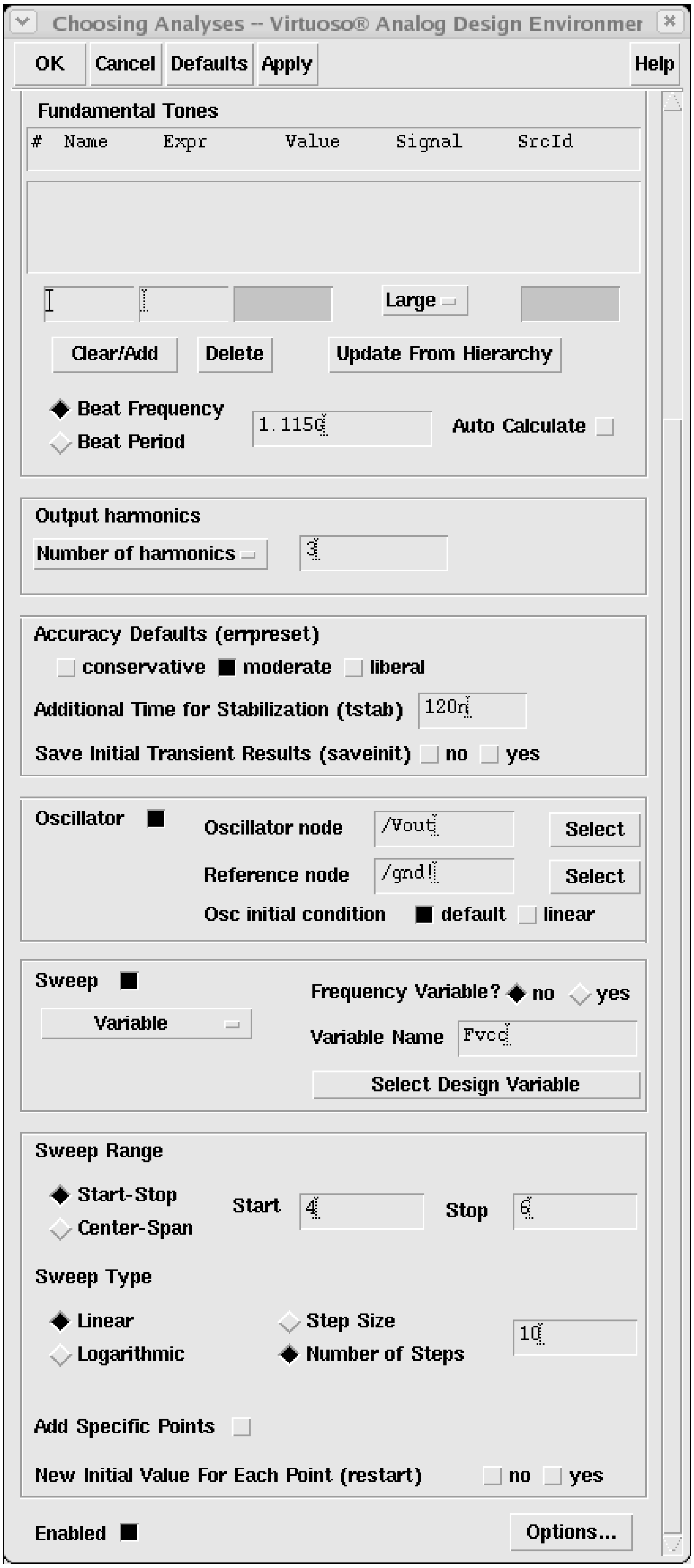
Action 2-3: (Optional) Choose **Session — Load State**, select **Cellview** in **Load State Option** and load the state "**Lab2_Frequency_Pushing_PSS**", and skip to Action 2-8.

Action 2-4: In the Virtuoso Analog Design Environment window, choose **Analyses — Choose…**.

Action 2-5: In the Choosing Analyses window, select **pss** in the **Analysis** field.

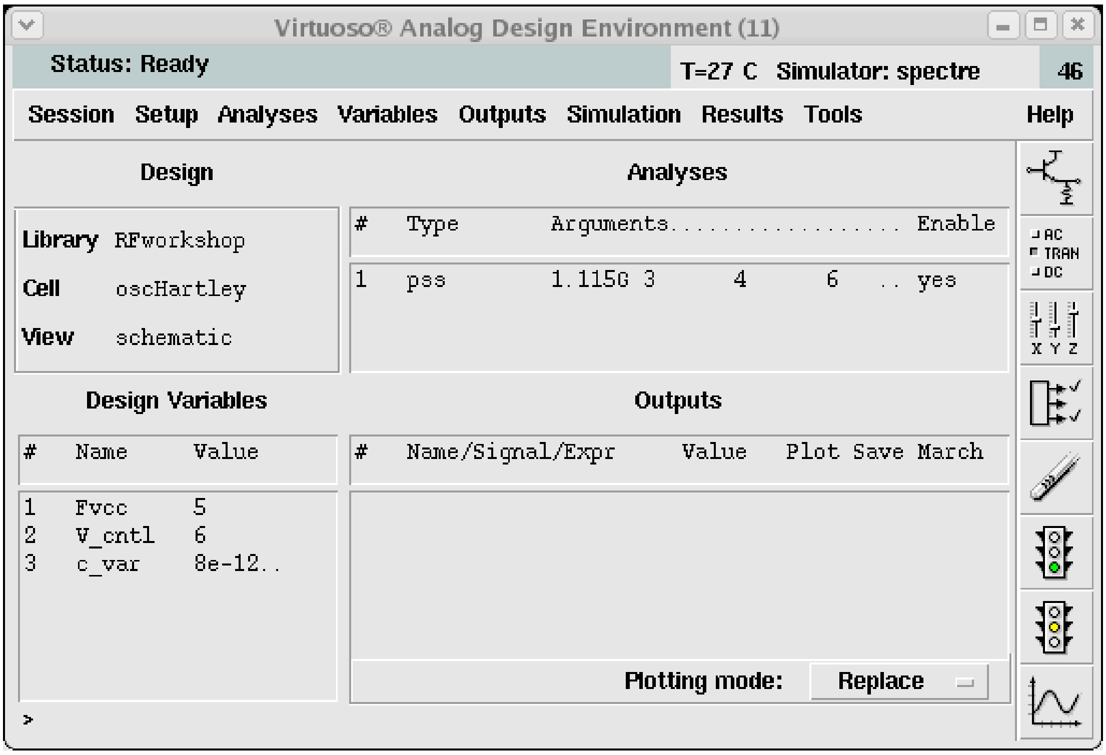
Action 2-6: Set up a swept PSS analysis with a **Beat Frequency** = 1.115 G; **Number of Harmonics** = 3; **errpreset** = **moderate**; **tstab** = 120 n; turn on **Oscillator**; set **Oscillator node** = /Vout; and **Reference node** = /gnd!; enable **Sweep**; enter fvcc as **Variable Nam**e; set the **Sweep Range Start** = 4 and **Stop** = 6; set **Sweep Type** = **Linear**; and **Number of Steps** = 10.

The Periodic Steady State Analysis window looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environmer
OK
Cancel
Defaults
Apply
Help
Fundamental Tones
Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Hierarchy
Beat Frequency
Beat Period
1.115G
Auto Calculate
Output harmonics
Number of harmonics
3
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Additional Time for Stabilization (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Oscillator
Oscillator node
/Vout
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Sweep
Variable
Frequency Variable?
no
yes
Variable Name
Fvcc
Select Design Variable
Sweep Range
Start-Stop
Center-Span
Start
4
Stop
6
Sweep Type
Linear
Logarithmic
Step Size
Number of Steps
10
Add Specific Points
New Initial Value For Each Point (restart)
no
yes
Enabled
Options...

Action 2-7: Turn on **Enabled**, and click **OK** in the Choosing Analyses form.

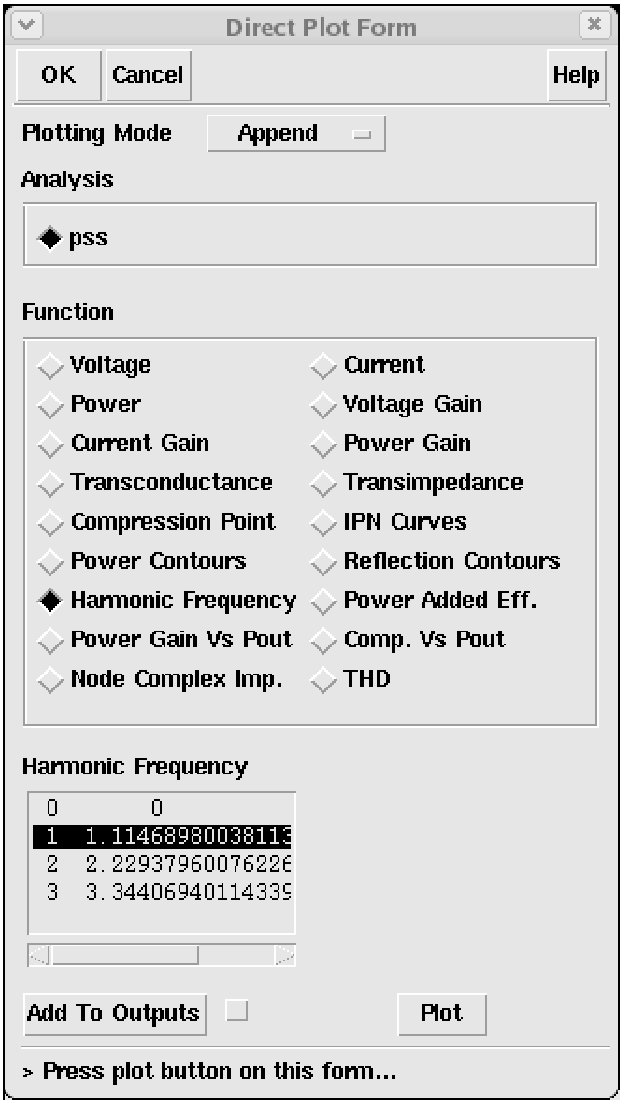
The Virtuoso Analog Design Environment window looks like this:



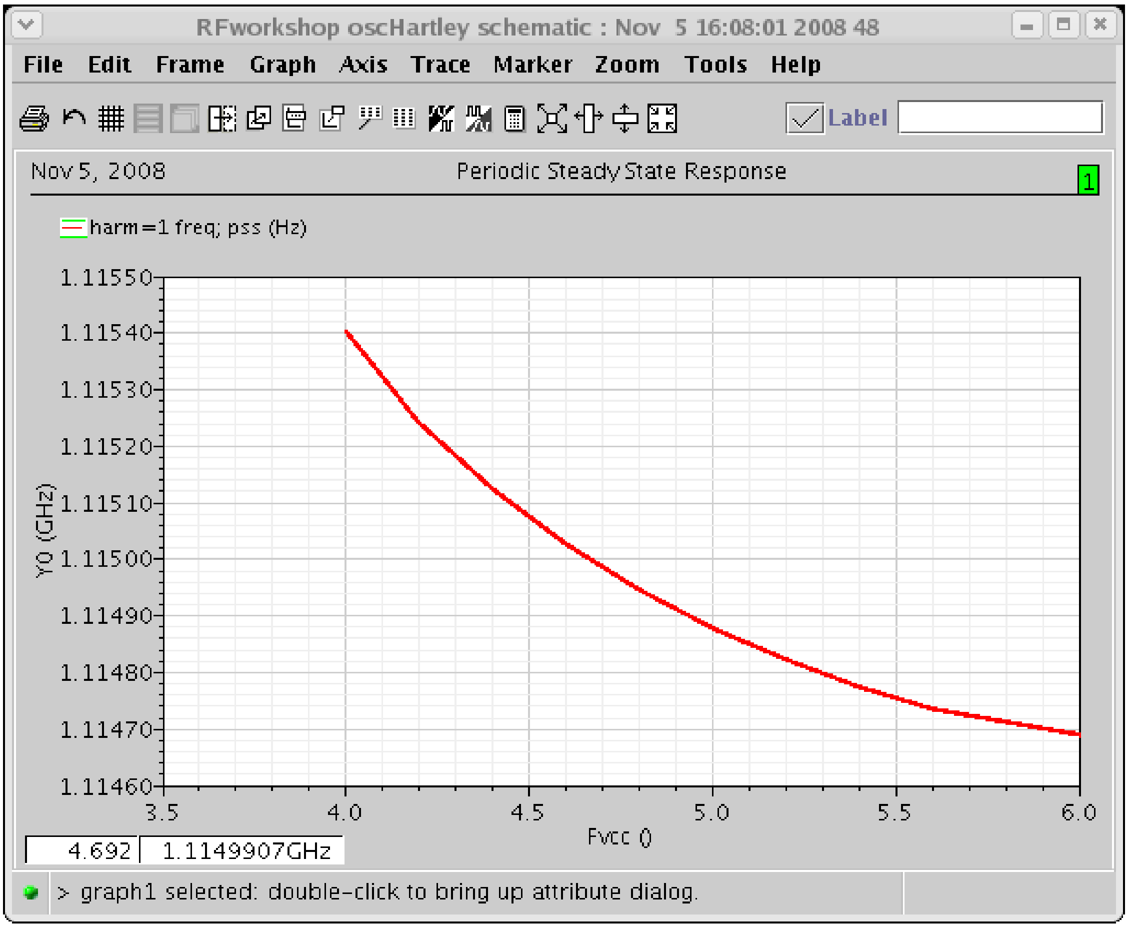
Action 2-8: In your Analog Design Environment, choose **Simulation — Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action 2-9: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form**.

Action 2-10: In the Direct Plot Form, select **pss**, select **Harmonic Frequency**, and highlight the 1st harmonic in the Harmonic Frequency section. The form looks like this:



Action 2-11: Click **Plot**. The following plot appears.



Action 2-12: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

Lab 3: Tuning Sensitivity and Linearity (Swept PSS)

Tuning sensitivity is defined as the frequency change per unit of tuning voltage. Ideally, tuning sensitivity would be constant but in practice this is generally not the case. In this lab, you

1. Compute the VCO frequency for different tuning voltages.

2. Plot VCO frequency measurements against tuning voltage. The slope of this characteristic is the tuning voltage sensitivity which you can calculate at different tuning voltages.

The tuning sensitivity is expressed in Hz/V.

Action 3-1: If it is not already open, open the *schematic* view of the design *oscHartley* in the library *Rfworkshop*.

Action 3-2: From the *oscHartley* schematic, start the Virtuoso Analog Design Environment with the **Tools — Analog Environmen**t command.

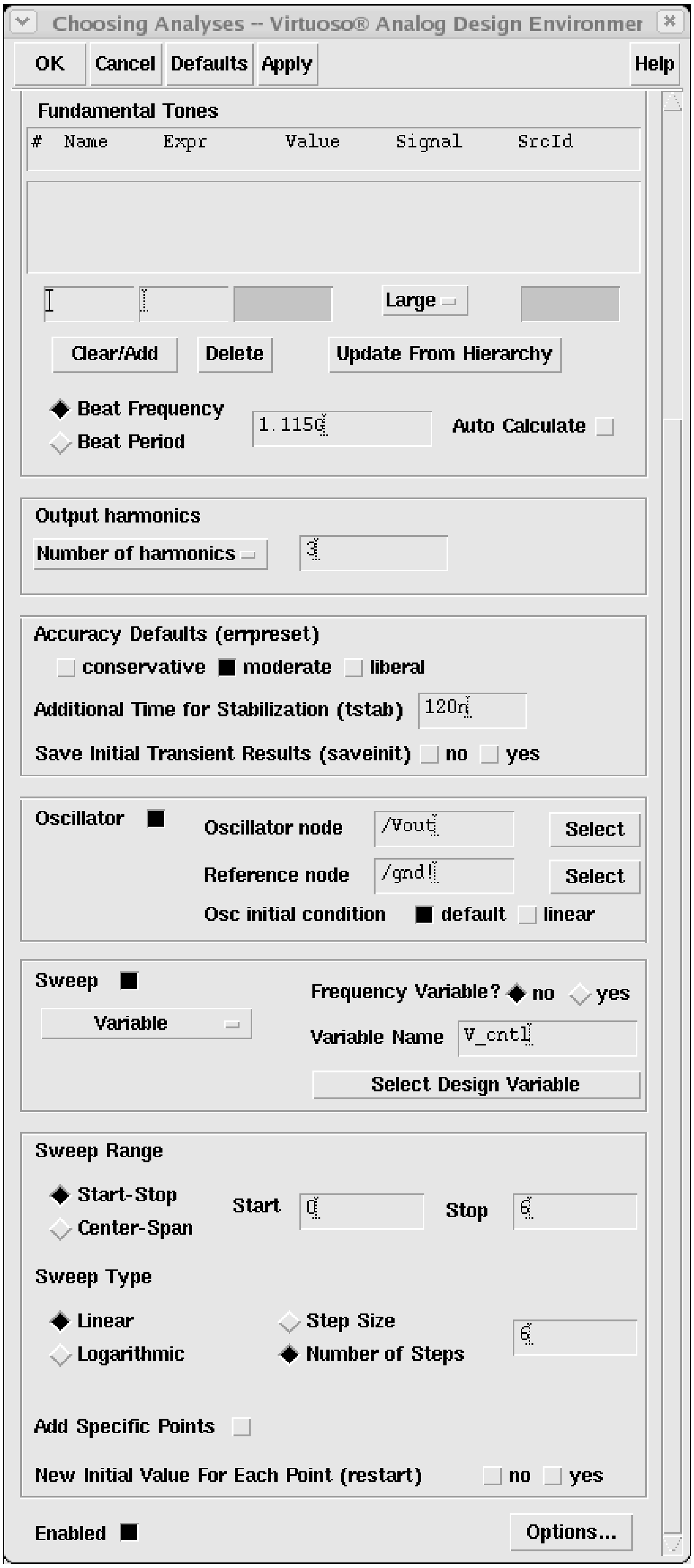
Action 3-3: (Optional) Choose **Session — Load State**, select **Cellview** in **Load State Option** and load the state "**Lab3_Sensitivity_Linearity_PSS**", and skip to Action 3-8.

Action 3-4: In the Virtuoso Analog Design Environment window, choose **Analyses — Choose….**

Action 3-5: In the Choosing Analyses window, select **pss** in the **Analysis** field.

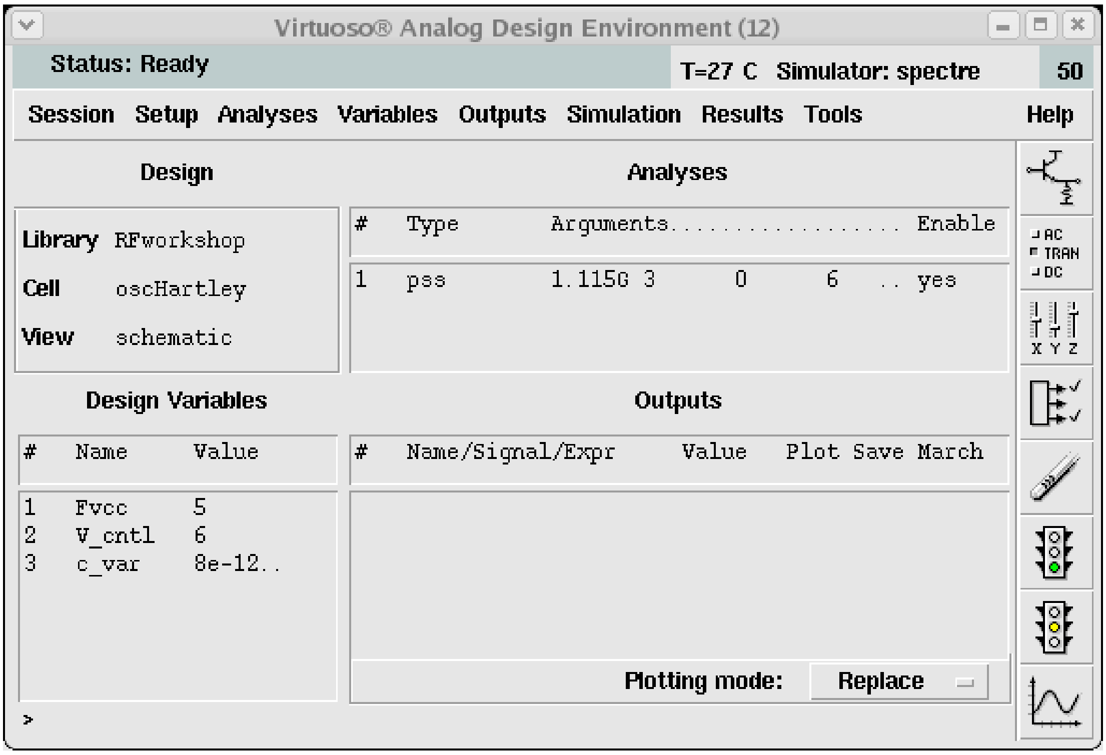
Action 3-6: Set up a swept PSS analysis with a **Beat Frequency** = 1.115G; **Number of Harmonics** = 3, **errpreset** = **moderate**, **tstab** = 120n. Turn on **Oscillator**, set **Oscillator node** = /Vout, and Reference node = /gnd!. Turn on **Sweep**, enter V_cntl as **Variable Name** (this is the tuning voltage), set the **Sweep Range Start** = 0 and **Stop** = 6, set **Sweep Type** = **Linear**, and **Number of Steps** = 6.

The Periodic Steady State Analysis window looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environmer
OK
Cancel
Defaults
Apply
Help
Fundamental Tones
Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Hierarchy
Beat Frequency
Beat Period
1.115G
Auto Calculate
Output harmonics
Number of harmonics
3
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Additional Time for Stabilization (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Oscillator
Oscillator node
/Vout
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Sweep
Variable
Frequency Variable?
no
yes
Variable Name
V_cntl
Select Design Variable
Sweep Range
Start-Stop
Center-Span
Start
0
Stop
6
Sweep Type
Linear
Logarithmic
Step Size
Number of Steps
6
Add Specific Points
New Initial Value For Each Point (restart)
no
yes
Enabled
Options...

Action 3-7: Turn on **Enabled**, and click **OK** in the Choosing Analyses form.

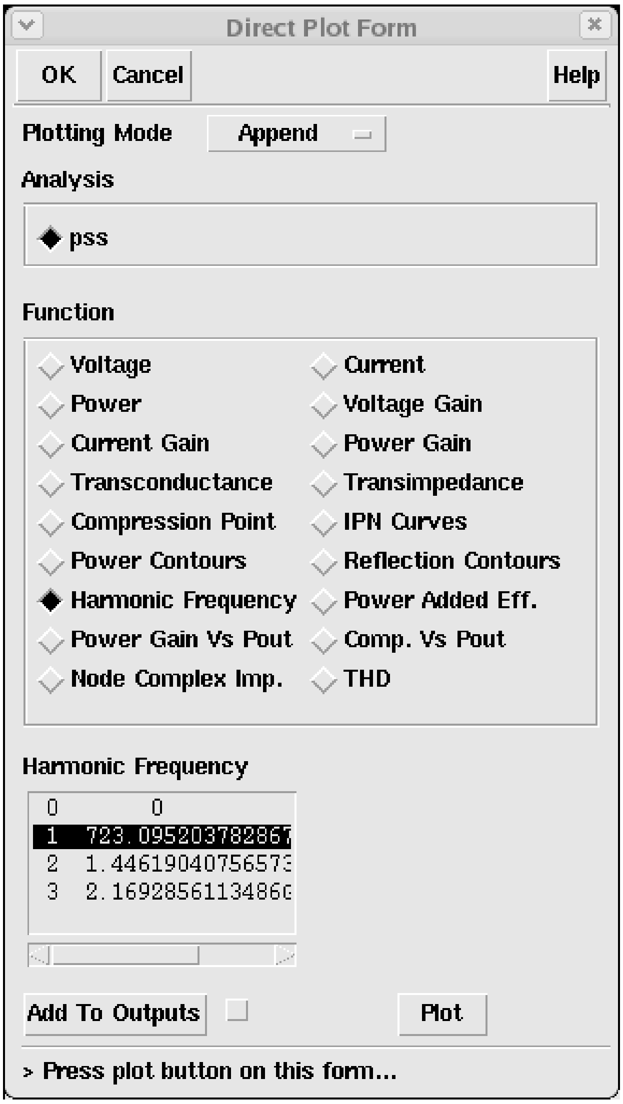
The Virtuoso Analog Design Environment window looks like this:



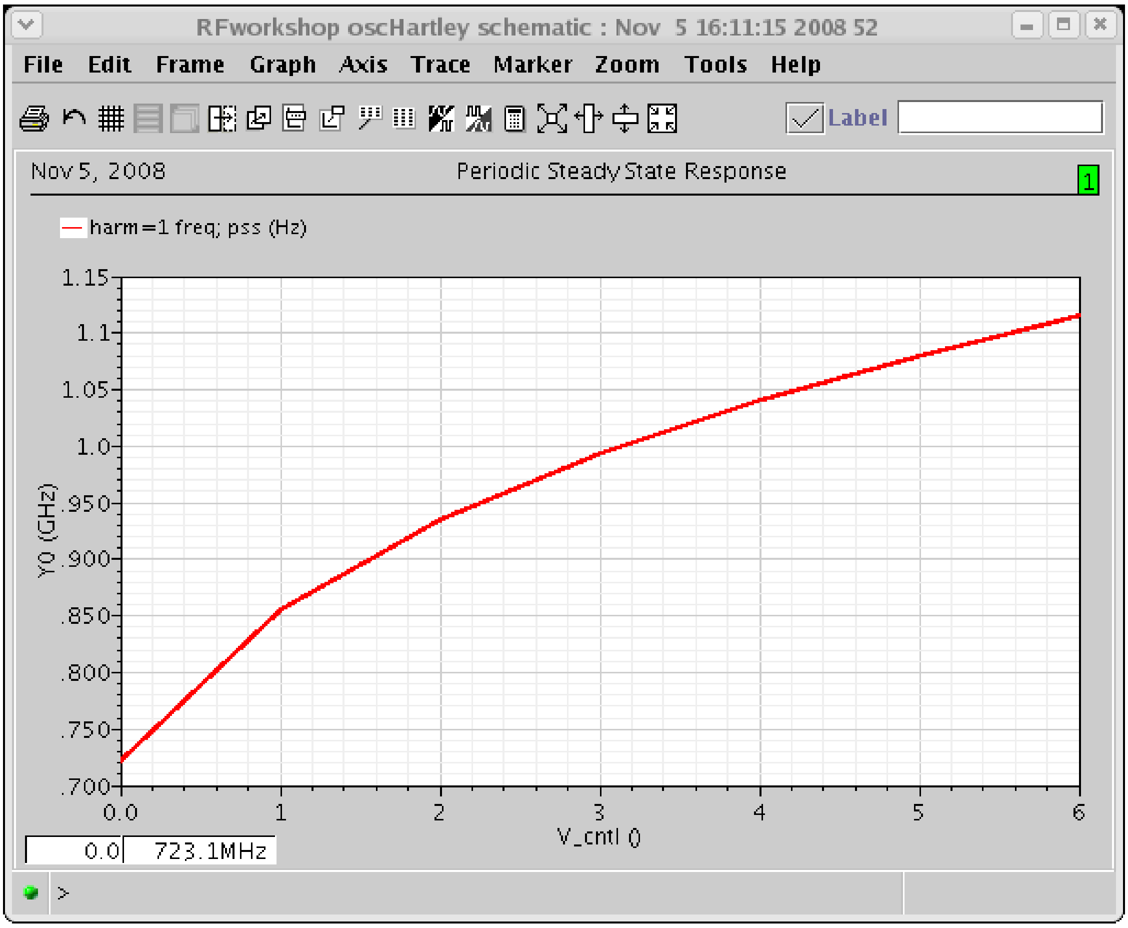
Action 3-8: In the Analog Design Environment window, choose **Simulation — Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action 3-9: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form**.

Action 3-10: In the Direct Plot Form, select **pss**, click **Harmonic Frequency**, and highlight the 1st harmonic in the **Harmonic Frequency** section. The form looks like this:



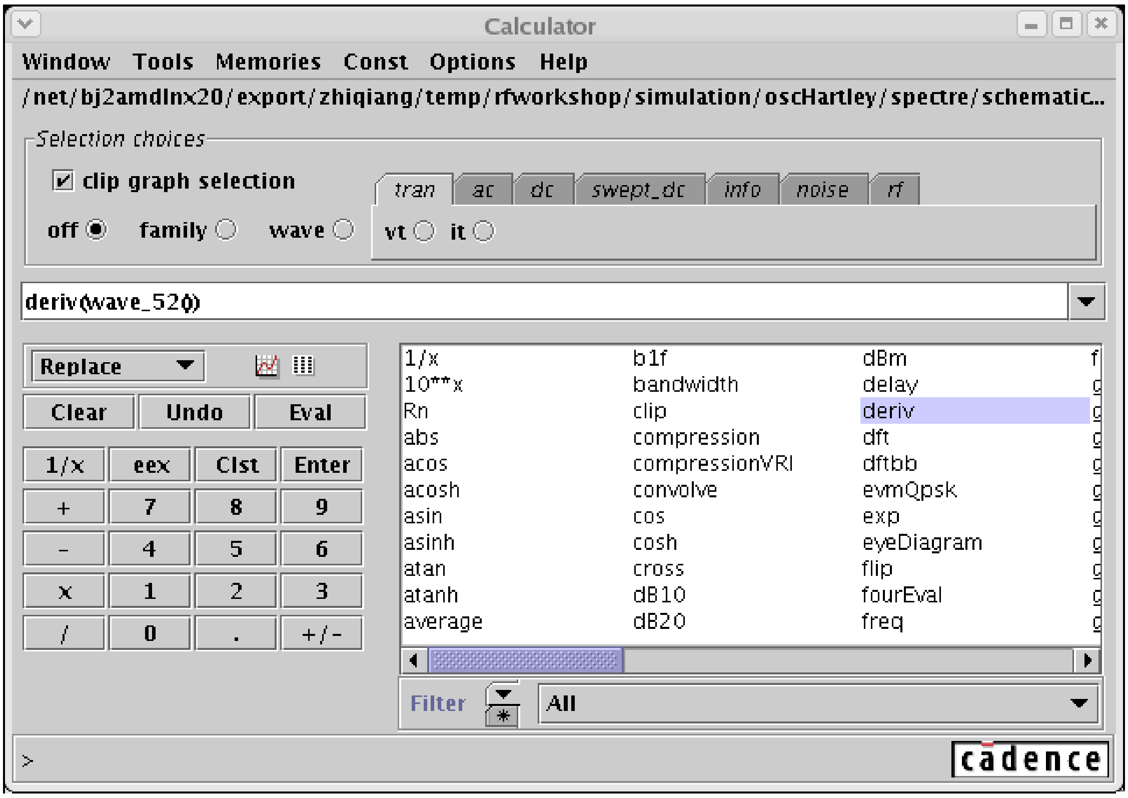
Action 3-11: Click **Plot**. The following plot appears.



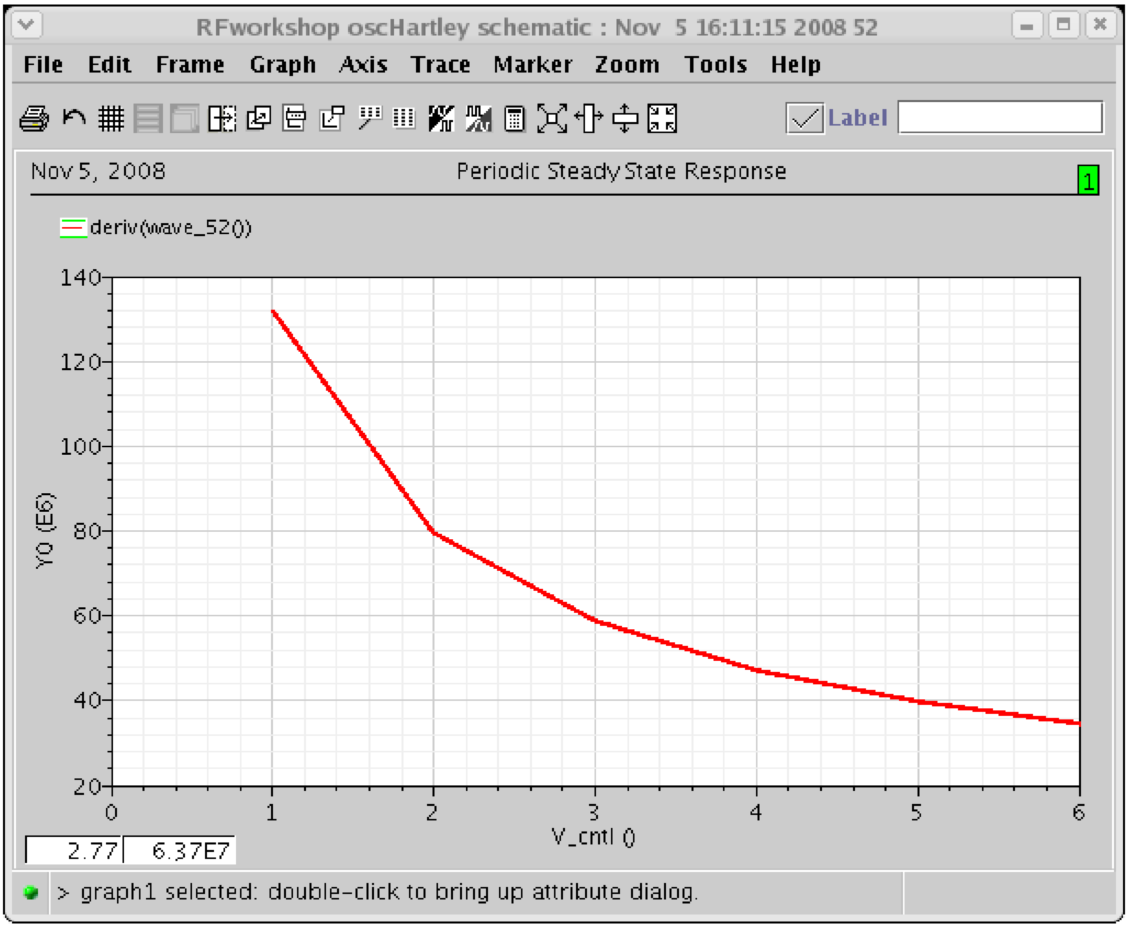
Action 3-12: In the waveform window, choose **Tools — Calculator.**

Action 3-13: In the Calculator window, highlight **wave**, and select the sensitivity curve in the waveform window. Choose **deriv** in the special function field. Change the plotting mode to **Replace**.

The calculator window looks like this:



Action 3-14: Click the icon in the calculator window. The following plot represents the frequency change per unit volt of tuning voltage.



Action 3-15: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

Lab 4: Power Dissipation (PSS)

Power dissipation arises from the following sources:

- Dynamic power dissipation due to switching current from charging and discharging parasitic capacitance.
- Dynamic power dissipation due to short-circuit current when both n-channel and p-channel transistors are momentarily on at the same time.
- Static power dissipation due to leakage current and subthreshold current. VCOs suffer from trade-offs between speed, power dissipation, and noise. Typically, they drain from 1 to megawatts, mW, of power.

Action 4-1: If it is not already open, open the *schematic* view of the design *oscHartley* in the library *RFworkshop*

Action 4-2: From the *oscHartley* schematic, start the Virtuoso Analog Design Environment with the **Tools — Analog Environment** command.

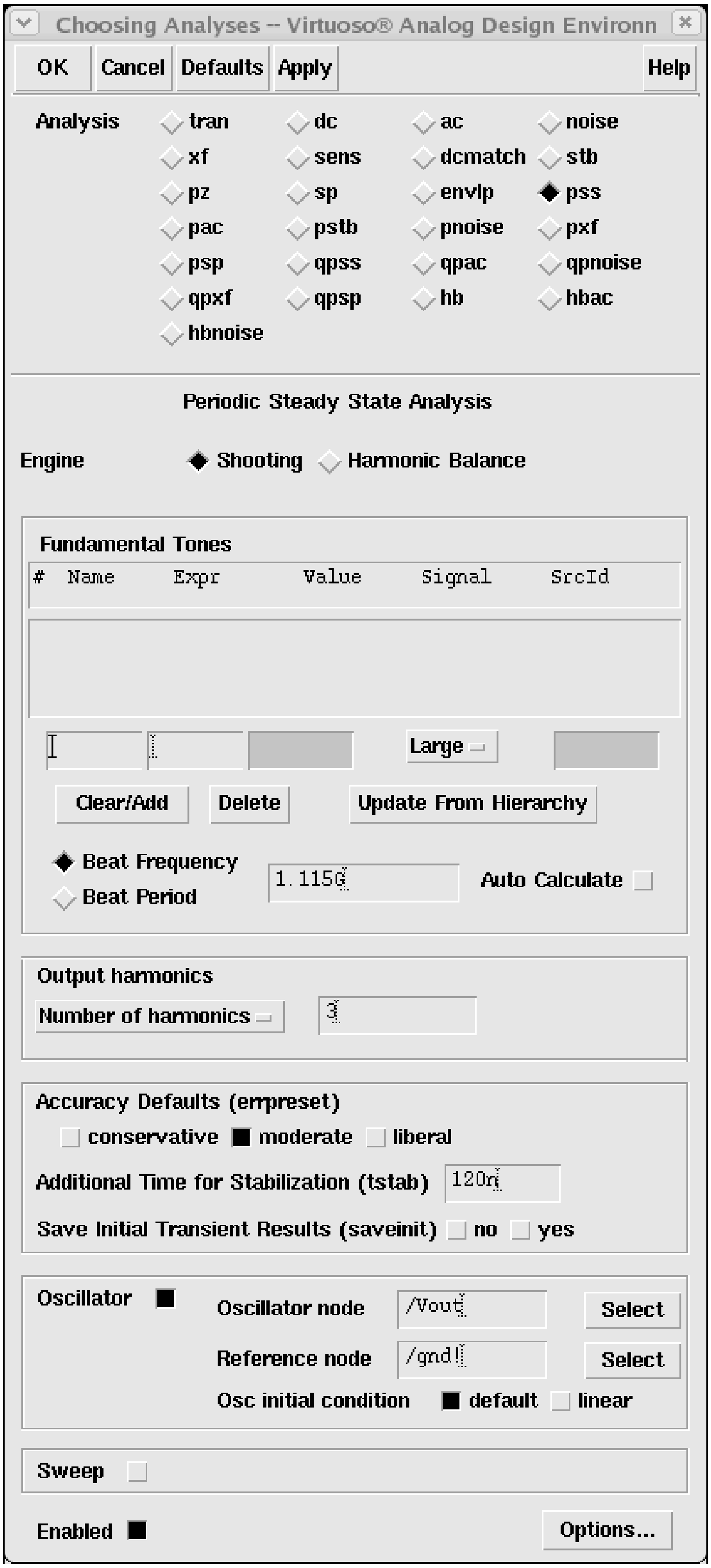
Action 4-3: (Optional) Choose **Session — Load State**, select **Cellview** in **Load State Option** and load the state "**Lab4_Power_Dissipation_PSS**", and skip to Action 4-11.

Action 4-4: In the Virtuoso Analog Design Environment window, choose **Analyses — Choose…**.

Action 4-5: In the Choosing Analyses window, select **pss** in the **Analysis** field.

Action 4-6: Set up a PSS analysis with a **Beat Frequenc**y = 1.115G; **Number of Harmonics** = 3; **errpreset** = **moderate**; **tstab** = 120n; turn on **Oscillator**; set **Oscillator node** = /Vout; and **Reference node** = /gnd!.

The Periodic Steady State Analysis window looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environn
OK
Cancel
Defaults
Apply
Help
Analysis
tran
dc
ac
noise
xf
sens
dcmatch
stb
pz
sp
envlp
pss
pac
pstb
pnoise
pxf
psp
qpss
qpac
qpnoise
qpxf
qpsp
hb
hbac
hbnoise
Periodic Steady State Analysis
Engine
Shooting
Harmonic Balance
Fundamental Tones
Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Hierarchy
Beat Frequency
Beat Period
1.115G
Auto Calculate
Output harmonics
Number of harmonics
3
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Additional Time for Stabilization (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Oscillator
Oscillator node
/Vout
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Sweep
Enabled
Options...

Action 4-7: Turn on **Enabled** and click **OK** in the Choosing Analyses form.

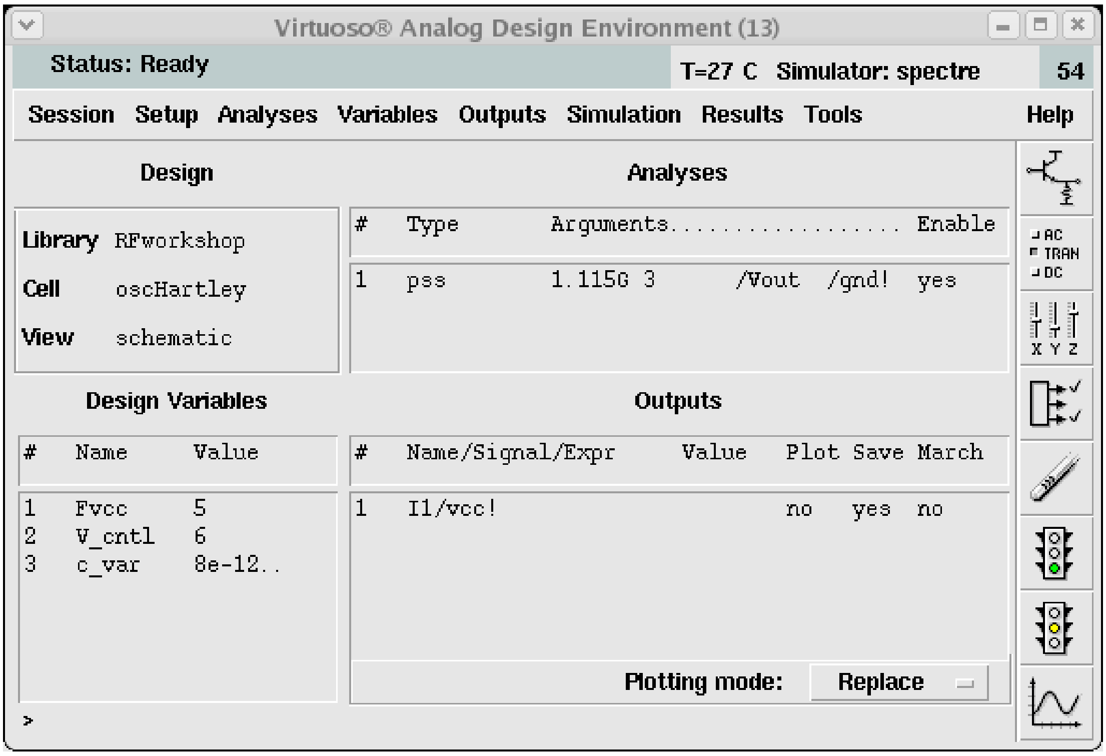
To obtain the power dissipation, before you run the PSS analysis, you must save data at the Vcc terminal through the analog design environment.

Action 4-8: In the Virtuoso Analog Design Environment window, choose **Outputs — To Be Saved — Select On Schematic**.

Action 4-9: In the schematic, select the *Vcc* terminals. The **Outputs** section of the analog design environment window must display, **I1/vcc!** with the Save column set to yes.

Action 4-10: Press **Esc** with your cursor in the *oscHartley* schematic window to end the selections.

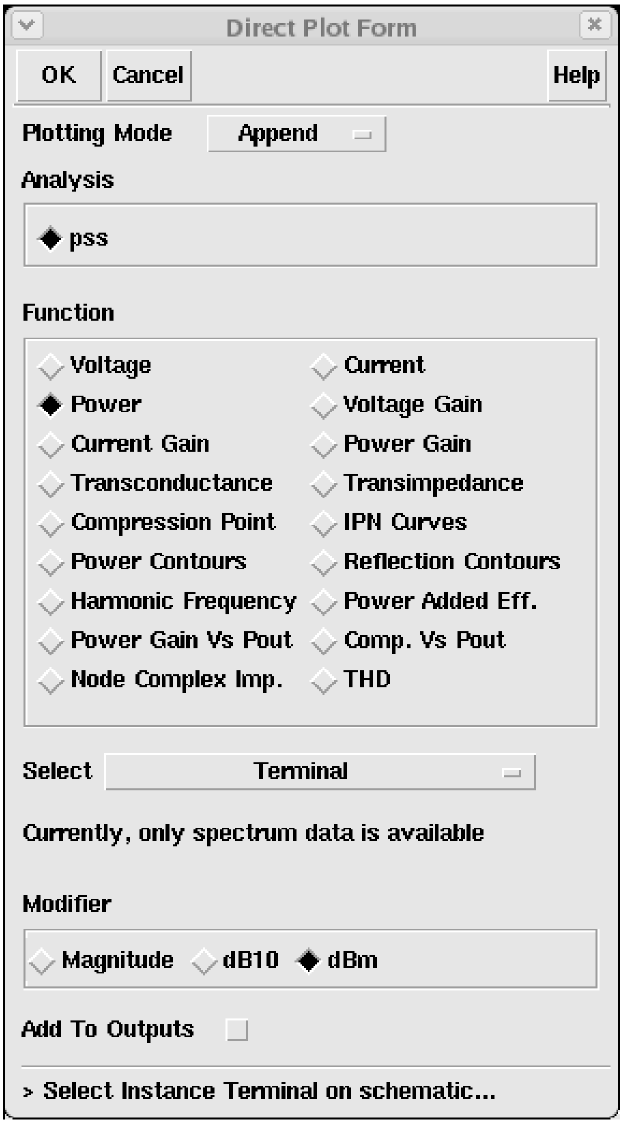
The Virtuoso Analog Design Environment window looks like this:



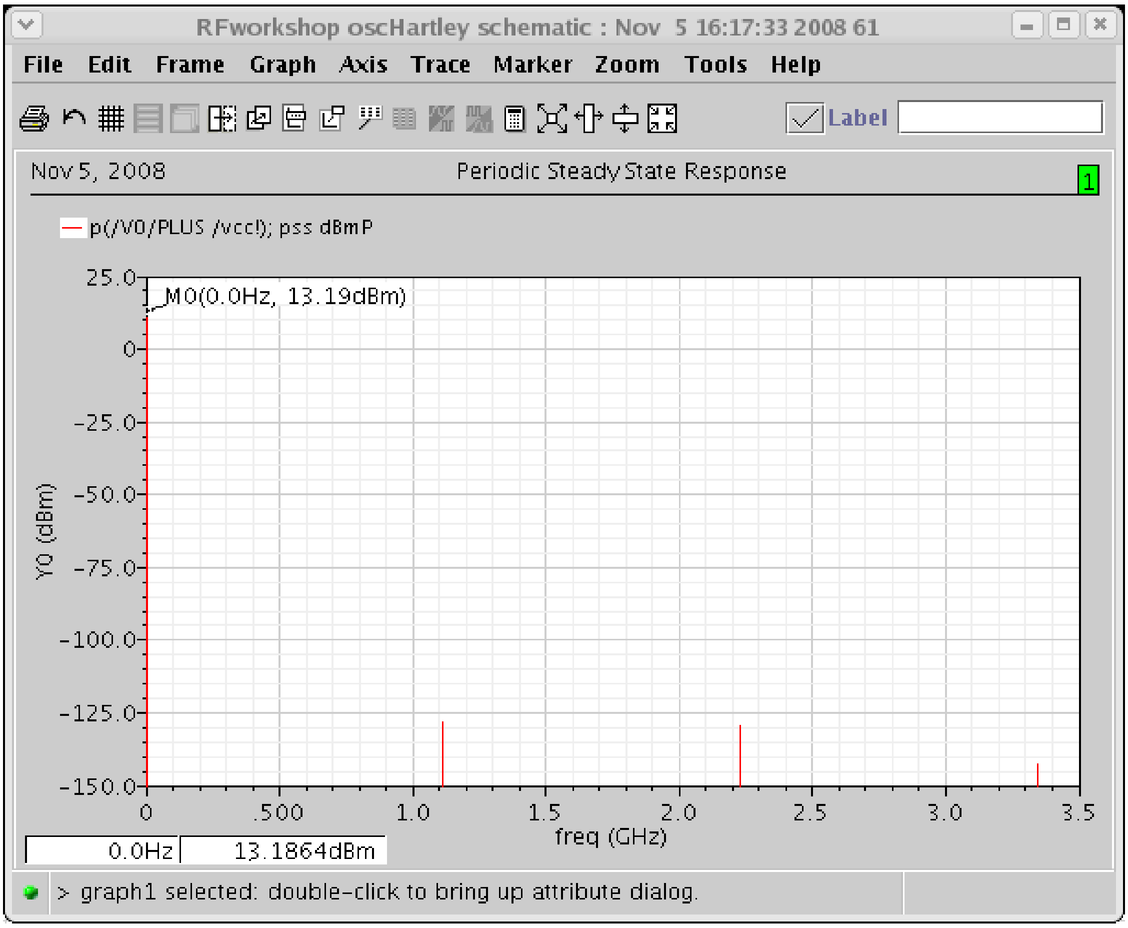
Action 4-11: In your Analog Design Environment, choose **Simulation — Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action 4-12: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form**.

Action 4-13: In the Direct Plot Form, select **pss**, click **power**, and choose **dBm** as **Modifier**. The form looks like this:



Action 4-14: Click the positive terminal of pulse source on the schematic. . The following plot appears.



The DC value in the above figure corresponds to power dissipation; that is, at freq = 0.0, power dissipation is equal to -16.81 dB or 13.19 dBm.

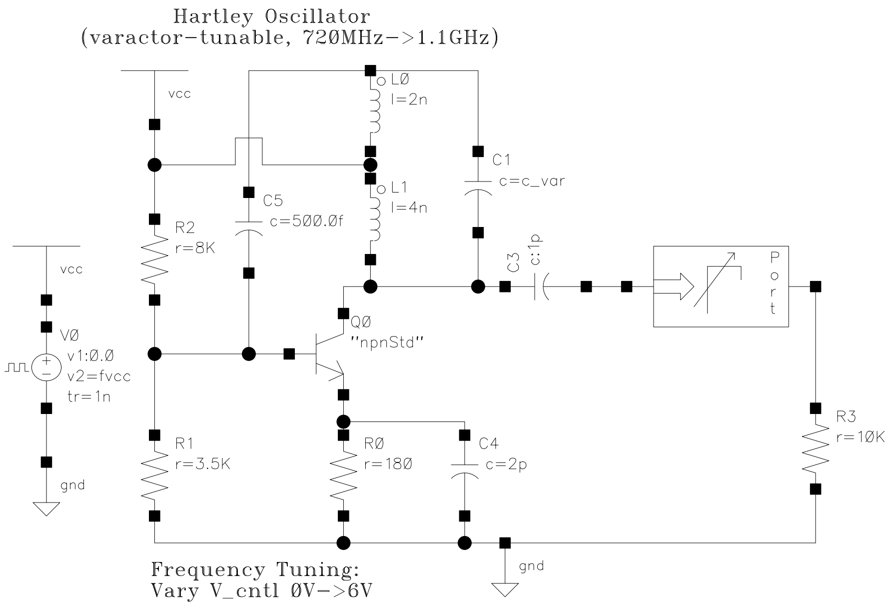
Action 4-15: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window. Close the *oscHartley* schematic.

Lab 5: Frequency Pulling (Swept PSS)

Frequency pulling is a measure of frequency change due to a non-ideal load. You measure frequency pulling by noting the frequency change caused by a load having a nominal 12 dB return loss with all possible phases. You should minimize frequency pulling, especially in cases where power stages are close to the VCO unit and short pulses might affect the output frequency.

Action 5-1: In the Library Manager window, open the *schematic* view of the design *freqpull* in the library *RFworkshop*

The following figure shows the modified Hartley Oscillator schematic for frequency pull calculations.



An instance of a PortAdaptor is connected to the load. The PortAdaptor is set to have the following properties:

- Frequency = 1.115 G;
- Phase of Gamma = theta;
- Mag of Gamma = 0.2512
- Reference Resistance = 10K (this value must equal the load).

Frequency pulling is the measurement of frequency change caused by a load having a nominal 12 dB return loss with all possible phases. The value of Mag of Gamma, 0.2512, is computed from the return loss value, rl, using the following formula:

$rl = -20 Log \mid \Gamma \mid$

where Γ is the reflection coefficient with respect to source impedance .

Action 5-2: From the *freqpull* schematic, start the Virtuoso Analog Design Environment with the **Tools — Analog Environment** command.

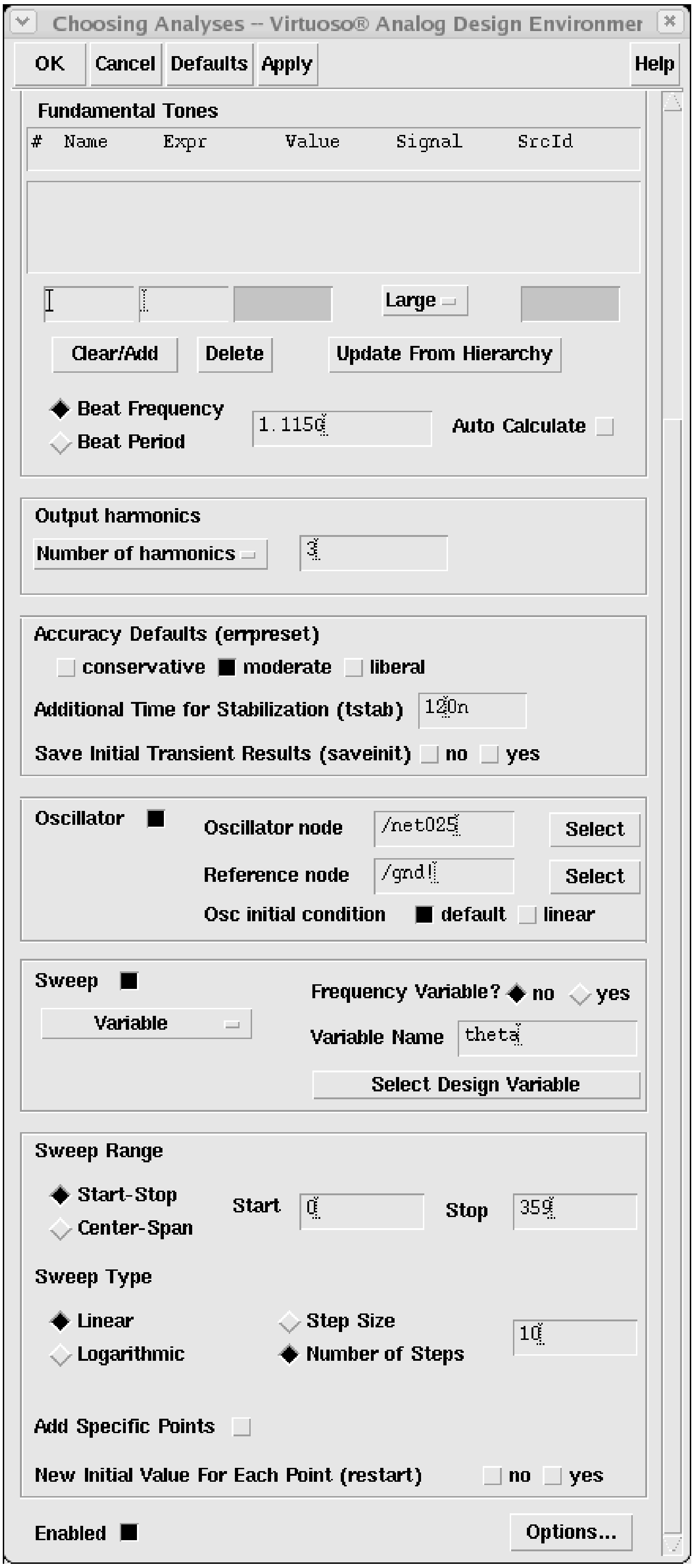
Action 5-3: (Optional) Choose **Session — Load State**, select **Cellview** in **Load State Option** and load the "**Lab5_Frequency_Pulling_PSS**" state, and skip to Action 5-8.

Action 5-4: In the Virtuoso Analog Design Environment window, choose **Analyses — Choose…**.

Action 5-5: In the Choosing Analyses window, select **pss** in the **Analysis** field of the window.

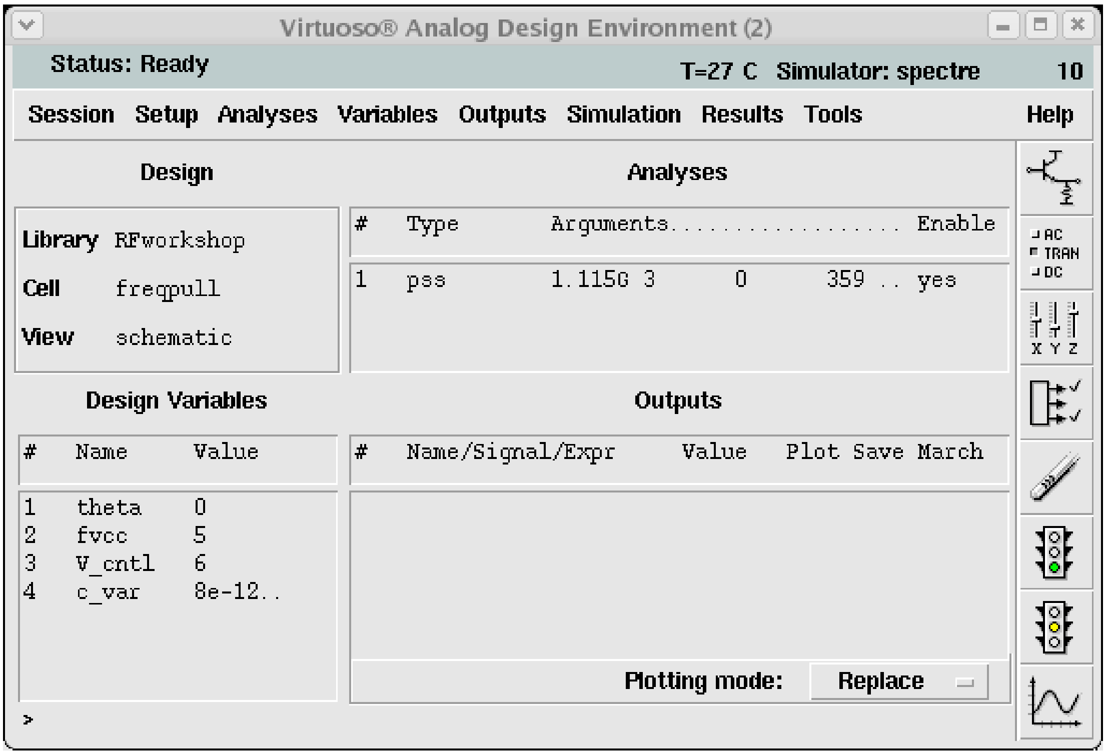
Action 5-6: Set up a swept PSS analysis with the theta parameter varying from 0 to 359 degrees. Set **Beat Frequency** = 1.115G; **Number of Harmonics** = 3; **errpreset** = **moderate**; **tstab** = **120n**; turn on **Oscillator**; set **Oscillator node** = **/Vout**; and **Reference node** = **/gnd!;** enable **Sweep**; enter theta as Variable Name; set the Sweep Range Start = 0 and Stop = 359; set Sweep Type = linear; and Number of Steps = 10.

The Periodic Steady State Analysis window looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environmer
OK
Cancel
Defaults
Apply
Help
Fundamental Tones
Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Hierarchy
Beat Frequency
Beat Period
1.115G
Auto Calculate
Output harmonics
Number of harmonics
3
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Additional Time for Stabilization (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Oscillator
Oscillator node
/net025
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Sweep
Variable
Frequency Variable?
no
yes
Variable Name
theta
Select Design Variable
Sweep Range
Start-Stop
Center-Span
Start
0
Stop
359
Sweep Type
Linear
Logarithmic
Step Size
Number of Steps
10
Add Specific Points
New Initial Value For Each Point (restart)
no
yes
Enabled
Options...

Action 5-7: Turn on **Enabled**, and click **OK** in the Choosing Analyses form.

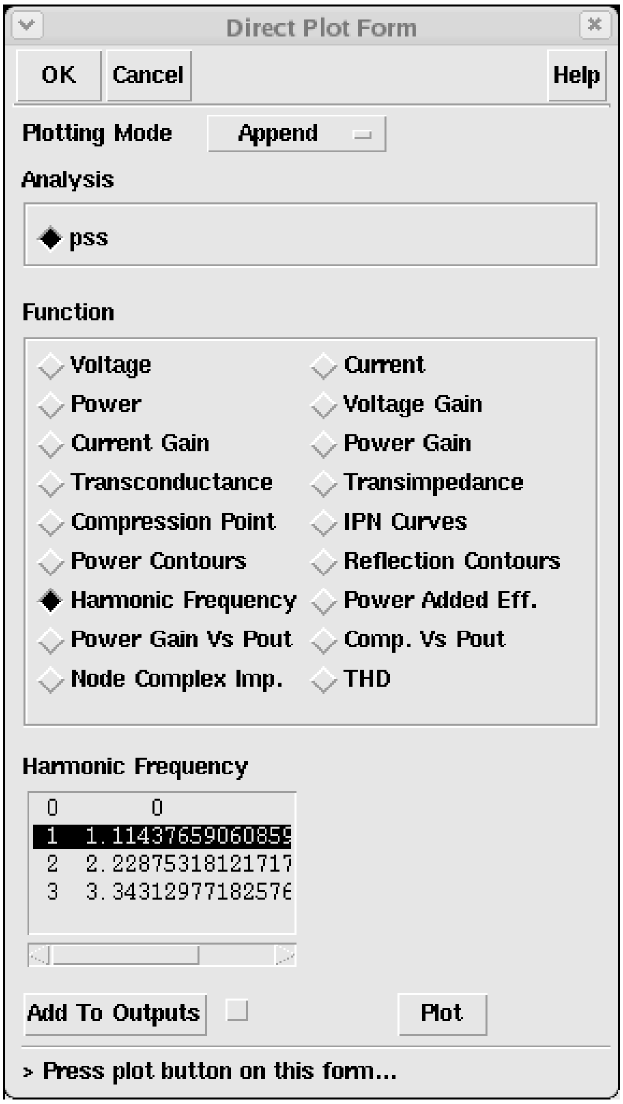
The Virtuoso Analog Design Environment window looks like this:



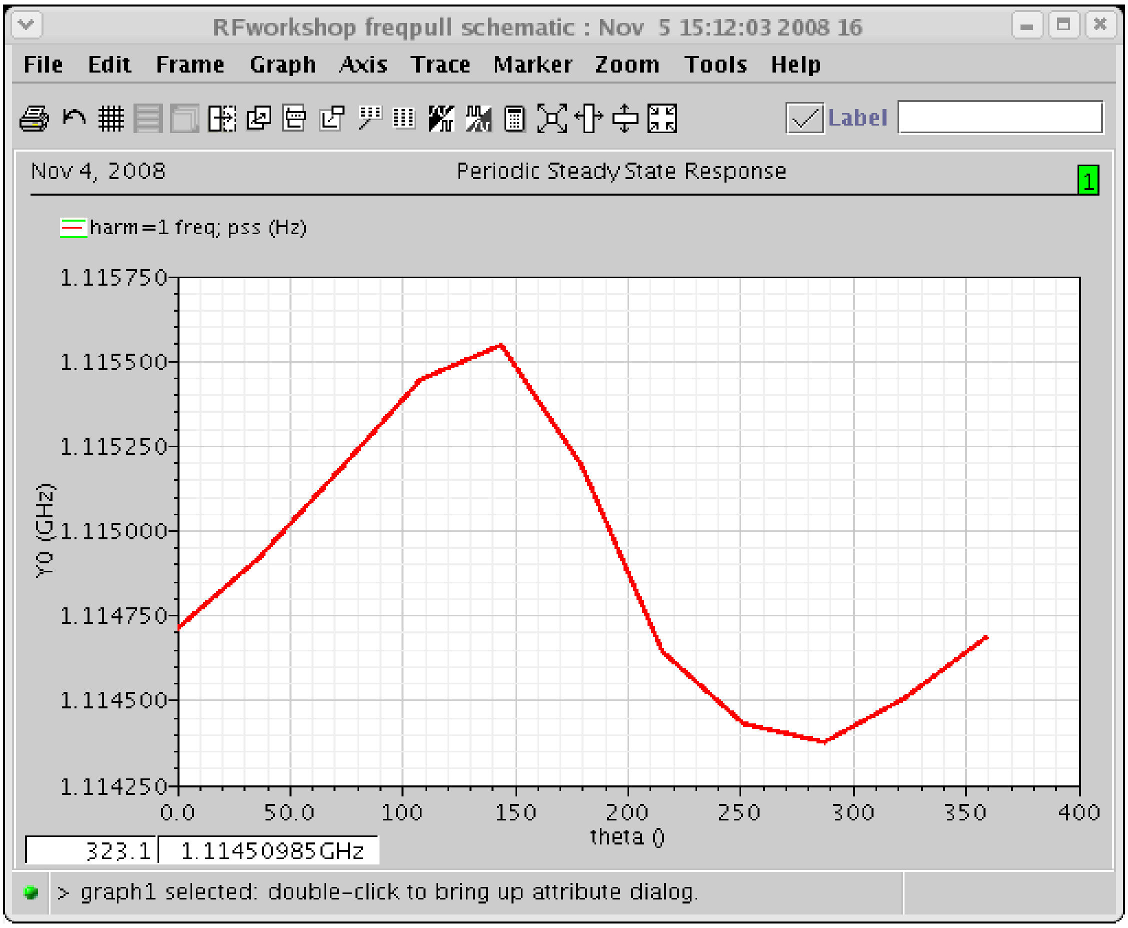
Action 5-8: In the Analog Design Environment window, choose **Simulation — Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action 5-9: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form**.

Action 5-10: In the Direct Plot Form, select **pss**, click **Harmonic Frequency**, and highlight the 1st harmonic in the **Harmonic Frequency** section. The form looks like this:



Action 5-11: Click **Plot**. The following plot appears.



The peak to peak difference in the displayed frequency in the above figure gives the load pull.

Action 5-12: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

Lab 6: Stability Analysis (PSS/Pstb)

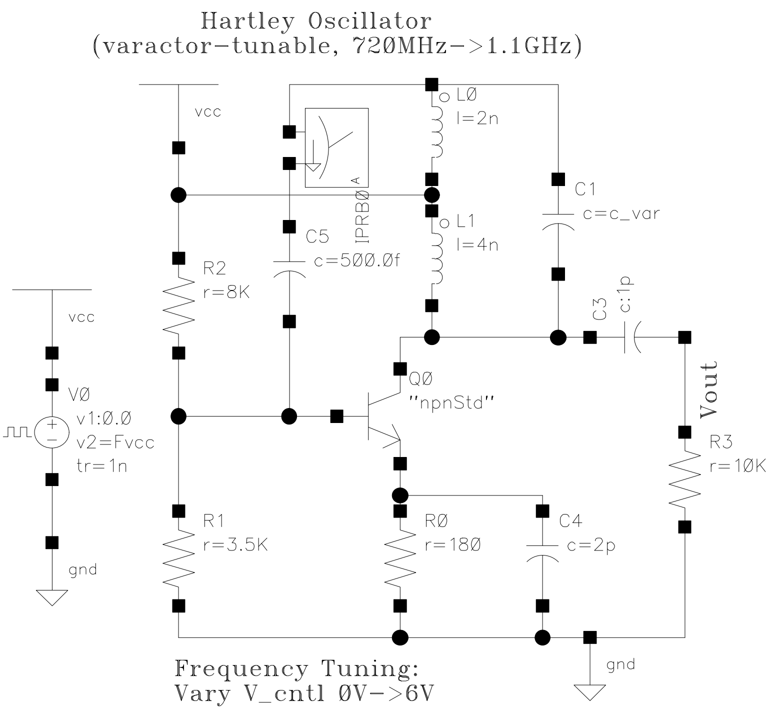
Spectre stability analysis (STB) rapidly evaluates the stability of feedback circuits. As is true for all small signal analyses, the circuit under analysis must first be linearized about a DC operating point before STB small signal analysis is performed. After the circuit is linearized about a DC operating point, STB analysis then calculates the loop gain, gain margin, and phase margin for the circuit using a subset of Nyquist criteria.

It is also important to evaluate the stability of periodic steady state regimes of nonlinear circuits such as power amplifiers, injection locked oscillators and dividers. STB analysis cannot 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. Small signal PSTB analysis calculates the loop gain, gain margin, and phase margin for circuits with a periodically time-varying operating point.

Action 6-1: In the Library Manager window, open the *schematic* view of the design *oschartley_pstb* in the library *RFworkshop*

The following figure shows the modified Hartley Oscillator schematic for stability analysis.



Notice that in the above schematic, a current probe, *IPROB0*, is inserted in the feedback loop.

The stability evaluation 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 a 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.

Then, the small stimulus is applied to compute the loop gain of the zero sideband with a probe component. The local stability can be evaluated using gain margin, phase margin, or a Nyquist plot of the loop gain. To perform PSTB analysis, you must use a probe instance and specify it with the probe parameter.

Action 6-2: From the *oschartley_pstb* schematic, start the Virtuoso Analog Design Environment with the **Tools — Analog Environment** command.

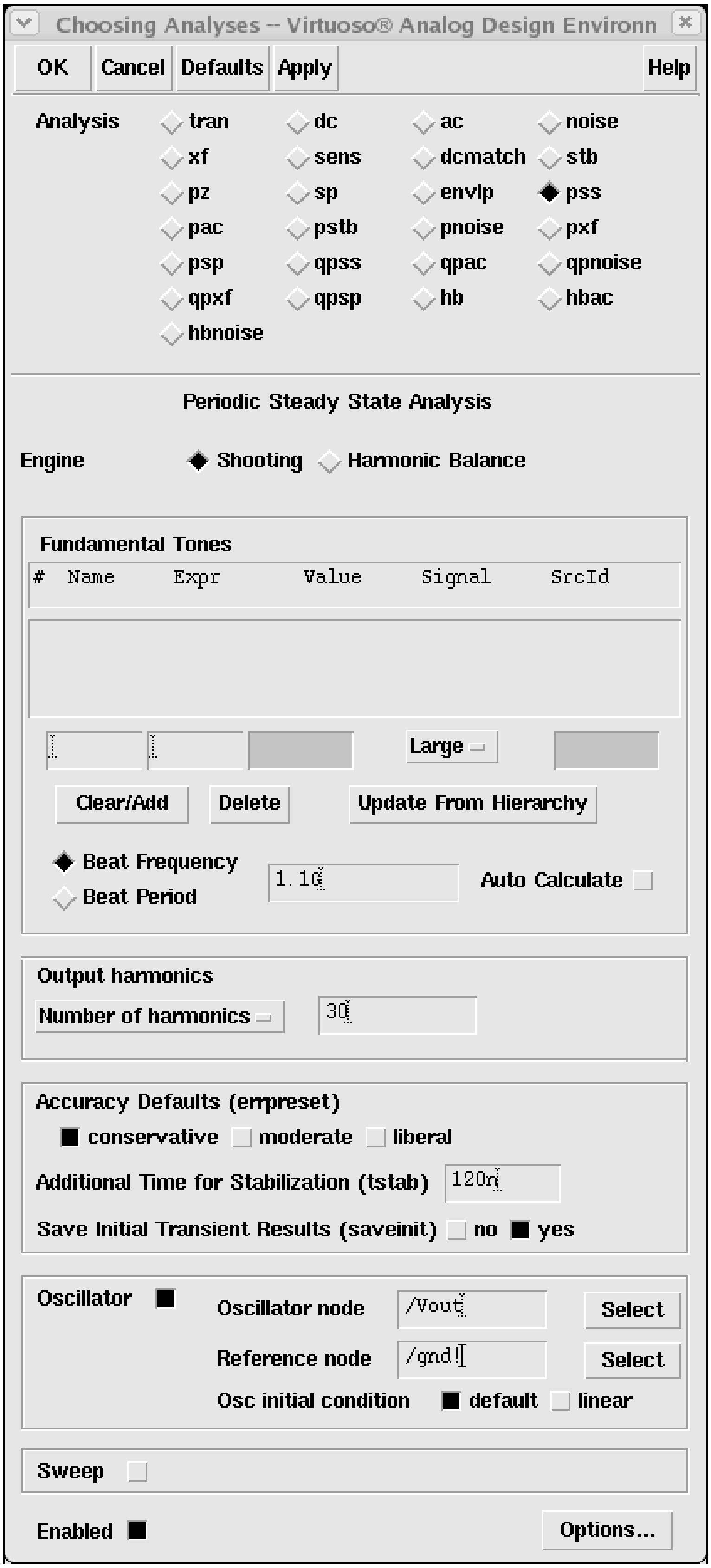
Action 6-3: (Optional) Choose **Session — Load State**, select **Cellview** in **Load State Option** and load the "**Lab6_stability_PSTB**" state, and skip to Action 6-10.

Action 6-4: In the Virtuoso Analog Design Environment window, choose **Analyses — Choose…**.

Action 6-5: In the Choosing Analyses window, select **pss** in the **Analysis** field.

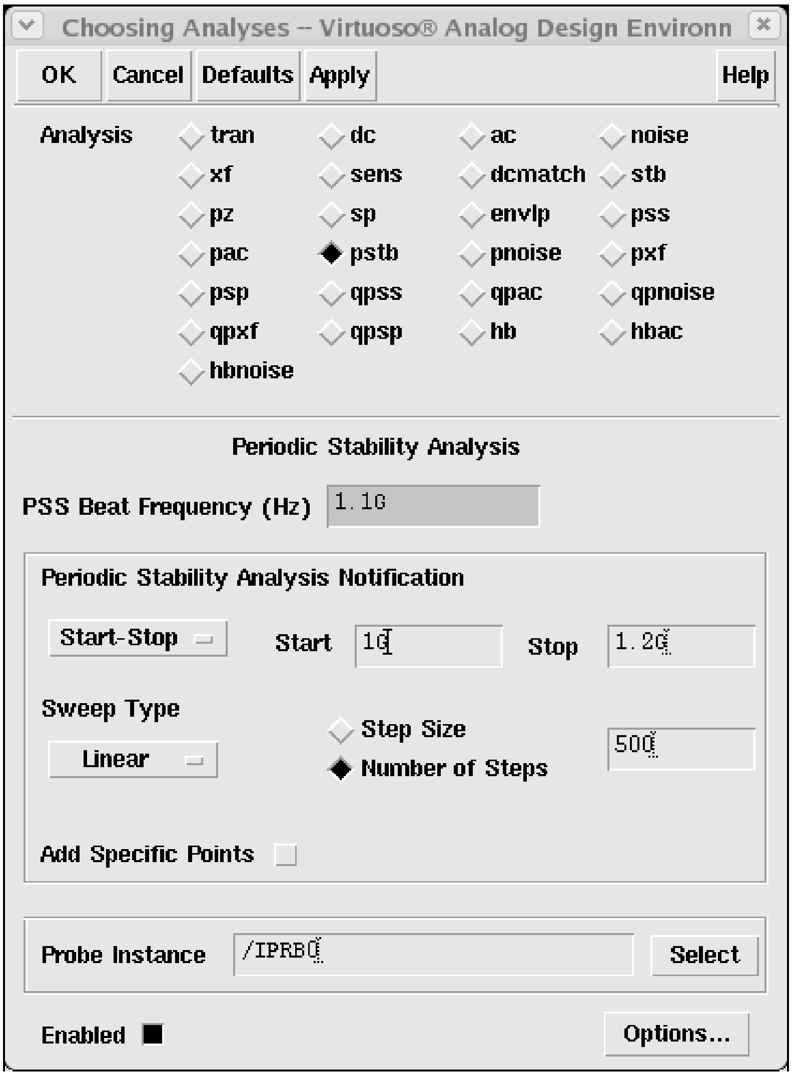
Action 6-6: Set up the PSS analysis. Set **Beat Frequency** = 1.1G; Number of **Harmonics** = **30**; **errpreset** = **conservative**; **tstab** = **120n**; turn on **Oscillator**; set **Oscillator node** = **/Vout**; and **Reference node** = **/gnd!.**

The Periodic Steady State Analysis window looks like this:


Choosing Analyses -- Virtuoso® Analog Design Environn
OK
Cancel
Defaults
Apply
Help
Analysis
tran
dc
ac
noise
xf
sens
dcmatch
stb
pz
sp
envlp
pss
pac
pstb
pnoise
pxf
psp
qpss
qpac
qpnoise
qpxf
qpsp
hb
hbac
hbnoise
Periodic Steady State Analysis
Engine
Shooting
Harmonic Balance
Fundamental Tones
Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Hierarchy
Beat Frequency
Beat Period
1.1G
Auto Calculate
Output harmonics
Number of harmonics
30
Accuracy Defaults (errpreset)
conservative
moderate
liberal
Additional Time for Stabilization (tstab)
120n
Save Initial Transient Results (saveinit)
no
yes
Oscillator
Oscillator node
/Vout
Select
Reference node
/gnd!
Select
Osc initial condition
default
linear
Sweep
Enabled
Options...

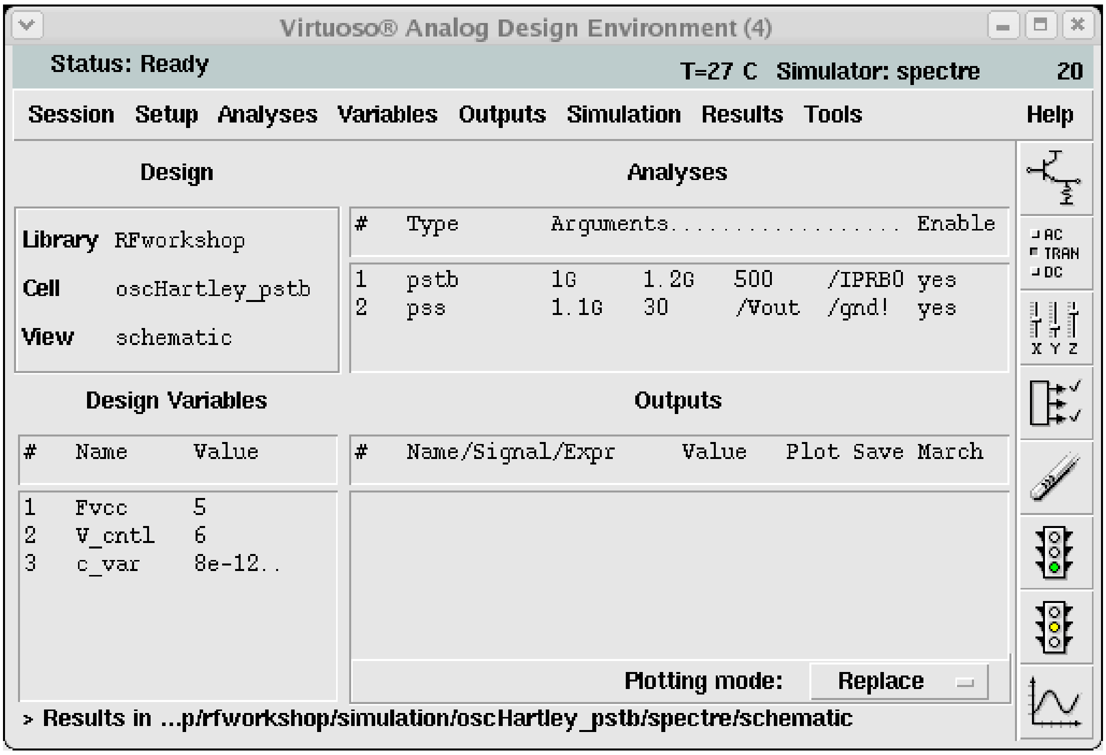
Action 6-7: Turn on **Enabled**, and click **pstb** in the Choosing Analyses form.

Action 6-8: In the Periodic Stab Analysis form, set the **Start** to 1G to and **Stop** to 1.2G, **Sweep Type**=**Linear**, **Number of Steps**=500, and select **/IPROB0** as **Probe Instance.**



Action 6-9: Turn on **Enabled**, and click **OK** to close the Choosing Analyses form.

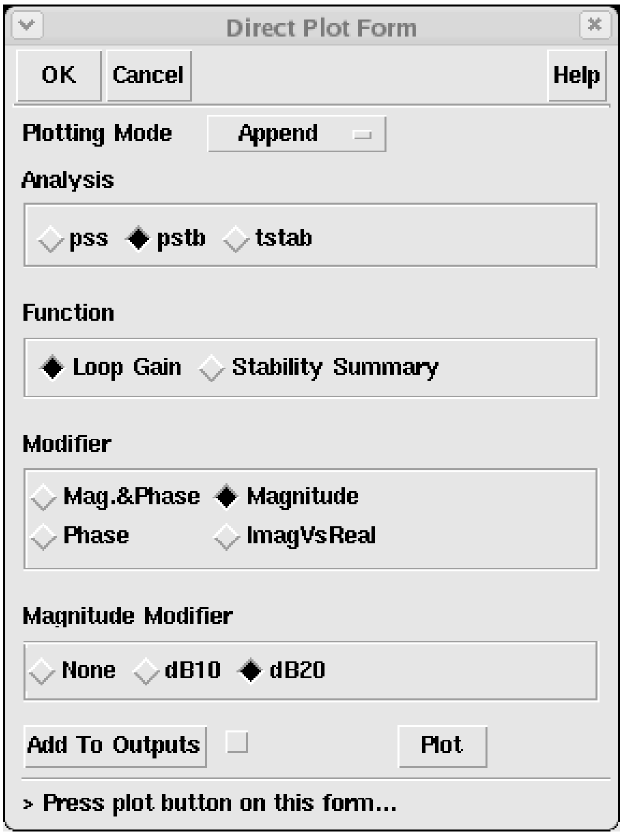
Your Virtuoso Analog Design Environment window looks like this:



Action 6-10: In the Analog Design Environment window, choose **Simulation — Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

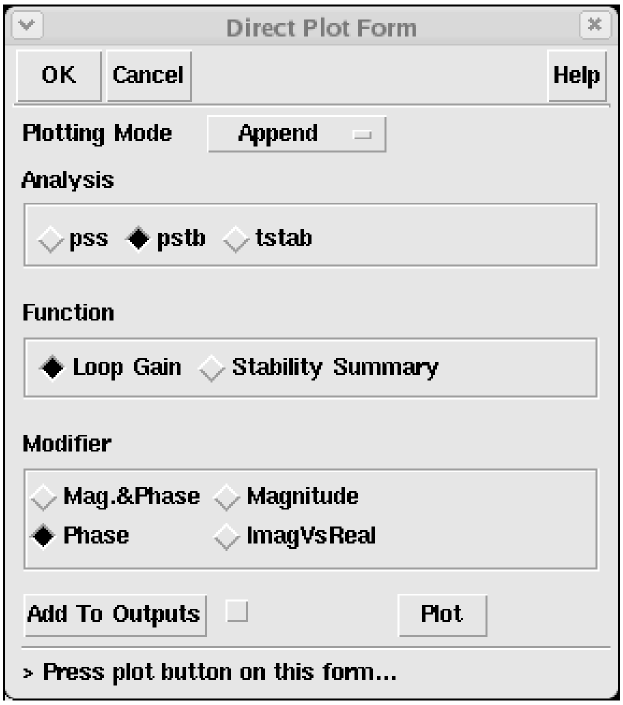
Action 6-11: In the Virtuoso Analog Design Environment window, choose **Results — Direct Plot — Main Form**.

Action 6-12: In the Direct Plot Form, select **pstb**, and set up the form as follows:

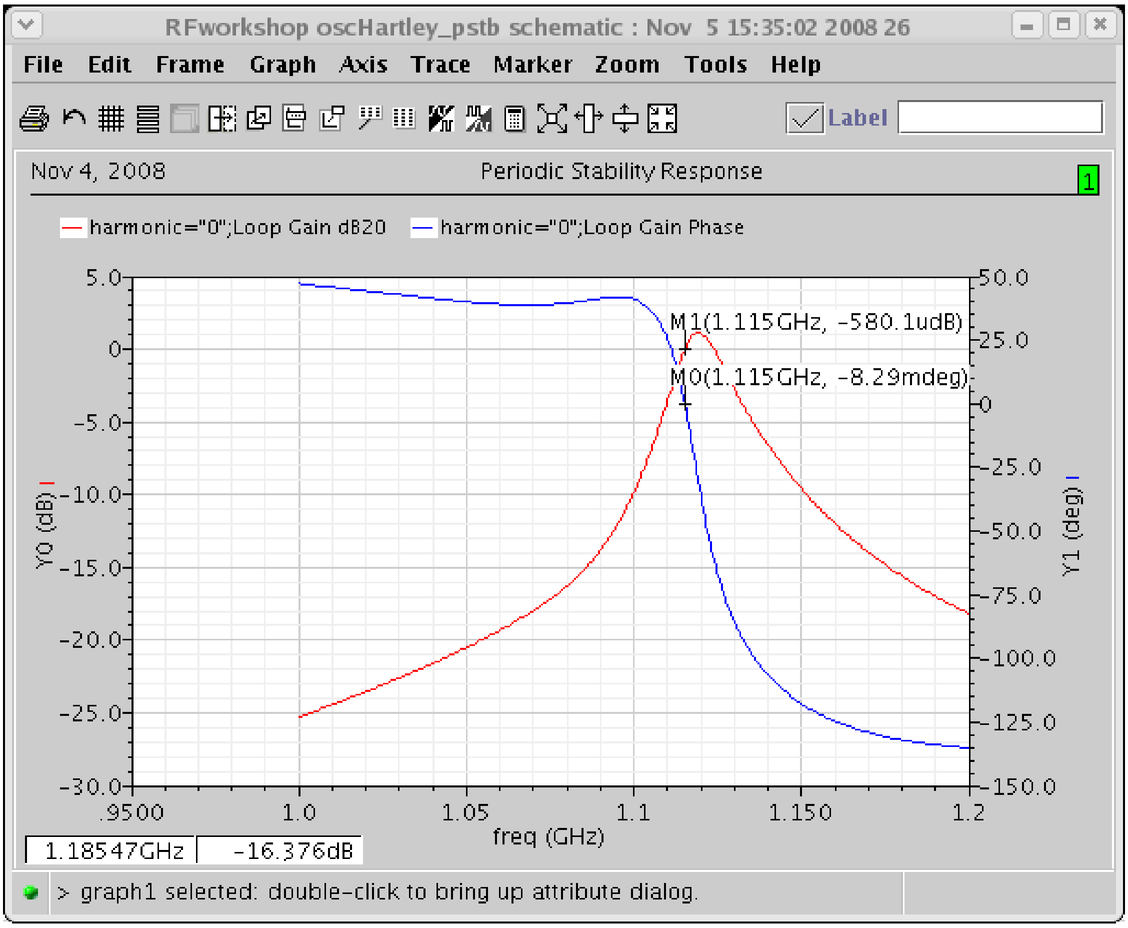


Action 6-13: Click **Plot**.

Action 6-14: In the direct plot form, select **Phase** as **Modifier**. Click **Plot**.



You get the following plot:



For an ideal oscillator, the loop gain at the oscillating frequency, *f0*, should be 1, so the dB20 (loopgain) and the Phase (loopgain) should both be zero at *f0* and the phase should change abruptly at *f0*. In practice, both dB20 (loopgain) and the Phase (loopgain) should be close to zero.

For this autonomous circuit, the PSS analysis gives *f0* = 1.11515G (check the simulation log file for details). From the above figure, the PSTB analysis gives dB20 (loopgain) = 0 and Phase (loopgain) = 0.

To setup an STB simulation, you can also get the stability information for this circuit. The results show that a PSTB analysis gives more accurate stability information for nonlinear circuits than does a STB analysis. For more details, please refer to *Application Note: Stability Analysis of Linear Periodical Time-Varying Circuit Using SpectreRF PSTB Analysis.*

Action 6-15: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

Conclusions

This workshop describes some of the most useful measurements for VCOs. SpectreRF measurements such as Frequency Pushing, Frequency Pulling, Tuning Sensitivity, Power Dissipation, and Linearity are discussed. The new Harmonic Balance engine is introduced and is used by some of these measurements.

References

[1] Ken Kundert, "Introduction to RF Simulation and Its Application", www.designers-guide.com

[2] Ken Kundert, "Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers", www.designers-guide.com

[3] Ken Kundert, ”Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers”, *The Designer’s Guide,* www.designers-guide.com, 2005

[4] Ken Kundert, Manolis Terrovitis, "Converting Phase-Noise to Jitter", Cadence report.