# SpectreRF Workshop

## VCO Design Using SpectreRF

## MMSIM6.0USR2

November 2005

# Contents

# Voltage Controlled Oscillator Design Measurements

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

## Purpose

This workshop presents how to use SpectreRF in the Analog Design Environment to measure parameters which are important in design verification of voltage controlled oscillators, or VCO. New features of MMSIM6.0USR2 are included.

## Audience

Users of SpectreRF in the Analog Design Environment.

## Overview

This application note provides the user with a basic set of common measurements for VCO.

# Introduction to VCOs

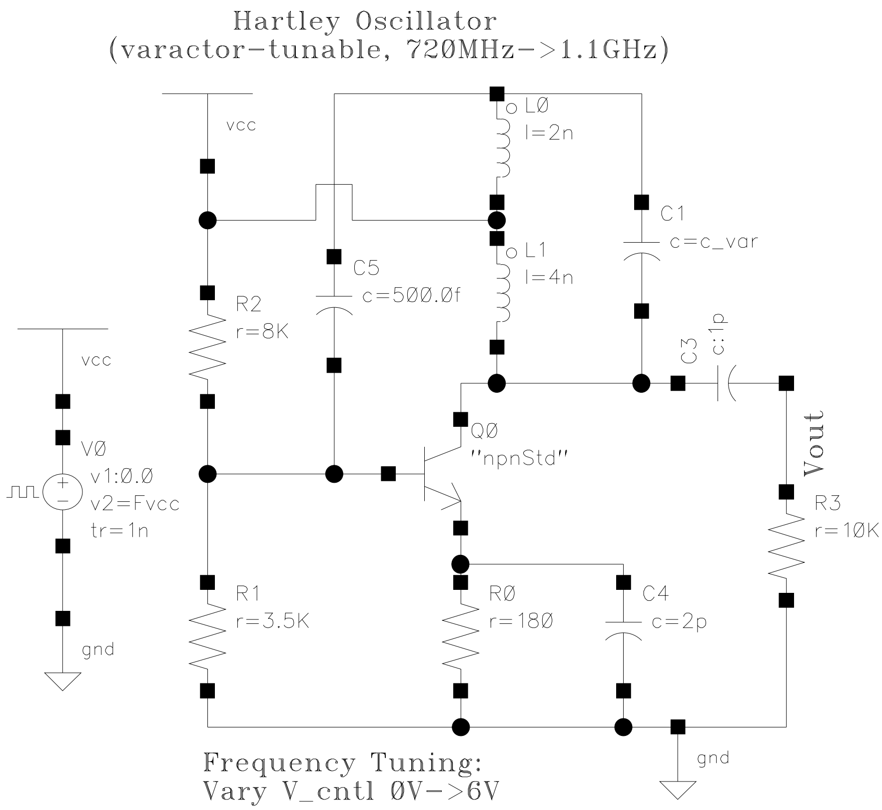
Oscillators generate a reference signal at a particular frequency. In voltage controlled oscillators, or VCOs, the frequency of the output varies in proportion to some control signal. Oscillators are generally used in RF circuits to generate the local oscillator, or LO, signal for mixers. VCOs are used 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, $C_{\text{var}}$, is a function of the applied voltage as shown in the following equation.

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

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 $C_{\text{var}}$ is inversely proportional to V, and Fo is inversely proportional to $C_{\text{var}}$, the oscillation frequency is proportional to V. In other words, as you increase V, $C_{\text{var}}$ 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 since 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 since 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.

There are many parameters you must evaluate that are not covered in this workshop. The parameters that are covered in this workshop are listed and described in the following labs.

We'll begin our examination of the flow by bringing up the Cadence Design Framework II environment and look at a full view of our reference design:

Change directory to…

Action: cd to **./vco** directory

Action: Invoke tool **icfb&**

Action: In the CIW window, select **Tools->Library Manager…**

## Lab1: Output Frequency, Output Power, Phase Noise and Jitter Measurement (Pnoise with shooting or Flexible 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 SpectreRF will compute the correct period. The output power of a VCO is typically expressed in dBm.

Phase noise is random phase variation in the VCO's output oscillating signal. 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. It causes an increase in the channel separation. Spectre RF 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, two Pnoise analyses (modulated Pnoise analyses) are run by SpectreRF to extract jitter information.

Action1-1: In the Library Manager window, open the *schematic* view of the design *oscHartley* in the library *RFworkshop*.

To start an oscillator circuit, apply initial conditions or kickstart the oscillator. This is not necessary for the *oscHartley* circuit in this lab, because it contains a pulse source that kickstarts the oscillator. Therefore, no real initial conditions are necessary.

Action1-2: In the Virtuoso Schematic Editing window, select **Tools->Analog**

**Environment**

Action1-3: You can choose **Session—Load State** in Virtuoso Analog Design Environment load state "**Lab1_Pnoise_shooting**", then skip to Action1-12 or …

Action1-4: In Analog Design Environment window, select **Analyses->Choose…**

Action1-5: In the Choosing Analyses window, select the **pss** button in the **Analysis** field of the window. Set **Fundamental Frequency** of **600 MHz** based on estimation. Set the **Number of Harmonics** to **3** for this oscillator. Set the **tstab** to add additional time for the oscillator to converge on its operating frequency. Click on yes to save the initial transient results. The **Oscillator** button should be enabled. When activated, this 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.

Action1-6: Click the **Options** button at the bottom of the Choosing Analyses form.

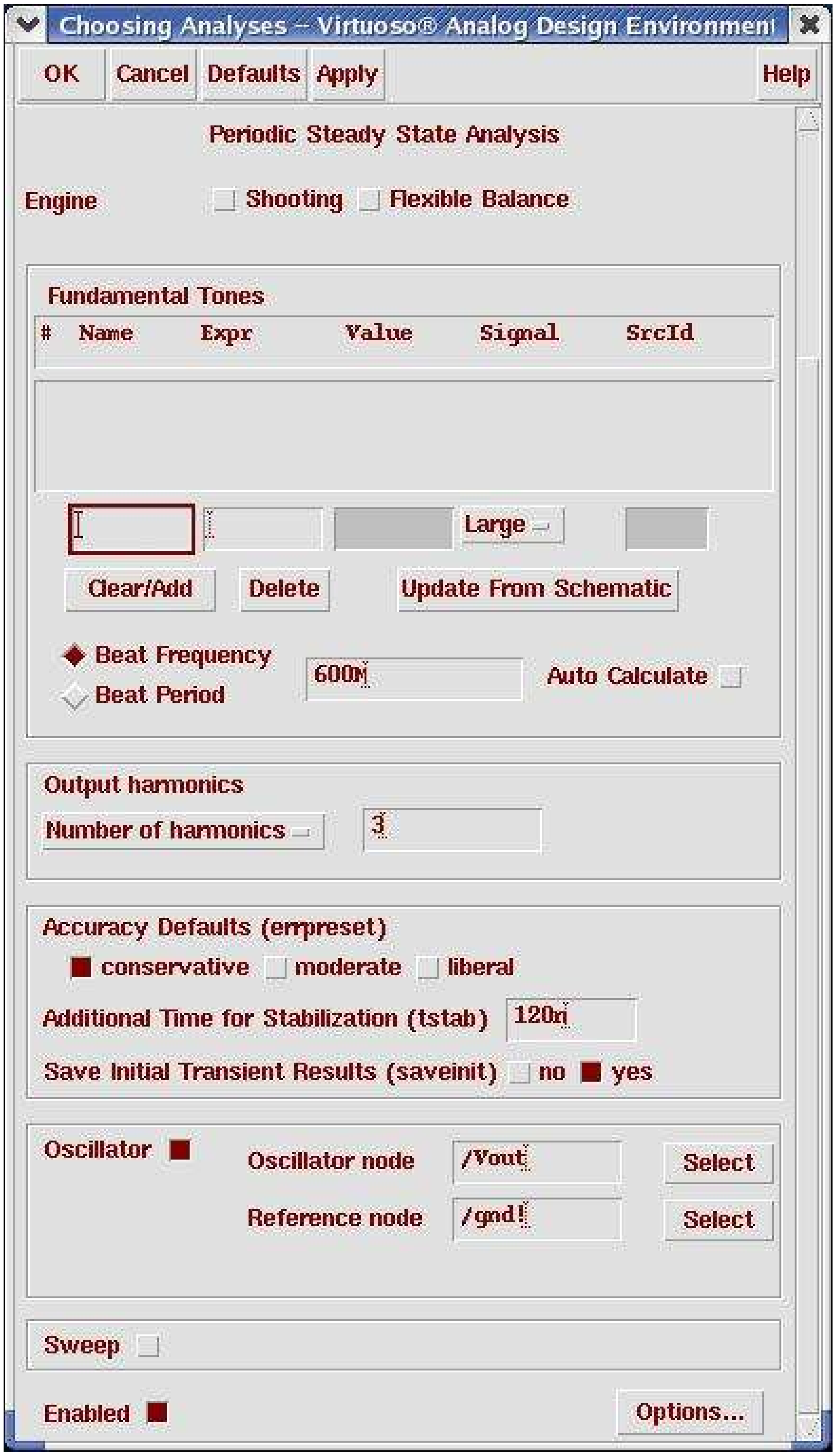
The Periodic Steady State Options form appears.

Action1-7: In the INTEGRATION METHOD PARAMETERS section, verify that the method is set to **gear2only**.

Action1-8: Click **OK** in the PSS Options form.

Action1-9 Click **Apply** in the Choosing Analyses form.

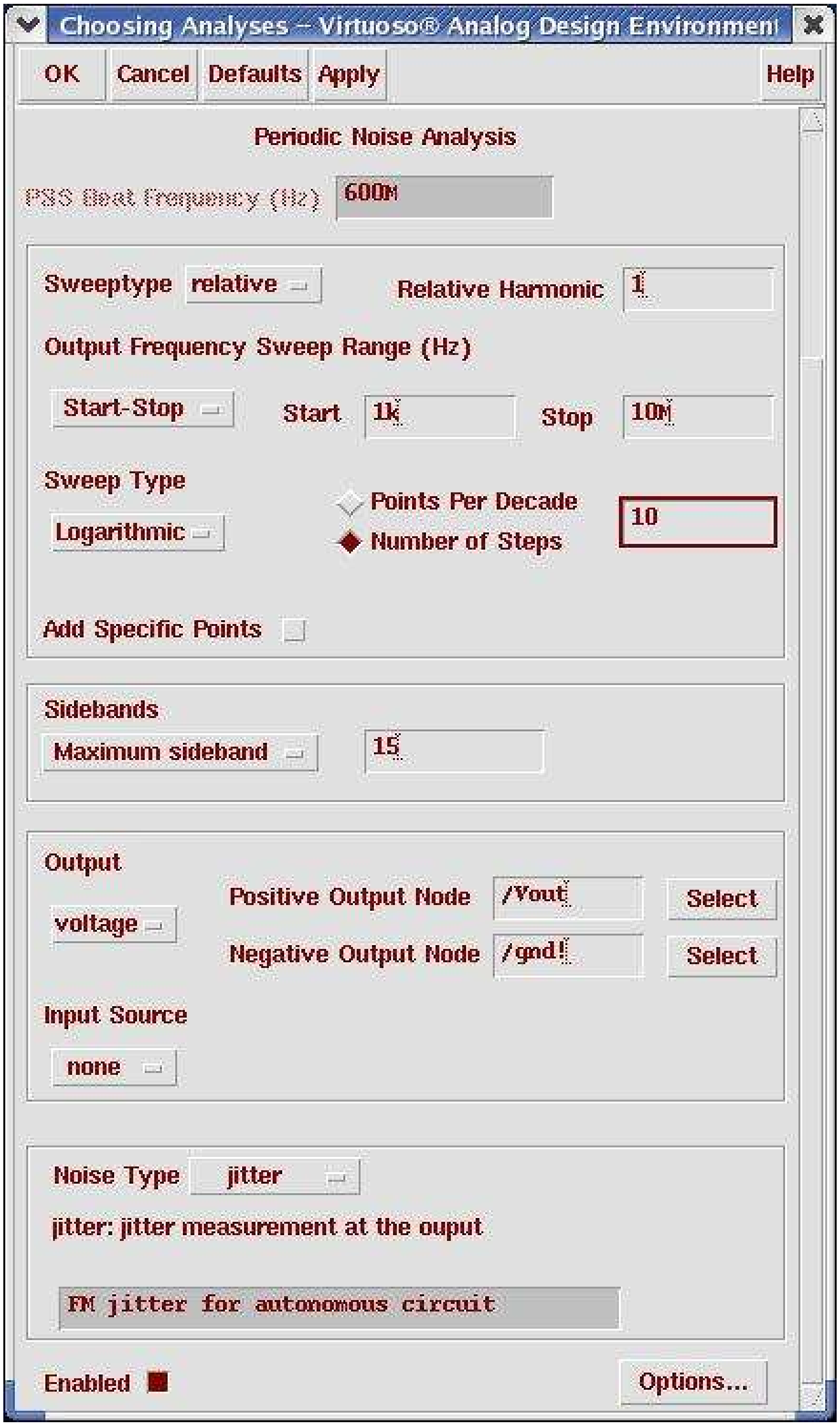
The PSS analysis form should look like this:




Action1-10: Click the **pnoise** button. In the Periodic Noise Analyses form, set **Sweep type** to **relative**, and **relative Harmonic** to **1**. Set **Start-stop** to **1K** 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 **Maxim**

**sideband** to **15**. Set Output to **voltage**. Select the Positive and Negative nodes in the oscillator schematic. Set **Input Source** to **none**.

The form should look like this:



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*, will be 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 one has to realize that at the frequencies close to the LO the small signal approximation will break 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) since we are only interested in the phase noise around the output harmonic. The bandwidth of the circuit will often be the factor here too. More frequency points are always helpful to increase an accuracy of the jitter computations. You will have to trade off the frequency range, number of point per decade and the maximum sidebands that will be 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.

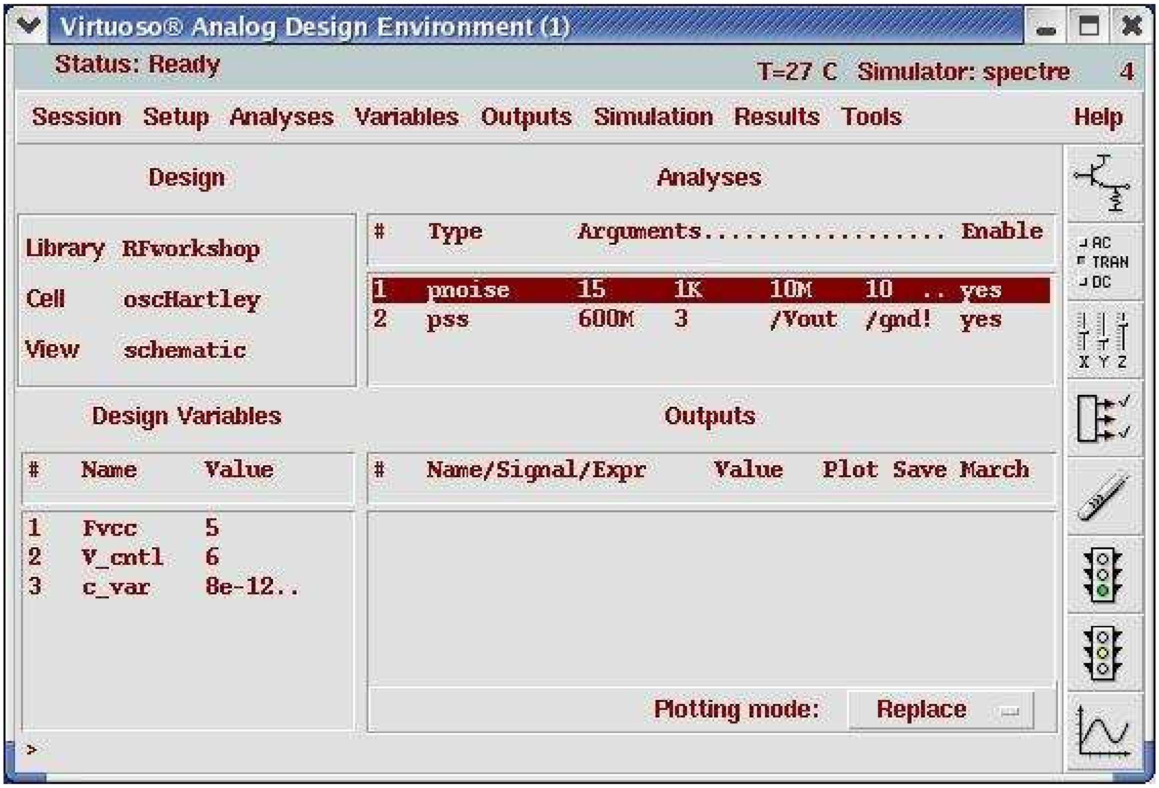
Because the up-converted noise appears at the oscillator output, the *Vout* node was selected as the **Positive Output Node** in this analysis. The noise power of each noise contributor in the circuit is stored, but the “output node of interest” needs to be specified to tell the Virtuoso® Spectre® RF 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 will be used to compute the jitter. Since we are using the phase noise to calculate the jitter, the netlist will have two pnoise analyses in it. for pnoise jitter for autonomous, pnoise modulated is run in the background. Consequently, pnoise modulated results are available in the direct plot form.

Action1-11: Make sure that the Enabled button is on. Click OK in the Choosing Analyses form.

Now your Virtuoso Analog Design Environment looks like this:

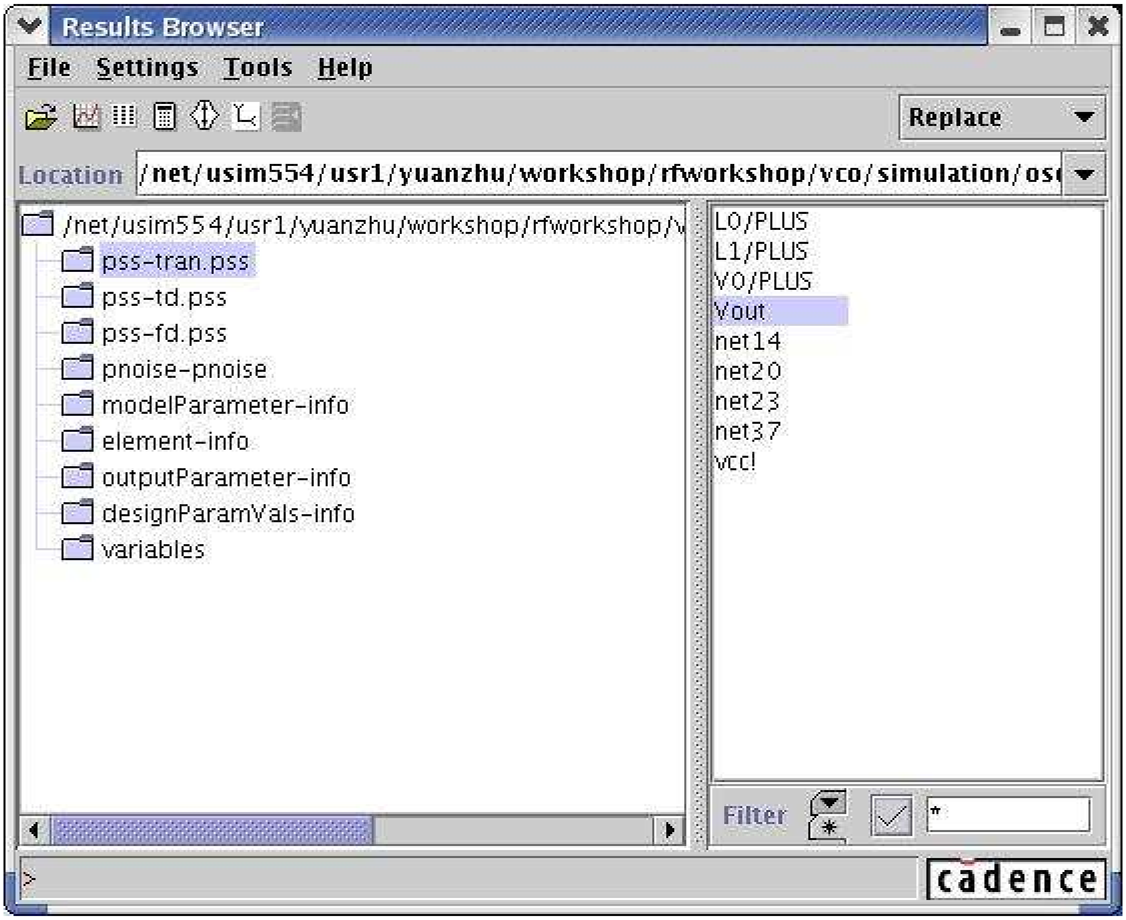



Action1-12: Choose **Simulation—Netlist and Run** to start the simulation or click on the **netlist and Run** icon in the Virtuoso Analog Design Environment window.

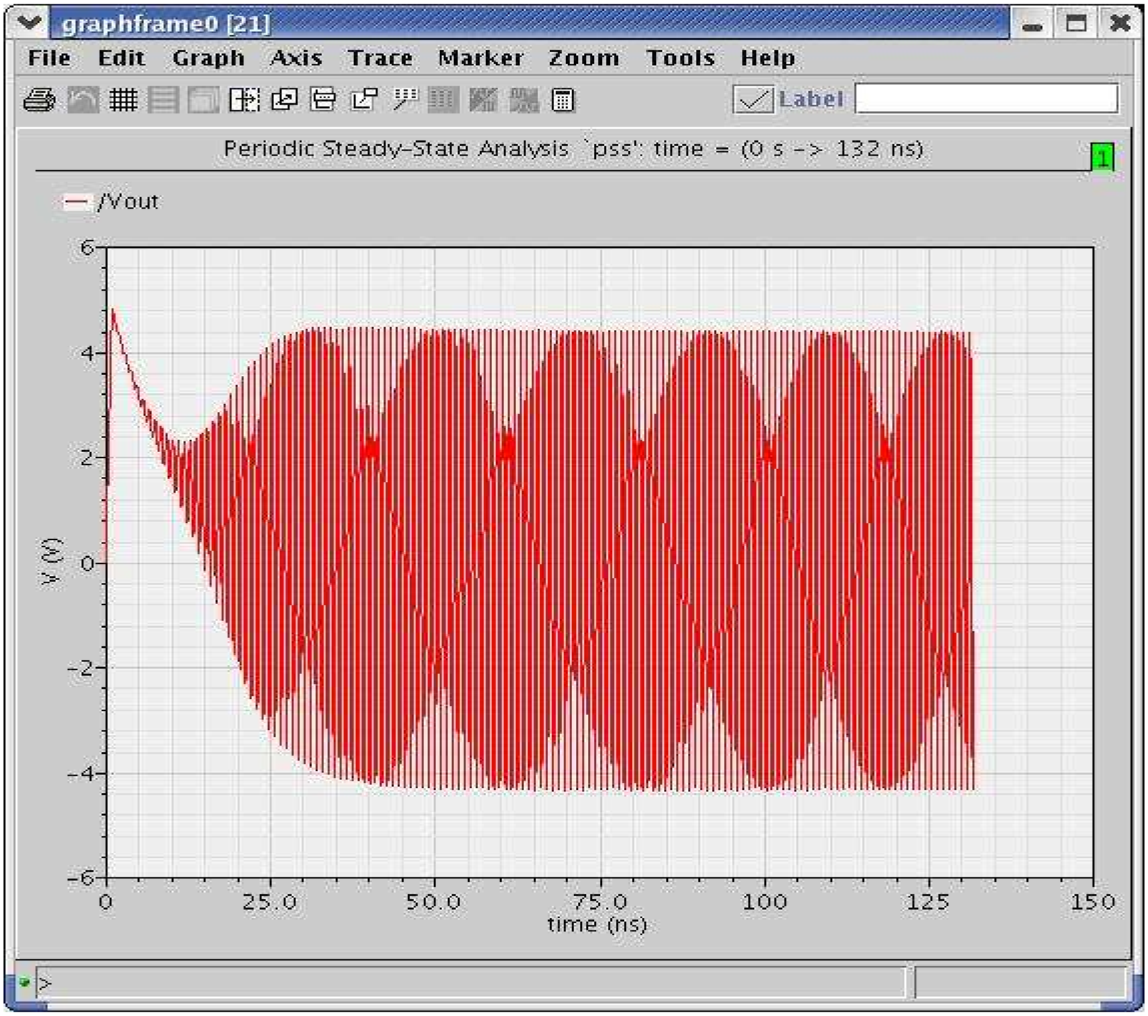
After the simulation has finished, plot the simulation results.

Action1-13: In the Virtuoso Analog Design Environment window, choose **Tools—Results Browser**.

Action1-14: In the Results Browser form, expand **pss-tran.pss.**




Action1-15: Double click on Vout. The *Vout* transient node voltage appears in the Waveform window.

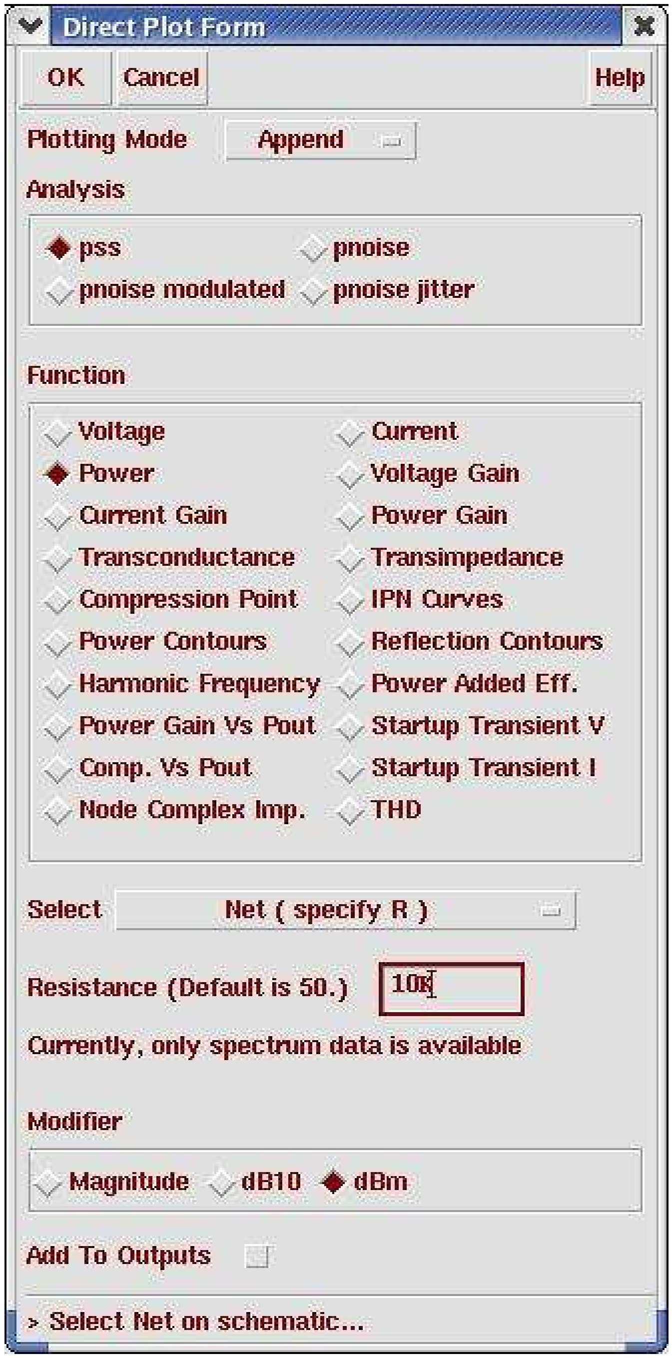



Action1-16: Close the waveform window.

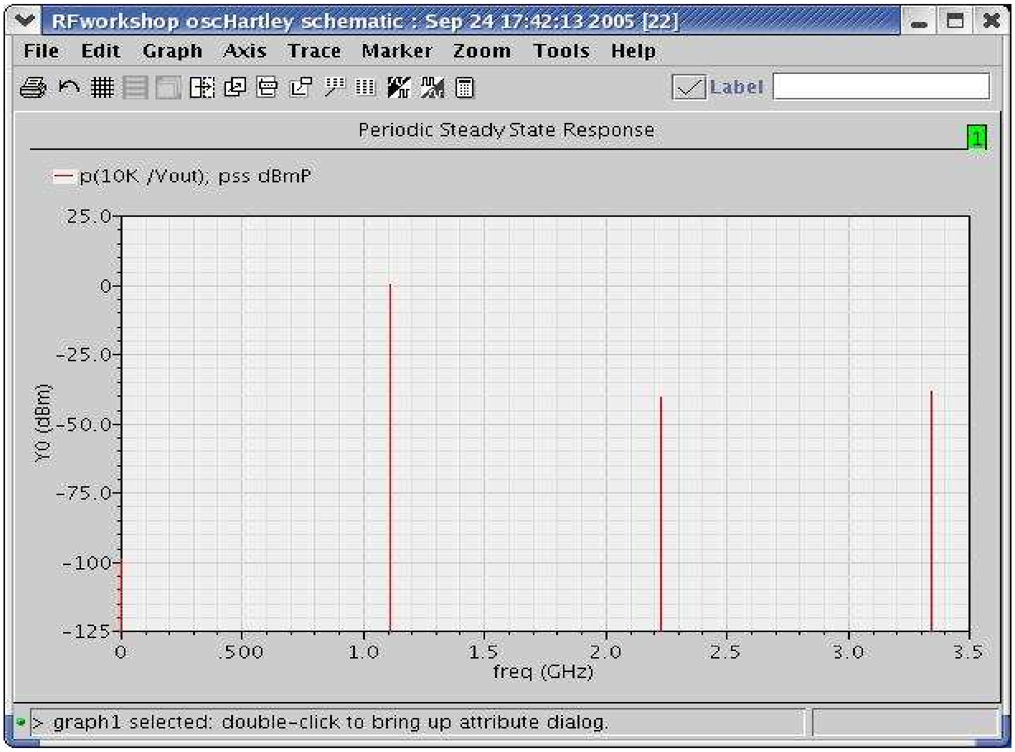
Action1-17: In the Virtuoso Analog Design Environment window, choose **Results—Direct Plot—Main Form** to plot the calculated oscillation frequency, output power and output noise.

The Direct Plot form appears.

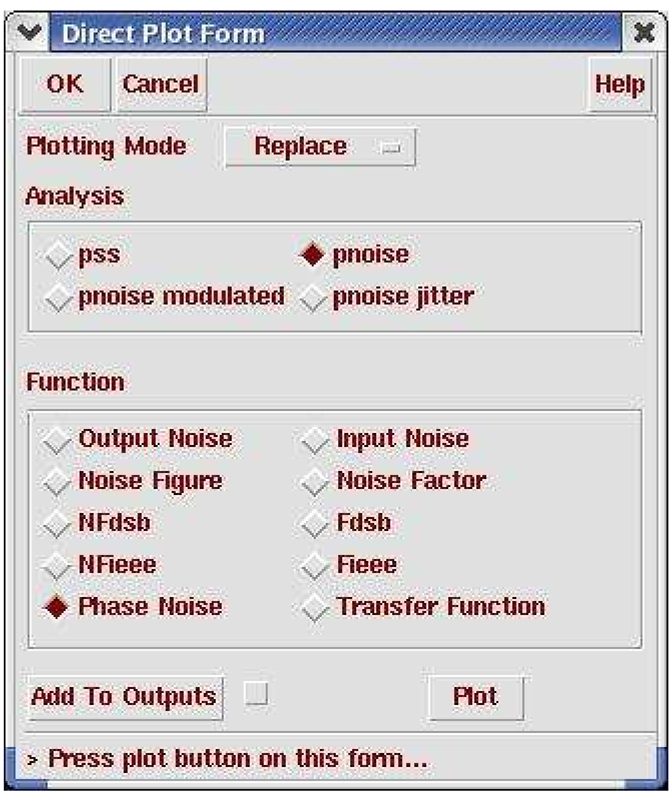
Action1-18: In Direct Plot Form window, choose **pss** as the Analysis type and configure the form as follows:



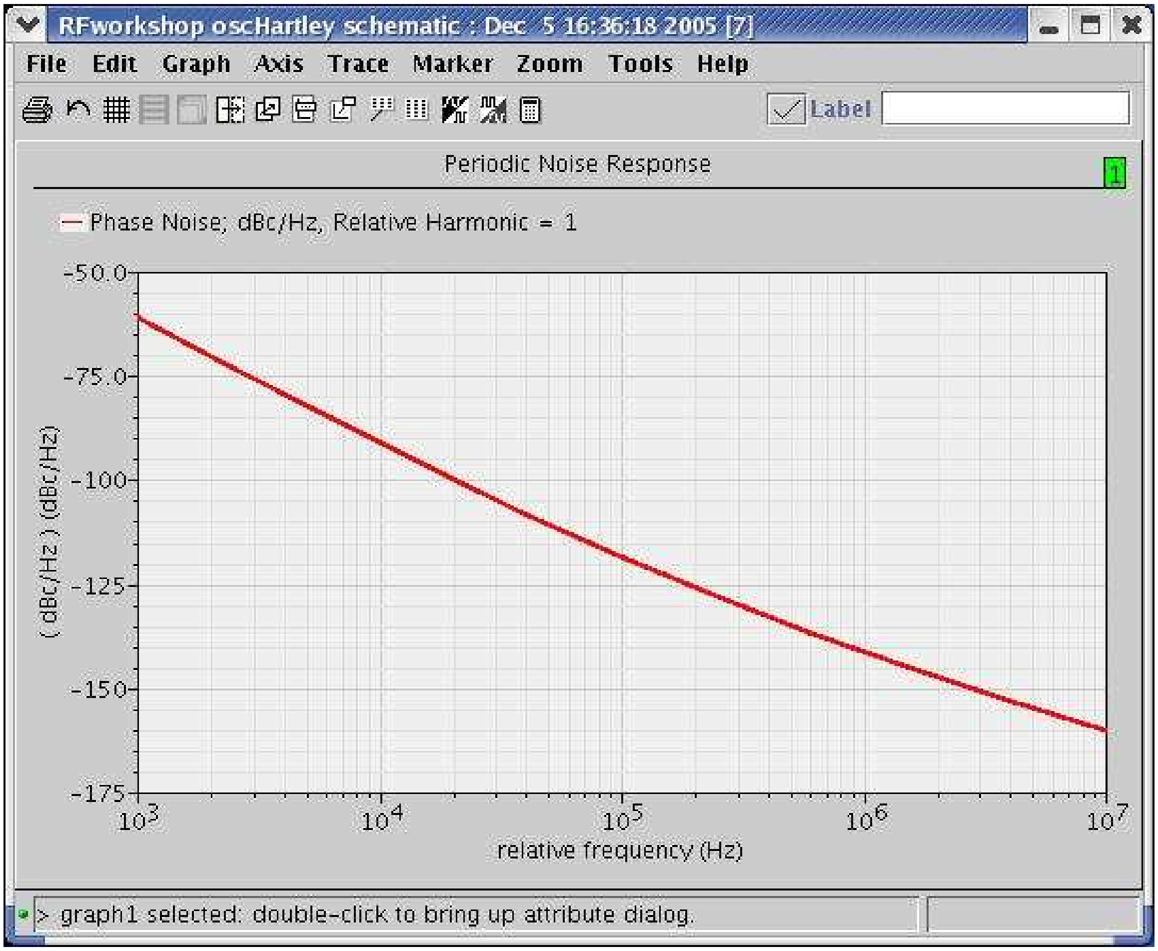

Action1-19: Select net Vout on the schematic. The results show the oscillation frequency is around 1.1G, and output power is around 0 dBm.




Action1-20: In the Direct Plot form, change the Plot Mode to Replace and the Analysis type to **pnoise**. Change the Function to **Phase Noise**, and click the **Plot** button.



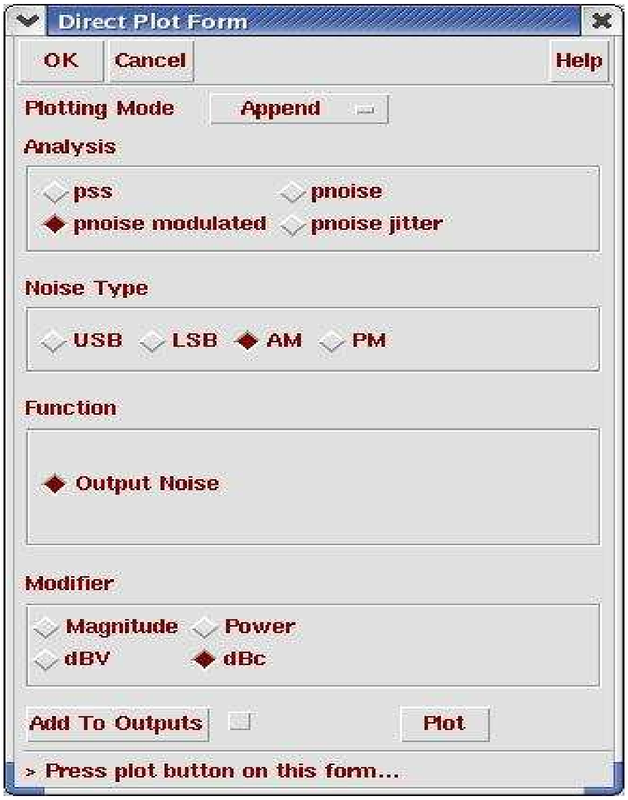
Now you should get the following waveforms:



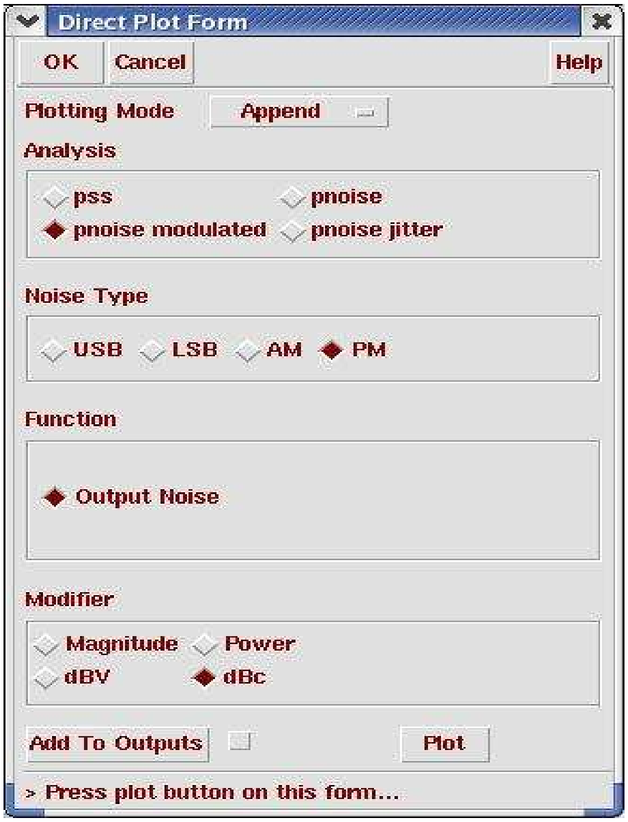

SpectreRF determines autonomous Jitter by first determining noise versus frequency with correlations (modulated Pnoise). When running pnoise modulated, SpectreRF compute correlation between sidebands by using pnoise=correlation in the background. Using spectreRF Pnoise modulated analysis, the user can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

Action1-21: In the waveform window, click the Add Subwindow icon.

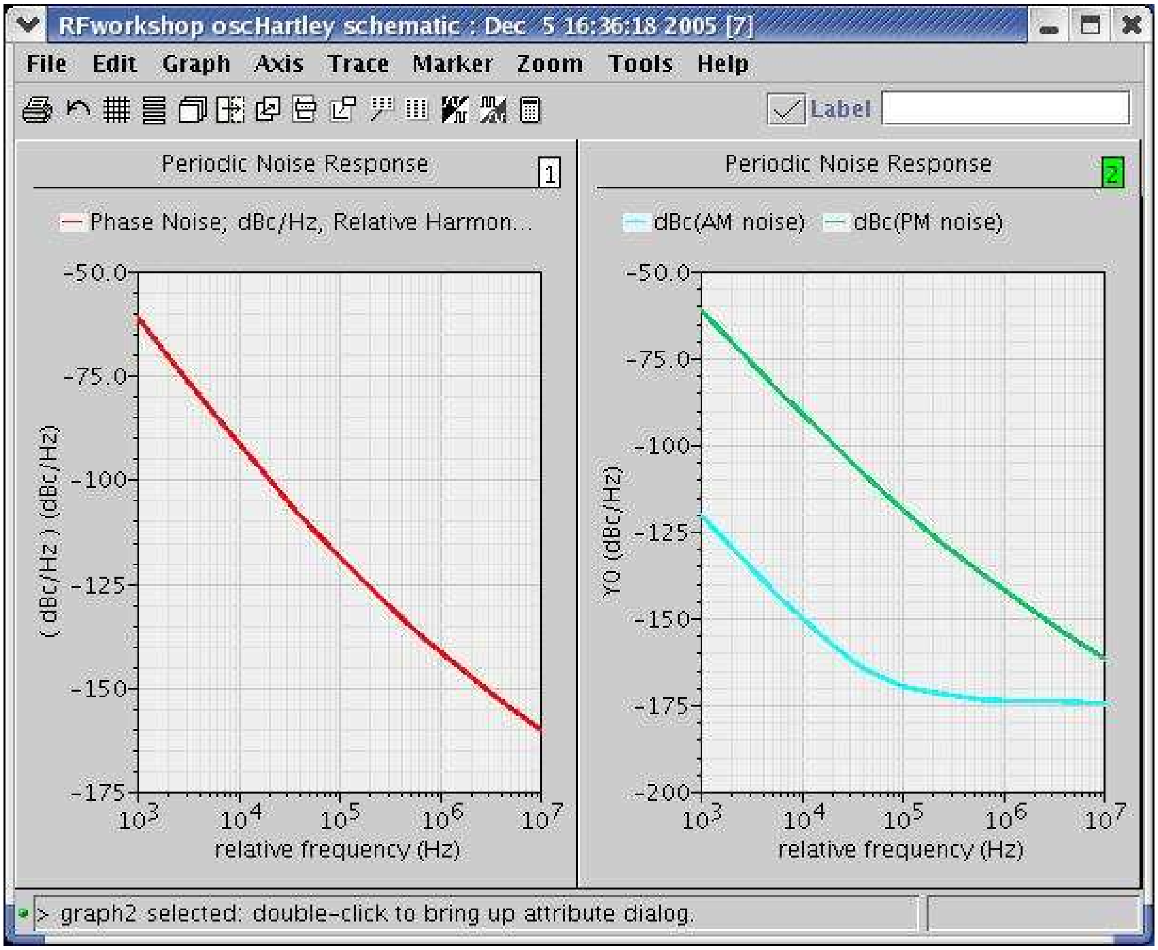
Action1-22: In the Direct Plot form, In the Direct Plot form set the Plotting Mode to **Append**. Select **pnoise modulated**. Select **AM**. Select **dBc**. Push the **Plot** button.




Action1-23: In the Direct Plot form, Select PM and push the Plot button.



The waveform window updates.

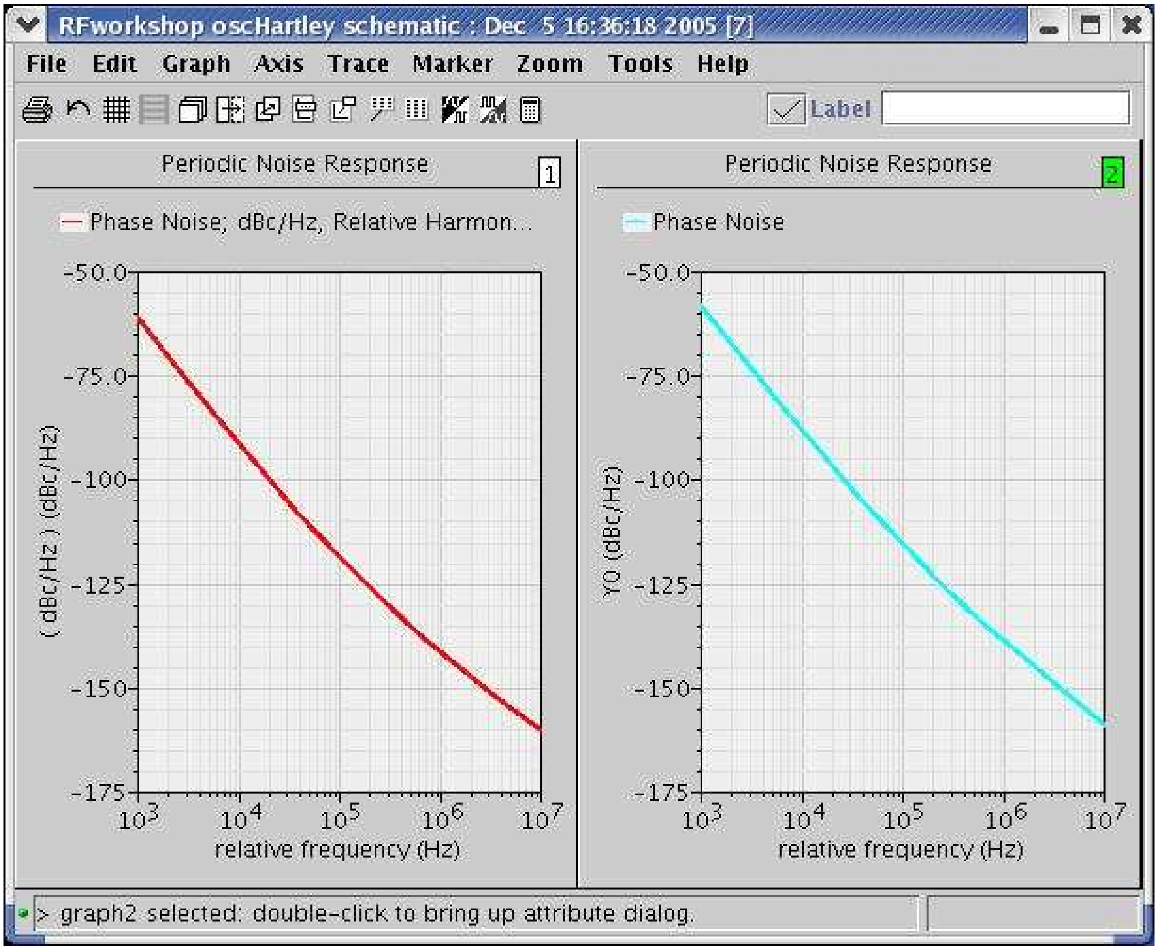



Action1-24: Compare the total phase noise to the PM noise. Move your mouse cursor in the waveform window and read the phase noise plot and the PM noise at 10K Hz. They are the same.

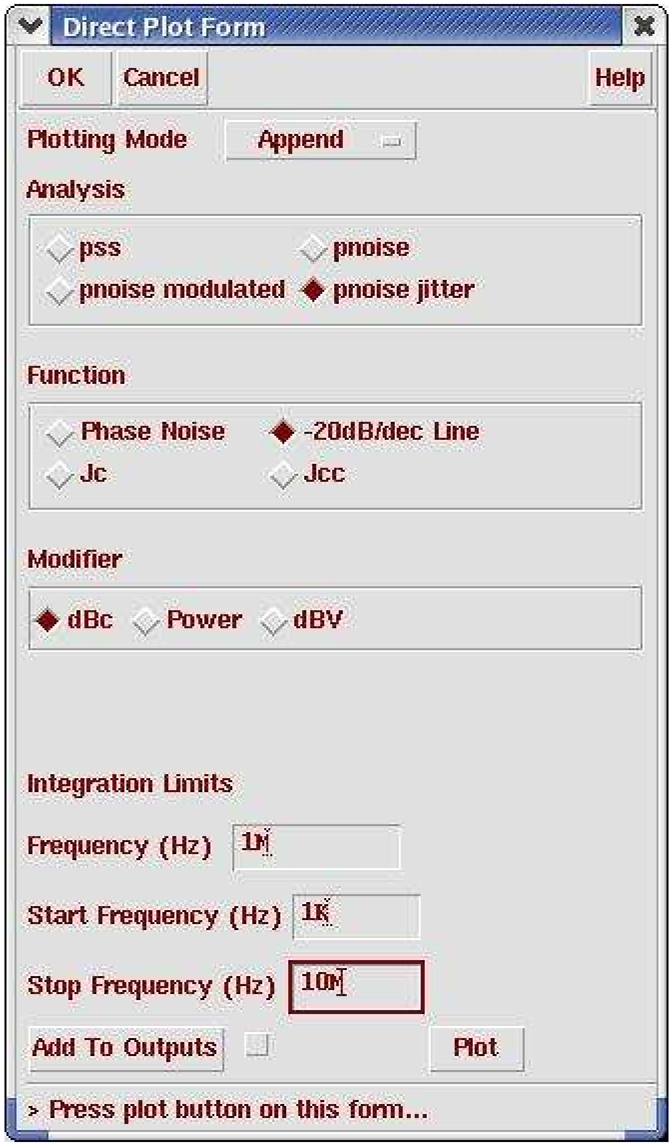
Action1-25: In the waveform window, select window 2.

Action1-26: 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.

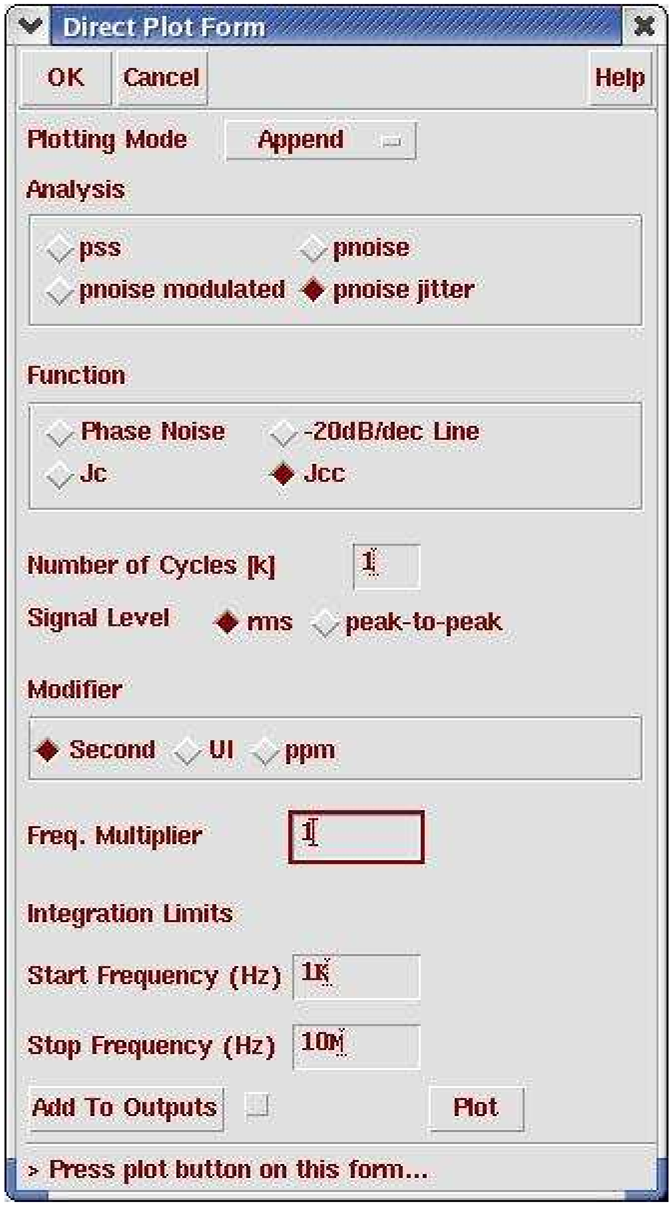



Action1-27: In the Direct Plot form, select **pnoise jitter**. Select **-20dB/dec Line** and configure the form as follows:

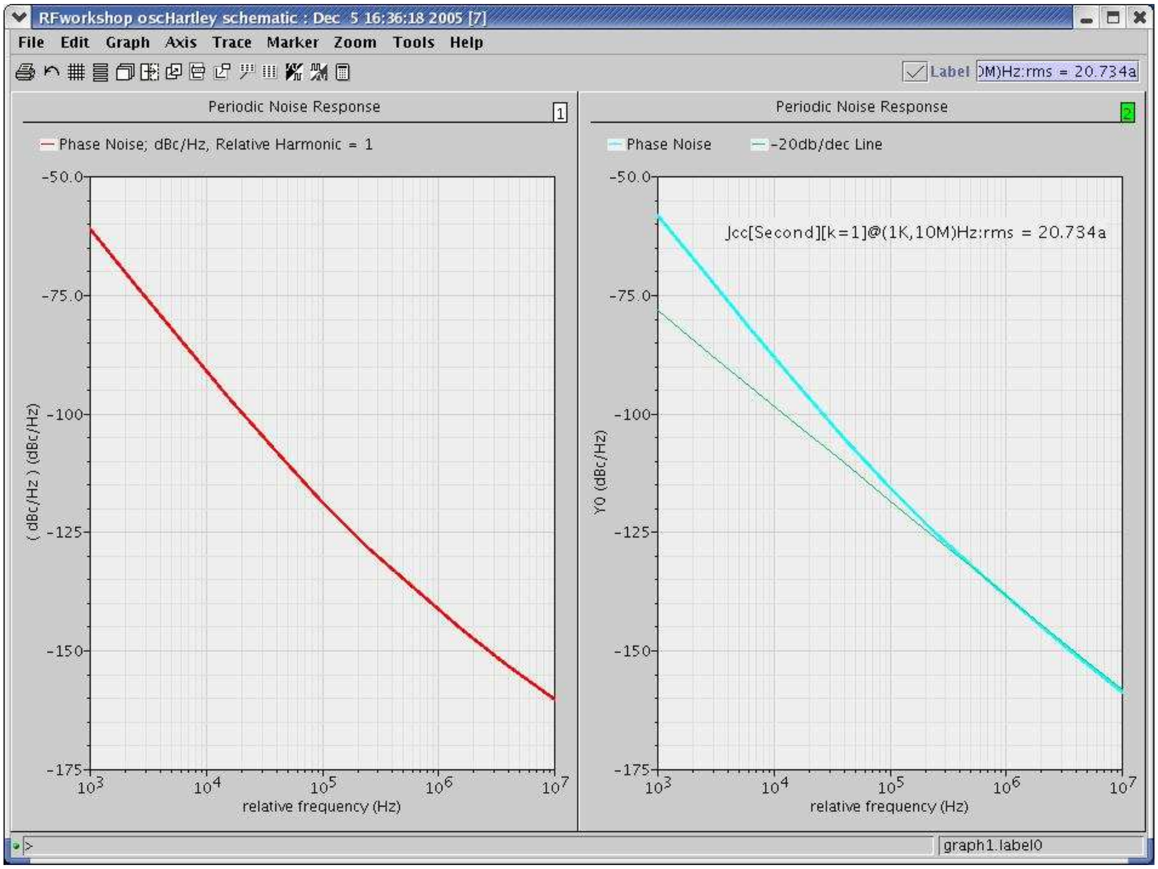



The "-20dB/dec" curve could be placed on the phase noise curve at any frequency. The slope of -20dB/dec "assistant" let us to distinguish the regions of the 1/f^3, 1/f^2 and 1/f phase noise PSD. If you prefer, the quick manual calculations using simple white noise jitter approximation could be used after 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 the user to select the point for the approximation of the slope. In the later releases of Direct Plot, the integration is numerical and the selection of the point is not needed anymore. Therefore, the slope assistant is for informative purposes only now.

Action1-28: 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 choose analysis form.




The waveform window updates with the jitter calculation.




MMSIM60USR2 introduces Flexible Balance for autonomous circuit application. In this release FlexBalance is supported for oscillators and phase-noise simulation. Flexbalance 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.

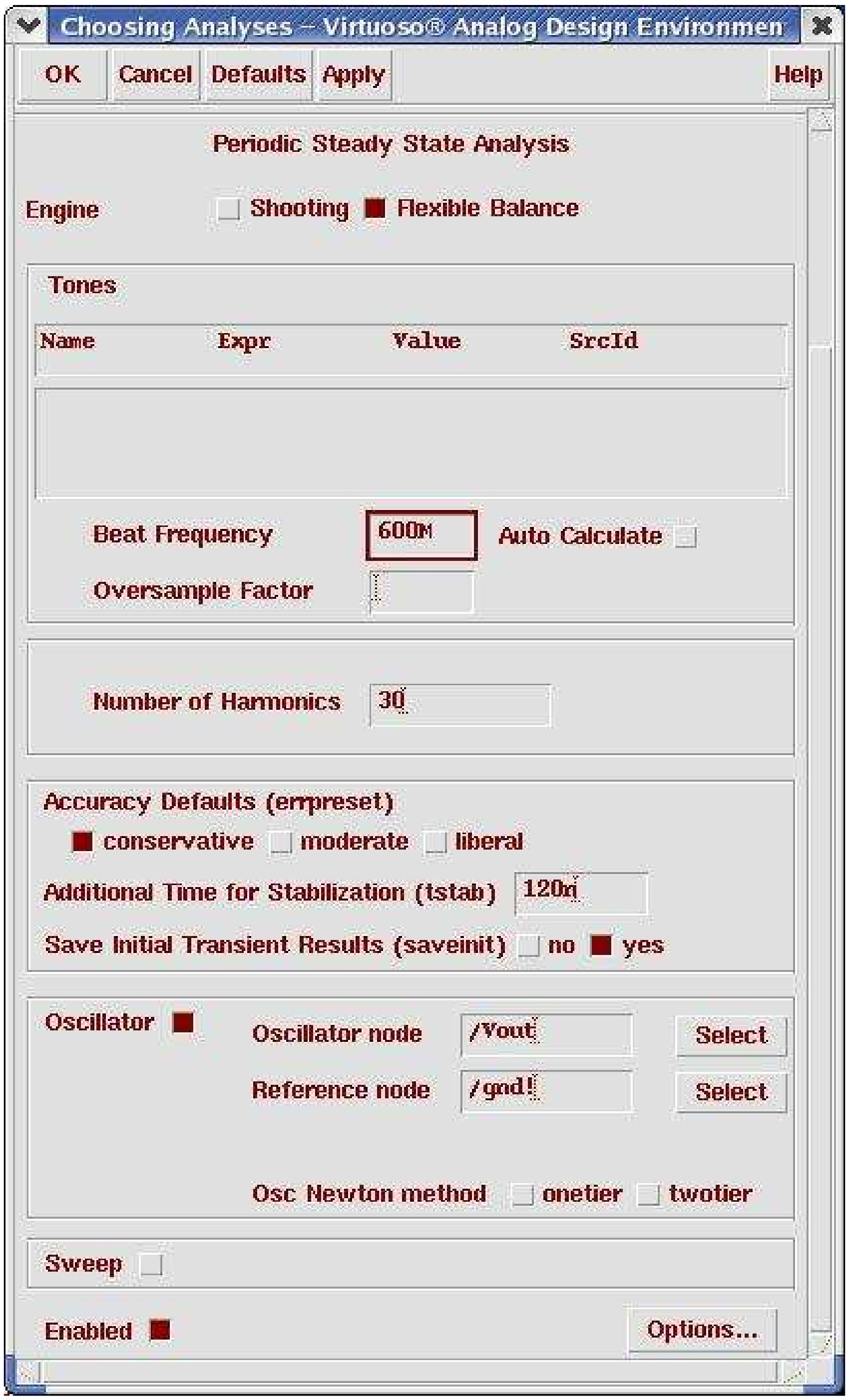
Action1-29: You can choose **Session—Load State** in Virtuoso Analog Design Environment load state “**Lab1_Pnoise_FB**”, then skip to Action1-35 or …

Action1-30: In Analog Design Environment window, select **Analyses->Choose…**

Action1-31: In the Choosing Analyses window, select the **pss** button in the **Analysis** field of the window.

Action1-32: In the PSS analysis form, choose **Flexible Balance** engine. Set **Fundamental Frequency** of **600 MHz**. Set the **Number of Harmonics** to **30** for this oscillator. Set the tstab to add additional time for the oscillator to converge on its operating frequency. Click on yes to save the initial transient results. The **Oscillator** button should be enabled. Keep other old set up.

The pss analysis form looks like this:



Currently, there are two Flexbalance methods implemented: onetier or twotier. They are set by parameter "oscmethod". In onetier method, the frequency and voltage spectrum are solved simultaneously in one single set of nonlinear equations. In twotier method, the nonlinear equations are split into two sets: the inner set of nonlinear equations solves the spectra of node voltage equation; 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 will automatically choose the pinning node. By default, SpectreRF runs onetier first for n iterations; if necessary, it runs twotier next n iterations (n is set by "maxperiods"). Users can also run only onetier or twotier by specifying "oscmethod". Twotier has larger convergence zone as its convergence is slightly more robust. This method is however slower than onetier method.

SpectreRF Flexbalance is transient-assisted. SpectreRF runs transient analysis (length is specified by "tstab") and then switch to FlexBalance. Stable oscillation in transient analysis can help FlexBalance to converge. It can be achieved by:

1.) Setting an initial condition for a particular node (for example: ic net01=5.0)

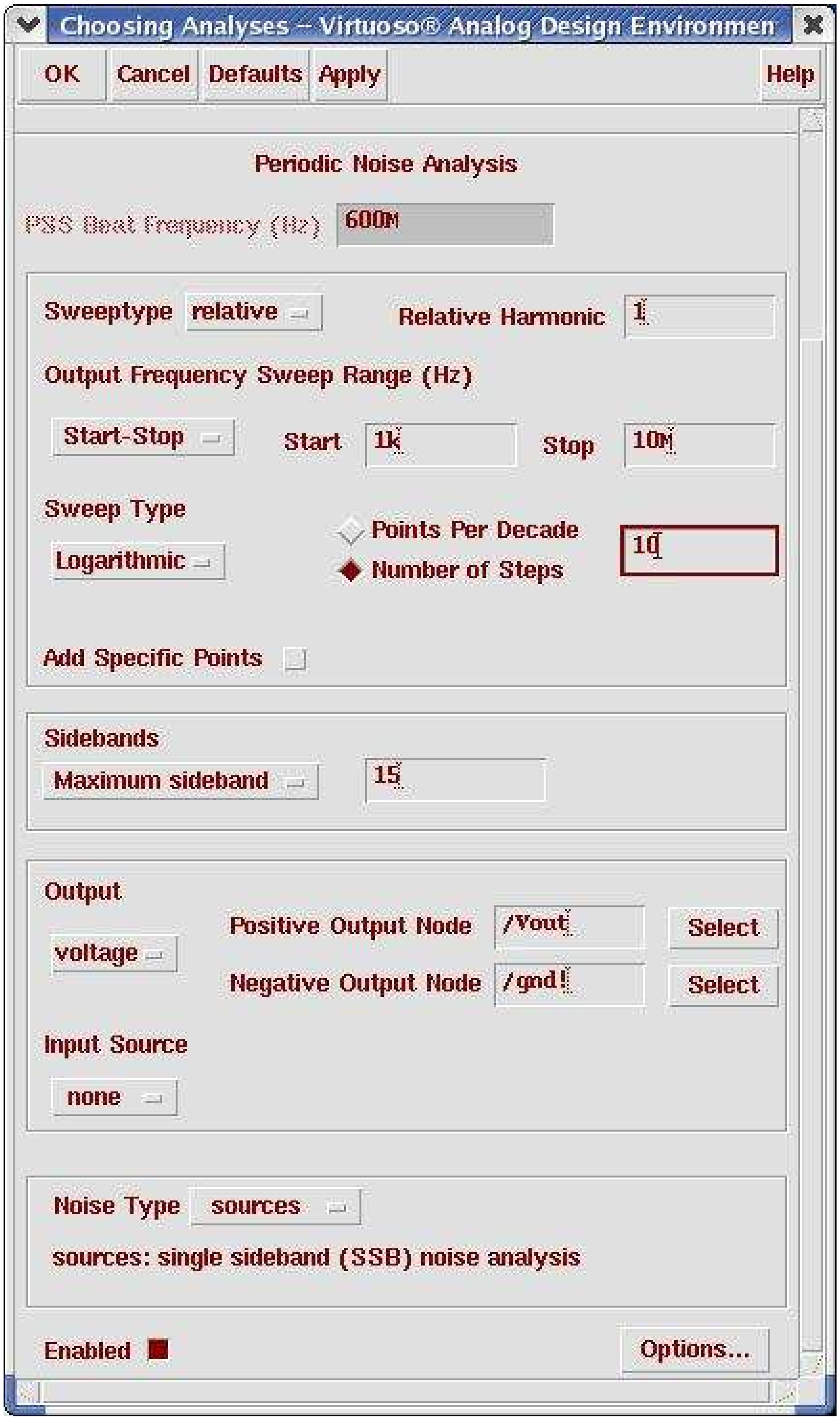
2.) Setting an initial condition for the inductor/capacitor in the resonator in OSC (for example: L14 (Xtal02 net01) inductor l=6.m ic=0.5m)

3.) 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

4.) Adding a voltage pulse (for example: vdd (vdd 0) vsource type=pulse val0=0.0 val1=3.3 rise=1n)

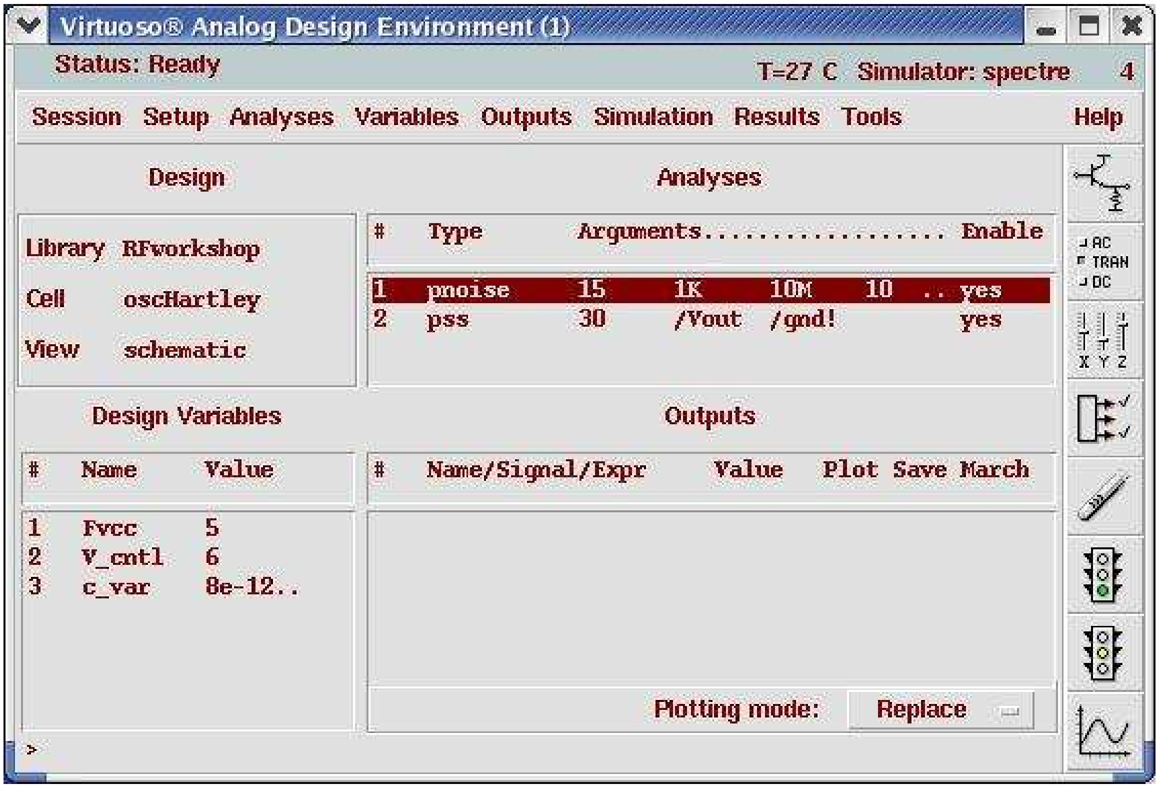
5.) Specifying "oscIC=defaultIC" or "oscIC=linearIC".

Action1-33: Click the **pnoise** button. In the Periodic Noise Analyses form, set noise type to sources. Current SpectreRF doesn't support jitter analysis with Flexible Balance engine. With Flexbalance engine, the Number of harmonics is input parameter. Even we set the Maximum sideban=15 here, 30 sidebands are to be calculated. The form should look like this:



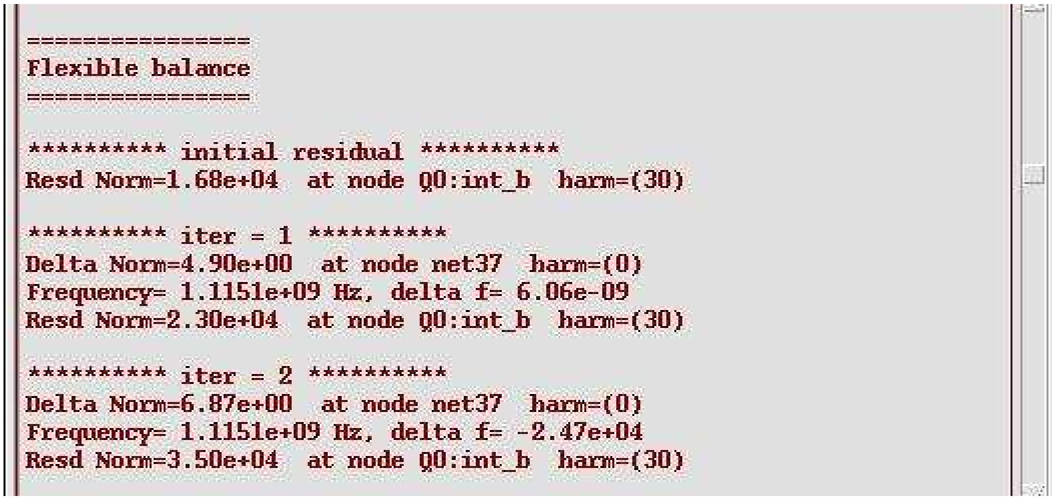
Action1-34: Make sure that the Enabled button is on. Click OK in the Choosing Analyses form.

Now your Virtuoso Analog Design Environment looks like this:



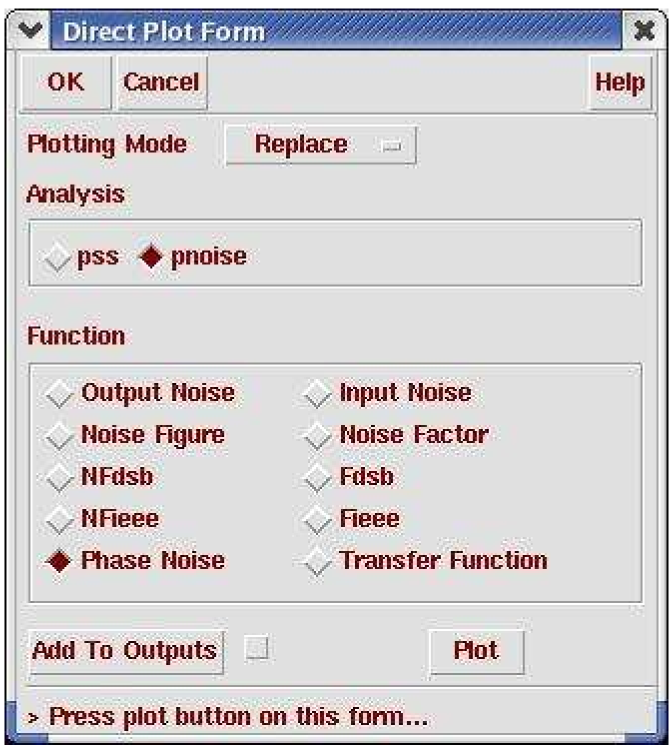

Action1-35: In your Analog Design Environment, Choose **Simulation—Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

As the simulation progresses, note messages in the simulation output log window that are different from time domain pss:



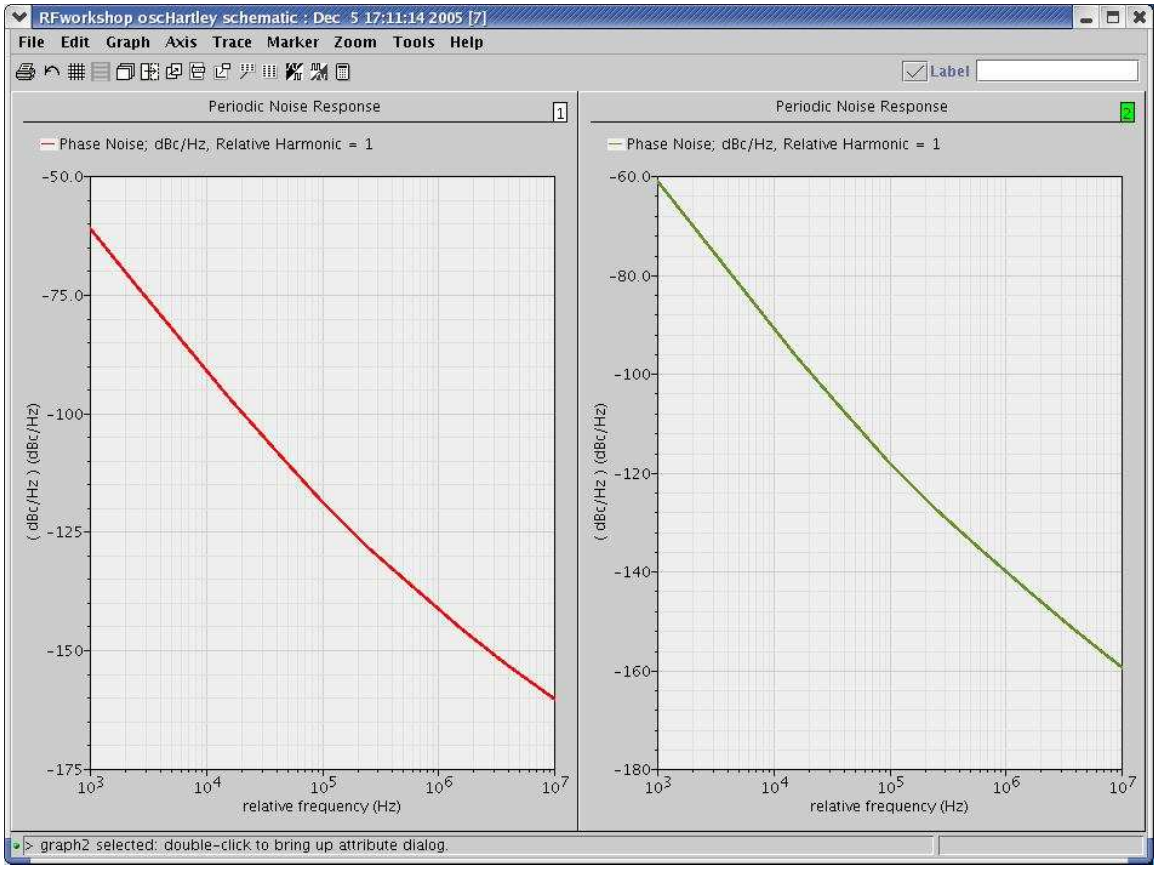

Action1-36: In the Virtuoso Analog Design Environment window, choose **Results—Direct Plot—Main Form** to plot the phase noise.

Action1-37: In Direct Plot Form window, choose **pss** as the Analysis type and configure the form as follows:




Action1-38: In the Direct Plot form, click the **Plot** button.

Now you should get the following waveforms:




FlexBalance method is a good candidate to simulate mildly-nonlinear oscillators with resonators, such as LC oscillators, negative-gain oscillators, and crystal oscillators. Shooting PSS method is a good candidate to simulate strongly-nonlinear non-resonator oscillators, such as ring oscillators, relaxation oscillators, or oscillators containing digital control components.

Action1-39: Close the waveform window, the Direct Plot form and Virtuoso Analog Design Environment window.

## Lab2: 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.

Action2-1: If not already open, open the *schematic* view of the design *oscHartley* in the library *RFworkshop.*

Action2-2: From the oscHartley schematic, start the Virtuoso Analog Design Environment with the **Tools—Analog** Environment command.

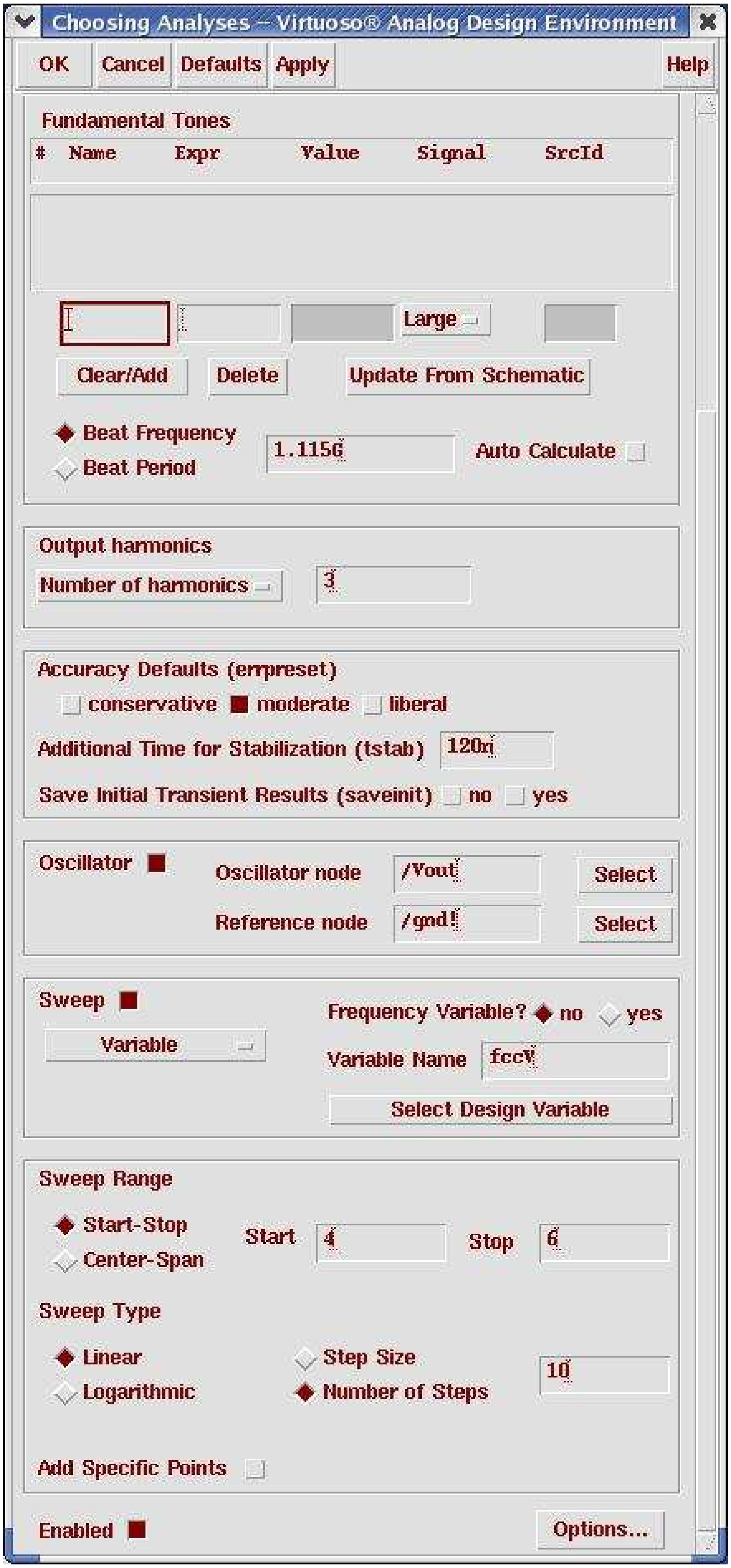
Action2-3: You can choose **Session—Load State**, load state "**Lab2_Frequency_Pushing_PSS**" and skip to Action2-8 or …

Action2-4: In Virtuoso Analog Design Environment, choose **Analyses—Choose…**

Action2-5: In the Choosing Analyses window, select the **pss** button in the **Analysis** field of the window.

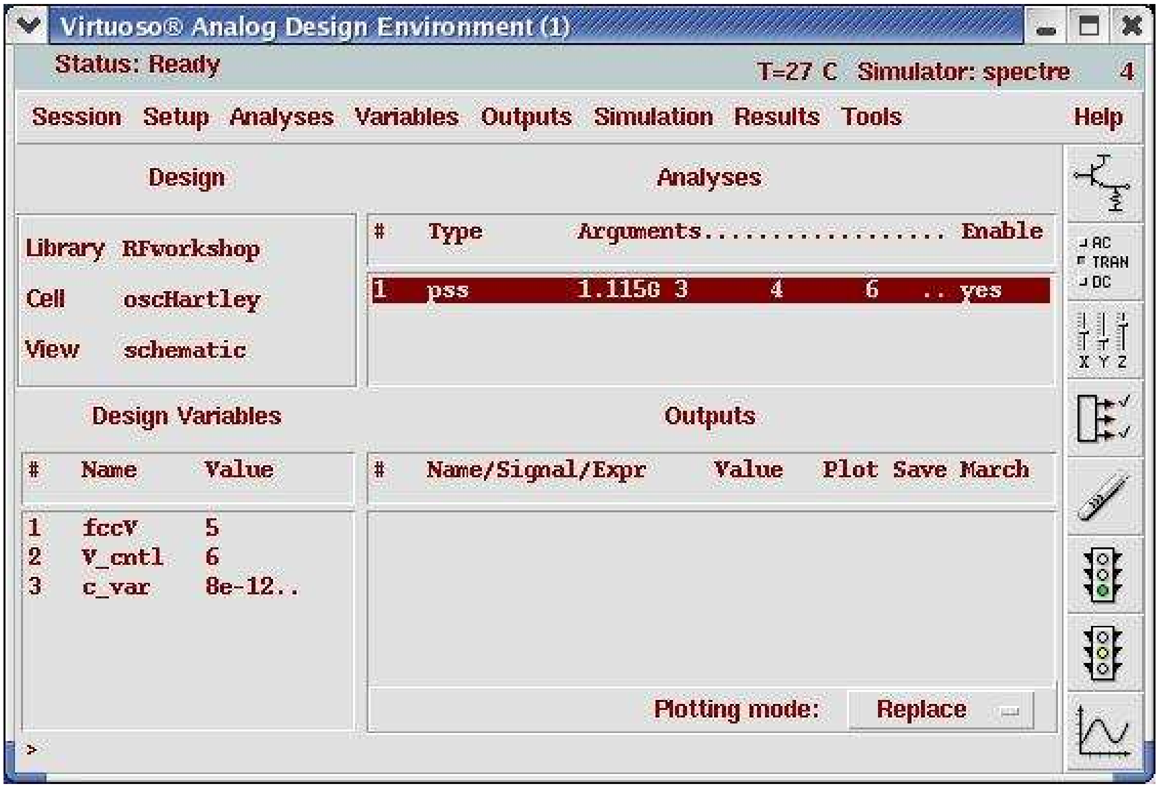
Action2-6: Set up a swept PSS analysis with a Beat Frequency = 1.115 G; Number of Harmonics = 3; errpreset = moderate; tstab = 120 n; enable the oscillator button; set Oscillator node = /Vout; and Reference node = /gnd!; enable the Sweep button; enter fvcc as Variable Name; set the Sweep Range Start = 4 and Stop = 6; set Sweep Type = linear; and Number of Steps = 10.

Your PSS analysis window should look like…


Choosing Analyses -- Virtuoso® Analog Design Environment
OK
Cancel
Defaults
Apply
Help
Fundamental Tones
# Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Schematic
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
Sweep
Variable
Frequency Variable?
no
yes
Variable Name
fccV
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
Enabled
Options...

Action2-7: Make sure the **Enabled** button is active, and click **OK** in the Choosing Analyses form.

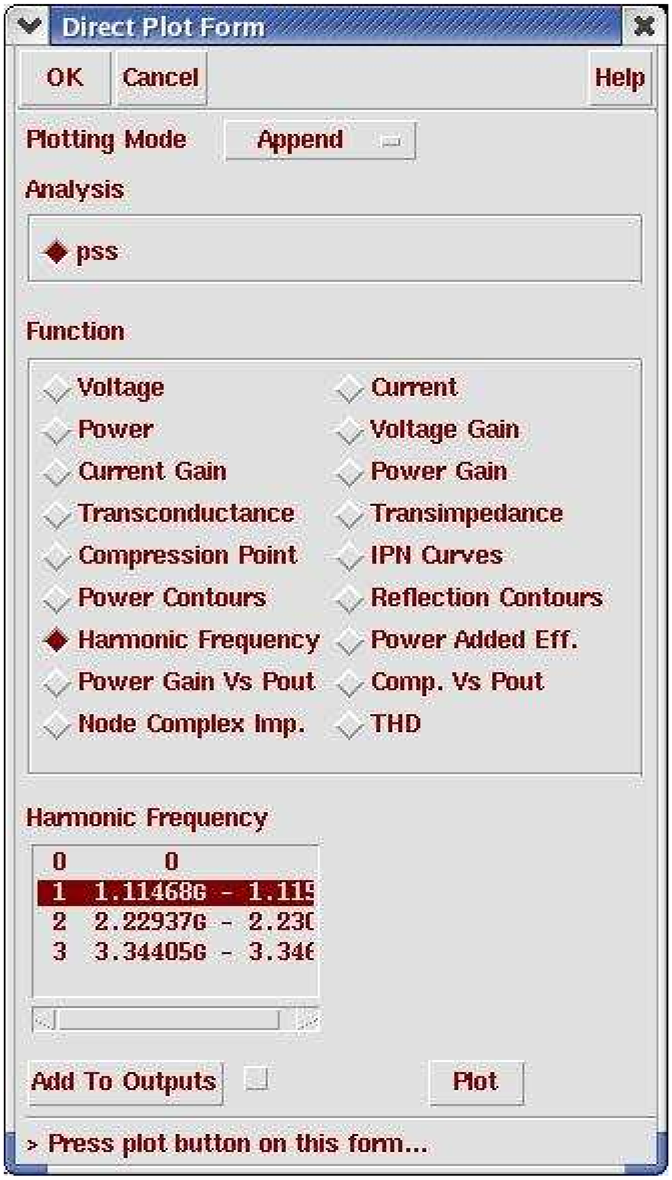
Your Virtuoso Analog Environment will look like this:



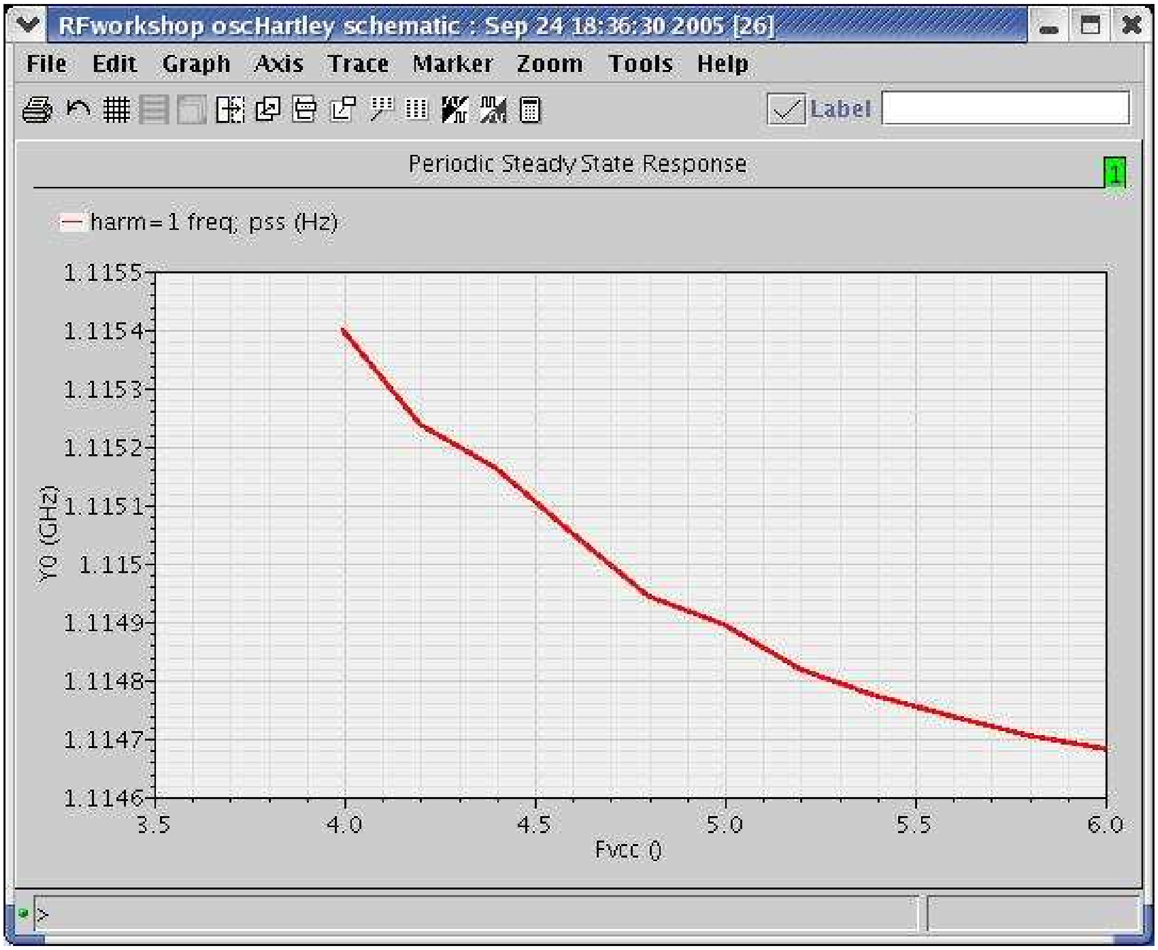

Action2-8: In your Analog Design Environment, Choose **Simulation—Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action2-9: In the Virtuoso Analog Design Environment, Choose **Results—Direct Plot—Main Form**.

Action2-10: In the Direct Plot Form, select the **pss** button, click on the Harmonic Frequency button, and highlight the 1st harmonic in the Harmonic Frequency section. The form should look like this:




Action2-11: Click the **Plot** button. The following plot will show up.




Action2-12: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

## Lab3: 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.

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.

Action3-1: If not already open, In, open the *schematic* view of the design *oscHartley* in the library *RFworkshop*

Action3-2: From the oscHartley schematic, start the Virtuoso Analog Design Environment with the **Tools—Analog** Environment command.

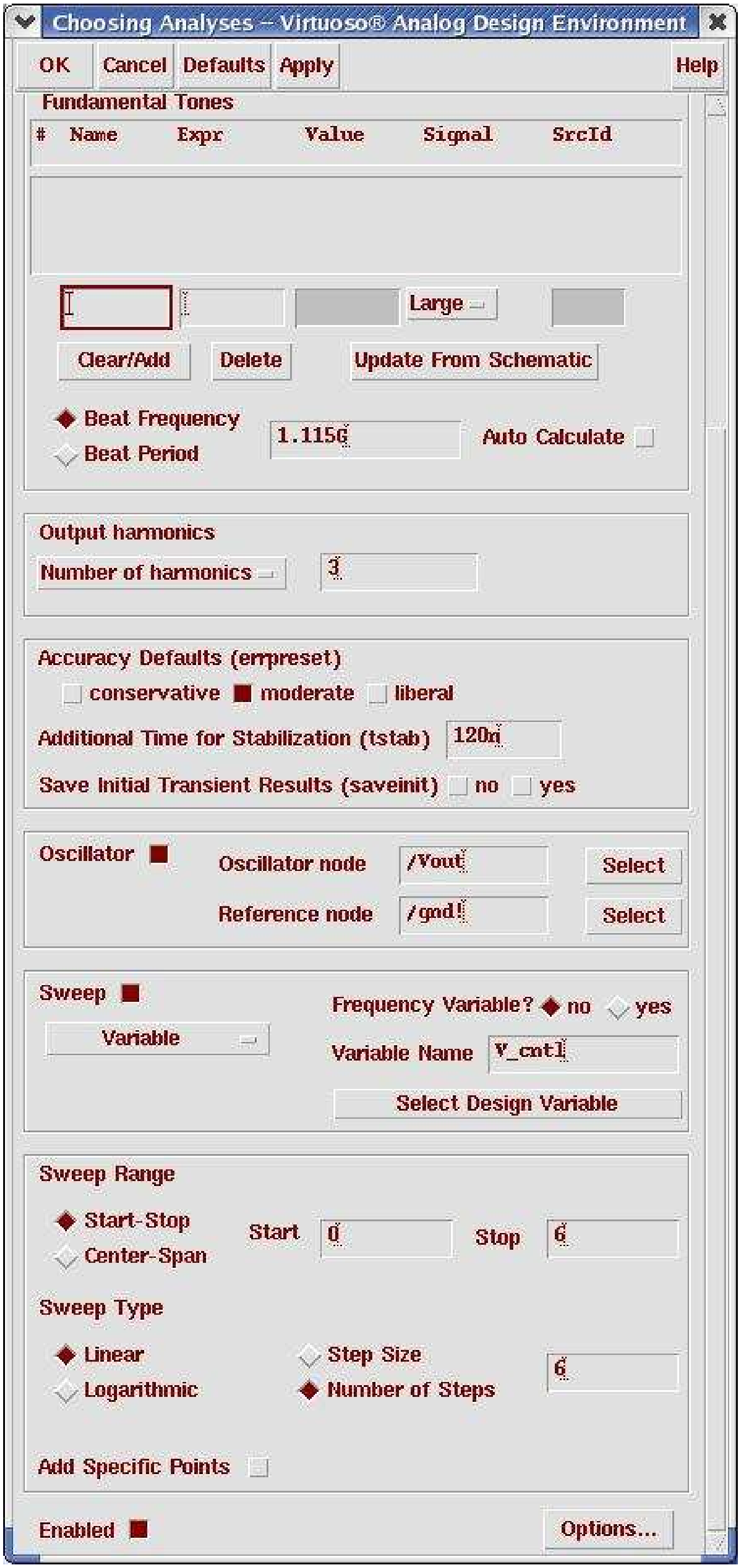
Action3-3: You can choose **Session—Load State**, load state "**Lab3_Sensitivity_Linearity_PSS**" and skip to Action3-8 or …

Action3-4: In Virtuoso Analog Design Environment, choose **Analyses—Choose…**

Action3-5: In the Choosing Analyses window, select the **pss** button in the **Analysis** field of the window.

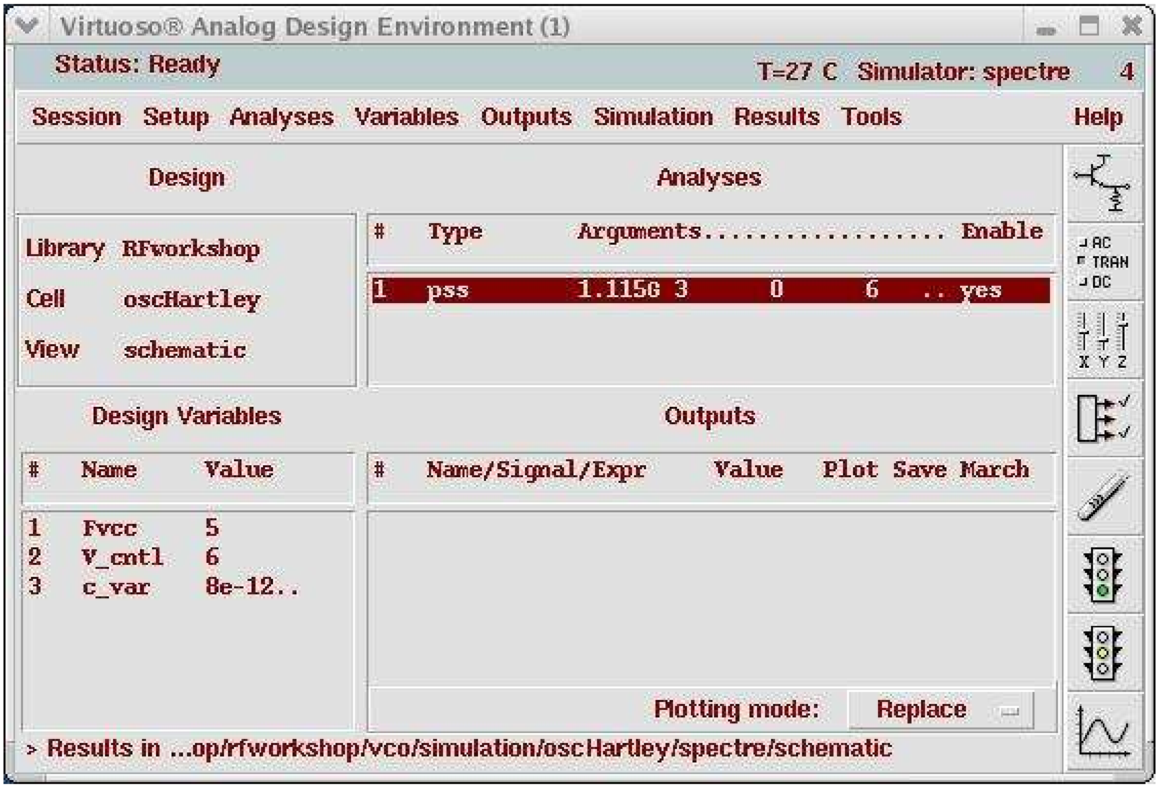
Action3-6: Set up a swept PSS analysis with a Beat Frequency = 1.115G; Number of Harmonics = 3, errpreset = moderate, tstab = 120n. Enable the oscillator button, set Oscillator node = /Vout, and Reference node = /gnd!. Enable the Sweep button, enter V_cntl as Variable Name (this is the tuning voltage), set the Sweep Range Start = 0 and Stop = 6, set Sweep Type = linear, and Number of Steps = 6.

Your PSS analysis window should look like…


Choosing Analyses -- Virtuoso® Analog Design Environment
OK
Cancel
Defaults
Apply
Help
Fundamental Tones
# Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Schematic
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
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
Enabled
Options...

Action3-7: Make sure the **Enabled** button is active, and click **OK** in the Choosing Analyses form.

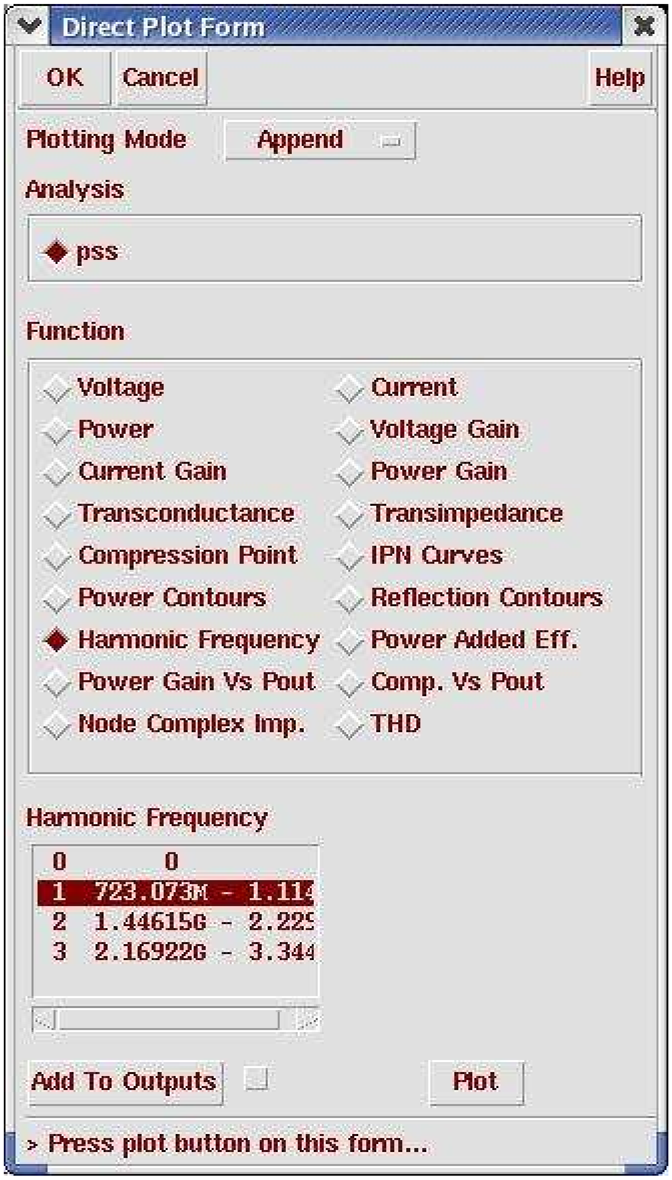
Your Virtuoso Analog Environment will look like this:



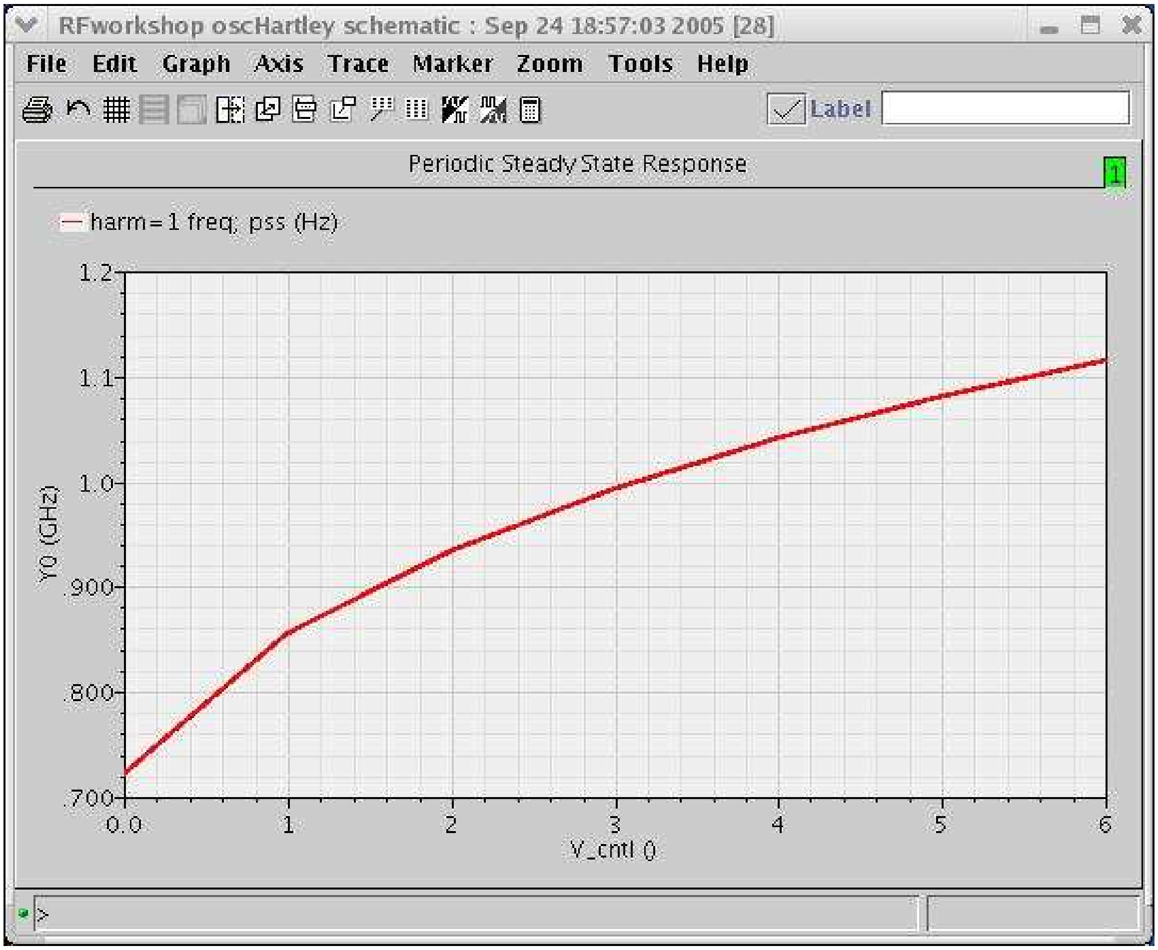

Action3-8: In your Analog Design Environment, Choose **Simulation—Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action3-9: In the Virtuoso Analog Design Environment, Choose **Results—Direct Plot—Main Form**.

Action3-10: In the Direct Plot Form, select the **pss** button, click on the Harmonic Frequency button, and highlight the 1-st harmonic in the Harmonic Frequency section. The form should look like this:



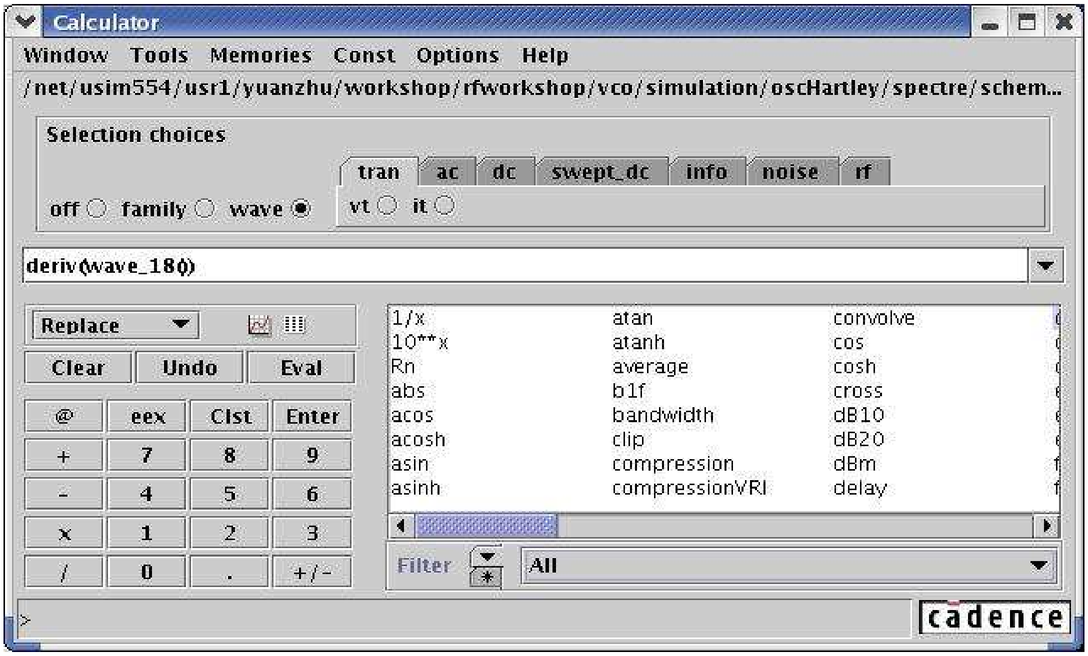

Action3-11: Click the **Plot** button. The following plot will show up.



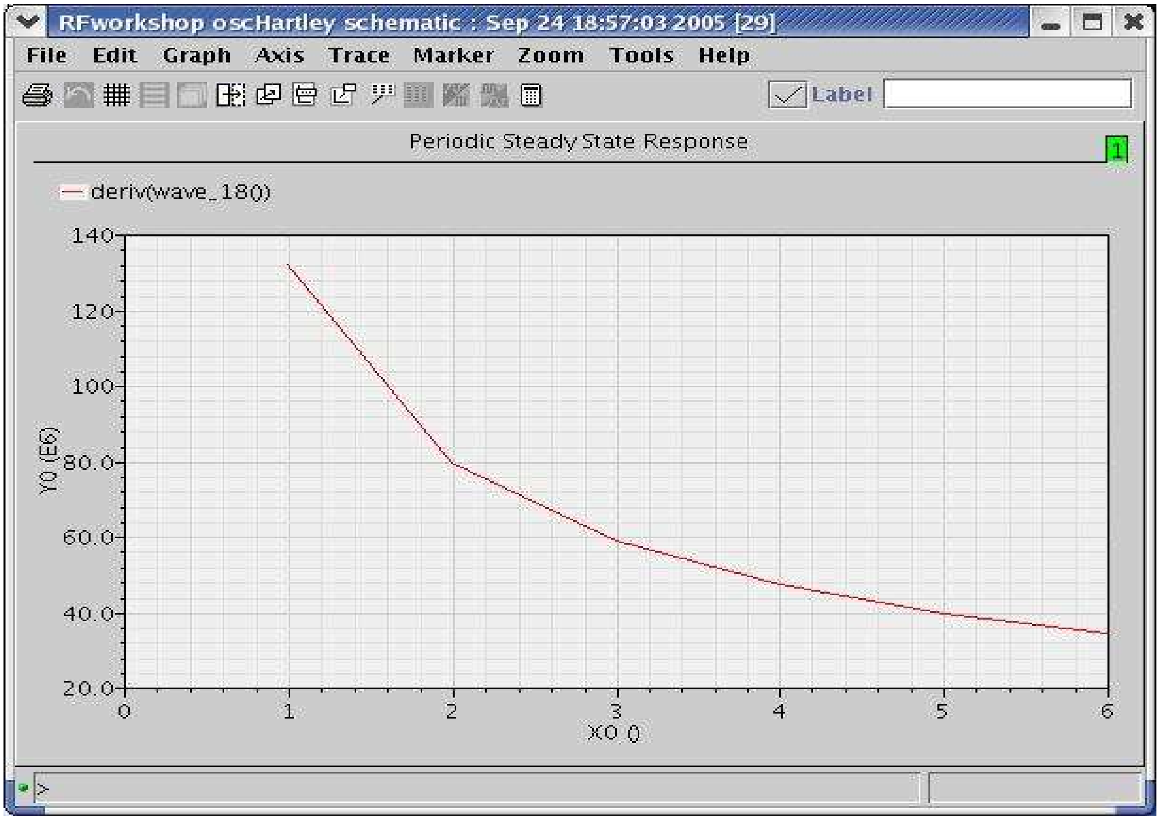

Action3-12: In the Virtuoso Analog Design Environment window, Choose **Tools—Calculator.**

Action3-13: In the Calculator window, highlight **wave**, and select the sensitivity curve in the waveform window. Choose deriv in the special function field. Change the Plot Mode to Replace.

The calculator window should look like:




Action3-14: Click on the button on the calculator window. The following plot represents the frequency change per unit volt of tuning voltage.




Action3-15: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

## Lab4: 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.

Action4-1: If not already open, open the *schematic* view of the design *oscHartley* in the library *RFworkshop*

Action4-2: From the oscHartley schematic, start the Virtuoso Analog Design Environment with the **Tools—Analog** Environment command.

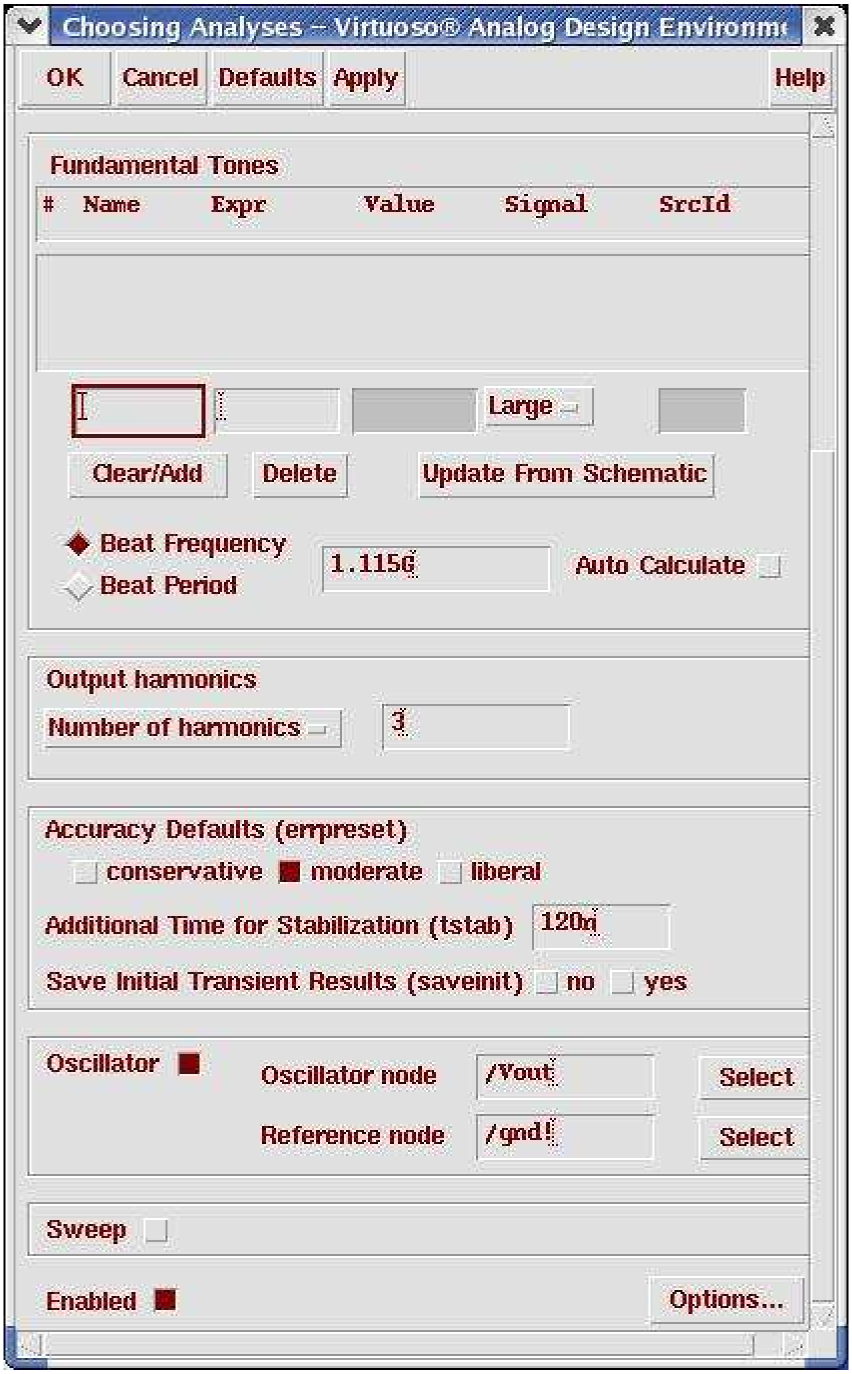
Action4-3: You can choose **Session—Load State**, load state "**Lab4_Power_Dissipation_PSS**" and skip to Action4-11 or …

Action4-4: In Virtuoso Analog Design Environment, choose **Analyses—Choose…**

Action4-5: In the Choosing Analyses window, select the **pss** button in the **Analysis** field of the window.

Action4-6: Set up a PSS analysis with a Beat Frequency = 1.115G; Number of Harmonics = 3; errpreset = moderate; tstab = 120n; enable the oscillator button; set Oscillator node = /Vout; and Reference node = /gnd!.

Your PSS analysis window should look like…




Action4-7: Make sure the **Enabled** button is active, and click **OK** in the Choosing Analyses form.

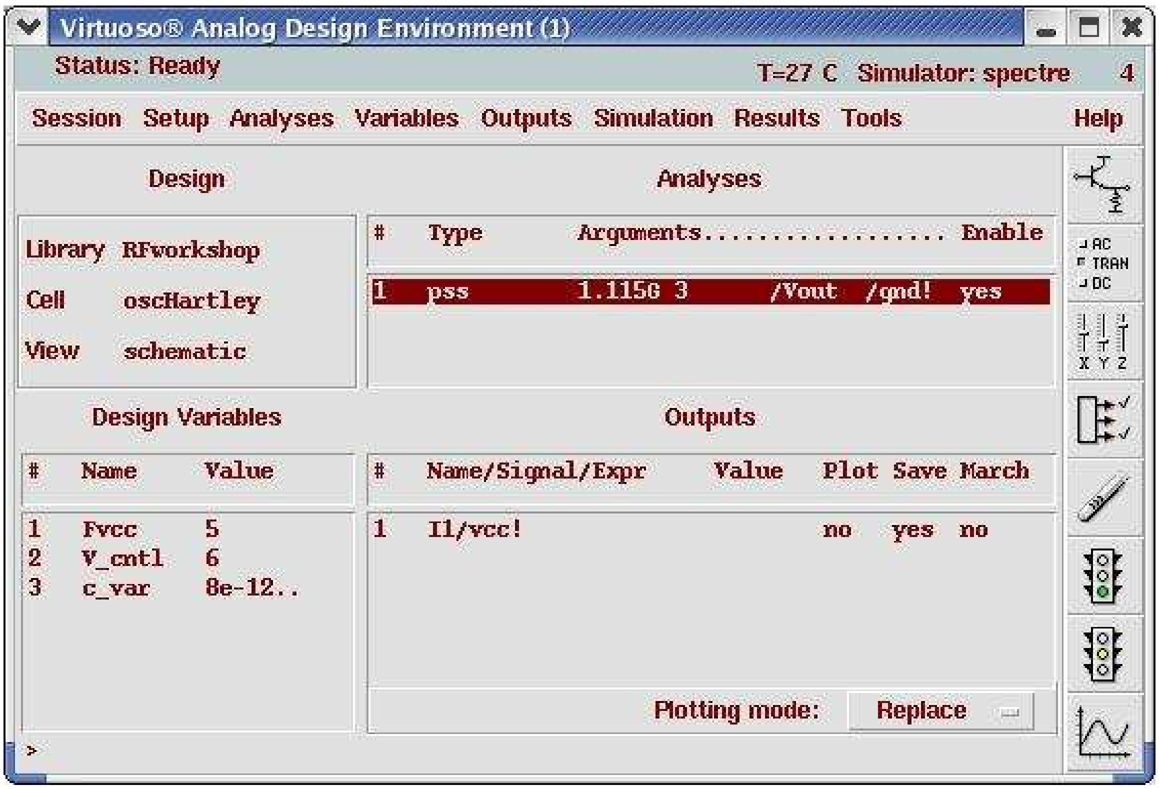
To obtain the Power Dissipation, before you run the PSS analysis, you must save data at the Vcc terminal through the analog design environment.

Action4-8: In the Virtuoso Analog Design Environment window, choose **Outputs—To Be Saved—Select On Schematic**.

Action4-9: 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.

Action4-10: Press **Esc** with your cursor in the *oscHartley* schematic window to end the selections.

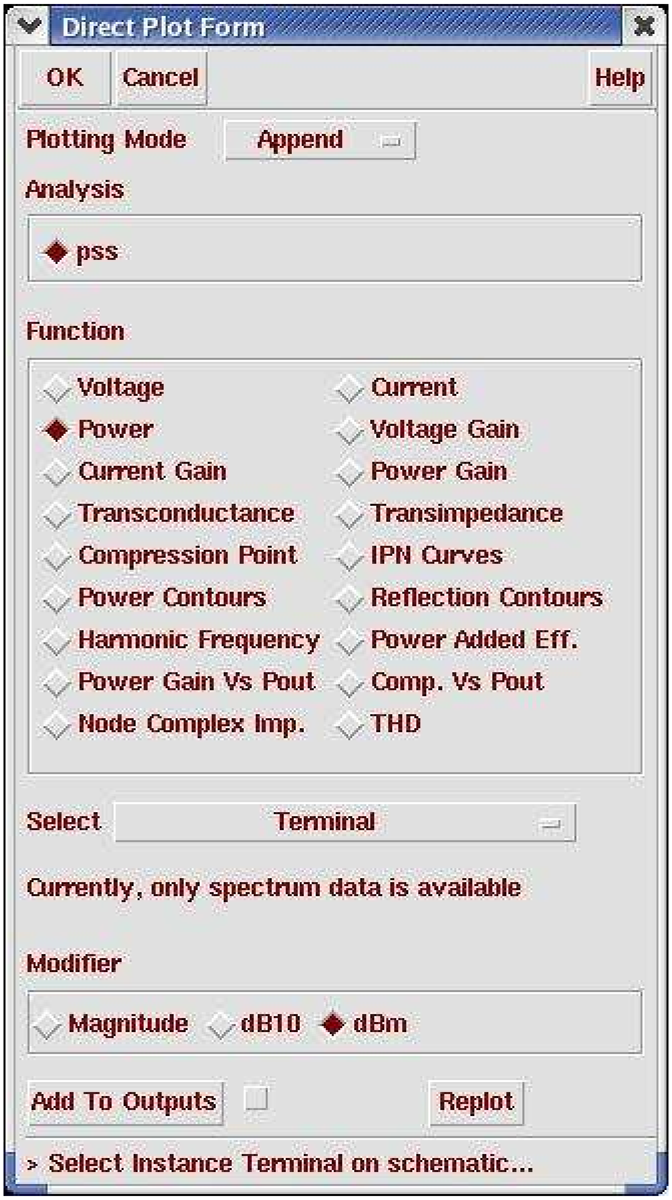
Your Virtuoso Analog Environment will look like this:



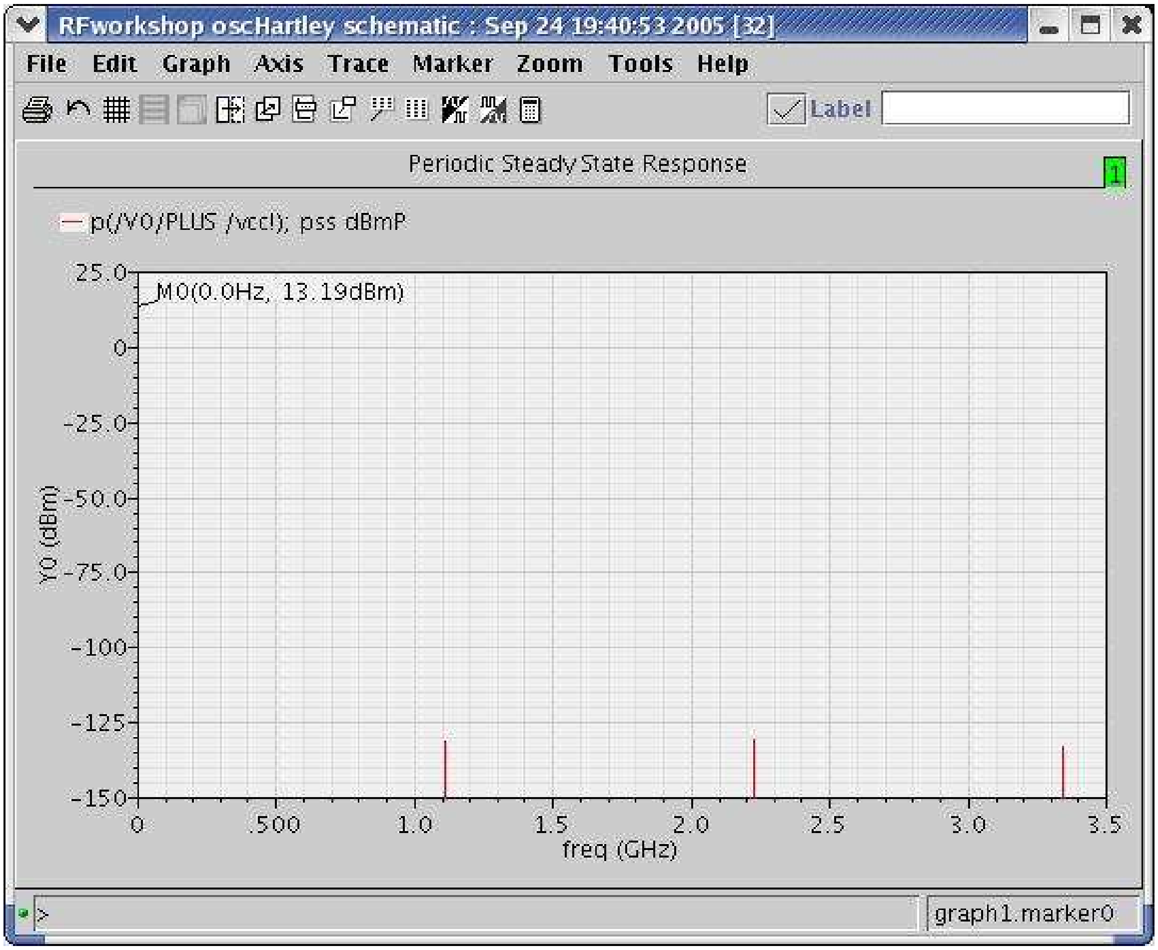

Action4-11: In your Analog Design Environment, Choose **Simulation—Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action4-12: In the Virtuoso Analog Design Environment, Choose **Results—Direct Plot—Main Form**.

Action4-13: In the Direct Plot Form, select the **pss** button, click on the **power** button, and choose **dBm** as **Modifier**. The form should look like this:




Action4-14: Click the positive terminal of pulse source on the schematic. . The following plot will show up.




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

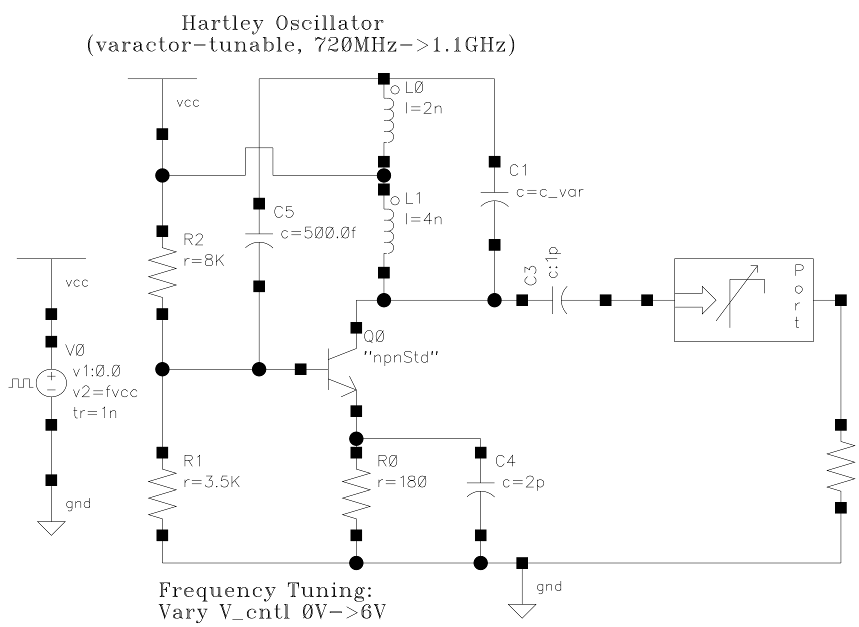
Action4-15: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window. Close the *oscHartley* schematic.

## Lab5: 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.

Action5-1: In the Library Manager window, open the *schematic* view of the design *freqpull* in the library *RFworkshop*

The following figure shows the modified Hartley Oscillation 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:

$$rl = -20Log\,|\,\Gamma\,|$$

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

Action5-2: From the *freqpull* schematic, start the Virtuoso Analog Design Environment with the **Tools—Analog** Environment command.

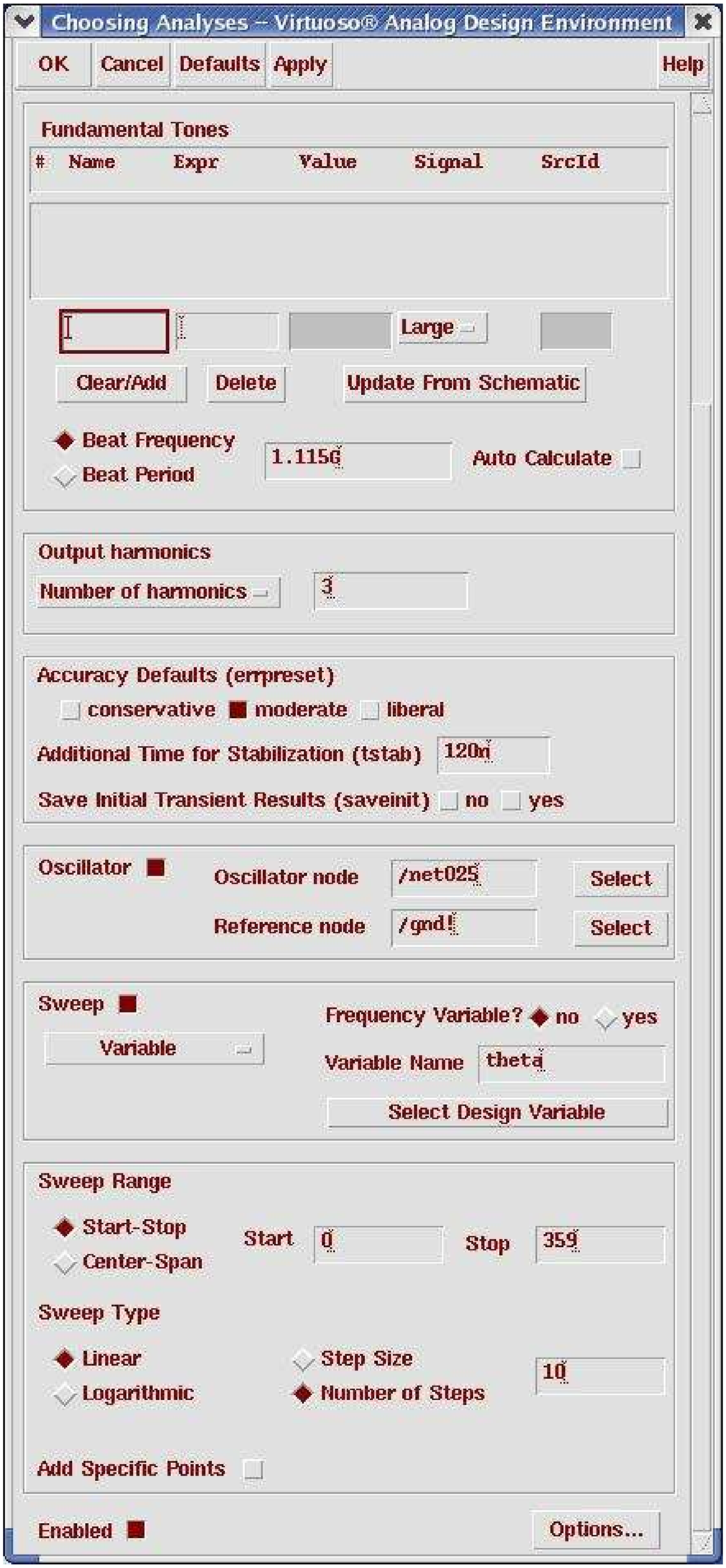
Action5-3: You can choose **Session—Load State**, load state "**Lab5_Frequency_Pulling_PSS**" and skip to Action5-8 or …

Action5-4: In Virtuoso Analog Design Environment, choose **Analyses—Choose…**

Action5-5: In the Choosing Analyses window, select the **pss** button in the **Analysis** field of the window.

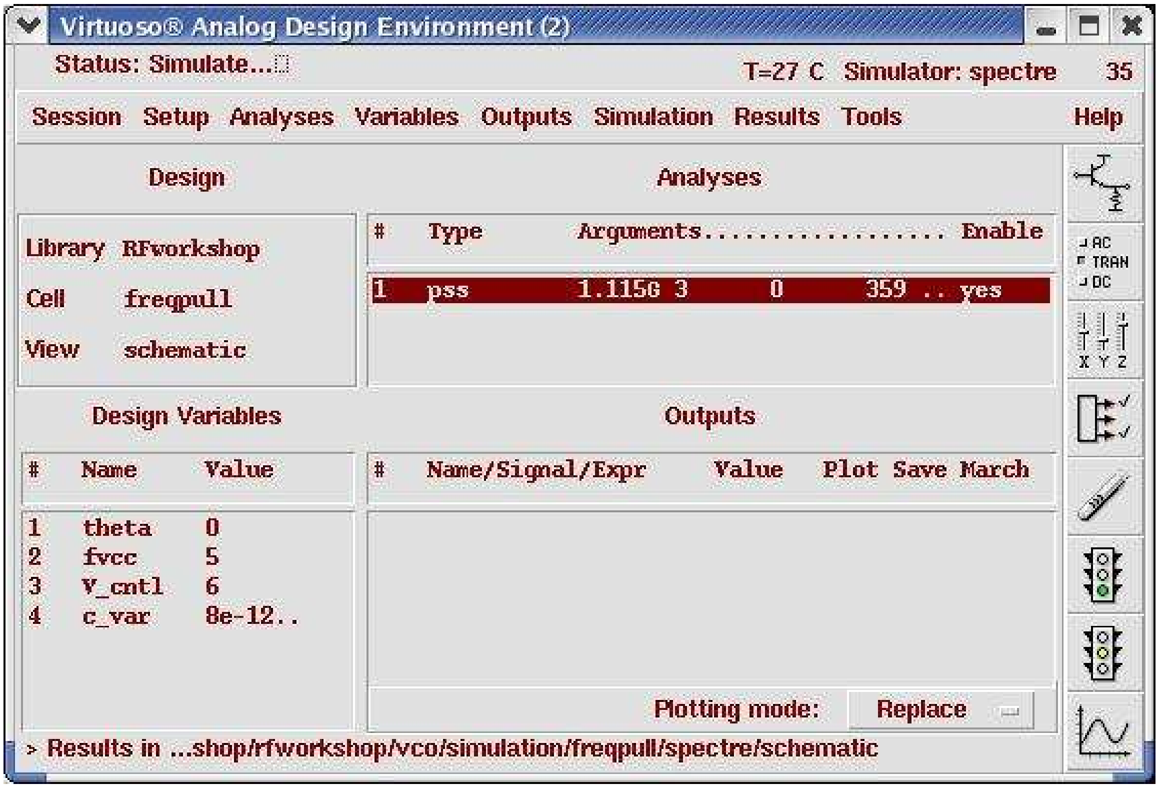
Action5-6: Set up a swept PSS analysis with the theta parameter varying from 0 to 359 degrees. Set Beat Frequency = 1.115G; Number of Harmonics = 3; errpreset = moderate; tstab = 120n; enable the oscillator button; set Oscillator node = /Vout; and Reference node = /gnd!; enable the Sweep button; enter theta as Variable Name; set the Sweep Range Start = 0 and Stop = 359; set Sweep Type = linear; and Number of Steps = 10.

Your PSS analysis window should look like…


Choosing Analyses -- Virtuoso® Analog Design Environment
OK
Cancel
Defaults
Apply
Help
Fundamental Tones
# Name Expr Value Signal SrcId
Large
Clear/Add
Delete
Update From Schematic
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
Sweep
Frequency Variable?
no
yes
Variable
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
Enabled
Options...

Action5-7: Make sure the **Enabled** button is active, and click **OK** in the Choosing Analyses form.

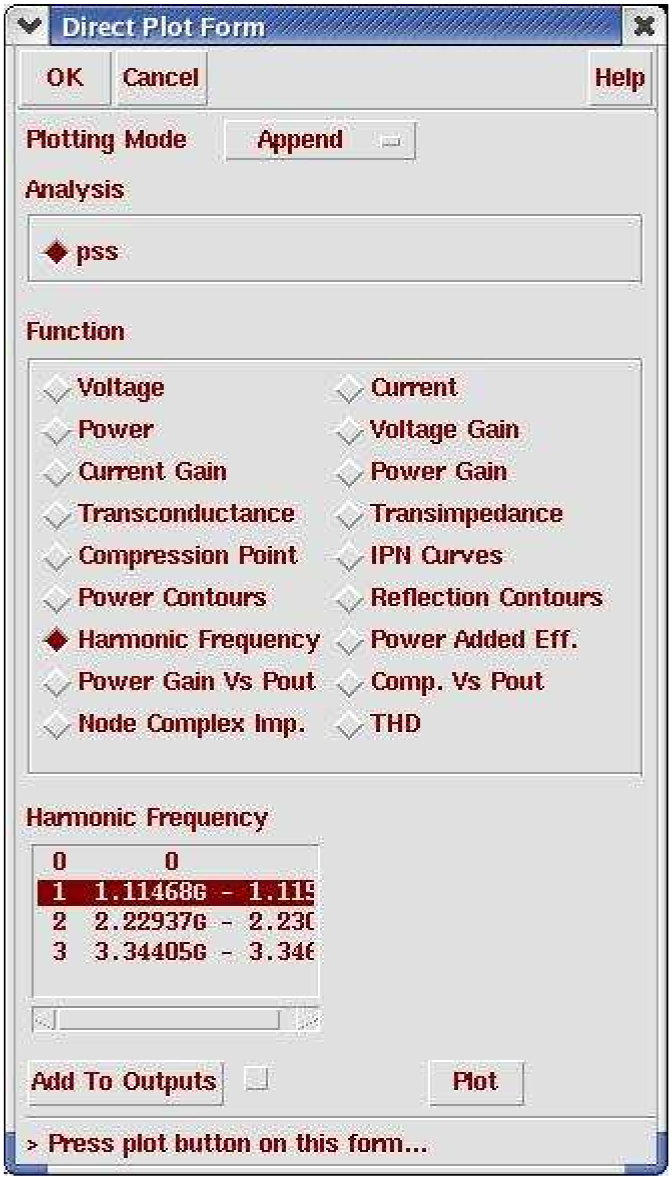
Your Virtuoso Analog Environment will look like this:



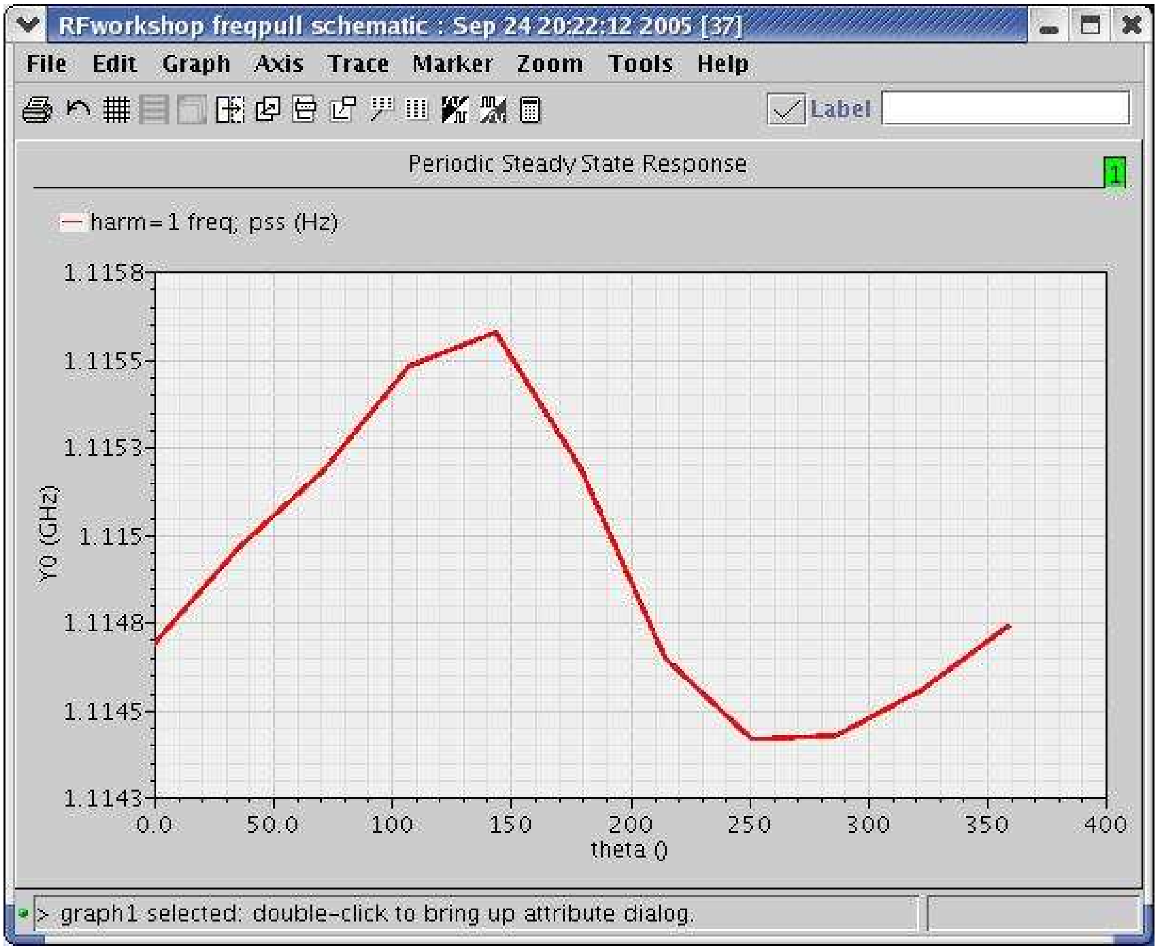

Action5-8: In your Analog Design Environment, Choose **Simulation—Netlist and Run** or click the **Netlist and Run** icon to start the simulation.

Action5-9: In the Virtuoso Analog Design Environment, Choose **Results—Direct Plot—Main Form**.

Action5-10: In the Direct Plot Form, select the **pss** button, click on the Harmonic Frequency button, and highlight the 1-st harmonic in the Harmonic Frequency section. The form should look like this:




Action5-11: Click the **Plot** button. The following plot will show up.




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

Action5-12: Close the waveform window. Click **Cancel** on the Direct Plot form. Close the Virtuoso Analog Design Environment window.

# Conclusion

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. FlexBalance as new engine is introduced and is used by some of those measurements.

# Reference

[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.