ASITIC & Eldo Tutorial for EE6240 – RF Integrated Circuits

A. Lab, servers etc

You may use the FPGA lab to run your ASITIC and Eldo simulations for this course. Both simulation softwares are installed on all machines in this lab. The machine ip addresses are 10.7.8.41-80, and you can run simulations remotely on these through an "ssh -X" command.

You may also install and run Eldo on your personal computers. You can find more information about this here. Similarly, you can download and install the latest "Grackle" release of ASITIC on your personal computers.

B. Setting up your terminal session to run Eldo & ASITIC

Your default login to the FPGA lab machines uses the Z shell (zsh). Add the following lines to the .zshrc file in your home directory:

```
export LM_LICENSE_FILE=1717@10.7.9.34:1717@10.7.9.35:1717@10.7.9.36
export MGC_AMS_HOME=/tools/Mentor/AMS_2008_2_IXL
export PATH=$PATH:$MGC_AMS_HOME/bin:/tools/asitic
```

If you don't have a .zshrc file, create one in your home directory and add these lines to it. These are sourced everytime you login, but for the first time alone you will have to source it manually with a "source .zshrc" command (unless you want to log out and log back in). You are now ready to run Eldo.

C. Sample ASITIC sessions

Sample ASITIC sessions are available on the ASITIC website here. It is a good idea to go through this before starting your inductor design.

D. Eldo Documentation

Eldo documentation is available on the FPGA lab machines in the following locations.

HTML: /tools/Mentor/AMS_2008_2_IXL/docs/htmldocs/eldo_rf/eldo_rf.htm

PDF: /tools/Mentor/AMS_2008_2_IXL/docs/pdfdocs/eldo_rf.pdf

E. Setting up an Eldo netlist

An Eldo netlist file is of the form "filename.cir". You will need to create this file and add all appropriate portions to it: active/passive network to be analysed, voltage/current sources, simulation

options, plotting functions, comments etc. To run Eldo on your netlist, use this command on the terminal window: “eldo filename.cir”

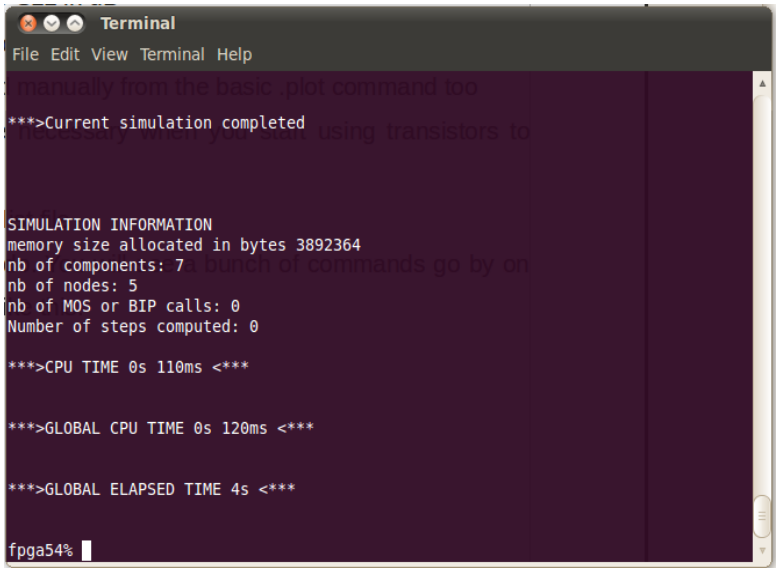
F. Example circuits and simulations

Example #1: L-match circuit AC and S-parameter simulation

Look at sample netlist #1 on the class website: Lmatch.cir. This file simulates the high-pass L-match circuit we designed in class to transform a 200Ω load impedance down to 50Ω.

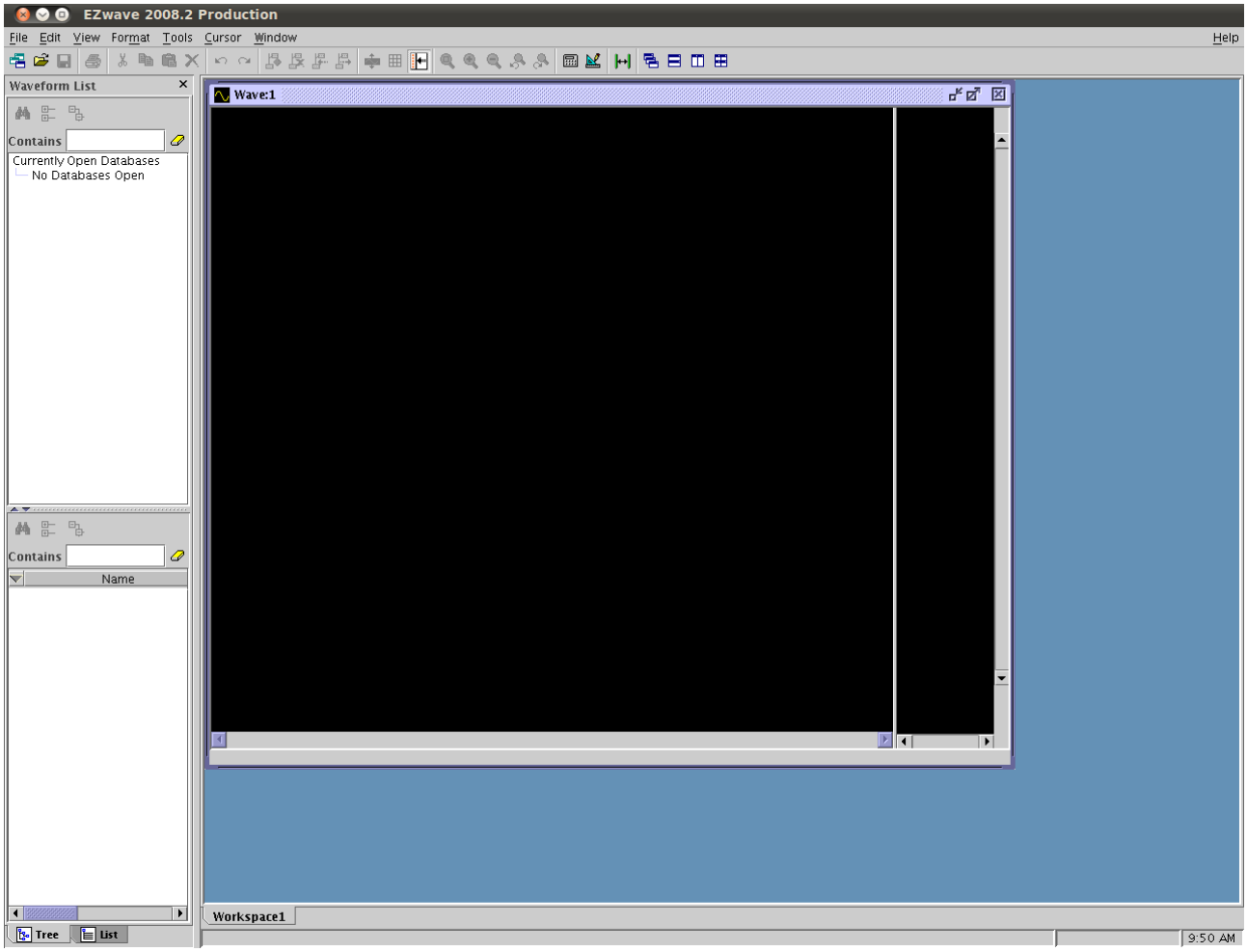
- The first line of the .cir file is a comment. You can use this to describe what you are doing with the netlist e.g. “S-Parameter analysis for Impedance Matching using L-match”. Any other lines in the body of your netlist that you want to use for comments should start with a “*”; e.g. “*-----LC circuit-----”
- C1 is a capacitor between nodes 1 and 2 of value 0.765pF
- L1 is an inductor between nodes 2 and 0 of value 7.66nH
- RL is the 200Ω load resistor
- node '0' is the reference ground node
- V1 is a 'port' source at the input. You will need to add a port whenever you want to run an S-parameter simulation. In this case, the load resistor RL is known, and we are interested only in transforming it to a desired impedance at the input. Therefore, we will run a 1-port S-parameter simulation with the input port source set to 50Ω. This will allow us to simulate and plot S_{11}.
- The “.OP” statement simulates the DC operating point of the circuit.
- The “.AC” statement performs an AC analysis. In this case, we are sweeping between 10MHz and 10GHz, with 100 points per decade (of frequency sweep)
- The “.plot” command here plots the S_{11} in dB
- The “.extract” can be used to extract further values from the .AC simulation (such as Zin, etc). All of these can be plotted manually from the basic .plot command too
- The “.option compat” will become necessary when you start using transistors to assure spice model compatibility
- “.END” indicates the end of the netlist file

Now, let us try running this netlist with Eldo (run this command: “eldo Lmatch.cir”). You will see a bunch of commands go by on your screen, and it will end in something like this:

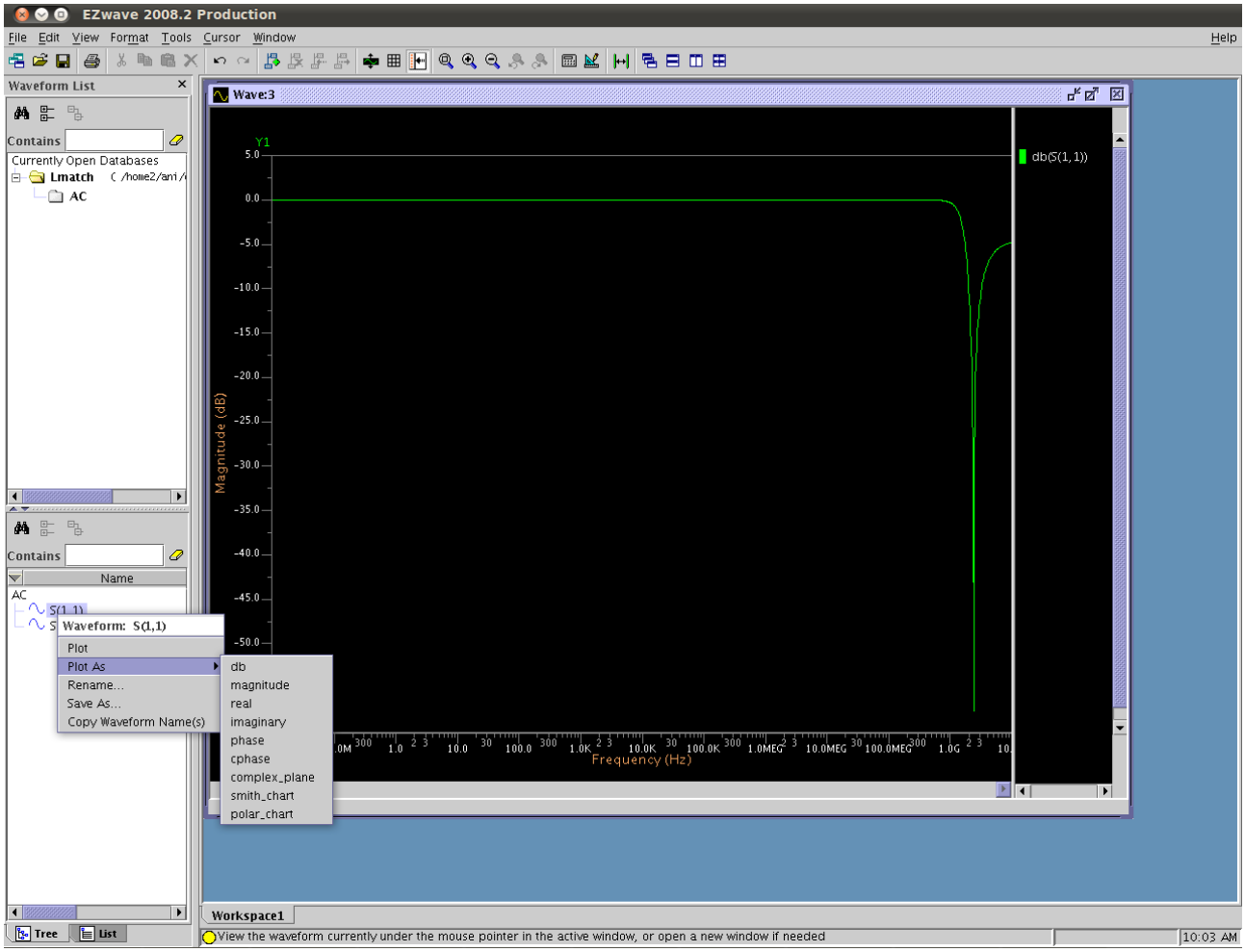


If you look at the files in your working directory, you will see a bunch of other files named Lmatch.*, which are the output files, log files etc. The actual simulation output is contained in Lmatch.wdb.

Viewing the output: The output can be graphically viewed by running the “ezwave” command. When you do this, the ezwave window opens up, and looks something like this:



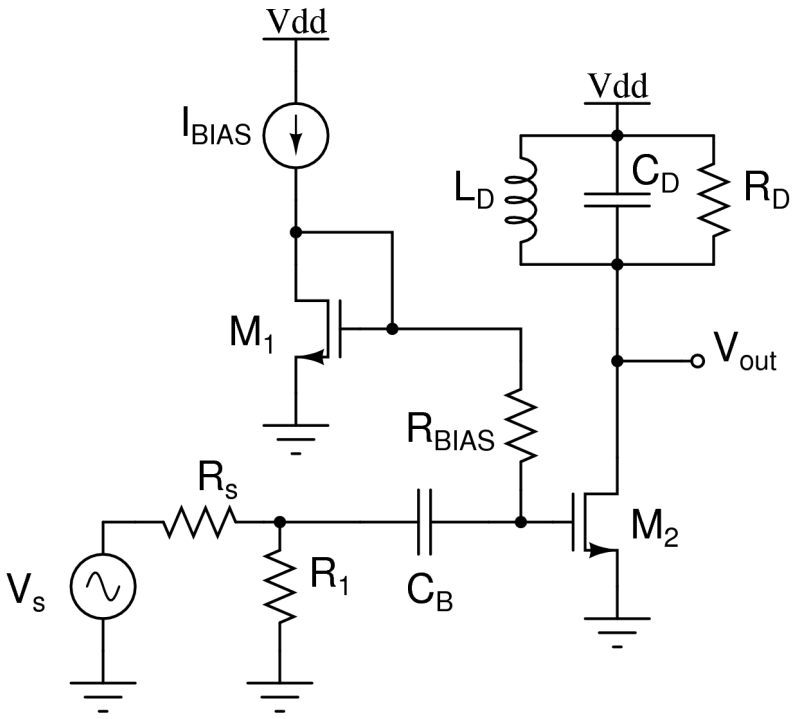
You will now open the database by navigating to: File --> Open --> Lmatch.wdb. On the top left pane, you will see 'Lmatch' under currently open databases. If you click on the '+' sign, it should open up the AC simulation result. When you left-click on 'AC', the bottom left pane should show all the simulation results – S(1,1) [all S_{11} results] and SDB(1,1) [S_{11} magnitude in dB from .extract command]. To plot, right-click on the appropriate result. Right-clicking on SDB(1,1) will allow you to plot only S_{11} in dB versus frequency. Right-clicking on S(1,1) will allow you to plot S_{11} in a variety of ways (through the 'plot as' choice that pops up): dB, magnitude, real, imaginary, smith-chart etc, as shown in the screenshot below.



As you can see, the S_{11} at 2.4GHz is around -45dB, which indicates an excellent match to 50Ω at the input.

Example #2: LNA circuit – Gain, NF and IIP3 simulations

In this example, we will simulate the 900MHz LNA circuit shown in the figure below. The sample LNA netlist is available on the class website: lna.cir. This netlist performs Gain, NF and IIP3 simulations. Recall the F_{min} for this circuit (it was covered in class). You can compare it to the simulated value to see how far you are from the optimum noise match situation.



- The ".include ibm013.lib" line adds the transistor models to the netlist.
- .subckt creates a sub-circuit definition. In this case, the sub-circuit models the additional drain and gate noise sources, as well as the correlation coefficient.
- An example transistor definition is (in quotes, bold) "**M1 N3 N3 0 0 CMOSN W=1.0U L=0.13u**". This indicates the following: transistor name is M1; the order of nodes is drain-gate-source-bulk; model name is CMOSN; Width W and Length L are 1um and 0.13um respectively.
- For input matching, the simulation is similar to that used in HW1. For IIP_3 and Gain simulations, we use a V1 port definition that includes two tones fund1 and fund2.
- The ".param" statement is used to define variable values, in this case Pin, f1 and f2.
- The ".step" statement is used to sweep variables. In this case Pin is swept from -30dBm to +10dBm in steps of 1dB.
- The ".sst" statement performs a steady state analysis. This is very useful in analysing circuits of a large-signal nature, but having a periodically varying operating point. In this

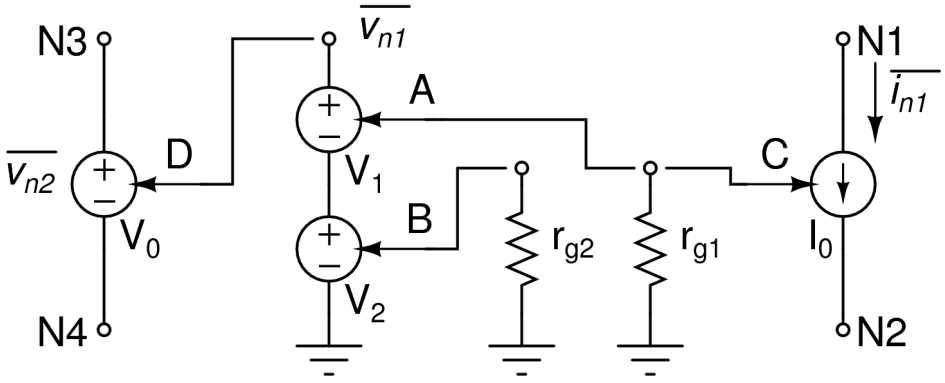
case, we are using it to analyse the intermodulation performance of the LNA. You should be careful in specifying the number of harmonics of the fundamental that you want to analyse. In the case of nominally small-signal, mildly non-linear circuits like LNAs, the number of harmonics will not affect the simulation results too much {nharm1=10 will usually be sufficient}. However, simulations on mixers and VCOs will show greater impact.

Modelling noise correlation and short-channel effects:

Eldo inherently models noise based on long-channel assumptions i.e. $\gamma = 2/3$. Therefore we need to model an additional noise power with an equivalent $\gamma = 2 - 2/3 = 4/3$. We will also need to include $\alpha = 0.85$, $\delta = 4$ and $|c| = 0.395$ in this additional noise power.

To keep things relatively simple, we will model all additional noise sources as resistors. Remember that drain thermal noise current, induced gate noise voltage (in series with the gate) and resistor thermal noise current/voltage are all proportional to 4kT in some form or the other. We can therefore model all MOSFET noise sources by choosing appropriate resistor values and/or applying scaling factors on these in the form of different controlled sources (VCVS, VCCS, CCVS, CCCS). We will show below that we can also model the correlation-coefficient c using controlled sources.

Start off with two 1-Ω resistors, r_{g1} and r_{g2}, with each having one side grounded and the other left open as shown below. The DC voltage on the second node is also 0.



Let us start off with drain current noise – it is the easier of the two to model. The noise of a 1-Ω resistor is $4kT\Delta f$. We want a current noise power of $\overline{i^2_{n1}} = 4kT\cdot(4/3)\cdot(g_m/0.85)\cdot\Delta f$, which is easy to model: determine g_m from the operating point of the transistor, and set the gain of the VCCS I_0 as $C = [(\gamma - 2/3)\cdot(g_m/\alpha)]^{1/2}$. Nodes N1 and N2 of I_0 are connected to D and S respectively of the appropriate transistor.

Gate noise is a bit more difficult to model, due to correlation. We will achieve this in the following way: remember that the thermal noise of resistors r_{g1} and r_{g2} are completely uncorrelated, and the noise voltages are therefore in quadrature. We will scale these by two parameters $A = |c|$ & $B = (1 - |c|^2)^{1/2}$, and add them. The resulting noise power $\overline{v^2_{n1}}$ is equal to $4kT\Delta f$, but the noise voltage is rotated by an appropriate angle relative to the noise voltage of r_{g1} such that the correlation between

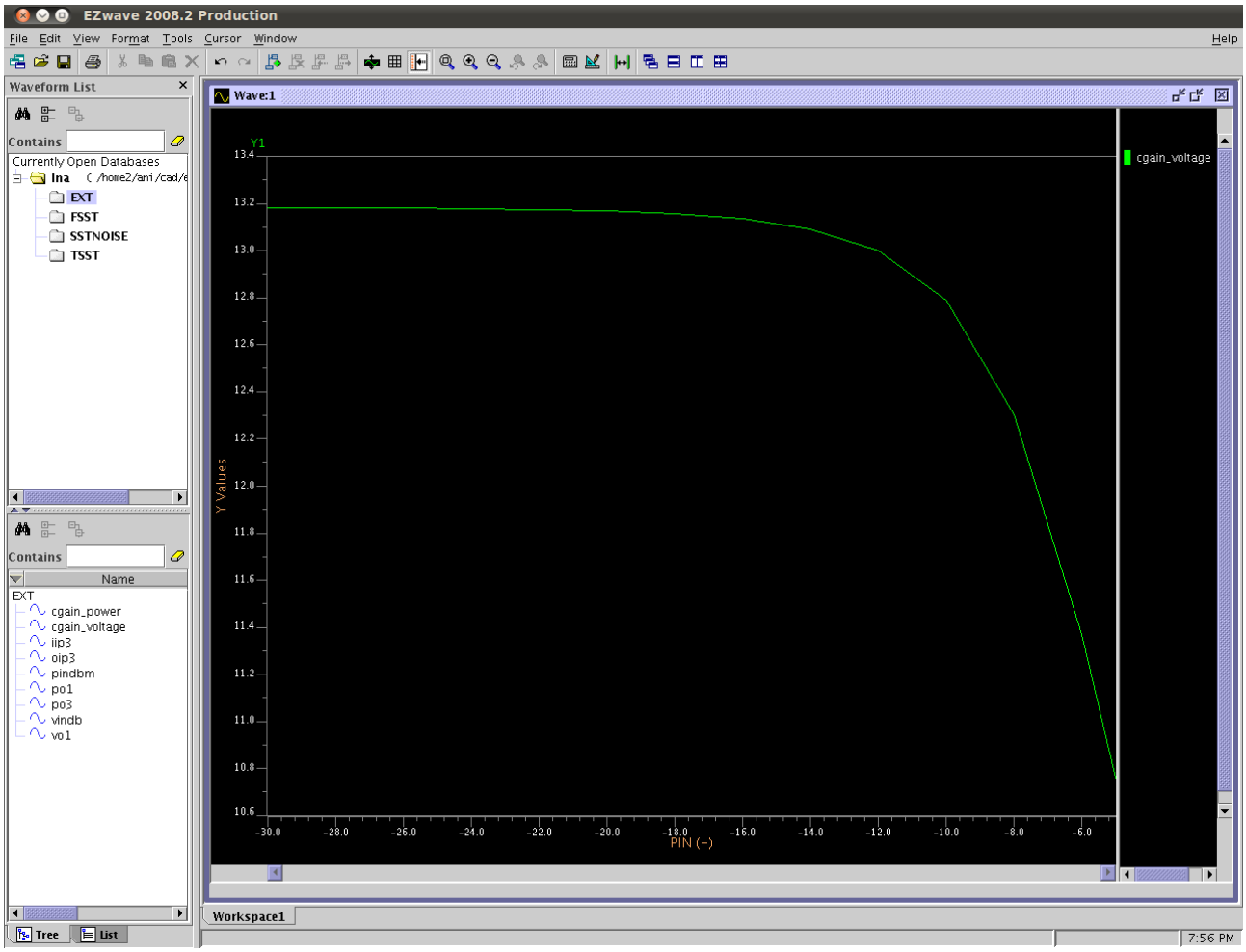
the two is |c| = 0.395. Finally, voltage $\overline{v}_{n1}$ is now scaled by a value D = $[(\delta)\cdot\{\alpha/(5g_m)\}]^{1/2}$ to create a noise voltage $\overline{v}_{n2}$ across N3 and N4. This voltage is to be connected in series with the gate of the appropriate transistor.

Note that γ = 2, δ = 4, α = 0.85 and |c| = 0.395 can be (and have been, in the example circuit) easily parameterised. However, g_m is a function of the size and bias current of the transistor. Every time these are modified, g_m has to be determined from an operating point simulation and the value appropriately updated in the netlist.

Simulations and results:

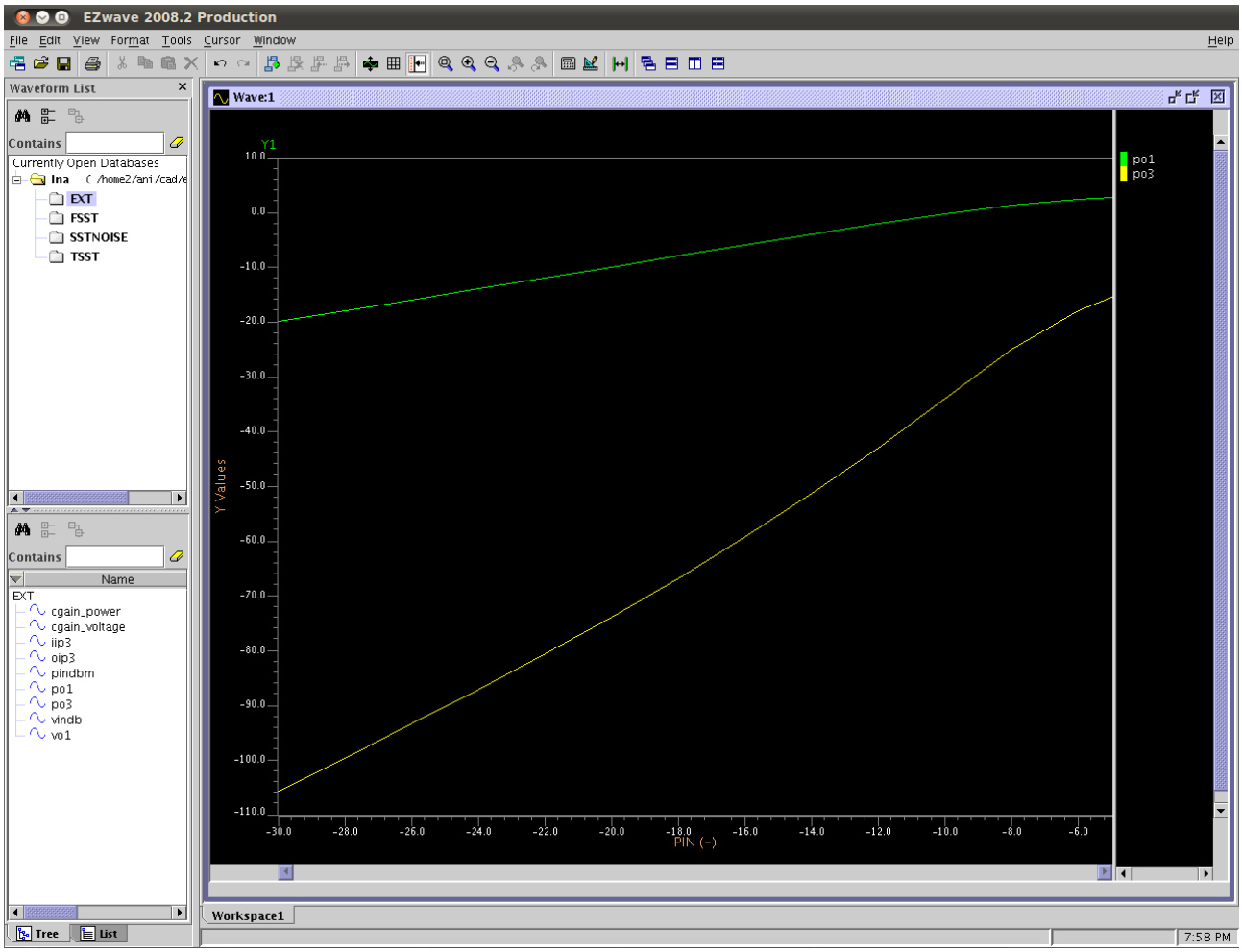
Running the command “eldo lna.cir” will perform only a .op simulation and will enable you to determine g_m value(s). Running the command “eldo lna.cir -define rf” will also run the .sst simulations for NF and IIP3.

Voltage Gain as a function of Pin (plot cgain_voltage under EXT):

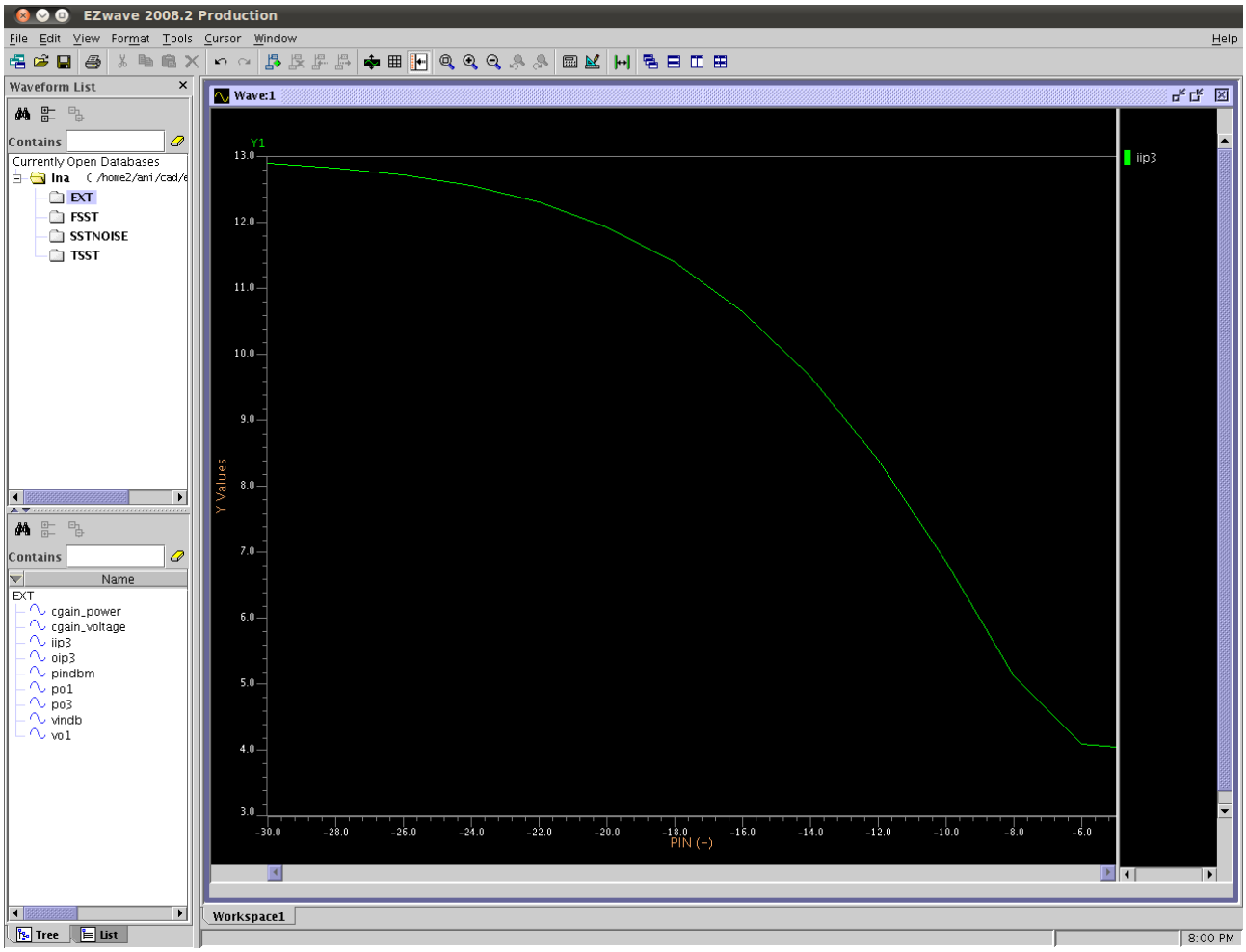


The actual voltage gain value is the one at low powers (~13.2dB).

Fundamental and IM3 powers as a function of Pin (plot po1 and po3 under EXT):

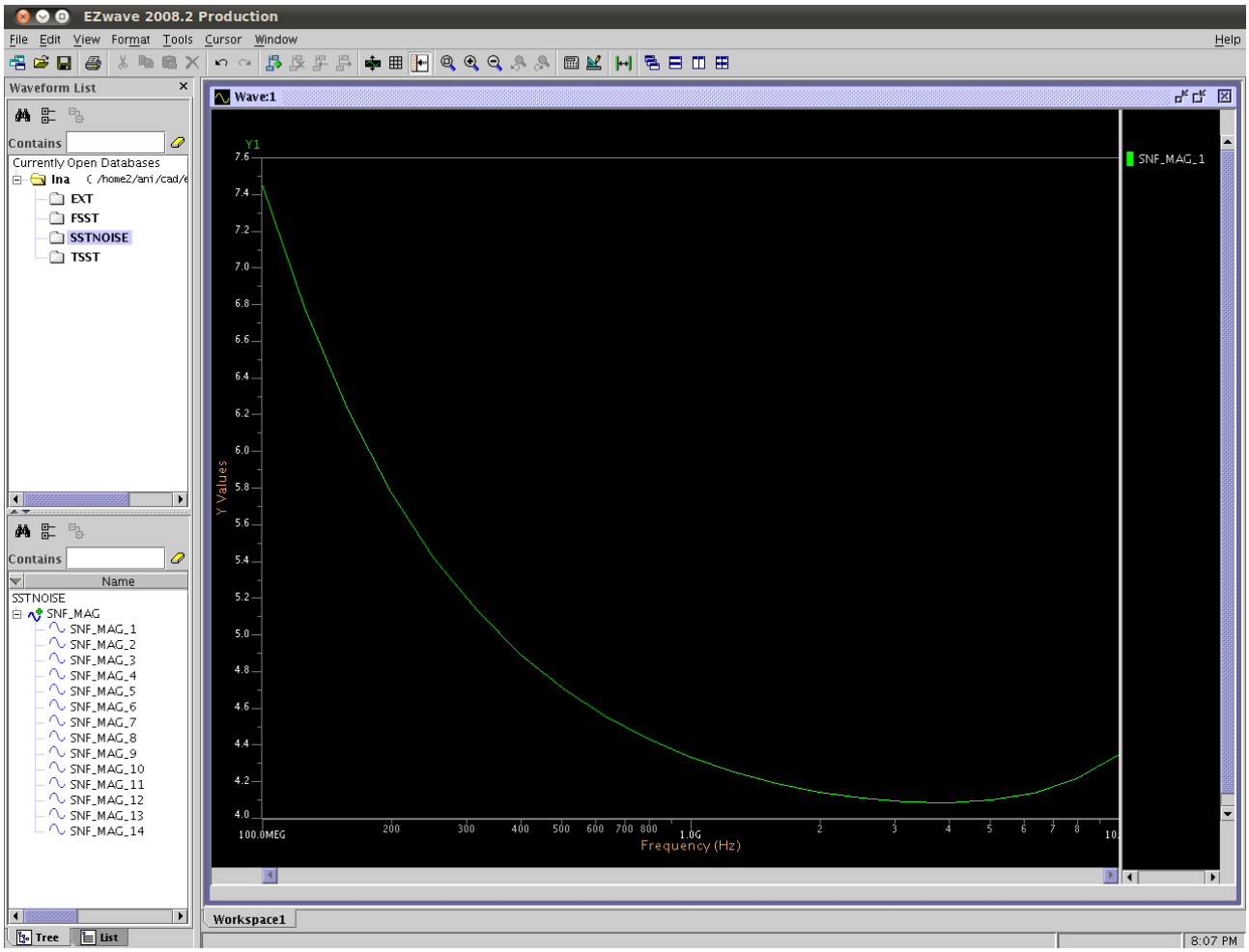


IIP3 as a function of Pin (plot cgain_voltage under EXT):



The actual IIP3 value is the one based on extrapolation at low input power (12.89dBm in this case).

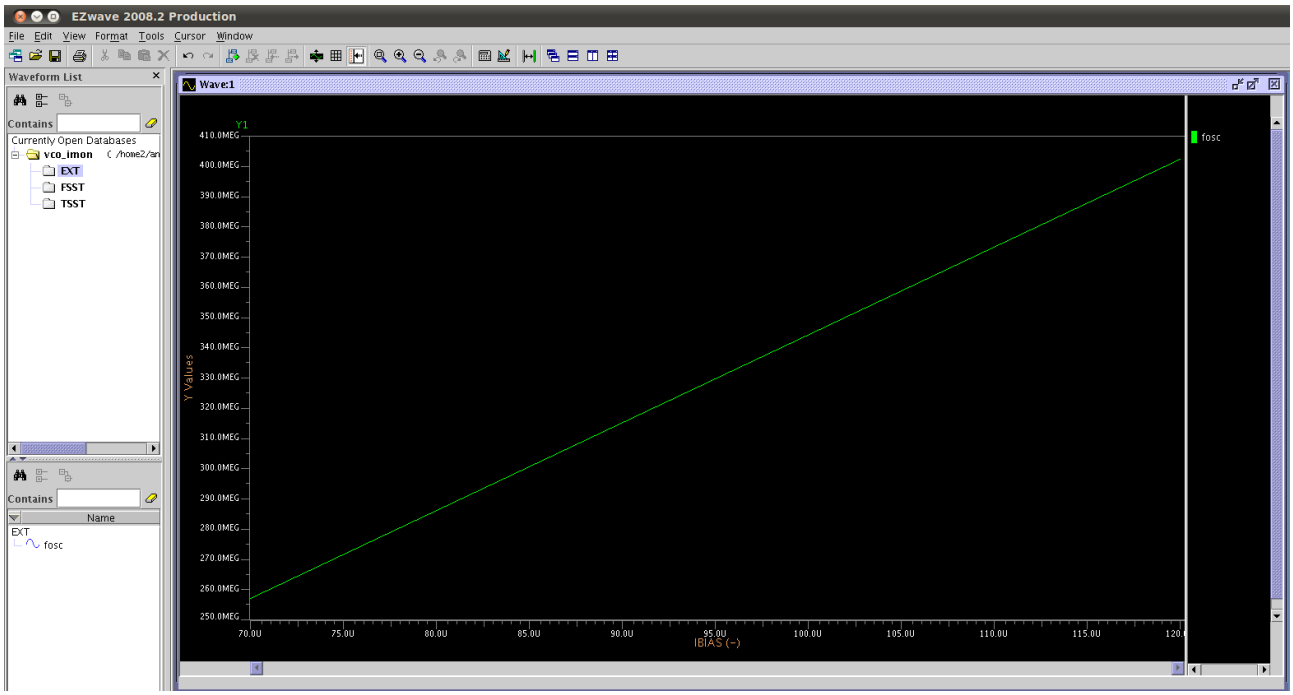
NF as a function of frequency (plot SNF_MAG_1 under SSTNOISE):



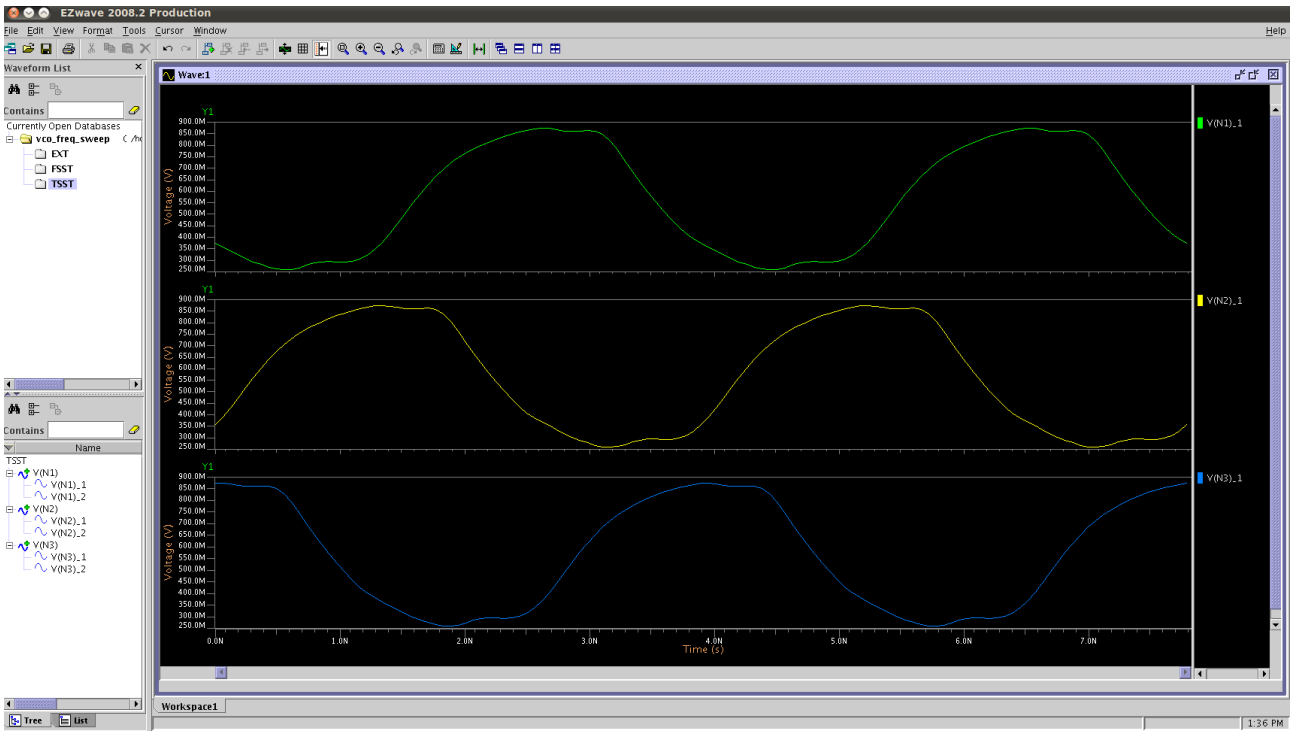
Example #3: VCO circuit – Phase noise and frequency-vs-control voltage/current simulations

In this example, we will simulate a 3-stage current-starved Ring Oscillator circuit. Remember that the delay of an inverter is a function of it's 'bias' current. If the current is reduced, the inverter delay increases, and therefore the frequency of oscillation decreases. The VCO is controlled by a current input, and it tunes between 250MHz and 400MHz. The sample VCO netlist is available on the class website: vco.cir. Since oscillators are autonomous circuits, the steady state analysis syntax is slightly different for VCOs as compared to other RF circuits. This netlist performs phase noise and frequency versus control-current simulations.

The netlist is set to sweep current I_C and and plot f vs I_C. Remember that for each change in I_C, the circuit reaches a new steady-state oscillatory condition. Study the netlist to see how it extracts f vs I_C behaviour shown below. Phase noise simulation has been disabled because you don't need phase noise (at this time) for each control current setting; phase noise simulations are typically time consuming.



You can also plot the transient outputs at different points using the **.plot tsst ***** command. The voltages at nodes n1, n2, n3 are shown below.



The netlist needs to be slightly modified to run phase noise simulations at a single control voltage/current value, because this does not need a steady-state **sweep** analysis. For this simulation, comment out this line:

.step param ibias 70u 120u 10u

and uncomment these lines:

***.sstnoise v(n3) harm (1) dec 100 10K 10G**

***.plot sstnoise db(sphi)**

The ring oscillator phase noise versus frequency offset is shown below.