TSEK38 Radio Frequency Transceiver Design: Laboratory Exercise

Lab manual for TSEK38 2019-VT1.

Lisam course page:
https://liuonline.sharepoint.com/sites/TSEK38/TSEK38_2019VT_FU

Course responsible: Ted Johansson (ted.johansson@liu.se)

You can work on your own with this lab or use the schedules lab session to get support from a lab assistant.

Please document all answers to the exercises in the text (screen dumps, etc.). It will be used to verify PASS on this lab with the examiner.

Contents

LINKÖPING UNIVERSITY

Introduction

1.1 What is ADS?

Keysight Advanced Design System (ADS) is a powerful electronic design automation software system for high-frequency design. It supports the design of systems and RF designs for applications such as RF/microwave modules, integrated MMICs for communications, and aerospace/defense applications. The software supports several different types of simulation technologies such as circuit frequency- and time-domain simulations, and electromagnetic field simulations, including optimization capabilities.

Usually, ADS is run in the Linux environment, which also enable integration with Cadence IC design system, but ADS (virtually identical) can also be installed and run in Windows (no Cadence integration) using the same software licenses and simulation files. In this lab, we will use the ADS for Linux.

1.2 Objective of the laboratory exercise

The objective of this laboratory exercise and project work manual is that the student learns how to use ADS for RF system level design. The first circuits that will be simulated are an amplifier and a mixer. They will be evaluated for different simulation modes: harmonic balance simulation (periodic steady-state), S-parameters simulation, envelope simulation, noise simulation, and harmonic and intermodulation distortions. During the exercise, the student will learn how to setup a design project and display data in several ways and combinations, such as optimization of impedance matching networks, analysis of an EDGE RF signal in terms of trajectory and spectral components, noise power simulations, intermodulation terms in a mixer or blockers in a receiver. The last part of the lab combines the analog/RF simulation engine together with the Ptolemy simulator for a receiver model with a QPSK modulated signal applied. The ADS Ptolemy software provides simulation tools, which can be used to evaluate and design modern communication systems.

At the beginning of the lab, the instructions are very detailed, but when one step is completed and well understood, it can be easily used in subsequent simulations. The laboratory exercises are useful both as an introduction to ADS and also as basis for the project work in the course.

2 Creating a new ADS project

2.1 Starting ADS

ADS is updated about each year with a new major version. This lab manual is using ADS2019. The screen dumps were done using ADS2016, but everything in ADS for RF/microwave simulation is virtually the same with ADS2019.

Open a terminal window and establish an ssh connection to the ixtab server through the command: “ssh -X ixtab.edu.isy.liu.se”, then input your credentials.

Write the following two commands to load the module and start running ADS (Keysight is known previously both as "Agilent" and "HP", therefore the module path "agilent"):

```
module add agilent/ADS2019
ads &
```

A first window may pop up, asking to choose the simulation software license. Select the first alternative, the **ADS Inclusive**.

A new window, “Get Started” pops up. The first time, you may get a question to choose the design flow. Click on **RF/Microwave** and then click on Finish.

2.2 Creating a new workspace

A new window, “Get Started” pops up. Create a new workspace by clicking on ‘New workspace’ in Figure 1.

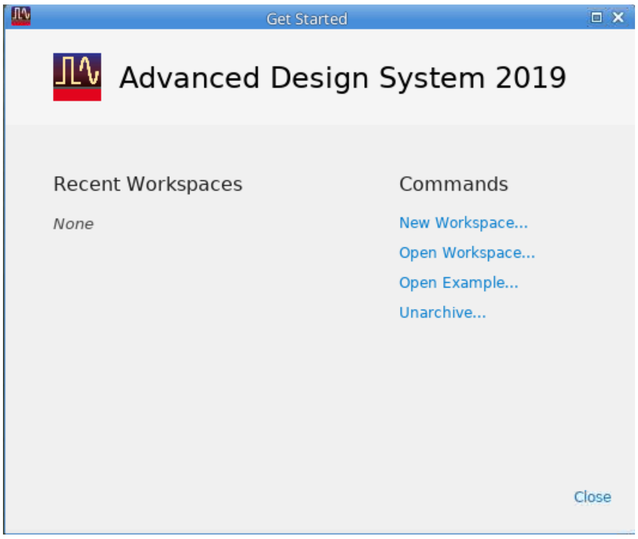


Figure 1. The “Get started” dialogue. Use it to create a New workspace.

A new window pops-up, click on Next. Then, write the name of the workspace, typically **‘Lab_TSEK38_wrk’**, and its location (use you own directory ("Create in") of choice) as in Figure 2. Click on ‘**Finish**’ to create the workspace.

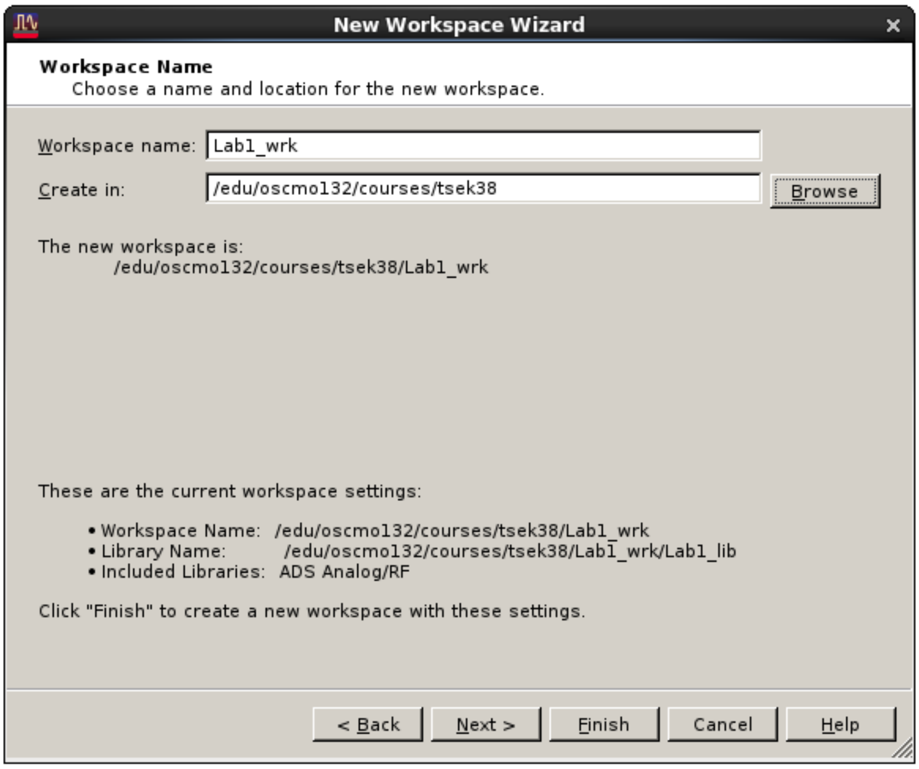


Figure 2. Name and location of the workspace

Open a new schematic window by clicking on 'New Schematic' icon, , as in Figure 3. A new schematic window will be displayed on the screen. Call the cell **'Lab_HB'** and then click OK. The design file is now available in the 'Folder View' (in the 'Main Window'). Now, go to the schematic window.

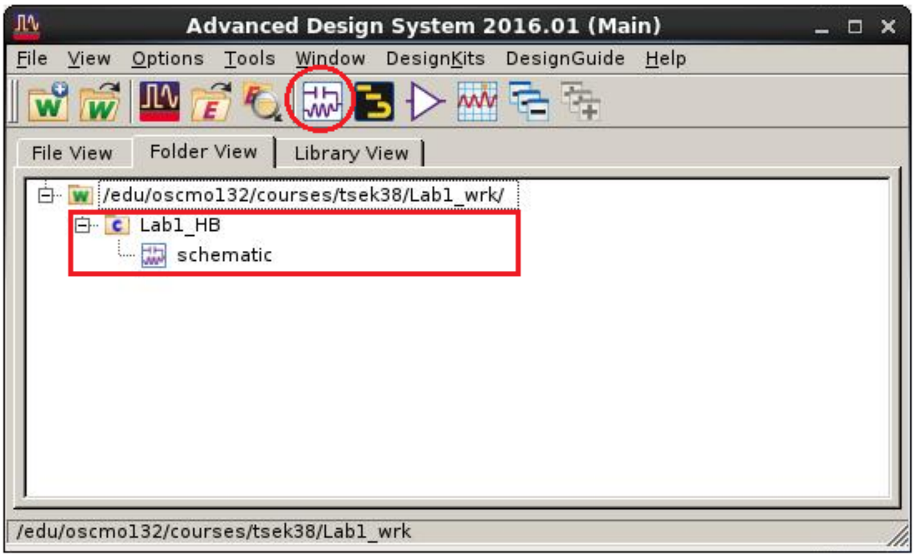


Figure 3. Create new schematic

2.3 Simulation of an amplifier

In this section we will simulate a simple amplifier in different ways in order to understand the available capabilities in ADS to display and evaluate simulation data. There are several simulation options in ADS, but we will focus on the following simulation types: S-parameter (SP) simulation, Harmonic Balance (HB), and Envelope (ENV) simulation, which all will be described shortly. All simulation components and part components can be found in the drop-down lists and palettes to the left in the schematic window, see Figure 4.

Note: The specifications of a component can easily be shown by placing it in a schematic, double-click, and select Help.

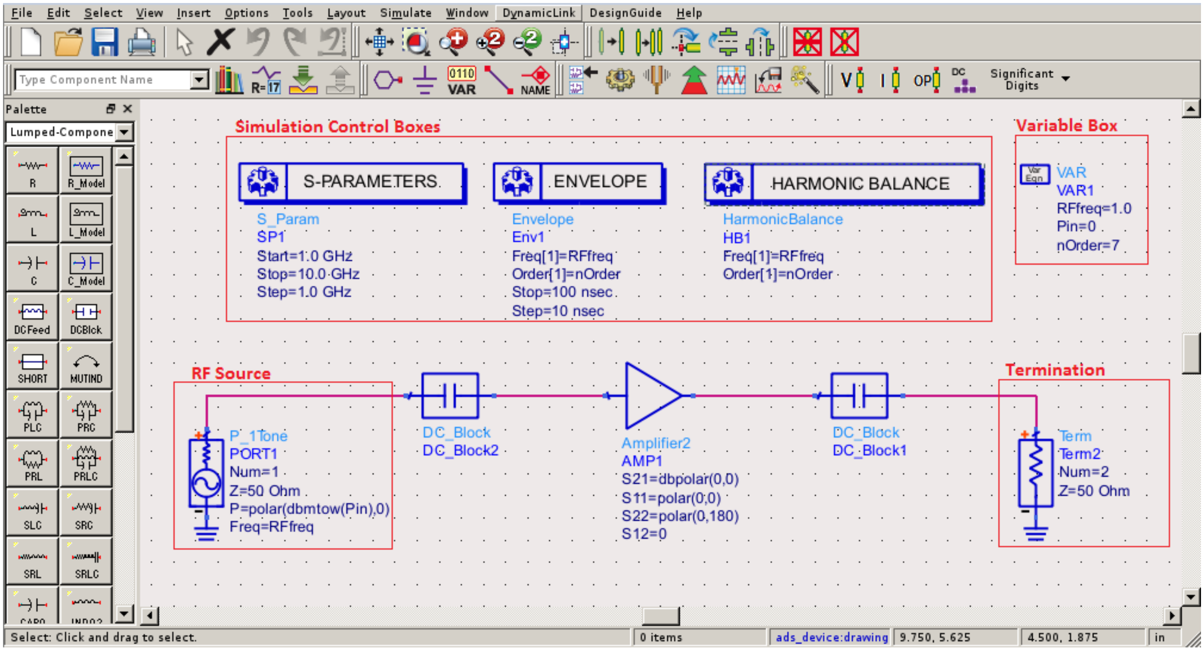


Figure 4. ADS schematic.

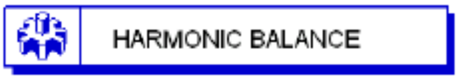
2.4 Simulation modes

A description of all simulation options can be found in the ADS Help (search for "Simulation Controllers"), where other simulation modes not covered by this laboratory exercise are included.

2.4.1 S-parameter (SP) simulation

The S-parameter controller ('S_Param') is used to define the signal-wave response of an n-port electrical element at a given frequency. It is a type of small-signal AC simulation that is commonly used to characterize a passive RF component and establish the small-signal characteristics of a device at a specific bias and temperature. The simulation tasks involving S-parameters also include information about how to optimize component values and simulation expressions.

2.4.2 Harmonic Balance (HB)



The Harmonic Balance (‘HarmonicBalance’) controller is used for simulating analog/RF and microwave circuits considered as nonlinear. The simulation is done in the frequency domain (using harmonics of a fundamental frequency components), compared to conventional time-domain simulations (as in SPICE and many other simulators). Within the context of high-frequency circuit and system simulation, harmonic balance offers benefits over conventional time-domain transient analysis. Harmonic balance captures the steady-state spectral response directly while conventional transient methods must integrate over many periods of the lowest-frequency sinusoid to reach steady state. Harmonic balance is faster at solving typical high-frequency problems that transient analysis cannot accurately solve or can solve at prohibitive costs. Harmonic balance is more accurate at solving high frequencies where many linear models are best represented in the frequency domain. Use the Harmonic Balance controller to determine the spectral content of voltages or currents, calculate quantities such as third-order intercept points, total harmonic distortion, intermodulation distortion components, and noise.

2.4.3 Envelope (ENV) simulation

The Envelope (‘Envelope’) controller is best suited for a fast and complete analysis of circuits with complex signals such as digitally-modulated RF signals. It combines features of time and frequency-domain representation by permitting input waveforms to be represented in the frequency domain as RF carriers with modulation envelopes that are represented in the time domain. The Circuit Envelope Simulation is highly efficient in analyzing circuits with modulated signals, because the transient simulation takes place only around the carrier and its harmonics. In addition, its calculations are not made where the spectrum is empty.

3 Creation of simulation schematic

We will start by examining the amplifier to be used in the simulations. It can be found in category 'System-Amps & Mixers' in the 'Palette', on the left side of the schematic window, the palette symbol says 'Amp' and part 'Amplifier2' when placed in the schematic. Add one amplifier to the schematic. In this exercise we will use an amplifier with a gain of 0 dB, which means that the gain in linear scale is 1.

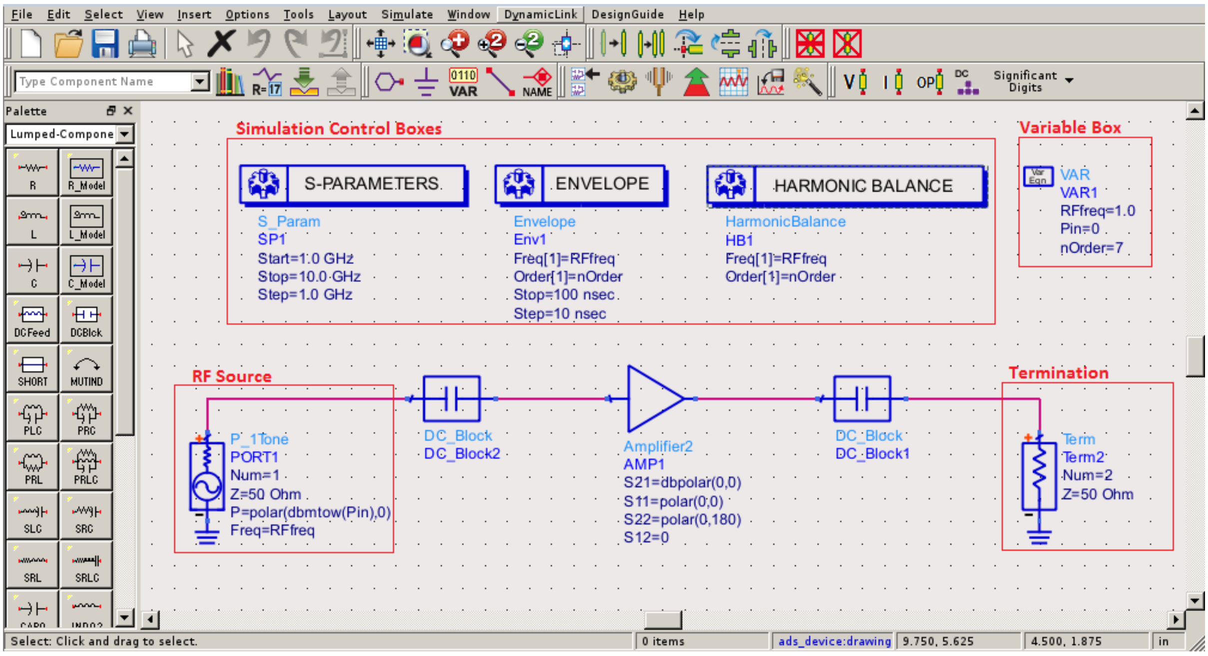


Figure 5. Adding components to the schematic

Add two DC-blockers (decoupling caps) by typing 'DC_Block' in the 'Search all libraries' and then click/drag the search result. Add the DC_Block at the input and output of the amplifier as in Figure 5.

Add the simulation control boxes for S-parameters, Envelope simulation, and Harmonic Balance to the schematic. The controllers can be found in the 'Parts search drop-down list under 'Simulation-S_Param', 'Simulation-Envelope', and 'Simulation-HB'.

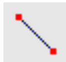
Add an input port 'P_1Tone' to the schematic from category 'Sources-Freq domain', where you also can find several different types of sources. At the output of the amplifier you should place a termination port, 'Term', found in 'Simulation-HB' and 'Simulation-S_Param'.

Add ground connections, which are available next to the 'Part' drop-down list. For the Envelope and Harmonic Balance control boxes, and 'P_1Tone' source, change the following variables by double-clicking the symbol:

```
Value (Old)            Value (New)
Freq[1]=1.0 GHz        Freq[1]=RFfreq GHz
                       (note that "GHz" is missing in
                       figure below)
Order[1]=3             Order[1]=nOrder
```

The value 'Pin' denotes the input power in dBm. For the 'P_1Tone' source, change the following variable:

```
P=polar(dbmtow(0),0)   P=polar(dbmtow(Pin),0)
```

Connect all components by using wires. Click on the 'wire', , in the menu or use Crtl+w to connect all components as shown in Figure 6. Double-click on the wire at the termination port and name it 'vload'.

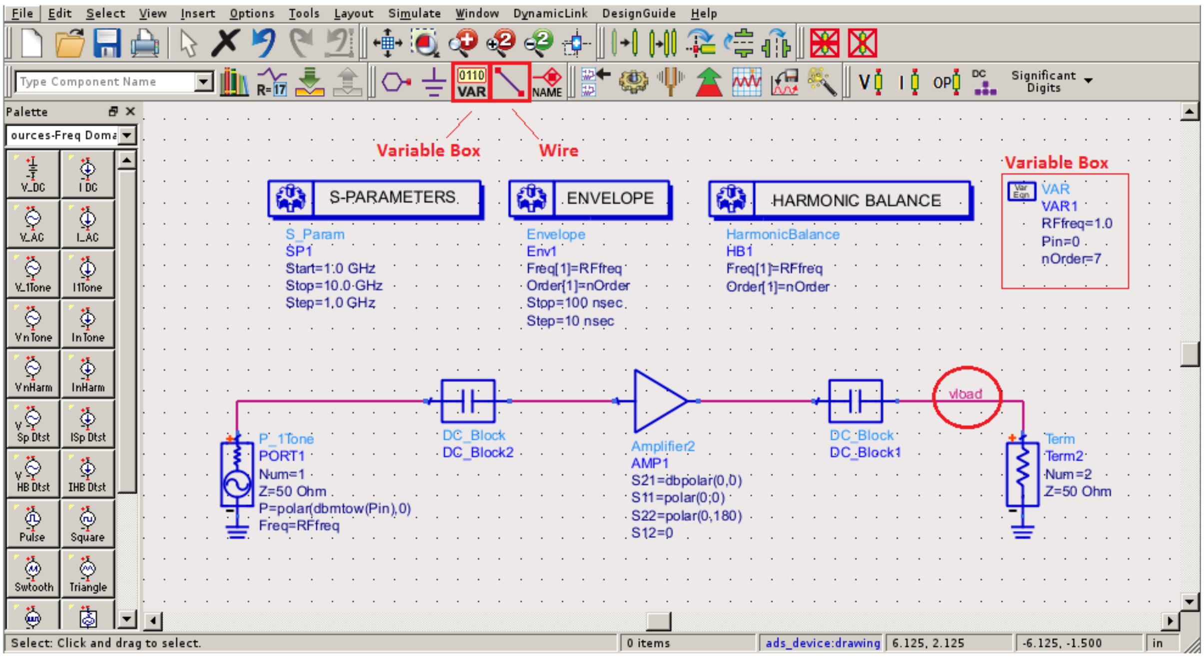


Figure 6. "Var" box and "Wire" command

Add a variable box, , to the schematic as shown in Figure 6. The 'Var' box contains variables that can be used in the simulation. It is convenient to use a 'Var' box since several different components can be updated by just changing one value instead of two or more, or it can even be used in optimization operations. Double-click on the 'Var' box and add the following variables. Write the name and its value, then press 'Add', otherwise by pressing 'Apply' the previous variable will be replaced by the new one.

Name	**Value**
RFfreq	1.0
nOrder	7
Pin	0

You should now have the same schematic as in Figure 6.

3.1 Properties of the amplifier – 'Amplifier2'

Double-click on the amplifier symbol in the schematic to examine the properties of the amplifier. Additional information is obtained using the Help button.

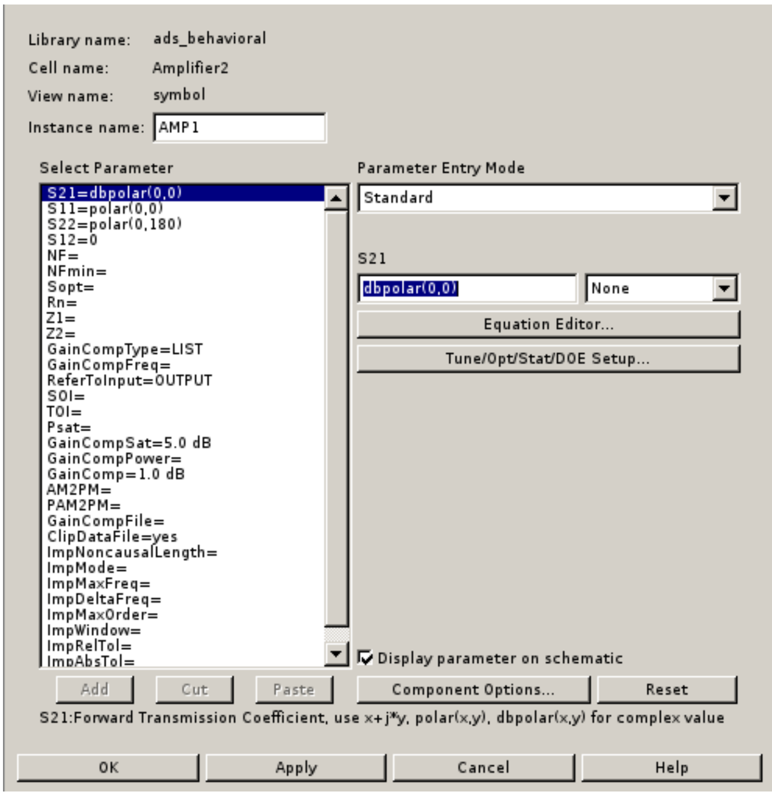


Figure 7. Amplifier properties

3.1.1 S_{ij} settings

From the property window we have the following S-parameter settings.

```
S21=dbpolar(0,0)
S11=polar(0,0)
S22=polar(0,180)
S12=0
```

It means that the reverse transmission coefficient (related to S_{12}) is 0, the input/output reflection coefficients (related to S_{11} and S_{22}) are 0, and that the forward transmission coefficient (S_{21}) is 0 dB (i.e. 1). For the moment we will leave these parameters as they are.

3.1.2 Amplifier compression point parameters

The amplifier model is based on an n^{th}-order polynomial. The polynomial with x as the input voltage and y(x) as the output voltage, can be defined in two ways that depend on the amplifier specification. If SOI, the second-order intercept point or IIP2, is specified for the amplifier, then equation (1) is applied, whereas for all other cases equation (2) is applied.

(1) $$y(x) = a_1 x + a_2 x^2 + a_3 x^3$$

(2) $$y(x) = a_1 x + a_3 x^3 + a_5 x^5 \dots$$

There are several combinations of gain-limiting variables that can be used to make the model behave like a real amplifier. However, not all of them work well together in the simulator. For more information please refer to the Help for the Amp2.

We will initially consider a combination of the third-order intercept (TOI or IIP3) and 1 dB gain compression point parameters (P_{1dB}). The constraint for this combination for GainComp = 1 dB is:

TOI > GainCompPower + 10.8 dB.

However, if only one of them is declared then according to the polynomial model

IIP3 = P_{1dB} + 9.6 dB.

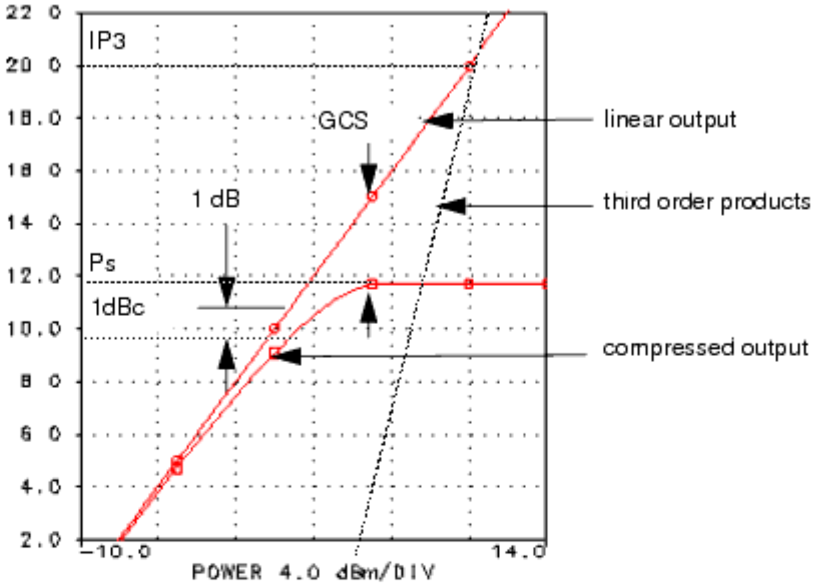


Figure 8. Amplifier model with IIP3 ≈10.6 dBm and P1dB ≈1 dBm.

4 Harmonic Balance Simulation

Initially, we will not run different simulation types at the same time. Therefore, choosing at first the Harmonic Balance, we would like to disable the 'S-parameters' and 'Envelope' simulation boxes, which can be achieved by selecting the both boxes and clicking on the symbol, , in the menu as shown in Figure 9. Now, only the Harmonic Balance simulation mode is active.

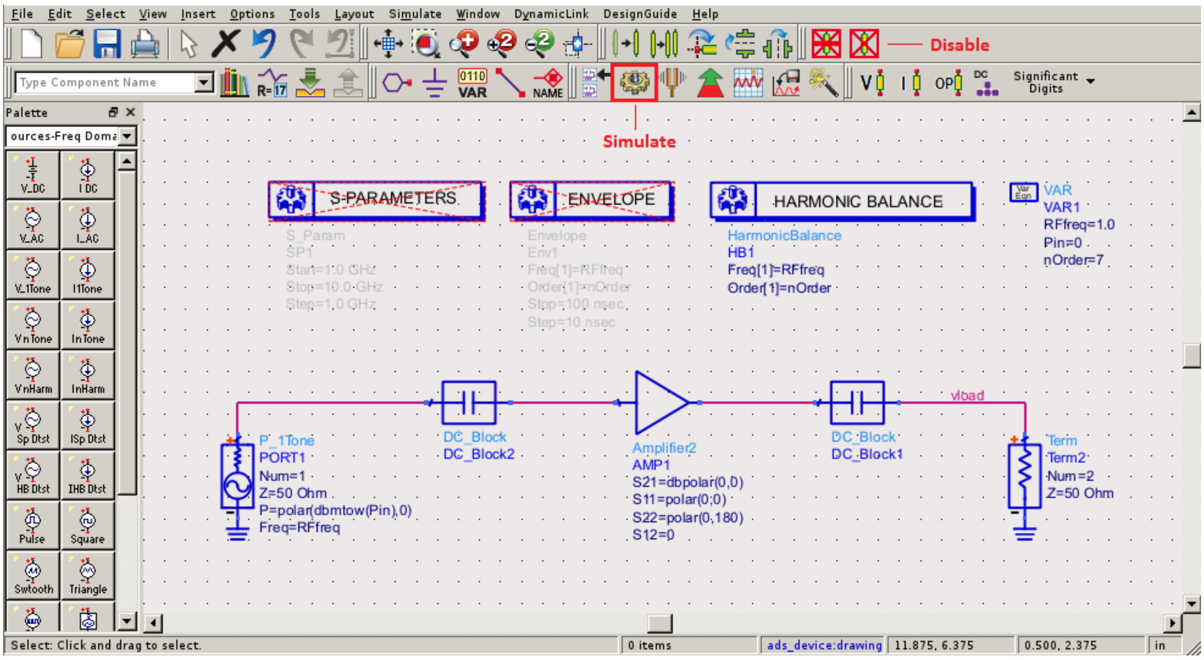


Figure 9. Disabling simulation control boxes

4.1 Running Harmonic Balance

To run the simulation, press the simulation button, , in the menu. A small simulation window will pop-up and tell the user about the status of the simulation.

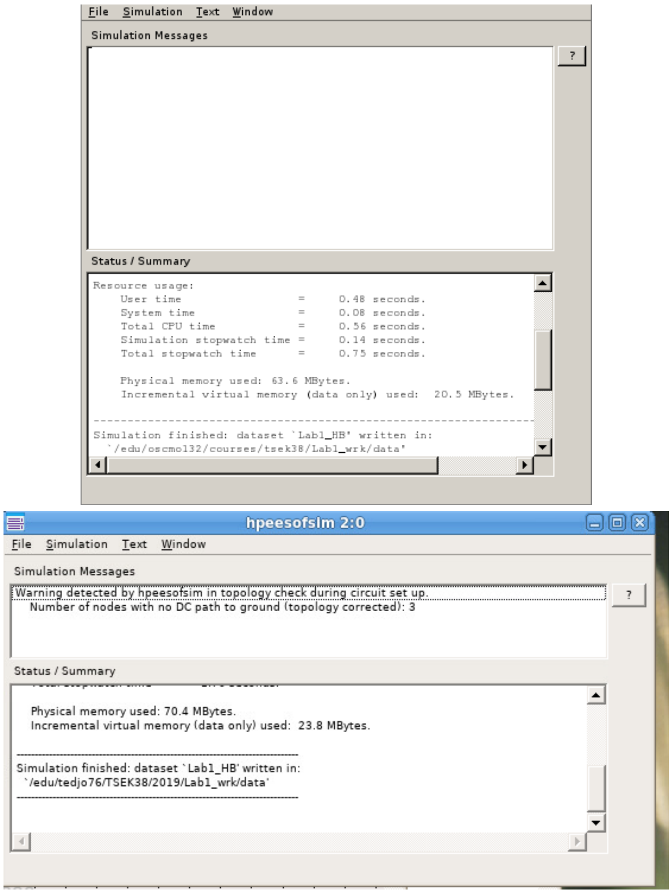


Figure 10. Simulation status. You may get as in window below with a warning message about "no DC path to ground".

If the simulation is successfully completed, a 'Data display window' (DDW) with the same name as the design will pop-up, in this case **'Lab_HB'**.

In ADS there are three types of files related to an actual design. They are:

- Design file (.dsn): schematic (and layout) for each design in the networks folder.
- Data display window (.dds): the window where the simulation results can be displayed. This file is saved in the project directory.
- Data set (.ds): the simulation results are saved in the 'data' folder in the project directory and are created/updated when a simulation is completed.

The DDW is initially empty and can be filled with any operation on the simulation data.

To verify that the amplifier gives a gain of 0 dB (1 in linear scale), we simply plot the harmonic contents of the node 'vload'. In the list of different display options (see

Figure 11) we choose the regular plot function, . Click on the symbol and place it in the empty white space.

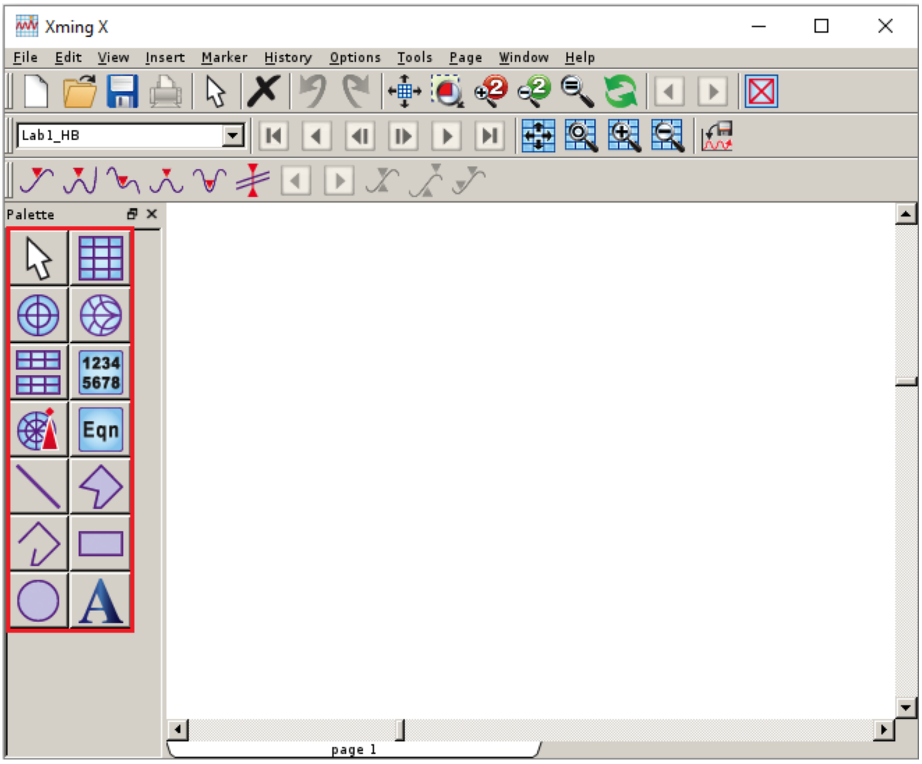


Figure 11. 'Data display window', DDW

A window labeled "Plot Traces & Attributes" is displayed.

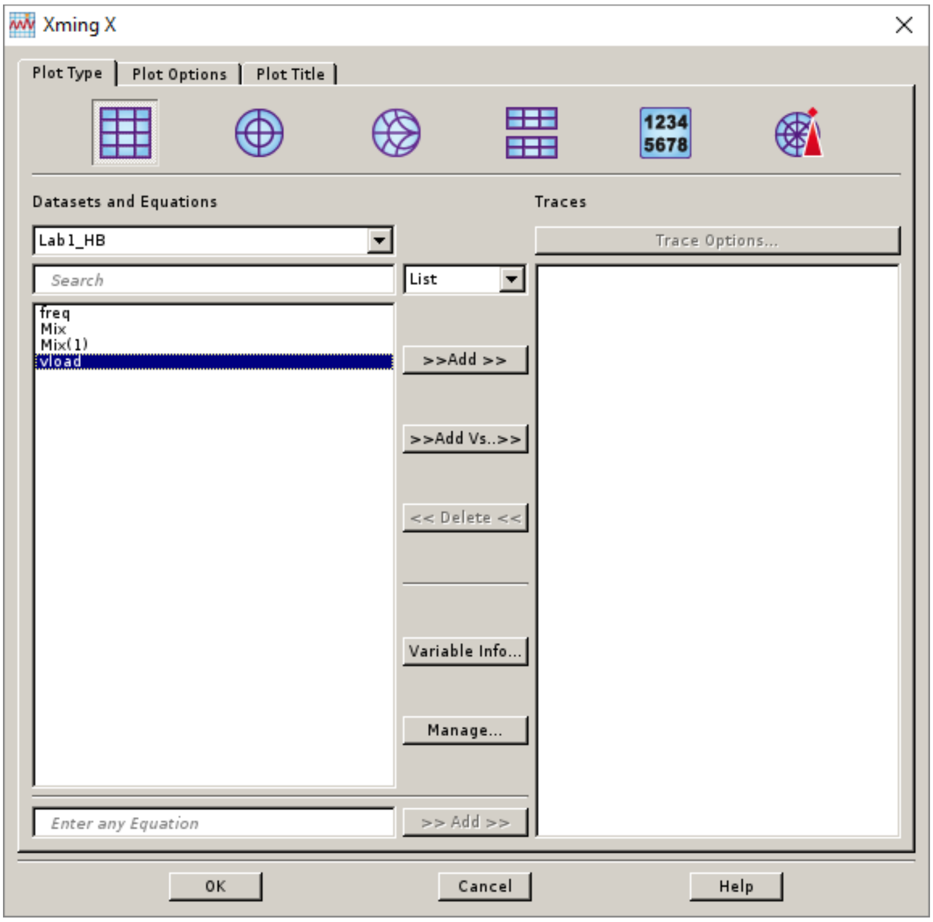


Figure 12. 'Plot Traces & Attributes

Click on the 'vload' in the 'Available data' column, then click on the 'Add' button. A radio dialog window is opened.

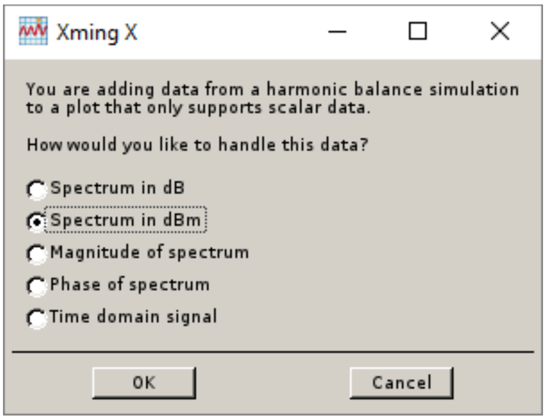


Figure 13. Plot options

Choose 'Spectrum in dBm' and then 'OK'. Now, the power of the fundamental and harmonics dissipated in the load will be plotted. Note: 'dBm' assumes that the power is measured for a 50 Ohm load. The resulting plot is in Figure 14.

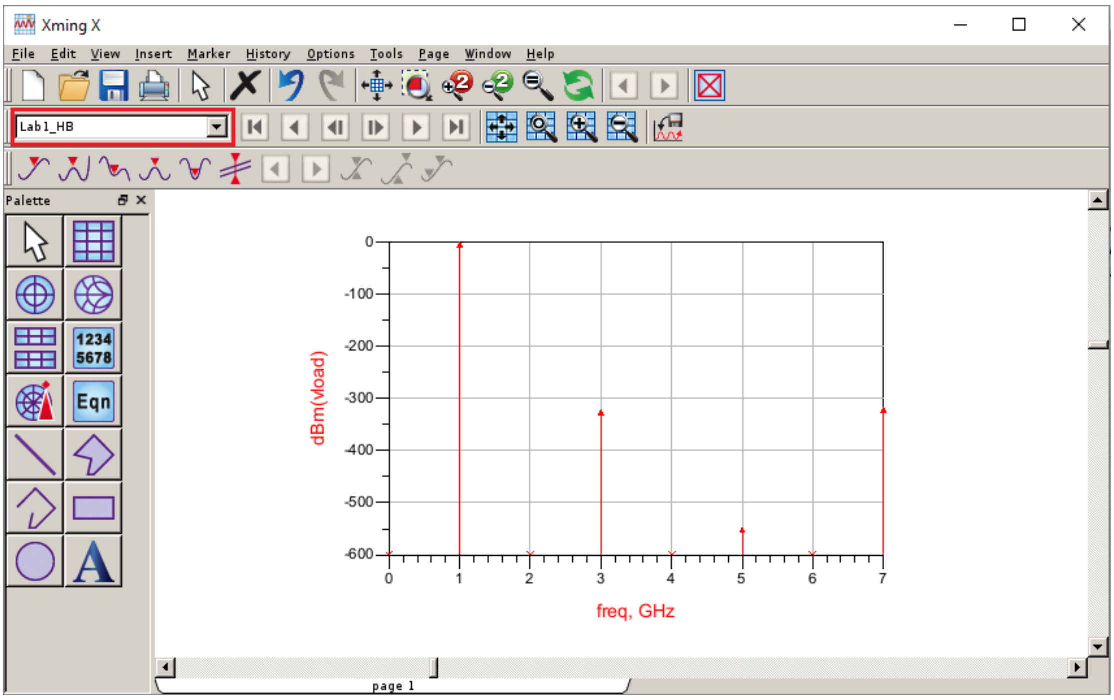


Figure 14. Harmonic balance simulation

As seen in Figure 14, the plot only contains data at the fundamental and harmonic components (see description of the harmonic balance simulation). In the drop-down list (marked in red in Figure 14) you can choose which dataset to plot in the case of several simulation results, but for only one data display window. Put a marker on the fundamental tone, at 1 GHz, and verify that the power is 0 dBm. Note that also very small harmonics may be present (< -300 dBm), but they are only there due to limited accuracy of the computations.

4.2 Harmonic balance with compression point parameters

For the amplifier, set the following variables:

```
Name          Value
GainComp      1.0 dB
GainCompPower 3
TOI           15
GainCompSat   - (Delete value)
```

It means that at the output power level of 3 dBm, the gain is compressed by 1 dB.

Exercise 4.2a: Re-run the simulation and observe the harmonic distortion that occurs at the output.

Start by re-running the simulation with the new amplifier parameters
Double-click on the plot in the DDW and go to the 'Plot options' tab and set the y-axis to range between -170 and 10 with a step of 10.
Make three copies of the plot window, by using 'ctrl-c' and 'ctrl-v'. Place them next to each other. In the first copy, click on the y-axis label 'dBm (vload)' and go to "Trace Expression". Change the expression to 'dBm (vload[1])'. In the two other copies, set the indexes to 3 and 5, respectively. In that way you can plot the power dissipated in the load resistance for the third and fifth harmonics.
Make two more copies of the original plot window (y autoscale). In these copies, set the "Trace Expression" to:

```
dBm(vload[1])-dBm(vload[3])
```

```
dBm(vload[1])-dBm(vload[5])
```

Then the difference in power between the fundamental, third, and fifth harmonic can be measured, and is denoted as 'dBc'.

A convenient way to measure power, signal levels, and gains during a simulation is to specify 'Simulation Measurement Equations'. To add such equations, go the schematic and enter 'MeasEqn' in the 'Part' drop-down list as previously described and place the component in the schematic. Double-click on the 'MeasEqn' component and add the following equations.

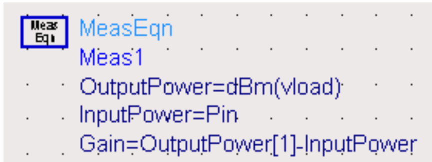


Figure 15. Measurement equations. 'MeasEqn'

When running the simulation these equations will be accessible in the 'Available data' box as in Figure 12. Re-run the simulation and add a new plot with the 'Gain' to see the gain for the fundamental component (change the y scale). Now the DDW should look like Figure 16.

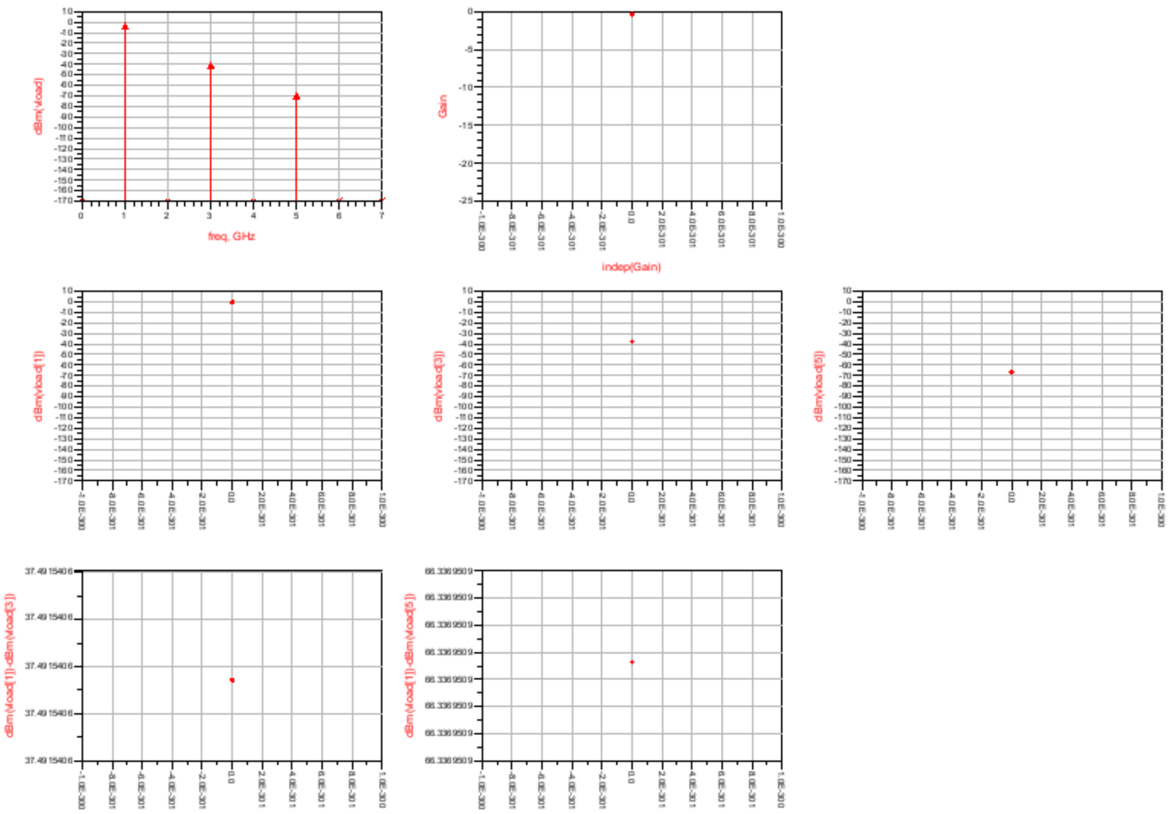


Figure 16. Harmonic contents

As you may already have concluded, this simulation only shows the data for one specific input power level and cannot tell what the compression or TOI are. Go back to the schematic and double-click on the 'Harmonic Balance' control box, and go to the 'Sweep' tab.

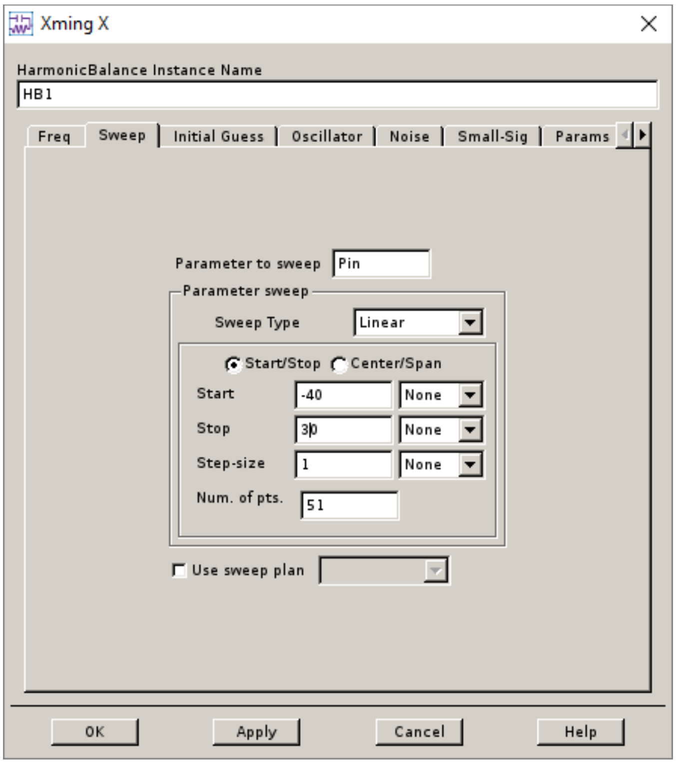


Figure 17. Input power (Pin) sweep

In order to sweep a parameter, you type the name of the parameter in the 'Parameter to sweep' box. You also need to specify the start, stop, and step values in the parameter sweep. A typical setup is shown in Figure 17. Run the simulation.

In the first plot you can see that all harmonics are plotted on top of each other, but in the other plots the fundamental component, third component, fifth component, 'dBc', and 'Gain' are plotted for each input power level.

Note that this is power gain, which means that a gain of 0 dB (1 in the linear scale).

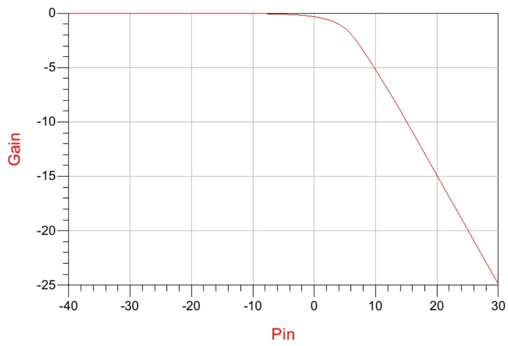


Figure 18. Gain for fundamental component

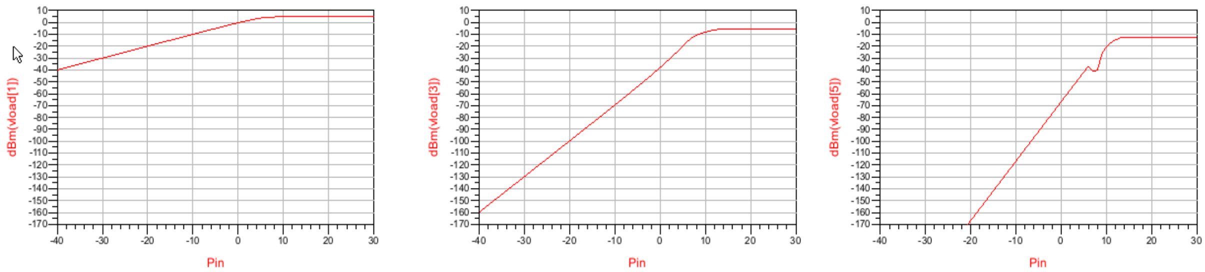


Figure 19. Plots of fundamental, third, and fifth component

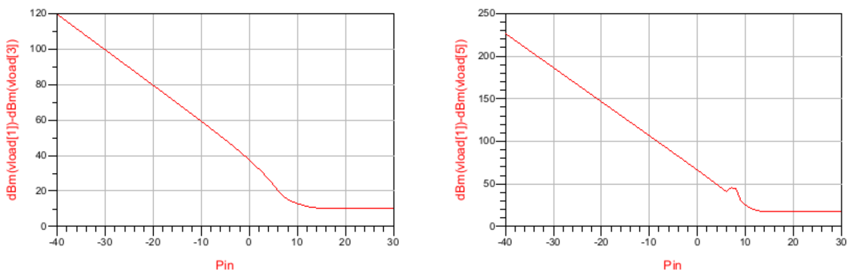


Figure 20. Difference between fundamental and third/fifth components

For clarity you can change the y-axis for the harmonic components, or you can add marker to the plots as in the picture below. The markers are found in the 'Marker' menu in the DDW. Go the 'Marker' menu, choose 'New…' and click on the desired curve to place a marker.

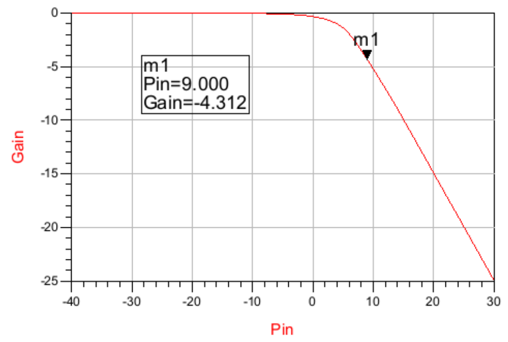


Figure 21. Marker

Exercise 4.2b: Adjust the y-axes scale for fundamental component. Add a new marker to the plot. What is the output power for input power levels of -40, 4, and 30 dBm? Can you find the 1 dB compression point?

4.3 Two-Tone Test of Amplifier

In this section we will compute the IP3 based on the harmonic contents of the signal at the load (see [1], p. 21). Save the schematic **'Lab_HB'** as a new design called **'Lab_TOI'**. The IP3 is referred here to as the Third Order Intercept (TOI) Point. The measured TOI of an amplifier is dependent mainly on the two-tone signal power but to some extent also on the spacing between the two tones that are used in the simulations. For this reason, it is important to simulate an amplifier under the same conditions in any described target specification.

Replace the 'P_1Tone' source with a 'P_nTone', available in the category 'Source-Freq Domain'. Set its 'Num' parameter to 1. The two tones should be very closely spaced relative to the operating frequency. In this example, the spacing can be defined by 'fspacing', and set to 100 MHz centered around 1 GHz. For the 'P_nTone' source, change the following variables:

Value (Old)	**Value (New)**
`Freq[1]=1.0 GHz`	`Freq[1]=(RFfreq+fspacing/2) GHz`
`P[1]=polar(dbmtow(0),0)`	`P[1]=polar(dbmtow(Pin),0)`

The value 'Pin' denotes the input power in dBm and should be set to a low value of typically -30 in order not to push the amplifier into compression. Double-click on the source and click on 'Freq[1]' and 'Add'. Similarly, click on 'P[1]' and 'Add'. Another 'Freq' and 'P' components are added and labeled 'Freq[2]' and 'P[2]', and change the component values to:

Value (Old)	**Value (New)**
`Freq[2]=1.0 GHz`	`Freq[2]=(RFfreq-fspacing/2) GHz`
`P[2]=polar(dbmtow(0),0)`	`P[2]=polar(dbmtow(Pin),0)`

Add the following variable to a variable box:

Var	**Value**
`fspacing`	`0.1`

Also set Pin to -30. In order to capture the intermodulation products in the simulation, we need to change the harmonic balance setup. Double-click on the 'Harmonic Balance' controller and click on the first listed frequency entry and set the 'Frequency' to 'RFfreq+fspacing/2 GHz' with the 'Order' set to 'nOrder'. Click 'Add' and change the setting of the second frequency entry to 'RFfreq-fspacing/2 GHz' with the 'Order' set to 'nOrder'. See Figure 22. On the 'Sweep' tab, delete 'Pin' as 'Parameter to sweep'. Close the 'Harmonic Balance' controller.

LINKÖPING UNIVERSITY

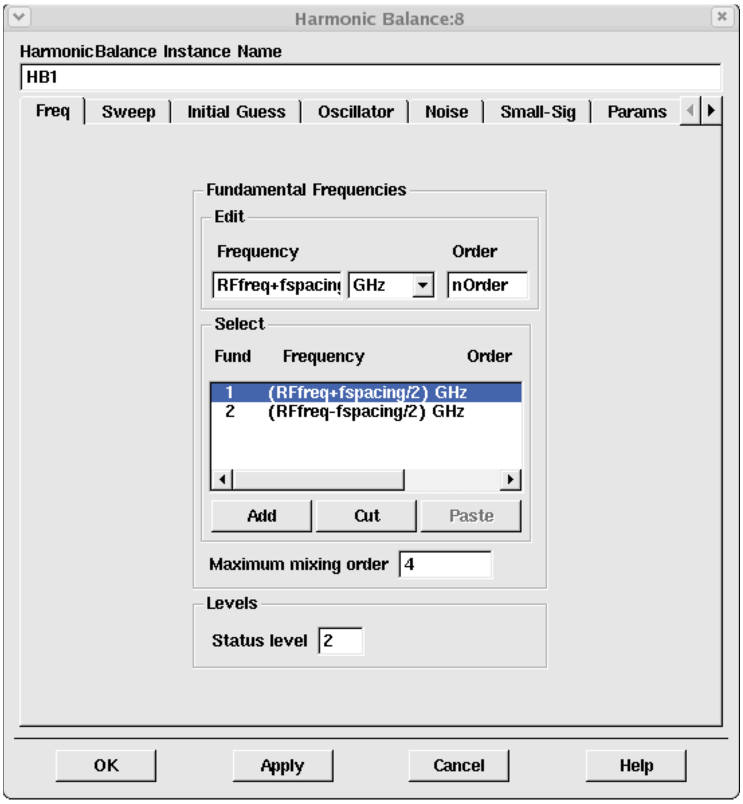


Figure 22. HB controller setup

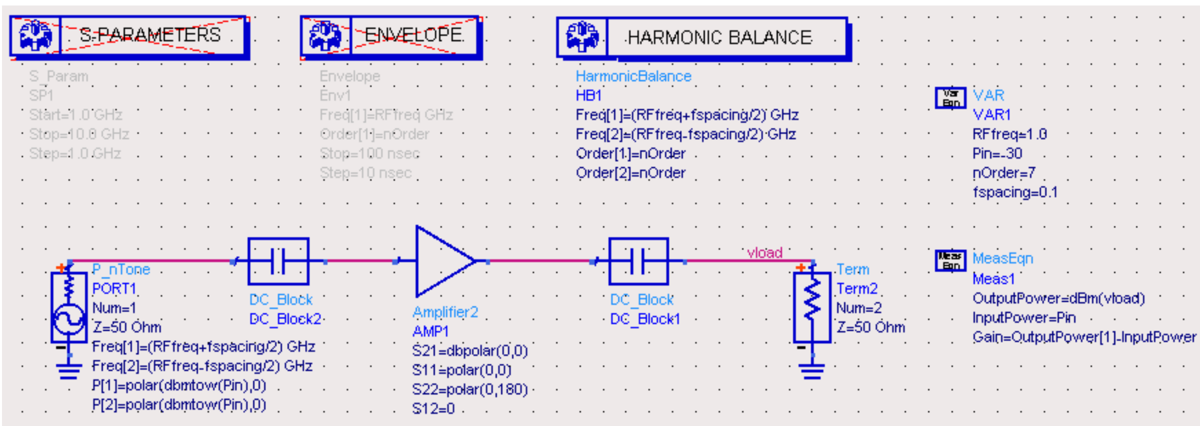


Figure 23. Two tone testbench

4.3.1 Computation of IP3

Click on the simulation button. When the results window pops up, plot the intermodulation terms by plotting 'dBm(vload)'. Add two markers in the plot. One marker at 1.05GHz (Freq[1]) and one marker at 1.15GHz (2*Freq[1]-Freq[2]). The results should be as in Figure 24.

Observe that the distance between the markers is IM3 (ΔP) equal to 90 dB. Hence, IIP3 = $P_{in}+\Delta P/2$ = -30 dBm+90 dB/2 = +15 dBm as defined in the model.

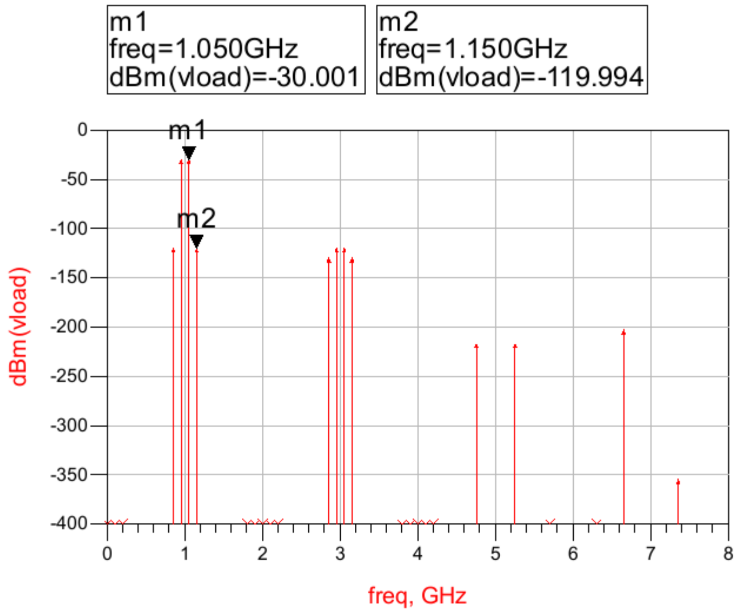


Figure 24. Two tone test simulation results

To see the ADS calculation capabilities, add two equations. The first equation computes the TOI from the markers. The second equation is based on the IP3 built-in function in ADS and refers to the fundamental and intermodulation tones.

Press the equation button, 

, in 'Palette' list in the DDW, and enter the following expression for the first equation:

```
TOI_spectral=m1+(m1-m2)/2
```

Add the second equation to calculate IP3 in different way:

```
TOI_builtin=ip3_out(vload,{1,0},{2,-1},50,Mix)
```

Add a new list, , and add the two following expressions to be printed (select the Equations section in the Datasets and Equations:

```
TOI_spectral[0]
TOI_builtin[0]
```

Exercise 4.3.1: If the simulation has been performed correctly, the IP3 is equal to TOI specified for the amplifier. Verify! Also verify that the first equation is equal to the well-known formula for IIP3 since the gain is 0 dB.

$P_{in} + \Delta P/2 = P_{in} + (P_{out}-P_{IM3})/2$

4.3.2 Computation of IP2

As you could see in the plot of the two-tone simulation results, no second-order intermodulation terms were generated. To enable the generation of the second-order terms, set 'SOI' to 50 and **delete the settings for 'GainCompPower'. Re-run the simulation and observe the strength** of the second-order intermodulation terms.

Exercise 4.3.2: If the simulation has been performed correctly, the IIP2 is equal to SOI specified for the amplifier. Verify!

$IIP2 = P_{in} + (P_{out}-P_{IM3})$

---

5 S-parameter Simulation

This section requires basic knowledge about S-parameters, see [1], p. 71.

In the previous simulation we have assumed that there is a perfect match between the output port and input port of the following block. In the S-parameter simulation we will adjust the input and output impedances of the amplifier to create some mismatch, and then create matching networks in order to correct this mismatch and achieve good amplification.

Save the original schematic **'Lab_HB'** as a new design, typically **'Lab_SP'**. Click on the HB control box and disable it by clicking the 'disable' button, , in the toolbar. Then click on the SP control box and enable this control by clicking on the 'disable' button. Consequently, the disable button can be used for both disabling and enabling components, not only control boxes but also e.g. resistors and signal sources.

Add five variables to a 'Var' box (we intentionally introduce impedance mismatch knowing that Zs = 50 Ohm):

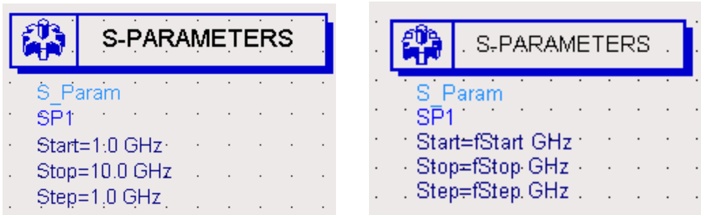
Name	**Value**
`Zin`	`100`
`Zout`	`25`
`fStart`	`0.1`
`fStop`	`1.9`
`fStep`	`0.025`

Double-click on the amplifier symbol and set the parameters as follows:

Name	**Value**
`Z1`	`Zin`
`Z2`	`Zout`

By setting these impedance parameters it means that the amplifier will relate its S-parameters to these impedance values.

In the S-parameters control box change:



a) **b)**

Figure 25. S-parameters control box, a) default values, b) modifications used for simulations

Use 'P_1Tone' as the driving source with the same settings as in chapter 3.

Exercise: Run the simulation.

In the simulation status window, the following message will be displayed:

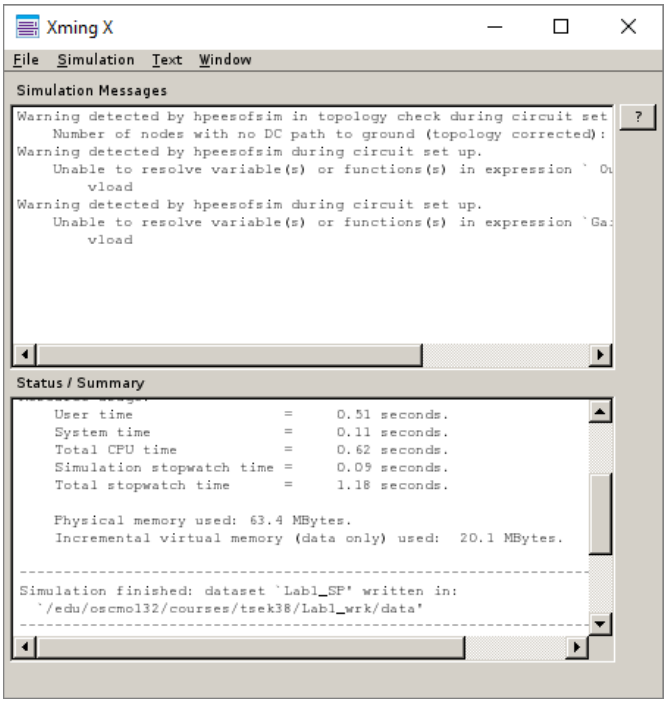


Figure 26. Error message

The warning message relates to the measurement equations defined previously and is based on the HB simulation, but since the HB simulation is deactivated the expressions are not valid and a warning message is displayed. For the S-parameter simulation there is no need to change anything, but the message will disappear if the 'MeasEqn' box is disabled.

5.1 Plot of S-parameters

As before an empty new DDW is displayed on the screen. Click on the plot symbol, . In the 'Available data' section in Figure 27, several new data items are available.

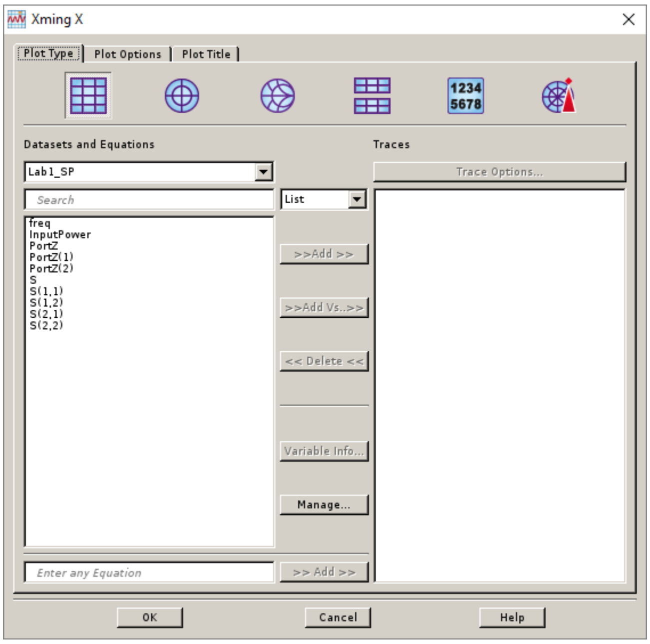


Figure 27. S-parameter simulation data

The port numbers 1 and 2 in the S-parameter simulation data relate to the input port number, 'Num=1', and the output termination number, 'Num=2'. Click on 'S(1,1)' and then on the buttons '>>Add>>'. In the radio button windows that pops-up, select to plot the data in 'dB' (<u>NOT dBm</u>) and then click 'OK'. Plot 'S(2,1)' in a similar way, either in the same plot or using new plots (). If the data is plotted correctly you should now have the following data plots:

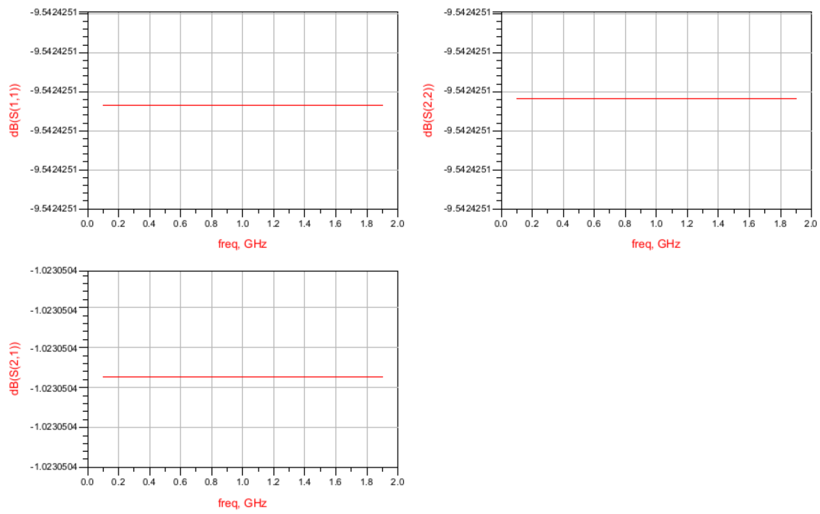


Figure 28. S-parameters for the unmatched amplifier

Observe that the result of S_{11} can be verified using the well-known formula: $S_{11} = (Z_{in} - Z_s)/(Z_{in}+Z^s) = (100-50)/(100+50) = 1/3$ and $20\log(1/3) = -9.6$ dB. Similarly, $S_{22} = (Z_{out} - Z_L)/(Z_{out}+Z_L) = -1/3$.

As seen in Figure 28, 'S(1,1)' and 'S(2,2)' are not perfectly matched since they are only -9.5 dB. Also 'S(2,1)' is only -1.02 dB which means that the gain of the amplifier is lower than when the amplifier was perfectly matched as in the HB simulation.

We can also see that the behavior of the amplifier is independent of frequency which stems from the fact that we don't have any frequency-dependent matching components, e.g. inductors and capacitors.

5.2 LC matching networks

Since the impedance of the signal generator, 'PORT1', is lower than the input impedance of the amplifier, a simple LC matching network suitable for narrowband operation can be applied. For simplicity we assume no losses in the matching components.

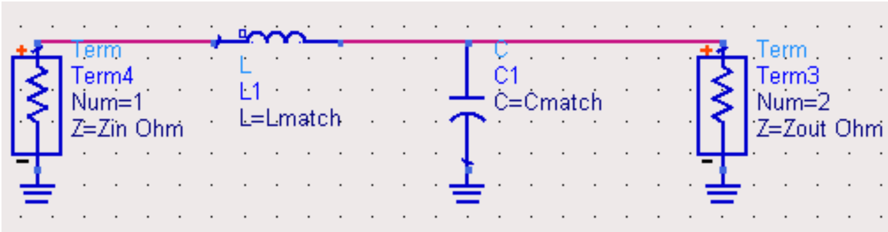


Figure 29. Typical LC matching network

For a general LC-matching network as in Figure 29, the impedance seen at the input port, 'Num=1', should be equal to 'Zin'. In this setup at the resonance frequency ω_0 the load impedance, 'Zout', is transformed to a lower value equal to Zout/Q^2 where $Q = \omega_0 C_{match}$ Zout and $\omega_0 = (LC)^{-1/2}$. Using this matching network, the input impedance of the amplifier (100 Ohm) should be transformed to 50 Ohm of the source (input port). Similarly, the load impedance (50 Ohm) should be transformed to the output impedance of the amplifier, which is 25 Ohm.

5.3 Adding matching components

Set the operating frequency, 'RFfreq', to 1 GHz (then $\omega_0 = 2\pi \times 1$ GHz). Add the matching components (capacitors and inductors) to the schematic as in Figure 30, and update all other simulation components. Add the matching component name to the 'Var' box and fill in the right component values (Cmatch1, Cmatch2, Lmatch1, and Lmatch2) as 1.519, 3.091, 7.461, and 3.963. Note that the units, pF and nH are already filled in for these components.

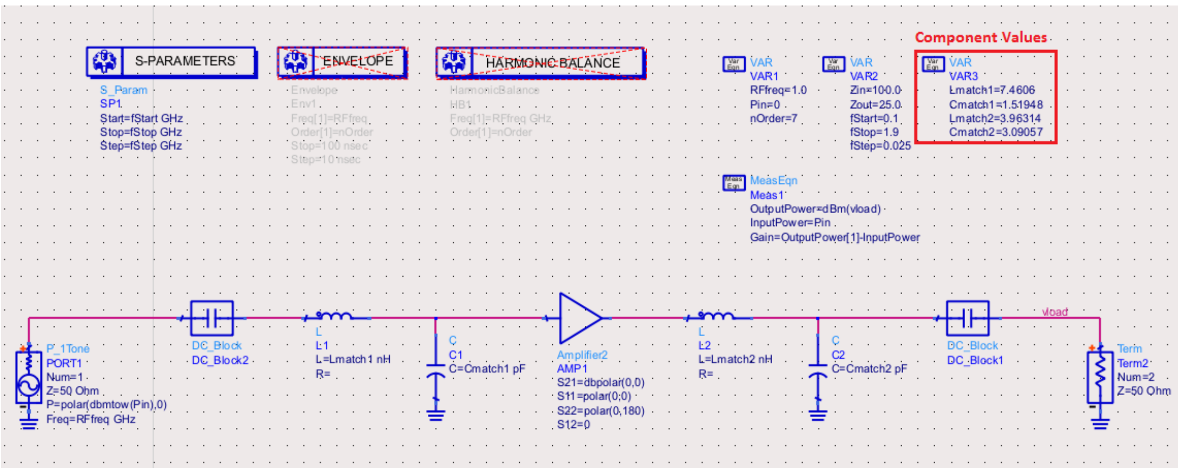


Figure 30. Schematic with matching networks

When you have filled in the component values, run the simulation. The following S-parameter plots appear. Add markers to the plots and put the marker at 1 GHz.

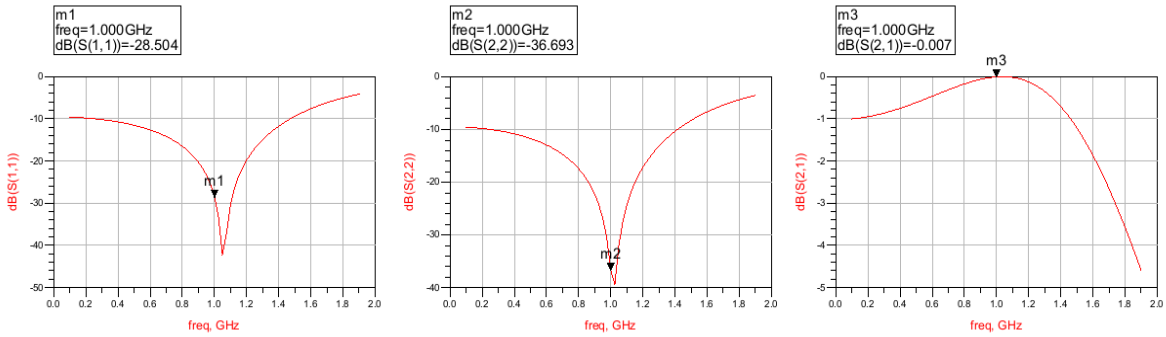


Figure 31. Well-matched amplifier

The values at the markers may differ depending on the number of decimals and frequency step used, but we can conclude that the amplifier is matched at the input and output and therefore the gain is also 0 dB. Instead of calculating the component values in the matching networks by hand, you can also let the software optimize the values. However, first we will run another simulation mode.

5.4 Using matched schematic for HB

To see another advantage of the matching circuits, disable the SP control box, and enable the HB control box. Next, restore the 'GainCompPower' of the amplifier as in section 4.2. Make sure that only a single tone is simulated, with parameters for Harmonic Balance controller as in Figure 9 and then run the simulation.
As seen in Figure 32, there is no SP data available. Instead of adding more plots to the current DDW, we can open the previously created DDW where we have plots for HB simulations. To open the DDW window for HB simulations, go to the schematic, then go to the menu 'Window', and then select 'Open Data Display...'. In the file browser, select **'Lab_HB.dds'**.

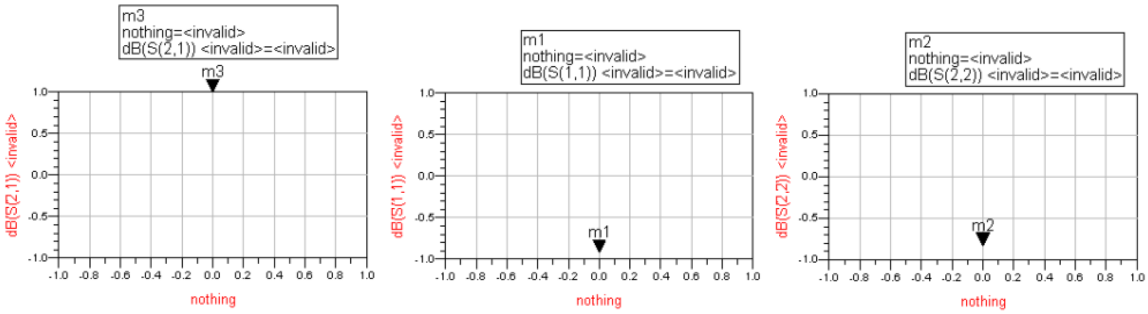


Figure 32. No SP simulation data available

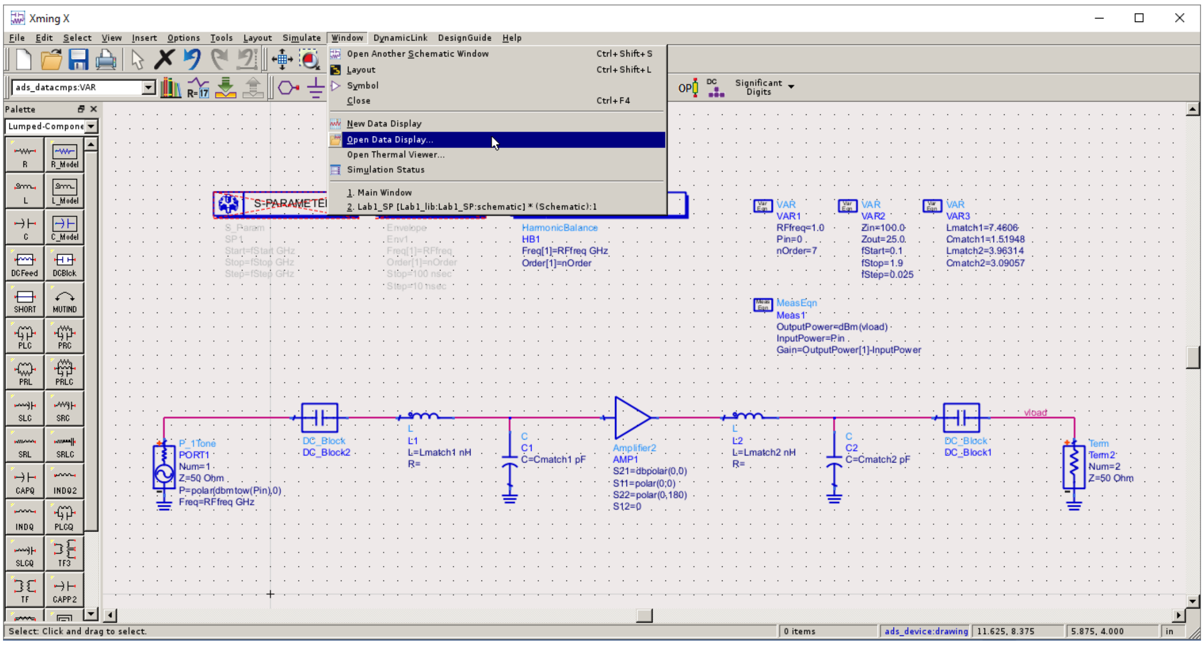


Figure 33. Open the HB data display window

In the top middle of the plot window, there is a scroll-down box. Here you select the data set you want to plot. Switch to 'Lab_SP' to show the data from this simulation.

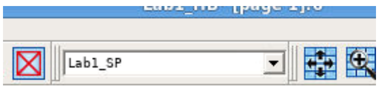


Figure 34. Data Set selection in the plot window

Exercise 5.4: Sweep the input power like in section 4.2. (*Figure 17*). Plot gain at the fundamental component (1 GHz) and power of the harmonics. Compare with previous results and explain the difference.

Go back to the **'Lab_SP'** schematic window. Disable the HB control box and once again, enable the SP control box.

5.5 Optimization of matching networks

In case of more complicated networks including parasitic inductances, resistances, and capacitances, it might be hard to calculate the matching networks components by hand. Even if the matching problem in this example is relatively simple, it is instructive to let ADS automatically calculate the component values.

Select 'Optim/Stat/Yield/DOE' from the 'Category' drop-down list. Place an 'Optim' part from the palette in the schematic. Place three 'Goal' in the schematic. A 'Goal' part is found next to the 'Optim' part.

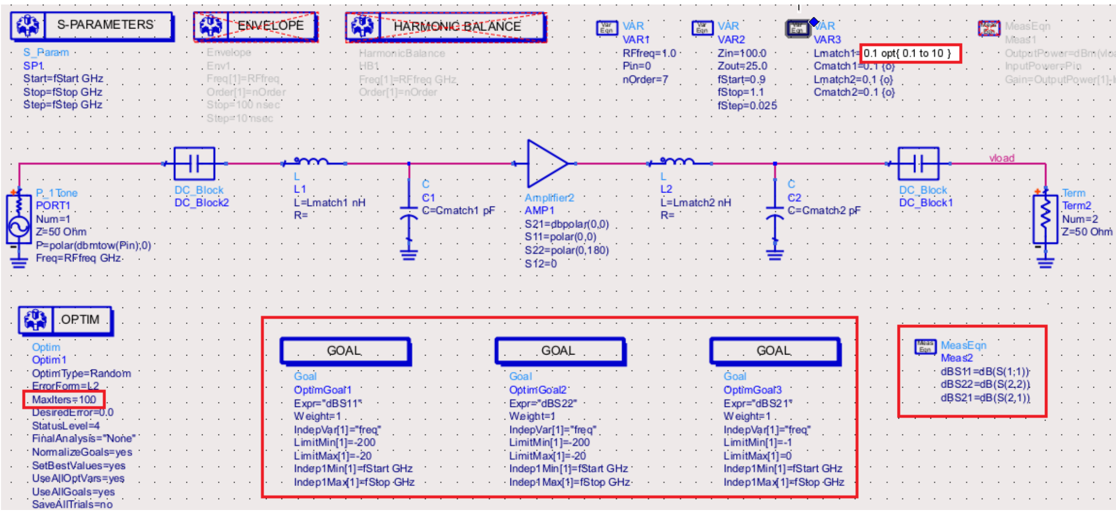


Figure 35. Optimization of component values

In the 'Optim' component, adjust 'MaxIters' to 100 or higher value.
To optimize certain component values, you need to specify the boundaries for the component values. For simplicity set all optimization boundaries as for Lmatch1 in Figure 35, '0.1 opt{0.1 to 10}'. It means that all inductors can be as small and as large as 0.1 nH and 10 nH, respectively. Similarly, 0.1 pF and 10 pF for the capacitors.
Add a new (or copy the previous) simulation measurement equations, 'MeasEqn', box and add the items as in Figure 35.
In the S-parameters control box, set 'fStop' and 'fStart' to 1.1 and 0.9, respectively.
Also modify the three optimization goals according to Figure 35. Note that all values for S_{11}, S_{22}, and S_{21} are measured in dB. Since we would like to optimize the design around 1 GHz, we tighten the frequency range in the optimization by setting 'fStart' and 'fStop' to 0.9 and 1.1, respectively.

When clicking the optimization button , optimization will be performed for an SP simulation with the constraints that S_{11} and S_{22} must be lower than -20 dB, and that S_{21} must be larger than -1 dB. After the simulation we get the following SP plots of S_{11}, S_{22}, and S_{21}. After the optimization, set the component values to the optimized values by accessing the 'Simulate' menu in the schematic window and select 'Update Optimization Values'.

Note that the exact numbers and simulated performance (as shown below) may differ, due to software versions and other number of iterations in the optimization process have been chosen in the simulation. If you want to fit the data in the window, click in the specific, and then click on the data fit button . If you are not satisfied with the scale on the x and y axis, double-click on the window and go to the 'Plot options' tab. Here you can manually set the scales.

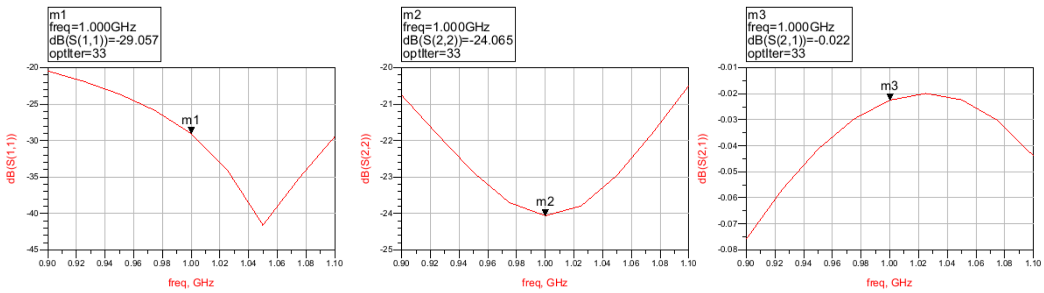


Figure 36. Optimized S-parameters

It is interesting the follow the optimization. Set MaxIters to e.g. 1000 and the Optimization Type to Random MiniMax. Then observe the 'Optimization Cockpit' window where you can follow and control the optimization process.

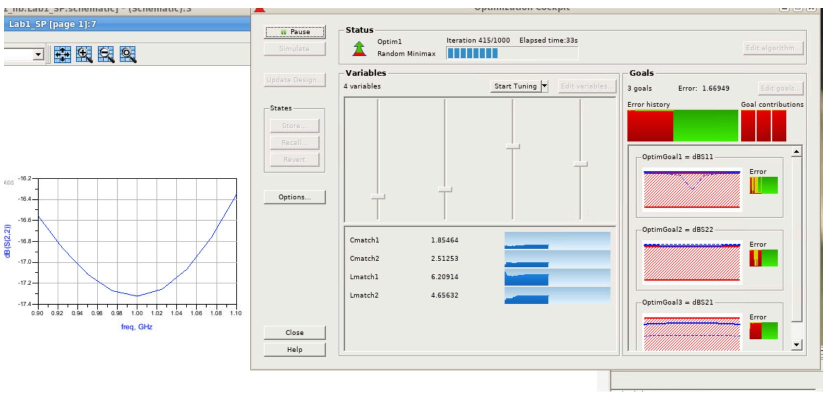


Figure 37. The Optimization Cockpit window

6 Envelope Simulation

6.1 EDGE Signal Analysis

As already described, Envelope Simulation can be used for analyzing digitally modulated RF signals and perform a combination of time and frequency representation of the signal. Briefly, this simulator permits input waveforms to be represented in the frequency domain as RF carriers, with modulation "envelopes" that are represented in the time domain as shown in Figure 38.

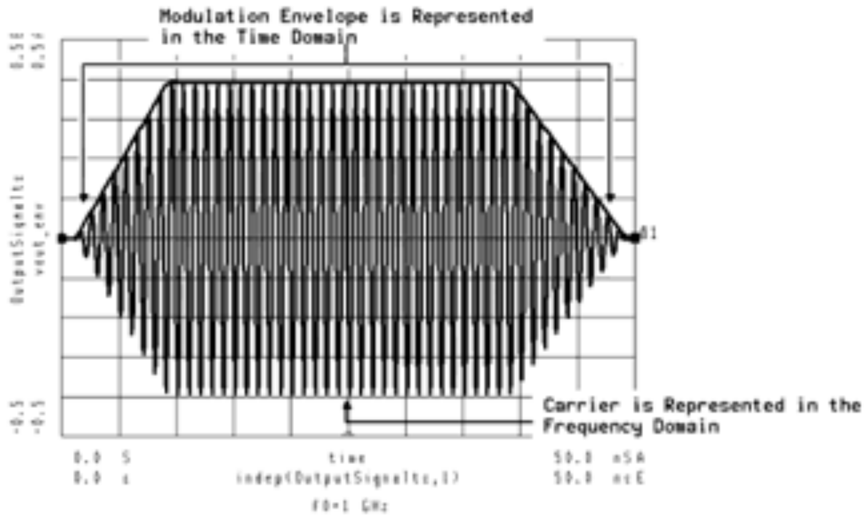


Figure 38. Envelope analysis

To describe the circuit envelope simulation process more specifically, in an envelope simulation each node voltage is represented by a discrete spectrum having time-varying Fourier coefficients. The set of spectral frequencies is user-defined; the amplitude and phase at each spectral frequency can vary with time, so the signal representing the harmonic is no longer limited to a constant, as it is with harmonic balance. Each spectral frequency can be thought of as the center frequency of a spectrum; the width of each spectrum is ±0.5/*Time step*. Figure 39 illustrates this, where the minimum envelope bandwidth is equal to the bandwidth of the modulation signal. In most cases the bandwidth of the modulation signal is much smaller than the lowest user-defined spectral frequency (which corresponds to the "carrier" frequency), unlike what is shown in the figure.

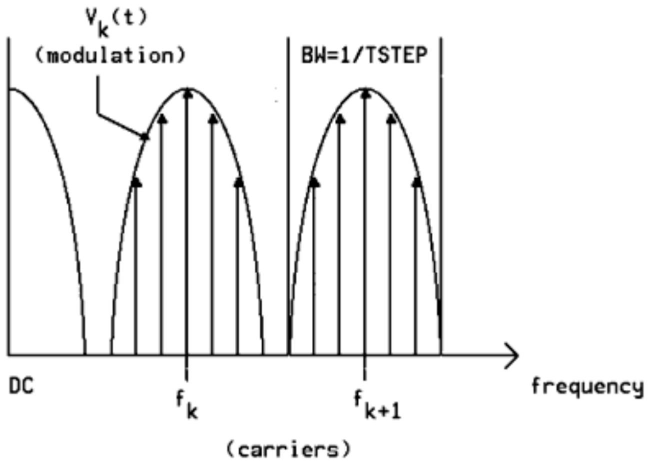


Figure 39. Spectrum in frequency domain.

The bandlimited signal within each spectrum can contain periodic, transient, or random tones. The actual time-domain waveform is represented as a sum of carriers (with harmonics and intermodulation products), where each envelope can vary with time,

$$v(t) = \text{real}\left[\sum_{k=0}^{N} V_k(t) e^{j2\pi f_k t}\right]$$

where v(t) is a voltage at any node in the circuit, including the input. The Fourier coefficients, $V_k(t)$, are allowed to vary with time and may represent an arbitrary modulation of each carrier. Since each time-varying spectrum $V_k(t)$ can be thought of as a modulation waveform with a center frequency f_k, these are often referred to as "envelopes." This spectrum may represent transient signals with continuous spectra, such as a digital modulation envelope over an RF carrier, or periodic signals with discrete spectral lines, such as the two RF tones required for intermodulation distortion analysis.

The following figure illustrates a modulated signal and the time-varying spectrum that results from the simulation. Any spectral component obtained from the simulation can be processed and displayed in amplitude or phase, I or Q. By computing the Fourier transform of the spectral component, the simulator can present the spectrum around the component, as in a spectrum analyzer display.

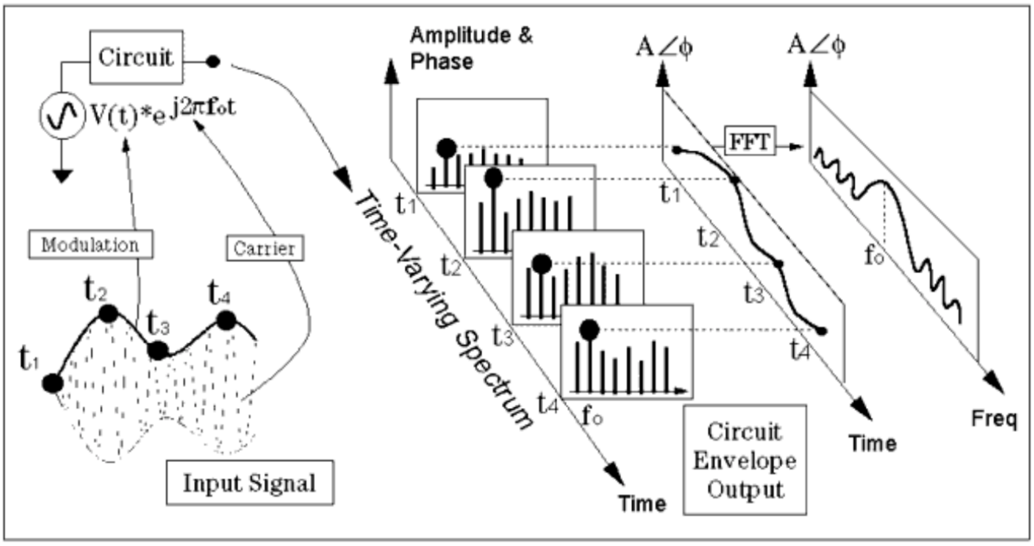


Figure 40. Modulated signal and its simulated time-varying spectrum.

In this part of the lab, we will use an EDGE signal generator (GSM with amplitude and phase modulation, also known as 2.5G mobile communication) and feed it into the amplifier. The data file of the generator contains 1 TDMA frame (120/26 msec) of EDGE data (1250 symbols at 48/13 usec per symbol). One EDGE frame contains 8 time slots with each time slot containing 156.25 symbols. The EDGE frame generated by this source contains data (normal burst with 8PSK modulation) in the second time slot, all other seven time slots are idle (no signal). This frame represents one active user in the EDGE uplink. Due to the modulation of EDGE signals, we have both amplitude and phase modulation of the RF signals, and a peak-to-average-power ratio (PAPR) of approximately 3.5 dB. We will examine what happens to the frequency spectrum and how the trajectory diagram (related to I and Q) signals look like after passing through the amplifier.

6.1.1 Simulation setup

Create a new schematic based on the SP simulation setup. Go to the 'File' menu and select 'Save Design As…'. Name it **'Lab_ENV'**. The EDGE signal generator can be found in 'Sources-Modulated'. Place one part called 'EDGE Uplink' ('PtRF_EDGE_Uplink') in the schematic. Change the following parameters:

Name	**Value**
`Freq`	`RFfreq GHz`
`Power`	`dbmtow(Pin)`

Disable the previously used signal generator and connect the EDGE signal generator to the DC-blocker at the input matching network.

Add a new variable box, ‘Var’, with the following variables.

Name	**Value**
numSymbols	256
sam_per_sym	8
tstep	1/(sym_rate*sam_per_sym)
tstop	numSymbols/sym_rate
sym_rate	270.8333 kHz

Disable the SP, HB, Goals, Optim control boxes, and enable the ENV control box. Add a new variable to a ‘MeasEqn’ box and name it ‘VloadFund’ and set it to ‘vload[1]’.

Set the parameters of the ENV control box accordingly:

Name	**Value**
Freq[1]	RFfreq GHz
Order[1]	nOrder
Stop	2*tstop0
Step	tstep

Run the simulation to open a new DDW for an input power, ‘Pin’, of -10. As for all other simulation modes examined before, the DDW is empty.

Click on the equation button in the palette 

in Figure 11 and place the equation in the DDW. Equation boxes can be used to compute any combination of simulation data, and in our case, we would like to normalize the constellation diagram of the received EDGE signal. Enter the equation in the equation field, as in Figure 41:
`Vn = VloadFund/abs(max(VloadFund)).`

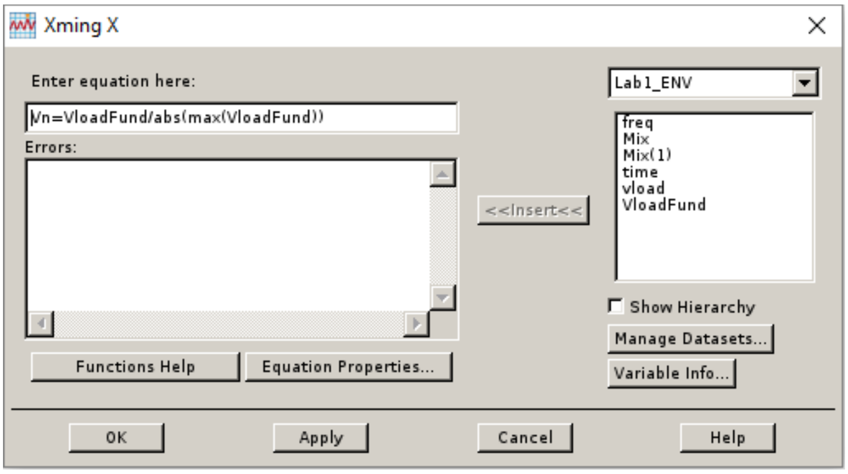


Figure 41. Normalization of the received signal

As ‘VloadFund’ contains the amplitude and phase information of the received signal, we now normalize the received signal to the peak amplitude, which is 1 in the constellation diagram.

6.1.2 Trajectory diagram and spectral components

To plot the trajectory diagram of the received signal, which can be related to I (real) and Q (imaginary/phase) components, we click on the plot button, . Type the expression `plot_vs(imag(Vn),real(Vn))` in the blank space “Enter any equation” as in Figure 42 and then click Add >>.

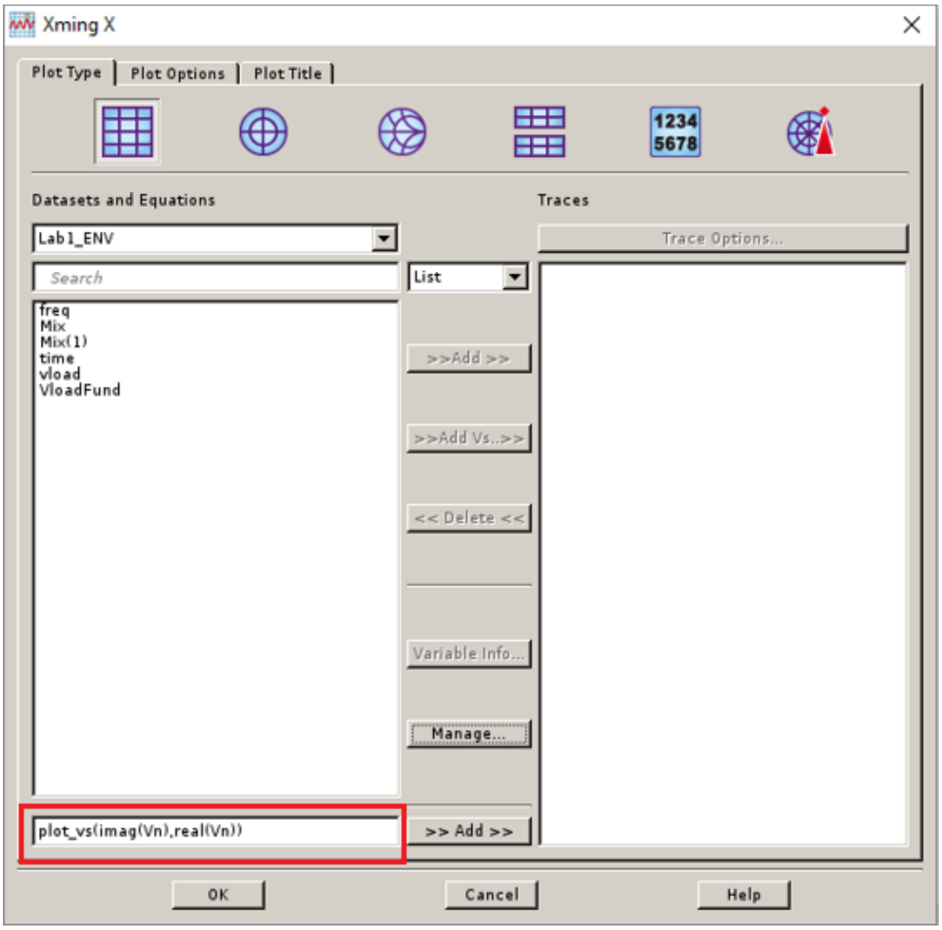


Figure 42. Trajectory diagram

It is important to check the spectral content, so that radio transmissions in the adjacent channels are not disturb. To plot the Fourier spectrum, add a new equation as in Figure 41, but this time the expression should be `Spectrum=dBm(fs(VloadFund,,,,,5))` . (The number 5 is just the filter used while displaying the data, see "Measurement Expressions" and "Filters for System Models" in the ADS Help.)
Add a new plot to the DDW and add 'Spectrum' to the traces. Finally, you should have a DDW with the following contents as below.

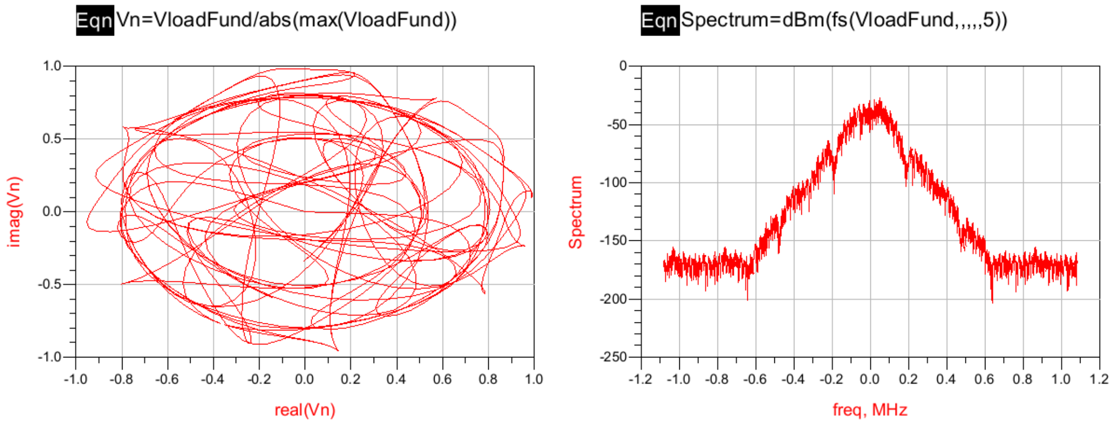


Figure 43. Trajectory diagram and spectral contents of the amplifier for an input power of -10 dBm

Adjust the input power to 10 dBm. Run the simulation again. The resulting plots can be seen below.

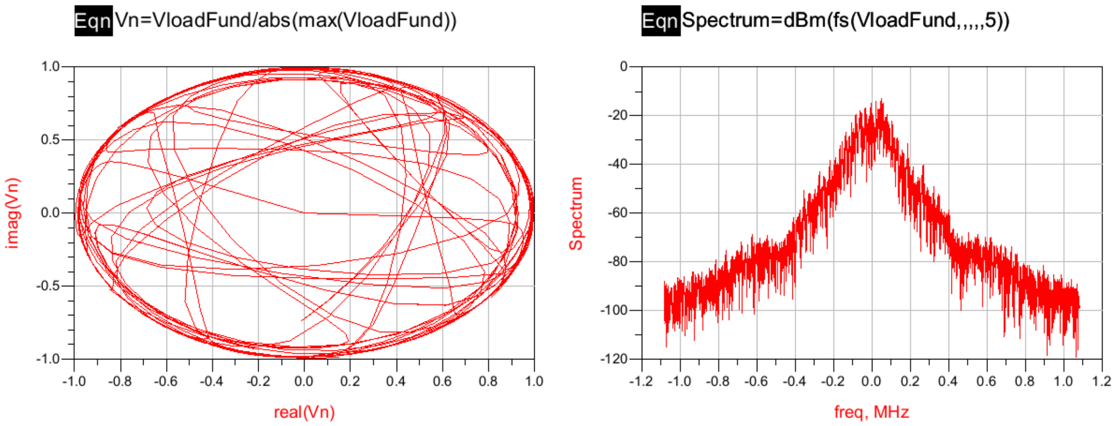


Figure 44. Trajectory diagram and spectral contents of the amplifier for an input power of +10 dBm

Exercise 6.1.2: Can you explain the differences in the simulated spectrum, and the trajectory diagram? Hint: plot the absolute value of the time domain signal of 'vload', where you will also see that there is no constant signal transmission, the signal is bursted. Comparing the minimum and maximum amplitudes of the signal for both cases also give a hint about what has happened.

6.2 Noise and Noise Figure simulation using Envelope Analysis

All real systems are not as ideal as the presented ones, since more realistic systems are influenced by noise, which is particularly important in low-noise amplifiers (LNA). The following simulation will evaluate the amplifier in terms of noise.

Copy the schematic **'Lab_ENV'** to **'Lab_NOISE'** by using 'Save Design As...'.
Double-click on the wire at the source port and name it 'vsource'.
Disable the EDGE-port and enable the 'P_1Tone' which has been disabled.
Double-click on the source port and set 'Noise' to 'yes' if it is not already set, also check the box 'Display parameter on schematic'.
Double-click on the load port and set 'Noise' to 'no', also check the box 'Display parameter on schematic'.
Add a new measurement equation called 'VsourceFund' and set it to 'vsource[1]'..
In the ENVELOPE controller, go to tab 'Env_Params' and check 'Turn on all noise'.
Double-click on the amplifier in the schematic and set the Noise Figure (NF) of the amplifier field to 'NFuser'.
Add a new variable to the variable box with the name 'NFuser' and value 10.
Add a 'ParamSweep' controller box from the Simulation-Envelope palette to the schematic. Set 'SweepVar' to "Pin" (including the "), 'SimInstanceName[1]' to "Env1" (including the "), 'Start' to '-300' (no signal in practice), 'Stop' to '-50', with a total number of points set to 2 (step size 250).

Since we are running an envelope simulation, the noise is limited to the envelope bandwidth as 1/(tstep). To set the temperature to the desired level, an 'Options' controller is needed. Write 'Options' in the 'Part' drop-down list and place the controller in the schematic. Set 'Temp' and 'Tnom' to 25.

Run the simulation with the specific parameter sweep. Two simulations will be performed, one simulation with the input power set to -300 dBm (with index 0), and one simulation with -50 dBm (index 1). After the simulation, a new empty data display window pops up.
In a similar way as before, we also need to fill the data display with some useful handling controllers, such as equations and plots. Initially, in DDW add three equations with the following expressions:

```
time_vals=indep(VloadFund[0,::])        //return the independent
                                        //attached to the data
time_step=time_vals[1]
EnvBandwidth=1/time_step
```

The equations bring out the envelope bandwidth from the simulation, which is needed to be able to compute the noise power levels at the input and output correctly and make the computation independent on the time step used in the simulation.

To compute SNR and NF of the circuit, we start by considering the input. Add the following equations to compute the signal power at the input:

```
VspecIn=fs(VsourceFund[1,::],,,,"Kaiser")    //collected from '1'
                                             //sweep for -50 dBm
PspecIn=(VspecIn**2)/100                     //power calculation
SignalInW=mag(max(PspecIn))                  //signal picked up by
                                             //max function
```

To compute the noise level at the input as power per Hz and the SNR, add the following equations:

```
NoiseInW=mean(mag(VsourceFund[0,::])**2/100)     //collected from '0'
                                                 //sweep for -300 dBm
NoiseInW_per_Hz_dBm=10*log(mean(NoiseInW/1e-3))-
10*log(EnvBandwidth)

SNRin_dB=10*log(SignalInW)-NoiseInW_per_Hz_dBm
```

Alternatively,

```
SNRin_dB=10*log(SignalInW/NoiseInW)
```

Add the same equations for the output by replacing ”In” by ”Out” and ”VsourceFund” by ”VloadFund”. Then the Noise Figure (NF) can be computed as the difference in SNR at the input and output.

```
NF=SNRin_dB-SNRout_dB
```

Add the following expressions to a list box, , to compute the input power in dBm, the output in dBm, and the noise power in dBm per Hz at the input and output.

```
10*log(SignalInW/1e-3)
10*log(SignalOutW/1e-3)
NoiseInW_per_Hz_dBm
NoiseOutW_per_Hz_dBm
```

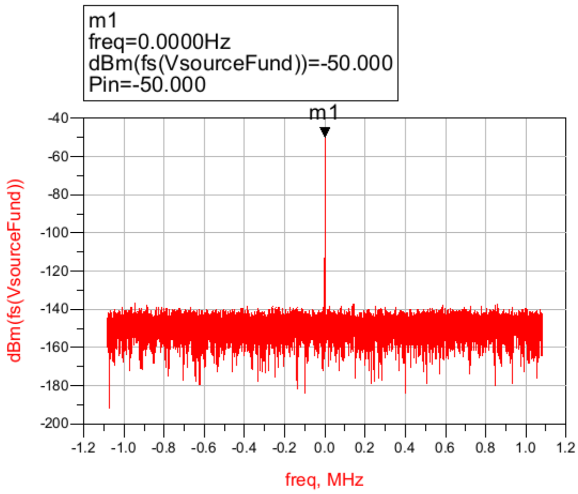
If everything is set up correctly, the noise power is about -174 dBm/Hz at the input, which is the reference noise spectral density, also calculated as:

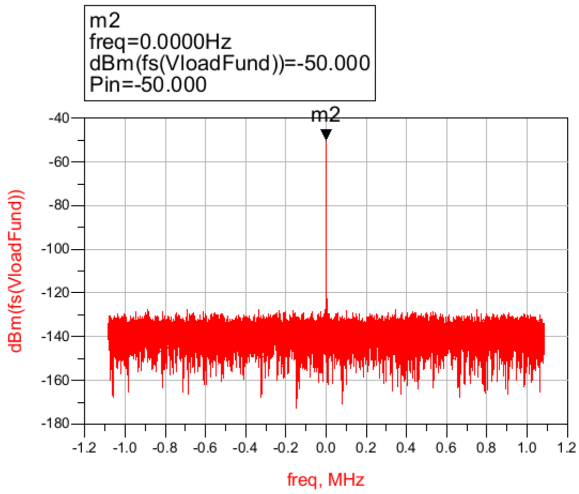
$$10\log(kT_0/1\ \mathrm{mW}) = 10\log(1.38\cdot 10^{-23}\times 290\times 1000) = 10\log(4\cdot 10^{-18}) = -174\ \mathrm{dBm/Hz}.$$

Plot the spectrum at the input and outputs in two rectangular plots by plotting the following expressions:

```
dBm(fs(VsourceFund))
dBm(fs(VloadFund))
```

Note that both simulations are then plotted on top of each other and if the ‘ParamSweep’ controller box is disabled, only the equations without the indexes are valid. If all equations and spectrum plots are plotted as described, the data display window should now have the contents as shown in Figure 45.





Eqn time_vals=indep(VloadFund[0,::])

Eqn time_step=time_vals[1]

Eqn EnvBandwidth=1/time_step

10*log(EnvBandwidth)
63.358

Input Signal

Eqn VspecIn=fs(VsourceFund[1,::],,,,"Kaiser")

Eqn PspecIn=(VspecIn**2)/100

Eqn SignalInW=mag(max(PspecIn))

Output Signal

Eqn VspecOut=fs(VloadFund[1,::],,,,"Kaiser")

Eqn PspecOut=(VspecOut**2)/100

Eqn SignalOutW=mag(max(PspecOut))

Noise

Eqn NoiseInW=mean(mag(VsourceFund[0,::])**2/100)

Eqn NoiseInW_per_Hz_dBm=10*log(mean(NoiseInW/1e-3))-10*log(EnvBandwidth)

Eqn NoiseOutW=mean(mag(VloadFund[0,::])**2/100)

Eqn NoiseOutW_per_Hz_dBm=10*log(mean(NoiseOutW/1e-3))-10*log(EnvBandwidth)

SNR

Eqn SNRin_dB=10*log(SignalInW)-NoiseInW_per_Hz_dBm

Eqn SNRout_dB=10*log(SignalOutW)-NoiseOutW_per_Hz_dBm

NF

Eqn NF=SNRin_dB-SNRout_dB

10*log(SignalInW/1e-3)	10*log(SignalOutW/1e-3)	NoiseInW_per_Hz_dBm	NoiseOutW_per_Hz_dBm
-50.000	-50.000	-173.949	-163.963

SNRin_dB	SNRout_dB	NF
93.949	83.963	9.986

Figure 45. Spectrum for a -50 dBm input and NF calculation

Exercise 6.2: What can be concluded about signal power levels relative the noise power at the input and output ports (look at SNR and NF)? Note that if the channel bandwidth is larger than 1 Hz (most likely), the noise power has to be integrated over the that bandwidth. Compare the noise floor in the plots with the signal spectrum plotted in dBm for the input and output signals. Play around with the 'NFuser' variable and simulate.

	Input	**Output**
Signal power:		
Noise Power / Hz:		
SNR in / out:		
NF dB:		

7 Simulation of Mixer

7.1 Mixer model

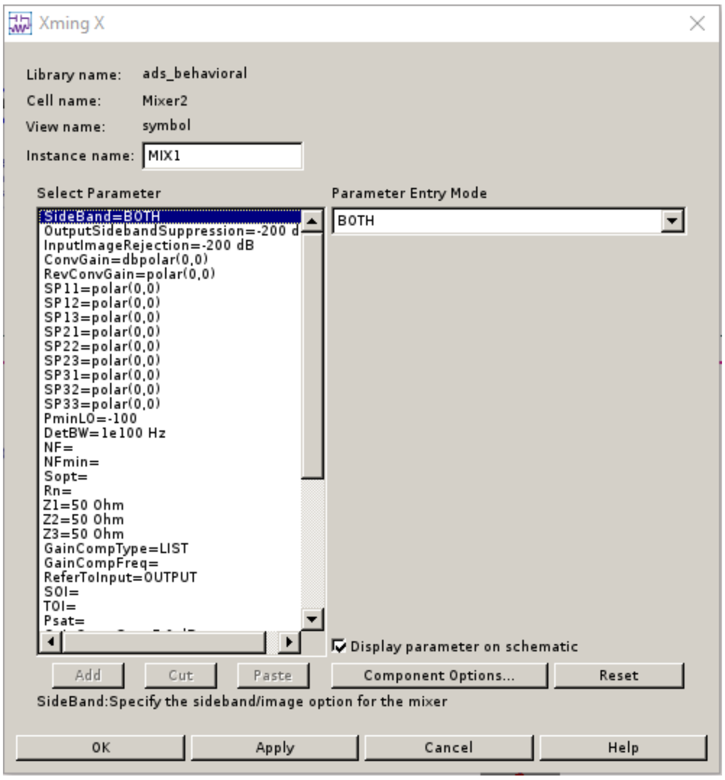


Figure 46. Mixer properties

Like the amplifier used previously, also the mixer has certain properties controlling its behavior, as shown in Figure 46. The linear behavior of Mixer2 is described by the conversion gain 'ConvGain', the reverse conversion gain 'RevConvGain', and the nine reflection/leakage/isolation parameters (S-parameters) SP_{ij} (i,j=1,2,3). The 'ConvGain' parameter is the conversion gain from RF to IF. It is applied to the lower sideband |RF-LO| and the upper sideband RF+LO. The SP_{ij} (i,j=1,2,3) parameters describe the port reflection and port-to-port leakage/isolation for the mixer. Mixer2 is a three-port device and in line with established theory for generalized S-parameters, we denote the voltages and currents at port n by v_n and i_n and define the input and output waves at each port as:

$$a_n = (v_n + Z_n i_n) / \left(2\sqrt{Z_n}\right)$$

$$b_n = (v_n + Z_n i_n) / \left(2\sqrt{Z_n}\right)$$

with Z_n being the reference impedance for port n. See the Help for more information. Amplifier2 applies compression from its input to its output. In an analogous manner, Mixer2 applies compression from RF to IF in the same way. In addition, Mixer2 also applies the same compression from RF to LO.

Under typical operating conditions, the mixer LO port is saturated. This means that the RF to IF mixing process is insensitive to amplitude fluctuations of the LO signal. A small fluctuation in LO power will not change the RF to IF mixing. In another perspective, a mixer is more of a switch than a voltage multiplier. To mimic this behavior for Mixer2, the LO is limited.

7.2 Basic simulation of mixer

In the first simulation we will not use our amplifier, but only focus on the mixer and mixing terms created in a mixer operation. After verifying the functionality, we will incorporate the amplifier in the simulation.

Create a new schematic based on **'Lab_NOISE'** and save it as **'Lab_MIXER'**. After the DC-blocker at the input, insert a 'Mixer2' and a termination after the mixer as in the figure below. Disable the matching network components located after the termination, which means that the amplifier and its input matching network do not affect the simulation.

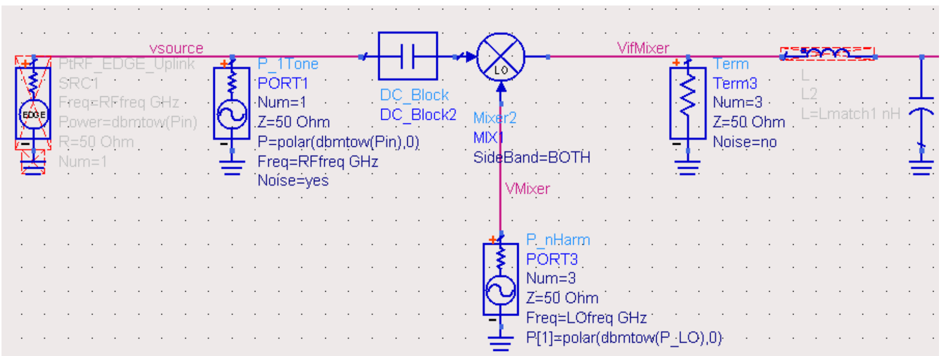


Figure 47. Insertion of mixer

Use 'P_1Tone' source at the input, with the following parameters:

`Name`	**`Value`**
`Freq`	`RFfreq GHz`
`P`	`polar(dbmtow(Pin),0)`

At the second port of the mixer, attach a 'P_nHarm' part. Set the following parameters:

Name	**Value**
`Freq`	`LOfreq GHz`
`P[1]`	`polar(dbmtow(P_LO),0)`

Two new variables are introduced, 'LOfreq' and 'P_LO'. Add these variables to the variable box and set them to 1.0 and 0.
Disable the 'Envelope', 'S-Parameters', and 'Parameter Sweep' controller boxes.
Enable the HB controller box and double-click on the controller and go to the 'Freq' tab. In the 'Frequency' field, write 'LOfreq', press Add, and now both 'LOfreq' and 'RFfreq' should be visible in the frequency list. Set 'Maximum mixing order' to 'nOrder'. Disable any sweep specified in the 'Sweep' tab.
After the mixer, label the wire 'VifMixer', and the wire between the LO and the mixer, 'VMixer'.

Set 'RFfreq' to 2.0, and 'Pin' to -50. It means that the IF frequency becomes 1 GHz, which is not a very realistic choice of IF frequency in real applications, however since our matching networks are tuned to 1 GHz, it is for demonstration purposes a convenient choice.

Exercise 7.2: Run the simulation and plot the mixing terms for 'vsource', 'VMixer', and 'VifMixer'. The generated frequencies can be plotted according to 'dBm(NETNAME[::])', where NETNAME is either 'VifMixer', 'vsource' or 'VMixer'. Write down the power level of the signal and image of the IF signal ('VifMixer').

Disable the termination port after the mixer and enable the disabled matching network components. Run the simulation again. What is the signal strength of the signal and the image the output port? Can you explain the differences in power strength between the signal and the image?

7.2.1 IF signal of 100 MHz

A more realistic choice would be an IF frequency of 100 MHz. Disable the matching network components in front of the amplifier (short-circuit the inductor and disable the capacitor) and set 'RFfreq' to 1.1 (for 1.1 GHz) and 'LOfreq' to 1.0 (for 1.0 GHz). We shall also modify the parameters for the mixer to achieve a more realistic behavior in terms of mixing terms.

Double-click on the mixer. Set the following variables:

```
Name           Value
ConvGain       dbpolar(5,0)
GainComp       1
GainCompPower  0
TOI            11
GainCompSat    - (Delete value)
DetBW          1e5
PminLO         -         //This is P_LO value at which ConvGain is
                         reached that is useful to model e.g.
                         self-mixing when a signal leaking to LO
                         port can take effect, but for now it can
                         be deleted.
```

In the mixer a new variable is introduced: conversion gain. In a mixer, two different kinds of conversion gain can be identified [1]. The gain used in ADS relates the amplitude of the signal at the RF carrier frequency to the amplitude of the downconverted signal at the IF frequency. The gain compression is defined slightly different compared to the amplifier. Here, it means that the gain is compressed by 1 dB at the output power level of 0 dBm corresponding to -4 dBm at the input. We have also adjusted the detector bandwidth to 100 MHz (see Help for further details).

Exercise 7.2.1a: Re-run the simulation of the mixer and verify that the power of the downconverted signal is compressed by 1 dB for an input signal of -4 dBm by sweeping the input power, Pin, to observe the gain of the mixer (disable matching components after the mixer). What is the saturated output power? Note: To operate the mixer at saturation makes it only possible to have constant envelope signals.
Verify that the mixer gives a gain of 5 dB by setting the input power to a low value, e.g. -50 dBm.

Exercise 7.2.1b: Run the simulation for Pin equal to -50 dBm and identify the mixing terms in the node 'VifMixer'. Also plot the spectrum of 'VMixer'. Notice the power levels (> -200 dBm) of the mixing terms and the IF and image signals. In these initial simulations the power levels of mixing terms should be very low. Also, increase the power to -4 dBm (to reach 1 dB compression point) and see what happens to the mixing terms. Note that no second-order intermodulation terms are present as this parameter is not defined in the mixer.

7.2.2 Switching mixer

In practice the LO is not a pure sinusoidal source, but rather a periodic square wave. However, as seen in the previous simulations, the output spectrum is quite clean from mixing terms. To mimic a more realistic behavior of a mixer with LO harmonics, we can insert a periodic switching source driving the LO port of the mixer (second port). Insert a source called 'Vf_Square' and disable the 'P_nHarm' source driving the LO input of the mixer.

Set the following variables of the source:

Name	**Value**
Vpeak	0.221 V
Freq	LOfreq GHz
Rise	1 psec
Fall	1 psec
Harmonics	16

Exercise 7.2.2: We have now created an LO signal with harmonic components. Simulate the testbench and plot the power of the frequency components of the node 'VMixer'. What is the power difference (in dB) between the fundamental and third harmonic, the fundamental and fifth harmonic, and the fundamental and the seventh harmonic?
Look at the spectrum of 'VifMixer'. What additional terms are found in the spectrum compared to the situation when a sinusoidal LO was used?

8 Simulation of Receiver Front-End with Quadrature Downconversion Mixer

8.1 Quadrature Downconversion Mixer and Filter

Before we proceed to the complete receiver front-end simulation, we introduce a quadrature downconversion mixer, where several simulation techniques and methods used so far will be combined. First, create a new schematic named **'Lab_QUAD'** based on the **'Lab_MIXER'** schematic. See Figure 48 for the final schematic.

8.1.1 Quadrature Downconversion Mixer Model

The circuit will be rearranged in the following manner.

- Remove the matching components and set 'Zin' and 'Zout' to 50 ohm (or 'Z1' and 'Z2') equal to 50 ohm of the amplifier, as we will not consider matching issues in this simulation. Change the gain of the amplifier to 20 dB by setting S21 equal to 'dbpolar(20,0)'. Also set 'GainComp', 'GainCompPower', and 'TOI' to '1.0 dB', '-15', and '-4'. Initially, set the noise figure variable 'NFuser dB' to '2 dB'.
- **Make sure to lower the number of symbols to 32 in order to reduce simulation time, especially since we are not using a modulated signal.**
- Add a power splitter 'PwrSplit2' after the amplifier with S21 and S31 equal to 0.7071 (power gain = 0.5) so that the power is equally split between the two paths (The splitter can also be used to add signals together). Also change the isolation to be 100000 dB, so that port 2 and 3 are isolated, otherwise the noise floor will be significantly higher.
- Add a power splitter 'PwrSplit2' after the mixer with S21 and S31 equal to 0.7071. Place the same power splitter in the path from the sinusoidal and switching mixers, it means that the power seen in 'VMixer' is present at both mixers. From the two splitter branches connect two mixers of type 'Mixer2'.
- In the LO path for one of the mixers add a 90 degrees phase shift 'PhaseShiftSML'. In the signal paths after the two mixers, add one phase shift of '0' degrees and one with '90' degrees.
- For the mixer, set the noise figure to be 'NFmixer dB'.
- For all components where the circuit parameters are set by external variables, double-click on the component and then click on the variable in the list. Check the box labeled 'Display parameter on schematic'.
- Set temperature to be 25 degrees Celsius and in order to reduce simulation time and complexity, set 'nOrder' to 5.
- Set the 'Noise' parameter of the signal generator to 'yes'.
- *When you are done placing the components the final schematic should look like the one in* Figure 48. Additionally, you can consider using a BPF in the LO path to make the LO signal cleaner as well.

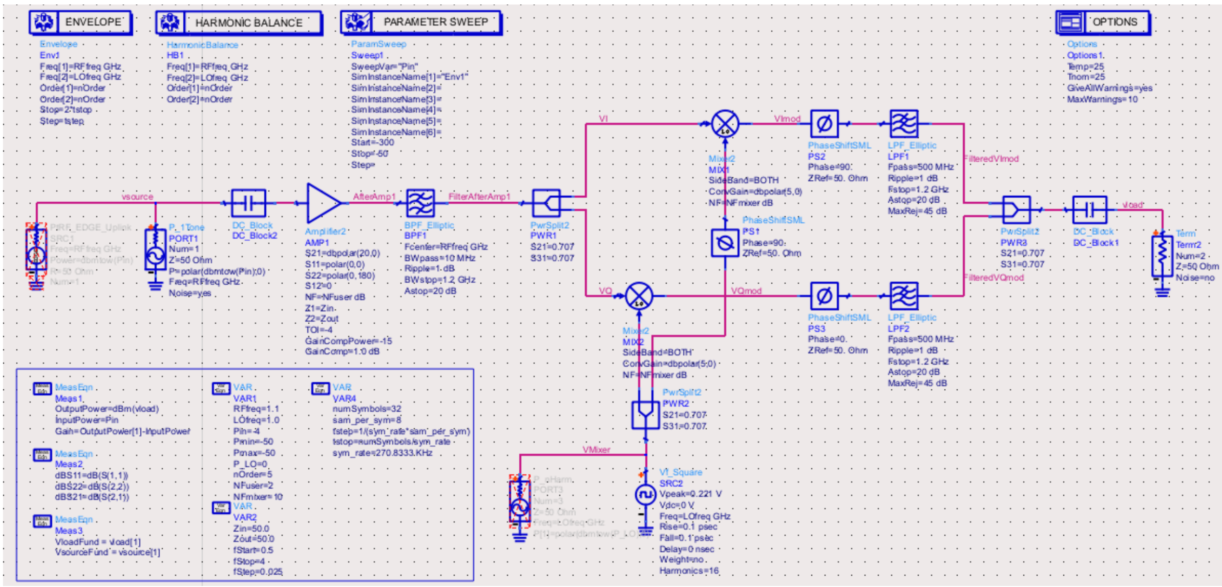


Figure 48. Schematic overview

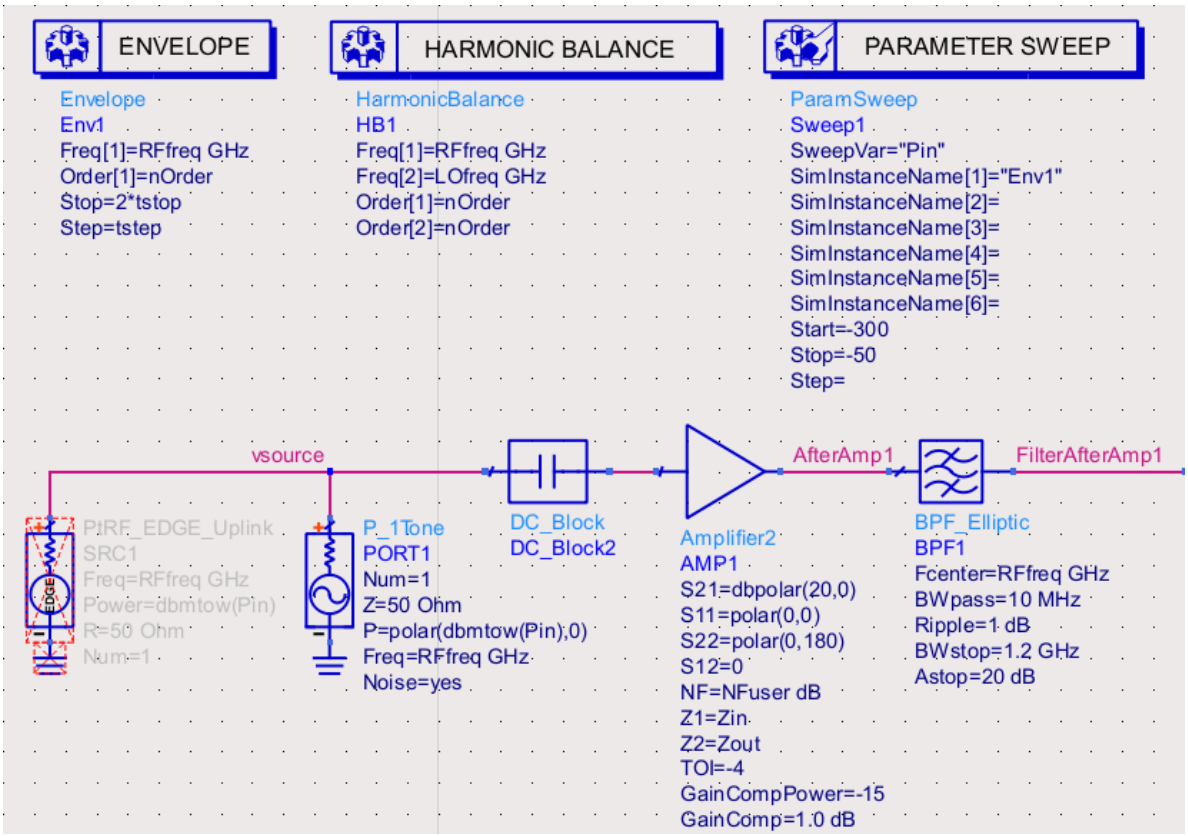


Figure 49. Schematic overview - zoom 1

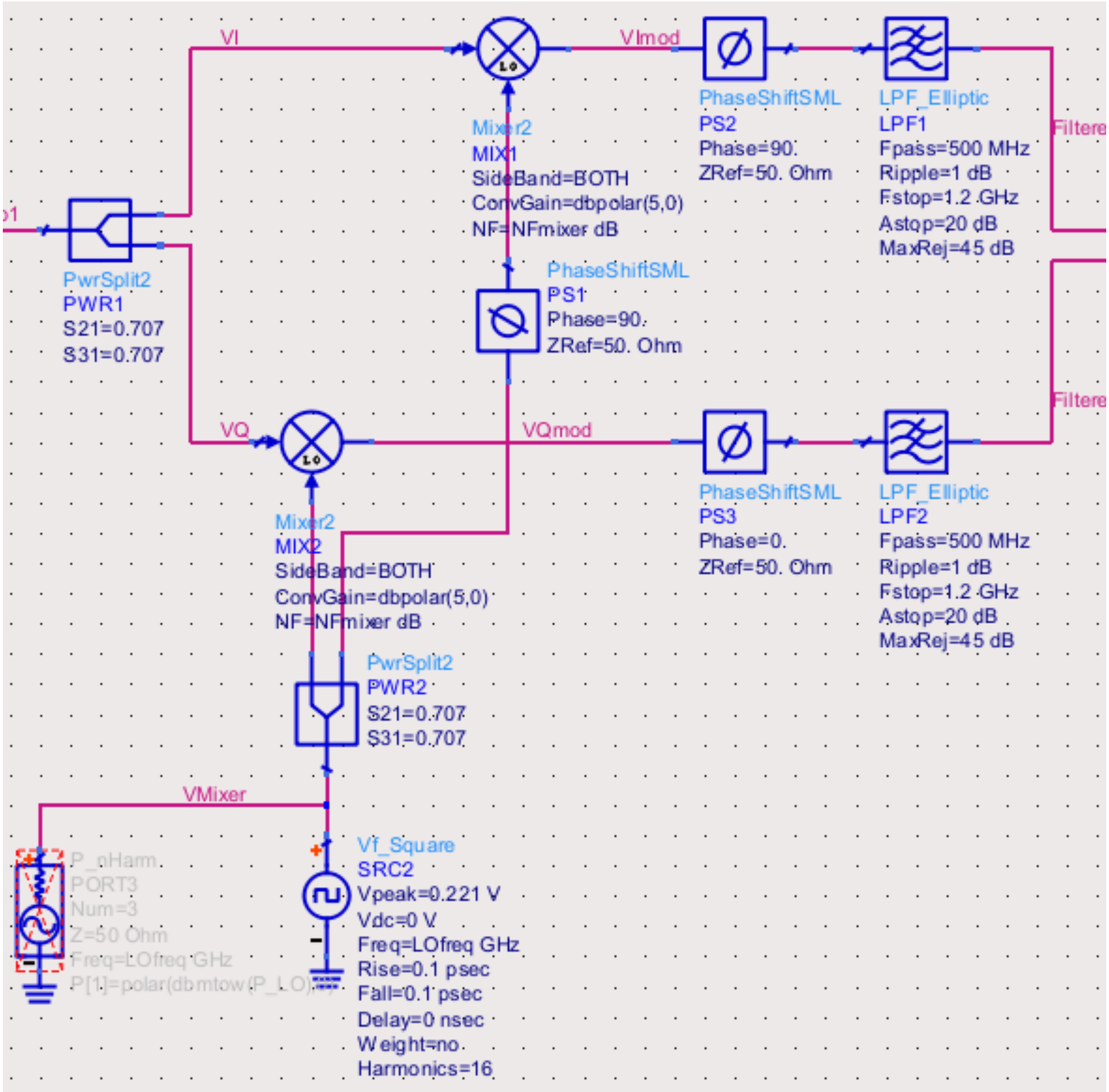


Figure 50. Schematic overview - zoom 2

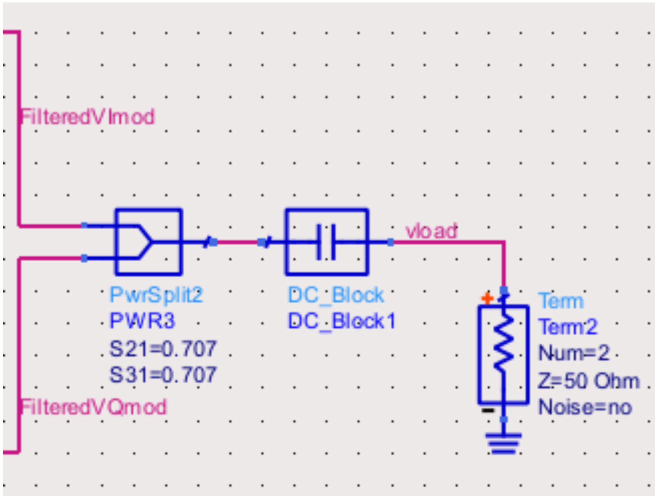


Figure 51. Schematic overview - zoom 3

A more detailed description of the parameters in the Harmonic Balance and Envelope controllers will be given shortly.

8.1.2 Elliptic filter properties - 'LPF_Elliptic'

In the previous simulation setups, we have not used any kind of filters. It can be of great use to filter unwanted harmonics or intermodulation terms. In ADS, several different types of filters exist, but here only Elliptic filters will be presented.

The typical parameters of filters available in ADS are:
Fpass = passband edge frequency in Hz
Fstop = stopband edge frequency in Hz
Ripple = Stopband ripple in dB
Astop = attenuation at stopband edges in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level in dB
N = filter order. If not given, it is calculated based on BWpass, Ripple, BWstop, and Astop
IL = insertion loss in dB
Z1 (and Z2) = reference impedance for port 1 (and 2), in ohms

After the first amplifier, which can be considered as an LNA in a real RF front-end, we place a bandpass filter ('BPF_Elliptic') with center frequency ('Fcenter') at the carrier frequency 'RFfreq GHz'. It is used to illustrate a possible attenuation of the LNA harmonics.
Set the other parameters of the filter as below:

Name	**Value**
Fcenter	RFfreq GHz
BWpass	10 MHz
Ripple	1.0 dB
BWstop	1.2 GHz
Astop	20 dB
MaxRej	45 dB
Temp	27

The second filter that will be used is a low-pass filter ('LPF_Elliptic') with 'Fpass' and 'Fstop' of 500 MHz and 1.2 GHz, respectively. This filter should be used after the mixers and before the load resistance, see Figure 48. As we use a sinusoidal and switching LO, we place a filter after the switching LO to filter the driving signal of the mixer.

Exercise 8.1.2 To make sure about the filter characteristics, prepare a simple testbench with 'P_1Tone' and connect it to the first port of the filter. Connect a termination port at the second port of the filter.
Put labels on the wires at the input and output, 'Vin' and 'Vout', respectively. Since this circuit is linear you don't need HB simulation. Rather you can choose 'AC' simulation. For this purpose, from the 'Simulation-AC' palette open 'AC' controller box and define the desired frequency range and step. Run the AC simulation. The frequency of the 'P_1Tone' will be swept as defined in the 'AC' controller. In the DDW you can define the filter gain using 'Eqn' box as follows:

```
FilterGain = dB(mag(Vout/Vin))
```

The filter characteristics is plotted using  ***box, where the variable 'FilterGain' is introduced.***

8.2 Simulation control boxes

As seen in the noise simulations of the mixer, the output of the Harmonic Balance simulation corresponds to IF frequency components. However, when the schematic becomes larger with several important nodes, we must be able to monitor the signal quality at each node. By using one Harmonic Balance controller and one Envelope controller taking two tones into account, all important nodes can be monitored. The controllers are shown in Figure 52. To be able to compute the noise floor at the input and output, we also need to simulate the circuit with a very low signal power (below noise level). This is performed by using a 'Parameter Sweep' where two simulation runs are defined.

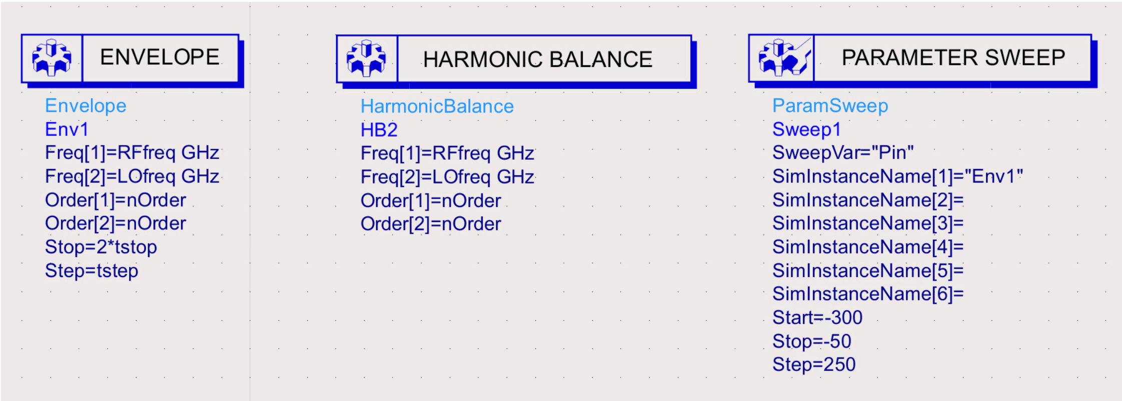


Figure 52. Harmonic Balance and Envelope controllers

When the simulation is completed, the results will be denoted as 'HB1.HB.SIGNALNAME' for a signal called 'SIGNALNAME', obtained during the harmonic balance simulation of HB1. In that way all important nodes can be monitored. Double-click on the HB box and go to the 'Sweep' tab. Set the sweep variable to be 'Pin' with min and max value, 'Pmin' and 'Pmax' with a step size of one. Also, two new variables labeled 'Pmin' and 'Pmax' should be defined in the schematic. With these variables a sweep of input power will be performed for all harmonic balance simulations at the same time. However, setting 'Pmin' and 'Pmax' to the same value makes it simple to evaluate the signal quality for a specific input power level. Then, if the performance of the receiver over some input power range is of interest, you just redefine their values accordingly.

The corresponding Envelope simulation results can be accessed by 'Env1.HB.SIGNALNAME' and as in previous simulations we also need a parametric sweep of the input power to calculate the noise floor and output power. The two power levels used in this example is -300 dBm (which is a negligible power, much less than the noise floor to be picked up) and -50 dBm as seen in Figure 52.

Note: If only one simulation control box is enabled when the simulation is started, the signal name is labeled 'SIGNALNAME' and not

'Env1.HB.SIGNALNAME' or 'HB1.HB.SIGNALNAME', which makes it hard to directly re-use the plots or equations.

8.2.1 Harmonic contents and filtering at the mixer

In the simulation of the complete receiver, we will investigate the influence of filters, simulate the SNR at the input and output of the receiver, compute the noise figure of the receiver, and see how different circuit variants or specifications affect the final result. From the theory and previous simulations, we know that there are mixing terms generated by a mixer due to nonlinearities and also harmonic components of LO.

Exercise 8.2.1: Disable the Envelope controller to reduce the simulation time. Set both 'Pmin' and 'Pmax' to '-50' as we just want to generate one set of harmonics. Simulate and plot the harmonic contents of the signal before the mixer, 'VI', by 'dBm(VI)' (or 'dBm(HB1.HB.VI)'). The harmonic contents of the signal after the mixer, 'VImod', can be plotted by 'dBm(VImod)'. Similarly, the contents of the signal after the filter, 'FilteredVImod', can be plotted by 'dBm(FilteredVImod)'.

At the output, labeled 'vload', you can see some spectral components to be largely suppressed due to quadrature downconversion and signal combination (remove the LPFs and compare 'dBm(vload)' to 'dBm(VImod)'. Before running the simulation without the LPFs, go to History in the toolbar of the DDW window and click "on". This will allow to keep previous simulations in the same charts, making easy to compare with new results.

Obviously, extra advantage can be taken from the LPFs. Activate the filters and adjust their 'Fpass' and 'Fstop' frequencies properly. Use markers in the HB plots to clearly see what is happening. Adjust Astop until you can see a clear damping of the spurious spectral components, but then reset it to 20 dB.

This is a good exercise to be done for each pair of signals in the schematic: plot the harmonic content of all signals before and after a device in the schematic. As for the amplifier in Section 4.3, apply a two-tone test and note that you have to set the Harmonic Balance controller box correctly to be able to see the intermodulation tones.

8.2.2 Gain, SNR and NF

Calculation of signal amplitudes as well as noise floor levels in the receiver can be simplified by taking advantage of the ADS built-in functions.

Enable the Envelope Controller and re-simulate the circuit. When the circuit simulation has been performed, the data display window pops up. Initially, we will plot the spectrum at the input and output of the receiver. Create two equations and plot them in two separate windows:

```
Spectrum_vsource=dBm(fs(mix(Env1.HB.vsource,{1,0})))
Spectrum_vload=dBm(fs(mix(Env1.HB.vload,{1,-1})))
```

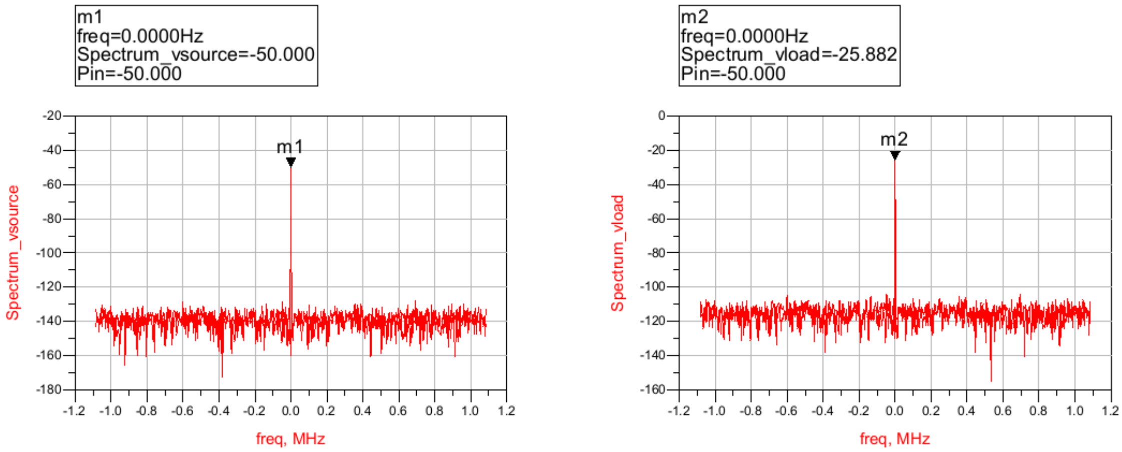


Figure 53. Input and output spectrum

(Carefully click the inserted Marker to find the -50 dBm sweep, if you get Pin=-300 instead).

In the first equation, we compute the spectrum of node 'vsource' from the 'Env1' simulation at frequency '1*RFfreq+0*LOfreq' (by using 'mix' command). Next, we perform a time-to-frequency transform (fs) and return the result in dBm ('dBm'). Similarly, in the second equation the 'vload' spectrum is calculated around IF frequency '1*RFfreq-1*LOfreq'.

If we assume a channel bandwidth of 1 MHz, the band around the carrier is from -0.5 to 0.5 MHz. However, due to the Fourier transform the frequency span seen in Figure 52 is `1/tstep = 270.833 kHz × 8 = 2166.6 kHz.`

Next, we define two new equations, which are very similar to the two spectrum equations recently defined.

```
Input=spec_power(dBm(fs(mix(Env1.HB.vsource,{1,0}))),-
0.5MHz,0.5MHz)
```

```
Output=spec_power(dBm(fs(mix(Env1.HB.vload,{1,-1}))),-
0.5MHz,0.5MHz)
```

The ‘spec_power’ returns power integrated between the two specified frequencies, in this case -0.5 MHz and 0.5 MHz, at ‘1*RFfreq+0*LOfreq’ (or at ‘1*RFfreq-1*LOfreq’) while the power of the spectrum is given in dBm.

As the input power has been swept, we have two sets of data with index ‘0’ for the first simulation when Pin is -300 dBm (the signal is negligible while noise is emphasized) and index ‘1’ for the second simulation when Pin is -50 dBm. Then, the input signal (Pin_signal), input noise (Pin_noise), output signal (Pout_signal), and output noise (Pout_noise) can be computed accordingly. From the signal and noise power levels, also the noise figure and gain can be computed as shown in Figure 54. Note that equations similar to the used above can be applied to compute the Adjacent Channel Power Ratio (ACPR) in a transmitter, but integration is performed over different frequency ranges.

Keep in mind that the noise power levels and receiver noise figure are very sensitive to simulation parameters! Initially set ‘NFuser’ of LNA and ‘NFmixer’ to ‘2’ dB and ‘10’ dB, respectively.

```
Eqn Spectrum_vsource=dBm(fs(mix(Env1.HB.vsource,{1,0})))
Eqn Input=spec_power(dBm(fs(mix(Env1.HB.vsource,{1,0}))),-0.5MHz,0.5MHz)

Eqn Pin_signal=Input[1]
Eqn Pin_noise=Input[0]
Eqn SNR_in=Pin_signal-Pin_noise

Eqn NF=SNR_in-SNR_out
Eqn Gain=Pin_signal-Pout_signal

Eqn Spectrum_vload=dBm(fs(mix(Env1.HB.vload,{1,-1})))
Eqn Output=spec_power(dBm(fs(mix(Env1.HB.vload,{1,-1}))),-0.5MHz,0.5MHz)

Eqn Pout_signal=Output[1]
Eqn Pout_noise=Output[0]
Eqn SNR_out=Pout_signal-Pout_noise
```

SNR_in	SNR_out	NF	Gain
63.414	61.036	2.378	-24.118

Figure 54. Formulas used for SNR and NF calculations with typical simulation results (may differ from yours)

Exercise 8.2.2: As we have several different filters and noise figure values of the amplifier and mixer, we can see how combinations of the filters affect the SNR and NF of the receiver. Simulate (at least) combination 1 and 2 and two other combinations as shown in Table 1. For each combination of input parameters you should simulate the testbench to find Pout, Gain, SNRin, SNRout, and NF as in Table 1. The button used to short the filter is located to the left of the 'disable' button in the toolbar  ***. The NF is given in dB. The first value of Pin means that the noise floor level is simulated for an input power of -300 dBm, and the second value stated is the input power when Pout, Gain, and SNR are calculated. Simulate with both a sinusoidal LO and a switching LO as discussed in Section 7.2.2.***

Estimate also the overall NF of the receiver by using Friis formula. Note that it might be cumbersome when the filters are on unless a very narrow bandwidth is defined.

	Input parameters			**Simulated performance**					
#	**Filter**	**NF (dB)**	**Pin (dBm)**	**Pout (dBm)**	**Gain (dB)**	**SNRin (dB)**	**SNRout (dB)**	**NF (dB)**	**NF (dB) by Friis**
1	**All filters**	**NFuser = 2** **NFmixer = 10**	**-300** **-50**						
2	**No filters**	**NFuser = 2** **NFmixer = 10**	**-300** **-50**						
3		NFuser = NFmixer =							
4		NFuser = NFmixer =							
5		NFuser = NFmixer =							

Table 1. Simulation of the receiver for several filter and parameter combinations

8.3 Noise Computation under Blocker and Phase Noise

In this section we will compute the noise power over the bandwidth of interest when a blocker (strong in-band interferer) is elevating the noise floor due to LO phase noise. For this purpose, we save **'Lab_QUAD'** design as **'Lab_QUAD_BLOCKER'**. Some changes are necessary to run the simulations. The LO is replaced by an oscillator with phase noise, 'OSCwPhNoise'. The phase noise (PN) is specified as a list of frequency offsets where the corresponding phase noise values are expressed in dB units. An example of PN characteristics is shown in Figure 55.

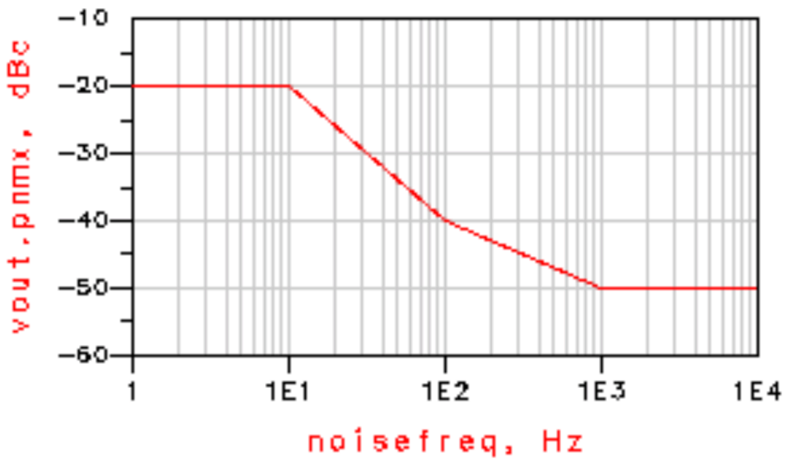


Figure 55. Phase noise characteristics

This PN characteristics can be defined in the 'OSCwPhNoise' component by e.g. the following data:

```
list(10Hz, -20dB, 100Hz,-40dB, 1kHz,-50dB)
```

A more realistic PN of the LO used in our simulations is defined by the following list:

```
list(31.25e3Hz, -85dB, 62.5e3Hz, -94dB, 125e3Hz, -103dB,
250e3Hz, -112dB, 500e3Hz, -118dB, 1e6Hz, -140dB)
```

The corresponding LO model is also illustrated in Figure 56.

Figure 56. Phase noise characteristics of LO used in simulations

Set the 'Fpass' and 'Fstop' parameters of the filters in the I and Q paths to 100.5 MHz and 1.2 GHz, respectively.
The time step and total time of the Envelope simulation should also be modified, so that the obtained frequency spectrum is correct. For the interferer (blocker) at offset frequency of a few MHz, we will choose the simulation bandwidth equal to 5 MHz. Consequently, we introduce two new variables 'SimBandwidth' and 'SimFreqReso' and set them to '5 MHz' and '50 kHz', respectively. Then we set 'tstep' and 'tstop' to '1/(2*SimBandwidth)' and '1/SimFreqReso'. Set Step and Stop time in the envelope controller to tstep and tstop, respectively.

When phase noise is added to the LO, we should also use a 'NoiseCon' (HB Noise Controller), which increases the accuracy of the noise computed around the "carrier" (the IF) at the output node 'vload'. Double-click on the 'NoiseCon' component. There are five tabs. Make the following changes:

Freq: Sweep type: Log, Start/Stop/Pts./decade: 30kHz/1MHz/5
Nodes: Pos node/Neg node: vload/(empty), click add
Misc: Input frequency: RFfreq GHz
PhaseNoise: Phase Noise Type: Phase noise spectrum, Frequency: (RFfreq-LOfreq) GHz
Display: NLNoiseStart, NLNoiseStop, NLNoiseDec, InputFreq, CarrierFreq, PhaseNoise, NoiseNode

If all the components are set correctly, the testbench should look like in Figure 57.

IMPORTANT: Make sure to use a low-pass filter, like 'LPF_Butterworth' (or 'LPF_Elliptic'), after the mixers. If a filter is wanted after the amplifier, the BPF should be replaced by a LPF to make sure that noise power calculations are valid. The BPF has shown to create problems in these simulations.

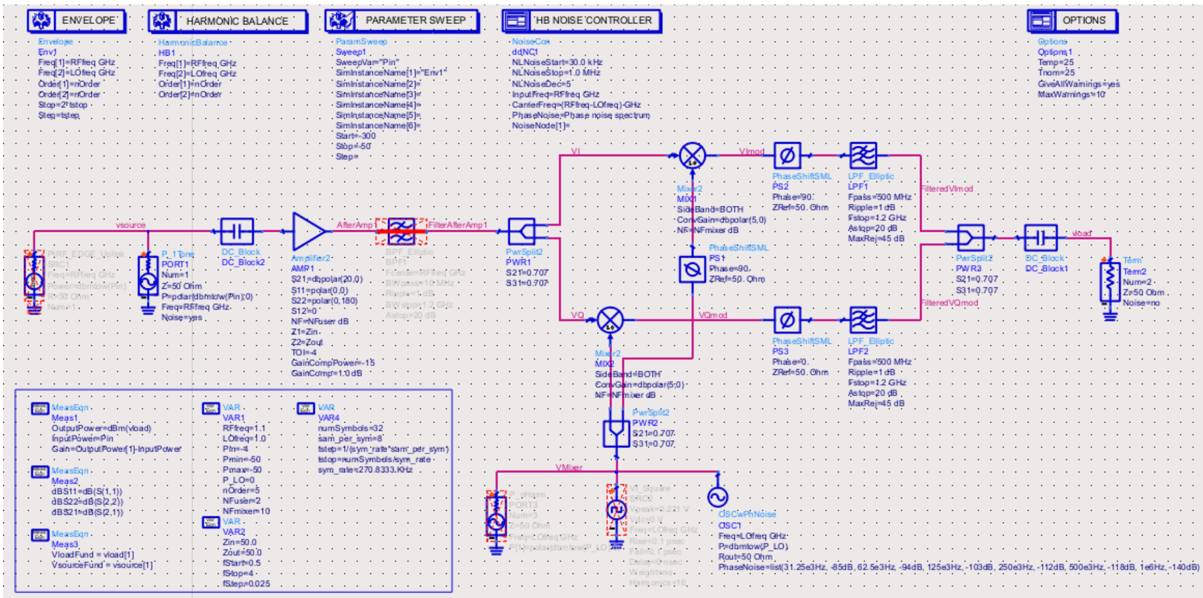


Figure 57. Impact on noise floor due to in-band blocker

Simulate the testbench. When the simulation is performed a new data display window pops up. Open the data display for 'Lab_QUAD' and save it as 'Lab_QUAD_ BLOCKER'. Note that during the first parameter sweep the influence of PN will not take effect since the interferer power is negligible (-300 dBm). During the second sweep the reciprocal mixing occurs and the output noise level around the blocker is elevated.

To plot the spectrum at the load for the first and second sweep, use  box. Initially, define Spectrum_vload as in section 8.2.2.
Then, 'Spectrum_vload_Refnoise=Spectrum_vload[0,::]' and
'Spectrum_vload_PN=Spectrum_vload[1,::]', respectively.

Next, assume our signal band is placed between -4 MHz to -3 MHz from the blocker. Using the Eqn box (and the List box) you can calculate the noise power which decides SNR under reference conditions and under the blocker as follows.

```
Output_noise_ref=spec_power(Spectrum_vload[0,::],-4MHz,-3MHz)

Output_noise_PN=spec_power(Spectrum_vload[1,::],-4MHz,-3MHz)
```

Please note that the results can vary from one simulation to another even by a few dB due to noise variations.

Exercise 8.3a: Compare the noise values for increased blocker power (e.g. by 10 dB).
Verify the effect of IF filters.
Compare also the results for different LNA gain. In this case not only the blocker power, but also the reference noise at the output is affected.

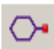
Finally, replace the input source 'P_1Tone' with 'P_nTone'. Define two tones (as in Two-Tone test) at RFfreq–1.5*f_offset and RFfreq+0.5*f_offset. Set 'f_offset' to 0.001 (1 MHz) and 'Pin' to -50 dBm. With this spacing the IM3 product falls in the middle of the signal band (-3.5 MHz) while each tone undergoes reciprocal mixing elevating thereby the noise floor.

Exercise 8.3b: Run the simulation and compare the noise values measured over the signal band (-4 MHz, -3 MHz). The 'HB' box can be disabled.

8.4 QPSK Modulated Signals in a Zero-IF Receiver

In this section of the lab, we aim at simulating our receiver with a QPSK-modulated signal based on data generation blocks, so that you learn how to model more realistic signals. The signal at the output of the receiver will be analyzed in the frequency domain and also EVM will be calculated. The schematics we have used so far are based on the Analog/RF simulation engine in ADS, but in ADS there is another simulation engine called **Ptolemy,** which is used for digital/DSP simulations. These two engines can be used together for ‘Envelope’ and ‘Tran’ (transient) simulations, which is how we will simulate our receiver front-end. The receiver schematic will be created in several steps, to be finally considered as a building block with an RF input signal and two output signals at baseband, I and Q, respectively.
In the project work in the course, you need an RX testbench and a TX testbench. The receiver testbench will be developed in the remaining part of this document. It uses a QPSK modulated signal, but the testbench is very generic and can be used for different modulation schemes.
Additionally, at the Lisam Course Room, in the folder Project, there is an ADS work file (TSEK38_2019_TransmitterTestbench_wrk.7zads) that contains a TX testbench you can modify and use for TX part of the project work. It is important to identify the key concepts presented in the both testbenches (receiver and transmitter) to facilitate your project work.

8.4.1 Receiver Building Block

Copy the design **‘Lab_QUAD’** and name it **‘Lab_MODULATION_SYMBOL**’. Remove the signal sources driving the receiver such that there is no source at all. Insert a Pin, , component from the toolbar and connect it to the input with the name ‘RFin’. After the mixer and the filters, there is a power combiner, a filter, a DC blocker, and a terminal. Remove all of them and put two ports on the wires called ‘FilteredVQmod’ and ‘FilteredVImod’. Make sure that the ports use different ‘Num’, which is important to ensure correct simulation results. Call the ports ‘IFoutI’ and ‘IFoutQ’. In parallel with ground also put two 50-ohm resistors as termination of the IF filters. Depending on how ‘IFoutI’ and ‘IFoutQ’ signals are connected outside the **‘Lab_MODULATION_SYMBOL’,** there will be a need for these termination resistances. However, in our simulations these resistors are not needed and only put there for demonstration purposes and should be disabled. Also disable ALL filters (including the phase shifts in the signal paths of ‘VImod’ and ‘VQmod’) in the receiver since we aim at introducing distortions on the modulated signals. Move the phase shift in the LO path to the other mixer to ensure correct demodulation in the receiver section. To be able to visualize the received data in the time domain, we downconvert the signal to DC (zero-IF). Consequently, we also need to set ‘LOfreq’ to equal to ‘RFfreq’ at 1.1 GHz. Also, adjust ‘numSymbols’ to 128, and 'sam_per_sym' to 16. If the instructions have been followed correctly, a schematic similar to Figure 58 is obtained.

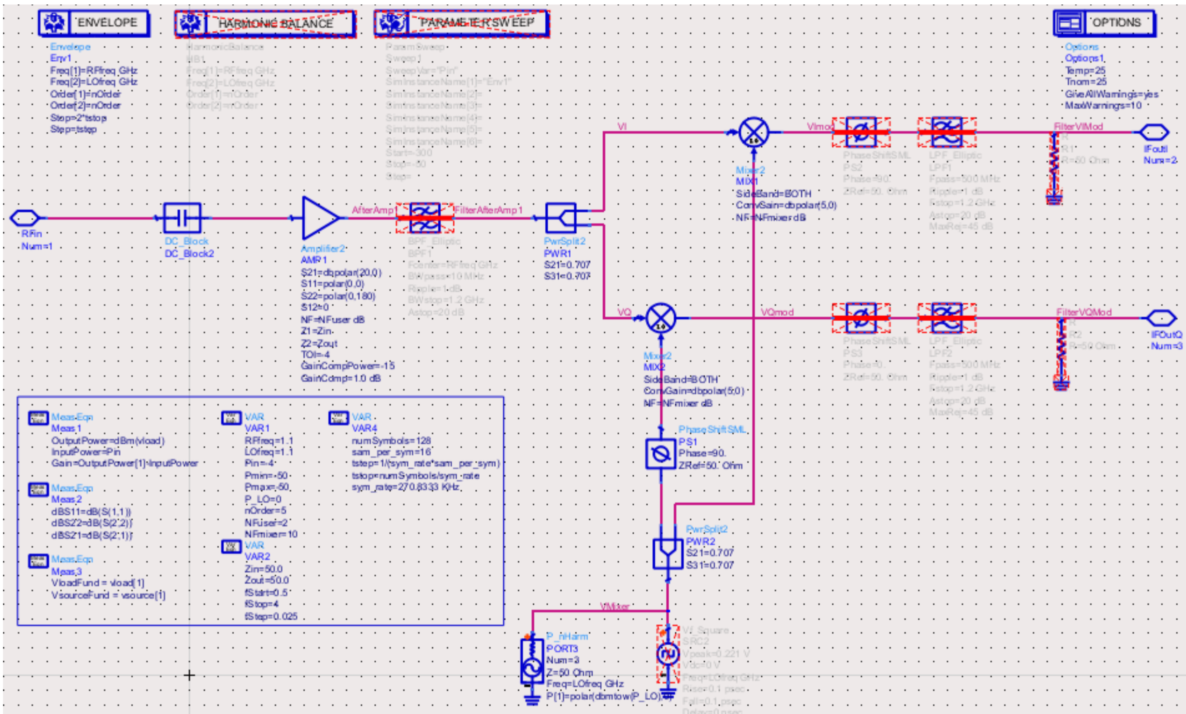


Figure 58. "Lab_MODULATION_SYMBOL" schematic

The next step is to create a symbol that will be used in the digital/DSP (Ptolemy) simulator. To create the symbol, go to menu 'Window -> Symbol' and then click 'OK' to proceed. A new window pops up with the Auto-Generate option. Click 'OK' to create the symbol. Double-click on each pin and make sure that the "Number" matches the "Num" field of the ports in the schematic. Even if there is no match, the simulation will run but with incorrect result.

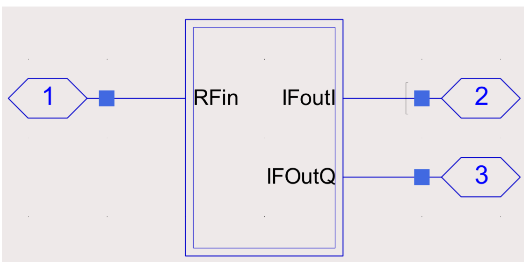


Figure 59. 'Lab_MODULATION_SYMBOL' symbol

8.5 Digital Signal Processing Network

Initially, in the main menu choose 'DesignKits -> Manage Libraries -> Add Design Kit from Favorites and select DSP.
Create a new schematic and call it **'Lab_MODULATION_Ptolemy'**. In the design we distinguish between different networks, as outlined in Figure 60. The different circuit blocks will be described individually in this section.

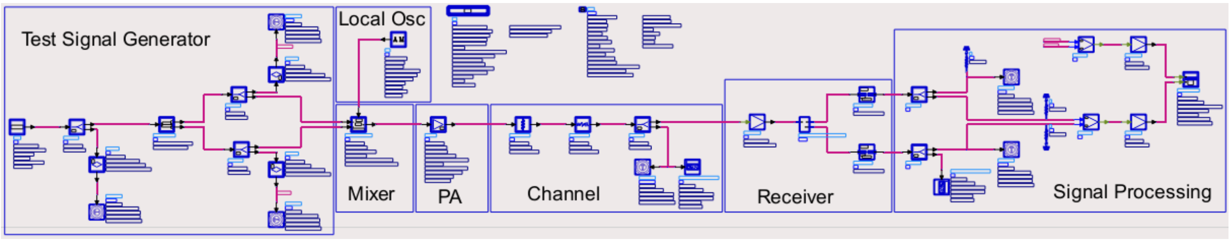


Figure 60. Final schematic

8.5.1 Test Signal Generator

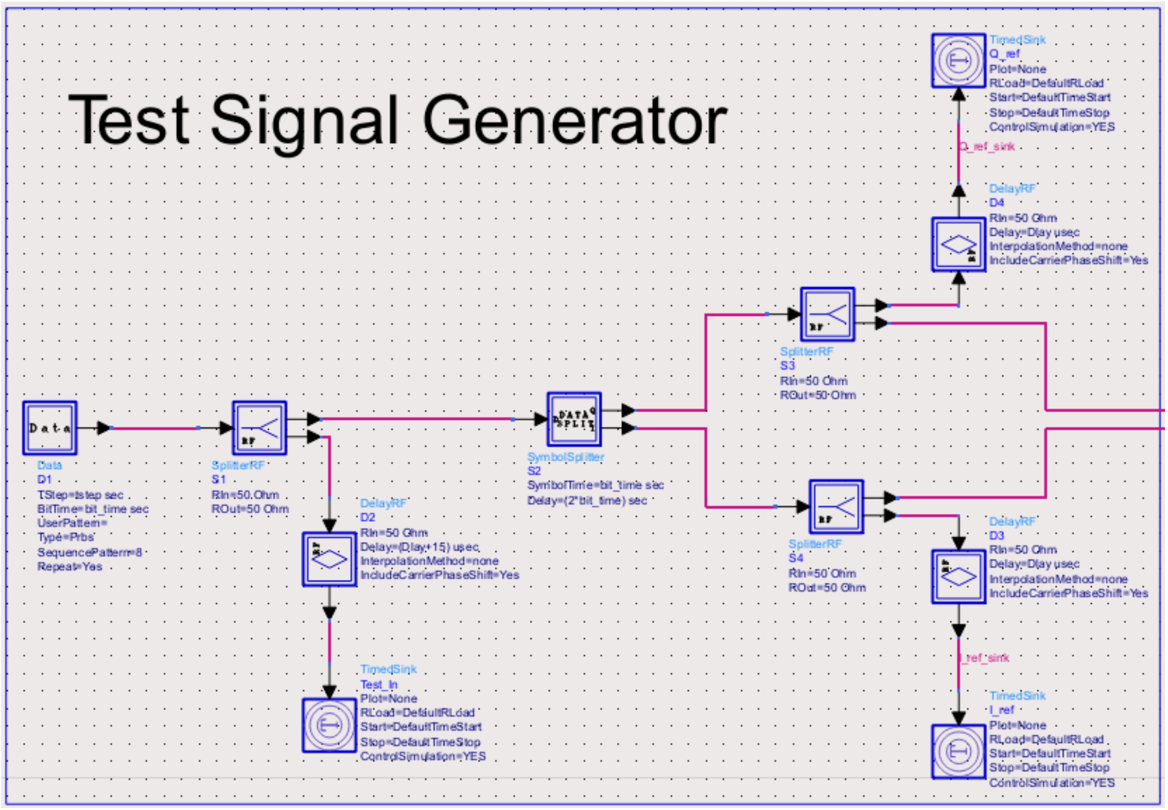


Figure 61. Test Signal Generator

The test signal generator will generate the digital data stream that will be used in the IQ modulator and then fed to the 'Mixer'. The 'Test Signal Generator' is needed in order to make a full testbench, so the 'Receiver' can be evaluated. Add an instance of 'Data' with the following parameters:

Name	Value
Rout	50.0 Ohm
TStep	tstep
BitTime	bit_time
UserPattern	""
Type	Prbs
SequencePattern	8
Repeat	Yes

At the output of the 'Data' instance, add a 'SplitterRF' with input and output impedance of 50 Ohm.
In one branch, add one 'DelayRF' instance with input and output impedance of 50 Ohm, and 'Delay' of '(Dlay+15) usec'.
Put a 'TimedSink' after the delay component to capture the signal in that node. Name it 'Test_In'.
In the second branch of the splitter, add a 'SymbolSplitter' with input and output impedance of 50 Ohm. Set 'SymbolTime' and 'Delay' to 'bit_time' and '2*bit_time'.
After the symbol splitter at the 'I' output, add a 'SplitterRF' with input and output impedance of 50 Ohm.
At the output of the splitter, add another 'DelayRF' block with input and output impedance of 50 Ohm, and 'Delay' of 'Dlay usec'. After the delay element add a 'TimedSink' and labeled it as 'I_ref'. Connect the second output of the splitter to the 'I' input of a 'QAM_ModExtOsc' with properties as below:

Name	Value
ROut	50.0 Ohm
RIn	50.0 Ohm
Power	dbmtow(Pin)
VRef	0.425V
GainImbalance	GainImbalance
PhaseImbalance	PhaseImbalance

As for the 'I' output of the symbol splitter, add the same components for the 'Q' output of the symbol splitter and connect everything to the 'QAM_ModExtOsc'. 'VRef' is used to scale the output signal and is adjusted such that the gain of the amplifier is correct relative the input power, 'Pin'.

8.5.2 Upconversion Mixer and Local Osc

The 'Mixer' consists of the 'QAM_ModExtOsc' that will take the baseband I and Q data and mix it up to the RF carrier by the 'Local OSC'. The oscillator, in its initial appearance, only consists of a sinusoidal source comprised of a 'AM' source with the following properties:

Name	**Value**
ROut	50.0 Ohm
RTemp	-274
TStep	tstep
Type	Conventional AM
FCarrier	RFfreq GHz
Power	dbmtow(0)
Phase	alpha
VRef	1.0 V
FSignal	0 kHz
Vpeak	0.0

Combining 'Upconversion Mixer' and 'Local OSC' the following schematic is obtained.

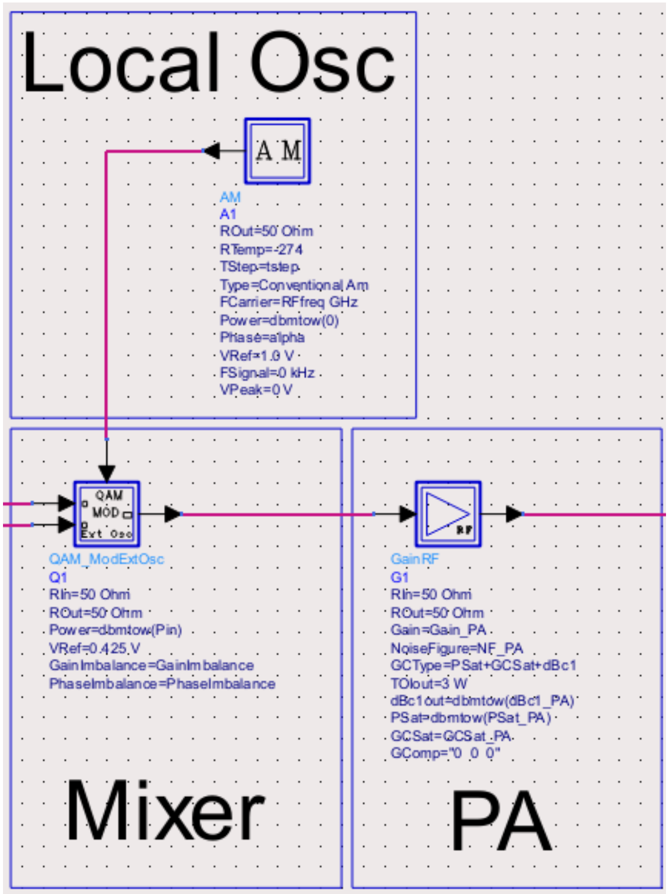


Figure 62. 'Mixer',' Local Osc', and 'PA'

8.5.3 Power Amplifier

As in the analog/RF environment, there is an amplifier in the Ptolemy environment called ‘GainRF’. Place the power amplifier after the ‘Upconversion Mixer’. Set the parameters as in Figure 63.

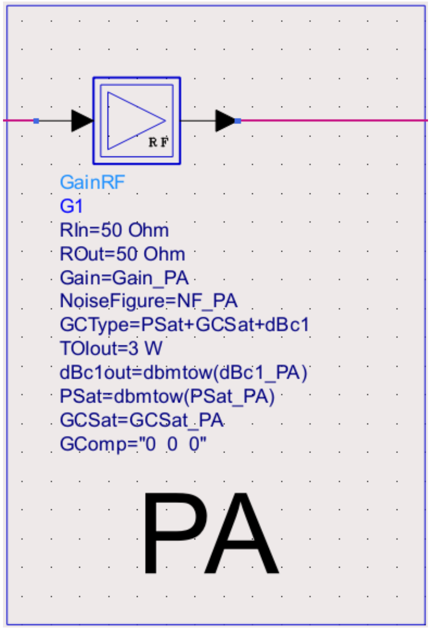


Figure 63. 'Power Amplifier' section

8.5.4 Channel

The radio channel is modeled as a certain loss in dB, a 'MatchedLoss' component', with some noise added, a 'AddNDensity' component, as shown in Figure 64. Set 'Loss' to the 'channel_loss' variable.

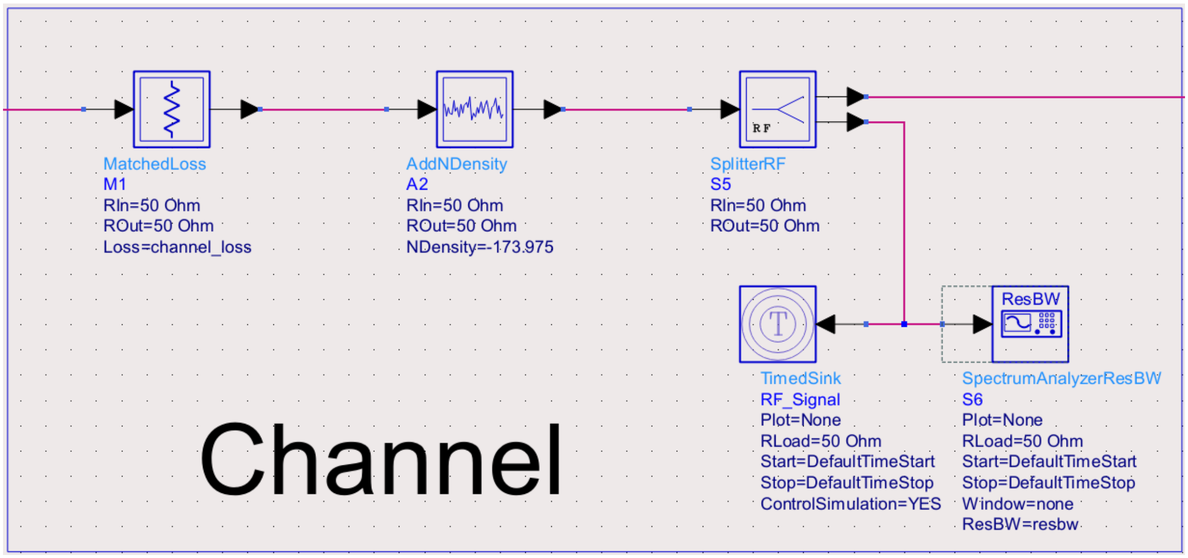


Figure 64. Channel model

Add a signal splitter after the noise component and connect a 'SpectrumAnalyzerResBW' with 'RLoad' and 'ResBW' set to '50' and 'resbw', respectively. Also add a 'TimedSink' labeled 'RF_Signal'.

8.5.5 Receiver

After the 'Channel' we connect the 'Receiver'. As we come from the 'digital' simulation environment with complex valued data and enter the analog/RF time domain simulation environment section we need to connect a 'CxToTimed' instance, which converts a complex data variable to the time domain. After the 'CxToTimed' component we insert the **'Lab_MODULATION_SYMBOL'** component by using the menu 'Insert -> Component –> Component Library…'. Browse to your project library and click on the component name drag it and place in the schematic.

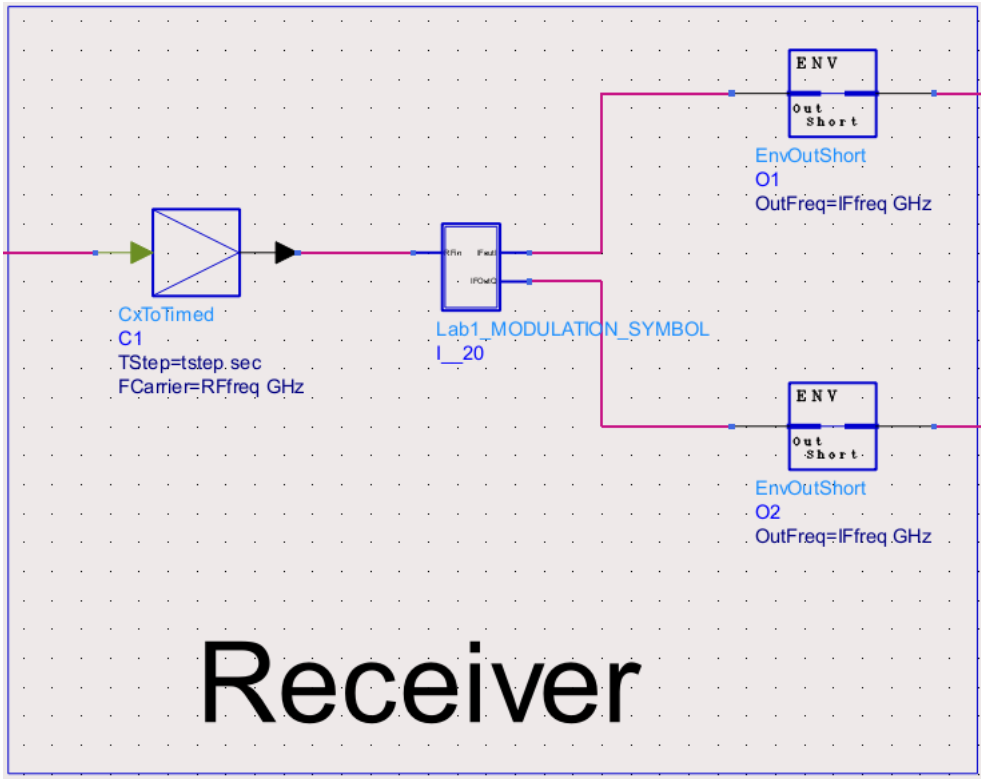


Figure 65. 'Receiver' section

As we return to the digital simulation world after the receiver, we need two signal converters here as well. The converters used in this case are called 'EnvOutShort', which means that the output of the receiver is shorted to the component after the signal converter, which in this case is the first component in the 'Signal Processing' section. Note how the RF carrier frequency and IF frequency variables are used in the signal converter blocks.

8.5.6 Signal Processing

The first components in the 'Signal Processing' section are the signal splitters. The 'Q' signal from the 'Receiver' goes to a dummy 50 Ohm resistor, and in the second branch a 'TimedSink' component labeled 'Q_test', a 50 Ohm load resistor. The 'I' signal from the 'Receiver' goes to a spectrum analyzer, and in the second branch a 'TimedSink' component is labeled as 'I_test', and it is also connected to a 50 Ohm load. The new spectrum analyzer should use the same resolution bandwidth as the previous one. Many signal converters, 'RectToCx' (combining I and Q) and 'CxToTimed' (discrete to time domain), are used as shown in Figure 66. To measure the EVM, an 'EVM_WithRef'-component labeled 'EVM' is used. Connect all components as in Figure 66. Note that the carrier frequency of the 'CxToTimed' blocks cannot be zero.

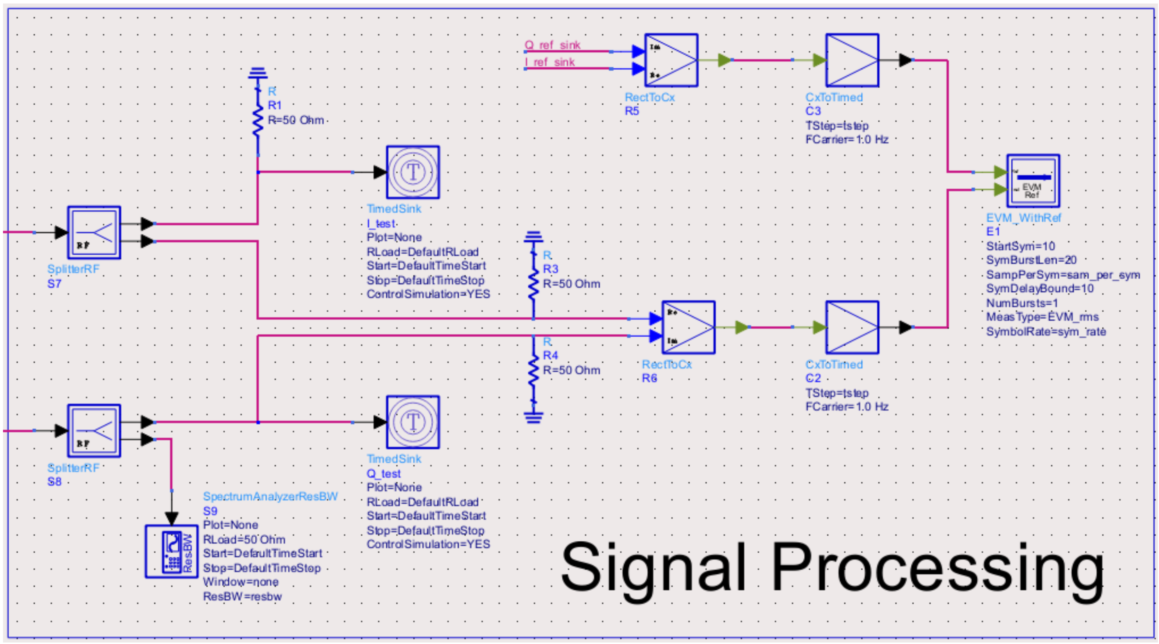


Figure 66. 'Signal processing' network

Note: The 'RectToCx' components are mirrored about the X axis (right-click and "mirror about X"), and that the Q signal branch is the one at the bottom.

The properties of the EVM component, 'EVM_WithRef', are as follows:

Name	**Value**
StartSym	10
SymBurstLen	20
SampPerSym	sam_per_sym
SymDelayBound	10
NumBursts	1
MeasType	EVM_rms
SymbolRate	sym_rate

8.5.7 Data Flow Controller

The last component needed is the 'Data Flow Controller', with instance name 'DF'. Set the parameters of the control like below. The parameters to be shown are controlled under the 'Display' tab in the 'DF' controller.

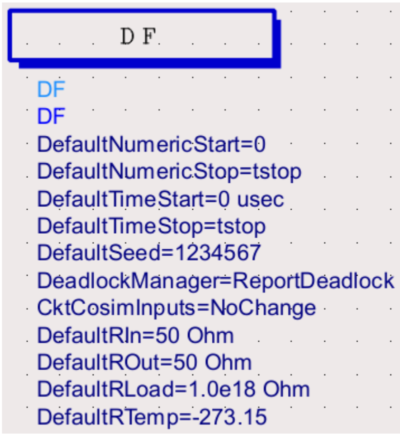


Figure 67. 'Data Flow Controller'

Now it is time to run the first simulation. However, before doing this we need to set a number of variables. Use the following initial variable values:

Name	**Value**
PhaseImbalance	0
Gain_PA	25
GCSat_PA	5
PSat_PA	26
NF_PA	5
dBc1_PA	22 (PA settings => TOI ~35 dBm)
channel_loss	0
alpha	0
GainImbalance	0
num_ave	8 (used for averaging in power spectrum)

sam_per_sym	16
num_symbols	128
tstep	1/(2*bit_rate*sam_per_sym)
symbol_time	2/sym_rate
tstop	num_symbols/sym_rate
bit_rate	270833.3
bit_time	1/bit_rate
resbw	1/(tstop/num_ave)
Dlay	0
RFfreq	1.1
IFfreq	0.0
Pin	-25
sym_rate	bit_rate/2

Run the simulation. When completed, open a new data display window and give it the same name as the schematic. Several Rectangular Plots will be used to plot the digital, transient, and spectral data.

Use two different plots to plot the expressions: 'vs(Q_ref,I_ref)' and 'vs(Q_test,I_test)'

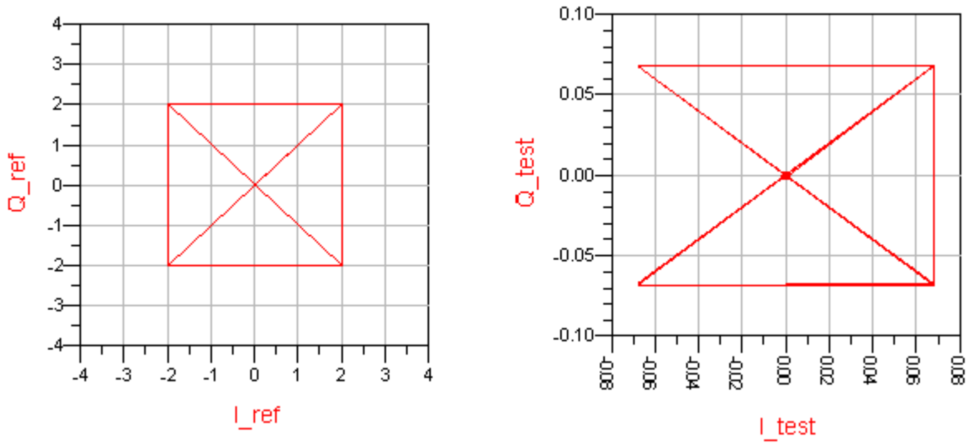


Figure 68. I and Q reference signals, and received I and Q

In next plots: ‘Q_test’, ‘I_test’, ‘Q_ref’, and ‘I_ref’.

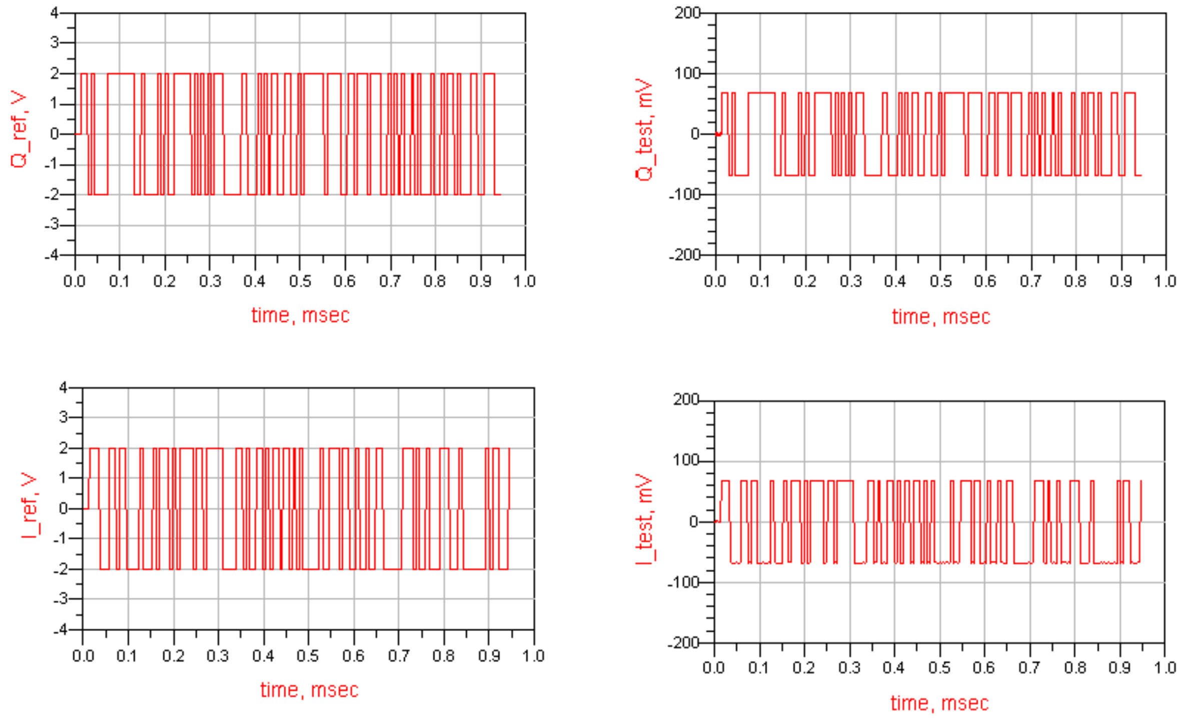


Figure 69. I and Q in the time domain

In next two plots: 'dBm(S6)' and 'dBm(S9)'. Note that S6 and S9 are the names of the 'SpectrumAnalyzerResBW', Figure 64 and Figure 66, for the 'rfSignal' and 'Q_test'. It means that if you have named the spectrum analyzers differently, these labels should be used.

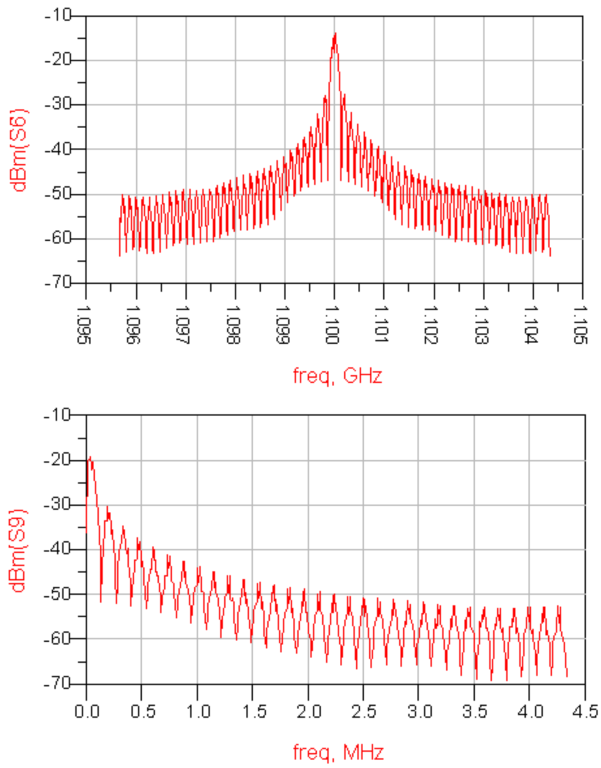


Figure 70. Spectrum of received RF signal, and baseband signal (Q or I)

To calculate the power of the RF signal ('totalP_RF'), the Q signal ('totalP_Q'), and the power in the adjacent channel ('totalP_AC') you can use some equations and print in three list boxes:

```
totalP_RF=spec_power(dBm(S6),1.0995 GHz, 1.1005 GHz)
totalP_Q=spec_power(dBm(S9),0, 0.3 MHz)
totalP_AC=spec_power(dBm(S6),1.1005 GHz, 1.1015 GHz)
```

In this case we have assumed that the power is located within a bandwidth of 1 MHz.

In the list box, print the expression 'EVM_Results'. **Note that e.g. 0.05 means 5%.**

Exercise 8.5.7: Evaluate the impact on 'Q_test', 'I_test', 'EVM', due to 'PhaseImbalance' (degrees), 'GainImbalance', 'channel_loss' (dB), 'Pin' (dB).

This completes the lab!

References

[1] B. Razavi, *RF Microelectronics*, 2nd ed., Pearson Education Inc., 2012

LINKÖPING UNIVERSITY