Visual System Simulator™ 2007

Modeling Guide

Visual System Simulator Modeling Guide

Version 7.5, July 2007

Applied Wave Research, Inc.
1960 E. Grand Avenue, Suite 430
El Segundo, CA 90245

Phone 310.726.3000
Fax 310.726.3005
Website *www.appwave.com*
Technical Support Email *support@appwave.com*
Technical Support Phone 888.349.7610

CONTENTS

1 SIMULATION BASICS 1-1

1.1 Basic Concepts 1-1

1.1.1 RF Budget Analysis and RF Inspector Simulations 1-2

1.1.2 Time Domain Simulations 1-2

1.1.3 Sweeping Simulations 1-3

1.2 Data Signals 1-11

1.2.1 Digital Signals 1-12

1.2.2 Analog Signals 1-13

1.2.3 Complex Envelope Signals 1-13

1.2.4 Sampling Frequencies, Data Rates and Oversampling 1-20

1.2.5 Generic Signals 1-26

1.2.6 Fixed Point Signals 1-26

1.2.7 Signal Properties 1-26

1.2.8 Primary Data Path 1-27

1.3 Automatic Configuration 1-28

1.4 Spectral Analysis in Time Domain Simulations 1-29

1.4.1 Fourier Transforms 1-29

1.4.2 Averaging, Windowing and Power Spectrum Estimation 1-32

1.4.3 Frequency Resolution and Video Bandwidth 1-34

1.4.4 Negative Complex Envelope Frequencies 1-34

1.4.5 Measuring Channel Power 1-37

1.5 References 1-38

2 RF MODELING IN VSS 2-1

2.1 RF Blocks In VSS 2-2

2.1.1 Linear Filters 2-3

2.1.2 Amplifiers 2-5

2.1.3 Mixers 2-6

2.1.4 Miscellaneous Blocks 2-7
2.1.5 Working with Non-RF Blocks 2-8
2.2 RF Budget Analysis 2-8
2.2.1 RF Budget Analysis Simulation Basics 2-9
2.2.2 Frequency Dependent Signal Properties 2-10
2.2.3 Additional Notes 2-14
2.3 RF Inspector 2-18
2.3.1 RF Inspector Dialog Box 2-20
2.3.2 Modulated Signals 2-28
2.3.3 Performance Considerations 2-33
2.3.4 How RFI Works 2-37
2.3.5 RFI Impedance Mismatch Modeling 2-38
2.3.6 Feedback Loops in RFI 2-38
2.3.7 Additional Notes 2-39
2.4 Impedance Mismatch Modeling 2-40
2.4.1 Block Support for Impedance Mismatch Modeling 2-41
2.4.2 Modeling Basics 2-43
2.4.3 Source and Load Impedances 2-45
2.4.4 Interfacing to Non-RF Blocks 2-46
2.4.5 Additional Notes 2-46
2.5 Filter Issues 2-47
2.5.1 FIR Filters 2-49
2.5.2 IIR Filters 2-56
2.5.3 Working With Circuit Filter Based Blocks 2-61
2.6 Nonlinear Modeling Issues 2-63
2.6.1 Aliasing 2-63
2.6.2 Saturation Effects 2-67

3 NOISE MODELING IN VSS 3-1

3.1 Modeling Noise in VSS 3-1
3.2 Circuit Noise 3-2
3.2.1 Linear RF Blocks 3-3
3.2.2 Amplifiers 3-4
3.2.3 Mixers 3-6
3.2.4 Additional Notes 3-6
3.3 Channel Noise 3-7

3.4 Phase Noise....................3-7
3.4.1 Generating Phase Noise....................3-9
3.4.2 Measuring Phase Noise....................3-9
3.5 SNR, Eb/N0 and Es/N0....................3-11
3.5.1 SNR....................3-12
3.5.2 Eb/N0 and Es/N0....................3-13
3.6 Noise and Negative Frequency Folding....................3-13

1 SIMULATION BASICS

Visual System Simulator™ (VSS) is the system level design component of the AWR® Design Environment™ suite. With VSS you can analyze a complete communications system, from data encoding, through transmission, reception and data decoding.

1.1 BASIC CONCEPTS

VSS is a system level simulator. A system level simulation models designs at the component level, such as amplifiers, mixers, or encoders.

VSS models the effects of a system upon one or more signals which can be digital data, modulated analog signals, continuous wave (CW) tones, or other types. VSS designs start with a system diagram and are composed of interconnected blocks representing individual components in a design. The connections between the blocks describe the flow of data through the system.

Blocks typically have one or more *ports*, which serve as the connection points to other blocks. The connection point between ports is called a *node*. Due to historical reasons, ports are also often referred to as nodes.

There are two types of ports: input ports and output ports. Input ports are the entry point of data into a block, and receive data from an output port of another block. When a simulation runs, data flows from the output port of one block to one or more input ports of other blocks connected to the output port. When a block receives data it applies its behavior to the data and generates any appropriate output.

At least one block must be a source block, which is a block that generates a signal without requiring input from another block. Examples of source blocks are the modulated sources such as the QAM Modulated Signal block QAM_SRC, or the RF Tone Source block TONE.

To perform most measurements there must also be at least one ‘meter’ block with an input port connected to the system. Annotations, which are applied to an entire system diagram, are the exception.

The meter blocks are located in the **Meters** category of the Element Browser. The most commonly used meter is the Test Point block TP. TP lets you monitor any signal.

VSS supports three types of simulations: time domain simulations, RF Budget Analysis simulations, and RF Inspector simulations.

1.1.1 RF Budget Analysis and RF Inspector Simulations

The RF Budget Analysis measurements, in the **System > RF Budget Analysis** measurement category, invoke the RF Budget Analysis (RFB) simulator. These measurements include cascaded noise figure, gain, and IP3 measurements.

The RF Inspector measurements, in the **System > RF Inspector** measurement category, invoke the RF Inspector (RFI) simulator.

Both RFB and RFI simulations are generally restricted to the RF components of a system design. Both simulators operate in the frequency domain and assume steady state behavior. Whereas time domain simulations are started and then run until stopped, RFB and RFI simulations occurs in one step. This allows efficient use of the optimization and yield analysis features of the AWR Design Environment.

RFB and RFI simulations are run similar to Microwave Office® simulations. They are run when the **Analyze** command is chosen, and are only run if the system diagram has been modified. In addition, they are also run when the **Run/Stop System Simulators** command is chosen if the system diagram has been modified.

See the *RF Modeling in VSS* chapter for a more detailed description of RF Budget Analysis and RF Inspector simulations.

1.1.2 Time Domain Simulations

NOTE. Time domain simulations require a VSS-250 or greater license. The **Run/Stop System Simulators** command, which is required to run time domain simulations, is not available without a VSS-250 license.

In a time domain simulation, data is represented as a stream of samples, with each sample representing the value of the signal at a specific point in time. The time difference between each sequential sample in a signal is called the *time step*. The inverse of the time step is the *sampling frequency*, commonly denoted as fs.

In VSS the time step is fixed during the main simulation. However, different signals may have different time steps. For many blocks the sampling frequency of the output signals are the same as the sampling frequency of the input signals. Other blocks may modify the sampling frequency. For example, a digital-to-analog converter would typically generate several analog output samples for each digital value input.

The time step and sampling frequency can be viewed at each output port in a system diagram using the SMPFRQ and TSTEP annotations.

When a simulation runs, the source blocks generate a sequence of data samples representing the time varying data. This sequence of samples is passed on to any connected blocks. These connected blocks in turn 'read' each incoming sample, process them, and if necessary generate new samples for the blocks connected to their output ports.

Time domain simulations typically require many thousands of samples be generated and processed before any significant measurement can be made. Because it takes real time to process all these samples, time domain simulations are normally started and then run in the background. While the simulation is running, the graphs update based on the samples available to the individual measurements.

VSS time domain simulations are started by choosing **Simulate > Run System Simulators**. The simulation runs until you pause or stop it, or it completes processing the required number of samples. Each time you choose **Run System Simulators** new simulations are started and you lose all measurement results from previous simulations.

1.1.3 Sweeping Simulations

It is often desirable to obtain simulation results for several values of a design parameter. For example, bit error rate (BER) measurements are typically made over several signal to noise ratio values, or adjacent channel power is measured for several signal power levels.

When one or more design parameters are modified during a simulation, the simulation is a *swept simulation*. Each *sweep* consists of running the simulation for a specific set of design parameter values. The design parameters are called the *swept variables.*

All VSS simulators support sweeping. Multiple swept variables are supported for performing multi-dimensional sweeps.

Each sweep can be viewed as a separate simulation. When a new sweep begins, most blocks reset their state. The exceptions are blocks that need to track inter-sweep information such as the BER block. For time domain simulations the sweep runs until a control block determines that the simulation has run long enough for that sweep.

There are several elements to a simulation sweep:

- Design Parameters - Defining the design parameter(s) to sweep
- Controlling Sweeps - Defining the criteria for starting a new sweep (time domain simulations)
- Swept Measurements - Viewing the results

DESIGN PARAMETERS

Design parameters are the values that are varied from sweep to sweep. Having more than one design parameter results in a multi-dimensional sweep.

VSS supports two types of design parameters. The first is an equation variable that is assigned to one or more block parameters. The second is a built-in CW power level generated by the large signal vector network analyzer VNA_LS.

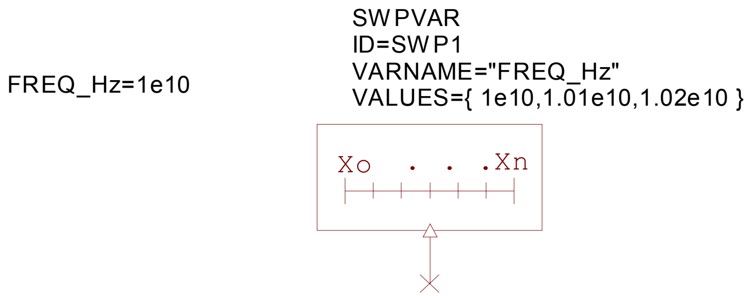
In order for an equation variable to be a design parameter, the set of values to be assigned for each sweep must be defined. This is normally done using one of the following blocks:

- BER, BER_EXT, FER_EXT, SER and SER_EXT - These are the BER meters found under **Meters > BER**.
- SWPVAR - The swept variable control block, found under **Simulation Control**.
- VSA - The vector signal analyzer block, found under **Meters > Network Analyzers**.

These block have a VARNAME and a VALUES parameter. The VARNAME parameter is set to the name of the variable to be swept, entered in double quotes. The VALUES parameter is set to a vector containing the value for each sweep.

The variable to be swept must be in the system diagram. It must also be assigned a constant as its value and not an equation or other variable.

The following illustrates using SWPVAR to define FREQ_Hz as a design parameter with the sweep values of 10e9, 10.1e9, and 10.2e9:



Note that the value for the VARNAME parameter, “FREQ_Hz”, is enclosed in double-quotation marks as it is referring to the name of the variable, not the values of the variable.

While the equation variable must be assigned a constant value, the VALUES parameter may contain an equation, particularly equation functions. One of the more useful functions is `stepped()`. This function automatically creates a vector starting from a given value and containing equally spaced values up to a given end value. For example:

```
stepped(5,10,1)
```

is equivalent to the vector:

```
{5,6,7,8,9,10}
```

VNA_LS can also be used to define an equation variable as a design parameter. The VARNAME parameter is used as above. The values for each sweep, however, are automatically set to the power levels generated by the block. For example, if PSTART is 5, PSTOP is 10, and PSTEP is 1, and the variable “PWR_dBm” were assigned to the VARNAME parameter, PWR_dBm would be a design parameter with the following values:

```
{5,6,7,8,9,10}
```

Although you can have more than one VNA_LS in a system diagram, only the VNA_LS block that has the largest number of power steps defines a design parameter. This VNA_LS is the *master VNA_LS*. Any other VNA_LS block in the system diagram will still generate a swept power signal. However once those blocks reach their PSTOP power level they stop participating in the remaining sweeps of the master VNA_LS.

Once a design parameter has been defined, it can be used just like any other equation variable. In order for the design parameter to affect the simulation, however, at least one block's parameter must refer to that design parameter, either directly or indirectly through other equations.

NOTE. Previous versions of VSS supported limited sweeping through the use of the `sweep()` equation function. This functionality is still supported. However, it is strongly recommended that the new sweep mechanism be used in place of the `sweep()` equation function. The new mechanism is required in order to use the generic swept measurement capabilities.

The most common use of the `sweep()` function was to define the swept ***Es/N0, Eb/N0 or SNR*** values for a BER simulation. As an example, suppose SNR were being swept. The following equation may have been defined:

```
SNR = sweep(stepped(0,4,1))
```

The variable might then have been assigned to the PWR parameter of the additive white Gaussian noise block AWGN.

To convert this to the current sweep mechanism, you would do the following:

- Cut 'sweep(stepped(0,4,1))' from the equation and replace it with "0":

  ```
  SNR = 0
  ```

- Paste 'sweep(stepped(0,4,1))' as the value of the VALUE parameter of the BER block and remove the `sweep()` function:

  ```
  stepped(0,4,1)
  ```

- Set the VARNAME parameter of the BER block to "SNR", including the double quotation marks.

CONTROLLING SWEEPS

When performing a swept time domain simulation, the simulator needs to be told when to start the next sweep. This is done through a sweep control block. Frequency analysis simulations do not need a sweep control block.

The same blocks that define design parameters are also used to control sweeps: the BER meters, SWPVAR, VSA, and VNA_LS.

Only one block should control sweeping in any system diagram. The block should be located as far downstream from the sources as possible. This is to give all other blocks a chance to process samples before a new sweep is started.

The BER meters by default start a new sweep after a specified number of errors have been detected. This can be disabled by setting the DETACT secondary parameter of the BER block to "Do not sweep".

The SWPVAR, VSA and VNA_LS blocks can be configured to start a new sweep by connecting the block's input to an output port and entering a value for either the block's SWPDUR or SWPCNT secondary parameter. The SWPCNT parameter specifies a minimum number of samples to receive before starting a new sweep. The SWPDUR parameter is similar, except a simulation stop time is specified. A new sweep is started after a minimum of $SWPDUR \cdot fs$ samples have been received, where *fs* is the sampling frequency of the input signal.

SWEPT MEASUREMENTS

Most VSS measurements support displaying results from one or more sweeps. They typically allow you to view the results from either an individual sweep or to display the results of all the sweeps of a swept variable. When displaying the results of an individual sweep, the sweep can be the active sweep, a specific pre-selected sweep, or selected on the fly with the tuner.

When displaying the results of all the sweeps, the format will depend upon the measurement. For measurements with a pre-defined x-axis, such as the waveform measurement WVFM (time is the x-axis) or the power spectrum measurement PWR_SPEC (frequency is the x-axis), the results of each sweep are displayed as separate traces overlaid on the same graph.

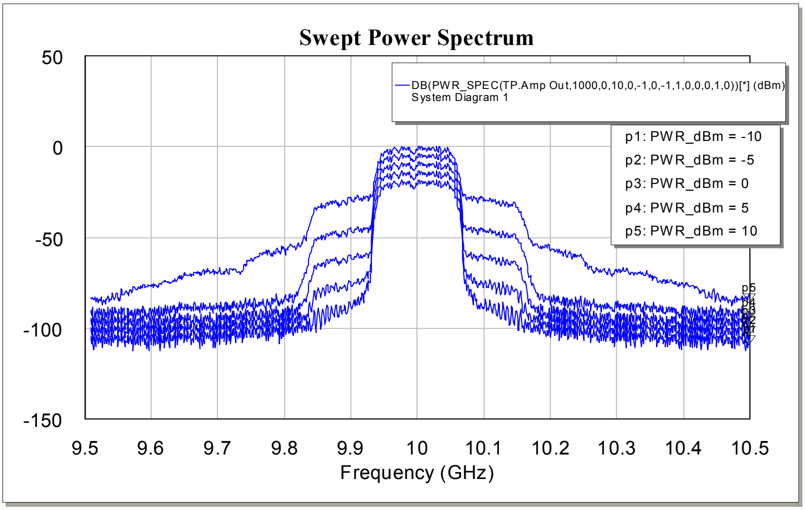


Figure 1-1. PWR_SPEC displaying all sweeps.

Measurements that inherently support sweeping, such as the adjacent channel power measurement ACPR or the AM to AM measurement AMtoAM_PS, support using the values of one of the swept variables for the x-axis. For example, by default S21_PS displays S21 versus measured input power. If the simulation swept frequency, S21_PS could be configured to display S21 versus frequency.

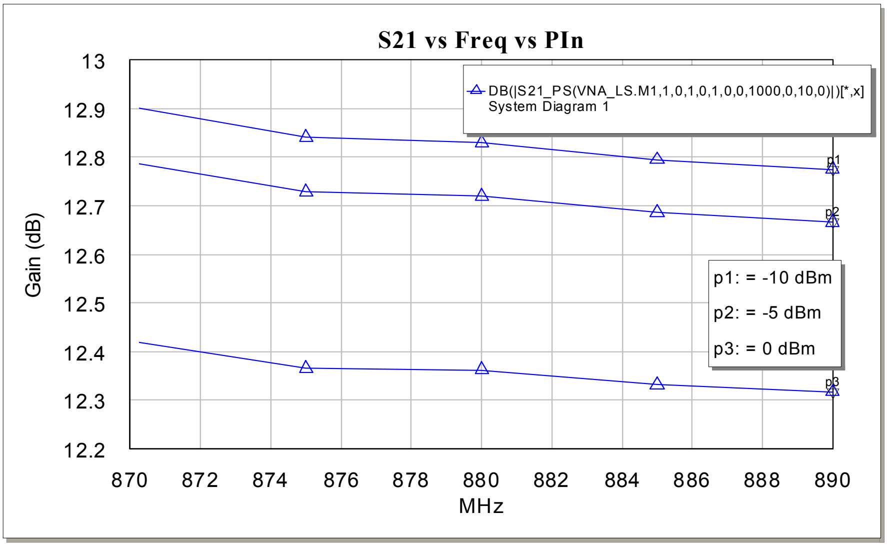


Figure 1-2. S21_PS displaying S21 vs. frequency vs. input power.

When a system diagram contains swept variables, the measurement properties dialog box, opened by choosing either **Project > Add Measurement** or right-clicking on an existing measurement and choosing **Properties**, displays additional controls for configuring the sweeps to plot.

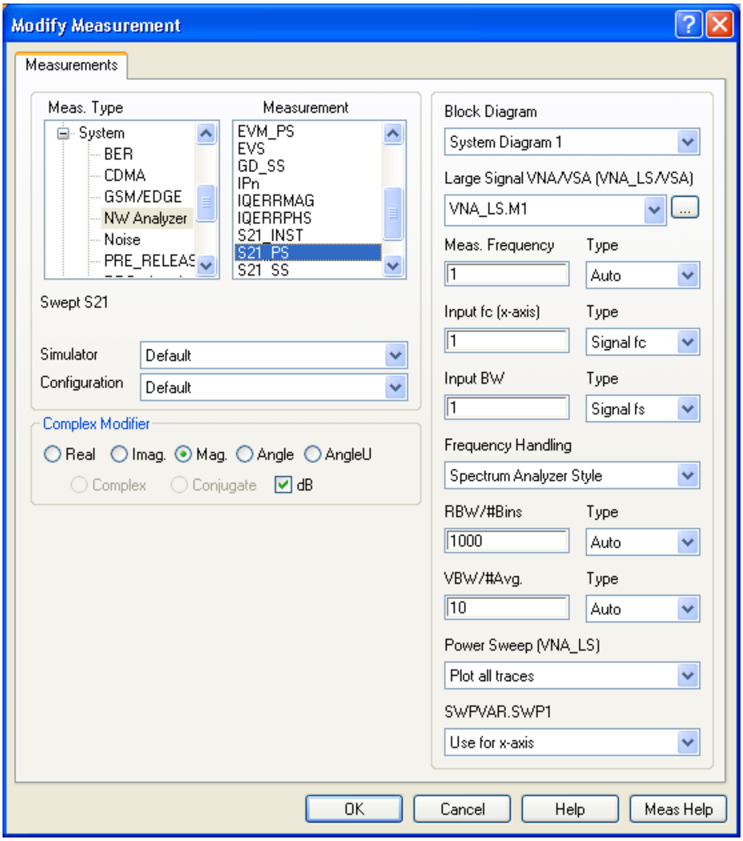


Figure 1-3. Measuremens property tab displaying drop-downs for two swept variables.

If there is a VNA_LS configured to sweep, a drop-down list box with the title **Power Sweep (VNA_LS)** is displayed. For other swept variables, a drop-down list box with a title containing the name of the block defining the swept variable and the ID parameter of the block is displayed - "SWPVAR.SWP1" for example.

The drop-down list(s) contains a number of options, including a list of all the values for the swept variable represented by the list, a "Plot all traces" option, and a "Select with tuner" option.

Measurements that have a pre-defined x-axis such as WVFM or PWR_SPEC also have a "Plot active sweep" option. Measurements that directly support sweeping such as ACPR or AMtoAM_PS also have a "Use for x-axis" option.

Measurements that directly support sweeping typically have 'Swept' in their description. The BER measurements also directly support sweeping.

If a specific swept variable value is selected, the measurement will display only the results for that sweep value. If "Select with tuner" is selected, the measurement will display the results for one sweep value. The specific value can be changed by opening the tuner through **Simulate > Tune**.

If "Plot all traces" is selected, the measurement will display the results for all the sweeps of that swept variable. For measurements with a pre-defined x-axis the results are overlaid as separate traces. Measurements that directly support sweeping will plot their measured values using their 'built-in' x-axis. For the swept power measurements such as ACPR or AMtoAM_PS this is the measured input power from the VSA. For BER this is SNR, Es/N0 or Eb/N0.

If "Plot active sweep" is chosen, only the results of the current sweep are displayed. Note that for multi-dimensional sweeps, only the first drop-down list contains this option. Selecting it will override the settings for the remaining drop-down lists.

1.2 DATA SIGNALS

Data signals form the means of passing simulation data from one block to another. There are several different types of data signals, for the different types of processing supported. They include:

- Digital signals
- Analog signals
- Complex envelope signals
- Generic signals
- Fixed-point signals

Data signals have signal properties associated with them. Signal properties define static attributes of the signal, which are used by various blocks and measurements to aid in the simulation. Examples of signal properties include sampling frequency and data rate, estimated signal power, and estimated phase rotation.

Closely associated with signal properties is the concept of a primary data path. The primary data is important in blocks that have multiple main inputs, such as adders and combiners. The primary data path determines which input signal's properties are used for the output signal.

1.2.1 Digital Signals

Digital signals typically represent digital data to be transmitted or data that has been received.

Digital signals are signals with values that are restricted to a range of non-negative whole numbers. Every digital signal has an *alphabet size* property, commonly referred to as *M*, which defines the range of values. A digital signal with an alphabet size of *M* has values that fall within the following range:

$$\{0, 1, \ldots, M-1\} \tag{1}$$

Each digital sample is also called a *digital symbol*. If M=2 then each sample represents a bit. Normally, when a digital signal is converted to an analog waveform such as in a digital-to-analog converter or a digital transmitter, the number of analog samples representing each digital symbol is maintained as a property of the analog signal. This is the *samples per symbol property*. It is also known as the o*versampling rate*. In the current release the samples per symbol property is always a positive whole number.

The time step of a digital signal is called the *symbol period*. The inverse of the time step is the *data rate*. The data rate and time step at every digital output port can be viewed on a system diagram using the DRATE and TSTEP annotations.

When working with digital sources, the data rate by default is automatically set. If the source is eventually connected to a block downstream that defines the data rate or sampling frequency, such as a transmitter, the data rate of the source is set to produce the proper data rate or sampling frequency downstream. The data rate automatically compensates for any intermediate blocks that may adjust the data rate, such as encoders.

If the data rate cannot be determined from the downstream blocks, it is set to the data rate setting under the **Simulator** tab of the Options dialog box of the system diagram.

1.2.2 Analog Signals

Analog signals are used to represent analog waveforms such as voltage or current. In the current release almost all analog signals represent instantaneous total voltage seen at an output port. There are several PLL blocks (CHPMP_L, PFDCP and LIN_S_PLL) that work with instantaneous total current.

Analog signals are usually *oversampled*, that is the sampling frequency is a multiple of the largest frequency of interest in the signal. To satisfy the Nyquist criterion the signal must be oversampled by at least 2. However, for modeling analog signals an oversampling rate of 8 or more should normally be used.

The oversampling rate is usually determined automatically. Blocks that specify the oversampling rate contain a SMPSYM parameter to let you explicitly set the oversampling rate. SMPSYM can be found in analog signal source blocks such as the TONE and SINE blocks, or in blocks that convert another signal to analog, such as a digital-to-analog converter or a QAM transmitter. (The parameter is named SMPSYM because it originally represented the number of samples per symbol used to represent a digital signal in a transmitter. To maintain consistency, the name was kept for pure analog blocks. Samples per symbol and oversampling rate have the same meaning within VSS.)

If the oversampling rate is not explicitly specified at a block that generates an analog signal, it is normally determined either automatically from the topology of the system diagram or from the oversampling rate option in the **Simulator** tab of the Options dialog box of a system diagram.

1.2.3 Complex Envelope Signals

Often an analog signal is a narrowband signal, where the frequency content of interest is within a narrow frequency band that is centered at a frequency (called the *center frequency*) much larger than the width of the narrow frequency band. An example is a modulated RF signal. These signals can often benefit from *complex envelope* representation.

In complex envelope, or CE, representation, the signal is modeled relative to the center frequency, and the sampling frequency only needs to accommodate the narrow frequency band. This can reduce simulation time significantly, especially if the bandwidth of interest is much smaller than the center frequency.

Take for example a 1 GHz carrier modulated by a 10 MHz data signal. If you wanted to monitor the IM3 and IM5 characteristics of the modulated signal near to the carrier, the frequency band from 950 to 1050 MHz would be more than sufficient to represent those frequencies.

If that signal were to be represented as a real valued analog signal (or *real signal*), you would need a sampling frequency of at least 2100 MHz (twice the maximum frequency of interest, 1050 MHz) to model the 950 to 1050 MHz band. Using a complex envelope representation, however, the sampling frequency would only need to be 100 MHz with a center frequency of 1 GHz.

Mathematically, the relationship between a real signal and its complex envelope representation is:

$$x(t) = Re\left\{c(t) \cdot e^{j2\pi f_c t}\right\} \tag{2}$$

where $x(t)$ is the real signal, $c(t)$ is the complex envelope representation, f_c is the center frequency, and $Re\{a\}$ is the real component of the complex value *a*. $x(t)$ is also called the *real passband signal*.

As a practical interpretation of the above, consider a narrowband modulated signal centered about a high frequency sinusoidal carrier with frequency f_c. Such a signal can be mathematically expressed as:

$$x(t) = x_c(t) \cdot \cos 2\pi f_c t - x_s(t) \cdot \sin 2\pi f_c t \tag{3}$$

$x_c(t)$ and $x_s(t)$ are real valued lowpass or *baseband* signals representing the signal modulating the carrier. They have a bandwidth much smaller than the center frequency. $x_c(t)$ and $x_s(t)$ are the in-phase and quadrature (I and Q) components, respectively, of the real passband signal $x(t)$.

If the complex envelope term of equation 2 is represented as:

$$c(t) = I(t) + j \cdot Q(t) \tag{4}$$

and multiplied by the $e^{j2\pi f_c t}$ term, the argument of $Re\{\ \}$ in equation 2 becomes:

$$I(t) \cdot \cos 2\pi f_c t - Q(t) \cdot \sin 2\pi f_c t + j \cdot (I(t) \cdot \sin 2\pi f_c t + Q(t) \cdot \cos 2\pi f_c t) \tag{5}$$

and equation 2 becomes:

$$x(t) = I(t) \cdot \cos 2\pi f_c t - Q(t) \cdot \sin 2\pi f_c t \tag{6}$$

Equation 6 shows that for a narrowband modulated signal, the complex envelope representation is the same as the baseband modulating signal.

The following graph further illustrates the complex envelope representation of a modulated signal. In this graph, a QAM signal with a 100 MHz data rate is modulating a 500 MHz carrier. The blue curve is the real passband signal. The red and magenta curves are the real and imaginary components of the complex envelope signal. The green curve is the magnitude of the imaginary signal. The graph illustrates that the envelope of the modulated carrier signal is the same as the magnitude of the complex envelope signal at the carrier's center frequency.

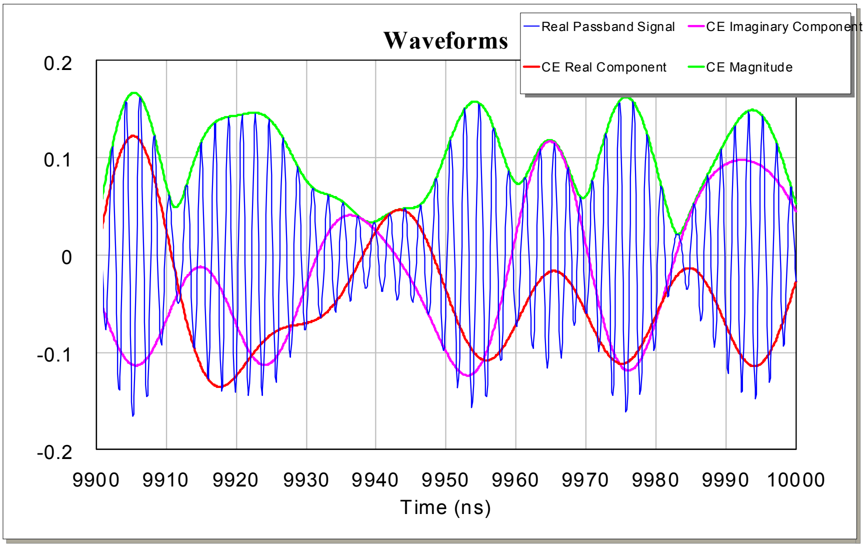


Figure 1-4. Real and complex envelope waveforms for a QAM modulated signal.

WORKING WITH COMPLEX ENVELOPE SIGNALS

Many VSS blocks support complex envelope signals directly. For example, the circuit filters in the **Filters** category, the transmitters, receivers, modulators and demodulators in the **Modulation** category, and many of the RF blocks in the **RF Blocks** category support complex envelope signals directly. Many of the RF blocks also support real signals.

Other blocks, such as many of the math operators in the **Math Tools** category or many of the signal processing blocks in **Signal Processing**, work with complex envelope signals, though they generally treat a complex envelope signal the same as a generic complex signal.

Internally, VSS represents a complex envelope signal using a complex signal with an associated center frequency tag. The value of the center frequency tag at all output ports with such a tag can be viewed on a system diagram using the CTRFRQ annotation.

Complex envelope signals can only be generated by blocks that satisfy two conditions:

- The output signal must be complex.
- A center frequency must be specified.

Blocks that generate complex envelope signals usually have a CTRFRQ parameter for specifying the center frequency.

Most of the modulators and transmitters in the **Modulation** category satisfy these conditions, as do the complex sources in **Sources**. Several other source blocks provisionally satisfy these conditions. They provide a CTRFRQ parameter for specifying a center frequency, but support real as well as complex signals. The TONE block in **RF Blocks > Tones** and the SINE block in **Sources > Waveforms** are such blocks.

In most cases, working with complex envelope signals in VSS simply involves specifying the center frequency at one or two blocks. After that, you can usually specify frequencies of interest using absolute frequency values. For example, cutoff frequencies for circuit filters are specified in absolute frequencies.

Most measurements also automatically detect and handle complex envelope signals. For example, when entering frequencies for the adjacent channel power measurement ACPR, the channel center is specified as an absolute frequency.

There may be cases, however, where you would want to convert a complex envelope signal to the equivalent real passband signal. For example, if you wanted to view the modulated carrier signal itself, you would need the real passband signal.

This conversion can be accomplished using the CE2R block in **Converters > Complex Envelope**. This block allows you to view and manipulate the modulated carrier signal rather than just the complex envelope. Of course, in order to properly represent the complex envelope signal as a real signal the sampling frequency of the generated real signal may be very large compared to the sampling frequency of the complex envelope signal.

A real passband signal can also be converted into a complex envelope signal. This is done using the R2CE block in **Converters > Complex Envelope**. Note that while converting a complex envelope signal to its real signal equivalent is fairly straightforward, generating a complex envelope representation of an arbitrary real signal is not. R2CE works by essentially down converting the real signal to baseband using the specified center frequency, lowpass filtering that result, and downsampling to reduce the sampling frequency.

CENTER FREQUENCY OF ZERO

VSS allows a complex signal with a center frequency of zero. However, when the center frequency is zero, the signal may have special characteristics depending on what the signal represents.

For CW signals such as those generated by TONE or SINE, a center frequency of zero is treated simply as $f_c = 0$.

Modulated signals treat a center frequency of zero differently. In this case the complex signal is treated as representing two separate real signals. The real component is the I channel of a baseband signal while the imaginary component is the Q channel. The signal is called a *baseband I/Q channel* signal.

Several RF blocks implement different behaviors for baseband I/Q channel signals and complex envelope signals. These blocks include the behavioral amplifier AMP_B and the behavioral mixers. Figure 1-5 illustrates the equivalent diagram for an amplifier-mixer link.

There is one significant difference between the two layouts in Figure 1-5. The output of the mixer when using complex baseband I/Q channels is a complex envelope signal, while the equivalent layout using real signals results in a real signal. If the LO frequency is high relative to the data rate, a very large sampling frequency must be used for the real signal equivalent. The sampling frequency for the baseband I/Q channel based layout only requires the sampling frequency be adequate to represent the data rate.

The default behavior of the spectrum based measurements is to display the full spectrum when the signal is a baseband I/Q channel signal. This is similar to what network analyzers do when displaying I/Q channel signals.

The circuit filter blocks also support baseband I/Q channel signals. The main difference between treating a complex signal as complex envelope versus a baseband I/Q channel signal is that the coupling between the real and imaginary components of the input signal.

In the complex envelope, complex math is applied directly, which results in coupling between the two components. For baseband I/Q channel signals, the signals are treated as two separate signals, so there is no coupling between the two components.

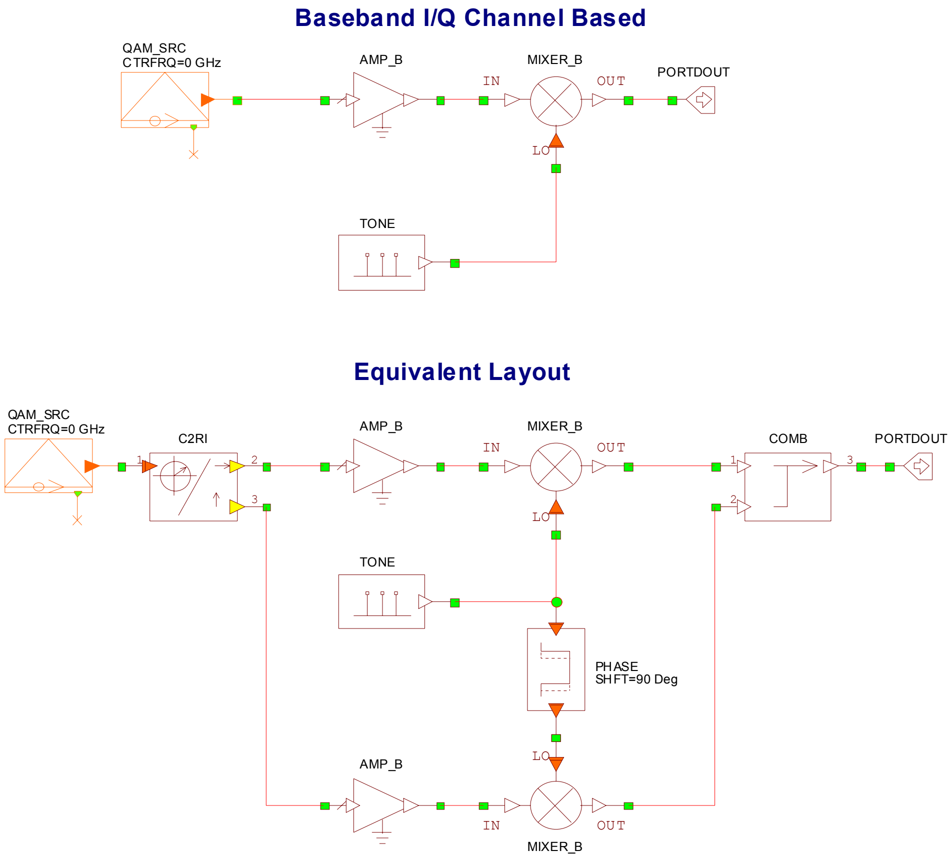


Figure 1-5. Equivalent baseband I/Q channel amplifier and mixer

OTHER BENEFITS AND LIMITATIONS

One advantage of using complex envelope signals in a time domain simulation is the fact that phase information is present in the complex signal. The phase information does not have to be extracted from the real signal.

One of the big disadvantages of complex envelope signals is the modeling of wideband signals, particularly noise, when the center frequency is small enough that the complex envelope frequency band crosses DC. This occurs when the following is true:

$$f_c < f_s / 2 \tag{7}$$

For a complex envelope signal, any negative frequency content can be 'folded' across DC into an equivalent positive frequency content. The problem with a wideband or noise signal is any negative frequency content is already has a positive frequency content, and the net effect is that the signal is doubled over the overlapping frequency range. This topic is covered in more detail in the *Negative Complex Envelope Frequencies* section of this chapter.

Another limitation of complex envelope signals involves the modeling of nonlinear operations. If the intent of a nonlinear operation is to apply the nonlinearity to the equivalent real passband signal, the operation must take into account the complex envelope nature of the signal.

Many of the RF nonlinear blocks do apply their nonlinearities to complex envelope signals with this adjustment. In particular, the nonlinear amplifiers that incorporate the polynomial based nonlinearity found in AMP_B, which include AMP_BV, AMP_F, and VGA_F, along with the behavioral mixers MIXER_B, MIXER_F and MIXER_S, all apply their nonlinearities to the equivalent real passband signal. The AM/AM-AM/PM based amplifiers such as NL_F and NL_S also operate properly with complex envelope signals.

Other RF nonlinear blocks, such as V_LIM, do not take into account the complex envelope nature of the signal, and must be used with caution with complex envelope signals. For these blocks, the safest, though slowest simulation time-wise, approach is to convert the complex envelope signal to the real passband equivalent using CE2R, apply the nonlinearities, then, if desired, convert the signal back to complex envelope form using R2CE.

RF BUDGET ANALYSIS SIMULATIONS

Complex envelope signals are primarily a time domain simulation concept. However, they are also useful when performing RF Budget Analysis simulations with mixers. While these simulations do not directly use the complex envelope signal, they do use the center frequency tag. RF Budget Analysis measurements are made at frequency offsets from the center frequency.

When a mixer is present, the frequency shift imparted by the mixer is reflected in the center frequencies of the input and output signals. With a complex envelope signal, the frequency analysis measurements will reflect the effect of the mixer on a specific input frequency and on its corresponding up or down converted output frequency. Without a complex envelope signal, the frequency analysis measurements can only show the effects at absolute frequencies.

1.2.4 Sampling Frequencies, Data Rates and Oversampling

There are several key concepts related to sampling frequency in VSS time domain simulation signals. First and foremost is the sampling frequency itself. Every time domain signal has a sampling frequency. The sampling frequency is the inverse of the simulation time duration represented by each sample of the signal.

Signals also have a *data rate*, which represents the bandwidth of the 'interesting' portion of the signal. For digital signals the data rate and the sampling frequency are the same. Digital signals that have been converted to analog signals such as with an ADC or a transmitter usually have a sampling frequency that is a multiple of the data rate. This multiple is the *samples per symbol*, as each digital symbol is represented by that many analog samples. It is also called the *oversampling rate*.

For pure analog signals such as tones and other waveforms, the data rate does not have a direct interpretation. Instead, it is used to indicate the interesting portion of the signal to the simulator, and is called the *signal bandwidth*. The sampling frequency is the signal bandwidth multiplied by the oversampling rate. The use of the signal bandwidth in VSS is described in the Signal Bandwidth section.

The following equations show the relationships between these values:

$$f_s = \frac{1}{\Delta t} \tag{8}$$

$$f_s = DRATE \cdot SMPSYM = DRATE \cdot OVRSMP = SIGBW \cdot OVRSMP \qquad (9)$$

Data rate, oversampling rate and sampling frequency can be specified in several ways in VSS. The simplest method is to use the settings in the Simulator tab of the System Simulator Options dialog box either for all system diagrams or for a specific system diagram. The settings in the dialog box are used as the default settings for the system diagram.

Source and transmitter blocks also allow you to specify these settings on an individual block level. Blocks that work with digital signals such as the QAM transmitter QAM_TX have a DRATE parameter for setting the data rate and a SMPSYM parameter for setting the samples per symbol/oversampling rate. More general purpose blocks such the RF tone TONE have a SMPFRQ parameter for setting the sampling frequency. They also have a SMPSYM parameter for setting the oversampling rate. (The parameter is SMPSYM for historical reasons and to illustrate that it has the same effect as the SMPSYM parameter for the digital transmitters.)

Note that the default data rate, sampling frequency and samples per symbol/oversampling rate values specified in the System Simulation Options dialog box are available as built-in variables in the system diagram. These variables are:

Sampling frequency	_SMPFRQ
Data rate	_DRATE
Samples per symbol/oversampling rate	_SMPSYM

The data rate and sampling frequency at all the output ports in a system diagram can be viewed using the DRATE and SMPFRQ annotations. The FRQ_PROP measurement in the **System > Tools** category can be used to display the data rate, sampling frequency and center frequency for specific ports in a graph table.

The proper selection of sampling frequency is important in time domain simulations. The sampling frequency, and for complex envelope signals the center frequency as well, determines the range of frequencies that may successfully be simulated.

SAMPLING THEORY AND ALIASING

Sampling theory states that the minimum sampling frequency required to represent a given frequency is twice that frequency. For real signals the frequency range is:

$$-f_s/2 \le f < f_s/2 \tag{10}$$

For complex envelope signals, the *sampling frequency band* is:

$$f_c - f_s/2 \le f < f_c + f_s/2 \tag{11}$$

Note the use of $\le$ at the lower limit and $<$ at the upper limit. This is because the frequency at $-f_s/2$ (or $f_c - f_s/2$) has the same value as $f_s/2$ (or $f_c + f_s/2$

Any frequency content outside these ranges is aliased into the sampling frequency band. The following diagram illustrates the effect of aliasing.

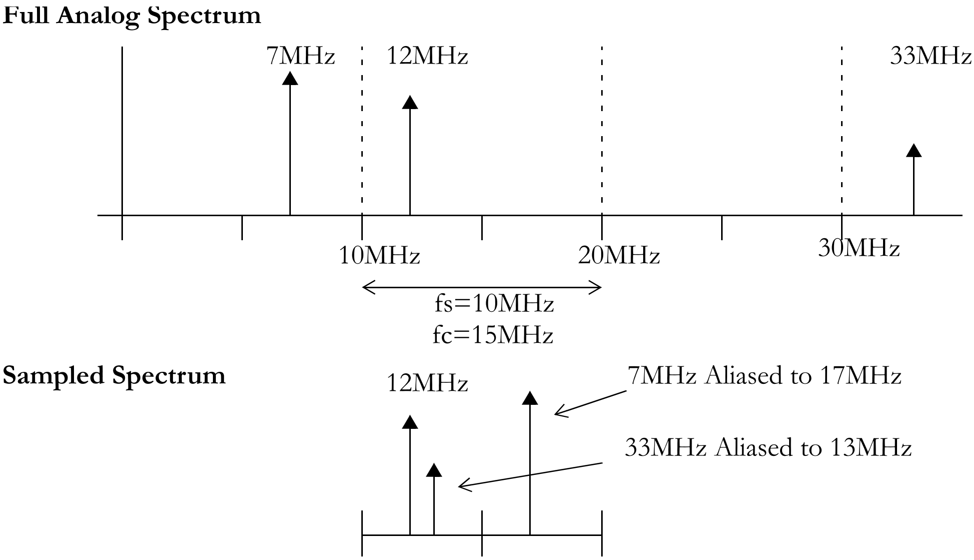


Figure 1-6. Aliasing of frequencies outside sampling frequency band

In this example, there are three tones: 7MHz, 12MHz and 33MHz. This analog signal is then represented as a complex envelope signal with sampling frequency of 10MHz and center frequency of 15MHz. The sampling frequency band is then 10MHz to 20MHz.

The 12MHz tone is sampled correctly since it is within the sampling frequency band. The 7MHz tone is outside the band, so it gets aliased into the sampling frequency band. The aliasing results in the 7MHz tone appearing at 17MHz. The 33MHz tone is also aliased, it appears at 13MHz.

SELECTING A SAMPLING FREQUENCY

In practice, the sampling frequency for analog signals should generally be several times larger than the minimum sampling frequency required. This is particularly important when using the circuit filters or RF blocks. The reason for this is many of these blocks incorporate filters as part of internal upsampling and downsampling. This resampling is a key element in the modeling performed by these blocks.

The filters are used to eliminate frequency content outside the original sampling frequency band when the input signal is upsampled, and again prior to downsampling. Ideally, the filters would be bandpass filters with a passband matching the original sampling frequency band and infinitely sharp cutoffs at the edges of the band. However, ideal filters with infinitely sharp cutoffs are not possible, and the filters must exhibit a transition region at the band edges.

The end result is some roll-off near the edges of the sampling frequency band. The following figure illustrates the roll-off effect on a tone with white noise passed through the nonlinear behavioral amplifier AMP_B. The blue curve is the signal going into the amplifier, the pink curve is the signal output by the amplifier, with slight attenuation of the noise near the edges of the sampling frequency band.

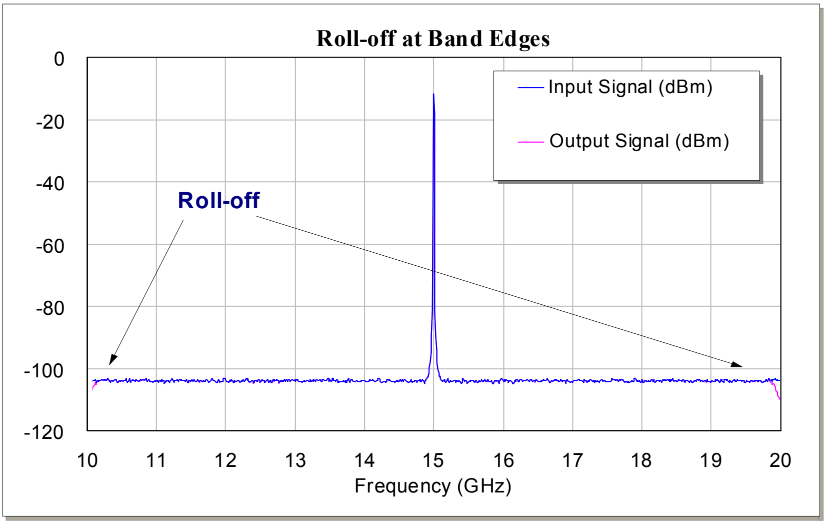


Figure 1-7. Roll-off near sampling frequency band edges due to internal resampling.

SIGNAL BANDWIDTH

The signal bandwidth, signal band, and oversampling rate for analog signals are used several different ways in VSS.

The circuit filter blocks such as BPFB or LIN_S use the signal bandwidth when in IIR mode to determine the frequency alignment points for the bilinear transform. The bilinear transform maps s-domain frequencies to the z-domain in a nonlinear fashion. The frequency alignment point is where the s-domain frequency exactly matches the z-domain frequency. The *Filter Issues* section in the *RF Modeling in VSS* chapter covers circuit filters in detail.

The RF behavioral amplifiers and mixers may also use the signal bandwidth. The RF blocks that support internal resampling can be configured to use the signal bandwidth in their determination of the amount of upsampling required.

Consider a very simplified example: suppose the sampling frequency band is 10GHz to 20GHz (sampling frequency of 10GHz and center frequency of 15GHz). If the signal were passed through a 5th order polynomial (which is what the behavioral amplifiers use if both P1dB and IP3 are specified), the maximum frequency that may be generated is $5 \cdot 20 = 100\text{GHz}$. The upsampling rate that would be required for a given frequency can be found from:

$$UPRATE = roundup\left(\frac{f_{max} - f_c}{f_s}\right) \tag{12}$$

For this example the upsampling rate required would be 9. However, if the signal can be presumed to be entirely contained within the signal band, then the maximum frequency that would need to be accommodated would be the maximum signal band frequency rather than the maximum sampling frequency band signal. If, for this example the sampling frequency band were only 1GHz, then the maximum frequency that may be generated is $5 \cdot 15.5 = 77.5\text{GHz}$, and the upsampling rate would only have to be 7.

Note that the default behavior for the amplifiers and mixers is to use the sampling frequency band. The SIGBW parameter must be cleared to use the signal frequency band. Also note that this is only of value when the center frequency is near the sampling frequency, where the ratio of the maximum signal band frequency to the maximum sampling frequency band frequency is largest.

Mixers can also use the signal frequency band to limit the spurs to be synthesized. In these cases the SIGBW parameter is also used to limit the frequency band used to determine which spurs to generate.

Many of the VSS measurements use the oversampling rate to determine the base number of samples to 'slide' their snapshot of the time domain samples. For example, the WVFM measurement advances the window of time displayed by oversampling rate samples. If the time step is 0.1 ns and the oversampling rate is 10, the WVFM measurement lines up the start of the time axis on 1 ns boundaries.

The spectrum measurements also use the oversampling rate in a similar manner. The point at which samples for the FFTs are taken will start on a multiple of the oversampling rate. This is one reason why a spectrum of a CW signal may appear to jump as the simulation runs. If the frequency of the CW signal relative to the sampling frequency and the oversampling rate are not related by a whole number, they may become out of sync, affecting the display of the spectrum.

1.2.5 Generic Signals

The term generic signals is used to classify signals that do not represent digital nor analog signals directly. For example, control signals are generic signals. The complex valued input to the I/Q Modulator block IQ_MOD is also a generic signal, as is the output of the QAM Mapper block QAM_MAP.

1.2.6 Fixed Point Signals

Fixed point signals are a special case of generic signals. Fixed point signals are used to model the limited floating point precision commonly found in DSP applications. The resolution available for representing floating point values in DSP applications is generally much less than that available in the general purpose CPUs used in personal computers. The fixed point library allows the modeling of those limitations.

The "*Fixed-Point Simulations*" chapter of the *Getting Started Guide* covers fixed-point simulations in detail.

1.2.7 Signal Properties

All signals in VSS have one or more properties associated with them. A *signal property* describes a particular characteristic of the signal during a simulation sweep. All signals have a time step property, which defines the time duration between each sample. The following table lists some of the more common properties:

Property	Data Types	Description
Time Step/ Sampling Frequency	All	The time span between samples. The inverse of the time step is the sampling frequency.
Samples per Symbol/ Oversampling Rate	All	The ratio of the data rate to the sampling frequency. The data rate is the sampling frequency divided by the samples per symbol.
Alphabet Size	Digital	The range of allowed values in a digital signal.
Center Frequency	Complex Envelope	The center frequency of a complex envelope signal.
Signal Power	Analog	The average power of the transmitted signal. This defines the operating point and is used for automatic gain control in receivers.

Property	Data Types	Description
Generated Noise PSD	Analog	The average noise power spectral density in a time domain signal. This is used by the BER meters to automatically determine Eb/N0, Es/N0 or SNR.
Phase Rotation	Analog	The average phase rotation that has been applied. This is used to perform automatic phase rotation compensation.
Signal Delay	All	The amount of delay that has been imparted on the signal since its generation.

Many of the properties are used for automatic compensation. For example, most receivers and demodulators use the signal power, phase rotation and signal delay properties to scale and align the signal prior to demodulation and detection.

The signal properties are referred to as '*static*' properties. They are static because they do not change during a simulation sweep. They are determined before any samples are generated, and are used to describe the expected state of the signal. Most of these properties can be viewed using either annotations or measurements found in the **System > Tools** category.

1.2.8 Primary Data Path

Signal properties are normally assigned starting from source blocks. Blocks connected to those blocks then assign properties to their output signals, and the process repeats until all blocks have assigned properties to their signals.

Most blocks in VSS have one input port and one output port for the *main signal*. Other input or output ports are generally secondary in nature. For these one input-one output main signal blocks, the signal properties from the input port are normally assigned to the output port unless the block changes a particular property.

However, blocks that combine multiple input signals, such as the adder block ADD or the combiner block COMB, have multiple main signals to choose from. To determine the signal properties for their output port, these blocks must select one of the input ports as the *primary input port* and use that port's signal properties as the basis for the output port. The signal path formed by this primary input port and the output port is called the primary data path.

The primary input port of blocks such as ADD or COMB can either be determined automatically or selected through a PRIMINP parameter. When automatic selection is used, the block chooses the primary input port based on the following rules:

- Modulated signals have the highest priority.
- With all else equal, the input port with the smallest node number is chosen.

There is no corresponding primary output port. A block with multiple output ports defines multiple primary data paths, one for each input port - output port path.

1.3 AUTOMATIC CONFIGURATION

VSS has the ability to configure many of its settings automatically. For example, when using a QAM transmitter block such as QAM_TX, you can use the general purpose receiver block RCVR to receive and detect the signal. The RCVR block automatically selects the appropriate demodulation based on the received signal's properties. You can change the number of symbol levels in the transmitter and RCVR will automatically be reconfigured to use the new symbol levels. You can even change the transmitter to MPSK_TX and RCVR will automatically reconfigure itself for PSK rather than QAM modulation.

At the same time, blocks that support automatic configuration normally have parameters that can be used to override the automatic configuration. These parameters are often secondary parameters since they are not frequently used. They also typically have no value assigned, which the blocks interpret as use automatic configuration.

Another example of automatic configuration is the ability to automatically set sampling and center frequencies at source blocks whose signals are eventually combined. In this case one block determines its sampling frequency and the other sources automatically set their sampling frequencies to be compatible. For example, if the LO input to a mixer does not have its sampling frequency explicitly set, it will be assigned the sampling frequency of the input or output signal of the mixer.

The block setting the sampling frequency does not necessarily have to be a source block. For example, most of the transmitters contain a DRATE parameter that lets you define the data rate at the output of the transmitter. The input signal to the transmitter will have its sampling frequency set to this data rate if it is specified, while the output signal will have its sampling frequency set to the data rate times the samples per symbol/oversampling rate. If the input to the transmitter consisted of a digital source followed by a rate 1/3 convolutional encoder, the digital source would be automatically configured to generate data at 1/3 the data rate of the transmitter.

When there are multiple data paths all set for automatic sampling frequency determination connected to a single block such as a combiner, the primary data path is used to determine which blocks define the sampling frequency.

1.4 SPECTRAL ANALYSIS IN TIME DOMAIN SIMULATIONS

Performing spectral analysis in sampled time domain simulations requires taking into account several factors that may affect the results. These are primarily related to the limitations imposed by the need to represent continuous signals as a sampled data stream.

This section points out some of these factors without going into depth with the mathematical theory behind them. For more in-depth details, refer to general digital signal processing texts such as Proakis [1].

1.4.1 Fourier Transforms

Spectral analysis in time domain simulations within VSS is performed using the *discrete Fourier transform* (DFT), *windowing* and *averaging*.

The DFT converts N equally spaced time domain samples into N frequency domain samples at equally spaced frequencies. The general equation is:

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot \exp\left(\frac{-j \cdot 2\pi \cdot k \cdot n}{N}\right) \qquad n = 0, 1, 2, \ldots, N-1 \tag{13}$$

The inverse DFT converts N equally spaced frequency domain samples into N equally spaced time domain samples according to:

$$x(k) = \frac{1}{N} \cdot \sum_{n=0}^{N-1} X(n) \cdot \exp\left(\frac{j \cdot 2\pi \cdot k \cdot n}{N}\right) \qquad k = 0, 1, 2, \ldots, N-1 \qquad (14)$$

The relationship between n and frequency is:

$$f = \frac{n}{N} \cdot f_s + f_c - \frac{f_s}{2} \qquad N \text{ even}$$

$$f = \frac{n}{N} \cdot f_s + f_c - \frac{f_s}{2} + \frac{f_s}{2 \cdot N} \qquad N \text{ odd} \qquad (15)$$

The two different equations result in one of the N frequencies always being at the center frequency.

These equations make a big assumption: that the signal is periodic over N samples. However, unless the signal is composed only of tones that fall exactly on one of the equally spaced frequencies, the frequency content will be 'smeared', or *leaked*, into nearby frequencies. This is most apparent when working with pure tone signals. When N results in the frequencies of the tones falling exactly on one of the N frequencies, the frequency spectrum is exact. However, increasing or decreasing N by 1 causes the spectrum to leak.

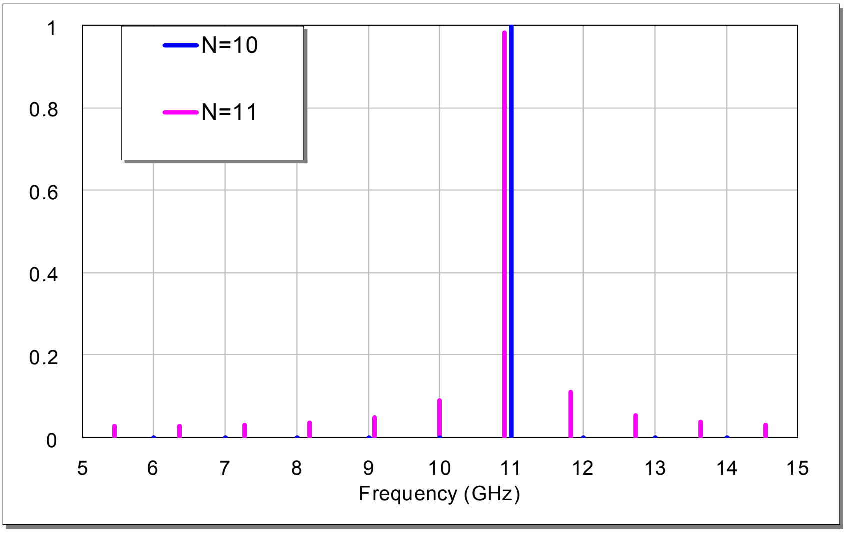


Figure 1-8. Voltage spectrum of an 11 GHz tone at center frequency 10 GHz

Figure 1-8 illustrates the leaking effect. The signal from both plots is the same: a single tone complex envelope signal at 11 GHz with center frequency of 10 GHz and sampling frequency of 10 GHz. The tone's amplitude is 1V, and the y-axis displays linear voltage.

The blue curve has N set to 10. This results in the following sampled frequencies:

5, 6, 7, 8, 9, 10, 11, 12, 13, 14 GHz

The pink curve has N set to 11. This results in the following sampled frequencies:

5.45, 6.36, 7.27, 8.18, 9.09, 10.00, 10.91, 11.82, 12.73, 13.64, 14.55 GHz

For N=1, 11 GHz is not one of the sampled frequencies. Therefore, the tone is spread into the available sampled frequencies.

1.4.2 Averaging, Windowing and Power Spectrum Estimation

An estimate of the power density spectrum called a *periodogram* can be made using equation 13 by applying the following:

$$P_{xx}(f) = \frac{1}{N} \cdot \frac{|X(f)|^2}{2 \cdot Z} \tag{16}$$

Unfortunately, the periodogram is not a consistent estimate of the true power density spectrum, particularly when working with non-periodic signals such as modulated signals.

One of the most common and simplest methods of improving the power spectrum estimate, and the approach taken by VSS, is the *Welch* method. This method involves computing the average of several overlapping periodograms with windowing applied to the time domain samples.

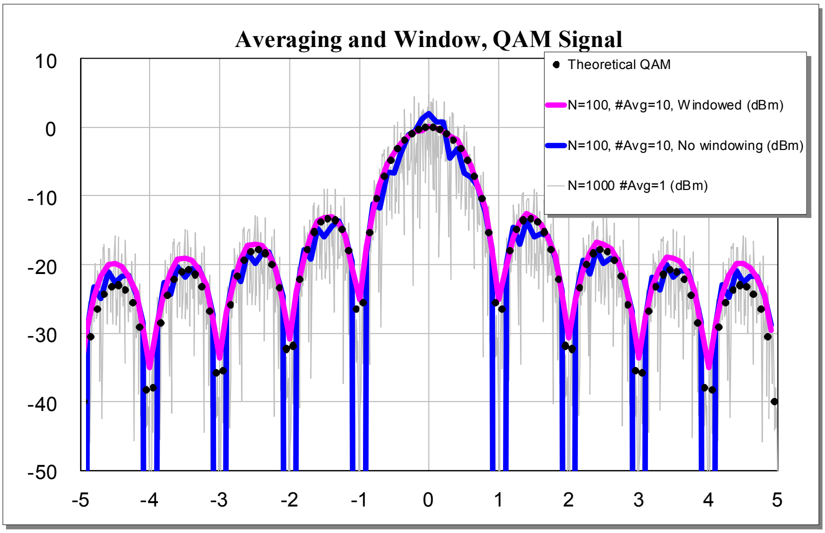


Figure 1-9. Effect of averaging and windowing on power spectrum estimates

Figure 1-9 illustrates the smoothing effect of averaging and windowing on a QAM signal. The QAM signal is not pulse shaped, so the theoretical power spectrum is the sinc function squared. The black markers indicate the theoretical spectrum. The pink curve is windowed and averaged 10 times with 50% overlap. The blue curve is not windowed and is averaged 10 times with

50% overlap. The gray curve is not windowed, has no averaging and uses a DFT with 1000 points. The averaged spectrums use DFTs with 100 points, resulting in a total of 1000 time domain samples being used for all three cases.

Note that with windowing and averaging, the spectrum matches the theoretical spectrum closely in the main lobe -1 GHz to +1 GHz. However, it deviates towards the ends of the sampling frequency band.

In general, windowing and averaging should be used when working with non-CW signals, such as modulated signals, but not with CW signals. The automatic configuration feature of VSS attempts to select the proper settings for performing power spectrums based on the signal properties. If the signal is a modulated signal, it uses windowing and averaging and a fairly large N (1024 in the current release) for the DFT.

If the signal is CW and the tone spacing is a sub-multiple of the sampling frequency, it attempts to find an N that results in all tones in the signal falling on frequency sample points. If an N with a reasonable value is found, averaging and windowing is disabled.

The automatic configuration settings can be overridden. The measurements that rely on power spectrum computations include settings for selecting N, the number of averages, and the windowing settings. These settings are generally secondary settings in the measurements. The various test points and meters also include similar settings. By default the measurement settings use the settings from the test point/meter.

The default settings when averaging and windowing are applied are:

- Number of FFT bins: 1024
- Number of averages: Cumulative
- Window type: Taylor
- Windowing parameter: 4.5 for “Kaiser-Bessel” windowing, 0.0 for all others.
- Sliding ratio: 0.5

1.4.3 Frequency Resolution and Video Bandwidth

Spectrum analyzer settings are normally in the form of frequency resolution and video bandwidth. There are equivalent settings for the DFT based spectrum measurements.

Frequency resolution is directly related to the number of samples used to compute the DFT:

$$f_{Res} = \frac{f_s}{N} \tag{17}$$

Video bandwidth does not have such a direct equivalent. However, an effect similar to that of video bandwidth is produced by averaging. In a spectrum analyzer decreasing the video bandwidth has the effect of smoothing the spectrum. A similar behavior can be accomplished with DFTs by applying averaging.

VSS uses the following relationship to convert from video bandwidth to number of averages:

$$N_{Avg} = ceil\left(\frac{f_s}{N \cdot f_{VBW}}\right) \tag{18}$$

Note that equation 18 is only used to approximate the effects of changing the video bandwidth. It does not necessarily shape the spectrum to what you would see in a spectrum analyzer for the given video bandwidth.

1.4.4 Negative Complex Envelope Frequencies

One of the more confusing aspects of working with complex envelope signals is how to interpret negative frequencies. When possible, you should avoid working with complex envelope signals that contain negative frequencies. This is accomplished by ensuring equation 19 is true:

$$f_c \geq \frac{f_s}{2} \tag{19}$$

However, this is not always possible.

There are three scenarios when a complex signal contains negative frequency content:

- Complex baseband signal representing I and Q channels.
- Complex envelope signal with center frequency of 0.
- Complex envelope signal with center frequency other than 0.

The first two cases both have a center frequency of 0. The difference between the two is the interpretation of the complex values. In general, a signal is treated as separate I and Q channels when it represents the output of an I/Q modulator. CW signals are treated as complex envelope signals.

The second and third cases both treat the signal as complex envelope signals and result in the same behavior.

Complex baseband signals representing I and Q channels are covered in the *Center Frequency Of Zero* sub-section of the *Complex Envelope Signals* section.

For complex envelope signals, negative frequency content has an equivalent positive frequency content. The equivalent positive frequency content is the complex conjugate of the negative frequency content. This is derived from the complex envelope equation:

$$S(f) = \frac{1}{2} \cdot (S_{CE}(f - f_c) + S_{CE}^*(-f - f_c)) \tag{20}$$

where $S(f)$ is the spectrum of the real signal, $S_{CE}(f)$ is the complex envelope spectrum and a^* indicates the complex conjugate of *a*.

Figure 1-10 illustrates this interpretation further. The upper diagram shows the complex envelope and the conjugate of the complex envelope when fc>fs/2. The lower diagram shows that as fc becomes less than fs/2, the complex envelope and its conjugate begin to overlap. The cross-hatched area is the negative frequency content of the complex envelope.

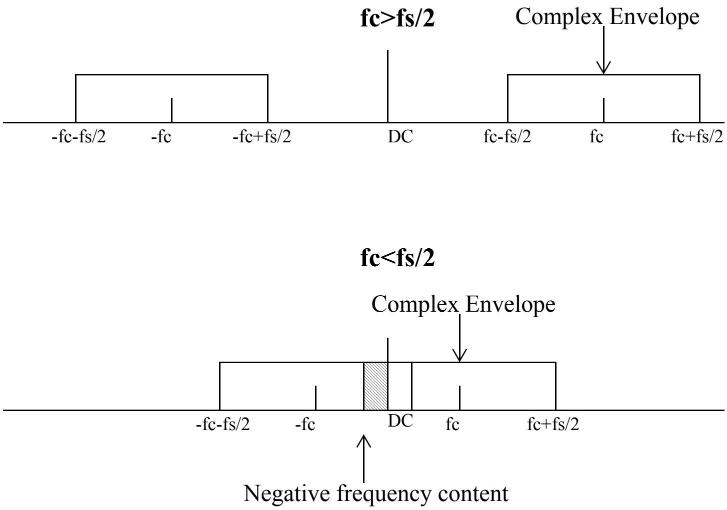


Figure 1-10. Negative frequencies in complex envelope signals

In most cases, when interpreting a spectrum any negative frequency content should be '*folded*' into the equivalent positive frequencies. This results in a spectrum containing only non-negative frequencies, which is what is normally expected when working with real signals.

The default behavior of the spectrum based measurements is to display negative frequencies folded into their equivalent positive frequencies. This presents the spectrum as you would see it on a spectrum analyzer.

One case where negative frequency folding does not work is the modeling of white noise or any other broad spectrum signal where the signal crosses DC. What happens in these cases is there is overlapping frequency content from both the positive and negative frequencies, and folding results in a doubling of the spectrum over the portion of the spectrum where the overlap occurs.

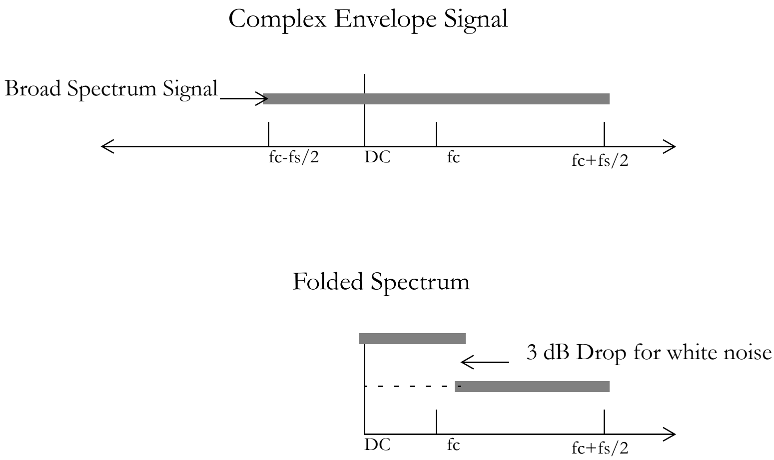


Figure 1-11. Negative frequency folding and broad spectrum signals

As Figure 1-11 illustrates, the resulting folded spectrum shows a sharp drop where the negative frequency overlap ends. This is most apparent when the signal contains white noise, since the white noise extends across the full sampling frequency band. Theoretically, the white noise should end at DC - there should not be any negative frequency content. However, in the current release white noise is always generated across the full sampling frequency band.

The *Noise Modeling in VSS* chapter of this guide contains more information on noise modeling within VSS.

1.4.5 Measuring Channel Power

Many VSS measurements use *channel power* in their computations. These include the power meter measurement PWR_MTR, the adjacent channel power measurement ACPR, and the swept AM to AM measurement AMtoAM_PS.

Channels are normally defined using a center frequency and a bandwidth. The channel center frequency should not be confused with the signal center frequency. The channel center frequency defines the center of the channel and the bandwidth defines the width. The channel frequencies are the frequencies that satisfy:

$$f_{c,\,Ch} - \frac{f_{BW}}{2} \le f < f_{c,\,Ch} + \frac{f_{BW}}{2} \tag{21}$$

Channel power is computed by first obtaining a power spectrum. The use of the DFT divides the frequency spectrum into N frequency bins, where the value of each bin represents the power within the frequency range:

$$\begin{aligned} &f_c - \frac{f_s}{2} + \frac{f_s}{N}\cdot n \le f < f_c + \frac{f_s}{2} + \frac{f_s}{N}\cdot(n+1) \qquad \text{for even } N \\ &f_c - \frac{f_s}{2} + \frac{f_s}{2N} + \frac{f_s}{N}\cdot n \le f < f_c - \frac{f_s}{2} + \frac{f_s}{2N} + \frac{f_s}{N}\cdot(n+1) \qquad \text{for odd } N \end{aligned} \tag{22}$$

for $n = 0, 1, \ldots, N-1$. Note that for $n = 0$ the lower frequency bound is the lower edge of the sampling frequency band and for $n = N-1$ the upper frequency bound is the upper edge of the sampling frequency band.

Channel power is obtained by first summing the power spectrum values for the bins that fall entirely within the channel frequency band. If one of the edge frequencies does not fall exactly on a edge of a frequency bin, power proportional to the amount of the bin occupied by the edge frequency is added to the channel power. For example, if the lower frequency edge were 3.75 Hz and it fell in the frequency bin bounded by 4 Hz and 5 Hz, 25% of the power in that bin would be included in the channel power.

When specifying channel bandwidth, VSS also supports using a single frequency bin. The "1 RBW (bin)" option for the channel bandwidth type, when available, does this.

1.5 REFERENCES

[1] Proakis, J. and Manolakis, D., *Digital Signal Processing*

RF MODELING IN VSS

2

This chapter details the RF modeling capabilities of VSS. These capabilities include:

- RF Budget Analysis
- RF Inspector
- Impedance mismatch modeling.
- Behavioral amplifier, mixer and circuit filter blocks.
- Blocks for modeling Microwave Office (MWO) linear and nonlinear circuits within VSS.

RF modeling applies to the portions of a system design that represent analog voltage signals, usually at RF frequencies but not always. These are the portions of the design that include behavioral amplifiers, mixers, and circuit based filters as well as blocks for incorporating Microwave Office circuit designs within a system simulation.

RF modeling typically only applies when using the RF blocks. These blocks include all the blocks within the **RF Blocks** category. They also include the circuit based filters, which are found in the **Bandpass**, **Bandstop**, **Highpass** and **Lowpass** sub-categories of the **Filters** category.

The "*RF Blocks In VSS*" section outlines the available blocks and offers suggestions on when to use various blocks. It also includes information on incorporating signals from non-RF blocks into RF modeling by using the RF_START and RF_END blocks.

RF modeling lets you perform quick frequency based cascaded measurements such as cascaded noise figure or operating gain using the RF Budget Analysis simulator. The RF Budget Analysis simulator is one part of the frequency analysis capability of VSS. The "*RF Budget Analysis*" section explains RF Budget Analysis simulations.

RF modeling lets you identify the makeup of frequency content at any point in an RF link using the RF Inspector simulator (RFI). With RFI you can identify sources of distortion and interference, such as what aspect of a nonlinear RF component is causing a particular IM product. The RF Inspector simulator is the other part of the frequency analysis capability of VSS. The "*RF Inspector*" section covers RFI in detail.

RF modeling allows you to take into account the effects of impedance mismatches, particularly when working with filters. The "*Impedance Mismatch Modeling*" section details these capabilities.

A key component of RF modeling is the support for nonlinear behavior. VSS includes several RF blocks for modeling nonlinear behavior. They include the behavioral amplifier blocks AMP_B, AMP_BV, AMP_F, VGA_F, the behavioral mixer blocks MIXER_B and MIXER_F, and the nonlinear MWO simulation based amplifier and mixer blocks, NL_S and MIXER_S. As with any RF model, they have limitations. These limitations are described in the "*Nonlinear Modeling Issues*" section.

2.1 RF BLOCKS IN VSS

The VSS RF blocks can be grouped into four main groups: linear filters, nonlinear amplifiers, mixers and miscellaneous.

The linear filters consist of purely linear blocks that can be modeled using S and Y parameters. These include LIN_F, LIN_S, RFATTEN, and QHYB_12, as well as the circuit filter blocks such as BPFB or LPFC.

The nonlinear amplifiers are blocks that apply a nonlinearity to the signal without shifting the frequency. These include AMP_B, AMP_BV, AMP_F, VGA_F, NL_S and NL_F.

The mixers are three port blocks that multiply two signals to obtain a third frequency shifted signal. These include MIXER_B, MIXER_F and MIXER_S.

The miscellaneous group includes blocks such as TONE and LOAD, and the network analyzer blocks VNA_LS and VSA.

Two miscellaneous blocks, RF_START and RF_END, let you incorporate signals from non-RF sources into RF modeling.

There are also several other specialized blocks such as the PLL blocks and V_LIM. Because of the nature of these blocks, they generally do not fully incorporate the RF modeling features described here and are not discussed further in this chapter.

Finally, non-RF block generated signals such as modulated signals can be used in time domain RF and RF Budget Analysis simulations through the use of the RF_START and RF_END blocks.

2.1.1 Linear Filters

The linear filters group consists of RF blocks that have linear behavior. Their behavior can be represented as S or Y matrices. This group consists of the following RF blocks:

Block	Description
CIRC_12	Circulator (1-Input, 2-Output)
CIRC_21	Circulator (2-Input, 1-Output)
DCOUPLER_3	Directional Coupler, 3-port Internal Termination
DCOUPLER_4	Directional Coupler, 4-port External Termination
ISOLATOR	Isolator
LIN_F	Linear Behavioral Model (File-Based)
LIN_S	Linear Behavioral Model (Simulation-Based)
QHYB_12	Quadrature Hybrid (1-Input, 2-Output)
QHYB_21	Quadrature Hybrid (2-Input, 1-Output)
RF_START	Start RF Signal
RFATTEN	RF Attenuator
RFSPDT_12ST	RF Single Pole Double Throw Switch (static 1 input 2 output)
RFSPDT_21ST	RF Single Pole Double Throw Switch (static 2 input 1 output)
S2P_BLK	Two Port S Parameter Block
SPLIT	Splits One Signal Into N or Fewer Signals
Y2P_BLK	Two Port Y Parameter Block
Z2P_BLK	Two Port Z Parameter Block

The circuit filter blocks in the **Bandpass**, **Bandstop**, **Highpass** and **Lowpass** sub-categories of the **Filters** category are also linear filters.

NOTE. RF_START is included in the above table because it is implemented as a linear filter. Its use, however, is described in the "*Working with Non-RF Blocks*" section.

The linear filters all utilize the same core implementation, which internally represents the filters in Y matrix form. The Y matrices are used to compute the various frequency dependent properties for frequency analysis simulations. These include linear gain, available gain, and noise generation. When impedance mismatch modeling is enabled, the Y matrices are used by the simulator to determine the impedances looking into the ports of the block.

For time domain simulations, the linear filters rely on either IIR, FIR or a combination of both IIR and FIR digital filters. Each type has its advantages and limitations. In general, IIR filters provide a better overall response for arbitrary signals, though this is not guaranteed. The FIR filters on the other hand can provide much better frequency responses provided the signal is composed of tones that fall exactly on the frequencies corresponding to the FIR taps. The FIR filter is also generally much faster for large filter orders. The combination FIR and IIR filter provides a last-ditch model, with a typically good match for the magnitude response and a rough approximation of the group delay response.

A more comprehensive discussion of issues related to time domain filtering is found in the "*Filter Issues*" section.

The blocks in the linear filters group support both frequency analysis simulations and impedance mismatch modeling. See the "*RF Budget Analysis*", "*RF Inspector*" and "*Impedance Mismatch Modeling*" sections for details.

LIN_F VERSUS LIN_S

The LIN_F and LIN_S blocks are both used to model frequency dependent S parameter based linearities. The difference between the two depends on where the S parameter data comes from.

The following summarizes when to use which block:

If the S parameter data is:	Use:
S parameters from a Microwave Office schematic	LIN_S - You can drag MWO circuits from the Subcircuits > LIN_S category of the Element Browser.

If the S parameter data is:	Use:
Touchstone format	LIN_S - import the data as S-parameter data into the **Data Files** node of the Project Browser. You can drag the Touchstone circuit from the Subcircuits > LIN_S category of the Element Browser.
Frequency response data corresponding to S, Y or Z parameters.	LIN_F - convert the data to Text Data File format and import it into the **Data Files** node of the Project Browser.

2.1.2 Amplifiers

The amplifiers group consists of the following one input-one output nonlinear amplifiers:

Block	Description
AMP_B	Behavioral Amplifier
AMP_BV	Behavioral Amplifier, Voltage Based
AMP_EQN	Equation-based Nonlinear Amplifier
AMP_F	Frequency Dependent Behavioral Amplifier, File-based
NL_F	Nonlinear Behavioral Model (File-Based)
NL_S	Nonlinear Behavioral Model (Simulation-Based)
VGA_F	Nonlinear Variable Gain Amplifier, File-based

These blocks are all found within the **RF Blocks > Amplifiers** category. There are other blocks within that category, such as LOGAMP and VGA_L. However, those blocks are currently not part of the amplifiers group.

The blocks in the amplifiers group support both frequency analysis simulations and impedance mismatch modeling. See the "*RF Budget Analysis*", "*RF Inspector*" and "*Impedance Mismatch Modeling*" sections for details.

The following summarizes when to use which block:

If you have:	Use:
General amplifier characteristics such as P1dB and IP3	AMP_B if characteristics are power based, AMP_BV if characteristics are voltage based.
Frequency dependent amplifier characteristics such as P1dB and IP3, or a combination of voltage and power based characteristics	AMP_F - AMP_F is a very flexible form of AMP_B and AMP_BV, with a variety of ways to enter the characteristics.
A Microwave Office circuit whose AM/AM and AM/PM characteristics you wish to include in a VSS simulation (the MWO circuit must sweep power)	NL_S - Allowed MWO schematics can be dragged from the Subcircuits > NL_S category of the Element Browser.
A table of AM/AM - AM/PM characteristics, including frequency dependent characteristics	NL_F - Create a Text Data File node in the **Data Files** node of the Project Browser and enter the characteristics. See the online help for NL_F for format details.

2.1.3 Mixers

The mixer group consists of the following mixer blocks:

Block	Description
MIXER_B	Behavioral Mixer
MIXER_F	File-based Behavioral Mixer
MIXER_S	Simulation-based Behavioral Mixer

The blocks in the mixers group support both frequency analysis simulations and impedance mismatch modeling. See the "*RF Budget Analysis*", "*RF Inspector*" and "*Impedance Mismatch Modeling*" sections for details.

The following summarizes when to use which block:

If you:	Use:
Do not have spur information, but may have gain conversion and IP3	MIXER_B - A spur table for a double balanced diode mixer is generated.

If you:	Use:
Have spur table information for the mixer	MIXER_F - Create a Text Data File node in the **Data Files** node of the Project Browser and enter the spur table. See the online help for MIXER_F for format details.
Have a Microwave Office mixer design you wish to include in a VSS simulation (the MWO schematic must contain at least one power sweep with a minimum of 3 values)	MIXER_S - Allowed MWO schematics can be dragged from the Subcircuits > MIXER_S category of the Element Browser.

2.1.4 Miscellaneous Blocks

The miscellaneous group includes the following blocks:

Block	Description
COMB	Combiner (N Inputs to 1 Output)
LOAD	Grounded Resistor
OSC_S	Oscillator with Optional Phase Noise Effects (Simulation-Based)
RF_END	End of RF Signal
RN	Resistor Noise Source
TONE	Tone(s) Source
TP	Test Point
VN	Voltage Noise Source
VNA_LS	Vector Network Analyzer (Large Signal, Power Sweep)
VNA_SS	Vector Network Analyzer (Small Signal)
VSA	Vector Signal Analyzer (Complex Envelope)

NOTE. VNA_SS is designed to only be used to measure the small signal response of linear circuits. It operates by sending an impulse through the device under test (DUT) and measuring the impulse response. Because the impulse's magnitude is based on the number of samples to be measured for the impulse response, it should not be passed through a nonlinear circuit because it does not reflect the operating point and would most likely be compressed.

The TP block is found in the **Meters** category, while the VNA_LS, VNA_SS and VSA blocks are found in the **Meters > Network Analyzers** category. The other blocks are in sub-categories of the **RF Blocks** category.

For impedance mismatch purposes, TP, VNA_LS, VNA_SS and VSA can all be configured as either a voltage probe with infinite impedance or as a termination.

2.1.5 Working with Non-RF Blocks

In many instances you can incorporate signals that are not generated by RF blocks within time domain RF and RF Budget Analysis simulations. You can also use the IQ and OFDM based modulated signal sources and transmitters in RF Inspector simulations. The modulated signal support in RF Inspector is described fully in the "*Modulated Signals*" section.

To use non-RF signals within a time domain RF or RF Budget Analysis simulation add an RF_START block between the non-RF signal and the first RF block. The signal must be either complex or real. If further processing is required after the RF link add an RF_END block after the last RF block. Both RF_START and RF_END are found under the **RF Blocks > Impedance Mismatch** category.

RF_START can be thought of as representing the port of a signal generator, with any block before RF_START internal to the signal generator and the signal coming out of RF_START the generated analog RF signal. Likewise, RF_END can be thought of as representing the port of a network analyzer or other measurement device. Any block after RF_END would be internal to the measuring device.

RF_START primarily provides impedance mismatch support. This includes the ability to specify S22 and the characteristic impedance of the output.

RF_END lets you specify a frequency dependent load to be seen by the preceding RF block. The output of RF_END is the total voltage seen at its input.

2.2 RF BUDGET ANALYSIS

The RF Budget Analysis (RFB) simulator is a VSS frequency domain simulator typically used to perform cascaded measurements such as cascaded gain and noise figure. RFB is invoked by the RF Budget Analysis measurements found in the **System > RF Budget Analysis** measurement category. These measurements include Cascaded Noise Factor/Figure C_NF, Cascaded IP3 C_IP3, and Cascaded Signal-to-Noise Ratio C_SNR.

RFB simulations are performed on a per frequency basis. It can be thought of as measuring the response of the RF link to a CW signal at each frequency of interest. The CW signal has the same power as the source.

NOTE. Only CW RF sources directly support RFB simulations. These sources are the TONE, OSC_S and VNA_LS blocks. Non-RF analog signals may also be used provided an RF_START block is inserted between the non-RF signal and the first RF block. The starting point of the measurement must after the RF_START block.

The set of frequencies at which the simulations are performed is defined by frequency offsets from the signal's center frequency. These offsets are specified in the **Frequency Analysis** tab of the System Simulator Options dialog box of system diagrams. Offsets to the center frequency rather than absolute frequencies are used to simplify the tracking of frequencies when signals pass through mixers.

The signal properties are assumed to represent steady state conditions. Because they are steady state, RFB simulations tend to be much faster than time domain simulations. This lets you tune, optimize, and perform yield analysis quickly. You can optimize the noise figures and gains within an RF link using RF Budget Analysis, then run a time domain simulation to see the effects on a modulated signal (there are some limitations, particularly with noise modeling, see the *Noise Modeling in VSS* chapter for details).

Some of the frequency dependent properties computed within RF Budget Analysis include:

- *Operating point gain*
- *Noise temperatures*
- *Noise gain*
- *IP3*

2.2.1 RF Budget Analysis Simulation Basics

RF Budget Analysis simulations are performed using *frequency sets*. A frequency set represents the set of frequency dependent *static signal properties* associated with a node. Each RF output port shares a frequency set with all connected RF input ports.

One of the first steps in an RFB simulation is to determine the frequencies for each of these frequency sets. At a minimum, each set contains the frequencies at the frequency offsets from the center frequency of the signal used for the measurement. Blocks may modify or add additional frequencies as needed.

For example, mixers normally shift the input frequencies for their output frequency set and add the image frequencies to their input frequency set to compute the effects of image noise. The simulator ensures that the frequencies are consistent along the signal path.

If impedance mismatch modeling is enabled, the next step is to resolve the impedances seen at the various ports. If impedance mismatch modeling is not enabled the characteristic system impedance _Z0 is seen at all ports.

NOTE. Impedance mismatch modeling should generally be enabled when performing RF Budget Analysis measurements. RF Budget Analysis measurements tend to follow equivalent Microwave Office circuit measurements more closely when impedance mismatch modeling is enabled.

The computation of the various static signal properties in the frequency sets occurs as part of parameter propagation. During parameter propagation each block is visited to allow it to update the various signal properties. This occurs for both the time and frequency domain simulations. This way frequency dependent properties are available to blocks for both time and frequency domain simulations.

2.2.2 Frequency Dependent Signal Properties

This section describes several of the signal properties used in RFB simulations and how they are modified by typical RF blocks.

OPERATING POINT GAIN

The operating point gain property is actually determined from the change in a frequency dependent operating point power level property. The use of operating point power level lets nonlinear blocks determine the operating point gain.

Linear blocks simply scale the operating point power level property by the power gain they apply. Nonlinear blocks, on the other hand, use a more elaborate algorithm for determining operating point gain.

At each frequency point, the nonlinear block creates a set of data samples whose average power matches the operating point power level at the input. For real signals the data samples represent a single period of a sine wave. For complex signals the data samples are evenly distributed on a circle in the real/imaginary plane, representing a single period of a complex sinusoid.

These data samples are then passed through the nonlinearity without applying any memory effects such as filters or resampling. The average power of the output samples is then used as the output operating point power level. The operating point gain is the ratio of the operating point power level at an output to the operating point power level at an input.

The operating point gain at a given frequency therefore represents the operating point gain if a single tone at that frequency were passed through the RF link.

NOISE TEMPERATURE

The noise temperature property represents the thermal noise present at a node under matched conditions. The change in noise temperature along with the change in noise power gain can be used to compute cascaded noise figure.

In VSS 7.5 only the RF tone sources TONE, OSC_S and VNA_LS, the RF noise sources RN and VN set the initial noise temperature in an RF link. Without one of these blocks, the noise temperature is 0K until an RF block adds noise. However, the cascaded noise RF budget analysis measurements such as Cascaded Noise Figure/Factor C_NF and Cascaded Equivalent Input Noise Temperature C_TE will compensate for this by assuming an input noise temperature of 290K and adding the equivalent noise with the noise power gain applied to the output noise temperature to allow the measurements to compute their values.

Blocks normally adjust the noise temperature according to the following:

$$T_{Out} = T_{In} \cdot G_{NPwr} + T_N \tag{1}$$

where G_{NPwr} is the noise power gain property and T_N is the equivalent output noise temperature for the block.

For blocks with an explicit noise figure parameter, T_N is computed using the block's linear gain and the standard temperature of 290K:

$$T_N = (F_{DS} - 1) \cdot G_L \cdot 290 \tag{2}$$

F_{DS} is the double-sided noise factor.

Linear blocks compute T_N from their noise correlation matrix.

Note that VSS treats T_N as independent of the input noise temperature.

The "*Noise Modeling in VSS*" chapter covers noise modeling in detail.

NOISE POWER GAIN

The noise power gain property is used in computing cascaded noise figure. It represents the gain that would be applied to the thermal noise at the input if the output were impedance matched. The noise power gain property is initially set to 1. Blocks then apply their computed noise power gain to the property to obtain the noise power gain associated with their output.

For linear blocks, the noise power gain can be computed directly from the S parameters and reflection coefficients as available power gain:

$$G_{NPwr} = \frac{\left|1 - |\Gamma_S|^2\right|}{|1 - (S11 \cdot \Gamma_S)|^2 \cdot (1 - |\Gamma_{Out}|^2)} \cdot |S21|^2 \qquad (3)$$

Nonlinear blocks must estimate their noise power gain. They do this by linearizing their nonlinearity at an approximated noise power operating point. At a given frequency, this operating point is set to:

$$N_f = T_f \cdot k \cdot f_s \qquad (4)$$

where T_f is the noise temperature at frequency *f*. The full sampling frequency bandwidth is used to attempt to reflect the gain that would be applied to a time domain noise signal.

The linearization of the nonlinearity involves estimating S21 at the noise power operating point, which is based on the same mechanism used to compute operating point gain. The noise power gain is then set to the available power gain using the estimated S21 in equation 3.

IP3

The IP3 property represents the third order intercept point at the output of a block relative to the signal source. RFB simulations use output referred IP3

(OIP3) for their computations. For a signal without any nonlinearities, OIP3 is infinite. OIP3 at the output of a source block is infinite.

Nonlinear blocks that support OIP3 adjust the OIP3 property. If OIP3 at the input is infinite, the output OIP3 can simply be set to the OIP3 of the block. If OIP3 at the input is not infinite, the process is more involved.

OIP3 is defined as the output power level where the linear output power level matches the extrapolated 3rd order output power level, or in equation form:

$$OIP3 = P_{In,\, dB} + G_{dB} = 3 \cdot P_{In,\, dB} + b_{IP3,\, dB} = P_{Out3,\, dB} \tag{5}$$

where $P_{In,\, dB}$ is the input power level, G_{dB} is the linear gain, and $b_{IP3,\, dB}$ is a constant. By definition compression is ignored.

Solving for $b_{IP3,\, dB}$ we have:

$$b_{IP3,\, dB} = 3 \cdot G_{dB} - 2 \cdot OIP3 \tag{6}$$

For a given $P_{In,\, dB}$ the 3rd order output power level is:

$$P_{Out3,\, dB} = 3 \cdot P_{In,\, dB} + b_{IP3,\, dB} \tag{7}$$

The relationship between OIP3, gain, input power and 3rd order output power is then:

$$OIP3 = \frac{3 \cdot (G_{dB} + P_{In,\, dB}) - P_{Out3,\, dB}}{2} \tag{8}$$

The effect of a nonlinear block with OIP3 of $OIP3_{Blk}$ on the OIP3 property can then be computed by determining $P_{Out3,\, dB}$ at the output of the block. There are two components to $P_{Out3,\, dB}$: the OIP3 contribution from the block and the 3rd order power level at the input of the block. To apply voltage corrections due to impedance mismatches, we convert to voltage.

We then have for the OIP3 contribution of the block:

$$P_{OIP3,\,Blk} = \frac{\left(G_P \cdot \dfrac{Z_{L,\,Out}}{Z_{L,\,Inp} \cdot \left|Corr_{Inp} \cdot Corr_{Out}\right|^2} \cdot P_{In} \cdot \left|Corr_{Inp}\right|^2 \cdot \dfrac{Z_{L,\,Inp}}{Z_0}\right)^3}{(OIP3_{Blk,\,Lin})^2} \tag{9}$$

$$V_{OIP3,\,Blk} = \sqrt{P_{OIP3,\,Blk} \cdot 2 \cdot Z_0} \cdot \left|Corr_{Out}\right| \tag{10}$$

For the contribution from the 3rd order power level at the input of the block we have:

$$P_{In3} = \frac{P_{In}^3}{OIP3_{In,\,Lin}^2} \tag{11}$$

$$V_{In3,\,Blk} = \sqrt{G_P \cdot P_{In3} \cdot 2 \cdot Z_{L,\,Out}} \tag{12}$$

$P_{Out3,\,dB}$ is then:

$$P_{Out3,\,Lin} = \frac{(V_{OIP3,\,Blk} + V_{In3,\,Blk})^2}{2 \cdot Z_{L,\,Out}} \tag{13}$$

and OIP3 is:

$$OIP3 = 10 \cdot \log 10\left(\sqrt{\frac{(G_P \cdot P_{In})^3}{P_{Out3,\,Lin}}}\right) \tag{14}$$

2.2.3 Additional Notes

The following are some items to note when working with RF Budget Analysis.

PRIMARY DATA PATHS

The cascaded RF budget analysis measurements such as C_NF or C_GA require two test points to perform their measurement. The first test point identifies the reference point for the cascaded measurement. The second test point identifies the end of the cascaded measurement. The second test point must be downstream from the block that connects to the first test point.

In the current release, the test points are further restricted to lie on a primary data path. The primary data path defines the main signal of a link. In many cases this is straightforward. However, when a signal is the result of the combination of two or more input signals, such as in a combiner block, only one of the input signals represents the primary data path.

The port corresponding to the primary data path can usually be explicitly set through a PRIMINP parameter in the block. In many cases VSS can also determine the primary data path based on the properties of the input signals. VSS uses the following rules when automatically determining the primary data path:

- Modulated signals are given the highest priority.
- If all else is equal, the signal at the port with the smallest node number is given the highest priority.

UNSUPPORTED TOPOLOGIES

Because of the primary data path restriction, RF Budget Analysis does not in general support RF link topologies that contain loops. This includes the following:

- Feedback loops.
- Splitting then recombining a signal. For example using QHYB_12 to separate the signal into I and Q channels, processing, then recombining the channels with QHYB_21 is not supported.

Switched branches using RFSPDT_12 and RFSPDT_21 to switch separate paths on or off are supported.

MIXERS

The mixer blocks MIXER_B, MIXER_F and MIXER_S include image noise when they compute the noise temperature at the output of the mixer. To do so, the mixer determines the two input frequencies that contribute to each output frequency being calculated. For example, if the LO center frequency is 10 GHz and the output center frequency is 1 GHz, then the two fundamental input frequencies that contribute to the output frequency 1.1 GHz are 11.1 GHz (1,1) and 8.9 GHz (1,-1). On the other hand, if the output center frequency is 12 GHz then the two fundamental input frequencies that contribute to it are 2 GHz (1,1) and 22 GHz (-1,1).

When the mixer computes the output noise temperature, it applies the appropriate conversion gain to the noise temperature at each input noise frequency, applies the input to output feedthrough to the input noise with the output frequency, applies the LO to output feedthrough to the LO noise with the output feedthrough, sums the four, and adds the noise added by the mixer itself to obtain the output noise temperature:

$$\begin{aligned} T_{Out}(f) = {} & T_{Inp}(|f-f_{LO}|) \cdot G_{Conv}(f_{LO}, |f-f_{LO}|) \\ & + T_{Inp}(f+f_{LO}) \cdot G_{Conv}(f_{LO}, f+f_{LO}) \\ & + T_{Inp}(f) \cdot G_{Inp2Out} + T_{LO}(f) \cdot G_{LO2Out} \\ & + T_N \end{aligned} \tag{15}$$

Note that if the input signal is band pass filtered to attenuate the image, the effect of the attenuation will be reflected in the noise temperature computation:

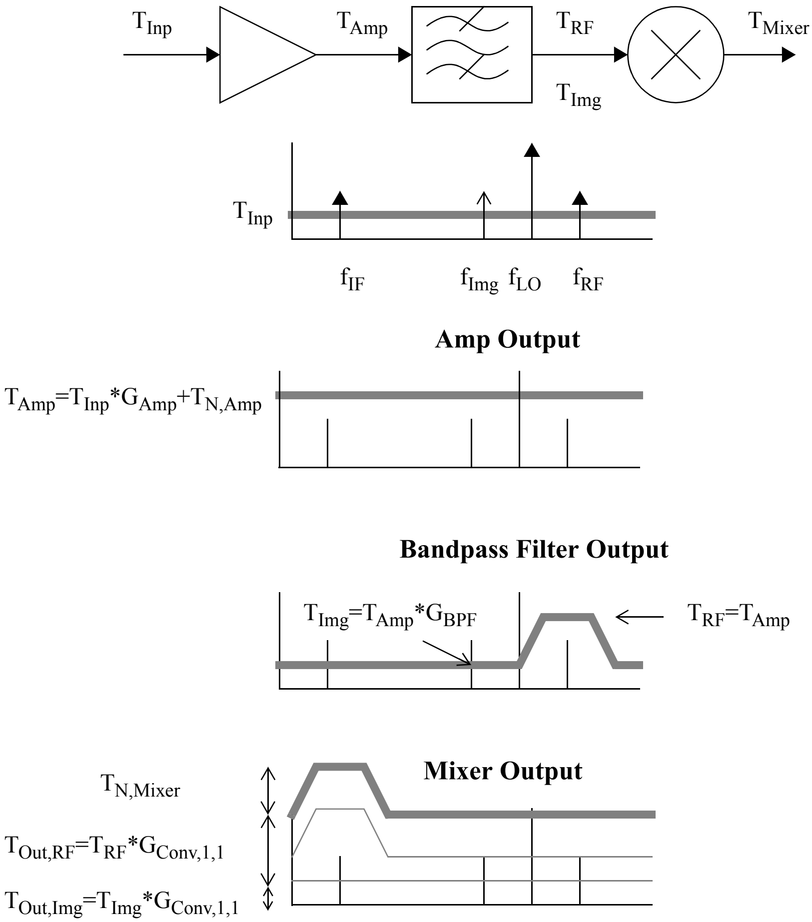


Figure 2-1. Image noise in a mixer

When computing noise power gain, however, only the gain from the signal sideband is used. The image is not part of the noise power gain computation.

The cascaded noise figure measurement C_NF displays the single sided noise figure when measuring noise figure across a mixer. This is because the image is not considered as part of the input noise in the noise figure equation:

$$F = \frac{T_{Out}}{T_{Inp} \cdot GConv} \tag{16}$$

DIFFERENCES FROM TIME DOMAIN SIMULATIONS

When comparing results from RF Budget Analysis and time domain simulations, you may notice some discrepancies between the two results. These discrepancies are, in general, the result of time domain filter limitations.

The filters used to implement frequency dependent behavior in the time domain have limitations as to how well they can match the desired frequency response. The "*Filter Issues*" section of the guide details these limitations.

2.3 RF INSPECTOR

RF Inspector is a VSS frequency domain tool that lets you inspect frequency content anywhere along an RF link. With RF inspector you can:

- Determine the various contributors to the frequency content.
- Identify the path of the contributors.
- Examine the voltage, current and power of each contributor.
- Classify the contributors as signal of interest, distortion or interference.

RF Inspector uses the RF Inspector simulator (RFI). It is invoked by using any of the measurements in the **System > RF Inspector** measurement category.

The most commonly used measurements are the power and voltage spectrum measurements:

- RFI Power Spectrum (RFI_PWR_SPEC)
- RFI Voltage Spectrum (RFI_V_SPEC)

These measurements display the power and voltage spectrum graphs, respectively. With these graphs you can select individual spectral components for further inspection. By pressing and holding the mouse button down on a spectral component, you can view summary information about the selected component:

3.1 GHz
-3.19 dBm
Full spectrum
-3.06 dBm AMP_B.A2@2: LINEAR 3.1 GHz (LINEAR) (LINEAR)
-39.4 dBm AMP_B.A2@2: IM 3.1 GHz (IM 2 (1,-1)) (IM 2 (1,-1)) (IM 2 (1,-1))
(IM 2 (1,1)) (IM 2 (1,-1)) (IM 3 (0,1)) (IM 3 (2,-1))
(...)

Figure 2-2. Summary Information in RF Inspector

Figure 2-2 shows the summary information presented when the 3.1 GHz tone in the Chain Spectrum graph of the VSS example project Visual_System_Simulator\RF_Inspector\Chain.emp is selected with the cursor. The information indicates the tone has a power level of -3.19 dBm, of which -3.06 dBm is from a 3.1 GHz tone passing linearly through AMP_B.A2 and -39.4 dBm is from IM products falling at 3.1 GHz generated by AMP_B.A2. The “IM 2(x,y)” items indicate some of the types of IM products, in this case several 2nd order IM products contributed.

Double clicking on the same 3.1 GHz tone opens up the RF Inspector dialog box for the measurement data, which contains detailed information about the content of the selected spectrum component:

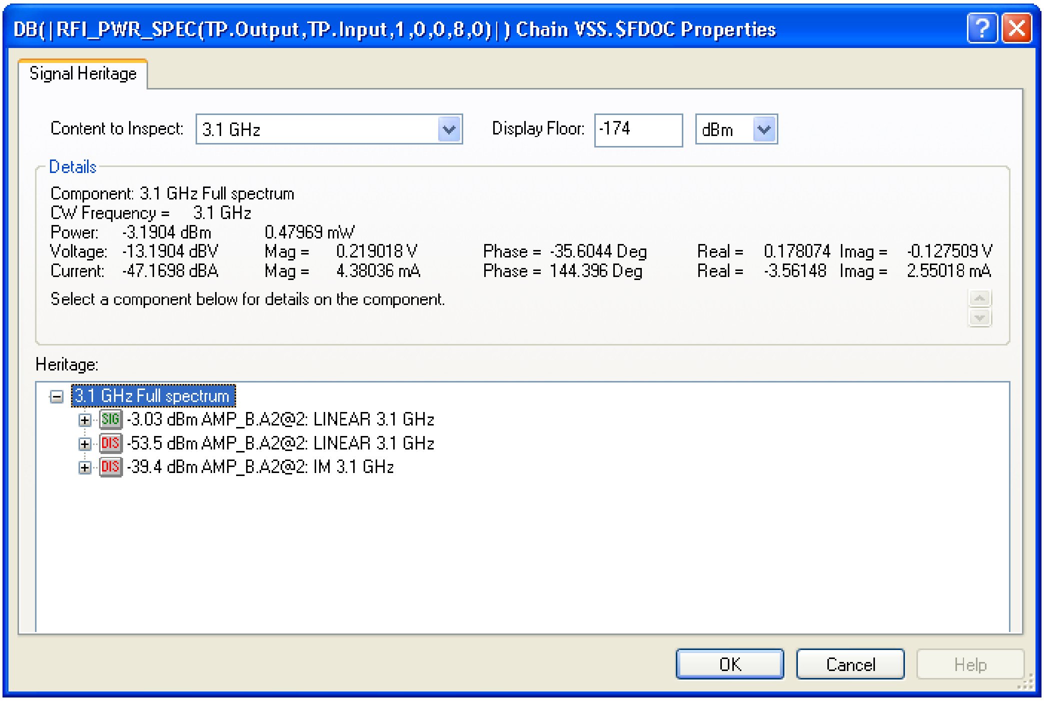


Figure 2-3. Signal Heritage Tab of the RF Inspector window

2.3.1 RF Inspector Dialog Box

The RF Inspector dialog box contains several sections. At the very top are controls for selecting different spectral components and for filtering the results shown in the **Heritage Window**.

The **Content to Inspect** control lists the different spectral components available for inspection in the window. Selecting a different component updates the **Details** section and **Heritage Window** appropriately.

The **Display Floor** controls let you set a minimum threshold for the contributors displayed in the **Heritage Window**. This is quite useful when the signal passes through several nonlinearities, as quite often there are a large number of contributors with very small values compared to the overall signal and these contributors can be ignored.

DETAILS SECTION

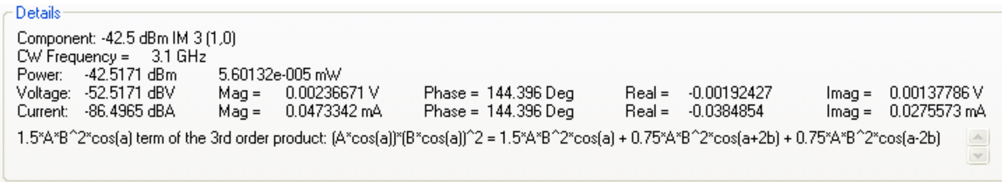


Figure 2-4. Details section with an IM product of a tone selected.

The **Details** section provides detailed information on the item selected in the **Heritage Window**. In Figure 2-4, the -42.5 dBm IM 3 product has been selected. The upper portion of the section displays component details such as power, voltage and current information for the contribution.

The lower portion identifies, when the information is available, the equation responsible for the contribution.

In this case, the contribution is due to the $(A \cdot \cos(a)) \cdot (B \cdot \cos(b))^2$ 3rd order product of two tones $A \cdot \cos(a)$ and $B \cdot \cos(b)$, which expands to:

$$\frac{3}{2} \cdot A \cdot B^2 \cdot \cos(a) + \frac{3}{4} \cdot A \cdot B^2 \cdot \cos(a + 2 \cdot b) + \frac{3}{4} \cdot A \cdot B^2 \cdot \cos(a - 2 \cdot b) \qquad (17)$$

This particular contribution is from the $\frac{3}{2} \cdot A \cdot B^2 \cdot \cos(a)$ term of (17).

The two tones for this particular contribution can be viewed by opening the -42.5 dBm IM 3 (1,0) item in the **Heritage Window**:

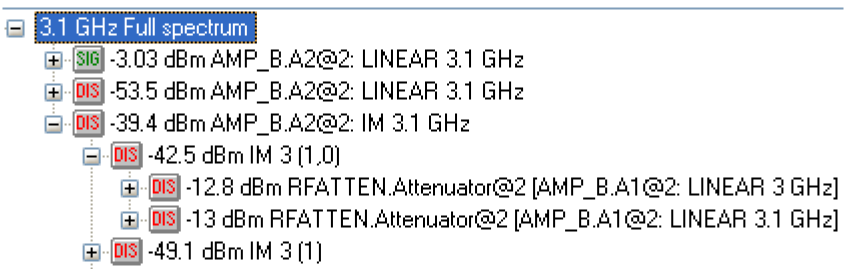


Figure 2-5. Tones contributing to -42.5 dBm IM 3(1,0)

Here the two tones are a -12.8 dBm tone at 3 GHz and a -13 dBm tone at 3.1 GHz.

The contents of each portion of the **Details** section can be copied to the clipboard. You can right click on the text in the section, choose **Select All** from the popup menu, then right click again and choose **Copy**. You can also use the cursor to directly select the text, then right click and choose **Copy** from the popup menu. The third technique is to right click in the **Heritage Window** then choose **Copy > Component Details** or **Coefficient Details** from the popup menu.

HERITAGE WINDOW

The lower section of the RF Inspector dialog box is used to browse the heritage of the selected spectral component. You can trace the various contributors all the way back to their sources.

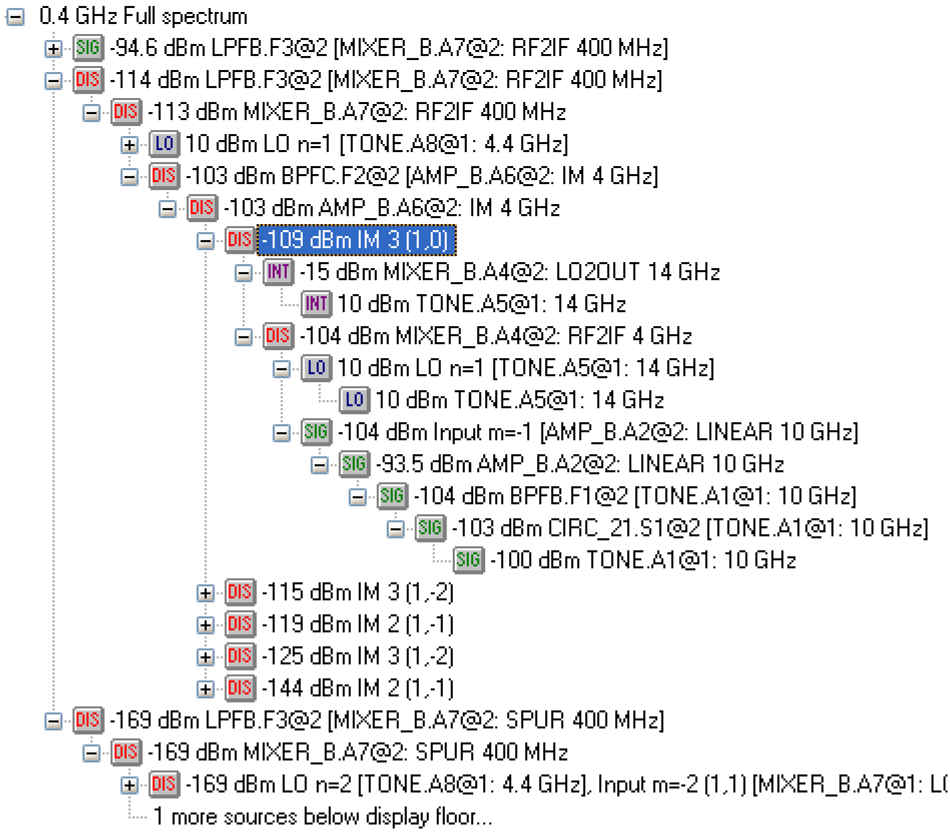


Figure 2-6. Tracing contributions back to source

Figure 2-6 illustrates the path back to several of the sources contributing to the -112 dBm 400 MHz tone. Figure 2-7 shows the system diagram used to generate the signal. Note that this diagram is not intended to necessarily represent the layout of an actual receiver, but is designed to illustrates features of the **Heritage Window**.

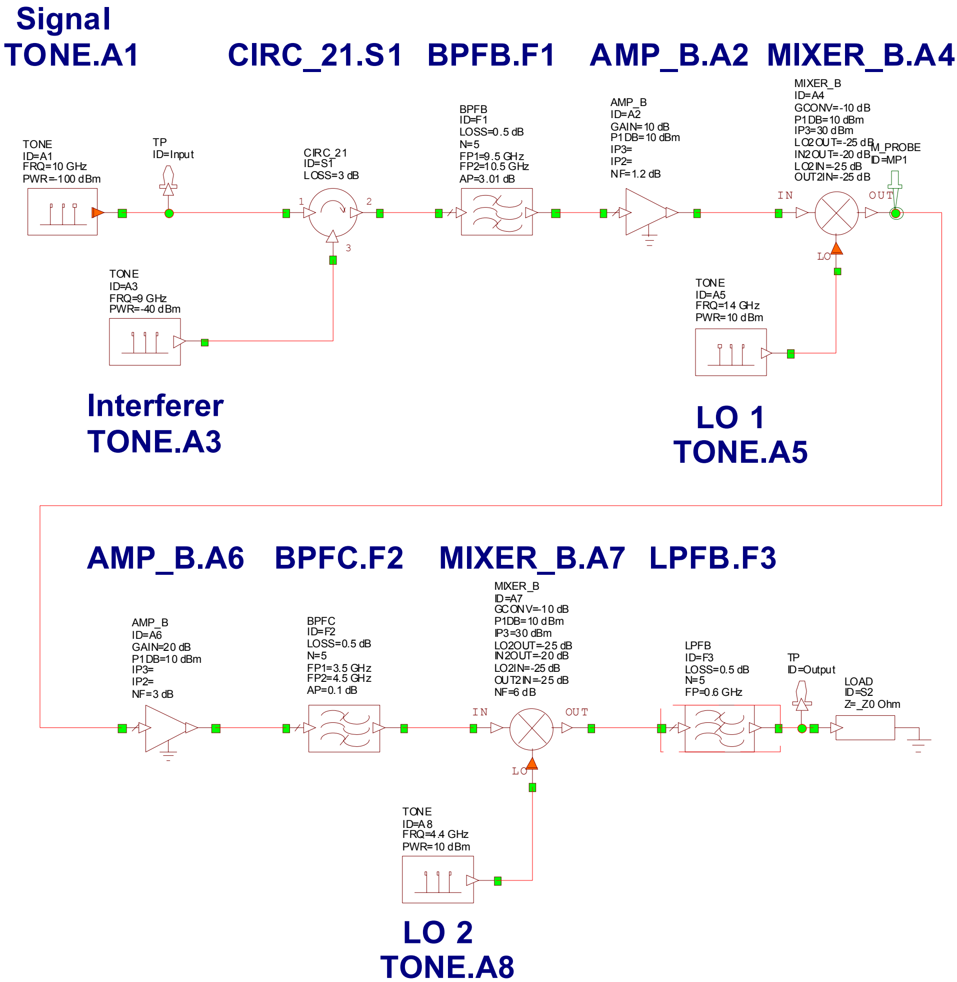


Figure 2-7. Example double down conversion receiver with interferer.

In Figure 2-6 the highlighted item "`-114 dBm LPFB.F3@2 [MIXER_B.A7@2: RF2IF 400 MHz]`" refers to a -114 dBm tone coming from port 2 of block LPFB.F3, which is the last filter in the system diagram. Since we are examining the 400 MHz tone, this tone is at 400 MHz.

The text in brackets, "`[MIXER_B.A7@2: RF2IF 400 MHz]`", indicates the source of that signal relative to LPFB.F3. In this case, it is a 400 MHz tone generated at port 2 of the mixer MIXER_B.A7. This tone is part of the (1,-1) RF to IF downconversion, as indicated by "`RF2IF`".

The next item, "`-113 dBm MIXER_B.A7@2: RF2IF 400 MHz`", tells you that the contribution at the output of the mixer is -113 dBm.

The next two items identify how the mixer generated this particular -113 dBm tone at 400 MHz. The first item identifies the LO, which is a 10 dBm tone at 4.4 GHz generated by TONE.A8. The second item identifies the input, which is a -103 dBm tone at 4 GHz coming from port 2 of BPFC.F2.

Each nested level of items represents the contributors to the item above.

Skipping to the item "`-103 dBm AMP_B.A6@2: IM 4 GHz`" we see contributions from the nonlinear amplifier AMP_B.A6. This line indicates that this amplifier is generating a -103 dBm tone at 4 GHz that is due to IM products. The items at the next nested level, such as "`-109 dBm IM 3 (1,0)`", "`-115 dBm IM 3 (1,-2)`" and "`-119 dBm IM 2 (1,-1)`", identify the individual IM products generated by the amplifier for this particular contribution.

The item "`-109 dBm IM 3 (1,0)`" indicates a -109 dBm 3rd order IM product for two tones with the frequency multipliers 1 and 0. The term from the 2 tone 3rd order equation is displayed in the **Details** section when the item is selected. In this case the following text is displayed:

```
1.5*A*B^2*cos(a) term of the 3rd order product:
(A*cos(a))*(B*cos(a))^2 = 1.5*A*B^2*cos(a) +
0.75*A*B^2*cos(a+2b) + 0.75*A*B^2*cos(a-2b)
```

If you continue proceeding through the nested items of "`-109 dBm IM 3 (1,0)`" you will see that this particular contribution is comes from TONE.A5 and TONE.A1, where the LO TONE.A5 contributes two ways:

- More strongly as LO to output feedthrough at MIXER_B.A4.
- Less strongly as the LO for a mixing product in MIXER_B.A4.

SIGNAL CLASSIFICATION

If you look once again at Figure 2-6, notice that each contributor item has a small icon to its left. The icons indicate the classification assigned to the contribution by RFI. There are four classifications:

-  Signal: The signal of interest, which is indicated by the **Test Point Identifying Signal** setting of the measurement.

- Distortion: IM products, spurs, or any other unwanted contributions that is in any part generated by the Signal.
- Interference: Any contribution that is not Signal, Distortion or LO.
- LO: The LO input to a mixer.

Each RF Inspector measurement lets you select which classifications are to be included in the measurement's output. For the spectrum measurements, selecting multiple classifications displays a separate trace for each classification.

The Signal (signal of interest) consists of all the spectral components found at the point indicated by the **Test Point Identifying Signal** setting of the measurement that have only been through linear transformations. For example, if the setting points to a location after a combiner, the Signal will include signals from all the inputs of the combiner.

When considering Signals, the spectral components passing through mixers are further restricted to only those products that match the conversion indicated by the MODE parameter of the mixer. For example, if the MODE parameter is set to "SUM", then only the spectral components generated by the (1,1) conversion may be considered a Signal. Likewise, if MODE is set to "DIFF" then only the spectral components generated by the (1,-1) conversion may be considered a Signal.

Note that any potential Signal that is not classified as a Signal is classified as Interference. Using a combiner as any example once again, if the **Test Point Identifying Signal** setting points to a location on one of the inputs of a combiner, the spectral components on the other inputs that could potentially be classified as Signals are considered Interference.

RF Inspector tracks spectral components as they pass through the various blocks of the RF link. Spectral components that pass through linear blocks, the linear portion of nonlinear amplifiers, or the conversion indicated by the MODE parameter of mixers retain their classification.

The IM products generated by nonlinear amplifiers and the spurs and feedthroughs generated by mixers are classified as either Distortion or Interference. Spectral components are classified as Distortion if any of the contributions responsible for the product is classified as Signal or Distortion. All other spectral components are classified as Interference.

Simply put, if the Signal is turned off, the Signal and Distortion disappear, while the Interference remains.

The LO classification is used to quickly distinguish the LO item from the input item when viewing the contributors of a mixer.

OTHER FEATURES

At each level of nesting the items are listed in order of decreasing value for the measurements. For RFI_PWR_SPEC it is decreasing power, for RFI_V_SPEC it is decreasing voltage magnitude.

Selecting items in the **Heritage Window** highlights and centers the item's block in the corresponding System Diagram.

Selecting an item in the Heritage Window then right-clicking it displays a menu. with the following options:

Next Frequency, **Previous Frequency**: Advance to the next or previous frequency in the **Content to Inspect** control at the top of the inspector page.

Copy > Item: Copies the text of the selected item to the clipboard.

Copy > Branch: Copies the text of the selected item and all items nested within it to the clipboard.

Copy > All Items: Copies all the items to the clipboard.

Copy > Component Details: Copies the text of the Component Details portion of the Details section to the clipboard.

Copy > Coefficient Details: Copies the text of the Coefficient Details portion of the Details section to the clipboard. This is grayed out if nothing is displayed in that portion.

Expand Branch: Opens up all the nested items within the selected item, including items nested within those items.

Collapse: Collapses the nested items of the selected item, hiding them from view.

Note that the **Copy > Branch**, **Copy > All Items**, and **Expand Branch** commands may take a significant amount of time depending on the number of nested items. If it takes more than approximately 10 seconds to complete the command a dialog box will appear allowing you to cancel the command. The dialog box will reappear every 10 seconds to give you the opportunity to cancel once again.

When an item is copied to the clipboard, the classification is included as prefix text: SIG for Signal, DIS for Distortion, INT for Interference and LO for LO.

2.3.2 Modulated Signals

RF Inspector supports modulated signals that are based on the I/Q Modulator block IQ_MOD and the OFDM Modulator block OFDM_MOD. The following are the supported blocks:

Block	Description
BPSK_SRC	BPSK Modulated Signal
BPSK_TX	BPSK Transmitter
IQ_MOD	I/Q Modulator
MPSK_SRC	MPSK Modulated Signal
MPSK_TX	MPSK Transmitter
MSK_SRC	MSK Modulated Signal
MSK_TX	MSK Transmitter
OFDM_MOD	OFDM Modulator
OQPSK_SRC	Offset QPSK Modulated Signal
OQPSK_TX	Offset QPSK Transmitter
PAM_SRC	PAM Modulated Signal
PAM_TX	PAM Transmitter
PQPSK_SRC	Pi/4 Modulated Signal
PQPSK_TX	Pi/4 Transmitter
QAM_SRC	QAM Modulated Signal
QAM_TX	QAM Transmitter
QPSK_SRC	QPSK Modulated Signal
QPSK_TX	QPSK Transmitter

These blocks are all found within the **Modulation** category. There are other source and transmitter blocks within that category, such as AM_MOD, FM_MOD and FSK_SRC. However, those blocks do not currently support RF modeling.

Because RF Inspector utilizes a frequency domain simulator, only an approximation of the spectral shape of the modulated signal is possible. Since RF Inspector's primary task is to be able to identify contributors to spectral content throughout an RF link, the modeled signal is band limited - the side lobes are not modeled. This can be seen in Figure 2-8 and Figure 2-9.

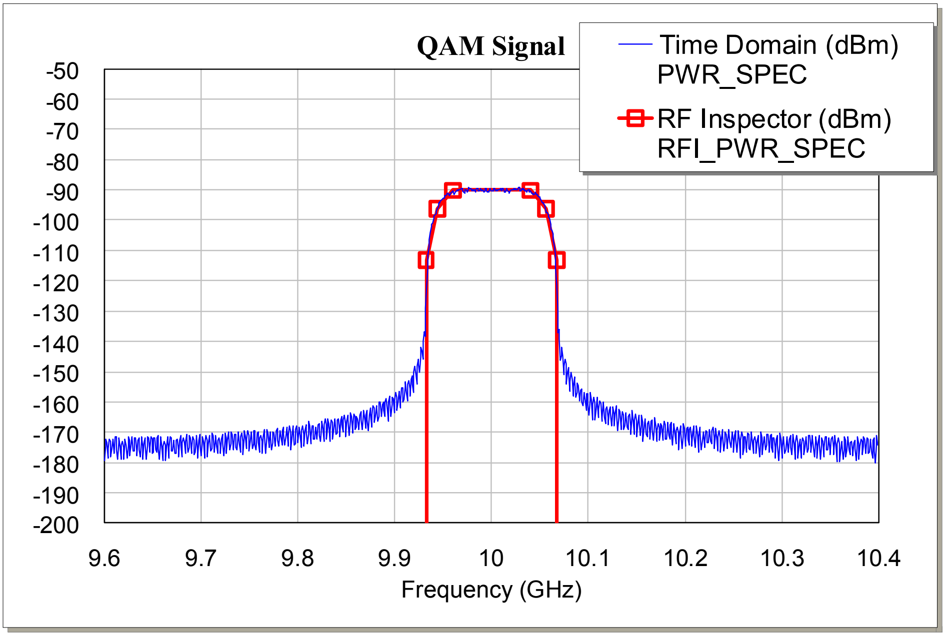


Figure 2-8. Time domain versus RF Inspector measurements of a QAM Signal

In Figure 2-8 the red curve shows a QAM signal modeled by RFI. The blue curve is the signal measured in the time domain with the PWR_SPEC measurement. The QAM signal was generated by the QAM Modulated Source block QAM_SRC with a data rate of 0.1 GHz, center frequency of 10 GHz, and root raised cosine pulse shaping with alpha of 0.35. Both measurements are displaying power spectral density (PSD) in dBm/Hz.

In Figure 2-9 the red curve shows an OFDM signal modeled by RFI while the blue curve is the time domain measurement. The OFDM signal was generated using 256 subcarriers, subcarrier spacing of 0.1/256 GHz, and a guard interval of 1/8. No windowing was applied.

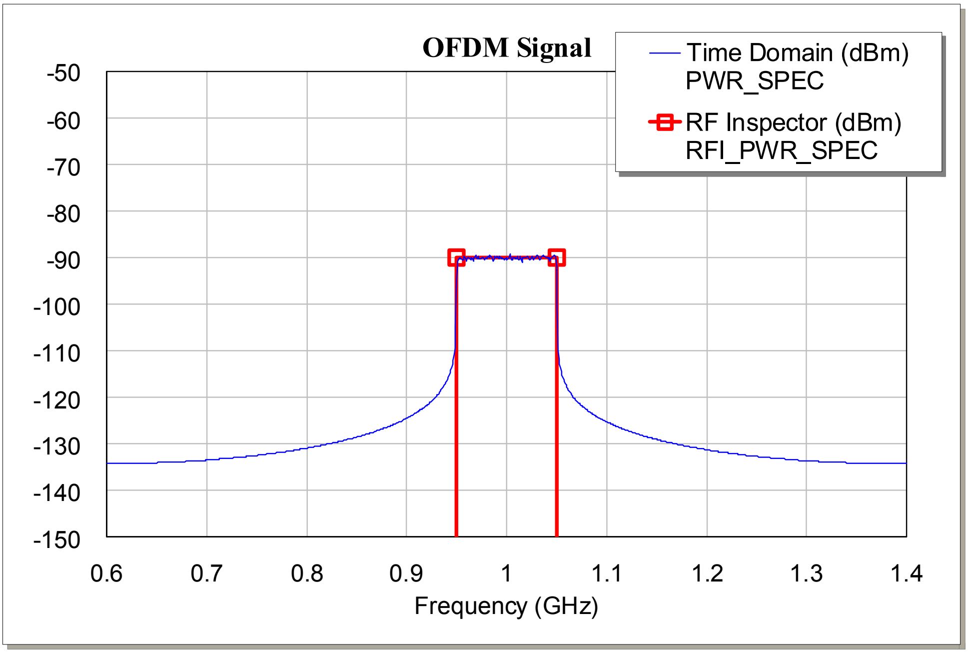


Figure 2-9. Time domain versus RF Inspector measurements of an OFDM signal.

For IQ based modulated signals, RFI uses six frequency points to represent the power distribution within the main lobe. In Figure 2-10 f1 through f6 represent the six frequency points. PSD1 through PSD6 are the power spectral density values assigned to the six frequency points. The profile is symmetric, so PSD4=PSD3, PSD5=PSD2, PSD6=PSD1, f6-f5=f2-f1 and f5-f4=f3-f2.

The PSD values are computed such that the total power in the profile:

$$P_T = \sum_{i=1}^{5} \frac{(PSD[i] + PSD[i+1])}{2} \cdot (f[i+2] - f[i]) \qquad (18)$$

is the total power of the modulated signal. This includes power in the sidelobes.

To simplify the simulation, the same profile is used for all IQ based modulated signals, regardless of the symbol mapping (such as PAM, QPSK or QAM) and the pulse shaping. The profile used is based on a QAM signal with root raised cosine pulse shaping and an alpha of 0.35.

For OFDM based modulated signals, RFI uses a flat two point profile, which is clearly visible in Figure 2-9.

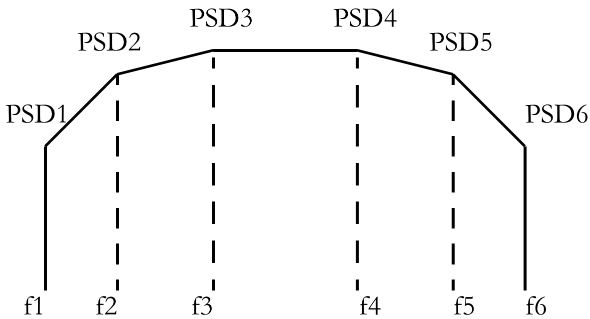


Figure 2-10. Profile of IQ modulated signals in RF Inspector

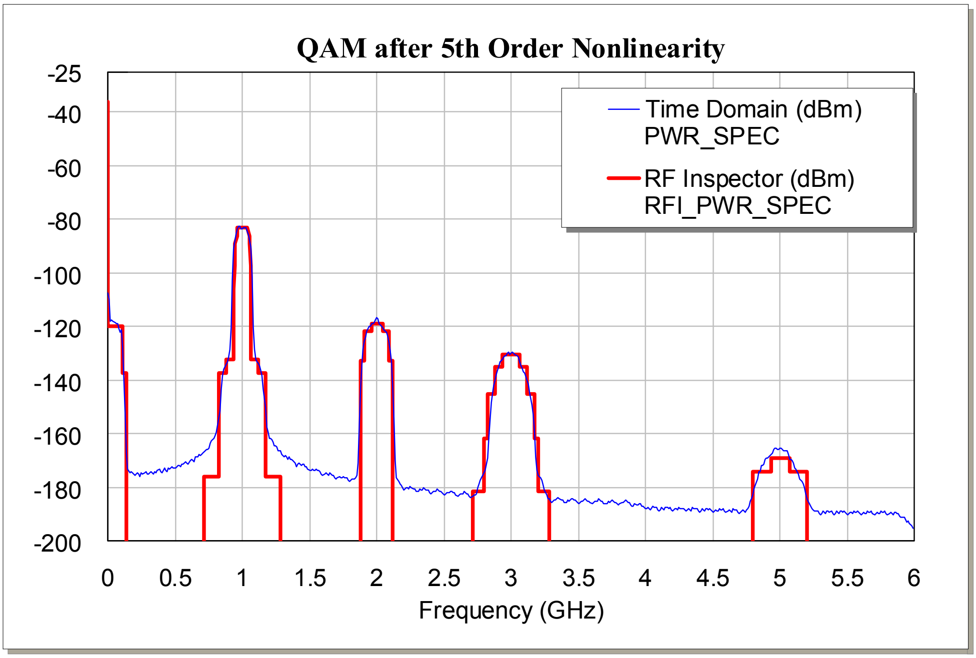


Figure 2-11. QAM signal after 5th order nonlinear amplifier.

The IQ and OFDM profiles are maintained as long as the modulated signal remains linear. Once the signal undergoes a nonlinearity, the profile is reduced to a set of overlapping 'brick wall' segments. This can be seen in Figure 2-11, where a QAM signal has been passed through a 5th order nonlinear amplifier.

The segments are stacked, as illustrated in Figure 2-12. The total power of the stacked segments roughly approximates the total power in the modulated signal at the given harmonic or IM product. This power can give you an idea of the magnitude of the IM product, but should not be used as an exact measurement of the power, particularly if the signal has passed through several nonlinearities.

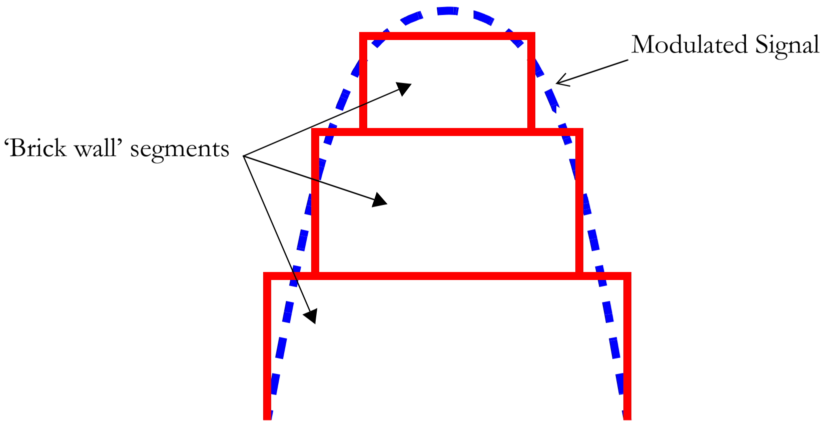


Figure 2-12. 'Brick wall' segments

CW AND MODULATED SIGNAL DISPLAY

The RFI_PWR_SPEC measurement displays its values as power spectral density (PSD), or power per Hertz. For CW signals it simply uses 1 Hz so the power displayed is the same as the power for the signal. For modulated signals the total power in a brick wall segment is divided by the bandwidth of the segment to obtain the PSD value. The net effect is RFI_PWR_SPEC measures PSD using an equivalent resolution bandwidth of 1 Hz.

This must be taken into account when comparing RFI and time domain power spectrum results for signals that contain both CW and modulated components. In most cases, the bandwidth of the signal is large enough that a 1Hz RBW is not practical for the time domain measurement.

When measuring the modulated signal components, this is not an issue if the time domain measurement is configured with the **Y-Axis Output** setting set to "PSD (Pwr/Hz)". The scaling will be similar to the RFI measurement.

When measuring CW signals, this is not an issue if the time domain measurement is configured with the **Y-Axis Output** setting set to “Spectrum Analyzer Spectrum”.

However, when displaying a combination of modulated and CW signals (a modulated signal passed through a 2nd or 4th order nonlinearity will generate a DC term) this becomes a problem. In this situation either the modulated signals or the CW signals will be displayed with the correct scaling, but not both.

When the time domain measurement is set to “PSD (Pwr/Hz)” the CW portion of the signal is usually displayed with a much lower power level than seen in the RFI measurement. This is because the measurement must use its RBW to convert power per FFT bin to power per Hertz. The RBW will typically be much larger than 1 Hz and CW signals will appear with a much lower power level since the power of the CW signal is divided by the RBW and not 1 Hz.

When the time domain measurement is set to “Spectrum Analyzer Spectrum” the modulated signal portion is displayed with a much higher power level than seen in the RFI measurement. This is because the power level is not displayed as PSD, which is the scaling used by the RFI measurement.

2.3.3 Performance Considerations

In order to determine the heritage of spectral components, each component generated, whether by a source or a nonlinearity, must be tracked as it passes through the RF link. For the linear portions of the RF link this is typically not a problem since for each source component no new components are generated.

Nonlinearities, however, can pose a significant problem, especially if the signal passes through several nonlinearities. The problem is due to the rapid growth of the number of spectral components generated by a nonlinearity as the number of input components increases.

Take for example a simple 3rd order nonlinearity with no 2nd order term:

$$y = a_1 \cdot x + a_3 \cdot x^3 \tag{19}$$

A single tone passed into the nonlinearity will generate three components for tracking: the linear component, the 3rd order harmonic, and the IM product that falls on the fundamental:

$$y = a_1 \cdot A \cdot \cos(\omega t) + a_3 \cdot (A \cdot \cos(\omega t))^3 \tag{20}$$

$$(A \cdot \cos(\omega t))^3 = \frac{3}{4} \cdot A^3 \cdot \cos(\omega t) + \frac{1}{4} \cdot A^3 \cdot \cos(\omega t) \qquad (21)$$

Two tones passed into the nonlinearity will generate nine components for tracking: the linear component and 8 nonlinear components. The nonlinear components fall on the following frequencies:

```
(1,0), (0,1), (3,0), (0,3), (2,1), (1,2), (2,-1), (1,-2)
```

where each pair of values (m,n) produces the frequency $f = |m \cdot f_A + n \cdot f_B|$.

Three tones passing into the 3rd order nonlinearity will generate 28 nonlinear components at 14 different frequencies (assuming the three tones are spaced such that none of their nonlinear products overlap).

The situation is much worse with five tones passing into a 5th order nonlinearity even with no 2nd, 3rd or 4th order terms: over 75 different frequencies are generated, of which most of those frequencies have multiple components. Add in the 2nd and 3rd order terms and the number of frequencies quickly expands.

To put the order of the nonlinearities into perspective, a 5th order nonlinearity is required in order to model a nonlinear amplifier with an |IP3-P1dB|not approximately 10 dB.

RF Inspector supports several techniques for reducing the number of components to be processed. These include ignoring components below a specified level, ignoring components outside specified frequency ranges, and limiting the total number of components generated at any given port.

These settings are all configured through the **RF Inspector** tab of the System Simulator Options dialog box, illustrated in Figure 2-13.

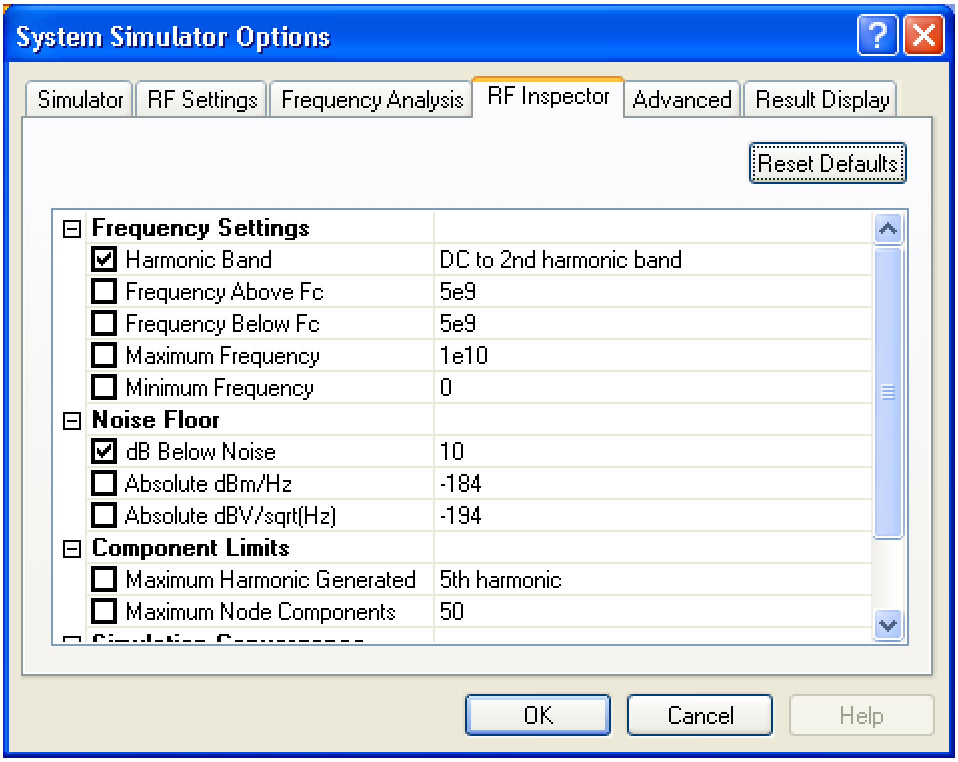


Figure 2-13. RF Inspector tab of the System Simulator Options dialog box.

FREQUENCY SETTINGS

These settings set frequency limits on the components.

The **Harmonic Band** setting is useful for restricting components to only those around the fundamental frequencies. This is particularly useful in narrowband designs, where frequencies far outside the fundamental are assumed to be severely attenuated. The default setting is "DC to 2nd harmonic band", which works well for most cases.

The **Frequency Above Fc** and **Frequency Below Fc** settings provide more control over the frequency band. These settings are useful when working with mixers as the frequency band will shift as the center frequency shifts passing through the mixers. By default these settings are disabled.

The **Minimum Frequency** and **Maximum Frequency** settings set absolute frequency limits. These set absolute limits on the frequencies generated -

components beyond these limits are ignored. By default these settings are disabled.

The frequency settings are cumulative. Only components that satisfy all the enabled constraints are generated.

NOISE FLOOR

The noise floor settings set a minimum power or voltage level required for components to be generated. Components that do not satisfy these levels are ignored.

The **dB Below Noise** setting specifies a level below the ambient noise floor. Note that negative values will set the level above the ambient noise floor. The ambient noise floor is determined by the ambient system temperature, and is approximately -174 dBm/Hz for the default temperature of 290 K. The default setting is 10 dB.

The **Absolute dBm/Hz** and **Absolute dBV/sqrt(Hz)** settings let you specify the noise floor in terms of absolute power and voltage spectral density. These settings are disabled by default.

The noise floor settings are cumulative. The smallest noise floor value computed is used. Note that when determining the noise floor RFI converts power levels to voltage levels using the characteristic impedance of the output port of the block.

The noise floor constraint is applied at both at the input and the output ports. Input signals that do not satisfy the input noise floor are not further processed. Output components that do not satisfy the output noise floor are not generated at the output port.

COMPONENT LIMITS

The **Maximum Harmonic Generated** setting lets you globally restrict the orders of the nonlinearity. Nonlinearity orders can usually be restricted on a block by block basis. For the behavioral amplifiers this can be done by leaving either the P1dB or IP3 setting blank. For mixers the secondary IMPROD, LOHMAX and INHMAX parameters control the nonlinearity orders. This setting is disabled by default.

The **Maximum Node Components** setting allows you to limit the total number of components generated at any output port. If more components than this limit are generated by the nonlinearity, only the components with the largest voltage levels are output. This setting is disabled by default.

SIMULATION CONVERGENCE

RF Inspector utilizes an iterative algorithm to handle nonlinearities as explained the "*How RFI Works*" section.

The **Maximum Iterations** setting limits the number of iterations allowed. If the simulation exceeds this number of iterations the simulation halts and a warning message is reported.

The **Change in dB** setting defines the tolerance used for determining convergence. If the change in voltage levels for individual components between iterations falls below this level for all components the simulation is considered converged and the simulation is completed.

2.3.4 How RFI Works

RF Inspector can be thought of as a system level frequency domain circuit simulator. It solves for voltages and currents at each RF node for each generated spectral component. For independent CW sources such as the TONE block and for the supported modulated signal sources, this is a fairly straightforward solution.

For each solution, Y matrices are obtained from all the RF blocks in an RF link and assembled into a global Y matrix according to the node connectivity. Appropriate terminations are added. The matrix is then solved using the source current for the particular solution. Voltages at each node and the current flowing into the ports of each block are obtained.

NOTE. In RF Inspector, the distinction between input and output port depends on the direction of the signals being processed. For example, for an amplifier the nonlinearity is applied in the forward direction. Therefore, when describing the ports for the nonlinearity, the input port is port 1 of the block while the output port is port 2. On the other hand, when computing reverse leakage, the input port is port 2 while the output port is port 1.

Nonlinearities are treated as voltage sources that are dependent upon the incident voltage at the input port of the nonlinearity. Each individual component generated by the nonlinearity is a separate source. The nonlinear dependency requires the use of an iterative algorithm.

In simplified form the algorithm is:

1. Compute voltages and currents due to independent sources.
2. Compute nonlinear sources based on the components at the corresponding input ports.
3. Compute voltages and currents due to the nonlinear sources.
4. If the change in the voltages and currents satisfy the convergence criteria, stop. Otherwise repeat from step 2.

NOTE. When solving for the node voltages and port currents, the voltage sources are applied in series with an impedance equal to the characteristic impedance of the output port.

2.3.5 RFI Impedance Mismatch Modeling

RF Inspector always operates with impedance mismatches modeled. This is because RF Inspector solves for voltages and currents in the ports of linear RF blocks using a typical Y matrix based linear circuit solver.

There is one difference between simulations with impedance mismatch modeling enabled in the **RF Settings** tab of the System Simulator Options dialog box. When impedance mismatch modeling is disabled, RF Inspector will automatically terminate any ports that are not connected to another port. The termination will be set to the system characteristic impedance _Z0.

2.3.6 Feedback Loops in RFI

RF Inspector only supports linear feedback loops. All components in the loop must be linear RF blocks, with the exception of the Negate block NEG, the Phase block PHASE, and the delay blocks DLY_SMP, DLY_SYM and DELAY. Note that delay blocks are ignored in RFI.

Nonlinear feedback loops, such as feedback control for a VGA_F block are not supported.

Because RFI uses a frequency domain simulator, feedback loop results may differ from time domain simulations. This is most apparent with issues of stability. Consider the linear feedback loop in Figure 2-14. The closed loop gain is expressed as:

$$G = \frac{A}{1 + A \cdot B} \tag{22}$$

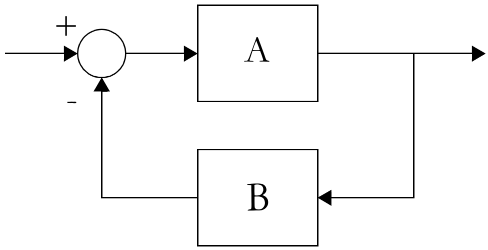


Figure 2-14. Example Frequency Domain Linear Feedback Loop

In the VSS time domain simulator, which is essentially a discrete time simulator and operates in the z-domain, a minimum of one sample delay must be introduced. The closed loop gain in the z-domain is:

$$G(z) = \frac{A \cdot z}{z + A \cdot B} \tag{23}$$

In order for the loop to be stable in the z-domain the following must be true:

$$A \cdot B \leq 1 \tag{24}$$

However, there is no such restriction in the frequency domain.

Another difference to note when working with feedback loops in the time domain is the extra delay introduced by filters due to their implementation as FIR and/or IIR filters.

2.3.7 Additional Notes

MIXERS

The mixer model used in RF Inspector differs from the model used in the time domain simulator. In time domain simulations mixers are modeled using a nonlinear amplifier followed by a spur generator. The spur generator synthesizes the spurs that fall within the output sampling frequency band.

In RFI simulations, mixers use a polynomial model to generate the input products. The corresponding LO products are then applied to obtain the output spurs. There is no nonlinear amplifier to generate the IP3 and P1dB effects. Instead, these effects are handled by the polynomial applied to the input signal.

The net effect is in RFI simulations IP3 and P1dB directly affect the levels of spurs with 3rd and 5th order input terms, whereas in time domain simulations those spurs are independent of IP3 and P1dB.

SATURATION EFFECTS

The modeling of saturation effects in nonlinear amplifiers and mixers in RF Inspector differs from the modeling used in the time domain simulator.

In time domain simulations saturation shaping is applied directly to the instantaneous voltage and is a function of that voltage. The shaping has the effect of generating many more nonlinear products that would be produced by the 5th polynomial model. The "*Saturation Effects*" sub-section of the "*Nonlinear Modeling Issues*" section describes the time domain saturation model.

In RF Inspector, the saturation is applied to the voltage levels of the individual components. Additional nonlinear products are not generated. The effect in RFI simulations is a decrease in the gains for both the fundamental and nonlinear products as the signal saturates.

The signal levels will therefore tend to differ once saturation effects are applied to the signal. Note that saturation effects are only applied after the input signal reaches the P1dB point.

2.4 IMPEDANCE MISMATCH MODELING

NOTE. The Impedance Mismatch Modeling section only applies to RF Budget Analysis and time domain simulations. RF Inspector simulations always include impedance mismatch effects as described in the section "*RFI Impedance Mismatch Modeling*".

Many RF blocks support the modeling of impedance mismatch effects. By default impedance mismatch modeling is disabled. It must be explicitly enabled through the **RF Settings** tab of the Options dialog box of individual system diagrams, or in the System Simulator Options dialog box. When enabled, impedance mismatch modeling is performed for both RF Budget Analysis and time domain simulations.

When impedance mismatch modeling is enabled, the impedances seen at each RF node are taken into account when computing signal voltages. In time domain simulations the data samples at each RF node represent total instantaneous voltage and are transformed to incident voltage on input and back to total voltage on output. This means that a waveform measurement displays the total instantaneous voltage.

2.4.1 Block Support for Impedance Mismatch Modeling

Because impedance mismatch modeling applies to circuits, only a limited set of VSS blocks directly support impedance mismatch modeling. The RF blocks listed in the "*RF Blocks In VSS*" section support impedance mismatch modelling unless otherwise noted.

Several utility blocks are transparent to impedance - the impedance seen looking into a port is the same as the impedance looking out the opposite port. The following blocks are transparent to impedance mismatch modeling:

Block	Description
ABS	Absolute Value
ACOS	Arc cosine
ACOSH	Inverse Hyperbolic Cosine
ALIGN	Align Signal (Gain, Phase and Delay Compensate)
ARG	Calculates phase angle
ASIN	Arc sine
ASINH	Inverse Hyperbolic Sine
ATAN	Arctangent
ATANH	Inverse Hyperbolic Tangent
ATTEN	Attenuator
CE2R	Complex Envelope to Real Converter
CONJ	Complex Conjugate
COS	Cosine
COSH	Hyperbolic Cosine

Block	Description
DIV	Divider
EXP	Exponential
EXPJ	Complex Exponential
FRQSHFT	Carrier Frequency Shifter
GAIN	Gain Block
LN	Natural Logarithm
LOG	Logarithm
NORM	Complex Magnitude Squared
POW	Raise to the Power
R2CE	Real to Complex Envelope Converter
RECIP	Reciprocal (1/x)
RESAMPLER	Rational Resampler
SCALE	Scale and Offset
SIGN	Signum Function
SIN	Sine
SINH	Hyperbolic Sine
SQR	Square
SQRT	Square root
TAN	Tangent
TANH	Hyperbolic Tangent

NOTE. In VSS 7.0.x the NEG and PHASE blocks were transparent to impedance mismatch modeling. They now appear as terminations and can be used in RF Inspector simulations.

NOTE. The above blocks do not in general support RF Inspector simulations.

2.4.2 Modeling Basics

There are two main steps involved in impedance mismatch modeling within VSS:

- *Determining Impedances*
- *Applying Voltage Corrections*

DETERMINING IMPEDANCES

The first step in impedance mismatch modeling is to determine the impedances throughout the system diagram. Impedances need to be determined only at ports for blocks that support impedance mismatching and at ports connected to those ports, either directly or through impedance transparent blocks.

Blocks model impedances in one of three ways:

- Ports appear as terminations.
- Blocks are represented as frequency dependent Y parameters.
- Blocks are transparent to impedances.

Blocks that do not support impedance mismatch and are not transparent automatically appear as terminations. The termination impedance is the system characteristic impedance _Z0, which is set in the **RF Options** tab of the Options dialog box of system diagrams or the System Simulator Options dialog box.

Some of the blocks that support impedance mismatch also appear as terminations. Examples are the mixers, the AM/PM based nonlinear amplifiers NL_F and NL_S, and the TONE source.

The linear RF blocks present themselves as frequency dependent Y parameters. In many cases the Y parameters are obtained from S parameters using the system characteristic impedance _Z0. Other blocks, such as the circuit filters, have parameters specifying the characteristic impedances. LIN_S obtains the Y parameters directly from the Microwave Office circuit.

The behavioral amplifier AMP_B also presents itself in Y parameter form. It does this by linearizing its response at the input signal's operating point.

Once all the blocks have presented their impedance models to the simulator, the simulator resolves the impedances seen looking out of and into every port where impedances are needed. The following diagram illustrates the impedances available to block B and the other impedances used in computing those impedances.

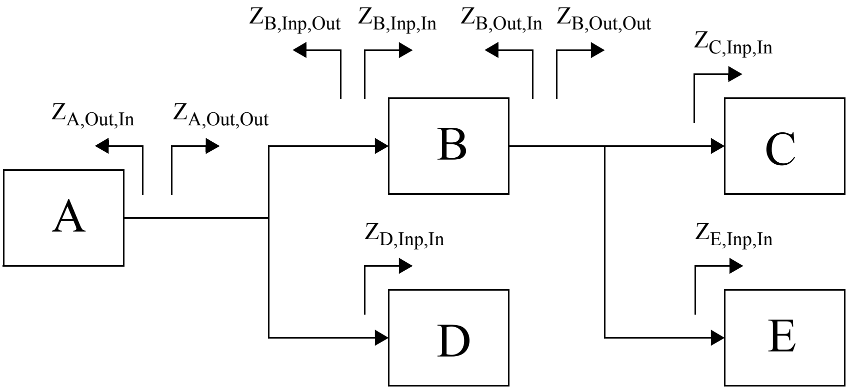


Figure 2-15. Impedances available to a block

The impedance $Z_{B,Inp,Out}$ is computed from $Z_{A,Out,In}$ and $Z_{D,Inp,In}$. $Z_{B,Out,Out}$ is computed from $Z_{C,Inp,In}$ and $Z_{E,Inp,In}$. For Y parameter based blocks $Z_{B,Inp,In}$ and $Z_{B,Out,In}$ are computed from:

$$Z_{B,\ Inp,\ In} = \frac{1}{Y_B} + Z_{B,\ Out,\ Out} \tag{25}$$

$$Z_{B,\ Out,\ In} = \frac{1}{Y_B} + Z_{B,\ Inp,\ Out} \tag{26}$$

For termination blocks, $Z_{B,Inp,In}$ and $Z_{B,Out,In}$ are the termination impedance.

The impedance looking into or out of any port can be displayed using the Z_node RF Budget Analysis measurement.

VOLTAGE CORRECTIONS

Once the various impedances are available to a block, the block uses the impedances to compute any necessary voltage corrections for its inputs and outputs. The corrections are required to convert from total voltage to incident voltage at the inputs and from reflected voltage to total voltage at the outputs.

The input correction is:

$$Corr_{Inp} = \frac{1}{|1 + \Gamma_{In}|} \tag{27}$$

where:

$$\Gamma_{In} = \frac{Z_{Inp,\,In} - Z_0}{Z_{Inp,\,In} + Z_0} \tag{28}$$

The output correction is:

$$Corr_{Out} = \frac{1 + \Gamma_L}{1 - S22 \cdot \Gamma_L} \tag{29}$$

where:

$$\Gamma_L = \frac{Z_{Out,\,Out} - Z_0}{Z_{Out,\,Out} + Z_0} \tag{30}$$

Z_0 is the characteristic impedance of the block, which in many cases is the system characteristic impedance _Z0. Blocks that support specifying characteristic impedances usually do so through ZINP and ZOUTP parameters.

2.4.3 Source and Load Impedances

When performing impedance mismatch modeling, it is important to properly terminate outputs. When impedance mismatch modeling is not being performed, a block's output always sees the system characteristic impedance _Z0 at its output, regardless of how many blocks are connected to the output. When impedance mismatching modeling is enabled, the impedance seen at the outputs of RF blocks (as well as the inputs) will depend on what is connected to the block. If the output is left unconnected, it will be treated as an open circuit. The same holds true if the output is connected only to a test point with default settings.

The Grounded Resistor block LOAD and the End of RF Signal block RF_END, both found in **RF Blocks > Impedance Mismatch**, can be used to explicitly terminate an RF output. Both LOAD and RF_END let you define a frequency dependent complex impedance for the load. RF_END also provides the voltage seen at the load as a non-RF signal output.

The test point blocks TP, VNA_LS, VNA_SS and VSA can all be configured as a load. By default they are configured as probes, with infinite impedance. The ZL parameter can be used to specify the desired frequency independent complex load impedance.

NOTE. Experience has shown that LOAD blocks be used rather than the ZL parameter of the test point blocks when specifying terminations. Using a LOAD block makes it clear when a termination has been set. Using the test point blocks, on the other hand, can lead to unintentionally specifying multiple loads at a node.

The RF source blocks also allow you to specify a frequency independent source impedance. The TONE, OSC_S, VNA_LS and VNA_SS blocks all have a ZS parameter for specifying the source impedance.

2.4.4 Interfacing to Non-RF Blocks

Blocks that do not directly support impedance mismatch modeling can still be connected to RF blocks. For example, the transmitter blocks do not directly support impedance mismatch modeling, yet you can still connect an amplifier to their output. However, if the impedance seen looking into the amplifier's input port is not the system characteristic impedance then the signal voltage must be corrected to properly reflect the effect of the impedance mismatch at the input to the amplifier.

The Start RF Signal block RF_START in **RF Blocks > Impedance Mismatch** is used to apply the necessary mismatch correction. It supports an S22 parameter and an output characteristic impedance parameter ZOUT. RF_START's input port appears as a termination with impedance _Z0.

When the load impedance at the output port of RF_START is conjugate matched to ZOUT, the power delivered to the load is the power delivered to RF_START's input port.

2.4.5 Additional Notes

There are several additional items to be aware of when interpreting the results of an impedance mismatch modeled simulation.

In time domain simulations transmittance in the reverse direction through a block (such as S12) cannot be modeled due to the fact that data samples only flow in the forward direction.

In time domain simulations frequency dependent voltage corrections must be implemented with filters. Unfortunately, due to digital filter design constraints the time domain response of the filter may not represent the desired corrections accurately. The "*Filter Issues*" section has more information on some of these design issues. Frequency independent voltage corrections do not normally have these problems as they are a simple complex scaling.

FEEDBACK LOOPS

Also related to the filter problem is the handling of impedance mismatch in feedback loops. In general, frequency dependent impedance mismatch modeling is not supported in feedback loops because of the need to use filters to perform the voltage corrections in the time domain. These filters would typically introduce a filter delay which may affect the overall response of the feedback loop. The exception to this is the linear filter based RF blocks. These blocks, since they already utilize a filter when impedance mismatch modeling is not enabled, can implement the voltage corrections by modifying the frequency response of their filter.

TOPOLOGY LIMITATIONS

The algorithm used to compute the impedances has limited support for circuit layouts that contain loops in the signal paths. For example, the algorithm cannot accommodate splitting a signal and then recombining the components. In these cases, the simulator generates a warning message similar to:

```
Caution: The impedance seen from one or more nodes could not be
resolved, the system impedance of 50 Ohms has been used.
```

2.5 FILTER ISSUES

VSS relies upon Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) digital filters to implement frequency dependent behavior in time domain simulations. Unfortunately, there are limitations and trade-offs as to how well these digital filters can apply a desired frequency response to a time domain signal.

FIR filters are the simpler of the two filters in terms of implementation. An FIR filter has M filter coefficients, or taps, and has order M-1. When implemented using FFTs, they are significantly faster than IIR filters of much smaller order. The primary drawback of the FIR filter is its introduction of ripple in the frequency response. The ripple is due to the finite length of the impulse response.

The IIR filter implementation used by VSS typically provides a better overall approximation than the FIR filter, particularly when working with modulated signals or arbitrary tones. However, the IIR filter implementation has its own shortcomings, including sensitivity to numerical limitations such as truncation and round-off errors. The IIR filter design also utilizes the bilinear transform, which introduces a nonlinear mapping of analog frequencies to discrete frequencies. Upsampling and downsampling of the signal is required to reduce some of the effects of the bilinear transform, which introduces signal distortion as well as excessive signal delay in feedback loops due to the non-ideal lowpass anti-aliasing filter required before downsampling. The IIR filter design also requires a pole-residue approximation of the desired frequency response, which has its own limitations.

Figure 2-16 illustrates the ripple in the stopband of an FIR implementation of a notch filter. It also shows the response of an IIR filter implementation that provides a much better response.

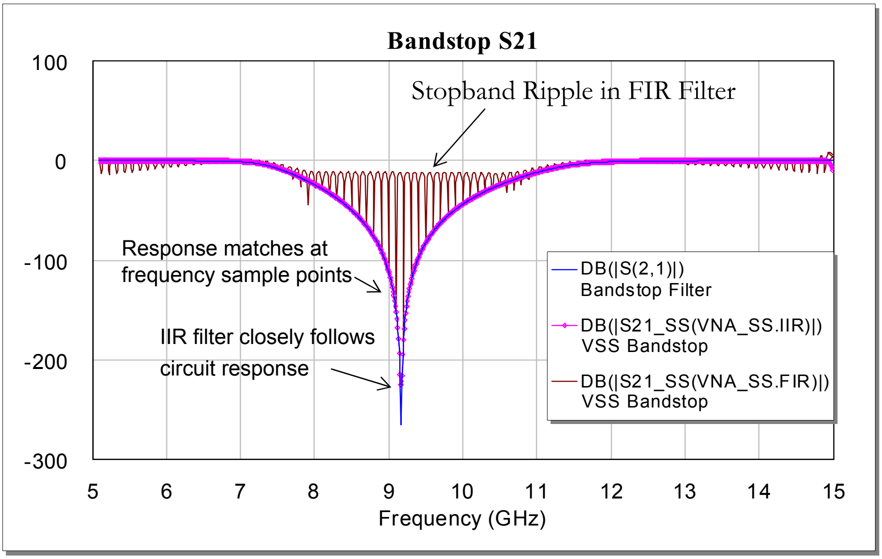


Figure 2-16. Comparison of FIR and IIR filters for a notch filter

Figure 2-17 illustrates some of the distortion introduced by the IIR filter implementation.

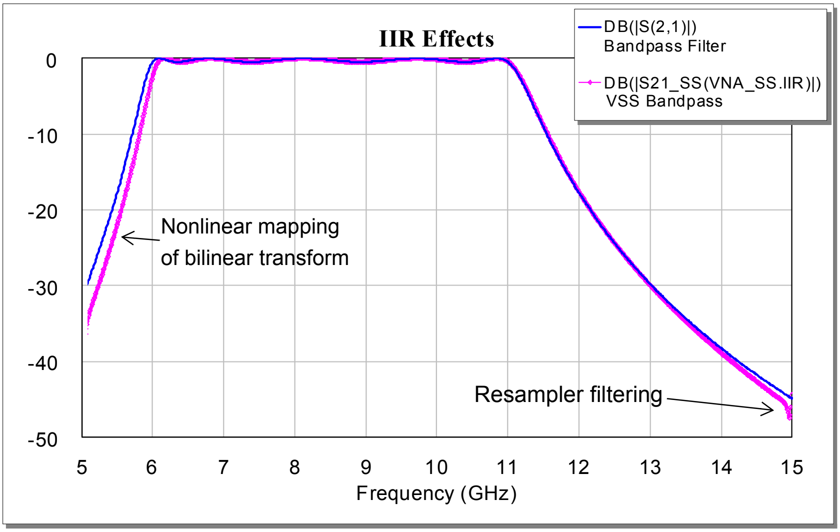


Figure 2-17. IIR filter distortion

2.5.1 FIR Filters

Finite Impulse Response, or FIR, filters are fairly simple filters to implement. The time-domain equation for the n'th output sample is:

$$y(n) = \sum_{k=0}^{M-1} b_k \cdot x(n-k) \tag{31}$$

The filter response in z-domain is:

$$H(z) = \sum_{k=0}^{M-1} b_k \cdot z^{-k} \tag{32}$$

FIR filters can be efficiently implemented using fast Fourier transforms (FFT) provided the filter is not part of a feedback loop. When implemented using an FFT, an FIR filter with M=1000 will normally process samples faster than an FIR filter using equation 31 directly with M=100.

The primary drawback of the FIR filters is ripple in the frequency response. Figure 2-18 illustrates ripple in a fairly simple lowpass filter.

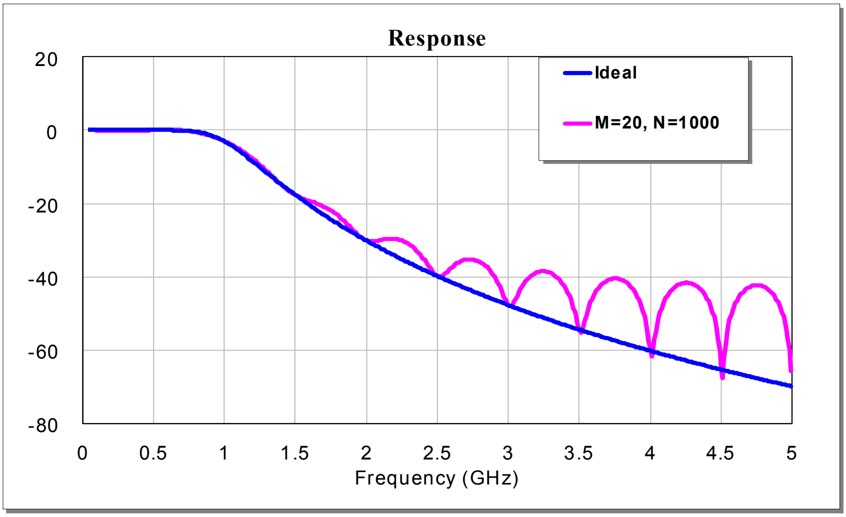


Figure 2-18. Ripple in an FIR filter implementation of a 7th order Butterworth filter, M=20

In the figure, the blue curve represents the ideal response of a 7th order Butterworth filter with a 1 GHz cutoff frequency and a maximum passband attenuation of 3.0103 dB. The pink curve shows the response of an FIR filter implementation of the same filter with M=20. The frequency response curve was generated by passing 1000 samples of an impulse through the filter and then applying the discrete Fourier transform (DFT) to obtain the frequency response. Note that in this case the sampling frequency fs is 10 GHz centered at 0 GHz. The frequency response is symmetric about DC.

This filter was designed using the frequency sampling design procedure. VSS supports two other FIR filter implementations: a windowed FIR filter and a combination windowed FIR filter and an allpass IIR filter.

FREQUENCY SAMPLING

Frequency sampling is a fairly simple procedure. The FIR filter coefficients b_k are obtained by applying the inverse DFT to the desired frequency response sampled at frequencies:

$$
\begin{aligned}
f_i &= f_c - \frac{f_s}{2} + \frac{i \cdot f_s}{M} && \text{for } M \text{ even} \\
f_i &= f_c - \frac{f_s}{2} + \frac{(i + 1/2) \cdot f_s}{M} && \text{for } M \text{ odd}
\end{aligned}
\qquad (33)
$$

The number of frequencies *M* is determined by the NFREQFIR parameter. If NFREQFIR is empty, the number will be automatically determined.

Frequency sampling results in a filter design that matches the desired frequency response at the sampled frequencies f_i. However, between those frequencies the response deviates from the desired frequency response, which appears as the bumps, or ripple, in Figure 2-18.

Increasing M may improve the response as illustrated in Figure 2-19.

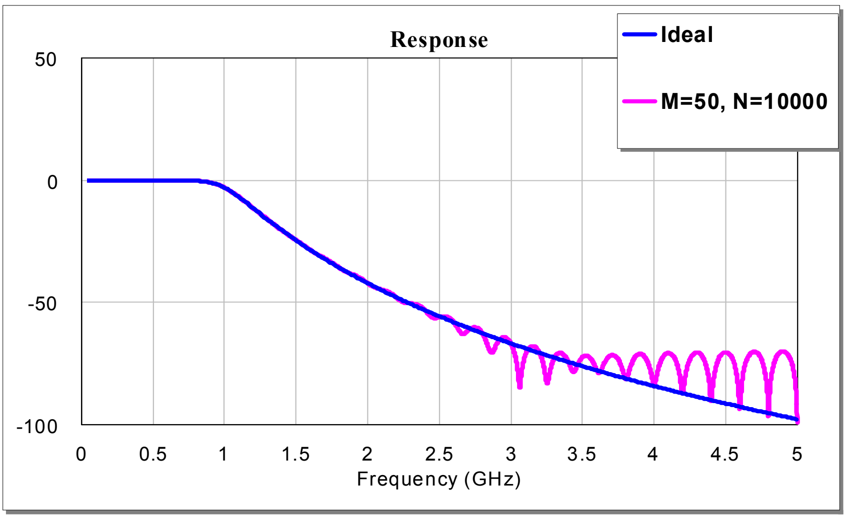


Figure 2-19. Ripple in an FIR filter implementation of a 7th order Butterworth filter, M=50

There will always be some amount of ripple, particularly in the stopband. Depending on the requirements of your design, the ripple may be acceptable. However, when working with notch filters, the attenuation in the notch will typically fall short of the desired attenuation.

WINDOWED FIR FILTERS

Windowed FIR filter design is a design technique that typically (though not always) provides an overall magnitude response that is superior to that of the frequency sampling technique, particularly between frequency sample points. The trade-offs are the filter has a linear phase response and there is a smoothing of the magnitude response.

This technique involves creating a frequency response with the desired magnitudes at the frequencies given by equation 33. The phase response is set to a linear phase response of the form:

$$\begin{aligned} \theta_i &= -\pi \cdot (i - M/2) && \text{for } M \text{ even} \\ \theta_i &= -\pi \cdot (i - (M-1)/2) && \text{for } M \text{ odd} \end{aligned} \tag{34}$$

The inverse DFT is then applied to this response to obtain the time-domain impulse response.

A windowing function is applied to the impulse response. The windowing function has the effect of smoothing the ends of the impulse response, reducing ripple due to the truncated nature of the FIR filter. VSS uses a Blackman-Harris window function.

The windowed impulse response becomes the coefficients of the FIR filter.

Figure 2-20 shows the magnitude response of a windowed FIR design for the filter response in Figures 2-18 and 2-19. Figure 2-21 shows the constant group delay imparted by the windowed FIR design compared to the group delay of the ideal filter.

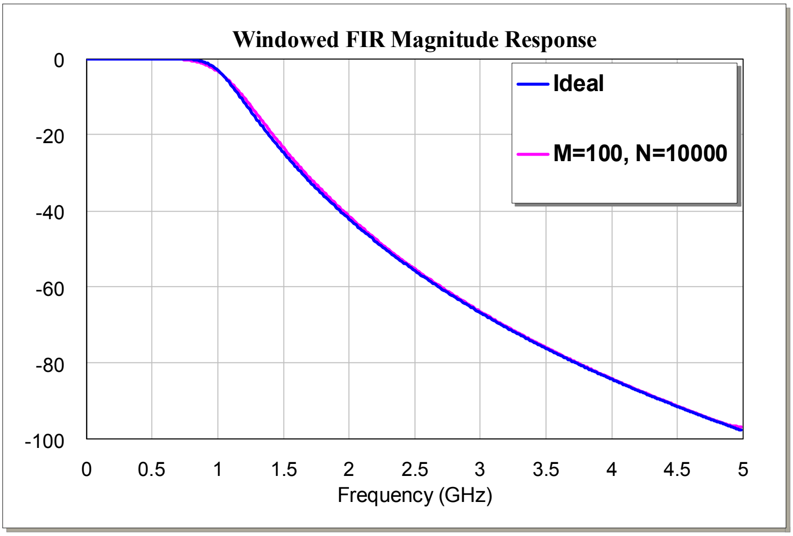


Figure 2-20. Magnitude response of windowed FIR filter design.

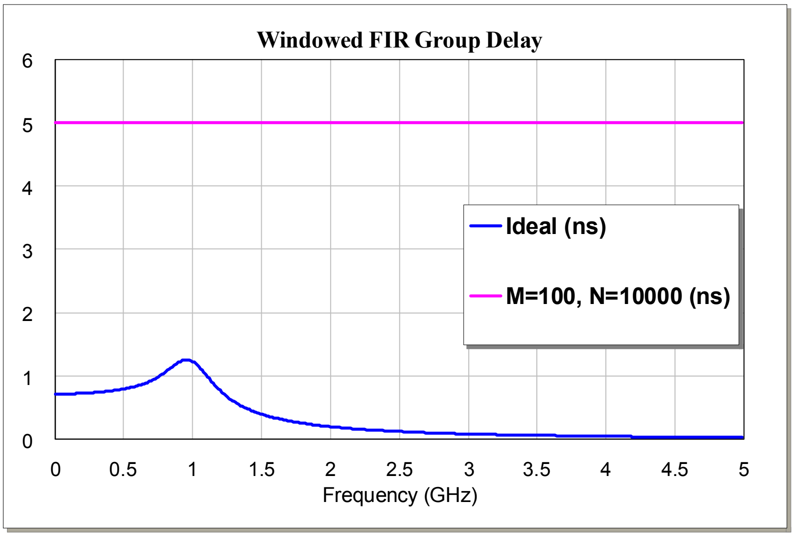


Figure 2-21. Group delay response of windowed FIR filter design.

WINDOWED FIR + ALLPASS IIR FILTER

The windowed FIR + allpass IIR filter design attempts to obtain a frequency response with the same magnitude response as the windowed FIR filter design, but with a better group delay response. This is done by combining a windowed FIR filter with an allpass IIR filter. The allpass IIR filter is used to add a non-constant group delay response to the windowed FIR filter's constant group delay response.

The allpass IIR filter is implemented with the following z-domain transfer function:

$$H(z) = \prod_{i=0}^{M-1} \frac{1 + c_{1i} \cdot z + c_{0i} \cdot z^2}{c_{0i}^* + c_{1i}^* \cdot z + z^2} \qquad (35)$$

Because of the nonlinear nature of equation 35, the allpass IIR filter is designed using an iterative algorithm that attempts to minimize the error between the IIR filter's group delay response and the desired group delay response. Because of the difficulty inherent in designing an IIR filter to an arbitrary frequency response, the group delay of the IIR filter will rarely closely match the desired response. However, it will often have the same general shape.

Figure 2-22 shows the group delay response of the windowed FIR + allpass IIR filter design for the ideal group delay response in Figure 2-21.

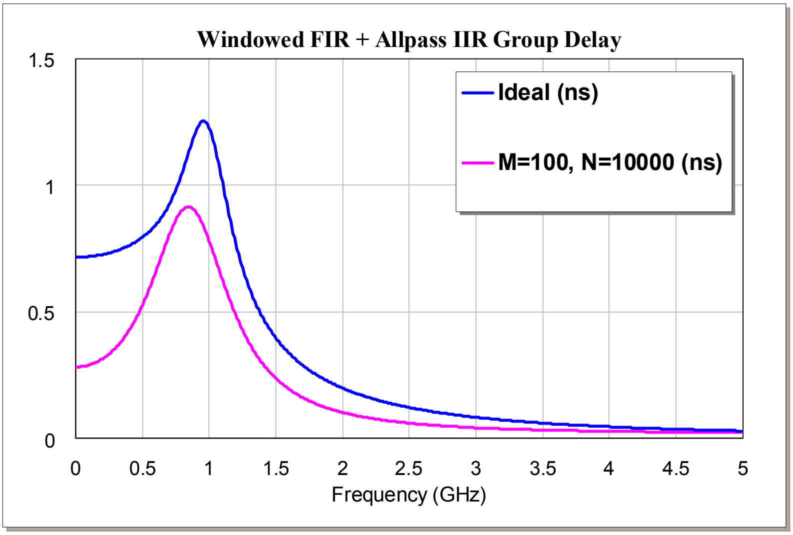


Figure 2-22. Group delay response of the windowed FIR + allpass IIR design.

OTHER FIR DESIGN TECHNIQUES

There are other techniques for designing FIR filters. Most of these techniques are based on minimizing the overall frequency response error or minimizing the amount of ripple. The most effective design techniques partition the desired frequency response into three types of regions: passband, stopband, and transition. They then attempt to reduce the ripple in the passbands and stopbands while not restricting the ripple in the transition regions.

Unfortunately, when attempting to match a general frequency response such as the frequency responses for applying impedance mismatch voltage corrections or the frequency response for an analog filter such as the Butterworth filter, the partitioning techniques often cannot be applied since transition regions cannot be specified.

2.5.2 IIR Filters

Infinite impulse response, or IIR filters, are more complex and generally slower than equivalent FIR filters. However, they do typically provide a better overall frequency response match than a frequency sampled FIR filter.

The general time-domain equation for the n'th output sample of an IIR filter is:

$$y(n) = -\sum_{k=1}^{N} a_k \cdot y(n-k) + \sum_{k=0}^{M} b_k \cdot x(n-k) \tag{36}$$

or in z-domain form:

$$H(z) = \frac{\sum_{k=0}^{M} b_k \cdot z^{-k}}{1 + \sum_{k=1}^{N} a_k \cdot z^{-k}} \tag{37}$$

Note that these relations can be expressed in other ways.

Many of the difficulties of working with IIR filters arise from the feedback terms, those with coefficients a_k. On the design side, the feedback terms introduce poles into the frequency response, which may result in an unstable filter. Determining the coefficients is a complex problem in itself, particularly when arbitrary frequency responses are desired.

On the implementation side, the feedback terms make IIR filters much more sensitive than FIR filters to numerical issues such as quantization and round-off due to the finite resolution of floating point values in digital computers. Fortunately, by choosing the proper structure and order of application these numerical issues can be minimized.

The following briefly summarizes the steps involved in the IIR filter implementation:

1. An s-domain transfer function approximation of the desired frequency response is made. The transfer function is in pole-residue form.
2. The s-domain transfer function is mapped into the z-domain using the bilinear transform. The transfer function is also frequency-shifted prior to

the application of the bilinear transform to align the center frequency with z = 0.

3. At simulation time, each sample input is fed into the IIR filter. If UPRATE is greater than 1, UPRATE-1 zeros are then input into the IIR filter, effectively upsampling the filtered signal.
4. If UPRATE is greater than 1, the output of the IIR filter is passed through a lowpass filter (relative to the center frequency). The output of the lowpass filter is then decimated by UPRATE-1, effectively downsampling the previously upsampled filtered signal.

UPRATE is a control parameter that determines the upsampling rate used by the IIR filter implementation. If a block does not have this parameter available a value of 4 is used.

The s-domain transfer function approximation results in a transfer function of the form:

$$H(s) = C_0 + C_1 \cdot s + \sum_{i=1}^{N} \left(\frac{R_i}{s - P_i} + \frac{R_i^*}{s - P_i^*} \right) \qquad N \le MAXNPOLE \tag{38}$$

where (x)* indicates the complex conjugate of x. Analog frequencies f_a are related to s by:

$$s = j\Omega = j \cdot 2\pi \cdot f_a \tag{39}$$

MAXNPOLE is a control parameter that is provided to allow you to limit the number of poles in the transfer function. The default value for MAXNPOLE varies depending on the number of frequencies supplied, but the maximum default value is 40.

By applying the bilinear transform:

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \tag{40}$$

an IIR filter can be designed that approximates H(s) in the z-domain. T is a design parameter.

The bilinear transform maps the analog frequencies Ω to the z-domain frequencies ω according to the following equation:

$$\omega = 2 \cdot \operatorname{atan}\frac{\Omega T}{2} \tag{41}$$

If the signal representation is complex envelope, the frequencies must be shifted so the z-domain frequency band is centered at the center frequency, or $\omega = 0$ when $\Omega = 2\pi \cdot f_c$. This is easily accomplished by adding $-j \cdot 2\pi \cdot f_c$ to s prior to applying the bilinear transform.

The bilinear transform maps the entire analog frequency range $-\infty \le \Omega \le \infty$ into the z-domain frequency range $-\pi \le \omega \le \pi$. The result is that all frequencies, including unwanted frequencies, appear within the z-domain response. Additionally, severe frequency compression occurs as ω approaches $\pm\pi$ due to the arctangent relation in equation 41. The following illustrates the mapping process:

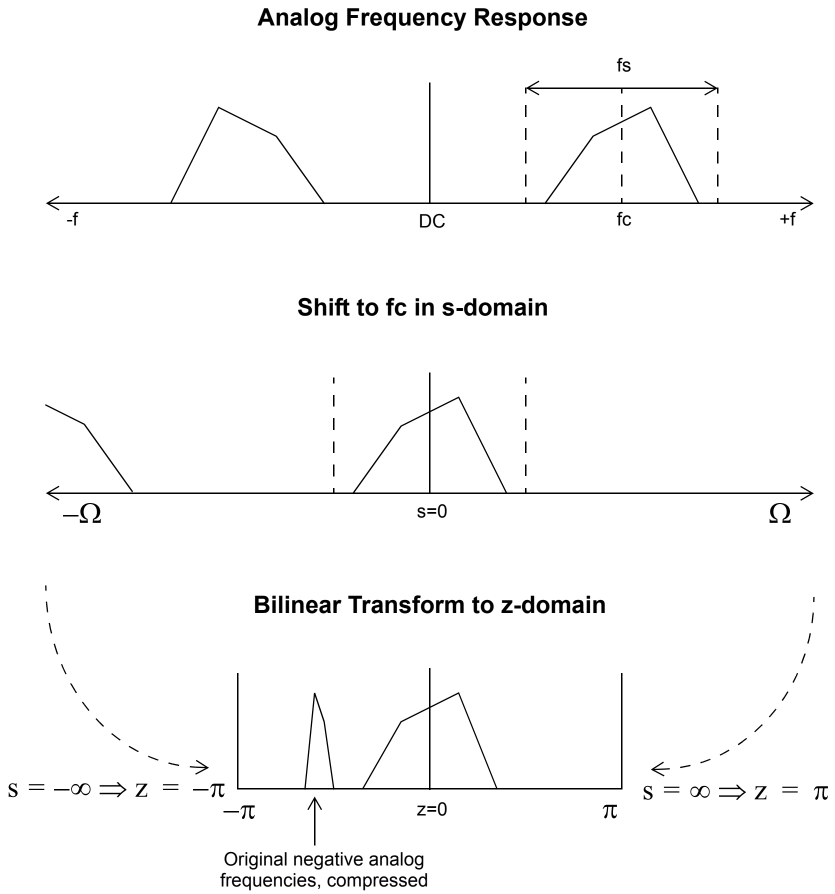


Figure 2-23. Frequency mapping from analog to Z-domain

The effects of the bilinear transform can be controlled to a certain extent by upsampling the signal prior to passing it through the IIR filter. By properly designing the IIR filter, the frequencies outside the desired analog frequency band can be made to fall outside of the original pre-upsampled z-domain frequency band fs. The signal is then passed through a lowpass z-domain filter with bandwidth fs, attenuating the unwanted frequencies. The signal is then decimated back to the original sampling frequency.

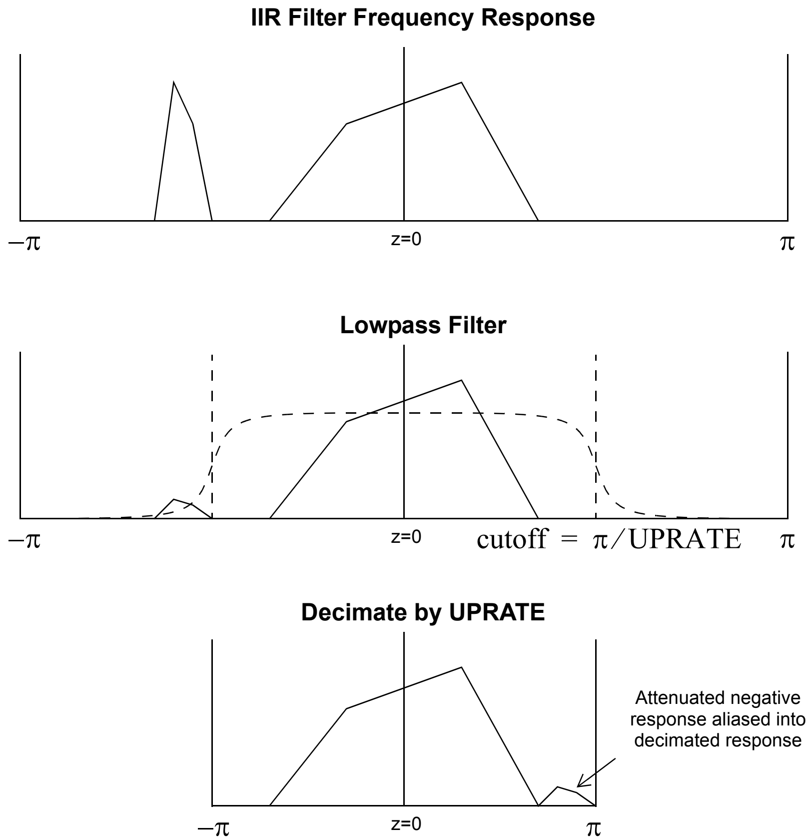


Figure 2-24. Upsampling to shift unwanted frequencies out of sampling frequency band

The variable T in equations 40 and 41 is determined by both UPRATE and FRQALIGN. FRQALIGN defines the frequency at which analog frequency and z-domain frequency match:

$$f = \frac{f_s}{2} \cdot FRQALIGN \tag{42}$$

FRQALIGN is a control parameter like UPRATE. If a block does not have FRQALIGN available, it is generally set to the inverse of the oversampling rate of the input signal.

Substituting equation 42 into equation 41 and solving for T we obtain:

$$T = \frac{2 \cdot \tan\left(\frac{\pi \cdot FRQALIGN}{2 \cdot UPRATE}\right)}{\pi \cdot f_s \cdot FRQALIGN} \tag{43}$$

After transformation into the z-domain, the transfer function has the following form:

$$H(z) = G + C \cdot \frac{1 - z^{-1}}{1 + z^{-1}} + \sum_{i=1}^{2N} \left(\frac{\beta_i \cdot (1 + z^{-1})}{1 + \alpha_i \cdot z^{-1}} \right) \tag{44}$$

This form is readily implemented by a parallel IIR filter structure.

2.5.3 Working With Circuit Filter Based Blocks

The linear filter RF blocks all share the same core implementation. There may be some differences with the default configurations used, but they all support the same core features.

The four digital filter implementations described in the previous sections are available:

- Frequency sampled FIR filter
- Windowed FIR filter
- Windowed FIR + allpass IIR filters
- S-domain derived IIR filter

By default the circuit filter blocks are configured to automatically select a filter implementation. The IMPL parameter is used to select a specific implementation.

The selection algorithm can be described as:

1. If the filter is in a feedback loop, select frequency sampled FIR filter.

2. Design a frequency sampled FIR filter and an s-domain derived IIR filter. Compute error metrics for both designs. If one of the filters has a sufficiently small error metric, select that filter. If the block is one of the blocks in the **Filters** category, select the filter with the smallest error metric.

3. If step 2 did not result in a selected filter implementation, design a windowed FIR + allpass IIR filter. Select the filter implementation from the three designs with the smallest error metric.

Note that the blocks in the **Filters** category do not attempt a windowed FIR + allpass IIR design. This is because they are based on s-domain filter responses and the s-domain IIR design most often provides a better response than the windowed FIR + allpass IIR design.

In most cases either the frequency sampled FIR or the s-domain derived IIR designs are sufficient. The windowed FIR + allpass IIR design is only attempted if neither of the other designs is satisfactory. This is because the windowed FIR + allpass IIR design algorithm takes significantly longer than the other design processes.

NFREQ AND NFREQFIR

The number of frequencies used in the different filter designs depends on the NFREQ or NFREQFIR parameters. The NFREQ parameter is used primarily for the s-domain derived IIR filter design, though if NFREQFIR is left empty it may also be used for the FIR filter designs. The NFREQFIR parameter is used only for the FIR filter designs.

The effect of the number of frequencies used in a filter design is dependent upon the filter design algorithm. The s-domain derived IIR filter design algorithm is quite sensitive to the number of frequencies used. Changing the number of frequencies by 1 or 2 may sometimes produce significantly different responses. Too many frequencies or too few frequencies may produce a response that does not resemble the desired response at all.

Experience has shown that between 50 and 200 frequency points tends to work best for the s-domain derived IIR filter design. You can easily experiment with the circuit filters in the **Filters** category using the **Filter Design** tab of the Element Options dialog box.

When working with LIN_S or LIN_F with 50 or more frequency points in the MWO circuit or data file that fall within the sampling frequency band, NFREQ should be left alone. This lets the blocks use the actual frequency points from the MWO circuit simulation or the text data file as the design response. Doing so avoids interpolating frequency response information, which may yield an undesirable response.

For the FIR filter based designs, including the windowed FIR + allpass IIR design, the number of frequencies used directly affects the resolution of the filter. Unless the filter is used in a feedback loop, using between 1000 and 10,000 frequency points generally yields satisfactory results in both response resolution and simulation speed. When used in a feedback loop, the number of frequencies should be much lower, since the FIR filter is implemented in convolutional mode, which is much slower than the FFT mode.

2.6 NONLINEAR MODELING ISSUES

There are a number of issues that need to be kept in mind when modeling nonlinearities in time domain simulations.

2.6.1 Aliasing

Aliasing is a result of the fact that time domain simulations have a bandwidth limited to the sampling frequency. Center frequency also contributes to the problem when working with complex envelope signals.

Aliasing problems can arise because of the frequency bandwidth expansion that occurs when a signal passes through a nonlinearity. Any generated components that fall outside the sampling frequency band will be aliased back into the sampling frequency band. Figure 2-25 illustrates the bandwidth expansion for a 3rd order nonlinearity due to harmonics. Intermodulation products would further expand the signal at each harmonic.

When aliasing occurs, signal content that is generated outside of the sampling frequency band is ‘folded’ back into the sampling frequency band, or aliased. Figure 2-26 illustrates aliasing for the example 3rd order nonlinearity. Note that intermodulation products are not illustrated in the figure.

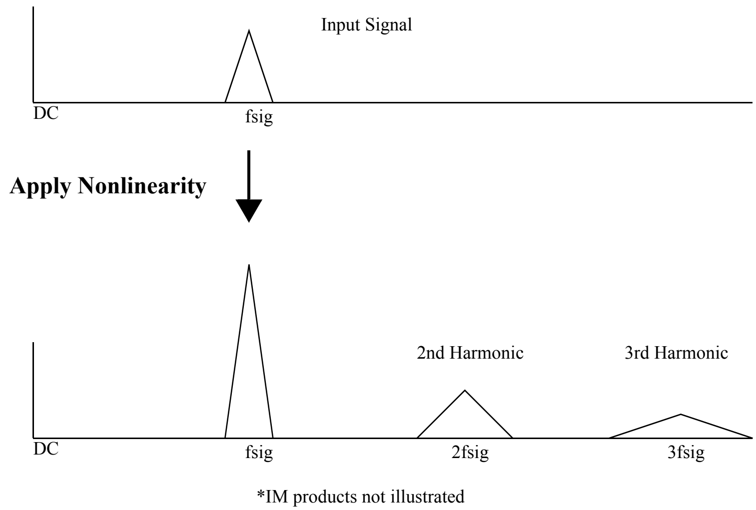


Figure 2-25. Frequency expansion due to 3rd order nonlinearity

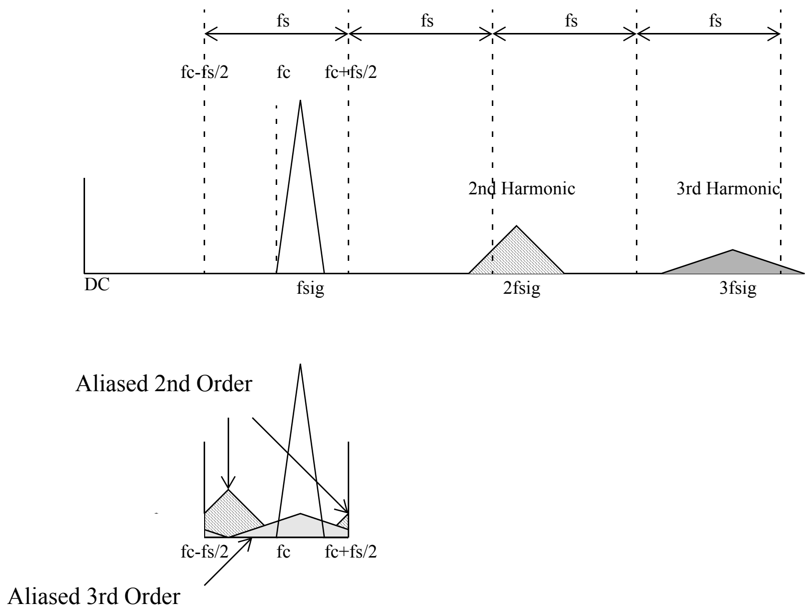


Figure 2-26. Aliasing of a 3rd order nonlinearity

If the largest signal frequency is sufficiently small, the expanded signal bandwidth falls entirely within the sampling frequency band and all is well. If that is not the case, however, special accommodations may have to be made if the input signal level is sufficiently high to generate significant harmonics. This is usually the case when working with complex envelope signals with non-zero center frequencies.

One approach to reducing the effects of the aliased harmonics is to temporarily expand the sampling frequency bandwidth, apply the nonlinearity, filter out the content outside the original sampling frequency bandwidth, then downsample back to the original sampling frequency bandwidth. VSS calls this process anti-aliasing resampling. Figure 2-27 illustrates these steps.

One limitation of anti-aliasing resampling is that the maximum harmonic generated must be known to determine the upsampling rate. Because of this, only the nonlinear blocks that use polynomials to model their nonlinearity currently use anti-aliasing resampling. The behavioral amplifiers AMP_B, AMP_BV, AMP_F and VGA_F, and the mixers MIXER_B, MIXER_F and MIXER_S support this behavior.

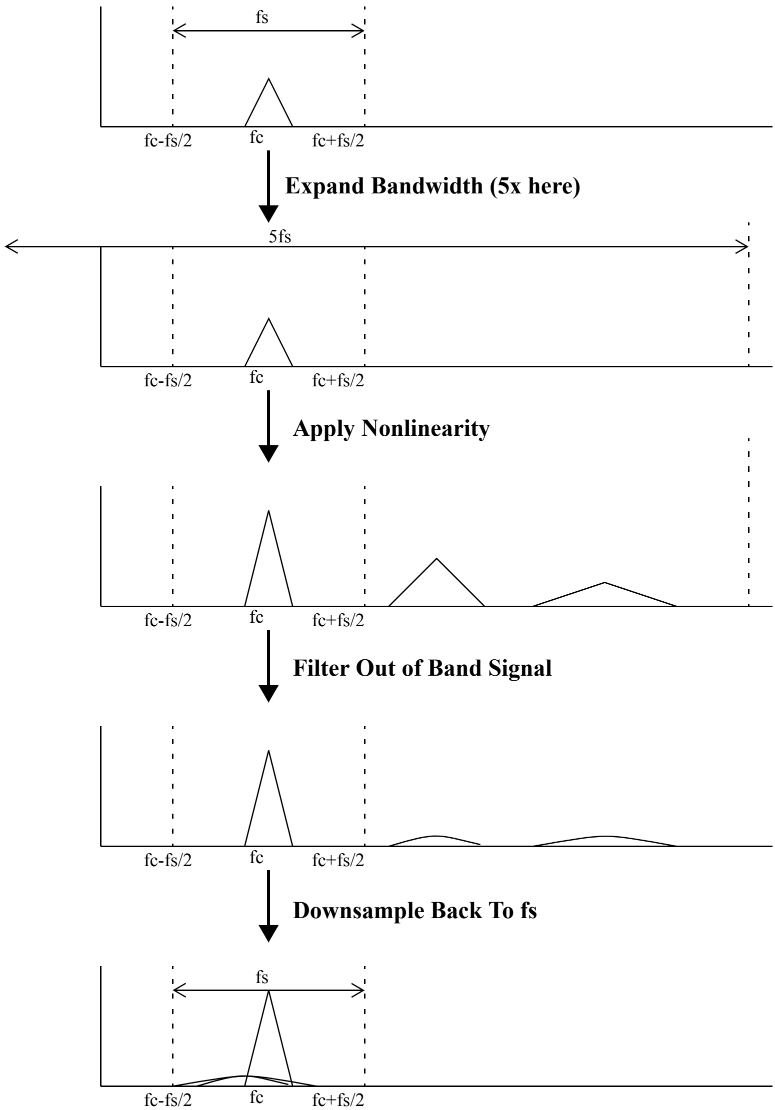


Figure 2-27. Anti-aliasing resampling

The biggest limitation in using anti-aliasing resampling is the amount of resampling that may be needed. This amount is related to the ratio of the largest signal frequency to the sampling frequency bandwidth. The larger the ratio, the more resampling is required.

This problem is most apparent when working with narrowband signals at high center frequencies. Take for example a 2 MHz bandwidth signal centered at 2 GHz. With a oversampling rate of 10, this signal can be efficiently modeled as a complex envelope signal with a sampling frequency of 20 MHz and a center frequency at 2 GHz. If this signal were to pass through a 3rd order nonlinearity, the largest signal component generated by the nonlinearity would be $3 \cdot 2.001$ GHz or 6.003 GHz.

To expand the sampling frequency so the sampling frequency band includes 6.003 GHz, we would need to upsample by:

$$UPRATE = \frac{(f_{max} - f_c) \cdot 2}{f_s} \tag{45}$$

or 4003!

VSS uses two approaches to avoid this problem. The first approach uses a proprietary algorithm to limit the upsampling rate needed for a given polynomial order. This algorithm results in a maximum upsampling rate of 7 for 3rd order nonlinearities and 18 for 5th order nonlinearities. The upsampling rate for 5th order nonlinearities can often be reduced to 9.

The second approach is to use a different nonlinearity implementation when the signal is narrowband. The behavioral amplifiers AMP_B and AMP_BV, and the mixers MIXER_B, MIXER_F and MIXER_S support this approach. In this case the block replaces the polynomial based nonlinearity with an AM/PM conversion table based nonlinearity that is derived from passing a single tone signal near the center frequency through the original polynomial based nonlinearity.

2.6.2 Saturation Effects

NOTE. This section applies to time domain simulations. The "*Saturation Effects*" sub-section of the "*RF Inspector*" section describes the saturation effects in RF Inspector simulations.

The behavioral amplifiers AMP_B, AMP_BV, AMP_F and VGA_F, and the mixers MIXER_B, MIXER_F and MIXER_S use as their primary nonlinearity a 3rd or 5th order polynomial. This polynomial allows gain, P1dB, IP2 and IP3 to be modeled up to the P1dB point. To model saturation effects, however, the instantaneous voltage must be modified differently. For these blocks a second polynomial is used to taper the output voltage to the saturation voltage once the input voltage exceeds a threshold value. See the AMP_B documentation for more information.

However, the transition from one polynomial to the other at the threshold is in itself a nonlinear operation, and results in additional harmonic and intermodulation products being generated. These products will typically not conform to any predictable pattern, so cannot be relied upon to determine IM product behavior near the saturation point.

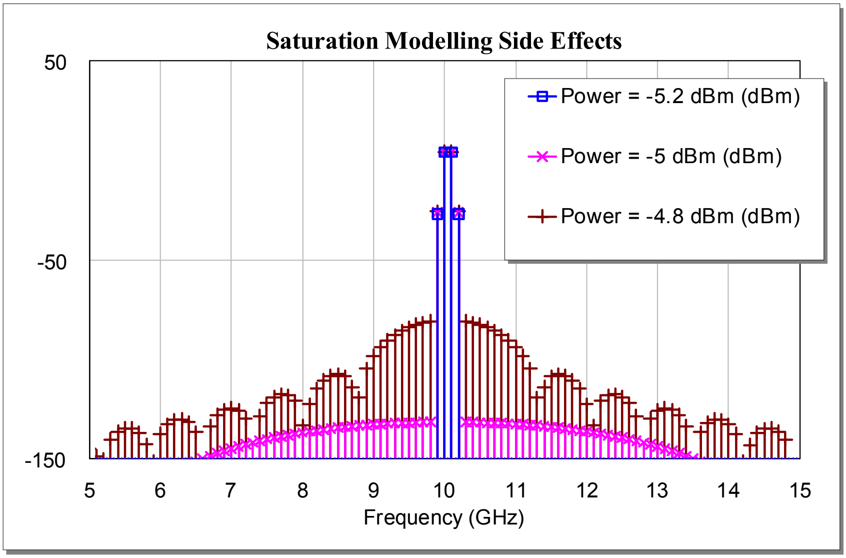


Figure 2-28. Side effects of saturation modeling

Figure 2-28 illustrates the generated IM products. The input signals are two tones, 10 and 10.1 GHz, fed into AMP_B. The blue plot is the output of the amplifier when the tone power is -5.2 dBm. For this case the instantaneous voltages of the input signal fall within the tolerance so the output is based solely on the base polynomial. In the pink plot, the tone power has been increased to -5 dBm, which results in some instantaneous voltages exceeding the threshold. This is evident from the appearance of the frequency content above 10.2 GHz and below 9.8 GHz. For the brown plot, the tone power has been increased to -4.8 dBm. Note the large increase in frequency content outside 9.8 to 10.2 GHz.

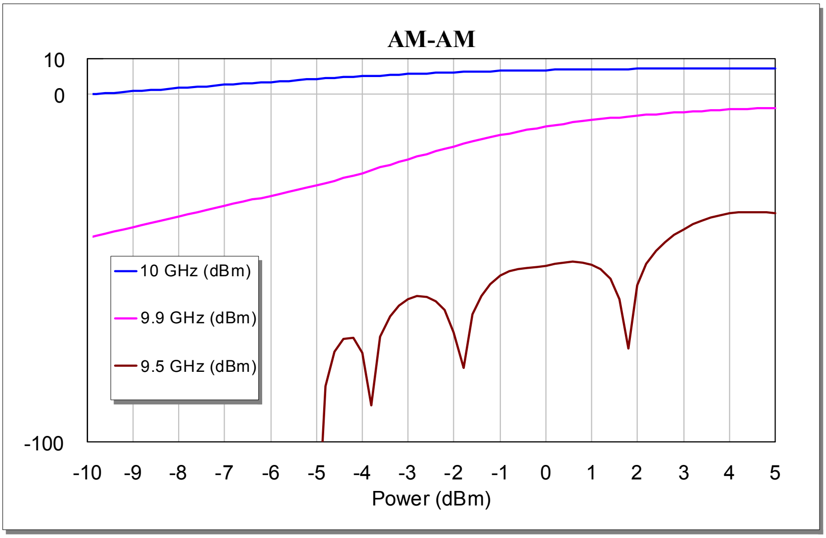


Figure 2-29. Side effects of saturation modeling in an AM-AM plot

Figure 2-29 is another view of the effects. The brown curve is the 9.5 GHz AM-AM plot, which should be negligible (< -300). When the saturation polynomial begins to take effect, however, that frequency suddenly appears.

NOISE MODELING IN VSS

3

This chapter describes the noise modeling capabilities of Visual System Simulator (VSS).

3.1 MODELING NOISE IN VSS

The two main forms of noise modeled in VSS are *thermal circuit noise* (which is called circuit noise in VSS) and *channel noise*. Both forms are based on *white Gaussian noise*. Circuit noise is primarily specified as a noise temperature, a noise figure, or equivalent noise voltage or current sources. Channel noise is typically specified in noise power spectral density (noise PSD) which are typically in units of dBm/Hz.

RF Budget Analysis simulations primarily work with circuit noise, which is frequency dependent, although they also provide limited support for channel noise. Time domain simulations are most effective when modeling white channel noise.

Time domain simulations also support *phase noise*. VSS phase noise modeling is covered in the "*Phase Noise*" section.

Related to noise are the signal to noise and energy to noise ratios used in VSS. These are discussed in the "*SNR, Eb/N0 and Es/N0*" section.

A limitation to modeling noise in the time domain occurs when working with complex envelope signals whose sampling frequency band crosses into negative frequencies. This is discussed in the "*Noise and Negative Frequency Folding*" section.

3.2 CIRCUIT NOISE

The RF blocks that support thermal noise generation have a NOISE parameter which allows you to control which simulations include noise. The NOISE parameter has four options:

- RF Budget only
- RF Budget + Time Domain
- Noiseless
- Auto

The default setting is “Auto”. With this setting, the noise modeling setting is taken from the “**Default RF block noise modeling**” setting on the “**RF Settings**” tab of the Options dialog box for the system diagram. This allows you to change the RF noise modeling settings for all RF blocks in a system diagram at one time. You still have the option of changing settings for individual blocks using the NOISE parameter.

In RF Budget Analysis simulations, noise is modeled by a block when either “RF Budget only” or “RF Budget + Time Domain” is selected. In time domain simulations, noise is modeled by a block when “RF Budget + Time Domain” is selected.

The different RF blocks determine how much noise is to be added. Noise is added only to the output signal. Because of this, the noise added represents the noise that is seen by the load. The noise is computed on a frequency dependent basis.

There are actually two different values computed for the noise to be added. The first value is the noise that would be added if the block were connected to a matched load, and is computed as noise power spectral density (noise PSD). This noise is used for the RF budget analysis noise figure measurements. It also forms the basis of the second computed value.

The second value is the noise voltage spectral density (noise VSD) that would be seen by the load if the input to the block were noiseless. If impedance mismatch modeling has been enabled in the Options dialog box of the system diagram, this value is adjusted to accommodate any impedance mismatch with the load.

3.2.1 Linear RF Blocks

The linear RF blocks, which include the circuit filter blocks in the **Bandpass**, **Bandstop**, **Highpass** and **Lowpass** sub-categories of the **Filters** category, determine the noise to be added using noise correlation matrices. LIN_S obtains a normalized noise correlation matrix $\mathbf{C_i}$ directly from the Microwave Office simulation. The other blocks obtain $\mathbf{C_i}$ from their Y matrix using the following:

$$C_i(i,j) = \frac{1}{2} \cdot (Y(i,j) + Y(j,i)^*) \tag{1}$$

If impedance mismatch modeling is enabled, $\mathbf{C_i}$ is adjusted for any voltage divisions on the inputs due to branches.

For a 2-port with port 2 the output port, the noise PSD added at the output, *NN*, is:

$$NN = \frac{4 \cdot k \cdot \mathrm{T} \cdot |VV|}{R_{Out}} \tag{2}$$

$$VV = \begin{bmatrix} z(2,1) & z(2,2) \end{bmatrix} \begin{bmatrix} C_i(1,1) & C_i(1,2) \\ C_i(2,1) & C_i(2,2) \end{bmatrix} \begin{bmatrix} z(2,1)^* \\ z(2,2)^* \end{bmatrix} \tag{3}$$

$$z = \begin{bmatrix} Y(1,1) + \dfrac{1}{Z_S} & Y(1,2) \\ Y(2,1) & Y(2,2) + \dfrac{1}{Z_L} \end{bmatrix}^{-1} \tag{4}$$

The noise VSD added at the output is computed directly from NN:

$$VN = \sqrt{NN \cdot R_{Out}} \tag{5}$$

Z_S is the impedance seen looking into the source's output port.

For the matched load case, Z_L is the conjugate of Z_S. R_{Out} is obtained from:

$$R_{Out} = Re\left\{\frac{\text{ZOUTP}}{1-\Gamma_{Out}}\right\} \tag{6}$$

ZOUTP is the characteristic impedance of the output port.

For computing noise VSD, Z_L is the load impedance and R_{Out} is the real component of ZOUTP.

The temperature T in equation 2 is either the temperature parameter from the block if the block has a temperature (RF_START has a temperature parameter) or is the ambient system temperature setting from the **RF Settings** tab of the Options dialog box of the system diagram.

3.2.2 Amplifiers

There are several ways of specifying noise added by amplifier blocks. These include noise figure NF, equivalent input referred noise voltage, and equivalent input referred noise current. The voltage and current are RMS spectral densities and are normally specified in units of nV/sqrt(Hz) and pA/sqrt(Hz), respectively.

The amplifiers AMP_B, AMP_BV, AMP_F and VGA_F support all three methods. AMP_F supports the specification of frequency dependent noise values. NL_S obtains its noise information from the MWO nonlinear noise simulation.

If noise figure is specified, it is converted to an equivalent input referred noise current using the following:

$$In = \sqrt{\frac{(10^{NF/10}-1)\cdot 4\cdot k\cdot T_0}{Re\{\text{ZINP}\}}} \tag{7}$$

where ZINP is the characteristic impedance of the amplifier's input port.

The amplifiers interpret input referred noise voltage and current using the following model:

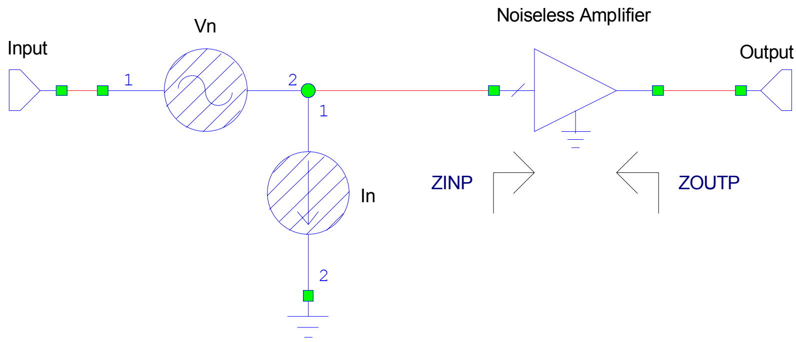


Figure 3-1. Interpretation of Vn and In in VSS amplifiers.

Note that the voltage and current sources are not correlated.

Noise PSD and noise VSD are computed using noise correlation matrices as for the linear RF filters. The normalized noise correlation matrix $\mathbf{C_i}$ is computed as:

$$C_i = \frac{1}{4 \cdot k \cdot T_0} \cdot (C_{i,\,Vn} + C_{i,\,In}) \tag{8}$$

$$C_{i,\,Vn} = \begin{bmatrix} V_n^2 \cdot Y(1,1) \cdot Y(1,1)^* & V_n^2 \cdot Y(1,1) \cdot Y(2,1)^* \\ V_n^2 \cdot Y(2,1) \cdot Y(1,1)^* & V_n^2 \cdot Y(2,1) \cdot Y(2,1)^* + Y(2,2) \cdot Y(2,2)^* \end{bmatrix} \tag{9}$$

$$C_{i,\,In} = \begin{bmatrix} In^2 & 0 \\ 0 & 0 \end{bmatrix} \tag{10}$$

The Y matrix is obtained from the linear voltage gain, S11 and S22 parameters of the amplifier.

3.2.3 Mixers

The VSS behavioral mixers MIXER_B and MIXER_F use single sided noise figure to specify noise. The noise figure is converted to an equivalent input referred noise current similar to the amplifiers. However, because the noise figure is single sided the conversion is:

$$In = \sqrt{\frac{(10^{NF/10} - 2) \cdot 4 \cdot k \cdot T_0}{Re\{\mathrm{ZINP}\}}} \qquad (11)$$

3.2.4 Additional Notes

The following are additional items to note when working with circuit noise.

In time domain simulations VSS adds noise directly to the samples as the noise is generated. While simplifying the simulation, it presents several problems. One problem occurs when the sampling frequency band crosses DC - noise folding occurs. This problem is discussed in detail in the "*Noise and Negative Frequency Folding*" section.

The second problem occurs when combining signals whose noise levels represent the ambient noise temperature.

COMBINING SIGNALS WITH AMBIENT NOISE

Because noise is directly added to the signal samples as the samples are generated, the noise cannot later be separated from the signal. In most cases this is not an issue.

However, in the case of a passive lossy combiner this becomes a problem when the signals being combined have noise near the ambient noise temperature. In a physical passive combiner, if the input signals all have noise levels at the ambient noise temperature, the noise level at the output of the combiner is also at the ambient noise temperature.

Unfortunately in VSS time domain simulations, because the noise is incorporated directly into the signal samples, the noise at the output of the combiner is effectively the sum of the noise at all the inputs. For a two input combiner with both inputs having a noise temperature of 290 K the output noise will be at 580 K, or 3.01 dB too high.

The workaround is to only generate noise on one of the inputs to the combiner, preferably the primary input.

3.3 CHANNEL NOISE

Channel noise is primarily used when performing BER measurements on a system design. The additive white Gaussian noise channel block AWGN models white Gaussian channel noise.

Note that when circuit noise samples are being generated in a time domain simulation, the circuit noise is automatically included in the channel noise.

3.4 PHASE NOISE

The following blocks support the generation of phase noise:

Block	Description
Channels > Phase Noise > PHSNOISE_CH	Phase Noise Channel
RF Blocks > Sources > TONE	Tone(s) Source
RF Blocks > Sources > Simulation Based > OSC_S	Oscillator with Optional Phase Noise Effects
Sources > Noise > PHASENS	Phase Noise Source

With the exception of OSC_S, phase noise is specified as a phase noise mask in dBc/Hz at specific frequency values. The mask can be entered either in a text data file or as a vector of frequency-dBc/Hz pairs of values. OSC_S obtains its phase noise mask directly from the Microwave Office phase noise simulation.

The phase noise channel block PHSNOISE_CH is useful for modeling phase noise from an LO that would be added to a narrowband signal by an ideal mixer (a mixer that only generates the 1,1 spurs) without incorporating a mixer in the design. For example, a QAM signal generated by QAM_SRC with center frequency set to 1 GHz passing through a PHSNOISE_CH block would produce similar results to a QAM signal at baseband passing through an ideal mixer with a 1 GHz LO with a similar phase noise mask:

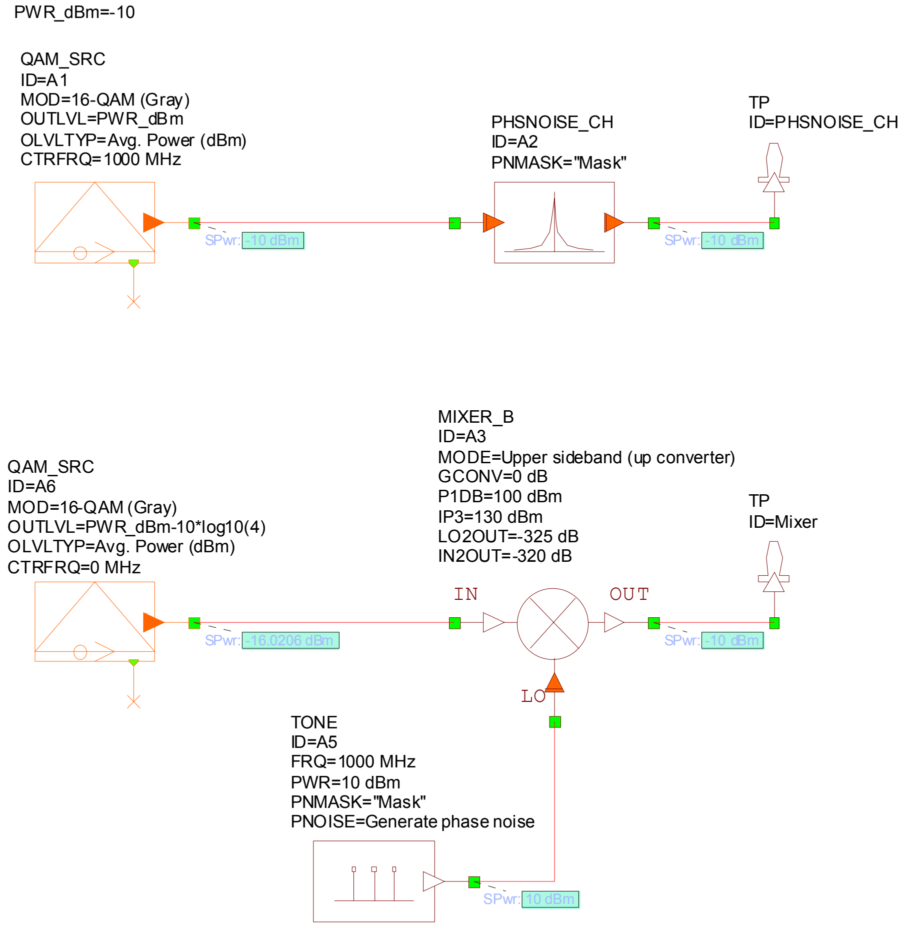


Figure 3-2. Use of PHSNOISE_CH with a modulated signal to model baseband upconversion.

Note the adjustment to the OUTLVL parameter of the baseband QAM_SRC. This is to reflect the fact that OUTLVL for QAM_SRC at baseband represents the average power of each channel, which are then combined by the upconversion mixer.

3.4.1 Generating Phase Noise

VSS generates phase noise by applying an FIR filter to a white Gaussian noise source. The FIR filter is used to shape the noise to match the phase noise mask.

PHSNOISE_CH, TONE and OSC_S then add the phase noise samples to the phase of a complex signal to apply the phase noise to the signal.

A windowed FIR frequency sampling design algorithm is used to design the FIR filter. The windowing is used to reduce the sidelobes inherent in the FIR frequency sampling design technique. The "*Windowed FIR Filters*" section of the "*RF Modeling in VSS*" chapter further discusses windowed FIR frequency sampling design.

3.4.2 Measuring Phase Noise

There are two VSS measurements for working with phase noise in time domain simulations: Swept Integrated Phase Noise INTG_PHS_NOISE and Phase Noise (dBc/Hz, log frequency) PHS_NOISE. Both measurements are found in the **System > Noise** measurement category.

INTG_PHS_NOISE measures phase noise within a frequency band. It can present the phase noise as phase jitter, time jitter, or dBc. The frequency band is specified relative to a carrier frequency.

PHS_NOISE displays phase noise in dBc/Hz versus frequency. The frequency axis should normally be set to a log scale.

Both measurements operate by performing a power spectrum measurement, converting the measured spectrum to dBc/Hz. By default they utilize a large number of FFT bins along with windowing and cumulative averaging.

Measuring phase noise can be tricky, particularly when measuring close to the carrier. There are two main reasons for this. First is the issue of resolution bandwidth. The closer to the carrier the frequency of interest, the smaller the resolution bandwidth must be, and the larger the number of FFT bins required to perform the spectrum measurement. The order of the filter used to shape the phase noise must also be taken into account, as a larger filter order may be needed.

The second problem is due to the phase noise making the signal non-periodic. When an FFT is performed to obtain the power spectrum, the signal is truncated. This effectively spreads the power of the signal over the bins of the FFT. The solution to this is to apply a window function to the signal prior to the

FFT. The window essentially smooths the edges of the time domain signal, reducing the effects of truncation and thereby reducing the spreading of the power. The drawback is that the application of the window function spreads the carrier due to the wide main lobe. The net result is a more accurate spectrum in the majority of the sampling frequency band, but a less accurate spectrum near the carrier.

Figure 3-3 compares a PHS_NOISE measurement with windowing set to the default Blackman-Harris 4 term window function, a PHS_NOISE measurement with no windowing applied, and the phase noise mask used to generate the signal. With windowing, the carrier has been spread to about 3.6 kHz as can be seen by the wide bump ending at 3.6 kHz. Beyond 3.6 kHz, however, the windowed curve follows the phase noise mask closely.

Without windowing, the carrier has been spread to about 1 kHz, allowing a measurement of the phase noise starting from that point. However, the curve does deviate from the phase noise mask until approximately 200 kHz.

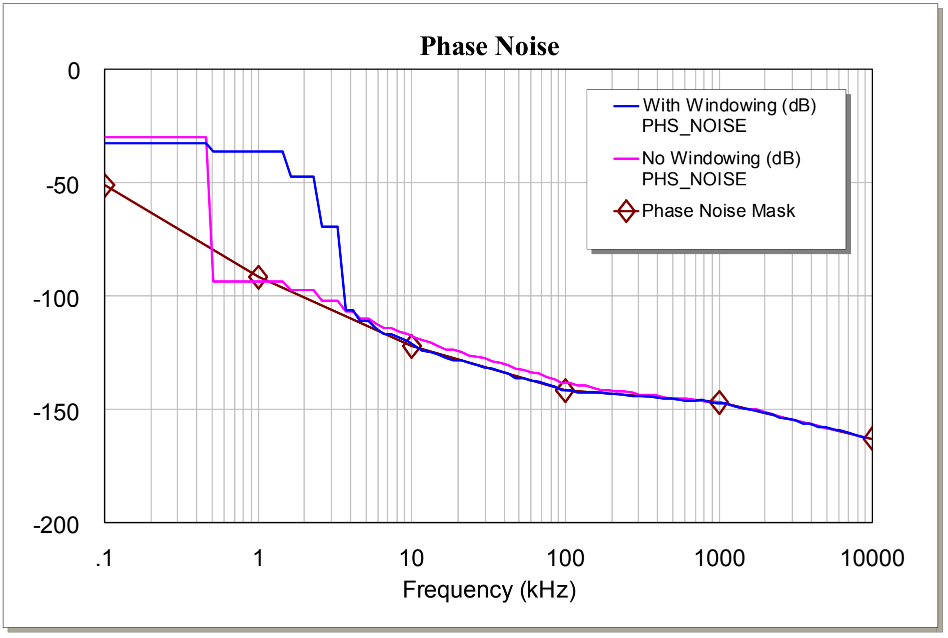


Figure 3-3. Comparison of phase noise measurements with and without windowing

In the PHS_NOISE measurement, the range of frequencies measured is limited by default to ignore the main lobe spectral spreading due to the windowing and to ignore the frequencies near the edges of the sampling frequency band. The frequencies near the edges of the sampling frequency band are generally not meaningful when performing RF simulations, as they contain artifacts such as resampling filter transition bands used by nonlinear and circuit filter models, or the wrap-around nature of digital filters.

The online help for PHS_NOISE and INTG_PHS_NOISE contains more information on using these measurements.

3.5 SNR, EB/N0 AND ES/N0

VSS supports the three noise ratios SNR (signal-to-noise ratio), Eb/N0 (bit energy to noise PSD ratio) and Es/N0 (symbol energy to noise PSD ratio) in several different ways.

All three ratios base the signal power component on the *static signal power property*. This is a signal property that represents the average signal power as the signal progresses through the link. The property is static because it is determined at the start of each simulation sweep and does not vary during the sweep.

The signal source blocks are generally blocks that have a power level parameter. These include the RF tone source block TONE and the various transmitters such as QAM_TX or QAM_SRC. The signal power property is updated by blocks along the signal path that change the signal power level, such as amplifiers, filters and mixers.

While used to convey the signal power portion of the noise ratios, the signal power property's primary use is for automatic gain control in receivers. The receiver blocks use the signal power property to determine an appropriate scaling for demodulation and detection of the received signal.

The static signal power property at each output port can be displayed using the SIGPWR annotation as well as the static signal power properties measurement PWR_PROP, found under the **System > Tools** category.

Eb/N0, Es/N0 and one form of SNR also use the *static channel noise PSD property* to determine their noise component. The channel noise PSD is an estimate of the power spectral density of channel noise generated as it passes through the link. The channel noise PSD property can be displayed using the NOISEPSD annotation with output type set to “Generated Noise PSD”. It can also be displayed using the NOISE_PROP measurement in the **System > Tools** category.

The channel noise PSD property is adjusted differently depending on the block it passes through. For filters, the property is adjusted by applying the average voltage gain over the signal bandwidth, centered at the center frequency. The signal bandwidth is the sampling frequency divided by the oversampling rate or samples per symbol associated with the signal.

For nonlinear blocks such as the RF amplifiers and mixers, the property is adjusted by creating a set of samples with average power equal to the noise power over the sampling frequency, passing those samples through the nonlinearity, then converting the average power of the modified samples back to noise PSD over the sampling frequency.

3.5.1 SNR

VSS supports two forms of signal-to-noise ratio measurements. The first form normally represents the frequency dependent circuit SNR and is available only in RF Budget Analysis through the cascaded signal to noise ratio measurement C_SNR found in the **System > RF Budget Analysis** category. This SNR uses for its noise component the frequency dependent node noise temperature property. It can also optionally include channel noise.

The second form of SNR is similar to Eb/N0 and Es/N0. It is normally used when performing BER measurements in time domain simulations and is only available in time domain simulations. This form of SNR is frequency independent and uses the static channel noise PSD property for its noise component. The noise power is computed over the signal bandwidth, which is the sampling frequency divided by the signal’s oversampling rate. Equation 12 illustrates this computation:

$$SNR_{BER} = \frac{SIGPWR}{NOISEPSD \cdot f_s / OVRSMP} \tag{12}$$

The second form of SNR can be displayed using the EsN0_PROP measurement found in the **System > Tools** category.

3.5.2 Eb/N0 and Es/N0

Eb/N0 and Es/N0 are normally used when working with modulated signals, particularly when performing bit error rate measurements in the time domain. They are only available in time domain simulations. Eb/N0 and Es/N0 can be displayed using the EsN0 annotation or the EsN0_PROP measurement found in the **System > Tools** category.

One of the features of VSS is the ability to automatically determine Eb/N0 and Es/N0 at a BER meter based on the signal and noise characteristics of the received signal. This can, of course, be overridden at the BER meter by specifying explicit values for the SWPVAR parameter.

Note that when you use a transmitter and AWGN channel, Eb/N0 or Es/N0 at the BER meter will include the effects of all the blocks between the transmitter and receiver. Therefore, the Eb/N0 and Es/N0 values displayed in a BER may not match the ratio of the transmitter's power to the AWGN's channel, particularly if the signal encounters compression.

Eb/N0 and Es/N0 use the static channel noise PSD property for their noise component. Equations 13 and 14 illustrate the computations.

$$E_b/N_0 = \frac{\left(\frac{SIGPWR}{OVRSMP \cdot BITSYM}\right)}{NOISEPSD \cdot Scale} \qquad \begin{array}{rcl} Scale & = & 1 \text{ for complex signals} \\ Scale & = & 2 \text{ for real signals} \end{array} \tag{13}$$

$$E_s/N_0 = \frac{\left(\frac{SIGPWR}{OVRSMP}\right)}{NOISEPSD \cdot Scale} \qquad \begin{array}{rcl} Scale & = & 1 \text{ for complex signals} \\ Scale & = & 2 \text{ for real signals} \end{array} \tag{14}$$

3.6 NOISE AND NEGATIVE FREQUENCY FOLDING

Negative frequency folding occurs when working with complex envelope signals whose center frequency is less than 1/2 the sampling frequency. When that occurs, part of the sampling frequency band contains negative frequencies. Conceptually, negative frequency content is equivalent to the complex conjugate of the corresponding positive frequency content. When the center frequency is greater than 0, the default behavior of VSS spectrum measurements is to automatically convert negative frequency content to the equivalent positive frequency content, or to "fold" the negative frequency content into the positive frequencies.

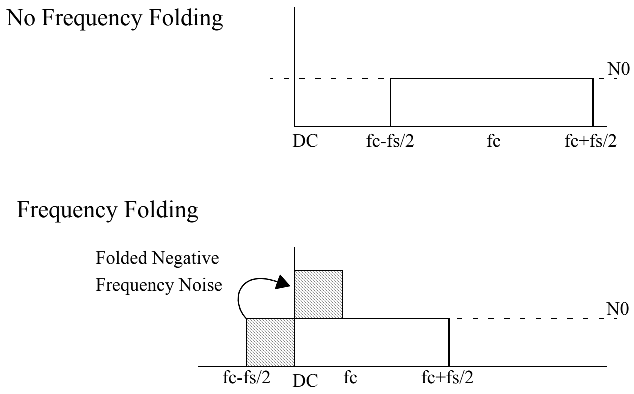


Figure 3-4. Noise and negative frequency folding

However, when working with noise, frequency folding results in a doubling of noise PSD where the frequencies have been folded. The noise is no longer white but has a 3 dB step. To avoid this scenario when modeling noise, the center frequency should either be 0 or greater than 1/2 the sampling frequency to avoid negative frequency folding.

The "*Negative Complex Envelope Frequencies*" section of the *Simulation Basics* chapter explores negative frequencies in detail.

INDEX

A

aliasing *1-22*, *2-63*
allpass IIR filter, windowed FIR+ *2-54*
alphabet size *1-12*
amplifiers
  VSS *2-5*
analog signals *1-13*
analysis, frequency *2-8*, *2-18*
automatic VSS configuration *1-28*
averaging, spectrum *1-29*, *1-32*

B

bandwidth expansion, nonlinear modeling *2-63*
baseband I/Q channel signal *1-17*
baseband signals *1-14*
BER
  design parameters *1-4*
BER_EXT
  design parameters *1-4*
bit energy-to-noise PSD ratio *3-11*
budget analysis *2-8*
Budget Analysis, RF
  simulation basics *2-9*

C

center frequencies
  signals *1-20*
center frequency *1-13*
channel noise *3-1*, *3-7*
channel noise PSD property *3-11*
channel power
  measuring *1-37*
circuit filters *2-47*
  FRQALIGN *2-60*
  IMPL *2-61*
  implementation selection algorithm *2-61*
  MAXNPOLE *2-57*
  NFREQ *2-62*
  NFREQFIR *2-62*
  number of frequencies *2-62*
  UPRATE *2-57*
circuit noise, VSS *3-1*, *3-2*
Combiners, noise in VSS *3-6*
complex envelope *1-13*
  benefits and limitations *1-19*
complex envelope, working with *1-15*
complex envelopes, RF Budget Analysis simulations *1-20*
configuration
  automatic *1-28*
context menu
  RF Inspector *2-27*
controlling sweeps *1-6*
copying
  RF Inspector *2-27*

D

data paths
  primary *1-27*, *2-14*
data rate *1-12*, *1-20*
data signals *1-11*
design parameters
  sweeps *1-4*
Details section, RF Inspector *2-21*
DFT *1-29*
digital signals *1-12*
digital symbol *1-12*
DIS icon, RF Inspector *2-25*
discrete Fourier transform *1-29*

E

Eb/N0 *3-11*, *3-13*
Es/N0 *3-11*, *3-13*
estimation, power spectrum
  VSS *1-32*

F

feedback loops
  RF Budget Analysis *2-15*
  RF Inspector *2-38*
FER_EXT
  design parameters *1-4*
filter issues, VSS *2-47*
filters
  FIR *2-49*

FRQALIGN *2-60*
IIR *2-56*
IMPL *2-61*
implementation selection algorithm *2-61*
linear RF *2-3*
MAXNPOLE *2-57*
NFREQ *2-62*
NFREQFIR *2-62*
number of frequencies *2-62*
RF modeling, VSS *2-47*
UPRATE *2-57*
windowed FIR *2-52*
windowed FIR+allpass IIR *2-54*
filters, VSS circuit *2-47*
Finite Impulse Response (FIR) filter *2-47*, *2-49*
FIR filters *2-47*, *2-54*
frequency sampling *2-50*
windowed *2-52*
fixed point signals *1-26*
folding, negative frequency *3-13*
Fourier transforms *1-29*
frequencies
negative complex envelope *1-34*
frequency analysis *2-8*, *2-18*
frequency dependent signal properties *2-10*
frequency expansion, nonlinear modeling *2-63*
frequency resolution *1-34*
frequency sampling, FIR filters *2-50*
frequency set *2-9*
FRQALIGN, circuit filters *2-60*

G

gain
noise power, VSS *2-12*
Gaussian noise *3-1*
generic signals *1-26*

H

Heritage Window, RF Inspector *2-22*

I

I/Q channel signal *1-17*
IIR circuit filters
FRQALIGN *2-60*
MAXNPOLE *2-57*
UPRATE *2-57*
IIR filters *2-47*, *2-54*
impedance mismatch modeling *2-40*
frequency analysis *2-10*
interfacing to non-RF blocks *2-46*
RF Inspector *2-38*
voltage corrections *2-44*
impedances
determining in VSS *2-43*
source and load, VSS *2-45*
IMPL
circuit filters *2-61*
Infinite Impulse Response (IIR) filter *2-47*, *2-56*
input node
primary *1-27*
INT icon, RF Inspector *2-25*

L

leaking, spectral *1-30*
LIN_F *2-4*
LIN_S *2-4*
linear RF filters *2-3*
LO icon, RF Inspector *2-25*
load impedance, VSS *2-45*

M

main signal *1-27*
master VNA_LS *1-6*
MAXNPOLE, circuit filters *2-57*
measurements
swept, VSS *1-7*
measuring channel power *1-37*
MIXER_B
RF Budget Analysis *2-15*
MIXER_F
RF Budget Analysis *2-15*
MIXER_S
RF Budget Analysis *2-15*
mixers
RF Budget Analysis *2-15*
RF Inspector *2-39*
VSS *2-6*
modeling
basics *2-43*
nonlinear in VSS *2-63*
modulated signals
impedance mismatch modeling *2-46*
RF Inspector *2-28*

N

negative complex envelope frequencies *1-34*
negative frequency folding *3-13*
NFREQ, circuit filters *2-62*
NFREQFIR, circuit filters *2-62*
node, VSS *1-1*
noise *3-13*
amplifiers, VSS *3-4*
channel *3-1*, *3-7*
circuit, VSS *3-2*

combiners *3-6*
Gaussian *3-1*
linear RF blocks, VSS *3-3*
mixers, VSS *3-6*
phase, VSS *3-7*
thermal in VSS *3-1*
thermal, VSS *3-2*
white *3-1*
noise figure, mixers *2-15*
noise power gain, VSS *2-12*
noise temperature, VSS *2-11*
nonlinear amplifiers
VSS *2-5*
nonlinear modeling
aliasing *2-63*
frequency expansion *2-63*
saturation effects, VSS *2-67*
VSS *2-63*
non-RF blocks, interfacing to for impedance mismatch modeling *2-46*

O

operating point gain, VSS *2-10*
oversampling *1-13*, *1-20*
oversampling rate *1-12*

P

P1dB
RF Inspector *2-40*
passband signal *1-14*
periodgram *1-32*
phase noise
generating *3-9*
measuring *3-9*
phase noise modeling, VSS *3-7*
Plot active sweep *1-11*
Plot all traces, VSS *1-11*
ports, VSS *1-1*
power
measuring channel *1-37*
power spectrum
estimation, VSS *1-32*
Welch method *1-32*
primary data path *1-27*, *2-14*
primary input node *1-27*
properties
signal *1-26*
static channel noise PSD *3-11*
static signal power *3-11*

R

RBW *1-34*
real passband signal *1-14*
real signals *1-14*
resolution
frequency *1-34*
resolution bandwidth *1-34*
RF blocks *2-2*
RF Budget Analysis
simulation basics *2-9*
RF budget analysis *2-8*
RF Budget Analysis, complex envelopes and *1-20*
RF Inspector *2-18*
modulated signals *2-28*
settings *2-33*
RF Inspector dialog box *2-20*
RF_START *2-46*

S

samples per symbol *1-12*, *1-20*
sampling frequencies
signals *1-20*
sampling frequency *1-3*
selecting *1-23*
sampling frequency band *1-22*
sampling theory *1-22*
saturation
RF Inspector *2-40*
saturation effects, nonlinear modeling, VSS *2-67*
Select with tuner, VSS *1-11*
SER
design parameters *1-4*
SER_EXT
design parameters *1-4*
SIG icon, RF Inspector *2-25*
signal bandwidth *1-20*, *1-24*
signal classification
RF Inspector *2-25*
signal heritage, RF Inspector *2-22*
signal power property *3-11*
signal properties *1-26*
frequency dependent *2-10*
static *1-27*
signals
aliasing *1-22*
analog *1-13*
center frequencies *1-20*
data *1-11*
digital *1-12*
fixed point *1-26*
generic *1-26*
main *1-27*
sampling frequencies *1-20*
selecting sampling frequency *1-23*
signal bandwidth *1-24*
signals samples per symbol *1-20*
signals, data rate *1-20*

signals, oversampling *1-20*
signal-to-noise ratio *3-11*
simulations
basics *1-1*
RF Budget Analysis *1-2*
sweeping *1-3*
time domain *1-2*, *1-29*, *2-18*
SNR *3-11*
SNR, VSS *3-12*
source impedance, VSS *2-45*
spectral analysis *1-29*
spectral leakage *1-30*
spectrum
power in VSS *1-32*
VSS *1-29*
Welch method *1-32*
spectrums
leaking *1-30*
static channel noise PSD property *3-11*
static properties
channel noise PSD *3-11*
signal power *3-11*
static signal power property *3-11*
static signal properties *1-27*
sweep
VSS *1-3*
sweeping simulations *1-3*
sweeps
controlling *1-6*
design parameters *1-4*
measurements, VSS *1-7*
swept measurements, VSS *1-7*
swept variables
VSS *1-4*
SWPVAR
design parameters *1-4*
symbol energy-to-noise PSD ration *3-11*
symbol period *1-12*
System Simulation Options
RF Inspector tab *2-33*

T

thermal noise, VSS *3-1*
time domain simulations *1-2*, *1-29*, *2-18*
time step *1-3*
transforms, Fourier *1-29*

U

UPRATE, circuit filters *2-57*

V

variables
swept,VSS *1-4*
VBW *1-34*
video bandwidth *1-34*
VNA_LS
design parameters *1-4*
VNA_LS, master *1-6*
VNA_SS
limitations *2-7*
voltage corrections, VSS *2-44*
VSA
design parameters *1-4*
VSS concepts *1-1*

W

Welch method *1-32*
white noise, VSS *3-1*
windowed FIR filters *2-52*
windowed FIR+allpass IIR *2-54*
windowed FIR+allpass IIR filter *2-54*
windowed+allpass IIR *2-54*
windowing
spectrum *1-32*
VSS *1-29*