RF Blockset™ 2
User's Guide

MATLAB®
& SIMULINK®

The MathWorks™
Accelerating the pace of engineering and science

How to Contact The MathWorks

www.mathworks.com	Web
comp.soft-sys.matlab	Newsgroup
www.mathworks.com/contact_TS.html	Technical Support
suggest@mathworks.com	Product enhancement suggestions
bugs@mathworks.com	Bug reports
doc@mathworks.com	Documentation error reports
service@mathworks.com	Order status, license renewals, passcodes
info@mathworks.com	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

RF Blockset™ User's Guide

Revision History

June 2004	Online only	New for Version 1.0 (Release 14)
August 2004	Online only	Revised for Version 1.0.1 (Release 14+)
March 2005	Online only	Revised for Version 1.1 (Release 14SP2)
September 2005	Online only	Revised for Version 1.2 (Release 14SP3)
March 2006	Online only	Revised for Version 1.3 (Release 2006a)
September 2006	Online only	Revised for Version 1.3.1 (Release 2006b)
March 2007	Online only	Revised for Version 2.0 (Release 2007a)
September 2007	Online only	Revised for Version 2.1 (Release 2007b)
March 2008	Online only	Revised for Version 2.2 (Release 2008a)

Contents

1 Getting Started

2 Modeling an RF System

3 Plotting Model Data

4 Block Reference

5 Blocks — Alphabetical List

A RF Blocks et™ Algorithms

B Modeling Mixers

C Examples

Index

1

Getting Started

Product Overview

RF Blockset™ software e xtends your Simulink ® modeling environment with a library of blocks for modeling R F systems that include RF filters, transmission lines, amplifiers, and mixers. For more information about creating and running Simulink models, see "Creating a Simulink Model" in the Simulink User's Guide.

You useRF Blockset blocks to represent the components of your RF system in a Simulink model. The blockset provides several types of component representations using network parameters (S, Y, Z, ABCD, H, and T format), mathematical descriptions, and physical properties.

In the Simulink model, you cascade the components to represent your RF architecture and run the simulation. During the simulation, all blocks are modeled using a time-domain, complex -baseband representation. This way of modeling results in fast simulation of the quadrature modeling schemes used in today's communication systems a nd enables compati bility with other Simulink blocks.

The blockset lets you visualize the network parameters of the blocks using plots and Smith ® Charts.

A validated Simulink model of an RF system can provide an executable specification for RF circuit design fo r wireless communication systems.

You can also use the blockset in conj unction withReal-Time Workshop ® software to automatically generate embeddable C code for real-time execution.

Required and Related Products

In addition to MATLAB® and Simulink®, you must have the following products installed to use RF Blockset™ software:

- RF Toolbox™ — Provides MATLAB functions for defining, simulating, and visualizing RF components.
- Signal Processing Toolbox™ — Provides MATLAB functions for filtering wireless communication signals.
- Signal Processing Blockset™ — Provides Simulink blocks for time-domain simulation of communication signals.

You can build sophisticated wireless communication system models by incorporating blocks from other blocksets, such as Signal Processing Blockset and Communications Blockset™.

The MathWorks™ provides several products that are especially relevant to the kinds of tasks you can perform with RF Blockset software. The following table summarizes the related products and describes how they complement the features of this product.

Product	Description
Communications Blockset	Simulink blocks for time-domain simulation of modulation and demodulation of a wireless communications signal.
Communications Toolbox™	MATLAB functions for signal modulation and demodulation.
Filter Design Toolbox™	MATLAB functions for filtering the modulated communication signal.

Product Demos

You can find interactive RF Blockset™ demos in the MATLAB® Help browser, as shown in the following figure.

This example shows you how to locate and open an RF Blockset demo:

1 Type `demos` at the MATLAB prompt to open the Help browser to the **Demos** tab.

2 Select **Blocksets > RF** in the **Demos** tab to see a list of demo categories.

3. Select a model, and click **Open this model** in the upper-right corner of the demo window to display the Simulink® model for this demo.

4. In the model window, select **Simulation > Start** to run the demo simulation.

RF Blockset™ Libraries

In this section...

Overview of RF Blockset™ Libraries

The RF Blockset™ product consists of the Physical and Mathematical libraries of components for modeling RF systems within the Simulink® environment. An RF model can contain blocks from both the Physical and Mathematical libraries. It can also include Simulink blocks and blocks from other blocksets, such as those described in “Required and Related Products” on page 1-3.

Opening RF Blockset™ Libraries

To open the main library window, type the following at the MATLAB® prompt:

```
rflib
```

The window is shown in the following figure. Each yellow icon in the window represents a library. Double-click an icon to open the corresponding library.

The Physical and Mathematical libraries are discussed in the following sections.

Note The blue icons take you to the MATLAB Help browser.

- Double-click the Demos icon to open the RF Blockset demos.
- Double-click the Info icon to open the RF Blockset documentation.

Physical Library

Use blocks from the Physical library to model physical and electrical components by specifying physical properties or by importing measured data. The Physical library includes several sublibraries, as shown in the following figure.

The following table describes the sublibraries and how they can be used.

Sublibrary	Description
Amplifiers	RF amplifie rs, specified using network parameters, noise data, and nonlinearity data, or a data file containing these parameters.
Ladder Filters	RF filters, specified using LC parameters. The software calculates the network parameters and noise data of the blocks from the topology of the filter and the LC values.
Series/Shunt RLC	Series and shunt RLC components for designing lumped element cascades, specified using RLC parameters. The software calculates the network parameters and noise data of the blocks from the topology of the components and the RLC values.
Mixers	RF mixers that contain local oscillators, specified using network parameters, noise data, and nonlinearity data, or a data file containing these parameters.
Transmission Lines	RF filters, specified using physical dimensions and electrical characteristics. The software calculates the network parameters and noise data of the blocks from the specified data.

Sublibrary	Description
Black Box	Passive RF components, specified using network parameters, or a data file containing these parameters. The software calculates the network parameters and noise data of the blocks from the specified data.
Input/Output Ports	Blocks for specifying simulation information that pertains to all blocks in a physical subsystem, such as center frequency and sample time. **Note** A physical subsystem is a collection of one or more physical blocks bracketed by an Input Port block and an Output Port block that bridge the physical and mathematical parts of the model.

For more information on defining components, see “Specifying or Importing Component Data” on page 2-7.

Mathematical Library

The Mathematical library contains mathematical representations of the amplifier, mixer, and filter blocks. Use a block from the Mathematical library to model an RF component in terms of mathematical equations that describe how the block operates on an input signal.

Mathematical blocks assume perfect impedance matching and a nominal impedance of 1 ohm. This means there is no loading and the power flow is unidirectional. As such, they are similar to standard Simulink blocks. In contrast, the physical blocks do not assume perfect matching—these blocks model the reflections that occur between blocks. Physical blocks model bidirectional power flow, and include loading effects. For these blocks, you can specify the source and load impedances using the Input Port and Output Port blocks.

The mathematical library is sh own in the following figure.

Product Workflow

When you analyze an RF system using RF Blockset™ software, your workflow might include the following tasks:

1. Create a Simulink® model that includes RF components.

   For more information, see “Modeling RF Components” on page 2-2.

2. Define component data by

   - Specifying network parameters, mathematical relationships, or physical properties
   - Importing data from an industry-standard Touchstone® file, a MathWorks™ AMP file, an Agilent® P2D or S2D file, or the MATLAB® workspace

     The product lets you access component data in Touchstone SnP, YnP, ZnP, HnP, and GnP formats. You can also import amplifier network parameters and power data from a MathWorks AMP file.

   For more information, see “Specifying or Importing Component Data” on page 2-7.

3. Where applicable, add the following information to the component definition:

   - Operating condition values (see “Specifying Operating Conditions” on page 2-24).
   - Nonlinearity data (see “Modeling Nonlinearity” on page 2-26).
   - Noise data (see “Modeling Noise” on page 2-29).

4. Validate the behavior of individual blocks by plotting component data.

   Note You can plot data for individual blocks from the RF Physical library that model physical components either before or after you run a simulation.

   For more information, see “Creating Plots” on page 3-2.

5 Run the simulation.

For more information on how the product performs time-domain simulation of an RF system, see “Simulating an RF Model” on page A-2.

6 Generate plots to gain insight into system behavior.

For more information, see “C reating Plots” on page 3-2.

The following plots and charts are available:

- Rectangular plots
- Polar plots
- Smith ® Charts
- Composite plots
- Budget plots

Example — Modeling an LC Bandpass Filter

Overview of LC Bandpass Filter Example

In this example, you model the signal attenuation caused by an RF filter by comparing the signals at the input and output of the filter.

The RF filter you use in this example is an LC bandpass filter with a bandwidth of 200 MHz, centered at 700 MHz. You use a three-tone input signal to stimulate a range of in-band and out-of-band frequencies of the filter. The input signal has the following tones:

- 700 MHz — Center of the filter
- 600 MHz — Lower edge of the filter passband
- 900 MHz — Outside the filter passband

You simulate the effects of the filter over a bandwidth of 500 MHz.

Selecting Blocks to Represent System Components

In this part of the example, you select the blocks to represent the input signal, the RF filter, and the signal displays.

You model the RF filter using a physical subsystem, which is a collection of one or more physical blocks bracketed by an Input Port block and an Output Port block. The RF filter subsystem consists of an LC Bandpass Pi block, and the Input Port and Output Port blocks. The function of the Input Port and

Output Port blocks is to bridge the phy sical part of the model, which uses bidirectional RF signals, and the rest of the model that uses unidirectional Simulink ® signals.

The following table lists the blocks that represent the system components and a description of the role of each block.

Block	Description
Sine Wave	Generates a three-channel signal.
Matrix Sum	Combines the three channel signal into a single three-tone source signal.
Input Port	Establishes parameters that are common to all blocks in the RF filter subsystem, including the source impedance of the subsystem that is used to convert Simulink signals to the RF Blockset™ physical modeling environment.
LC Bandpass Pi	Models the signal attenuati on caused by the RF filter which, in this example, is the LC Bandpass Pi filter.
Output Port	Establishes parameters that are common to all blocks in the RF filter subsystem. These parameters include the load impedance of the subsystem, which is used to convert RF signals to Simulink signals.
Spectrum Scope	Displays signals at the input to and output of the filter.

Building the Model

In this part of the example, you create a Simulink model, add blocks to the model, and connect the blocks.

1 Create a model.

   If you are new to Simulink, see the introductory example, “Creating a Simulink Model”, for information on how to create a model.

2 Add to the model the blocks shown in the following table. The Library column of the table specifies the h ierarchical path to each block.

Block	Library Path	Quantity
Sine Wave	Signal Processing Blockset > Signal Processing Sources	1
Matrix Sum	Signal Processing Blockset > Math Functions > Matrices and Linear Algebra > Matrix Operations	1
Spectrum Scope	Signal Processing Blockset > Signal Processing Sinks	2
Input Port	RF Blockset > Physical > Input/Output Ports	1
LC Bandpass Pi	RF Blockset > Physical > Ladder Filters	1
Output Port	RF Blockset > Physical > Input/Output Ports	1

3 Connect the blocks as shown in the following figure.

For more information on connecting physical and mathematical blocks, see “Connecting Model Blocks” on page 2-3.

Now you are ready to specify block parameters.

Specifying Model Parameters

In this part of the example , you specify the following parameters to represent the behavior of the system components:

- "Input Signal Parameters" on page 1-16
- "Filter Subsystem Parameters" on page 1-18
- "Signal Display Parameters" on page 1-21

Input Signal Parameters

You generate the three-tone source signal using two blocks. You use the Sine Wave block to generate a complex three -channel signal, where each channel corresponds to a different frequency. Then, you use the Matrix Sum block to combine the channels into a single three-tone source signal. Without this block, the signal in all subsequent blocks would have three independent channels.

The RF Blockset algorithm requires you to shift the frequencies of the input signal. The software simulates the filt er subsystem using a complex-baseband modeling technique, which automatically shifts the filter response so it is centered at zero. You must shift the frequencies of the signals outside the physical subsystem by the same amount.

For more information on complex-ba seband modeling, see "Creating a Complex Baseband-Equivalent Model" on page A-13.

Note All signals in the RF model must be complex to match the signals in the physical subsystem, so you create a complex input signal.

The center frequency of the LC ban dpass filter is 700 MHz, so you use a three-tone source signal with tones t hat are 700 MHz below the actual tones, at -100 MHz, 0 MHz, and 200 MHz, respectively.

1 Set the Sine Wave block parameters as follows:

- Amplitude = 1e-6
- Frequency (Hz) = [-100 0 200]*1e6
- Output complexity = Complex
- Sample time = 1/500e6
- Samples per frame = 128

2 Set the Matrix Sum block **Sum along** parameter to `Rows`.

Filter Subsystem Parameters

In this part of the example, you configure the blocks that model the RF filter subsystem—the Input Port, LC Bandpass Pi, and Output Port blocks.

1 Set the Input Port block parameters as follows:

- **Center frequency** = `700e6`
- **Sample time(s)** = `1/500e6`

- Clear the **Add noise** check box so the software will not include noise in the simulation. To learn how to model noise, see “Modeling Noise” on page 2-29.

Note You must enter the **Sample time (s)** because the Input Port block does not inherit a sample time from the input signal. The specified sample time must match the sample time of the input signal. The **Sample time (s)** of `1/500e6` second used in this example is equivalent to a bandwidth of 500 MHz.

---

2 Accept default parameters for inductance and capacitance in the LC Bandpass Pi block. These parameters create a filter with the desired bandwidth of 200 MHz, centered at 700 MHz.

3 Accept the default parameters for the Output Port block to use a load impedance of 50 ohms.

Signal Display Parameters

In this part of the example, you specify the parameters of the Spectrum Scope block to display the source signal, and the parameters of the Spectrum Scope1 block to display the filtered signal.

For each scope, you set the range of the x- and y-axes to make sure that the entire signal is visible.

By default, the scope displays appear stacked on top of each other on the screen when you run the simulation, so you can only see one of them. To ensure that both scopes are visible during the simulation, you specify a different position for each scope on the screen.

1 Set the Spectrum Scope block parameters as follows:
   - Axis Properties tab, Frequency range = [-Fs/2...Fs/2]
   - Axis Properties tab, Minimum Y-limit = -160
   - Axis Properties tab, Maximum Y-limit = -100
   - Display Properties tab, Scope position = get(0,'defaultfigureposition').*[.15 1 1 1]

2 Set the Spectrum Scope1 block parameters as follows:
   - Axis Properties tab, Frequency range = [-Fs/2...Fs/2]
   - Axis Properties tab, Minimum Y-limit = -160
   - Axis Properties tab, Maximum Y-limit = -105
   - Display Properties tab, Scope position = get(0,'defaultfigureposition').*[1.85 1 1 1]

Note If you do not specify the position of the scopes using the Display Properties tab, you can click and drag the displays to arrange them on the screen after the simulation starts.

Validating Filter Components and Running the Simulation

In this part of the example, you validate the behavior of the LC Bandpass Pi filter block by plotting its frequency response and then run the simulation.

Note When you plot information about a physical block, the plot displays the actual frequency response of the block at the selected passband (i.e., the response at the unshifted frequencies), and *not* the response at the shifted frequencies. Recall that this shift was introduced in “Input Signal Parameters” on page 1-16.

---

1 Double-click the LC Bandpass Pi block to open the block dialog box.

2 Select the **Visualization** tab and click **Plot** to plot the frequency response of the filter. This plots the magnitude of S_{21} as a function of frequency, which represents the gain of the filter.

Filter Gain

Note The physical blocks only model a band of frequencies around the center frequency of the physical subsystem. You must choose the sample time and center frequency such that all important frequency characteristics of your physical subsystem fall in this band of frequencies. The plot shows the frequency response of the filter for the portion of the RF spectrum that the physical blocks model. In this example, the physical blocks model a 500 MHz band that is centered at 700 MHz, defined by the Input Port block.

---

3 In the model window, select **Simulation > Start** to run the simulation.

Analyzing the Simulation Results

In this part of the example, you analyze the results of the simulation. This section contains the following topics:

- "Comparing the Input and Output Signals of the RF Filter" on page 1-24
- "Plotting Model Parameters of the Filter Subsystem" on page 1-27

Comparing the Input and Output Signals of the RF Filter

You can view the source signal and the filtered signal in the Spectrum Scope windows while the model is running. These windows appear automatically when you start the simulation.

The Spectrum Scope blocks display the signals at the shifted (baseband-equivalent) frequencies, and not at the selected passband frequencies. You can relabel the x-axes of the Spectrum Scope windows to display the passband signal by entering the **Center frequency** parameter value of `700e6` (from the Input Port block) for the **Display DC as** parameter in the **Axis Properties** pane of the Spectrum Scope blocks. For more information on complex-baseband modeling, see "Creating a Complex Baseband-Equivalent Model" on page A-13.

The Spectrum Scope blocks display power spectral density normalized to unit sampling frequency. To display power per channel, insert a Gain block with the **Gain** parameter set to `1/N` before each Spectrum Scope block. `N` is the number of channels. The Gain block is in the **Simulink > Commonly Used Blocks** library.

In this example, `N` is `128` (the value of the **Samples per frame** parameter of the Sine Wave block, `128`).

Note RF Blockset signals represent amplitudes, not voltages. This means that in the product, power is defined as:

$$\text{Power (in watts)} = \left[\text{Amplitude (in volts} / \text{sqrt(ohm))}\right]^2$$

The following plot shows the RF filter input signal you specified in the Sine Wave block.

Input to RF Filter

The next plot shows the filtered signal. Notice that the amplitude of the peaks is attenuated as a result of the RF filter.

Attenuated Output of RF Filter

Together, the Input Port and Output Port blocks introduce a 6-dB attenuation into the physical system at all frequencies. You can see this loss in the plots by comparing the input and output signals at the center frequency of the filter. The magnitude of the input signal at the center frequency approaches -101 dB in the Spectrum Scope window. The magnitude of the output signal at the center frequency approaches -107 dB in the Spectrum Scope1 window. However, as shown in the plot in the previous section, Filter Gain on page 1-23, the filter does not attenuate the signal at the center frequency. The 6 dB of loss is caused by the source and the load in the model. For more information on why this loss occurs, see the note in "Converting to and from Simulink® Signals" on page A-33.

Plotting Model Parameters of the Filter Subsystem

After you simulate an RF model, you can evaluate the behavior of the physical subsystem by plotting the network parameters of the Output Port block.

Note When you plot information about a physical subsystem, the plot displays the actual frequency response of the subsystem at the selected passband (i.e. the response at the unshifted frequencies), and *not* the response at the shifted frequencies.

To understand the frequency response of the filter, examine the S-parameters as a function of frequency for the RF filter subsystem on a composite plot.

1 Open the dialog box of the Output Port block by double-clicking the block.

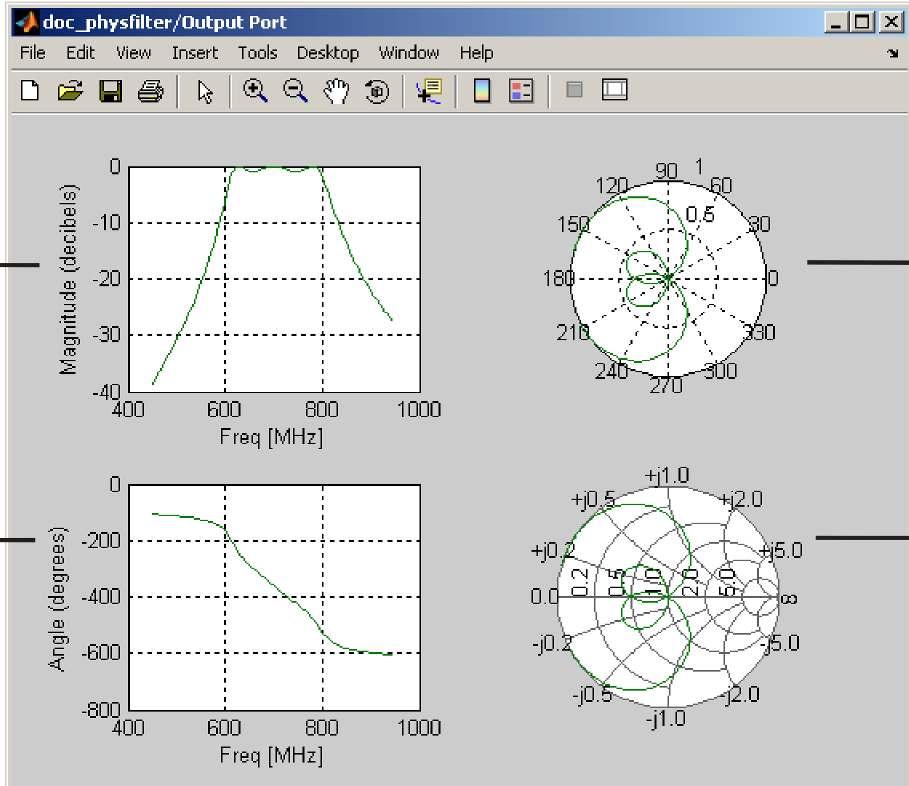
2 Select the **Visualization** tab, and click **Plot**.

The composite plot, shown in the following figure, contains four separate plots in one figure. For the Output Port block, the composite plot shows the following as a function of frequency (counterclockwise from the upper-left plot):

- An X-Y plane plot of the magnitude of the filter gain, S_{21}, in decibels.
- An X-Y plane plot of the phase of the filter gain, S_{21}, in degrees.
- A Z Smith chart showing the real and imaginary parts of the filter reflection coefficient, S_{11}.
- A Polar plane plot showing the magnitude and phase of the filter reflection coefficient, S_{11}.

Note In this example, the response of the filter subsystem is the same as the response of the filter block because the subsystem contains only a filter block.

! " # $ %


doc_physfilter/Output Port
File Edit View Insert Tools Desktop Window Help
Magnitude (decibels)
0
-10
-20
-30
-40
400 600 800 1000
Freq [MHz]
Angle (degrees)
0
-200
-400
-600
-800
400 600 800 1000
Freq [MHz]
90 1 120 60 0.5 150 30 180 0 210 330 240 300 270
+j1.0 +j0.5 +j2.0 +j0.2 +j5.0 0.0 0.2 0.5 1.0 2.0 5.0 ∞ -j0.2 -j5.0 -j0.5 -j2.0 -j1.0

$ %

Modeling an RF System

Modeling RF Components

In this section...

Adding RF Blocks to a Model

You can include blocks from the RF Blockset™ Physical and Mathematical libraries in a Simulink® model. For more information on the libraries and the available RF blocks, see "RF Blockset™ Libraries" on page 1-6.

This section contains the following topics:

Input Signal Requirements

Most RF Blockset blocks only support complex single-channel signals. The signals can be either sample-based or frame-based. The following blocks have this requirement:

- All Physical blocks
- Mathematical Amplifier and Mixer blocks

You can model the effect of these components on a multichannel signal as follows:

1. Use a Simulink Demux block to split the multichannel signal into single-channel signals.
2. Create duplicate RF models, with one model for each channel, and pass each single-channel signal into a separate model.
3. Use a Simulink Mux block multiplex the signals at the output of the RF models.

How to Add RF Blocks to a Model

To add RF blocks to a Simulink model:

1. Type `rflib` at the MATLAB® prompt to open the RF Blockset library.
2. Navigate to the desired library or sublibrary.
3. Drag instances of RF Blockset blocks into the model window using the mouse.

Note You can also access RF Blockset blocks and other Simulink blocks from the Simulink Library Browser window. Open this window by typing `simulink` at the MATLAB prompt. Add blocks to the model by dragging them from this window and dropping them into the model window.

Connecting Model Blocks

You follow the same procedure for connecting RF Blockset blocks as for connecting Simulink blocks: you click a port and drag the mouse to draw a line to another port on a different block. For more information on connecting blocks, see “Connecting Blocks in the Model Window” in the Simulink documentation.

You can only connect blocks that use the same type of signal. RF Blockset Physical blocks use different types of signals than Mathematical blocks, and are represented graphically by a different port style. Therefore, you can freely connect pairs of Mathematical modeling blocks. You can also freely connect pairs of Physical modeling blocks. However, you cannot directly connect Physical blocks to Mathematical blocks. Instead, you must use the Input Port and Output Port blocks to bridge them.

For more information on the RF Blockset libraries, including how to open the libraries and a description of the available blocks, see “RF Blockset™ Libraries” on page 1-6.

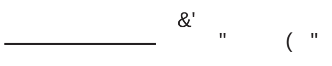
This section contains the following topics:

- “Connecting Mathematical Blocks” on page 2-4

- “Connecting Physical Blocks” on page 2-4
- “Bridging Physical and Mathematical Blocks” on page 2-5

Connecting Mathematical Blocks

The RF Blockset Mathematical blocks use the same input and output ports as standard Simulink blocks. These ports show the direction of the signal at the port, as shown in the following diagram.

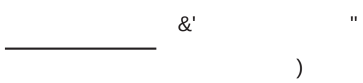


Similar to standard Simulink blocks, you draw lines between the ports of the Mathematical modeling blocks, called signal lines, to represent signals that are inputs to and outputs from the mathematical functions represented by the blocks. Therefore, you can connect Simulink, Signal Processing Blockset™, and RF Blockset mathematical blocks by drawing signal lines between their ports.

You can connect a port to multiple ports by branching the signal line, or you can leave a port unconnected. For more information on connecting blocks, see “Connecting Blocks in the Model Window” in the Simulink documentation.

Connecting Physical Blocks

The RF Blockset Physical blocks have specialized connector ports. These ports only represent physical connections; they do not imply signal direction.



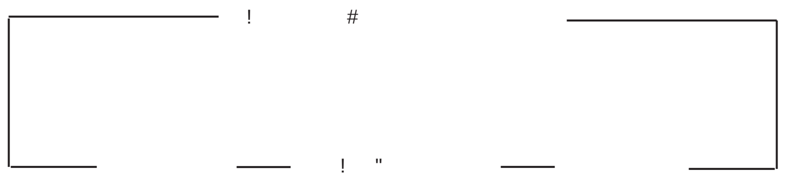
The lines you draw between the physical modeling blocks, called connection lines, represent physical connections among the block components. Connection lines appear as solid black when connected and as dashed red lines when either end is unconnected.

You can draw connection lines only between the connector ports of physical modeling blocks. You cannot branch these connection lines. You cannot leave connector ports unconnected.

Bridging Physical and Mathematical Blocks

The blockset provides the Input Port and Output Port blocks to connect the physical and mathematical parts of the model. These blocks convert mathematical signals to and from the physical modeling environment.

The Input Port and Output Port blocks have one of each kind of connector port: a standard Simulink style input port and a physical modeling port. These ports are shown in the following figure:



The Input Port and Output Port blocks must bound a physical subsystem to connect it to the mathematical part of a model.

For example, a simple RF model of a coaxial transmission line might resemble the following figure.

The Microstrip Transmission Line block uses an Input Port block to get its white noise input from a Random Source block, and an Output Port block to pass its output to a Spectrum Scope block. The Random Source and Spectrum Scope blocks are from Signal Processing Blockset library.

For information on how RF Blockset software converts mathematical signals to and from the physical modeling environment, see "Converting to and from Simulink® Signals" on page A-33.

Specifying or Importing Component Data

In this section...

Specifying Parameter Values

There are two ways to set block parameter values:

- Using the GUI — Enter information in the block dialog boxes, which open when you double-click a block in the Simulink® window.
- Using commands — Use the Simulink `set_param` and `get_param` commands to set and get parameter values of the blocks, respectively. For more information on these commands, see the `set_param` and `get_param` reference pages.

Supported File Types for Importing Data

The blockset also lets you import the following types of data files:

- Industry-standard file formats — Touchstone® S2P, Y2P, Z2P, and H2P formats specify the network parameters and noise information for measured and simulated data.

  For more information on Touchstone files, see http://www.vhdl.org/pub/ibis/connector/touchstone_spec11.pdf.
- Agilent® P2D file format — Specifies amplifier and mixer large-signal, power-dependent network parameters, noise data, and intermodulation tables for several operating conditions, such as temperature and bias values.

The P2D file format lets you import system-level verification models of amplifiers and mixers.

- Agilent S2D file format — Specifies amplifier and mixer network parameters with gain compression, power-dependent S_{21} parameters, noise data, and intermodulation tables for several operating conditions.

  The S2D file format lets you import system-level verification models of amplifiers and mixers.

- MathWorks™ amplifier (AMP) file format — Specifies amplifier network parameters, power data, noise data, and third-order intercept point

  For more information about `.amp` files, see "AMP File Format" in the RF Toolbox™ documentation.

- MATLAB® circuits — RF Toolbox circuit objects in the MATLAB workspace specify network parameters, noise data, and third-order intercept point information of circuits with different topologies.

  For more information about RF circuit objects, see "RF Circuit Objects" in the RF Toolbox documentation.

Importing Data Files into RF Blocks

The blockset lets you import industry-standard data files, Agilent P2D and S2D files, and MathWorks AMP files into specific blocks to simulate the behavior of measured components in the Simulink modeling environment.

This section contains the following topics:

- "Blocks Used to Import Data" on page 2-8
- "How to Import Data Files" on page 2-9

Blocks Used to Import Data

Three blocks in the Physical library accept data from a file. The following table lists the blocks and any corresponding data format that each supports.

Block	Description	Supported Format(s)
General Amplifier	Generic amplifier	Touchstone, AMP, P2D, S2D

Block	Description	Supported Format(s)
General Mixer	Generic mixer	Touchstone, AMP, P2D, S2D
General Passive Network	Generic passive component	Touchstone

How to Import Data Files

To import a data file:

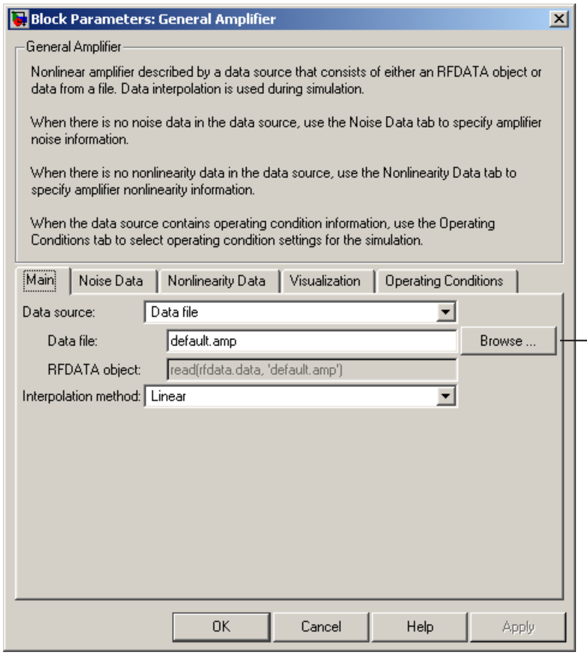
1. Choose the block that best represents your component from the list of blocks that accept file data shown in “Blocks Used to Import Data” on page 2-8.

2. Open the RF Blockset™ Physical library, and navigate to the sublibrary that contains the block.

3. Click and drag the block into your Simulink model.

4. In the block dialog box, enter the name of your data file for the **Data file** parameter. The file name must include the extension. If the file is not in your MATLAB path, specify the full path to the file or use the **Browse** button to find the file.

Note The **Data file** parameter is only enabled when the **Data source** parameter is set to `Data file`. This is the default setting and it means the block data comes from a file.



The following section shows an example of this procedure.

Example — Importing a Touchstone® Data File into an RF Model

In this example, you simulate the frequency response of a passive component using data from a Touchstone file, `passive.s2p`.

You use a model from one of the RF Blockset demos to perform the following tasks:

- “Importing Data into a General Passive Network Block” on page 2-11
- “Validating the Passive Component” on page 2-13
- “Running the Simulation and Analyzing the Results” on page 2-15

Importing Data into a General Passive Network Block

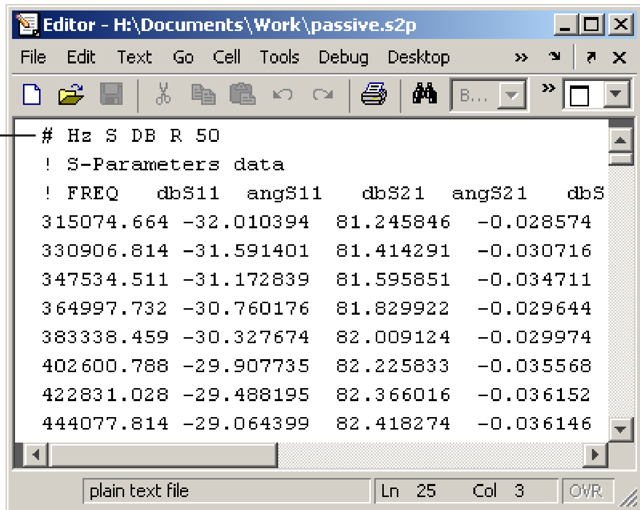
In this part of the example, you inspect the `passive.s2p` file and import data into the RF model using the General Passive Network block.

1 Type the following at the MATLAB prompt to open the `passive.s2p` file:

```
edit passive.s2p
```

The following figure shows a portion of the `.s2p` file.

* +



The option line

```
# Hz S DB R 50
```

specifies the following information about the contents of the data file:

- Hz — Frequency units.
- S — Network parameters are S-parameters.
- DB — Network parameters are specified as magnitude in dB and phase in degrees.
- R 50 — Reference impedance is 50 ohms.

For more information about the Touchstone specification, including the option line, see http://www.vhdl.org/pub/ibis/connector/touchstone_spec11.pdf.

2 At the MATLAB prompt, type

```
sparam_filter
```

This command opens the RF Blockset demo called “Touchstone Data File for 2-Port Passive Networks,” as shown in the following figure.

3 Double-click the General Passive Network block to display its parameters.

The Data source parameter is set to RFDATA object, so the RFDATA object parameter specifies the data file to import. The RFDATA object parameter is set to read(rfdata.data, 'passive.s2p'). This setting reads the data from the file passive.s2p into an rfdata.data object. The block uses this data with the other block parameters during simulation.

Note When the imported file contains data that is measured at frequencies other than the modeling frequencies, use the **Interpolation method** parameter to specify how the block determines the data values at the modeling frequencies. For more information, see "Determining the Modeling Frequencies" on page A-3 and "Mapping Network Parameters to Modeling Frequencies" on page A-5.

Validating the Passive Component

In this part of the example, you plot the network parameters of the General Passive Network block to validate the data you imported in "Importing Data into a General Passive Network Block" on page 2-11.

1 Open the General Passive Network block dialog box, select the **Visualization** tab, and click **Plot**.

This action creates a plot of S_{11} as a function of frequency.

S_{11} versus Frequency for the Imported Data

2 Open the General Passive Network block dialog box, and select the **Visualization** tab. Set the **Y Parameter1** parameter to `S21`, and click **Plot**.

The block adds the S_{21} data to the plot.

S_{11} and S_{21} versus Frequency for the Imported Data

Running the Simulation and Analyzing the Results

In this part of the example, you run the simulation and examine the frequency response of the passive component.

Start the simulation by selecting **Simulation > Start** in the model window. This action opens the Vector scope plot, which displays the amplitude of the transfer function of the system in decibels as a function of frequency. This is shown in the following figure.

Transfer Function of General Passive Network Subsystem

Note The transfer function differs from S_{21} because of the following factors:

- The 6 dB of loss introduced by the source and the load in the `sparam_filter` model

  This loss is described in the note in "Converting to and from Simulink® Signals" on page A-33.

- The numerical error introduced by the calculation of the transfer function

---

Importing Circuits from the MATLAB Workspace

You can only connect the RF Blockset Physical blocks in cascade. However, the blockset works with RF Toolbox software to let you include additional circuit topologies in an RF model. To model circuit topologies that contain other types of connections, you must define a circuit in the MATLAB workspace and import it into an RF model.

To import a circuit from the MATLAB workspace:

1. Define the circuit object in the MATLAB workspace using the RF Toolbox functions.

   For more information about RF circuit objects, see the RF Toolbox documentation for “RF Circuit Objects”.

2. Add a General Circuit Element block to your RF model from the Black Box Elements sublibrary of the Physical library. For information on how to open this library, see “Opening RF Blockset™ Libraries” on page 1-6.

3. Enter the circuit object name in the **RFCKT object** parameter in the General Circuit Element block dialog box.

This procedure is illustrated by example in the following section.

Example — Importing a Bandstop Filter into an RF Model

In this example, you simulate the frequency response of a filter that you model using circuit objects from the MATLAB workspace.

The filter in this example is the 50-ohm bandstop filter shown in the following figure.

Bandstop Filter Diagram

You represent the filter using four circuit objects that correspond to the four parts of the filter, `ckt1`, `ckt2`, `ckt3`, and `ckt4` in the diagram. You use an input signal with random, complex input values that have a Gaussian distribution to stimulate the filter. The scope block displays the output signal.

This example illustrates how to perform the following tasks:

- "Creating Circuit Objects in the MATLAB® Workspace" on page 2-18
- "Building the Model" on page 2-19
- "Specifying and Importing Component Data" on page 2-20
- "Running the Simulation and Plotting the Results" on page 2-22

Creating Circuit Objects in the MATLAB® Workspace

In this part of the example, you define MATLAB variables to represent the physical properties of the filter shown in the previous figure, Bandstop Filter Diagram on page 2-17, and use functions from RF Toolbox software to create RF circuit objects that model the filter components.

1 Type the following at the MATLAB prompt to define the filter's capacitance and inductance values in the MATLAB workspace:

```
C1 = 1.734e-12;
C2 = 4.394e-12;
C3 = 7.079e-12;
C4 = 7.532e-12;
C5 = 1.734e-12;
C6 = 4.394e-12;
L1 = 25.70e-9;
L2 = 3.760e-9;
L3 = 17.97e-9;
L4 = 3.775e-9;
L5 = 17.63e-9;
L6 = 25.70e-9;
```

2 Type the following at the MATLAB prompt to create RF circuit objects that model the components labeled ckt1 , ckt2 , ckt3 , and ckt4 in the circuit diagram:

```
ckt1 = ...
  rfckt.series('Ckts',{rfckt.shuntrlc('C',C1),...
  rfckt.shuntrlc('L',L1,'C',C2)});
ckt2 = ...
  rfckt.parallel('Ckts',{rfckt.seriesrlc('L',L2),...
  rfckt.seriesrlc('L',L3,'C',C3)});
ckt3 = ...
  rfckt.parallel('Ckts',{rfckt.seriesrlc('L',L4),...
  rfckt.seriesrlc('L',L5,'C',C4)});
ckt4 = ...
  rfckt.series('Ckts',{rfckt.shuntrlc('C',C5),...
  rfckt.shuntrlc('L',L6,'C',C6)});
```

For more information about the RF Toolbox objects used in this example, see the .series , parallel constructor , shuntrlc constructor , and seriesrlc constructor object reference pages in the RF Toolbox documentation.

Building the Model

In this portion of the example, you create a Simulink model. For more information about adding and connecting components, see “Modeling RF Components” on page 2-2.

1 Create a new model.

2 Add to the model the blocks shown in the following table. The Library column of the table specifies the hierarchical path to each block.

Block	Library	Quantity
Random Source	Signal Processing Blockset > Signal Processing Sources	1
Input Port	RF Blockset > Physical > Input/Output Ports	1

Block	Library	Quantity
General Circuit Element	RF Blockset > Physical > Black Box Elements	4
Output Port	RF Blockset > Physical > Input/Output Ports	1
Spectrum Scope	Signal Processing Blockset > Signal Processing Sinks	1

3 Connect the blocks as shown in the following figure.

Change the names of your General Circuit Element blocks to match those in the figure by double-clicking the text below the block and typing a new name.

Specifying and Importing Component Data

In this portion of the example, you specify block parameters. To open the parameter dialog box for each block, double-click the block.

1 Set the Random Source block parameters as follows:

- Source type = Gaussian
- Sample time = 1/(100e6)
- Samples per frame = 256
- Complexity = Complex

Selecting these settings creates an input signal with random, complex input values that have a Gaussian distribution.

2 Set the Input Port block parameters as follows:

- Finite impulse response filter length = 256
- Center frequency (Hz) = 400e6
- Sample time = 1/(100e6)
- Source impedance = 50
- Clear the Add noise check box.

Selecting these settings defines the physical characteristics and modeling bandwidth of the filter.

3 Set the parameters of the General Circuit Element blocks as follows:

- In the General Circuit Element1 block dialog box, set the RFCKT object parameter to ckt1.
- In the General Circuit Element2 block dialog box, set the RFCKT object parameter to ckt2.
- In the General Circuit Element3 block dialog box, set the RFCKT object parameter to ckt3.
- In the General Circuit Element4 block dialog box, set the RFCKT object parameter to ckt4.

Selecting these settings imports the circuit objects that model the filter components into the model.

4 Type 50 in the Load impedance field of the Output Port block to represent an impedance of 50 ohms.

5 Set the Spectrum Scope block parameters as follows:

- In the **Scope Properties** tab, set the **Number of spectral averages** parameter to `100`.

  This parameter establishes the number of spectra that the scope averages to produce the displayed signal. You use a value of 100 because the input signal is random and you want to display the average filter response over a large number of input values.

- In the **Axis Properties** tab, set the **Frequency range** parameter to `[-Fs/2 ... Fs/2]`, the **Minimum Y-limit** parameter to `-35`, and the **Maximum Y-limit** parameter to `3.6`.

  These values set the range of *x*- and *y*-values on the display such that the entire signal is visible when you run the simulation.

Running the Simulation and Plotting the Results

In this part of the example, you run the simulation and examine the frequency response of the filter.

Select **Simulation > Start** in the model window to start the simulation.

The Spectrum Scope window appears automatically and displays the following plot, which shows the frequency response of the filter.

Frequency Response of Bandstop Filter

The Spectrum Scope block displays the frequency response at the shifted (baseband-equivalent) frequencies, and *not* at the selected passband frequencies. You can relabel the *x*-axis of the Spectrum Scope window to display the passband signal by entering the **Center frequency** parameter value of `400e6` (from the Input Port block) for the **Display DC as** parameter in the **Axis Properties** pane of the Spectrum Scope block. For more information on complex-baseband modeling, see "Creating a Complex Baseband-Equivalent Model" on page A-13.

References

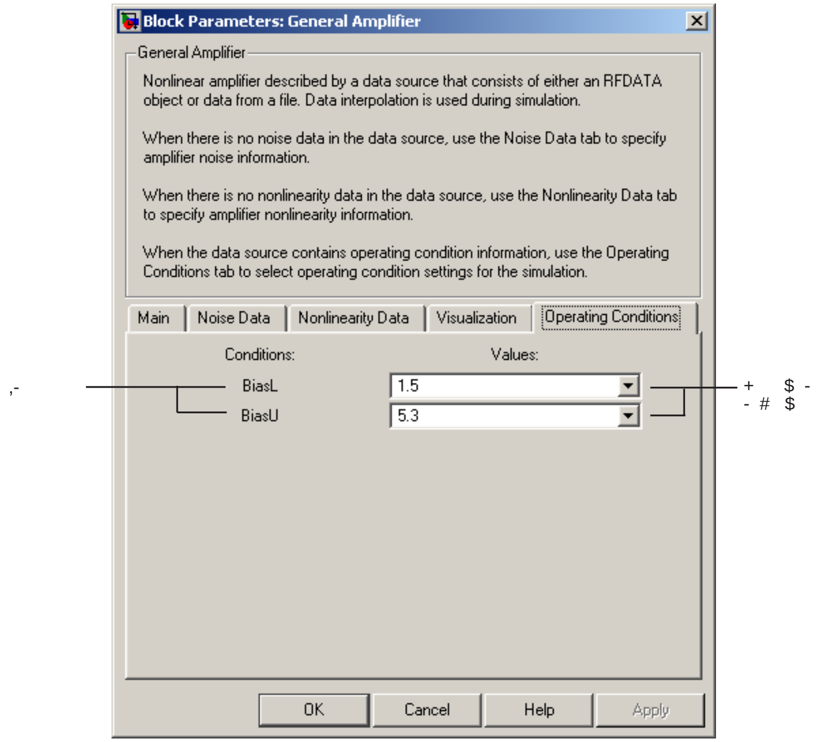
Geffe, P.R., "Novel designs for elliptic bandstop filters," *RF Design*, February 1999.

Specifying Operating Conditions

Agilent® P2D and S2D files contain simulation results at one or more operating conditions. Operating conditions define the independent parameter settings that are used when creating the file data. The specified conditions differ from file to file.

When you import component data from a `.p2d` or `.s2d` file into a General Amplifier or General Mixer block, the block contains parameter values for several operating conditions. The available conditions depend on the data in the file. By default, the blockset defines the object behavior using the property values that correspond to the operating conditions that appear first in the file. To use other property values, you must select a different operating condition in the block dialog box.

If the block contains data at multiple operating conditions, the Operating Conditions tab contains two columns. The Conditions column shows the available conditions, and the Values column contains a drop-down list of the available values for the corresponding condition.



Example Block Dialog Box Showing Operating Conditions

To specify the operating condition values for a simulation:

1 Double-click the block to open the block dialog box.

2 Select the **Operating Conditions** tab.

3 In the **Conditions** column, find the condition to specify. Select the corresponding pull-down list in the **Values** column, and choose the desired operating condition value.

Repeat the preceding step as needed to specify the desired operating condition values.

Modeling Nonlinearity

In this section...

Amplifier and Mixer Nonlinearity Specifications

You define nonlinearity for the physical amplifier and mixer blocks through one of the following specifications:

- Power data, consisting of output power as a function of input power, imported into the block.
- Third-order intercept data, with or without one or more power parameters, in the block dialog box. The power parameters are gain compression power (defined as the ratio of output power to input power at small input power) and output saturation power.

The following table summarizes the nonlinearity specification options for each type of physical amplifier and mixer block.

Block	Nonlinearity Specification
General Amplifier	Power data (using a P2D, S2D, or AMP data file) OR Third-order intercept data, with or without one or more power parameters, in the block dialog box.
S-Parameters Amplifier Y-Parameters Amplifier Z-Parameters Amplifier	Third-order intercept data, with or without one or more power parameters, in the block dialog box.

Block	Nonlinearity Specification
General Mixer	Power data (using a P2D, S2D, or AMP data file) OR Third-order intercept data, with or without one or more power parameters, in the block dialog box.
S-Parameters Mixer Y-Parameters Mixer Z-Parameters Mixer	Third-order intercept data, with or without one or more power parameters, in the block dialog box.

Adding Nonlinearity to Your System

To simulate the nonlinearity of an amplifier or mixer, you specify or import nonlinearity data into the block.

The method you use to add nonlinearity data to a block depends on whether you are specifying the data manually or importing the data into a block.

The following table provides instructions for adding nonlinearity data.

Nonlinearity Specification	Instructions
IP3	In the **Nonlinearity Data** tab of the block dialog box, set the **IP3 type** parameter to `IIP3` or `OIP3`, and enter the input third-order intercept value in the **IP3 (dBm)** parameter.
IP3 with power parameters	Specify the input third-order intercept value using the procedure described in the preceding row of the table. Then, enter the gain compression power in the **1 dB gain compression power (dBm)** parameter and, optionally, the saturation power in the **Output saturation power (dBm)** parameter. **Note** If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification.
Power data (from a file)	Import file data that includes power information into the **Data file** or **RFCKT object** parameter of the General Amplifier or General Mixer block.

Note If you import file data with no power information into a General Amplifier or General Mixer block, the **Nonlinearity Data** tab lets you add nonlinearity data manually in the block dialog box.

For information on how the blockset simulates nonlinearity data of an amplifier or mixer, see the block reference page.

Modeling Noise

Amplifier and Mixer Noise Specifications

You only need to specify noise information for the physical amplifier and mixer blocks that generate noise other than resistor noise. For the other blocks, the blockset calculates the noise automatically based on the resistor values.

You define noise for the physical amplifier and mixer blocks through one of the following specifications:

- Spot noise data in the data source.
- Spot noise data in the block dialog box.
- Spot noise data (.noise) object in the block dialog box.
- Frequency-independent noise figure, noise factor, or noise temperature value in the block dialog box.
- Frequency-dependent noise figure data (rfdata.nf) object in the block dialog box.

The following table summarizes the noise specification options for each type of physical amplifier and mixer block.

Block	Noise Specification
General Amplifier	Spot noise data (using a Touchstone®, P2D, S2D, or AMP data file) OR Spot noise data, noise figure value, noise factor value, noise temperature value, rfdata.noise, or rfdata.nf object in the block dialog box
S-Parameters Amplifier Y-Parameters Amplifier Z-Parameters Amplifier	Spot noise data, noise figure value, noise factor value, noise temperature value, rfdata.noise, or rfdata.nf object in the block dialog box
General Mixer	Spot noise data (using a Touchstone, P2D, S2D, or AMP data file) OR Spot noise data, noise figure value, noise factor value, noise temperature value, rfdata.noise, or rfdata.nf object in the block dialog box
S-Parameters Mixer Y-Parameters Mixer Z-Parameters Mixer	Spot noise data, noise figure value, noise factor value, noise temperature value, rfdata.noise, or rfdata.nf object in the block dialog box

Adding Noise to Your System

To simulate the noise of a physical subsystem, you perform the following tasks:

- “Specifying or Importing Noise Data” on page 2-30
- “Adding Noise to the Simulation” on page 2-33

Specifying or Importing Noise Data

The method you use to add noise data to a block depends on whether you are specifying noise data manually or importing spot-noise data.

The following table provides instructions for adding noise data.

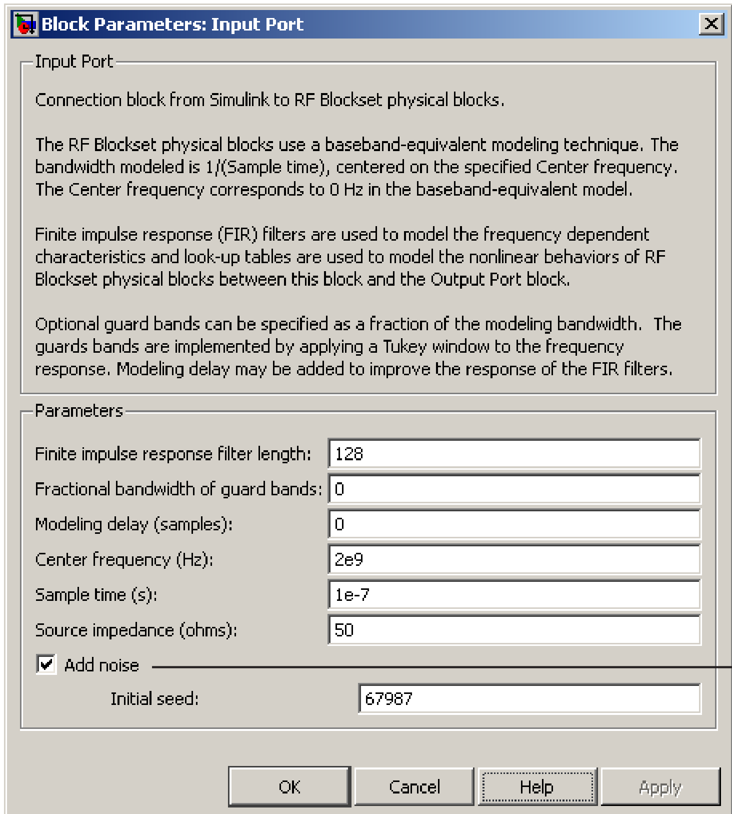
Noise Specification	Instructions
Frequency-independent noise figure	In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Noise figure`, and enter the noise figure value in the **Noise figure (dB)** parameter.
Frequency-dependent noise figure	In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Noise figure`, and enter the name of the `rfdata.nf` object in the **Noise figure (dB)** parameter.
Noise factor	In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Noise factor`, and enter the noise factor value in the **Noise factor** parameter.
Noise temperature	In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Noise temperature`, and enter the noise temperature value in the **Noise temperature (K)** parameter.
Spot noise data (in a block dialog box)	In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Spot noise data`. Enter the spot noise information in the **Minimum noise figure (dB)**, **Optimal reflection coefficient**, and **Equivalent normalized noise resistance** parameters.

Noise Specification	Instructions
Spot noise data (from a data object)	In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Noise figure` and enter the name of the `rfdata.noise` object in the **Noise figure (dB)** parameter.
Spot noise data (from a file)	Import file data that includes noise information into the **Data file** or **RFCKT object** parameter of the General Amplifier or General Mixer block.

Note If you import file data with no noise information into a General Amplifier or General Mixer block, the **Noise Data** tab lets you add noise data manually in the block dialog box.

Adding Noise to the Simulation

To include noise in the simulation, you must select the **Add noise** check box on the Input Port block dialog box. This check box is selected by default.



For information on how the blockset simulates noise, see “Modeling Noise in an RF System” on page A-7.

Plotting Noise

RF Blockset™ software models communications systems. The noise in these systems has a very small amplitude, typically from 1e-6 to 1e-12 Watts. In contrast, the default signal power of a Communications Blockset™ modulator block is 1 Watt at a nominal 1 ohm. Therefore, the signal-to-noise ratio in an RF system simulation is large, making it difficult to view the noise RF Blockset blocks add to your signal.

To display the noise on a plot, you might need to attenuate the signal amplitude to a value within a couple orders of magnitude of the noise.

For example, suppose you have the following model that contains a multitone test signal source.

When you simulate this model, Simulink® brings up several windows showing the input and output for the physical subsystem. The Input - Frequency Domain window shown in the following figure displays the input signal in the frequency domain.

Input Signal Spectrum

The Real Part of Input - Time Domain window displays the real part of the complex-valued input signal in the time domain.

Real Part of Input Signal

In the model, the physical subsystem adds noise to the input signal. The Output - Frequency Domain window shows the noisy output signal in the frequency domain.

Output Signal Spectrum

The amplitude of the signal is large compared to the amplitude of the noise, so the noise is not visible in the Real Part of Output - Time Domain window that shows the real part of the time-domain output signal. Therefore, you must attenuate the amplitude of the input signal to display the noise of the time-domain output signal.

Real Part of Output Signal

Attenuate the amplitude of the input signal by setting the **Gain** parameter to `1e-3`. This is equivalent to attenuating the input signal by 60 dB. When you run the model again, the two signal peaks are not as pronounced in the Output - Frequency Domain window.

Output Signal Spectrum for Attenuated Input

You can now view the noise RF Blockset blocks add to your signal in the Real Part of Output - Time Domain window.

Real Part of Output Signal Showing Noise

3

Plotting Model Data

Creating Plots

In this section...

Available Data for Plotting

RF Blockset™ software lets you validate the behavior of individual RF components and physical subsystems in your model by plotting the following data:

- Large- and small-signal S-parameters
- Noise figure
- Output third-order intercept point
- Power data
- Phase noise
- Voltage standing-wave ratio

Note When you plot information about a physical block, the blockset plots the actual frequency response of the block, as specified in the block dialog box. The blockset does not plot the frequency response of the complex-baseband model that it uses to simulate the block, in which the frequency response is centered at zero. For more information on how the blockset simulates physical blocks, see Appendix A, "RF Blockset™ Algorithms".

Using Plots to Validate Individual Blocks and Subsystems

You can plot model data for an individual physical block or for a physical subsystem. A subsystem is a collection of one or more physical blocks bracketed by an Input Port block and an Output Port block. To understand the behavior of specific subsystems, plot the data of the corresponding Output Port block after you run a simulation.

To validate the behavior of individual RF components in the model, plot the data of the corresponding physical blocks. You can plot data for individual blocks from each of these components either before or after you run a simulation.

You create a plot by selecting options in the block dialog box, as shown in “Example — Creating and Modifying Subsystem Plots” on page 3-31. To learn about the available plots, see “Types of Plots” on page 3-3. For more information about creating plots, see “How to Create a Plot” on page 3-14.

Types of Plots

RF Blockset software provides a variety of plots for analyzing the behavior of RF components and subsystems. The following table summarizes the available plots and charts and describes each one.

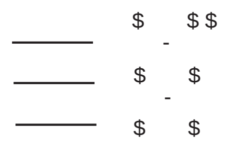
Plot Type	Plot Contents
X-Y Plane (Rectangular) Plot	Parameters as a function of frequency, input power, or operating condition, such as • S-parameters • Noise figure (NF) • Voltage standing-wave ratio (VSWR) • Output third-order intercept point (OIP3)

Plot Type	Plot Contents
Link Budget Plot (3-D)	Parameters as a function of frequency for each component in a physical subsystem, where the curve for a given component represents the cumulative contribution of each RF component up to and including the parameter value of that component. For more information, see “Link Budget” on page 3-10.
Polar Plane Plot	Magnitude and phase of S-parameters as a function of frequency or operating condition.
Smith® Chart	Real and imaginary parts of S-parameters as a function of frequency or operating condition, used for analyzing the reflections caused by impedance mismatch.
Composite Plot	Multiple plots and charts in one figure.

To learn how to create these plots, see “How to Create a Plot” on page 3-14.

Plot Formats

When you create a plot from a block dialog box, you must specify the format of the data for both the x- and y-axes.



These plot options define how RF Blockset software displays the data on the plot.

The available formats vary with the data you select to plot. The data you can plot depends on the plot type you select. The plot formats determine whether the blockset converts the data to a new set of units, or performs a calculation on the data. For example, setting the format to Real tells the blockset to compute and plot the real part of the parameter.

The following topics describe the available parameters and formats for each plot type:

- “Composite Data” on page 3-6
- “X-Y Plane” on page 3-8
- “Link Budget” on page 3-10
- “Polar Plane Plots and Smith Charts” on page 3-12

Composite Data

The composite data plot automatically generates four separate plots in one figure window, showing the frequency dependence of several parameters. The following figure shows an example of such a plot.

Example — Composite Data Plot

Note For composite data plots, you do not need to specify the parameters or the formats—they are set automatically.

The combination of plots differs based on the type of block and the specified block data. The following table describes the contents of the composite data plot for each specification. The Plot Contents column lists the types of plots as they appear on the composite plot, counterclockwise and starting in the upper-left corner. The blockset plots all data as a function of frequency.

Block	Specified Data	Plot Contents
General Amplifier or General Mixer	Network parameters OR Network parameters and noise	• X-Y plot, magnitude of S_{12} and S_{21} in decibels • X-Y plot, phase of S_{12} and S_{21} in degrees • Z Smith Chart, real and imaginary parts of S_{11} and S_{22} • Polar plot, magnitude and phase of S_{11} and S_{22}
	Network parameters and power OR Network parameters, noise, and power	• X-Y plot, magnitude of S_{12} and S_{21} in decibels • X-Y plot, output power (P_{out}) in dBm (decibels referenced to one milliwatt) • Z Smith Chart, real and imaginary parts of S_{11} and S_{22} • Polar plot, magnitude and phase of S_{11} and S_{22}
Other Physical block	Network parameters OR Network parameters and noise (S-, Y-, and Z-Parameters Amplifiers and Mixers only) **Note** Only the General Amplifier and General Mixer blocks accept power data.	• X-Y plot, magnitude of S_{12} and S_{21} in decibels • X-Y plot, phase of S_{12} and S_{21} in degrees • Z Smith Chart, real and imaginary parts of S_{11} and S_{22} • Polar plot, magnitude and phase of S_{11} and S_{22}

X-Y Plane

You can plot any parameters that are relevant to your block on an X-Y plane plot. For this type of plot, you specify data for both the x- and y-axes. If you specify two Y parameters, and you specify different formats for the two Y parameters, the blockset plots the second Y parameter on the right y-axis.

The following table summarizes the available Y parameters and formats. The parameters and formats are the same for both the left and right y-axes.

Note LS11, LS12, LS21, and LS22 are large-signal S-parameters. You can plot these parameters as a function of input power or as a function of frequency.

Y Parameter	Y Format
S11, S12, S21, S22 LS11, LS12, LS21, LS22 (General Amplifier and General Mixer blocks with multiple operating conditions only)	Magnitude (decibels) Magnitude (linear) Angle (degrees) Angle (radians) Real Imaginary
NF	Magnitude (decibels) Magnitude (linear)
OIP3 (Amplifier and Mixer blocks only)	dBm dBW W mW
VSWRIn, VSWROut	Magnitude (decibels) Magnitude (linear)
Pout (General Amplifier and General Mixer blocks with power data only)	dBm dBW W mW
Phase (General Amplifier and General Mixer blocks with power data only)	Angle (degrees) Angle (radians)

Y Parameter	Y Format
AM/AM (General Amplifier and General Mixer blocks with power data only)	Magnitude (decibels) Magnitude (linear)
AM/PM (General Amplifier and General Mixer blocks with power data only)	Angle (degrees) Angle (radians)
PhaseNoise (Mixer blocks only)	dBc/Hz
FMIN (Amplifier and Mixer blocks with spot noise data only)	Magnitude (decibels) Magnitude (linear)
GAMMAOPT (Amplifier and Mixer blocks with spot noise data only)	Magnitude (decibels) Magnitude (linear) Angle (degrees) Angle (radians) Real Imaginary
RN (Amplifier and Mixer blocks with spot noise data only)	None This format tells the blockset to plot the noise resistance as it is specified to the block.

The available X parameters depend on the Y parameters you select. The following table summarizes the available X parameters for each of the Y parameters in the preceding table.

Y Parameter	X Parameter
Pout, Phase, LS11, LS12, LS21, LS22	Pin Freq
S11, S12, S21, S22, NF, OIP3, VSWRIn, VSWROut, GAMMAIn, GAMMAOut, FMIN, GAMMAOPT, RN	Freq
AM/AM, AM/PM	AM

The following table shows the X formats that are available for the X parameters listed in the preceding table.

X Parameter	X Format
Pin	dBm dBW W mW
Freq	THz GHz MHz KHz Hz Auto (xformat is chosen to provide the best scaling for the given xparameter values.)
AM	Magnitude (dB) Magnitude (linear)

When you import block data from a .p2d or .s2d file, you can also plot Y parameters as a function of any operating condition from the file that has numeric values, such as bias. You can specify an operating condition as the X parameter only when validating individual blocks, and the format is always None. This format tells the blockset to plot the operating condition values as they are specified in the file.

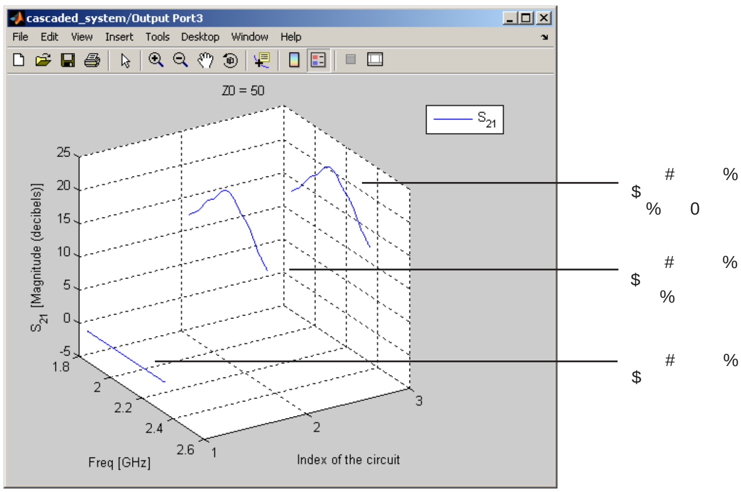
Link Budget

You use the Link budget plot to understand the individual contribution of each block to a plotted Y parameter value in a physical subsystem with multiple components between the Input Port and the Output Port blocks.

The link budget plot is a three-dimensional plot that shows one or more curves of parameter values as a function of frequency, ordered by the subsystem circuit index.

The following figure shows how the circuit index is assigned to a component in a physical subsystem based on its seq uential position in the subsystem.

A curve on the link budget plot for each circuit index represents the contributions to the parameter value of the RF components up to that index. The following figure shows an example of a link budget plot.



Example — Link Budget Plot

The following table summarizes the Y parameters and formats that are available for a link budget plot.

Y Parameter	Y Format
S11, S12, S21, S22	Magnitude (decibels) Magnitude (linear) Angle (degrees) Real Imaginary
VSWRIn, VSWROut	Magnitude (decibels) Magnitude (linear)
OIP3	dBm dBW W mW
NF	Magnitude (decibels) Magnitude (linear)

If you specify two Y parameters, the bloc kset puts both parameters in a single plot. The Y parameters must have the same formats.

For a link budget plot, the X parameter is always Freq. The format of the X parameter specifies the units of the x-axis.

Polar Plane Plots and Smith Charts

You can use RF Blockset software to generate Polar plots and Smith Charts. When you select these plot types, you do not need to specify the format of any Y parameters—the formats are set automatically. If you specify two Y parameters, the blockset puts bo th parameters in a single plot.

The following table describes the Polar plot and Smith Chart options, as well as the available Y parameters.

Plot Type	Y Parameter
Polar plane	S11, S12, S21, S22 LS11, LS12, LS21, LS22 (General Amplifier and General Mixer blocks with data from a P2D file only)
Z Smith chart	S11, S22 LS11, LS22 (General Amplifier and General Mixer blocks with data from a P2D file only)
Y Smith chart	S11, S22 LS11, LS22 (General Amplifier and General Mixer blocks with data from a P2D file only)
ZY Smith chart	S11, S22 LS11, LS22 (General Amplifier and General Mixer blocks with data from a P2D file only)

By default, the X parameter is Freq. The format of the X parameter specifies the units of the x-axis. When you import block data from a .p2d or .s2d file, you can also plot Y parameters as a function of any operating condition from the file that has numeric values, such as bias. You can specify an operating condition as the X parameter only when validating individual blocks, and the format is always None.

How to Create a Plot

1 Double-click the block to open the block dialog box, and select the Visualization tab. The following figure shows the contents of the tab.

2 Select the Source of frequency data .

$$ 1#
#
(

This value is the source of the frequ ency values at which to plot block data. The following table summarize s the available types of sources for the various types of blocks.

Source of frequency data	Description	Blocks
User-specified	Vector of frequencies that you enter. When you select User-specified in the Source of frequency data list, the Frequency range (Hz) field is displayed. Enter a vector specifying the range of frequencies you want to plot. For example, to plot block data from 0.3 MHz to 5 GHz by 0.1 MHz, enter [0.3e6:0.1e6:5e9]. **Note** When you select PhaseNoise in the Parameter list and User-specified in the Source of frequency data list, the Frequency range (Hz) field is disabled. You use the Phase noise frequency offset (Hz) block parameter to specify the frequency values at which to plot block data.	All physical blocks

Source of frequency data	Description	Blocks
Derived from Input Port parameters (Available after running a simulation or clicking the Update Diagram button)	Modeling frequencies derived from the Input Port block parameters. For information on how the blockset computes the modeling frequencies, see “Determining the Modeling Frequencies” on page A-3.	All physical blocks
Same as the s-parameters frequency	Frequency values specified in the Frequency block parameter.	S-Parameters Passive Network, S-Parameters Amplifier, S-Parameters Mixer
Same as the y-parameters frequency	Frequency values specified in the Frequency block parameter.	Y-Parameters Passive Network, Y-Parameters Amplifier, Y-Parameters Mixer
Same as the z-parameters frequency	Frequency values specified in the Frequency block parameter.	Z-Parameters Passive Network, Z-Parameters Amplifier, Z-Parameters Mixer
Extracted from data source	Frequency values imported into the Data file or RFDATA object block parameter.	General Passive Network, General Amplifier, and General Mixer

3 Enter the Reference impedance .

2
$

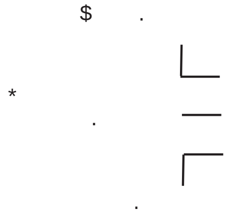
This value is the reference impedance to use when plotting small-signal parameters.

4 Select the Plot type .

This value is the type of plot. For a description of the options, see “Types of Plots” on page 3-3.

5 Select the following parameters:

- Y Parameter1 — The first parameter for the Y-axis.
- Y Parameter2 — The second parameter for the Y-axis (optional).
- X Parameter — The parameter for the X-axis.



These parameters specify the data to be plotted. The available choices vary with the type of plot. For a description of the options for a particular plot type, see the topic on that plot type in “Plot Formats” on page 3-4.

6 If you select a large-signal parameter for one or more y-axis parameters, select the Source of power data .

Note Large-signal parameters are available only for General Amplifier or General Mixer blocks that contain power data.

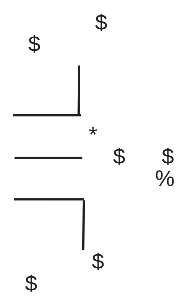
#
$ # (
- # (

This value is the source of the input power values at which to plot block data. The following table summarizes the available types of sources for the General Amplifier and General Mixer blocks.

Source of frequency data	Description
Extracted from data source	Input power values imported into the Data file or RFDATA object block parameter.
User-specified	Vector of power values that you enter. When you select User-specified in the Source of power data list, the Input power data (dBm) field is displayed. Enter a vector specifying the range of power values you want to plot. For example, to plot block data from 1 dBm to 10 dBm by 2 dBm, enter [1:2:10] .

7 Select the following formats:

- Y Format1 — The format for the first Y parameter.
- Y Format2 — The format for the second Y parameter (optional).
- X Format — The format for the X parameter.



These are the X and Y formats for plotting the selected parameter. The available choices vary based on the selected parameter. For a description of the options for a particular plot type, see the topic on that plot type in “Plot Formats” on page 3-4.

8 Select the **X Scale** and **Y Scale**.

These are the scales on which to plot the data. The available choices are `Linear` and `Log`.

9 Click **Plot**.

Note By default, the blockset does not add a legend to some plots. To display the plot legend, type `legend show` at the MATLAB® prompt.

Example — Plotting Component Data on a Z Smith® Chart

In this example, you simulate the frequency response of an amplifier using data from the `default.s2d` S2D file.

Using a model from one of the RF Block set demos, you import the data file into a General Amplifier block and validate the amplifier by plotting the S-parameters of the block on a Z Smith Chart.

1 Type sparam_amp at the MATLAB prompt to open the RF Blockset demo called “AMP Data File for Amplifier”.

2 Double-click the General Amplifier block to display its parameters.

As shown in the preceding figure, the **Data source** parameter is set to `Data file` and the **Data file** parameter is set to `default.s2d`. These values tell the blockset to import data from the file `default.s2d`. The block uses this data, along with the other block parameters, in simulation.

3 Select the **Visualization** tab and set the General Amplifier block parameters as follows:

- In the **Plot type** list, select `Z Smith chart`.
- In the **Y Parameter1** list, select `S22`.

4 Click **Plot**.

This action creates a Z Smith Chart of the S_{22} parameters using the frequency data from the `default.s2d` file.

General Amplifier Frequency Response

Note To display data tips for a plotted line, select **Tools > Data Cursor**. Click the data cursor on the plotted line to see the frequency and the parameter value at that point. See "Data Cursor — Displaying Data Values Interactively" in the MATLAB documentation for more information.

Updating Plots

When you run a simulation, the blockset continues to display any open plots, but does not update the plots to reflect new simulation results. You must update the subsystem plots after the simulation to display the behavior of the revised subsystem.

When you make changes to the parameters of blocks that represent individual RF components, you need to update any open plots, because the blockset does not automatically redraw the plots.

To update an existing plot:

1. Double-click the block to open the block dialog box, and select the **Visualization** tab.

   Example Block Dialog Box Showing Visualization Tab

2. Click **Plot**.

Modifying Plots

You can modify an existing plot by changing the plot options. The outcome depends on the parameter you change.

The following table summarizes the results of changing the plot options.

Block Parameter	Plot Change
Source of frequency data OR Frequency data (Hz)	Redraws plot using the new frequency data.
Source of power data OR Input power data (dBm)	Redraws plot using the new power data.
Plot type	Draws plot in a new figure using the new plot type. **Note** If the current plot options are valid for the new plot type, they retain their values. Otherwise, they revert to their default values.
Y Parameter1 OR Y Parameter2	If the new parameter has the same independent variable and format as the one on the plot, the blockset adds the new parameter to the existing plot. Otherwise, it redraws the plot for the new parameter and independent variable.
Y Format1 OR Y Format2	Redraws plot using the new format.

Block Parameter	Plot Change
X Parameter	Redraws plot using the new independent variable.
X Format	Redraws plot using the new format.
X Scale	Redraws plot using the new scale.
Y Scale	Redraws plot using the new scale.

To modify a plot:

1 Double-click the block to open the block dialog box, and select the **Visualization** tab.

Example Block Dialog Box Showing Plot Parameters

2 Change the plot options.

3 Click **Plot**.

Example — Creating and Modifying Subsystem Plots

In this section...

Plotting the Network Parameters of a Subsystem

In this part of the example, you open and run an RF Blockset™ demo that uses file data to specify an amplifier in a physical subsystem. Then, you plot the network parameters of the physical subsystem, which consists of the General Amplifier, the Input Port, and the Output Port blocks.

1 Type `sparam_amp` at the MATLAB® prompt to open the RF Blockset demo called “AMP Data File for Amplifier”.

2 In the model window, select **Simulation > Start** to run the simulation.

3 Double-click the Output Port block to open the block dialog box.

4 Select the **Visualization** tab and set the Output Port block parameters as follows:

- In the **Source of frequency data** list, select `Derived from Input Port parameters`.
- In the **Plot type** list, select `X-Y plane`.
- In the **Y Parameter1** list, select `S21`.

5 Click **Plot**.

This action plots the magnitude of S_{21} (in decibels) as a function of frequency on an X-Y plot.

S_{21} versus Frequency for a Physical Subsystem

Adding Data to an Existing Plot

In this part of the example, you add data to the plot you created in “Plotting the Network Parameters of a Subsystem” on page 3-31.

1 Double-click the Output Port block to open the block dialog box.

2 Select the **Visualization** tab and change the value of **Y Parameter1** to S22.

3 Click **Plot** .

This action adds S_{22} to the plot.

S_{21} and S_{22} versus Frequency for a Physical Subsystem

Changing Data on an Existing Plot

In this part of the example, you change the data on the plot you created in the previous steps of the example by modifying the **Plot type**.

1 Double-click the Output Port block to open the block dialog box.

2 Select the Visualization tab and change the value of Plot type to Polar plane , as shown in the following figure.

As the figure shows, the value of Y Parameter1 remains as S_{22}, the last parameter selected for the previous plot.

3 Click Plot .

This action creates a Polar plane plot of S_{22} as a function of frequency.

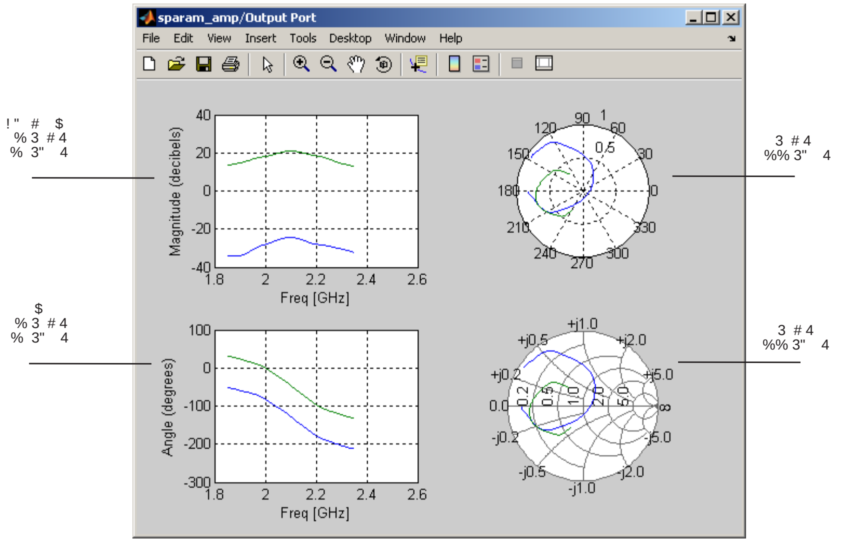
S_{22} versus Frequency for a Physical Subsystem

4 In the Output Port block dialog box, change the **Plot type** to `Composite data` to generate four plots in one figure. The parameters for the plots

are defined by the block, so the **Y Parameter1**, **Y Parameter2**, and **X Parameter** fields becomes invisible.

5 Click **Plot**.

This action creates a composite plot.



Composite Plot for a Physical Subsystem

Block Reference

Mathematical

Amplifier	Complex baseband model of amplifier with noise
Bandpass RF Filter	Standard bandpass RF filters in baseband-equivalent complex form
Bandstop RF Filter	Standard bandstop RF filters in baseband-equivalent complex form
Highpass RF Filter	Standard highpass RF filters in baseband-equivalent complex form
Lowpass RF Filter	Standard lowpass RF filters in baseband-equivalent complex form
Mixer	Complex baseband model of mixer and local oscillator with phase noise

Physical

Ladder Filters

LC Bandpass Pi	Model LC bandpass pi network
LC Bandpass Tee	Model LC bandpass tee network
LC Bandstop Pi	Model LC bandstop pi network
LC Bandstop Tee	Model LC bandstop tee network
LC Highpass Pi	Model LC highpass pi network
LC Highpass Tee	Model LC highpass tee network
LC Lowpass Pi	Model LC lowpass pi network
LC Lowpass Tee	Model LC lowpass tee network

Series/Shunt RLC

Series C	Model series capacitor
Series L	Model series inductor
Series R	Model series resistor
Series RLC	Model series RLC network
Shunt C	Model shunt capacitor
Shunt L	Model shunt inductor

Shunt R	Model shunt resistor
Shunt RLC	Model shunt RLC network

Transmission Lines

Coaxial Transmission Line	Model coaxial transmission line
Coplanar Waveguide Transmission Line	Model coplanar waveguide transmission line
Microstrip Transmission Line	Model microstrip transmission line
Parallel-Plate Transmission Line	Model parallel-plate transmission line
RLCG Transmission Line	Model RLCG transmission line
Transmission Line	Model general transmission line
Two-Wire Transmission Line	Model two-wire transmission line

Black Box Elements

General Circuit Element	Model two-port network described by rfckt object
General Passive Network	Model two-port passive network described by rfdata object
S-Parameters Passive Network	Model passive network using S-parameters
Y-Parameters Passive Network	Model passive network using Y-parameters
Z-Parameters Passive Network	Model passive network using Z-parameters

Amplifiers

General Amplifier	Model nonlinear amplifier described by rfdata object or file data
S-Parameters Amplifier	Model nonlinear amplifier using S-parameters
Y-Parameters Amplifier	Model nonlinear amplifier using Y-parameters
Z-Parameters Amplifier	Model nonlinear amplifier using Z-parameters

Mixers

General Mixer	Model mixer and local oscillator described by rfdata object
S-Parameters Mixer	Model mixer and local oscillator using S-parameters
Y-Parameters Mixer	Model mixer and local oscillator using Y-parameters
Z-Parameters Mixer	Model mixer and local oscillator using Z-parameters

Input/Output Ports

Connection Port	Connection port for RF subsystem
Input Port	Connection block from Simulink® environment to RF physical blocks
Output Port	Connection block from RF physical blocks to Simulink environment

5

Blocks — Alphabetical List

Purpose

Complex baseband model of amplifier with noise

Library

Mathematical

Description

The Amplifier block generates a complex baseband model of an amplifier with thermal noise. It provides six methods for modeling nonlinearity and three ways to specify noise.

Note This block assumes a nominal impedance of 1 ohm.

Modeling Nonlinearity

Use the **Method** parameter in the block dialog box to specify the method for modeling amplifier nonlinearity. The options for the **Method** parameter are

- Linear
- Cubic polynomial
- Hyperbolic tangent
- Saleh model
- Ghorbani model
- Rapp model

The linear method is implemented by a Gain block. The other nonlinear methods are implemented by subsystems underneath the block's mask. Each subsystem has the same basic structure, as shown in the following figure.

Application of Nonlinearity

All five subsystems for the nonlinear **Method** options apply a memoryless nonlinearity to the complex baseband input signal. Each one

1. Multiplies the signal by a gain factor.
2. Splits the complex signal into its magnitude and angle components.
3. Applies an AM/AM conversion to the magnitude of the signal, according to the selected nonlinearity method, to produce the magnitude of the output signal.
4. Applies an AM/PM conversion to the phase of the signal, according to the selected nonlinearity method, and adds the result to the angle of the signal to produce the angle of the output signal.
5. Combines the new magnitude and angle components into a complex signal and multiplies the result by a gain factor, which is controlled by the **Linear gain** parameter.

AM/AM and AM/PM Conversions

The subsystems for the nonlinear methods implement the AM/AM and AM/PM conversions differently, according to the nonlinearity method you specify. To see exactly how the Amplifier block implements the conversions for a specific method, you can view the AM/AM and AM/PM subsystems that implement these conversions as follows:

1. Right-click the Amplifier block.

2 Select Look under mask in the pop-up menu. This displays the block's configuration underneath the mask. The block contains five subsystems corresponding to the five nonlinearity methods.

3 Double-click the subsystem for the method in which you are interested. A subsystem displays similar to the one shown in the preceding figure.

4 Double-click one of the subsystems labeled AM/AM or AM/PM to view how the block implements the conversions.

The following figure shows, for the Saleh method, plots of

- Output voltage against input voltage for the AM/AM conversion
- Output phase against input voltage for the AM/PM conversion

Model Parameters and Characteristics of Nonlinearity Modeling Methods

The following sections discuss how the parameters specific to the following nonlinear amplifier models affect the AM/AM and AM/PM characteristics of the Amplifier block:

- Cubic Polynomial Model on page 5
- Hyperbolic Tangent Model on page 6
- Saleh Model on page 7
- Ghorbani Model on page 8
- Rapp Model on page 9

Note The Amplifier block also enables you to model a linear amplifier.

Cubic Polynomial Model

When you select `Cubic polynomial` for the nonlinearity modeling **Method** parameter, the Amplifier block models the AM/AM nonlinearity by:

1. Using the third-order input intercept point **IIP3 (dBm)** parameter to compute the factor, f, that scales the input signal before the Amplifier block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(\text{Watts})}} = \sqrt{\frac{3}{10^{(IIP3(\text{dBm}) - 30)/10}}}$$

2. Computing the scaled input signal by multiplying the amplifier input signal by f.

3. Limiting the scaled input signal to a maximum value of 1.

4 Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

The Amplifier block uses the AM/PM conversion (degrees per dB) parameter, which specifies the linear phase change, to add the AM/PM nonlinearity within the power limits specified by the Lower input power limit for AM/PM conversion (dBm) parameter and the Upper input power limit for AM/PM conversion (dBm) parameter. Outside those limits, the phase change is constant at the values corresponding to the lower and upper input power limits, which are zero and

$$(\text{AM/PM conversion})\,(\text{upper input power limit} - \text{lower input power limit}),$$

respectively.

The Linear gain (dB) parameter scales the output signal.

Hyperbolic Tangent Model

When you select Hyperbolic tangent for the nonlinearity modeling Method parameter, the Amplifier block computes and adds the AM/AM nonlinearity by:

1 Using the third-order input intercept point IIP3 (dBm) parameter to compute the factor, f, that scales the input signal before the Amplifier block applies the nonlinearity:

$$f = \sqrt{\frac{3}{\mathrm{IIP3}(\mathrm{Watts})}} = \sqrt{\frac{3}{10^{(\mathrm{IIP3(dBm)} - 30)/10}}}$$

2 Computing the scaled input signal by multiplying the amplifier input signal by f.

3 Limiting the scaled input signal to a maximum value of 1.

4 Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

The Amplifier block uses the AM/PM conversion (degrees per dB) parameter, which specifies the linear phase change, to add the AM/PM nonlinearity within the power limits specified by the Lower input power limit for AM/PM conversion (dBm) parameter and the Upper input power limit for AM/PM conversion (dBm) parameter. Outside those limits, the phase change is constant at the values corresponding to the lower and upper input power limits, which are zero and

(AM/PM conversion) (upper input power limit – lower input power limit),

respectively.

The Linear gain (dB) parameter scales the output signal.

Saleh Model

When you select Saleh model for the nonlinearity modeling Method parameter, the Input scaling (dB) parameter scales the input signal before the nonlinearity is applied. The block multiplies the input signal by the parameter value, converted from decibels to linear units. If you set the parameter to be the inverse of the input signal amplitude, the scaled signal has amplitude normalized to 1.

The AM/AM parameters, alpha and beta, are used to compute the amplitude gain for an input signal using the following function

where u is the magnitude of the scaled signal.

The AM/PM parameters, alpha and beta, are used to compute the phase change for an input signal using the following function

where u is the magnitude of the input signal. Note that the AM/AM and AM/PM parameters, although similarly named alpha and beta, are distinct.

The Output scaling (dB) parameter scales the output signal similarly.

Ghorbani Model

When you select Ghorbani model for the nonlinearity modeling Method parameter, the Input scaling (dB) parameter scales the input signal before the nonlinearity is applied. The block multiplies the input signal by the parameter value, converted from decibels to linear units. If you set the parameter to be the inverse of the input signal amplitude, the scaled signal has amplitude normalized to 1.

The AM/AM parameters, $[x_1\ x_2\ x_3\ x_4]$, are used to compute the amplitude gain for an input signal using the following function

where u is the magnitude of the scaled signal.

The AM/PM parameters, $[y_1\ y_2\ y_3\ y_4]$, are used to compute the phase change for an input signal using the following function

where u is the magnitude of the scaled signal.

The Output scaling (dB) parameter scales the output signal similarly.

Rapp Model

When you select Rapp model for the nonlinearity modeling Method parameter, the Smoothness factor and Output saturation level parameters are used to compute the amplitude gain for an input signal by the following function

where u is the magnitude of the scaled signal, S is the Smoothness factor and O_{sat} is the Output saturation level .

The Rapp model does not apply a phase change to the input signal.

The Output saturation level parameter limits the output signal level. The Smoothness factor parameter controls the transition for the amplitude gain as the input amplitude approaches saturation. The smaller the smoothness factor, the smoother the curve.

Thermal Noise Simulation

You can specify the amount of thermal noise in three ways, according to the Specification method parameter you select.

- Noise temperature — Specifies the noise in kelvin.
- Noise factor — Specifies the noise by the following equation:

$$\text{Noise factor} = 1 + \frac{\text{Noise temperature}}{290}$$

- Noise figure — Specifies the noise in decibels relative to a noise temperature of 290 kelvin. In terms of noise factor,

$$\text{Noise figure} = 10\log(\text{Noise factor})$$

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Dialog Box

The parameters displayed in the dialog box vary for different methods of modeling nonlinearity. Only some of these parameters are visible in the dialog box at any one time.

You can change tunable parameters while the model is running.

Method
Method used to model the nonlinearity. The choices are `Linear`, `Cubic polynomial`, `Hyperbolic tangent`, `Saleh model`, `Ghorbani model`, `Rapp model`. Tunable.

Linear gain (dB)
Scalar specifying the linear gain for the output function. This field becomes visible if you select `Linear`, `Cubic polynomial`, `Hyperbolic tangent`, or `Rapp model` as the **Method** parameter. Tunable.

IIP3 (dBm)
Input power intercept point as a scalar value. This field becomes visible if you select `Cubic polynomial` or `Hyperbolic tangent` as the **Method** parameter. For both of these methods, the nominal impedance is 1 ohm. Tunable.

AM/PM conversion (degrees per dB)
Scalar specifying the AM/PM conversion in degrees per decibel. This field becomes visible if you select `Cubic polynomial` or `Hyperbolic tangent` as the **Method** parameter. Tunable.

Lower input power limit for AM/PM conversion (dBm)
Scalar specifying the minimum input power for which AM/PM conversion scales linearly with input power value. Below this value, the phase shift resulting from AM/PM conversion is zero. This field becomes visible if you select `Cubic polynomial` or `Hyperbolic tangent` as the **Method** parameter. Tunable.

Upper input power limit for AM/PM conversion (dBm)
Scalar specifying the maximum input power for which AM/PM conversion scales linearly with input power value. Above this value, the phase shift resulting from AM/PM conversion is constant. The value of this maximum shift is given by:

(AM/PM conversion) (upper input power limit − lower input power limit), This field becomes visible if you select Cubic polynomial or Hyperbolic tangent as the Method parameter. Tunable.

Input scaling (dB)
: Number that scales the input signal level. This field becomes visible if you select Saleh model or Ghorbani model as the Method parameter. Tunable.

Output scaling (dB)
: Number that scales the output signal level. This field becomes visible if you select Saleh model or Ghorbani model as the Method parameter. Tunable.

AM/AM parameters [alpha beta]
: Vector specifying the AM/AM parameters. This field becomes visible if you select Saleh model as the Method parameter. Tunable.

AM/PM parameters [alpha beta]
: Vector specifying the AM/PM parameters. This field becomes visible if you select Saleh model as the Method parameter. Tunable.

AM/AM parameters [x1 x2 x3 x4]
: Vector specifying the AM/AM parameters. This field becomes visible if you select Ghorbani model as the Method parameter. Tunable.

AM/PM parameters [y1 y2 y3 y4]
: Vector specifying the AM/PM parameters. This field becomes visible if you select Ghorbani model as the Method parameter. Tunable.

Smoothness factor
: Scalar specifying the smoothness factor. This field becomes visible if you select Rapp model as the Method parameter. Tunable.

Output saturation level
: Scalar specifying the output saturation level. This field becomes visible if you select Rapp model as the Method parameter. Tunable.

Specification method
: The method by which you specify the amount of noise. The choices are Noise temperature, Noise figure, and Noise factor. Tunable.

Noise temperature (K)
: Scalar specifying the amount of noise. This field becomes visible if you select Noise temperature as the Specification method parameter. Tunable.

Noise figure (dB)
: Scalar specifying the amount of noise relative to a noise temperature of 290 kelvin. A Noise figure setting of 0 decibels indicates a noiseless system. Th is field becomes visible if you select Noise figure as the Specification method parameter. Tunable.

Noise factor
: Scalar specifying the amount of noise relative to a noise temperature of 290 kelvin. This field becomes visible if you select Noise factor as the Specification method parameter. Tunable.

Initial seed
: Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise.

Examples

You can see the effect of the Amplifier block in the demo Intermodulation: Mathematical Amplifier.

This demo uses a baseband-equivalent multitone signal as input to the Amplifier block. A Simulink® Slider Gain block enables you to vary the gain from 1 to 10. The following figure shows the input signal with gain set to the default 1.

The next figure shows the same signal after it passes through the Amplifier block, with the Method parameter set to Hyperbolic tangent. The demo uses the default Amplifier block IIP3 (dBm) value of 30. It uses no AM/PM conversion. The demo specifies thermal noise as Noise figure, for which it uses the default 3.01 dB.

References

[1] Ghorbani, A. and M. Sheikhan, “The Effect of Solid State Power Amplifiers (SSPAs) Nonlinearities on MPSK and M-QAM Signal

Transmission," Sixth Int'l Conference on Digital Processing of Signals in Comm., 1991, pp. 193-197.

[2] Rapp, C., "Effects of HPA-Nonlinearity on a 4-DPSK/OFDM-Signal for a Digital Sound Broadcasting System," in Proceedings of the Second European Conference on Satellite Communications, Liege, Belgium, Oct. 22-24, 1991, pp. 179-184.

[3] Saleh, A.A.M., "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," IEEE Trans. Communications, vol. COM-29, pp.1715-1720, November 1981.

See Also

Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Lowpass RF Filter, Mixer

Purpose

Standard bandpass RF filters in baseband-equivalent complex form

Library

Mathematical

Description

The Bandpass RF Filter block lets you design standard analog bandpass filters, implemented in baseband-equivalent complex form. The following table describes the available design methods.

Design Method	Description
Butterworth	The magnitude response of a Butterworth filter is maximally flat in the passband and monotonic overall.
Chebyshev I	The magnitude response of a Chebyshev I filter is equiripple in the passband and monotonic in the stopband.
Chebyshev II	The magnitude response of a Chebyshev II filter is monotonic in the passband and equiripple in the stopband.
Elliptic	The magnitude response of an elliptic filter is equiripple in both the passband and the stopband.
Bessel	The delay of a Bessel filter is maximally flat in the passband.

The block input must be a discrete-time complex signal.

Note This block assumes a nominal impedance of 1 ohm.

Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

Design Method	Filter Design Parameters
Butterworth	Order, lower passband edge frequency, upper passband edge frequency
Chebyshev I	Order, lower passband edge frequency, upper passband edge frequency, passband ripple
Chebyshev II	Order, lower stopband edge frequency, upper stopband edge frequency, stopband attenuation
Elliptic	Order, lower passband edge frequency, upper passband edge frequency, passband ripple, stopband attenuation
Bessel	Order, lower passband edge frequency, upper passband edge frequency

The Bandpass RF Filter block designs the filters using the Signal Processing Toolbox™ filter design functions `buttap`, `cheb1ap`, `cheb2ap`, `ellipap`, and `besselap`.

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Dialog Box

The parameters displayed in the dialog box vary for different design methods. Only some of these parameters are visible in the dialog box at any one time.

You can change tunable parameters while the model is running.

Design method
: Filter design method. The design method can be Butterworth, Chebyshev I, Chebyshev II, Elliptic, or Bessel. Tunable.

Filter order
: Order of the lowpass analog prototype filter that forms the basis for the bandpass filter design. The order of the final filter is twice this value.

Lower passband edge frequency (Hz)
Lower passband edge frequency for Butterworth, Chebyshev I, elliptic, and Bessel designs. Tunable.

Upper passband edge frequency (Hz)
Upper passband edge frequency for Butterworth, Chebyshev I, elliptic, and Bessel designs. Tunable.

Lower stopband edge frequency (Hz)
Lower stopband edge frequency for Chebyshev II designs. Tunable.

Upper stopband edge frequency (Hz)
Upper stopband edge frequency for Chebyshev II designs. Tunable.

Passband ripple in dB
Passband ripple for Chebyshev I and elliptic designs. Tunable.

Stopband attenuation in dB
Stopband attenuation for Chebyshev II and elliptic designs. Tunable.

Finite impulse response filter length
Desired length of the baseband-equivalent impulse response for the filter.

Center frequency (Hz)
Center of the modeling frequencies.

Sample time (s)
Time interval between consecutive samples of the input signal.

See Also

Amplifier, Bandstop RF Filter, Highpass RF Filter, Lowpass RF Filter, Mixer

buttap, cheb1ap, cheb2ap, ellipap, besselap (Signal Processing Toolbox)

Purpose

Standard bandstop RF filters in baseband-equivalent complex form

Library

Mathematical

Description

The Bandstop RF Filter block lets you design standard analog bandstop filters, implemented in baseband-equivalent complex form. The following table describes the available design methods.

Design Method	Description
Butterworth	The magnitude response of a Butterworth filter is maximally flat in the passband and monotonic overall.
Chebyshev I	The magnitude response of a Chebyshev I filter is equiripple in the passband and monotonic in the stopband.
Chebyshev II	The magnitude response of a Chebyshev II filter is monotonic in the passband and equiripple in the stopband.
Elliptic	The magnitude response of an elliptic filter is equiripple in both the passband and the stopband.
Bessel	The delay of a Bessel filter is maximally flat in the passband.

The block input must be a discrete-time complex signal.

Note This block assumes a nominal impedance of 1 ohm.

Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

Design Method	Filter Design Parameters
Butterworth	Order, lower passband edge frequency, upper passband edge frequency
Chebyshev I	Order, lower passband edge frequency, upper passband edge frequency, passband ripple
Chebyshev II	Order, lower stopband edge frequency, upper stopband edge frequency, stopband attenuation
Elliptic	Order, lower passband edge frequency, upper passband edge frequency, passband ripple, stop band attenuation
Bessel	Order, lower passband edge frequency, upper passband edge frequency

The Bandstop RF Filter block designs the filters using the Signal Processing Toolbox™ filter design functions `buttap`, `cheb1ap`, `cheb2ap`, `ellipap`, and `besselap`.

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Dialog Box

The parameters displayed in the dialog box vary for different design methods. Only some of these parameters are visible in the dialog box at any one time.

You can change tunable parameters while the model is running.

Design method
: Filter design method. The design method can be Butterworth, Chebyshev I, Chebyshev II, Elliptic, or Bessel. Tunable.

Filter order
: Order of the lowpass analog prototype filter that forms the basis for the bandstop filter design. The order of the final filter is twice this value.

Lower passband edge frequency (Hz)
: Lower passband edge frequency for Butterworth, Chebyshev I, elliptic, and Bessel designs. Tunable.

Upper passband edge frequency (Hz)
: Upper passband edge frequency for Butterworth, Chebyshev I, elliptic, and Bessel designs. Tunable.

Lower stopband edge frequency (Hz)
: Lower stopband edge frequency for Chebyshev II designs. Tunable.

Upper stopband edge frequency (Hz)
: Upper stopband edge frequency for Chebyshev II designs. Tunable.

Passband ripple in dB
: Passband ripple for Chebyshev I and elliptic designs. Tunable.

Stopband attenuation in dB
: Stopband attenuation for Chebyshev II and elliptic designs. Tunable.

Finite impulse response filter length
: Desired length of the baseband-equivalent impulse response for the filter.

Center frequency (Hz)
: Center of the modeling frequencies.

Sample time (s)
: Time interval between consecutive samples of the input signal.

See Also

Amplifier, Bandpass RF Filter, Highpass RF Filter, Lowpass RF Filter, Mixer

buttap, cheb1ap, cheb2ap, ellipap, besselap (Signal Processing Toolbox)

Purpose Model coaxial transmission line

Library Transmission Lines sublibrary of the Physical library

Description The Coaxial Transmission Line block models the coaxial transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A coplanar waveguide transmission line is shown in cross-section in the following figure. Its physical characteristics include the radius of the inner conductor *a* and the radius of the outer conductor *b*.

The block lets you model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model a coaxial transmission line as a stubless line, the Coaxial Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, *d*, and the complex propagation constant, *k*, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the resistance (R), inductance (L), conductance (G), and capacitance (C) per unit length (meters) as follows:

$$Z_0 = \sqrt{\frac{R + j2\ fL}{G + j2\ fC}}$$

where

In these equations, is the conductivity in the conductor and is the conductivity in the dielectric. is the permeability of the dielectric, is its permittivity, and skin depth is calculated as . f is a vector of modeling frequencies determined by the Output Port block.

The Coaxial Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Coaxial Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Outer radius (m)
Radius of the outer conductor of the coaxial transmission line.

Inner radius (m)
: Radius of the inner conductor of the coaxial transmission line.

Relative permeability constant
: Relative permeability of the dielectric expressed as the ratio of the permeability of the dielectric to permeability in free space.

Relative permittivity constant
: Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space.

Conductivity in conductor (S/m)
: Conductivity of the conductor in siemens per meter.

Conductivity in dielectric (S/m)
: Conductivity of the dielectric in siemens per meter.

Transmission line length (m)
: Physical length of the transmission line.

Stub mode
: Type of stub. Choices are Not a stub, Shunt, or Series.

Termination of stub
: Stub termination for stub modes Shunt and Series. Choices are Open or Short. This parameter becomes visible only when Stub mode is set to Shunt or Series.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, "Plotting Model Data".

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

See Also

Coplanar Waveguide Transmission Line, General Passive Network, Transmission Line, Microstrip Transmission Line, Parallel-Plate Transmission Line, Two-Wire Transmission Line

Purpose

Connection port for RF subsystem

Library

Input/Output Ports sublibrary of the Physical library

Description

The Connection Port block, placed inside a subsystem composed of RF Blockset™ blocks, creates an RF Blockset open round physical modeling connector port on the boundary of the subsystem. When it is connected to a connection line, the port becomes solid.

You connect individual RF Blockset blocks and subsystems made of RF Blockset blocks to one another with RF Blockset connection lines, instead of normal Simulink® signal lines. These blocks and subsystems are anchored at the open, round physical modeling connector ports. Subsystems constructed out of RF Blockset blocks automatically have such open round physical modeling connector ports. You can add additional connector ports by adding Connection Port blocks to your subsystem.

Dialog Box

Port number

This field labels the subsystem connector port created by this block. Multiple connector ports on the boundary of a single subsystem require different numbers as labels. The default value for the first port is `1`.

Port location on parent subsystem

Use this parameter to choose on which side of the parent subsystem boundary the Port is placed. The choices are Left or Right. The default choice is Left.

See Also

In Simulink, see Creating Subsystems.

Purpose

Model coplanar waveguide transmission line

Library

Transmission Lines sublibrary of the Physical library

Description

The Coplanar Waveguide Transmission Line block models the coplanar waveguide transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A coplanar waveguide transmission line is shown in cross-section in the following figure. Its physical characteristics include the conductor width (w), the conductor thickness (t), the slot width (s), the substrate height (d), and the relative permittivity constant ( ).

The block lets you model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model a coplanar waveguide transmission line as a stubless line, the Coplanar Waveguide Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, d, and the complex propagation constant, k, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the specified conductor strip width, slot width, substrate height, conductor strip thickness, relative permittivity constant, conductivity and dielectric loss tangent of the transmission line, as described in [1].

The Coplanar Waveguide Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Coplanar Waveguide Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Conductor width (m)
: Physical width of the conductor.

Slot width (m)
: Physical width of the slot.

Substrate height (m)
: Thickness of the dielectric on which the conductor resides.

Strip thickness (m)
: Physical thickness of the conductor.

Relative permittivity constant
: Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space.

Conductivity in conductor (S/m)
: Conductivity of the conductor in siemens per meter.

Loss tangent in dielectric
: Loss angle tangent of the dielectric.

Transmission line length (m)
: Physical length of the transmission line.

Stub mode
: Type of stub. Choices are Not a stub, Shunt, or Series.

Termination of stub
: Stub termination for stub modes Shunt and Series. Choices are Open or Short. This parameter becomes visible only when Stub mode is set to Shunt or Series.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, “Plotting Model Data”.

References

[1] Gupta, K. C., Ramesh Garg, Inder Bahl, and Prakash Bhartia, *Microstrip Lines and Slotlines*, 2nd Edition, Artech House, Inc., Norwood, MA, 1996.

See Also

Coaxial Transmission Line, General Passive Network, Transmission Line, Microstrip Transmission Line, Parallel-Plate Transmission Line, Two-Wire Transmission Line

Purpose Model nonlinear amplifier described by `rfdata` object or file data

Library Amplifiers sublibrary of the Physical library

Description The General Amplifier block models the nonlinear amplifier described by a data source that consists of either an RF Toolbox™ data (`rfdata.data`) object or data from a file.

Network Parameters

If network parameter data and their corresponding frequencies exist as S-parameters in the data source, the General Amplifier block interpolates the S-parameters to determine their values at the modeling frequencies. If the network parameters are Y- or Z-parameters, the block first converts them to S-parameters. See Appendix A, "RF Blockset™ Algorithms" for more details.

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from the power data. Power data determines both IP3 and 1 dB gain compression power.

If the data source contains no power data, then you can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the General Amplifier block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the General Amplifier block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1. Converting the specified value into IIP3 (if needed)
2. Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the General Amplifier block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3 Computing the scaled input signal by multiplying the amplifier input signal by f.

4 Limiting the scaled input signal to a maximum value of 1.

5 Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1 Converting the specified third-order intercept value into OIP3 (if needed)

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the data source):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the amplifier power gain, which is derived from the network parameters.

- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}} \left(\frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05} \right)$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the data source.
- Spot noise data in the block dialog box.
- Spot noise data (.noise) object in the block dialog box.

- Noise figure, noise factor, or noise temperature value in the block dialog box.
- Frequency-dependent noise figure data (rfdata.nf) object in the block dialog box.

The latter four options are only available if noise data does not exist in the data source.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Operating Conditions

Agilent® P2D and S2D files define block parameters for several operating conditions. Operating conditions are the independent parameter settings that are used when creating the file data. By default, RF Blockset™ software defines the block behavior using the parameter values that correspond to the operating conditions that appear first in the file. To use other property values, you must select a different operating condition in the General Amplifier block dialog box.

Data Consistency

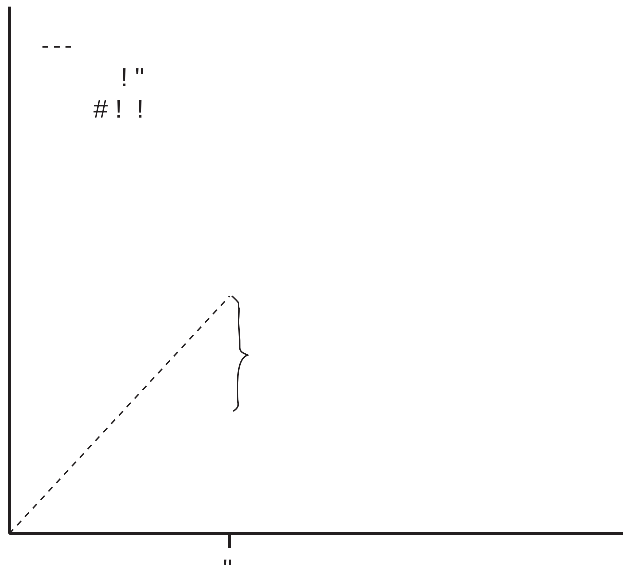
If the data source is a MathWorks™ AMP file or an Agilent S2D file that contains both network parameter data and power data, the blockset checks the data for consistency and reconciles it as necessary.

The blockset compares the small-signal amplifier gain defined by the network parameters, S_{21}, and by the power data, P_{out}-P_{in}. The discrepancy between the two is computed in dBm using the following equation:

$$\Delta P = S_{21}(f_P) - P_{out}(f_P) + P_{in}(f_P) \quad (\mathrm{dBm})$$

where f_P is the lowest frequency for which power data is specified.

If ΔP is more than 0.4 dB, a warning appears, and the blockset adds ΔP to the output power values at each specified input power value to resolve the discrepancy for simula tion. This discrepancy is shown in the following graph.



Dialog Box

Main Tab

Data source

Determines the source of the data that describes the amplifier behavior. The data source must contain network parameters and may also include noise data, nonlinearity data, or both. The value can be `Data file` or `RFDATA object`.

Data file

If **Data source** is set to `Data file`, use this field to specify the name of the file that contains the amplifier data. The file name must include the extension. If the file is not in your MATLAB®

path, specify the full path to the file or use the Browse button to find the file.

RFDATA object
: If Data source is set to `RFDATA object`, use this field to specify an RF Toolbox data (`rfdata.data`) object that describes an amplifier. You can specify the object as (1) the handle of a data object previously created using RF Toolbox software, (2) an RF Toolbox command such as `rfdata.data('Freq',1e9,'S_Parameters',[0 0; 0.5 0])`, which creates a data object, or (3) a MATLAB expression that generates such an object. See the RF Toolbox documentation for more information about data objects.

Interpolation method
: The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
`Linear` (default)	Linear interpolation
`Spline`	Cubic spline interpolation
`Cubic`	Piecewise cubic Hermite interpolation

Noise Data Tab

Noise type

Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature. This parameter is disabled if the data source contains noise data.

Noise figure (dB)

Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)

Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient

Optimal amplifier source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance

Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor

Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)

Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm)
: Scalar power value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if the IP3 value is unknown.

1 dB gain compression power (dBm)
Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1 dB compression point data. Use the default value, `Inf`, if the 1 dB compression point is unknown.

Output saturation power (dBm)
Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if 1 dB gain compression power (dBm) is specified. Use the default value, `Inf`, if saturation power is unknown.

Visualization Tab

For information about plotting the amplifier parameters, see Chapter 3, “Plotting Model Data”. Use rftool or the RF Toolbox plotting functions to plot other data.

Operating Conditions Tab

If the data source contains data at multiple operating conditions, the **Operating Conditions** tab contains two columns. The **Conditions** column shows the available conditions, and the **Values** column contains a drop-down list of the available values for the corresponding condition. Use the drop-down lists to specify the operating condition values to use in simulation.

Examples

Creating a General Amplifier Block from File Data

This example uses the default data source, which is the nonlinear amplifier in the file `default.s2d`. The file contains S-parameters for frequencies from 1.0 to 2.9 GHz at intervals of 0.01 GHz, power data at frequency 2.1 GHz, and active noise parameters. By default, the General Amplifier block uses linear interpolation to model the network described in the object.

1 On the **Main** tab, accept the default settings.

2 On the **Visualization** tab, set the parameters as follows:

- In the **Plot type** list, select `Z Smith chart`.
- In the **Y parameter1** list, select `S22`.

Click Plot. This action creates Z Smith chart of the S_{22} parameters using the frequencies taken from the data source.

For a demonstration of how to use an Agilent .s2d file in a Simulink® model, see Effect of Nonlinear Amplifier on QPSK Modulation.

See Also

Output Port, S-Parameters Amplifier, Y-Parameters Amplifier, Z-Parameters Amplifier

rfdata.data (RF Toolbox)

interp1 (MATLAB)

Purpose

Model two-port network described by `rfckt` object

Library

Black Box Elements sublibrary of the Physical library

Description

The General Circuit Element block models the two-port network described by an RF Toolbox™ circuit (`rfckt`) object.

The block uses the `rfckt/analyze` method to calculate the network parameters at the modeling frequencies.

Dialog Box

Main Tab

RFCKT object
: An RF Toolbox circuit (`rfckt`) object. You can specify the object as (1) the handle of a circuit object previously created using RF Toolbox software, (2) an RF Toolbox command such as `rfckt.txline`, `rfckt.coaxial`, or `rfckt.cascade` that creates a default circuit object of the specified type, or (3) a MATLAB® expression that generates such an object. See “RF Circuit Objects” in the RF Toolbox documentation for more information about circuit objects.

Visualization Tab

For infor mation about plotting circuit parameters, see Chapter 3, “Plottin g Model Data”.

Examples

Creating a General Circuit Element from an RF Toolbox Object

This example uses the `rfckt.txline` object, which describes a transmission line.

1 On the **Main** tab, set the **RFCKT object** parameter to `rfckt.txline`.

Click **Apply**. This action applies the specified settings.

2 Set the General Circuit Element block parameters on the **Visualization** tab as follows:

- In the **Y parameter1** list, select S12.

Click Plot . This action creates an X-Y Plane plot of the S_{12} parameters in the frequency range 1.9 to 2.2 GHz.

See Also

General Passive Network, S-Parameters Passive Network, Y-Parameters Passive Network, Z-Parameters Passive Network

interp1 (MATLAB)

Purpose Model mixer and local oscillator described by `rfdata` object

Library Mixer sublibrary of the Physical library

Description The General Mixer block models the mixer described by an RF Toolbox™ data (`rfdata.data`) object.

Network Parameters

The network parameter values all refer to the mixer input frequency. If network parameter data and their corresponding frequencies exist as S-parameters in the `rfdata.data` object, the General Mixer block interpolates the S-parameters to determine their values at the modeling frequencies. If the block contains network Y- or Z-parameters, the block first converts them to S-parameters. See “Mapping Network Parameters to Modeling Frequencies” on page A-5 for more details.

RF Blockset™ software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

$$\begin{bmatrix} b_1(f_{in}) \\ b_2(f_{out}) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1(f_{in}) \\ a_2(f_{out}) \end{bmatrix}$$

where

- f_{in} and f_{out} are the mixer input and output frequencies, respectively.
- a_1 and a_2 are the incident waves at the mixer input and output, respectively.

The interpolated S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency.

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the data source.

- Spot noise data in the block dialog box.
- Spot noise data (`.noise`) object in the block dialog box.
- Noise figure, noise factor, or noise temperature value in the block dialog box.
- Frequency-dependent noise figure data (`rfdata.nf`) object in the block dialog box.

The latter four options are only available if noise data does not exist in the data source.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Phase Noise

The General Mixer block applies phase noise to a complex baseband signal. The block first generates additive white Gaussian noise (AWGN) and filters the noise with a digital FIR filter. It then adds the resulting noise to the angle component of the input signal.

The blockset computes the digital filter by:

1. Interpolating the specified phase noise level to determine the phase noise values at the modeling frequencies.
2. Taking the IFFT of the resulting phase noise spectrum to get the coefficients of the FIR filter.

Note If you specify phase noise as a scalar value, the blockset assumes the phase noise is constant at all modeling frequencies and does not have a 1/*f* slope. This assumption differs from that made by the Mathematical Mixer block.

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from the power data. Power data determines both IP3 and 1 dB gain compression power.

If the data source contains no power data, then you can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the General Mixer block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the General Mixer block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1 Converting the specified value into IIP3 (if needed)

2 Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the General Mixer block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3 Computing the scaled input signal by multiplying the mixer input signal by f.

4 Limiting the scaled input signal to a maximum value of 1.

5 Applying an AM/AM conversion to the mixer gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1. Converting the specified third-order intercept value into OIP3 (if needed).

2. Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the data source):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

   where

   - GAIN is the mixer power gain, which is derived from the network parameters.
   - OIP3 is the output third-order intercept point.
   - PCOMP is the output power at the 1 dB compression point.

3. Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}} \frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05}$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Operating Conditions

Agilent® P2D and S2D files define block parameters for several operating conditions. Operating conditions are the independent parameter settings that are used when creating the file data. By default, the blockset defines the block behavior using the parameter values that correspond to the operating conditions that appear first in the file. To use other property values, you must select a different operating condition in the General Mixer block dialog box.

Dialog Box

Main Tab

Data source

Determines the source of the data that describes the mixer behavior. The data source must contain network parameters and may also include noise data, nonlinearity data, or both. The value can be `Data file` or `RFDATA object`.

Data file

If **Data source** is set to `Data file`, use this field to specify the name of the file that contains the mixer data. The file name must include the extension. If the file is not in your MATLAB® path, specify the full path to the file or use the **Browse** button to find the file.

Note If the data file contains an intermodulation table, the General Mixer block ignores the table. Use RF Toolbox software to ensure the cascade has no significant spurs in the frequency band of interest before running an RF Blockset simulation.

RFDATA object

If **Data source** is set to `RFDATA object`, use this field to specify an RF Toolbox data (`rfdata.data`) object that describes a mixer. You can specify the object as one of the following:

- The handle of a data object previously created using RF Toolbox software.
- An RF Toolbox command such as `rfdata.data('Freq',1e9,'S_Parameters',[0 0; 0.5 0])`, which creates a data object.
- A MATLAB expression that generates a data object.

For more information about data objects, see the `rfdata.data` reference page in the RF Toolbox documentation.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
`Linear` (default)	Linear interpolation
`Spline`	Cubic spline interpolation
`Cubic`	Piecewise cubic Hermite interpolation

Mixer Type

Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

LO frequency (Hz)

Local oscillator frequency. If you choose `Downconverter`, the blockset computes the mixer output frequency, f_{out}, from the mixer input frequency, f_{in}, and the local oscillator frequency, f_{lo}, as . If you choose `Upconverter`, .

Note The mixer output frequency must be positive. This means that if you choose a downconverting mixer, f_{in} must be greater than f_{lo}. Otherwise, an error appears.

Noise Data Tab

Phase noise frequency offset (Hz)
: Vector specifying the frequency offset.

Phase noise level (dBc/Hz)
: Vector specifying the phase noise level.

Noise type
: Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature. This parameter is disabled if the data source contains noise data.

Noise figure (dB)
: Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)
: Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient
: Optimal mixer source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance
: Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)
: Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm)
: Scalar power value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1 dB compression point data. Use the default value, Inf, if the 1 dB compression point is unknown.

Output saturation power (dBm)
: Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if 1 dB gain compression power (dBm) is specified. Use the default value, Inf, if saturation power is unknown.

Visualization Tab

For information about plotting the mixer parameters, see Chapter 3, “Plotting Model Data”. Use `rftool` or the RF Toolbox plotting functions to plot other data.

Operating Conditions Tab

If the data source contains data at multiple operating conditions, the **Operating Conditions** tab contains two columns. The **Conditions** column shows the available conditions, and the **Values** column contains a drop-down list of the available values for the corresponding condition. Use the drop-down lists to specify the operating condition values to use in simulation.

See Also

Output Port, S-Parameters Mixer, Y-Parameters Mixer, Z-Parameters Mixer

rfdata.data (RF Toolbox)

Purpose

Model two-port passive network described by rfdata object

Library

Black Box Elements sublibrary of the Physical library

Description

The General Passive Network block models the two-port passive network described by an RF Toolbox™ data (rfdata.data) object.

If network parameter data and their corresponding frequencies exist as S-parameters in the rfdata.data object, the General Passive Network block interpolates the S-parameters to determine their values at the modeling frequencies. If the block contains network Y- or Z-parameters, the block first converts them to S-parameters. See “Mapping Network Parameters to Modeling Frequencies” on page A-5 for more details.

Dialog Box

Main Tab

Data source
: Determines the source of the data that describes the amplifier behavior. The data source must contain network parameters and

may also include noise data, nonlinearity data, or both. The value can be Data file or RFDATA object.

RFDATA object

If Data source is set to RFDATA object, use this field to specify an RF Toolbox data (rfdata.data) object. You can specify the object as (1) the handle of a data object previously created using RF Toolbox software, (2) an RF Toolbox command such as rfdata.data('Freq',1e9,'S_Parameters',[0 0; 0.5 0]), which creates a data object, or (3) a MATLAB® expression that generates such an object. See the RF Toolbox documentation for more information about data objects.

Data file

If Data source is set to Data file, use this field to specify the name of the file that contains the amplifier data. The file name must include the extension. If the file is not in your MATLAB path, specify the full path to the file or use the Browse button to find the file.

Interpolation method

The method used to interpolate the network parameters. The following table lists the availa ble methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization Tab

For information about plotting circuit parameters, see Chapter 3, “Plotting Model Data”.

Examples

Creating a General Passive Network Block from File Data

This example creates a two-port passive network from the data in the file `passive.s2p`. The file contains S-parameters for frequencies from about 0.315 MHz to 6.0 GHz. The General Passive Network block uses linear interpolation to model the network described in the object.

1 On the **Main** tab, accept the default settings.

2 On the **Visualization** tab, set the parameters as follows:

- In the **Plot type** list, select `Z Smith chart`.

Click **Plot**. This action creates a Z Smith chart of the S_{11} parameters, using the frequencies taken from the **RFDATA object** parameter on the **Main** tab.

See Also

General Circuit Element, Output Port, S-Parameters Passive Network, Y-Parameters Passive Network, Z-Parameters Passive Network

rfdata.data (RF Toolbox)

interp1 (MATLAB)

Purpose

Standard highpass RF filters in baseband-equivalent complex form

Library

Mathematical

Description

The Highpass RF Filter block lets you design standard analog highpass filters, implemented in baseband-equivalent complex form. The following table describes the available design methods.

Design Method	Description
Butterworth	The magnitude response of a Butterworth filter is maximally flat in the passband and monotonic overall.
Chebyshev I	The magnitude response of a Chebyshev I filter is equiripple in the passband and monotonic in the stopband.
Chebyshev II	The magnitude response of a Chebyshev II filter is monotonic in the passband and equiripple in the stopband.
Elliptic	The magnitude response of an elliptic filter is equiripple in both the passband and the stopband.
Bessel	The delay of a Bessel filter is maximally flat in the passband.

The block input must be a discrete-time complex signal.

Note This block assumes a nominal impedance of 1 ohm.

Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block lets you specify the filter design parameters shown in the following table.

Design Method	Filter Design Parameters
Butterworth	Order, passband edge frequency
Chebyshev I	Order, passband edge frequency, passband ripple
Chebyshev II	Order, stopband edge frequency, stopband attenuation
Elliptic	Order, passband edge frequency, passband ripple, stopband attenuation
Bessel	Order, passband edge frequency

The Highpass RF Filter block designs the filters using the Signal Processing Toolbox™ filter design functions buttap, cheb1ap, cheb2ap, ellipap, and besselap.

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Dialog Box

The parameters displayed in the dialog box vary for different design methods. Only some of these parameters are visible in the dialog box at any one time.

You can change tunable parameters while the model is running.

Design method
: Filter design method. The design method can be `Butterworth`, `Chebyshev I`, `Chebyshev II`, `Elliptic`, or `Bessel`. Tunable.

Filter order
: Order of the filter.

Passband edge frequency (Hz)
: Passband edge frequency for Butterworth, Chebyshev I, elliptic, and Bessel designs. Tunable.

Stopband edge frequency (Hz)
: Stopband edge frequency for Chebyshev II designs. Tunable.

Passband ripple in dB
Passband ripple for Chebyshev I and elliptic designs. Tunable.

Stopband attenuation in dB
Stopband attenuation for Chebyshev II and elliptic designs. Tunable.

Finite impulse response filter length
Desired length of the baseband-equivalent impulse response for the filter.

Center frequency (Hz)
Center of the modeling frequencies.

Sample time
Time interval between consecutive samples of the input signal.

See Also

Amplifier, Bandpass RF Filter, Bandstop RF Filter, Lowpass RF Filter, Mixer

buttap, cheb1ap, cheb2ap, ellipap, besselap (Signal Processing Toolbox)

Purpose

Connection block from Simulink® environment to RF physical blocks

Library

Input/Output Ports sublibrary of the Physical library

Description

The Input Port block serves as a connecting port from the Simulink, or mathematical, part of the model to an RF physical part of the model. The Input Port block lets you provide the parameter data needed to calculate the modeling frequencies and the baseband-equivalent impulse response for the physical subsystem.

For more information about how the Input Port block converts the mathematical Simulink signals to RF Blockset™ physical modeling environment signals, see "Converting to and from Simulink® Signals" on page A-33. For more information about connecting mathematical and physical parts of a model, see Chapter 2, "Modeling an RF System".

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the Input Port mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Dialog Box

Finite impulse response filter length

Desired length of the baseband-equivalent impulse response for the physical model. The longer the FIR filter in the time domain, the finer the frequency resolution in the frequency domain. The frequency resolution is approximately equal to 1/ (**Finite impulse response filter length** * **Sample time (s)**). For a graphical representation of this parameter, see "Baseband-Equivalent Modeling" on page A-13.

Note The RF Blockset algorithm uses the next power of 2 greater than the specified filter length in its calculations and then truncates the impulse response to the specified length. As a result, you get different results when you set the **Finite impulse response filter length** parameter to a number that is not a power of 2. For more information about how the software calculates the baseband-equivalent impulse response, see "Calculating the Baseband-Equivalent Impulse Response" on page A-16.

Fractional bandwidth of guard bands

Fraction of modeling bandwidth over which to taper the edges of the transfer function of the system when creating the baseband-equivalent model. This parameter defines the ratio of the bandwidth of sections that are tapered using a Tukey, or cosine-tapered, window to the bandwidth of the constant, or untapered, sections.

A value less than or equal to 0 tells the Input Port block to use a rectangular (`rectwin`) window. A value greater than or equal to 1 tells the Input Port block to use a `hann` window.

The blockset uses the Signal Processing Toolbox™ `tukeywin` function to generate the window. The following figure shows the resulting frequency-domain window for several values of the **Fractional bandwidth of guard bands** parameter.

See “Creating a Complex Baseban d-Equivalent Model” on page A-13 for information about how the Input Port block applies this window to reduce the Gibbs phenomenon (also known as ringing), and other artifacts in the baseband-equivalent model of the system.

Modeling delay (samples)

Number of time samples by which to delay the impulse response of the baseband-equivalent model to ensure that the baseband-equivalent model has a causal response.

See “Creating a Complex Baseband-Equivalent Model” on page A-13 for information on how the Input Port block applies this delay to ensure a causal response.

Center frequency (Hz)
: Center of the modeling frequencies. See the Output Port block reference page for information about calculating the modeling frequencies.

Sample time (s)
: Time interval between consecutive samples of the input signal.

Note The Input Port block does not automatically inherit a sample time from its input signal. The specified **Sample time (s)** value must match the sample time of the input signal.

Source impedance (ohms)
: Source impedance of the RF network described in the physical model to which it connects.

Add noise
: If you select this parameter, noise data in the RF physical blocks that are bracketed by the Input Port block and Output Port block is taken into consideration. If you do not select this parameter, noise data is ignored.

Initial seed
: Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise. This parameter becomes visible if you select the **Add noise** parameter.

See Also

Output Port

Purpose

Model LC bandpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description

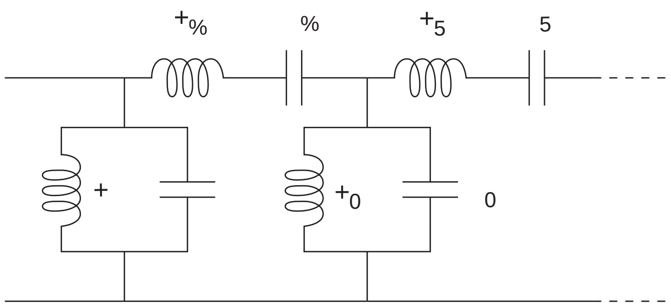
The LC Bandpass Pi block models the LC bandpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series pair, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series pair. For each shunt pair, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt pair.

The LC Bandpass Pi block then cascades the ABCD-parameters for each series and shunt pair at each of the modeling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block for information about determining the modeling frequencies.

The LC bandpass pi network object is a two-port network as shown in the following circuit diagram.



[L_1, L_2, L_3, L_4, ...] is the value of the `'L'` property, and [C_1, C_2, C_3, C_4, ...] is the value of the `'C'` property.

Dialog Box

Main Tab

Inductance (H)
: Vector containing the inductances, in order from source to load, of all inductors in the network. The inductance vector must contain at least three elements. All values must be strictly positive.

Capacitance (F)
: Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to the length of the vector you provide in the Inductance parameter. All values must be strictly positive.

Visualization Tab

For infor mation about plotting the filt er parameters, see Chapter 3, “Plottin g Model Data”.

Examples

Using a Ladder Filter Block to Filter Gaussian Noise

This example provides complex rand om noise in Gaussian form as input to an LC Bandpass Pi block. A Signal Processing Blockset™ Spectrum Scope block plots the filtered output.

The Signal Processing Blockset Random Source block produces frame-based output at 512 samples per frame. Its Sample time parameter is set to 1.0e-9. This sample time must match the sample time for the physical part of the model, which you provide in the Input Port block diagram.

The Input Port block specifies Finite impulse response filter length as 256, Center frequency as 700.0e6 Hz, Sample time as 1.0e-9, and Source impedance as 50 ohms.

The LC Bandpass Pi block provides the inductances for three inductors, in order from source to load, [1.4446e-9, 4.3949e-8, 1.4446e-9]. Similarly, it provides the capacitances for three capacitors [3.5785e-11, 1.1762e-12, 3.5785e-11].

The following plot shows a sample of the baseband-equivalent RF signal generated by this LC Bandpass Pi block. Zero (0) on the frequency axis corresponds to the center frequency specified in the Input Port block. The bandwidth of the frequency spectrum is 1/sample time. You specify the **Sample time** parameter in the Input Port block.

The Axis Properties of the Spectrum Scope block have been adjusted to show the frequencies above and below the carrier. The Minimum Y-limit parameter is -90, and Maximum Y-limit is 0.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

LC Bandpass Tee

Purpose Model LC bandpass tee network

Library Ladder Filters sublibrary of the Physical library

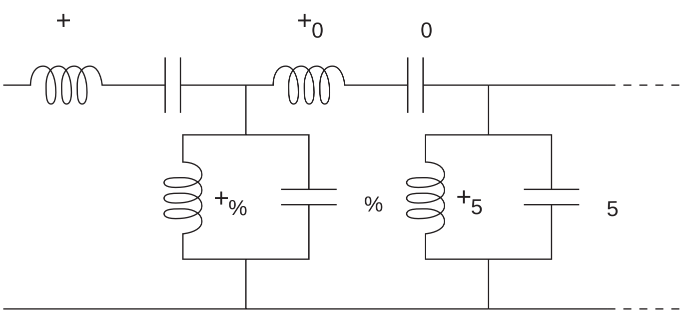
Description The LC Bandpass Tee block models the LC bandpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series pair, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series pair. For each shunt pair, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt pair.

The LC Bandpass Tee block then cascades the ABCD-parameters for each series and shunt pair at each of the modeling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block reference page for information about determining the modeling frequencies.

The LC bandpass tee network object is a two-port network as shown in the following circuit diagram.



[L_1, L_2, L_3, L_4, ...] is the value of the `'L'` property, and [C_1, C_2, C_3, C_4, ...] is the value of the `'C'` property.

Dialog Box

Main Tab

Inductance (H)

Vector containing the inductances, in order from source to load, of all inductors in the network. The inductance vector must contain at least three elements. All values must be strictly positive.

Capacitance (F)

Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model LC bandstop pi network

Library

Ladder Filters sublibrary of the Physical library

Description

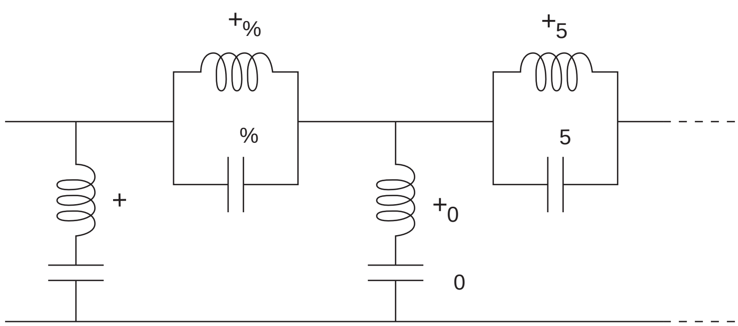
The LC Bandstop Pi block models the LC bandstop pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series pair, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series pair. For each shunt pair, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt pair.

The LC Bandstop Pi block then cascades the ABCD-parameters for each series and shunt pair at each of the modeling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block for information about determining the modeling frequencies.

The LC bandstop pi network object is a two-port network as shown in the following circuit diagram.



$[L_1, L_2, L_3, L_4, ...]$ is the value of the `'L'` property, and $[C_1, C_2, C_3, C_4, ...]$ is the value of the `'C'` property.

Dialog Box

Main Tab

Inductance (H)

Vector containing the inductances, in order from source to load, of all inductors in the network. The inductance vector must contain at least three elements. All values must be strictly positive.

Capacitance (F)

Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to the length of the vector you provide in the Inductance parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model LC bandstop tee network

Library

Ladder Filters sublibrary of the Physical library

Description

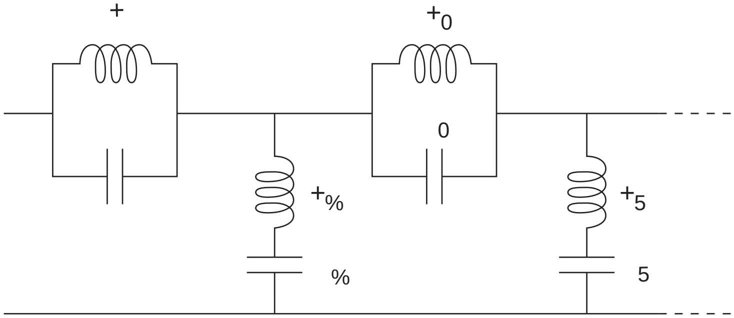
The LC Bandstop Tee block models the LC bandstop tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series pair, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series pair. For each shunt pair, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt pair.

The LC Bandstop Tee block then cascades the ABCD-parameters for each series and shunt pair at each of the modeling frequencies, and converts the cascaded parame ters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block for information about determining the modeling frequencies.

The LC bandstop tee network object is a two-port network as shown in the following circuit diagram.



[L_1, L_2, L_3, L_4, ...] is the value of the 'L' property, and [C_1, C_2, C_3, C_4, ...] is the value of the 'C' property.

Dialog Box

Main Tab

Inductance (H)
Vector containing the inductances, in order from source to load, of all inductors in the network. The inductance vector must contain at least three elements. All values must be strictly positive.

Capacitance (F)
Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model LC highpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description

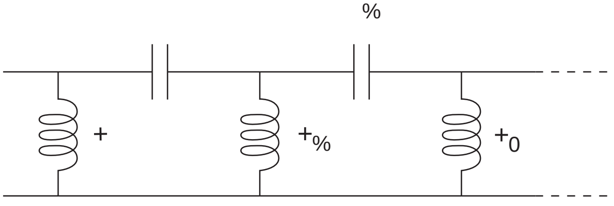
The LC Highpass Pi block models the LC highpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series circuit, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series circuit. For each shunt, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt circuit.

The LC Highpass Pi block then cascades the ABCD-parameters for each circuit element at each of the modeling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block reference page for information about determining the modeling frequencies.

The LC highpass pi network object is a two-port network as shown in the following circuit diagram.



[L_1, L_2, L_3, ...] is the value of the `'L'` property, and [C_1, C_2, ...] is the value of the `'C'` property.

Dialog Box

Main Tab

Inductance (H)
: Vector containing the inductances, in order from source to load, of all inductors in the network. The inductance vector must contain at least two elements. All values must be strictly positive.

Capacitance (F)
: Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to or one less than the length of the vector you provide in the Inductance parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

LC Highpass Tee

Purpose

Model LC highpass tee network

Library

Ladder Filters sublibrary of the Physical library

Description

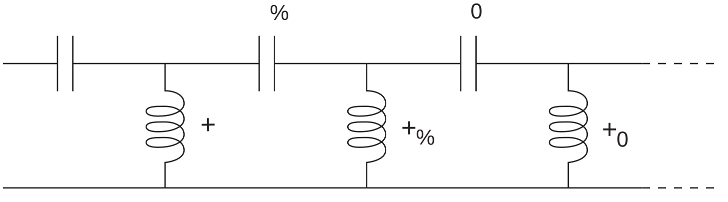
The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series circuit, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series circuit. For each shunt, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt circuit.

The LC Highpass Tee block then cascades the ABCD-parameters for each circuit element at each of the modeling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block reference page for information about determining the modeling frequencies.

The LC highpass tee network object is a two-port network as shown in the following circuit diagram.



$[L_1, L_2, L_3, ...]$ is the value of the `'L'` property, and $[C_1, C_2, C_3, ...]$ is the value of the `'C'` property.

Dialog Box

Main Tab

Inductance (H)
Vector containing the inductances, in order from source to load, of all inductors in the network. All values must be strictly positive. The vector cannot be empty.

Capacitance (F)
Vector containing the capacitances, in order from source to load, of all capacitors in the network. The capacitance vector must contain at least two elements. Its length must be equal to or one greater than the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model LC lowpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description

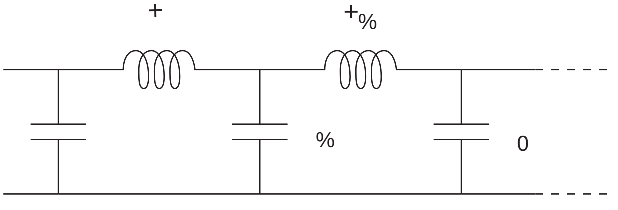
The LC Lowpass Pi block models the LC lowpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. For each series circuit, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series circuit. For each shunt, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt circuit.

The LC Lowpass Pi block then cascades the ABCD-parameters for each circuit element at each of the modeling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block reference page for information about determining the modeling frequencies.

The LC lowpass pi network object is a two-port network as shown in the following circuit diagram.



[L_1, L_2, ...] is the value of the `'L'` property, and [C_1, C_2, C_3, ...] is the value of the `'C'` property.

Dialog Box

Main Tab

Inductance (H)

Vector containing the inductances, in order from source to load, of all inductors in the network. All values must be strictly positive. The vector cannot be empty.

Capacitance (F)

Vector containing the capacitances, in order from source to load, of all capacitors in the network. The capacitance vector must contain at least two elements. Its length must be equal to or one greater than the length of the vector you provide in the Inductance parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model LC lowpass tee network

Library

Ladders Filters sublibrary of the Physical library

Description

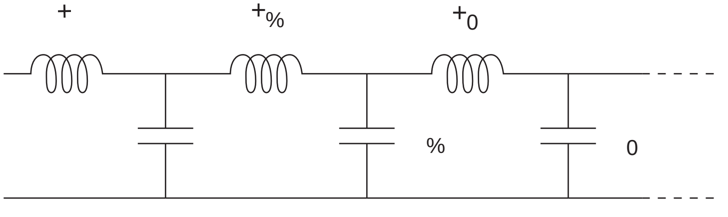
The LC Lowpass Tee block models the LC lowpass tee network described in the block dialog box in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each freq uency contained in the vector of modeling frequencies. For each series circuit, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series circuit. For each shunt, A = 1, B = 0, C = Y, and D = 1, where Y is the admittance of the shunt circuit.

The LC Lowpass Tee block then cascades the ABCD-parameters for each circuit element at each of the mo deling frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox™ `abcd2s` function.

See the Output Port block reference page for information about determining the modeling frequencies.

The LC lowpass tee network object is a two-port network as shown in the following circuit diagram.



[L_1, L_2, L_3, ...] is the value of the 'L' property, and [C_1, C_2, C_3, ...] is the value of the 'C' property.

Dialog Box

Main Tab

Inductance (H)

Vector containing the inductances, in order from source to load, of all inductors in the network. The inductance vector must contain at least two elements. All values must be strictly positive.

Capacitance (F)

Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to or one less than the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

Examples

See the LC Bandpass Pi block for an example of an LC filter.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Standard lowpass RF filters in baseband-equivalent complex form

Library

Mathematical

Description

The Lowpass RF Filter block lets you design standard analog lowpass filters, implemented in baseband-equivalent complex form. The following table describes the available design methods.

Design Method	Description
Butterworth	The magnitude response of a Butterworth filter is maximally flat in the passband and monotonic overall.
Chebyshev I	The magnitude response of a Chebyshev I filter is equiripple in the passband and monotonic in the stopband.
Chebyshev II	The magnitude response of a Chebyshev II filter is monotonic in the passband and equiripple in the stopband.
Elliptic	The magnitude response of an elliptic filter is equiripple in both the passband and the stopband.
Bessel	The delay of a Bessel filter is maximally flat in the passband.

The block input must be a discrete-time complex signal.

Note This block assumes a nominal impedance of 1 ohm.

Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

Design Method	Filter Design Parameters
Butterworth	Order, passband edge frequency
Chebyshev I	Order, passband edge frequency, passband ripple
Chebyshev II	Order, stopband edge frequency, stopband attenuation
Elliptic	Order, passband edge frequency, passband ripple, stopband attenuation
Bessel	Order, passband edge frequency

The Lowpass RF Filter block designs the filters using the Signal Processing Toolbox™ filter design functions `buttap`, `cheb1ap`, `cheb2ap`, `ellipap`, and `besselap`.

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Dialog Box

The parameters displayed in the dialog box vary for different design methods. Only some of these parameters are visible in the dialog box at any one time.

Parameters that are tunable can be changed while the model is running.

Design method
: Filter design method. The design method can be Butterworth, Chebyshev I, Chebyshev II, Elliptic, or Bessel. Tunable.

Filter order
: Order of the filter.

Passband edge frequency (Hz)
: Passband edge frequency for Butterworth, Chebyshev I, elliptic, and Bessel designs. Tunable.

Stopband edge frequency (Hz)
: Stopband edge frequency for Chebyshev II designs. Tunable.

Passband ripple in dB
: Passband ripple for Chebyshev I and elliptic designs. Tunable.

Stopband attenuation in dB
: Stopband attenuation for Chebyshev II and elliptic designs. Tunable.

Finite impulse response filter length
: Desired length of the baseband-equivalent impulse response for the filter.

Center frequency (Hz)
: Center of the modeling frequencies.

Sample time (s)
: Time interval between consecutive samples of the input signal.

See Also

Amplifier, Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Mixer

buttap, cheb1ap, cheb2ap, ellipap, besselap (Signal Processing Toolbox)

Purpose

Model microstrip transmission line

Library

Transmission Lines sublibrary of the Physical library

Description

The Microstrip Transmission Line block models the microstrip transmission line described in the block dialog in terms of its frequency-dependent S-parameters. A microstrip transmission line is shown in cross-section in the following figure. Its physical characteristics include the microstrip width (w), the microstrip thickness (t), the substrate height (d), and the relative permittivity constant ( ).

The block lets you model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model a microstrip transmission line as a stubless line, the Microstrip Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, d, and the complex propagation constant, k, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the specified conductor strip width, substrate height, conductor strip thickness, relative permittivity constant, conductivity, and dielectric loss tangent of the microstrip line, as described in [1].

The Microstrip Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Microstrip Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Strip width (m)
: Width of the microstrip transmission line.

Substrate height (m)
: Thickness of the dielectric on which the microstrip resides.

Strip thickness (m)
: Physical thickness of the microstrip.

Relative permittivity constant
: Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space .

Conductivity in conductor (S/m)
: Conductivity of the conductor in siemens per meter.

Loss tangent in dielectric
Loss angle tangent of the dielectric.

Transmission line length (m)
Physical length of the transmission line.

Stub mode
Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`.

Termination of stub
Stub termination for stub modes `Shunt` and `Series`. Choices are `Open` or `Short`. This parameter becomes visible only when **Stub mode** is set to `Shunt` or `Series`.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, "Plotting Model Data".

References

[1] Gupta, K.C., G. Ramesh, I. Bahl, and P. Bhartia, *Microstrip Lines and Slotlines*, Second Edition, Artech House, 1996. pp. 102-109.

See Also Coaxial Transmission Line, Coplanar Waveguide Transmission Line, General Passive Network, Transmission Line, Parallel-Plate Transmission Line, Two-Wire Transmission Line

Purpose

Complex baseband model of mixer and local oscillator with phase noise

Library

Mathematical

Description

The Mixer block generates a complex baseband model of the following:

- A mixer
- A local oscillator with phase noise whose spectrum is characterized by a 1/f slope

The Mixer block includes both the IF and RF signals as complex-baseband equivalent signals. Both the IF and RF center frequencies are represented as 0 hertz. The amplitude of the noise spectrum is specified by the noise power contained in a 1-hertz bandwidth offset from the carrier by a specified frequency.

Note This block assumes a nominal impedance of 1 ohm.

The block applies the phase noise to the signal as follows:

1. Generates additive white Gaussian noise (AWGN) and filters it with a digital filter.
2. Adds the resulting phase noise to the angle component of the input signal.

You can view the block's implementation of phase noise by right-clicking the block and selecting **Look under mask** from the pop-up menu. The following figure shows the implementation.

You can view th e construction of the Noise Source subsystem by double-click ing it.

Dialog Box

You can change parameters that are marked as tunable in the following descriptions while the model is running.

Conversion gain (dB)
: Scalar specifying the conversion gain for the mixer. Use a negative value to specify loss. Tunable.

Phase noise level (dBc/Hz)
: Scalar specifying the phase noise level in decibels relative to the carrier, per hertz. Tunable.

Frequency offset (Hz)
: Scalar specifying the frequency offset. Tunable.

Initial seed
: Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise.

References

[1] Kasdin, N.J., “Discrete Simulation of Colored Noise and Stochastic Processes and 1/(f^alpha); Power Law Noise Generation,” The Proceedings of the IEEE, May, 1995, Vol. 83, No. 5.

See Also

Amplifier, Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Lowpass RF Filter

Purpose

Connection block from RF physical blocks to Simulink® environment

Library

Input/Output Ports sublibrary of the Physical library

Description

The Output Port block produces the baseband-equivalent time-domain response of an input signal traveling through a series of RF physical components. The Output Port block

1. Partitions the RF physical components into linear and nonlinear subsystems.
2. Extracts the complex impulse response of the linear subsystem for baseband-equivalent modeling of the RF linear system.
3. Extracts the nonlinear AMAM/AMPM modeling for RF nonlinearity.

The Output Port block also serves as a connecting port from an RF physical part of the model to the Simulink, or mathematical, part of the model. For more information about how the Output Port block converts the RF Blockset™ physical modeling environment signals to mathematical Simulink signals, see “Converting to and from Simulink® Signals” on page A-33. For more information about connecting mathematical and physical parts of a model, see Chapter 2, “Modeling an RF System”.

Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

Linear Subsystem

For the linear subsystem, the Output Port block uses the Input Port block parameters and the interpolated S-parameters calculated by each of the cascaded physical blocks to calculate the baseband-equivalent impulse response. Specifically, it

1 Determines the modeling frequencies f as an N-element vector. The modeling frequencies are a function of the center frequency f_c, the sample time t_s, and the finite impulse response filter length N, all of which you specify in the Input Port block dialog box.

The nth element of f, f_n, is given by

$$f_n = f_{min} + \frac{n-1}{t_s N} \quad n = 1, \ldots, N$$

where

$$f_{min} = f_c - \frac{1}{2t_s}$$

2 Calculates the passband transfer function for the frequency range as

$$H(f) = \frac{V_L(f)}{V_S(f)}$$

where V_S and V_L are the source and load voltages, and f represents the modeling frequencies. More specifically,

$$H(f) = \frac{S_{21} * (1 + {}_{l}) * (1 - {}_{s})}{2 * (1 - S_{22} * {}_{l})(1 - {}_{in} * {}_{s})}$$

where

$${}_{l} = \frac{Z_l - Z_o}{Z_l + Z_o}$$

$${}_{s} = \frac{Z_s - Z_o}{Z_s + Z_o}$$

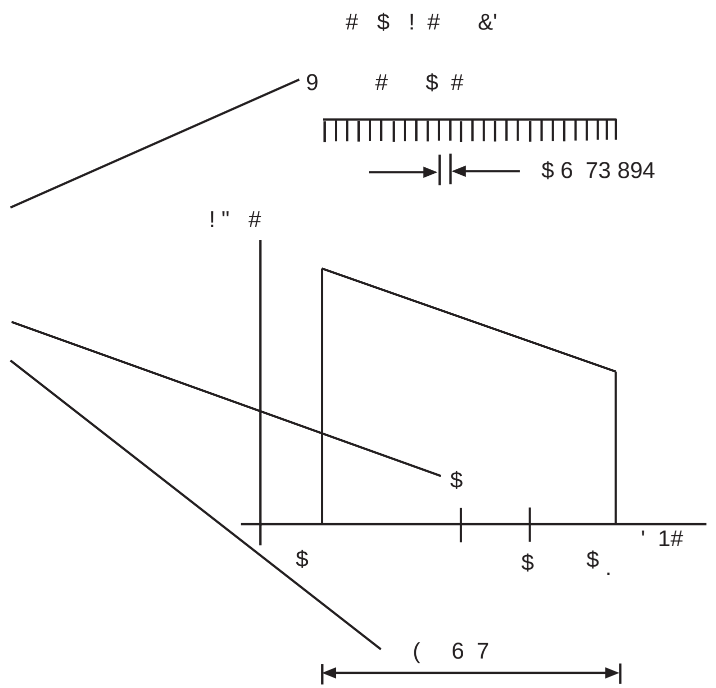
$${}_{in} = S_{11} + S_{12} * S_{21} * \frac{{}_{l}}{(1 - S_{22} * {}_{l})}$$

and

- Z_S is the source impedance.
- Z_L is the load impedance.
- S_{ij} are the S-parameters of a two-port network.

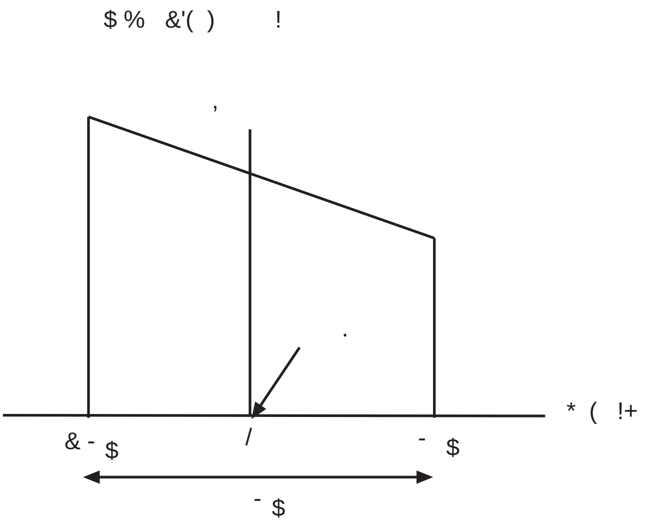
The blockset derives the passband transfer function from the Input Port block parameters as shown in the following figure:

: #



3 Transla tes the passband transfer function to baseband as ,
where f_c is the specified center frequency.

The baseband transfer function is shown in the following figure.



4. Obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. For faster simulation, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. Then, it truncates the impulse response to a length equal to the filter length specified.

For the linear subsystem, the Output Port block uses the calculated impulse response as input to the Signal Processing Blockset™ Digital Filter block to determine the output.

Nonlinear Subsystem

The nonlinear subsystem is implemented by AM/AM and AM/PM nonlinear models, as shown in the following figure.

The nonlinearities of AM/AM and AM/PM conversions are extracted from the power data of an amplifier or mixer by the equations

where AM_{in} is the AM of the input voltage, AM_{out} and PM_{out} are the AM and PM of the output voltage, R_s, is the source resistance (50 ohms), R_l is the load resistance (50 ohms), P_{in} is the input power, P_{out} is the output power, and Phase is the phase shift between the input and output voltage.

Note You can provide power data via a `.amp` file. See "AMP File Format" in the RF Toolbox™ documentation for information about this format.

The following figure shows the original power data of an amplifier.

This figure shows the extracted AM/AM nonlinear conversion.

Dialog Box

Main Tab

Load impedance (ohms)
: Load impedance of the RF network described in the physical model to which it connects.

Visualization Tab

This tab shows parameters for creating plots if you display the Output Port mask after you perform one or more of the following actions:

- Run a model with two or more blocks between the Input Port block and the Output Port block.
- Click the Update Diagram button to initialize a model with two or more blocks between the Input Port block and the Output Port block.

For information about plotting, see Chapter 3, “Plotting Model Data”.

See Also

Input Port

s2y (RF Toolbox)

Purpose Model parallel-plate transmission line

Library Transmission Lines sublibrary of the Physical library

Description The Parallel-Plate Transmission Line block models the parallel-plate transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A parallel-plate transmission line is shown in cross-section in the following figure. Its physical characteristics include the plate width and the plate separation .

The block lets you model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model a parallel-plate transmission line as a stubless line, the Parallel-Plate Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, d, and the complex propagation constant, k, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the resistance (R), inductance (L), conductance (G), and capacitance (C) per unit length (meters) as follows:

$$Z_0 = \sqrt{\frac{R + j2\ fL}{G + j2\ fC}}$$

where

In these equations, is the conductivity in the conductor and is the conductivity in the dielectric. is the permeability of the dielectric, is its permittivity , and skin depth is calculated as . f is the vector of modeling frequencies for the specified parameters. See the

Output Port block reference page for information about determining the modeling frequencies.

The Parallel-Plate Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Parallel-Plate Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Plate width (m)
: Physical width of the parallel-plate transmission line.

Plate separation (m)
: Thickness of the dielectric separating the plates.

Relative permeability constant
: Relative permeability of the dielectric expressed as the ratio of the permeability of the dielectric to permeability in free space.

Relative permittivity constant
: Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space.

Conductivity in conductor (S/m)
: Conductivity of the conductor in siemens per meter.

Conductivity in dielectric (S/m)
: Conductivity of the dielectric in siemens per meter.

Transmission line length (m)
: Physical length of the transmission line.

Stub mode

Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`.

Termination of stub

Stub termination for stub modes `Shunt` and `Series`. Choices are `Open` or `Short`. This parameter becomes visible only when **Stub mode** is set to `Shunt` or `Series`.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, “Plotting Model Data”.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

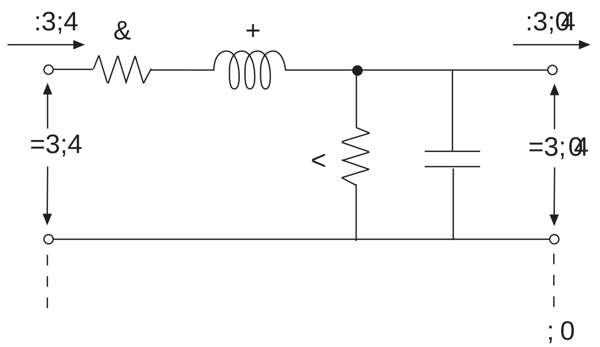
See Also

Coaxial Transmission Line, Coplanar Waveguide Transmission Line, General Passive Network, Transmission Line, Microstrip Transmission Line, Two-Wire Transmission Line

Purpose Model RLCG transmission line

Library Transmission Lines sublibrary of the Physical library

Description The RLCG Transmission Line block models the RLCG transmission line described in the block dialog box in terms of its frequency-dependent resistance, inductance, capacitance, and conductance. The transmission line, which can be lossy or lossless, is treated as a two-port linear network.



where .

The block lets you model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model an RLCG transmission line as a stubless line, the RLCG Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, d, and the complex propagation constant, k, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the resistance (R), inductance (L), conductance (G), and capacitance (C) per unit length (meters) as follows:

$$Z_0 = \sqrt{\frac{R + j2\ fL}{G + j2\ fC}}$$

The RLCG Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the RLCG Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Resistance per length (ohms/m)
Vector of resistance values in ohms per meter.

Inductance per length (H/m)
Vector of inductance values in henries per meter.

Capacitance per length (F/m)
Vector of capacitance values in farads per meter.

Conductance per length (S/m)
Vector of conductance values in siemens per meter.

Frequency (Hz)
Vector of frequency values at which the resistance, inductance, capacitance, and conductance values are known.

Interpolation method
: Specify the interpolation method the block uses to calculate the parameter values at the modeling frequencies. Your choices are `Linear`, `Spline`, or `Cubic`.

Transmission line length (m)
: Physical length of the transmission line.

Stub mode
: Type of stub. Your choices are `Not a stub`, `Shunt`, or `Series`.

Termination of stub
: Stub termination for stub modes `Shunt` and `Series`. Choices are `Open` or `Short`. This parameter becomes visible only when **Stub mode** is set to `Shunt` or `Series`.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, “Plotting Model Data”.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

See Also

Coaxial Transmission Line, Coplanar Waveguide Transmission Line, General Passive Network, Parallel-Plate Transmission Line, Transmission Line, Microstrip Transmission Line, Two-Wire Transmission Line

Purpose

Model nonlinear amplifier using S-parameters

Library

Amplifiers sublibrary of the Physical library

Description

The S-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent S-parameters, the frequencies and reference impedance of the S-parameters, noise data, and nonlinearity data.

Network Parameters

In the **S-parameters** field of the block dialog box, provide the S-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the S-parameters as an M-element vector. The elements of the frequencies vector must be in the same order as the S-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the S-parameters array and the vector of frequencies.

The S-Parameters Amplifier block interpolates the given S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, "RF Blockset™ Algorithms" for more details.

Nonlinearity

You can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the S-Parameters Amplifier block dialog box. You can also specify the

1 dB gain compression power and the output saturation power in the S-Parameters Amplifier block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1 Converting the specified value into IIP3 (if needed)

2 Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the S-Parameters Amplifier block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3 Computing the scaled input signal by multiplying the amplifier input signal by f.

4 Limiting the scaled input signal to a maximum value of 1.

5 Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1 Converting the specified third-order intercept value into OIP3 (if needed).

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the block dialog box):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the amplifier power gain, which is derived from the S-parameters.
- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}} \left(\frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05} \right)$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the S-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the S-Parameters Amplifier block dialog box.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Dialog Box

Main Tab

S-Parameters

S-parameters for a nonlinear amplifier in a 2-by-2-by-M array. M is the number of S-parameters.

Frequency (Hz)

Frequencies of the S-parameters as an M-element vector. The order of the frequencies must correspond to the order of the S-parameters in **S-Parameters**. All frequencies must be positive.

Reference impedance (ohms)

Reference impedance of the S-parameters as a scalar or a vector of length M. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data Tab

Noise type

Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature.

Noise figure (dB)

Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if **Noise type** is set to Noise figure.

Minimum noise figure (dB)

Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, ($S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient

Optimal amplifier source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance

Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor

Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)

Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be IIP3 (input intercept point) or OIP3 (output intercept point).

IP3 (dBm)
: Scalar power value of third-order intercept point. Use the default value, Inf, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. Use the default value, Inf, if the 1 dB compression point is unknown.

Output saturation power (dBm)

Scalar output power value the amplifier produces when fully saturated. The value is only used if 1 dB gain compression power (dBm) is specified. Use the default value, Inf, if saturation power is unknown.

Visualization Tab

For information about plotting the amplifier parameters, see Chapter 3, “Plotting Model Data”.

Examples

Plotting Parameters with the S-Parameters Amplifier Block

The following example specifies S-parameters [-.33+.71i, -.03i; 8.12-.02i, -.37-.37i] and [0.16+.20i, -.03-.04i; 7.71-8.04i, -.70-.12i] at frequencies 2.0 GHz and 2.1 GHz respectively, with a reference impedance of 50 ohms. The example uses the MATLAB® `cat` function to create the 2-by-2-by-2 S-parameters array.

```
cat(3,[-.33+0.71i,      -.03i;   8.12-.02i,  -.37-.37i],...
      [ .16+0.20i,  -.03-.04i;  7.71-8.04i,  -.70-.12i])
```

1 Type the following command at the MATLAB prompt to create a variable called `sparams` that stores the values of the S-parameters.

```
sparams = cat(3,...
      [-.33+0.71i,  -.03i;   8.12-.02i,  -.37-.37i],...
      [ .16+0.20i,  -.03-.04i;  7.71-8.04i,  -.70-.12i])
```

2 Set the S-Parameters Amplifier block parameters on the **Main** tab as follows:

- Set the **S-Parameters** parameter to `sparams`.
- Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

Click **Apply**. This action applies the specified settings.

3 Set the S-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Plot type** list, select X-Y plane.
- In the **Y parameter1** list, select S11.

Click **Plot**. This action creates an X-Y Plane plot of the S_{11} parameters using the frequencies taken from the **Frequency (Hz)** parameter on the **Main** tab.

See Also

General Amplifier, Output Port, Y-Parameters Amplifier, Z-Parameters Amplifier

interp1 (MATLAB)

S-Parameters Mixer

Purpose

Model mixer and local oscillator using S-parameters

Library

Mixer sublibrary of the Physical library

Description

The S-Parameters Mixer block models the nonlinear mixer described in the block dialog box, in terms of its frequency-dependent S-parameters, the frequencies and reference impedance of the S-parameters, noise data (including phase noise data), and nonlinearity data.

Network Parameters

The S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency. The other S-parameters also refer to the mixer input frequency.

The S-Parameters Mixer block interpolates the given S-parameters to determine their values at the modeling frequencies the Output Port block calculates. For more details about how the Output Port block calculates the modeling frequencies, see Appendix A, "RF Blockset™ Algorithms".

RF Blockset™ software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

$$\begin{bmatrix} b_1(f_{in}) \\ b_2(f_{out}) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1(f_{in}) \\ a_2(f_{out}) \end{bmatrix}$$

where

- f_{in} and f_{out} are the mixer input and output frequencies, respectively.
- a_1 and a_2 are the incident waves at the mixer input and output, respectively.

The interpolated S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency.

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the S-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the S-Parameters Mixer block dialog box.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Phase Noise

The S-Parameters Mixer block applies phase noise to a complex baseband signal. The block first generates additive white Gaussian noise (AWGN) and filters the noise with a digital FIR filter. It then adds the resulting noise to the angle component of the input signal.

The blockset computes the digital filter by:

1. Interpolating the specified phase noise level to determine the phase noise values at the modeling frequencies.
2. Taking the IFFT of the resulting phase noise spectrum to get the coefficients of the FIR filter.

Note If you specify phase noise as a scalar value, the blockset assumes the phase noise is constant at all modeling frequencies and does not have a 1/*f* slope. This assumption differs from that made by the Mathematical Mixer block.

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from the power data. Power data determines both IP3 and 1 dB gain compression power.

If the data source contains no power data, then you can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the S-Parameters Mixer block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the S-Parameters Mixer block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1. Converting the specified value into IIP3 (if needed)

2. Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the S-Parameters Mixer block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3. Computing the scaled input signal by multiplying the mixer input signal by f.

4. Limiting the scaled input signal to a maximum value of 1.

5. Applying an AM/AM conversion to the mixer gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

   where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1. Converting the specified third-order intercept value into OIP3 (if needed).

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the data source):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the mixer power gain, which is derived from the network parameters.
- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}} \left(\frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05} \right)$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Dialog Box

Main Tab

S-Parameters

S-parameters for a nonlinear mixer in a 2-by-2-by-M array. M is the number of S-parameters.

Frequency (Hz)

Frequencies of the S-parameters as an M-element vector. The order of the frequencies must correspond to the order of the S-parameters in S-Parameters . All frequencies must be positive. The following figure shows the correspondence between the S-parameters array and the vector of frequencies.

Reference impedance (ohms)

Reference impedance of the S-parameters as a scalar or a vector of length M. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type

Type of mixer. Choices are Downconverter (default) and Upconverter.

LO frequency (Hz)

Local oscillator frequency. If you choose Downconverter, the blockset computes the mixer output frequency, f_{out}, from the mixer input frequency, f_{in}, and the local oscillator frequency, f_{lo}, as . If you choose Upconverter, .

Note The mixer output frequency must be positive. This means that if you choose a downconverting mixer, f_{in} must be greater than f_{lo}. Otherwise, an error appears.

Noise Data Tab

Phase noise frequency offset (Hz)
: Vector specifying the frequency offset.

Phase noise level (dBc/Hz)
: Vector specifying the phase noise level.

Noise type
: Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature. This parameter is disabled if the data source contains noise data.

Noise figure (dB)
: Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at

the output, (S_i/N_i)/(S_o/N_o). This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)
: Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, (S_i/N_i)/(S_o/N_o). This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient
: Optimal mixer source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance
: Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, (S_i/N_i)/(S_o/N_o). This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)
: Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm)
: Scalar power value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data

or 1 dB compression point data. Use the default value, Inf, if the 1 dB compression point is unknown.

Output saturation power (dBm)
: Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if 1 dB gain compression power (dBm) is specified. Use the default value, Inf, if saturation power is unknown.

Visualization Tab

For information about plotting the amplifier parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Mixer, Output Port, Y-Parameters Mixer, Z-Parameters Mixer

Purpose

Model passive network using S-parameters

Library

Black Box Elements sublibrary of the Physical library

Description

The S-Parameters Passive Network block models the two-port passive network described in the block dialog box, in terms of its S-parameters and the frequencies and reference impedance of the S-parameters.

In the **S-Parameters** field of the block dialog box, provide the S-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the S-parameters as an M-element vector. The elements of the vector must be in the same order as the S-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the S-parameters array and the vector of frequencies.

The S-Parameters Passive Network block interpolates the given S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, “RF Blockset™ Algorithms” for more details.

Dialog Box

Main Tab

S-Parameters

S-parameters for a two-port passive network in a 2-by-2-by-M array. M is the number of S-parameters.

Frequency (Hz)

Frequencies of the S-parameters as an M-element vector. The order of the frequencies must correspond to the order of the S-parameters in **S-Parameters**. All frequencies must be positive.

Reference impedance (ohms)

Reference impedance of the network as a scalar or a vector of length M. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization Tab

For information about plotting circuit parameters, see Chapter 3, “Plotting Model Data”.

Examples

Plotting Parameters with the S-Parameters Passive Network Block

The following example specifies S-parameters [-.96-.23i, .03-.12i; .03-.12i, -.96-.23i] and [-.96-.11i, .02-.21i; .02-.21i, -.96-.11i] at frequencies 2.0 GHz and 2.1 GHz respectively. The example then plots these parameters.

The example first uses the MATLAB® `cat` function to create the 2-by-2-by-2 S-parameters array.

```
cat(3,[-.96-.23i, .03-.12i; .03-.12i, -.96-.23i],...
          [-.96-.11i, .02-.21i; .02-.21i, -.96-.11i])
```

You could also use the MATLAB `reshape` function. The following command produces the same result as previous command.

```
reshape([-.96-.23i;.03-.12i;.03-.12i;-.96-.23i;...
              -.96-.11i;.02-.21i;.02-.21i;-.96-.11i],2,2,2)
```

1 Type the following command at the MATLAB prompt to create a variable called `sparams` that stores the values of the S-parameters.

```
sparams = cat(3,...
          [-.96-.23i, .03-.12i; .03-.12i, -.96-.23i],...
          [-.96-.11i, .02-.21i; .02-.21i, -.96-.11i])
```

2 Set the S-Parameters Passive Network block parameters on the **Main** tab as follows:

- Set the **S-Parameters** parameter to `sparams`.
- Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

Click **Apply**. This action applies the specified settings.

3 Set the S-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Y parameter1** list, select `S21`.

Click **Plot**. This action creates an X-Y Plane plot of the magnitude of the S_{21} parameters, in decibels, in the frequency range 1.9 to 2.2 GHz.

See Also

General Circuit Element, General Passive Network, Output Port, Y-Parameters Passive Network, Z-Parameters Passive Network

interp1 (MATLAB)

Purpose

Model series capacitor

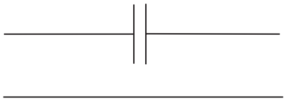
Library

Ladders Filters sublibrary of the Physical library

Description

The Series C block models the series capacitor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series C object is a two-port network, as shown in the following circuit diagram.



Dialog Box

Main Tab

Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab

For information about plotting the series capacitor parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model series inductor

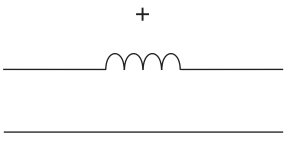
Library

Ladders Filters sublibrary of the Physical library

Description

The Series L block models the series inductor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series L object is a two-port network, as shown in the following circuit diagram.



Dialog Box

Main Tab

Inductance (H)
  Scalar value for the inductance. The value must be nonnegative.

Visualization Tab

For information about plotting the series inductor parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series R, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model series resistor

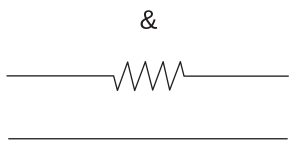
Library

Ladders Filters sublibrary of the Physical library

Description

The Series R block models the series resistor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series R object is a two-port network, as shown in the following circuit diagram.



Dialog Box

Main Tab

Resistance (ohms)

Scalar value for the resistance. The value must be nonnegative.

Visualization Tab

For information about plotting the series resistor parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series RLC, Shunt C, Shunt L, Shunt R, Shunt RLC

Purpose

Model series RLC network

Library

Ladders Filters sublibrary of the Physical library

Description

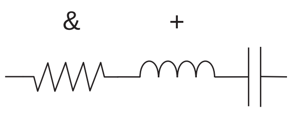
The Series RLC block models the series RLC network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox™ `abcd2s` function. See the Output Port block reference page for information about determining the modeling frequencies.

For this circuit, A = 1, B = Z, C = 0, and D = 1, where

and .

The series RLC object is a two-port network as shown in the following circuit diagram.



Dialog Box

Main Tab

Resistance (ohms)

Scalar value for the resistance. The value must be nonnegative.

Inductance (H)

Scalar value for the inductance. The value must be nonnegative.

Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, "Plotting Model Data".

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Shunt C, Shunt L, Shunt R, Shunt RLC

Shunt C

Purpose

Model shunt capacitor

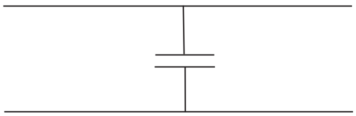
Library

Ladders Filters sublibrary of the Physical library

Description

The Shunt C block models the shunt capacitor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt C object is a two-port network, as shown in the following circuit diagram.



Dialog Box

Main Tab

Capacitance (F)
Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab

For information about plotting the shunt capacitor parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt L, Shunt R, Shunt RLC

Purpose

Model shunt inductor

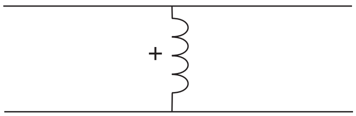
Library

Ladders Filters sublibrary of the Physical library

Description

The Shunt L block models the shunt inductor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt L object is a two-port network, as shown in the following circuit diagram.



Dialog Box

Main Tab

Inductance (H)
Scalar value for the inductance. The value must be nonnegative.

Visualization Tab

For information about plotting the shunt inductor parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt R, Shunt RLC

Purpose

Model shunt resistor

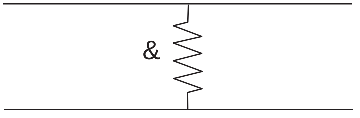
Library

Ladders Filters sublibrary of the Physical library

Description

The Shunt R block models the shunt resistor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt R object is a two-port network, as shown in the following circuit diagram.



Dialog Box

Main Tab

Resistance (ohms)
Scalar value for the resistance. The value must be nonnegative.

Visualization Tab

For information about plotting the shunt resistor parameters, see Chapter 3, "Plotting Model Data".

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt RLC

Purpose

Model shunt RLC network

Library

Ladders Filters sublibrary of the Physical library

Description

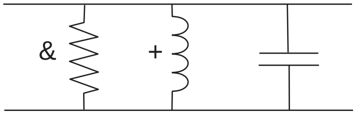
The Shunt RLC block models the shunt RLC network described in the block dialog box, in terms of its fre quency-dependent S-parameters.

For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and t hen converts the ABCD-parameters to S-parameters using the RF Toolbox™ `abcd2s` function. See the Output Port block reference page fo r information about determining the modeling frequencies.

For this circuit, A = 1, B = 0, C = Y, and D = 1, where

and .

The shunt RLC object is a two-port network as shown in the following circuit diagram.



Dialog Box

Main Tab

Resistance (ohms)
Scalar value for the resistance. The value must be nonnegative.

Inductance (H)
Scalar value for the inductance. The value must be nonnegative.

Capacitance (F)
Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab

For information about plotting the filter parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Highpass Tee, LC Lowpass Pi, LC Lowpass Tee, Series C, Series L, Series R, Series RLC, Shunt C, Shunt L, Shunt R

Purpose

Model general transmission line

Library

Transmission Lines sublibrary of the Physical library

Description

The Transmission Line block models the transmission line described in the block dialog box in terms of its physical parameters. The transmission line, which can be lossy or lossless, is treated as a two-port linear network.

The block enables you to model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model the transmission line as a stubless line, the Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the abcd2s function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, d, and the complex propagation constant, k, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 is the specified characteristic impedance. k is a vector whose elements correspond to the elements of f, a vector of modeling frequencies. The block calculates k from the specified parameters as , where is the attenuation coefficient and is the wave number. The attenuation coefficient is related to the specified loss, , by

and the wave number is related to the specified phase velocity, V_P, by

The phase velocity V_P is also known as the wave propagation velocity.

The Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Characteristic impedance (ohms)
: Characteristic impedance of the transmission line. The value can be complex.

Phase velocity (m/s)
: Propagation velocity of a uniform plane wave on the transmission line.

Loss (dB/m)
: Reduction in strength of the signal as it travels over the transmission line. Must be positive.

Transmission line length (m)
: Physical length of the transmission line.

Stub mode
: Type of stub. Choices are Not a stub, Shunt, or Series.

Termination of stub

Stub termination for stub modes Shunt and Series. Choices are Open or Short. This parameter becomes visible only when **Stub mode** is set to Shunt or Series.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, “Plotting Model Data”.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

See Also

Coaxial Transmission Line, Coplanar Waveguide Transmission Line, General Passive Network, Microstrip Transmission Line, Parallel-Plate Transmission Line, Two-Wire Transmission Line

Purpose

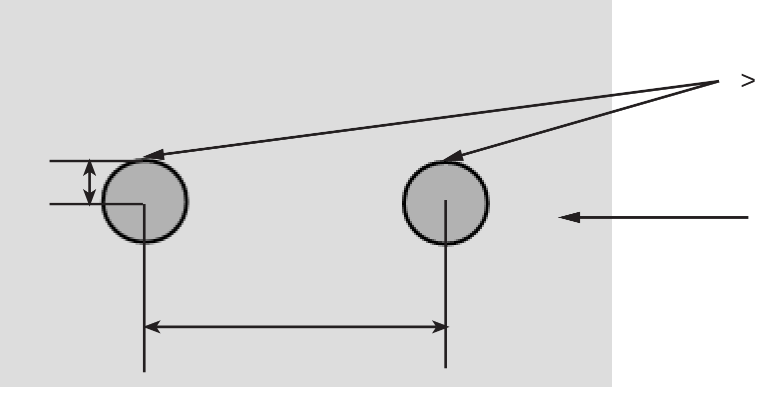
Model two-wire transmission line

Library

Transmission Lines sublibrary of the Physical library

Description

The Two-Wire Transmission Line block models the two-wire transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A two-wire transmission line is shown in cross-section in the following figure. Its physical characteristics include the radius of the wires a, the separation or physical distance between the wire centers S, and the relative permittivity and perme ability of the wires. RF Blockset™ software assumes the relative permittivi ty and permeability are uniform.



The block enables you to model the transmission line as a stub or as a stubless line.

Stubless Transmission Line

If you model a two-wire transmission line as a stubless line, the Two-Wire Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

The block calculates the ABCD-parameters using the physical length of the transmission line, d, and the complex propagation constant, k, using the following equations:

$$A = \frac{e^{kd} + e^{-kd}}{2}$$

$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

$$C = \frac{e^{kd} - e^{-kd}}{2 * Z_0}$$

$$D = \frac{e^{kd} + e^{-kd}}{2}$$

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the resistance (R), inductance (L), conductance (G), and capacitance (C) per unit length (meters) as follows:

$$Z_0 = \sqrt{\frac{R + j2\pi fL}{G + j2\pi fC}}$$

where

In these equations, is the conductivity in the conductor and is the conductivity in the dielectric. is the permeability of the dielectric, is its permittivity, and skin depth is calculated as . f is the vector of modeling frequencies for the specified parameters. See the Output Port block reference page for information about determining the modeling frequencies.

The Two-Wire Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Two-Wire Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Shunt`, the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

Series ABCD-Parameters

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.

Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

Dialog Box

Main Tab

Wire radius (m)
: Radius of the conducting wires of the two-wire transmission line.

Wire separation (m)
: Physical distance between the wires.

Relative permeability constant
: Relative permeability of the dielectric expressed as the ratio of the permeability of the dielectric to permeability in free space .

Relative permittivity constant
: Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space .

Conductivity in conductor (S/m)
: Conductivity of the conductor in siemens per meter.

Conductivity in dielectric (S/m)
: Conductivity of the dielectric in siemens per meter.

Transmission line length (m)
: Physical length of the transmission line.

Stub mode

Type of stub. Choices are Not a stub, Shunt, or Series.

Termination of stub

Stub termination for stub modes Shunt and Series. Choices are Open or Short. This parameter becomes visible only when Stub mode is set to Shunt or Series.

Visualization Tab

For information about plotting the transmission line parameters, see Chapter 3, “Plotting Model Data”.

References

[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

See Also

Coaxial Transmission Line, Coplanar Waveguide Transmission Line, General Passive Network, Transmission Line, Microstrip Transmission Line, Parallel-Plate Transmission Line

Purpose

Model nonlinear amplifier using Y-parameters

Library

Amplifiers sublibrary of the Physical library

Description

The Y-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent, the frequencies of the Y-parameters, noise data, and nonlinearity data

Network Parameters

In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Y-parameters as an M-element vector. The elements of the frequencies vector must be in the same order as the Y-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Y-parameters array and the vector of frequencies.

The Y-Parameters Amplifier block uses the RF Toolbox™ `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, “RF Blockset™ Algorithms” for more details.

Nonlinearity

You can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the Y-Parameters Amplifier block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the Y-Parameters Amplifier block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1. Converting the specified value into IIP3 (if needed)
2. Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the Y-Parameters Amplifier block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3. Computing the scaled input signal by multiplying the amplifier input signal by f.
4. Limiting the scaled input signal to a maximum value of 1.
5. Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

   where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1 Converting the specified third-order intercept value into OIP3 (if needed).

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the block dialog box):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the amplifier power gain, which is derived from the Y-parameters.
- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}} \frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05}$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the Y-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the Y-Parameters Amplifier block dialog box.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Dialog Box

Main Tab

Y-Parameters

Y-parameters for a nonlinear amplifier in a 2-by-2-by-M array. M is the number of Y-parameters.

Frequency (Hz)

Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in **Y-Parameters**. All frequencies must be positive.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data Tab

Noise type

Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature.

Noise figure (dB)
: Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)
: Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient
: Optimal amplifier source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance
: Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)
: Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IP3 (dBm)
: Scalar power value of third-order intercept point. Use the default value, `Inf`, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. Use the default value, `Inf`, if the 1 dB compression point is unknown.

Output saturation power (dBm)

Scalar output power value the amplifier produces when fully saturated. The value is only used if **1 dB gain compression power (dBm)** is specified. Use the default value, `Inf`, if saturation power is unknown.

Visualization Tab

For information about plotting the amplifier parameters, see Chapter 3, “Plotting Model Data”.

Examples

Plotting Parameters with the Y-Parameters Amplifier Block

The following example specifies Y-parameters [-.06+.58i, -.08i; 1.14-1.82i, -.07+.28i] and [.02-.21i, 0.03i; -.21+.72i, .03-.11i] at frequencies 2.0 GHz and 2.1 GHz respectively. It uses the MATLAB® `cat` function to create the 2-by-2-by-2 Y-parameters array

```
cat(3,[-.06+.58i, -.08i; 1.14-1.82i, -.07+.28i],...
      [ .02-.21i, 0.03i;  -.21+.72i,  .03-.11i])
```

1 Type the following command at the MATLAB prompt to create a variable called `yparams` that stores the values of the Y-parameters.

```
yparams = cat(3,...
      [-.06+.58i, -.08i; 1.14-1.82i, -.07+.28i],...
      [ .02-.21i, 0.03i;  -.21+.72i,  .03-.11i])
```

2 Set the Y-Parameters Amplifier block parameters on the **Main** tab as follows:

- Set the **Y-Parameters** parameter to `yparams`.
- Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

Click **Apply**. This action applies the specified settings.

3 Set the Y-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select User-specified.
- Set the **Frequency data (Hz)** parameter to [1.8e9:1.0e8:2.3e9].
- In the **Plot type** list, select X-Y plane.
- In the **Y parameter1** list, select S11.

Click **Plot**. This action creates an X-Y Plane plot of the S_{11} parameters in the frequency range 1.8 to 2.3 GHz.

See Also

General Amplifier, Output Port, S-Parameters Amplifier, Z-Parameters Amplifier

y2s (RF Toolbox)

interp1 (MATLAB)

Purpose

Model mixer and local oscillator using Y-parameters

Library

Mixer sublibrary of the Physical library

Description

The Y-Parameters Mixer block models the nonlinear mixer described in the block dialog box in terms of its frequency-dependent Y-parameters, the frequencies of the Y-parameters, noise data (including phase noise data), and nonlinearity data.

Network Parameters

The Y-parameter values all refer to the mixer input frequency.

The Y-Parameters Mixer block uses the RF Toolbox™ `y2s` function to convert the Y-parameters to S-parameters and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, "RF Blockset™ Algorithms" for more details.

RF Blockset™ software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

$$\begin{bmatrix} b_1(f_{in}) \\ b_2(f_{out}) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1(f_{in}) \\ a_2(f_{out}) \end{bmatrix}$$

where

- f_{in} and f_{out} are the mixer input and output frequencies, respectively.
- a_1 and a_2 are the incident waves at the mixer input and output, respectively.

The interpolated S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency.

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the Y-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the Y-Parameters Mixer block dialog box.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Phase Noise

The Y-Parameters Mixer block applies phase noise to a complex baseband signal. The block first generates additive white Gaussian noise (AWGN) and filters the noise with a digital FIR filter. It then adds the resulting noise to the angle component of the input signal.

The blockset computes the digital filter by:

1. Interpolating the specified phase noise level to determine the phase noise values at the modeling frequencies.
2. Taking the IFFT of the resulting phase noise spectrum to get the coefficients of the FIR filter.

Note If you specify phase noise as a scalar value, the blockset assumes the phase noise is constant at all modeling frequencies and does not have a 1/f slope. This assumption differs from that made by the Mathematical Mixer block.

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from the power data. Power data determines both IP3 and 1 dB gain compression power.

If the data source contains no power data, then you can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the Y-Parameters Mixer block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the Y-Parameters Mixer block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1 Converting the specified value into IIP3 (if needed)

2 Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the Y-Parameters Mixer block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3 Computing the scaled input signal by multiplying the mixer input signal by f.

4 Limiting the scaled input signal to a maximum value of 1.

5 Applying an AM/AM conversion to the mixer gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1 Converting the specified third-order intercept value into OIP3 (if needed).

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the data source):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the mixer power gain, which is derived from the network parameters.
- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2 OIP3_0}$$

$$c_5 = \frac{c_1^5}{4 P_0^2 10^{0.2}} \left(\frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05} \right)$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Dialog Box

Main Tab

Y-Parameters

Y-parameters for a nonlinear mixer in a 2-by-2-by-M array. M is the number of Y-parameters.

Frequency (Hz)

Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in Y-Parameters . All frequencies must be positive. The following figure shows the correspondence between the Y-parameters array and the vector of frequencies.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type

Type of mixer. Choices are Downconverter (default) and Upconverter .

LO frequency (Hz)

Local oscillator frequency. If you choose Downconverter , RF

Blockset software computes the mixer output frequency, f_{out},

from the mixer input frequency, f_{in}, and the local oscillator frequency, f_{lo}, as . If you choose Upconverter ,
.

Note The mixer output frequency must be positive. This means that if you choose a downconverting mixer, f_{in} must be greater than f_{lo}. Otherwise, an error appears.

Noise Data Tab

Phase noise frequency offset (Hz)
: Vector specifying the frequency offset.

Phase noise level (dBc/Hz)
: Vector specifying the phase noise level.

Noise type
: Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature. This parameter is disabled if the data source contains noise data.

Noise figure (dB)
: Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)
: Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient
: Optimal mixer source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance
: Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)

Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type

Type of third-order intercept point. The value can be IIP3 (input intercept point) or OIP3 (output intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm)
: Scalar power value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1 dB compression point data. Use the default value, `Inf`, if the 1 dB compression point is unknown.

Output saturation power (dBm)
: Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if 1 dB gain compression power (dBm) is specified. Use the default value, `Inf`, if saturation power is unknown.

Visualization Tab

For information about plotting the mixer parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Mixer, Output Port, S-Parameters Mixer, Z-Parameters Mixer

Purpose

Model passive network using Y-parameters

Library

Black Box Elements sublibrary of the Physical library

Description

The Y-Parameters Passive Network block models the two-port passive network described in the block dialog box, in terms of its Y-parameters and their associated frequencies.

In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Y-parameters as an M-element vector. The elements of the vector must be in the same order as the Y-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Y-parameters array and the vector of frequencies.

The Y-Parameters Passive Network block uses the RF Toolbox™ `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, “RF Blockset™ Algorithms” for more details.

Dialog Box

Main Tab

Y-Parameters
: Y-parameters for a two-port passive network in a 2-by-2-by-M array. M is the number of Y-parameters.

Frequency (Hz)
: Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in **Y-Parameters**. All frequencies must be positive.

Interpolation method
: The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation

Method	Description
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization Tab

For information about plotting circuit parameters, see Chapter 3, “Plotting Model Data”.

Examples

Plotting Parameters with the Y-Parameters Passive Network Block

The following example specifies Y-parameters [.23i, -.12i; -.12i, .23i] and [.02-.13i, -.02+.25i; -.02+.25i, .02-.13i] at frequencies 2.0 GHz and

2.1 GHz respectively. It uses the MATLAB® `cat` function to create the 2-by-2-by-2 Y-parameters array.

```
cat(3,[.23i,-.12i;-.12i,.23i],...
        [.02-.13i,-.02+.25i;-.02+.25i, .02-.13i])
```

1 Type the following command at the MATLAB prompt to create a variable called `yparams` that stores the values of the Y-parameters.

   ```
   yparams = cat(3,[.23i,-.12i;-.12i,.23i],...
           [.02-.13i,-.02+.25i;-.02+.25i, .02-.13i])
   ```

2 Set the Y-Parameters Passive Network block parameters on the **Main** tab as follows:

   - Set the **Y-Parameters** parameter to `yparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

   Click **Apply**. This action applies the specified settings.

3 Set the Y-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Plot type** list, select `Polar plane`.

Click **Plot**. This action creates a polar plane plot of the S_{11} parameters in the frequency range 1.9 to 2.2 GHz.

See Also

General Circuit Element, General Passive Network, Output Port, S-Parameters Passive Network, Z-Parameters Passive Network

y2s (RF Toolbox)

interp1 (MATLAB)

Z-Parameters Amplifier

Purpose

Model nonlinear amplifier using Z-parameters

Library

Amplifiers sublibrary of the Physical library

Description

The Z-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent Z-parameters, the frequencies of the Z-parameters, noise data, and nonlinearity data

Network Parameters

In the **Z-Parameters** field of the block dialog box, provide the Z-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Z-parameters as an M-element vector. The elements of the frequencies vector must be in the same order as the Z-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Z-parameters array and the vector of frequencies.

The Z-Parameters Amplifier block uses the RF Toolbox™ `z2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, “RF Blockset™ Algorithms” for more details.

Nonlinearity

You can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the Z-Parameters Amplifier block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the Z-Parameters Amplifier block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1. Converting the specified value into IIP3 (if needed)
2. Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the Z-Parameters Amplifier block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3. Computing the scaled input signal by multiplying the amplifier input signal by f.
4. Limiting the scaled input signal to a maximum value of 1.
5. Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

   where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1 Converting the specified third-order intercept value into OIP3 (if needed).

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the block dialog box):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the amplifier power gain, which is derived from the Z-parameters.
- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}} \frac{P_0}{OIP3_0} 10^{0.1} - 1 + 10^{-0.05}$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the Z-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the Z-Parameters Amplifier block dialog box.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Dialog Box

Main Tab

Z-Parameters
Z-parameters for a nonlinear amplifier in a 2-by-2-by-M array. M is the number of Z-parameters.

Frequency (Hz)
Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in **Z-Parameters**. All frequencies must be positive.

Interpolation method
The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data Tab

Noise type

Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature.

Noise figure (dB)
: Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)
: Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient
: Optimal amplifier source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance
: Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)
: Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IP3 (dBm)
: Scalar power value of third-order intercept point. Use the default value, `Inf`, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. Use the default value, `Inf`, if the 1 dB compression point is unknown.

Output saturation power (dBm)

Scalar output power value the amplifier produces when fully saturated. The value is only used if **1 dB gain compression power (dBm)** is specified. Use the default value, `Inf`, if saturation power is unknown.

Visualization Tab

For information about plotting the amplifier parameters, see Chapter 3, "Plotting Model Data".

Examples

Plotting Parameters with the Z-Parameters Amplifier Block

The following example specifies Z-parameters [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i] and [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i] at frequencies 2.0 GHz and 2.1 GHz respectively. It uses the MATLAB® cat function to create the 2-by-2-by-2 Z-parameters array.

```
cat(3,...
    [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i],...
    [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i])
```

1. Type the following command at the MATLAB prompt to create a variable called zparams that stores the values of the Z-parameters.

   ```
   zparams = cat(3,...
    [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i],...
    [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i])
   ```

2. Set the Z-Parameters Amplifier block parameters on the **Main** tab as follows:

   - Set the **Z-Parameters** parameter to zparams.
   - Set the **Frequency (Hz)** parameter to [2.0e9,2.1e9].

   Click **Apply**. This action applies the specified settings.

3 Set the Z-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Plot type** list, select `X-Y plane`.
- In the **Y parameter1** list, select `S11`.

Click Plot . This action creates an X-Y Plane plot of the S_{11} parameters in the frequency range 1.9 to 2.2 GHz.

See Also

General Amplifier, Output Port, S-Parameters Amplifier, Y-Parameters Amplifier

z2s (RF Toolbox)

interp1 (MATLAB)

Purpose

Model mixer and local oscillator using Z-parameters

Library

Mixer sublibrary of the Physical library

Description

The Z-Parameters Mixer block models the nonlinear mixer described in the block dialog box in terms of its frequency-dependent Z-parameters, the frequencies of the Z-parameters, noise data (including phase noise data), and nonlinearity data.

Network Parameters

The Z-parameter values all refer to the mixer input frequency.

The Z-Parameters Mixer block uses the RF Toolbox™ `z2s` function to convert the Z-parameters to S-parameters and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, “RF Blockset™ Algorithms” for more details.

RF Blockset™ software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

$$\begin{bmatrix} b_1(f_{in}) \\ b_2(f_{out}) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1(f_{in}) \\ a_2(f_{out}) \end{bmatrix}$$

where

- f_{in} and f_{out} are the mixer input and output frequencies, respectively.
- a_1 and a_2 are the incident waves at the mixer input and output, respectively.

The interpolated S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency.

Active Noise

You can specify active block noise in one of the following ways:

- Spot noise data in the Z-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the Z-Parameters Mixer block dialog box.

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

Phase Noise

The Z-Parameters Mixer block applies phase noise to a complex baseband signal. The block first generates additive white Gaussian noise (AWGN) and filters the noise with a digital FIR filter. It then adds the resulting noise to the angle component of the input signal.

The blockset computes the digital filter by:

1. Interpolating the specified phase noise level to determine the phase noise values at the modeling frequencies.
2. Taking the IFFT of the resulting phase noise spectrum to get the coefficients of the FIR filter.

Note If you specify phase noise as a scalar value, the blockset assumes the phase noise is constant at all modeling frequencies and does not have a 1/*f* slope. This assumption differs from that made by the Mathematical Mixer block.

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from the power data. Power data determines both IP3 and 1 dB gain compression power.

If the data source contains no power data, then you can enter either OIP3 or IIP3 data as a scalar value for nonlinearity in the Z-Parameters Mixer block dialog box. You can also specify the 1 dB gain compression power and the output saturation power in the Z-Parameters Mixer block dialog box.

If you do not specify the 1 dB gain compression power, the block ignores the output saturation power specification. The block computes and adds the nonlinearity from the OIP3 or IIP3 value by:

1 Converting the specified value into IIP3 (if needed)

2 Using the third-order input intercept point value in dBm to compute the factor, f, that scales the input signal before the Z-Parameters Mixer block applies the nonlinearity:

$$f = \sqrt{\frac{3}{IIP3(Watts)}} = \sqrt{\frac{3}{10^{(IIP3(dBm) - 30)/10}}}$$

3 Computing the scaled input signal by multiplying the mixer input signal by f.

4 Limiting the scaled input signal to a maximum value of 1.

5 Applying an AM/AM conversion to the mixer gain, according to the following cubic polynomial equation:

$$F_{AM/AM}(u) = u - \frac{u^3}{3}$$

where u is the magnitude of the scaled input signal, which is a unitless normalized input voltage.

If you specify the 1 dB gain compression power, the block computes and adds the nonlinearity to the input signal by:

1 Converting the specified third-order intercept value into OIP3 (if needed).

2 Converting the gain, OIP3, and 1dB compression data to linear, unitless values, normalized to 1 volt and the reference impedance Z_0 (which is specified in the data source):

$$G_0 = 10^{\frac{GAIN}{10dB}}$$

$$OIP3_0 = 10^{\frac{OIP3}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

$$P_0 = 10^{\frac{PCOMP}{10dBm}} \cdot 10^{-3} \cdot Z_0$$

where

- GAIN is the mixer power gain, which is derived from the network parameters.
- OIP3 is the output third-order intercept point.
- PCOMP is the output power at the 1 dB compression point.

3 Computing the coefficients of the polynomial, $F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s$, that determines the AM/AM conversion for the input signal s:

$$c_1 = \sqrt{G_0}$$

$$c_3 = -\frac{c_1^3}{2OIP3_0}$$

$$c_5 = \frac{c_1^5}{4P_0^2 10^{0.2}}\left(\frac{P_0}{OIP3_0}10^{0.1} - 1 + 10^{-0.05}\right)$$

4 Computing the input power at which the output saturates, if it is not specified, according to the following function:

$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9c_3^2 - 20c_1c_5}}{-10c_5}}$$

This value is the input power above which the block replaces the AM/AM conversion model with a constant output power value A_{sat}.

5 Applying the AM/AM conversion to the input signal

Dialog Box

Main Tab

Z-Parameters

Z-parameters for a nonlinear mixer in a 2-by-2-by-M array. M is the number of Z-parameters.

Frequency (Hz)

Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in Z-Parameters. All frequencies must be positive. The following figure shows the correspondence between the Z-parameters array and the vector of frequencies.

Interpolation method

The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type

Type of mixer. Choices are Downconverter (default) and Upconverter.

LO frequency (Hz)

Local oscillator frequency. If you choose Downconverter, the blockset computes the mixer output frequency, f_{out}, from the mixer input frequency, f_{in}, and the local oscillator frequency, f_{lo}, as . If you choose Upconverter , .

Note The mixer output frequency must be positive. This means that if you choose a downconverting mixer, f_{in} must be greater than f_{lo}. Otherwise, an error appears.

Noise Data Tab

Phase noise frequency offset (Hz)
Vector specifying the frequency offset.

Phase noise level (dBc/Hz)
: Vector specifying the phase noise level.

Noise type
: Type of noise data. The value can be Noise figure, Spot noise data, Noise factor, or Noise temperature. This parameter is disabled if the data source contains noise data.

Noise figure (dB)
: Scalar ratio, in decibels, of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise figure.

Minimum noise figure (dB)
: Minimum scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Spot noise data.

Optimal reflection coefficient
: Optimal mixer source impedance. This parameter is enabled if Noise type is set to Spot noise data.

Equivalent normalized resistance
: Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if Noise type is set to Spot noise data.

Noise factor
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if Noise type is set to Noise factor.

Noise temperature (K)
: Equivalent temperature that produces the same amount of noise power as the amplifier. This parameter is enabled if Noise type is set to Noise temperature.

Nonlinearity Data Tab

IP3 type
: Type of third-order intercept point. The value can be IIP3 (input intercept point) or OIP3 (output intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm)
: Scalar power value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, Inf, if the IP3 value is unknown.

1 dB gain compression power (dBm)
: Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data

or 1 dB compression point data. Use the default value, Inf, if the 1 dB compression point is unknown.

Output saturation power (dBm)
: Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if **1 dB gain compression power (dBm)** is specified. Use the default value, Inf, if saturation power is unknown.

Visualization Tab

For information about plotting the mixer parameters, see Chapter 3, “Plotting Model Data”.

See Also

General Mixer, Output Port, S-Parameters Mixer, Y-Parameters Mixer

Purpose

Model passive network using Z-parameters

Library

Black Box Elements sublibrary of the Physical library

Description

The Z-Parameters Passive Network block models the two-port passive network described in the block dialog box, in terms of its Z-parameters and their associated frequencies.

In the Z-Parameters field of the block dialog box, provide the Z-parameters for each of M frequencies as a 2-by-2-by-M array. In the Frequency field, specify the frequencies for the Z-parameters as an M-element vector. The elements of the vector must be in the same order as the Z-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Z-parameters array and the vector of frequencies.

The Z-Parameters Passive Network block uses the RF Toolbox™ `z2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See Appendix A, “RF Blockset™ Algorithms” for more details.

Dialog Box

Main Tab

Z-Parameters
: Z-parameters for a two-port passive network in a 2-by-2-by-M array. M is the number of Z-parameters.

Frequency (Hz)
: Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in **Z-Parameters**. All frequencies must be positive.

Interpolation method
: The method used to interpolate the network parameters. The following table lists the available methods describes each one.

Method	Description
Linear (default)	Linear interpolation

Method	Description
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization Tab

For information about plotting circuit parameters, see Chapter 3, “Plotting Model Data”.

Examples

Plotting Parameters with the Z-Parameters Passive Network Block

The following example specifies Z-parameters [0.13 - 5.93i, .03-3.16i; 0.03-3.16i, .13-5.93i] and [0.27-2.86i, -.09-5.41i; -.09-5.41i, .27-2.86i] at frequencies 2.0 GHz and 2.1 GHz respectively. It uses the MATLAB® `cat` function to create the 2-by-2-by-2 Z-parameters array.

```
cat(3,[0.13-5.93i, .03-3.16i; 0.03-3.16i, .13-5.93i],...
      [0.27-2.86i,-.09-5.41i; -.09-5.41i, .27-2.86i])
```

1. Type the following command at the MATLAB prompt to create a variable called `zparams` that stores the values of the Z-parameters.

   ```
   zparams = cat(3,...
         [0.13-5.93i,.03-3.16i; 0.03-3.16i,.13-5.93i],...
         [0.27-2.86i,-.09-5.41i; -.09-5.41i,.27-2.86i])
   ```

2. Set the Z-Parameters Passive Network block parameters on the **Main** tab as follows:

   - Set the **Z-Parameters** parameter to `zparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

   Click **Apply**. This action applies the specified settings.

3 Set the Z-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Y parameter1** list, select `S12`.

Click **Plot**. This action creates an X-Y plane plot of the S_{12} parameters in the frequency range 1.9 to 2.2 GHz.

See Also

General Circuit Element, General Passive Network, Output Port, S-Parameters Passive Network, Y-Parameters Passive Network

z2s (RF Toolbox)

interp1 (MATLAB)

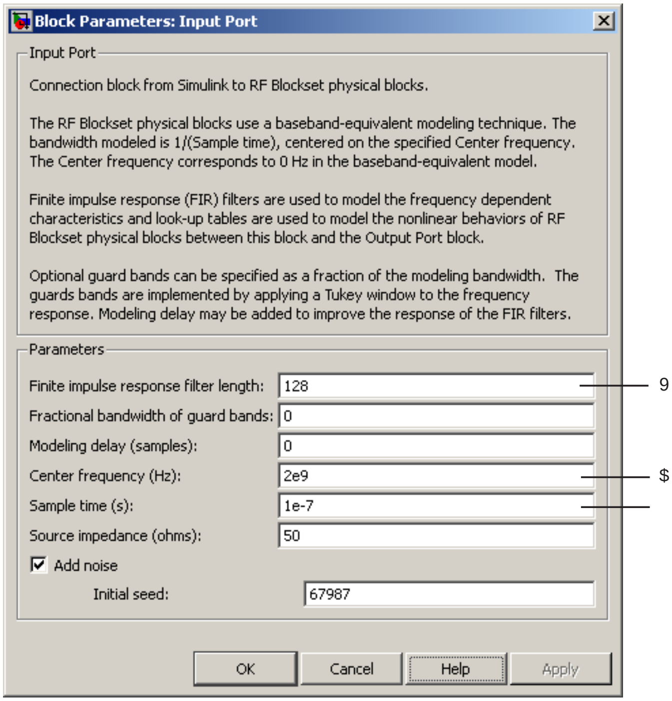
RF Blockset™ Algorithms

Simulating an RF Model

When you simulate a model that contains physical blocks, RF Blockset™ software determines the modeling frequencies of the physical system using the Input Port block parameters. The modeling frequencies are the frequencies at which the blockset takes information from the blocks to construct the baseband-equivalent model. Then, the software determines the block parameter values at those frequencies and uses the information to create a baseband-equivalent model for Simulink® time-domain simulation.

Determining the Modeling Frequencies

When you simulate an RF model, the Output Port block uses Input Port block parameters to determine the modeling frequencies f for the physical system that is bracketed between the Input Port block and the Output Port block. f is an N-element vector, where N is the finite impulse response filter length. The modeling frequencies are a function of the center frequency f_c and the sample time t_s. The following figure shows the Input Port block parameters that determine the modeling frequencies.



f_n is the nth element of the vector of modeling frequencies, f, and is given by

$$f_n = f_{min} + \frac{n - 1}{t_s N} \quad n = 1, \ldots, N$$

where

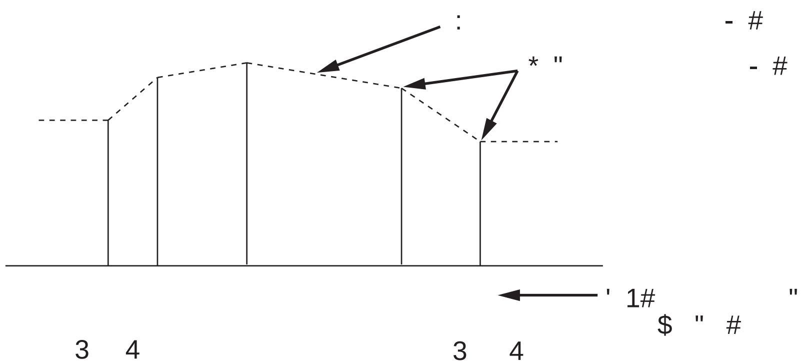
$$f_{min} = f_c - \frac{1}{2t_s}$$

Mapping Network Parameters to Modeling Frequencies

In a physical system, each block provides network parameters at different frequencies. These frequencies are not necessarily the modeling frequencies for the physical system in which the block resides. To create a baseband-equivalent model, RF Blockset™ software must calculate the values of the S-parameters at the modeling frequencies.

Individual physical blocks calculate the S-parameters at the modeling frequencies determined by the Input Port block parameters. Each block interpolates its S-parameters to determine the S-parameters at the modeling frequencies. If the block contains network Y- or Z-parameters, it first converts them to S-parameters.

Specifically, the block orders the S-parameters in the ascending order of their frequencies, f_n. Then, it interpolates the S-parameters using the MATLAB® `interp1` function. For example, the curve in the following diagram illustrates the result of interpolating the S_{11} parameters at original frequencies f_1 through f_5.



The **Interpolation method** field in the individual block dialog boxes enables you to specify the interpolation method as `Cubic`, `Linear` (default), or `Spline`. For more information about these methods, see the `interp1` reference page in the MATLAB documentation.

As shown in the previous diagram, each block uses the parameter values at f_{min} for all modeling frequencies smaller than f_{min}. The block uses the parameter values at f_{max} for all modeling frequencies greater than f_{max}. In both cases, the results may not be accurate, so you need to specify network parameter values over a range of frequencies that is wide enough to account for the block behavior.

Modeling Noise in an RF System

The RF Blockset™ physical blocks can model noise. The Input Port block parameters specify whether to include noise in a simulation. When you include noise information in your model, the blockset simulates the noise of the physical system by combining the noise contributions from each individual block. This section explains how the blockset simulates noise from user-specified information. For information on how to add noise to an RF model, see “Modeling Noise” on page 2-29.

In this section...

Output-Referred Noise in RF Models

In general, you can specify output-referred noise in one of three ways:

- Noise temperature — Specifies the noise in kelvin.
- Noise factor — Specifies the noise by the following equation:

$$\text{Noise factor} = 1 + \frac{\text{Noise temperature}}{290}$$

- Noise figure — Specifies the noise in decibels relative to the standard reference noise temperature of 290 K. In terms of noise factor

$$\text{Noise figure} = 10\log(\text{Noise factor})$$

These three specifications are equivalent, because you can compute each one from any of the others.

The blockset lets you simulate the noise associated with any physical block in your RF model.

The blockset automatically determines the noise properties of passive blocks from their network parameters. The blockset gets these network parameters either explicitly from the block dialog box or specified data files, or implicitly by calculating them from the specified block parameters.

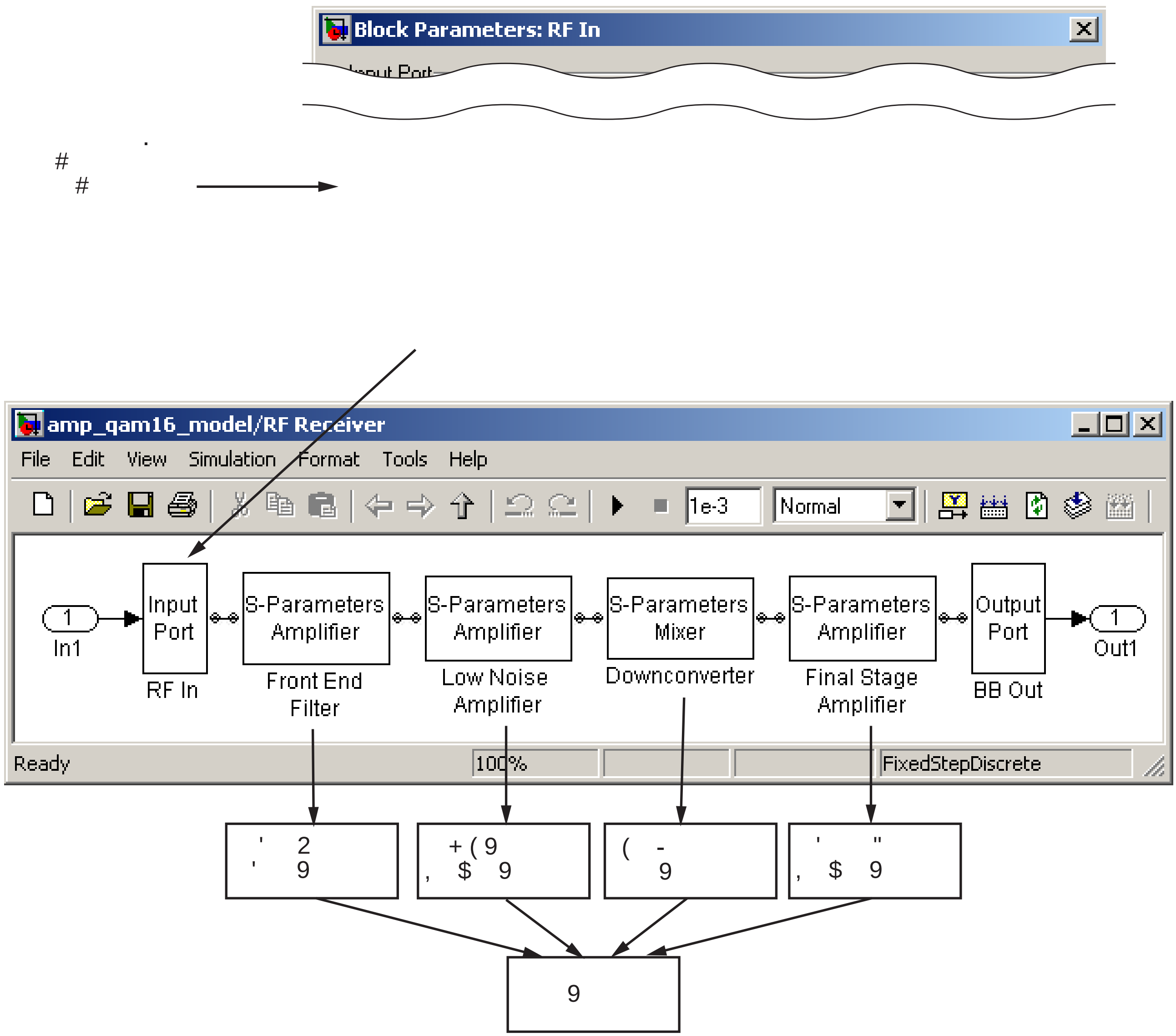
For active devices such as amplifiers and mixers, the noise properties can not be inferred from network parameters. Therefore, for the amplifier and mixer blocks, you must specify the noise information explicitly, either in the dialog block or the associated data file.

For RF Blockset physical amplifier and mixer blocks, you can specify active block noise in one of the following ways:

- Spot noise data
- Frequency-independent noise figure, noise factor, or noise temperature values
- Frequency-dependent noise figure data (rfdata.nf) or spot noise data (rfdata.noise) object

These noise specification options are described in “Amplifier and Mixer Noise Specifications” on page 2-29.

When you run the simulation, the blockset first computes the noise figure values for each individual block at the modeling frequencies. Then, it computes the noise figure of the physical system from the individual noise figure values and uses the system noise figure information to calculate the output noise power. This process is shown in the following figure.


Block Parameters: RF In
amp_qam16_model/RF Receiver
File Edit View Simulation Format Tools Help
1e-3
Normal
1
In1
Input Port
RF In
S-Parameters Amplifier
Front End Filter
S-Parameters Amplifier
Low Noise Amplifier
S-Parameters Mixer
Downconverter
S-Parameters Amplifier
Final Stage Amplifier
Output Port
BB Out
1
Out1
Ready
100%
FixedStepDiscrete

Calculating Noise Figure at Modeling Frequencies

To include noise information in a simulation, the blockset must compute the noise figure values of each individual block at the modeling frequencies.

If you specify the frequency-independent noise figure value directly, or if the blockset computes the noise figure value from the block resistance, the blockset uses this value for the noise figure value at each of the modeling frequencies.

If you specify the noise factor or noise temperature value, the blockset computes the noise figure value from the specified value using the equations in the preceding section and uses the computed value for the noise figure value at each of the modeling frequencies.

If you specify frequency-dependent noise figure values using an `rfdata.nf` object, the blockset interpolates the values using the **Interpolation method** specified in the block dialog box to get the noise figure value at each of the modeling frequencies.

If you specify spot noise data, the blockset computes frequency-dependent noise figure information from this data. It takes the minimum noise figure, NF_{min}, equivalent noise resistance, R_n, and optimal source admittance, Y_{opt}, values in the file and interpolates to find the values at the modeling frequencies. Then, the blockset uses the following equation to calculate the noise correlation matrix, C_A:

$$C_A = 2kT \begin{bmatrix} R_n & \frac{NF_{min} - 1}{2} - R_n Y_{opt} \\ \frac{NF_{min} - 1}{2} - R_n Y_{opt} & R_n \left|Y_{opt}\right|^2 \end{bmatrix}$$

where k is Boltzmann's constant, and T is the noise temperature in kelvin.

The blockset then calculates the noise factor, F, from the noise correlation matrix as follows:

$$F = 1 + \frac{z^{+} C_A z}{2kT\,\mathrm{Re}\{Z_S\}}$$

$$z = \frac{1}{Z_S^{*}}$$

In the two preceding equations, Z_S is the nominal impedance, which is 50 ohms, and z^{+} is the Hermitian conjugation of z.

The blockset obtains the noise figure, NF, from the noise factor:

$$NF = 10\log(F)$$

For more information about these calculation techniques, see the following article:

Hillbrand, H. and P.H. Russer, “An Efficient Method for Computer Aided Noise Analysis of Linear Amplifier Networks,” IEEE Transactions on Circuits and Systems, Vol. CAS-23, Number 4, pp. 235–238, 1976.

Calculating System Noise Figure

After the blockset computes the noise figure values for each individual block at the modeling frequencies, it calculates the noise figure for the N-element system using the following equation:

where G_n is the gain of the *n*th element of the system and NF_n is the noise figure of the *n*th element.

Calculating Output Noise Power

The blockset uses noise power to determine the amplitude of the noise that it adds to the physical system using a Gaussian distributed pseudorandom number generator. It uses both the noise temperature and the modeling bandwidth to calculate the noise power:

Noise power = kTB

where k is Boltzmann’s constant, which is 1.38e-23 J/K, T is the noise temperature in kelvin, and B is the bandwidth in hertz.

The blockset computes noise temperature from the specified or calculated noise figure values for the system, and it computes the modeling bandwidth from the model’s sample time and center frequency.

Creating a Complex Baseband-Equivalent Model

In this section...

Baseband-Equivalent Modeling

RF Blockset™ software simulates the physical system in the time domain using a complex baseband-equivalent model that it creates from the passband frequency-domain parameters of the physical blocks. This type of modeling is also known as lowpass equivalent (LPE), complex envelope, or envelope modeling.

To create a complex baseband-equivalent model in the time domain based on the network parameters of the physical system, the blockset performs a mathematical transformation that consists of the following three steps:

1. "Calculating the Passband Transfer Function" on page A-13
2. "Calculating the Baseband-Equivalent Transfer Function" on page A-16
3. "Calculating the Baseband-Equivalent Impulse Response" on page A-16

Calculating the Passband Transfer Function

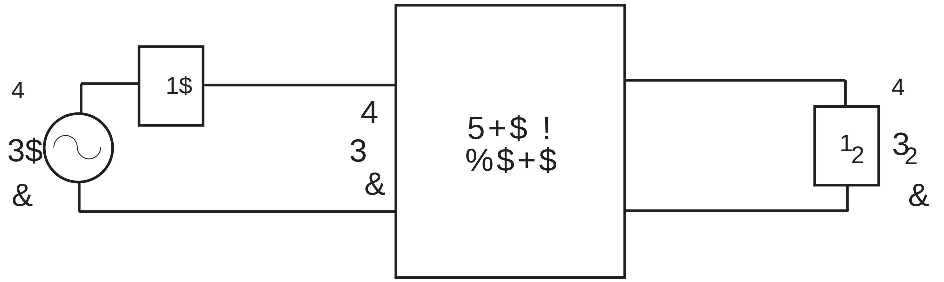
The blockset calculates the passband transfer function from the physical block parameters at the modeling frequencies by calculating the transfer function of the physical subsystem and then applying the Tukey window to obtain the passband transfer function.

Note To learn how the blockset uses the specified network parameters to compute the network parameters at the modeling frequencies, see "Mapping Network Parameters to Modeling Frequencies" on page A-5.

The transfer function of the physical subsystem is defined as

$$H(f) = \frac{V_L(f)}{V_S(f)}$$

where V_S and V_L are the source and load voltages shown in the following figure, and f represents the modeling frequencies.



More specifically,

$$H(f) = \frac{S_{21} * (1 + \Gamma_l) * (1 - \Gamma_s)}{2 * (1 - S_{22} * \Gamma_l)(1 - \Gamma_{in} * \Gamma_s)}$$

where

$$\Gamma_l = \frac{Z_l - Z_o}{Z_l + Z_o}$$

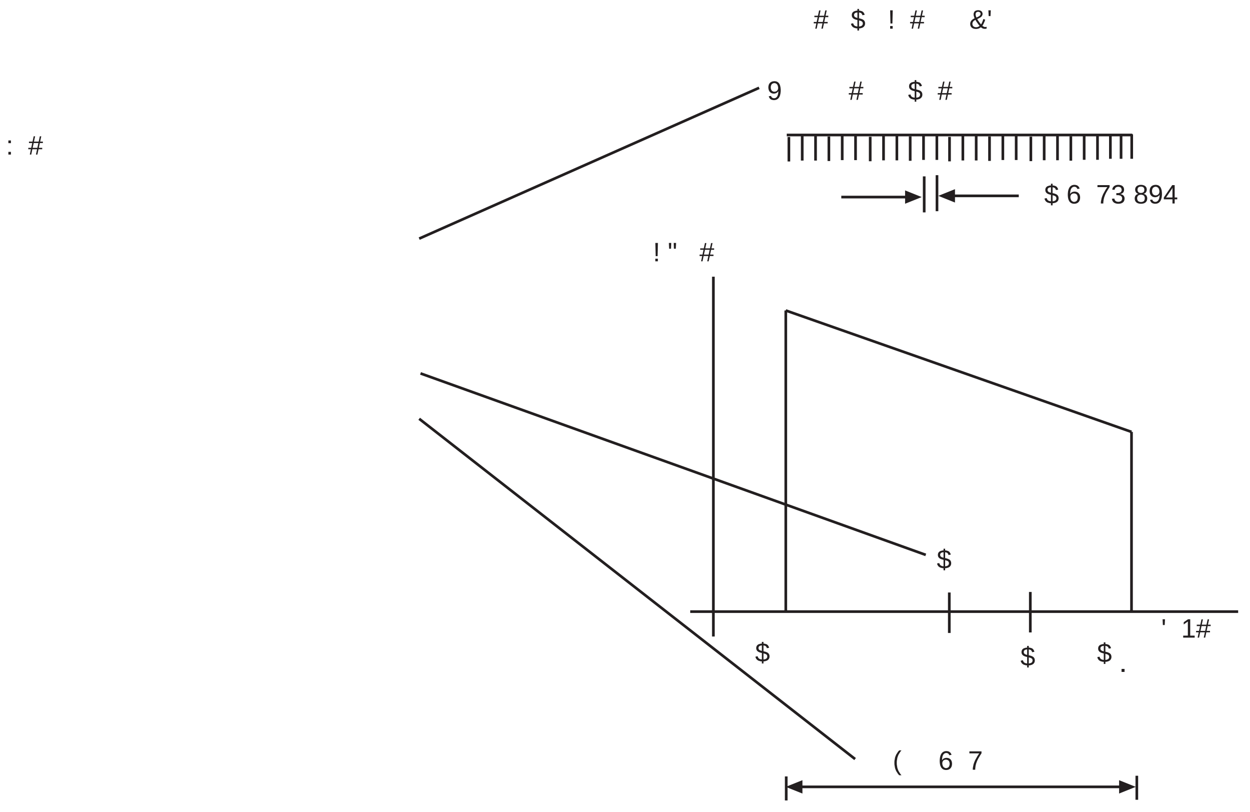
$$\Gamma_s = \frac{Z_s - Z_o}{Z_s + Z_o}$$

$$\Gamma_{in} = S_{11} + S_{12} * S_{21} * \frac{\Gamma_l}{(1 - S_{22} * \Gamma_l)}$$

and

- Z_S is the source impedance.
- Z_L is the load impedance.
- S_{ij} are the S-parameters of a two-port network.

The blockset derives the transfer function of the physical subsystem from the Input Port block parameters as shown in the following figure.



The blockset then applies the Tukey window to obtain the passband transfer function:

$$H_{passband}(f) = H(f) * \text{tukeywin}(N, F)$$

where `tukeywin` is the Signal Processing Toolbox™ `tukeywin` function.

Calculating the Baseband-Equivalent Transfer Function

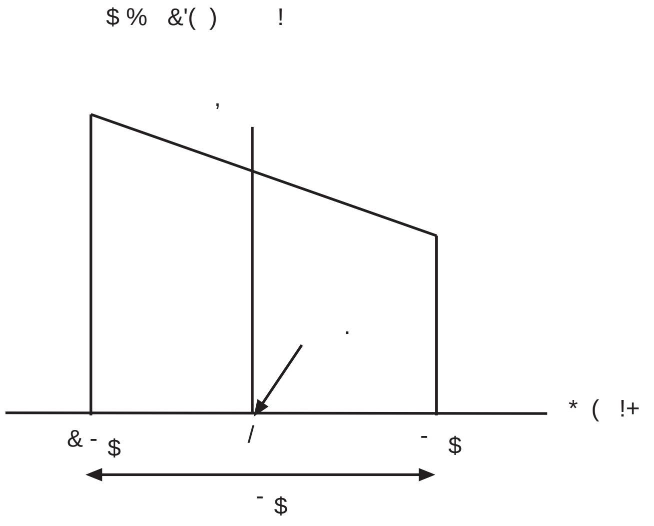
The blockset calculates the baseband transfer function, $H_{baseband}(f)$, by translating the passband transfer function to its equivalent baseband transfer function:

$$H_{baseband}(f) = H_{passband}(f + f_c)$$

where f_c is the specified center frequency.

The resulting baseband-equivalent spectrum is centered at zero, so the blockset can simulate the system using a much larger time step than Simulink® can use for the same system. For information on why this translation allows for a larger time step, see “Simulation Efficiency of a Baseband-Equivalent Model” on page A-18.

The baseband transfer function is shown in the following figure.

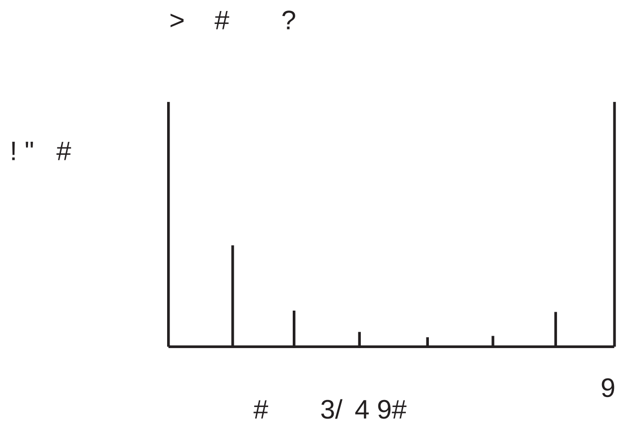


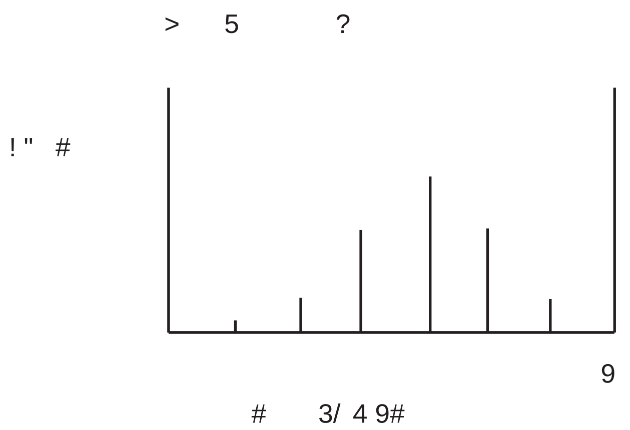
Calculating the Baseband-Equivalent Impulse Response

The blockset calculates the baseband-equivalent impulse response by performing the following steps:

1 Calculate the inverse FFT of the baseband transfer function. For faster simulation, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. Then, it truncates the impulse response to a length equal to the filter length specified. When the finite impulse response is truncated to the length specified by the user, the effect of the truncation is similar to windowing with a rectangular window.

2 Apply the delay specified by the **Modeling delay (samples)** parameter in the Input Port block dialog box. Selecting an appropriate value for this delay ensures that the baseband-equivalent model has a causal response by moving the time window such that the model energy is concentrated at the center of the window, as shown in the following figure:





Simulation Efficiency of a Baseband-Equivalent Model

The baseband-equivalent modeling technique improves simulation speed by allowing the simulator to take larger time steps. To simulate a system in the time domain, Simulink would require a step size of:

$$t_{step} = \frac{1}{2 f_{max}}$$

Using the baseband-equivalent model of the same system, whose spectrum has been shifted down by f_c, allows for a much larger time step of: