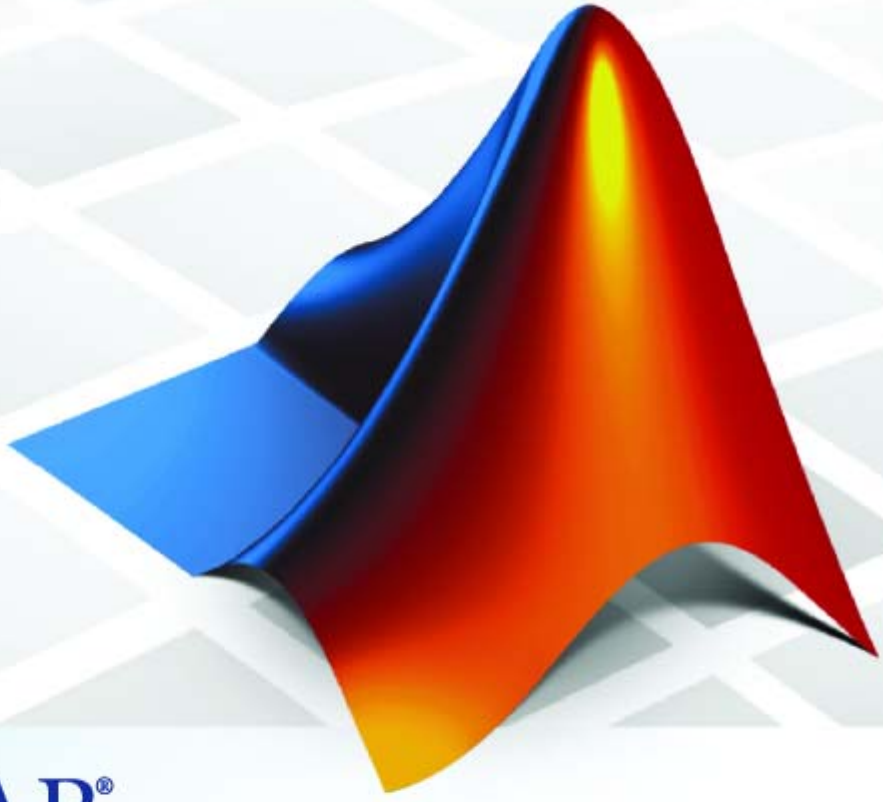
# RF Blockset 2
User's Guide



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*

© COPYRIGHT 2004–2007 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

#### Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks, and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

#### Patents

The MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

**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) |

# Contents

## 1 Getting Started


**What Is RF Blockset?** . . . . . 1-2

**Required and Related Products** . . . . . 1-3

**Product Demos** . . . . . 1-4

**RF Blockset Libraries** . . . . . 1-6
- Opening RF Blockset Libraries . . . . . 1-6
- Physical Library . . . . . 1-7
- Mathematical Library . . . . . 1-9

**RF Blockset Workflow** . . . . . 1-10

**Example — Modeling an LC Bandpass Filter** . . . . . 1-12
- Selecting Blocks to Represent System Components . . . . . 1-12
- Building the Model . . . . . 1-13
- Specifying Model Parameters . . . . . 1-15
- Validating Filter Components and Running the Simulation . . . . . 1-22
- Analyzing the Simulation Results . . . . . 1-24


## 2 Modeling an RF System


**Modeling RF Components** . . . . . 2-2
- Adding RF Blocks to a Simulink Model . . . . . 2-2
- Connecting Model Blocks . . . . . 2-2

**Specifying or Importing Component Data** . . . . . 2-6

Specifying Parameter Values . . . . . 2-7
Importing Data Files into RF Blocks . . . . . 2-7
Example — Importing a Touchstone Data File into an RF Model . . . . . 2-9
Importing Circuits from the MATLAB Workspace . . . . . 2-15
Example — Importing a Bandstop Filter into an RF Model . . . . . 2-16

**Specifying Operating Conditions** . . . . . 2-23

**Modeling Nonlinearity** . . . . . 2-25
Amplifier and Mixer Nonlinearity Specifications . . . . . 2-25
Adding Nonlinearity to Your System . . . . . 2-26

**Modeling Noise** . . . . . 2-28
Amplifier and Mixer Noise Specifications . . . . . 2-28
Adding Noise to Your System . . . . . 2-29
Plotting Noise . . . . . 2-32

## 3 Plotting Model Data

**Creating Plots** . . . . . 3-2
Validating Individual Blocks and Subsystems . . . . . 3-2
Types of Plots . . . . . 3-3
Plot Formats . . . . . 3-4
How to Create a Plot . . . . . 3-13
Example — Plotting Component Data on a Z Smith Chart . . . . . 3-20

**Updating Plots** . . . . . 3-25

**Modifying Plots** . . . . . 3-26

**Example — Creating and Modifying Subsystem Plots** . . . . . 3-29
Plotting the Network Parameters of a Subsystem . . . . . 3-29
Adding Data to an Existing Plot . . . . . 3-31

Changing Data on an Existing Plot . . . . . 3-33

## 4 Blocks — By Category

**Mathematical** . . . . . 4-2

**Physical** . . . . . 4-3
Ladder Filters . . . . . 4-3
Transmission Lines . . . . . 4-3
Black Box Elements . . . . . 4-4
Amplifiers . . . . . 4-4
Mixers . . . . . 4-5
Input/Output Ports . . . . . 4-5

## 5 Blocks — Alphabetical List

## A RF Blockset Algorithms

**Simulating an RF Model** . . . . . A-2

**Determining the Modeling Frequencies** . . . . . A-3

**Mapping Network Parameters to Modeling Frequencies** . . . . . A-4

**Modeling Noise in an RF System** . . . . . A-6
Output-Referred Noise in RF Models . . . . . A-6
Calculating Noise Figure at Modeling Frequencies . . . . . A-9
Calculating System Noise Figure . . . . . A-10

Calculating Output Noise Power . . . . . A-10

**Creating a Complex Baseband-Equivalent Model** . . . . . A-12
Baseband-Equivalent Modeling . . . . . A-12
Simulation Efficiency of a Complex Baseband-Equivalent Model . . . . . A-15

**Converting to and from Simulink Signals** . . . . . A-17

## B Modeling Mixers

**2-Port Mixer Blocks** . . . . . B-2

**Modeling a Mixer Chain** . . . . . B-4

**Quadrature Mixers** . . . . . B-6
Modeling Upconversion I/Q Mixers . . . . . B-6
Modeling Downconversion I/Q Mixers . . . . . B-7
Simulating I/Q Mixers . . . . . B-8

## C Examples

**Examples** . . . . . C-2

## Index

1

# Getting Started


| | |
|---|---|
| What Is RF Blockset? (p. 1-2) | Introduces RF Blockset and describes its capabilities |
| Required and Related Products (p. 1-3) | Describes products you can use to extend the capabilities of RF Blockset |
| Product Demos (p. 1-4) | Describes how to access the RF Blockset demos in the Help browser |
| RF Blockset Libraries (p. 1-6) | Summarizes the RF Blockset libraries |
| RF Blockset Workflow (p. 1-10) | Describes a typical workflow |
| Example — Modeling an LC Bandpass Filter (p. 1-12) | Describes how to build an RF model, modify the parameters, run the model in Simulink, and examine the results |

# What Is RF Blockset?

RF Blockset extends Simulink® with a library of blocks for modeling RF systems that include RF filters, transmission lines, amplifiers, and mixers. For more information about creating and running Simulink models, see "Working with a Simple Model" in the Simulink User's Guide.

You use blocks from RF Blockset 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 and enables compatibility with other Simulink blocks.

RF 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 for wireless communication systems.

# Required and Related Products

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

- 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.

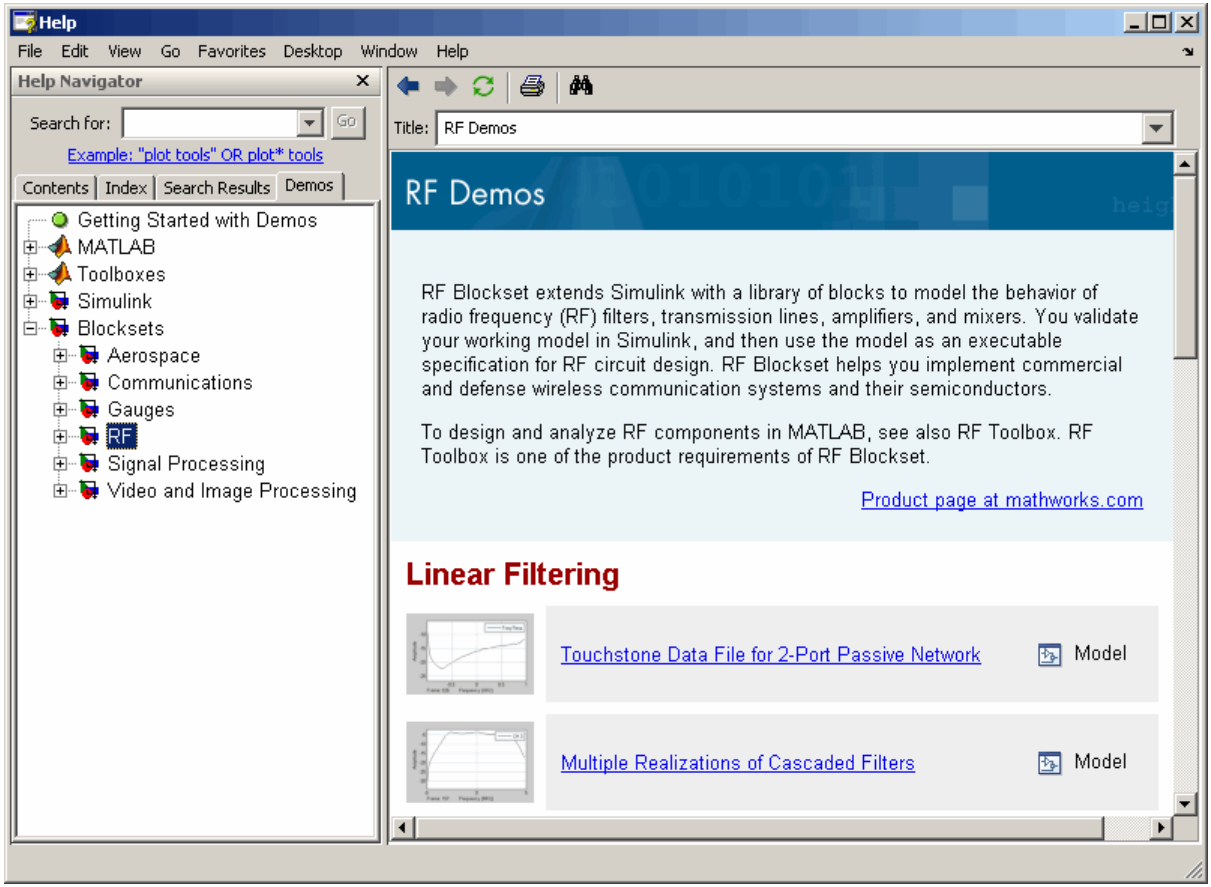
You can build sophisticated wireless communication system models by incorporating blocks from other blocksets, such as Signal Processing Blockset and Communications Blockset.

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

| Product | Description |
|---|---|
| Communications BlocksetCommunications 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

RF Blockset 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 MathWorks blocksets, such as those described in “Required and Related Products” on page 1-3.

This section contains the following topics:

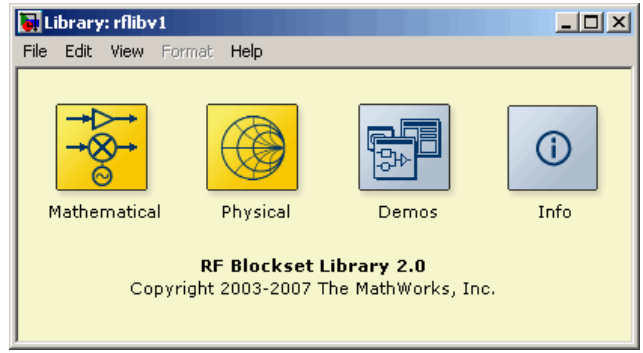

- “Opening RF Blockset Libraries” on page 1-6
- “Physical Library” on page 1-7
- “Mathematical Library” on page 1-9


## 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.



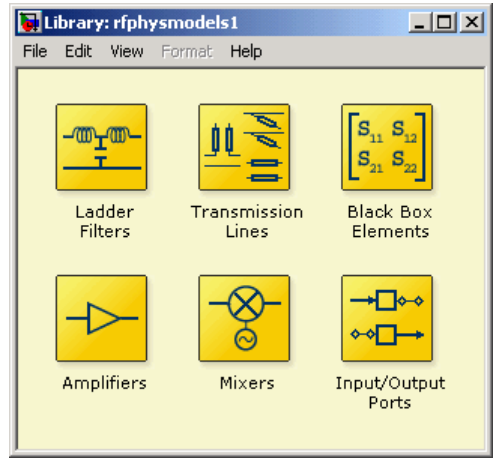

The Physical and Mathematical libraries are discussed in the following sections.

**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 amplifiers, specified using network parameters, noise data, and nonlinearity data, or a data file containing these parameters. |

| Sublibrary | Description |
|---|---|
| Ladder Filters | RF filters, specified using RLC parameters. RF Blockset calculates the network parameters and noise data of the blocks from the topology of the filter 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. RF Blockset calculates the network parameters and noise data of the blocks from the specified data. |
| Black Box | Passive RF components, specified using network parameters, or a data file containing these parameters. RF Blockset 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.<br><br>**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. |

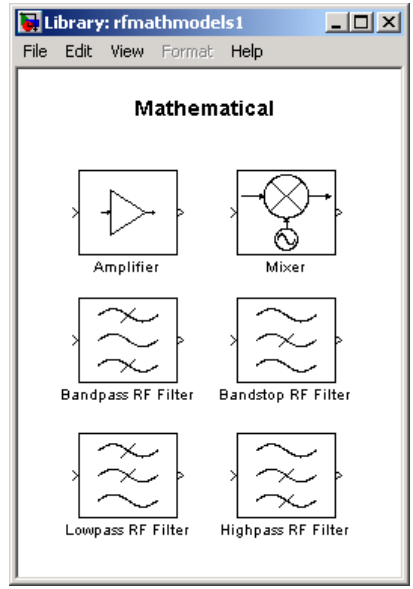
For more information on defining components, see “Specifying or Importing Component Data” on page 2-6.

## 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 shown in the following figure.



# RF Blockset Workflow

When you analyze an RF system using RF Blockset, 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

     RF Blockset 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 2-6.

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

   - Operating condition values (see “Specifying Operating Conditions” on page 2-23).
   - Nonlinearity data (see “Modeling Nonlinearity” on page 2-25).
   - Noise data (see “Modeling Noise” on page 2-28).

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 RF Blockset 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 “Creating Plots” on page 3-2.

The following plots and charts are available in RF Blockset:

- Rectangular plots
- Polar plots
- Smith charts
- Composite plots
- Budget plots

# Example — Modeling an LC Bandpass Filter

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.

This example illustrates how to perform the following tasks:


- "Selecting Blocks to Represent System Components" on page 1-12
- "Building the Model" on page 1-13
- "Specifying Model Parameters" on page 1-15
- "Validating Filter Components and Running the Simulation" on page 1-22
- "Analyzing the Simulation Results" on page 1-24


## 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 physical 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 attenuation 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 Simulink model.

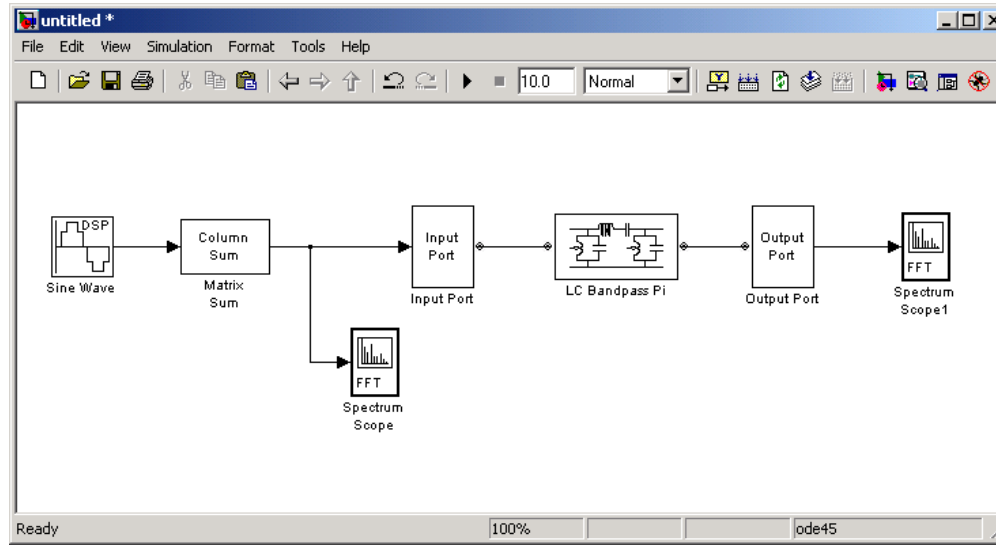
   If you are new to Simulink, see the introductory Simulink example, “Working with a Simple Model”, for information on how to create a Simulink model.

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 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 2-2.




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 1-15
- “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 algorithm used by RF Blockset requires you to shift the frequencies of the input signal. RF Blockset simulates the filter 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-baseband modeling, see "Creating a Complex Baseband-Equivalent Model" on page A-12.

---

**Note** RF Blockset requires that all signals in the RF model be complex to match the signals in the physical subsystem, so you create a complex input signal.

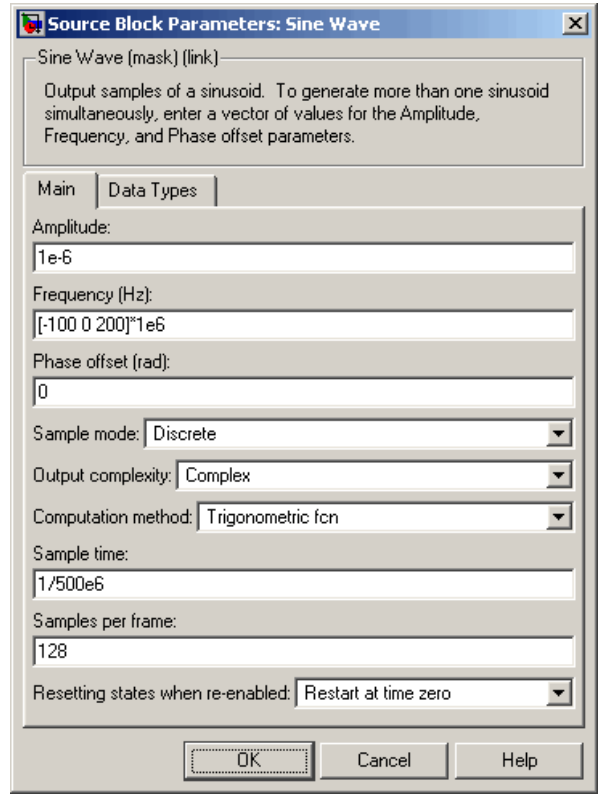
---

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

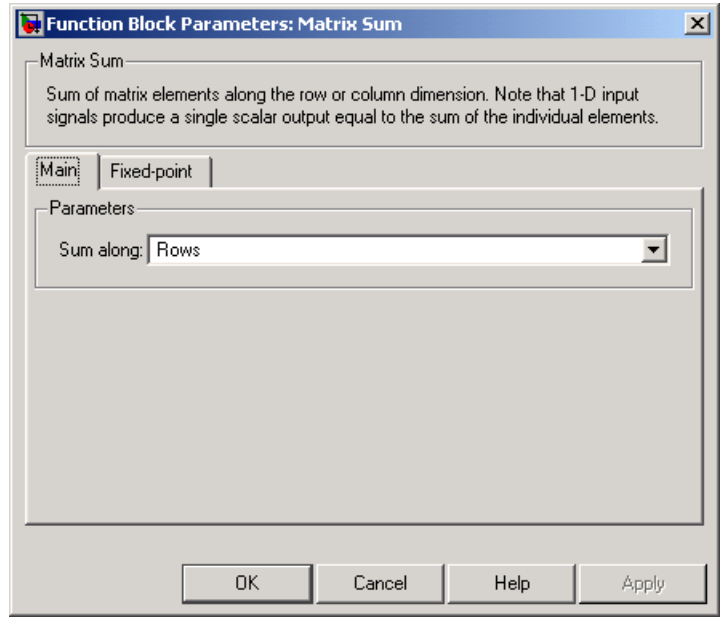
1 Set the Sine Wave block parameters as follows:

- **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`.



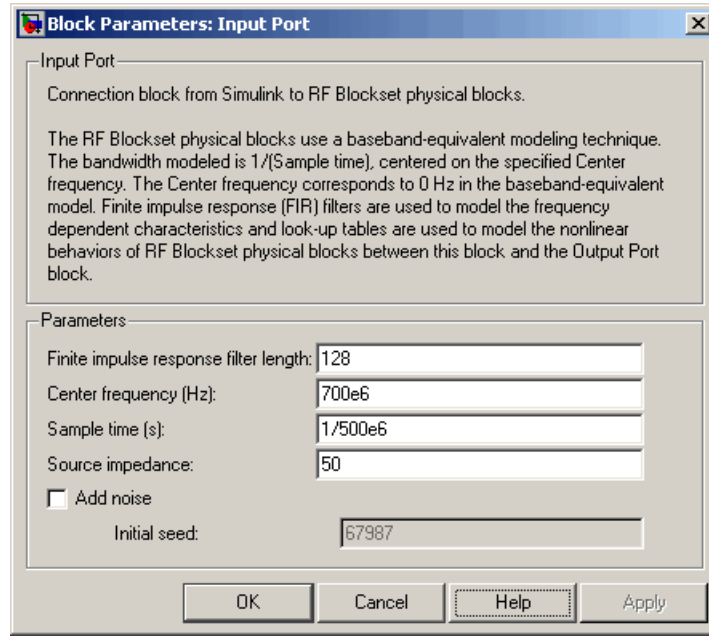

## Filter Subsystem Parameters

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:

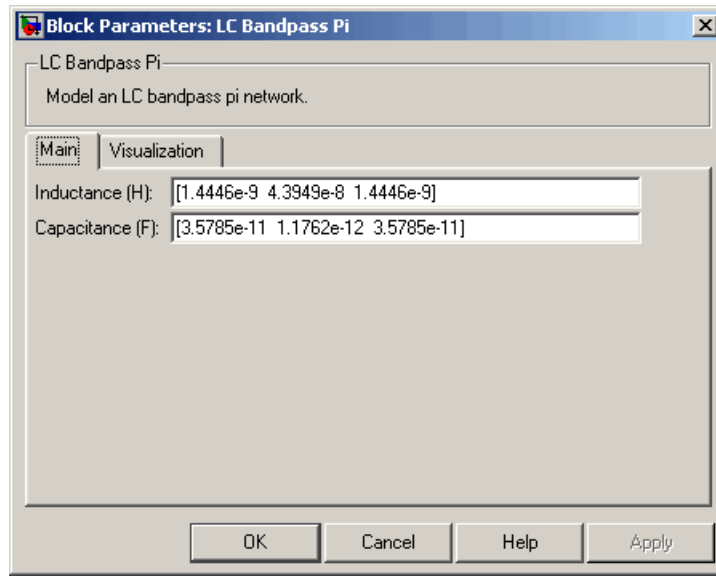
- **Center frequency** = `700e6`
- **Sample time(s)** = `1/500e6`

- Clear the **Add noise** check box so RF Blockset will not include noise in the simulation. To learn how to model noise, see “Modeling Noise” on page 2-28.

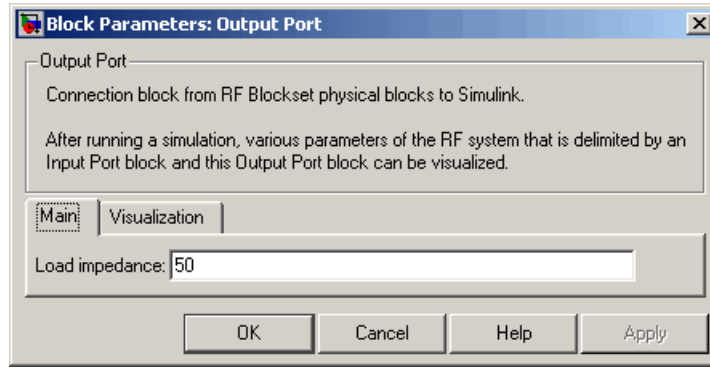



**Note** You must enter the **Sample time (s)** because the Input Port block does not inherit a sample time from the input signal. The specified sample time must match the sample time of the input signal. The **Sample time (s)** of `1/500e6` second used in this example is equivalent to a bandwidth of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

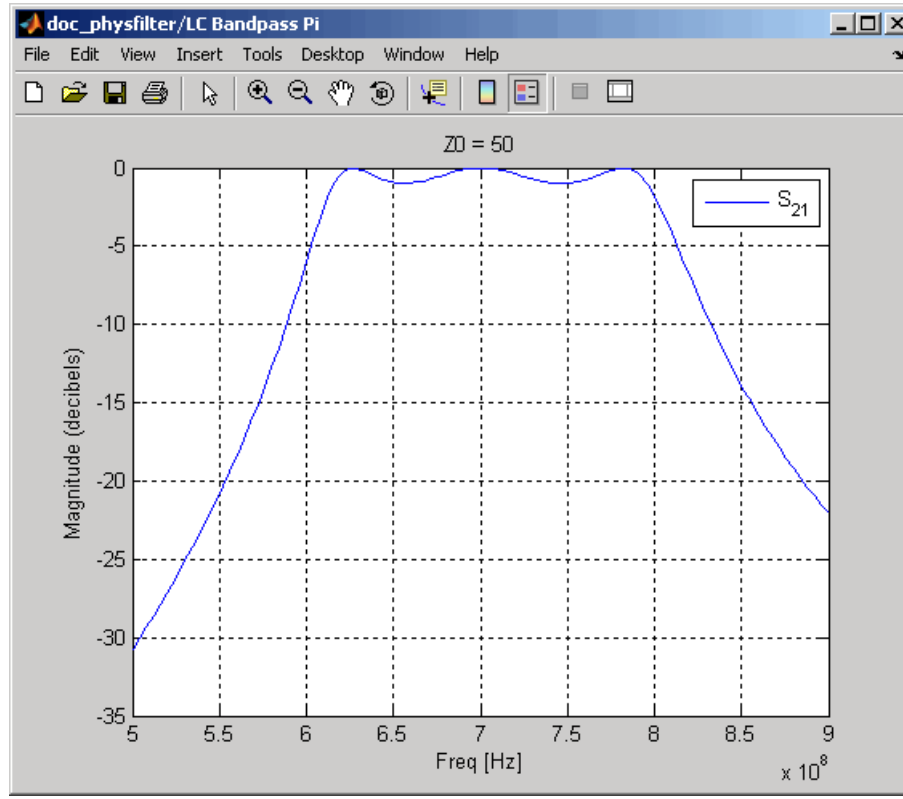
---

**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 1-15.

---

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 Simulink 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 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`).

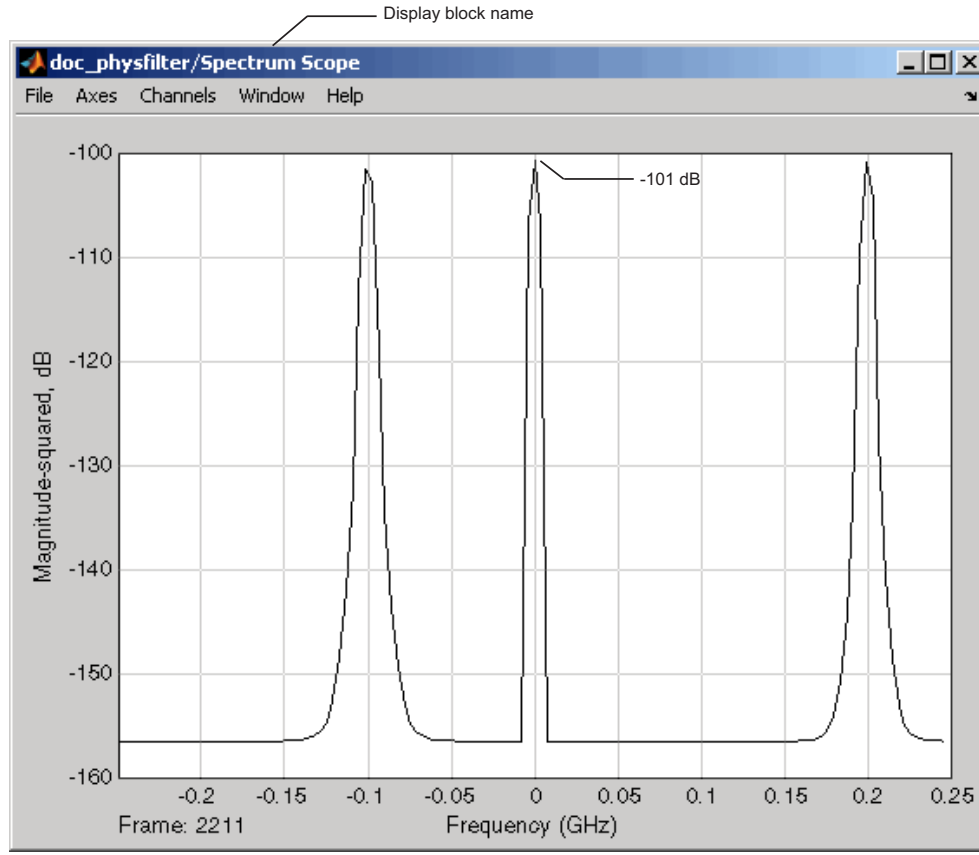
---

**Note** Simulink signals represent amplitudes, not voltages. This means that in Simulink, power is defined as:

$$Power(in\,watts) = [Amplitude(in\,volts / sqrt(ohm))]^2$$

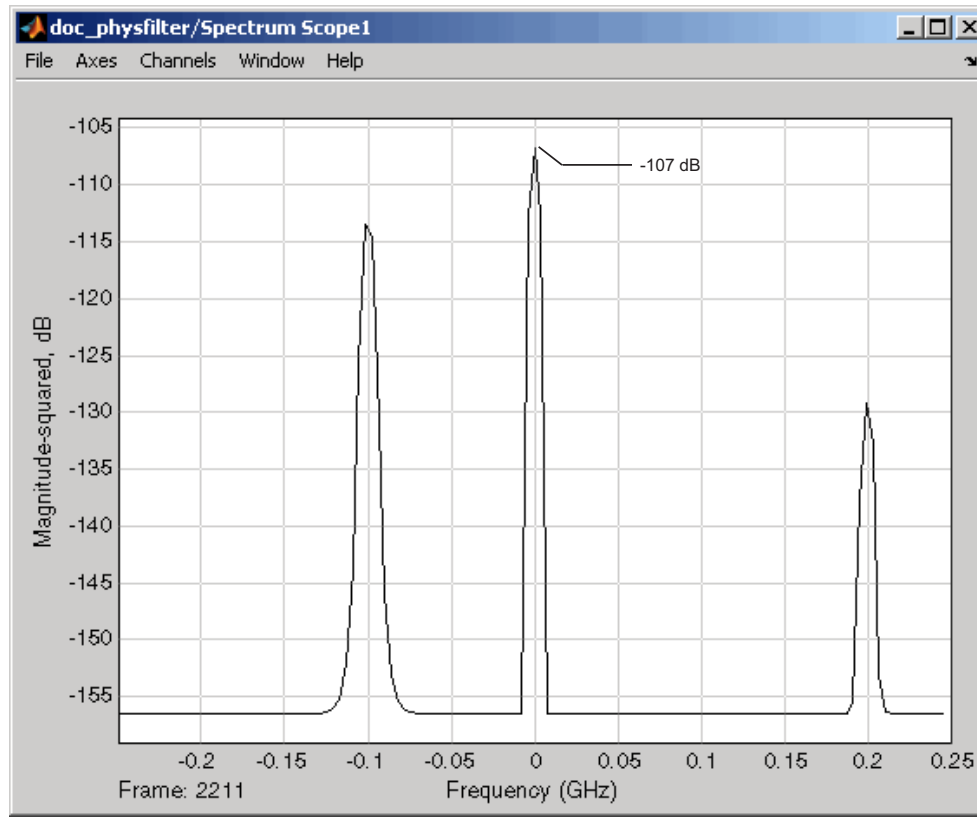
---

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-17.

## 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**.

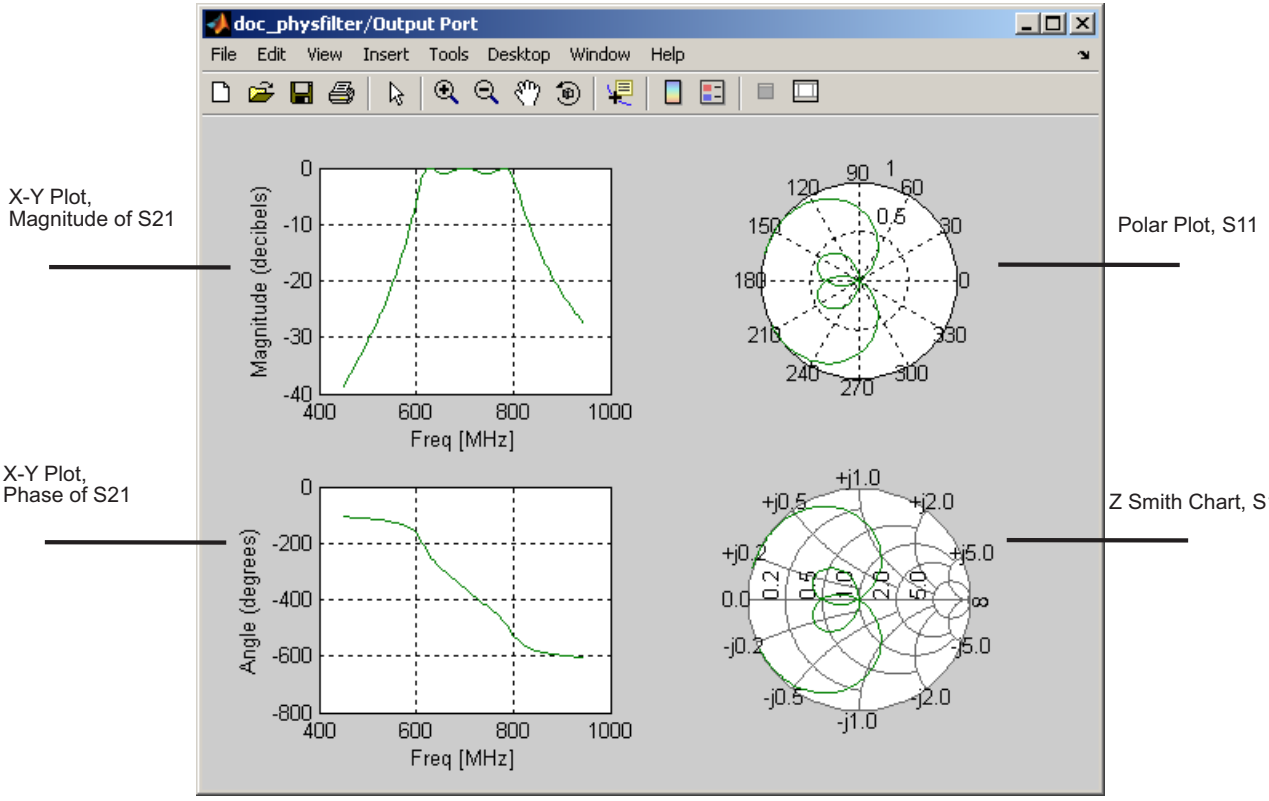
The composite plot, shown in the following figure, contains four separate plots in one figure. For the Output Port block, the composite plot shows the following as a function of frequency (counterclockwise from the upper-left plot):

- 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.

---


X-Y Plot,
Magnitude of S21
X-Y Plot,
Phase of S21
Polar Plot, S11
Z Smith Chart, S11
doc_physfilter/Output Port
File Edit View Insert Tools Desktop Window Help
Magnitude (decibels)
Freq [MHz]
Angle (degrees)
Freq [MHz]

2

# Modeling an RF System


| | |
|---|---|
| Modeling RF Components (p. 2-2) | Describes how to add and connect blocks in a Simulink model to represent RF components |
| Specifying or Importing Component Data (p. 2-6) | Describes how to define block behavior by entering parameter values and importing file and workspace data |
| Specifying Operating Conditions (p. 2-23) | Explains how to set and retrieve operating condition values. |
| Modeling Nonlinearity (p. 2-25) | Explains how to specify amplifier and mixer nonlinearity in a physical system |
| Modeling Noise (p. 2-28) | Explains how to model noise in a physical system |

# Modeling RF Components

RF Blockset lets you model an RF system using Simulink. This section describes how to build a system model that incorporates RF components and contains the following topics:


- "Adding RF Blocks to a Simulink Model" on page 2-2
- "Connecting Model Blocks" on page 2-2


## Adding RF Blocks to a Simulink 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.

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 Simulink blocks, see "Connecting the Blocks" 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.

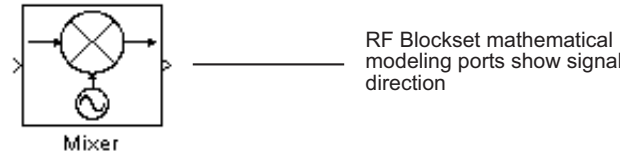
For more information on the RF Blockset libraries, including how to open the libraries and a description of the available blocks, see “RF Blockset Libraries” on page 1-6.

This section contains the following topics:


- “Connecting Mathematical Blocks” on page 2-3
- “Connecting Physical Blocks” on page 2-4
- “Bridging Physical and Mathematical Blocks” on page 2-4


## 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.

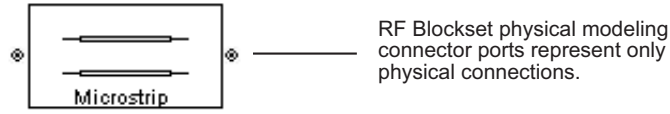



Similar to standard Simulink blocks, you draw lines between the ports of the Mathematical modeling blocks, called *signal lines*, to represent signals that are inputs to and outputs from the mathematical functions represented by the blocks. Therefore, you can connect Simulink, Signal Processing, and 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 Simulink blocks, see “Connecting the Blocks” 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.



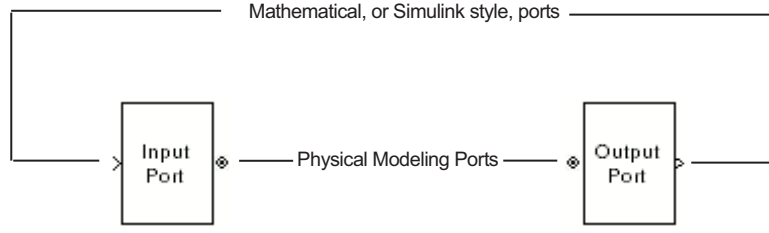

The lines you draw between the RF Blockset physical modeling blocks, called *connection lines*, represent physical connections among the block components. Connection lines appear as solid black when connected and as dashed red lines when either end is unconnected.

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

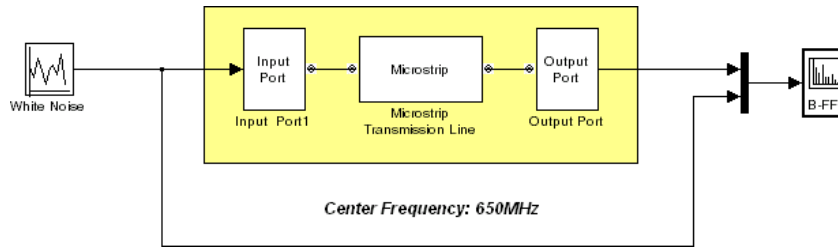
RF Blockset provides the Input Port and Output Port blocks to connect the physical and mathematical parts of the model. These blocks convert mathematical signals to and from the physical modeling environment.

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 converts mathematical signals to and from the physical modeling environment, see “Converting to and from Simulink Signals” on page A-17.

# Specifying or Importing Component Data

You can specify RF Blockset blocks by entering the RF component parameter values. RF Blockset also lets you import

- 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.

This section contains the following topics:

- “Specifying Parameter Values” on page 2-7
- “Importing Data Files into RF Blocks” on page 2-7

- "Example — Importing a Touchstone Data File into an RF Model" on page 2-9
- "Importing Circuits from the MATLAB Workspace" on page 2-15
- "Example — Importing a Bandstop Filter into an RF Model" on page 2-16


## 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.

## Importing Data Files into RF Blocks

RF 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 Simulink.

This section contains the following topics:


- "Blocks Used to Import Data" on page 2-7
- "How to Import Data Files" on page 2-8


### 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

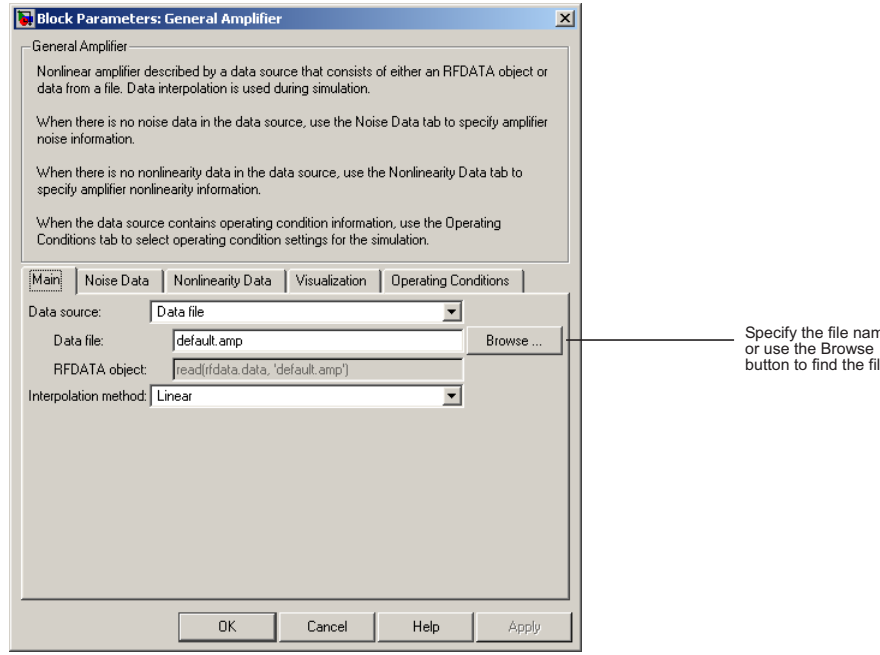
To import a data file:

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 2-7.
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 2-10
- “Validating the Passive Component” on page 2-12
- “Running the Simulation and Analyzing the Results” on page 2-14

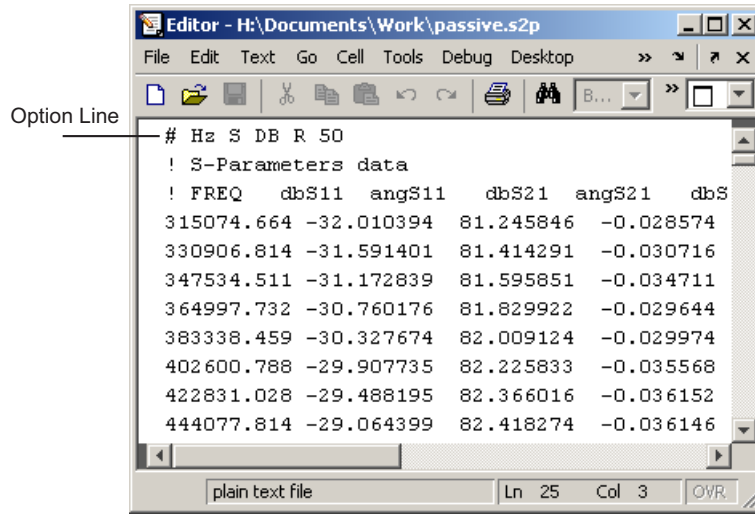
## Importing Data into a General Passive Network Block

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:

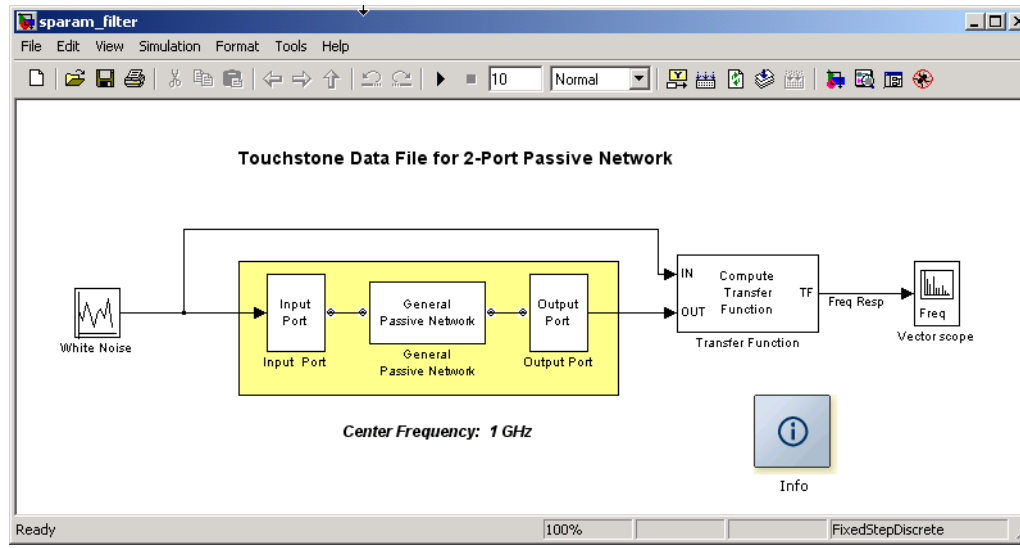
- Hz — Frequency units.
- S — Network parameters are S-parameters.
- DB — Network parameters are specified as magnitude in dB and phase in degrees.
- R 50 — Reference impedance is 50 ohms.

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.

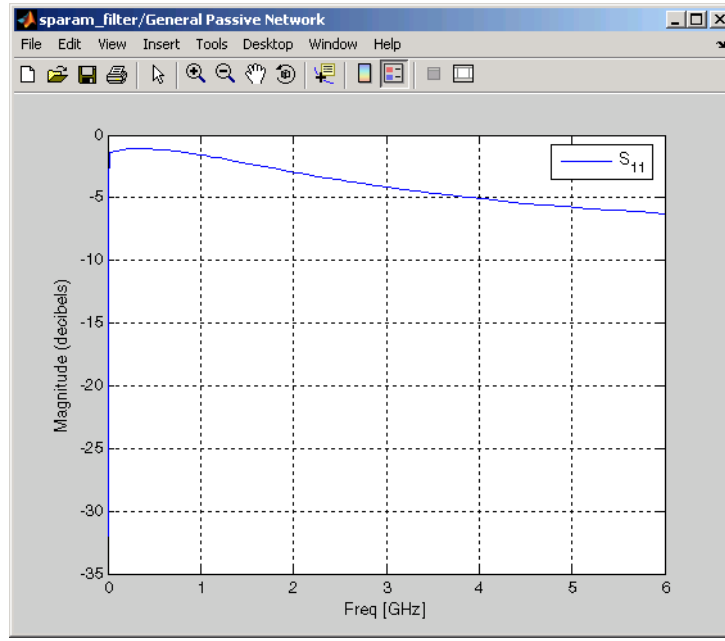
**Note** When the imported file contains data that is measured at frequencies other than the modeling frequencies, use the **Interpolation method** parameter to specify how the block determines the data values at the modeling frequencies. For more information, see "Determining the Modeling Frequencies" on page A-3 and "Mapping Network Parameters to Modeling Frequencies" on page A-4.

## 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 2-10.

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

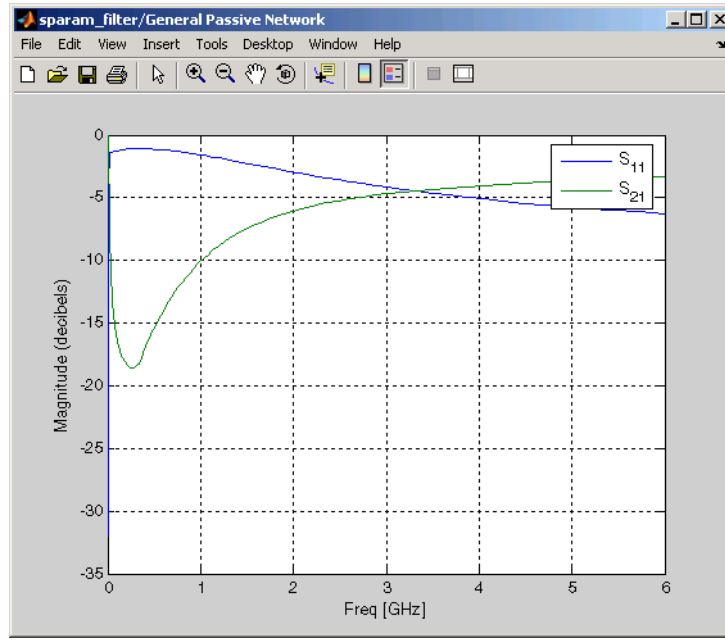
This action creates a plot of $S_{11}$ as a function of frequency.




**$S_{11}$ 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.

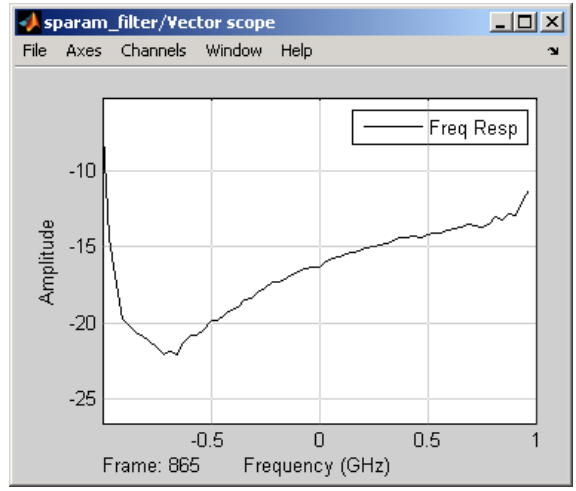



**$S_{11}$ and $S_{21}$ versus Frequency for the Imported Data**

## 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 Simulink 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-17.

- 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, RF Blockset works with RF Toolbox 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.

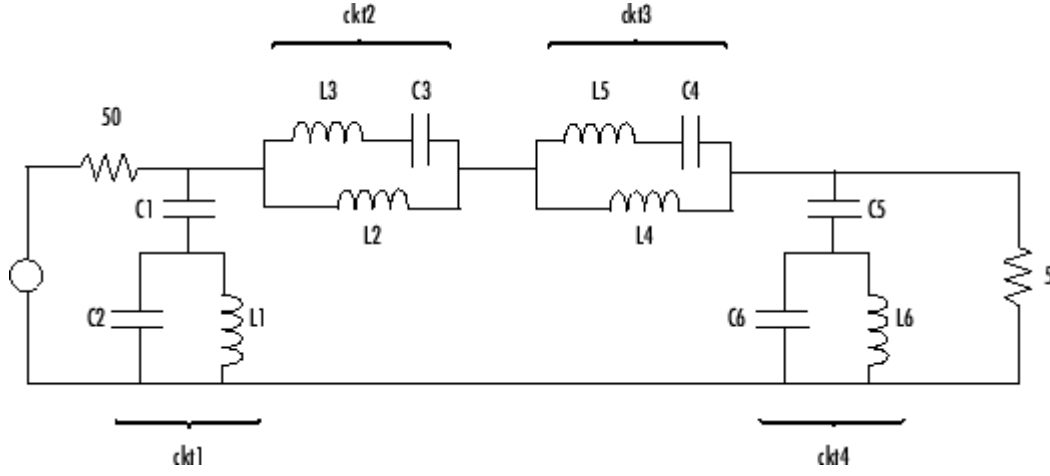
3. Enter the circuit object name in the **RFCKT object** parameter in the General Circuit Element block dialog box.

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 2-17
- "Building the Model" on page 2-18
- "Specifying and Importing Component Data" on page 2-19
- "Running the Simulation and Plotting the Results" on page 2-21


## 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 2-16, and use functions from RF Toolbox 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 Simulink model.

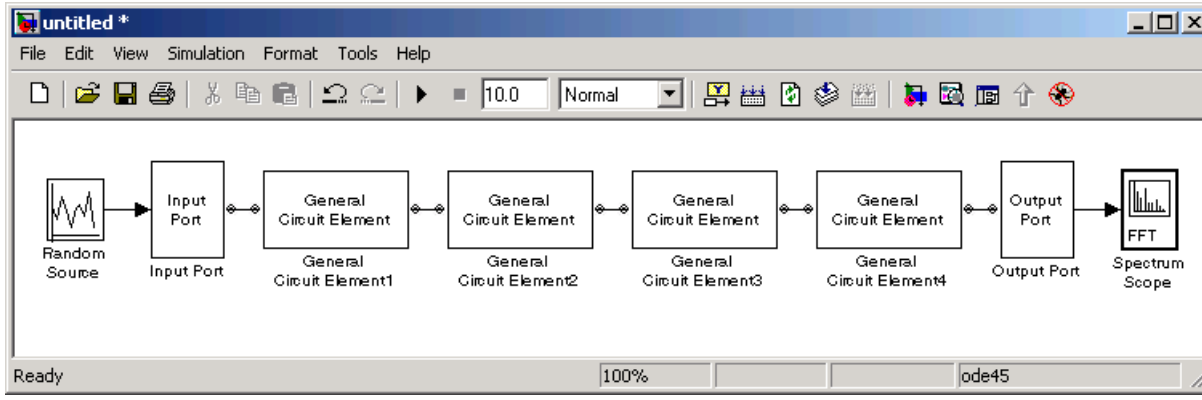
2 Add to the model the blocks shown in the following table. The Library column of the table specifies the hierarchical path to each block.

| Block | Library | Quantity |
|---|---|---|
| **Random Source** | **Signal Processing Blockset > Signal Processing Sources** | 1 |

| Block | Library | Quantity |
|---|---|---|
| **Input Port** | **RF Blockset > Physical > Input/Output Ports** | 1 |
| **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 Simulink 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`.

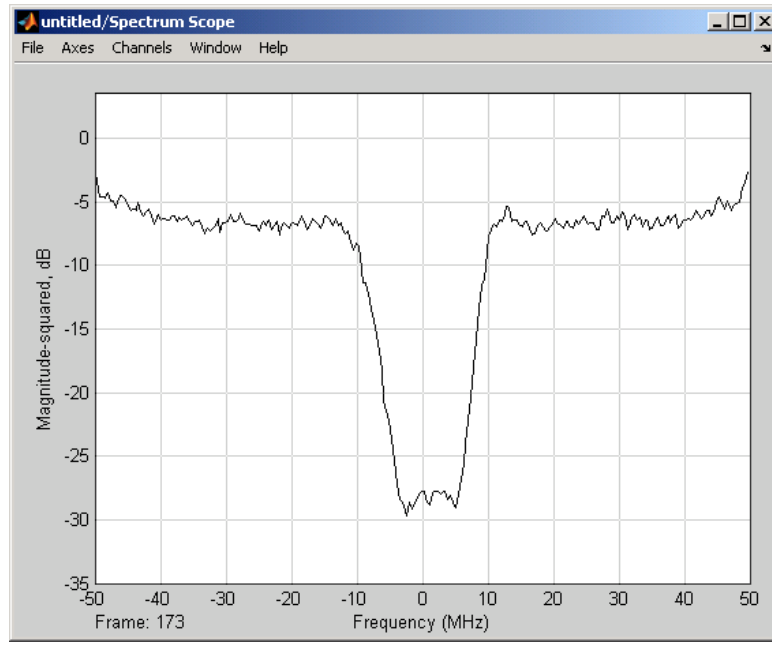
  These values set the range of X- and Y-values on the display such that the entire signal is visible when you run the simulation.

## 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 Simulink 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**

## References

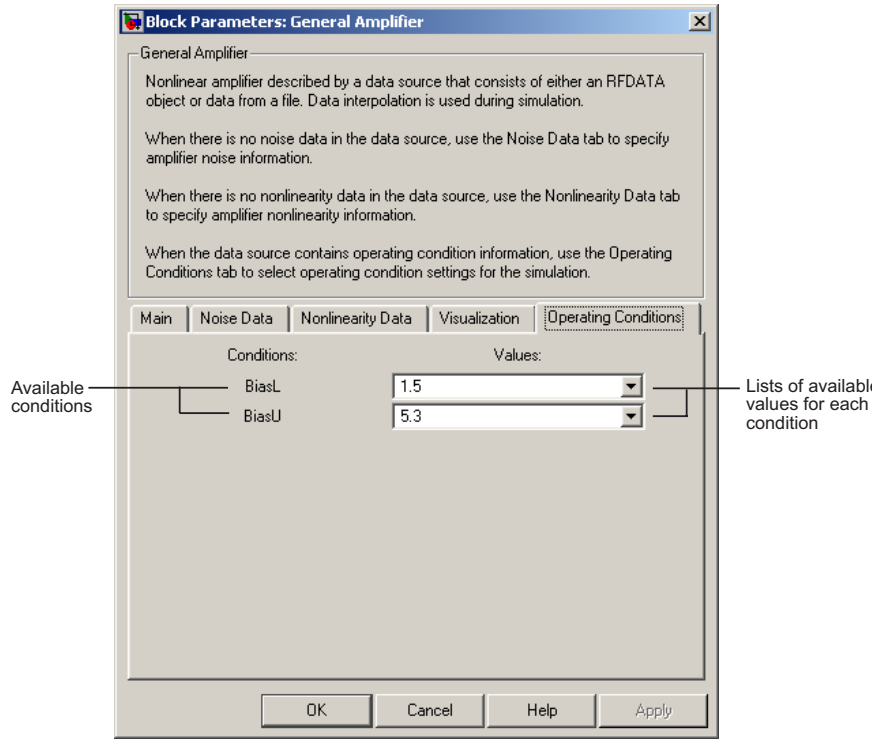
Geffe, P.R., "Novel designs for elliptic bandstop filters," *RF Design*, February 1999.

# 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, RF 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

RF Blockset lets you include the nonlinearity generated by the physical amplifier and mixer blocks in your RF model. This section describes how to model nonlinearity and contains the following topics:


- "Amplifier and Mixer Nonlinearity Specifications" on page 2-25
- "Adding Nonlinearity to Your System" on page 2-26


## 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)<br>**OR**<br>Third-order intercept data, with or without one or more power parameters, in the block dialog box. |
| S-Parameters Amplifier<br>Y-Parameters Amplifier<br>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)<br>**OR**<br>Third-order intercept data, with or without one or more power parameters, in the block dialog box. |
| S-Parameters Mixer<br>Y-Parameters Mixer<br>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.<br><br>**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 RF Blockset simulates nonlinearity data of an amplifier or mixer, see the block reference page.

# Modeling Noise

RF Blockset lets you include the noise generated by any physical block in your RF model. 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, RF Blockset calculates the noise automatically based on the resistor values.

This section contains the following topics:


- "Amplifier and Mixer Noise Specifications" on page 2-28
- "Adding Noise to Your System" on page 2-29
- "Plotting Noise" on page 2-32


## Amplifier and Mixer Noise Specifications

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)<br>**OR**<br>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<br>Y-Parameters Amplifier<br>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)<br>**OR**<br>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<br>Y-Parameters Mixer<br>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 2-29
- "Adding Noise to the Simulation" on page 2-32


### 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.

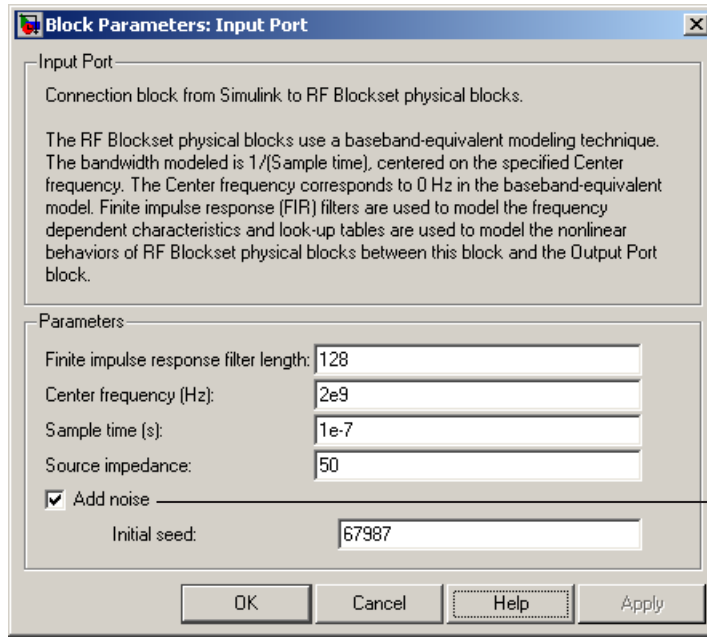
| Noise Specification | Instructions |
|---|---|
| Frequency-independent noise figure | In the **Noise Data** tab of the block dialog box, set the **Noise type** parameter to `Noise figure`, and enter the noise figure value in the **Noise figure (dB)** parameter. |
| Frequency-dependent noise figure | 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.



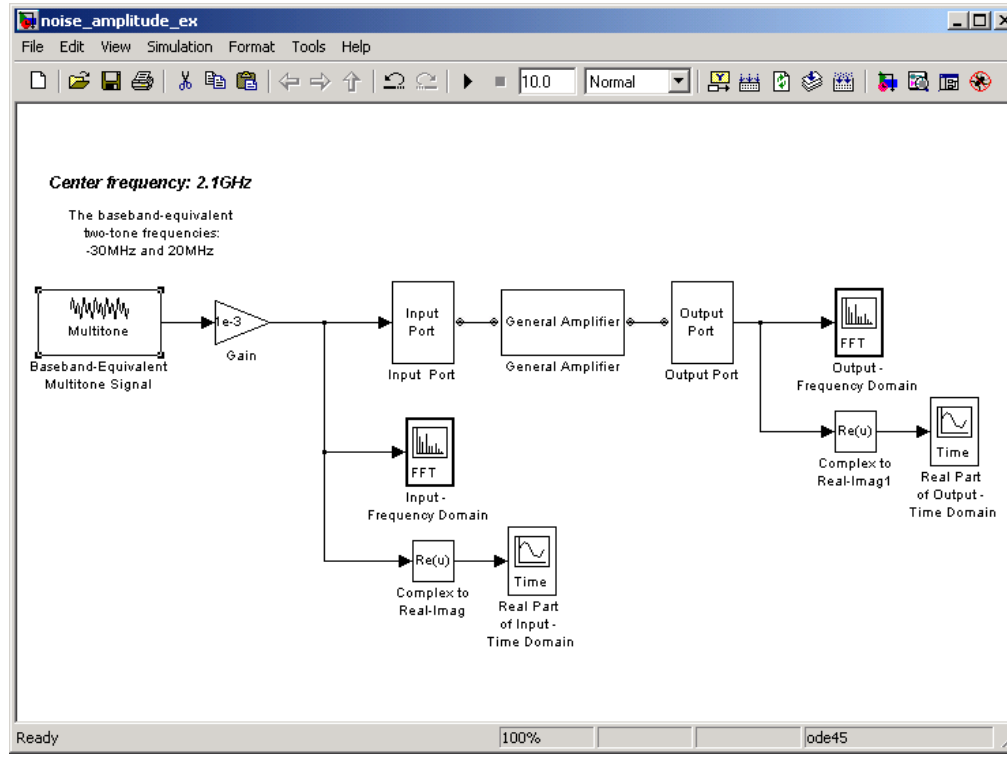

For information on how RF Blockset simulates noise, see “Modeling Noise in an RF System” on page A-6.

## Plotting Noise

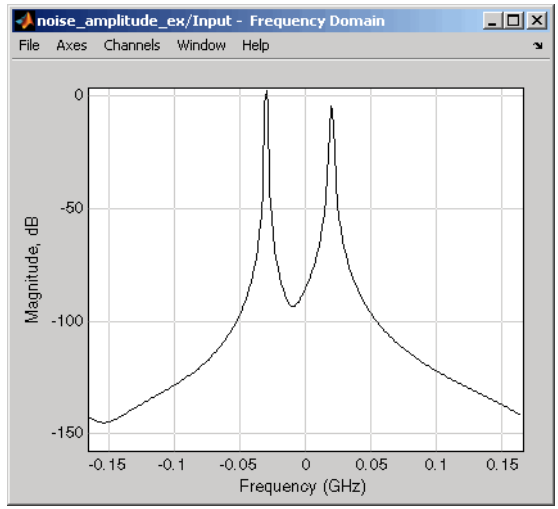
RF Blockset models communications systems. The noise in these systems has a very small amplitude, typically from 1e-6 to 1e-12 Watts. In contrast, the default signal power of a Communications Blockset modulator block is 1 Watt at a nominal 1 ohm. Therefore, the signal-to-noise ratio in an RF system simulation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

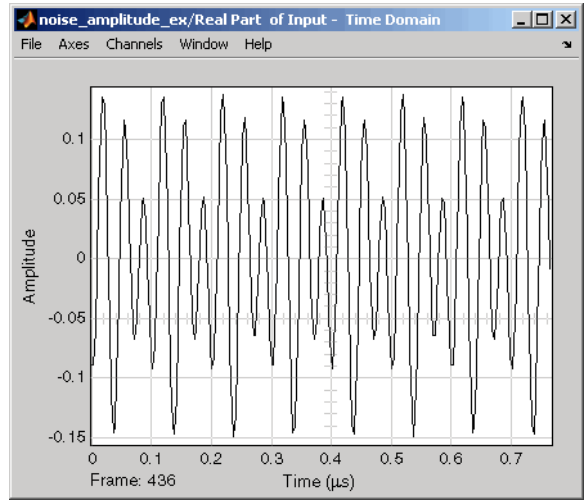


When you simulate this model, Simulink brings up several windows showing the input and output for the physical subsystem. The Input - Frequency Domain window shown in the following figure displays the input signal in the frequency domain.



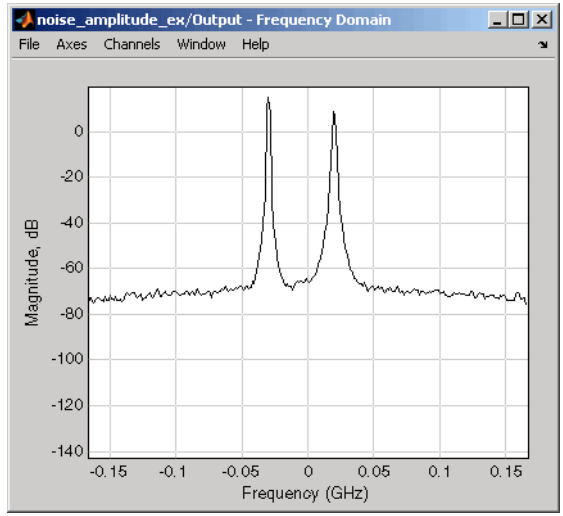

**Input Signal Spectrum**

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



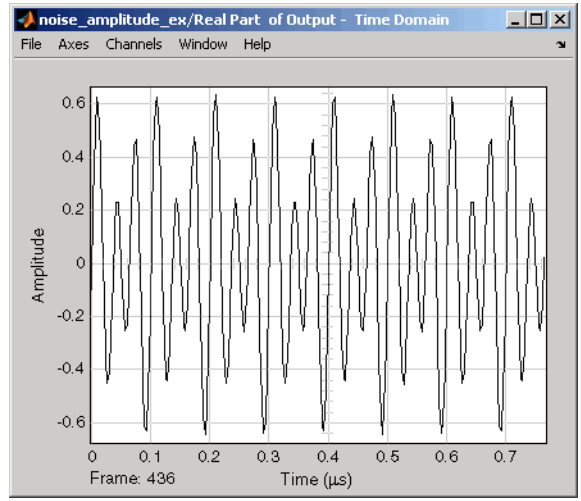

**Real Part of Input Signal**

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.



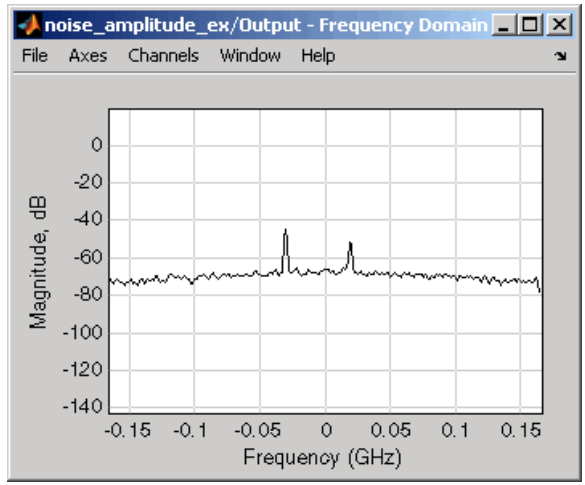

**Output Signal Spectrum**

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.



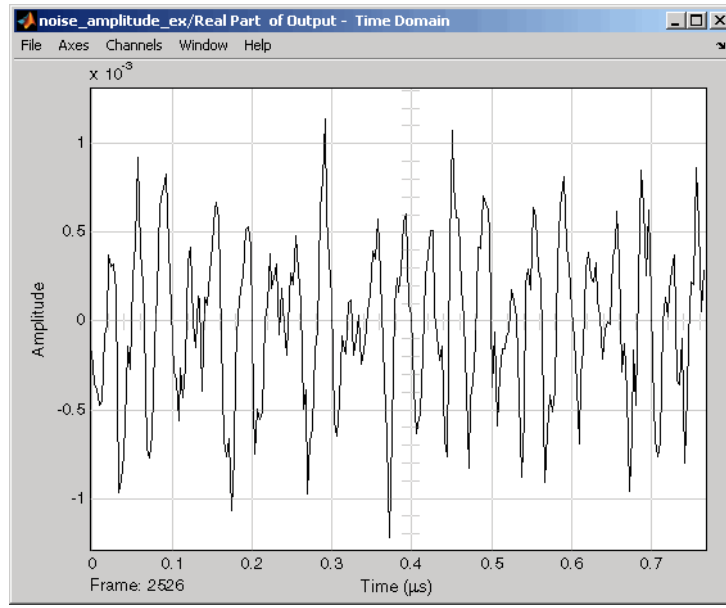

**Real Part of Output Signal**

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 adds 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 (p. 3-2) | Describes the available plots and explains how to plot data for components and subsystems |
| Updating Plots (p. 3-25) | Describes how to update existing plots after changing model parameters |
| Modifying Plots (p. 3-26) | Describes how to add data to a plot and how to change the plot type, format, and frequency range |
| Example — Creating and Modifying Subsystem Plots (p. 3-29) | Shows how to plot RF subsystem data, add data to the plot, and change the plot type |

# Creating Plots

RF Blockset 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

The following topics describe the RF Blockset plotting options, explain the procedure for creating a plot, and illustrate this procedure with an example:


- "Validating Individual Blocks and Subsystems" on page 3-2
- "Types of Plots" on page 3-3
- "Plot Formats" on page 3-4
- "How to Create a Plot" on page 3-13
- "Example — Plotting Component Data on a Z Smith Chart" on page 3-20


## Validating 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 3-29. 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 3-13.

## Types of Plots

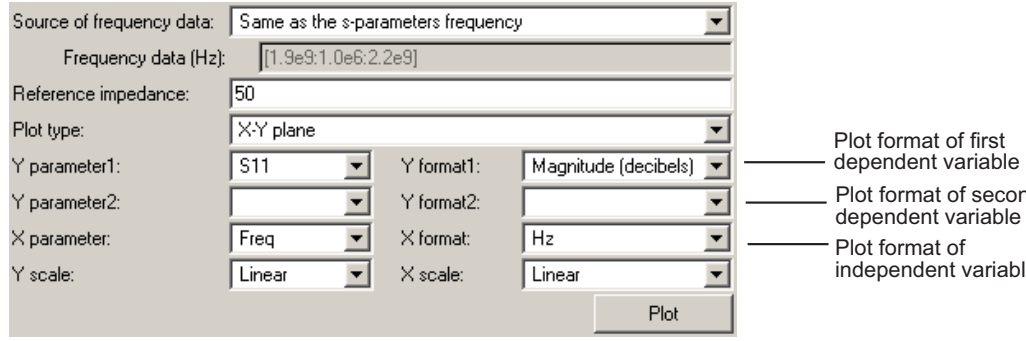
RF Blockset provides a variety of plots for analyzing the behavior of RF components and subsystems. The following table summarizes the available plots and charts and describes each one.

| Plot Type | Plot Contents |
|---|---|
| X-Y Plane (Rectangular) Plot | Parameters as a function of frequency, input power, or operating condition, such as<br>• S-parameters<br>• Noise figure (NF)<br>• Voltage standing-wave ratio (VSWR)<br>• Output third-order intercept point (OIP3) |
| 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 3-9. |
| 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 3-13.

## 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 displays the data on the plot.

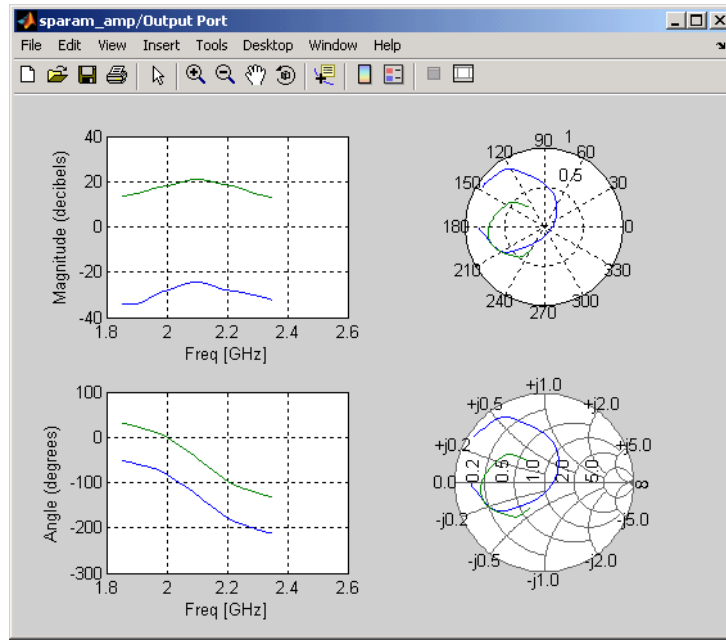
The available formats vary with the data you select to plot. The data you can plot depends on the plot type you select. The plot formats determine whether RF Blockset converts the data to a new set of units, or performs a calculation on the data. For example, setting the format to `Real` tells RF Blockset 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 3-5
- "X-Y Plane" on page 3-7
- "Link Budget" on page 3-9
- "Polar Plane Plots and Smith Charts" on page 3-11

## 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. RF Blockset plots all data as a function of frequency.

| Block | Specified Data | Plot Contents |
|---|---|---|
| General Amplifier or General Mixer | Network parameters<br>**OR**<br>Network parameters and noise | • X-Y plot, magnitude of $S_{12}$ and $S_{21}$ in decibels<br>• X-Y plot, phase of $S_{12}$ and $S_{21}$ in degrees<br>• Z Smith chart, real and imaginary parts of $S_{11}$ and $S_{22}$<br>• Polar plot, magnitude and phase of $S_{11}$ and $S_{22}$ |
| | Network parameters and power<br>**OR**<br>Network parameters, noise, and power | • X-Y plot, magnitude of $S_{12}$ and $S_{21}$ in decibels<br>• X-Y plot, output power ($P_{out}$) in dBm (decibels referenced to one milliwatt)<br>• Z Smith chart, real and imaginary parts of $S_{11}$ and $S_{22}$<br>• Polar plot, magnitude and phase of $S_{11}$ and $S_{22}$ |
| Other Physical block | Network parameters<br>**OR**<br>Network parameters and noise (S-, Y-, and Z-Parameters Amplifiers and Mixers only)<br><br>**Note** Only the General Amplifier and General Mixer blocks accept power data. | • X-Y plot, magnitude of $S_{12}$ and $S_{21}$ in decibels<br>• X-Y plot, phase of $S_{12}$ and $S_{21}$ in degrees<br>• Z Smith chart, real and imaginary parts of $S_{11}$ and $S_{22}$<br>• 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, RF 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`<br>`LS11`, `LS12`, `LS21`, `LS22` (General Amplifier and General Mixer blocks with multiple operating conditions only) | `Magnitude (decibels)`<br>`Magnitude (linear)`<br>`Angle (degrees)`<br>`Angle (radians)`<br>`Real`<br>`Imaginary` |
| `NF` | `Magnitude (decibels)`<br>`Magnitude (linear)` |
| `OIP3` (Amplifier and Mixer blocks only) | `dBm`<br>`dBW`<br>`W`<br>`mW` |
| `VSWRIn`, `VSWROut` | `Magnitude (decibels)`<br>`Magnitude (linear)` |
| `Pout` (General Amplifier and General Mixer blocks with power data only) | `dBm`<br>`dBW`<br>`W`<br>`mW` |
| `Phase` (General Amplifier and General Mixer blocks with power data only) | `Angle (degrees)`<br>`Angle (radians)` |

| Y Parameter | Y Format |
|---|---|
| AM/AM (General Amplifier and General Mixer blocks with power data only) | Magnitude (decibels)<br>Magnitude (linear) |
| AM/PM (General Amplifier and General Mixer blocks with power data only) | Angle (degrees)<br>Angle (radians) |
| PhaseNoise (Mixer blocks only) | dBc/Hz |
| FMIN (Amplifier and Mixer blocks with spot noise data only) | Magnitude (decibels)<br>Magnitude (linear) |
| GAMMAOPT (Amplifier and Mixer blocks with spot noise data only) | Magnitude (decibels)<br>Magnitude (linear)<br>Angle (degrees)<br>Angle (radians)<br>Real<br>Imaginary |
| RN (Amplifier and Mixer blocks with spot noise data only) | None<br>This format tells RF 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<br>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`<br>`dBW`<br>`W`<br>`mW` |
| `Freq` | `THz`<br>`GHz`<br>`MHz`<br>`KHz`<br>`Hz`<br>`Auto` (`xformat` is chosen to provide the best scaling for the given `xparameter` values.) |
| `AM` | `Magnitude (dB)`<br>`Magnitude (linear)` |

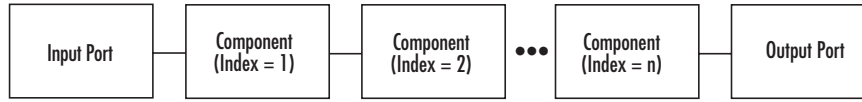
When you import block data from a `.p2d` or `.s2d` file, you can also plot Y parameters as a function of any operating condition from the file that has numeric values, such as bias. You can specify an operating condition as the X parameter only when validating individual blocks, and the format is always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

## Link Budget

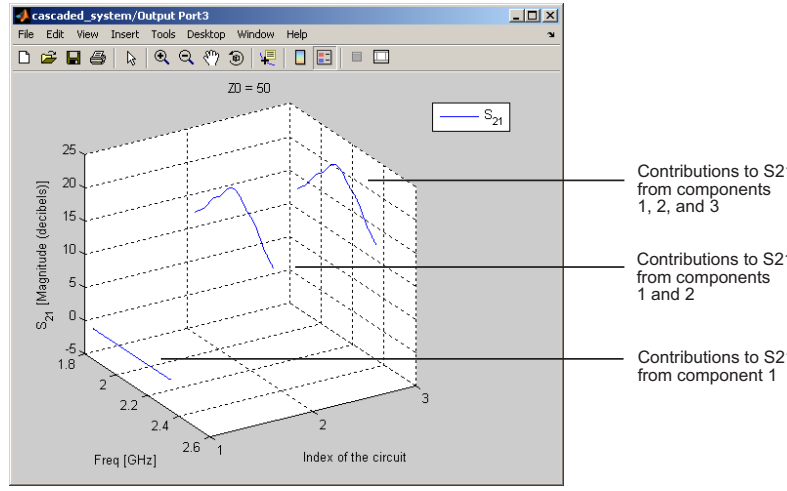
You use the `Link budget` plot to understand the individual contribution of each block to a plotted Y parameter value in a physical subsystem with multiple components between the Input Port and the Output Port blocks.

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 sequential 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)`<br>`Magnitude (linear)`<br>`Angle (degrees)`<br>`Real`<br>`Imaginary` |
| `VSWRIn`, `VSWROut` | `Magnitude (decibels)`<br>`Magnitude (linear)` |
| `OIP3` | `dBm`<br>`dBW`<br>`W`<br>`mW` |
| `NF` | `Magnitude (decibels)`<br>`Magnitude (linear)` |

If you specify two Y parameters, RF Blockset 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 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, RF Blockset puts both parameters in a single plot.

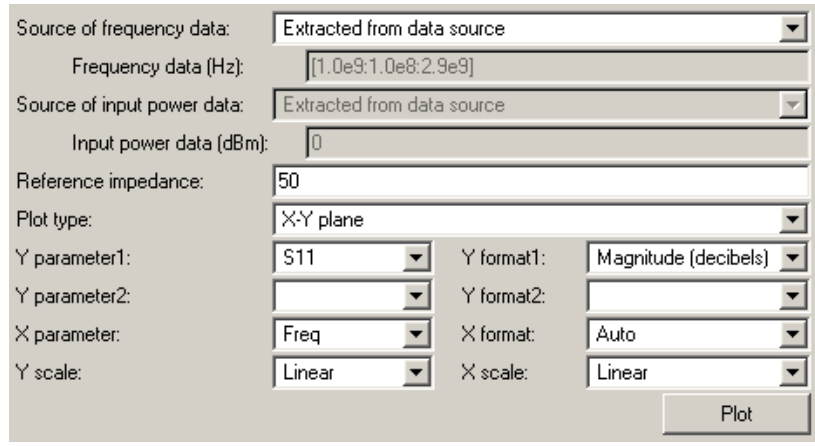
The following table describes the Polar plot and Smith chart options, as well as the available Y parameters.

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

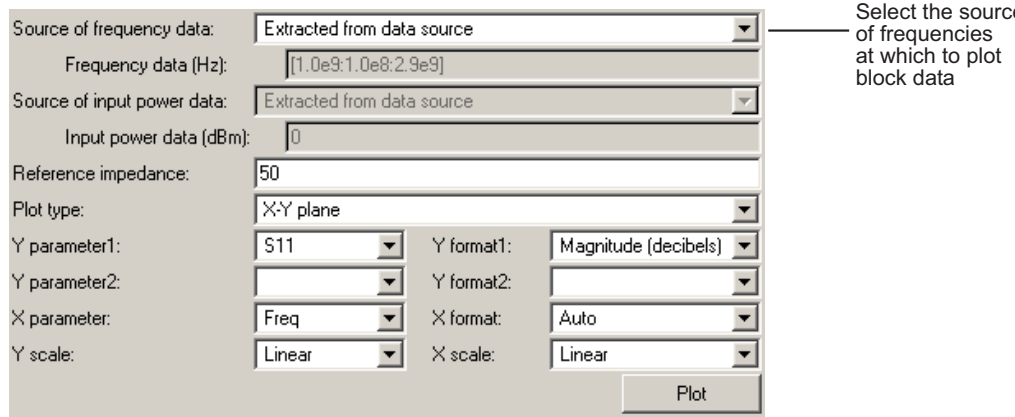
By default, the X parameter is `Freq`. The format of the X parameter specifies the units of the X-axis. When you import block data from a `.p2d` or `.s2d` file, you can also plot Y parameters as a function of any operating condition from the file that has numeric values, such as bias. You can specify 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**.

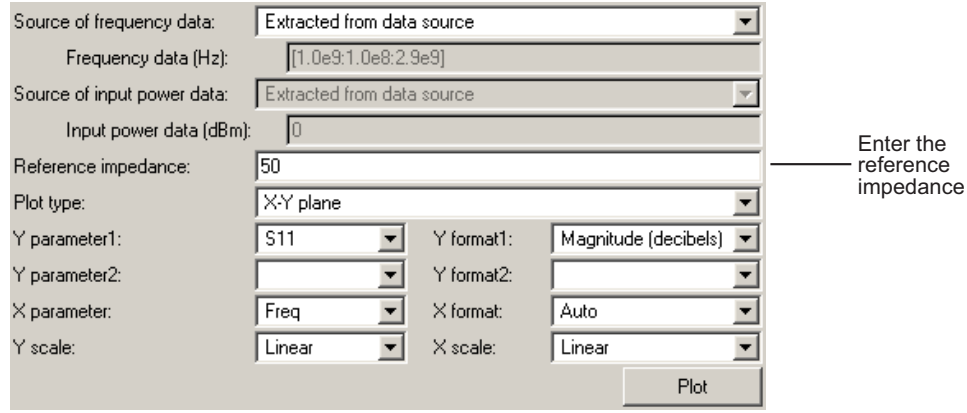



This value is the source of the frequency values at which to plot block data. The following table summarizes the available types of sources for the various types of blocks.

| Source of frequency data | Description | Blocks |
|---|---|---|
| `User-specified` | Vector of frequencies that you enter.<br><br>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.<br><br>For example, to plot block data from 0.3 MHz to 5 GHz by 0.1 MHz, enter `[0.3e6:0.1e6:5e9]`.<br><br>**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 |

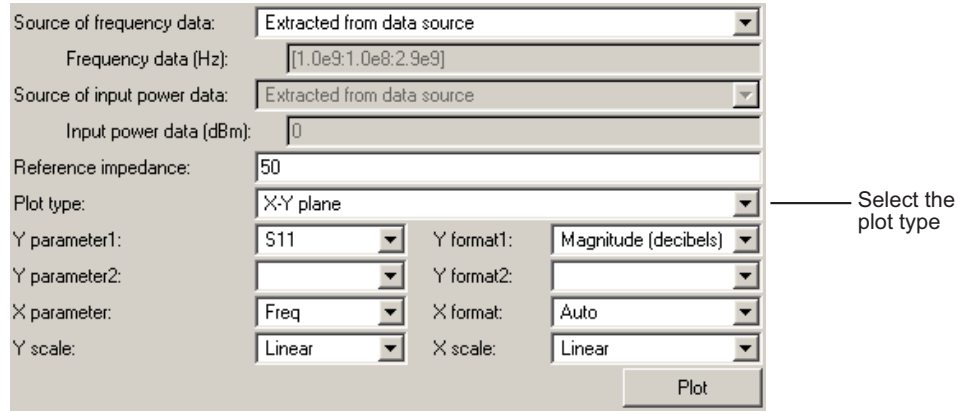
| Source of frequency data | Description | Blocks |
|---|---|---|
| `Derived from Input Port parameters` (Available after running a simulation or clicking the **Update Diagram** button ) | Modeling frequencies derived from the Input Port block parameters. For information on how RF Blockset computes the 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**.




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

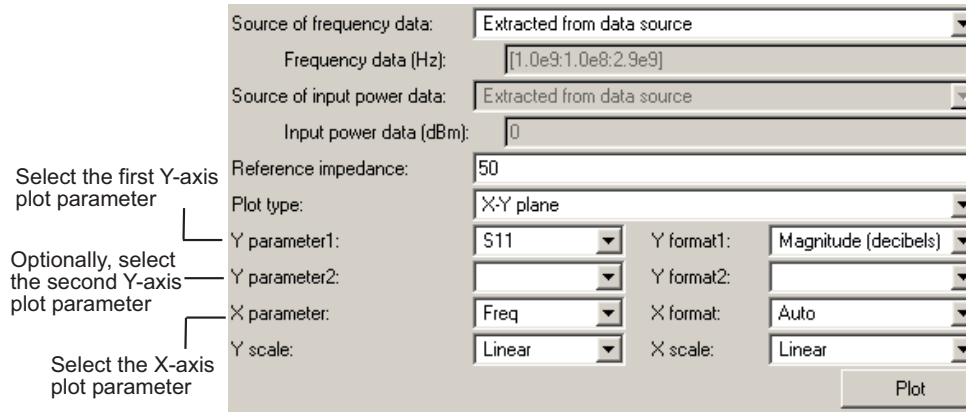
**4** Select the **Plot type**.




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.



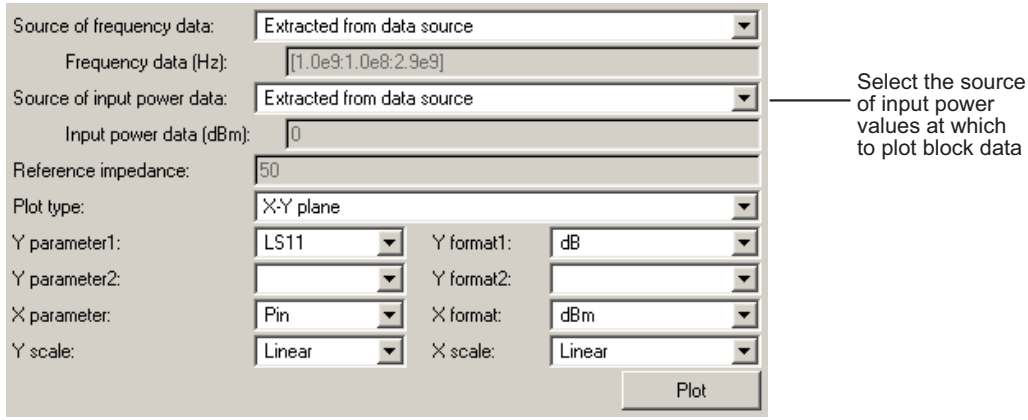

These parameters specify the data to be plotted. The available choices vary with the type of plot. For a description of the options for a particular plot type, see the topic on that plot type in “Plot Formats” on page 3-4.

**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.

---

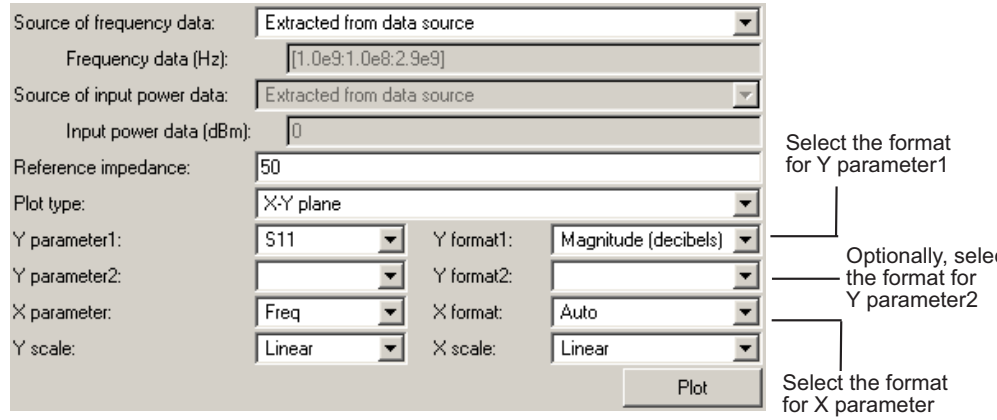



This value is the source of the input power values at which to plot block data. The following table summarizes the available types of sources for the General Amplifier and General Mixer blocks.

| 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.<br><br>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.<br><br>For example, to plot block data from 1 dBm to 10 dBm by 2 dBm, enter `[1:2:10]`. |

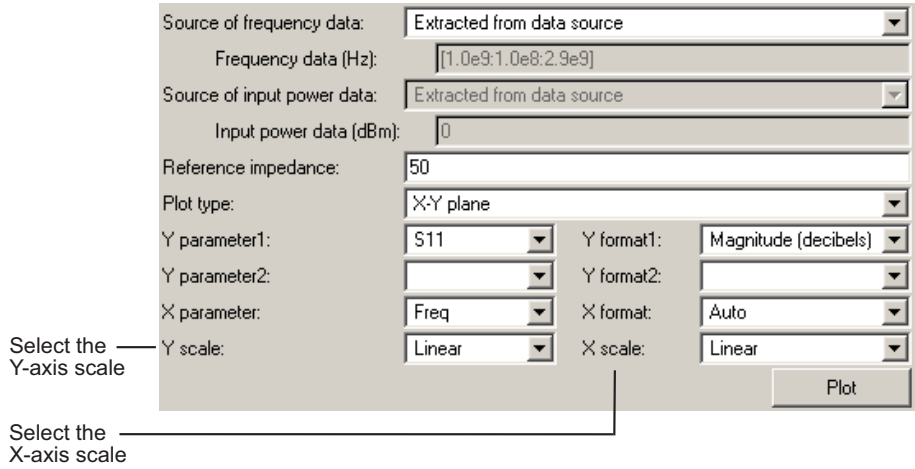
**7** Select the following formats:

- **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, RF Blockset does not add a legend to some plots. To display the plot legend, type `legend show` at the MATLAB prompt.

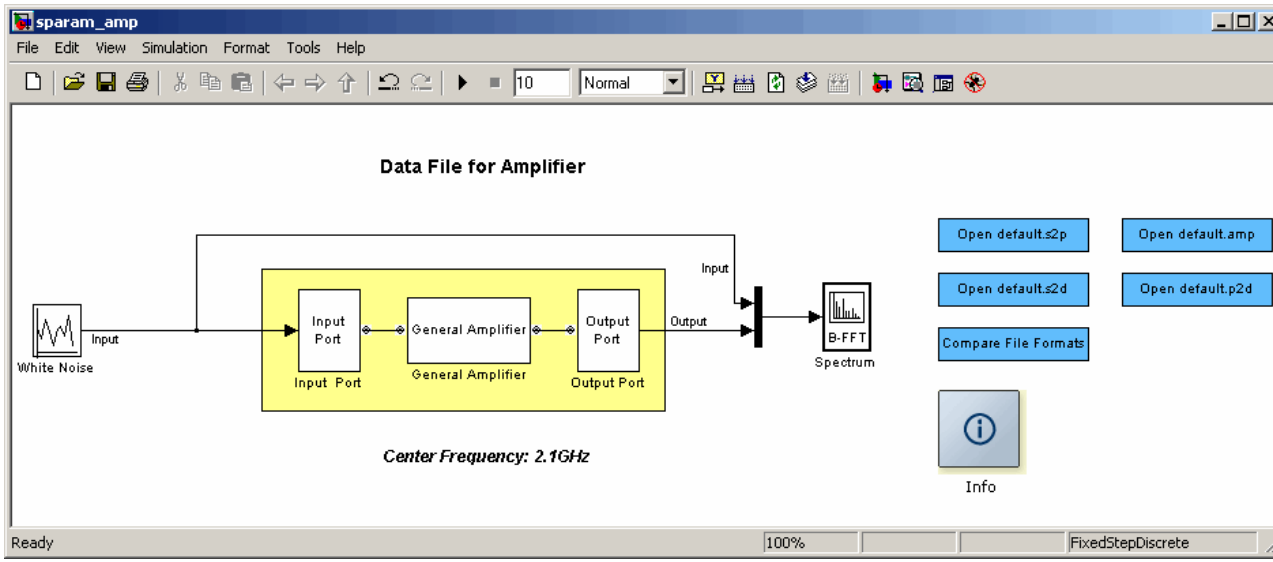
---

## Example — Plotting Component Data on a Z Smith Chart

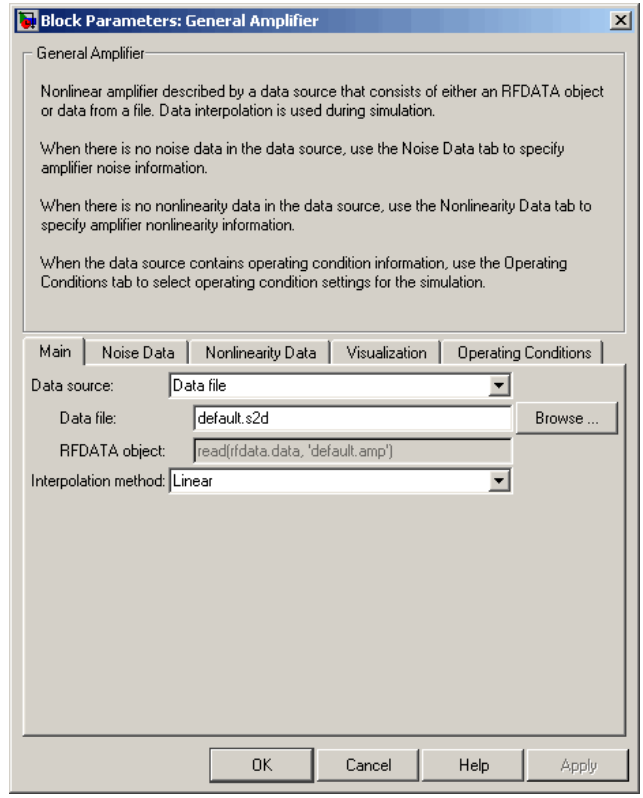
In this example, you simulate the frequency response of an amplifier using data from the `default.s2d` S2D file.

Using a model from one of the RF Blockset 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”.



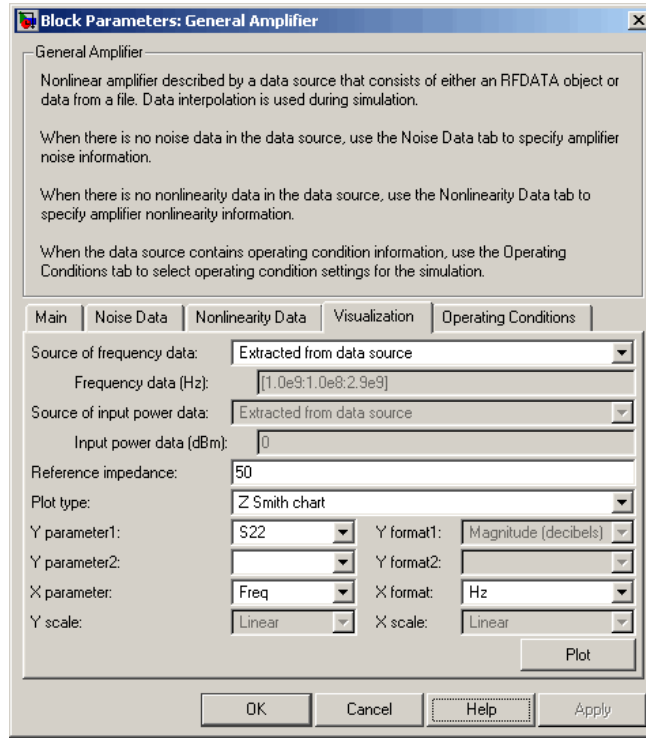
**2** Double-click the General Amplifier block to display its parameters.




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 RF Blockset to import data from the file `default.s2d`. The block uses this data, along with the other block parameters, in simulation.

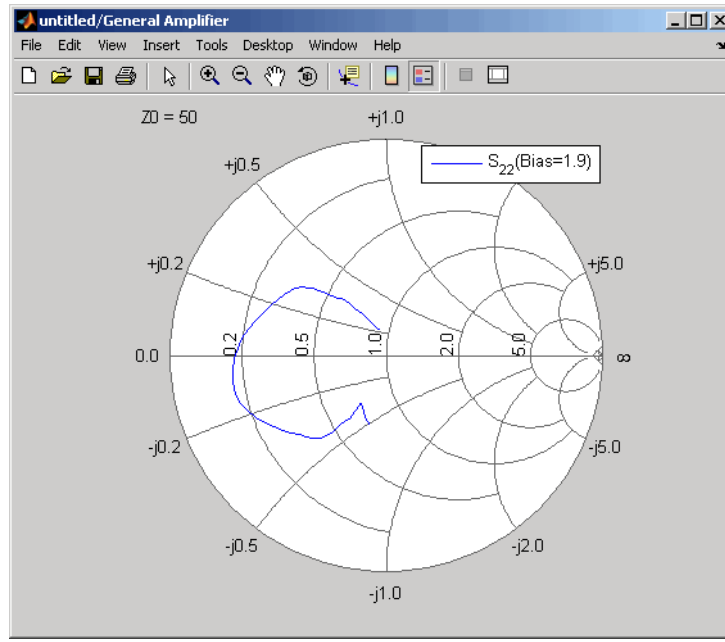
**3** Select the **Visualization** tab and set the General Amplifier block parameters as follows:

- 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.

---

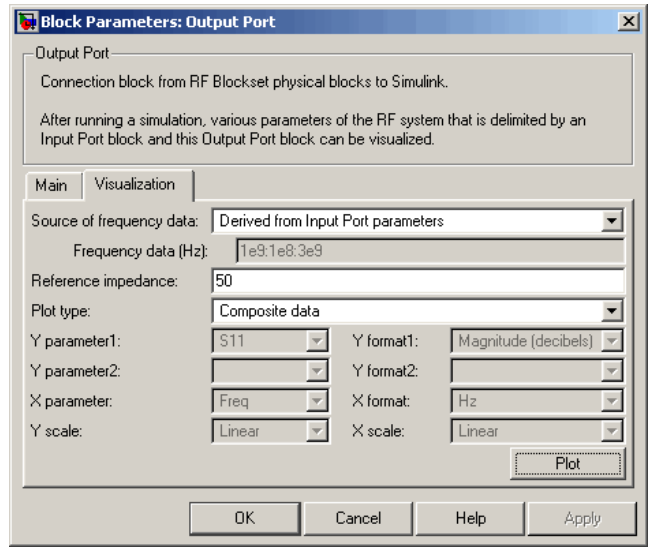
# Updating Plots

When you run a simulation, RF 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 RF 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.

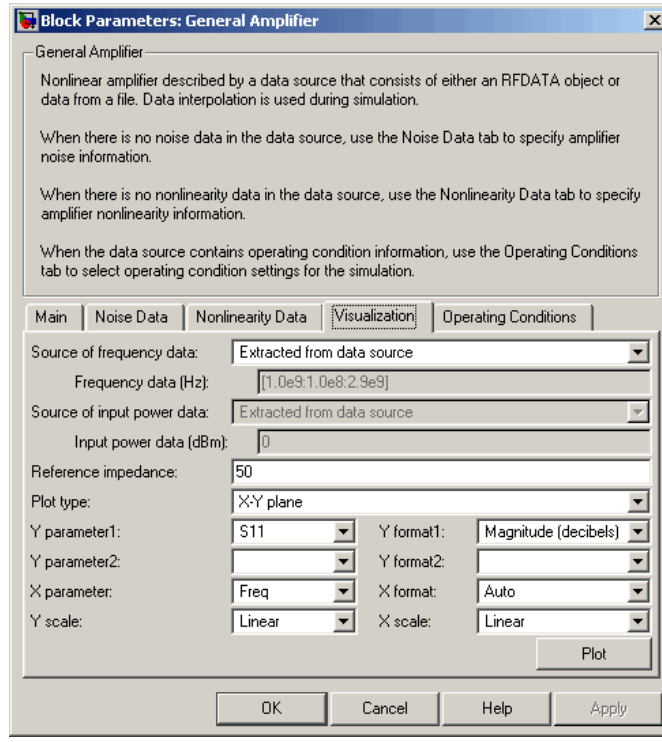
The following table summarizes the results of changing the plot options.

| Block Parameter | Plot Change |
|---|---|
| **Source of frequency data**<br>OR<br>**Frequency data (Hz)** | Redraws plot using the new frequency data. |
| **Source of power data**<br>OR<br>**Input power data (dBm)** | Redraws plot using the new power data. |
| **Plot type** | Draws plot in a new figure using the new plot type.<br><br>**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**<br>OR<br>**Y Parameter2** | If the new parameter has the same independent variable and format as the one on the plot, RF Blockset adds the new parameter to the existing plot. Otherwise, it redraws the plot for the new parameter and independent variable. |
| **Y Format1**<br>OR<br>**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 example, you perform the following tasks:


- "Plotting the Network Parameters of a Subsystem" on page 3-29
- "Adding Data to an Existing Plot" on page 3-31
- "Changing Data on an Existing Plot" on page 3-33

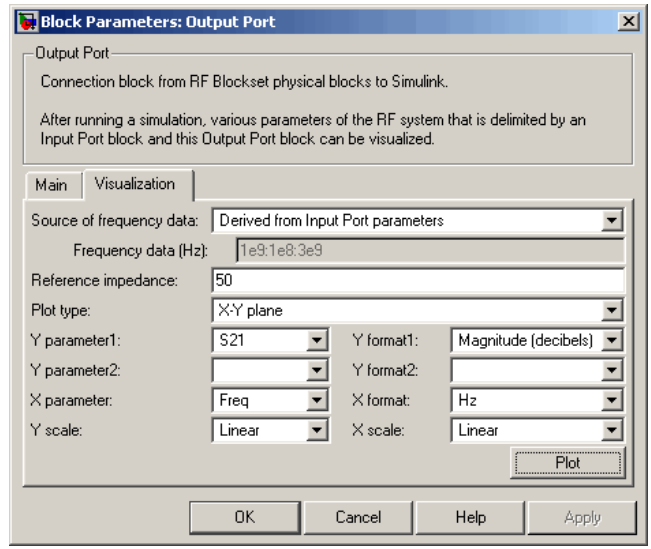

## Plotting the Network Parameters of a Subsystem

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 Simulink model window, select **Simulation > Start** to run the simulation.
3. Double-click the Output Port block to open the block dialog box.

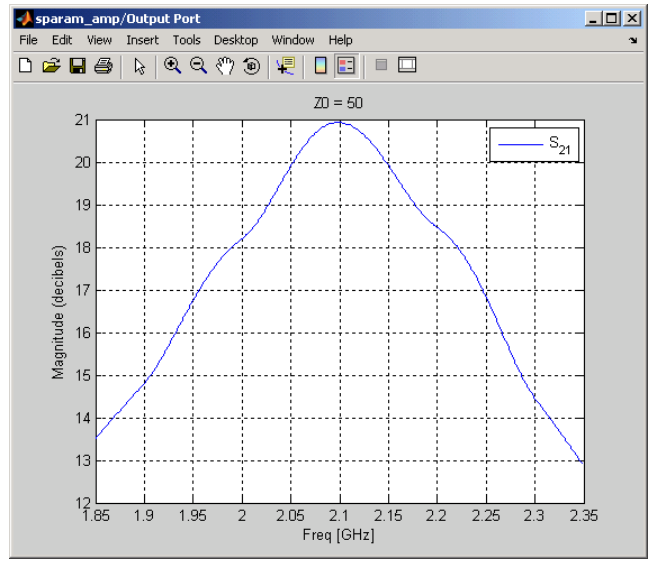
4 Select the **Visualization** tab and set the Output Port block parameters as follows:

- 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.

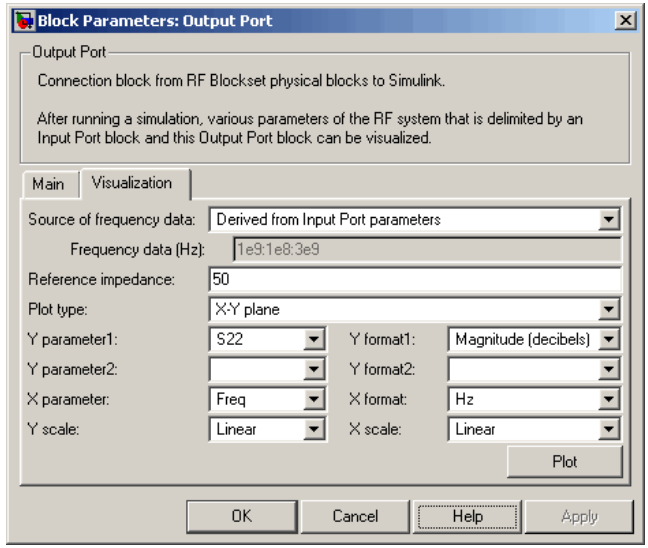



**$S_{21}$ versus Frequency for a Physical Subsystem**

## Adding Data to an Existing Plot

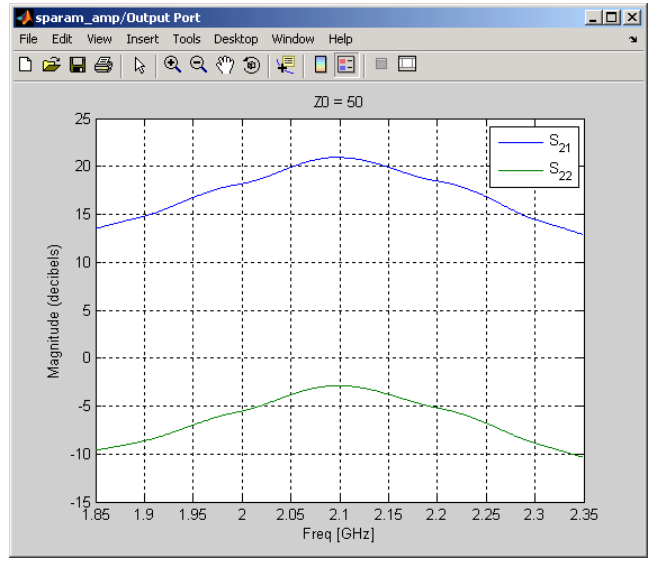
In this part of the example, you add data to the plot you created in “Plotting the Network Parameters of a Subsystem” on page 3-29.

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**

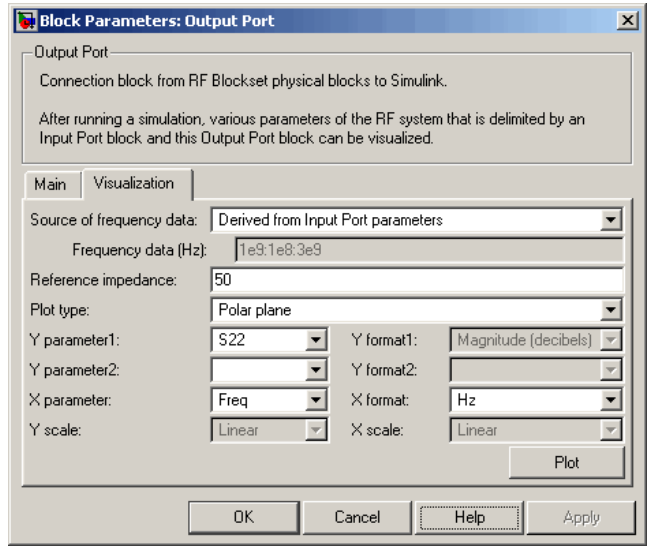
## Changing Data on an Existing Plot

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.

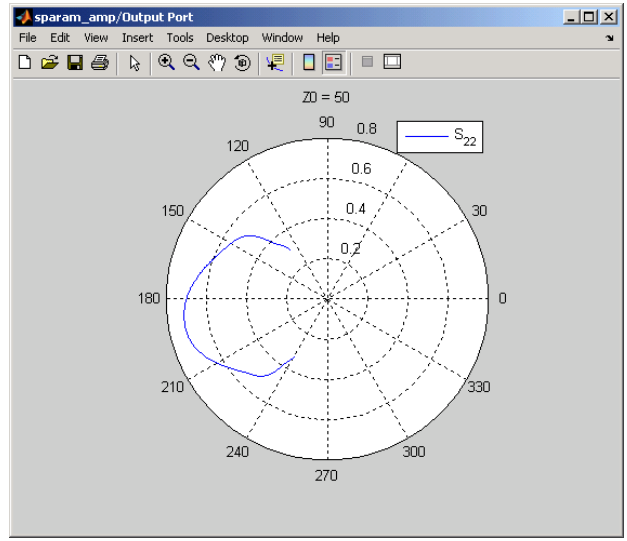
2 Select the **Visualization** tab and change the value of **Plot type** to `Polar plane`, as shown in the following figure.

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




3 Click **Plot**.

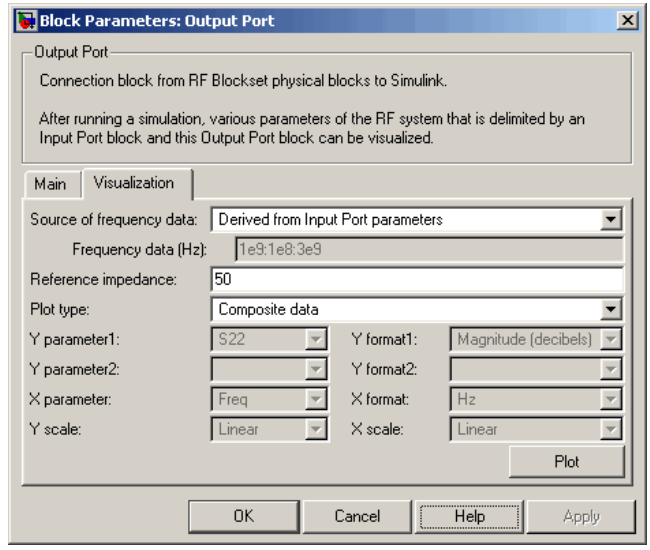
This action creates a Polar plane plot of $S_{22}$ as a function of frequency.




**$S_{22}$ versus Frequency for a Physical Subsystem**

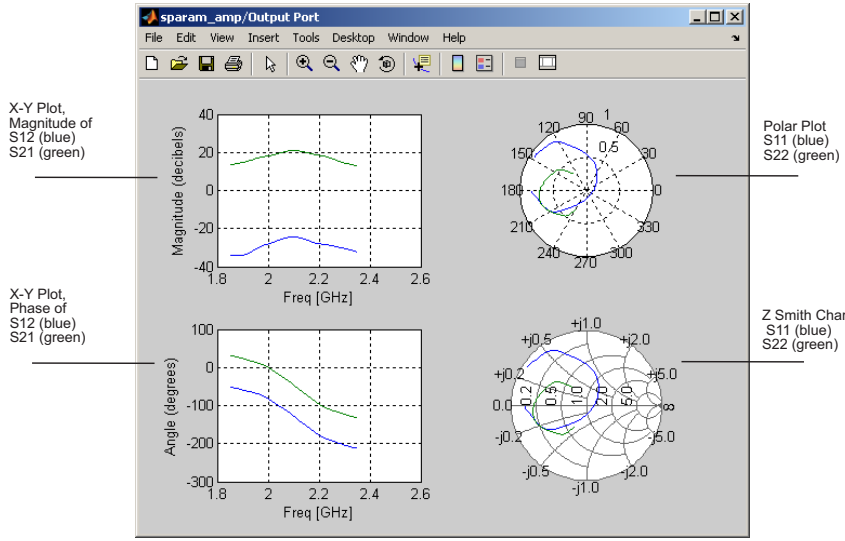
**4** In the Output Port block dialog box, change the **Plot type** to `Composite data` to generate four plots in one figure. The parameters for the plots

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**

4

# Blocks — By Category


| | |
|---|---|
| Mathematical (p. 4-2) | Model RF components in terms of mathematical equations. |
| Physical (p. 4-3) | Model RF components in terms of physical properties or measured data |

# 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 (p. 4-3) | Ladder filter blocks |
| Transmission Lines (p. 4-3) | Transmission line blocks |
| Black Box Elements (p. 4-4) | Black box elements blocks |
| Amplifiers (p. 4-4) | Amplifier blocks |
| Mixers (p. 4-5) | Mixer blocks |
| Input/Output Ports (p. 4-5) | Connector blocks |


## 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 RLC | Model series RLC network |
| 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

# Amplifier

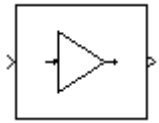
## Purpose

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.

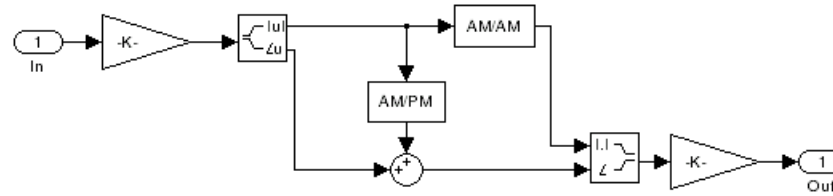
**Note** This block assumes a nominal impedance of 1 ohm.

### Modeling Nonlinearity

Specify the method of your choice using this parameter in the block dialog box. 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 methods 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 interpolation 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 interpolation 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 interpolation 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.

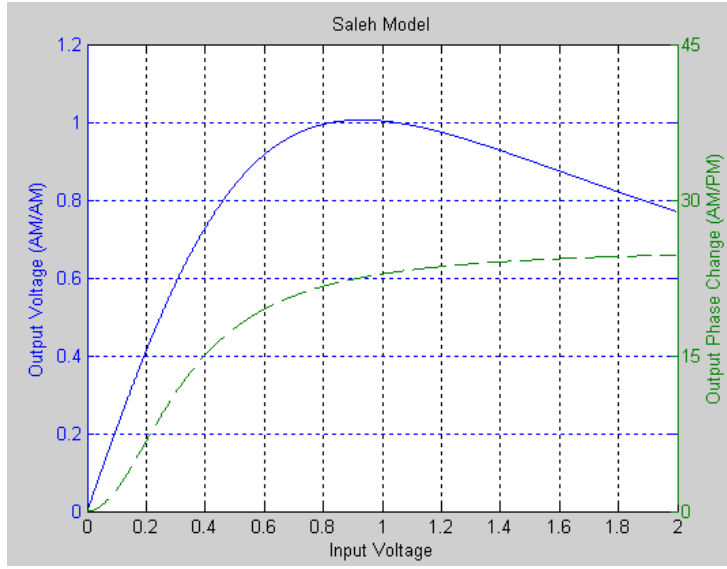
2 Select **Look under mask** in the pop-up menu. This displays the block's configuration underneath the mask. The block contains five subsystems corresponding to the five nonlinearity methods.

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.

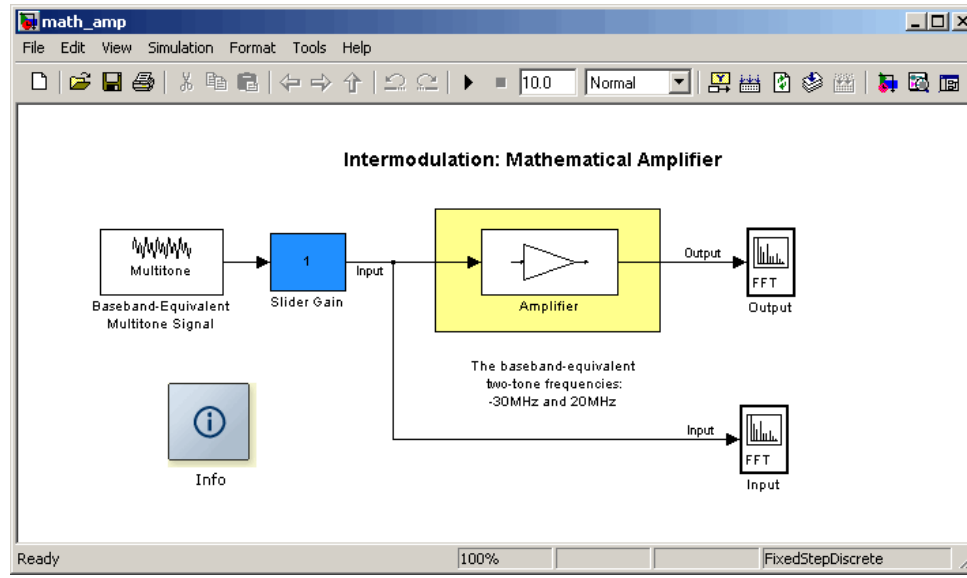
The following figure shows, for the Saleh method, plots of

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

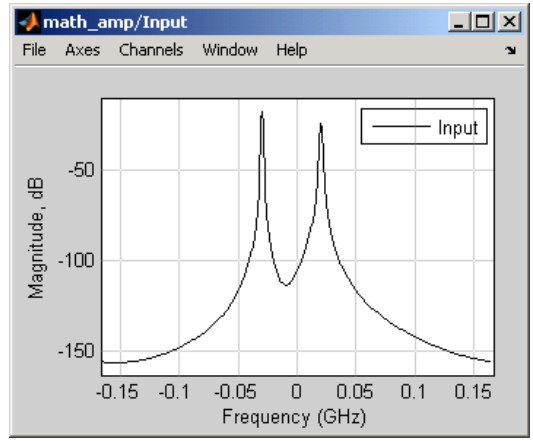


## Effects of the Amplifier Block

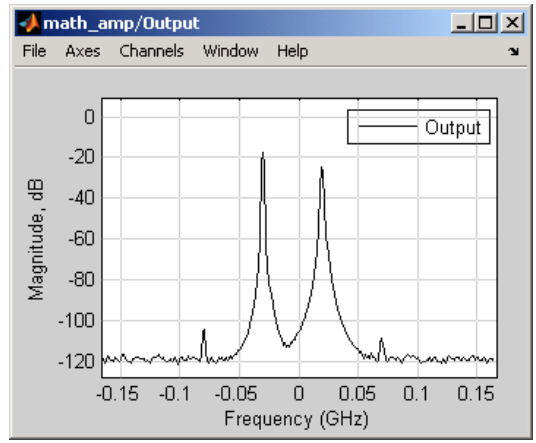
You can see the effect of the Amplifier block in the demo Intermodulation: Mathematical Amplifier.




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.



### Parameters for Nonlinear Methods

The following sections discuss parameters specific to the following models:


- "Cubic Polynomial Model" on page 5-7
- "Hyperbolic Tangent Model" on page 5-8
- "Saleh Model" on page 5-8
- "Ghorbani Model" on page 5-9
- "Rapp Model" on page 5-10


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

### Cubic Polynomial Model

The third-order input intercept point **IIP3 (dBm)** parameter is used to compute a scaling factor, which is then applied to the input signal. The scaling factor is equal to 3 divided by the **IIP3** parameter value, converted from decibels to linear units.

The scaled input is then limited to a maximum value of 1 and the amplitude gain is applied according to the following function

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

where $u$ is the magnitude of the scaled signal.

The **AM/PM conversion (degrees per dB)** parameter specifies the linear phase change. This is applied 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.

### Hyperbolic Tangent Model

The third-order input intercept point **IIP3 (dBm)** parameter is used to compute a scaling factor, which is then applied to the input signal. The scaling factor is equal to 3 divided by the **IIP3** parameter value, converted from decibels to linear units.

The amplitude gain is then applied to the scaled input according to the following function

$$F_{AM/AM}(u) = \tanh(u)$$

where $u$ is the magnitude of the scaled signal.

The **AM/PM conversion (degrees per dB)** parameter specifies the linear phase change. This is applied 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

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

$$F_{AM/AM}(u) = \frac{alpha * u}{1 + beta * u^2}$$

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

$$F_{AM/PM}(u) = \frac{alpha * u^2}{1 + beta * u^2}$$

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

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

$$F_{AM/AM}(u) = \frac{x_1 u^{x_2}}{1 + x_3 u^{x_2}} + x_4 u$$

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

$$F_{AM/PM}(u) = \frac{y_1 u^{y_2}}{1 + y_3 u^{y_2}} + y_4 u$$

where $u$ is the magnitude of the scaled signal.

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

### Rapp Model

The **Smoothness factor** and **Output saturation level** parameters are used to compute the amplitude gain for an input signal by the following function

$$F_{AM/AM}(u) = \frac{u}{\left(1+\left(\frac{u}{O_{\text{sat}}}\right)^{2dS}\right)^{1/2S}}$$

where $u$ is the magnitude of the scaled signal, $S$ is the **Smoothness factor** and $O_{\text{sat}}$ 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:

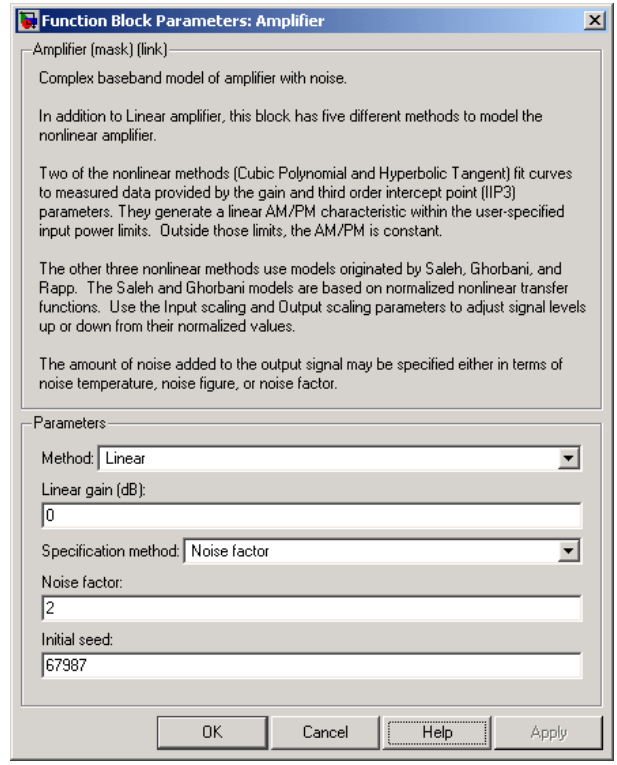
$$\text{Noise factor} = 1 + \frac{\text{Noise temperature}}{290}$$

- `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:

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

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. This 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.

## 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

# Bandpass RF Filter

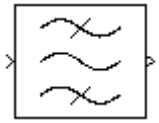
## Purpose

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.

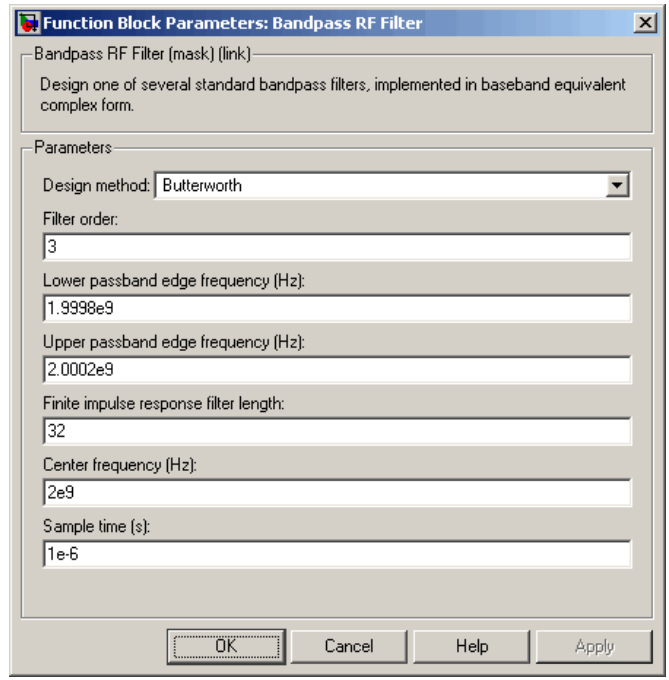
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

| 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)

# Bandstop RF Filter

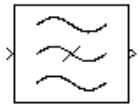
## Purpose

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.

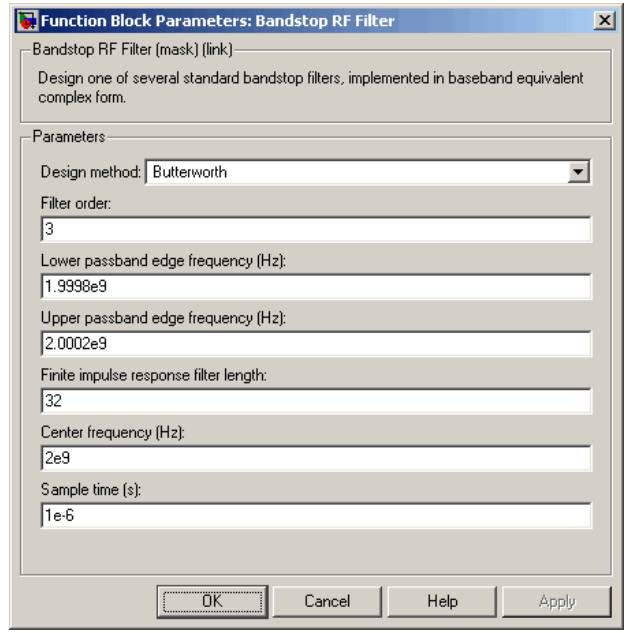
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

| 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, stopband 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)

# Coaxial Transmission Line

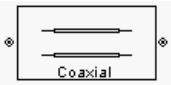
## Purpose

Model coaxial transmission line

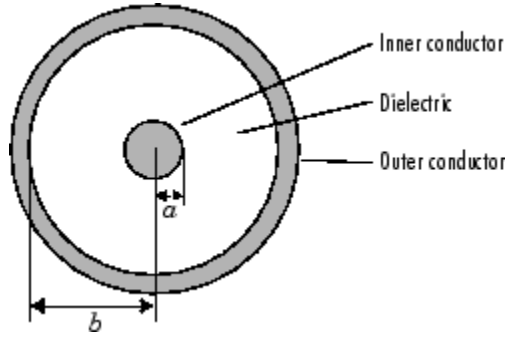
## Library

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 calculates the frequency-dependent S-parameters using the physical length of the transmission line, $D$, and the complex propagation constant, $k$.

$$S_{11} = 0$$

$$S_{12} = e^{-kD}$$

$$S_{21} = e^{-kD}$$

$$S_{22} = 0$$

$k$ is a vector whose elements correspond to the elements of $f$, a vector of modeling frequencies. It can be expressed in terms of the resistance ($R$), inductance ($L$), conductance ($G$), and capacitance ($C$) per unit length (meters) as

$$k = k_r + jk_i = \sqrt{(R + j2\pi fL)(G + j2\pi fC)}$$

where

$$R = \frac{1}{2\pi\sigma_{\text{cond}}\delta}\left(\frac{1}{a} + \frac{1}{b}\right)$$

$$L = \frac{\mu}{2\pi}\ln(b/a)$$

$$G = \frac{2\pi\sigma_{\text{diel}}}{\ln(b/a)}$$

$$C = \frac{2\pi\varepsilon}{\ln(b/a)}$$

In these equations, $\sigma_{\text{cond}}$ is the conductivity in the conductor and $\sigma_{\text{diel}}$ is the conductivity in the dielectric. $\mu$ is the permeability of the dielectric, $\varepsilon$ is its permittivity, and skin depth $\delta$ is calculated as $1/\sqrt{\pi f \mu \sigma_{\text{cond}}}$. $f$ 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.

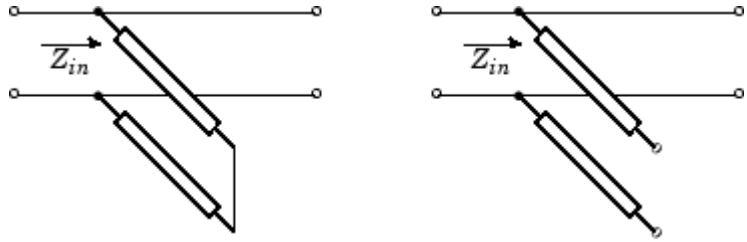
### Shunt and Series Stubs

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.

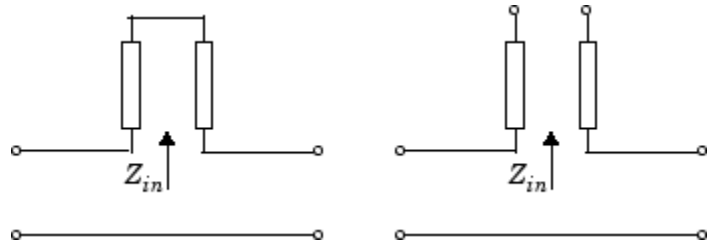



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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.

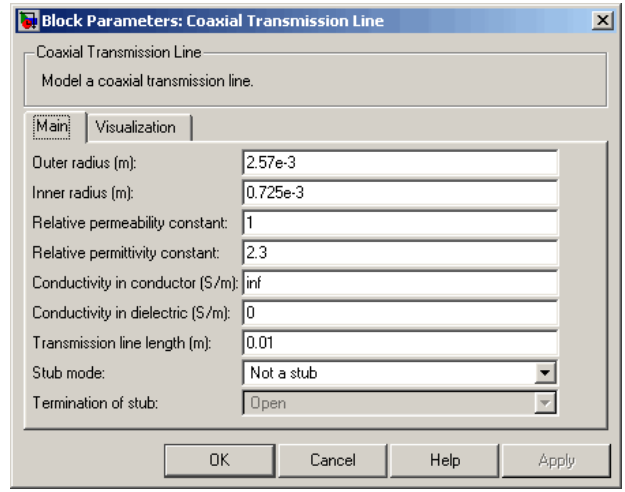



$Z_{in}$ is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## 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 $\mu_0$.

**Relative permittivity constant**
Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space $\varepsilon_0$.

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

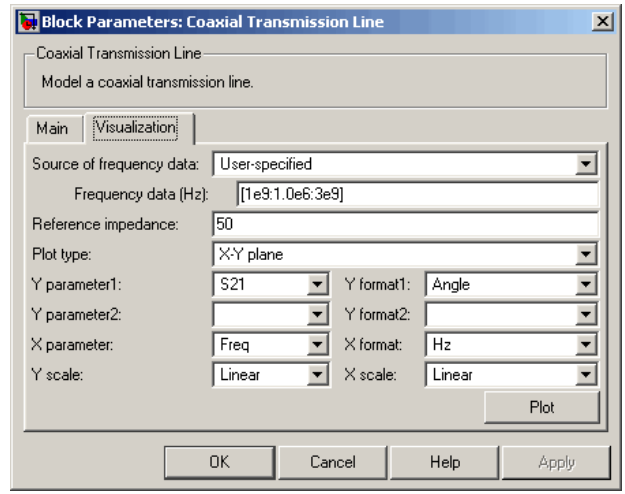
**Conductivity in dielectric (S/m)**
Conductivity of the dielectric in siemens per meter.

**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

# Connection Port

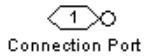
## Purpose

Connection port for RF subsystem

## Library

Input/Output Ports sublibrary of the Physical library

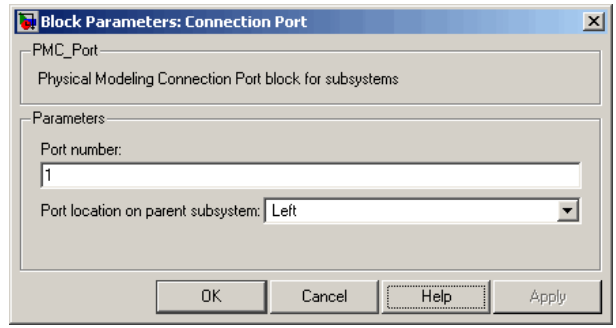
## Description




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.

# Coplanar Waveguide Transmission Line

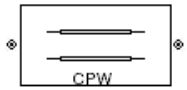
## Purpose

Model coplanar waveguide transmission line

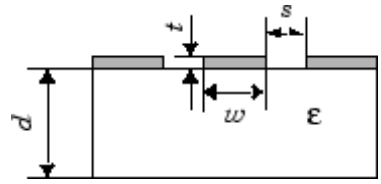
## Library

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 ($\varepsilon$).




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 calculates the frequency-dependent S-parameters using the physical length of the transmission line, $D$, and the complex propagation constant, $k$.

$$S_{11} = 0$$

$$S_{12} = e^{-kD}$$

$$S_{21} = e^{-kD}$$

$$S_{22} = 0$$

$k = \alpha_a + i\beta$, where $\alpha_a$ is the attenuation coefficient and $\beta$ is the wave number. The attenuation coefficient $\alpha_a$ is related to the loss, $\alpha$, by

$$\alpha_a = -\ln 10^{\frac{\alpha}{20}}$$

where $\alpha$ is the reduction in signal strength, in dB, per unit length. $\alpha$ combines both conductor loss and dielectric loss and is derived from the physical parameters specified in the Coplanar Waveguide Transmission Line block dialog box.

The wave number $\beta$ is related to the phase velocity, $V_P$, by

$$\beta = \frac{2\pi f}{V_p}$$

$V_P = c/\sqrt{\varepsilon_{\text{eff}}}$, where $\varepsilon_{\text{eff}}$ is the frequency-dependent effective dielectric constant. $f$ is a vector of modeling frequencies determined by the Output Port block. The phase velocity $V_P$ is also known as the wave propagation velocity.

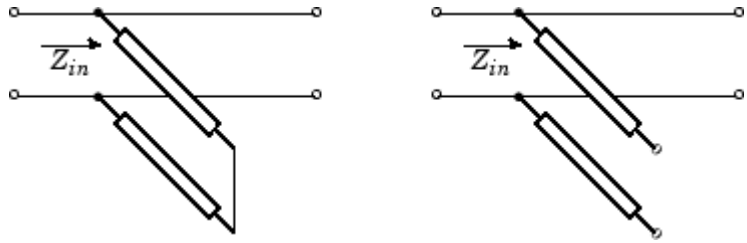
The Coplanar Waveguide Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

### 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.

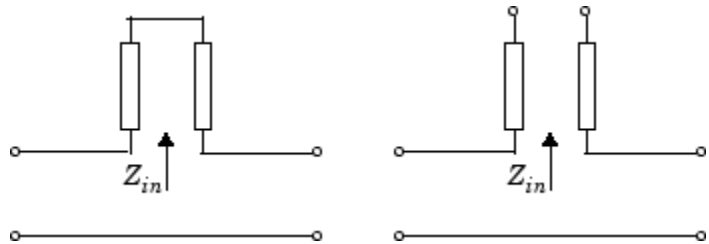



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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.

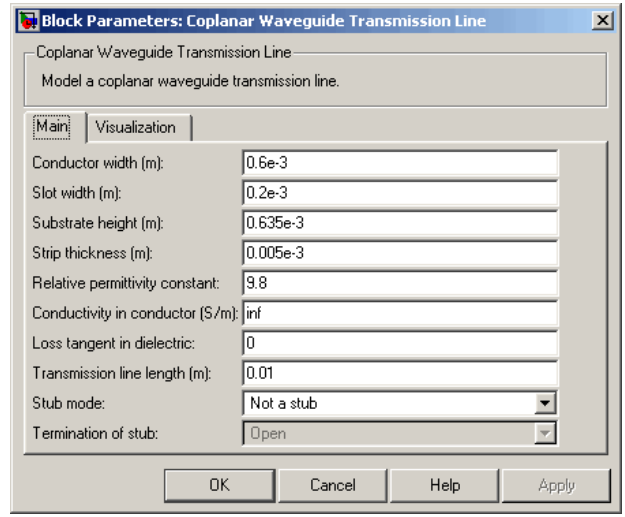



$Z_{in}$ is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## 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 $\varepsilon_0$.

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

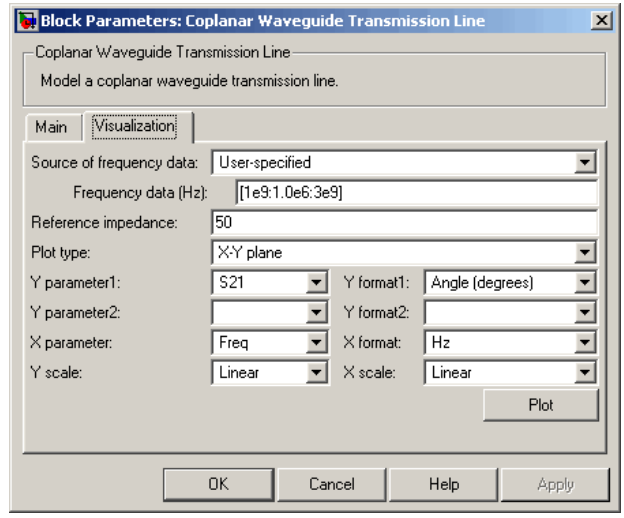
**Loss tangent in dielectric**
Loss angle tangent of the dielectric.

**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

# General Amplifier

## 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value

4 Applying the scaling factor

5 Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

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

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 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

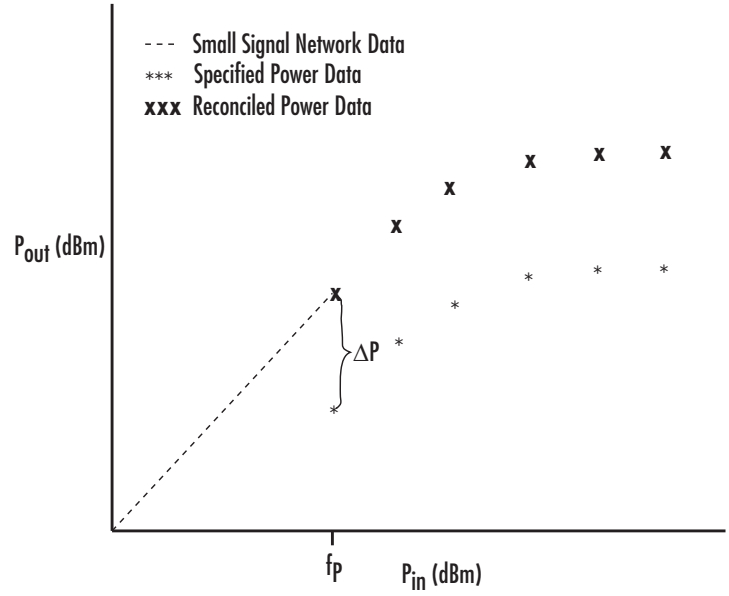
If the data source is a MathWorks AMP file or an Agilent S2D file that contains both network parameter data and power data, RF Blockset checks the data for consistency and reconciles it as necessary.

RF 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:

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

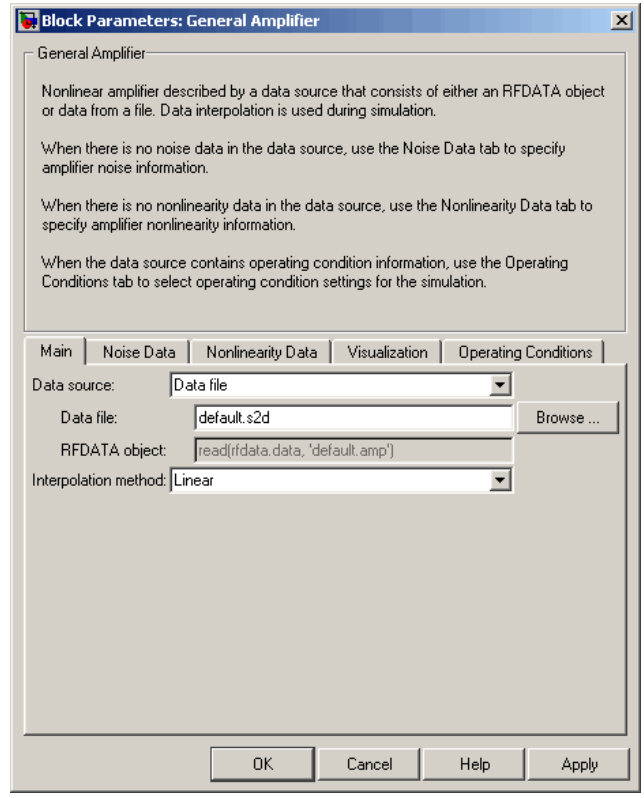
where $f_P$ is the lowest frequency for which power data is specified.

If $\Delta$P is more than 0.4 dB, a warning appears, and RF Blockset adds $\Delta$P to the output power values at each specified input power value to resolve the discrepancy for simulation. This discrepancy is shown in the following graph.


--- Small Signal Network Data
*** Specified Power Data
xxx Reconciled Power Data
Pout (dBm)
ΔP
fP
Pin (dBm)

## 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**

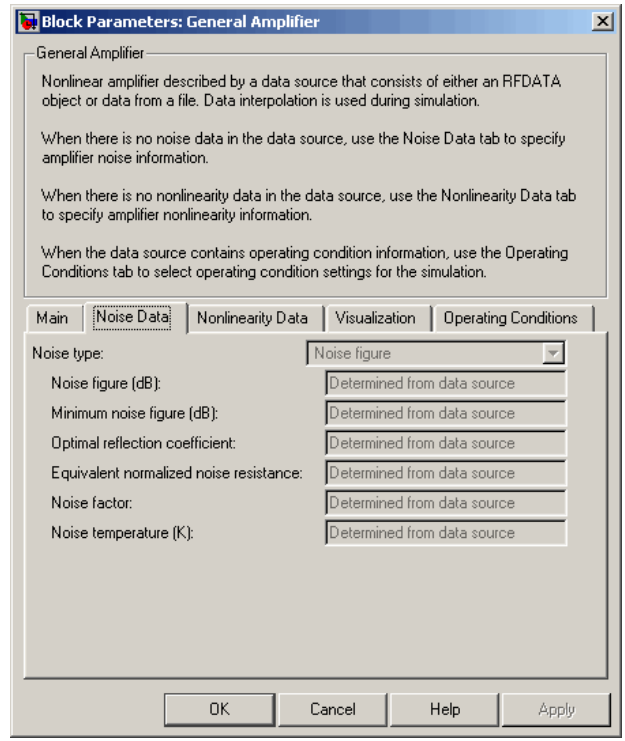
If **Data source** is set to `RFDATA object`, use this field to specify an RF Toolbox data (`rfdata.data`) object that describes an amplifier. You can specify the object as (1) the handle of a data object previously created using RF Toolbox, (2) an RF Toolbox command such as `rfdata.data('Freq',1e9,'S_Parameters',[0 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`.

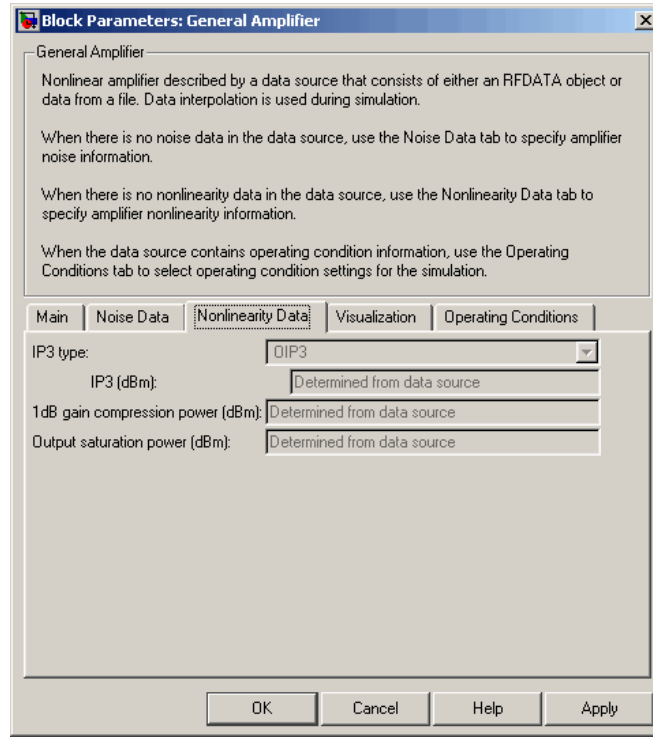
**Noise factor**

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.

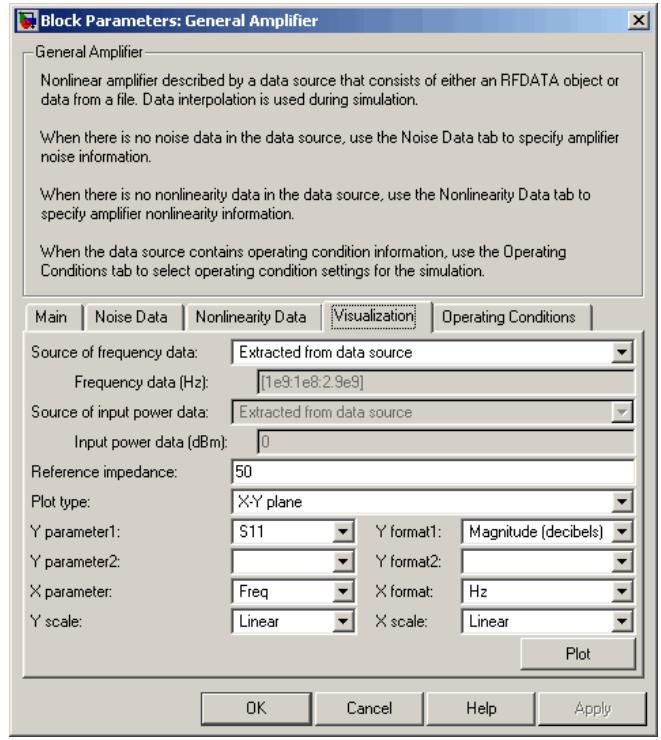
**1 dB gain compression power (dBm)**

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)**

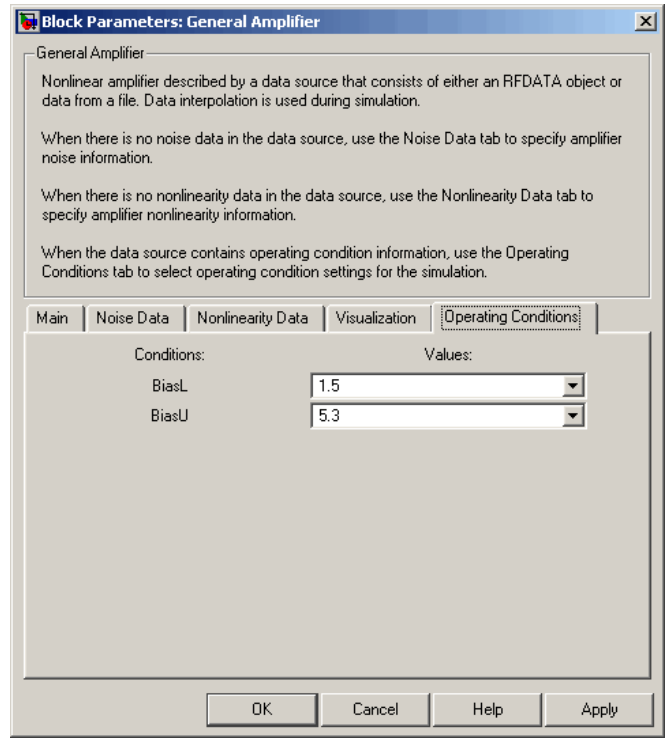
Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if **1 dB gain compression power (dBm)** is specified. Use the default value, `Inf`, if saturation power is unknown.

## 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



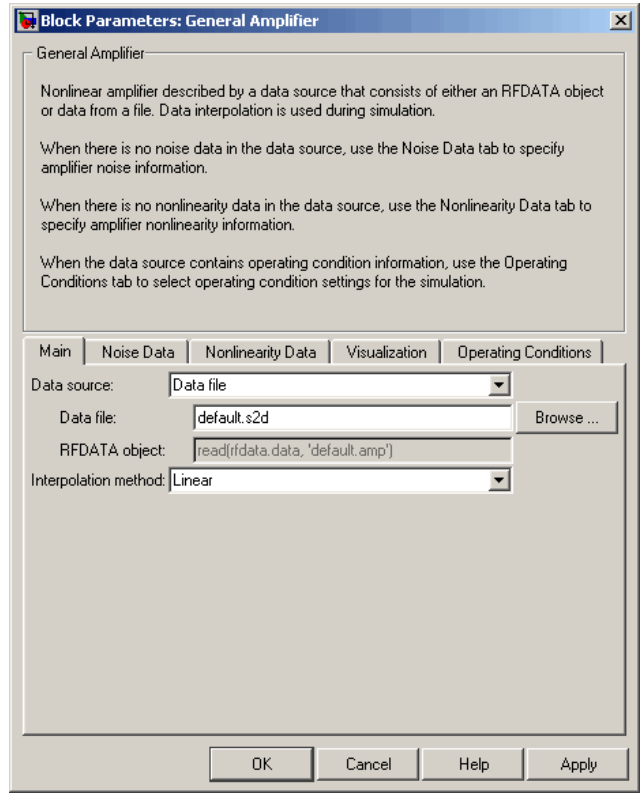

If the data source contains data at multiple operating conditions, the **Operating Conditions** tab contains two columns. The **Conditions** column shows the available conditions, and the **Values** column contains a drop-down list of the available values for the corresponding condition. Use the drop-down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

## Examples

### Creating a General Amplifier Block from File Data

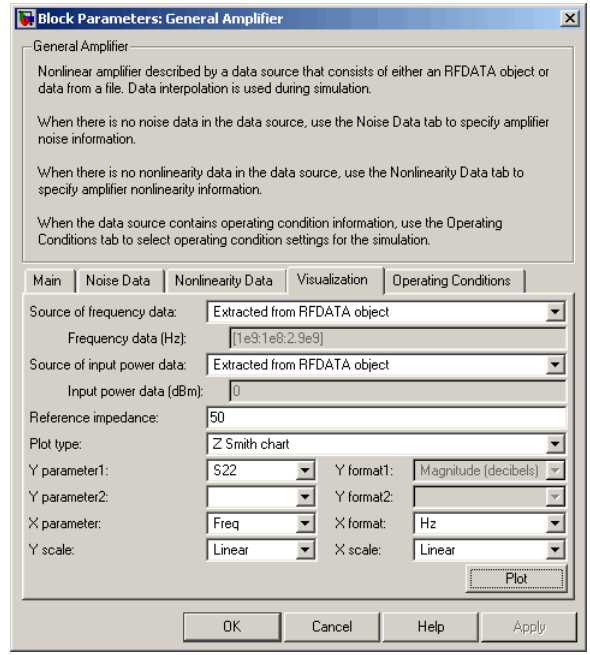
This example uses the default data source, which is the nonlinear amplifier in the file `default.s2d`. The file contains S-parameters for frequencies from 1.0 to 2.9 GHz at intervals of 0.01 GHz, power data at frequency 2.1 GHz, and active noise parameters. By default, the General Amplifier block uses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

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

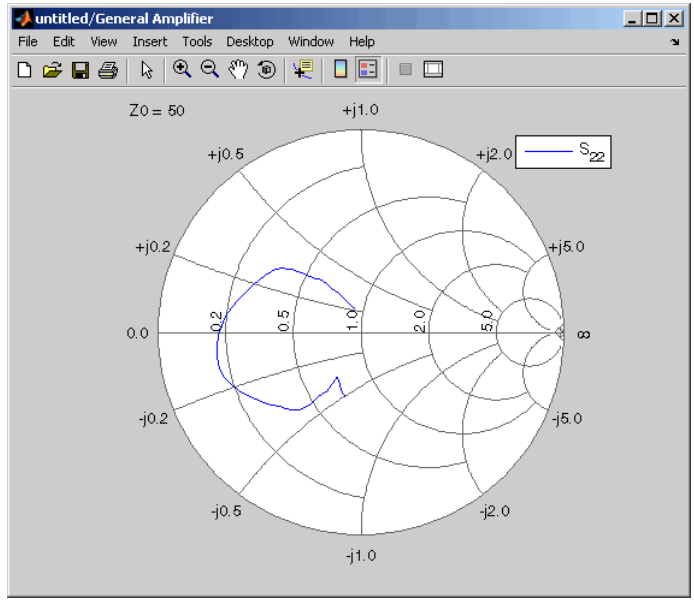



2 On the **Visualization** tab, set the parameters as follows:

- 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.




## See Also

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

`rfdata.data` (RF Toolbox)

`interp1` (MATLAB)

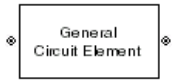
## Purpose

Model two-port network described by `rfckt` object

## Library

Black Box Elements sublibrary of the Physical library

## Description

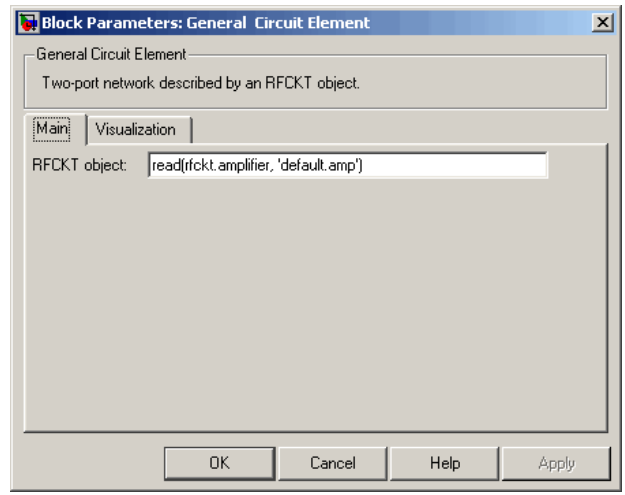



The General Circuit Element block models the two-port network described by an RF Toolbox circuit (`rfckt`) object.

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

## Dialog Box

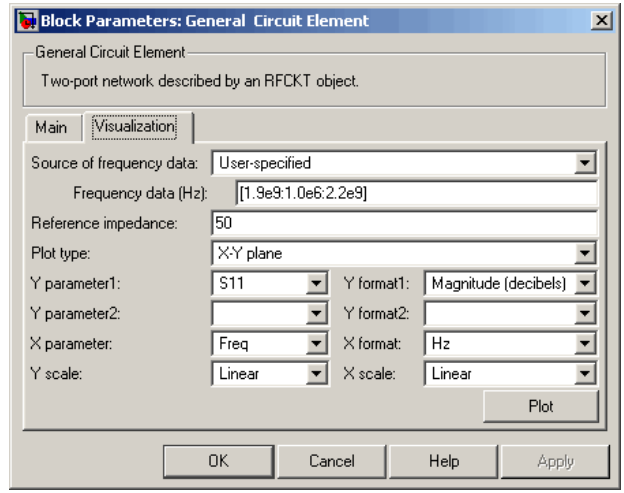
### Main Tab




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

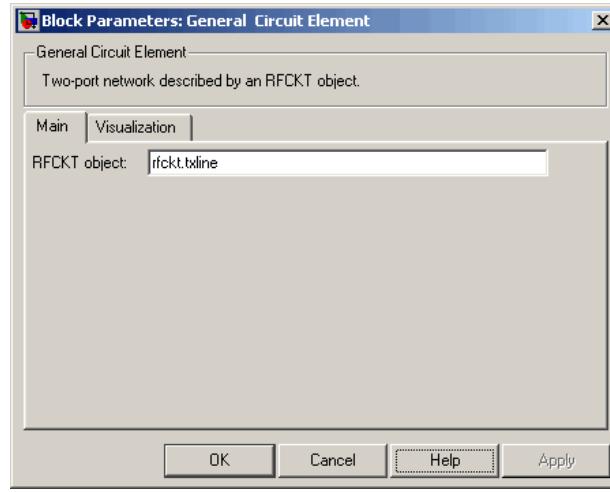
## Examples

### Creating a General Circuit Element from an RF Toolbox Object

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

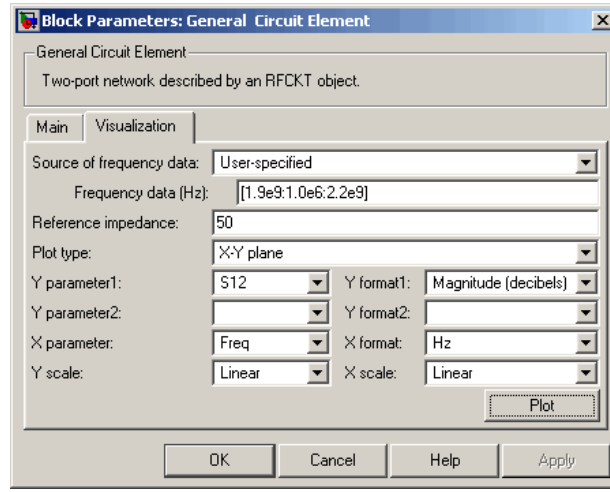
1 On the **Main** tab, set the **RFCKT object** parameter to `rfckt.txline`.

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

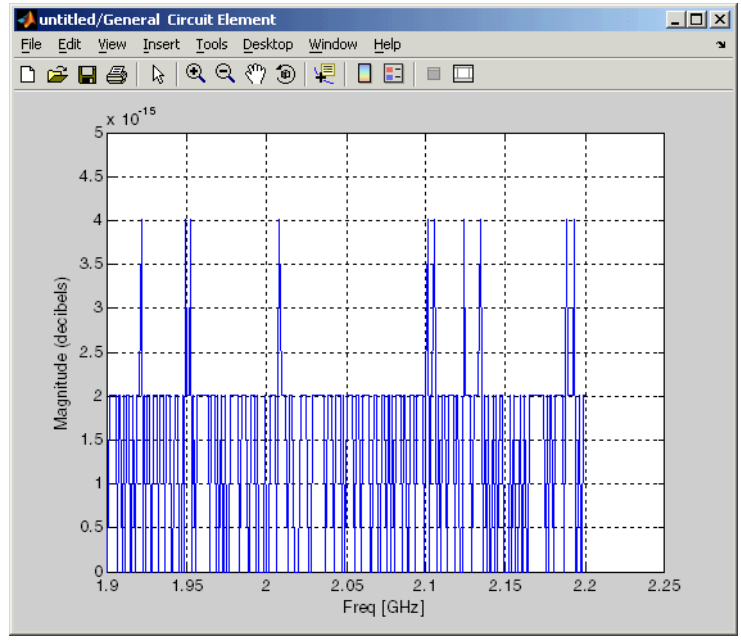



**2** Set the General Circuit Element block parameters on the **Visualization** tab as follows:

- 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)

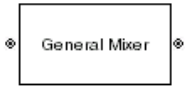
## Purpose

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-4 for more details.

RF Blockset 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.

RF 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, RF 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

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

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

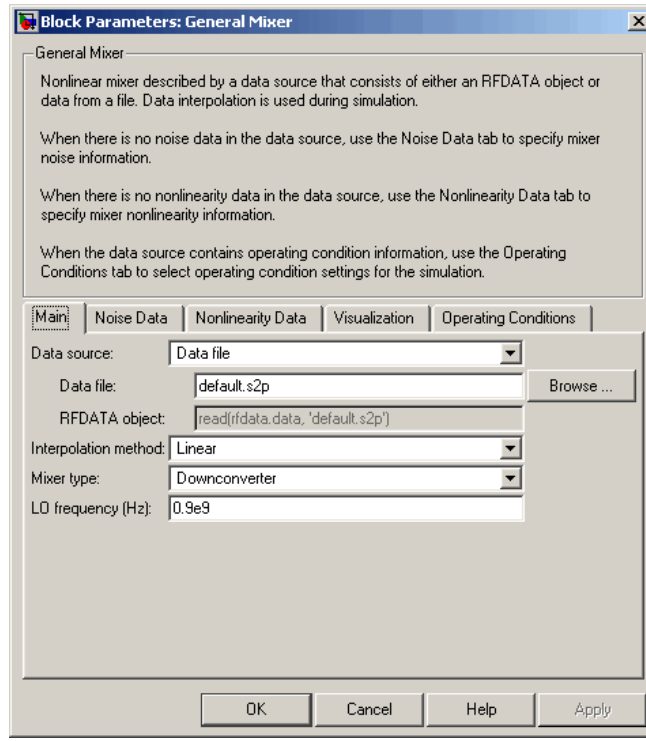
**5** Applying the AM/AM conversion to the input signal

## 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, RF 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. Therefore, it is recommended that you use RF Toolbox 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 (1) the handle of a data object previously created using RF Toolbox, (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 |

**Mixer Type**

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

**LO frequency (Hz)**

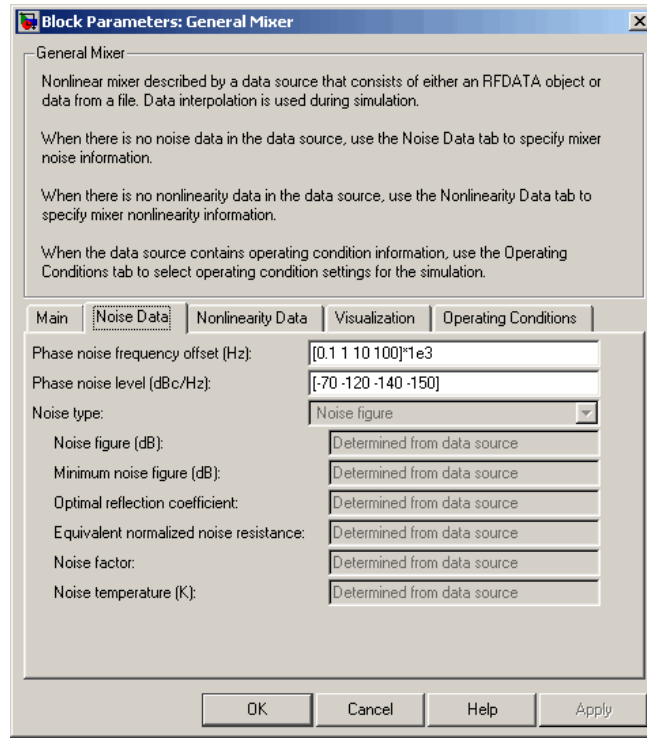
Local oscillator frequency. If you choose `Downconverter`, RF Blockset computes the mixer output frequency, $f_{out}$, from the mixer input frequency, $f_{in}$, and the local oscillator frequency, $f_{lo}$, as $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

---

**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`.

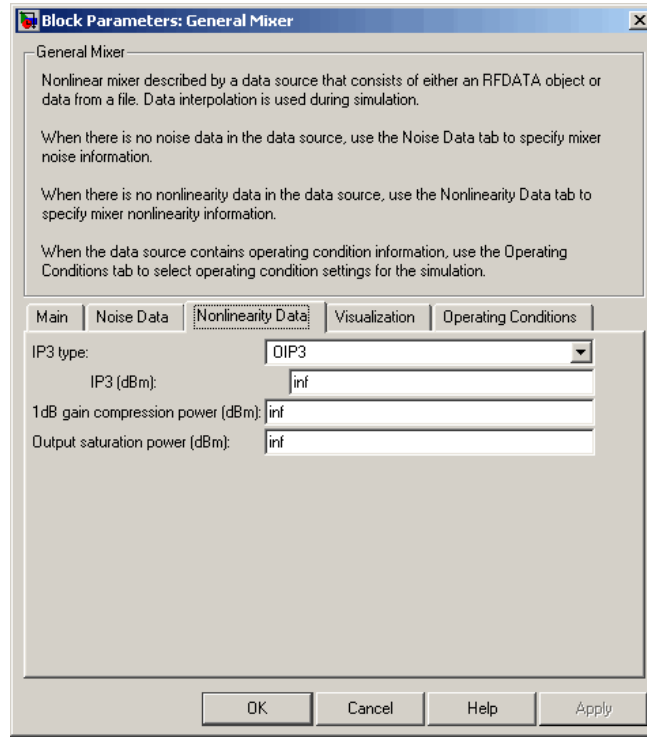
**Noise factor**

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.

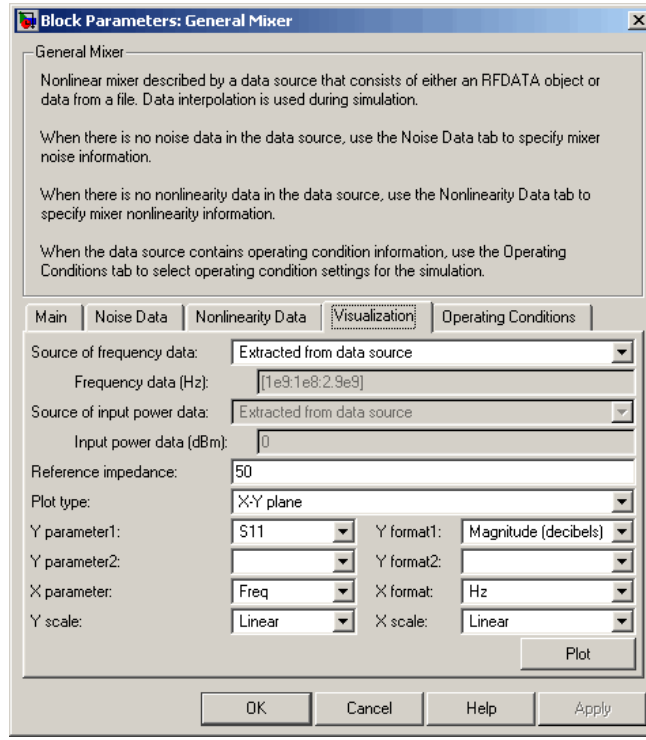
**IP3 (dBm)**

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.

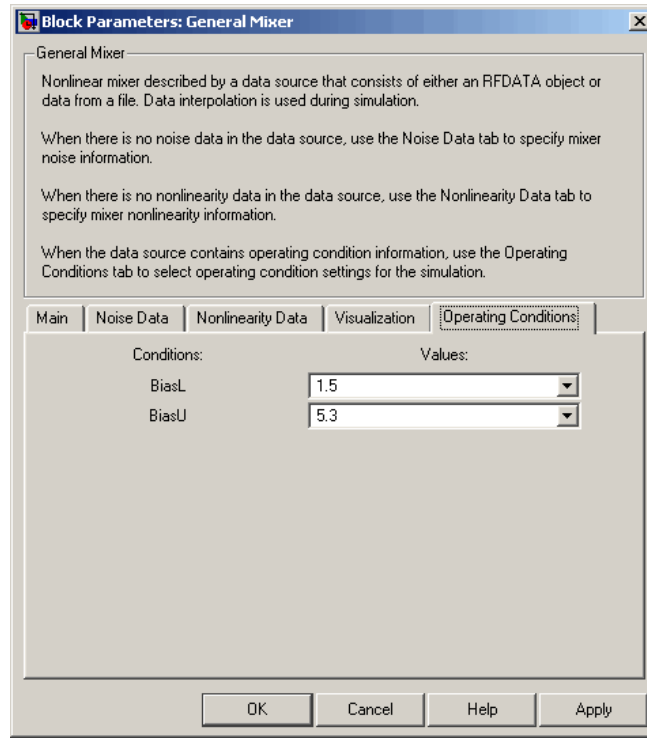
**Output saturation power (dBm)**
Scalar output power value the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Its value is only used if **1 dB gain compression power (dBm)** is specified. Use the default value, `Inf`, if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

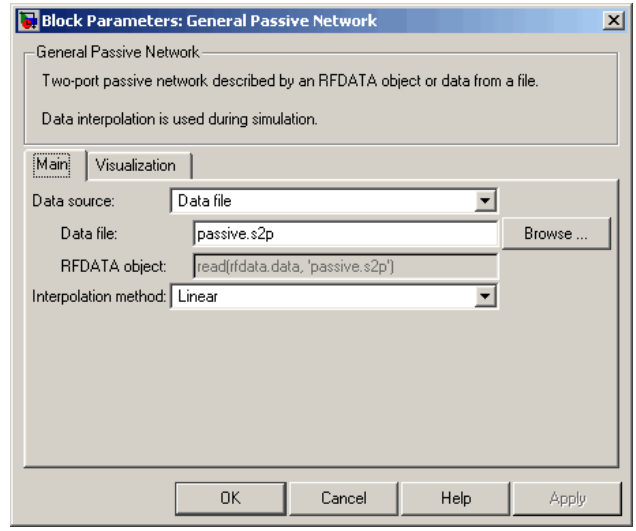
## Description

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-4 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, (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**

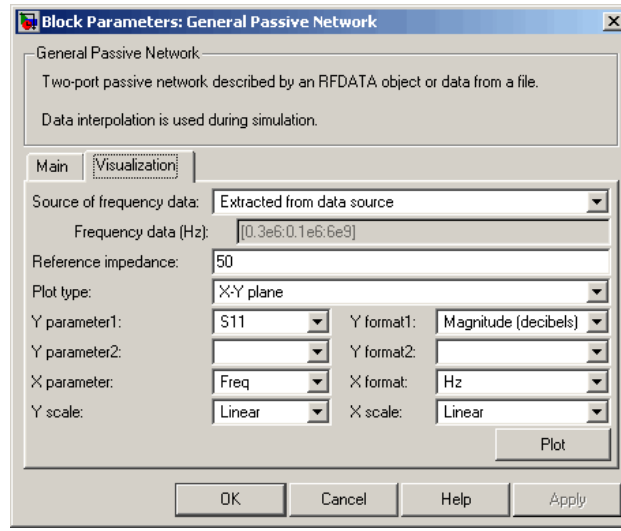
If **Data source** is set to `Data file`, use this field to specify the name of the file that contains the amplifier data. The file name must include the extension. If the file is not in your MATLAB path, specify the full path to the file or use the **Browse** button to find the file.

**Interpolation method**

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

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

### Visualization Tab



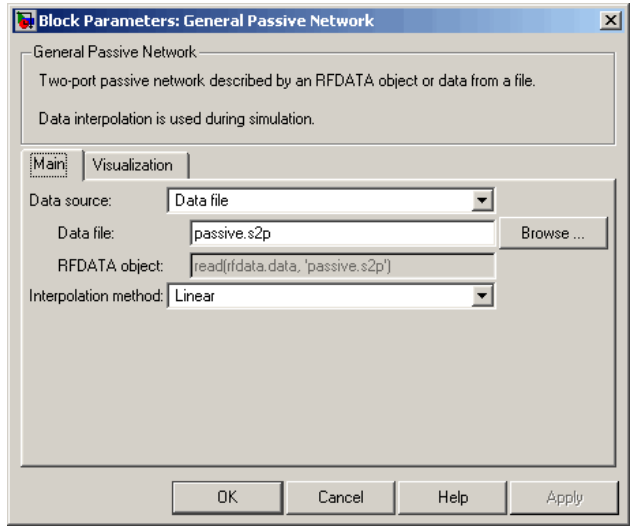

For information about plotting circuit parameters, see Chapter 3, “Plotting Model Data”.

## Examples

### Creating a General Passive Network Block from File Data

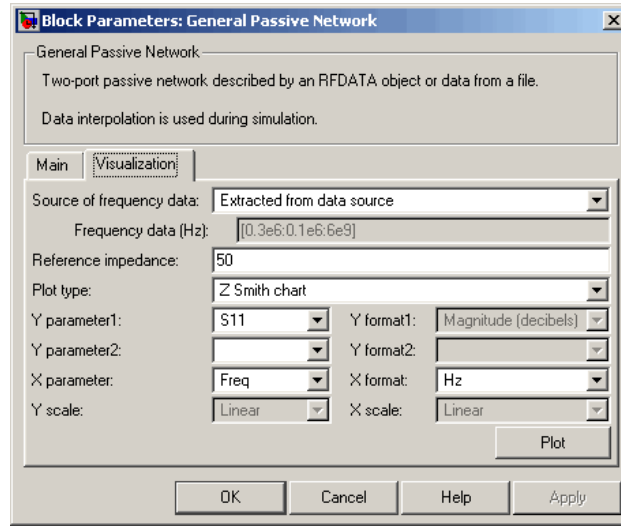
This example creates a two-port passive network from the data in the file `passive.s2p`. The file contains S-parameters for frequencies from about 0.315 MHz to 6.0 GHz. The General Passive Network block uses linear interpolation to model the network described in the object.

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

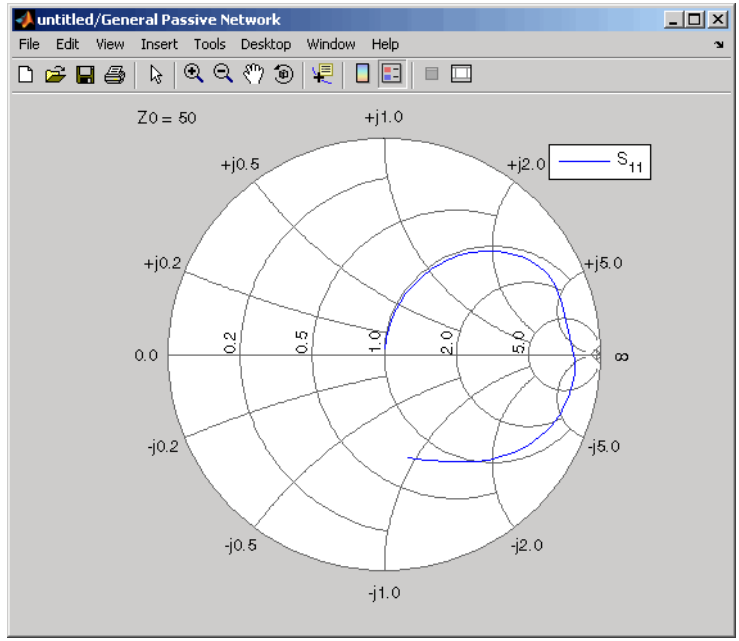



**2** On the **Visualization** tab, set the parameters as follows:

- 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)

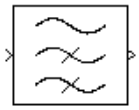
## Purpose

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.

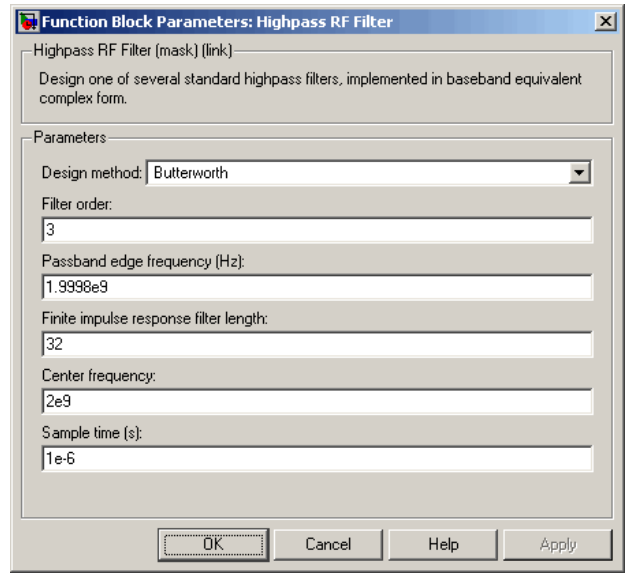
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block lets you specify the filter design parameters shown in the following table.

| 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)

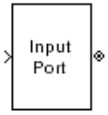
## Purpose

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.

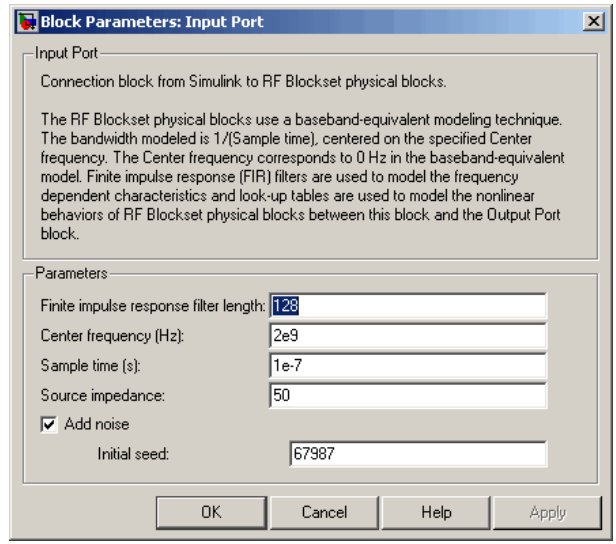
For more information about how the Input Port block converts the mathematical Simulink signals to RF Blockset physical modeling environment signals, see “Converting to and from Simulink Signals” on page A-17. For more information about connecting mathematical and physical parts of a model, see Chapter 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). For a graphical representation of this parameter, see “Baseband-Equivalent Modeling” on page A-12.

**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.

**Source impedance**

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 block, 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

# LC Bandpass Pi

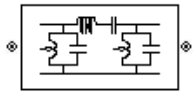
## Purpose

Model LC bandpass pi network

## Library

Ladder Filters sublibrary of the Physical library

## Description



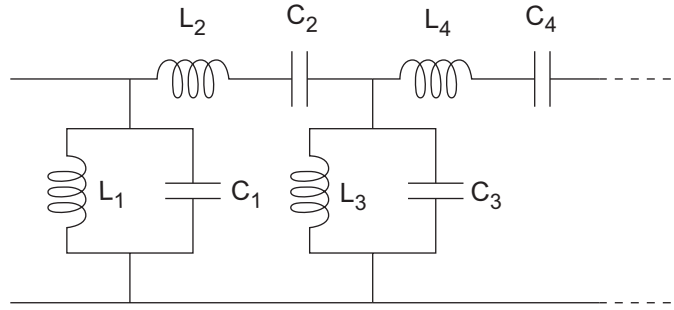
The LC Bandpass Pi block models the LC bandpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

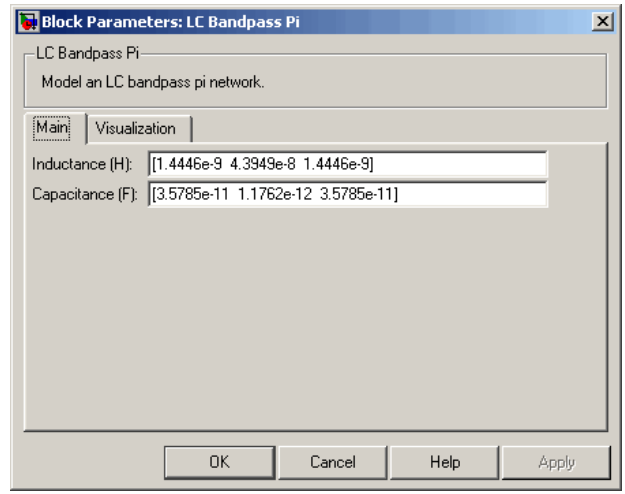
The LC bandpass pi network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, $L_3$, $L_4$, ...] 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



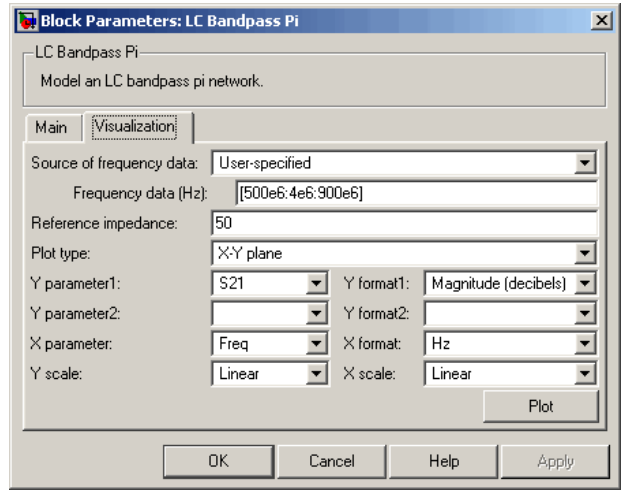

**Inductance (H)**

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

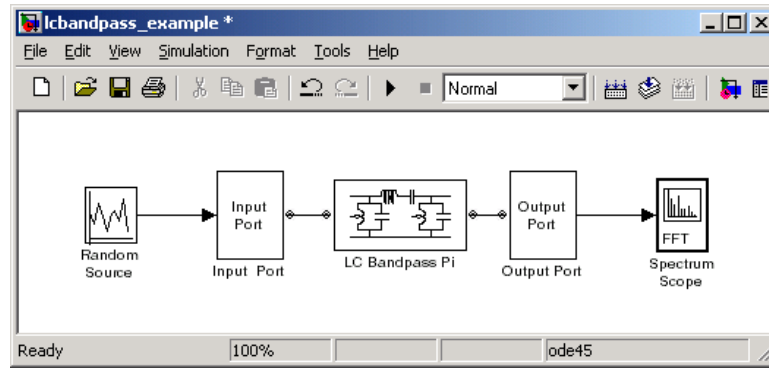



For information about plotting the filter parameters, see Chapter 3, “Plotting Model Data”.

## Examples

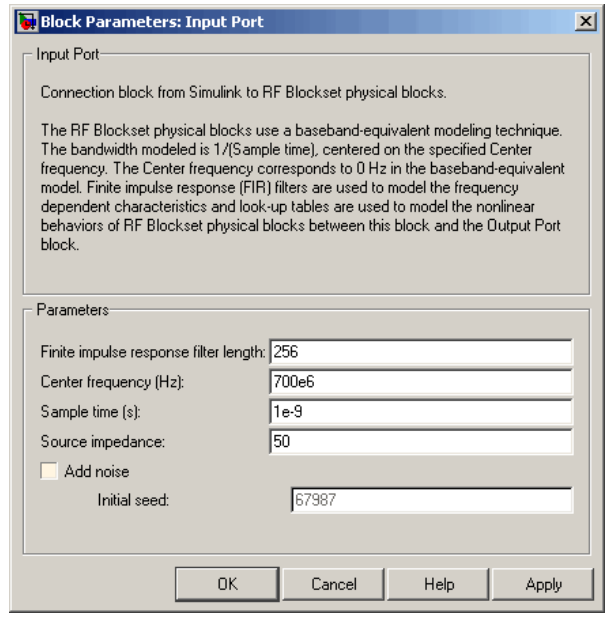
### Using a Ladder Filter Block to Filter Gaussian Noise

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

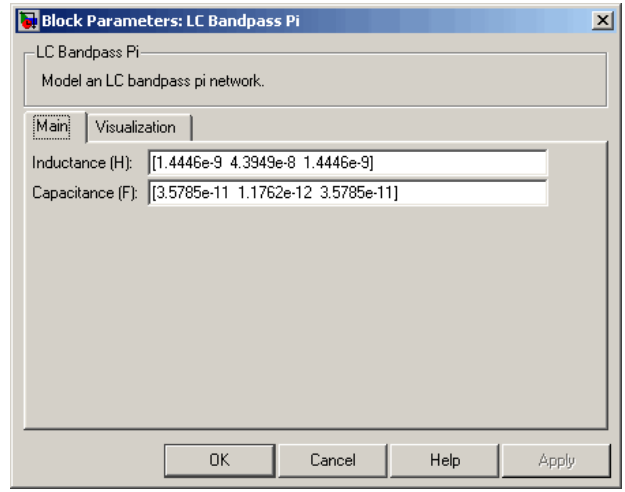



The Random Source block (Signal Processing Blockset) produces frame-based output at 512 samples per frame. Its **Sample time** parameter is set to 1.0e-9. This sample time must match the sample time for the physical part of the model, which you provide in the Input Port block diagram.

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.

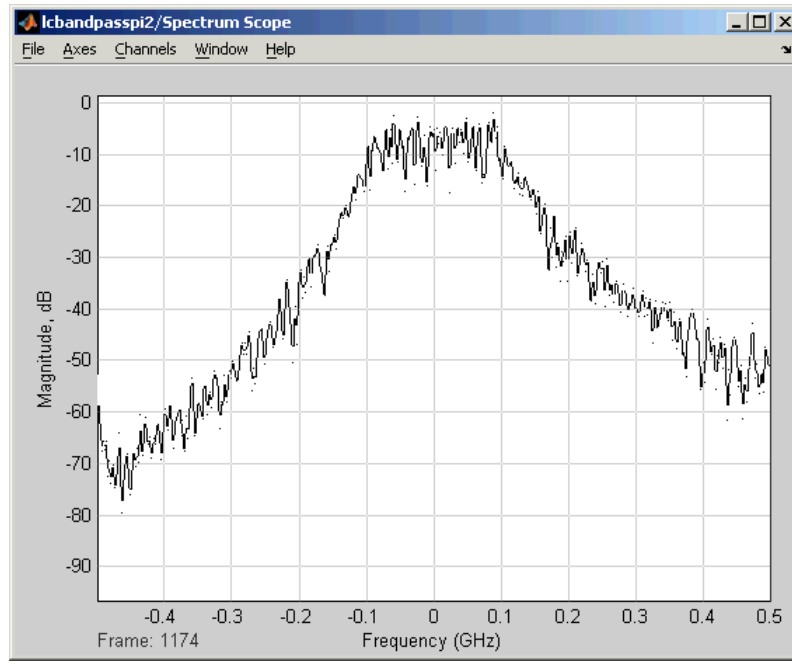



The LC Bandpass Pi block provides the inductances for three inductors, in order from source to load, [1.4446e-9, 4.3949e-8, 1.4446e-9]. Similarly, it provides the capacitances for three capacitors [3.5785e-11, 1.1762e-12, 3.5785e-11].




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 RLC, Shunt RLC

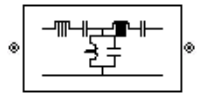
## Purpose

Model LC bandpass tee network

## Library

Ladder Filters sublibrary of the Physical library

## Description



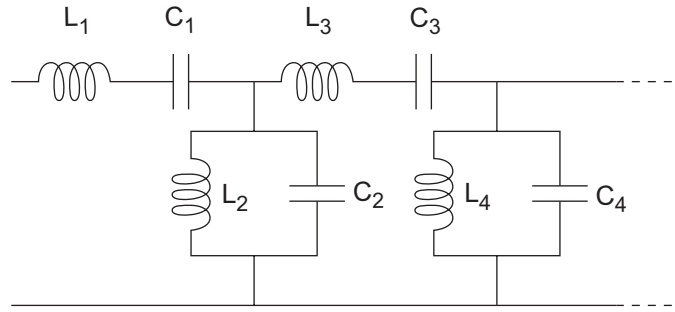
The LC Bandpass Tee block models the LC bandpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

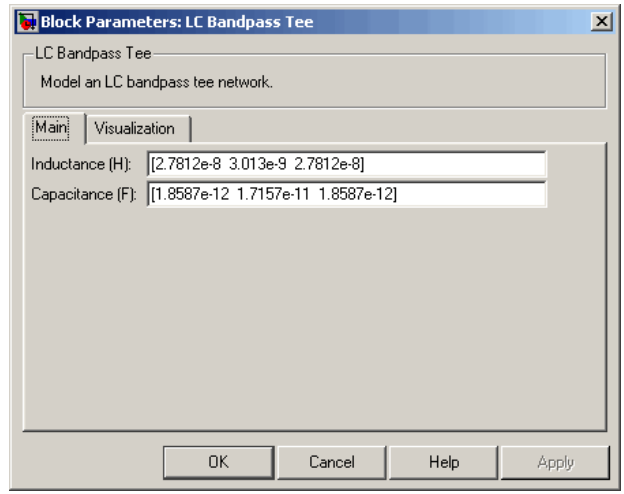
The LC bandpass tee network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, $L_3$, $L_4$, ...] 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



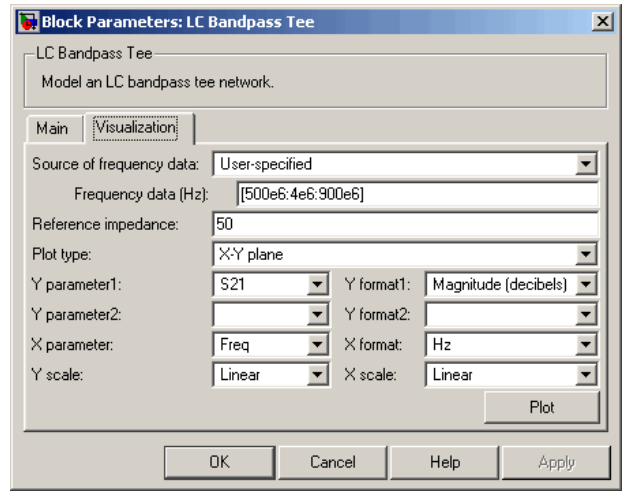

**Inductance (H)**

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 RLC, Shunt RLC

# LC Bandstop Pi

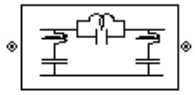
## Purpose

Model LC bandstop pi network

## Library

Ladder Filters sublibrary of the Physical library

## Description



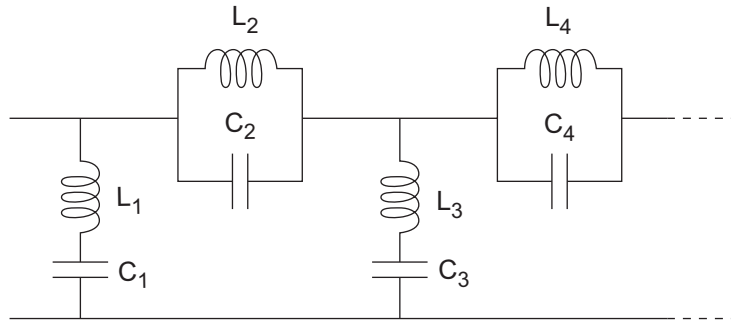
The LC Bandstop Pi block models the LC bandstop pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

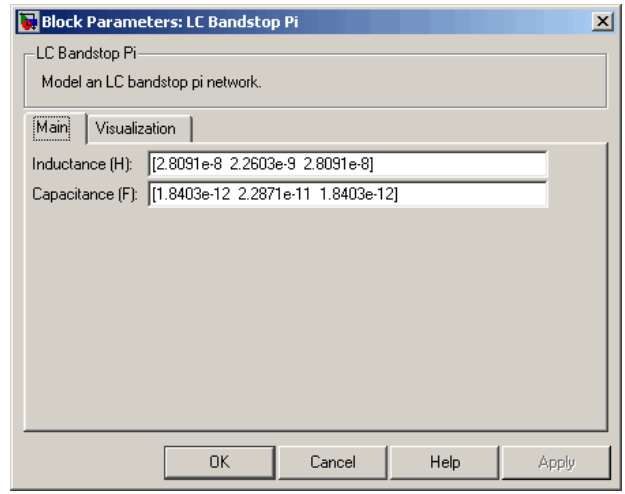
The LC bandstop pi network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, $L_3$, $L_4$, ...] 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



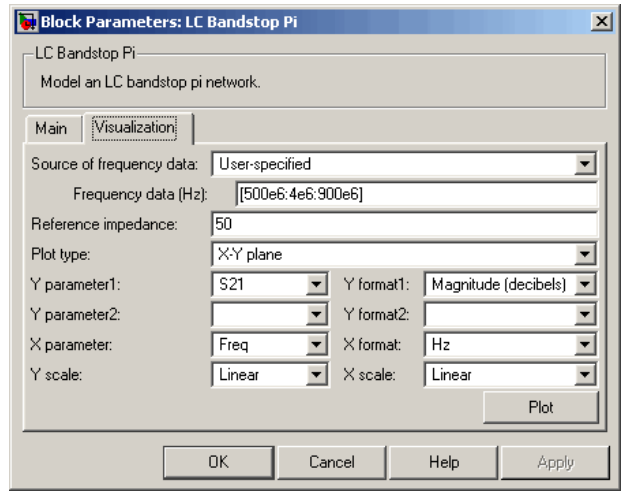

**Inductance (H)**

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 RLC, Shunt RLC

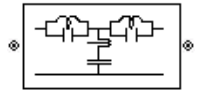
## Purpose

Model LC bandstop tee network

## Library

Ladder Filters sublibrary of the Physical library

## Description



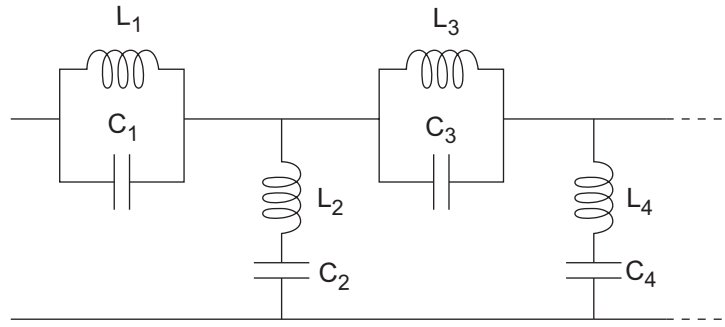
The LC Bandstop Tee block models the LC bandstop tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 parameters to S-parameters using the RF Toolbox `abcd2s` function.

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

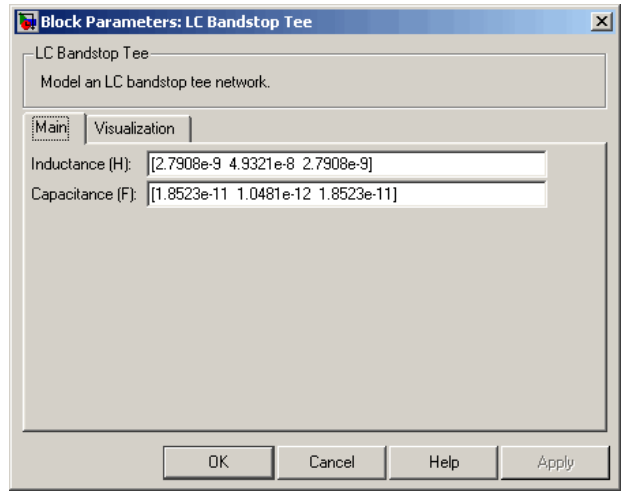
The LC bandstop tee network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, $L_3$, $L_4$, ...] 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



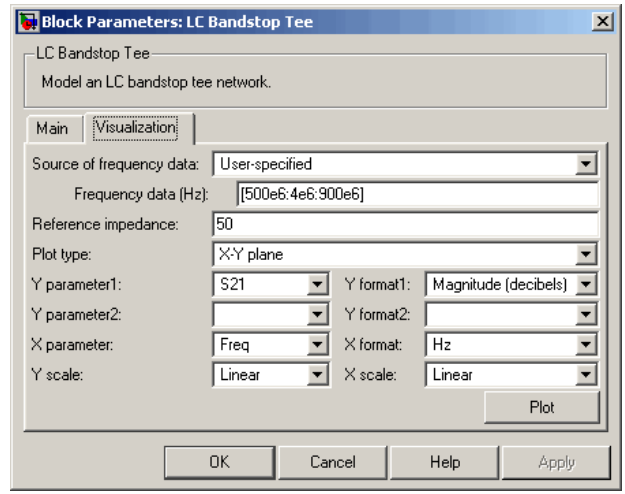

**Inductance (H)**

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 RLC, Shunt RLC

# LC Highpass Pi

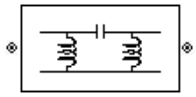
## Purpose

Model LC highpass pi network

## Library

Ladder Filters sublibrary of the Physical library

## Description



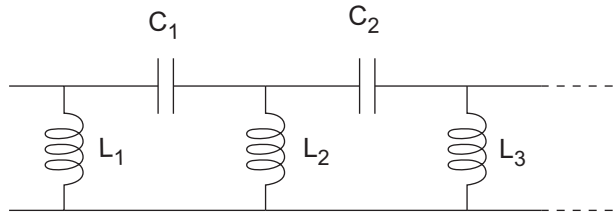
The LC Highpass Pi block models the LC highpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

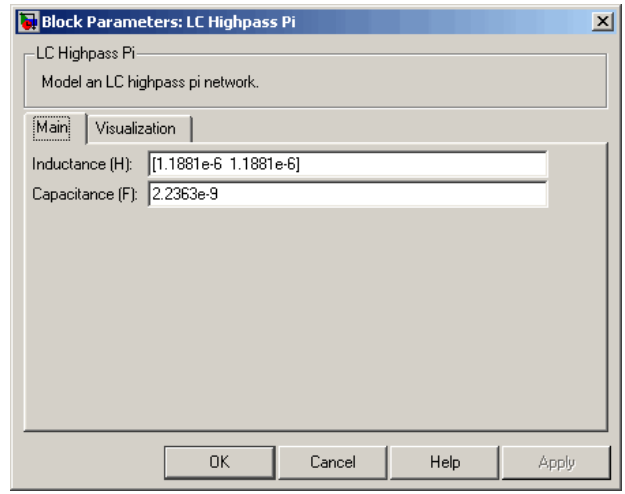
The LC highpass pi network object is a two-port network as shown in the following circuit diagram.




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

## Dialog Box

### Main Tab



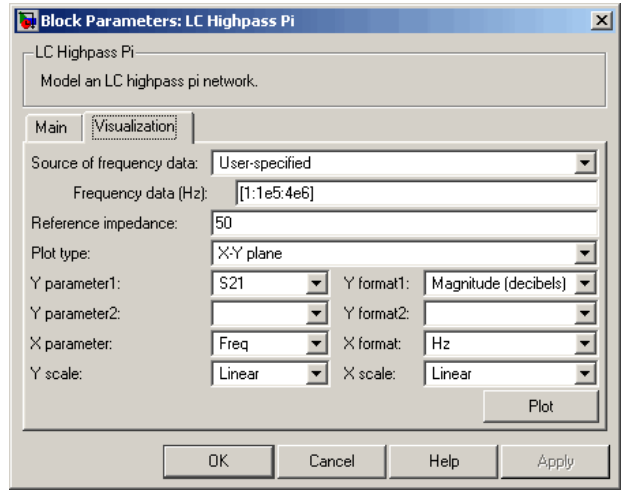

**Inductance (H)**

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 RLC, Shunt RLC

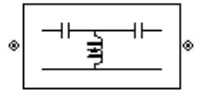
## Purpose

Model LC highpass tee network

## Library

Ladder Filters sublibrary of the Physical library

## Description



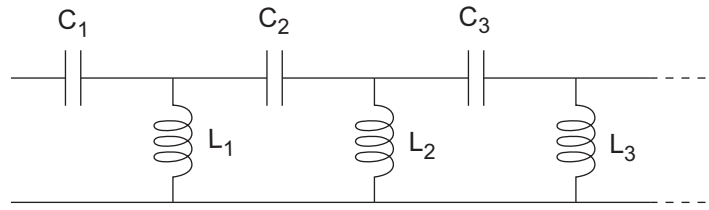
The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

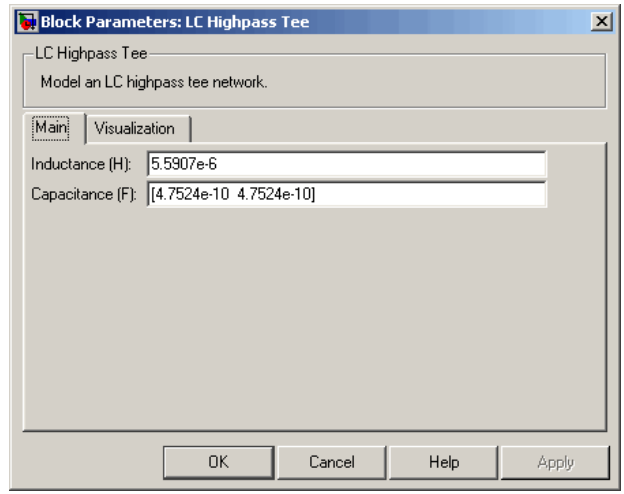
The LC highpass tee network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, $L_3$, ...] 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



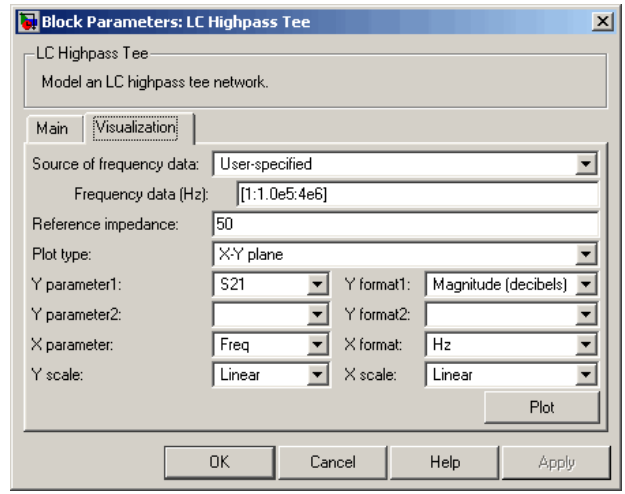

**Inductance (H)**

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 RLC, Shunt RLC

# LC Lowpass Pi

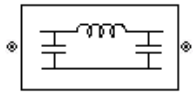
## Purpose

Model LC lowpass pi network

## Library

Ladder Filters sublibrary of the Physical library

## Description



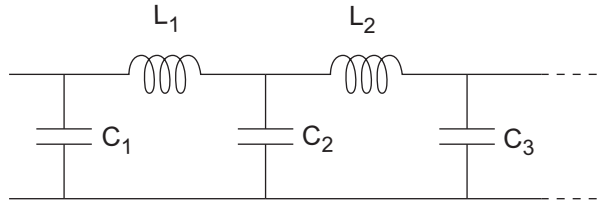
The LC Lowpass Pi block models the LC lowpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

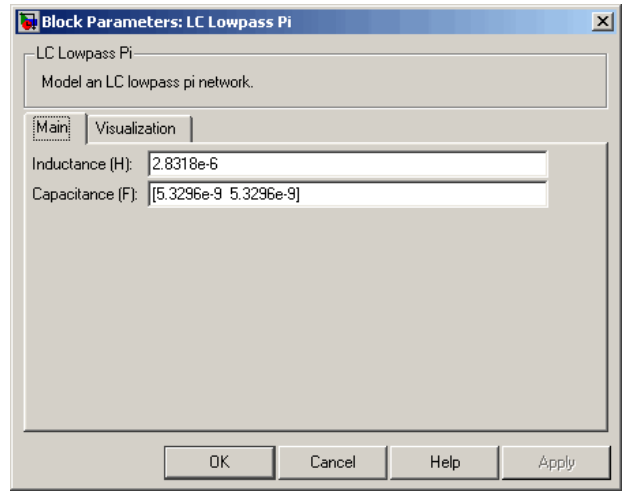
The LC lowpass pi network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, ...] 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



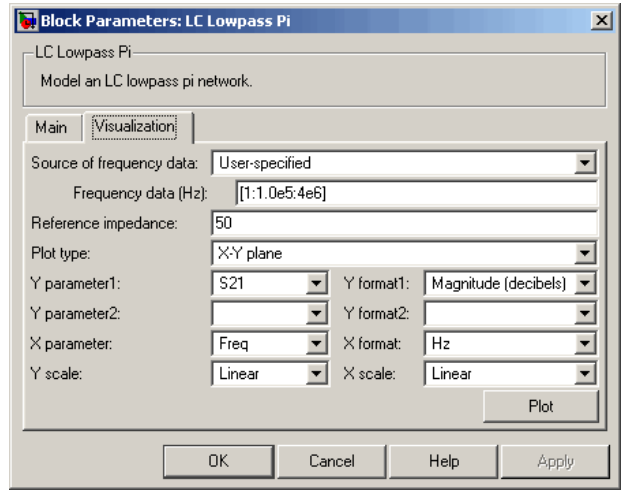

**Inductance (H)**

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 RLC, Shunt RLC

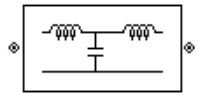
## Purpose

Model LC lowpass tee network

## Library

Ladders Filters sublibrary of the Physical library

## Description



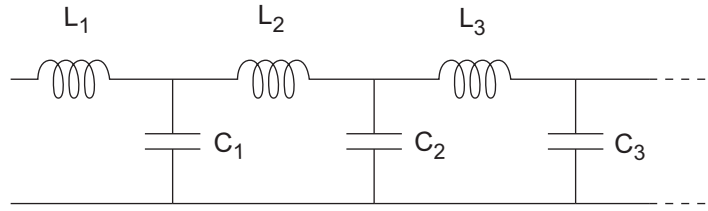
The LC Lowpass Tee block models the LC lowpass tee network described in the block dialog box in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each 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 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.

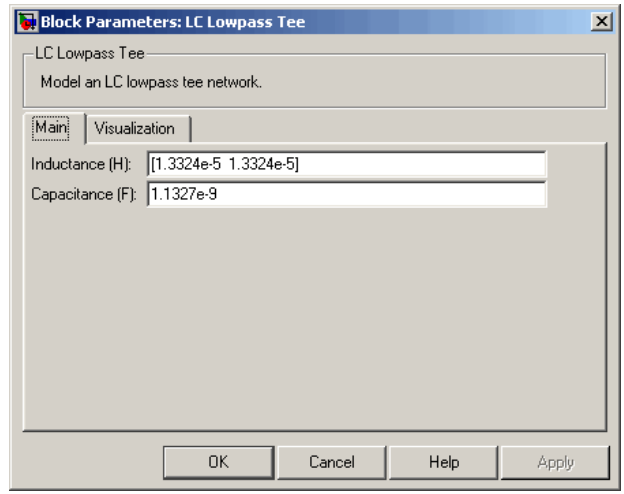
The LC lowpass tee network object is a two-port network as shown in the following circuit diagram.




[$L_1$, $L_2$, $L_3$, ...] 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



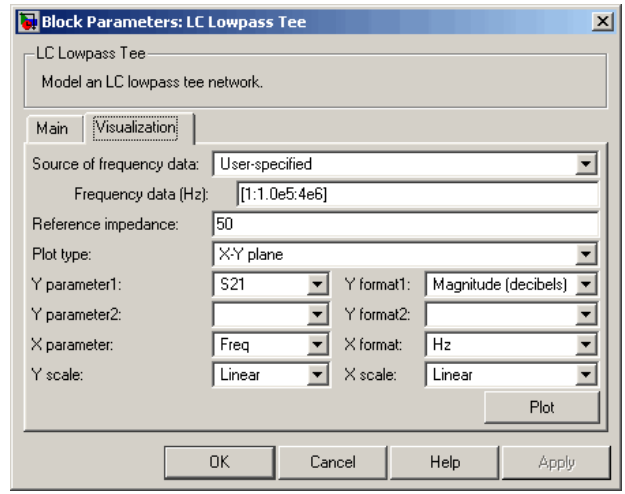

**Inductance (H)**

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 RLC, Shunt RLC

# Lowpass RF Filter

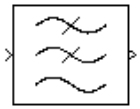
## Purpose

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.

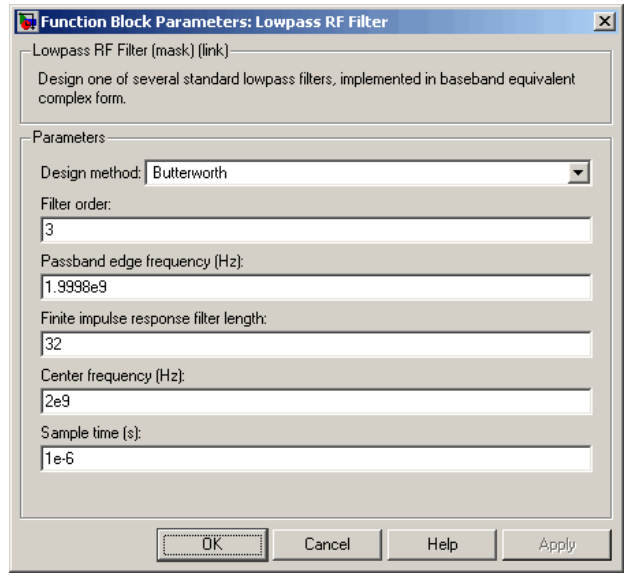
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

| 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)

# Microstrip Transmission Line

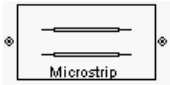
## Purpose

Model microstrip transmission line

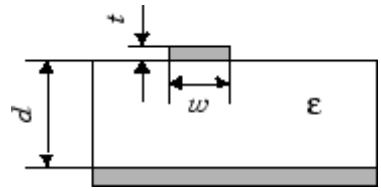
## Library

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 ($\varepsilon$).




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 calculates the frequency-dependent S-parameters using the physical length of the transmission line, $D$, and the complex propagation constant, $k$.

$$S_{11} = 0$$

$$S_{12} = e^{-kD}$$

$$S_{21} = e^{-kD}$$

$$S_{22} = 0$$

$k = \alpha_a + i\beta$, where $\alpha_a$ is the attenuation coefficient and $\beta$ is the wave number. The attenuation coefficient $\alpha_a$ is related to the loss, $\alpha$, by

$$\alpha_a = -\ln 10^{\frac{\alpha}{20}}$$

where $\alpha$ is the reduction in signal strength, in dB, per unit length. $\alpha$ combines both conductor loss and dielectric loss and is derived from the physical parameters specified in the Microstrip Transmission Line block dialog box. The wave number $\beta$ is related to the phase velocity, $V_P$, by

$$\beta = \frac{2\pi f}{V_p}$$

where $V_P = c/\sqrt{\varepsilon_{\text{eff}}}$, and $\varepsilon_{\text{eff}}$ is the frequency dependent effective dielectric constant. $f$ is the vector of modeling frequencies determined by the Output Port block. The phase velocity $V_P$ is also known as the wave propagation velocity.

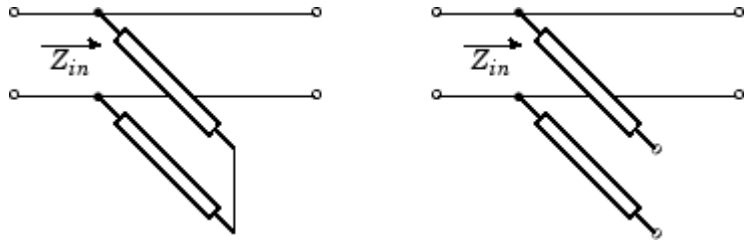
The Microstrip Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

### 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.

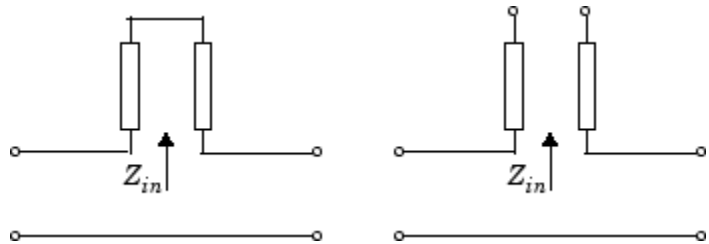



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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.

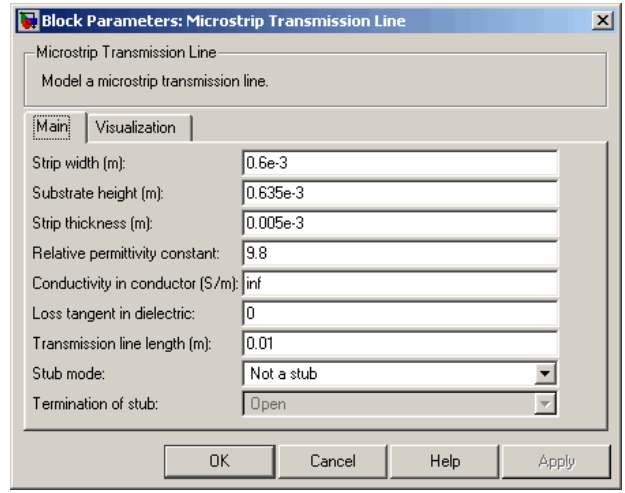



$Z_{in}$ is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## 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 $\varepsilon_0$.

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

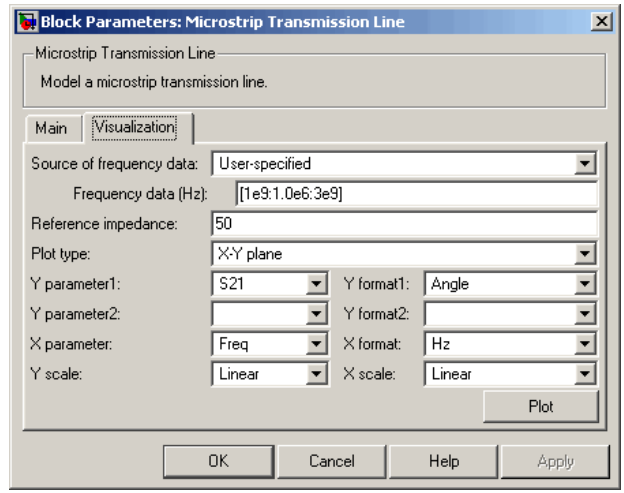
**Loss tangent in dielectric**
Loss angle tangent of the dielectric.

**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

# Mixer

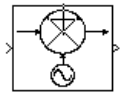
## Purpose

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 baseband-complex 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.

---

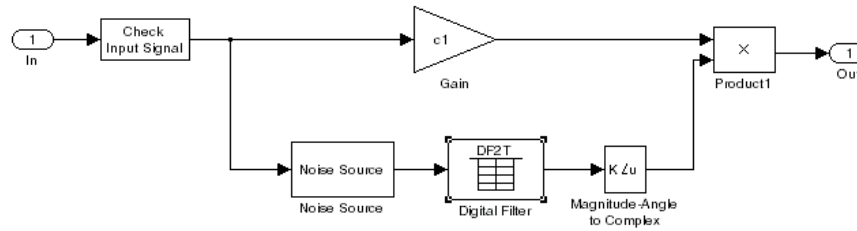
**Note** This block assumes a nominal impedance of 1 ohm.

---

The block applies the phase noise to the signal as follows:

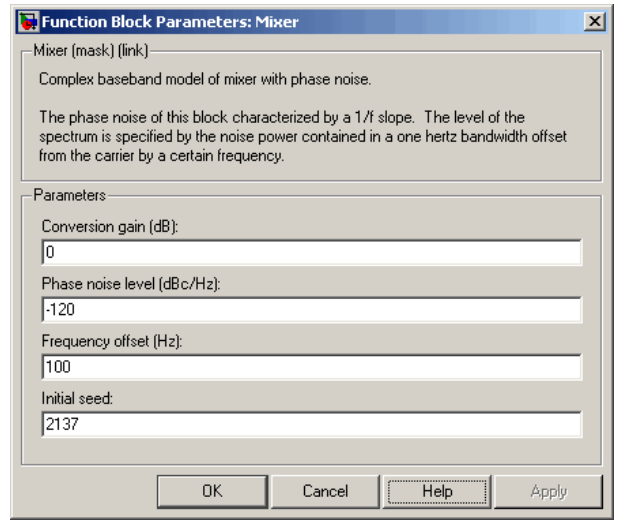
1. Generates additive white Gaussian noise (AWGN) and filters it with a digital filter.
2. Adds the resulting phase noise to the angle component of the input signal.

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 the construction of the Noise Source subsystem by double-clicking 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

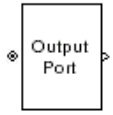
## Purpose

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-17. 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 $n$th 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,

   $$H(f) = \frac{S_{21} * (1+\Gamma_l) * (1-\Gamma_s)}{2 * (1-S_{22} * \Gamma_l)(1-\Gamma_{in} * \Gamma_s)}$$

where

$$\Gamma_l = \frac{Z_l - Z_o}{Z_l + Z_o}$$

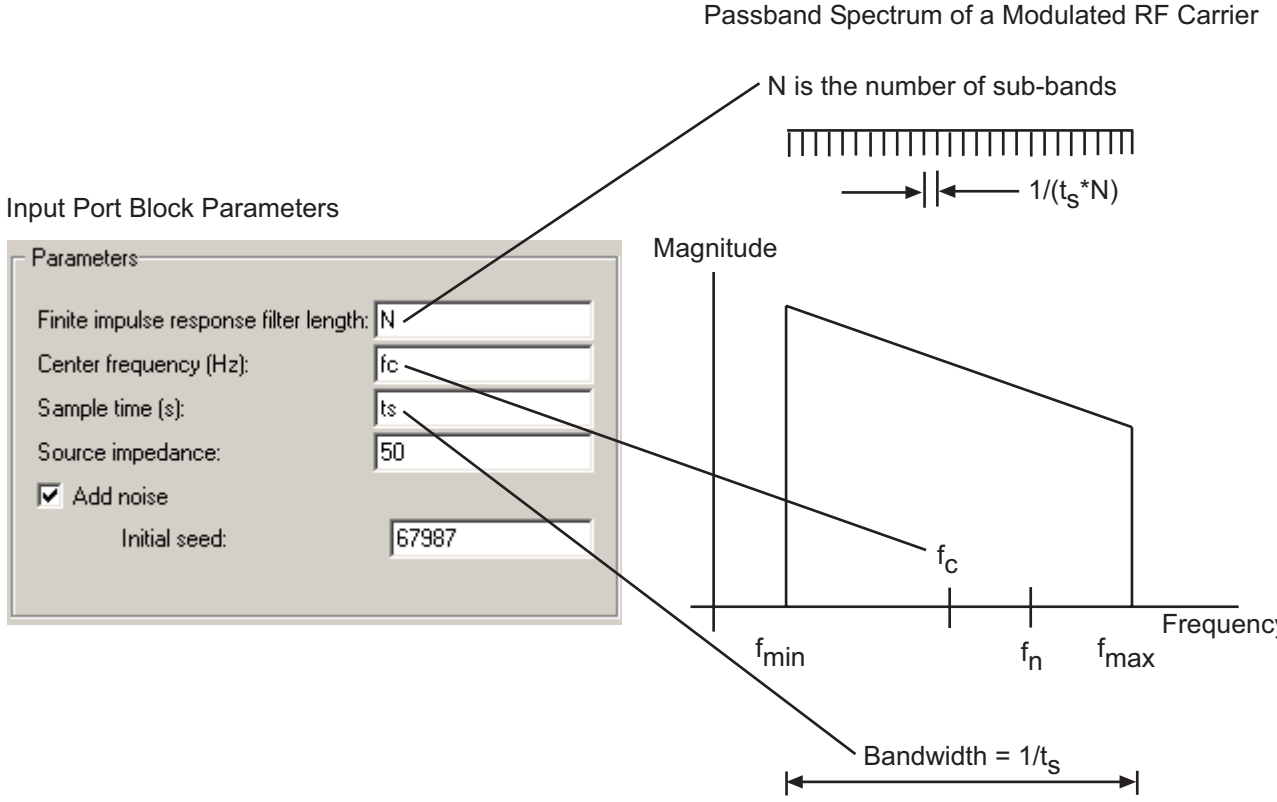
$$\Gamma_s = \frac{Z_s - Z_o}{Z_s + Z_o}$$

$$\Gamma_{in} = S_{11} + \left( S_{12} * S_{21} * \frac{\Gamma_l}{(1 - S_{22} * \Gamma_l)} \right)$$

and

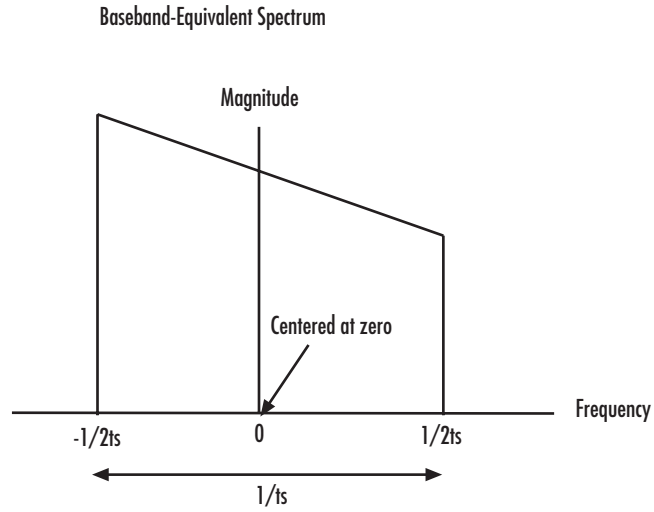
- $Z_S$ is the source impedance.
- $Z_L$ is the load impedance.
- $S_{ij}$ are the S-parameters of a two-port network.

RF Blockset derives the passband transfer function from the Input Port block parameters as shown in the following figure:




**3** Translates the passband transfer function to baseband as $H(f - f_c)$, where $f_c$ is the specified center frequency.

The baseband transfer function is shown in the following figure.

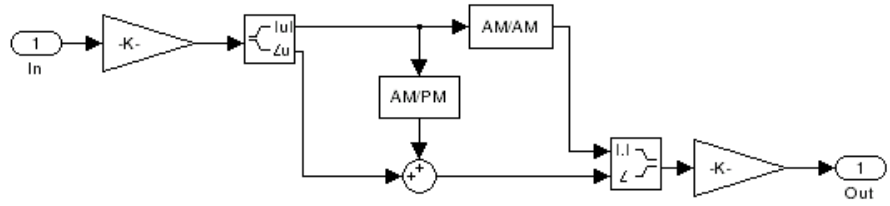



4 Obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. For faster simulation, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. Then, it truncates the impulse response 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

$$AM_{\text{out}} = \sqrt{R_l \cdot P_{\text{out}}}$$

$$PM_{\text{out}} = \text{Phase}$$

$$AM_{\text{in}} = \sqrt{R_s \cdot P_{\text{in}}}$$

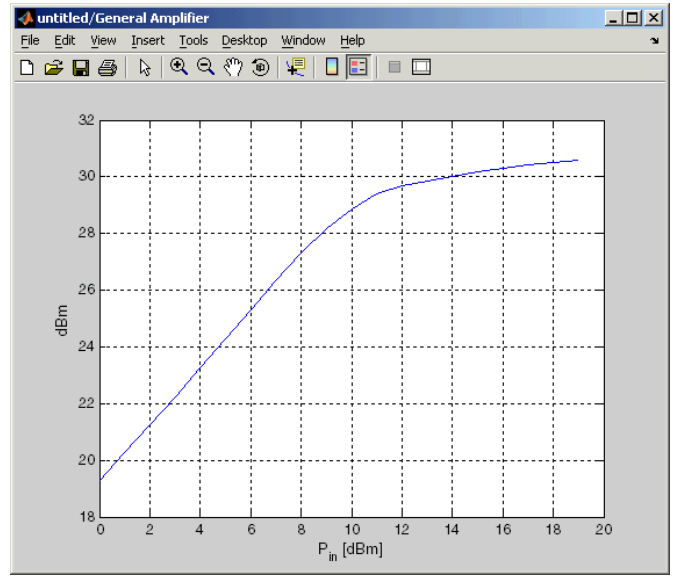
where $AM_{in}$ is the AM of the input voltage, $AM_{out}$ and $PM_{out}$ are the AM and PM of the output voltage, $R_s$, is the source resistance (50 ohms), $R_l$ is the load resistance (50 ohms), $P_{in}$ is the input power, $P_{out}$ is the output power, and *Phase* is the phase shift between the input and output voltage.

---

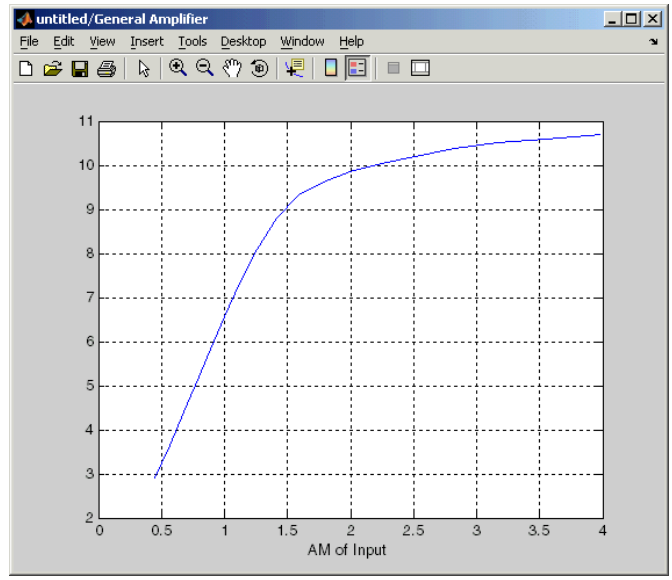
**Note** You can provide power data via a `.amp` file. See "AMP File Format" in the RF Toolbox documentation for information about this format.

---

The following figure shows the original power data of an amplifier.

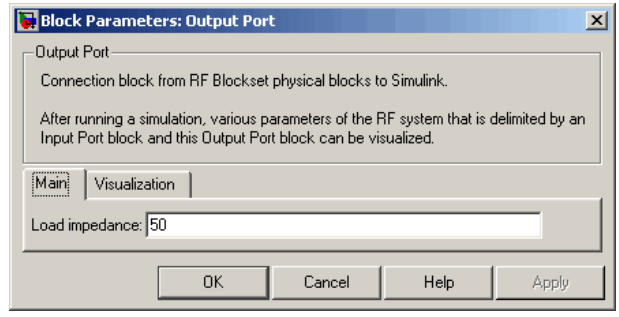


This figure shows the extracted AM/AM nonlinear conversion.




## Dialog Box

### Main Tab




**Load impedance**

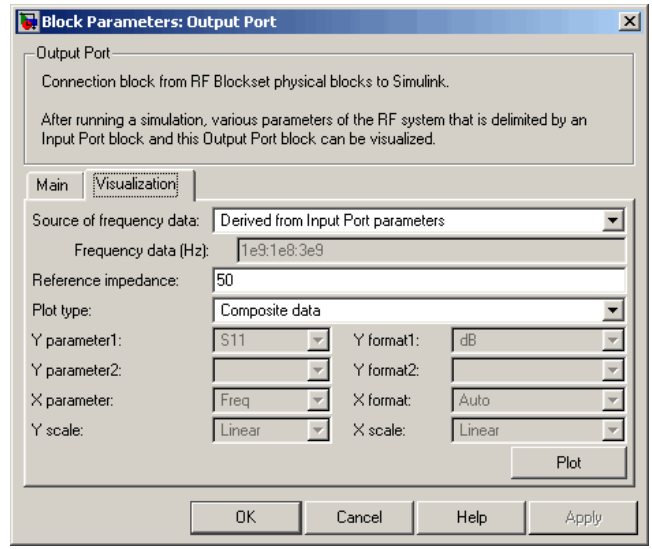
Load impedance of the RF network described in the physical model to which it connects.

### 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)

# Parallel-Plate Transmission Line

## Purpose

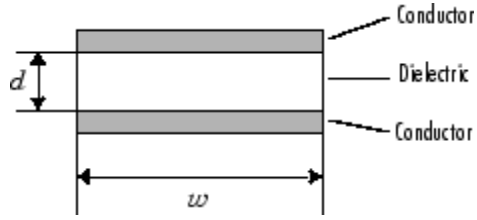
Model parallel-plate transmission line

## 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 $w$ and the plate separation $d$.




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 calculates the frequency-dependent S-parameters using the physical length of the transmission line, $D$, and the complex propagation constant, $k$.

$$S_{11} = 0$$

$$S_{12} = e^{-kD}$$

$$S_{21} = e^{-kD}$$

$$S_{22} = 0$$

$k$ can be expressed in terms of the resistance ($R$), inductance ($L$), conductance ($G$), and capacitance ($C$) per unit length (meters) as

$$k = k_r + jk_i = \sqrt{(R + j2\pi fL)(G + j2\pi fC)}$$

where

$$R = \frac{2}{w\sigma_{cond}\delta}$$

$$L = \mu\frac{d}{w}$$

$$G = \sigma_{diel}\frac{w}{d}$$

$$C = \varepsilon\frac{w}{d}$$

In these equations, $\sigma_{cond}$ is the conductivity in the conductor and $\sigma_{diel}$ is the conductivity in the dielectric. $\mu$ is the permeability of the dielectric, $\varepsilon$ is its permittivity, and skin depth $\delta$ is calculated as $1/\sqrt{\pi f\mu\sigma_{cond}}$. $f$ is the vector of modeling frequencies for the specified parameters. See the Output Port block reference page for information about determining the modeling frequencies.

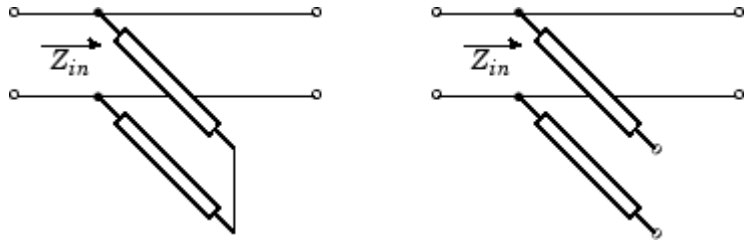
The Parallel-Plate Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

### 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.

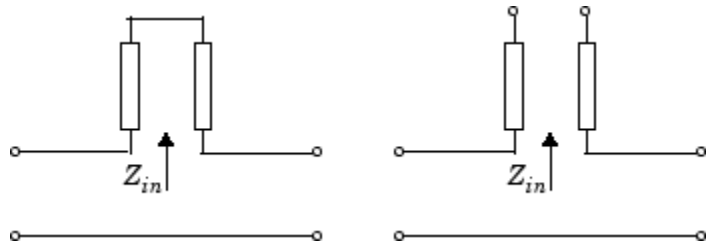



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## Dialog Box

### Main Tab

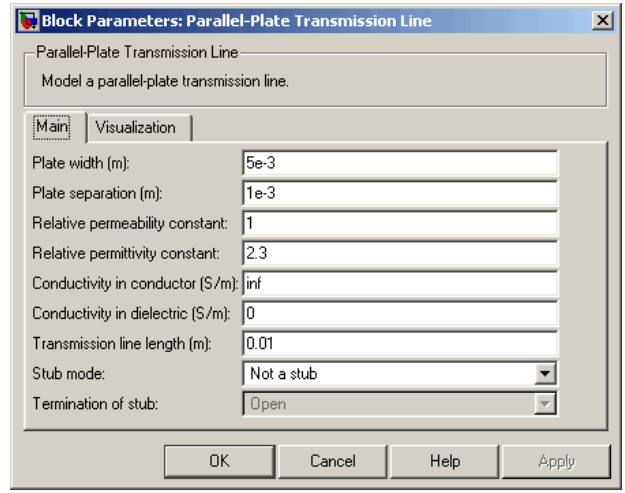



**Plate width (m)**

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 $\mu_0$.

**Relative permittivity constant**

Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space $\varepsilon_0$.

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

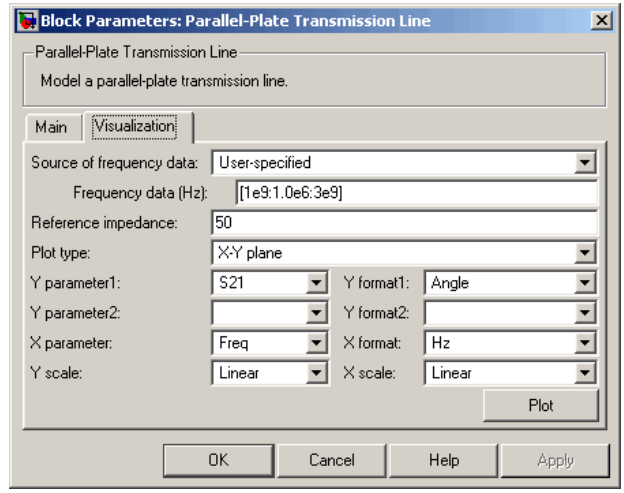
**Conductivity in dielectric (S/m)**
Conductivity of the dielectric in siemens per meter.

**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

# RLCG Transmission Line

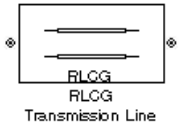
## Purpose

Model RLCG transmission line

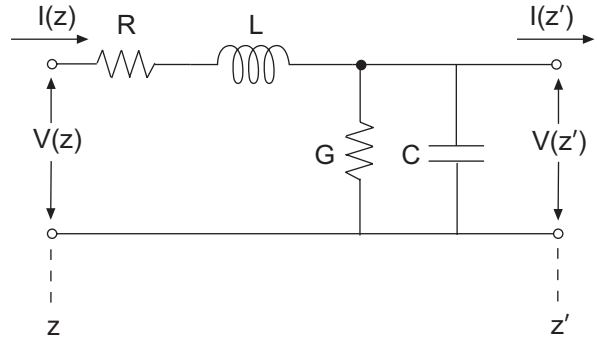
## Library

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 $z' = z + \Delta z$.

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

### Stubless Transmission Line

If you model a RLCG transmission line as a stubless line, the RLCG Transmission Line block calculates the frequency-dependent S-parameters using the physical length of the transmission line, *D*, and the complex propagation constant, *k*.

$$S_{11} = 0$$
$$S_{12} = e^{-kD}$$
$$S_{21} = e^{-kD}$$
$$S_{22} = 0$$

$k$ can be expressed in terms of the resistance ($R$), inductance ($L$), conductance ($G$), and capacitance ($C$) per unit length (meters) as

$$k = k_r + jk_i = \sqrt{(R + j2\pi fL)(G + j2\pi fC)}$$

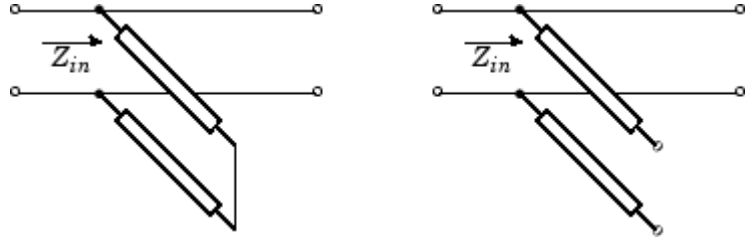
The RLCG Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

### 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.

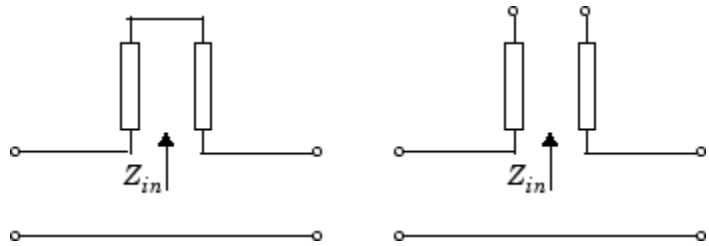



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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.

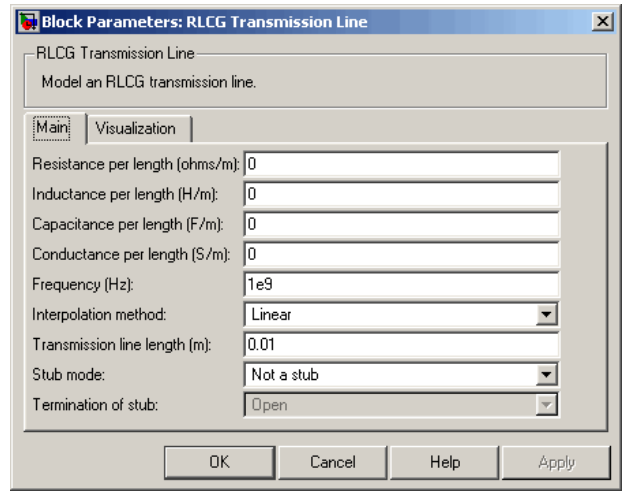



$Z_{in}$ is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## 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.

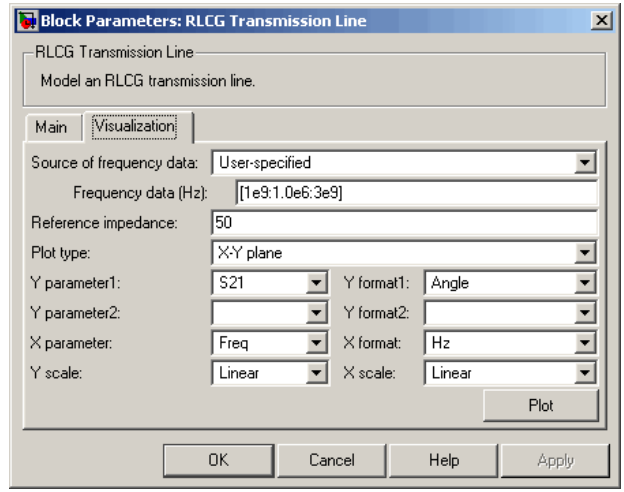
**Interpolation method**
Specify the interpolation method the block uses to calculate the parameter values at the modeling frequencies. Your choices are `Linear`, `Spline`, or `Cubic`.

**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

# S-Parameters Amplifier

## Purpose

Model nonlinear amplifier using S-parameters

## Library

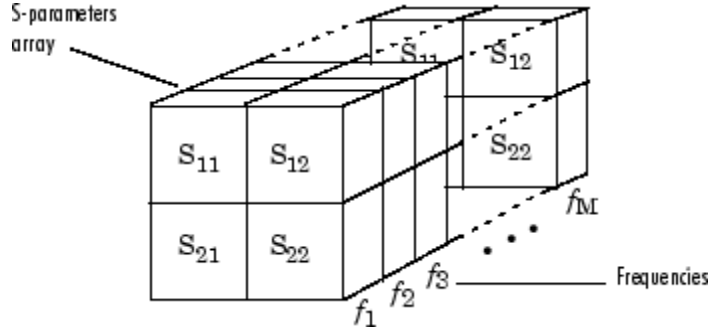
Amplifiers sublibrary of the Physical library

## 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

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

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

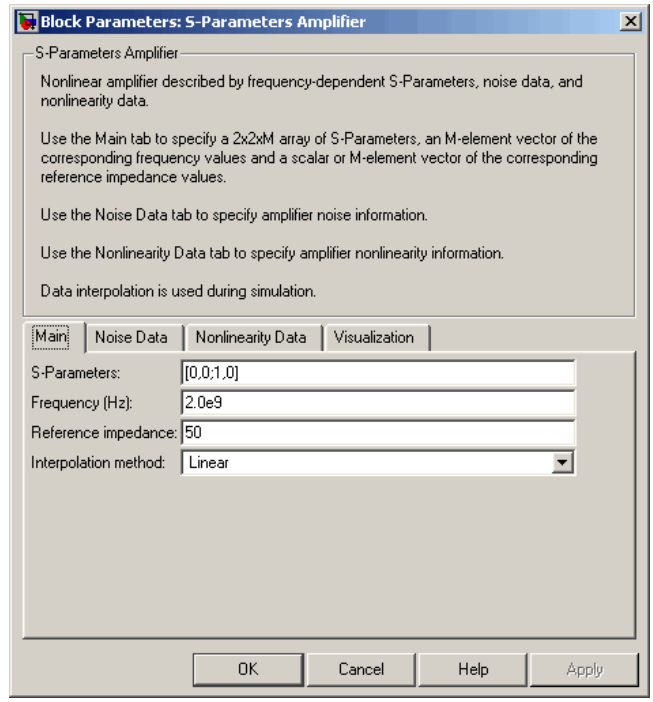
You can specify active block noise in one of the following ways:

- 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**

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.

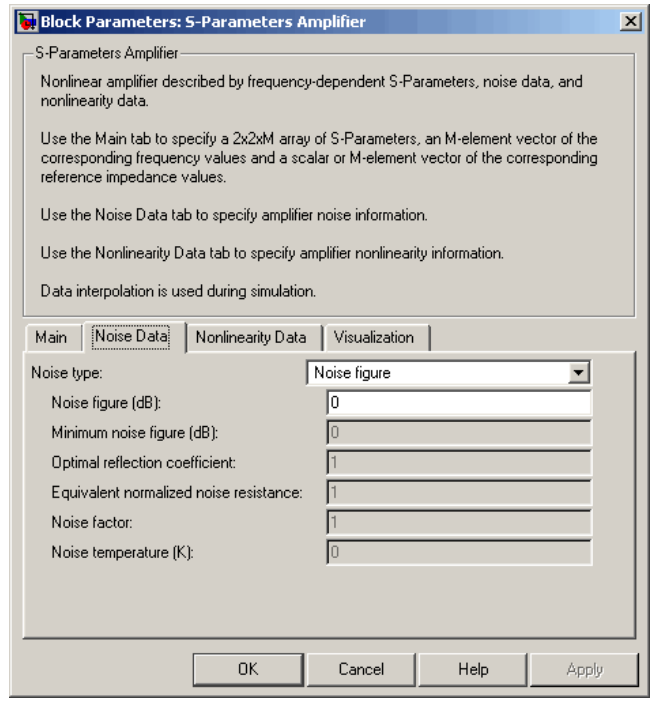
If you provide a scalar value, then that value is applied to all frequencies.

**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`.

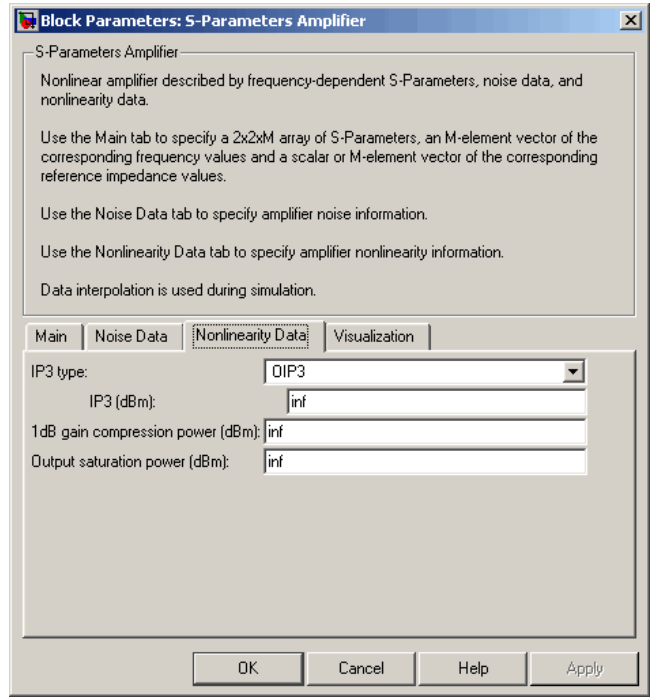
**Noise factor**

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.

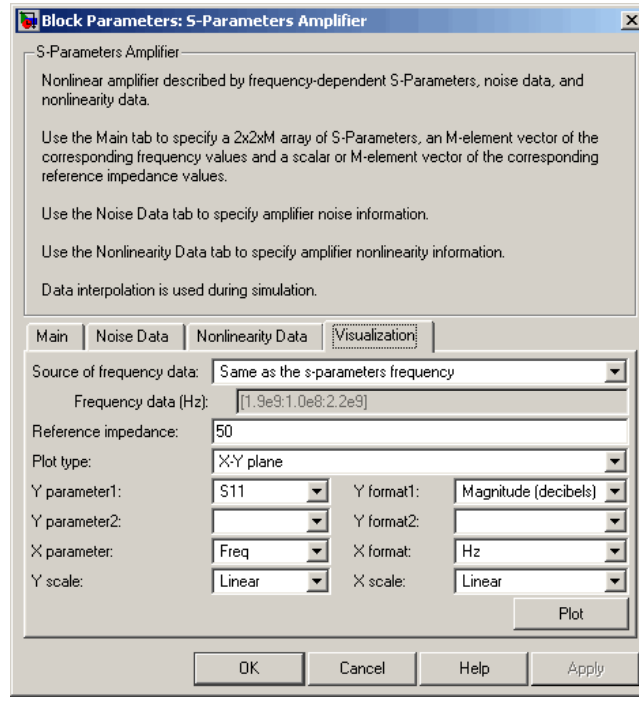
**1 dB gain compression power (dBm)**

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])
```

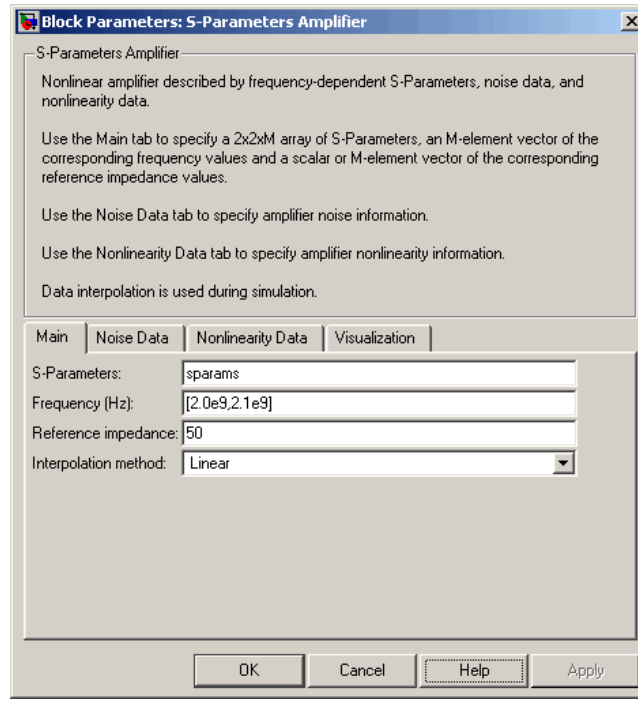
1. Type the following command at the MATLAB prompt to create a variable called `sparams` that stores the values of the S-parameters.

   ```
   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:

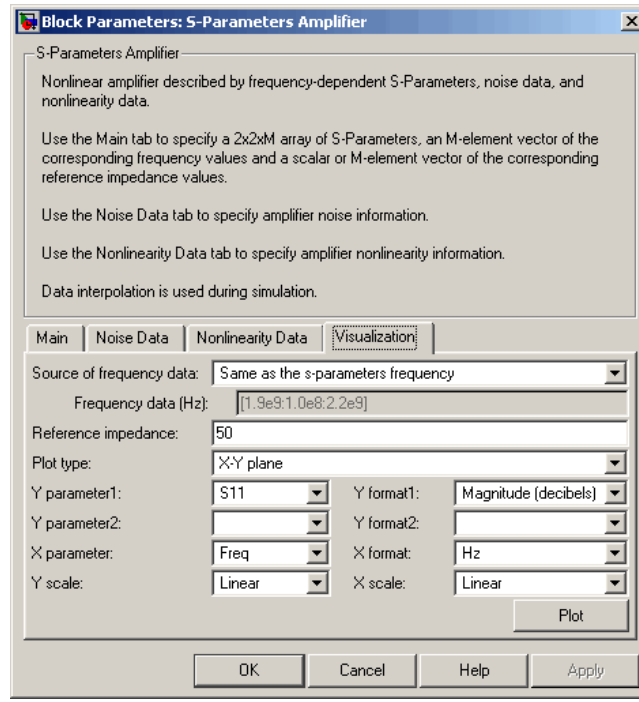
   - Set the **S-Parameters** parameter to `sparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

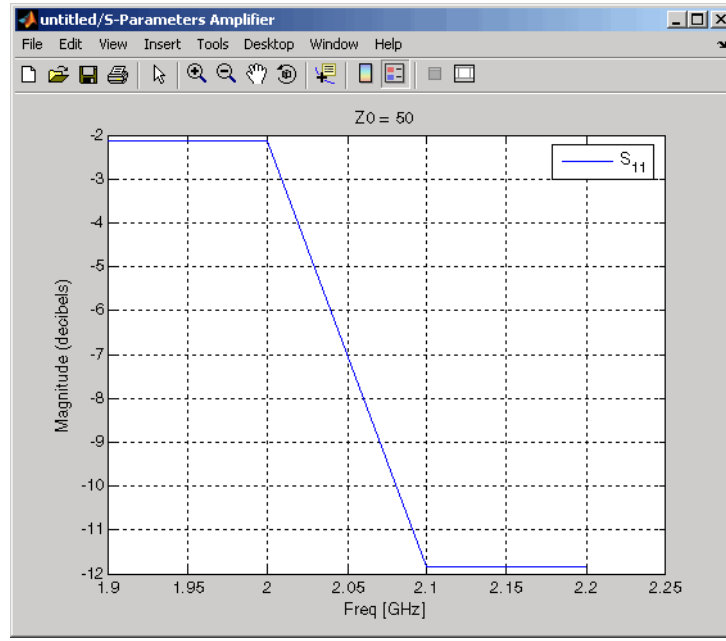



**3** Set the S-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- 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 modeling frequencies are determined by the Output Port block. See Appendix A, "RF Blockset Algorithms" for more details.

RF Blockset 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.

RF 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, RF 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

   where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

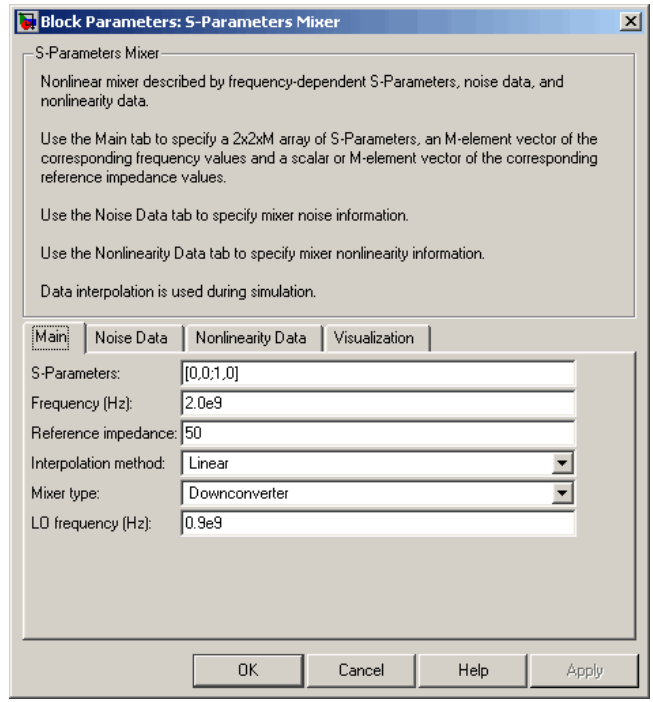
$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{{9c_3}^2 - 20c_1c_5}}{-10c_5}}$$

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

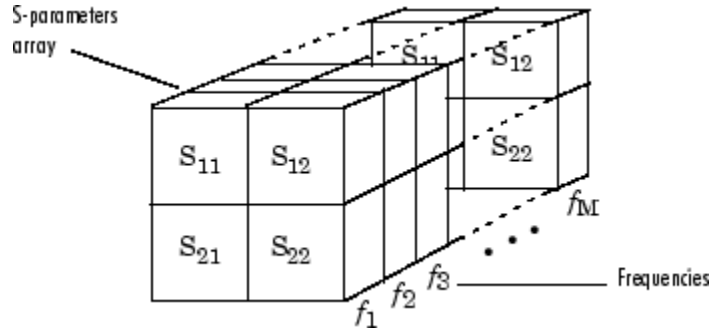



**S-Parameters**

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**

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)**

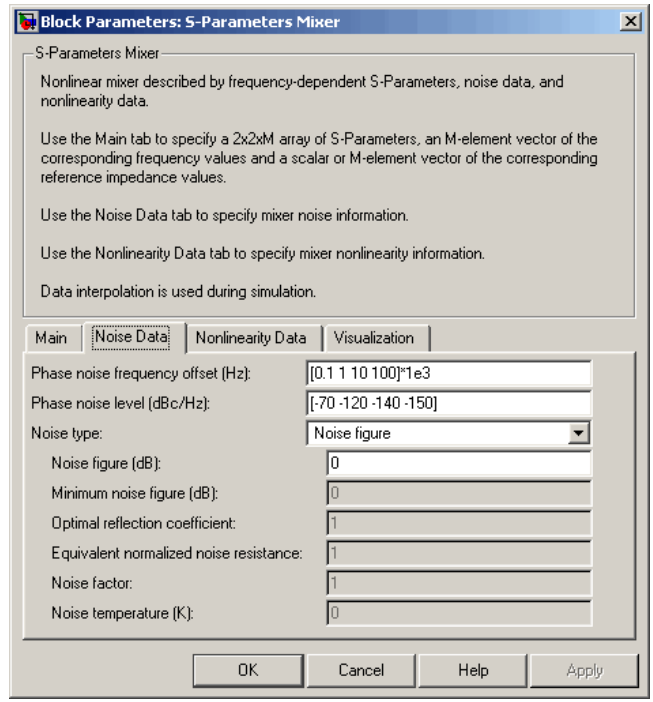
Local oscillator frequency. If you choose `Downconverter`, RF

Blockset computes the mixer output frequency, $f_{out}$, from the

mixer input frequency, $f_{in}$, and the local oscillator frequency, $f_{lo}$, as $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

**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`.

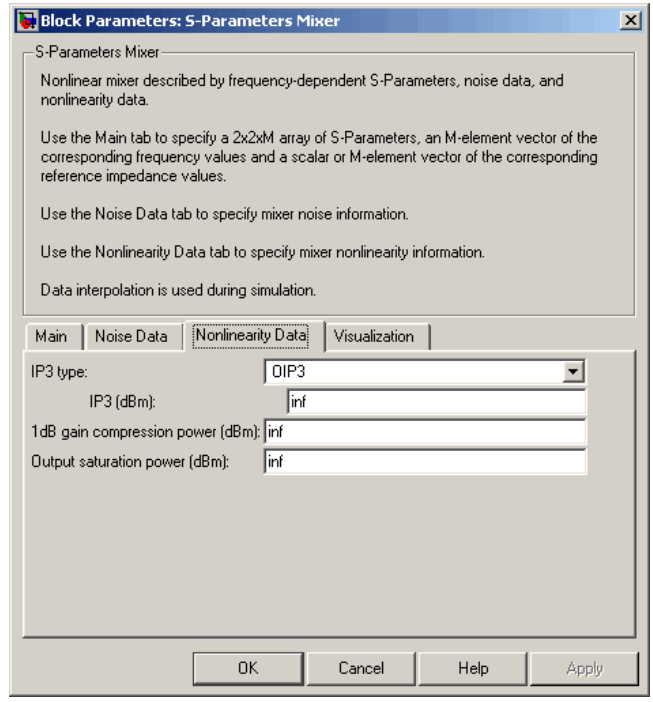
**Equivalent normalized resistance**
Resistance normalized to the resistance used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

**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.

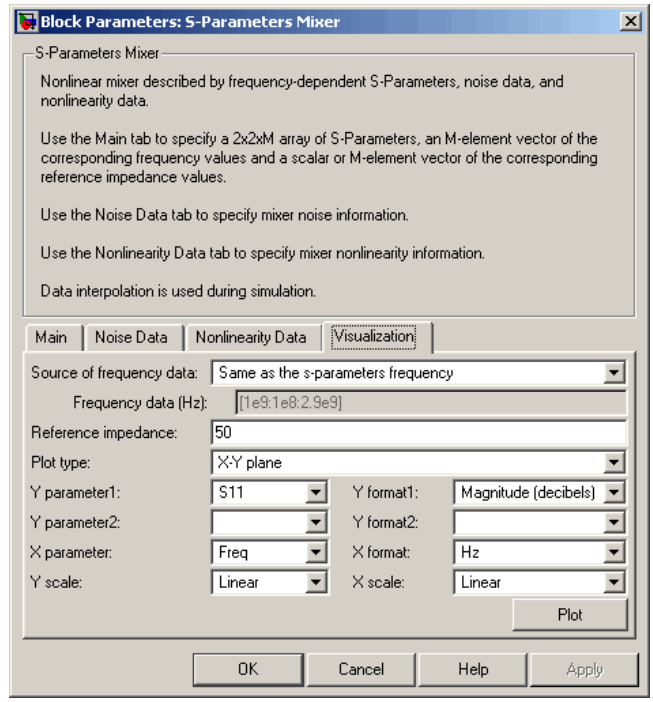
**1 dB gain compression power (dBm)**

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

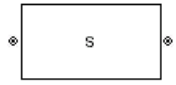
## Purpose

Model passive network using S-parameters

## Library

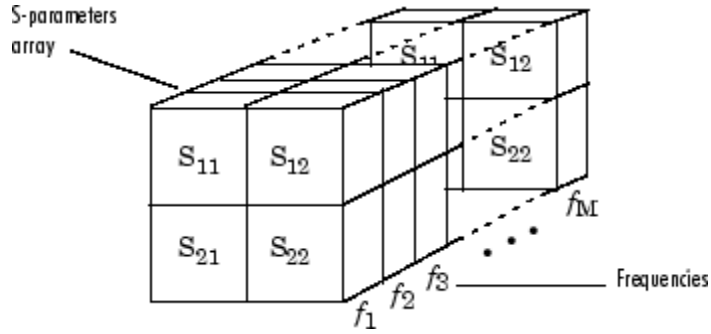
Black Box Elements sublibrary of the Physical library

## 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.

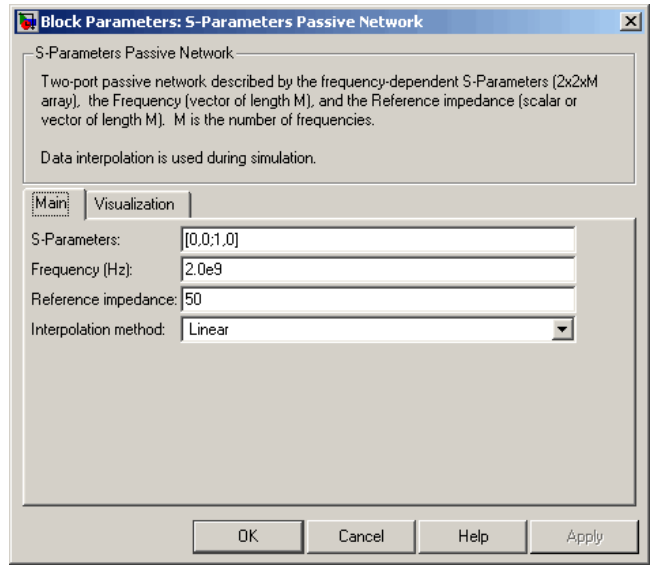
In the **S-Parameters** field of the block dialog box, provide the S-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the S-parameters as an M-element vector. The elements of the vector must be in the same order as the S-parameters. 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**

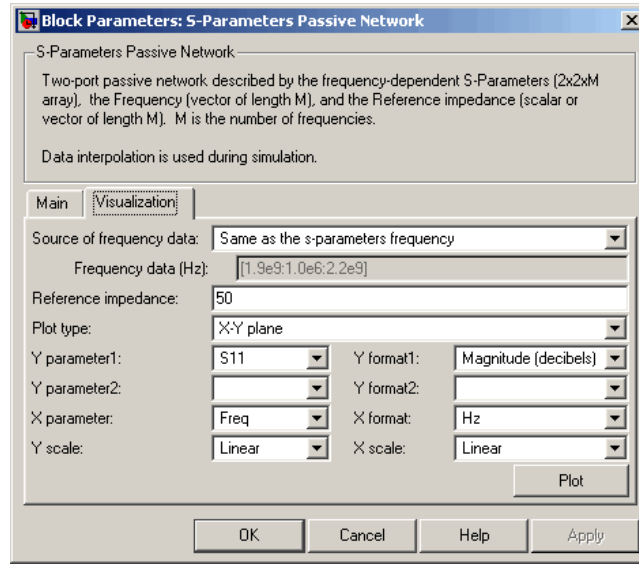
Reference impedance of the network as a scalar or a vector of length M. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

**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)
```

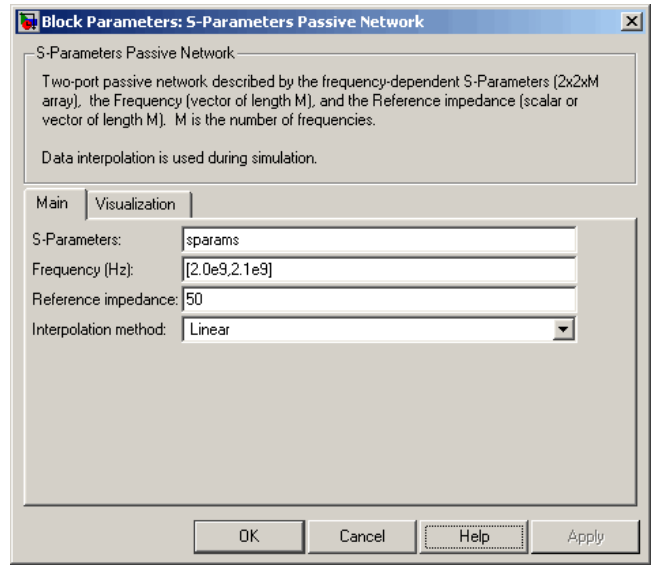
1. Type the following command at the MATLAB prompt to create a variable called `sparams` that stores the values of the S-parameters.

   ```
   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:

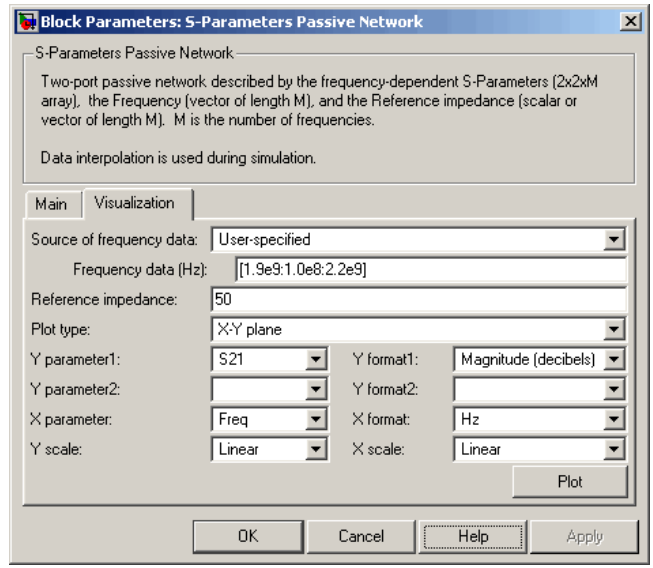
   - Set the **S-Parameters** parameter to `sparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

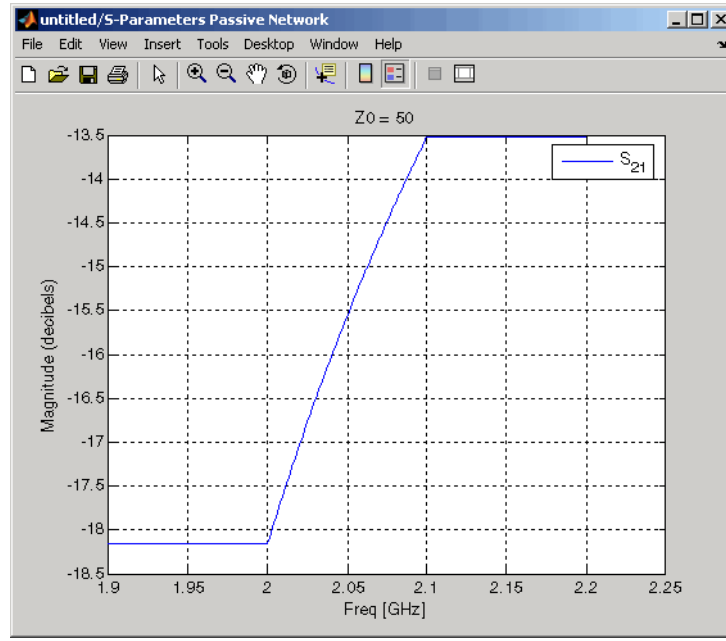



**3** Set the S-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- 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)

# Series RLC

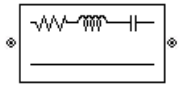
## Purpose

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.

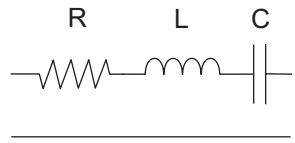
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference page for information about determining the modeling frequencies.

For this circuit, A = 1, B = Z, C = 0, and D = 1, where

$$Z = \frac{-LC\omega^2 + jRC\omega + 1}{jC\omega}$$

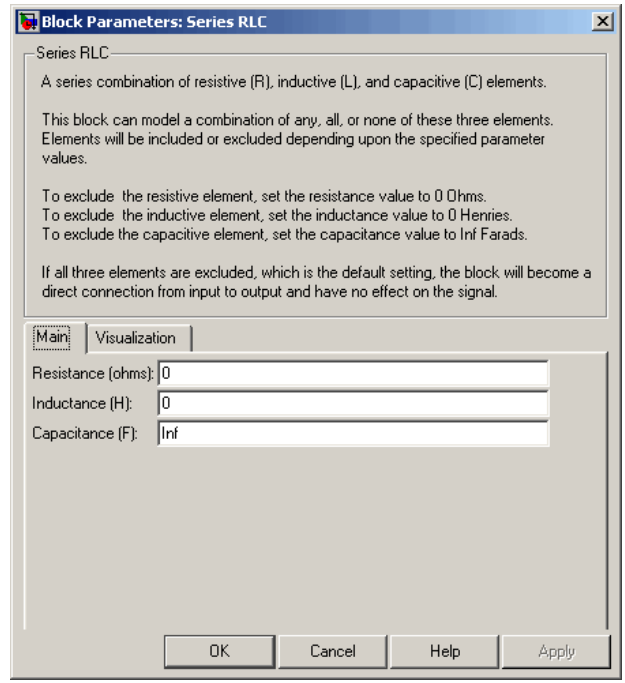
and $\omega = 2\pi f$.

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.

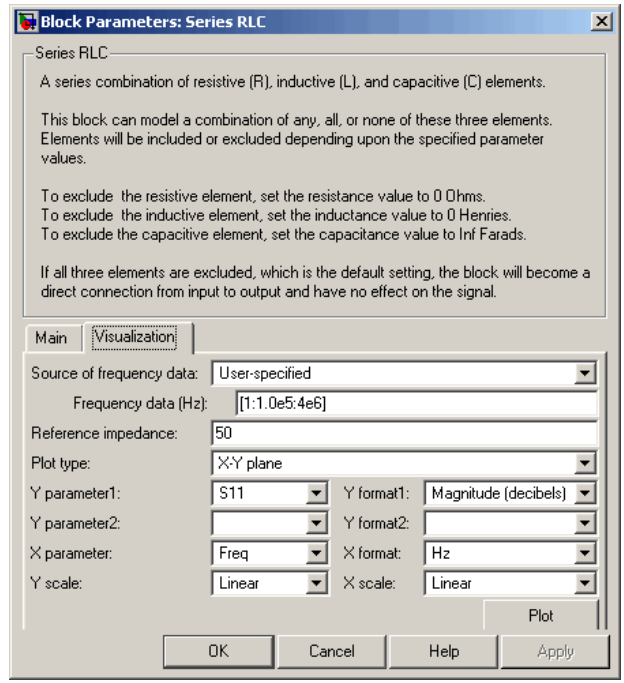
**Inductance (H)**

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, Shunt RLC

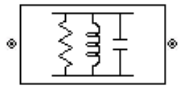
## Purpose

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 frequency-dependent S-parameters.

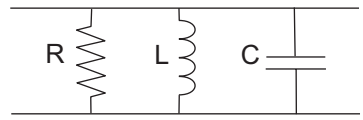
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference page for information about determining the modeling frequencies.

For this circuit, A = 1, B = 0, C = Y, and D = 1, where

$$Y = \frac{-LC\omega^2 + j(L/R)\omega + 1}{jL\omega}$$

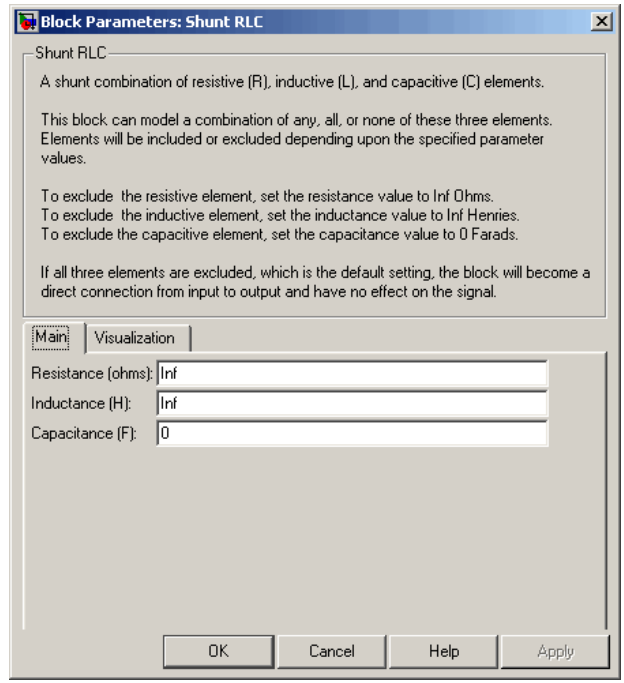
and $\omega = 2\omega f$.

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.

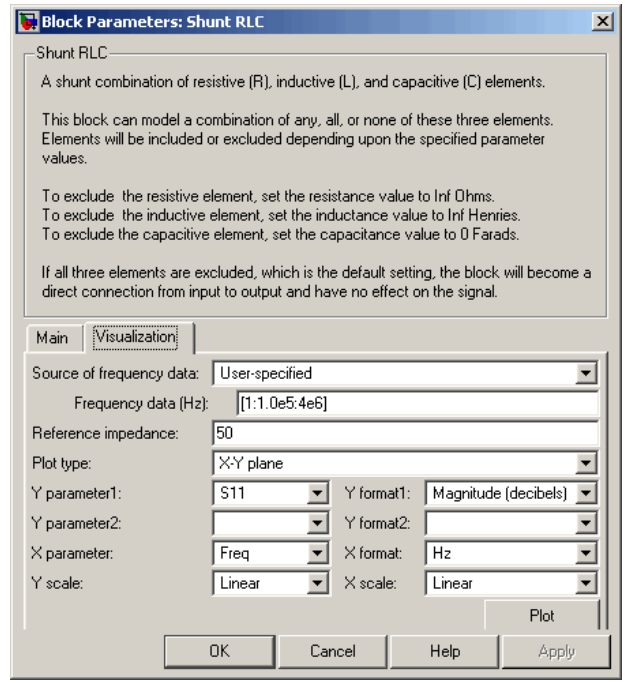
**Inductance (H)**

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 RLC

# Transmission Line

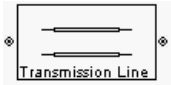
## Purpose

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 calculates the frequency-dependent S-parameters using the physical length of the transmission line, $D$, and the complex propagation constant, $k$.

$$S_{11} = 0$$

$$S_{12} = e^{-kD}$$

$$S_{21} = e^{-kD}$$

$$S_{22} = 0$$

$k = \alpha_a + i\beta$, where $\alpha_a$ is the attenuation coefficient and $\beta$ is the wave number. The attenuation coefficient $\alpha_a$ is related to the loss, $\alpha$, by

$$\alpha_a = -\ln 10^{-\frac{\alpha}{20}}$$

and the wave number $\beta$ is related to the phase velocity, $V_P$, by

$$\beta = \frac{2\pi f}{V_p}$$

where $f$ is the vector of modeling frequencies determined by the Output Port block. The phase velocity $V_P$ is also known as the wave propagation velocity.

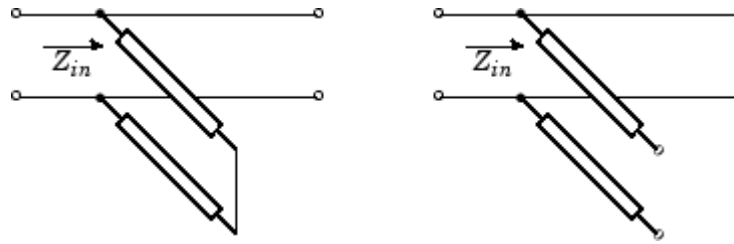
The Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

### 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.

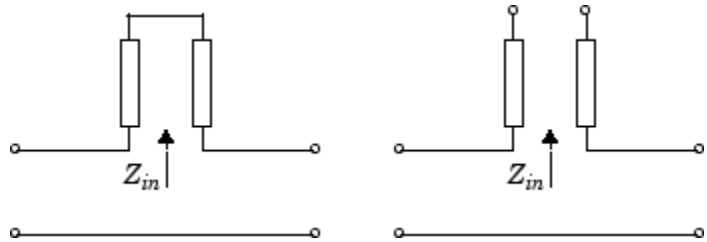



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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.

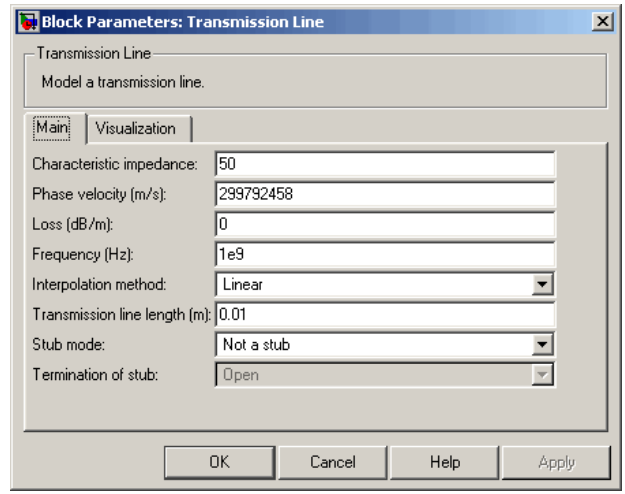



$Z_{in}$ is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## Dialog Box

### Main Tab




**Characteristic impedance**

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.

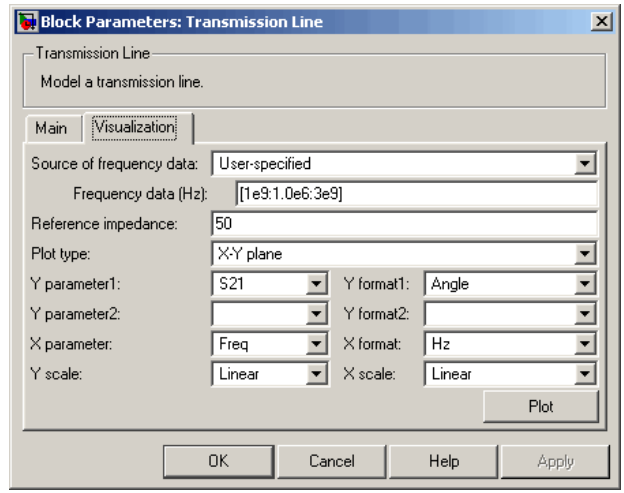
**Stub mode**

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

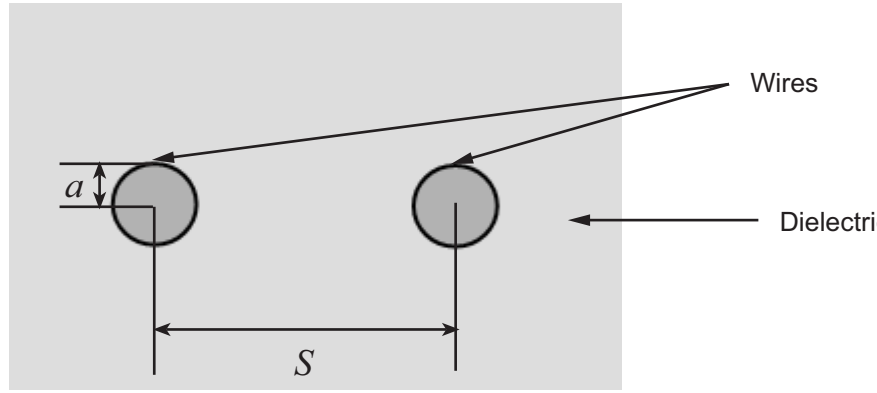
Model two-wire transmission line

## 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 permeability of the wires. RF Blockset assumes the relative permittivity 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 parallel-plate transmission line as a stubless line, the Two-Wire Transmission Line block calculates the frequency-dependent S-parameters using the physical length of the transmission line, $D$, and the complex propagation constant, $k$.

$$S_{11} = 0$$

$$S_{12} = e^{-kD}$$

$$S_{21} = e^{-kD}$$

$$S_{22} = 0$$

$k$ can be expressed in terms of the resistance ($R$), inductance ($L$), conductance ($G$), and capacitance ($C$) per unit length (meters) as

$$k = k_r + jk_i = \sqrt{(R + j2\pi fL)(G + j2\pi fC)}$$

where

$$R = \frac{1}{\pi a \sigma_{\text{cond}} \delta}$$

$$L = \frac{\mu}{\pi} \operatorname{acosh}\left(\frac{D}{2a}\right)$$

$$G = \frac{\pi \sigma_{\text{diel}}}{\operatorname{acosh}(D/(2a))}$$

$$C = \frac{\pi \varepsilon}{\operatorname{acosh}(D/(2a))}$$

In these equations, $\sigma_{\text{cond}}$ is the conductivity in the conductor and $\sigma_{\text{diel}}$ is the conductivity in the dielectric. $\mu$ is the permeability of the dielectric, $\varepsilon$ is its permittivity, and skin depth $\delta$ is calculated as $1/\sqrt{\pi f \mu \sigma_{\text{cond}}}$. $f$ 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.

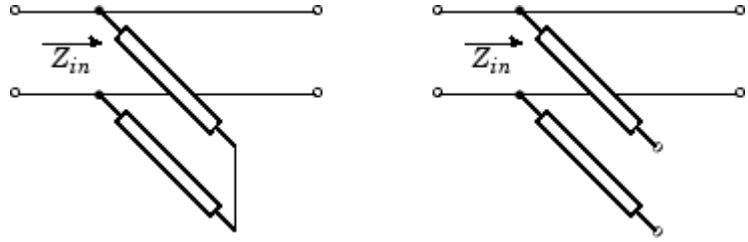
### Shunt and Series Stubs

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.

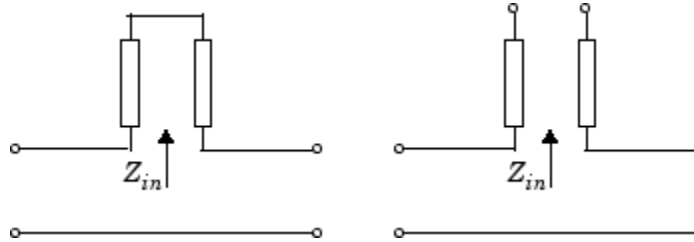



$Z_{in}$ is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$
$$B = 0$$
$$C = 1/Z_{in}$$
$$D = 1$$

### 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.

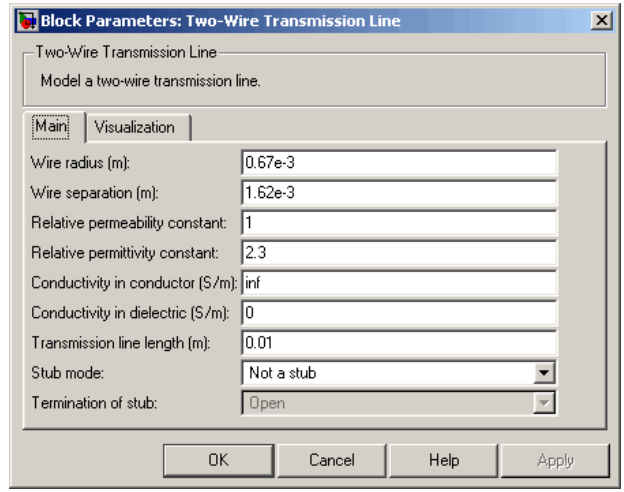



$Z_{in}$ is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

## 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 $\mu_0$.

**Relative permittivity constant**
Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space $\varepsilon_0$.

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

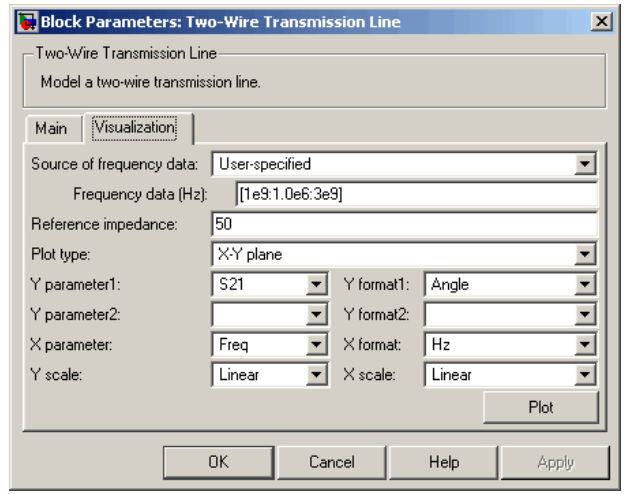
**Conductivity in dielectric (S/m)**
Conductivity of the dielectric in siemens per meter.

**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

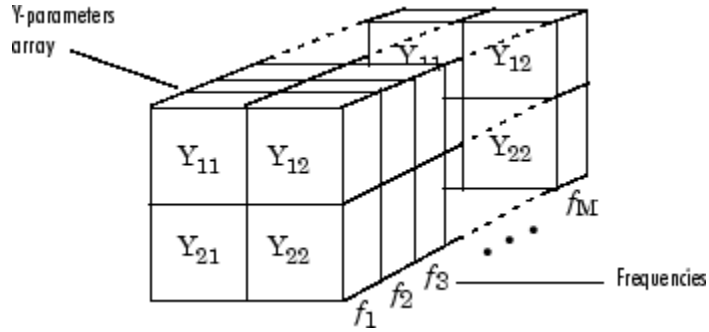
Amplifiers sublibrary of the Physical library

## 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

   where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

**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

### Active Noise

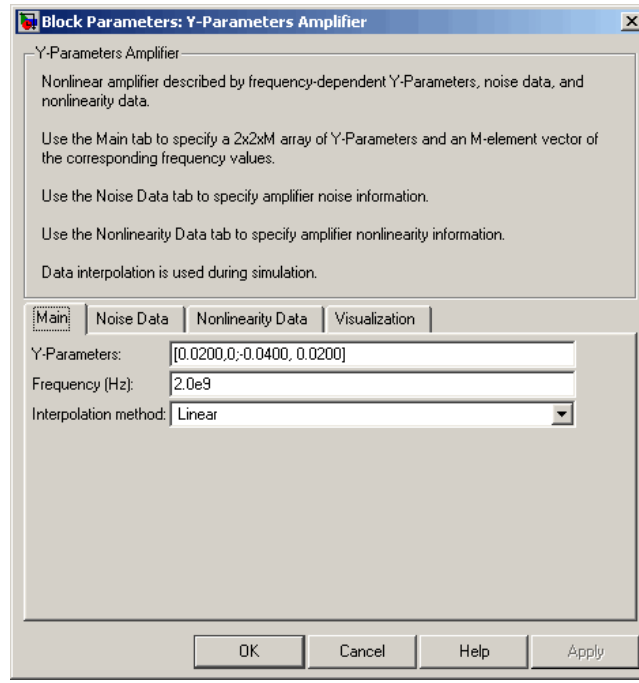
You can specify active block noise in one of the following ways:

- 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)**

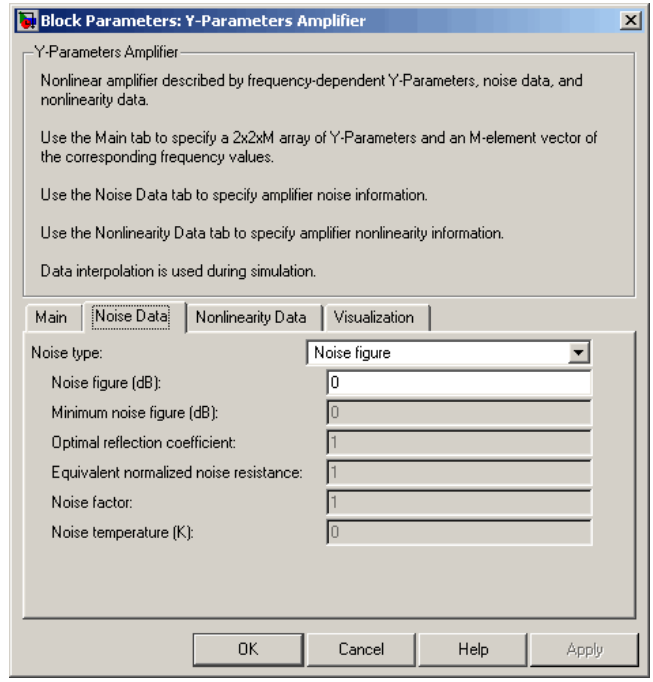
Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in **Y-Parameters**. All frequencies must be positive.

**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`.

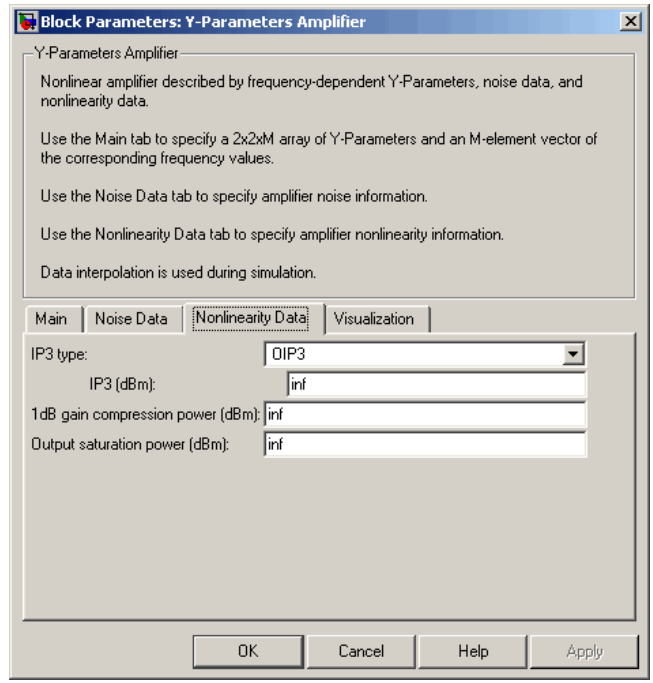
**Noise factor**

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.

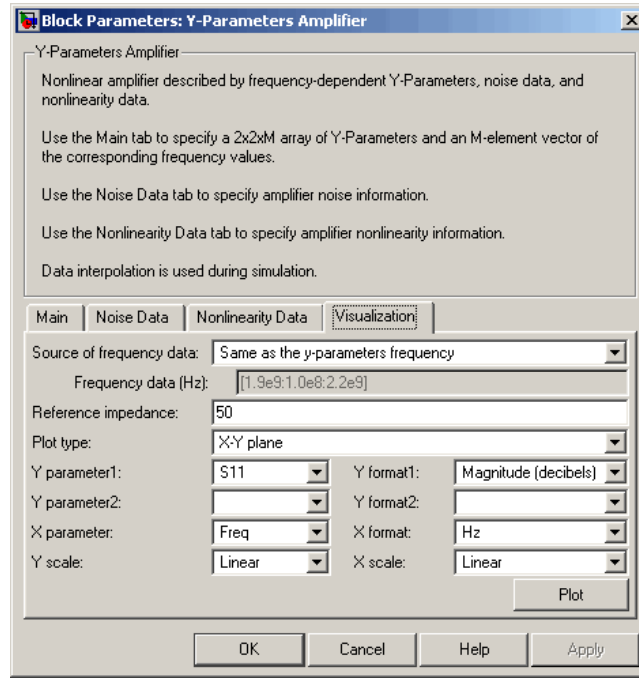
**1 dB gain compression power (dBm)**

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])
```

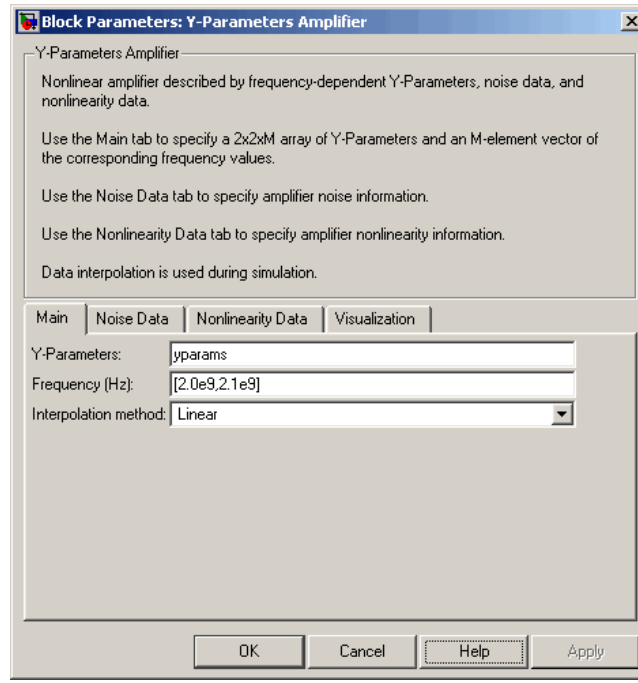
1. Type the following command at the MATLAB prompt to create a variable called `yparams` that stores the values of the Y-parameters.

   ```
   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:

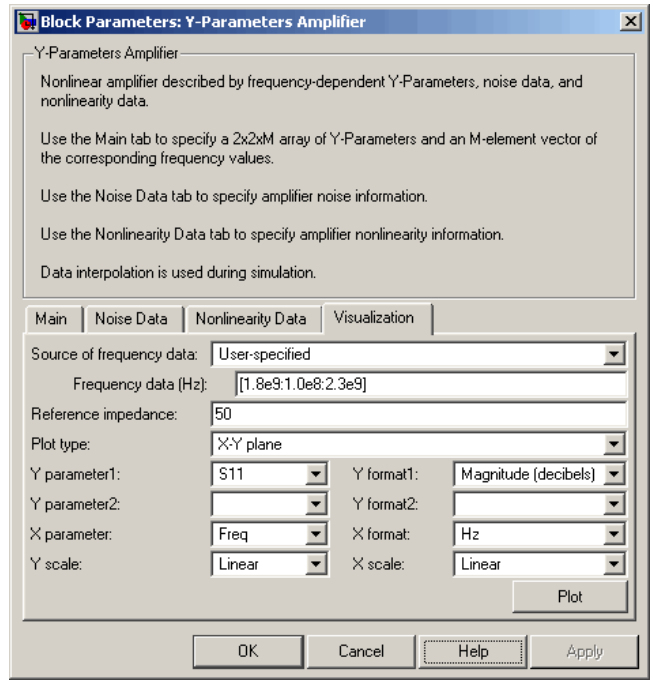
   - Set the **Y-Parameters** parameter to `yparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

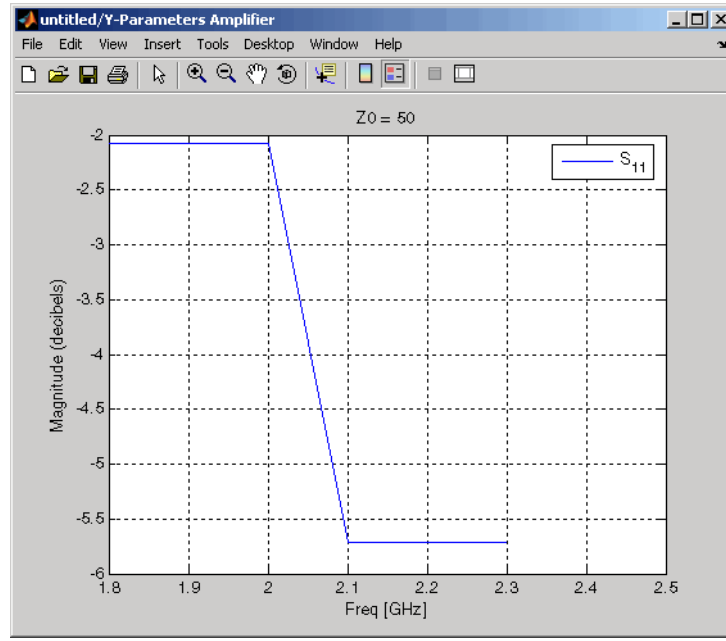



**3** Set the Y-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- 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)

# Y-Parameters Mixer

## 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 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.

RF 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, RF 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

   where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

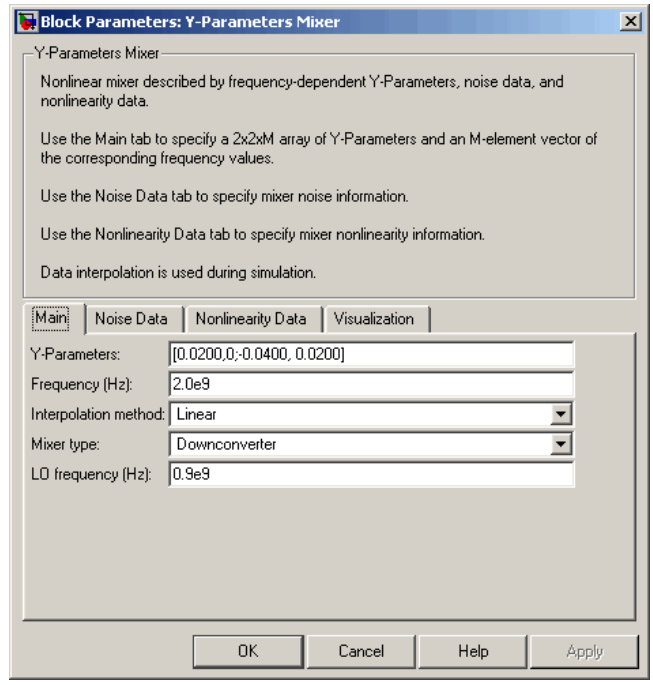
$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{{9c_3}^2 - 20c_1c_5}}{-10c_5}}$$

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

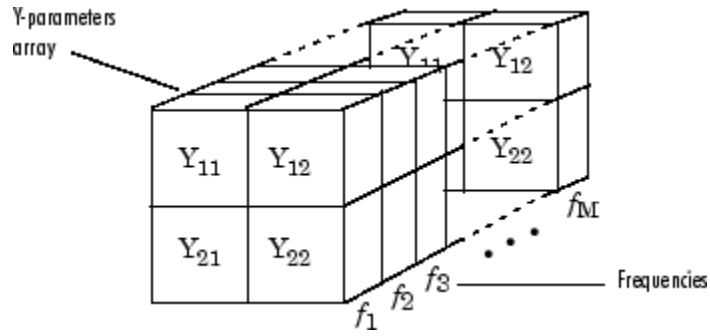



**Y-Parameters**

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**

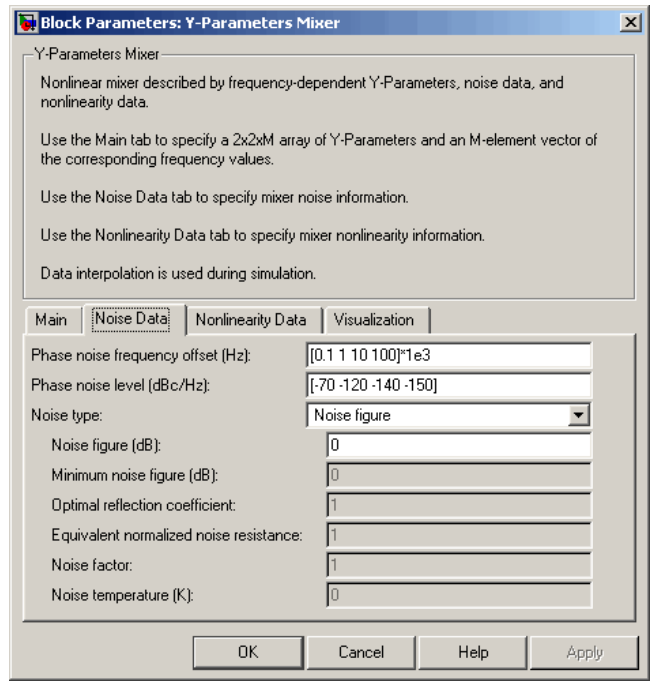
Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

**LO frequency (Hz)**

Local oscillator frequency. If you choose `Downconverter`, RF Blockset computes the mixer output frequency, $f_{out}$, from the mixer input frequency, $f_{in}$, and the local oscillator frequency, $f_{lo}$, as $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

**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`.

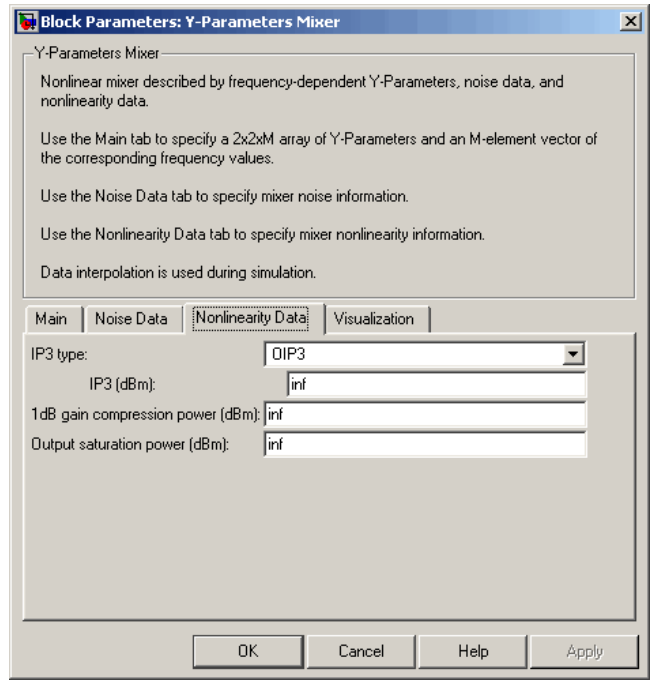
**Noise factor**

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)**

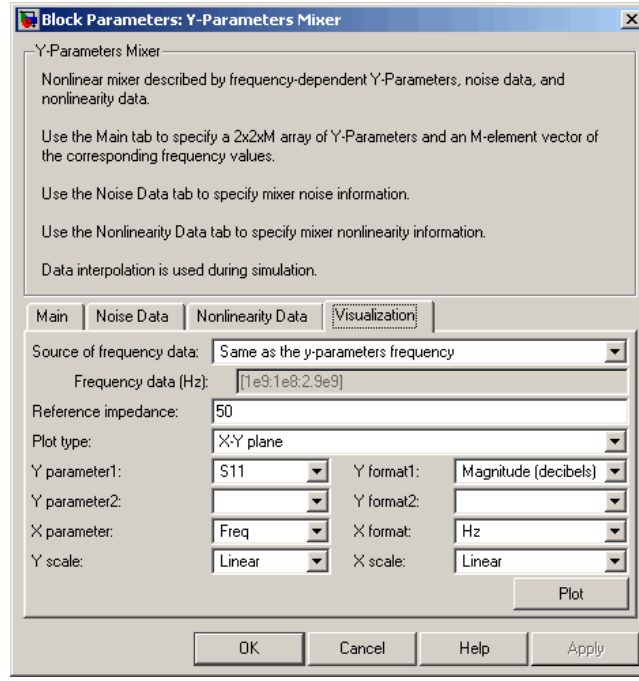
Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data

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

# Y-Parameters Passive Network

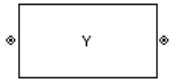
## Purpose

Model passive network using Y-parameters

## Library

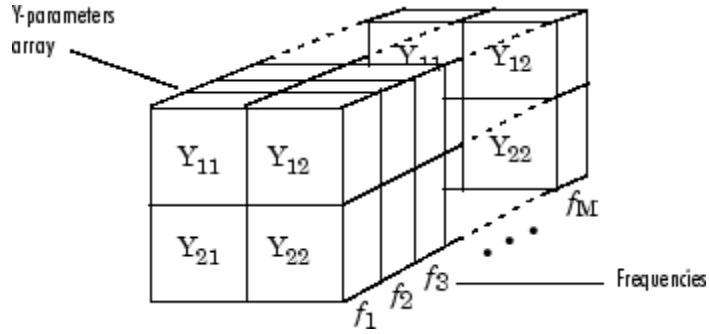
Black Box Elements sublibrary of the Physical library

## 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.

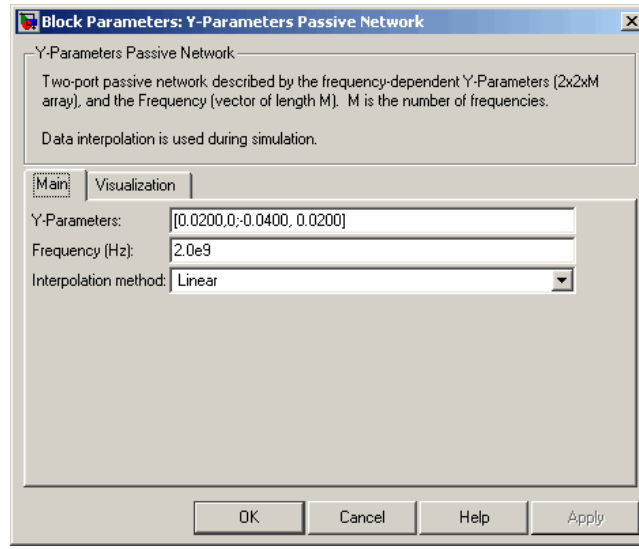
In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Y-parameters as an M-element vector. The elements of the vector must be in the same order as the Y-parameters. 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.

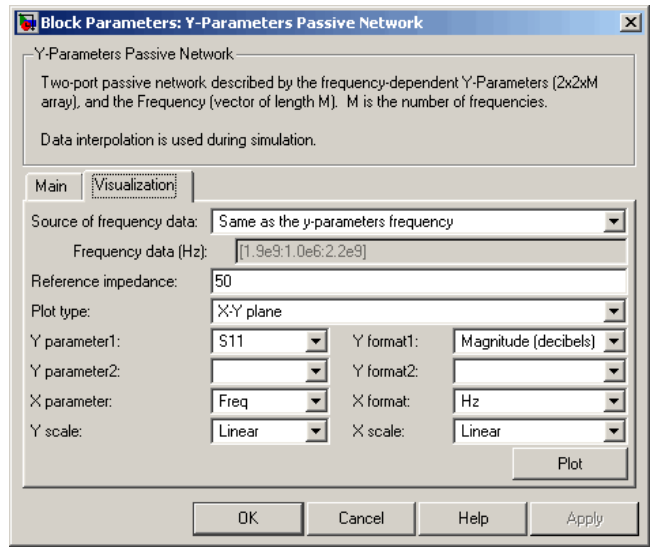
**Interpolation method**

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])
```

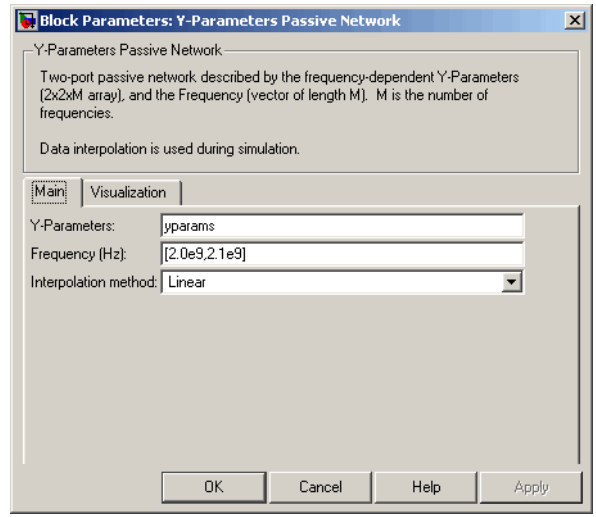
1 Type the following command at the MATLAB prompt to create a variable called `yparams` that stores the values of the Y-parameters.

```
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]`.

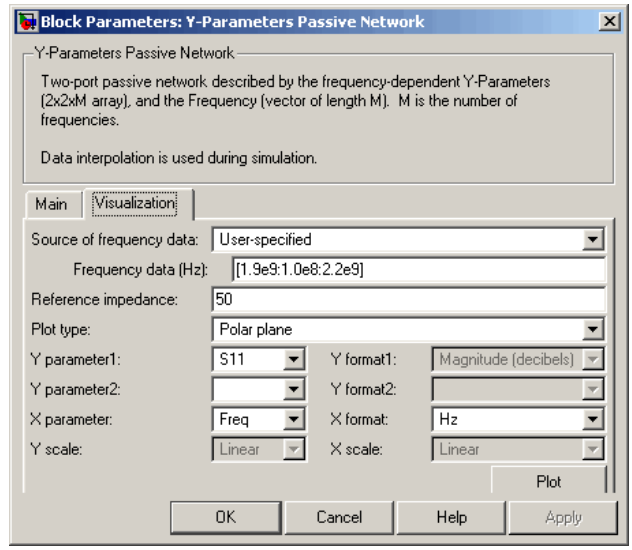
Click **Apply**. This action applies the specified settings.



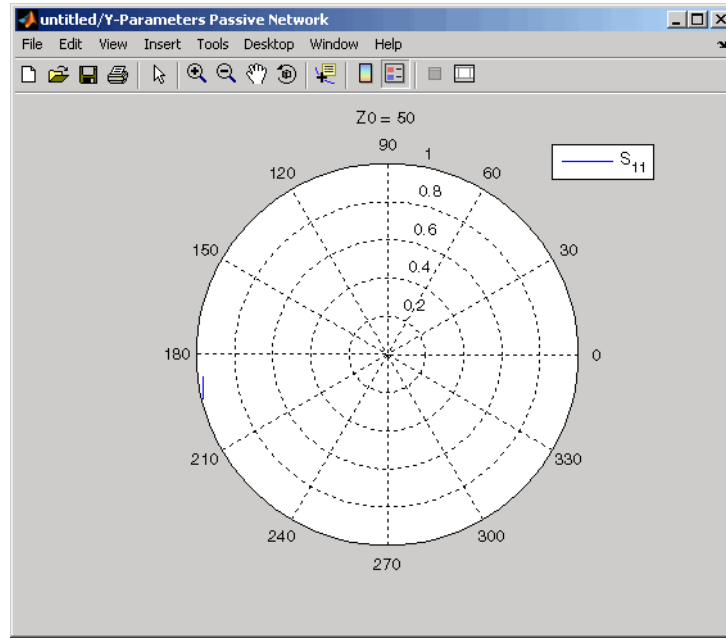

3 Set the Y-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- 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

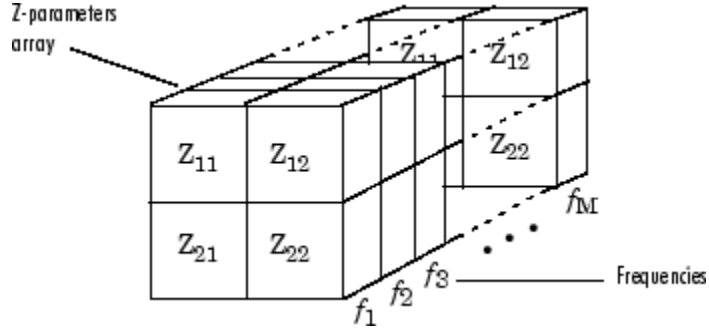
Amplifiers sublibrary of the Physical library

## 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

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

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

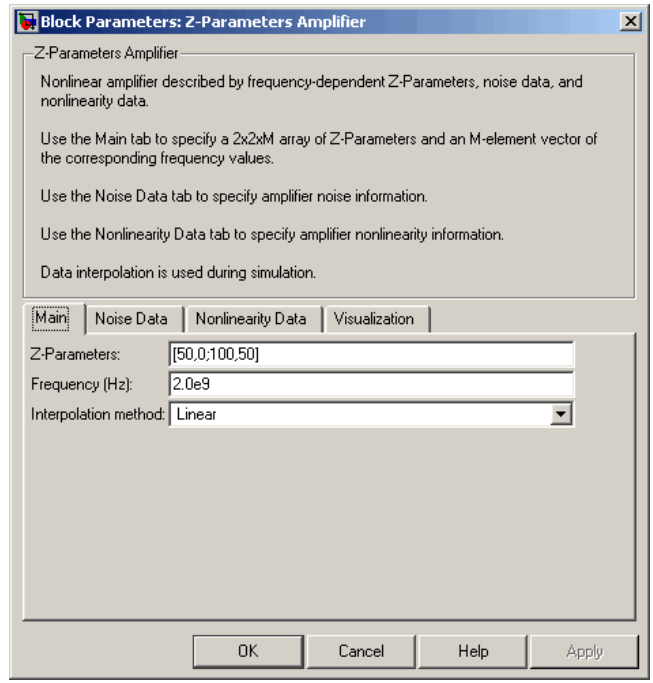
You can specify active block noise in one of the following ways:

- 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)**

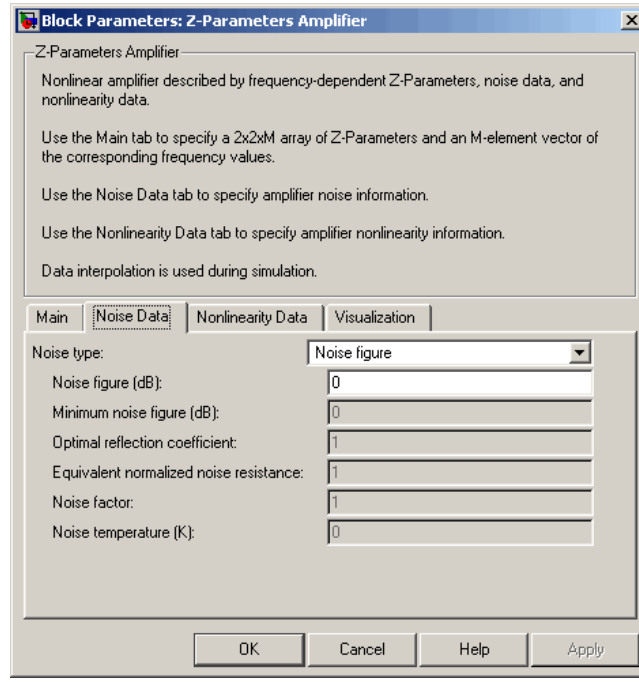
Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in **Z-Parameters**. All frequencies must be positive.

**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`.

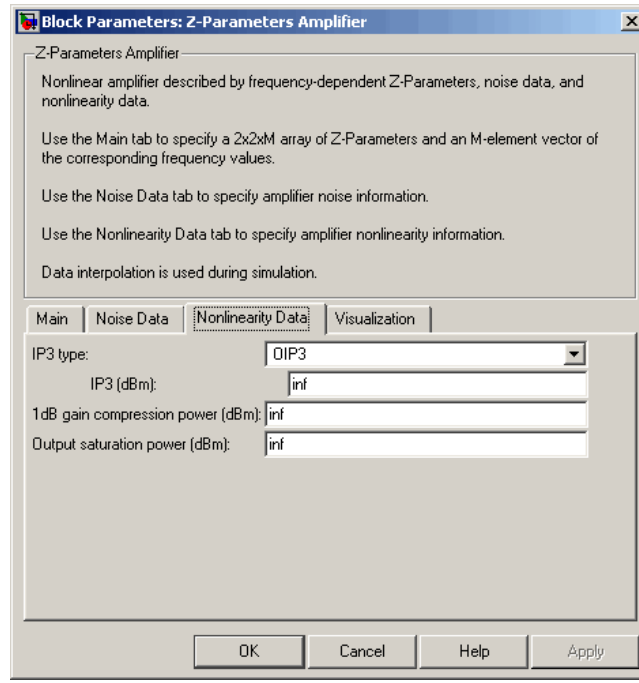
**Noise factor**

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.

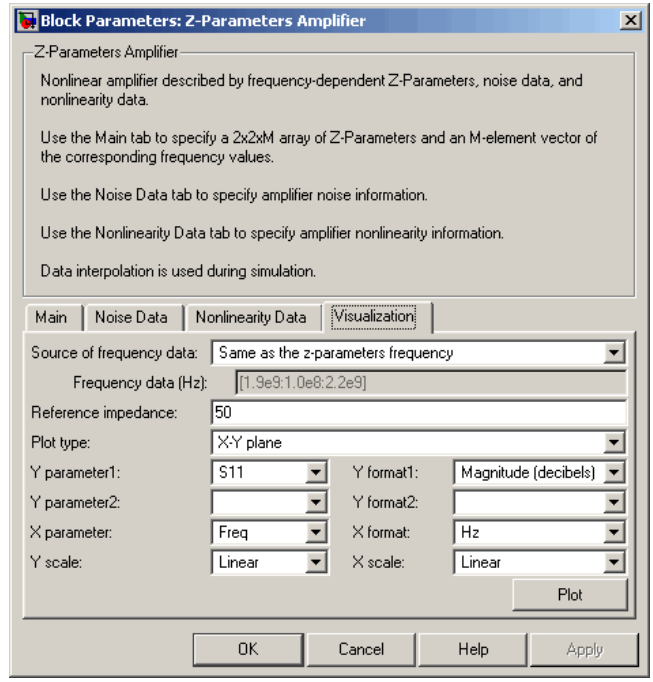
**1 dB gain compression power (dBm)**

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])
```

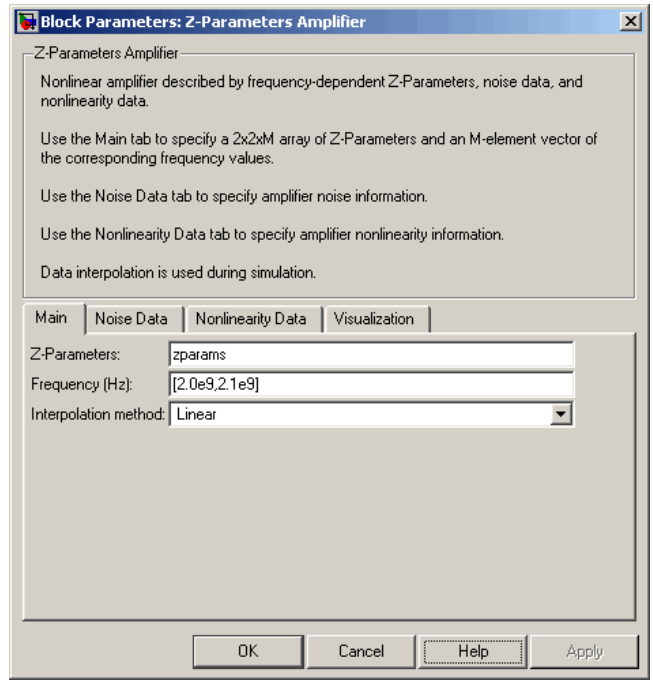
**1** Type the following command at the MATLAB prompt to create a variable called `zparams` that stores the values of the Z-parameters.

```
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:

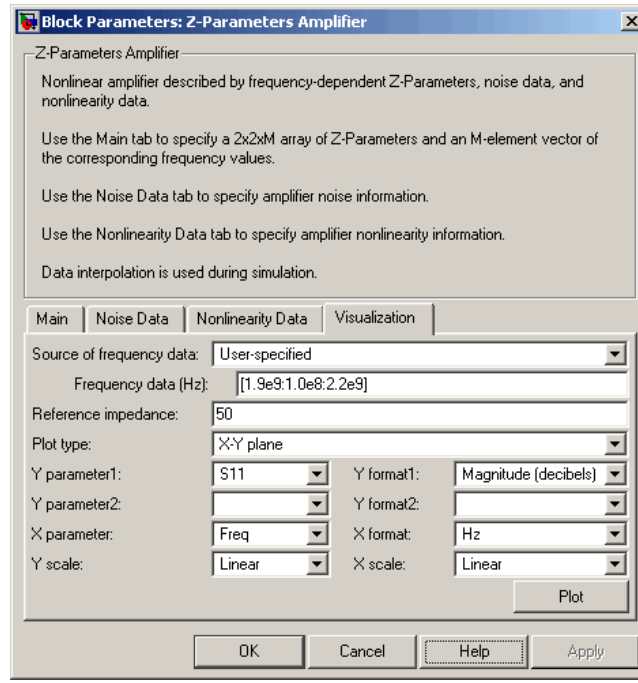
- Set the **Z-Parameters** parameter to `zparams`.
- Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

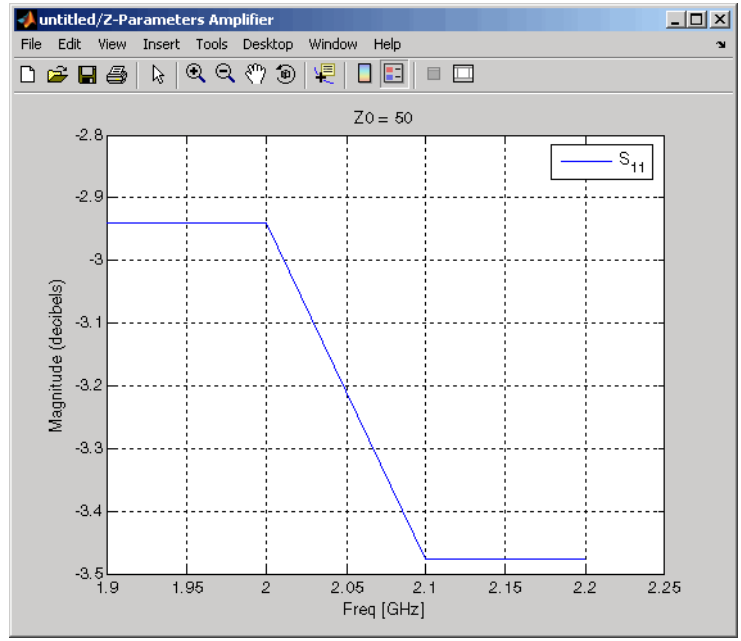



**3** Set the Z-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- 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 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.

RF 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, RF 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. Converting the IIP3 value from decibels to linear units
3. Computing a scaling factor, which is equal to 3 divided by the linear IIP3 value
4. Applying the scaling factor
5. Limiting the scaled input to a maximum value of 1 and applying an AM/AM conversion to the magnitude of the scaled signal according to the following function:

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

where $u$ is the magnitude of the scaled signal.

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`:

$$c_1 = \sqrt{G_0}$$

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

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

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

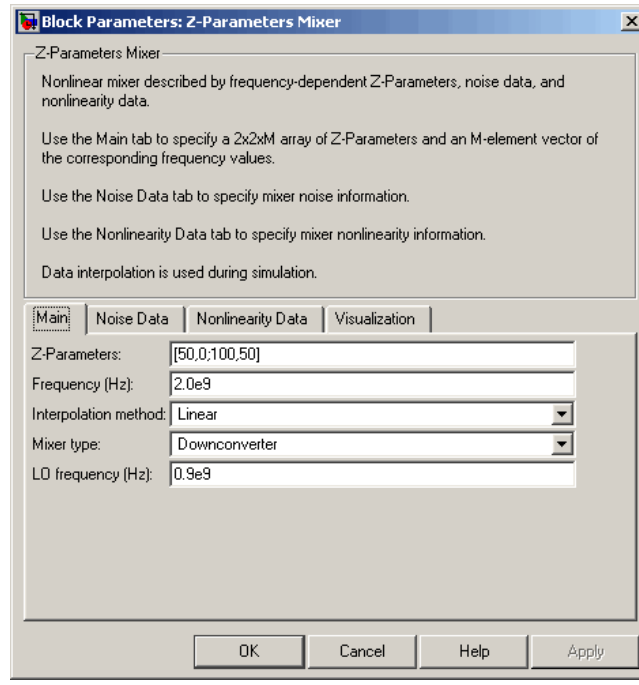
$$A_{sat} = \sqrt{\frac{3c_3 + \sqrt{9{c_3}^2 - 20c_1c_5}}{-10c_5}}$$

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

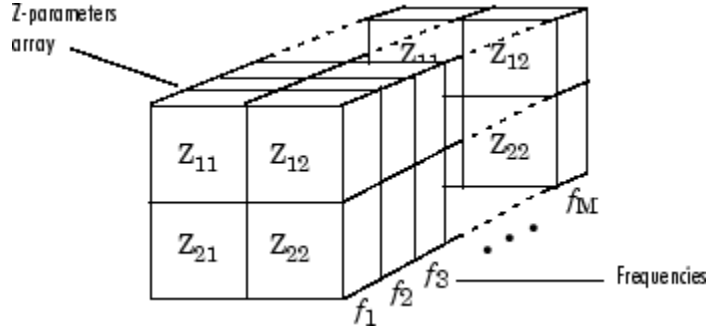



**Z-Parameters**

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**

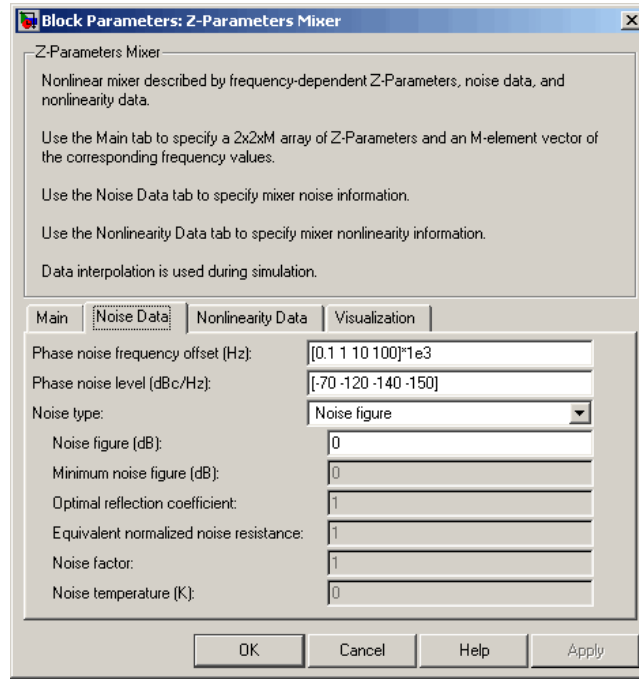
Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

**LO frequency (Hz)**

Local oscillator frequency. If you choose `Downconverter`, RF Blockset computes the mixer output frequency, $f_{out}$, from the mixer input frequency, $f_{in}$, and the local oscillator frequency, $f_{lo}$, as $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

**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`.

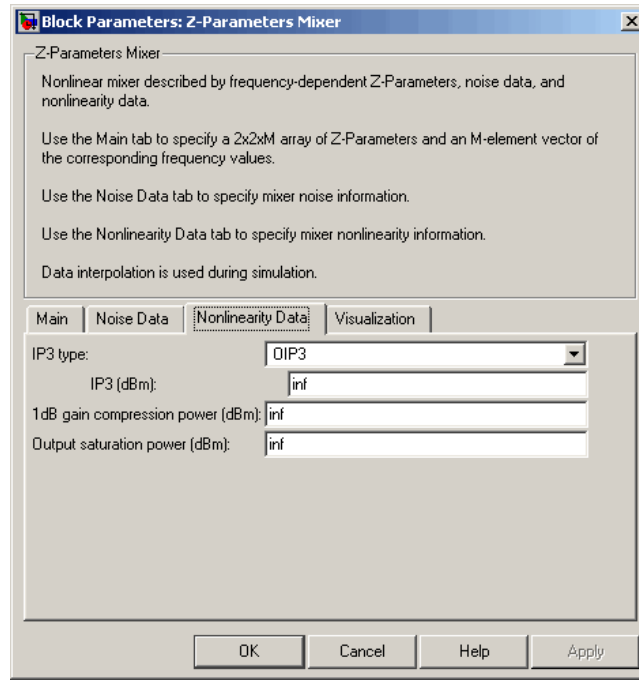
**Noise factor**

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)**

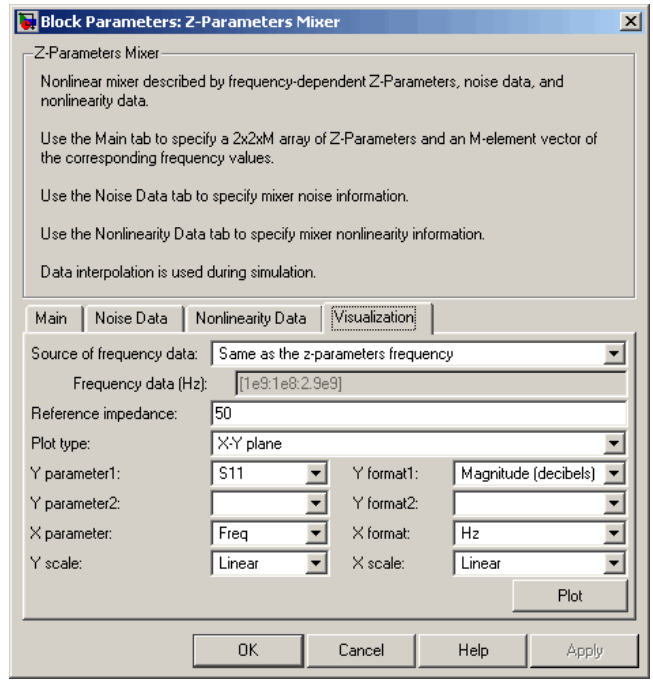
Scalar output power value at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data

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

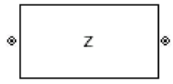
## Purpose

Model passive network using Z-parameters

## Library

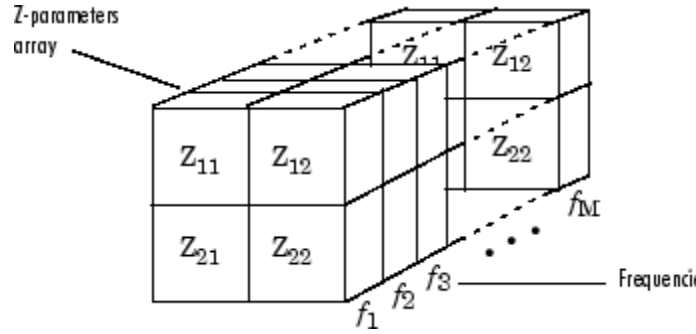
Black Box Elements sublibrary of the Physical library

## 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.

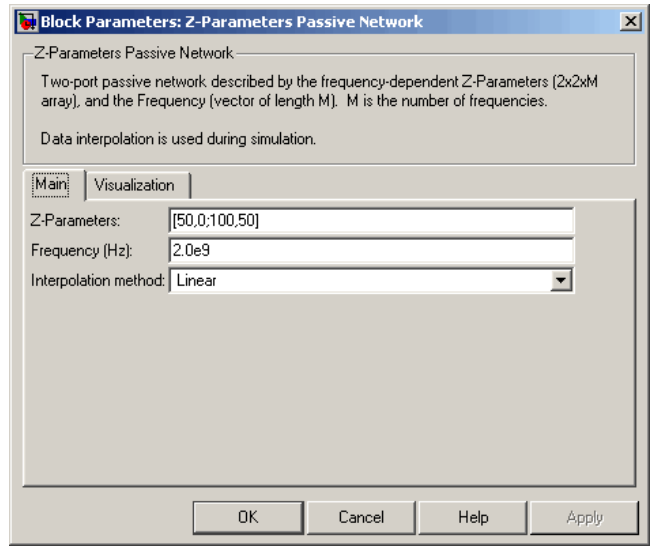
In the **Z-Parameters** field of the block dialog box, provide the Z-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Z-parameters as an M-element vector. The elements of the vector must be in the same order as the Z-parameters. 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.

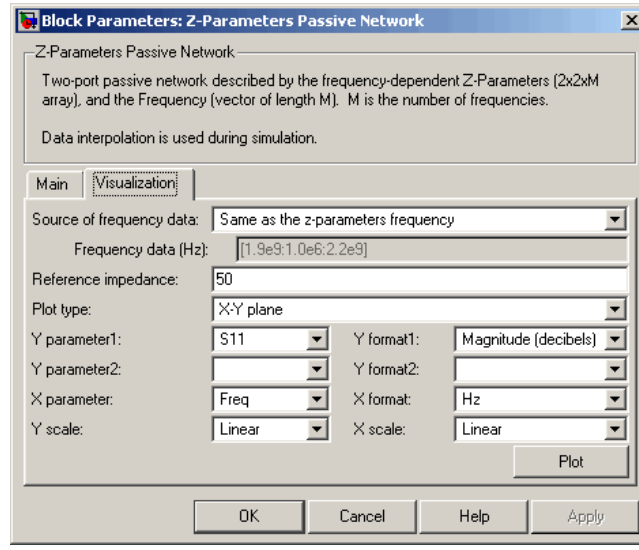
**Interpolation method**

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])
```

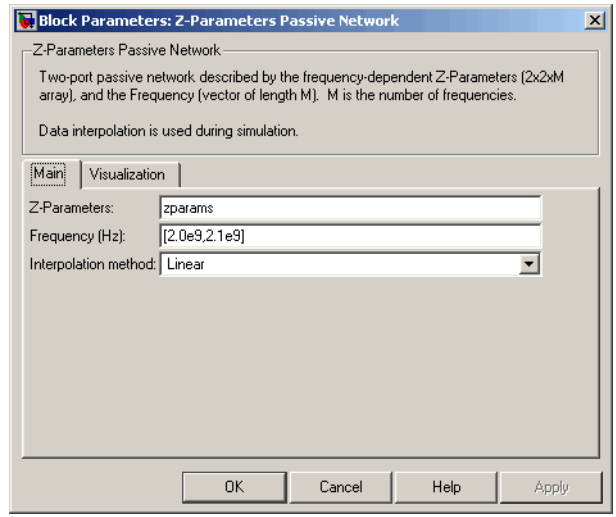
1 Type the following command at the MATLAB prompt to create a variable called `zparams` that stores the values of the Z-parameters.

```
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]`.

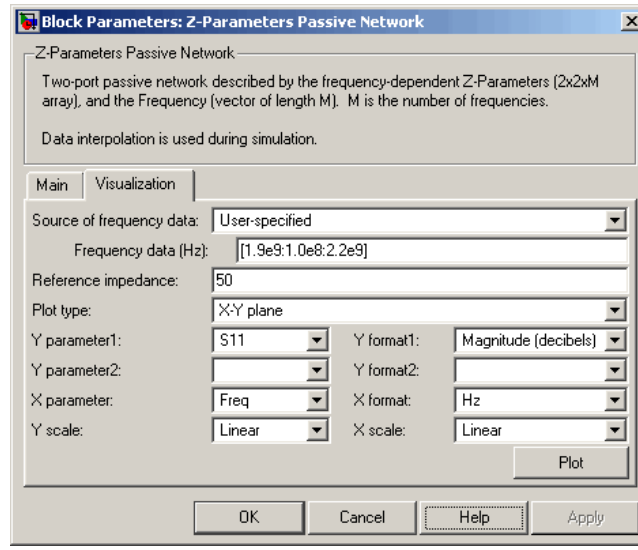
Click **Apply**. This action applies the specified settings.



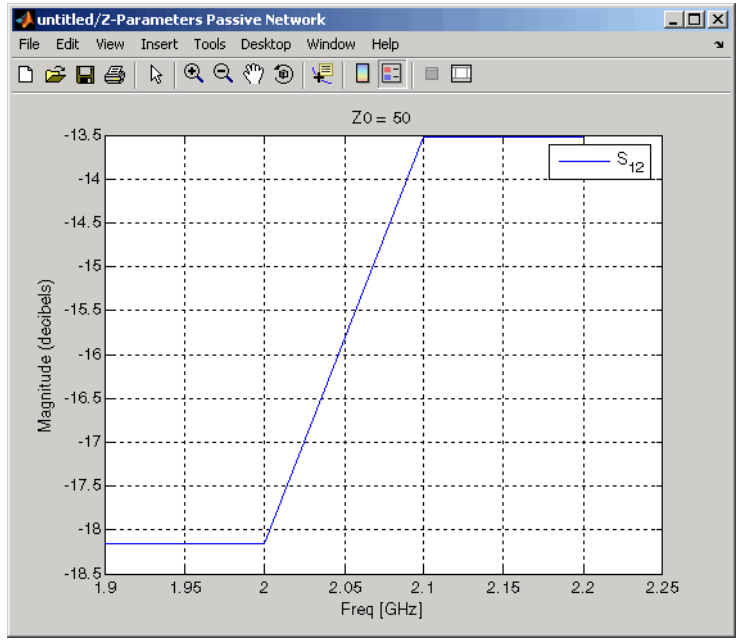

3 Set the Z-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- 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)

A

# RF Blockset Algorithms

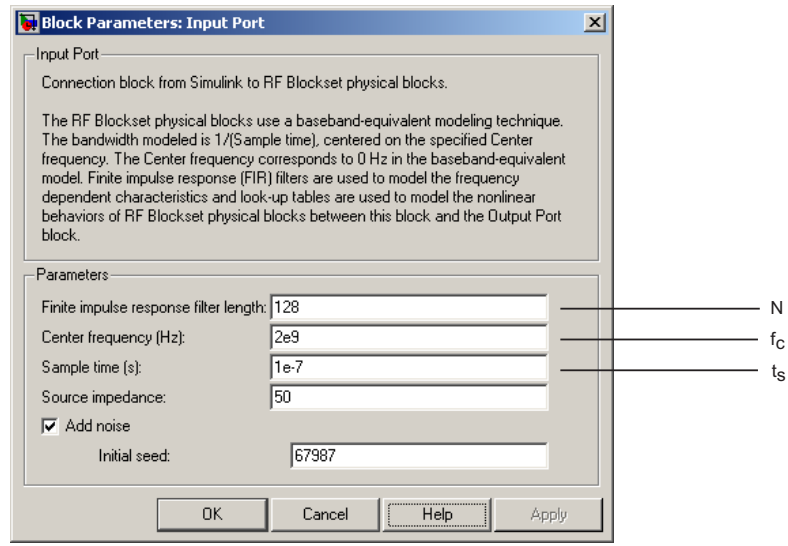
| | |
|---|---|
| Simulating an RF Model (p. A-2) | Gives an overview of how RF Blockset simulates an RF model. |
| Determining the Modeling Frequencies (p. A-3) | Describes how RF Blockset computes the modeling frequencies of a physical system. |
| Mapping Network Parameters to Modeling Frequencies (p. A-4) | Describes how RF Blockset determines the values of the S-parameters of the physical blocks at the modeling frequencies. |
| Modeling Noise in an RF System (p. A-6) | Describes how RF Blockset models noise in an RF system. |
| Creating a Complex Baseband-Equivalent Model (p. A-12) | Describes how RF Blockset uses the frequency-domain parameters of the RF blocks to create a baseband-equivalent model for time-domain simulation. |
| Converting to and from Simulink Signals (p. A-17) | Explains how the Input Port and Output Port convert Simulink signals to and from the RF Blockset physical modeling environment during a simulation. |

# Simulating an RF Model

When you simulate a model that contains physical blocks, RF Blockset 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, RF Blockset determines the block parameter values at those frequencies and uses the information to create a baseband-equivalent model for time-domain simulation in Simulink.

# 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 *n*th 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

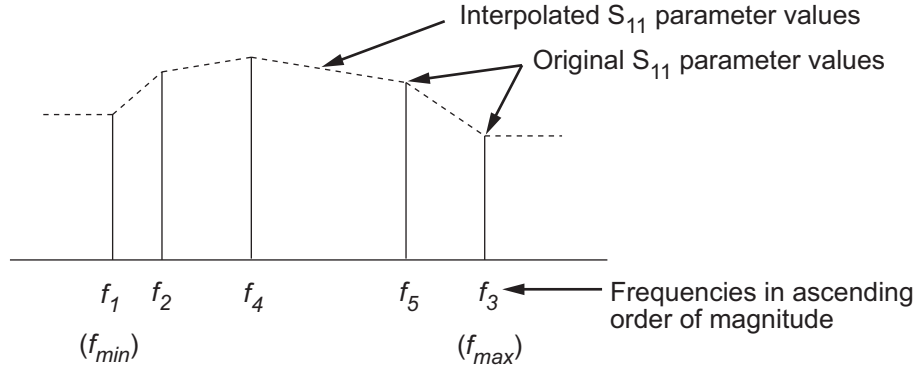
$$f_{\min} = f_c - \frac{1}{2t_s}$$

# 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 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 "One-Dimensional Interpolation" and 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, RF Blockset simulates the noise of the physical system by combining the noise contributions from each individual block. This section explains how RF Blockset simulates noise from user-specified information. For information on how to add noise to an RF model, see "Modeling Noise" on page 2-28.

This section contains the following topics:


- "Output-Referred Noise in RF Models" on page A-6
- "Calculating Noise Figure at Modeling Frequencies" on page A-9
- "Calculating System Noise Figure" on page A-10
- "Calculating Output Noise Power" on page A-10


## 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.

RF Blockset lets you simulate the noise associated with any physical block in your RF model.

RF Blockset automatically determines the noise properties of passive blocks from their network parameters. RF 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.

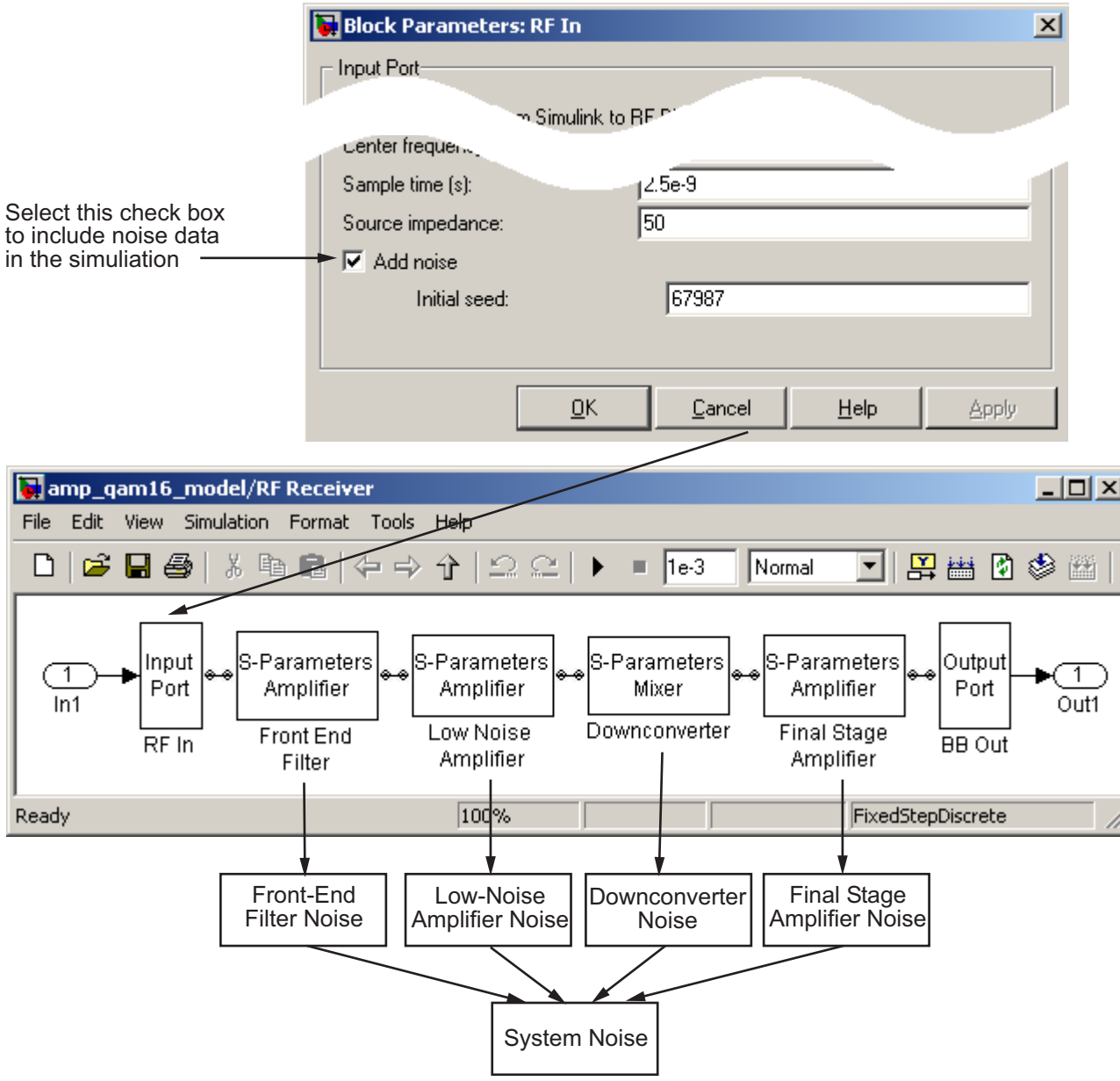
For active devices such as amplifiers and mixers, the noise properties can not be inferred from network parameters. Therefore, for the amplifier and mixer blocks, you must specify the noise information explicitly, either in the dialog block or the associated data file.

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 2-28.

When you run the simulation, RF 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
Input Port
Simulink to RF
Center frequency
Sample time (s):
2.5e-9
Source impedance:
50
Add noise
Initial seed:
67987
OK
Cancel
Help
Apply
Select this check box to include noise data in the simulation
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
Front-End Filter Noise
Low-Noise Amplifier Noise
Downconverter Noise
Final Stage Amplifier Noise
System Noise

## Calculating Noise Figure at Modeling Frequencies

To include noise information in a simulation, RF 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, RF 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, RF 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, RF 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, RF 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, RF 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.

RF Blockset then calculates the noise factor, `F`, from the noise correlation matrix as follows:

$$F = 1 + \frac{z^{+} C_A z}{2kT\,\mathrm{Re}\{Z_S\}}$$

$$z = \begin{bmatrix} 1 \\ Z_S{}^{*} \end{bmatrix}$$

In the two preceding equations, $Z_S$ is the nominal impedance, which is 50 ohms, and $z^+$ is the Hermitian conjugation of $z$.

RF 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 RF 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:

$$NF = NF_1 + \frac{NF_2 - 1}{G_1} + \frac{NF_3 - 1}{G_1 \cdot G_2} + \ldots + \frac{NF_N - 1}{G_1 \cdot G_2 \cdot \ldots \cdot G_{N-1}}$$

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

RF 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.

RF 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

RF Blockset simulates the physical system in the time domain using a baseband-equivalent model that it creates from the frequency-domain parameters of the physical blocks.

This section contains the following topics:


- "Baseband-Equivalent Modeling" on page A-12
- "Simulation Efficiency of a Complex Baseband-Equivalent Model" on page A-15


## Baseband-Equivalent Modeling

To create a complex baseband-equivalent model in the time domain based on the network parameters of the physical system, RF Blockset performs a mathematical transformation that consists of the following three steps:


1. "Calculating the Passband Transfer Function" on page A-12
2. "Calculating the Baseband Transfer Function" on page A-14
3. "Calculating the Baseband-Equivalent Impulse Response" on page A-15


### Calculating the Passband Transfer Function

RF Blockset calculates the passband transfer function from the physical block parameters at the modeling frequencies.

---

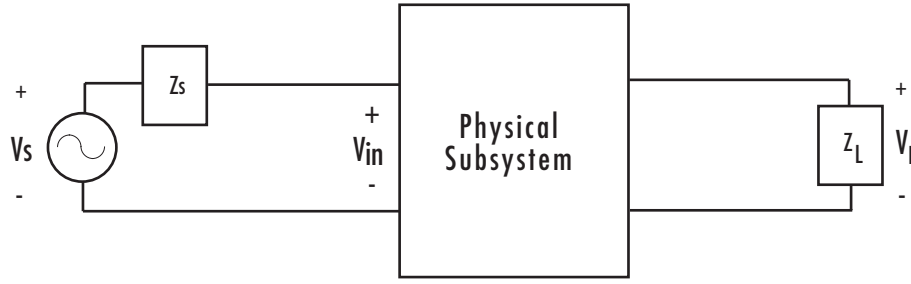
**Note** To learn how RF Blockset uses the specified network parameters to compute the network parameters at the modeling frequencies, see "Mapping Network Parameters to Modeling Frequencies" on page A-4.

---

The transfer function 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

$$\Gamma_l = \frac{Z_l - Z_o}{Z_l + Z_o}$$

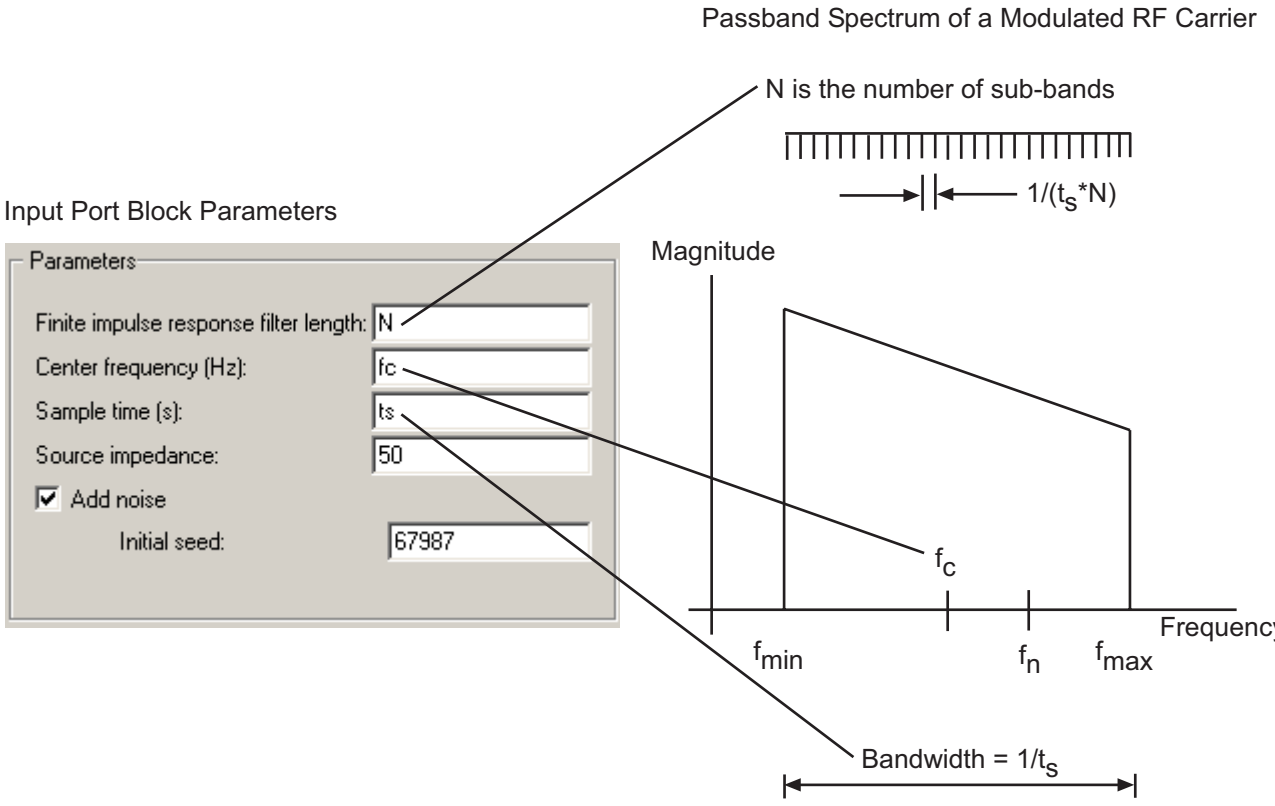
$$\Gamma_s = \frac{Z_s - Z_o}{Z_s + Z_o}$$

$$\Gamma_{in} = S_{11} + \left( S_{12} * S_{21} * \frac{\Gamma_l}{(1 - S_{22} * \Gamma_l)} \right)$$

and

- $Z_S$ is the source impedance.
- $Z_L$ is the load impedance.
- $S_{ij}$ are the S-parameters of a two-port network.

RF Blockset derives the passband transfer function from the Input Port block parameters as shown in the following figure:




## Calculating the Baseband Transfer Function

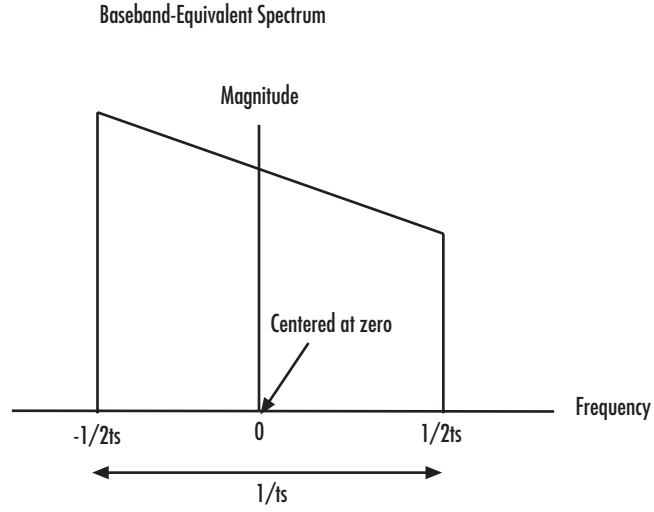
RF Blockset calculates the baseband transfer function, $H_{baseband}(f)$, by translating the passband transfer function to its equivalent baseband transfer function:

$$H_{baseband}(f) = H_{passband}(f + f_c)$$

where $f_c$ is the specified center frequency.

The resulting baseband-equivalent spectrum is centered at zero, so RF 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 Complex Baseband-Equivalent Model” on page A-15.

The baseband transfer function is shown in the following figure:




### Calculating the Baseband-Equivalent Impulse Response

RF Blockset calculates 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.

## Simulation Efficiency of a Complex 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:

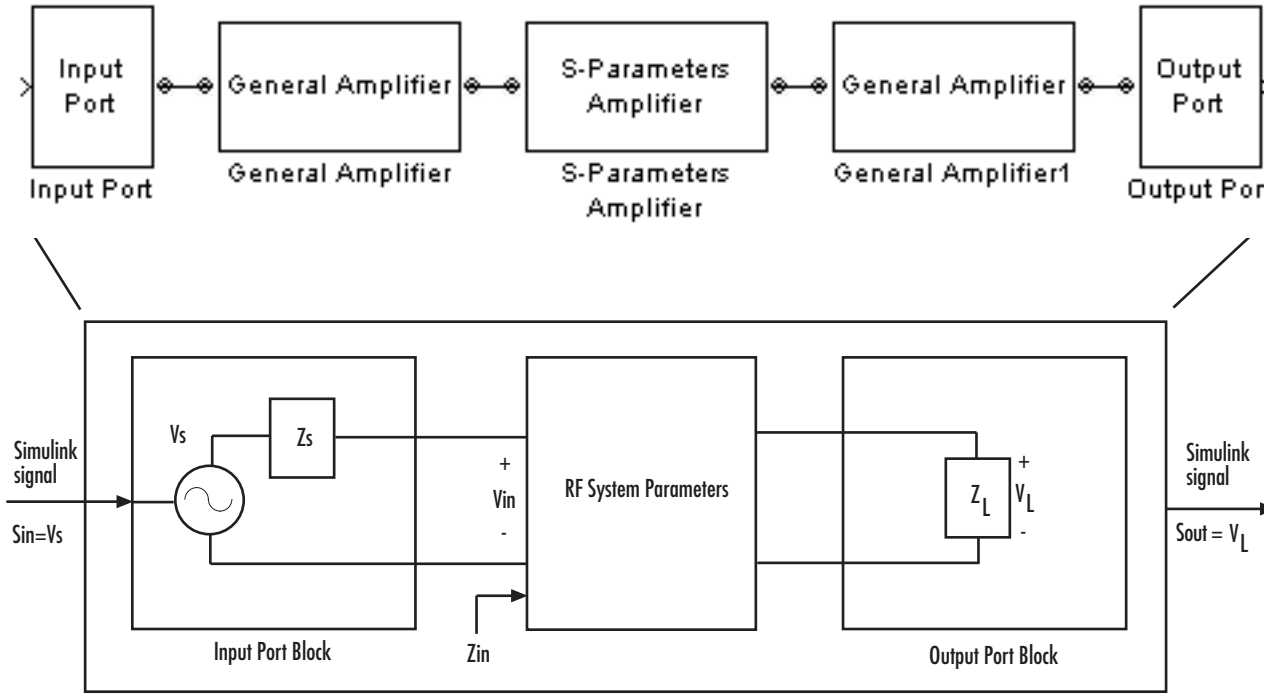
$$t_{step} = \frac{1}{2f_{\max}}$$

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:

$$t_{step} = \frac{1}{2(f_{\max} - f_c)} = \frac{1}{f_{\max} - f_{\min}}$$

# Converting to and from Simulink Signals

When you simulate an RF model, RF Blockset must convert the mathematical Simulink signals to and from the RF Blockset physical modeling environment. To perform this conversion, RF Blockset interprets the Simulink signal that enters the Input Port block, $S_{in}$, as the voltage $V_S$ across the source impedance $Z_S$. This process is shown in the following figure.




RF Blockset interprets the output Simulink signal as the voltage $V_L$ over the load impedance $Z_L$. You specify the source and load impedances in the Input Port and Output Port dialog boxes, respectively.

The RF Blockset interpretation of the input Simulink signal as the source voltage, $V_S$, produces different results than the interpretation that the input Simulink signal is the input voltage, $V_{in}$. When the source and load

impedances are the same, the former interpretation produces 6 dB of loss compared to the latter. You can see how this loss arises by analyzing the equivalent circuit model for each interpretation when there are no blocks between the Input Port and Output Port blocks.

The following figure shows the equivalent circuit model when the input Simulink signal, $S_{in}$, is taken to be the input voltage shown in the previous diagram.

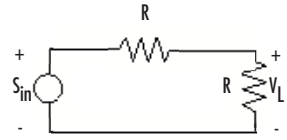
The load power is

$$P_L = \frac{V_L^2}{R} = \frac{S_{in}^2}{R}$$

In decibels, the load power is

$$10\log(P_L) = 10\log\left(\frac{S_{in}^2}{R}\right) \qquad \textbf{(A-1)}$$

The following figure shows the equivalent circuit model when the input Simulink signal, $S_{in}$, is taken to be the source voltage shown in the previous diagram.




The load power is

$$P_L = \frac{V_L^2}{R} = \frac{\left(S_{in}/2\right)^2}{R}$$

In decibels, the load power is

$$10\log(P_L) = 10\log\left(\frac{S_{in}^2}{4R}\right) = 10\log\left(\frac{S_{in}^2}{R}\right) - 6.02 \qquad \textbf{(A-2)}$$

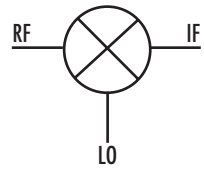
This load power is 6 dB less than the load power computed in Equation A-1.

# Modeling Mixers

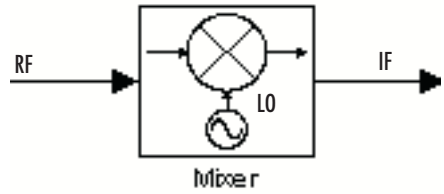

| | |
|---|---|
| 2-Port Mixer Blocks (p. B-2) | Explains why the RF Blockset mixer blocks have two ports rather than three. |
| Modeling a Mixer Chain (p. B-4) | Explains which frequencies RF Blockset models for each component when simulating a cascade that includes a mixer. |
| Quadrature Mixers (p. B-6) | Explains how to model upconversion and downconversion quadrature mixers. |

# 2-Port Mixer Blocks

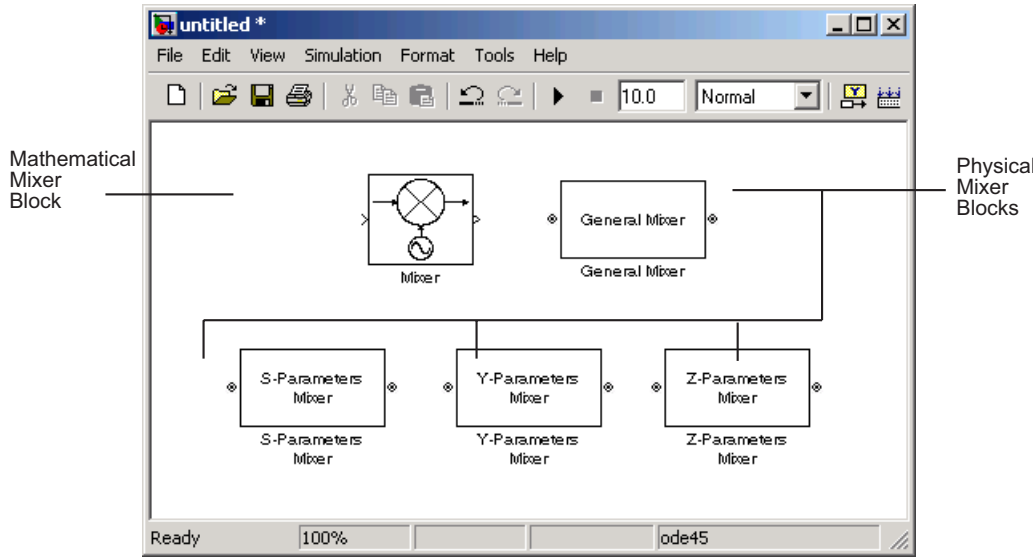
Typically, the block diagram of a mixer has three ports, as shown in the following representation of a downconversion mixer.




The RF Blockset mathematical and physical mixer blocks model both a mixer and a local oscillator, so they have only two ports.




The following figure shows the icons of the RF Blockset mixer blocks. The icon of the mathematical mixer block shows the internal local oscillator. The icons of the physical mixer blocks do not show the local oscillator, but it is in fact modeled in the same way.




For the physical blocks, you can use the **LO frequency (Hz)** parameter to specify the local oscillator's frequency.

For more information, see the individual block reference pages.

# Modeling a Mixer Chain

RF Blockset uses a baseband equivalent model to simulate RF components in the time domain, as described in Appendix A, “RF Blockset Algorithms”. The blockset only models a band of frequencies around the carrier frequency of each component; the frequency band is determined by the following parameters of the corresponding Input Port block:

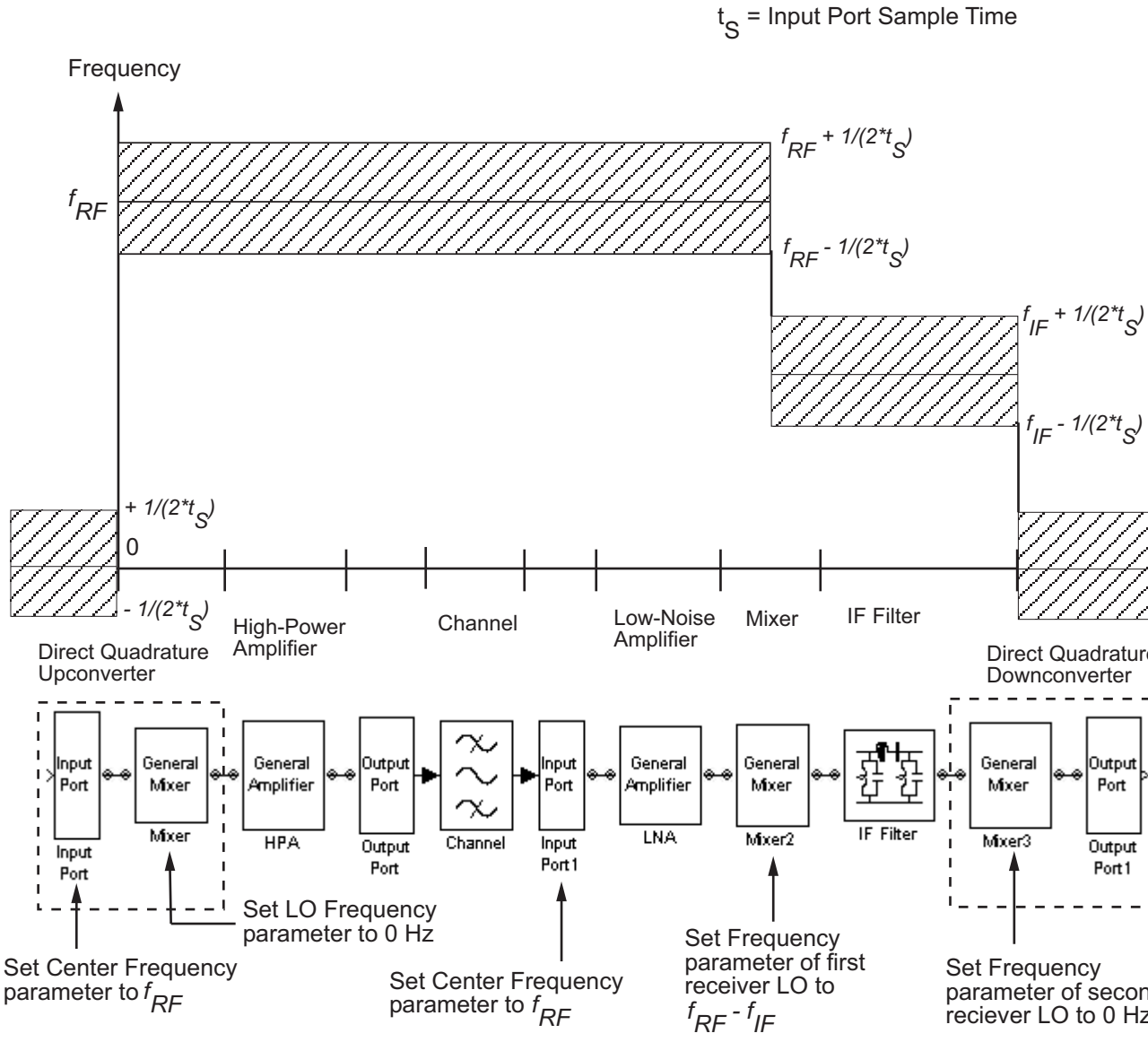
- Reciprocal of the **Sample time (s)**
- **Center frequency (Hz)**

When a mixer is present in a physical subsystem, it shifts the carrier frequency of the signal. This shift affects the frequencies that are used to create baseband equivalent model.

To illustrate how the mixer works, consider a typical RF mixer chain that consists of the following components:

- Direct Quadrature Upconverter
- High-Power Amplifier
- Channel
- Low-Noise Amplifier
- Downconverting Mixer
- IF Filter
- Direct Quadrature Downconverter

The following diagram shows these components and the band of frequencies that are simulated for each component. The signals at the input and output of the cascade are baseband complex. For the cascade, the diagram shows the real passband frequencies that are used to create the baseband-equivalent model, which is centered at zero. For a detailed explanation of how to use RF Blockset to model quadrature mixers, see “Quadrature Mixers” on page B-6.


$t_S$ = Input Port Sample Time
Frequency
$f_{RF}$ + 1/(2*$t_S$)
$f_{RF}$
$f_{RF}$ - 1/(2*$t_S$)
$f_{IF}$ + 1/(2*$t_S$)
$f_{IF}$ - 1/(2*$t_S$)
+ 1/(2*$t_S$)
0
- 1/(2*$t_S$)
Direct Quadrature Upconverter
High-Power Amplifier
Channel
Low-Noise Amplifier
Mixer
IF Filter
Direct Quadrature Downconverter
Input Port
General Mixer
Mixer
Input Port
General Amplifier
HPA
Output Port
Output Port
Channel
Input Port
Input Port1
General Amplifier
LNA
General Mixer
Mixer2
IF Filter
General Mixer
Mixer3
Output Port
Output Port1
Set LO Frequency parameter to 0 Hz
Set Center Frequency parameter to $f_{RF}$
Set Center Frequency parameter to $f_{RF}$
Set Frequency parameter of first receiver LO to $f_{RF}$ - $f_{IF}$
Set Frequency parameter of second reciever LO to 0 Hz

# Quadrature Mixers

RF Blockset lets you model upconversion and downconversion quadrature mixers using Physical blocks. These mixers convert a complex baseband signal up to and down from the desired carrier frequency by mixing the real and imaginary parts of the signal with a cosine and sine of the same frequency.

This section contains the following topics:


- "Modeling Upconversion I/Q Mixers" on page B-6
- "Modeling Downconversion I/Q Mixers" on page B-7
- "Simulating I/Q Mixers" on page B-8


## Modeling Upconversion I/Q Mixers

You use the Input Port block to model the upconversion of in-phase/quadrature baseband signals to modulated signals at a finite real carrier frequency. The real component of the block input represents the in-phase signal. The imaginary component of the block input represents the quadrature signal.

To model a perfect quadrature upconversion mixer, use the Input Port block with the **Center frequency (Hz)** parameter set to the carrier frequency.

To model an imperfect quadrature upconversion mixer, use the Input Port block with the **Center frequency (Hz)** parameter set to the carrier frequency. Follow this block immediately by a mixer block with the **LO frequency (Hz)** parameter set to `0`. Specify imperfections as follows:

- S-parameters — Use S-parameters to specify imperfections such as frequency response. For a mixer, $S_{21}$ describes the conversion gain, as explained in the Network Parameters section of the reference page for each mixer block. Use purely real and purely imaginary $S_{21}$ parameters to represent multiplying the input signal by a pure cosine and a pure sine, respectively. Use a complex $S_{21}$ parameter to represent multiplying the input signal by a combination of sine and cosine.
- Thermal noise — Use thermal noise to specify temperature-dependent random noise.

- Phase noise — Use the phase noise to specify noise to add to the angle component of the input signal.
- Nonlinearity — Use nonlinearity (specified as output power and phase as a function of input power and frequency in an AMP file or as third-order intercept point) to specify nonlinear mixer behavior as a function of input power.

---

**Note** If you specify a nonzero value for the local oscillator frequency of the mixer and set the **Type** parameter to `Upconverter`, RF Blockset converts the signal to a frequency above the center frequency. The final IF value is the sum of the Input Port center frequency and the mixer local oscillator frequency.

---

## Modeling Downconversion I/Q Mixers

You use the Output Port block to model the downconversion of in-phase/quadrature modulated carrier signals to baseband signals. The real component of the block output represents the in-phase signal. The imaginary component of the block output represents the quadrature signal.

The finite real carrier frequency is set automatically as the sum of the center frequency of the Input Port block and the LO frequencies in any mixer blocks in the cascade.

---

**Note** In the cascade, upconversion mixers increase the carrier frequency and downconversion mixers decrease the carrier frequency.

---

The Output Port block models a perfect quadrature downconversion mixer. To model an imperfect quadrature downconversion mixer, precede the Output Port block immediately by a mixer block with the **LO frequency (Hz)** parameter set to `0`. Specify imperfections as follows:

- S-parameters — Use S-parameters to specify imperfections such as frequency response. For a mixer, $S_{21}$ describes the conversion gain, as explained in the Network Parameters section of the reference page for each mixer block. Use purely real and purely imaginary $S_{21}$ parameters to represent multiplying the input signal by a pure cosine and a pure sine,

respectively. Use a complex $S_{21}$ parameter to represent multiplying the input signal by a combination of sine and cosine.

- Thermal noise — Use thermal noise to specify temperature-dependent random noise.
- Phase noise — Use the phase noise to specify noise to add to the angle component of the input signal.
- Nonlinearity — Use nonlinearity (specified as output power and phase as a function of input power and frequency in an AMP file or as third-order intercept point) to specify nonlinear mixer behavior as a function of input power.

## Simulating I/Q Mixers

When you model an I/Q mixer in RF Blockset, the center frequency you specify in the Input Port block dialog is only used to build a baseband-complex equivalent model of the cascade that represents the mixer. RF Blockset simulates this model using a fixed time step equal to the sample time that you specify in the Input Port block dialog box.

To examine the model in the Simulink window:

1. Select **Edit > Update Diagram** to update the model diagram.
2. Right-click the Output Port block and select **Look Under Mask**.

For more information on baseband-equivalent modeling, see Appendix A, “RF Blockset Algorithms”.

C

# Examples

Use this list to find examples in the documentation.

# Examples

"Example — Modeling an LC Bandpass Filter" on page 1-12
"Example — Importing a Touchstone Data File into an RF Model" on page 2-9
"Example — Importing a Bandstop Filter into an RF Model" on page 2-16
"Example — Plotting Component Data on a Z Smith Chart" on page 3-20
"Example — Creating and Modifying Subsystem Plots" on page 3-29

# Index

## A


active noise
  General Amplifier block 5-40
  General Mixer block 5-57
  S-Parameters Amplifier block 5-149
  S-Parameters Mixer block 5-161
  Y-Parameters Amplifier block 5-198
  Y-Parameters Mixer block 5-208
  Z-Parameters Amplifier block 5-227
  Z-Parameters Mixer block 5-237
adding
  noise 2-29
  nonlinearity 2-26
adding physical and mathematical blocks 2-2
algorithm
  simulation A-2
amplifier
  modeling from file data 5-50
  modeling from `rfdata` object 5-38
Amplifier block 5-2
  effects 5-5
  modeling nonlinearity 5-2
    methods 5-7
  sample time comparison 5-11
  thermal noise simulation 5-10
  *See also* General Amplifier block
amplifier blocks
  mathematical 4-2
  physical 4-4


## B


Bandpass RF Filter block 5-16
  sample time comparison 5-17
Bandstop RF Filter block 5-20
  sample time comparison 5-21
baseband-equivalent model
  creating A-12
  simulation efficiency A-15
baseband-equivalent time domain
    modeling 5-125
    linear subsystem 5-125
    nonlinear subsystem 5-129
Bessel filter design
    Bandpass RF Filter block 5-16
    Bandstop RF Filter block 5-20
    Highpass RF Filter block 5-77
    Lowpass RF Filter block 5-112
black box elements blocks 4-4
block parameters
    setting 2-7
blocks
    connecting 2-2
    mathematical 4-2
    physical 4-3
Butterworth filter design
    Bandpass RF Filter block 5-16
    Bandstop RF Filter block 5-20
    Highpass RF Filter block 5-77
    Lowpass RF Filter block 5-112


## C


Chebyshev I filter design
    Bandpass RF Filter block 5-16
    Bandstop RF Filter block 5-20
    Highpass RF Filter block 5-77
    Lowpass RF Filter block 5-112
Chebyshev II filter design
    Bandpass RF Filter block 5-16
    Bandstop RF Filter block 5-20
    Highpass RF Filter block 5-77
    Lowpass RF Filter block 5-112
circuit element
    modeling from `rfckt` object 5-54
circuit objects
    importing 2-15
Coaxial Transmission Line block 5-24
    shunt and series stubs 5-25

stubless 5-24
composite data plots 3-5
connecting blocks
RF Physical blocks to Simulink 2-2
Simulink to RF Physical blocks 2-2
Connection Port block 5-30
connector blocks 4-5
Connection Port 5-30
Input Port 5-81
Output Port 5-125
conversion
Simulink signal to RF signal A-17
Coplanar Waveguide Transmission Line
block 5-32
shunt and series stubs 5-33
stubless 5-32


## D


data consistency
General Amplifier block 5-41
data files
importing 2-7
demos
in Help browser 1-4


## E


Elliptic filter design
Bandpass RF Filter block 5-16
Bandstop RF Filter block 5-20
Highpass RF Filter block 5-77
Lowpass RF Filter block 5-112
example
creating and modifying plots 3-29
importing circuit objects 2-16
importing Touchstone file 2-9


## F


filter blocks
mathematical 4-2
physical 4-3
filter design
physical blocks
LC Lowpass Tee block 5-109
filter design mathematical blocks
Bandpass RF Filter block 5-16
Bandstop RF Filter block 5-20
Highpass RF Filter block 5-77
Lowpass RF Filter block 5-112
filter design physical blocks
LC Bandpass Pi block 5-84
LC Bandpass Tee block 5-91
LC Bandstop Pi block 5-94
LC Bandstop Tee block 5-97
LC Highpass Pi block 5-100
LC Highpass Tee block 5-103
LC Lowpass Pi block 5-106
format
changing in plot 3-26
frequency range
changing in plot 3-26


## G


General Amplifier block 5-38
active noise 5-40
generating from file data 5-50
nonlinearity 5-38
operating conditions 5-41
*See also* Amplifier block
General Circuit Element block 5-53
generating from `rfckt` object 5-54
General Mixer block 5-57
active noise 5-57
nonlinearity 5-59
operating conditions 5-61
phase noise 5-58
*See also* Mixer block
General Passive Network block 5-71

generating from file data 5-73
Ghorbani model 5-9

## H

Help browser
demos 1-4
Highpass RF Filter block 5-77
sample time comparison 5-78

## I

importing
circuit objects 2-15
data files 2-7
Input Port block 5-81
parameter input for physical subsystem 5-81
sample time comparison 5-81
*See also* Output Port block
interpolation
network parameters A-4
IP3
specifying mixer and amplifier 2-25

## L

ladder filter blocks 4-3
ladder filters
example to filter Gaussian noise 5-86
LC Bandpass Pi block 5-84
example to filter Gaussian noise 5-86
LC Bandpass Tee block 5-91
LC Bandstop Pi block 5-94
LC Bandstop Tee block 5-97
LC Highpass Pi block 5-100
LC Highpass Tee block 5-103
LC Lowpass Pi block 5-106
LC Lowpass Tee block 5-109
libraries
RF physical and mathematical 1-6
library
RF mathematical 1-9
RF physical 1-7
link budget plots 3-9
Lowpass RF Filter block 5-112
sample time comparison 5-113

## M

mathematical and physical models
connecting 2-2
mathematical modeling blocks 1-9
Microstrip Transmission Line block 5-116
shunt and series stubs 5-117
stubless 5-116
Mixer block 5-122
*See also* General Mixer block
mixer blocks
mathematical 4-2
physical 4-5
mixers
how to model B-1
up and down quadrature B-1
model
adding RF components to 2-2
modeling
amplifier from file data 5-50
amplifier from `rfdata` object 5-38
baseband-equivalent time domain 5-125
circuit element from `rfckt` object 5-54
passive network from file data 5-73
modeling frequencies
calculating A-3
calculating noise A-9
S-parameter interpolation A-4
models
physical and mathematical blocks 1-6

## N

network parameters

adding to a plot 3-26
interpolation at modeling frequencies A-4
noise
active
General Amplifier block 5-40
General Mixer block 5-57
S-Parameters Amplifier block 5-149
S-Parameters Mixer block 5-161
Y-Parameters Amplifier block 5-198
Y-Parameters Mixer block 5-208
Z-Parameters Amplifier block 5-227
Z-Parameters Mixer block 5-237
adding 2-29
calculating at modeling frequencies A-9
calculating subsystem A-10
phase
General Mixer block 5-58
S-Parameters Mixer block 5-161
Y-Parameters Mixer block 5-209
Z-Parameters Mixer block 5-238
plotting 2-32
specifying mixer and amplifier 2-28
noise output power
calculating A-10
noise simulation A-6
nonlinearity
adding 2-26
Amplifier block 5-2
General Amplifier block 5-38
General Mixer block 5-59
S-Parameters Amplifier block 5-146
S-Parameters Mixer block 5-161
specifying mixer and amplifier 2-25
Y-Parameters Amplifier block 5-195
Y-Parameters Mixer block 5-209
Z-Parameters Amplifier block 5-225
Z-Parameters Mixer block 5-238


## O


opening
block libraries 1-6
operating conditions
General Amplifier block 5-41
General Mixer block 5-61
Output Port block 5-125
baseband-equivalent time domain modeling 5-125
plotting RF subsystem parameters 5-133
*See also* Input Port block
output power
calculating noise A-10


## P


Parallel-Plate Transmission Line block 5-134
shunt and series stubs 5-135
stubless 5-134
parameter input
to physical subsystem 5-81
parameters
adding to a plot 3-26
passive network
modeling from file data 5-73
phase noise
General Mixer block 5-58
S-Parameters Mixer block 5-161
Y-Parameters Mixer block 5-209
Z-Parameters Mixer block 5-238
physical and mathematical models
connecting blocks 2-2
physical modeling blocks 1-7
physical subsystem
parameter input for 5-81
plot types
modifying 3-26
plots
budget 3-9
changing format of 3-26

changing frequency range of 3-26
composite 3-5
creating 3-13
displaying legend 3-20
example of creating and modifying 3-29
format 3-4
link budget 3-9
modifying 3-26
noise 2-32
RF subsystem parameters 5-133
Smith chart example 3-20
types of 3-3
updating 3-25
using block parameters to specify 3-13
X-Y plane 3-7
product demos 1-4

## Q

quadrature mixers B-1

## R

Rapp model 5-10
RF block libraries
Mathematical 4-2
Physical 4-3
RF Blockset
required and related products 1-3
RF Blockset libraries 1-6
how to open 1-6
RF models
connecting blocks 2-2
`rfckt` object
modeling general circuit element 5-53
`rfdata` object
modeling nonlinear amplifier 5-38
RLCG Transmission Line block 5-140
shunt and series stubs 5-141
stubless 5-140

## S

S-Parameters Amplifier block 5-146
active noise 5-149
nonlinearity 5-146
S-Parameters Mixer block 5-160
active noise 5-161
nonlinearity 5-161
phase noise 5-161
S-Parameters Passive Network block 5-171
Saleh model 5-8
Series RLC block 5-178
setting
block parameters 2-7
Shunt RLC block 5-181
simulation
noise A-6
simulation algorithm
overview A-2
Simulink signal
converting to and from A-17
Smith chart
example 3-20
specifying
mixer and amplifier noise 2-25
mixer and amplifier thermal noise 2-28
stubless transmission lines
Coaxial Transmission Line block 5-24
Coplanar Waveguide Transmission Line block 5-32
Microstrip Transmission Line block 5-116
Parallel-Plate Transmission Line block 5-134
RLCG Transmission Line block 5-140
Transmission Line block 5-184
Two-Wire Transmission Line block 5-189
stubs (shunt and series)
Coaxial Transmission Line block 5-25
Coplanar Waveguide Transmission Line block 5-33
Microstrip Transmission Line block 5-117
Parallel-Plate Transmission Line block 5-135

RLCG Transmission Line block 5-141
Transmission Line block 5-185
Two-Wire Transmission Line block 5-190
subsystem
calculating noise A-10

## T

thermal noise
specifying mixer and amplifier 2-28
time-domain modeling
baseband-equivalent 5-125
Touchstone file
importing 2-9
Transmission Line block 5-184
shunt and series stubs 5-185
stubless 5-184
transmission line blocks 4-3
Two-Wire Transmission Line block 5-189
shunt and series stubs 5-190
stubless 5-189

## V

viewing
demos 1-4

## W

workflow
RF Blockset 1-10
workflow example
RF Blockset 1-12

## Y

Y-Parameters Amplifier block 5-195
active noise 5-198
nonlinearity 5-195
Y-Parameters Mixer block 5-208
active noise 5-208
nonlinearity 5-209
phase noise 5-209
Y-Parameters Passive Network block 5-218

## Z

Z-Parameters Amplifier block 5-224
active noise 5-227
nonlinearity 5-225
Z-Parameters Mixer block 5-237
active noise 5-237
nonlinearity 5-238
phase noise 5-238
Z-Parameters Passive Network block 5-247