RF Blockset

For Use with Simulink®

- Modeling
- Simulation
- Implementation

User's Guide

Version 1

The MathWorks

How to Contact The MathWorks:

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

508-647-7000 (Phone)

508-647-7001 (Fax)

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

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

RF Blockset User's Guide

Revision History

June 2004	Online only	New for Version 1.0 (Release 14)
August 2004	Online only	Revised for Version 1.0.1 (Release 14+)
March 2005	Online only	Revised for Version 1.1 (Release 14SP2)
September 2005	Online only	Revised for Version 1.2 (Release 14SP3)
March 2006	Online only	Revised for Version 1.3 (Release 2006a)

Contents

1 Getting Started

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

RF Blockset Libraries **1-3**

Data Visualization **1-4**

Data Format Support **1-5**

Key Blockset Concepts **1-6**
Baseband-Equivalent Modeling **1-6**
Modeling Paradigm **1-9**
Noise **1-19**
Mixers **1-27**
Interpretation of Simulink Signals **1-28**

Demos **1-30**

2 Creating an RF Model

RF Models **2-2**

Creating a Model **2-3**
RF Blockset Libraries **2-3**
Connecting RF Blocks **2-4**
Setting RF Blockset Block Parameters **2-6**

3 Working with RF Models

S-Parameters at Simulation Frequencies **3-2**
Determining the Simulation Frequencies **3-2**
Interpolating the S-Parameters **3-2**

Using File Data **3-4**

Importing Circuits from the MATLAB Workspace **3-9**
References **3-13**

4 Plotting Network Parameters

Generating a Plot **4-2**
Block Parameters **4-2**
Example of a Z Smith Chart **4-5**
Examples of Other Plot Types **4-7**

Modifying a Plot **4-10**
Example of Plot Modification **4-10**

5 Blocks — By Category

Main RF Blockset Library **5-2**

Mathematical **5-3**

Physical **5-5**
Ladder Filters **5-5**
Transmission Lines **5-6**
Black Box Elements **5-7**

Amplifiers 5-8
Mixers 5-9
Input/Output Ports 5-10

6 Blocks — Alphabetical List

A Examples

Examples **A-2**

Index

1

Getting Started

What Is the RF Blockset? (p. 1-2)	Introduces the RF Blockset and describes its capabilities.
RF Blockset Libraries (p. 1-3)	Summarizes the RF Blockset libraries.
Data Visualization (p. 1-4)	Lists the plots you can generate.
Data Format Support (p. 1-5)	Lists the data formats that the RF Blockset supports.
Key Blockset Concepts (p. 1-6)	Describes how the RF Blockset works.
Demos (p. 1-30)	Explains where to find RF Blockset demos.

What Is the RF Blockset?

The RF Blockset is a tool for design, analysis, and simulation of the RF (Radio Frequency) portion of wireless communications systems. It uses complex baseband-equivalent behavioral modeling to model and analyze RF systems in the time domain.

RF technology is used to design and test RF circuits for cable television, wireless LAN, and other wireless applications such as broadcasting, radar, satellite communications, microwave relay, and mobile telephony.

The RF Blockset enables you to assemble complex RF systems from libraries of RF components such as ladder filters, transmission lines, black box elements, amplifiers, and mixers. You can include Simulink® Signal Processing Blockset, and Communications Blockset blocks in your model, as well as blocks from other MathWorks blocksets.

The RF Blockset contains

- Physical blocks, which describe RF components in terms of their geometry, physical interactions, and measured data. Internally, the RF Blockset characterizes these RF components in terms of their network parameters, and constructs the behavioral model for the cascade of components from those parameters.
- Mathematical blocks, which are Simulink style blocks that describe components in mathematical terms. Each block models the component's behavior according to predefined mathematical relationships.

Two specialized blocks let you include mathematical and physical blocks in the same model by providing transitions from a Simulink signal into the RF Blockset physical modeling environment and back into a Simulink signal. These blocks also enable you to specify certain parameters that are common to a physical modeling subsystem.

The RF Blockset works with the RF Toolbox. You can create complex topologies with the RF Toolbox, and then read them into RF Blockset blocks for inclusion in an RF model.

RF Blockset Libraries

The RF Blockset has two main libraries, Physical and Mathematical.

The Physical library includes the following sublibraries. Use blocks from these sublibraries to model the physical or electrical structure of portions of RF systems.

- Amplifiers — RF amplifiers described by S-, Y-, or Z-parameters, noise figure, and IP3, or a data file containing these parameters.
- Ladder Filters — RF filters whose network parameters can be calculated from their topologies.
- Mixers — RF mixers described by S-, Y-, or Z-parameters and phase noise, or a data file containing these parameters.
- Transmission Lines — RF filters whose network parameters can be calculated from their geometry.
- Black Box — Passive RF components described by S-, Y-, or Z-parameters, or a data file containing these parameters.
- Input/Output Ports — Blocks that connect mathematical portions of the model to the physical portions. Mathematical portions of the model may also include blocks from Simulink and other blocksets that describe components in mathematical terms.

The Mathematical library includes amplifier, mixer, and filter blocks. These blocks provide mathematical equivalents of the RF components.

Note Mathematical blocks assume perfect impedance matching and a nominal impedance of 1 ohm. In contrast, the physical blocks do not assume perfect matching. These blocks model the reflections that occur between blocks, and you can specify the source and load impedances using the Input Port and Output Port blocks.

For more information about these libraries, see Chapter 5, “Blocks — By Category”.

Data Visualization

The RF Blockset lets you plot the network parameters of the component blocks in the Physical library. For each such block, you can generate an X-Y plane plot, polar plane plot, or Smith® chart of selected network parameters in a specified frequency range. The dialog box for each block contains the parameters you need to specify the plot.

The RF Blockset also provides a composite plot that includes four separate plots in one figure. This predefined combination of plots differs based on the type of block.

Data Format Support

The RF Blockset supports the Touchstone® S2P, Y2P, Z2P, and H2P data file formats. It also introduces the MathWorks AMP format for amplifier data. The AMP format is intended for use as input to the General Amplifier block.

For information about the Touchstone file formats, see `http://www.eda.org/pub/ibis/connector/touchstone_spec11.pdf`.

For information about the AMP file format, see “AMP File Format” in the RF Toolbox documentation.

Key Blockset Concepts

The RF Blockset enables you to simulate the frequency response, noise, and nonlinearity in an RF system. In this section, you learn more about how the blockset works.

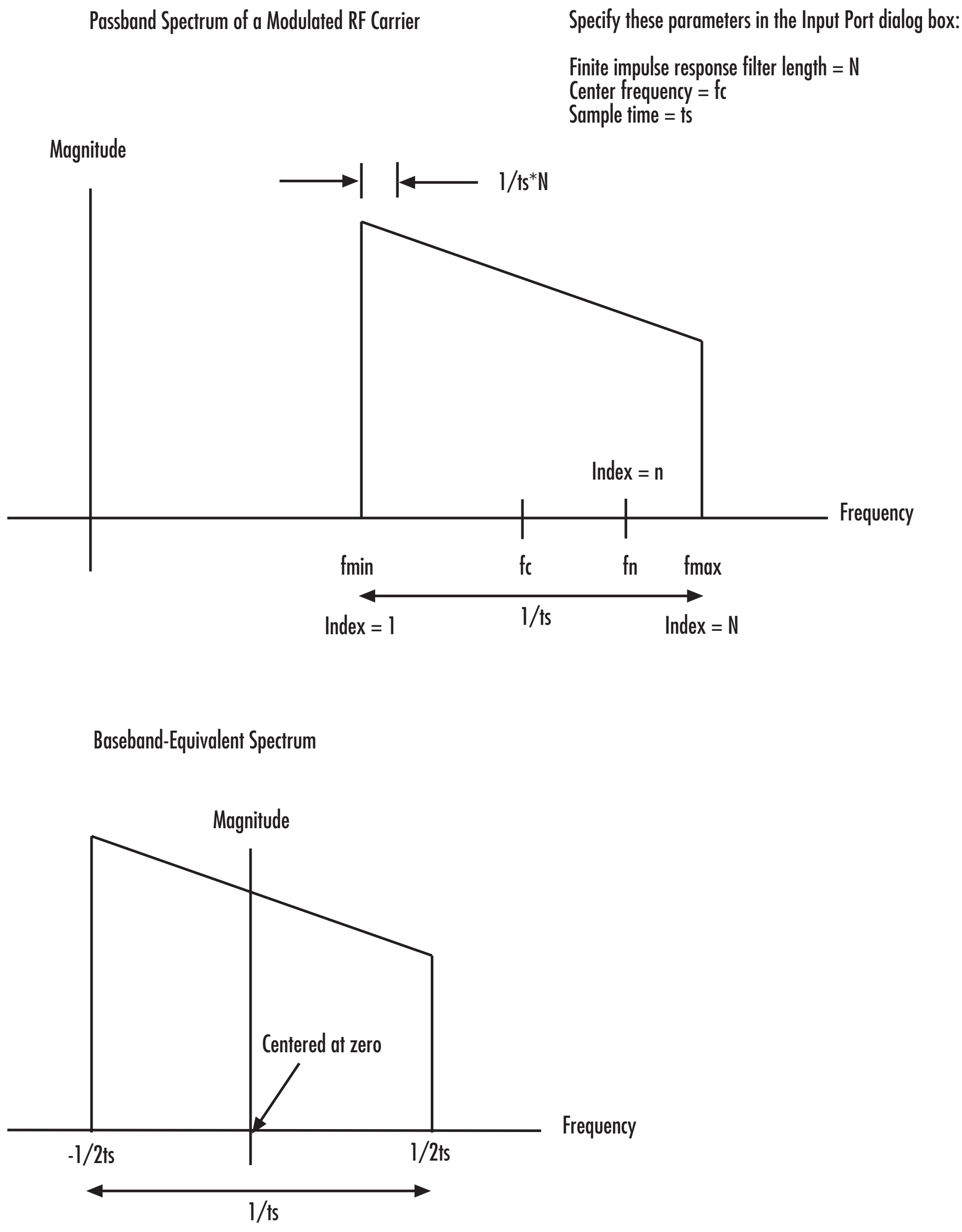
This section contains the following topics:

- "Baseband-Equivalent Modeling" on page 1-6 — Understand how the RF Blockset performs the two-step mathematical transformation to create a complex baseband-equivalent model in the time domain.
- "Modeling Paradigm" on page 1-9 — Learn how the RF Blockset uses a baseband-equivalent modeling paradigm to efficiently represent the behavior of RF signals and systems in the time domain.
- "Noise" on page 1-19 — Understand how the RF Blockset models noise.
- "Mixers" on page 1-27 — Learn how RF Blockset mathematical and physical mixer blocks model both mixers and local oscillators.
- "Interpretation of Simulink Signals" on page 1-28 — Understand how the RF Blockset interprets these input and output signals.

Baseband-Equivalent Modeling

When you run your RF model, the RF Blockset takes the frequency domain parameters you specified in the physical blocks and performs a two-step mathematical transformation to create a complex baseband-equivalent model in the time domain.

In the first step, the RF Blockset calculates the passband transfer function from the physical block parameters and translates the passband transfer function to its equivalent baseband transfer function. This is represented graphically in the following figures:


Passband Spectrum of a Modulated RF Carrier
Specify these parameters in the Input Port dialog box:
Finite impulse response filter length = N
Center frequency = fc
Sample time = ts
Magnitude
1/ts*N
Index = n
Frequency
fmin
fc
fn
fmax
Index = 1
1/ts
Index = N
Baseband-Equivalent Spectrum
Magnitude
Centered at zero
Frequency
-1/2ts
1/2ts
1/ts

In the second step, the RF Blockset calculates the baseband-equivalent impulse response by computing the inverse FFT of the baseband transfer function.

The two-step mathematical transformation used to create a complex baseband-equivalent model is reviewed in detail in the next two topics.

Step 1 — Calculating the Baseband Transfer Function

If the network parameters you supply for each block do not correspond to the simulation frequencies, the Output Block uses interpolation and extrapolation to estimate the network parameters at these frequencies. It ignores any network parameters that lie outside the simulation frequency range. Next, the block uses these estimated network parameters to calculate the passband transfer function using the following formula:

$$H(f) = \frac{V_L(f)}{V_S(f)}$$

where V_S and V_L are the source and load voltages and f represents the simulation frequencies. More specifically,

$$H(f) = \frac{-Y_{21}(f)}{(1 + Z_S(f) \cdot Y_{11}(f))(Y_{22}(f) + Z_L(f)^{-1}) - (Z_S(f) \cdot Y_{12}(f) \cdot Y_{21}(f))}$$

where

- Z_S is the source impedance.
- Z_L is the load impedance.
- Y_{ij} are the Y-parameters of a two-port network. The Output Port block derives the Y-parameters from the individual blocks' S-parameters.

The Output Port block then translates the passband transfer function to its equivalent baseband transfer function as $H(f - f_c)$, where f_c is the specified center frequency.

Step 2 — Calculating the Baseband-Equivalent Impulse Response

The Output Port block obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. If the specified finite impulse response filter length is not a power of 2, the Output Port block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. It then truncates the impulse response to a length equal to the filter length specified. This is done for reasons of efficiency. For a detailed example of this process, see "Modeling Paradigm" on page 1-9.

Modeling Paradigm

The RF Blockset uses a baseband-equivalent modeling paradigm to efficiently represent the behavior of RF signals and systems in the time domain. The example in this topic illustrates how the RF Blockset works.

Modeling an LC Bandpass Filter

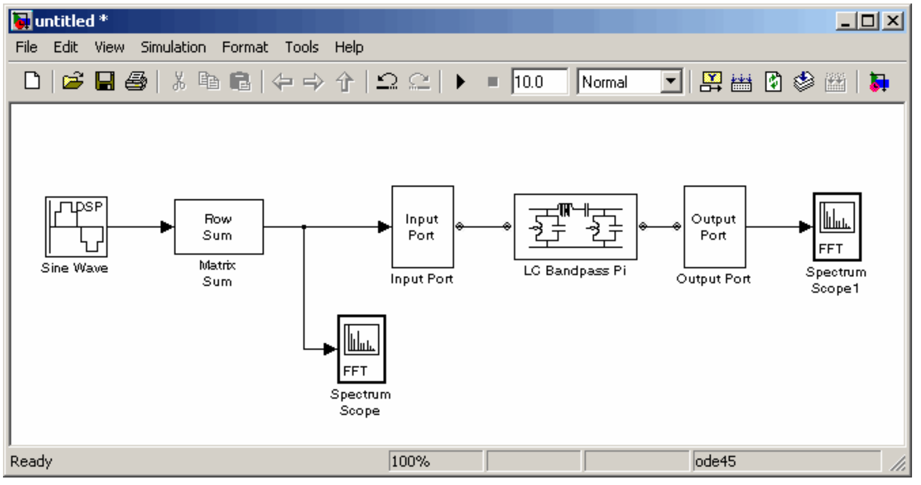
Create an LC bandpass filter with a bandwidth of 200 MHz centered at 700 MHz. You use this filter to attenuate a three-tone source signal.

1 Create a new Simulink model, and click-and-drag the following blocks into it.

Block	Library	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

Block	Library	Quantity
LC Bandpass Pi	RF Blockset / Physical / Ladder Filters	1
Output Port	RF Blockset / Physical / Input/Output Port	1

2 Connect the blocks so that your model resembles the following figure.

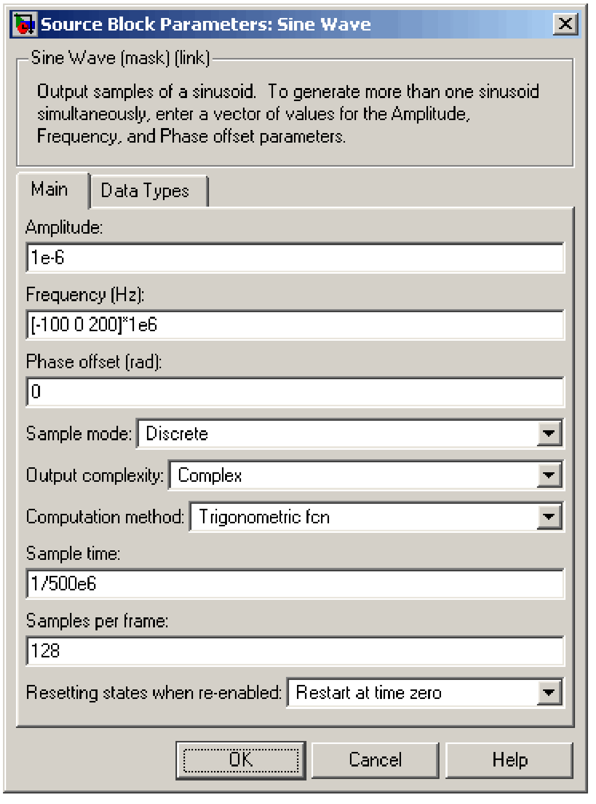


Note By default, the Matrix Sum block sums the matrix elements along the column dimension. In the next step you will change the block parameters to specify summation along the row dimension, and your model will match the one shown above.

3 Use the Sine Wave and Matrix Sum blocks to build a three-tone source signal. The actual signal you want to model has one tone at 700 MHz, the center of the filter, one tone at 600 MHz, the lower edge of the passband, and a third tone at 900 MHz, outside the passband. However, because the input to the Input Port block must be the baseband-equivalent signal of the actual RF signal you want to model, you create a three-tone source signal with three tones at -100 MHz, 0 MHz, and 200 MHz, respectively.

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`



Because the baseband-equivalent of a real passband signal is complex, you set the **Output complexity** parameter to `Complex`.

The output of the Sine Wave block is a three-channel signal, where each channel corresponds to a different frequency. Use the Matrix Sum block to combine the three channels by setting its **Sum along** parameter to `Rows`.

4 Use the Spectrum Scope block to view the three-tone source signal. On the **Axis Properties** pane, set the block parameters as follows:

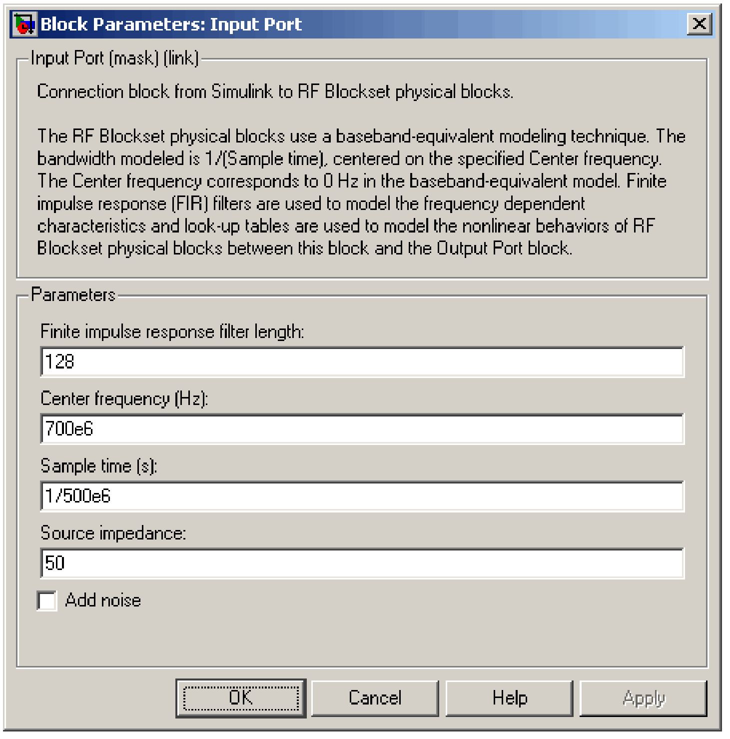
- **Frequency range** = `[-Fs/2...Fs/2]`
- **Minimum Y-limit** = `-160`
- **Maximum Y-limit** = `-100`

Setting the **Scope position** parameter lets you see both scope windows when you run the model.

5 Use the Input Port block to specify the portion of the RF spectrum that the physical blocks model. Set the block parameters as follows:

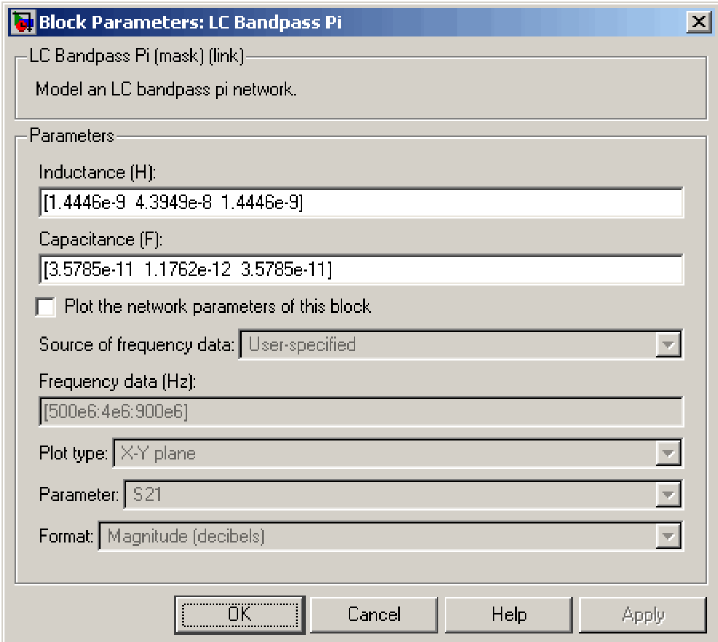
- **Center frequency** = `700e6`
- **Sample time(s)** = `1/500e6`

- Clear the **Add noise** check box.



The Input Port block cannot inherit a sample time so you must specify it in the block parameters dialog box. Set the **Sample time(s)** parameter to the same value as the **Sample time** parameter in the Sine Wave block. By specifying a sample time of `1/500e6` second, you establish a modeling bandwidth of 500 MHz. For more information about the **Source impedance** parameter, see "Interpretation of Simulink Signals" on page 1-28.

6 Use the LC Bandpass Pi block to attenuate the source signal. Use the following default parameters for inductance and capacitance.



Note If, on the LC Bandpass Pi dialog box, you select the **Plot the network parameters of this block** check box to see the frequency response of the filter, the plot displays the actual RF behavior of the block at the selected passband and *not* the complex baseband-equivalent response.

---

7 Use the Output Port block to specify the system's load impedance and to transition from the physical portion of the model to a Simulink signal. Use the default parameters.

Note After you run the model, the **Plot the network parameters of the RF system** check box appears on the Output Port dialog box, allowing you to plot the frequency response of the physical system. If you select this check box, the plot displays the actual RF behavior of the system and *not* the behavior of the baseband-equivalent model.

---

8 Use the Spectrum Scope1 block to view the attenuated three-tone source signal. On the **Axis Properties** pane, set the following parameters:

- **Frequency range** = `[-Fs/2...Fs/2]`
- **Minimum Y-limit** = `-160`
- **Maximum Y-limit** = `-105`

9 Run the model.

The RF Blockset takes the frequency domain parameters you specified in the physical blocks and performs a two-step mathematical transformation to create a complex baseband-equivalent model in the time domain. This process is called the *complex baseband-equivalent model technique*. It is analogous to the techniques used for the blocks in the Communications Blockset Channels and RF Impairments libraries. This technique assumes that the frequencies of interest lie between f_{min} and f_{max} and are centered around a center frequency, f_c. For efficient simulation, $f_{\max} - f_{\min} \ll f_c$. It also assumes that the out-of-band energy is negligible. For more information about this technique, see “Baseband-Equivalent Modeling” on page 1-6.

The RF Blockset uses the complex baseband-equivalent model technique because it improves simulation speed. To model a system in the time domain, Simulink would need to use a small step size.

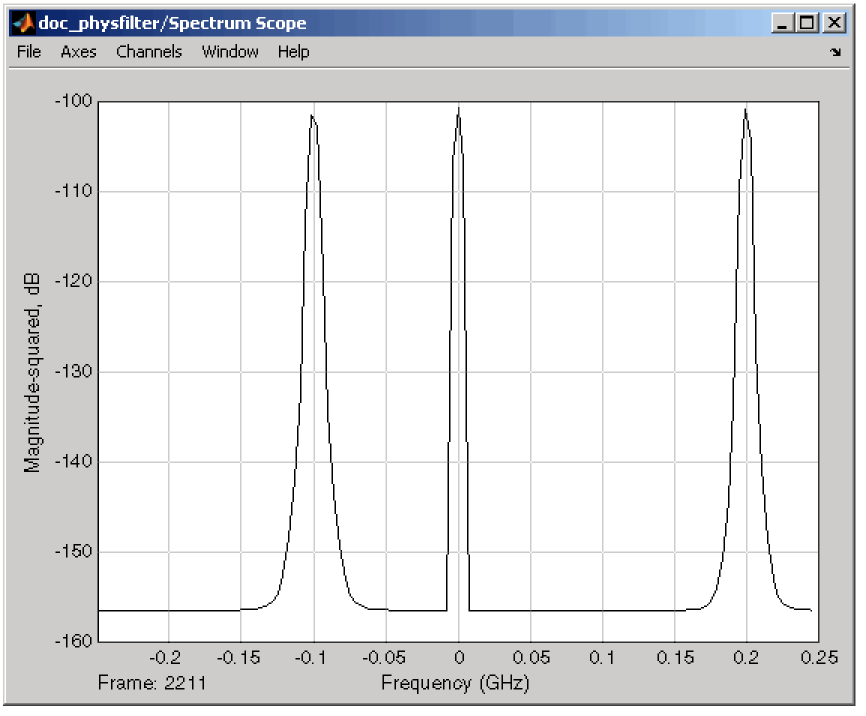
$$t_{step} = \frac{1}{2f_{\max}}$$

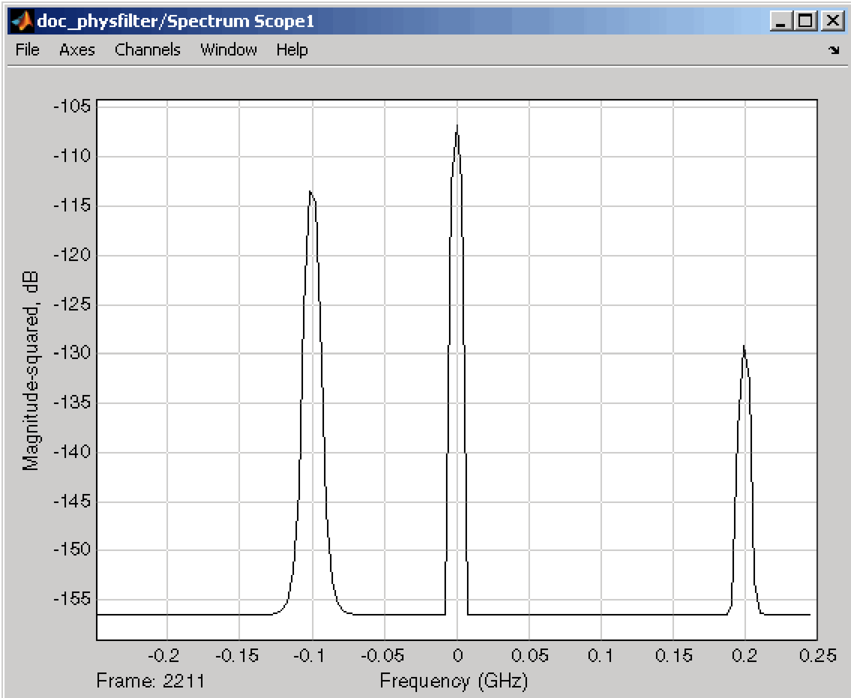
Consequently, the simulation speed would be slow. By creating a complex baseband-equivalent model in the time domain, the time step for the model can be much larger.

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

As a result, the model runs much faster.

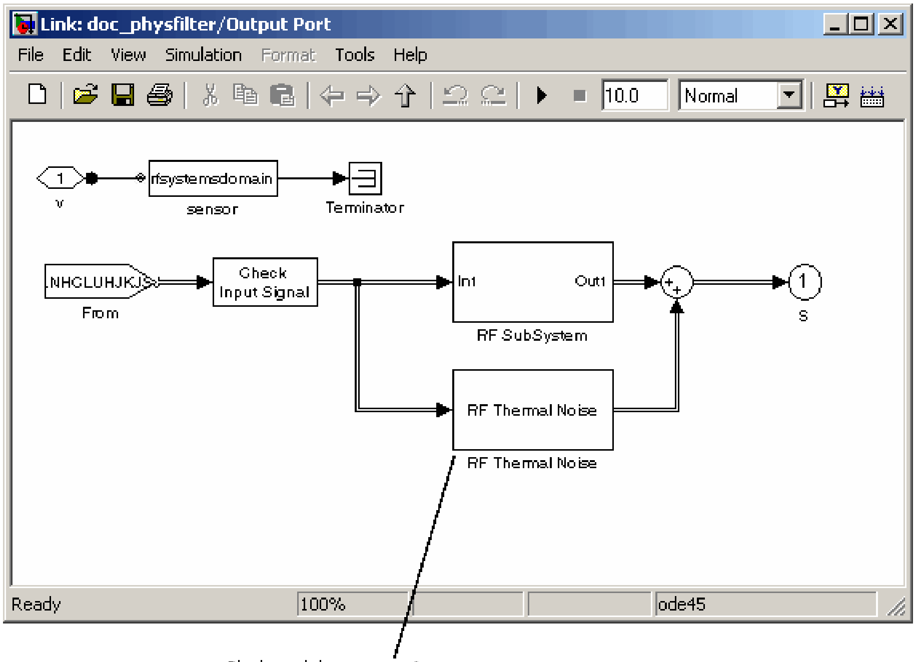
You can view the source signal and the attenuated signal in the Spectrum Scope windows while the model is running.





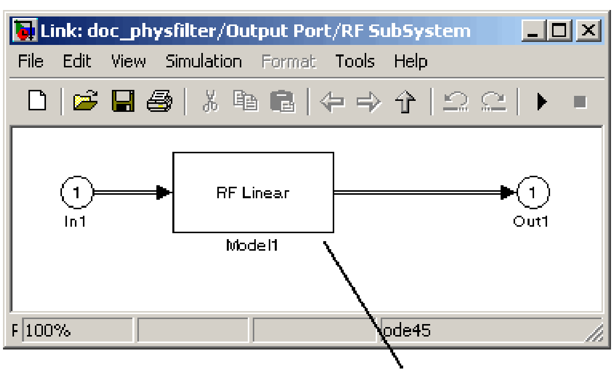
Note The magnitude of the input signal at the center frequency approaches -101 dB in the Spectrum Scope window. The magnitude of the output signal at the center frequency approaches -107 dB in the Spectrum Scope1 window. However, the filter does not attenuate the signal at the center frequency. The 6 dB of loss is caused by the source and the load in the model.

10 After you run your simulation or click the **Refresh Model blocks** button, you can view the baseband-equivalent model by right-clicking the Output Port block and selecting **Look Under Mask**. The following figure is an example of what you might see.



Block models system noise.

In the previous figure, you see a block that models thermal noise. If you right-click the RF SubSystem block and select **Look Under Mask**, you see the blocks that model the linear and nonlinear components of the RF system. In this example, the blockset only needed to model linear components.



Block models linear behavior.

For the linear components, the Output Port block uses the calculated impulse response as input to the Signal Processing Blockset Digital Filter block to determine the output.

You have now used the RF Blockset to model an LC bandpass filter. You also learned more about how the RF Blockset works.

Note Some RF Blockset mathematical blocks depend on the center frequency. To obtain accurate results, keep track of the center frequency when working with these blocks. All RF Blockset physical blocks keep track of the center frequency for you.

Noise

The RF Blockset physical blocks model thermal noise. Typically, 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 kelvin. In terms of noise factor,

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

Calculating Noise Figure

For RF Blockset physical amplifier and mixer blocks, noise can be characterized by noise figure values or the spot noise data, which is more accurate. If you specify the noise figure values directly, the RF Blockset interpolates to find the noise figure values at the simulation frequencies and uses these for its calculations. If you specify spot noise data using a Touchstone or AMP data file, the RF Blockset 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 simulation frequencies. Then it uses the following equation to calculate the noise correlation matrix, C_A.

$$C_A = 2kT\begin{bmatrix} R_n & \frac{NF_{\min} - 1}{2} - R_n Y_{opt}^{\;*} \\ \frac{NF_{\min} - 1}{2} - R_n Y_{opt} & R_n \left|Y_{opt}\right|^2 \end{bmatrix}$$

In the above equation, `k` is the Boltzmann constant and `T` is the noise temperature. Once you know the noise correlation matrix, you can use the following equation to calculate the noise factor, `F`.

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

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

In the above equations, Z_s is the nominal impedance, which is 50 ohms, and z^+ is the Hermitian conjugation of z. Finally, you can use the noise factor to calculate the noise figure, NF.

$$NF = 10\log(F)$$

For the other blocks, such as transmission lines and filters, the RF Blockset calculates the noise figure values using the network parameters. So, there are no noise parameters on the block dialog boxes.

For more information about these calculation techniques, see Hillbrand, Herbert and Peter H. Russer, "An Efficient Method for Computer Aided Noise Analysis of Linear Amplifier Networks," *IEEE Transactions on Circuits and Systems*, Vol. CAS-23, No. 4, April 1976, pp. 235-238.

Calculating Noise Power

The RF Blockset uses the noise temperature, which you specify in the block parameters dialog box, and the modeling bandwidth, which it calculates from the model's sample time and center frequency, to calculate the noise power:

$$\text{Noise power} = kTB$$

Here, `k` is the Boltzmann constant, which is 1.38e-23 Joules/kelvin, `T` is the noise temperature in kelvin and `B` is the bandwidth in hertz. It uses this noise power to determine the amplitude of the noise that it adds to the system using a Gaussian distributed pseudorandom number generator.

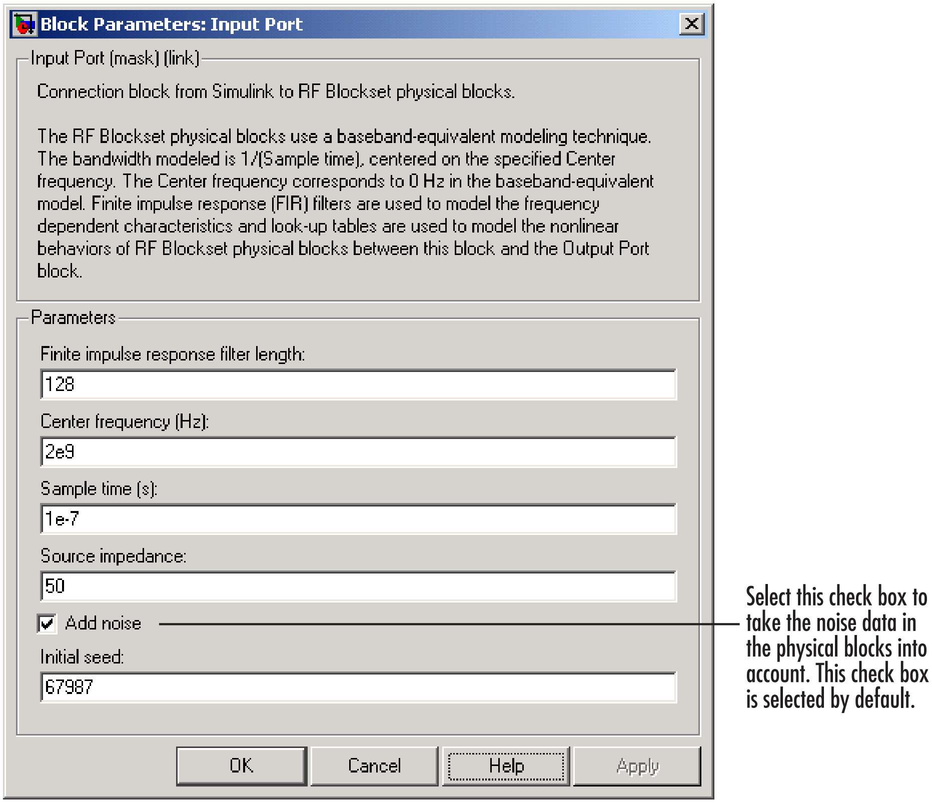
For more information, see the `rfdata.noise` and `rfdata.nf` reference pages.

Adding Noise to Your System

To add noise to your RF system:

1 Enter values for the **Noise figure (dB)** parameters in the physical blocks.

2 On the Input Port block dialog box, select the **Add noise** check box.

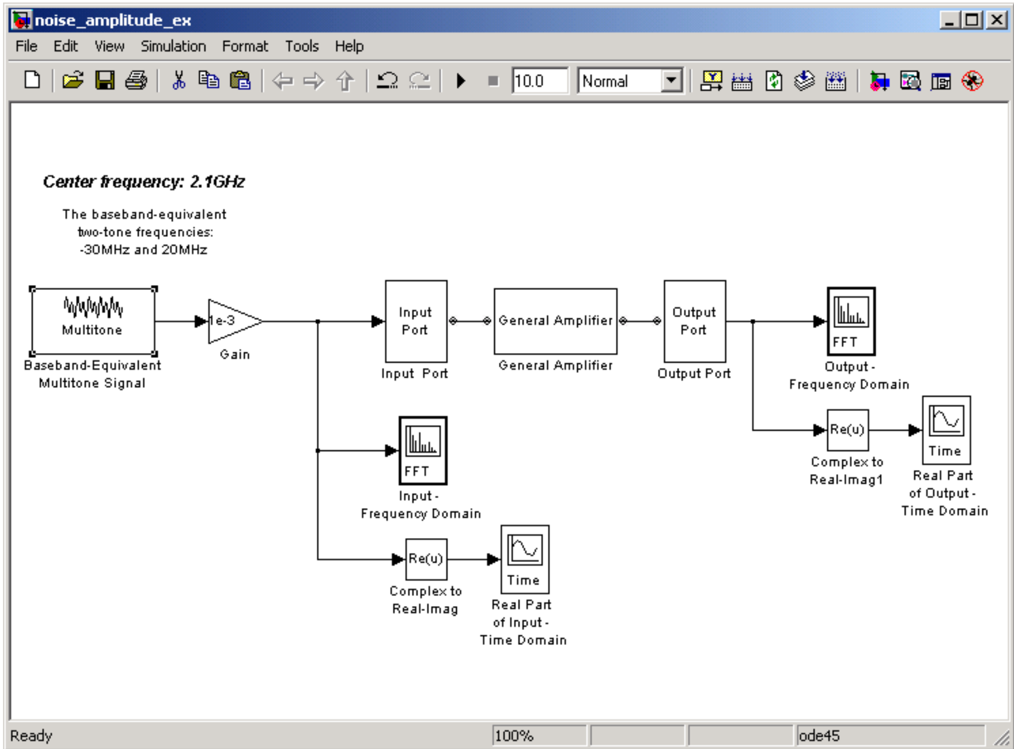


3 Run the model.

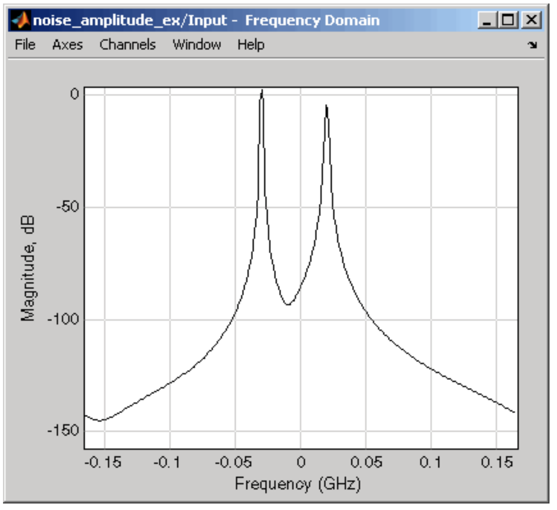
Noise Amplitude

The noise in communications 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. As a result, you won't see the noise the RF Blockset adds to your signal unless you attenuate the signal amplitude to obtain the desired signal-to-noise ratio.

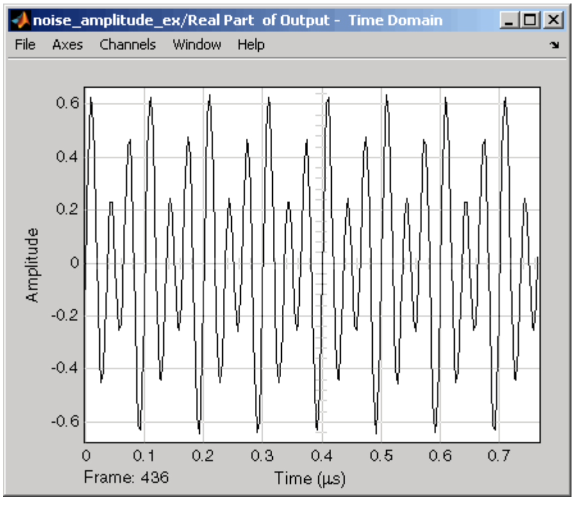
For example, suppose you have the following model that contains a multitone test signal source.



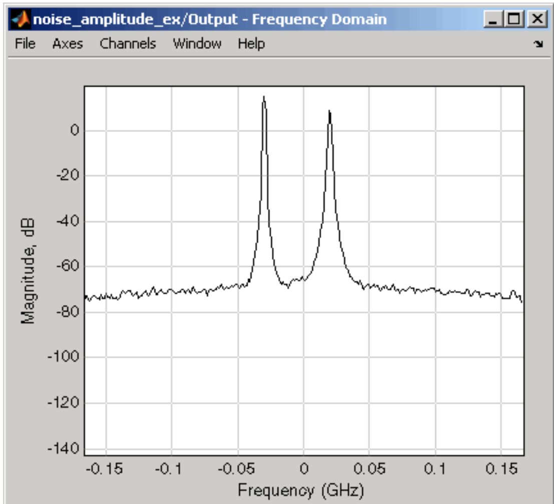
When you run this model, you can see the original signal in the frequency domain in the Input - Frequency Domain window.



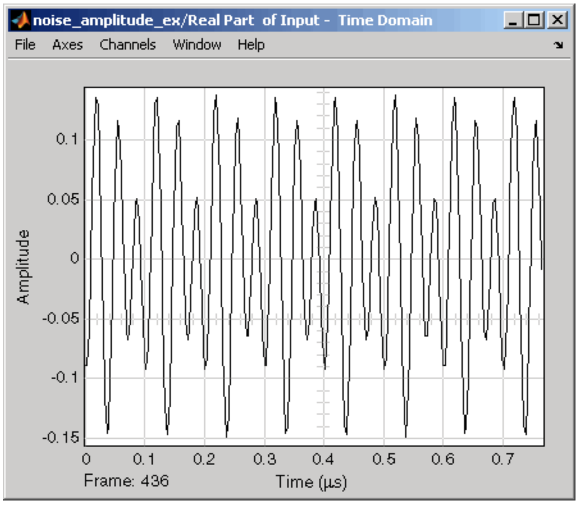
You can see the real part of the complex-valued original signal in the time domain in the Real Part of Input - Time Domain window.



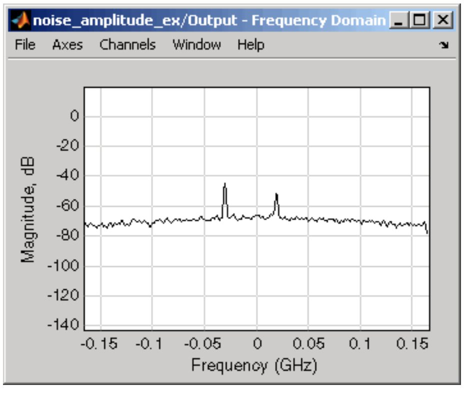
In the model, the Input Port block adds noise to the signal and the General Amplifier block amplifies it. You can see the amplified, noisy signal in the frequency domain in the Output - Frequency Domain window.



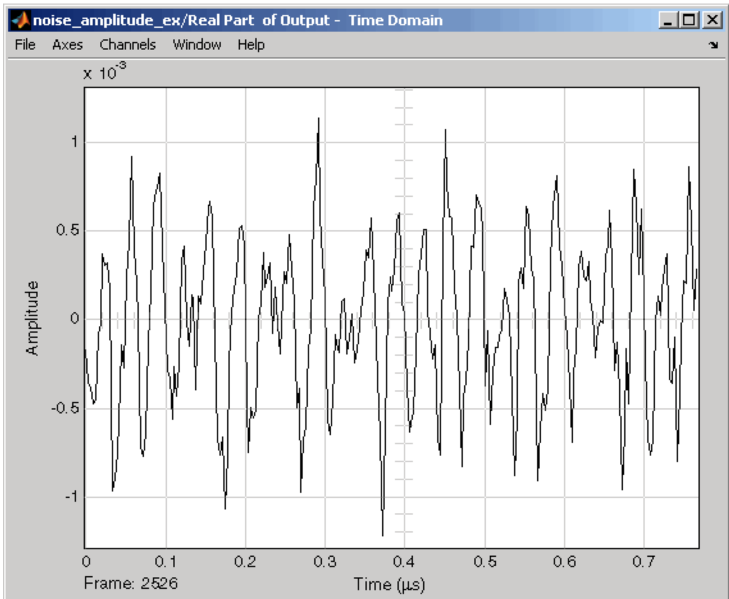
However, because the amplitude of the signal is large compared to the amplitude of the noise, you cannot see the noise in the Real Part of Output - Time Domain window.



To see the noise of the time-domain output signal, you must attenuate the amplitude of the original signal. For example, suppose you set the **Gain** parameter to `1e-3`. This is equivalent to attenuating the input signal by 60 dB. When you run the model again, you can see that the two signal peaks are not as pronounced in the Output - Frequency Domain window.

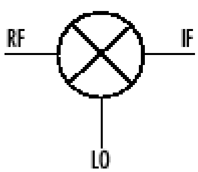


Also, you can see the noise the RF Blockset adds to your signal in the Real Part of Output - Time Domain window.

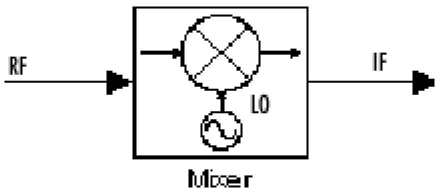


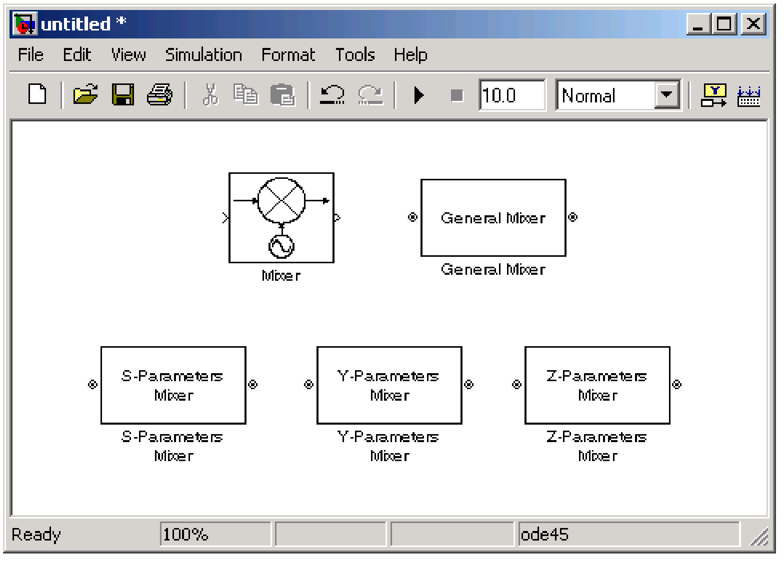
Mixers

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



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

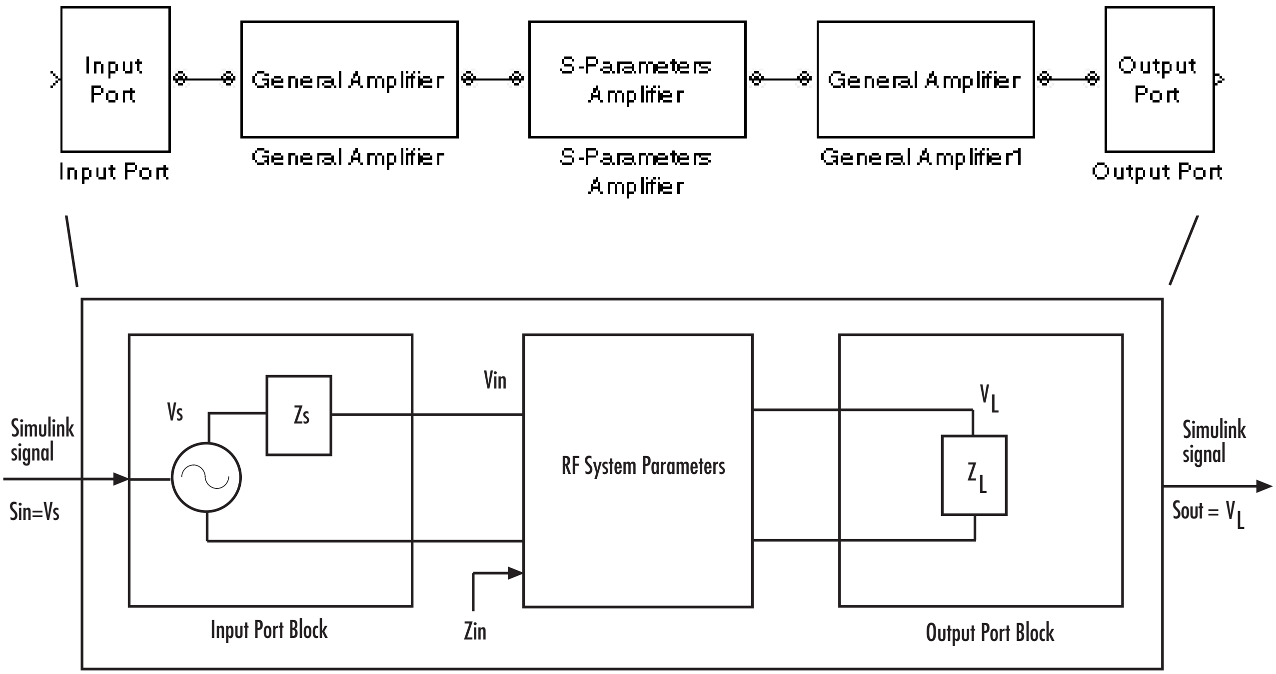




For the physical blocks, you can use the **LO frequency (Hz)** parameter to specify the local oscillator's frequency.

Interpretation of Simulink Signals

The RF Blockset interprets the Simulink signal input to the Input Port block as the voltage of the voltage source, as shown in the following figure. It interprets the output Simulink signal as the voltage over the load impedance.


Input Port
General Amplifier
S-Parameters Amplifier
General Amplifier
Output Port
Input Port
General Amplifier
S-Parameters Amplifier
General Amplifier1
Output Port
Simulink signal
Sin=Vs
Vs
Zs
Vin
Input Port Block
Zin
RF System Parameters
VL
ZL
Output Port Block
Simulink signal
Sout = VL

Demos

Demos of the RF Blockset capabilities are available on the **Demos** tab of the MATLAB Help browser. These demos show examples of linear filtering and nonlinear amplification, as well as transmitters and receivers.

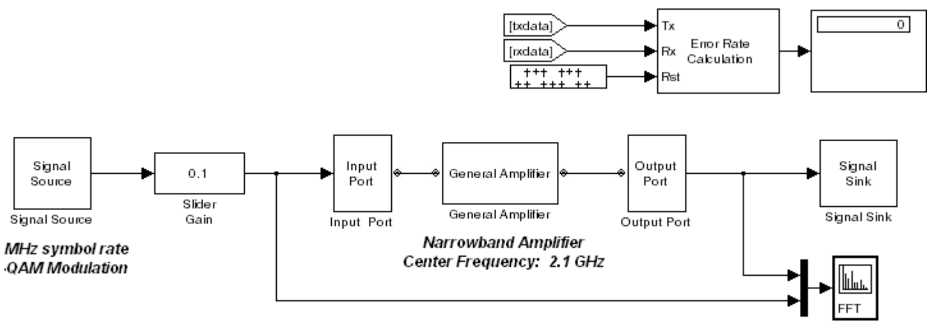
2

Creating an RF Model

RF Models (p. 2-2)	Introduces RF models and defines a requirement for connecting the different types of blocks within a model.
Creating a Model (p. 2-3)	Describes the mechanics of creating an RF model.

RF Models

An RF model consists of a block diagram that contains one or more schematics, each of which is a set of connected blocks representing an RF system. For example, the following model contains two schematics. The lower schematic displays the FFT of the input and output signals of a physical RF component, which is an amplifier.



In the diagram above, the Input Port, General Amplifier, and Output Port blocks represent a physical component network. Other blocks represent the mathematical portions of the model. The RF Blockset requires that you use the Input Port and Output Port blocks to connect the physical and mathematical portions of the model. “Creating a Model” on page 2-3 explains the use of these blocks in more detail.

Creating a Model

This section describes the mechanics of creating an RF model.

- "RF Blockset Libraries" on page 2-3
- "Connecting RF Blocks" on page 2-4
- "Setting RF Blockset Block Parameters" on page 2-6

RF Blockset Libraries

The RF Blockset consists of two main libraries, Physical and Mathematical. Physical blocks describe RF components in terms of their geometry, topologies, or network parameters. Mathematical blocks, which are Simulink style blocks, describe components in mathematical terms.

An RF model can contain blocks from both the Physical and Mathematical libraries. It can also include Simulink, Signal Processing Blockset, and Communications Blockset blocks, as well as blocks from other MathWorks blocksets.

Physical Library

The RF Blockset provides the following libraries of physical modeling blocks. Use these blocks to model the physical structure of portions of a component network. You may find it convenient to encapsulate these parts of the network into subsystems.

- Amplifiers
- Black Box Elements
- Ladder Filters
- Mixers
- Transmission Lines

The Physical Library also contains the Input/Output Connectors library. This library contains the Input Port and Output Port blocks that you use to connect the physical and mathematical portions of a model.

Mathematical Library

The RF Blockset includes the following libraries of mathematical modeling blocks. These blocks provide mathematical equivalents of RF components.

- Amplifiers
- Filters
- Mixers

For more information about these libraries, see Chapter 5, “Blocks — By Category”.

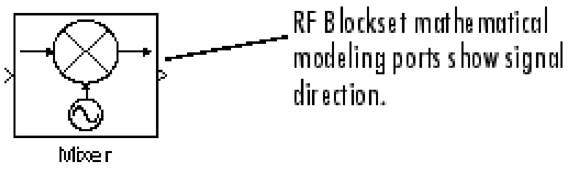
Connecting RF Blocks

To create an RF model, first open a model window and open the RF Blockset library by typing `rflib` at the Command Window prompt. Then drag instances of RF Blockset blocks into the model window and draw lines to connect the blocks.

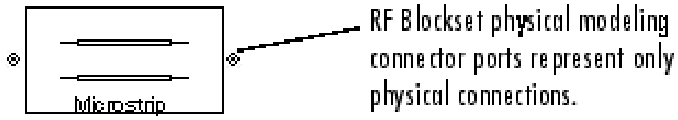
In general, you connect RF Blockset blocks in the same way you connect Simulink blocks. However, RF Blockset physical modeling blocks and mathematical blocks have different styles of ports and you cannot connect them directly.

Block Ports

Like standard Simulink blocks, the RF Blockset mathematical modeling blocks have input and output ports. The ports show the direction of the signal at the port.



In contrast, the RF Blockset physical modeling blocks have specialized *connector ports*. These ports represent only physical connections; they do not imply signal direction.



Connecting the Blocks

Just as for standard Simulink blocks, the *signal lines* that you draw between the ports of RF Blockset mathematical modeling blocks represent inputs to and outputs from the mathematical functions represented by the blocks.

In contrast, the lines you draw between RF Blockset physical modeling blocks, called *connection lines*, represent physical connections among the components represented by the blocks. You can draw connection lines only between the connector ports of physical modeling blocks. You cannot branch these connection lines. Connection lines appear as solid black when connected and as dashed red lines when either end is unconnected.

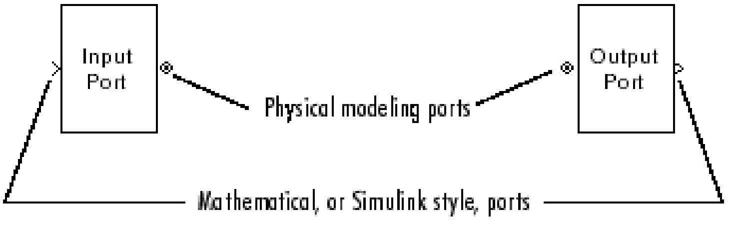
Note Physical blocks must be fully connected before you run the model. If they are not, an error message appears.

You can freely connect Simulink, Signal Processing, and RF Blockset mathematical modeling blocks. However, you cannot directly connect any of these blocks to an RF Blockset physical modeling block. The RF Blockset provides special blocks to connect the physical and mathematical parts of the model.

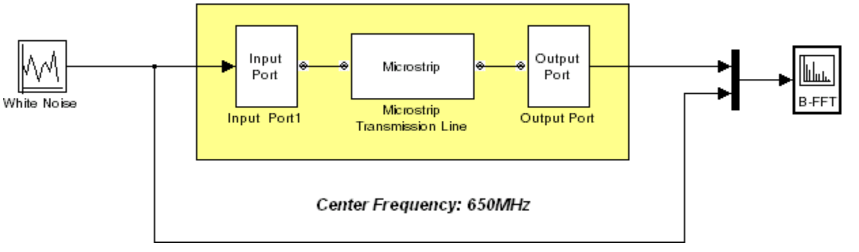
Connecting Physical and Mathematical Parts of a Model

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

The Input Port and Output Port blocks have one of each kind of connector port. The Input Port block has a standard Simulink style input port and one physical modeling port. The Output Port block has one physical modeling port and a standard Simulink style output port.



A simple RF model of a coaxial transmission line might look like 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 both Signal Processing Blockset blocks.



The Input Port block dialog box also enables you to provide the parameter data needed to calculate the baseband-equivalent impulse response for the bracketed physical subsystem. The Output Port block dialog box lets you specify the load impedance for the physical subsystem. See the Input Port and Output Port reference pages for more information about these blocks.

Setting RF Blockset Block Parameters

You can set RF Blockset block parameters using either the block dialog boxes or the Simulink `set_param` and `get_param` commands. Open the dialogs by double-clicking the block.

Working with RF Models

S-Parameters at Simulation Frequencies (p. 3-2)	Determine the S-parameters at the simulation frequencies for each block.
Using File Data (p. 3-4)	View the contents of a `.s2p` file and bring the data into an RF model.
Importing Circuits from the MATLAB Workspace (p. 3-9)	Use RF Blockset blocks to model circuits that contain series and parallel elements.

S-Parameters at Simulation Frequencies

In a physical system, each block provides network parameters at different frequencies. These are not necessarily the simulation frequencies for the physical system in which the block resides. A two-step process results in S-parameters at the simulation frequencies for each block:

1 "Determining the Simulation Frequencies" on page 3-2

2 "Interpolating the S-Parameters" on page 3-2

Determining the Simulation Frequencies

The Output Port block uses Input Port block parameters to determine the simulation frequencies f for the physical system 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 simulation frequencies are also a function of the center frequency f_c and the sample time t_s.

The nth element of f, f_n, is given by

$$f_n = f_{min} + \frac{n-1}{t_s N} \qquad n = 1, \ldots, N$$

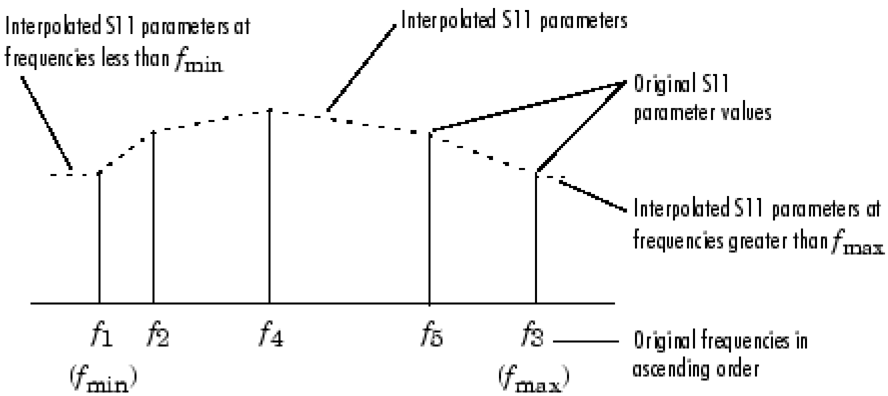
where

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

Interpolating the S-Parameters

Individual physical blocks calculate the S-parameters at the simulation frequencies determined by the Output Port block. Each block interpolates its S-parameters to determine the S-parameters at the simulation 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. It then interpolates the S-parameters using the MATLAB `interp1` function. For example, the curve in the following diagram illustrates the result of interpolating the S11 parameters at the original frequencies f_1 through f_5.



The **Interpolation** 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 diagram above, each block uses the parameter values at f_{min} for all simulation frequencies smaller than f_{min}. It uses the parameter values at f_{max} for all simulation frequencies greater than f_{max}. In both cases, the results may not be accurate.

Using File Data

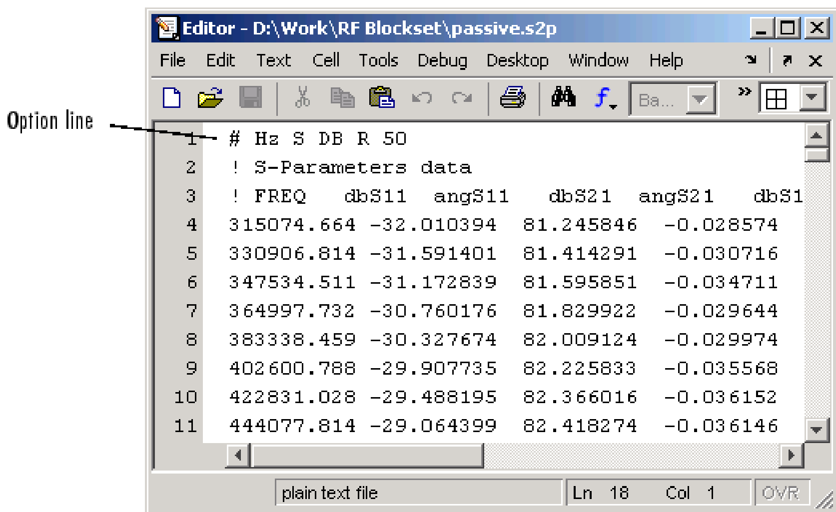
The RF Blockset can read data in Touchstone SnP, Y2P, Z2P, and H2P file formats. You can obtain this data from network analyzers, manufacturer Web sites, or other software packages. For Touchstone `.snp` files, `n` is an integer that represents the number of ports on the device to which the data corresponds. The RF Blockset only supports `.s2p` files. The scattering parameters in these files are represented by the complex variables S11, S12, S21, and S22.

In this example, you view the contents of `passive.s2p` and bring the data into an RF model using a General Passive Network block.

1 Open the `passive.s2p` file. At the MATLAB command prompt, type

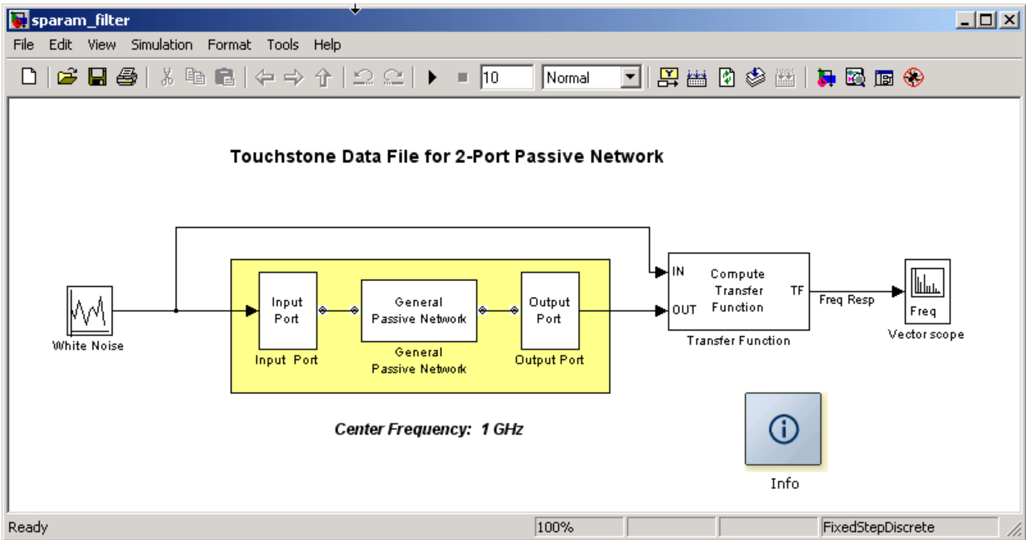
```
edit passive.s2p
```

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



Note that the option line reveals that the frequency is in Hz, the file contains S-parameters, the parameters are given in terms of dB magnitude and phase, and the reference impedance is 50 ohms. For more information about the option line, see `http://www.eda.org/pub/ibis/connector/touchstone_spec11.pdf`.

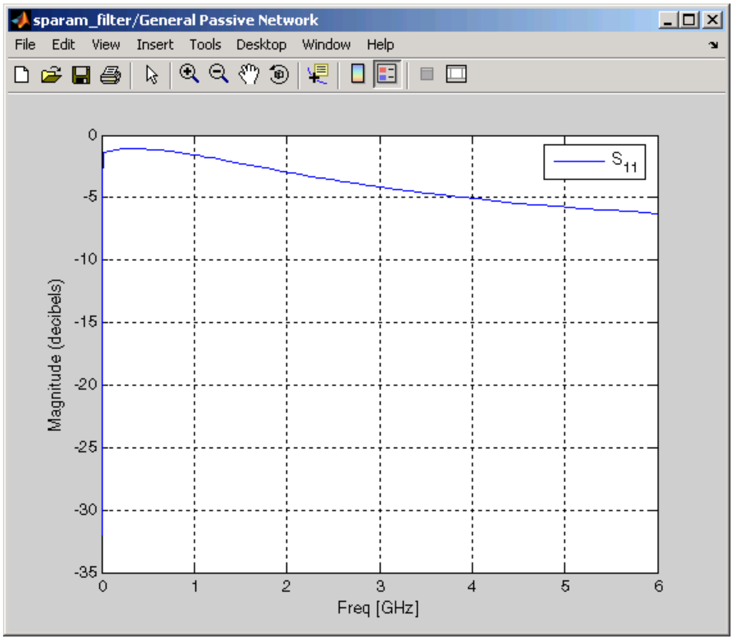
2 Open the RF Blockset demo called Touchstone Data File for 2-Port Passive Networks by typing `sparam_filter` at the MATLAB command prompt.



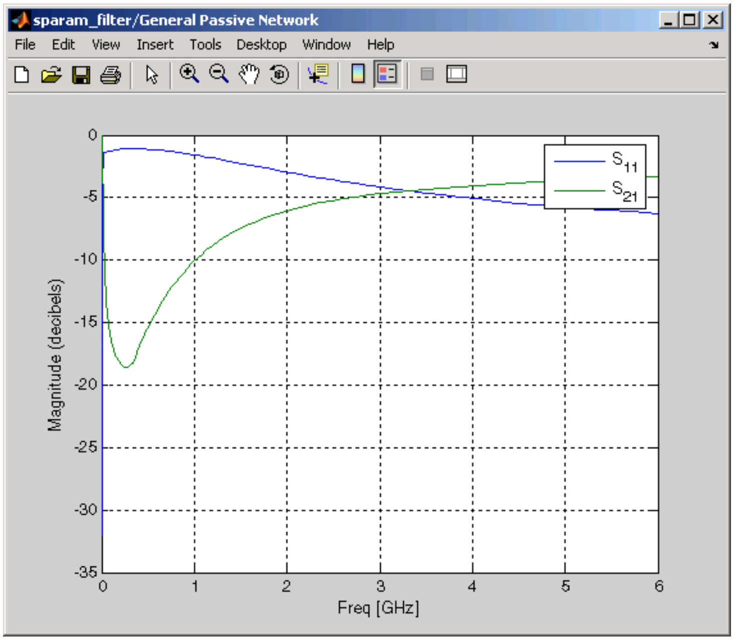
3 Use the General Passive Network block to import the data into the RF model. Double-click the General Passive Network block. The **RFDATA object** parameter is set to `read(rfdata.data, 'passive.s2p')`. This creates an `rfdata.data` object and reads the data from `passive.s2p` into it. The block uses this data as part of its parameters.

4 Plot the data you imported into the RF model. Select the **Plot the network parameters of this block** check box. From the **Parameter** list, select `S11`. Click **Apply**.

The plot of the S11 parameter appears in the General Passive Network window. Notice that the data is plotted at the RF frequencies and not at the baseband-equivalent frequencies.

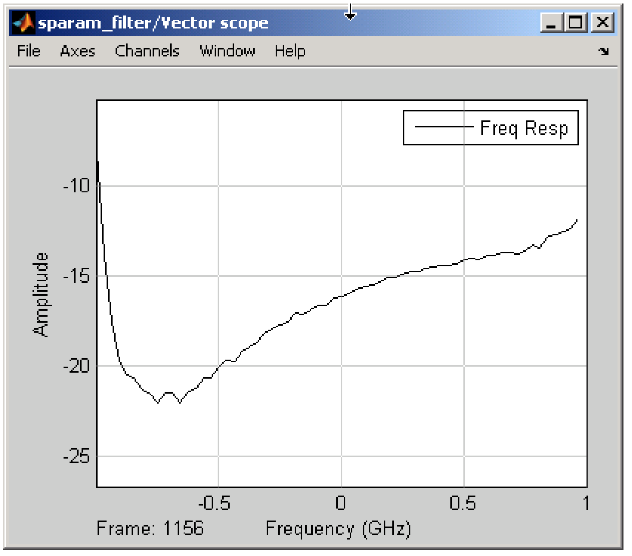


From the **Parameter** list, select `S21`. Click **Apply**. The block adds the S21 data to the plot. It continues to add to the plot until you close the plot window.



5 Run the demo model.

The model plots the amplitude of the transfer function of the system in decibels as a function of frequency in the Vector scope window. The transfer function is approximately the same as S21. The differences arise from the 6 dB of loss introduced by the source and the load in the `sparam_filter` model, and from the calculation of the transfer function.

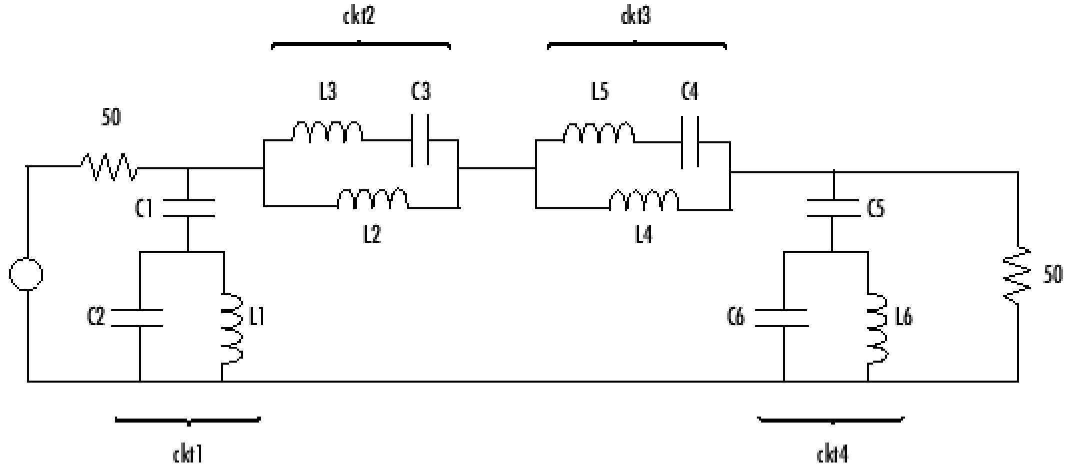


You have now brought the data from `passive.s2p` into an RF model using a General Passive Network block.

The `.s2p` file may contain data that is measured at frequencies that do not correspond to the simulation frequencies. In this case, you can use the **Interpolation method** parameter to determine how the block determines the data values at the simulation frequencies. For more information, see "Determining the Simulation Frequencies" on page 3-2 and "Interpolating the S-Parameters" on page 3-2.

Importing Circuits from the MATLAB Workspace

The physical blocks in the RF Blockset can only be connected in cascade. However, you can still use RF Blockset blocks to model circuits that contain series and parallel elements. In this example, you model the 50-ohm bandstop filter shown below by creating circuit objects in the MATLAB workspace. You import these circuit objects into a Simulink model using the RF Blockset General Circuit Element block.



1 Define the filter's resistance, capacitance, and inductance values in the MATLAB workspace. At the MATLAB command prompt, type

```
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 Use RF Toolbox functions to create RF circuit objects that model the components labeled ckt1, ckt2, ckt3, and ckt4 in the circuit diagram. At the MATLAB command prompt, type

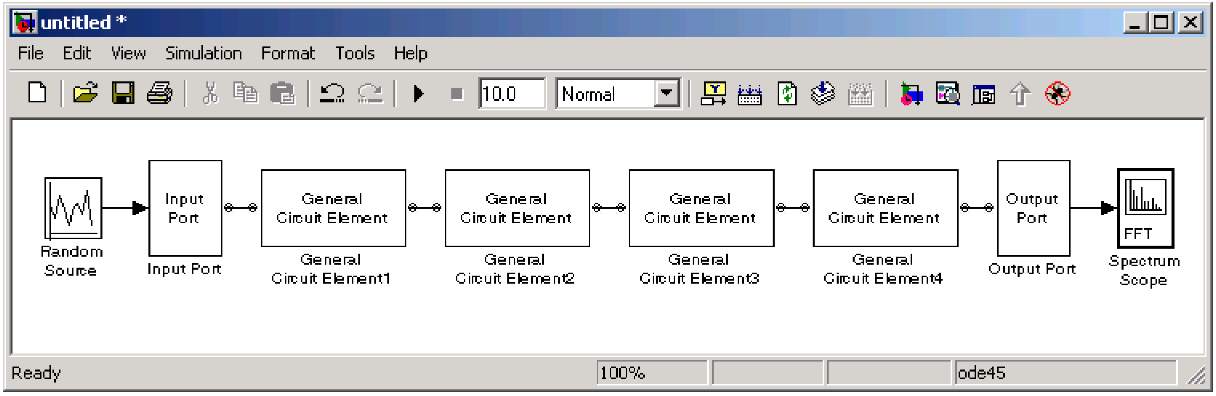
```
ckt1 =
rfckt.series('Ckts',{rfckt.shuntrlc('C',C1),rfckt.shuntrlc('L'
,L1,'C',C2)});
ckt2 =
rfckt.parallel('Ckts',{rfckt.seriesrlc('L',L2),rfckt.seriesrlc
('L',L3,'C',C3)});
ckt3 =
rfckt.parallel('Ckts',{rfckt.seriesrlc('L',L4),rfckt.seriesrlc
('L',L5,'C',C4)});
ckt4 =
rfckt.series('Ckts',{rfckt.shuntrlc('C',C5),rfckt.shuntrlc('L'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

3 Create a new Simulink model.

4 Click-and-drag the following blocks into your model.

Block	Library	Quantity
Random Source	Signal Processing Blockset / Signal Processing Sources	1
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

5 Connect the blocks so your model is similar to the figure below. You need to change the names of the General Circuit Element blocks for them to appear as they do in the figure.



6 Use the Random Source block to create random, complex input values that have a Gaussian distribution. Set the parameters as follows:

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

7 Use the Input Port block to define some of the physical characteristics of the filter. Set the 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.

8 Use the General Circuit Element blocks to import the circuit objects that model the filter components into the Simulink model.

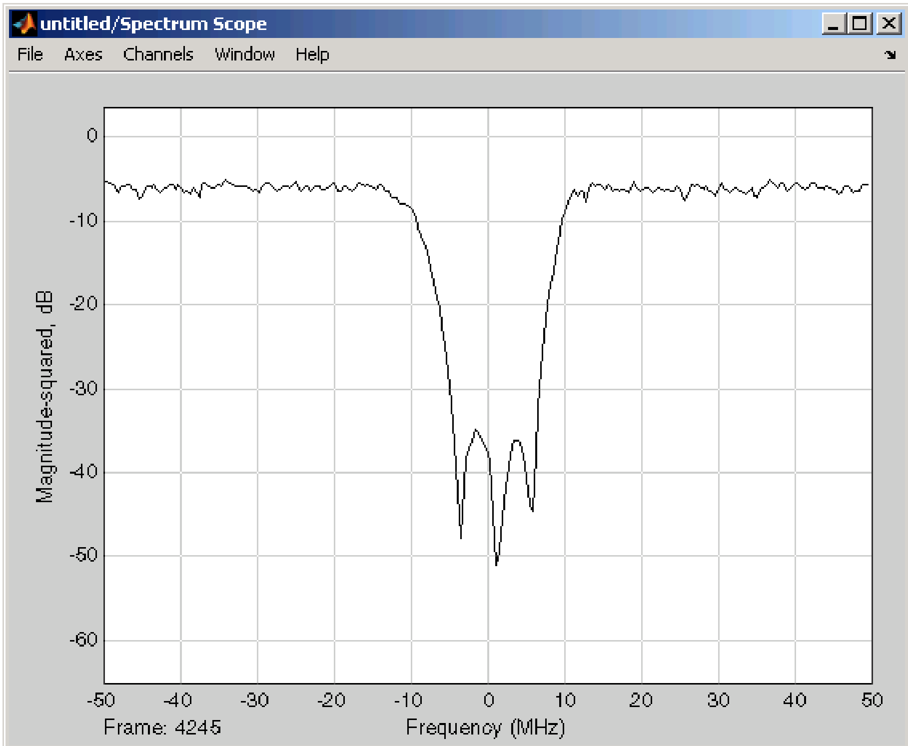
- In the General Circuit Element1 block dialog box, for the **RFCKT object** parameter, enter `ckt1`.
- In the General Circuit Element2 block dialog box, for the **RFCKT object** parameter, enter `ckt2`.
- In the General Circuit Element3 block dialog box, for the **RFCKT object** parameter, enter `ckt3`.
- In the General Circuit Element4 block dialog box, for the **RFCKT object** parameter, enter `ckt4`.

9 Use the Output Port block to define the load impedance. Set this parameter to `50`.

10 Use the Spectrum Scope block to view the bandstop filter response. Set the parameters as follows:

- On the **Scope Properties** pane, set the **Number of spectral averages** parameter to `100`.
- On the **Axis Properties** pane, set the **Frequency range** parameter to `[-Fs/2 ...  Fs/2]`, the **Minimum Y-limit** parameter to `-65`, and the **Maximum Y-limit** parameter to `3.6`.

11 Run the model and view the filter response in the Spectrum Scope window.



In this example, you modeled a 50-ohm bandstop filter by creating circuit objects in the MATLAB workspace and importing them into a Simulink model using the General Circuit Element block. For more information about the RF Toolbox functions used in this example, see the `rfckt.series`, `rfckt.parallel`, `rfckt.shuntrlc`, and `rfckt.seriesrlc` function reference pages in the RF Toolbox documentation.

References

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

Plotting Network Parameters

Generating a Plot (p. 4-2)	Describes how to generate plots and describes the available plots.
Modifying a Plot (p. 4-10)	Describes how to modify existing plots.

Generating a Plot

Blocks in the RF Physical library that model physical components provide a plotting capability that enables you to plot the network parameters of those components. The Output Port block provides a plotting capability only after you have run the model of which it is a part.

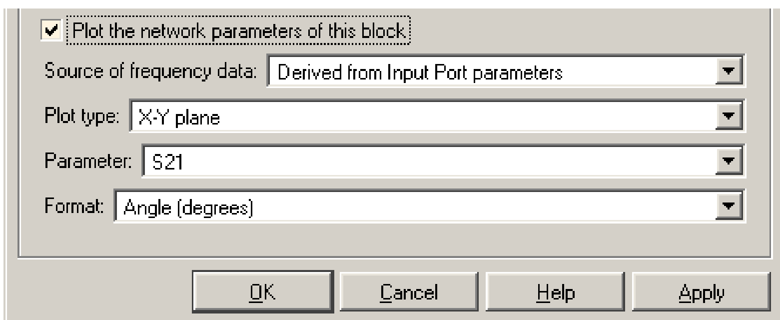
You can request a plot or change a plot whenever the model is not running. While the model is running, any existing plots continue to be displayed, but they are not updated.

This section contains these topics:

- "Block Parameters" on page 4-2
- "Example of a Z Smith Chart" on page 4-5
- "Examples of Other Plot Types" on page 4-7

Block Parameters

Blocks that enable you to request a plot provide plotting parameters in their block dialog boxes. Choose the plot parameters to define the plot you want, and then click **Apply**.



Plot the network parameters of this block
Select this check box to produce a plot. The plot is generated when you click **Apply** or **OK**.

Source of frequency data

Source of the frequencies at which the data is plotted. Choices vary for different blocks. All blocks can derive the frequencies from the Input Port block parameters, or can accept a user-specified vector of frequencies. Some blocks can use the specified mask **Frequency** parameter, or extract the frequencies from the object specified in the **RFDATA object** mask parameter. The Output Port block can use the derived simulation frequencies, once you have run the model.

Frequency range (Hz)

Specify, as a vector, the range of frequencies you want to plot. The frequencies must be strictly positive. This field becomes visible if you select `User-specified` as the source of frequency data.

Plot type

Type of plot. See the table below.

Parameter

Network parameter to be plotted. The available choices vary with the type of plot. See the table below.

Format

Format in which to plot the selected network parameter. The available choices vary based on the selected parameter. See the table below.

Plot Type	Parameter	Format
`Composite data`	Generates four separate plots in one figure. The predefined combination of plots differs based on the type of block.	

Plot Type	Parameter	Format
X-Y plane	S11, S12, S21, S22	Magnitude (decibels) Magnitude (linear) Angle (degrees) Real Imaginary
	VSWRIn, VSWROut	Magnitude (decibels) None
	OIP3 (Output Port block only)	dBm W mW
	NF (Output Port block only)	Magnitude (decibels) None
	Pout (General Amplifier block with power data only)	dBm dBW W mW
	Phase (General Amplifier block with power data only)	Angle (degrees) Angle (radians)
	AM/AM (General Amplifier block with power data only)	dB None
	AM/PM (General Amplifier block with power data only)	Angle (degrees) Angle (radians)
	PhaseNoise (Mixer blocks only)	dBc/Hz

Plot Type	Parameter	Format
Link budget (Output Port block only, for a cascaded network)	S11, S12, S21, S22	Magnitude (decibels) Magnitude (linear) Angle (degrees) Real Imaginary
	VSWRIn, VSWROut	Magnitude (decibels) None
	OIP3	dBm W mW
	NF	Magnitude (decibels) None
Polar plane	S11, S12, S21, S22	None
Z Smith chart	S11, S22	None
Y Smith chart	S11, S22	None
ZY Smith chart	S11, S22	None

Note The `Link budget` plot is available only for physical systems with more than one component between the Input Port block and the Output Port block. For systems with multiple components, this plot displays the contribution of each of the components to the metric being plotted.

Note To display the legend(s) for a plot, type `legend show` at the command line.

Example of a Z Smith Chart

In this example, you plot the network parameters of a General Amplifier block on a Z Smith Chart.

The General Amplifier block uses the RF Toolbox `read` function to create a data (`rfdata.data`) object that describes the nonlinear amplifier in the file `default.amp`. It then uses linear interpolation to model the network described in the data object.

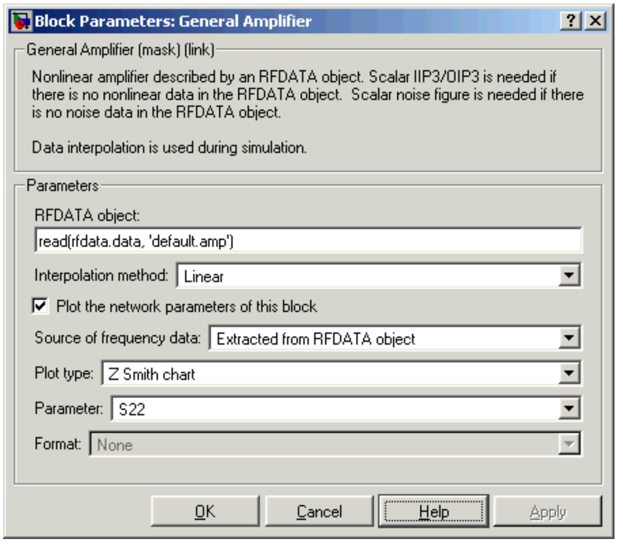
The file contains S-parameters for frequencies from 1.0 to 2.9 GHz at intervals of 0.01 GHz. To view the file, type

```
edit default.amp
```

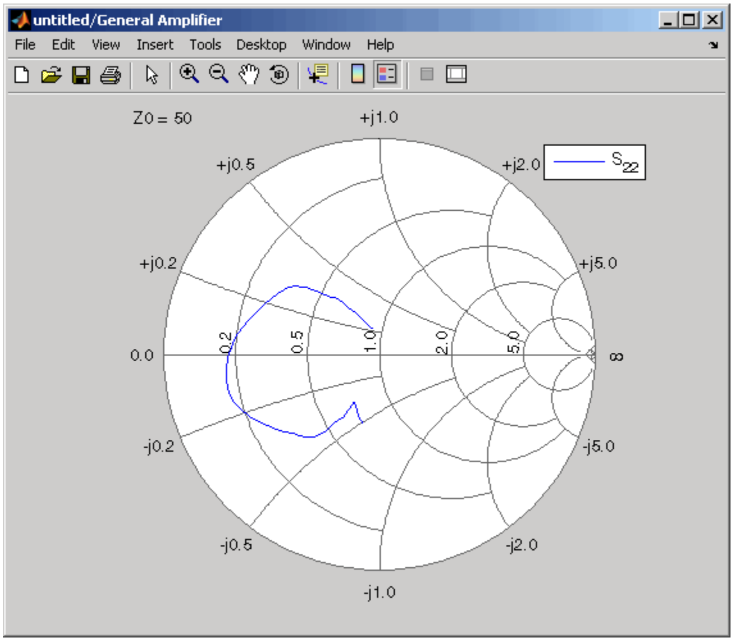
at the MATLAB command prompt.

Note The General Amplifier block models the nonlinear amplifier described by an `rfdata.data` object. See the RF Toolbox `rfdata` reference page for information about `rfdata` objects.

See "AMP File Format" in the RF Toolbox documentation for information about `.amp` files.



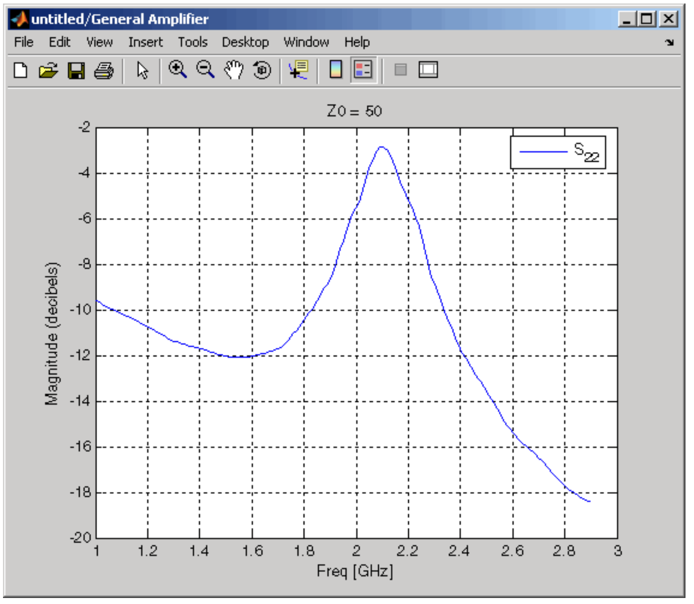
The plot parameters in the dialog request a Z Smith chart of the S22 parameters using frequency data extracted from the `rfdata` object.



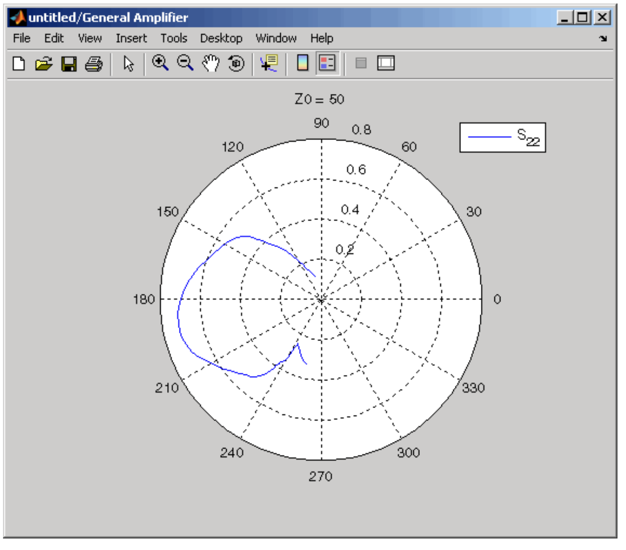
Note To display data tips for a plotted line, check **Data Cursor** on the **Tools** menu, and then click the data cursor on the plotted line to see the frequency and the plotted parameter at that point. See Data Cursor — Displaying Data Values Interactively in the MATLAB documentation for more information.

Examples of Other Plot Types

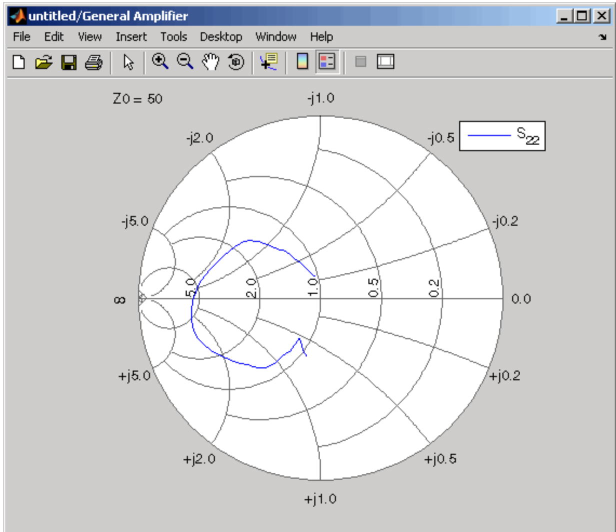
The following figures show an instance of each of the other plot types for the same data, at the same frequency range.



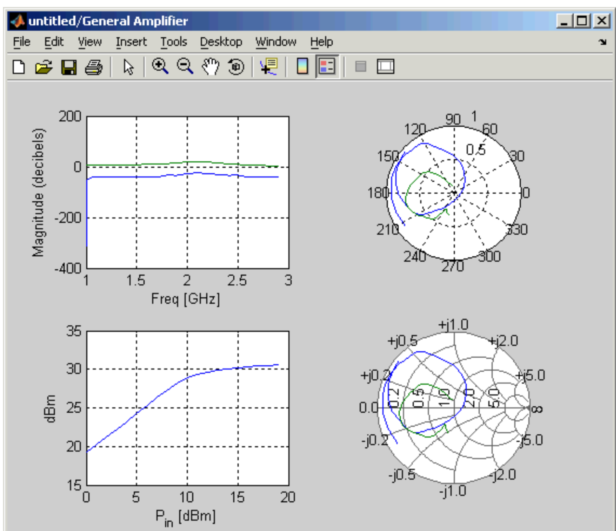
X-Y Plane Plot



Polar-Plane Plot



Y Smith Chart



Composite Data Chart

The actual combination of plots differs according to the type of block.

Modifying a Plot

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

Block Parameter	Result of Change
Frequency range	The plot is redrawn using the new frequency range.
Plot type	The figure is redrawn in the new plot type. Other plot parameters retain their values if those values are valid for the new plot type. Otherwise, they revert to their default values.
Parameter	The new parameter is added to the existing plot.
Format	The plot is redrawn using the new format.

Example of Plot Modification

In this example, you add parameters to an existing plot and modify the plot type.

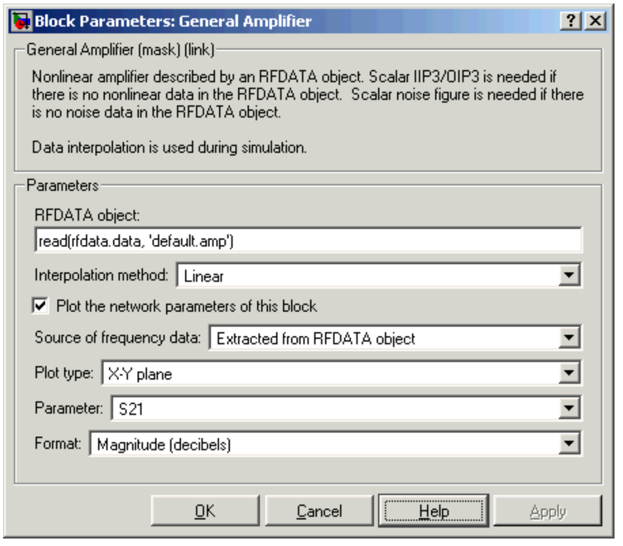
Start by generating an X-Y plane plot of the S21 parameters for the nonlinear amplifier described in the file `default.amp`. To view the file, type

```
edit default.amp
```

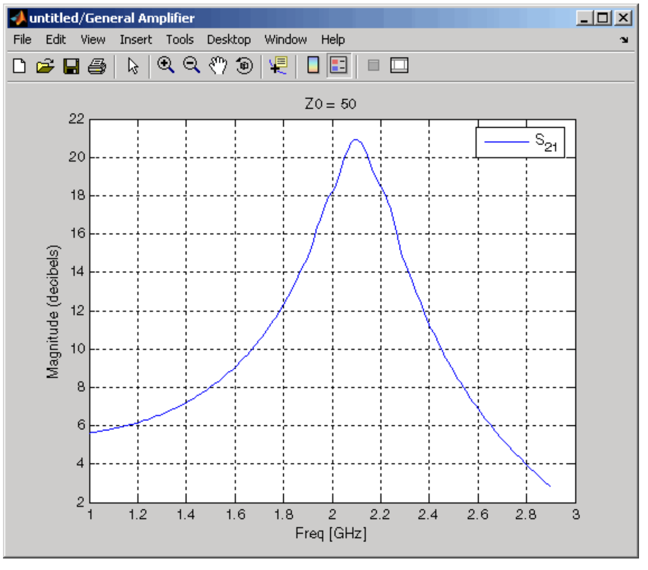
at the MATLAB command prompt.

Note The General Amplifier block creates an `rfdata.data` object to model the nonlinear amplifier described in the specified file. See the RF Toolbox `rfdata` reference page for information about `rfdata` objects.

See "AMP File Format" in the RF Toolbox documentation for information about `.amp` files.

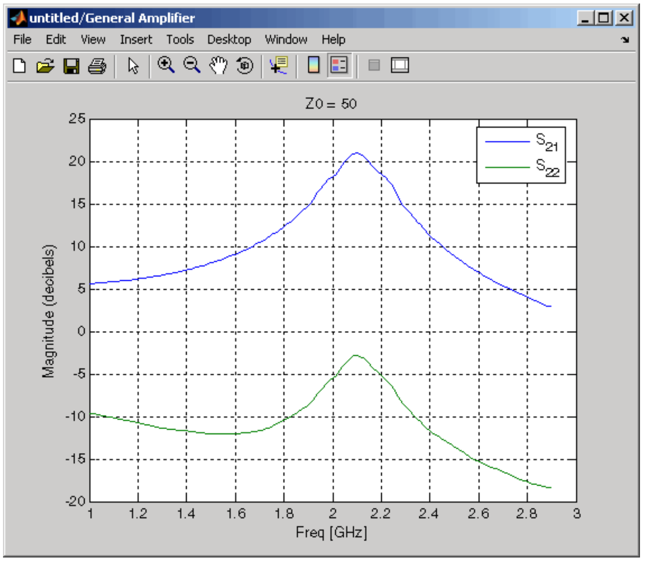


The dialog box above produces the plot below.

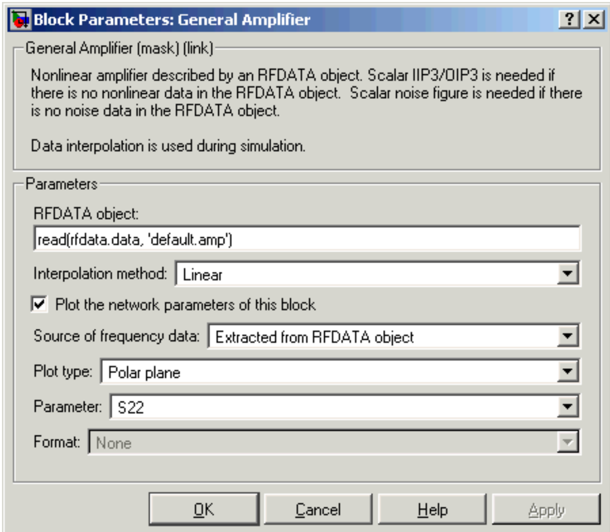


Add the S22 parameter to the plot by changing the value of **Parameter** to `S22`.

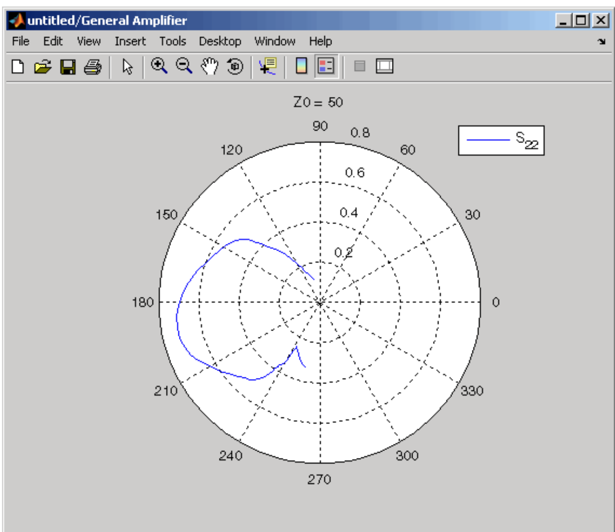
Note To display the legend for a plot, type `legend show` at the command line.



Change the **Plot type** to `Polar plane`. Note that the value of **Parameter** remains as `S22`, the last parameter selected for the previous plot.



The block dialog box above produces the plot below.



Blocks — By Category

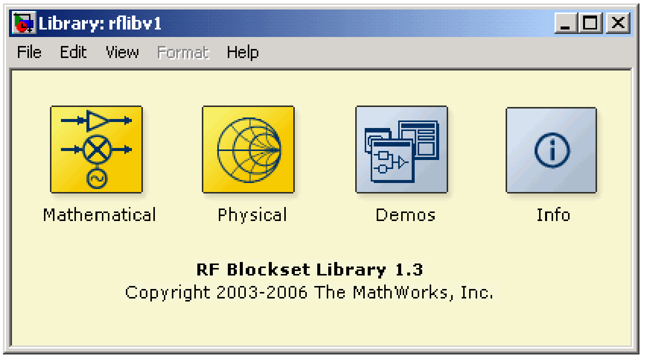
Main RF Blockset Library (p. 5-2)	Enables you to access all the blocks in the RF Blockset.
Mathematical (p. 5-3)	Contains Simulink style blocks that describe components in mathematical terms.
Physical (p. 5-5)	Contains blocks that model physical components in terms of their geometry and physical characteristics.

Main RF Blockset Library

You can open the main library by typing

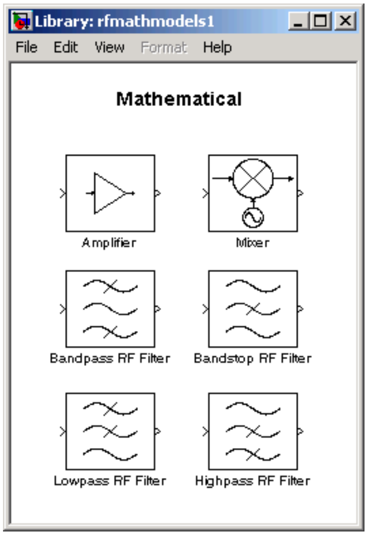
```
rflib
```

at the MATLAB prompt. Each yellow icon in the window represents a library. Double-click an icon to open the library.



Mathematical

You can open the Mathematical library by double-clicking its icon in the main RF Blockset library.



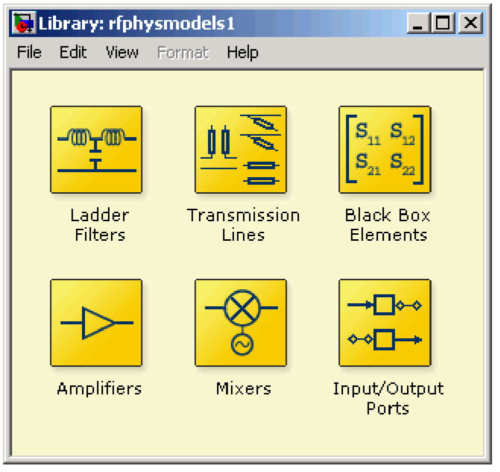
The following table lists the blocks in the Mathematical library. For information about a specific block, see its reference page.

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 with phase noise

Physical

You can open the Physical library by double-clicking its icon in the main RF Blockset library.

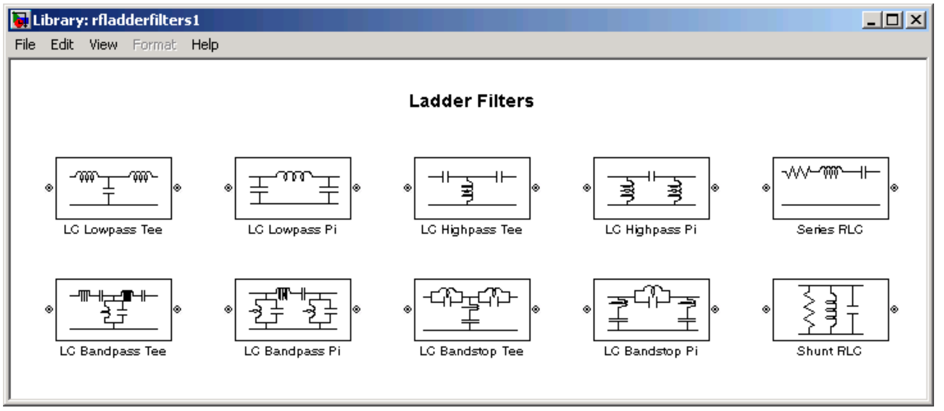


The Physical library contains these sublibraries:

- "Ladder Filters" on page 5-5
- "Transmission Lines" on page 5-6
- "Black Box Elements" on page 5-7
- "Amplifiers" on page 5-8
- "Mixers" on page 5-9
- "Input/Output Ports" on page 5-10

Ladder Filters

The Ladder Filters sublibrary contains RF filters whose network parameters can be calculated from their topologies. You can open the Ladder Filters sublibrary by double-clicking its icon in the Physical library.

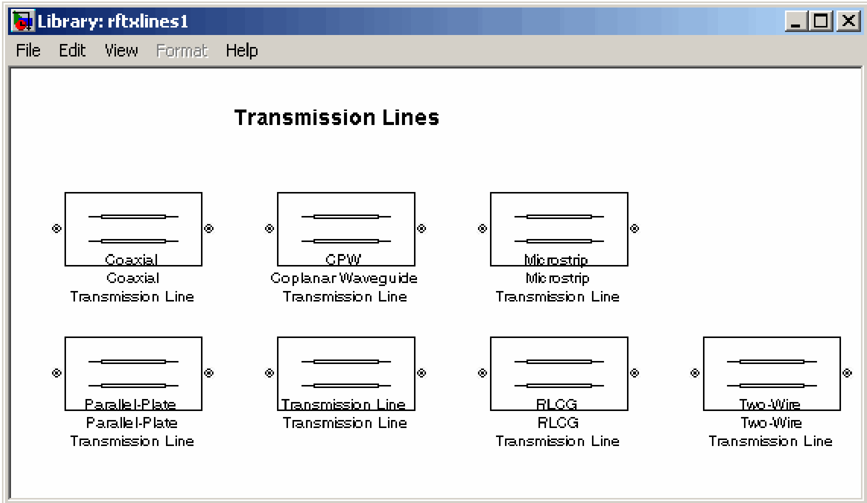


The following table lists the blocks in the Ladder Filters sublibrary. For information about a specific block, see its reference page.

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

The Transmission Lines sublibrary contains RF filters whose network parameters can be calculated from their geometry. You can open the Transmission Lines sublibrary by double-clicking its icon in the Physical library.

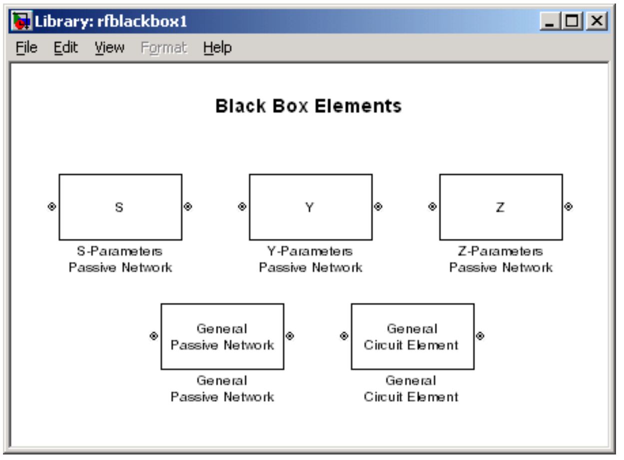


The following table lists the blocks in the Transmission Lines sublibrary. For information about a specific block, see its reference page.

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

Blocks in the Black Box Elements sublibrary describe passive RF components by their S-, Y-, or Z-parameters, or a data file containing these parameters. You can open the Black Box Elements sublibrary by double-clicking its icon in the Physical library.

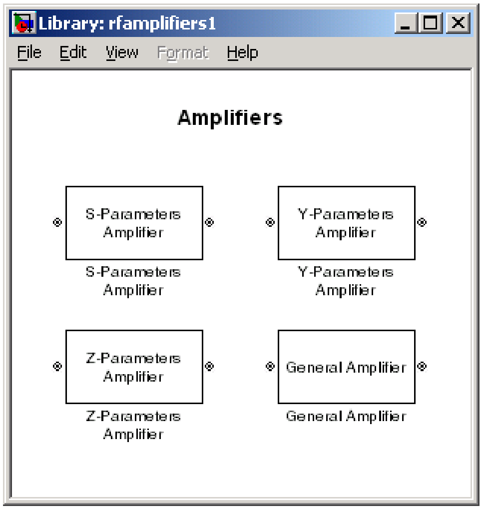


The following table lists the blocks in the Black Box Elements sublibrary. For information about a specific block, see its reference page.

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 its S-parameters
Y-Parameters Passive Network	Model passive network using its Y-parameters
Z-Parameters Passive Network	Model passive network using its Z-parameters

Amplifiers

Blocks in the Amplifiers sublibrary describe RF amplifiers by S-, Y-, or Z-parameters, or an object containing these parameters. You can open the Amplifiers sublibrary by double-clicking its icon in the Physical library.

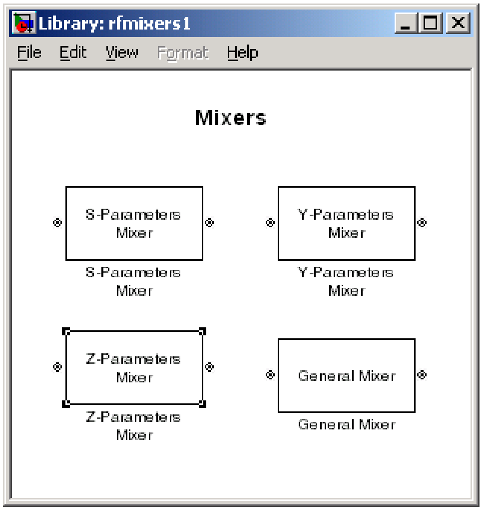


The following table lists the blocks in the Amplifiers sublibrary. For information about a specific block, see its reference page.

General Amplifier	Model nonlinear amplifier described by `rfdata` object
S-Parameters Amplifier	Model nonlinear amplifier using its S-parameters
Y-Parameters Amplifier	Model nonlinear amplifier using its Y-parameters
Z-Parameters Amplifier	Model nonlinear amplifier using its Z-parameters

Mixers

Blocks in the Mixers sublibrary describe RF mixers by S-, Y-, or Z-parameters, or an object containing these parameters. You can open the Mixers sublibrary by double-clicking its icon in the Physical library.

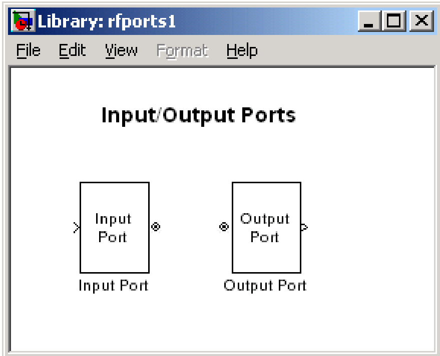


The following table lists the blocks in the Mixers sublibrary. For information about a specific block, see its reference page.

General Mixer	Model mixer described by `rfdata` object
S-Parameters Mixer	Model mixer using its S-parameters
Y-Parameters Mixer	Model mixer using its Y-parameters
Z-Parameters Mixer	Model mixer using its Z-parameters

Input/Output Ports

Blocks in the Input/Output Ports sublibrary enable you to transition from the mathematical environment into the physical modeling environment and from the physical modeling environment back into the mathematical environment. You can open the Input/Output Ports sublibrary by double-clicking its icon in the Physical library.



The following table lists the blocks in the Input/Output Ports sublibrary. For information about a specific block, see its reference page.

Input Port	Connection block from Simulink environment to RF physical blocks
Output Port	Connection block from RF physical blocks to Simulink environment

---

Note For information about using these block, see “Connecting RF Blocks” on page 2-4.

6

Blocks — Alphabetical List

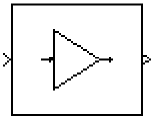
Purpose

Complex baseband model of amplifier with noise

Library

Mathematical

Description



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

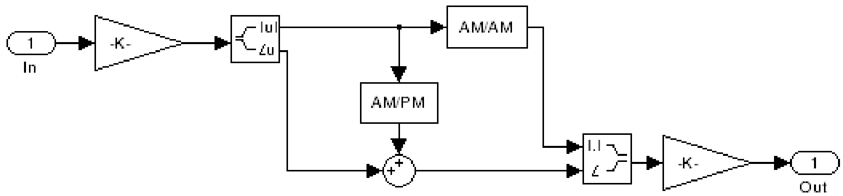
Note This block assumes a nominal impedance of 1 ohm.

Modeling Nonlinearity

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 figure below.



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.

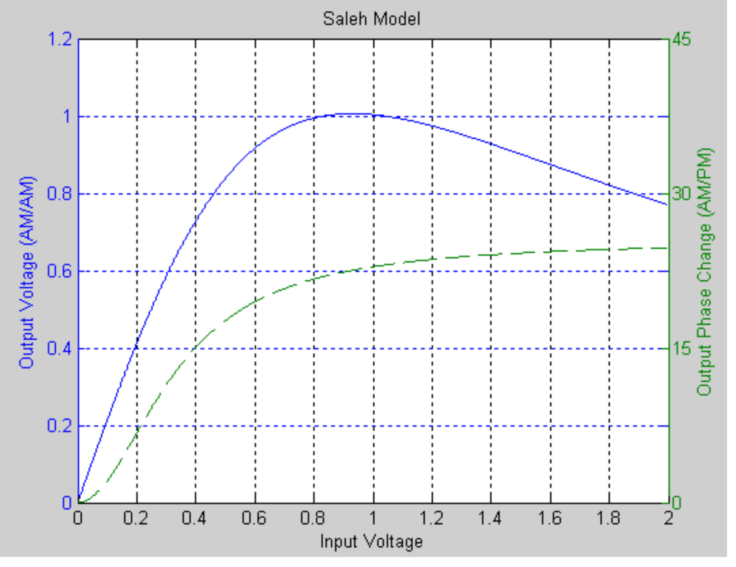
2 Select **Look under mask** in the pop-up menu. This displays the block's configuration underneath the mask. The block contains five subsystems corresponding to the five nonlinearity methods.

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.

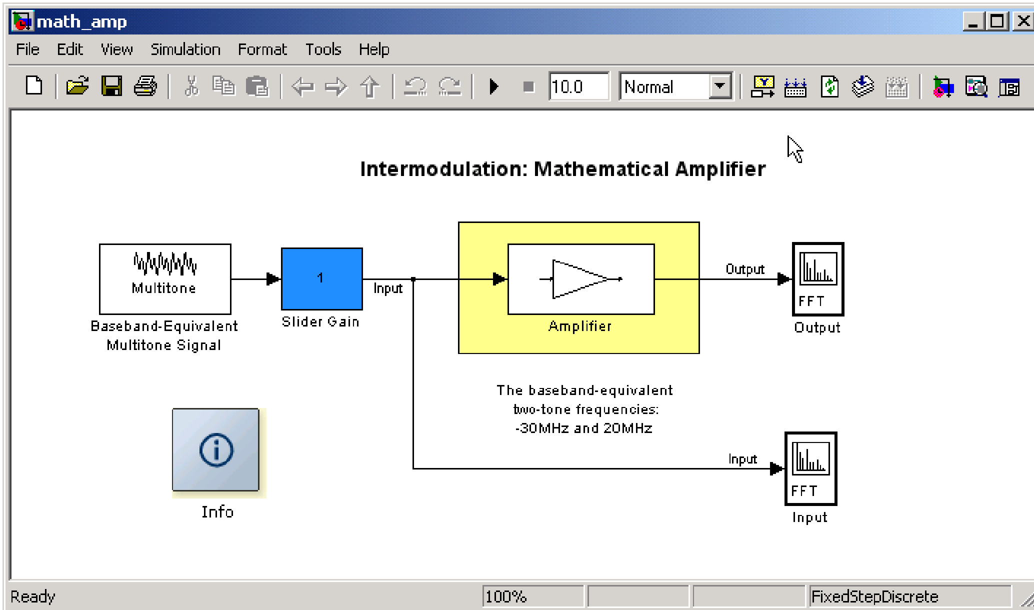
The following figure shows, for the Saleh method, plots of

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

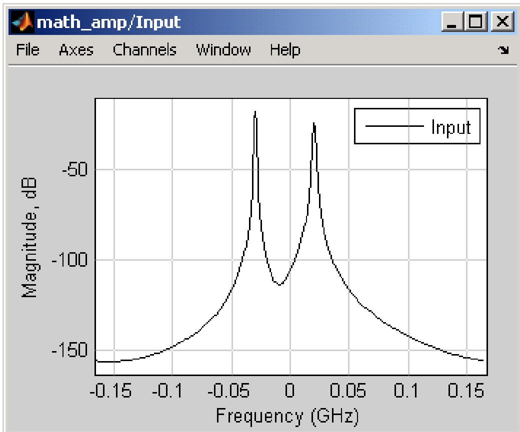


Effects of the Amplifier Block

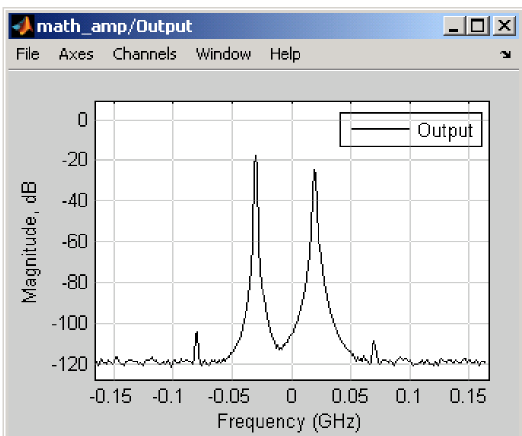
You can see the effect of the Amplifier block in the demo Intermodulation: Mathematical Amplifier.



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 6-7
- “Hyperbolic Tangent Model” on page 6-8
- “Saleh Model” on page 6-8
- “Ghorbani Model” on page 6-9
- “Rapp Model” on page 6-10

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

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

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

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

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

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

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

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

Thermal Noise Simulation

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

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

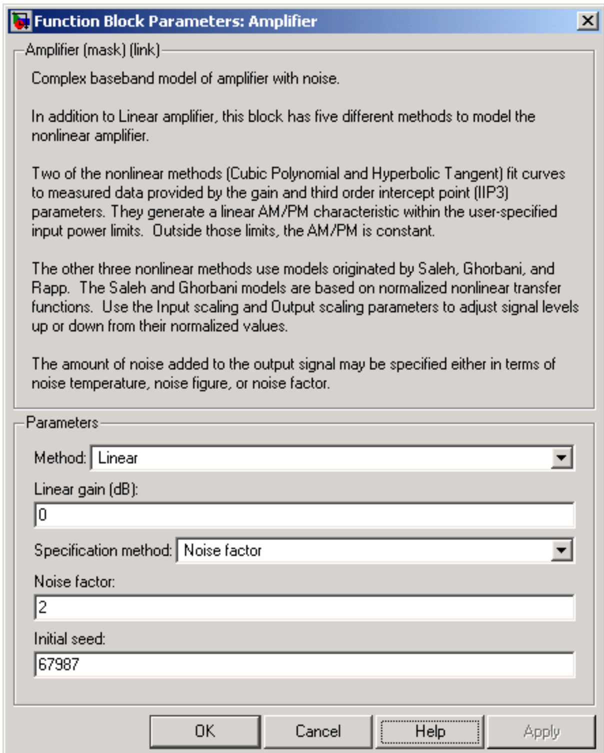
$$\text{Noise factor} = 1 + \frac{\text{Noise temperature}}{290}$$

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

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

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

Dialog Box



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

You can change tunable parameters while the model is running.

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

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

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

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

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

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

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

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 dB 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

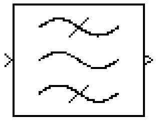
Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Lowpass RF Filter, Mixer

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.

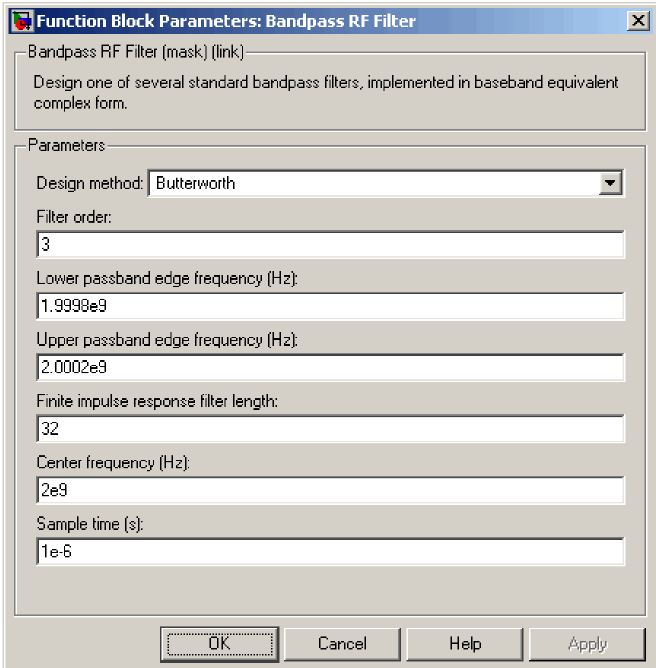
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

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 simulation frequencies.

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

See Also

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

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

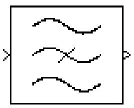
Purpose

Standard bandstop RF filters in baseband-equivalent complex form

Library

Mathematical

Description



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

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

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

Note This block assumes a nominal impedance of 1 ohm.

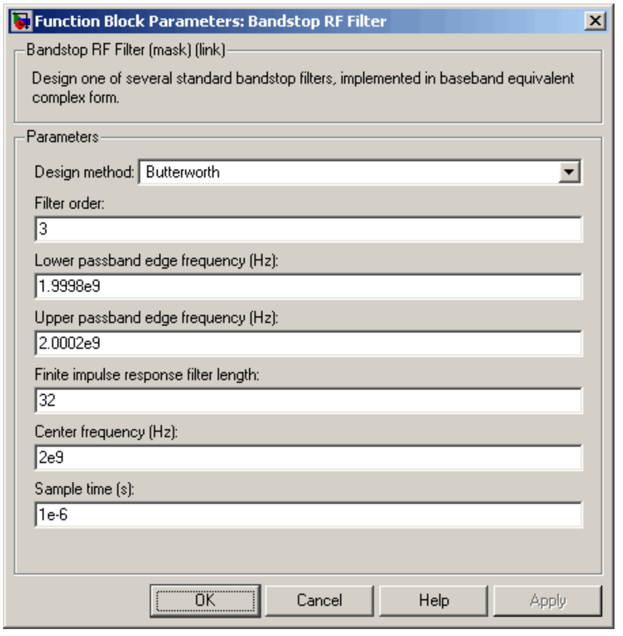
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

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

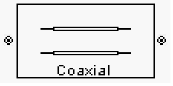
Purpose

Model coaxial transmission line

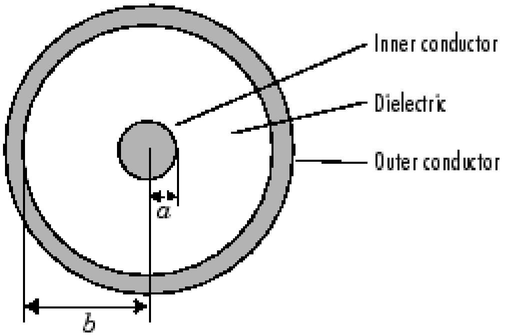
Library

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 here in cross-section. 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 simulation 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

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

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

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

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

In these equations, σ_{cond} is the conductivity in the conductor and σ_{diel} is the conductivity in the dielectric. μ is the permeability of the dielectric, ε is its permittivity, and skin depth δ is calculated as $1/\sqrt{\pi f\mu\sigma_{cond}}$. f is a vector of simulation frequencies determined by the Output Port block.

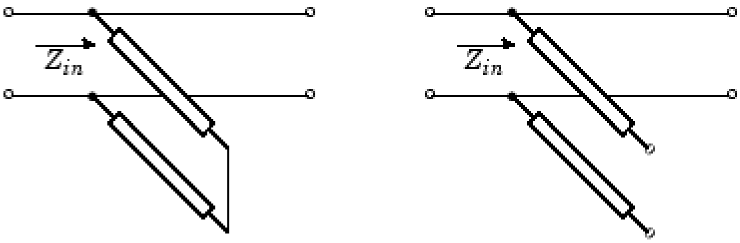
The Coaxial Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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

$A = 1$

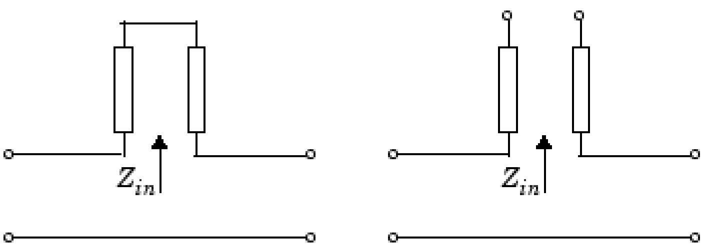
$B = 0$

$C = 1/Z_{in}$

$D = 1$

Series ABCD-Parameters

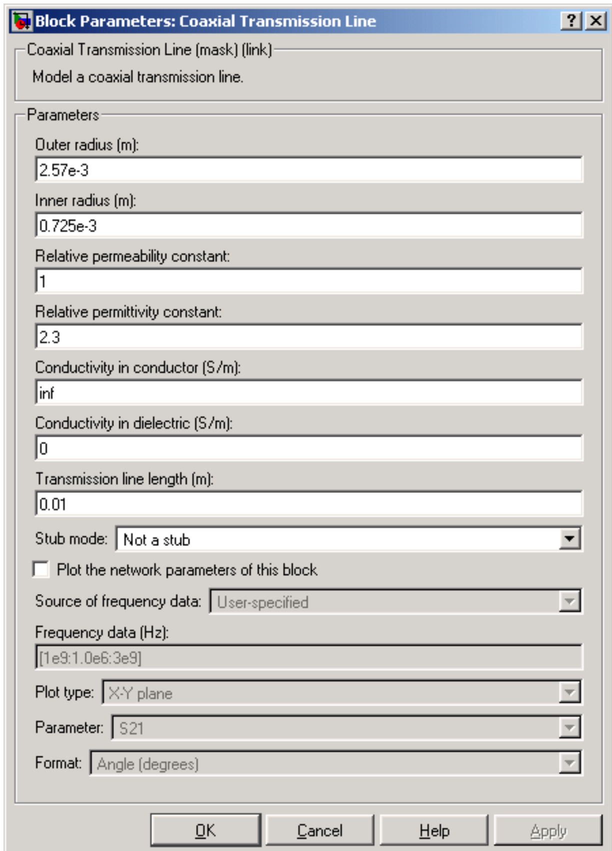
When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.



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



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 μ_0.

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

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

References

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

See Also

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

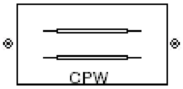
Purpose

Model coplanar waveguide transmission line

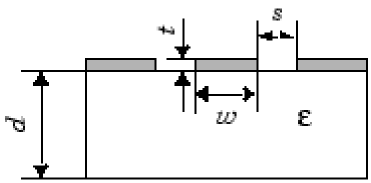
Library

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 here in cross-section. 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 (ε).



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 α_a is the attenuation coefficient and β is the wave number. The attenuation coefficient α_a is related to the loss, α, by

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

where α is the reduction in signal strength, in dB, per unit length. α 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 β is related to the phase velocity, V_P, by

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

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

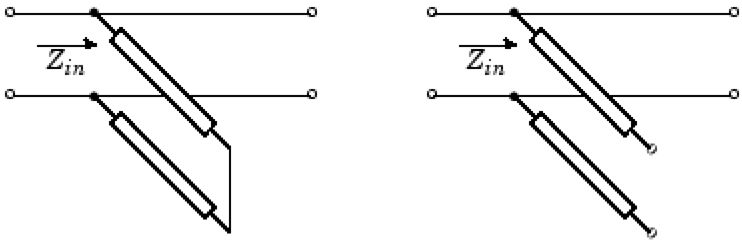
The Coplanar Waveguide Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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

$A = 1$

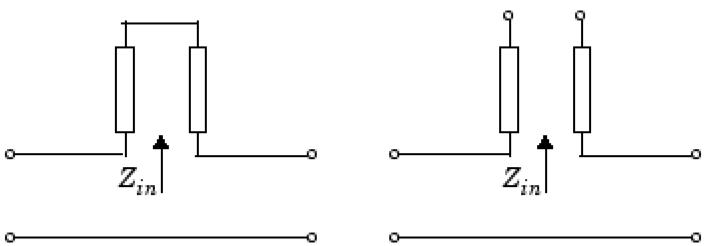
$B = 0$

$C = 1/Z_{in}$

$D = 1$

Series ABCD-Parameters

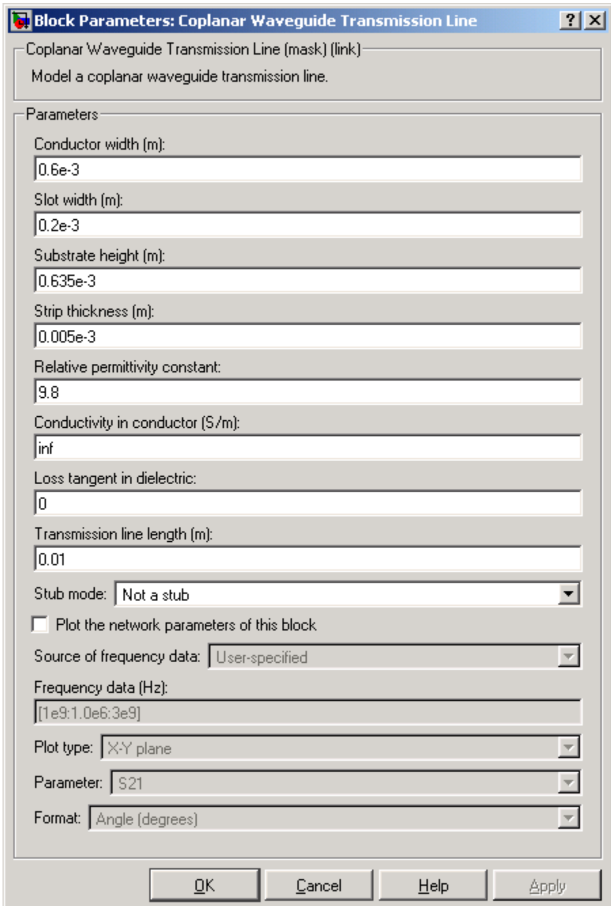
When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.



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



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 ε_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`.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

References

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

See Also

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

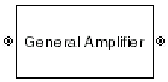
Purpose

Model nonlinear amplifier described by `rfdata` object

Library

Amplifiers sublibrary of the Physical library

Description



The General Amplifier block models the nonlinear amplifier described by an RF Toolbox data (`rfdata.data`) object.

Network Parameters

If network parameter data and their corresponding frequencies exist as S-parameters in the `rfdata.data` object, the General Amplifier block interpolates the S-parameters to determine the S-parameters at the simulation frequencies. If the block contains network Y- or Z-parameters, the block first converts them to S-parameters. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Nonlinearity

If power data exists in the `rfdata.data` object, the block extracts the AMAM/AMPM nonlinearities from the power data.

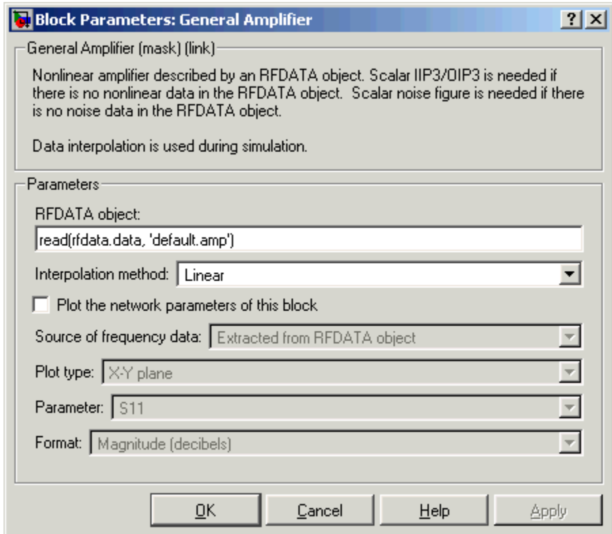
If the `rfdata.data` object contains no power data, the General Amplifier block dialog box lets you enter either the OIP3 or IIP3 as a scalar value for nonlinearity.

Active Noise

If active spot noise data exists in the `rfdata.data` object, the block uses the data to calculate the noise figure. It first interpolates the noise data for the simulation frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

If the `rfdata.data` object contains no noise data, the General Amplifier block dialog lets you enter a value for the noise figure.

Dialog Box



RFDATA object

An RF Toolbox data (`rfdata.data`) object. You can specify the object as (1) the handle of a data object previously created using the RF Toolbox, (2) an RF Toolbox command such as `rfdata.data`, which creates a default 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

For network data, the method used to interpolate the parameters contained in the `rfdata.data` object. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

IP3 type

Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point). This parameter becomes visible only if the `rfdata.data` object contains no power data.

IIP3 (dBm)
Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type. This parameter becomes visible only if the `rfdata.data` object contains no power data.

OIP3 (dBm)
Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type. This parameter becomes visible only if the `rfdata.data` object contains no power data.

Noise figure (dB)
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 becomes visible only if the `rfdata.data` object contains no noise data.

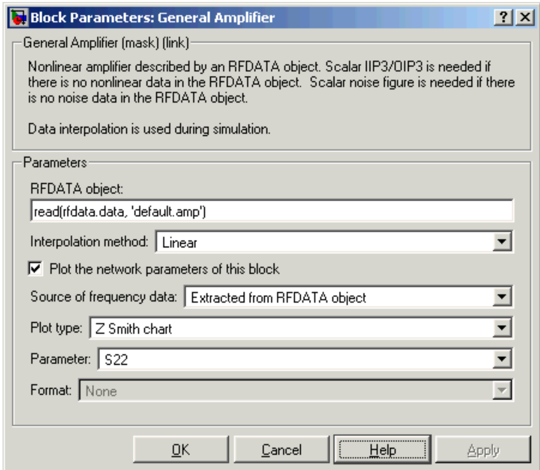
Note For information about plotting the network parameters, see Chapter 4, "Plotting Network Parameters". Use `rftool` or the RF Toolbox plotting functions to plot other data.

Examples

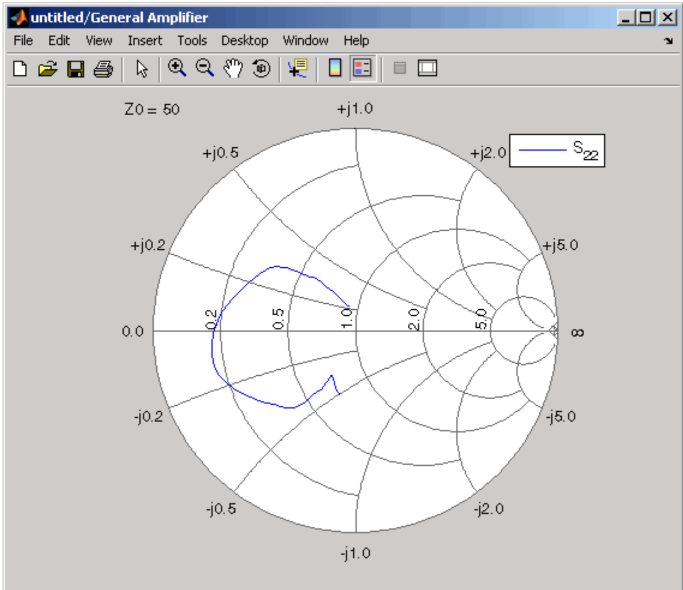
Creating a General Amplifier Block from File Data

This example uses the RF Toolbox `read` function to create an `rfdata.data` object that describes the nonlinear amplifier in the file `default.amp`. The file, which is read into an RF Toolbox data (`rfdata.data`) object, 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. The General Amplifier block uses linear interpolation to model the network described in the object.

Note See "AMP File Format" in the RF Toolbox documentation for information about `.amp` files.



The plot parameters in the dialog box request a Z Smith chart of the S22 parameters using the frequencies taken from the **RFDATA object** parameter.



See Also

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

`rfdata`, `rfdata.data` (RF Toolbox)

`interp1` (MATLAB)

General Circuit Element

Purpose

Model two-port network described by `rfckt` object

Library

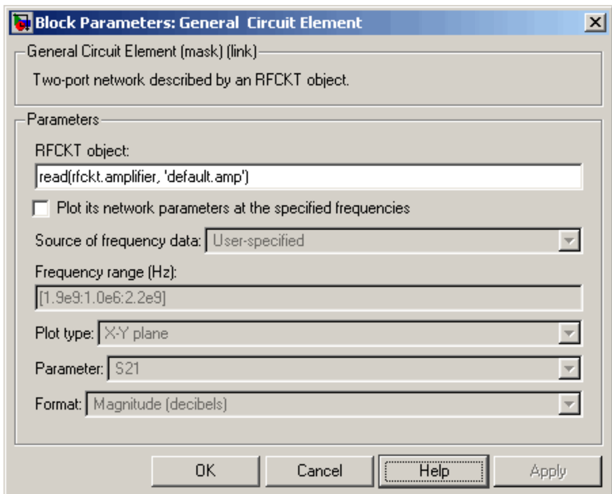
Black Box Elements sublibrary of the Physical library

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 simulation frequencies.

Dialog Box



RFCKT object

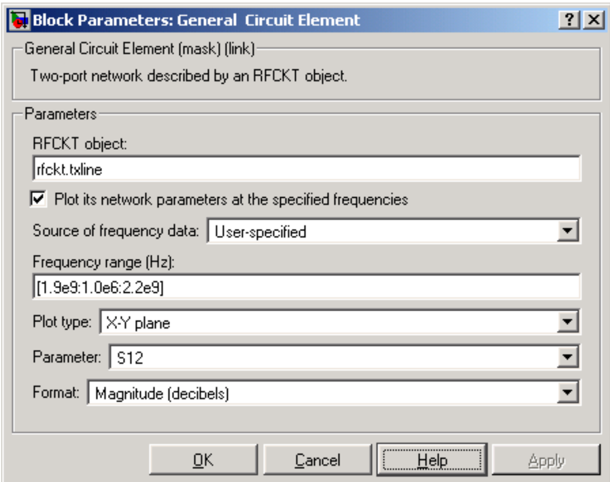
An RF Toolbox circuit (`rfckt`) object. You can specify the object as (1) the handle of a circuit object previously created using the RF Toolbox, (2) an RF Toolbox command such as `rfckt.txline`, `rfckt.coaxial`, or `rfckt.cascade` that creates a default circuit object of the specified type, or (3) a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

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

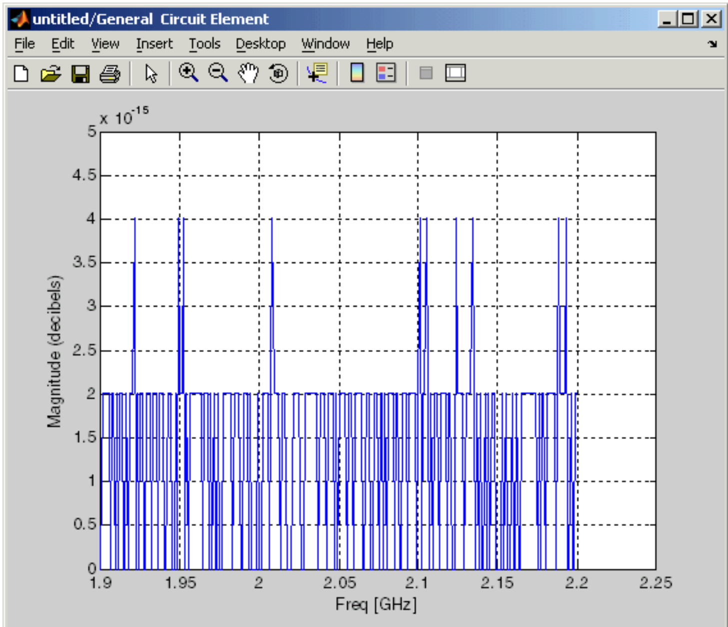
Examples

Creating a General Circuit Element from an RF Toolbox Object

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



The plot parameters in the dialog box request an X-Y plane plot of the S12 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

`rfckt` (RF Toolbox)

`interp1` (MATLAB)

Purpose

Model mixer 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

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 the S-parameters at the simulation frequencies. If the block contains network Y- or Z-parameters, the block first converts them to S-parameters. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Active Noise

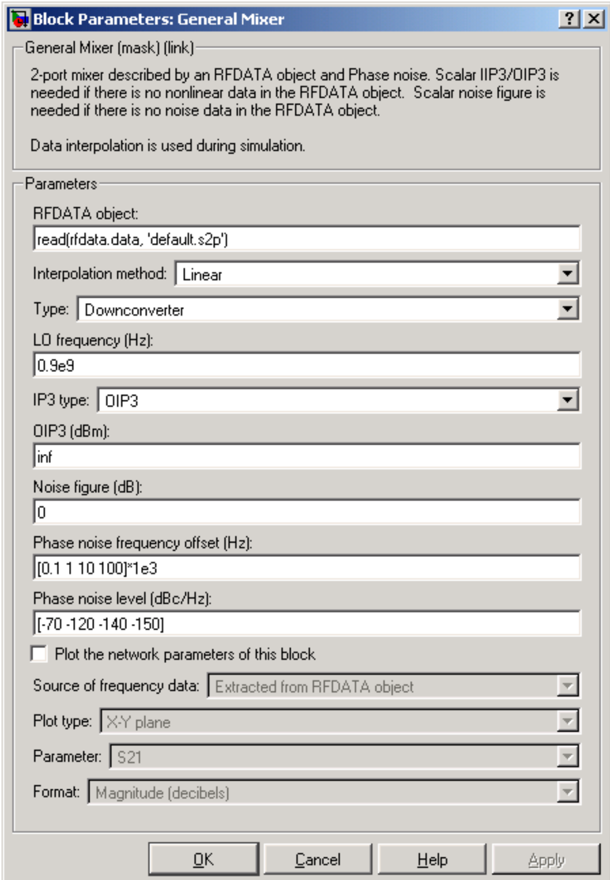
If active spot noise data exists in the `rfdata.data` object, the block uses the data to calculate the noise figure. It first interpolates the noise data for the simulation frequencies, using the specified **Interpolation method**. It then calculates the noise figure using the resulting values.

If the `rfdata.data` object contains no noise data, the General Mixer block dialog lets you enter a value for the noise figure.

Phase Noise

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

Dialog Box



RFDATA object

An RF Toolbox data object (`rfdata.data`) that describes a mixer. You can specify the object as (1) the handle of a data object previously created using the RF Toolbox, (2) an RF Toolbox command such as `rfdata.data` that creates a default 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

For network data, the method used to interpolate the parameters contained in the `rfdata.data` object. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

Type

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

LO frequency (Hz)

Local oscillator frequency. If you choose `Downconverter`, $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

IP3 type

Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point). This parameter becomes visible only if the `rfdata.data` object contains no power data.

IIP3 (dBm)

Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type. This parameter becomes visible if the `rfdata.data` object contains no power data and you select `IIP3` as the IP3 type.

OIP3 (dBm)

Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type. This parameter becomes visible if the `rfdata.data` object contains no power data and you select `OIP3` as the IP3 type.

Noise figure (dB)

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 becomes visible only if the `rfdata.data` object contains no noise data.

Phase noise frequency offset (Hz)

Vector specifying the frequency offset.

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

Note For information about plotting the network parameters, see Chapter 4, "Plotting Network Parameters". Use `RFTool` or the RF Toolbox plotting functions to plot other data.

See Also

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

`rfdata`, `rfdata.data` (RF Toolbox)

Purpose

Model two-port passive network described by `rfdata` object

Library

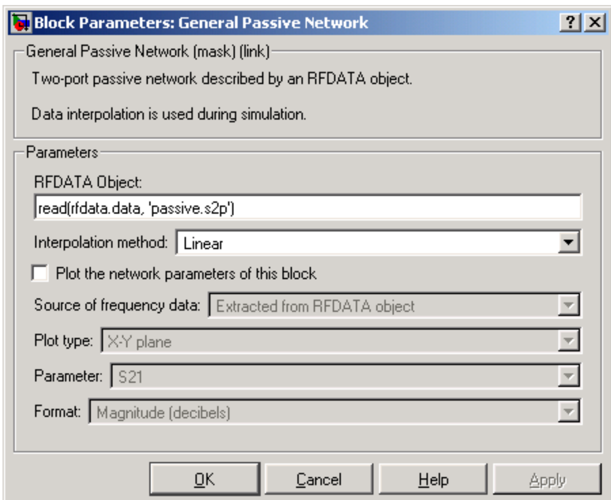
Black Box Elements sublibrary of the Physical library

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 the S-parameters at the simulation frequencies. If the block contains network Y- or Z-parameters, the block first converts them to S-parameters. See “S-Parameters at Simulation Frequencies” on page 3-2 for more details.

Dialog Box



RFDATA object

An RF Toolbox data (`rfdata.data`) object. You can specify the object as (1) the handle of a data object previously created using the RF Toolbox, (2) an RF Toolbox command such as `rfdata.data` that creates a default 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

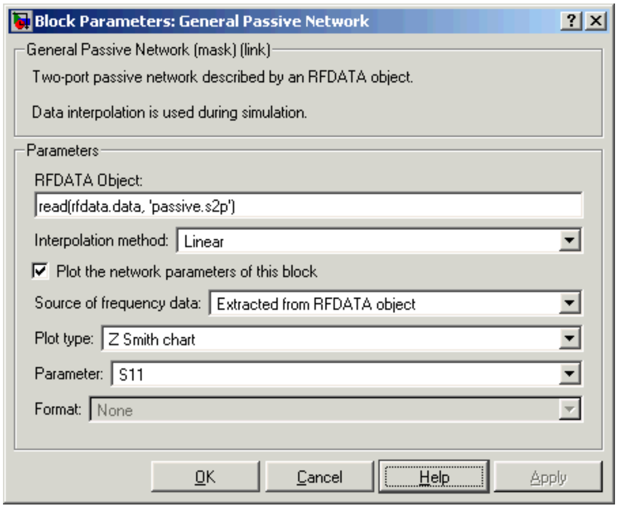
Method used to interpolate the parameters contained in the `rfdata` object. Interpolation can be `Linear` (default), `Spline`, or `Cubic`.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

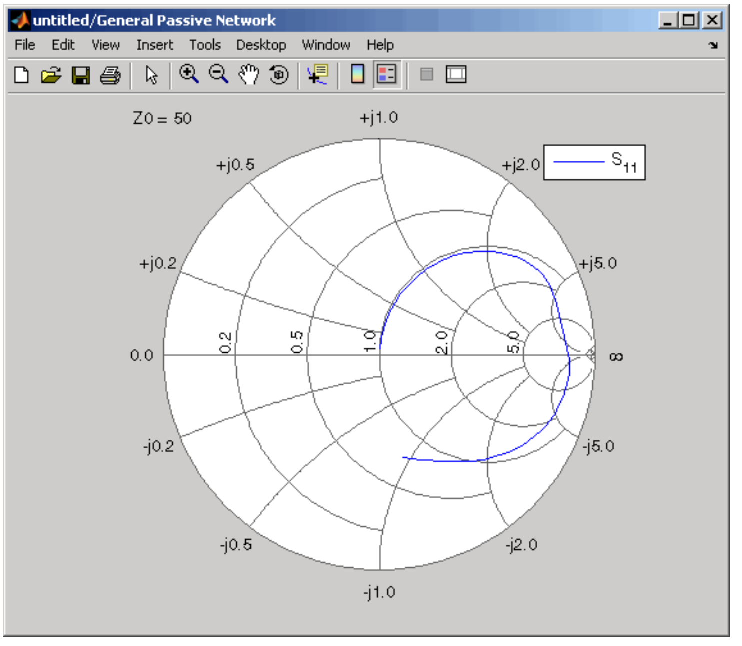
Examples

Creating a General Passive Network Block from File Data

This example uses the RF Toolbox read function to create an `rfdata.data` object that describes the two-port passive network 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.



The plot parameters in the dialog box request a Z Smith chart of the S11 parameters using the frequencies taken from the **RFDATA object** parameter.



See Also

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

`rfdata`, `rfdata.data` (RF Toolbox)

`interp1` (MATLAB)

Highpass RF Filter

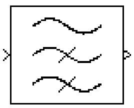
Purpose

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.

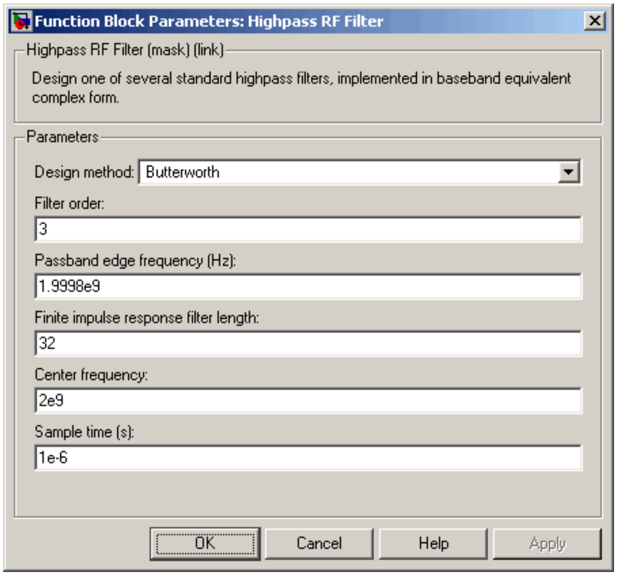
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block lets you specify the filter design parameters shown in the following table.

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 simulation 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)

Input Port

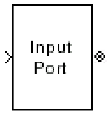
Purpose

Connection block from Simulink environment to RF physical blocks

Library

Input/Output Ports sublibrary of the Physical library

Description

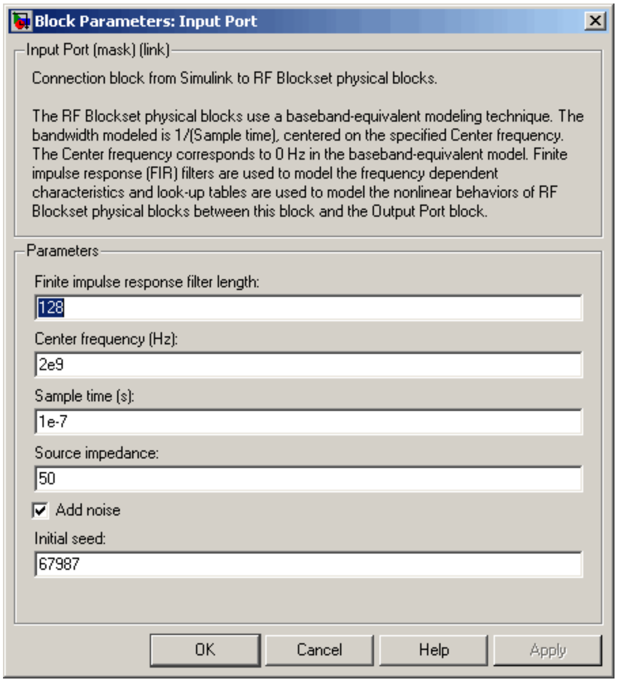


The Input Port block serves as a connecting port from the Simulink, or mathematical, part of the model to an RF physical part of the model. The Input Port block lets you provide the parameter data needed to calculate the simulation frequencies and the baseband-equivalent impulse response for the physical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

For more information about connecting mathematical and physical parts of a model, see Chapter 2, “Creating an RF Model”.

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 1-6.

Center frequency (Hz)

Center of the simulation frequencies. See the Output Port block reference page for information about calculating the simulation 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

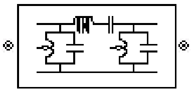
Purpose

Model LC bandpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description



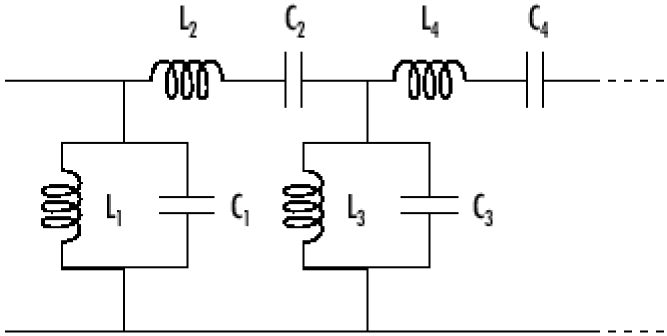
The LC Bandpass Pi block models the LC bandpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

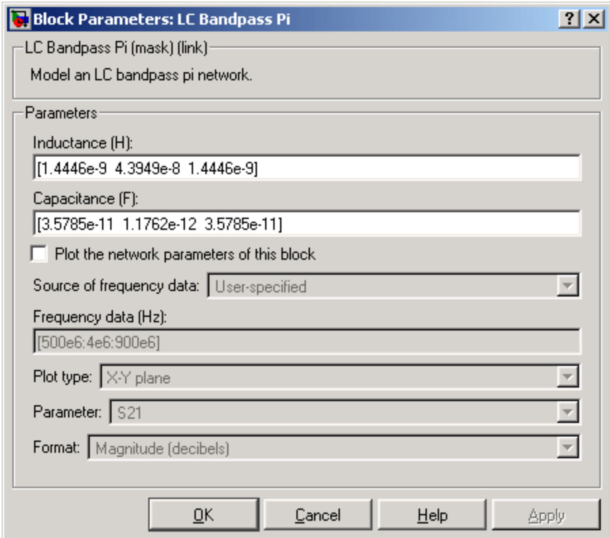
See the Output Port block for information about determining the simulation frequencies.

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



[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



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.

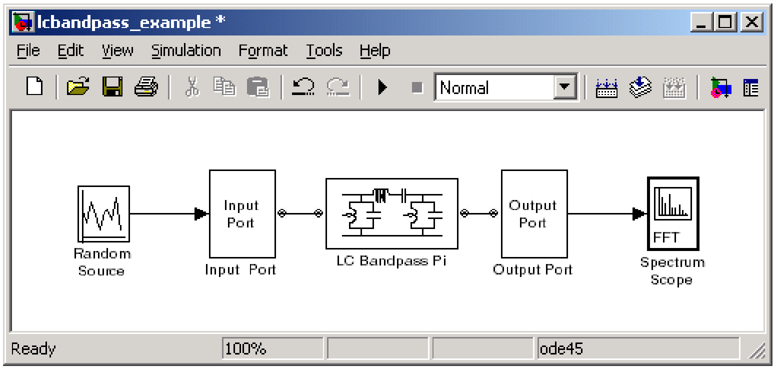
Capacitance (F)
: Vector containing the capacitances, in order from source to load, of all capacitors in the network. Its length must be equal to the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Note For information about plotting, see Chapter 4, “Plotting Network Parameters”.

Examples

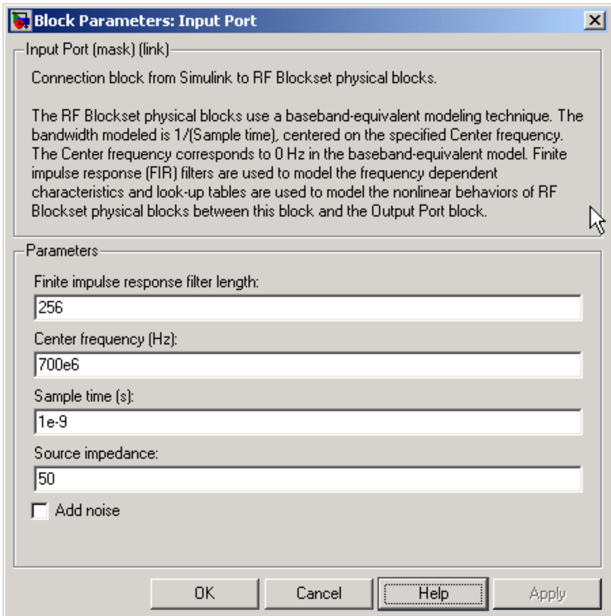
Using a Ladder Filter Block to Filter Gaussian Noise

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.

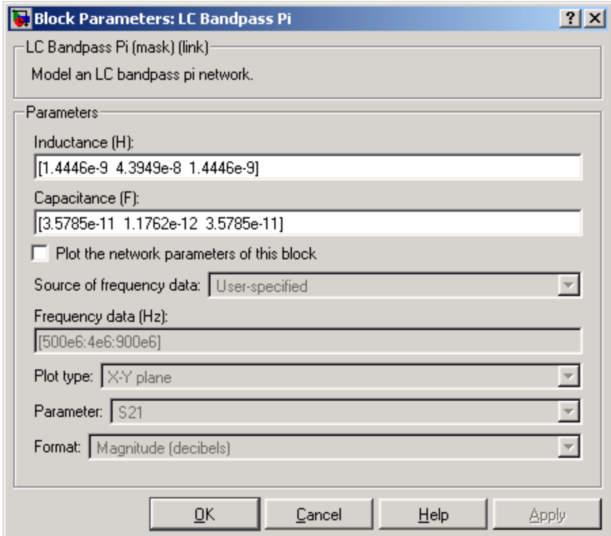


The Random Source block (Signal Processing Blockset) produces frame-based output at 512 samples per frame. Its **Sample time** parameter is set to 1.0e-9. This sample time must match the sample time for the physical part of the model, which you provide in the Input Port block diagram.

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.

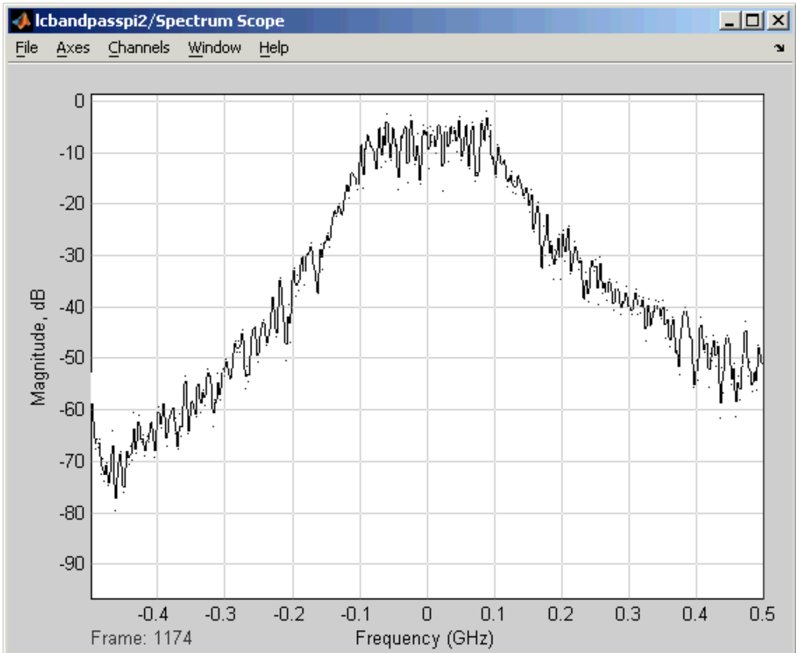


The LC Bandpass Pi block provides the inductances for three inductors, in order from source to load, [1.4446e-9, 4.3949e-8, 1.4446e-9]. Similarly, it provides the capacitances for three capacitors [3.5785e-11, 1.1762e-12, 3.5785e-11].



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.

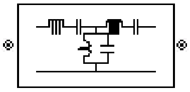
See Also

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.

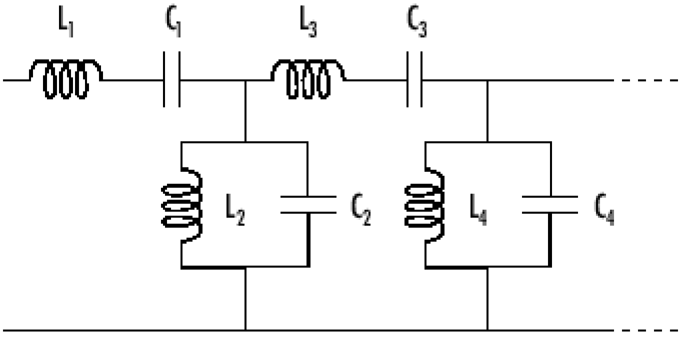


For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation frequencies. For each series pair, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series pair. For each shunt pair, A = 1, B = 0, 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

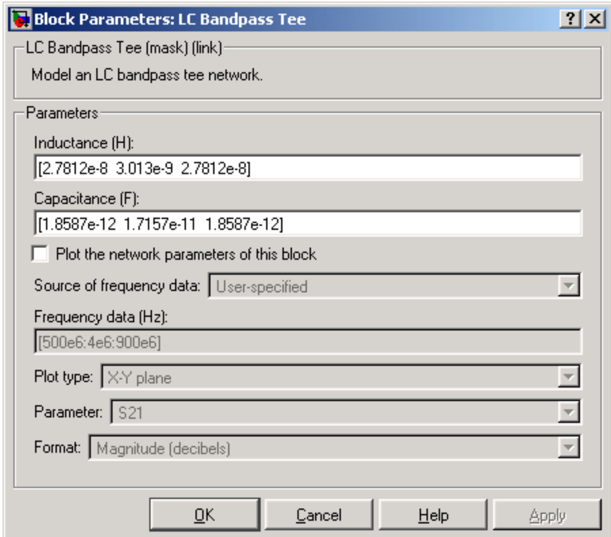
See the Output Port block reference page for information about determining the simulation frequencies.

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



[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



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.

Note For information about plotting, see Chapter 4, “Plotting Network Parameters”.

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

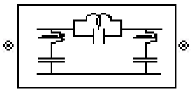
Purpose

Model LC bandstop pi network

Library

Ladder Filters sublibrary of the Physical library

Description



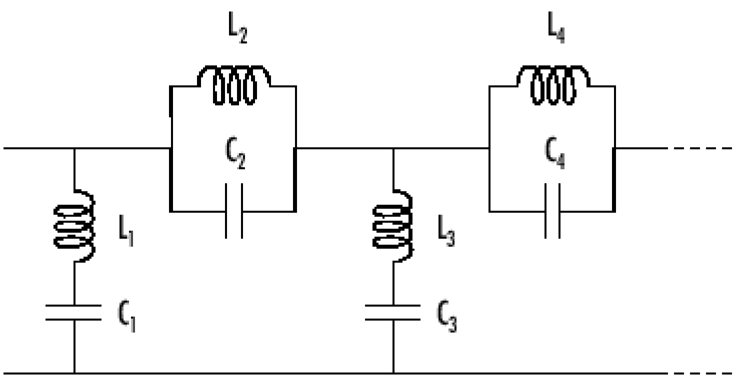
The LC Bandstop Pi block models the LC bandstop pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

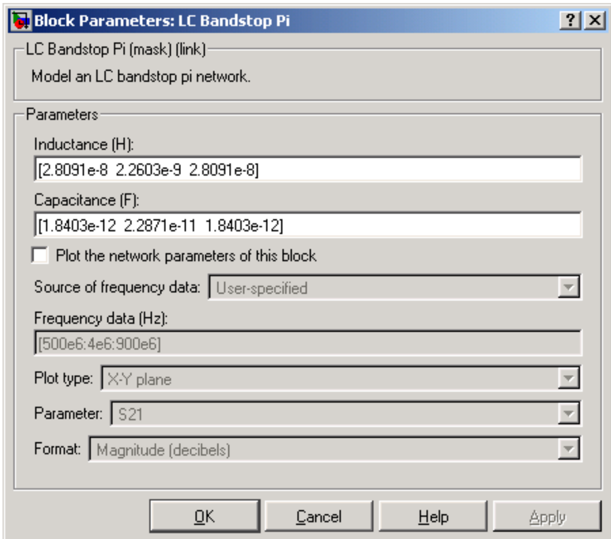
See the Output Port block for information about determining the simulation frequencies.

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



[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



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.

Note For information about plotting, see Chapter 4, “Plotting Network Parameters”.

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

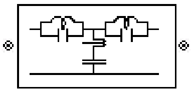
Purpose

Model LC bandstop tee network

Library

Ladder Filters sublibrary of the Physical library

Description



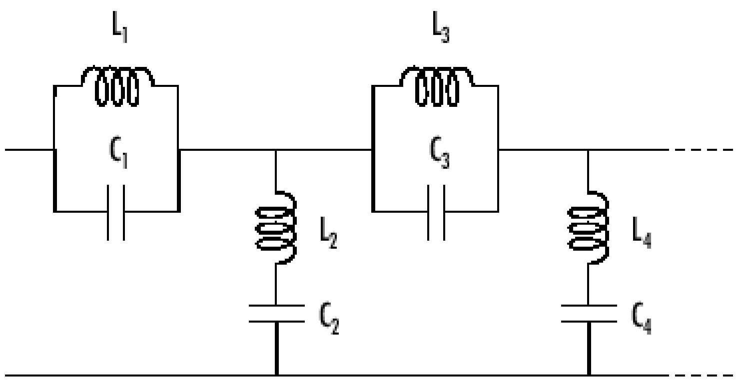
The LC Bandstop Tee block models the LC bandstop tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor pair in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

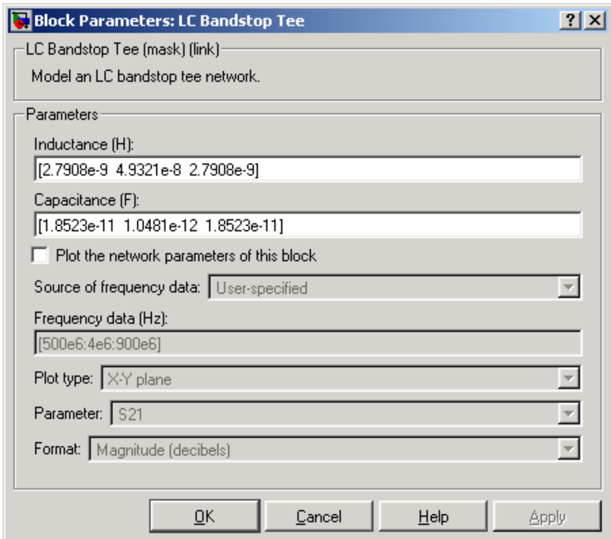
See the Output Port block for information about determining the simulation frequencies.

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



[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



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.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

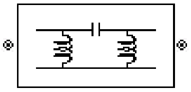
Purpose

Model LC highpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description



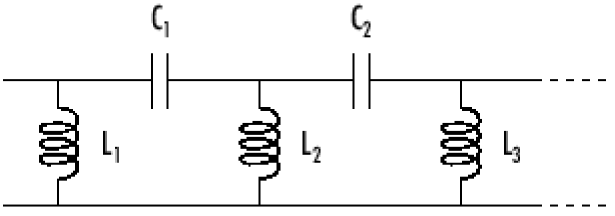
The LC Highpass Pi block models the LC highpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

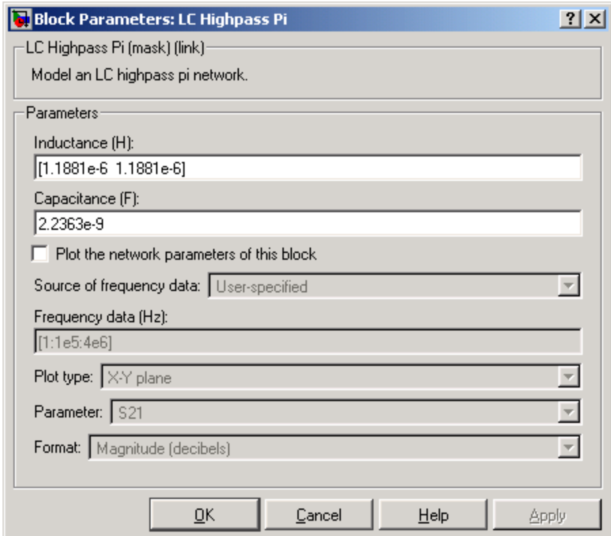
See the Output Port block reference page for information about determining the simulation frequencies.

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



[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



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.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

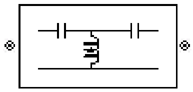
Purpose

Model LC highpass tee network

Library

Ladder Filters sublibrary of the Physical library

Description



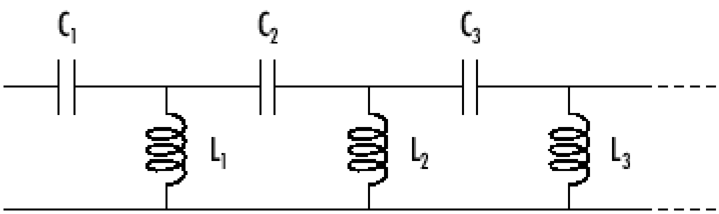
The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

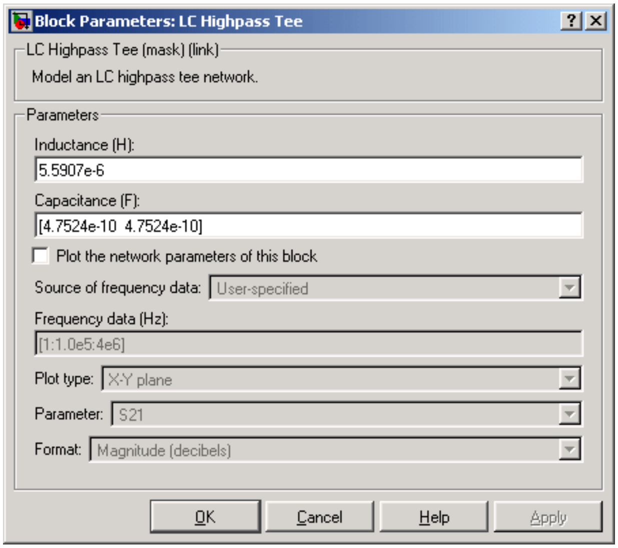
See the Output Port block reference page for information about determining the simulation frequencies.

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



[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



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.

Note For information about plotting, see Chapter 4, “Plotting Network Parameters”.

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

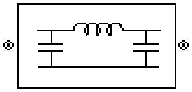
General Passive Network, LC Bandpass Pi, LC Bandpass Tee, LC Bandstop Pi, LC Bandstop Tee, LC Highpass Pi, LC Lowpass Pi, LC Lowpass Tee, Series RLC, Shunt RLC

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.

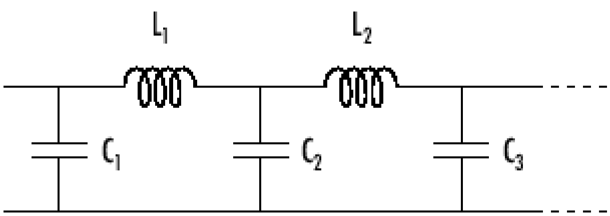


For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation frequencies. For each series circuit, A = 1, B = Z, C = 0, and D = 1, where Z is the impedance of the series circuit. For each shunt, A = 1, B = 0, C = 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

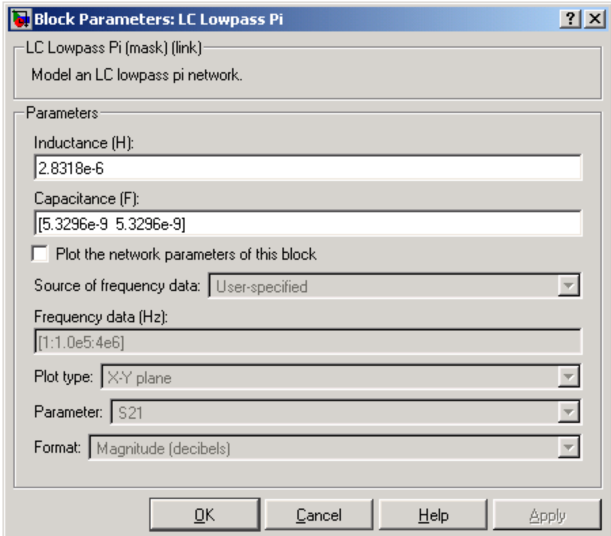
See the Output Port block reference page for information about determining the simulation frequencies.

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



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

Dialog Box



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.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

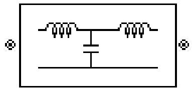
Purpose

Model LC lowpass tee network

Library

Ladders Filters sublibrary of the Physical library

Description



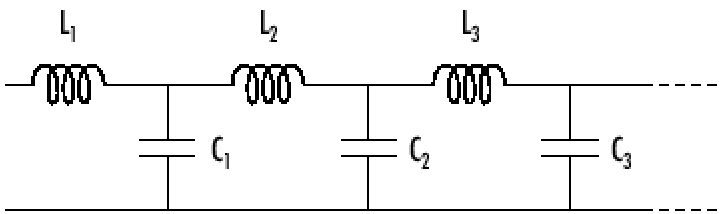
The LC Lowpass Tee block models the LC lowpass tee network described in the block dialog box in terms of its frequency-dependent S-parameters.

For each inductor and capacitor in the network, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation 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 simulation frequencies, and converts the cascaded parameters to S-parameters using the RF Toolbox `abcd2s` function.

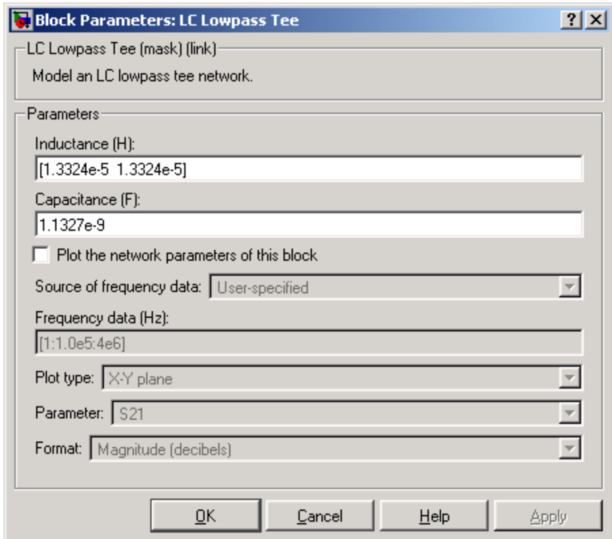
See the Output Port block reference page for information about determining the simulation frequencies.

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



[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



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.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

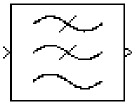
Purpose

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.

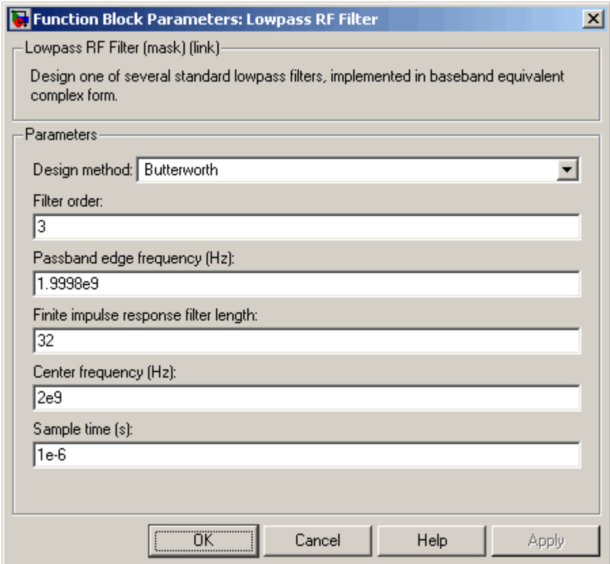
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

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 simulation 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

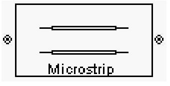
Purpose

Model microstrip transmission line

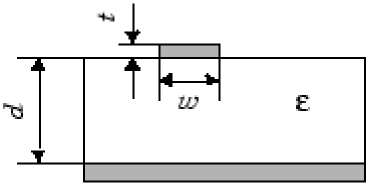
Library

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 here in cross-section. Its physical characteristics include the microstrip width (w), the microstrip thickness (t), the substrate height (d), and the relative permittivity constant (ε).



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 α_a is the attenuation coefficient and β is the wave number. The attenuation coefficient α_a is related to the loss, α, by

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

where α is the reduction in signal strength, in dB, per unit length. α 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 β is related to the phase velocity, V_P, by

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

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

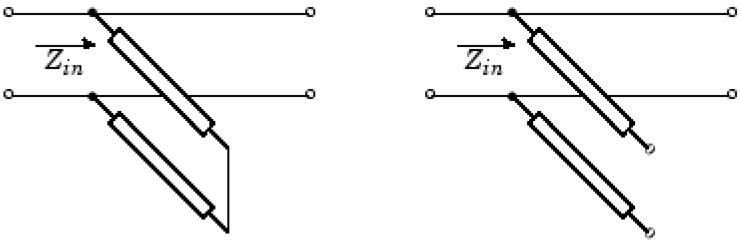
The Microstrip Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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

$A = 1$

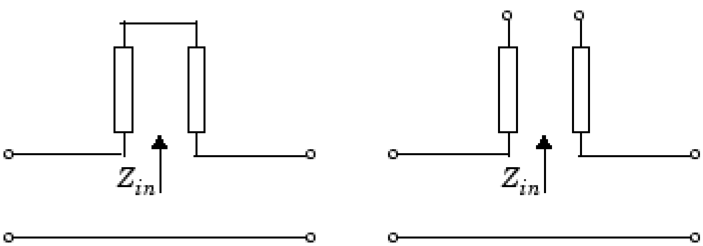
$B = 0$

$C = 1/Z_{in}$

$D = 1$

Series ABCD-Parameters

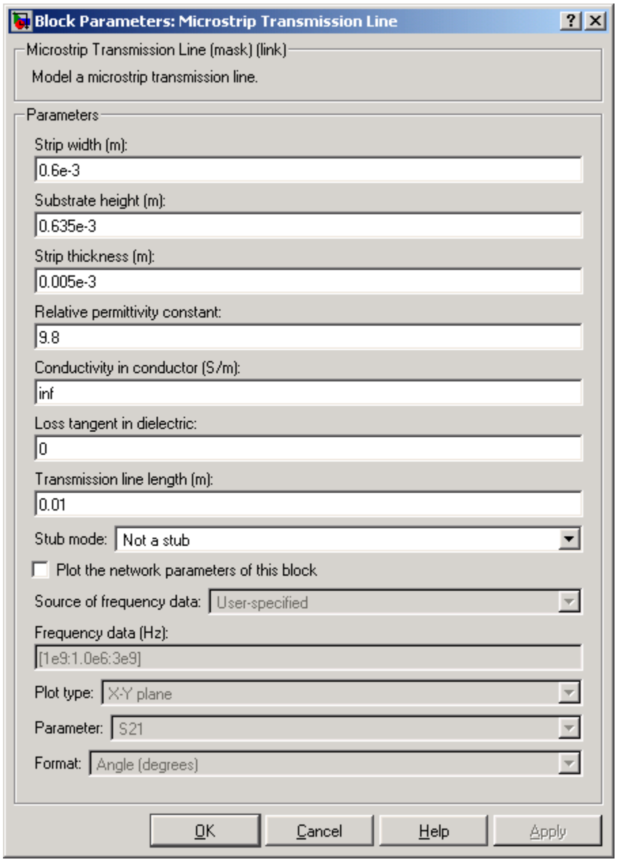
When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.



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



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 ε_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`.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

References

[1] Gupta, K.C., G. Ramesh, I. Bahl, and P. Bhartia, *Microstrip Lines and Slotlines*, Second Edition, Artech House, 1996. pp. 102-109.

See Also

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

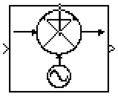
Purpose

Complex baseband model of mixer with phase noise

Library

Mathematical

Description



The Mixer block generates a complex baseband model of a mixer, with phase noise whose spectrum is characterized by a $1/f$ slope. The level of the spectrum is specified by the noise power contained in one hertz bandwidth offset from the carrier by a certain frequency.

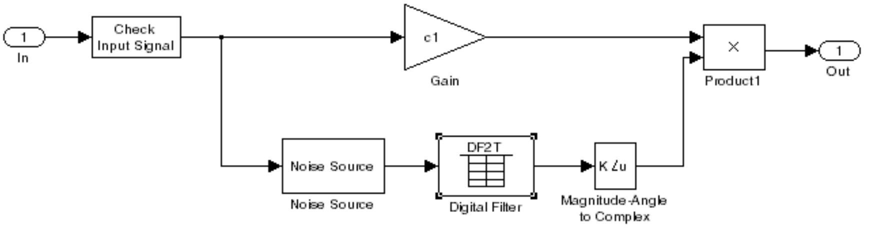
Note This block assumes a nominal impedance of 1 ohm.

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 noise to the angle component of the input signal.

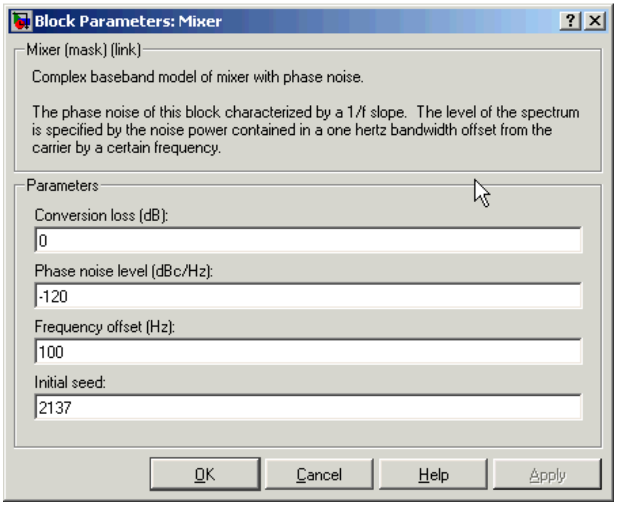
You can view the block's implementation of phase noise by right-clicking the block and selecting **Look under mask** from the pop-up menu. This displays the following figure.



You can view the construction of the Noise Source subsystem by double-clicking it.

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



You can change parameters that are marked as tunable while the model is running.

Conversion loss (dB)
: Scalar specifying the conversion loss for the mixer. 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

Output Port

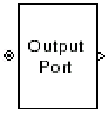
Purpose

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 connecting mathematical and physical parts of a model, see Chapter 2, "Creating an RF Model".

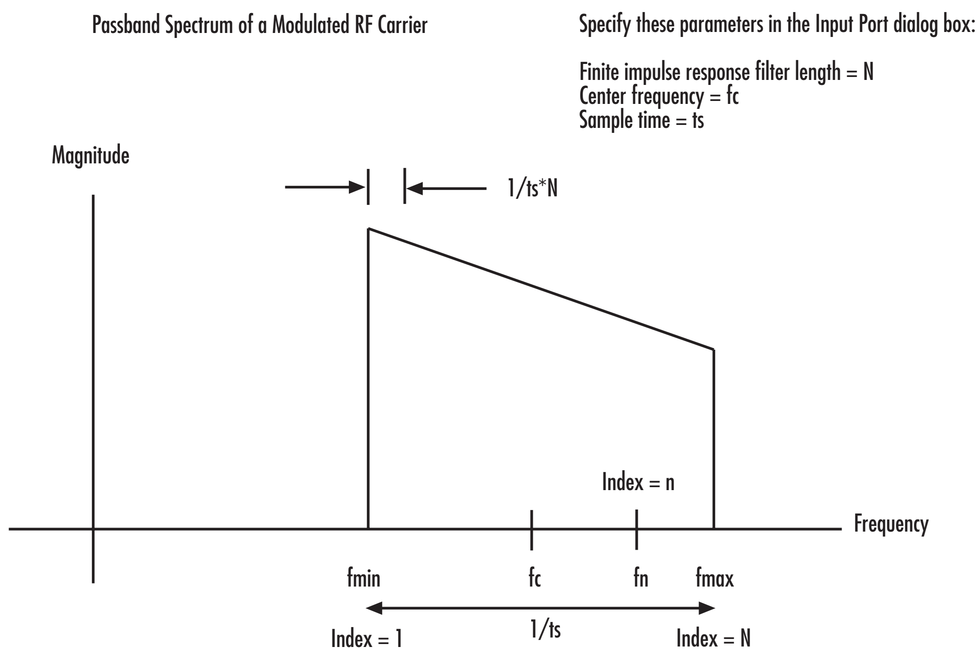
Note Some RF blocks require the sample time to perform baseband modeling calculations. To ensure the accuracy of these calculations, the Input Port block, as well as the mathematical RF blocks, compare the input sample time to the sample time you provide in the mask. If they do not match, or if the input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

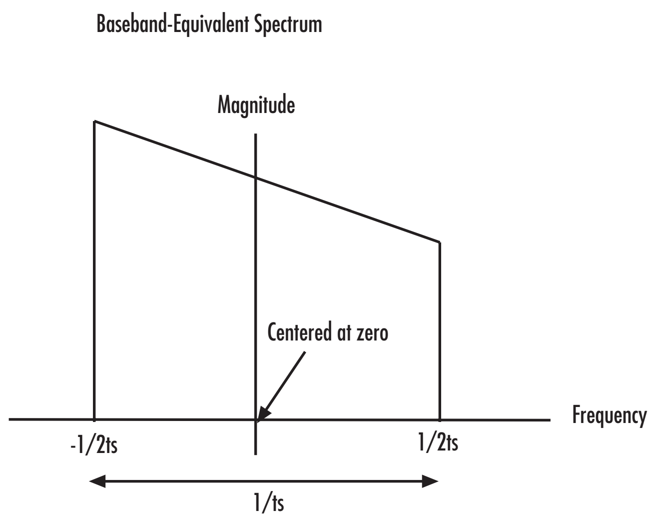
Linear Subsystem

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 Combines the S-parameters from the individual physical blocks and converts the resulting S-parameters to Y-parameters. See the RF Toolbox function `s2y` reference page for conversion details.

2 Determines the simulation frequencies f as an N-element vector. The simulation 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.


Passband Spectrum of a Modulated RF Carrier
Specify these parameters in the Input Port dialog box:
Finite impulse response filter length = N
Center frequency = fc
Sample time = ts
Magnitude
1/ts*N
Index = n
Frequency
fmin
fc
fn
fmax
1/ts
Index = 1
Index = N


Baseband-Equivalent Spectrum
Magnitude
Centered at zero
Frequency
-1/2ts
1/2ts
1/ts

The nth element of f, f_n, is given by

$$f_n = f_{min} + \frac{n-1}{t_s N} \qquad n = 1, \ldots, N$$

where

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

3 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 simulation frequencies. More specifically,

$$H(f) = \frac{-Y_{21}(f)}{(1 + Z_S(f) \bullet Y_{11}(f))(Y_{22}(f) + Z_L(f)^{-1}) - (Z_S(f) \bullet Y_{12}(f) \bullet Y_{21}(f))}$$

where

- Z_S is the source impedance.
- Z_L is the load impedance.
- Y_{ij} are the Y-parameters of a two-port network.

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

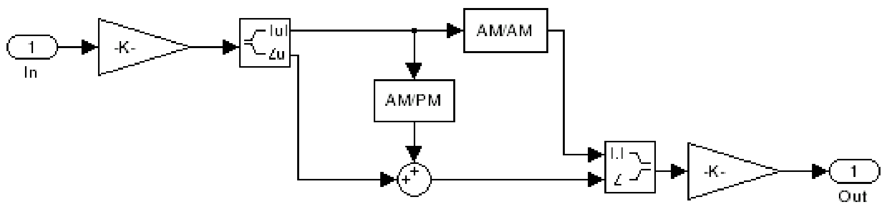
5 Obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. For reasons of efficiency, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. It

then truncates the impulse response to a length less than or equal to the 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 figure below.



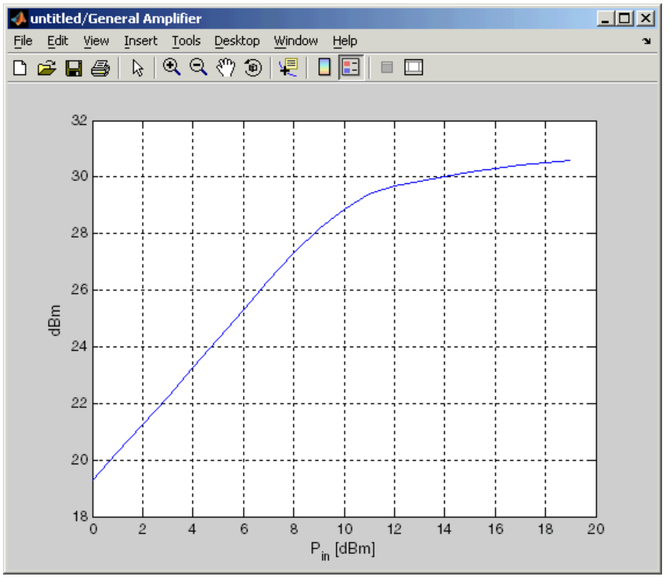
The nonlinearities of AM/AM and AM/PM conversions are extracted from the power data of an amplifier or mixer by the equations

$$AM_{\text{out}} = \sqrt{R_l \cdot P_{\text{out}}}$$
$$PM_{\text{out}} = \text{Phase}$$
$$AM_{\text{in}} = \sqrt{R_s \cdot P_{\text{in}}}$$

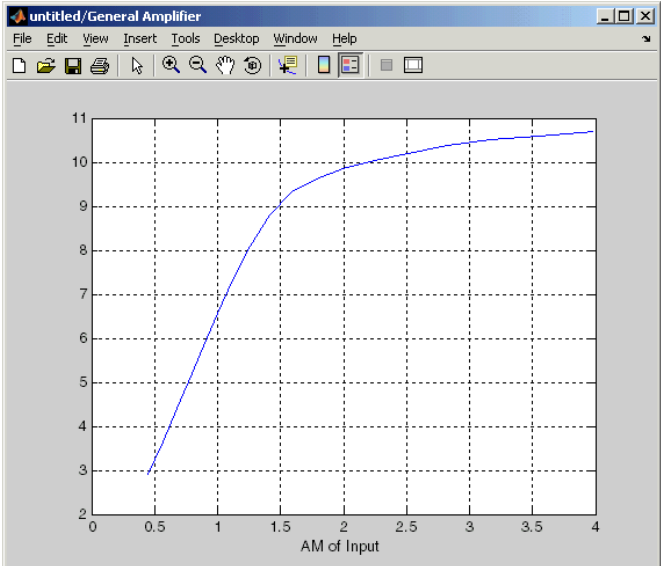
where AM_{in} is the AM of the input voltage, AM_{out} and PM_{out} are the AM and PM of the output voltage, R_s, is the source resistance (50 ohms), R_l is the load resistance (50 ohms), P_{in} is the input power, P_{out} is the output power, and Phase is the phase shift between the input and output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

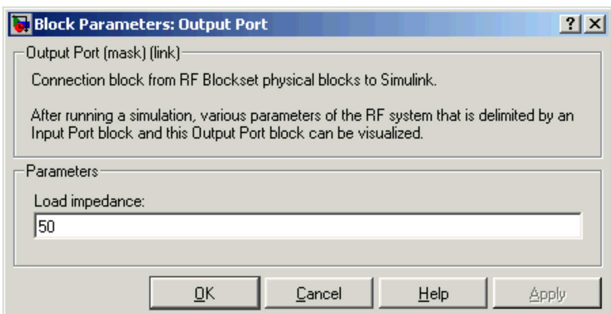
The following figure shows the original power data of an amplifier.



This figure shows the extracted AM/AM nonlinear conversion.



Dialog Box

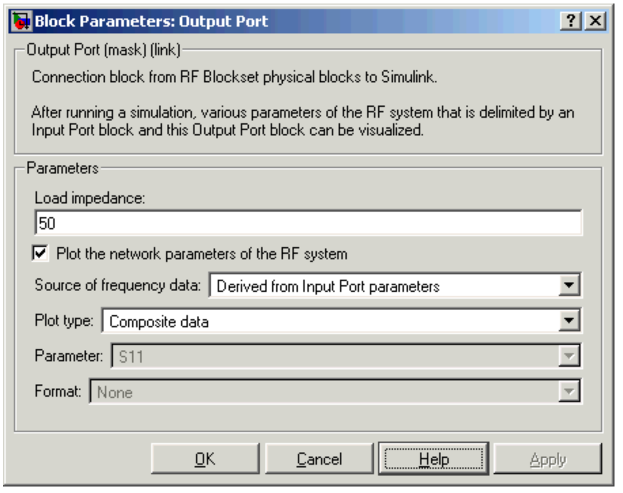


Load impedance

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

Plot the network parameters of the RF system

This parameter and the associated plotting parameters shown below become visible if you display the Output Port mask after you run the model. For more information about plotting, see Chapter 4, “Plotting Network Parameters”.



See Also

Input Port

`s2y` (RF Toolbox)

Parallel-Plate Transmission Line

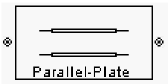
Purpose

Model parallel-plate transmission line

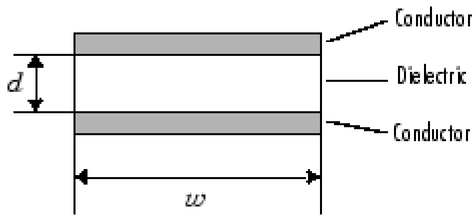
Library

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 here in cross-section. 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, σ_{cond} is the conductivity in the conductor and σ_{diel} is the conductivity in the dielectric. μ is the permeability of the dielectric, ε is its permittivity, and skin depth δ is calculated as $1/\sqrt{\pi f\mu\sigma_{cond}}$. f is the vector of simulation frequencies for the specified parameters. See the Output Port block reference page for information about determining the simulation frequencies.

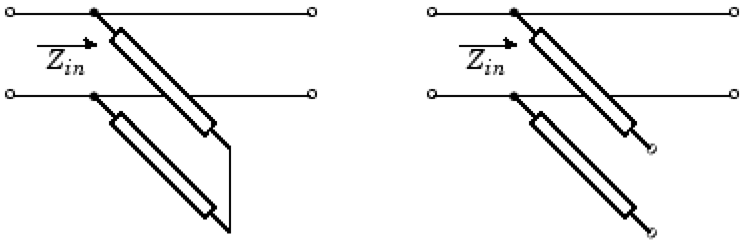
The Parallel-Plate Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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

$A = 1$

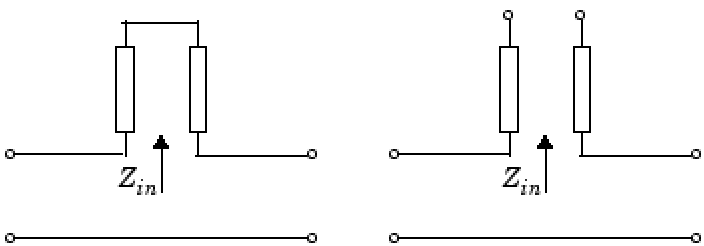
$B = 0$

$C = 1/Z_{in}$

$D = 1$

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

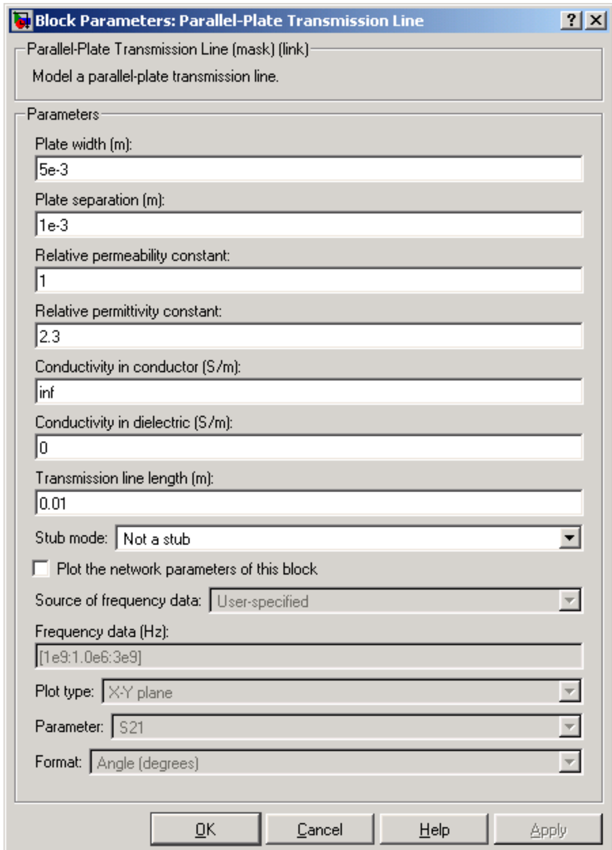


Plate width (m)

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 μ_0.

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

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

References

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

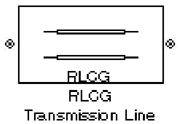
See Also

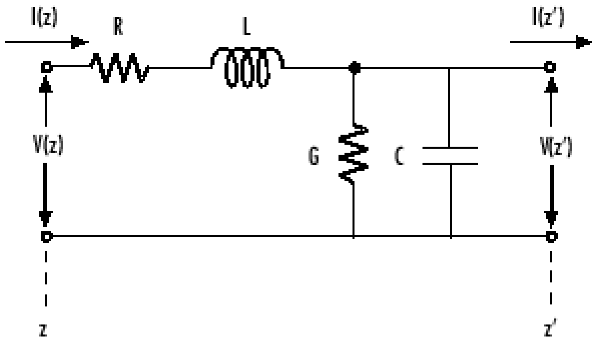
Coaxial Transmission Line, Coplanar Waveguide Transmission Line, General Passive Network, Transmission Line, Microstrip Transmission Line, Two-Wire Transmission Line

Purpose Model RLCG transmission line

Library Transmission Lines sublibrary of the Physical library

Description The RLCG Transmission Line block models the RLCG transmission line described in the block dialog box in terms of its frequency-dependent resistance, inductance, capacitance, and conductance. The transmission line, which can be lossy or lossless, is treated as a two-port linear network.





where $z' = z + \Delta z$.

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

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

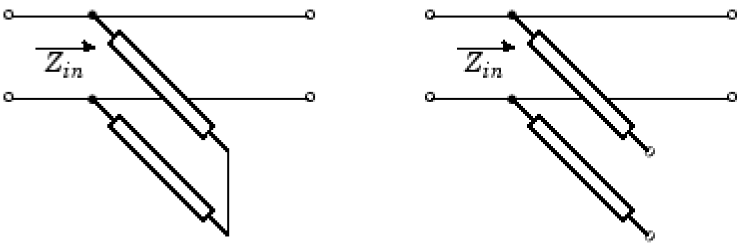
The RLCG Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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

$A = 1$

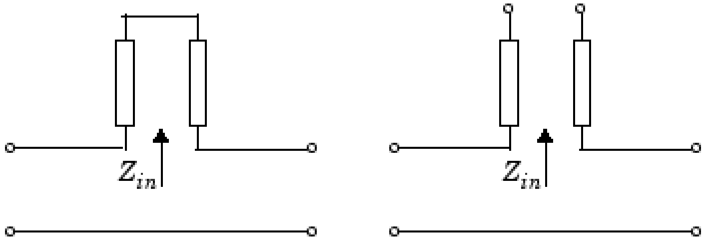
$B = 0$

$C = 1/Z_{in}$

$D = 1$

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

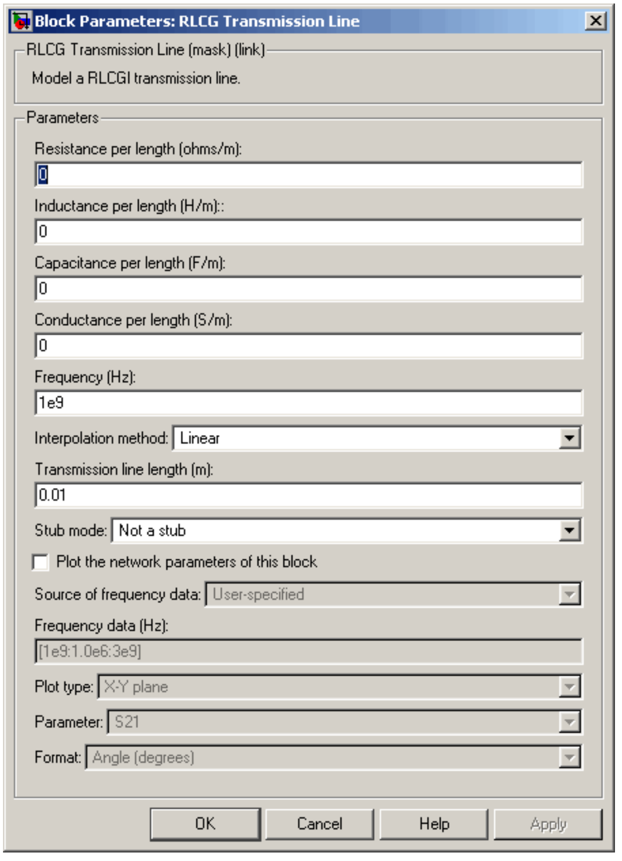
$A = 1$

$B = Z_{in}$

$C = 0$

$D = 1$

Dialog Box



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 simulation 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`.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

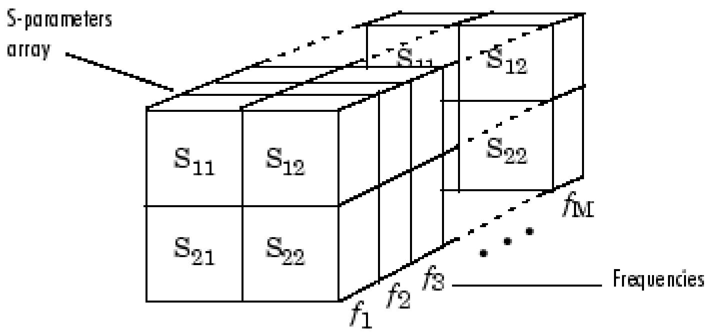
Purpose Model nonlinear amplifier using its S-parameters

Library Amplifiers sublibrary of the Physical library

Description

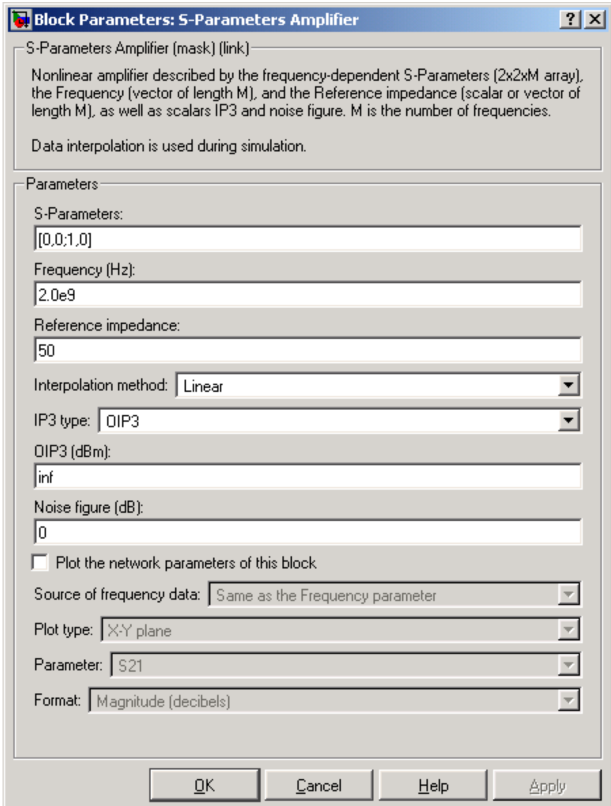
The S-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent S-parameters, their frequencies, and the reference impedance of the S-parameters. The block also takes into account the IP3 value and the noise figure.

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 the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Dialog Box



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, that value is applied to all frequencies.

Interpolation method
Method used to interpolate the given S-parameters over the range of frequencies. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

IP3 type
Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IIP3 (dBm)
Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type.

OIP3 (dBm)
Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type.

Noise figure (dB)
Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $((S_i/N_i)/(S_o/N_o))$.

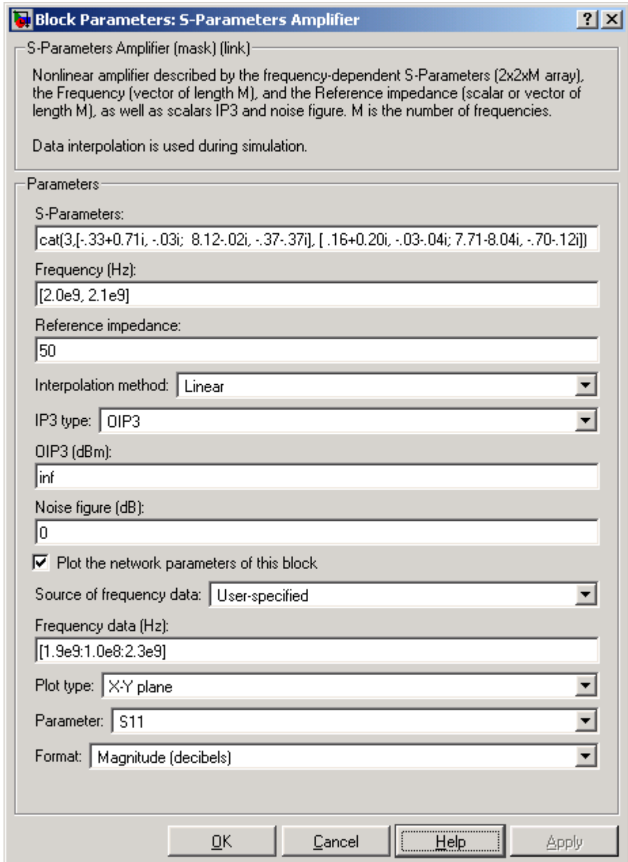
Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

Examples

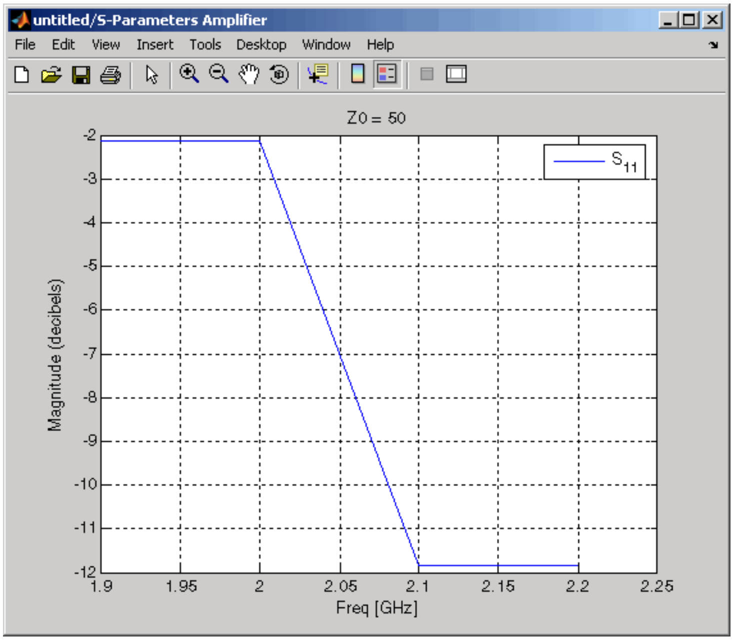
Plotting Parameters with the S-Parameters Amplifier Block

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. It 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])
```



The plot parameters in the dialog box request an X-Y Plane plot of the S11 parameters in the frequency range 1.8 to 2.3 GHz.



See Also

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

`interp1` (MATLAB)

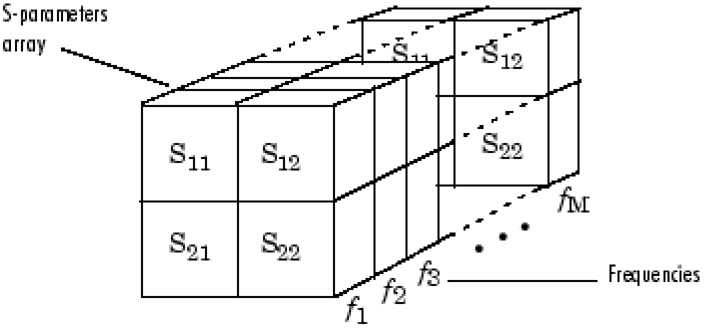
Purpose Model mixer using its S-parameters

Library Mixer sublibrary of the Physical library

Description

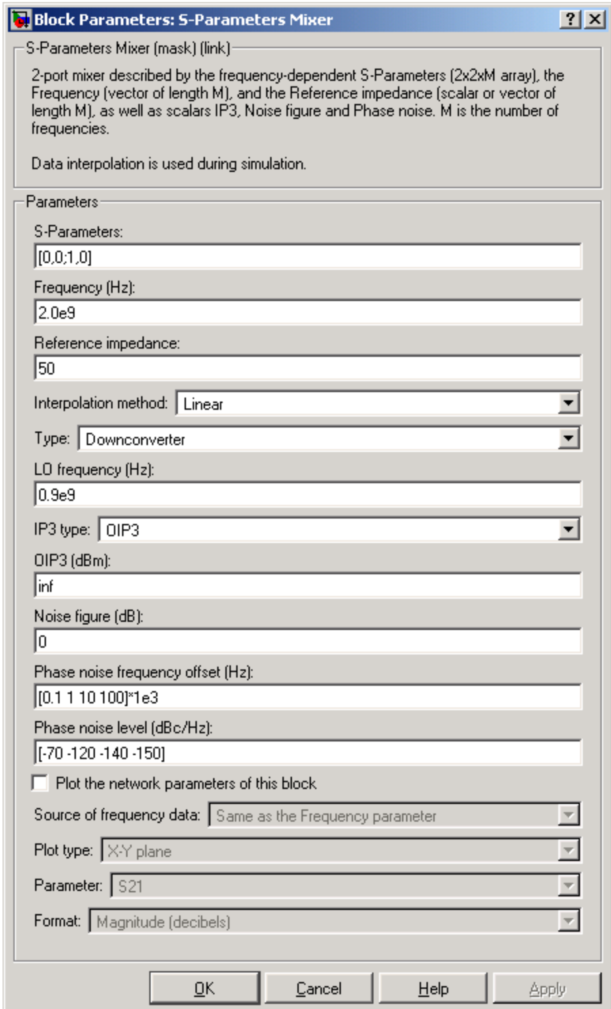
The S-Parameters Mixer block models the nonlinear mixer described in the block dialog box, in terms of its frequency-dependent S-parameters, their frequencies, and the reference impedance of the S-parameters. The block also takes phase noise into account.

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 Mixer block interpolates the given S-parameters to determine the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Dialog Box



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.

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, that value is applied to all frequencies.

Interpolation method
Method used to interpolate the given S-parameters over the range of frequencies. Interpolation can be `Linear` (default), `Spline`, or `Cubic`.

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

LO frequency (Hz)
Local oscillator frequency. If you choose `Downconverter`, $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

IP3 type
Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IIP3 (dBm)
Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type.

OIP3 (dBm)
Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type.

Noise figure (dB)
Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output $(S_i/N_i)/(S_o/N_o)$.

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

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

Note For information about plotting the network parameters, see Chapter 4, "Plotting Network Parameters".

See Also

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

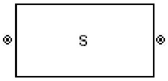
Purpose

Model passive network using its S-parameters

Library

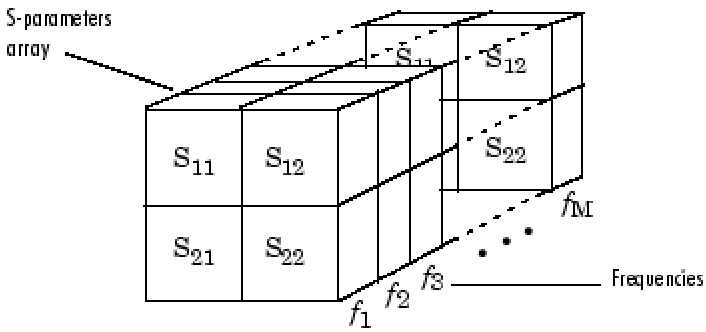
Black Box Elements sublibrary of the Physical library

Description



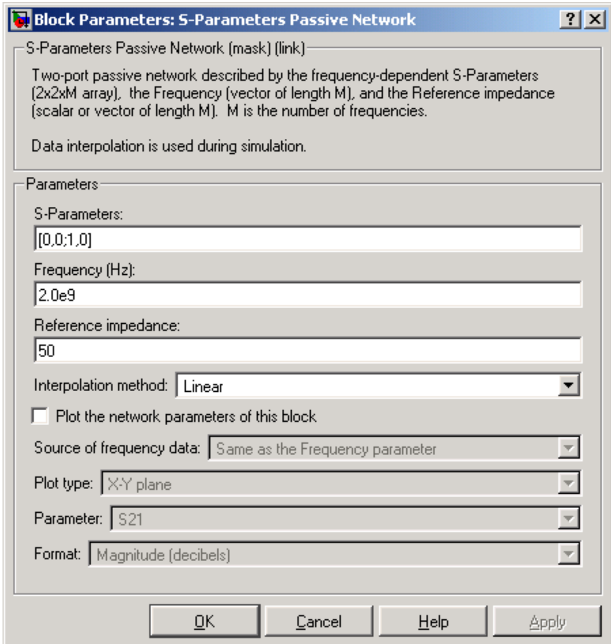
The S-Parameters Passive Network block models the two-port passive network described in the block dialog box, in terms of its S-parameters, their frequencies, and the reference impedance of the S-parameters.

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 the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See “S-Parameters at Simulation Frequencies” on page 3-2 for more details.

Dialog Box



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, that value is applied to all frequencies.

Interpolation method

Method used to interpolate the given S-parameters over the range of frequencies. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

Examples

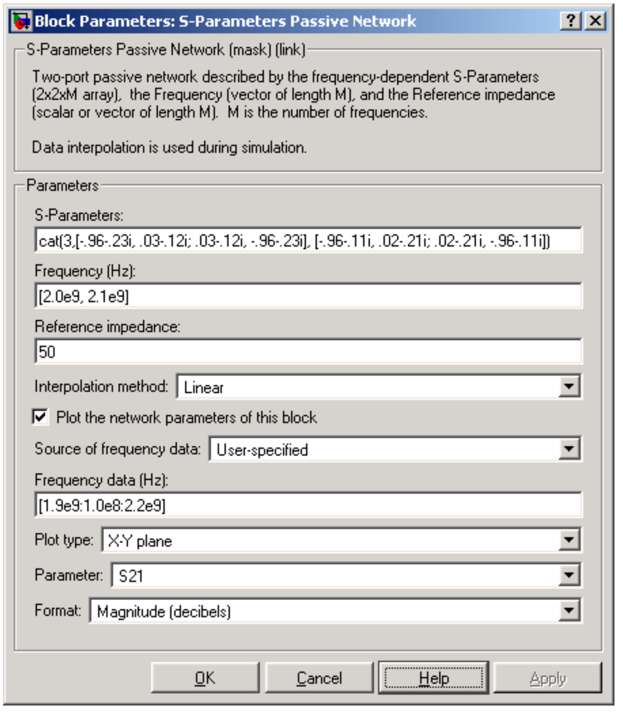
Plotting Parameters with the S-Parameters Passive Network Block

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. It uses the MATLAB `cat` function to create the 2-by-2-by-2 S-parameters array.

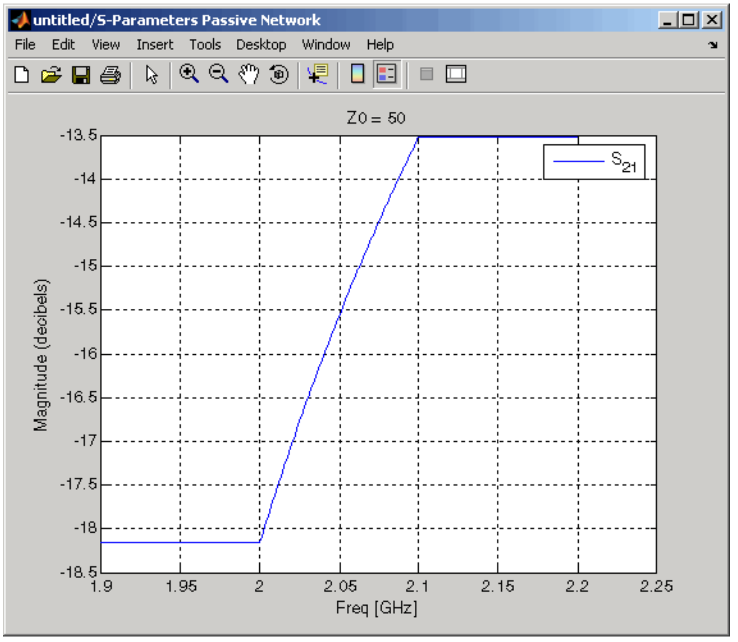
```
cat(3,[-.96-.23i, .03-.12i; .03-.12i, -.96-.23i],...
      [-.96-.11i, .02-.21i; .02-.21i, -.96-.11i])
```

You could also use the MATLAB `reshape` function. The following statement produces the same result as the one shown above.

```
reshape([-.96-.23i;.03-.12i;.03-.12i;-.96-.23i;...
         -.96-.11i;.02-.21i;.02-.21i;-.96-.11i],2,2,2)
```



The plot parameters in the dialog box request an X-Y Plane plot of the S21 S-parameter magnitudes, 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

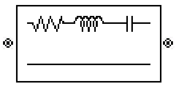
Purpose

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.

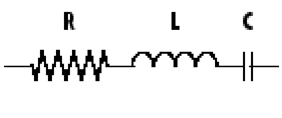
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference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

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

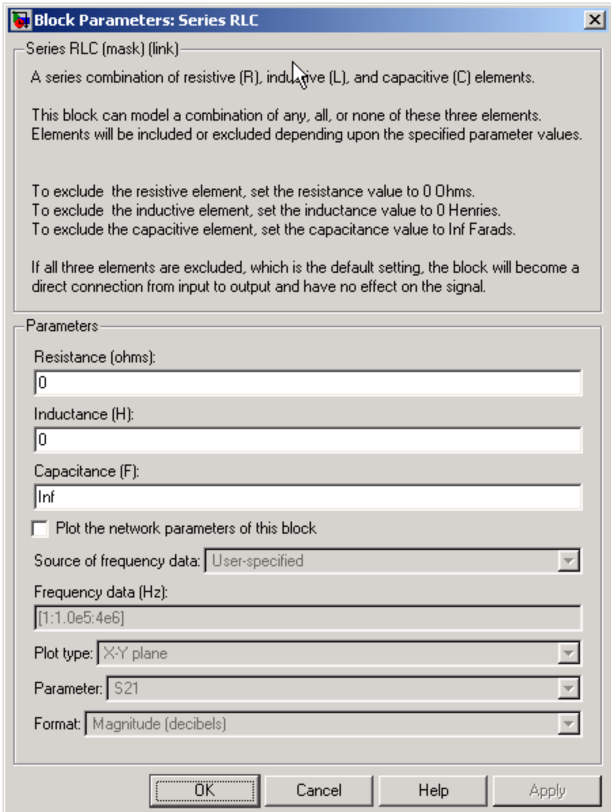
$$Z = \frac{-LC\omega^2 + jRC\omega + 1}{jC\omega}$$

and $\omega = 2\pi f$.

The series RLC object is a two-port network as shown in the circuit diagram below.



Dialog Box



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.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

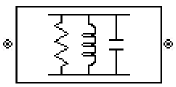
Purpose

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.

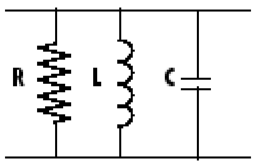
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of simulation frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference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

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

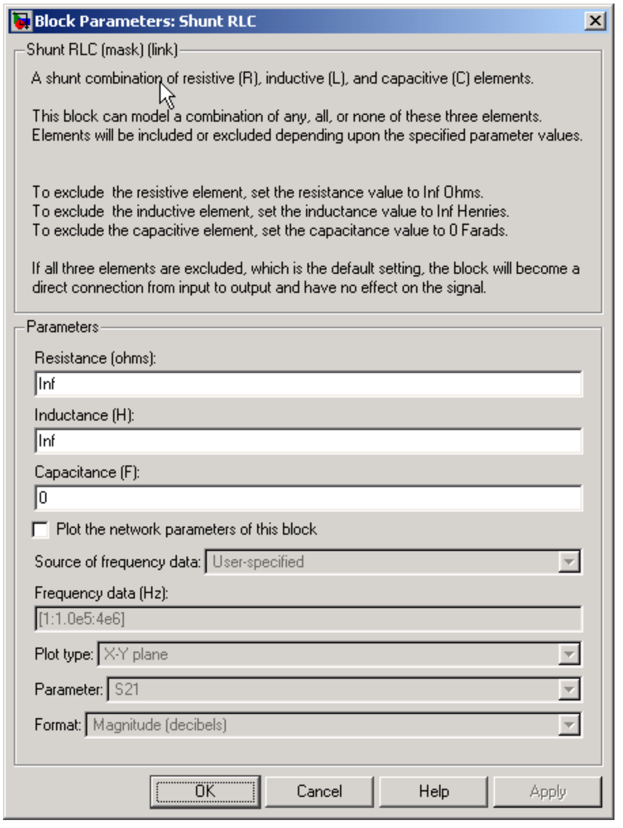
$$Y = \frac{-LC\omega^2 + j(L/R)\omega + 1}{jL\omega}$$

and $\omega = 2\omega f$.

The shunt RLC object is a two-port network as shown in the circuit diagram below.



Dialog Box



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.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

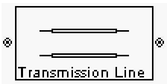
Purpose

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 α_a is the attenuation coefficient and β is the wave number. The attenuation coefficient α_a is related to the loss, α, by

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

and the wave number β is related to the phase velocity, V_P, by

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

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

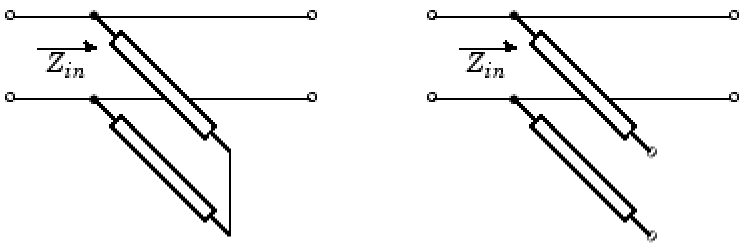
The Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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.

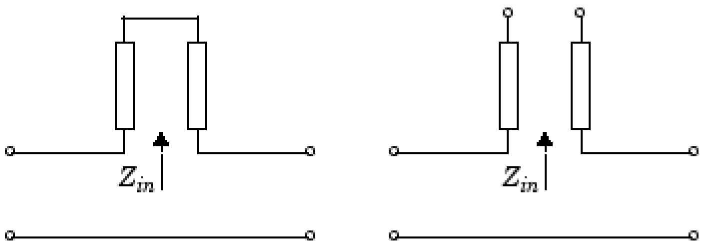


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

Series ABCD-Parameters

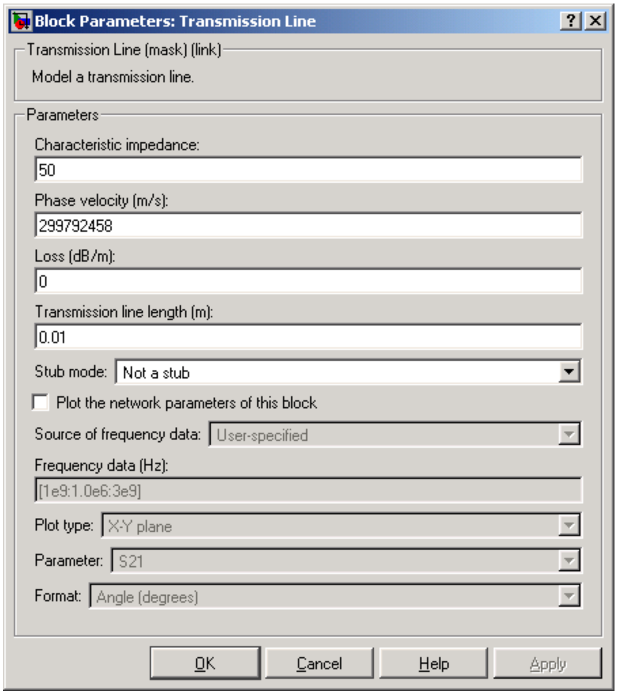
When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.



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



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`.

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

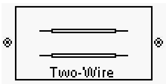
Purpose

Model two-wire transmission line

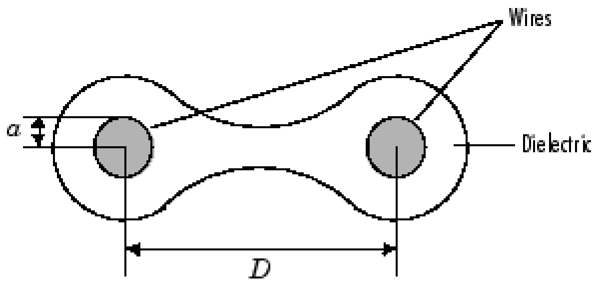
Library

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 parallel-plate transmission line is shown here in cross-section. Its physical characteristics include the radius of the wires a, and the distance between the wire centers D.



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

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

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

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

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

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

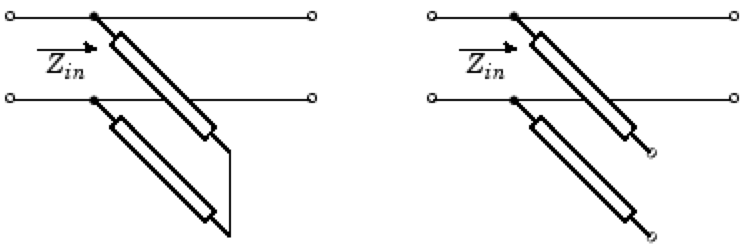
The Two-Wire Transmission Line block normalizes the resulting S-parameters to a reference impedance of 50 ohms.

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 simulation 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

$A = 1$

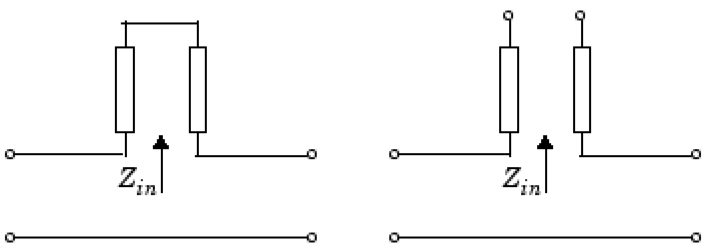
$B = 0$

$C = 1/Z_{in}$

$D = 1$

Series ABCD-Parameters

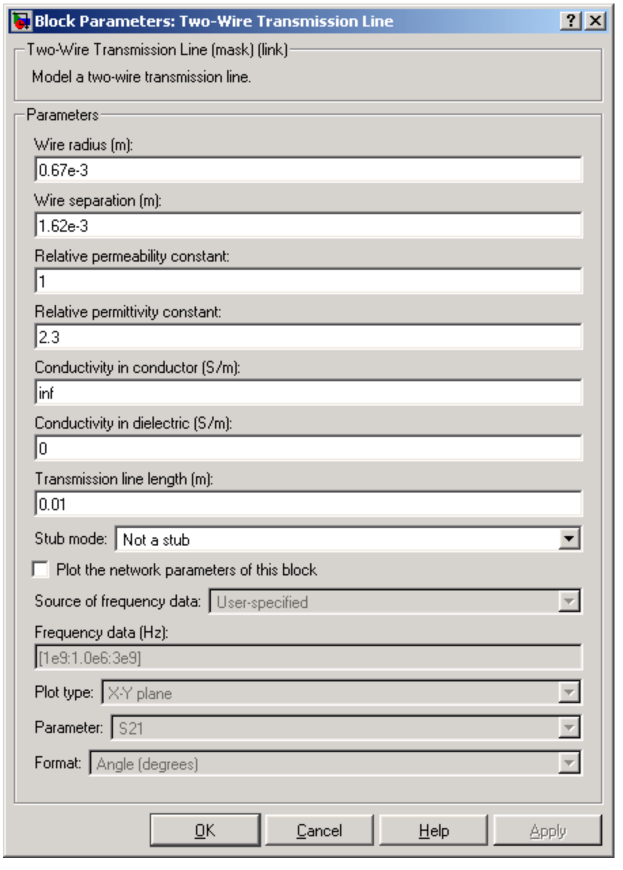
When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network consists of a series transmission line that you can terminate with either a short circuit or an open circuit as shown here.



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



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 μ_0.

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

Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

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

Y-Parameters Amplifier

Purpose

Model nonlinear amplifier using its Y-parameters

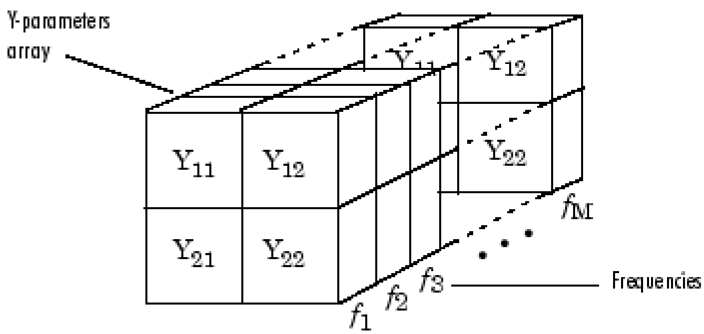
Library

Amplifiers sublibrary of the Physical library

Description

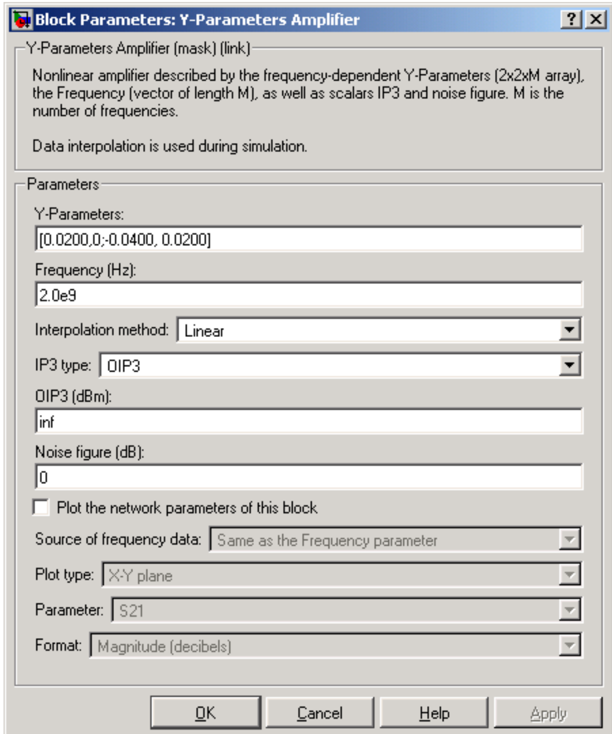
The Y-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent Y-parameters and their frequencies. The block also takes into account the IP3 value and the noise figure.

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 the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See “S-Parameters at Simulation Frequencies” on page 3-2 for more details.

Dialog Box



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

Method used to interpolate the S-parameters, as derived from the Y-parameters, over the range of frequencies. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

IP3 type
: Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IIP3 (dBm)
: Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type.

OIP3 (dBm)
: Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type.

Noise figure (dB)
: Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$.

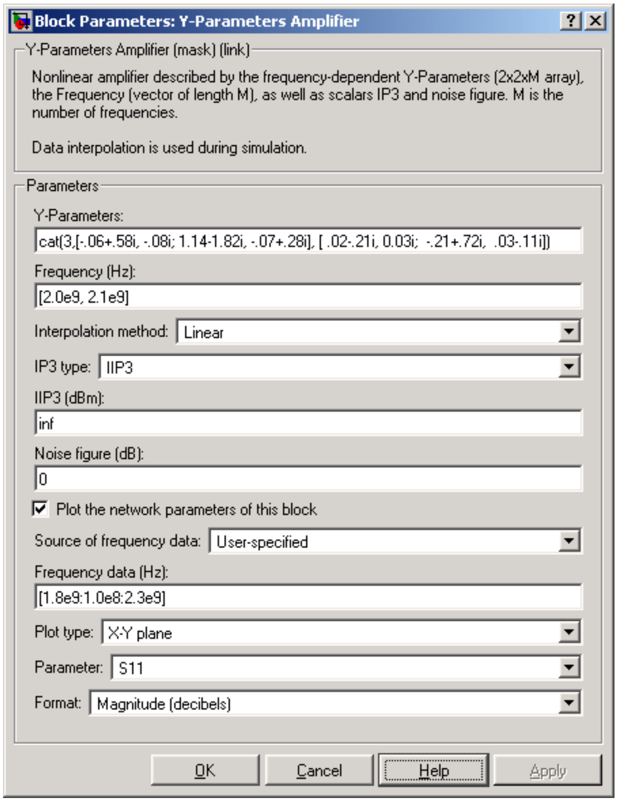
Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

Examples

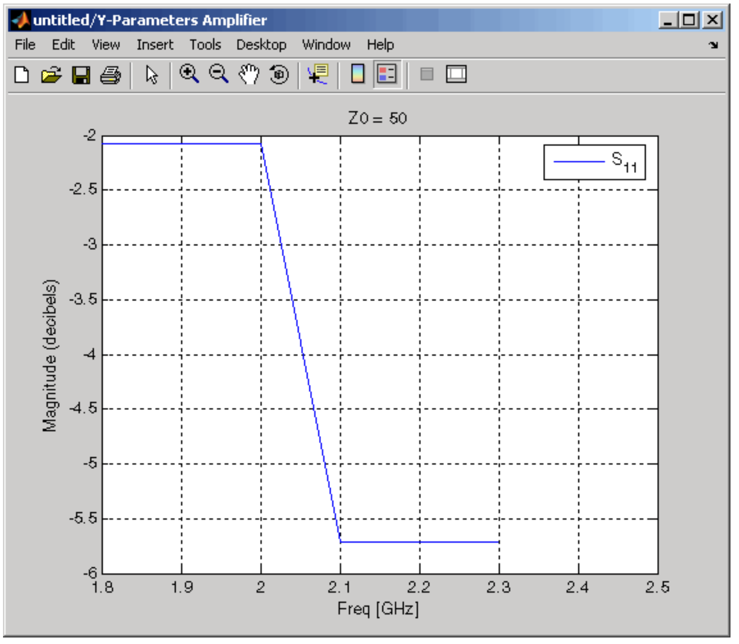
Plotting Parameters with the Y-Parameters Amplifier Block

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])
```



The plot parameters in the dialog box request an X-Y plane plot of the S11 parameters in the frequency range 1.8 to 2.3 GHz.



See Also

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

`y2s` (RF Toolbox)

`interp1` (MATLAB)

Purpose

Model mixer using its Y-parameters

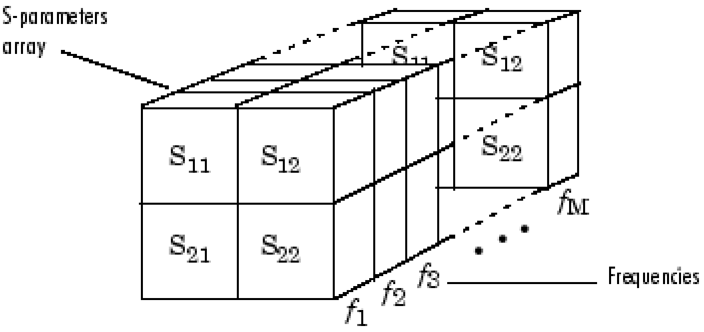
Library

Mixer sublibrary of the Physical library

Description

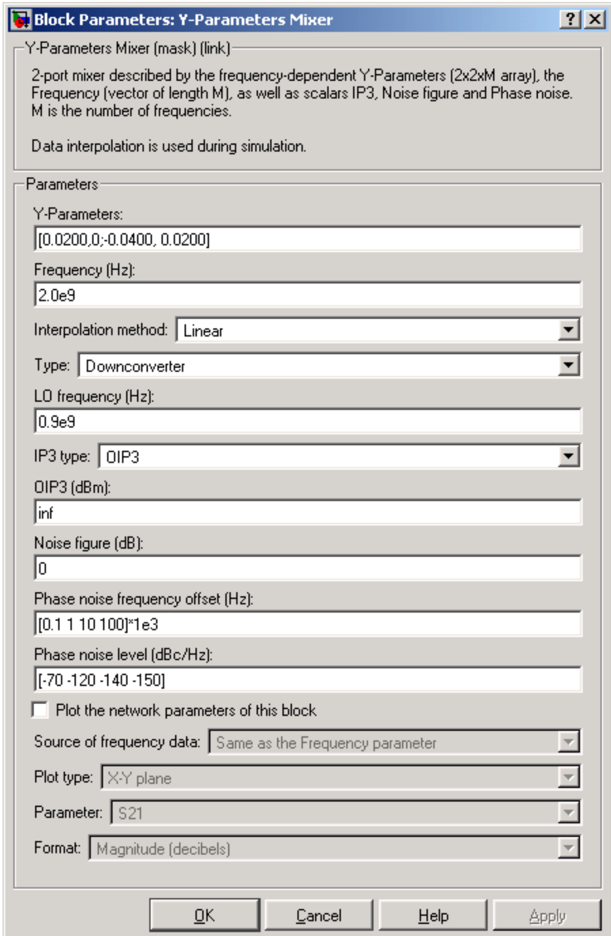
The Y-Parameters Mixer block models the nonlinear mixer described in the block dialog box, in terms of its frequency-dependent Y-parameters, their frequencies, and the reference impedance of the Y-parameters. The block also takes phase noise into account.

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 S-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 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 the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Dialog Box



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.

Interpolation method
Method used to interpolate the S-parameters, as derived from the Y-parameters, over the range of frequencies. Interpolation can be `Linear` (default), `Spline`, or `Cubic`.

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

LO frequency (Hz)
Local oscillator frequency. If you choose `Downconverter`, $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

IP3 type
Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IIP3 (dBm)
Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type.

OIP3 (dBm)
Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type.

Noise figure (dB)
Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$.

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

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

Note For information about plotting the network parameters, see Chapter 4, "Plotting Network Parameters".

See Also

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

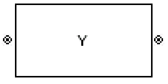
Purpose

Model passive network using its Y-parameters

Library

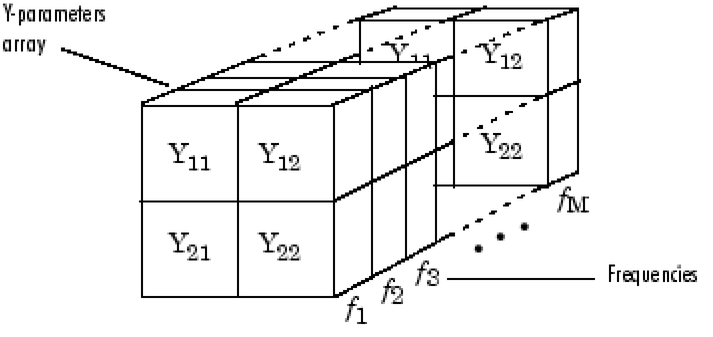
Black Box Elements sublibrary of the Physical library

Description



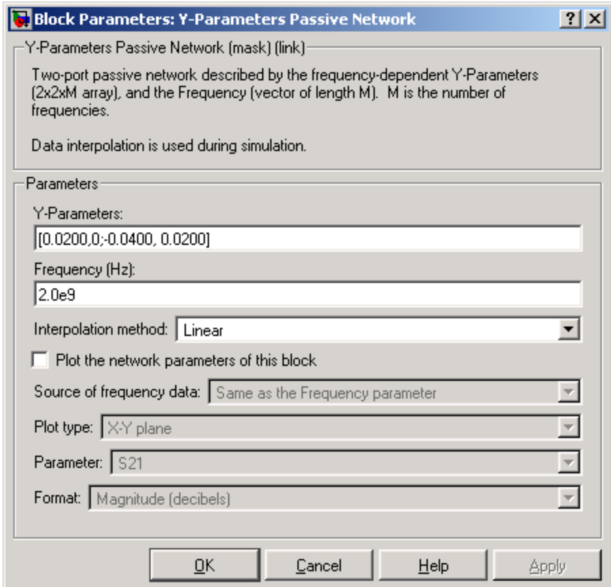
The Y-Parameters Passive Network block models the two-port passive network described in the block dialog box, in terms of its Y-parameters and their associated frequencies.

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 the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See “S-Parameters at Simulation Frequencies” on page 3-2 for more details.

Dialog Box



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
: Method used to interpolate the S-parameters, as derived from the Y-parameters, over the range of frequencies. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

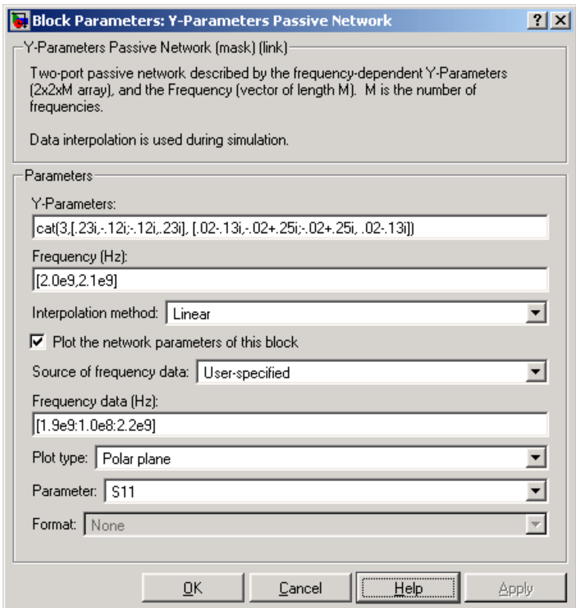
Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

Examples

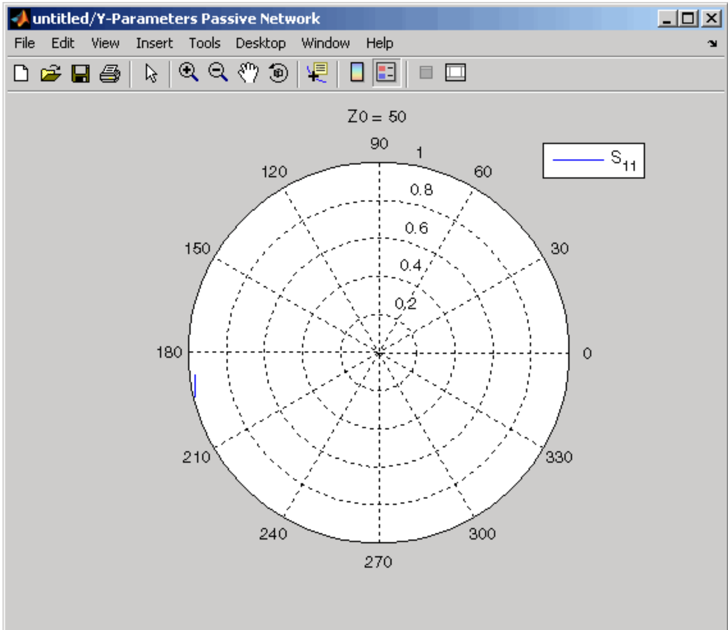
Plotting Parameters with the Y-Parameters Passive Network Block

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])
```



The plot parameters in the dialog box request a polar plane plot of the S11 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)

Purpose

Model nonlinear amplifier using its Z-parameters

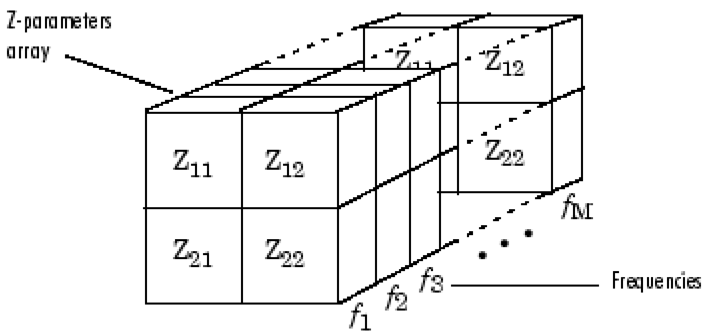
Library

Amplifiers sublibrary of the Physical library

Description

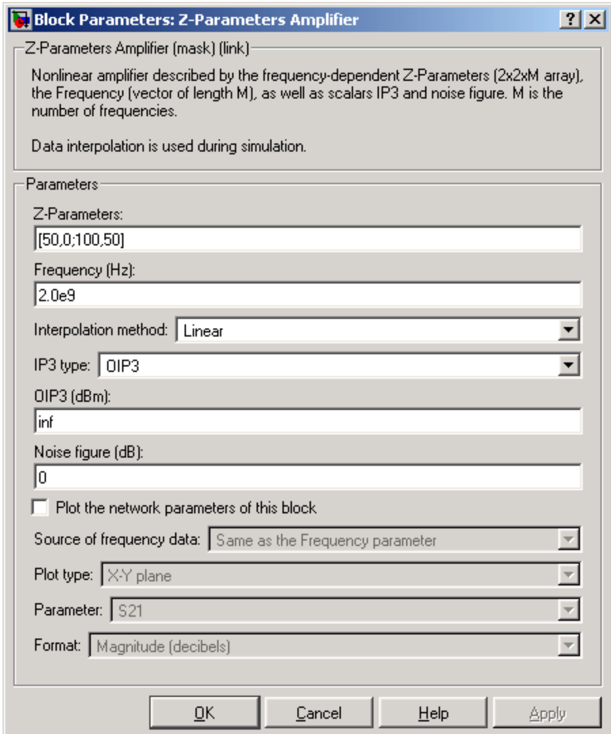
The Z-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent Z-parameters and their frequencies. The block also takes into account the IP3 value and the noise figure.

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 `y2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Dialog Box



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

Method used to interpolate the S-parameters, as derived from the Z-parameters, over the range of frequencies. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

IP3 type

Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IIP3 (dBm)

Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type.

OIP3 (dBm)

Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type.

Noise figure (dB)

Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$.

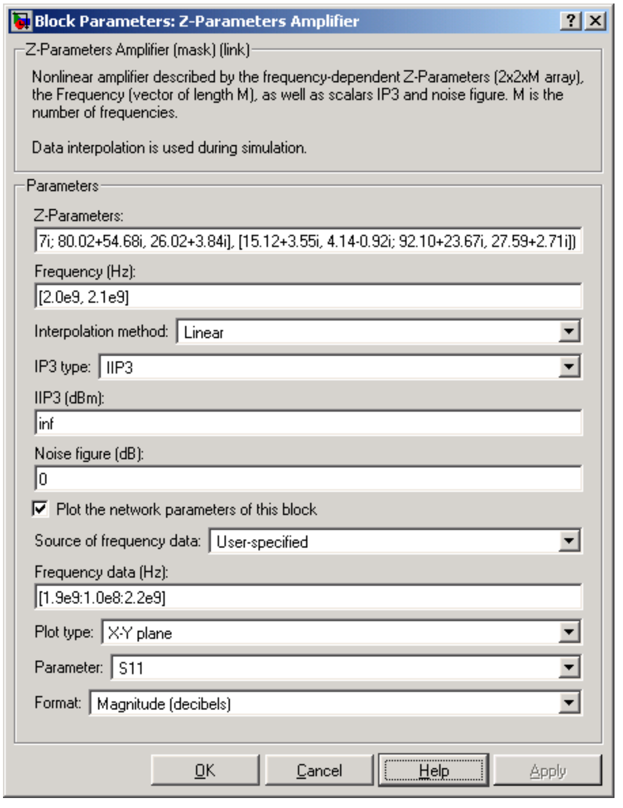
Note For information about plotting, see Chapter 4, "Plotting Network Parameters".

Examples

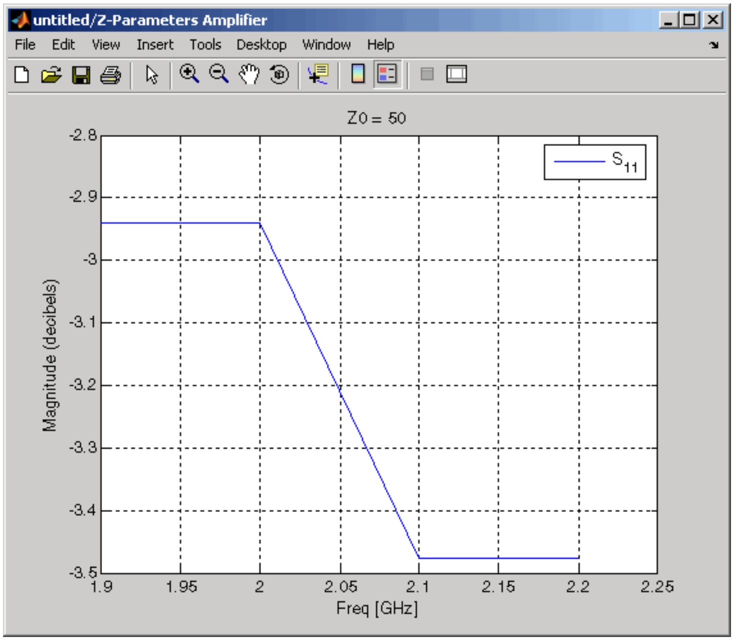
Plotting Parameters with the Z-Parameters Amplifier Block

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])
```



The plot parameters in the dialog box request an X-Y plane plot of the S11 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)

Z-Parameters Mixer

Purpose

Model mixer using its Z-parameters

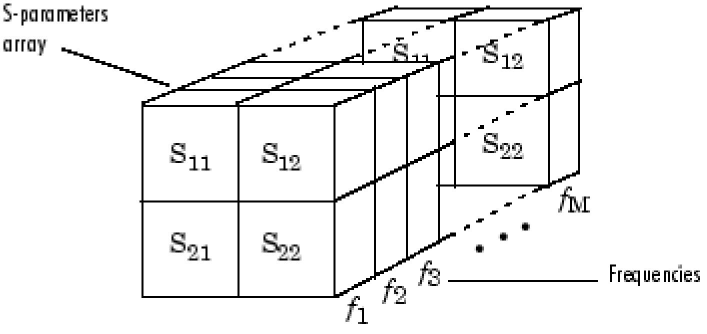
Library

Mixer sublibrary of the Physical library

Description

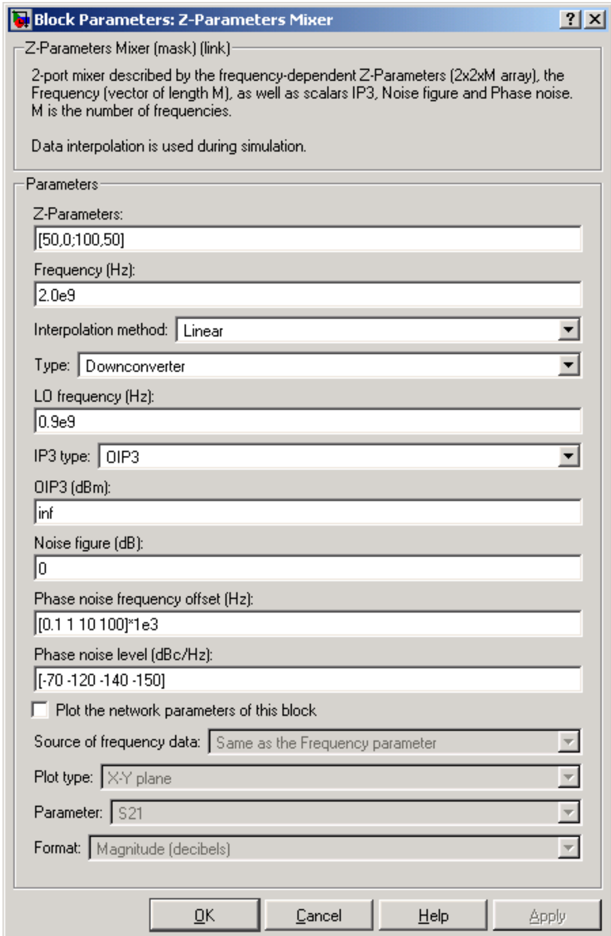
The Z-Parameters Mixer block models the nonlinear mixer described in the block dialog box, in terms of its frequency-dependent Z-parameters, their frequencies, and the reference impedance of the Z-parameters. The block also takes phase noise into account.

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 S-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 Mixer block uses the RF Toolbox `y2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Dialog Box



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.

Interpolation method
Method used to interpolate the S-parameters, as derived from the Z-parameters, over the range of frequencies. Interpolation can be `Linear` (default), `Spline`, or `Cubic`.

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

LO frequency (Hz)
Local oscillator frequency. If you choose `Downconverter`, $f_{out} = f_{in} - f_{lo}$. If you choose `Upconverter`, $f_{out} = f_{in} + f_{lo}$.

IP3 type
Type of third-order intercept point. The value can be `IIP3` (input intercept point) or `OIP3` (output intercept point).

IIP3 (dBm)
Input power intercept point as a scalar value. This field becomes visible if you select `IIP3` as the IP3 type.

OIP3 (dBm)
Output power intercept point as a scalar value. This field becomes visible if you select `OIP3` as the IP3 type.

Noise figure (dB)
Scalar ratio of the available signal-to-noise power ratio at the input to the available signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$.

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

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

Note For information about plotting the network parameters, see Chapter 4, “Plotting Network Parameters”.

See Also

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

Z-Parameters Passive Network

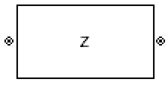
Purpose

Model passive network using its Z-parameters

Library

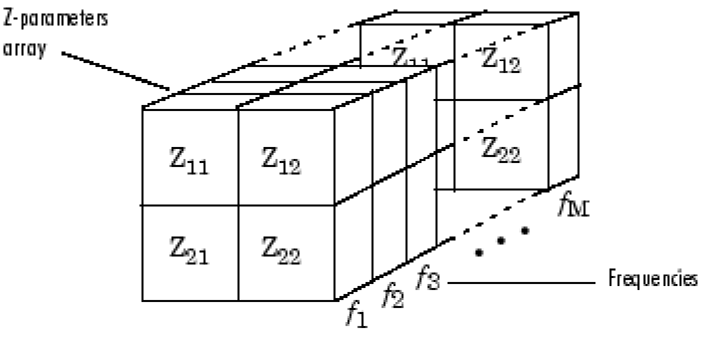
Black Box Elements sublibrary of the Physical library

Description



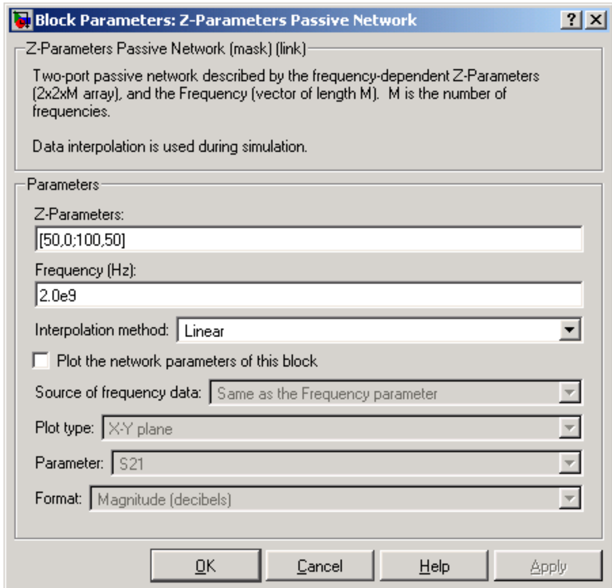
The Z-Parameters Passive Network block models the two-port passive network described in the block dialog box, in terms of its Z-parameters and their associated frequencies.

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 `y2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine the S-parameters at the simulation frequencies. The simulation frequencies are determined by the Output Port block. See "S-Parameters at Simulation Frequencies" on page 3-2 for more details.

Dialog Box



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

Method used to interpolate the S-parameters, as derived from the Z-parameters, over the range of frequencies. Interpolation can be `Cubic`, `Linear` (default), or `Spline`.

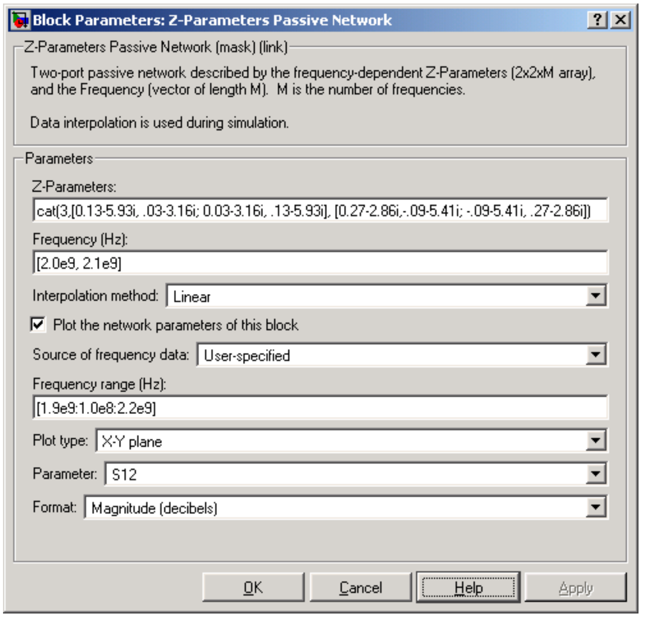
Note For information about plotting, see Chapter 4, “Plotting Network Parameters”.

Examples

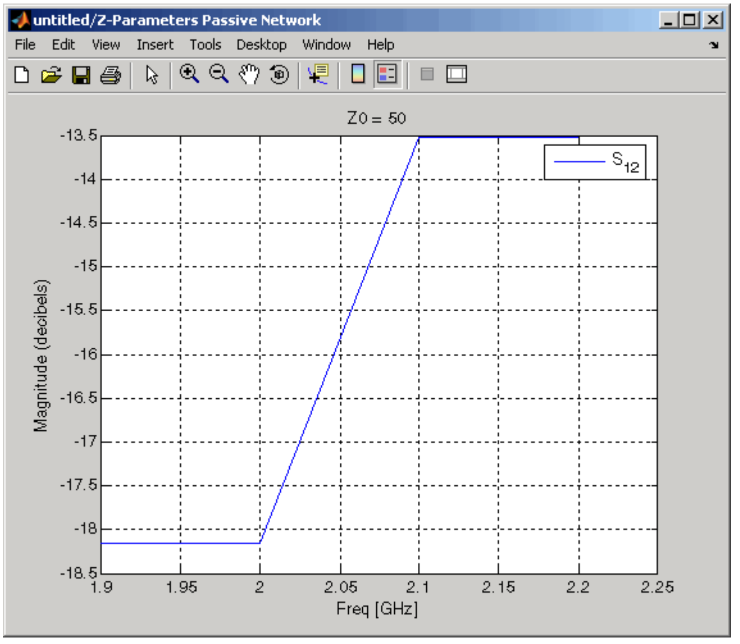
Plotting Parameters with the Z-Parameters Passive Network Block

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])
```



The plot parameters in the dialog box request an X-Y plane plot of the S12 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)

Examples

Use this list to find examples in the documentation.

Examples

"Modeling an LC Bandpass Filter" on page 1-9
"Using File Data" on page 3-4
"Importing Circuits from the MATLAB Workspace" on page 3-9

Index

A

active noise
  General Amplifier block 6-35
  General Mixer block 6-43
amplifier
  modeling from file data 6-37
  modeling from `rfdata` object 6-35
Amplifier block 6-2
  effects 6-5
  modeling nonlinearity 6-2
    methods 6-7
  sample time comparison 6-11
  thermal noise simulation 6-10
  See also General Amplifier block
amplifier blocks
  mathematical 5-3
  physical 5-8

B

Bandpass RF Filter block 6-16
  sample time comparison 6-17
Bandstop RF Filter block 6-20
  sample time comparison 6-21
baseband-equivalent time domain
  modeling 6-96
  linear subsystem 6-96
  nonlinear subsystem 6-100
Bessel filter design
  Bandpass RF Filter block 6-16
  Bandstop RF Filter block 6-20
  Highpass RF Filter block 6-50
  Lowpass RF Filter block 6-84
black box elements blocks 5-7
block parameters
  setting 2-6
blocks
  connecting 2-4
  mathematical 5-3
  physical 5-5
Butterworth filter design
  Bandpass RF Filter block 6-16
  Bandstop RF Filter block 6-20
  Highpass RF Filter block 6-50
  Lowpass RF Filter block 6-84

C

Chebyshev I filter design
  Bandpass RF Filter block 6-16
  Bandstop RF Filter block 6-20
  Highpass RF Filter block 6-50
  Lowpass RF Filter block 6-84
Chebyshev II filter design
  Bandpass RF Filter block 6-16
  Bandstop RF Filter block 6-20
  Highpass RF Filter block 6-50
  Lowpass RF Filter block 6-84
circuit element
  modeling from `rfckt` object 6-41
Coaxial Transmission Line block 6-24
  shunt and series stubs 6-25
  stubless 6-24
connecting blocks
  RF Physical blocks to Simulink 2-4
  Simulink to RF Physical blocks 2-4
connector blocks 5-10
  Input Port 6-54
  Output Port 6-96
Coplanar Waveguide Transmission Line
  block 6-29
  shunt and series stubs 6-30
  stubless 6-29

E

Elliptic filter design
  Bandpass RF Filter block 6-16
  Bandstop RF Filter block 6-20
  Highpass RF Filter block 6-50

Lowpass RF Filter block 6-84

F

filter blocks
mathematical 5-3
physical 5-5
filter design
physical blocks
LC Lowpass Tee block 6-81
filter design mathematical blocks
Bandpass RF Filter block 6-16
Bandstop RF Filter block 6-20
Highpass RF Filter block 6-50
Lowpass RF Filter block 6-84
filter design physical blocks
LC Bandpass Pi block 6-57
LC Bandpass Tee block 6-63
LC Bandstop Pi block 6-66
LC Bandstop Tee block 6-69
LC Highpass Pi block 6-72
LC Highpass Tee block 6-75
LC Lowpass Pi block 6-78
format
changing in plot 4-10
frequency range
changing in plot 4-10

G

General Amplifier block 6-35
active noise 6-35
generating from file data 6-37
nonlinearity 6-35
See also Amplifier block
General Circuit Element block 6-40
generating from `rfckt` object 6-41
General Mixer block 6-43
active noise 6-43
phase noise 6-43
See also Mixer block
General Passive Network block 6-47
generating from file data 6-48
Ghorbani model 6-9

H

Highpass RF Filter block 6-50
sample time comparison 6-51

I

Input Port block 6-54
parameter input for physical subsystem 6-54
sample time comparison 6-54
See also Output Port block
interpolation
network parameters 3-2

L

ladder filter blocks 5-5
ladder filters
example to filter Gaussian noise 6-59
LC Bandpass Pi block 6-57
example to filter Gaussian noise 6-59
LC Bandpass Tee block 6-63
LC Bandstop Pi block 6-66
LC Bandstop Tee block 6-69
LC Highpass Pi block 6-72
LC Highpass Tee block 6-75
LC Lowpass Pi block 6-78
LC Lowpass Tee block 6-81
Lowpass RF Filter block 6-84
sample time comparison 6-85

M

mathematical and physical models
connecting 2-4

mathematical modeling blocks 2-3
Microstrip Transmission Line block 6-88
  shunt and series stubs 6-89
  stubless 6-88
Mixer block 6-93
  sample time comparison 6-94
  See also General Mixer block
mixer blocks
  mathematical 5-3
  physical 5-9
modeling
  amplifier from file data 6-37
  amplifier from `rfdata` object 6-35
  baseband-equivalent time domain 6-96
  circuit element from `rfckt` object 6-41
  passive network from file data 6-48
models
  physical and mathematical blocks 2-3

N

network parameters
  adding to a plot 4-10
  interpolation at simulation frequencies 3-2
noise
  active
    General Amplifier block 6-35
    General Mixer block 6-43
  phase
    General Mixer block 6-43
nonlinearity
  Amplifier block 6-2
  General Amplifier block 6-35

O

Output Port block 6-96
  baseband-equivalent time domain
    modeling 6-96
  plotting RF subsystem network
    parameters 6-103
  See also Input Port block

P

Parallel-Plate Transmission Line block 6-104
  shunt and series stubs 6-105
  stubless 6-104
parameter input
  to physical subsystem 6-54
passive network
  modeling from file data 6-48
phase noise
  General Mixer block 6-43
physical and mathematical models
  connecting blocks 2-4
physical modeling blocks 2-3
physical subsystem
  parameter input for 6-54
plot types
  modifying 4-10
plots
  changing format of 4-10
  changing frequency range of 4-10
  displaying legend 4-12
  examples 4-2
  modifying 4-10
  RF subsystem network parameters 6-103
  types of 4-2
  using block parameters to specify 4-2

R

Rapp model 6-10
RF block libraries
  Mathematical 5-3
  Physical 5-5
RF models
  connecting blocks 2-4

description 2-2
`rfckt` object
modeling general circuit element 6-40
`rfdata` object
modeling nonlinear amplifier 6-35
RLCG Transmission Line block 6-109
shunt and series stubs 6-110
stubless 6-109

S

S-Parameters Amplifier block 6-114
S-Parameters Mixer block 6-119
S-Parameters Passive Network block 6-123
Saleh model 6-8
Series RLC block 6-128
setting
block parameters 2-6
Shunt RLC block 6-131
simulation frequencies
calculating 3-2
S-parameter interpolation 3-2
Smith chart
example 4-5
stubless transmission lines
Coaxial Transmission Line block 6-24
Coplanar Waveguide Transmission Line block 6-29
Microstrip Transmission Line block 6-88
Parallel-Plate Transmission Line block 6-104
RLCG Transmission Line block 6-109
Transmission Line block 6-134
Two-Wire Transmission Line block 6-139
stubs (shunt and series)
Coaxial Transmission Line block 6-25
Coplanar Waveguide Transmission Line block 6-30
Microstrip Transmission Line block 6-89
Parallel-Plate Transmission Line block 6-105
RLCG Transmission Line block 6-110
Transmission Line block 6-135
Two-Wire Transmission Line block 6-140

T

time-domain modeling
baseband-equivalent 6-96
Transmission Line block 6-134
shunt and series stubs 6-135
stubless 6-134
transmission line blocks 5-6
Two-Wire Transmission Line block 6-139
shunt and series stubs 6-140
stubless 6-139

Y

Y-Parameters Amplifier block 6-144
Y-Parameters Mixer block 6-149
Y-Parameters Passive Network block 6-153

Z

Z-Parameters Amplifier block 6-157
Z-Parameters Mixer block 6-162
Z-Parameters Passive Network block 6-166