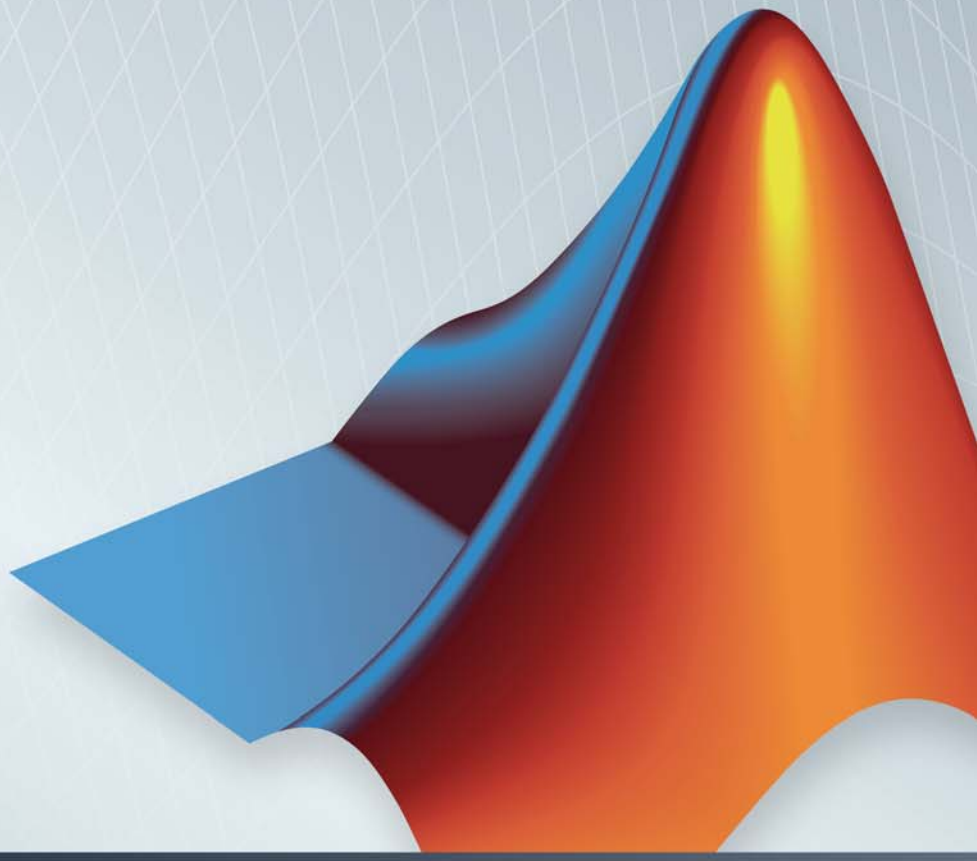
SimRF™

Reference

R2013a



MATLAB® & SIMULINK®

MathWorks®

How to Contact 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.

SimRF™ Reference

© COPYRIGHT 2010–2013 by The MathWorks, Inc.

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

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

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

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

Revision History

September 2010	Online only	New for Version 3.0 (Release 2010b)
April 2011	Online only	New for Version 3.0.2 (Release 2011a)
September 2011	Online only	New for Version 3.1 (Release 2011b)
March 2012	Online only	New for Version 3.2 (Release 2012a)
September 2012	Online only	New for Version 3.3 (Release 2012b)
March 2013	Online only	New for Version 4.0 (Release 2013a)

Contents

1 Blocks — Alphabetical List

2 Blocks — Alphabetical List

Index

1

Blocks — Alphabetical List

Amplifier

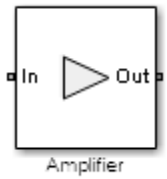
Purpose

Model amplifier in RF systems

Library

Elements

Description



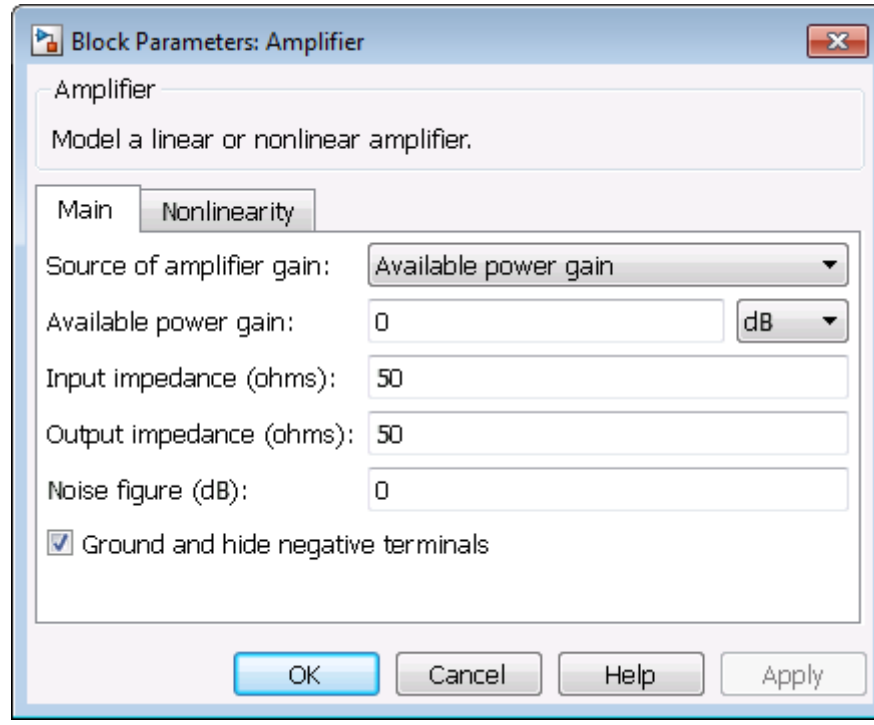
The Amplifier block models a linear or nonlinear amplifier, with or without noise. When the amplifier gain is described using a data source, this specification also includes input data visualization and modeling. The **Main** tab parameters specify amplifier gain and noise using data sheet values, standard `s2p` files, S-parameters or circuit envelope polynomial coefficients.

The amplifier is implemented as a polynomial, voltage-controlled voltage source (VCVS). The VCVS includes nonlinearities that are described using parameters listed in the **Nonlinearity** tab. To model linear amplification, the amplifier implements the relation $V_{out} = a_1 * V_{in}$ between the input and output voltages. The input voltage is $V_i(t) = A_i(t)e^{j\omega t}$, and the output voltage is $V_o(t) = A_o(t)e^{j\omega t}$ at each carrier $w = 2\pi f$ in the SimRF™ environment.

Nonlinear amplification is modeled as a polynomial (with the saturation power computed automatically). It is also produces additional intermodulation frequencies.

Dialog Box and Parameters

Main Tab



Source of amplifier gain

Specify the source parameter of the amplifier gain as:

- `Available power gain` — The block uses the value of the **Available power gain** parameter to calculate the linear voltage gain term of the polynomial VCVS, a_1. This calculation assumes a matched load termination for the amplifier.
- `Open circuit voltage gain` — The block uses the value of the **Open circuit voltage gain** parameter as the linear voltage gain term of the polynomial VCVS, a_1.

- `Data source` — The block calculates the linear voltage gain term of the polynomial VCVS from the specified network parameters:

  $$a_1 = \left| \frac{s_{21}}{1 + s_{11}} \right|$$

  for the maximal value of S_{21}.

  When using the data source option, S_{11} and S_{22} are used as the input and output impedances. Network parameter values are specified using either `Data file` or `S-parameters`, depending on the value of `Data source`.

- `Polynomial coefficients` — The block implements a nonlinear voltage gain according to the polynomial you specify. The order of the polynomial must be less than or equal to 9. The coefficients are ordered in ascending powers. If a vector has 10 coefficients, [a_0,a_1,a_2, ... a_9], the polynomial it represents is:

  $$V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^{\ 2} + \ldots + a_9 V_{in}^{\ 9}$$

  where a_1 represents the linear gain term, and higher-order terms are modeled according to [2].

  For example, the vector [a_0,a_1,a_2,a_3] specifies the relation $V_o = a_0 + a_1 V_I + a_2 V_I^{\ 2} + a_3 V_I^{\ 3}$. Trailing zeroes are omitted. If $a_3 = 0$, then [a_0,a_1,a_2] defines the same polynomial as [a_0,a_1,a_2,0]. The default value of this parameter is [0,1], corresponding to the linear relation $V_o = V_i$.

  The default value of this parameter is `Available power gain`.

Available power gain

When you set the **Source of amplifier gain** parameter to `Available power gain`, you can specify the available power gain of the amplifier. Specify the units from the corresponding drop-down list.

The default value of this parameter is `0 dB`.

Open circuit voltage gain

When you set the **Source of amplifier gain** to `Open circuit voltage gain`, you can specify the open circuit voltage gain of the amplifier. Specify the units from the corresponding drop-down list.

The default value of this parameter is `0 dB`.

Input impedance (ohms)

When you set the **Source of amplifier gain** to `Available power gain`, `Open circuit voltage gain`, or `Polynomial coefficients`, you can specify the scalar input impedance of the amplifier.

The default value of this parameter is `50` ohms.

Output impedance (ohms)

When you set the **Source of amplifier gain** to `Available power gain`, `Open circuit voltage gain`, or `Polynomial coefficients`, you can specify the scalar output impedance of the amplifier.

The default value of this parameter is `50` ohms.

Data source

When you set **Source of amplifier gain** to `Data source`, you can specify the data source as either `Data file` or `S-parameters`.

- `Data file` — Specify an `.s2p` file for the **Data file** parameter. The block ignores noise and nonlinearity data in imported files.
- `S-parameters` — Provide **S-parameters**, **Frequency**, and **Reference impedance (ohms)** data for the amplifier.

Noise figure (dB)

Specify the noise figure of the amplifier. The default value of this parameter is `0` dB, which implies that no noise is added to the system by this block.

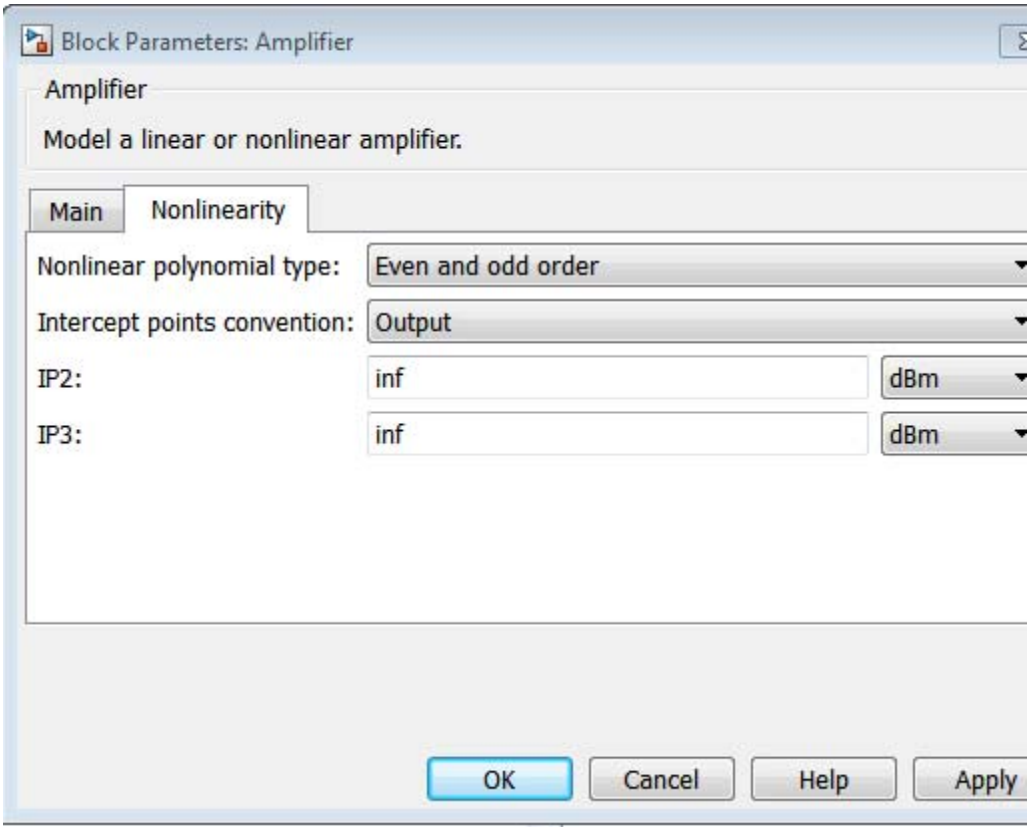
You can model noise in a SimRF model with a Noise, Resistor, Amplifier, or Mixer block. To do so, in the Configuration block dialog box, verify that the **Simulate noise** check box is selected (default).

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Nonlinearity Tab



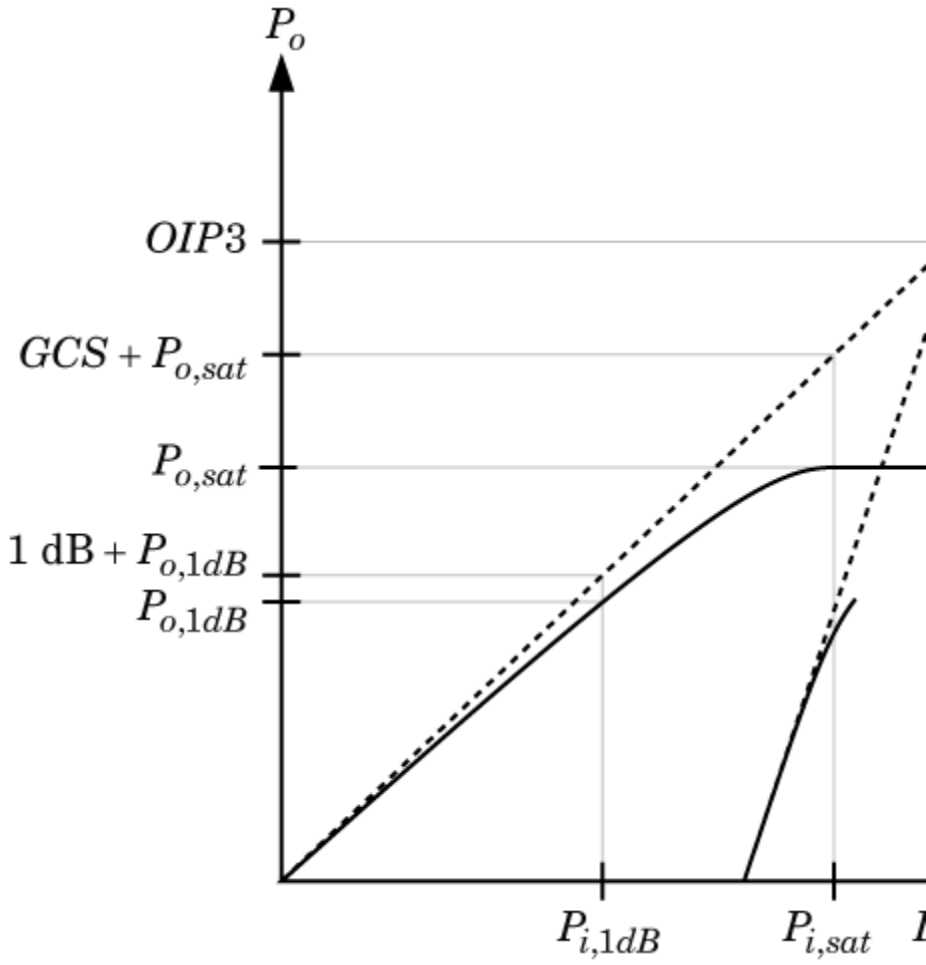
Nonlinear polynomial type

Specify either an `Even and odd order` or `Odd order` polynomial nonlinearity. The default value is `Even and odd order`.

- When you select `Even and odd order`, the amplifier can produce second- and third-order intermodulation frequencies in addition to a linear term.

- When you select `Odd order`, the amplifier generates only odd order intermodulation frequencies.

  The linear gain determines the linear a_1 term. The block calculates the remaining terms from the specified parameters. These parameters are **IP3**, **1-dB gain compression power**, **Output saturation power**, and **Gain compression at saturation**. The number of constraints you specify determines the order of the model.



The preceding figure shows the graphical definition of the nonlinear amplifier parameters.

Intercept points convention

Specify either an `Input`-referred or `Output`-referred convention. Use this specification for the intercept points, 1-dB gain compression power, and saturation power.

The default value is `Output`.

IP2

When **Nonlinear polynomial type** is `Even and odd order`, specify the second-order intercept point of the amplifier.

The default value is `inf dBm`, which corresponds to an unspecified point.

IP3

Specify the third-order intercept point of the amplifier. The default value is `inf dBm`, which corresponds to an unspecified point.

1-dB gain compression power

When **Nonlinear polynomial type** is `Odd order`, specify the 1-dB gain compression point. The 1-dB gain compression point must be less than the output saturation power.

The default value is `inf dBm`, which corresponds to an unspecified point.

Output saturation power

When **Nonlinear polynomial type** is `Odd order`, specify the output saturation power. The block uses this value to calculate the voltage saturation point used in the nonlinear model. In this case, the first derivative of the polynomial is zero, and the second derivative is negative.

The default value is `inf dBm`, which corresponds to an unspecified point in the polynomial model.

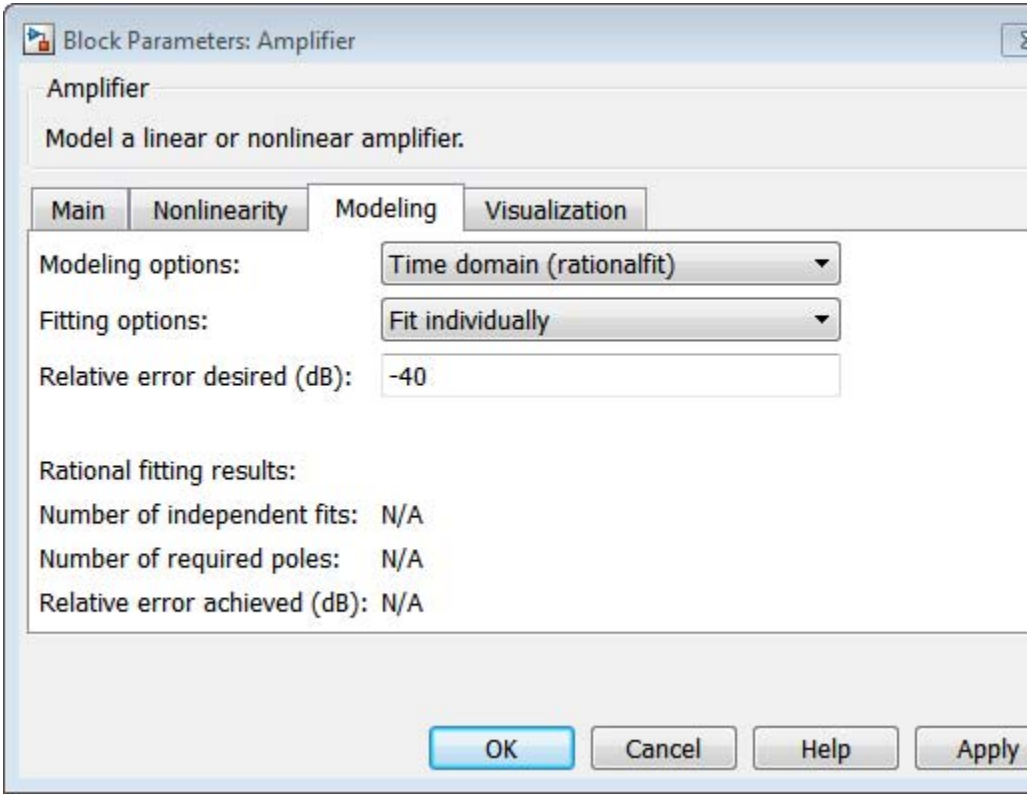
Gain compression at saturation

When **Nonlinear polynomial type** is `Odd order`, specify the gain compression at saturation. This parameter cannot be set unless **Output saturation power** is specified.

The default value is `inf dBm`.

Modeling Tab

Setting **Source of amplifier gain** to `Data source` activates the **Modeling** Tab.



Modeling options

Modeling domain. Select `Frequency domain` or `Time domain (rationalfit)` from the drop-down list.

For the Amplifier and S-parameters blocks, the default value is `Time domain (rationalfit)`. For the Transmission Line block, the default value is `Frequency domain`.

Fitting options

This field displays when you select `Time domain (rationalfit)` as the **Modeling options**. Valid values are `Share all poles`, `Share poles by columns`, and `Fit individually`.

For the Amplifier block, the default value is `Fit individually`. For the S-parameters block, the default value is `Share all poles`.

Relative error desired (dB)

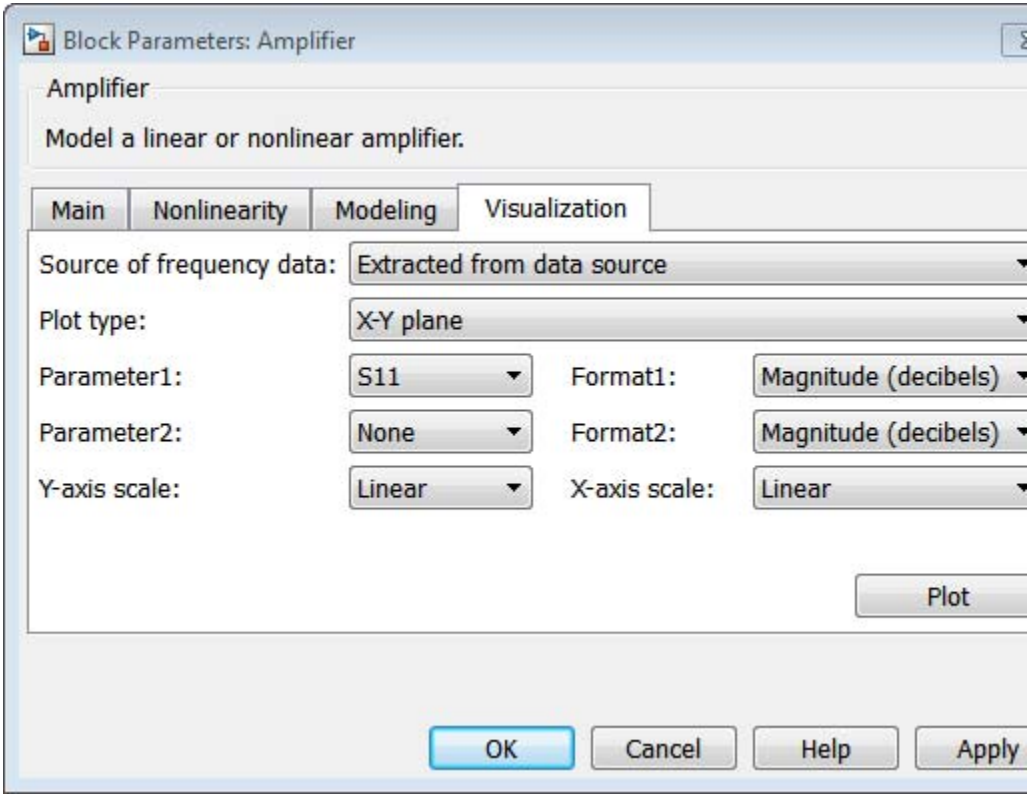
This field displays when you select `Time domain (rationalfit)` as the **Modeling options**. Enter the desired relative error in decibels (dB). The default value is `-40`.

Rational fitting results

These fields display the results of rational fitting when you select `Time domain (rationalfit)` as the **Modeling options**.

Visualization Tab

Setting **Source of amplifier gain** to `Data source` activates the Visualization tab.



Source of frequency data

Frequency data source. When **Source of frequency data** is `Extracted from data source`, the **Data source** must be set to `Data file`. Verify that the specified **Data file** contains frequency data.

When **Source of frequency data** is `User-specified`, specify a vector of frequencies in the **Frequency data** parameter. Also, specify units from the corresponding drop-down list.

For the Amplifier and S-parameters blocks, the default value is `Extracted from source data`. For the Transmission Line block, the default value is `User-specified`.

Plot type

Specify the type of plot that you want to produce with your data. The **Plot type** parameter provides the following options:

- `X-Y plane` — Generate a Cartesian plot of your data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- `Polar plane` — Generate a polar plot of your data. The block plots only the range of data corresponding to the specified frequencies.
- `Z smith chart`, `Y smith chart`, and `ZY smith chart` — Generate a Smith® chart. The block plots only the range of data corresponding to the specified frequencies.

The default value is `X-Y plane`.

Parameter #

Specify the S-parameters to plot. From the **Parameter1** and **Parameter2** drop-down lists, select the S-parameters that you want to plot. If you specify two parameters, the block plots both parameters in a single window.

The default value for **Parameter1** is `S11`. For the Amplifier and S-parameters blocks, the default value for **Parameter2** is `None`. For the Transmission Line block, the default value for **Parameter2** is `S22`.

Format #

For *X-Y* plots, format the units of the parameters to plot from the **Format1** and **Format2** drop-down lists. For polar plots and Smith charts, the formats are set automatically.

The default value is `Magnitude (decibels)`.

Y-axis scale
Scale for the *Y*-axis.

The default value is `Linear`.

X-axis scale
Scale for the *X*-axis.

The default value is `Linear`.

Examples

- The example, "Validating IP2/IP3 Using Complex Signals" verifies the nonlinear modeling capabilities of the amplifier block.
- The example, "Impact of an RF Receiver on Communication System Performance" performs quantitative noise analysis of the noise from an RF cascade.
- The example, Create a Low-IF Receiver Model, uses an amplifier in an IF receiver with specified gain and noise figure.

References

[1] Gonzalez, Guillermo. *Microwave Transistor Amplifiers: Analysis and Design*. Englewood Cliffs, N.J.: Prentice-Hall, 1984.

[2] Grob, Siegfried and Juergen Lindner. "Polynomial Model Derivation of Nonlinear Amplifiers". Department of Information Technology, University of Ulm, Germany.

[3] Kundert, Ken. "Accurate and Rapid Measurement of IP_2 and IP_3." *The Designers Guide Community*, Version 1b, May 22, 2002. http://www.designers-guide.org/analysis/intercept-point.pdf.

[4] Pozar, David M., *Microwave Engineering*. Hoboken NJ: John Wiley & Sons, 2005.

See Also

Mixer | S-Parameters

C (Capacitor)

Purpose

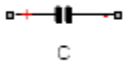
Model capacitor for circuit-envelope analysis

Library

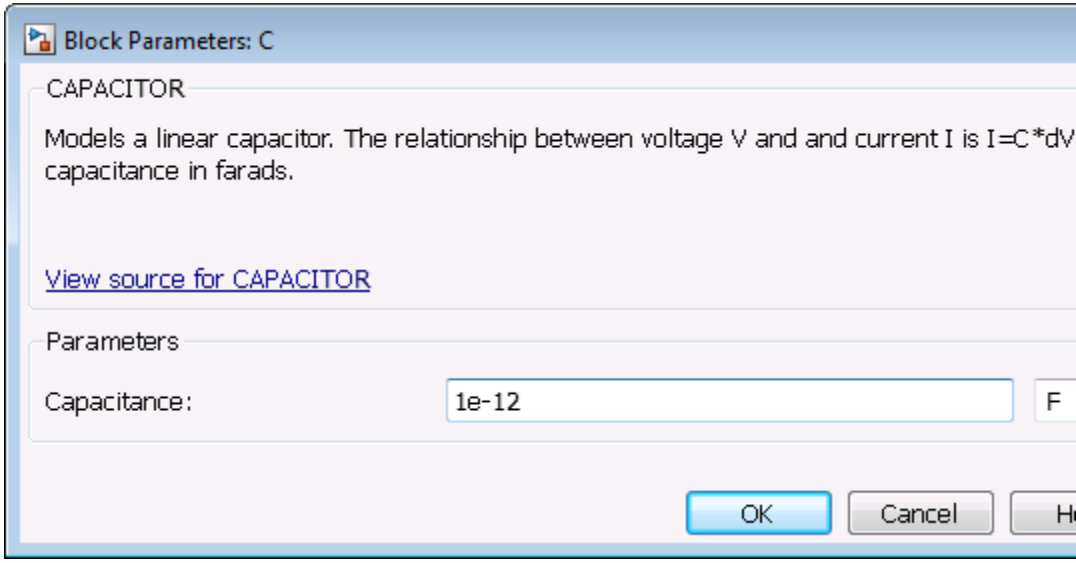
Elements

Description

The Capacitor block models a capacitor in circuit-envelope environment.



Dialog Box and Parameters



Capacitance

Specify the capacitance, *C*, as either `0` or a real number greater than or equal to `1e-18 F`. If you set this parameter to a value between `0` and `1e-18 F`, the block uses a value equal to `1e-18 F` during simulation. The default value of this parameter is `1e-12 F`.

Examples

- The example “AC Analysis of an RF System” analyzes an LC filter using harmonic balance.
- The example “Frequency Response of an RF Transmit/Receive Duplex Filter” simulates an analog RF filter comprised of Capacitor, Inductor, Resistor, and Impedance blocks.
- The section Filter Mixing Products walks through the construction of an LC low-pass tee filter in SimRF software.

See Also

Impedance | Inductor | Three-Winding Transformer | Resistor

Configuration

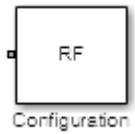
Purpose

Specify system-wide parameters for circuit-envelope analysis

Library

Utilities

Description

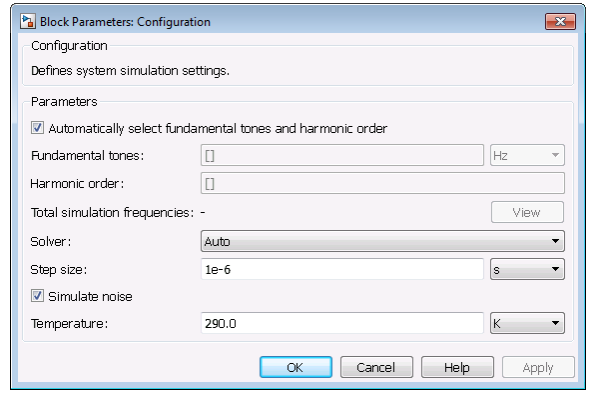


Use the Configuration block to set the model conditions for a circuit-envelope simulation. The block parameters define a set of simulation frequencies, solver attributes, and thermal noise.

Connect one Configuration block to each topologically distinct SimRF subsystem. Each Configuration block defines the parameters of the connected SimRF subsystem. To see an example of the Configuration block in a model, enter `simrfV2_noise` in a MATLAB Command Window.

For an introduction to RF simulation, see "Minimize Computations for RF Simulations".

Dialog Box and Parameters



Automatically select fundamental tones and harmonic order

When this check box is selected, the block determines the **Fundamental tones** and **Harmonic order** parameters automatically when you update the model. Automatic selection does not always return the smallest possible set of simulation frequencies.

Clearing this check box enables you to manually set the **Fundamental tones** and **Harmonic order** parameters. A smaller set of simulation frequencies decreases simulation time and decreases memory requirements. However, a decrease in simulation frequencies can reduce accuracy.

Fundamental tones

When **Automatically select fundamental tones and harmonic order** is cleared, specify a vector of positive frequencies. These frequencies represent the fundamental tones [f_1, f_2, ...] of the set of simulation frequencies. See the **Total simulation frequencies** parameter for additional information.

Harmonic order

When **Automatically select fundamental tones and harmonic order** is cleared, specify the harmonic order [h_1, h_2, ...] of each fundamental tone. Each h_i is a positive integer. You can specify a scalar that will be applied to each **Fundamental tones** parameter. See the **Total simulation frequencies** parameter for additional information.

Total simulation frequencies

The block displays the number of simulation frequencies for a nonlinear model. Because the solver computes a solution to the network at each simulation frequency, computation time scales according to the size of this value.

Click the **View** button to open a dialog box containing additional information about the simulation frequencies in your system. The **Configuration: Explaining simulation frequencies** dialog box lists tones and nonnegative simulation frequencies. By clicking on a simulation frequency in the list, you can see which linear combination or multiple combinations of fundamental tones represent that frequency. The dialog box also plots the positive simulation frequencies on a number line.

The block parameters define a set of simulation frequencies as linear combinations of fundamental tones: [$m*f_1 + n*f_2 + \ldots$],

where [f_1,f_2,...] is the set of fundamental tones, and the integers m and n are bounded by the corresponding **Harmonic order**, $|m| \leqq h1$, $|n| \leqq h_2$, etc.

For example, suppose you have a single fundamental tone f_1 = 2 GHz and corresponding harmonic order $h1$ = 3. Assume that m = [-3, -2, -1, 0, 1, 2, 3] defines the set of simulation frequencies, [-3f_1, -2f_1, -f_1, 0, f_1, 2f_1, 3f_1] = [-6GHz, -4GHz, -2GHz , 0GHz, 2GHz, 4GHz, 6GHz]. The total number of simulation frequencies is $2h_1$+1 = 7.

As a second example, suppose you have a circuit with two fundamental tones [f_1 = 2 GHz, f_2 = 50 MHz] and corresponding harmonic orders h_1 = h_2 = 1. Assume that m = [-1, 0, 1] and n = [-1, 0, 1]. This setup results in nine simulation frequencies ((2*h_1+1)*(2*h2+1)) with values [-f_1-f_2, -f_1, -f_1+f_2, -f_2, 0, f_2, f_1-f_2, f_1, f_1+f_2].

The set of simulation frequencies must include all carrier frequencies (inport and outport frequencies) specified in the SimRF subsystem.

Solver

Specify the fixed-step solver for the SimRF environment. When you are not sure which solver to use, set this parameter to `auto`. When manually choosing a solver, consider the following benefits and drawbacks:

- The `Backward Euler` solver is able to simulate the largest class of systems and signals. Damping effects make this solver suitable for wideband simulation, but overall accuracy is low.
- The `Trapezoidal Rule` solver is accurate for narrowband simulations. However, frequency warping and the lack of damping effects make this method inappropriate for most wideband simulations.
- The `NDF2` solver balances narrowband and wideband accuracy. This solver is suitable for situations where the frequency

content of the signals in the system is unknown relative to the Nyquist rate.

By default, **Solver** is set to `auto`.

The SimRF solver is an extension of the Simscape™ local solver. For more information on the Simscape local solver, see the Solver Configuration block reference page.

Step size

Specify a time step h for fixed-step integration. The default value is `1e-6 s`, which is sufficient for modeling envelope signals with bandwidths of up to $1/h$, or 1 MHz by default. However, simulation accuracy is reduced when simulating close to the maximum bandwidth. Reduce the step size to model signals with a larger bandwidth, or improve accuracy.

When the noise is simulated, the noise bandwidth for each simulation frequency is equal to $1/h$.

Simulate noise

Use this parameter to globally enable or disable noise modeling for SimRF blocks that support noise. When this check box is selected:

- Amplifier and Mixer blocks use the value of their respective **Noise figure (dB)** parameters.
- Amplifier and Mixer blocks simulate with thermal noise at the temperature specified by the **Temperature** parameter.
- Resistor blocks model thermal noise using the **Temperature** parameters.
- Noise blocks model a specified noise power as a voltage or current source.

Clearing this check box disables noise modeling in the SimRF environment. By default, this check box is selected.

Temperature

When **Simulate noise** is selected, specify a global noise temperature. The default value of this parameter is `290.0 K`.

References

Motchenbacher, C.D. and J.A. Connely. *Low Noise Electronic System Design*. New York: John Wiley & Sons, 1993.

Rodrigues, Paulo J. C. *Computer-Aided Analysis of Nonlinear Microwave Circuits*. Norwood, MA: Artech House, Inc., 1998.

Purpose

Model constant envelope source

Library

Sources

Description

The Continuous Wave block models a constant modulation on a carrier in the SimRF circuit-envelope simulation environment. For an introduction to RF simulation, see the example, “Minimize Computations for RF Simulations”.

The block implements the relation

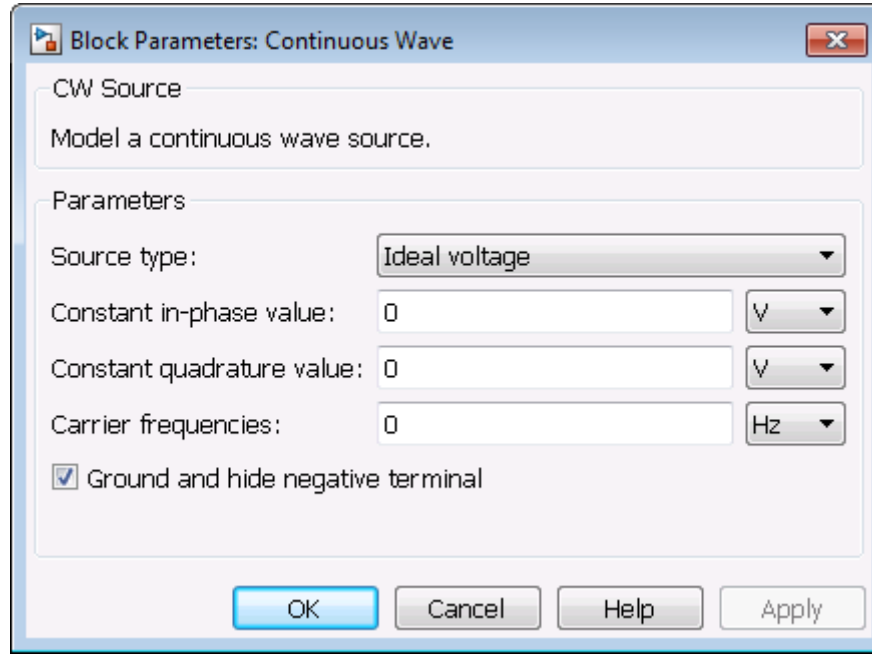
$$v(t) = v_0 e^{j\omega_k t}$$

or

$$i(t) = i_0 e^{j\omega_k t}$$

at the carrier ω_k, depending on the source type.

Dialog Box and Parameters



Source type

Specify the wave type as:

- `Ideal Voltage` — The block simulates a constant voltage envelope v_0 at the specified **Carrier frequencies**. The envelope has real and imaginary parts specified by the **Constant in-phase value** and **Constant quadrature value** parameters.
- `Ideal Current` — The block simulates a constant current envelope i_0 at the specified **Carrier frequencies**. The envelope has real and imaginary parts determined by the **Constant in-phase value** and **Constant quadrature value** parameters.
- `Power` — The block simulates the constant voltage envelope

$$v_0 = 2\sqrt{\mathrm{P_0 Re}(Z_s)}e^{j\frac{\pi}{180}\phi}$$

where:

- P_0 is the value of the **Available power** parameter
- Z_s is the value of the **Source impedance (ohms)** parameter.
- φ is the value of the **Angle (degrees)** parameter.

The default value of this parameter is `Ideal Voltage`.

Constant in-phase value

When the **Source type** is `Ideal Voltage` or `Ideal Current`, specify the in-phase modulations for each of the specified **Carrier frequencies** as a vector of real numbers. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 V`.

Constant quadrature value

When the **Source type** is `Ideal Voltage` or `Ideal Current`, specify the quadrature modulations for each of the specified **Carrier frequencies** as a vector of real numbers. Specify the units from the corresponding drop-down list. The quadrature value at baseband must equal `0`. The default value of this parameter is `0 V`.

Source impedance (ohms)

When the **Source type** is `Power`, specify the source impedance as either `0` or a complex number with real and imaginary parts that are greater than or equal to `1e-18 Ohm`. If you set this parameter to a value with real or imaginary part between `0` and `1e-18 Ohm`, the block uses a value with real or imaginary part equal to `1e-18 Ohm` during simulation. The default value of this parameter is `0 Ohm`.

Available power

When the **Source type** is `Power`, specify the available power at the specified **Carrier frequencies** as a vector of real numbers. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 W`.

Angle (degrees)

When the **Source type** is `Power`, specify the phase angles of the power waves at the specified **Carrier frequencies** as a vector of real numbers. The default value of this parameter is `0` degrees.

Carrier frequencies

Enter a vector of carrier frequencies whose elements are calculated from fundamental tones and corresponding harmonic in the Configuration block. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 Hz`.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

All models in the SimRF documentation that contain a Mixer block also contain a Continuous Wave block used as a local oscillator.

See Also

Inport | Sinusoid

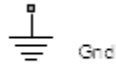
Purpose

Simulate connection to electrical ground

Library

Elements

Description



The Ground block represents an electrical ground in a SimRF circuit-envelope simulation environment. Connect at least one Ground to the SimRF environment; otherwise, models with SimRF blocks do not run.

Dialog Box and Parameters

The Ground block has no parameters.

Examples

All models in the SimRF documentation contain a Ground block. See Model an RF Mixer for an introduction to circuit-envelope simulation.

Ideal Transformer

Purpose

Model ideal transformer

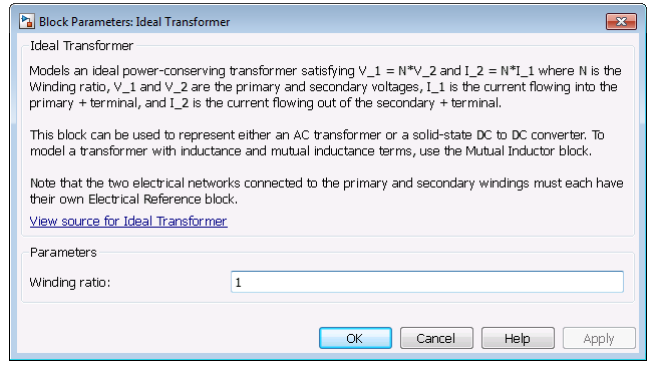
Library

Elements

Description

The Ideal Transformer block models a constant IV relationship within the SimRF circuit-envelope simulation environment. For an introduction to RF simulation, see the example, “Minimize Computations for RF Simulations”.

Dialog Box and Parameters



Winding ratio

Specify the winding ratio.

See Also

Mutual Inductor | Three Winding Transformer

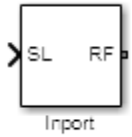
Purpose

Convert Simulink input signal to SimRF signal

Library

Utilities

Description



The Inport block imports sample-based Simulink® signals into the SimRF circuit-envelope simulation environment. For an introduction to RF simulation, see the example, "Minimize Computations for RF Simulations".

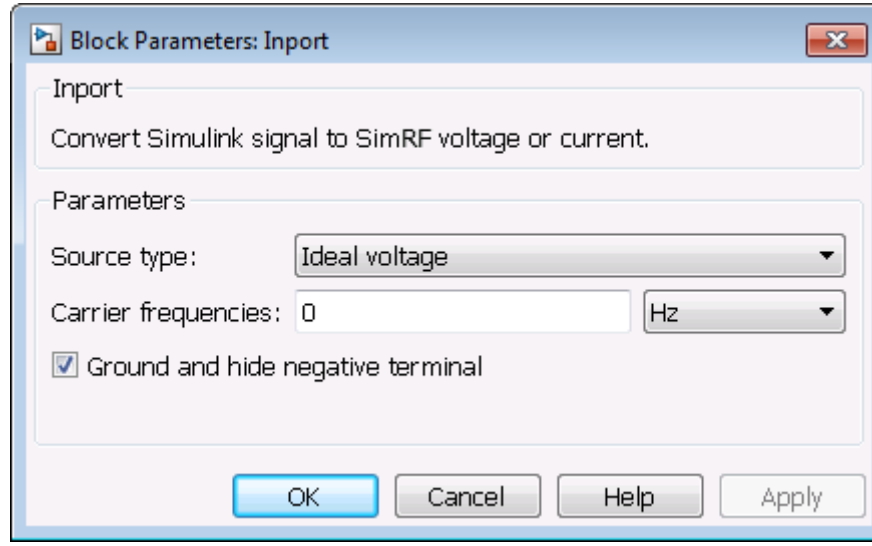
Complex-valued, sample-based input signals $I_k(t) + j \cdot Q_k(t)$ become the modulations of the frequencies $\{f_k\}$ specified in the **Carrier frequencies** parameter of the block:

$$i(t) = \sum_{\{f_k\}} \left(i_{I_k}(t) + j \cdot i_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

$$v(t) = \sum_{\{f_k\}} \left(v_{I_k}(t) + j \cdot v_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

The **Source type** parameter specifies the SimRF signal as either a current signal $i(t)$ or a voltage signal $v(t)$ in the SimRF environment.

Dialog Box and Parameters

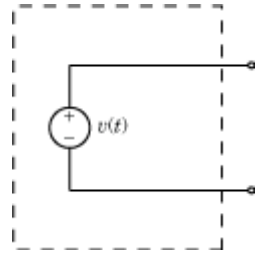


Source type

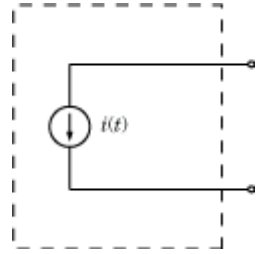
Specify a current, voltage, or power source. If the input signal is a vector containing more than one signal, the block assigns the kth input signal to the kth carrier.

Choose how the block interprets the Simulink signal:

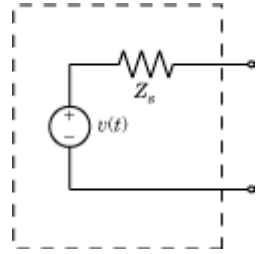
- `Ideal voltage` — The block outputs the Simulink signal as a voltage signal $v(t)$ in the SimRF environment. When you choose this option, the following figure illustrates the internal configuration of the block.



- `Ideal current` — The block outputs the Simulink signal as a current signal $i(t)$ in the SimRF environment. When you choose this option, the following figure illustrates the internal configuration of the block.



- `Power` — The block uses the Simulink signal, $v_{SL}(t)$, to output an available power of $|v_{SL}(t)|^2$ to a matched load. When you choose this option, in addition to a voltage source, a series impedance appears internally, as shown in the following figure.



The voltage $v(t)$ is a scaling of the Simulink signal $v_{SL}(t)$:

$$v(t) = 2\sqrt{\mathrm{Re}(Z_s)}v_{SL}(t)$$

In the preceding equation, Z_s is the value of the **Source impedance (ohms)** parameter.

The generator delivers a real power to a load Z_l:

$$P_{load} = |v(t)|^2 \frac{\mathrm{Re}(Z_s)}{|Z_s + Z_l|^2}$$

When $Z_l = Z_s^*$, this generator delivers the available power $|v_{SL}(t)|^2$.

Source impedance (ohms)

When **Input** is set to `Power`, specify a source impedance. When you use multiple Inport blocks as power sources at the same node in a given circuit, each block modifies the load for other components connected to that shared node, so the maximum available power is not achieved.

Carrier frequencies

Enter a vector of carrier frequencies whose elements are calculated from fundamental tones and corresponding harmonic in the Configuration block. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 Hz`.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

The example "Two-Tone Envelope Analysis Using Real Signals" shows how to use Inport and Outport blocks to position an RF system as part of a larger workflow.

See Also

Continuous Wave | Outport

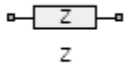
Purpose

Model complex impedance

Library

Elements

Description



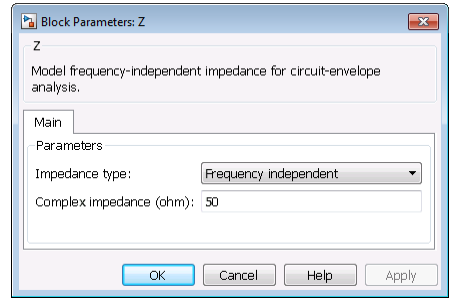
The Impedance block implements the relation, $v_k(t) = Z(f_k)*i_k(t)$, for each simulation frequency, f_k, where:

- $Z(f_k)$ represents complex-valued impedance at a specified simulation frequency.
- $v_k(t)$ represents the voltage across the terminals of the element at time t.
- $i_k(t)$ represents the current through the element at time t.

Circuit envelope current and voltage signals comprise in-phase, I_k, and quadrature, Q_k, components at each frequency, f_k.

Frequency-dependent impedance typically cannot be realized as a physical network such as an RLC chain. You can, however, use the Impedance block to model nonphysical behavior, such as frequency-independent negative capacitance or negative inductance. You can also use this block to specify resonant frequency offsets in filter networks.

Dialog Box and Parameters



Impedance type

Choose `Frequency independent` to apply the same impedance for all frequencies. Choose `Frequency dependent` to apply the impedance as a piecewise linear function.

Complex impedance (ohm)

When **Impedance type** is set to `Frequency independent`, impedance is a scalar complex number that is applied to all simulation frequencies. When **Impedance type** is set to `Frequency dependent`, impedance is a vector of complex numbers, $[Z_1, Z_2, Z_3, \ldots]$. This vector is linearly interpolated for all simulation frequencies.

In both cases, for zero simulation frequency, the imaginary part of the impedance is ignored. Also, the real part is forced to be positive to produce a stable simulation.

Frequency

When **Impedance type** is set to `Frequency independent`, specify a vector of nonnegative frequencies, $[f_1, f_2, f_3, \ldots]$. The pairs, f_i, Z_i, define a piecewise linear function, $Z(f)$, that is linearly interpolated for the simulation frequencies. For values outside the range, there is constant extrapolation. The default value of this parameter is `0` Hz.

Examples

The example, "Frequency Response of an RF Transmit/Receive Duplex Filter", simulates an analog RF filter comprising Capacitor, Inductor, Resistor, and Impedance blocks.

See Also

Capacitor | Inductor | Resistor

Purpose

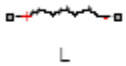
Model inductor for circuit-envelope analysis

Library

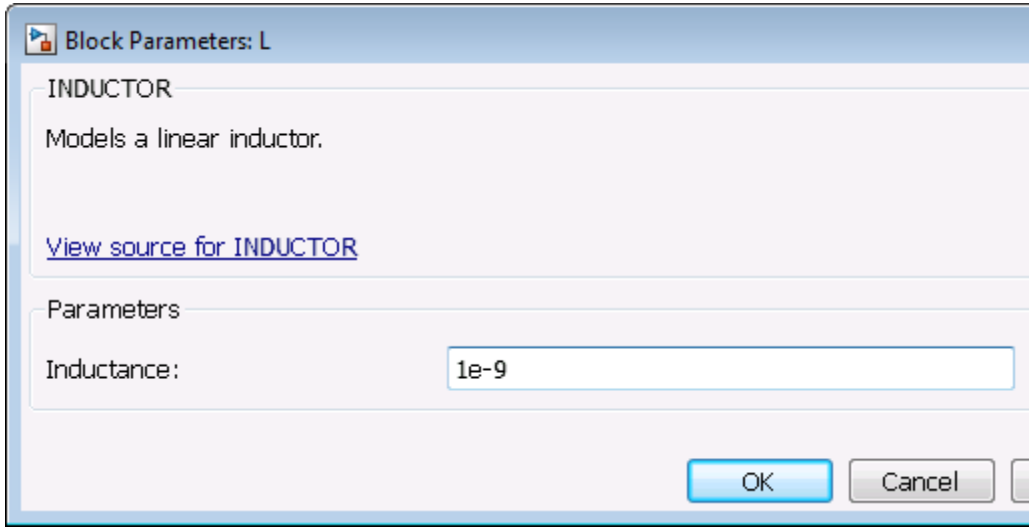
Elements

Description

The Inductor block models an inductor in circuit-envelope environment.



Dialog Box and Parameters



Inductance

Specify the inductance, *L*, as either `0` or a real number greater than or equal to `1e-18 H`. If you set this parameter to a value between `0` and `1e-18 H`, the block uses a value equal to `1e-18 H` during simulation. The default value of this parameter is `1e-9 H`.

Examples

- The example "AC Analysis of an RF System" analyzes an LC filter using harmonic balance.

- The example “Frequency Response of an RF Transmit/Receive Duplex Filter” simulates an analog RF filter comprised of Capacitor, Inductor, Resistor, and Impedance blocks.
- The section Filter Mixing Products walks through the construction of an LC lowpass filter in SimRF software.

See Also

Capacitor | Resistor | Mutual Inductor | Three-Winding Transformer

Purpose

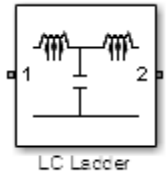
Model LC ladder networks

Library

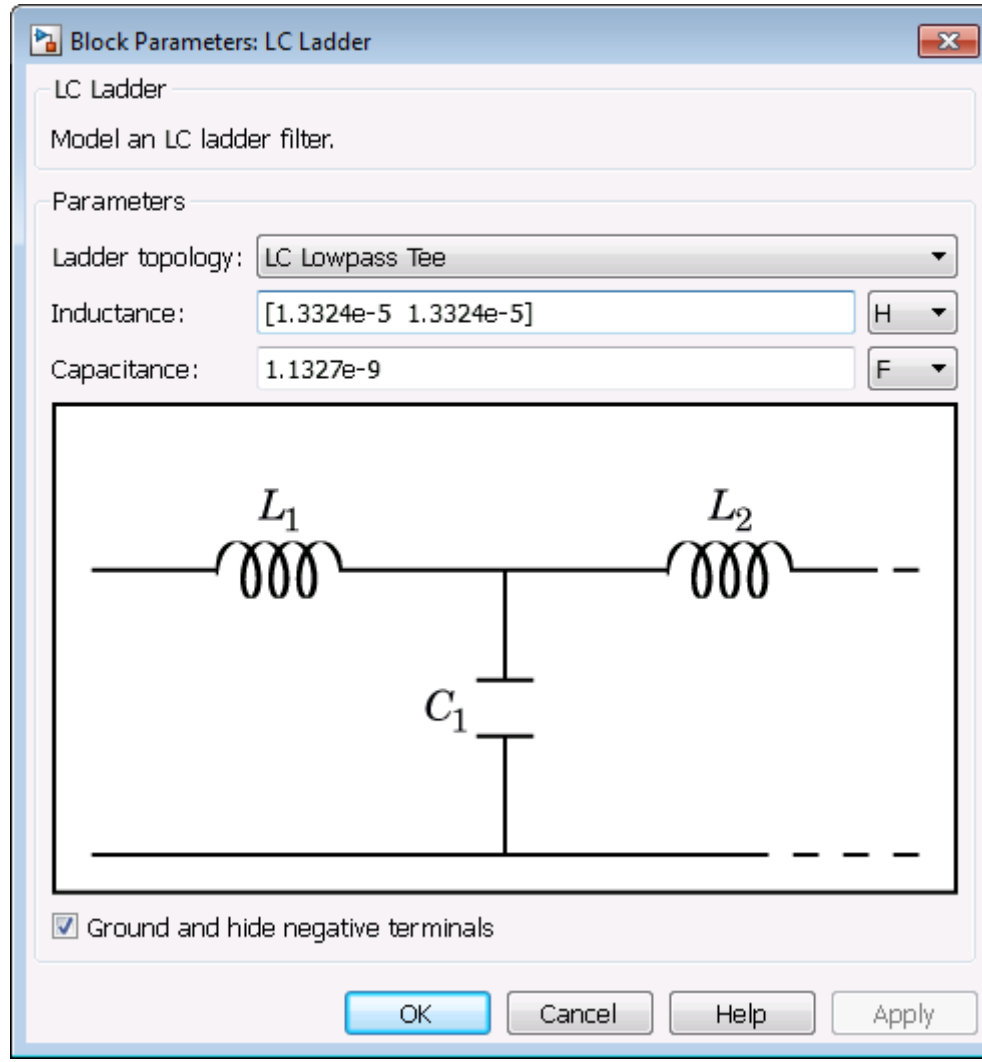
Elements

Description

The LC Ladder block models common two-port LC lossless networks.



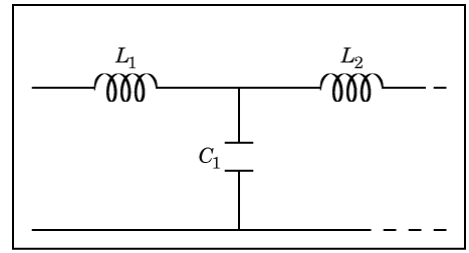
Dialog Box and Parameters



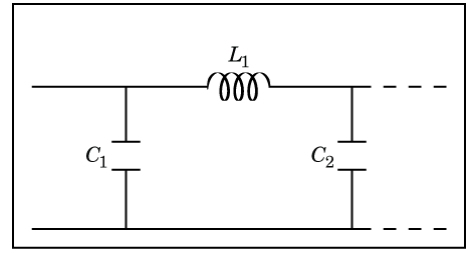
Ladder topology

Specify the ladder topology from the following options.

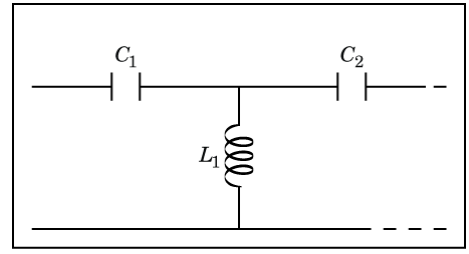
- LC Lowpass Tee



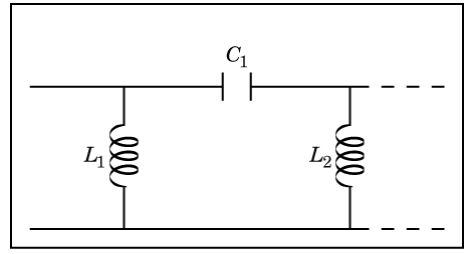
- LC Lowpass Pi



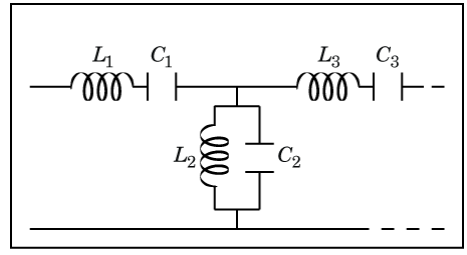
- LC Highpass Tee



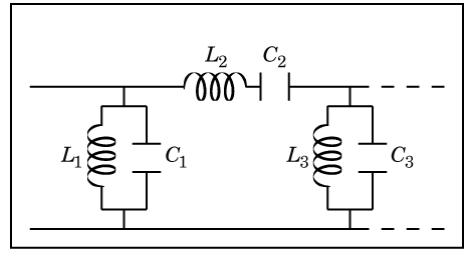
- LC Highpass Pi



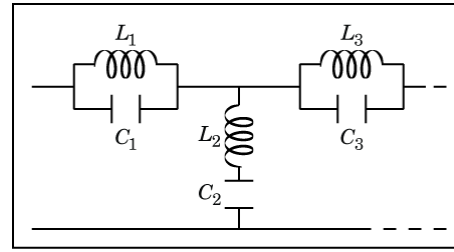
- LC Bandpass Tee



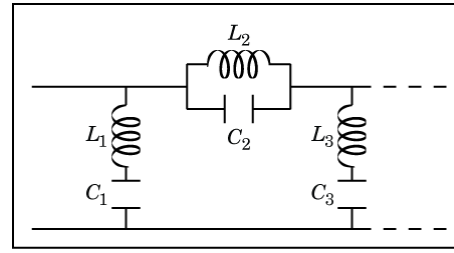
- LC Bandpass Pi



- LC Bandstop Tee



- `LC Bandstop Pi`



Inductance

Specify a vector of inductance values. The vector index of a value corresponds to the inductance index in the corresponding selected ladder topology. All values must be greater than zero.

Capacitance

Specify a vector of capacitance values. The vector index of a value corresponds to the capacitance index in the corresponding selected ladder topology. All values must be greater than zero.

The relationship between vector lengths of Capacitance and Inductance values must correspond to the Ladder topology selected.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

- The section “Model an RF Mixer” steps through setting up an RF mixer in the SimRF environment.

Purpose

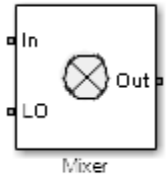
Model mixer in RF systems

Library

Elements

Description

The Mixer block performs signal frequency translation and nonlinear amplification.



Mixer

For a given RF input signal $V_{RF} = A_{RF}\cos(\omega_{RF}t)$ and an LO input signal $V_{LO} = A_{LO}\cos(\omega_{LO}t)$, the mixer multiplies the signals at the input ports:

$$V_{in}V_{LO} = A_{in}\cos(\omega_{in}t)A_{LO}\cos(\omega_{LO}t)$$
$$\frac{A_{in}A_{LO}}{2}\cos\left[(\omega_{in}+\omega_{LO})t\right]+\frac{A_{in}A_{LO}}{2}\cos\left[(\omega_{in}-\omega_{LO})t\right]$$

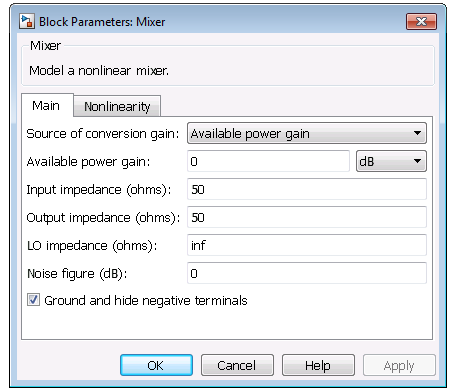
This mixing converts the frequency of RF signal to $\omega_{RF} + \omega_{LO}$ and $\omega_{RF} - \omega_{LO}$. For the mixer to perform this operation correctly, you must include the frequencies $\omega_{RF} + \omega_{LO}$ or $\omega_{RF} - \omega_{LO}$ in the simulation frequencies the Configuration block calculates.

The Power gain specification for this block relates the power of a single-sideband (SSB) to the input.

After mixing the RF and LO signals, the mixer block performs amplification. To model linear amplification, the mixer scales the signals by the coefficient a_1. A Voltage Controlled Voltage Source (VCVS), specified with a polynomial, implements nonlinear amplification. The polynomial includes saturation automatically and produces additional intermodulation frequencies.

Dialog Box and Parameters

Main Tab



Source of conversion gain

Specify the source parameter of the conversion gain as:

- `Available power gain` — The block uses the value of the Available power gain parameter to calculate the linear voltage gain term of the polynomial VCVS, a_1. This calculation assumes a matched load termination for the mixer.
- `Open circuit voltage gain` — The block uses the value of the Open circuit voltage gain parameter as the linear voltage gain term of the polynomial VCVS, a_1.
- `Polynomial coefficients` — The block implements a nonlinear voltage gain according to the polynomial you specify. The order of the polynomial must be less than or equal to 9 and the coefficients are ordered in ascending powers. If a vector a has 10 coefficients, $[a_0, a_1, a_2, \ldots, a_9,]$, the polynomial it represents is $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \ldots + a_9 V_{in}^9$. In this case, a_1 represents the linear gain term, and the modeling of higher-order terms is done according to [1].

  For example, the vector $[a_0, a_1, a_2, a_3]$ specifies the relation $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \ldots + a_3 V_{in}^3$.

Trailing zeroes are omitted; if $a_3 = 0$, $[a_0, a_1, a_2]$ defines the same polynomial as $[a_0, a_1, a_2, 0]$. The default value of this parameter is [0 1], corresponding to the linear relation $V_o = V_i$.

The default value of this parameter is `Available power gain`.

Available power gain

When **Source of conversion gain** is `Available power gain`, specify the linear gain of the mixer. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 dB`.

Open circuit voltage gain

When **Source of conversion gain** is `Open circuit voltage gain`, specify the open circuit voltage gain of the mixer. Specify the units from the corresponding drop-down list. If you specify the units as `None`, the gain must be positive. The default value of this parameter is `0 dB`.

Input impedance (ohms)

Specify the scalar impedance at the **In** port of the mixer. The default value of this parameter is `50 Ω`.

Output impedance (ohms)

Specify the scalar impedance at the **Out** port of the mixer. The default value of this parameter is `50 Ω`.

LO impedance (ohms)

Specify the scalar impedance at the **LO** port of the mixer. The default value of this parameter is `Inf Ω`.

Noise figure (dB)

Specify the single-sideband (SSB) noise figure of the mixer. The default value of this parameter is `0` dB.

To model noise in circuit envelope model with a Noise, Amplifier, or Mixer block, you must select the **Simulate noise** check box in the Configuration block dialog box.

The following table summarizes the two competing definitions for specifying SSB noise, where the image frequency (IM) is defined as $\omega_{IM} = \omega_{LO} + (\omega_{LO} - \omega_{RF})$.

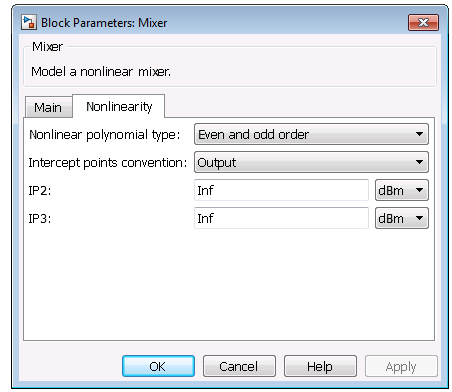
Noise Convention	Signal at RF Frequency	Signal at IM Frequency	Mixer Block Supports This Model?
Single-sideband noise (SSB)	*S* + *N*, signal with noise	*N*, noise only	Yes
IEEE definition of single-sideband noise (SSB_{IEEE})	*S* + *N*, signal with noise	No signal	No; you can create an equivalent model using an ideal filter created from an S-parameters block.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Nonlinearity Tab



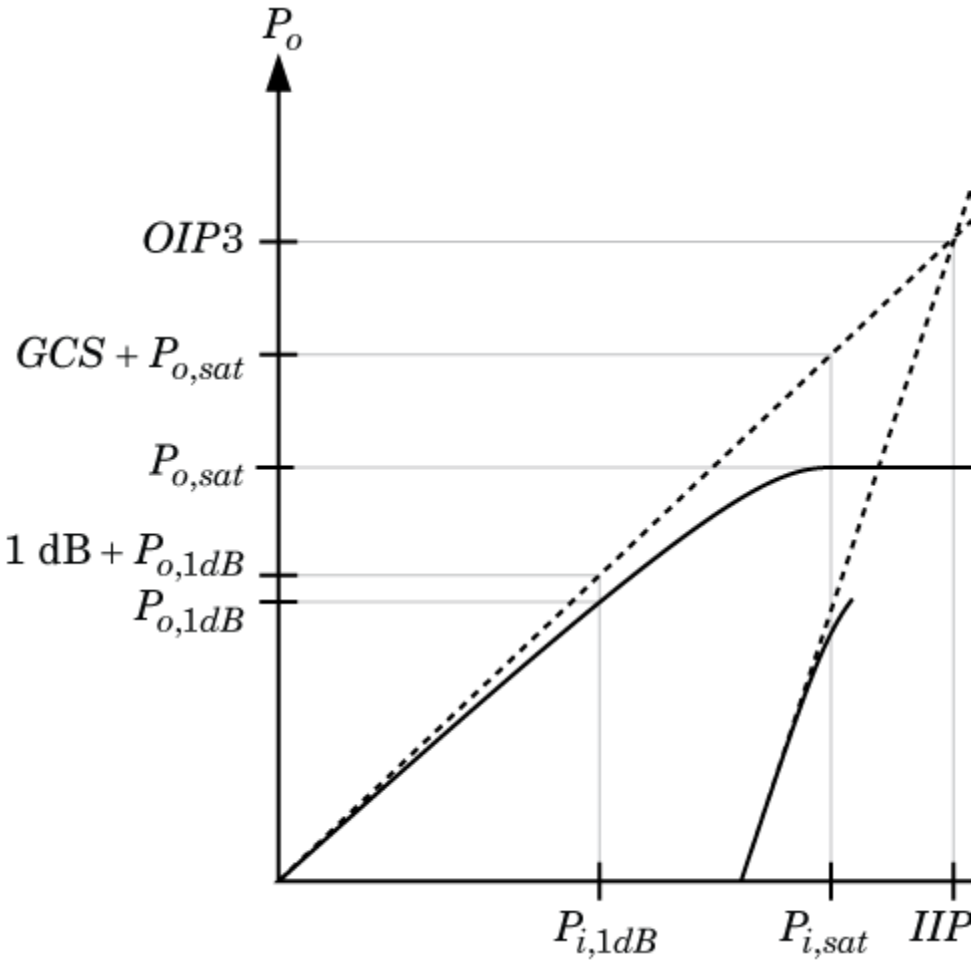
The specification is identical to that of the Amplifier block, except that it includes a scaling factor of 2 to account for the SSB mixer conversion gain.

Nonlinear polynomial type

Specify either an `Even and odd order` or `Odd order` polynomial nonlinearity. The default value is `Even and odd order`.

- When you select `Even and odd order`, the amplifier can produce second- and third-order intermodulation frequencies in addition to a linear term.
- When you select `Odd order`, the amplifier generates only odd order intermodulation frequencies.

  The linear gain determines the linear a_1 term. The block calculates the remaining terms from the specified parameters. These parameters are **IP3**, **1-dB gain compression power**, **Output saturation power**, and **Gain compression at saturation**. The number of constraints you specify determines the order of the model.



The preceding figure shows the graphical definition of the nonlinear amplifier parameters.

Intercept points convention

Specify either an `Input`-referred or `Output`-referred convention. Use this specification for the intercept points, 1-dB gain compression power, and saturation power.

The default value is `Output`.

IP2

When **Nonlinear polynomial type** is `Even and odd order`, specify the second-order intercept point of the amplifier.

The default value is `inf dBm`, which corresponds to an unspecified point.

IP3

Specify the third-order intercept point of the amplifier. The default value is `inf dBm`, which corresponds to an unspecified point.

1-dB gain compression power

When **Nonlinear polynomial type** is `Odd order`, specify the 1-dB gain compression point. The 1-dB gain compression point must be less than the output saturation power.

The default value is `inf dBm`, which corresponds to an unspecified point.

Output saturation power

When **Nonlinear polynomial type** is `Odd order`, specify the output saturation power. The block uses this value to calculate the voltage saturation point used in the nonlinear model. In this case, the first derivative of the polynomial is zero, and the second derivative is negative.

The default value is `inf dBm`, which corresponds to an unspecified point in the polynomial model.

Gain compression at saturation

When **Nonlinear polynomial type** is `Odd order`, specify the gain compression at saturation. This parameter cannot be set unless **Output saturation power** is specified.

The default value is `inf dBm`.

Examples

- The example, “Validating IP2/IP3 Using Complex Signals”, verifies the nonlinear modeling capabilities of the Amplifier block.
- The example, “Impact of an RF Receiver on Communication System Performance”, performs quantitative noise analysis of the noise from an RF cascade.
- The section, Create a Low-IF Receiver Model, uses an amplifier in an IF receiver with specified gain and noise figure.
- The section, “Model an RF Mixer”, steps through setting up an RF mixer in the SimRF environment.

References

[1] Grob, Siegfried and Lindner, Jurgen, “Polynomial Model Derivation of Nonlinear Amplifiers”, *Department of Information Technology*, University of Ulm, Germany.

See Also

Amplifier

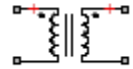
Purpose

Model two coupled inductors for circuit-envelope analysis

Library

Elements

Description



Mutual Inductor

The Mutual Inductor block models an inductor within the SimRF circuit-envelope simulation environment. For an introduction to RF simulation, see the example, "Minimize Computations for RF Simulations".

The block implements the relations

$$v_1(t) = L_1 \frac{d}{dt}\left[i_1(t)\right] + M \frac{d}{dt}\left[i_2(t)\right]$$

$$v_2(t) = M \frac{d}{dt}\left[i_1(t)\right] + L_2 \frac{d}{dt}\left[i_2(t)\right]$$

$$M = K\sqrt{L_1 L_2}$$

where:

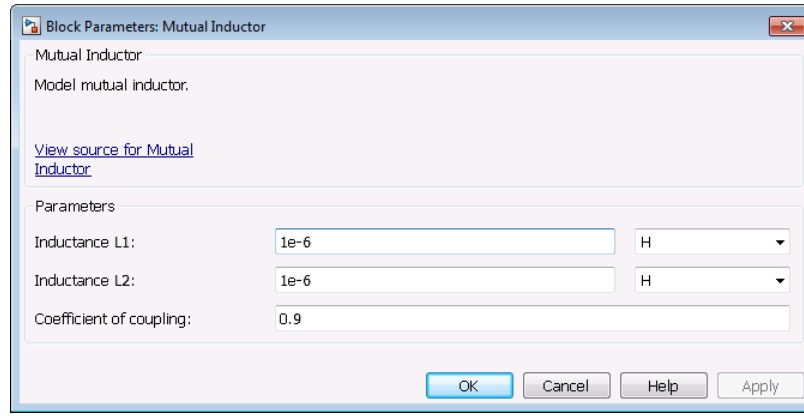
- L_1 and L_2 represent inductances.
- M represents a mutual inductance with coefficient of coupling K.
- $v_1(t)$ and $v_2(t)$ represent the voltage across the terminals of the inductors at time t.
- $i_1(t)$ and $i_2(t)$ represent the current through the inductors at time t. The block uses standard dot notation to indicate the direction of positive current flow relative to a positive voltage.

SimRF current and voltage signals consist of in-phase (I_k) and quadrature (Q_k) components at each frequency f_k specified in the Configuration block:

$$i(t) = \sum_{\{f_k\}} \left(i_{I_k}(t) + j \cdot i_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

$$v(t) = \sum_{\{f_k\}} \left(v_{I_k}(t) + j \cdot v_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

Dialog Box and Parameters



Inductance L1

Specify the inductance of the first inductor as a scalar value greater than or equal to `0`. Specify the units of the inductance from the corresponding drop-down list. The default value of this parameter is `1e-6 H`.

Inductance L2

Specify the inductance of the second inductor as a scalar value greater than or equal to `0`. Specify the units of the inductance from the corresponding drop-down list. The default value of this parameter is `1e-6 H`.

Coefficient of coupling

Specify the coefficient of coupling for the mutual inductance of the two inductors as a scalar value between `0` and `1`, inclusive. The default value of this parameter is `0.9`.

Note The minimum nonzero inductance value that the SimRF environment recognizes is `1e-18 H`. During simulation, the block uses a value of `1e-18 H` for any inductance and mutual inductance values specified between `0` and `1e-18 H`.

See Also

Inductor | Three-Winding Transformer

Noise

Purpose

Model noise using current or voltage noise source in RF systems

Library

Sources

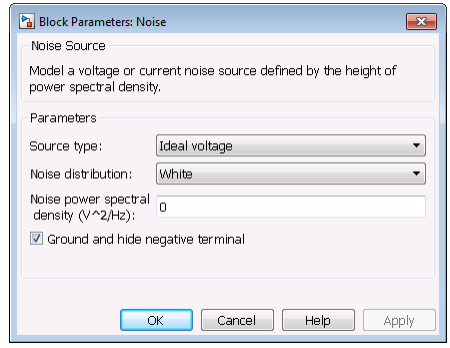
Description

When you use a Noise block to simulate noise in an RF model, also select the **Simulate noise** check box in the Configuration block. Otherwise, the model simulates without noise.

Tips

- The Noise block is useful for modeling noise as a current or voltage source for blocks other than Amplifier and Mixer blocks. For example, you can simulate global thermal noise in amplifiers and mixers. In the Configuration block dialog box, specify a noise temperature in the Temperature parameter.
- The Noise block does not depend on the Temperature parameter. This global Temperature parameter affects only Resistor, Amplifier, and Mixer noise simulation.

Dialog Box and Parameters



Source type
Specify the noise type as `Ideal Voltage` or `Ideal Current`. The default is `Ideal Voltage`.

Noise distribution
Specify the noise distribution type as `White` or `Piece-wise linear`. The default is `White`.

Noise power spectral density
Specify the single-sided noise power spectral distribution (PSD). The default value of this parameter is `0`.

When **Noise distribution** is set to:

- `White`, spectral density is a single non-negative value.
- `Piece-wise linear`, spectral density is a vector of values $[p_i]$.

When **Source type** is set to:

- `Ideal voltage`, spectral density units are V^2/Hz.
- `Ideal Current`, spectral density units are A^2/Hz.

Frequencies
This parameter appears only when **Noise distribution** is set to `Piece-wise linear`.

Parameter values are specified as a vector of non-negative frequencies $[f_i]$ with the same length as the vector of power spectral densities $[p_i]$. The pairs, (f_i,p_i), define an arbitrary piece-wise linear noise distribution. The default value of this parameter is `0` Hz.

Examples

The section, Create a Low-IF Receiver model, includes a model with a Noise block that models a thermal noise floor.

Explicitly Simulate Resistor Thermal Noise

The ex_simrf_noise_source model calculates the classic noise floor, kT, for a matched resistor circuit.

This example model simulates noise in the resistor, R, using a parallel current noise source of $(4kT/R)*(A^2/Hz)$. The power generated in a matching noiseless load is kTB watts. $B = 1/h$ is the bandwidth defined by the simulation time step, h, specified in the Configuration block. With bandwidth included, noise power equals 10*log10(1.3806503 × 10^{-23} m^2 kg s^{-2} K^{-1} * 290C / 10–3W/mW) = –173.98 dBm.

References

Motchenbacher, C.D. and J.A. Connely. *Low Noise Electronic System Design*. New York: John Wiley & Sons, 1993.

See Also

Configuration

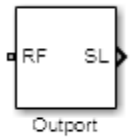
Purpose

Convert SimRF signal to Simulink output signal

Library

Utilities

Description



The Outport block outputs the modulation of a signal in the SimRF circuit-envelope simulation environment as a Simulink signal. For an introduction to RF simulation, see the example, “Minimize Computations for RF Simulations”.

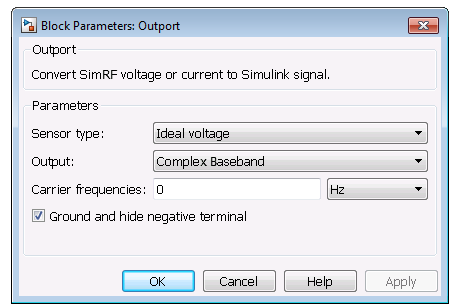
SimRF current and voltage signals consist of in-phase (I_k) and quadrature (Q_k) components at each frequency f_k specified in the Configuration block:

$$i(t) = \sum_{\{f_k\}} \left(i_{I_k}(t) + j \cdot i_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

$$v(t) = \sum_{\{f_k\}} \left(v_{I_k}(t) + j \cdot v_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

The **Sensor type** and **Output** parameters determine which signal the block measures.

Dialog Box and Parameters



Sensor type

Specify the type of signal measured by the sensor:

- `Ideal voltage` — The block outputs the modulations of the voltage signal

$$v(t) = \sum_{\{f_k\}} \left(v_{I_k}(t) + j \cdot v_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

at the specified **Carrier frequencies** in the format specified by the **Output** parameter.

- `Ideal current` — The block outputs the modulations of the current signal

$$i(t) = \sum_{\{f_k\}} \left(i_{I_k}(t) + j \cdot i_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

at the specified **Carrier frequencies** in the format specified by the **Output** parameter.

- `Power` — The block outputs the modulations of the voltage signal

$$\frac{\sqrt{\mathrm{Re}(Z_l)}}{Z_l} v(t)$$

at the specified **Carrier frequencies** in the format specified by the **Output** parameter. where Z_l is the value of the **Load impedance (ohms)** parameter.

If the **Carrier frequencies** parameter specifies more than one frequency, the block outputs a vector of sample-based signals. The kth output signal corresponds to the modulation of the kth carrier.

Load impedance (ohms)

When **Output** is set to `Power`, the Outport loads the circuit with the specified impedance. When you use multiple Outport blocks as power sources at the same node in a given circuit, the resulting load is the parallel combination of the specified load impedances.

Output

Specify the format of the output signals:

- `Complex Baseband` — The block outputs a vector of complex-valued, sample-based signals $I_k(t) + j \cdot Q_k(t)$ at the

SL port. The kth element of the vector is the kth frequency specified by the **Carrier frequencies** parameter.

- `In-phase and Quadrature Baseband` — The block outputs two vectors of real-valued, sample-based signals $I_k(t)$ and $Q_k(t)$ at the **I** port and **Q** port, respectively. The signal at the **I** port contains the in-phase components, and the signal at the **Q** port contains the quadrature components. The kth element of the vector is the kth frequency specified by the **Carrier frequencies** parameter.
- `Magnitude and Angle Baseband` — The block outputs two vectors of real-valued, sample-based signals whose elements are the magnitude and phase angle of the modulation. The **Mag** port outputs $|I_k(t) + j \cdot Q_k(t)|$ and the **Ang** port outputs $\text{Arg}[I_k(t) + j \cdot Q_k(t)]$. The kth element of the vector is the kth frequency specified by the **Carrier frequencies** parameter.
- `Real Passband` — The block outputs real passband signals corresponding to each carrier by multiplying the envelope signals by their carriers. The signal on the kth carrier is of the form $x(t) = I(t)\cos(2\pi f_k t) - Q(t)\sin(2\pi f_k t)$

Note When using the `Real Passband` option, a variable-step solver attempts to take time steps small enough to resolve the carrier. Simulation speed improvements from envelope simulation may be lost.

Carrier frequencies

Enter a vector of carrier frequencies whose elements are calculated from fundamental tones and corresponding harmonic in the Configuration block. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 Hz`.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals.

By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

- The "Minimize Computations for RF Simulations" example compares the `Real Passband` and `In-phase and Quadrature Baseband` output options of the Outport block.
- The example "Validating IP2/IP3 Using Complex Signals" shows how to use Outport blocks to probe RF systems in multiple locations.

See Also

Inport

Purpose

Model phase shift in RF systems

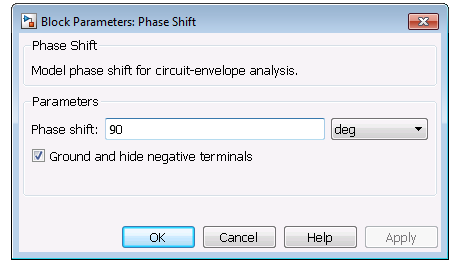
Library

Elements

Description

The Phase Shift block models an ideal phase shift in the circuit envelope environment.

Dialog Box and Parameters



Phase-shift

The default value of this parameter is `90 deg`. The phase shift is applied to all nonzero simulation frequencies. For zero (DC) frequency, the shift is always zero.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

The example, "", shows how to use Phase Shift blocks to model an LO phase offset in a receiver architecture.

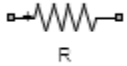
Purpose

Model resistor for circuit-envelope analysis

Library

Elements

Description



The Resistor block models a resistor within the SimRF circuit-envelope simulation environment. For an introduction to RF simulation, see the example, “Minimize Computations for RF Simulations”.

The block implements the relation

$$v(t) = Ri(t)$$

where:

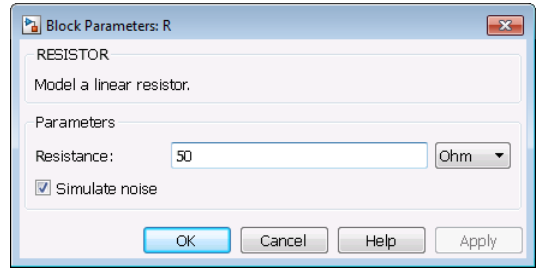
- R represents the resistance, as a function of temperature.
- $i(t)$ represents the current through the capacitor at time t.
- $v(t)$ represents the voltage across the terminals of the capacitor at time t.

SimRF current and voltage signals consist of in-phase (I_k) and quadrature (Q_k) components at each frequency f_k specified in the Configuration block:

$$i(t) = \sum_{\{f_k\}} \left(i_{I_k}(t) + j \cdot i_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

$$v(t) = \sum_{\{f_k\}} \left(v_{I_k}(t) + j \cdot v_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

Dialog Box and Parameters



Resistance

Specify the resistance, R_0, as a real number greater than zero. Specify the units of the resistance from the corresponding drop-down menu. The default value of this parameter is `50 Ohm`.

Simulate noise

To simulate thermal noise in a resistor, select this check box. Then, in the Configuration block dialog box, also select the **Simulate noise** check box. By default, both **Simulate noise** check boxes are selected.

This parameter inserts a current noise source with the single-sided power density of 4 kT/R A^2/Hz, where:

- k is the Boltzmann constant
- T is the value of the **Temperature** parameter, in degrees Kelvin. (Also located in the Configuration block.)

See Also

Capacitor | Inductor

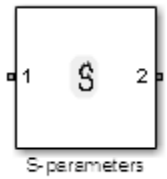
Purpose

Model S-parameter network

Library

Elements

Description

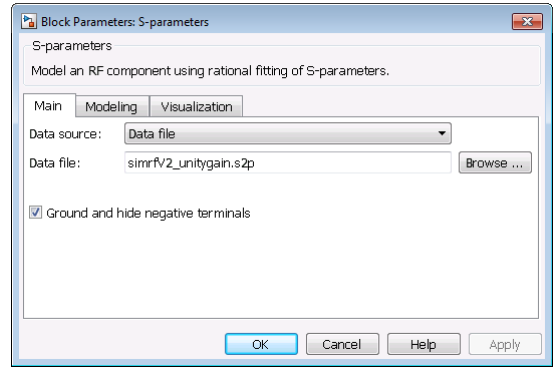


The S-Parameters block models a network defined by S-parameters in the SimRF circuit-envelope simulation environment. The device can have up to four ports. For an introduction to RF simulation, see the example, “Minimize Computations for RF Simulations”.

The block models S-parameter data in the SimRF environment by fitting a rational function to the specified data. For more information about rational fitting of S-parameters, see the RF Toolbox™ `rationalfit` function.

Dialog Box and Parameters

Main Tab



Data source

Specify the source of the data that describes the amplifier behavior. The data source must contain network parameters. The value can be `Data file`, `Network-parameters`, or `Rational model`. The default value of this parameter is `Data file`.

When you select `Rational model`, also select values for the **Residues**, **Poles**, and **Direct feedthrough** parameters corresponding to the equation for a rational model:

$$F(s) = \left(\sum_{k=1}^{n} \frac{C_k}{s - A_k} + D \right), \quad s = j2\pi f$$

In this equation, for a rational model $F(s)$, each C_k is the residue of the pole A_k with direct feedthrough D. The example, Model an RF Filter Using S-Parameter Data, shows how to use the RF Toolbox `rationalfit` function to create an `rfmodel.rational` object. This object has the properties `C`, `A`, and `D` that can be used to specify the **Residues**, **Poles**, and **Direct feedthrough** parameters.

Data file

When **Data source** is `Data file`, specify the name of the file that contains the S-parameter data. The file name must include the extension. If the file is not in your MATLAB® path, specify the full path to the file, or click the **Browse** button to locate the file. By default, this parameter specifies the data file `unitygain.s2p`.

Network parameter type

Specify the type of Network parameters: `S-parameters`, `Y-parameters`, or `Z-parameters`.

Network-parameters

When **Data source** is `Network-parameters`, specify a multidimensional array of network parameters. The third dimension of the S-parameter array must be the same length as the vector of frequencies specified by the **Frequency** parameter.

Frequency

Specify the frequency range of `Network-parameters`. Specify the units from the corresponding drop-down list.

Reference impedance (Ohms)

Specify the nonzero reference impedance of the S-parameter data. The default value is `50`.

Ground and hide negative terminals
Select this option to automatically ground and hide negative terminals in the model.

Modeling Tab

Modeling options
Modeling domain. Select `Frequency domain` or `Time domain (rationalfit)` from the drop-down list.

For the Amplifier and S-parameters blocks, the default value is `Time domain (rationalfit)`. For the Transmission Line block, the default value is `Frequency domain`.

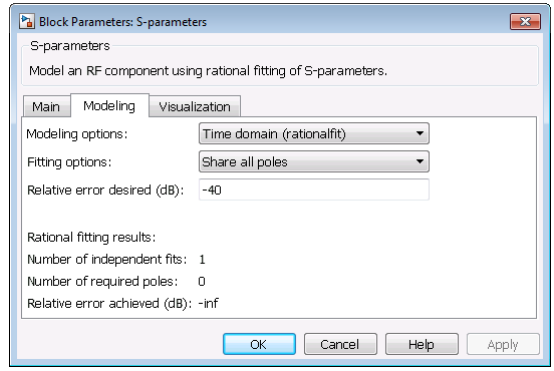
Fitting options
This field displays when you select `Time domain (rationalfit)` as the **Modeling options**. Valid values are `Share all poles`, `Share poles by columns`, and `Fit individually`.

For the Amplifier block, the default value is `Fit individually`. For the S-parameters block, the default value is `Share all poles`.

Relative error desired (dB)
This field displays when you select `Time domain (rationalfit)` as the **Modeling options**. Enter the desired relative error in decibels (dB). The default value is `-40`.

Rational fitting results
These fields display the results of rational fitting when you select `Time domain (rationalfit)` as the **Modeling options**.

Visualization Tab



For an example of using the Visualization tab, see Verify That the S-parameter Block Is Fitting a Rational Function with Acceptable Accuracy on page 1-73.

Source of frequency data

Frequency data source. When **Source of frequency data** is `Extracted from data source`, the **Data source** must be set to `Data file`. Verify that the specified **Data file** contains frequency data.

When **Source of frequency data** is `User-specified`, specify a vector of frequencies in the **Frequency data** parameter. Also, specify units from the corresponding drop-down list.

For the Amplifier and S-parameters blocks, the default value is `Extracted from source data`. For the Transmission Line block, the default value is `User-specified`.

Plot type

Specify the type of plot that you want to produce with your data. The **Plot type** parameter provides the following options:

- `X-Y plane` — Generate a Cartesian plot of your data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- `Polar plane` — Generate a polar plot of your data. The block plots only the range of data corresponding to the specified frequencies.
- `Z smith chart`, `Y smith chart`, and `ZY smith chart` — Generate a Smith chart. The block plots only the range of data corresponding to the specified frequencies.

The default value is `X-Y plane`.

Parameter #

Specify the S-parameters to plot. From the **Parameter1** and **Parameter2** drop-down lists, select the S-parameters that you want to plot. If you specify two parameters, the block plots both parameters in a single window.

The default value for **Parameter1** is `S11`. For the Amplifier and S-parameters blocks, the default value for **Parameter2** is `None`. For the Transmission Line block, the default value for **Parameter2** is `S22`.

Format #

For *X*-*Y* plots, format the units of the parameters to plot from the **Format1** and **Format2** drop-down lists. For polar plots and Smith charts, the formats are set automatically.

The default value is `Magnitude (decibels)`.

Y-axis scale

Scale for the *Y*-axis.

The default value is `Linear`.

X-axis scale

Scale for the *X*-axis.

The default value is `Linear`.

Examples

- "Architectural Design of a Low-IF Receiver System" uses an S-Parameters block to model a surface acoustic wave (SAW) filter.
- "Comparing Time- and Frequency-Domain Simulation Options for S-parameters" shows how to choose a selection for the **Modeling options** parameter based on the type of behavior you want to model in your system.

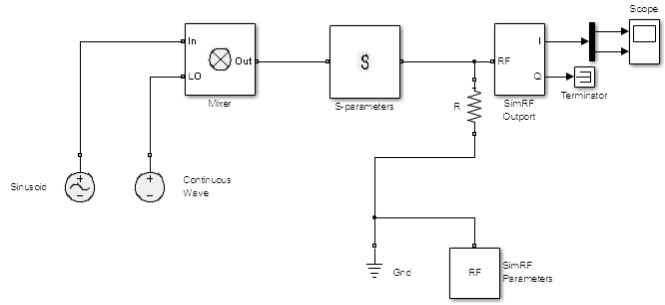
Model an RF Filter Using S-Parameter Data

The model

```
ex_simrf_sparams
```

replaces the LC lowpass tee filter in the model from the section, Model an RF Filter, with an S-parameters block. It reproduces the results of the original model. To recreate an S-parameter representation of the filter, the model uses MATLAB code in the `PreLoadFcn` callback to:

- Create an RF Toolbox `rfckt.lclowpasstee` object.
- Specify the filter coefficients used in the Model an RF Filter example.
- Analyze the filter object from 0 Hz to 2 GHz.
- Compute rational function coefficients using the RF Toolbox `rationalfit` function.

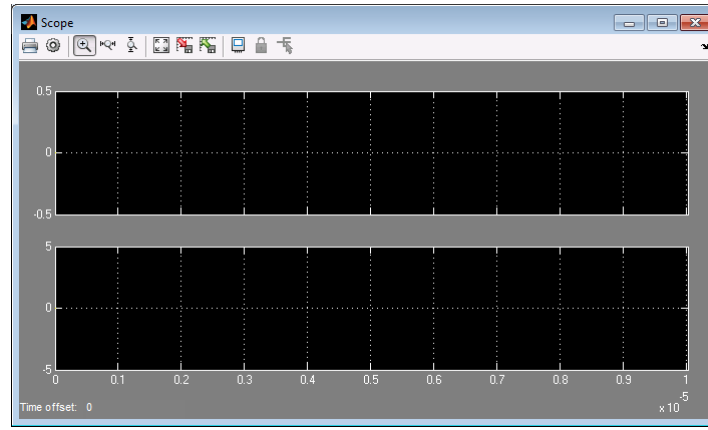


For information on using callbacks in SimRF models, see Model an RF Mixer.

To run the model:

1. Open the model. To do so, click the link, or, at the Command Window prompt, type the model name.
2. Select **Simulation > Run**.

To view the results of the simulation, double-click the scope.



The signals displayed in the scope match the results of the simulation in the Model an RF Filter section. The filter attenuates the high-frequency carrier and passes the low-frequency carrier. For more information on modeling RF components in with RF Toolbox objects, see Examples of Basic Operations with RF Toolbox Objects.

Model an RF Filter from Rational Function Coefficients

The model

```
ex_simrf_sparams
```

replaces the LC lowpass tee filter in the model from the section, Model an RF Filter, with an S-parameters block. It reproduces the results of the original model. To configure the S-parameters block in this model to simulate the same filter from rational function coefficients:

1. Set the **Data source** parameter to `Rational model`.
2. Set the **Residues** parameter to the variable `C`.
3. Set the **Poles** parameter to the variable `A`.
4. Set the **Direct feedthrough** parameter to the variable `D`.

To run the model, select **Simulation > Run**.

The `rfmodel.rational` object stores the residues, poles, and direct feedthrough data as vectors. However, because the S-parameters block requires matrices for the **Residues**, **Poles**, and **Direct feedthrough** parameters, rearrangement is necessary. To rearrange the vectors into matrices in the required format, the initialization function of the model transposes the output of the `reshape` function. For more information on rational function fitting with RF Toolbox software, see the `rationalfit` reference page.

Model an RF Filter from an S2P Data File

The model

```
ex_simrf_sparams
```

replaces the LC lowpass tee filter in the model from the section, Model an RF Filter, with an S-parameters block. It reproduces the results of the original model. To configure the S-parameters block in this model to simulate the same filter from an S2P data file:

1. Write the S-parameter data in this model to a data file. To do so, in the MATLAB Command Window, type:

   ```
   write(hfilt.AnalyzedResult,'teefilt.s2p')
   ```

2 Set the **Data source** parameter to `Data file`.

3 Set the **Data file** parameter to `teefilt.s2p`.

To run the model, select **Simulation > Run**. For more information on writing data files using RF Toolbox software, see Exporting Component Data to a File.

Verify That the S-parameter Block Is Fitting a Rational Function with Acceptable Accuracy

The S-Parameters block provides a visualization tool for validating RF-component data modeling. Use the visualization tool to verify that the S-parameter block is fitting a rational function with acceptable accuracy. To produce a data plot:

1 On the **Main** tab, specify the **Data source** parameter and requisite data to plot.

2 Specify frequency data:

- When **Source of frequency data** is `Extracted from data source`, the **Data source** must be set to `Data file`, and the specified **Data file** must contain frequency data.
- When **Source of frequency data** is `User-specified`, specify a vector of frequencies in the **Frequency data** parameter.

Also, specify units from the corresponding drop-down list.

3 Specify the type of plot that you want to produce with your data. The **Plot type** parameter provides the following options:

- `X-Y plane` — Generate a Cartesian plot of your data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- `Polar plane` — Generate a polar plot of your data. The block plots only the range of data corresponding to the specified frequencies.

- `Z smith chart`, `Y smith chart`, and `ZY smith chart` — Generate a Smith chart. The block plots only the range of data corresponding to the specified frequencies.

4. Specify the S-parameters to plot. From the **Parameter1** and **Parameter2** drop-down lists, select the S-parameters that you want to plot. If you specify two parameters, the block plots both parameters in a single window.

   For *X-Y* plots, format the units of the parameters to plot from the **Format1** and **Format2** drop-down lists. For polar plots and Smith charts, the formats are set automatically.

5. Click **Apply**.

6. Click **Plot**. The block plots your data and a rational function approximation.

See Also

Transmission Line | `rationalfit`

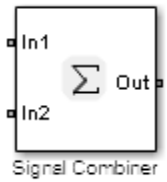
Purpose

Compute sum of RF signals

Library

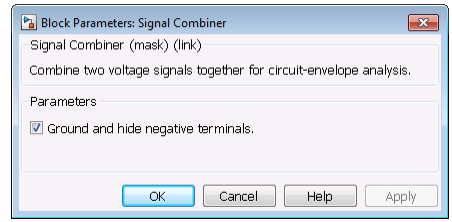
Elements

Description



Use the Signal Combiner block to sum signals across each carrier frequency in the SimRF circuit-envelope simulation environment. For more information about simulation RF systems, see Circuit Envelope Simulation.

Dialog Box and Parameters



Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

- The example, “Measuring Image Rejection Ratio in Receivers” shows how to use a signal combiner to perform image rejection.
- The example, Carrier to Interference Performance of a Weaver Receiver uses a Signal Combiner block as part of a realization of the Weaver receiver architecture.

Sinusoid

Purpose

Model DC offset and sinusoidal modulation

Library

Sources

Description

The Sinusoid block implements a voltage or current source that provides a DC offset and sine wave modulation. This block can be used with each listed block carrier in the circuit-envelope environment.

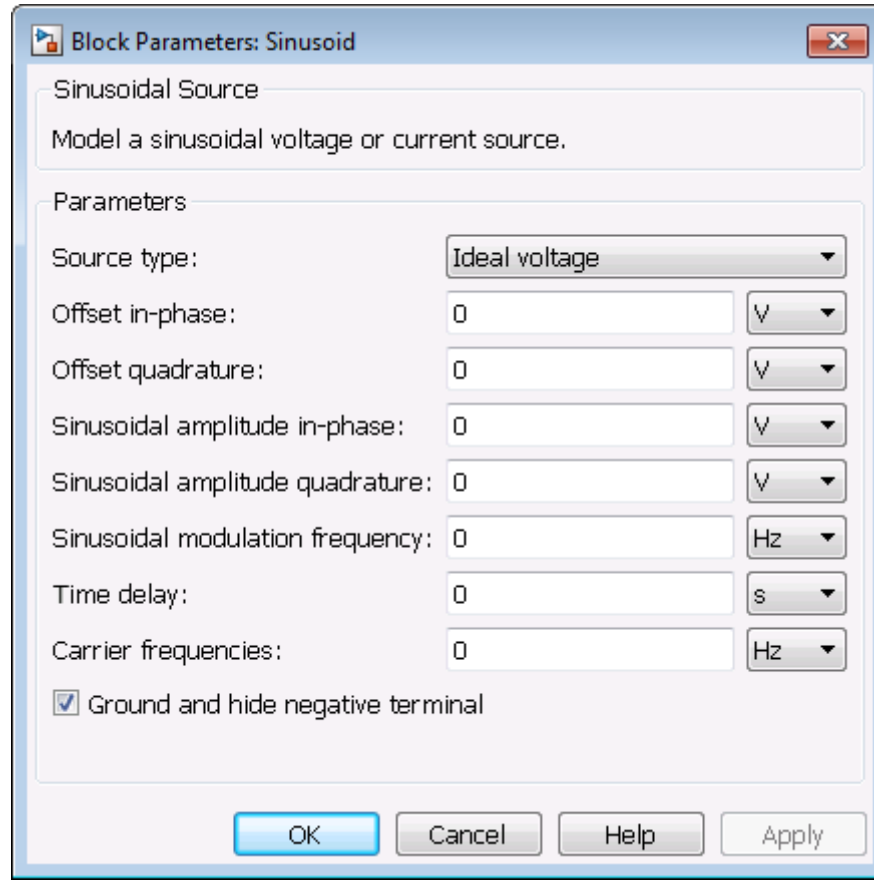
The block implements the following voltage (or current) relationships for the in-phase, (u_i), and quadrature, (u_q), components of the k^{th} listed block carrier:

$$u_{k,i}(t) = D_i + A_i \sin(w_k(t-\tau))$$
$$u_{k,q}(t) = D_q + A_q \sin(w_k(t-\tau))$$

where:

- D_i and D_q are DC offsets.
- A_i and A_q are in-phase and quadrature amplitudes.
- τ is the time delay.
- ω_k is the specified modulation frequency at a given carrier frequency f_k.
- t is the time.

Dialog Box and Parameters



Source type

Specify the wave type as:

- `Ideal Voltage` — The block simulates a voltage envelope $v(t)$ at the specified **Carrier frequencies**.
- `Ideal Current` — The block simulates a current envelope $i(t)$ at the specified **Carrier frequencies**.

The default value of this parameter is `Ideal Voltage`.

Offset in-phase

Specify the in-phase offset, D_i, as a scalar or vector. Specify the units of the offset from the corresponding drop-down list. The default value of this parameter is `0 V` for a voltage signal or `0 A` for a current signal.

Offset quadrature

Specify the quadrature offset, D_q, as a scalar or vector. Specify the units of the offset from the corresponding drop-down list. The default value of this parameter is `0 V` for a voltage signal or `0 A` for a current signal.

Sinusoidal amplitude in-phase

Specify the in-phase amplitude, A_i, as a scalar or vector. Specify the units of the offset from the corresponding drop-down list. The default value of this parameter is `0 V` for a voltage signal or `0 A` for a current signal.

Sinusoidal amplitude quadrature

Specify the quadrature amplitude, A_q, as a scalar or vector. Specify the units of the offset from the corresponding drop-down list. The default value of this parameter is `0 V` for a voltage signal or `0 A` for a current signal.

Sinusoidal modulation frequency

At each carrier f_k specified by the **Carrier frequencies** parameter, specify the modulation frequency m_k. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 Hz`.

Time delay

Specify the time delay, τ, of the modulation as a scalar or vector. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 s`.

Damping factor (1/s)

Specify the damping factor, ζ, of the modulation as a nonzero scalar or vector with nonzero elements. The default value of this parameter is `0` s^{-1}, representing an undamped sinusoid.

Carrier frequencies

Enter a vector of carrier frequencies whose elements are calculated from fundamental tones and corresponding harmonic in the Configuration block. Specify the units from the corresponding drop-down list. The default value of this parameter is `0 Hz`.

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

Examples

- The example "Validating IP2/IP3 Using Complex Signals" highlights the difference between sinusoidal SimRF and Simulink sources and shows how to use them in the same model.
- The section Model an RF Mixer uses a Sinusoid block to model an AM waveform at the input port of a mixer.

See Also

Continuous Wave

Three-Winding Transformer

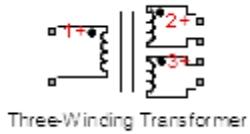
Purpose

Model three coupled inductors for circuit-envelope analysis

Library

Elements

Description



The Three-Winding Transformer block models three coupled inductors within the SimRF circuit-envelope simulation environment. For an introduction to RF simulation, see the example, "Minimize Computations for RF Simulations".

The block implements the relations

$$v_1(t) = L_1 \frac{d}{dt}[i_1(t)] + M_{12} \frac{d}{dt}[i_2(t)] + M_{13} \frac{d}{dt}[i_3(t)]$$

$$v_2(t) = M_{12} \frac{d}{dt}[i_1(t)] + L_2 \frac{d}{dt}[i_2(t)] + M_{23} \frac{d}{dt}[i_3(t)]$$

$$v_3(t) = M_{13} \frac{d}{dt}[i_1(t)] + M_{23} \frac{d}{dt}[i_2(t)] + L_3 \frac{d}{dt}[i_3(t)]$$

$$M_{pq} = K_{pq}\sqrt{L_p L_q}$$

where:

- L_1, L_2, and L_3 represent inductances.
- M_{pq} represents the mutual inductance between the pth and qth inductors, with coefficient of coupling K_{pq}.
- $v_1(t)$, $v_2(t)$, and $v_3(t)$ represent the voltage across the terminals of the inductors at time t.

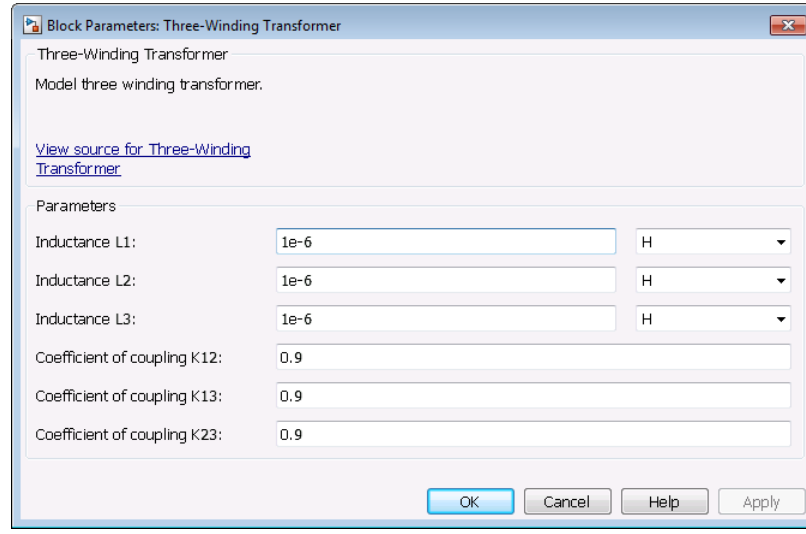
- $i_1(t)$, $i_2(t)$, and $i_3(t)$ represent the current through the inductors at time t. The block uses standard dot notation to indicate the direction of positive current flow relative to a positive voltage.

SimRF current and voltage signals consist of in-phase (I_k) and quadrature (Q_k) components at each frequency f_k specified in the Configuration block:

$$i(t) = \sum_{\{f_k\}} \left(i_{I_k}(t) + j \cdot i_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

$$v(t) = \sum_{\{f_k\}} \left(v_{I_k}(t) + j \cdot v_{Q_k}(t) \right) e^{j(2\pi f_k)t}$$

Dialog Box and Parameters



Inductance L1

Specify the inductance of the first inductor, L_1, as a scalar value greater than or equal to `0`. Specify the units of the inductance from the corresponding drop-down list. The default value of this parameter is `1e-6 H`.

Inductance L2

Specify the inductance of the second inductor, L_2, as a scalar value greater than or equal to `0`. Specify the units of the inductance from the corresponding drop-down list. The default value of this parameter is `1e-6 H`.

Inductance L3

Specify the inductance of the third inductor, L_3, as a scalar value greater than or equal to `0`. Specify the units of the inductance from the corresponding drop-down list. The default value of this parameter is `1e-6 H`.

Coefficient of coupling K12

Specify the coefficient of coupling for the mutual inductance of the first and second inductors, K_{12}, as a scalar value between `0` and `1`, inclusive. The default value of this parameter is `0.9`.

Coefficient of coupling K13

Specify the coefficient of coupling for the mutual inductance of the first and third inductors, K_{13}, as a scalar value between `0` and `1`, inclusive. The default value of this parameter is `0.9`.

Coefficient of coupling K23

Specify the coefficient of coupling for the mutual inductance of the second and third inductors, K_{23}, as a scalar value between `0` and `1`, inclusive. The default value of this parameter is `0.9`.

Note The minimum nonzero inductance value that the SimRF environment recognizes is `1e-18 H`. During simulation, the block uses a value of `1e-18 H` for any inductance and mutual inductance values specified between `0` and `1e-18 H`.

See Also

Inductor | Mutual Inductor

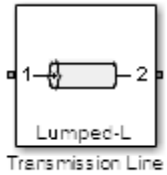
Purpose

Model transmission line

Library

Elements

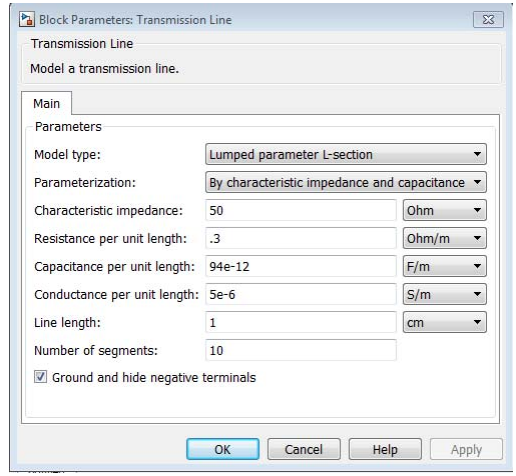
Description



The Transmission Line block models transmission lines. Lumped-element transmission line models are lossy and can only be simulated in the time domain. Distributed transmission models use an S-parameter representation. These models are simulated using either a time or frequency domain modeling option. For an introduction to RF simulation, see the example, “Minimize Computations for RF Simulations”.

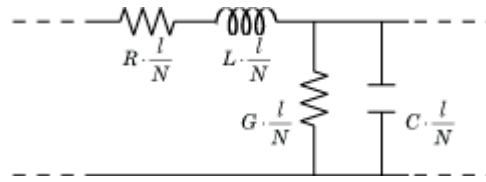
Dialog Box and Parameters

Main Tab



Model type

- **`Lumped parameter L-section`**
  Specify the transmission line as a number of RLGC L-sections.



When you select this option, the following dialog box parameters become available:

— Parameterization

Choose a parameterization for the segments. The default parameterization is `By characteristic impedance and capacitance`.

When you select `By characteristic impedance and capacitance`, these parameters appear:

Characteristic impedance

Specify the characteristic impedance of the transmission line. Specify the units from the corresponding drop-down list. The default value of this parameter is `50 Ohm`.

Resistance per unit length

Resistance per unit length, *R*. Specify the units from the corresponding drop-down list. The default value of this parameter is `.3 Ohm/m`.

Capacitance per unit length

Capacitance per unit length, *C*. Specify the units from the corresponding drop-down list. The default value of this parameter is `94e-12 F/m`.

Conductance per unit length

Conductance per unit length, *G*. Specify the units from the corresponding drop-down list. The default value of this parameter is `5e-6 S/m`.

Line length

Physical length of the transmission line, *l*. Specify the units from the corresponding drop-down list. The default length is `1 cm`.

Number of segments

Number of segments of the transmission line, *N*. The default number of segments is `10`.

When you select `By inductance and capacitance`, these parameters appear:

Resistance per unit length

Resistance per unit length, *R*. Specify the units from the corresponding drop-down list. The default value is `.3 Ohm/m`.

Inductance per unit length

Inductance per unit length, *G*. Specify the units from the corresponding drop-down list. The default value of this parameter is `235e-9 H/m`.

Capacitance per unit length

Capacitance per unit length, *C*. Specify the units from the corresponding drop-down list. The default value of this parameter is `94e-12 F/m`.

Conductance per unit length

Conductance per unit length, *G*. Specify the units from the corresponding drop-down list. The default value of this parameter is `5e-6 S/m`.

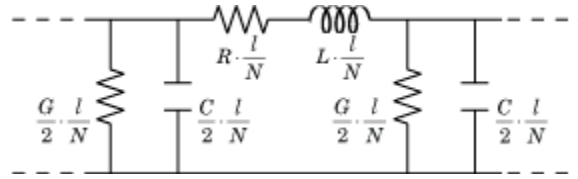
Line length

Physical length of the transmission line, *l*. Specify the units from the corresponding drop-down list. The default length is `1 cm`.

Number of segments

Number of segments of the transmission line, *N*. The default number of segments is `10`.

- **`Lumped parameter pi-section`**
  Specify the transmission line as a number of RLGC pi-sections.



When you select this option, the following dialog box parameters become available:

— Parameterization

Choose a parameterization for the segments. The default parameterization is `By characteristic impedance and capacitance`.

When you select `By characteristic impedance and capacitance`, these parameters appear:

Characteristic impedance

Specify the characteristic impedance of the transmission line. Specify the units from the corresponding drop-down list. The default value of this parameter is `50 Ohm`.

Resistance per unit length

Resistance per unit length, *R*. Specify the units from the corresponding drop-down list. The default value of this parameter is `.3 Ohm/m`.

Capacitance per unit length

Capacitance per unit length, *C*. Specify the units from the corresponding drop-down list. The default value of this parameter is `94e-12 F/m`.

Conductance per unit length

Conductance per unit length, *G*. Specify the units from the corresponding drop-down list. The default value of this parameter is `5e-6 S/m`.

Line length

Physical length of the transmission line, *l*. Specify the units from the corresponding drop-down list. The default length is `1 cm`.

Number of segments

Number of segments of the transmission line, *N*. The default number of segments is `10`.

When you select `By inductance and capacitance`, these parameters appear:

Resistance per unit length

Resistance per unit length, *R*. Specify the units from the corresponding drop-down list. The default value of this parameter is `.3 Ohm/m`.

Inductance per unit length

Inductance per unit length, *G*. Specify the units from the corresponding drop-down list. The default value of this parameter is `235e-9 H/m`.

Capacitance per unit length

Capacitance per unit length, *C*. Specify the units from the corresponding drop-down list. The default value of this parameter is `94e-12 F/m`.

Conductance per unit length

Conductance per unit length, *G*. Specify the units from the corresponding drop-down list. The default value of this parameter is `5e-6 S/m`.

Line length

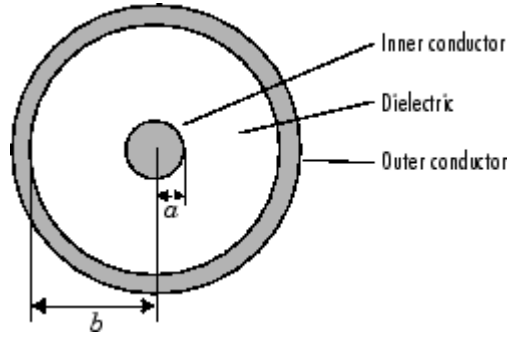
Physical length of the transmission line, *l*. Specify the units from the corresponding drop-down list. The default length is `1 cm`.

Number of segments

Number of segments of the transmission line, *N*. The default number of segments is `10`.

- **`Coaxial`**

  Specify the transmission line as a coaxial transmission line. A coaxial transmission line is shown in cross-section in the following figure. Its physical characteristics include the radius of the inner conductor, *a*, and the radius of the outer conductor *b*.

  When you select this option, the following dialog box parameters become available:

  — **Outer radius**

Radius of the outer conductor of the coaxial transmission line. The default value of this parameter is `2.57 mm`.

— Inner radius

Radius of the inner conductor of the coaxial transmission line. The default value of this parameter is `0.725 mm`.

— Relative permeability constant

Relative permeability of the dielectric. This value is expressed as the ratio of the permeability of the dielectric to permeability in free space, μ_0. The default value of this parameter is `1`.

— Relative permittivity constant

Relative permittivity of the dielectric. This value is expressed as the ratio of the permittivity of the dielectric to permittivity in free space, ε_0. The default value of this parameter is `2.2`.

— Loss Tangent of dielectric

Loss angle tangent of the dielectric. The default value of this parameter is `0`.

— Line length

Physical length of the transmission line. The default value of this parameter is `1 cm`.

— Conductivity of conductor

Conductivity of the conductor in siemens per meter. The default value of this parameter is `inf S/m`.

— Stub mode

Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

where

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

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

$$G = \frac{2\pi\omega\varepsilon''}{\ln\left(\frac{b}{a}\right)}$$

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

In these equations:

a is the radius of the inner conductor.

b is the radius of the outer conductor.

σ_{cond} is the conductivity in the conductor.

μ is the permeability of the dielectric.

ε is the permittivity of the dielectric.

ε'' is the imaginary part of ε, $\varepsilon'' = \varepsilon_0\varepsilon_r\tan\delta$, where:

ε_0 is the permittivity of free space.

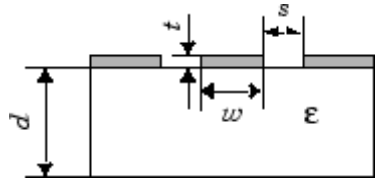
ε_r is the **Relative permittivity constant** parameter value.

tan δ is the **Loss tangent of dielectric** parameter value.

δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.

f is a vector of internal modeling frequencies.

- **`Coplanar waveguide`**
  Specify the transmission line as a coplanar waveguide. A coplanar waveguide transmission line is shown in cross-section in the following figure. Its physical characteristics include the conductor width, w, the conductor thickness, t, the slot width, s, the substrate height, d, and the relative permittivity constant, ε.



When you select this option, the following dialog box parameters become available:

— Conductor width

Physical width of the conductor. The default value of this parameter is `0.6 mm`.

— Slot width

Physical width of the slot. The default value of this parameter is `0.2 mm`.

— Substrate height

Thickness of the dielectric on which the conductor resides. The default value of this parameter is `0.635 mm`.

— Strip thickness

Physical thickness of the conductor. The default value of this parameter is `5 um`.

— Relative permittivity constant

Relative permittivity of the dielectric. This value is expressed as the ratio of the permittivity of the dielectric to permittivity in free space, ε_0. The default value of this parameter is `2.2`.

— Loss Tangent of dielectric

Loss angle tangent of the dielectric. The default value of this parameter is `1`.

— Line length

Physical length of the transmission line. The default value of this parameter is `1 cm`.

— Conductivity of conductor

Conductivity of the conductor in siemens per meter. The default value of this parameter is `inf S/m`.

— Stub mode

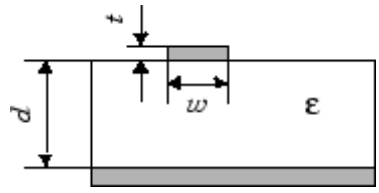
Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

Z_0 and k are vectors whose elements correspond to the elements of f, a vector of internal modeling frequencies. Both vectors can be expressed in terms of the specified conductor strip width, slot width, substrate height, conductor strip

thickness, relative permittivity constant, conductivity and dielectric loss tangent of the transmission line, as described in [3].

- **`Microstrip`**
  Specify the transmission line as a microstrip transmission line. A microstrip transmission line is shown in cross-section in the following figure. Its physical characteristics include the microstrip width, w, the microstrip thickness, t, the substrate height, d, and the relative permittivity constant, ε.



When you select this option, the following dialog box parameters become available:

— Strip width

Width of the microstrip transmission line. The default value of this parameter is `0.6 mm`.

— Substrate height

Thickness of the dielectric on which the microstrip resides. The default value of this parameter is `0.635 mm`.

— Strip thickness

Physical thickness of the microstrip. The default value of this parameter is `5 um`.

— 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. The default value of this parameter is `2.2`.

- **Loss Tangent of dielectric**

  Loss angle tangent of the dielectric. The default value of this parameter is `0`.

- **Line length**

  Physical length of the transmission line. The default value of this parameter is `1 cm`.

- **Conductivity of conductor**

  Conductivity of the conductor in siemens per meter. The default value of this parameter is `inf S/m`.

- **Stub mode**

  Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

  The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

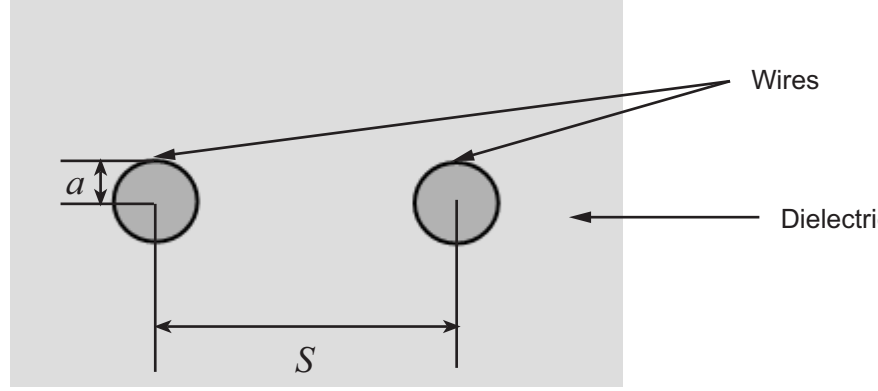
  Z_0 and k are vectors whose elements correspond to a vector of internal modeling frequencies. Both vectors can be expressed in terms of the specified conductor strip width, substrate height, conductor strip thickness, relative permittivity constant, conductivity, and dielectric loss tangent of the microstrip line.

-

`Two-wire`

Specify the transmission line as two-wire transmission line. A two-wire transmission line is shown in cross-section in the following figure. Its physical characteristics include the radius

of the wires, a, the separation or physical distance between the wire centers, S, and the relative permittivity and permeability of the wires. [2] SimRF Equivalent Baseband software assumes the relative permittivity and permeability are uniform.



When you select this option, the following dialog box parameters become available:

- **Wire radius**

  Radius of the conducting wires of the two-wire transmission line. The default value of this parameter is `0.67 mm`.

- **Wire separation**

  Physical distance between the wires. The default value of this parameter is `1.62 mm`.

- **Relative permeability constant**

  Relative permeability of the dielectric. This value is expressed as the ratio of the permeability of the dielectric to permeability in free space, μ_0. The default value of this parameter is `1`.

- **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. The default value of this parameter is `2.2`.

— Loss Tangent of dielectric

Loss angle tangent of the dielectric. The default value of this parameter is `0`.

— Line length

Physical length of the transmission line. The default value of this parameter is `1 cm`.

— Conductivity of conductor

Conductivity of the conductor in siemens per meter. The default value of this parameter is `inf S/m`.

— Stub mode

Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

where

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

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

$$G = \frac{\pi \omega \varepsilon''}{\operatorname{a\,cosh}\left(\frac{D}{2a}\right)}$$

$$C = \frac{\pi \varepsilon}{\operatorname{a\,cosh}\left(\frac{D}{2a}\right)}$$

and $\omega = 2\pi f$.

In these equations:

σ_{cond} is the conductivity in the conductor.

μ is the permeability of the dielectric.

ε is the permittivity of the dielectric.

ε'' is the imaginary part of ε, $\varepsilon'' = \varepsilon_0 \varepsilon_r \tan \delta$, where:

ε_0 is the permittivity of free space.

ε_r is the **Relative permittivity constant** parameter value.

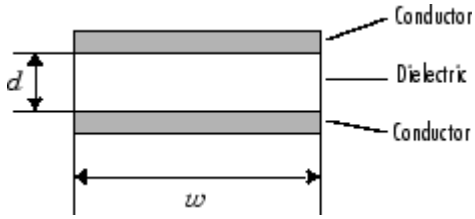
$\tan \delta$ is the **Loss tangent of dielectric** parameter value.

δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.

f is a vector of modeling frequencies determined by the Output Port block.

•

`Parallel plate`
Specify the transmission line as a parallel-plate transmission line. A parallel-plate transmission line is shown in cross-section in the following figure. Its physical characteristics include the plate width, w, and the plate separation, d. [2]



When you select this option, the following dialog box parameters become available:

- **Plate width**

  Physical width of the parallel-plate transmission line. The default value of this parameter is `5 mm`.

- **Plate separation**

  Thickness of the dielectric separating the plates. The default value of this parameter is `1 mm`.

- **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. The default value of this parameter is `1`.

- **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. The default value of this parameter is `2.2`.

- **Loss Tangent of dielectric**

Loss angle tangent of the dielectric. The default value of this parameter is `0`.

— Line length

Physical length of the transmission line. The default value of this parameter is `1 cm`.

— Conductivity of conductor

Conductivity of the conductor. The default value of this parameter is `inf mS/m`.

— Stub mode

Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

where

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

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

$$G = \omega\varepsilon''\frac{w}{d}$$

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

In these equations:

σ_{cond} is the conductivity in the conductor.

μ is the permeability of the dielectric.

ε is the permittivity of the dielectric.

ε'' is the imaginary part of ε, $\varepsilon'' = \varepsilon_0\varepsilon_r\tan\delta$, where:

ε_0 is the permittivity of free space.

ε_r is the **Relative permittivity constant** parameter value.

$\tan\delta$ is the **Loss tangent of dielectric** parameter value.

δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f\mu\sigma_{cond}}$.

f is a vector of modeling frequencies determined by the Output Port block.

- **`Equation-based`**
  Specify the transmission line as an equation-based transmission line. The transmission line, which can be lossy or lossless, is treated as a two-port linear network.

When you select this option, the following dialog box parameters become available:

- **Phase velocity (m/s**

  Propagation velocity of a uniform plane wave on the transmission line. The default value of this parameter is `299792458`.

- **Characteristic impedance**

  Characteristic impedance of the transmission line. The value can be complex. The default value of this parameter is `50 Ohm`.

- **Loss (dB/m)**

  Reduction in strength of the signal as it travels over the transmission line. This value must be positive. The default value of this parameter is `0`.

- **Line length**

  Physical length of the transmission line. The default value of this parameter is `1 cm`.

- **Frequency**

  Vector of modeling frequencies. The block performs the calculations listed in the Description section at each frequency you provide. The default value of this parameter is `1e9 Hz`.

- **Interpolation method**

  Specify the interpolation method the block uses to calculate the parameter values at the modeling frequencies. Your choices are `Linear`, `Spline`, or `Cubic`. The default value of this parameter is `Linear`.

- **Stub mode**

Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

Z_0 is the specified characteristic impedance. k is a vector whose elements correspond to the elements of the input vector, `freq`. The block calculates k from the specified parameters as $k = \alpha_a + i\beta$, where α_a is the attenuation coefficient and β is the wave number. The attenuation coefficient α_a is related to the specified loss, α, by

$$\alpha_a = -\ln\left(10^{\alpha/20}\right)$$

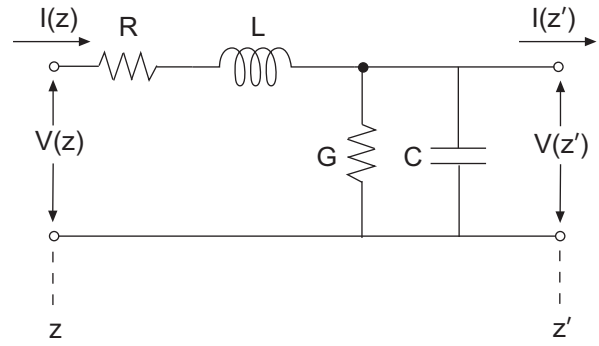
The wave number β is related to the specified phase velocity, V_p, by

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

The phase velocity V_P is also known as the *wave propagation velocity*.

- **RLCG**

  Specify the transmission line as an RLCG transmission line. This line is 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 $z' = z + \Delta z$.

When you select this option, the following dialog box parameters become available:

- **Resistance per unit length**

  Vector of resistance values in ohms per meter. The default value of this parameter is `0.3 Ohm/m`.

- **Inductance per unit length**

  Vector of inductance values in henries per meter. The default value of this parameter is `235e-9 H/m`.

- **Capacitance per unit length**

  Vector of capacitance values in farads per meter. The default value of this parameter is `94e-12 F/m`.

- **Conductance per unit length**

  Vector of conductance values in siemens per meter. The default value of this parameter is `5e-6 S/m`.

- **Line length**

  Physical length of the transmission line. The default value of this parameter is `1 cm`.

- **Frequency**

  Vector of frequency values at which the resistance, inductance, capacitance, and conductance values are known. The default value of this parameter is `1e9 Hz`.

- **Interpolation method**

  Specify the interpolation method the block uses to calculate the parameter values at the modeling frequencies. Your choices are `Linear`, `Spline`, or `Cubic`. The default value of this parameter is `Linear`.

- **Stub mode**

  Type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See Transmission Line with Stub on page 105 for more information.

  The following auxiliary equations are used for ABCD Parameter Calculations. For general information, see ABCD Parameter Calculations for a Transmission Line on page 105

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

Ground and hide negative terminals

Select this check box to internally ground and hide the negative terminals. Clear the check box to expose the negative terminals. By exposing these terminals, you can connect them to other parts of your model.

By default, this check box is selected.

ABCD Parameter Calculations for a Transmission Line

When modeling a transmission line, the block first calculates the ABCD-parameters at at a set of internal frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

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

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

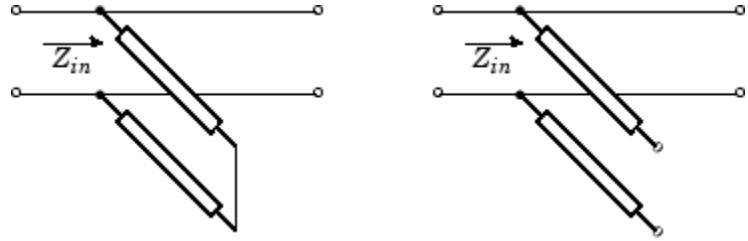
$$B = \frac{Z_0 * \left(e^{kd} - e^{-kd}\right)}{2}$$

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

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

Transmission Line with Stub

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



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

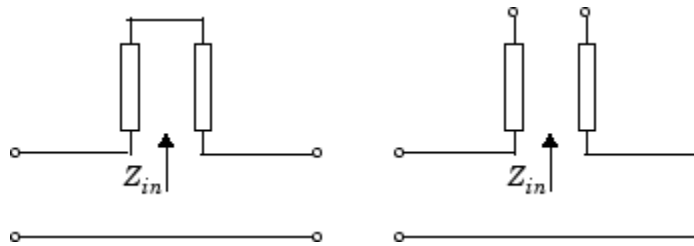
$A = 1$

$B = 0$

$C = 1 / Z_{in}$

$D = 1$

When you set the **Stub mode** parameter in the mask dialog box to `Series`, the two-port network comprises a series transmission line. You can terminate this line 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

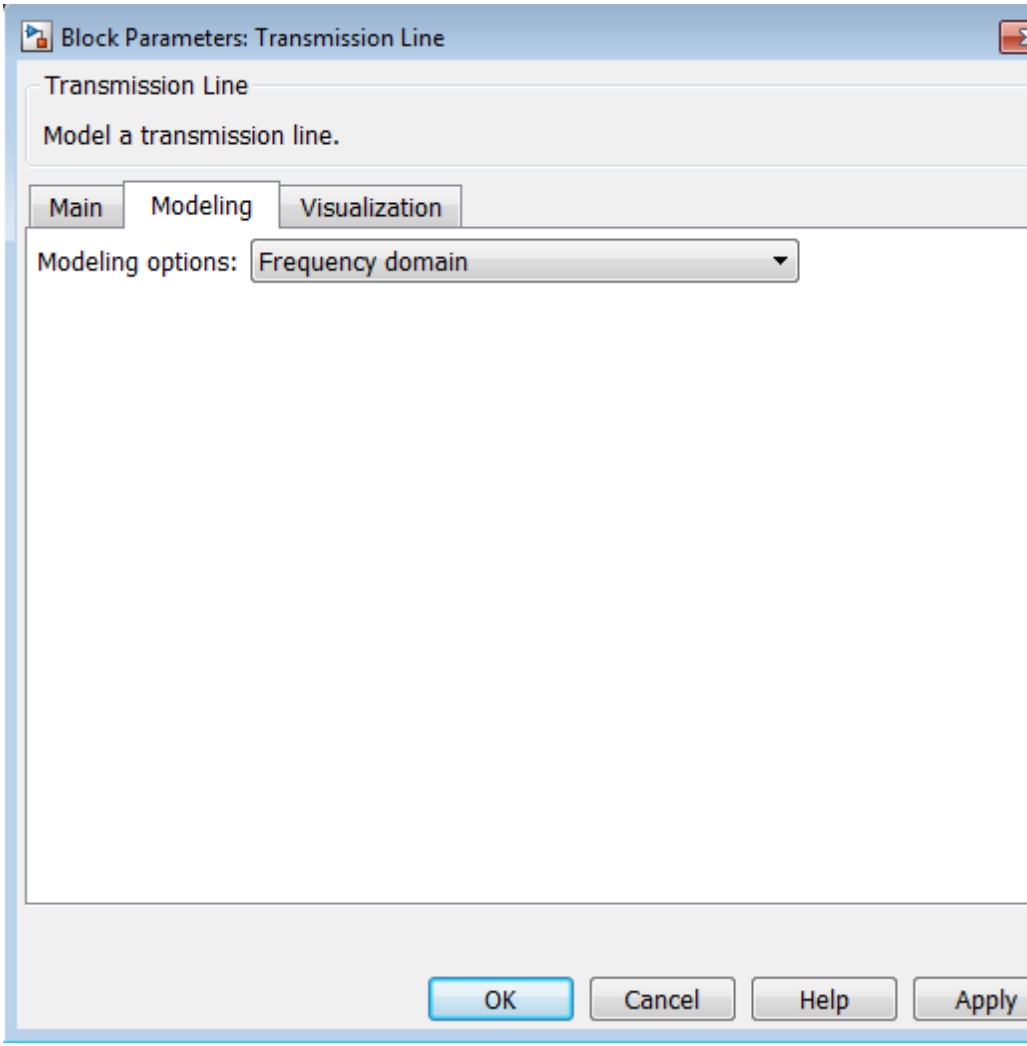
$A = 1$

$B = Z_{in}$

$C = 0$

$D = 1$

Modeling Tab



Modeling options

Modeling domain. Select `Frequency domain` or `Time domain (rationalfit)` from the drop-down list.

For the Amplifier and S-parameters blocks, the default value is `Time domain (rationalfit)`. For the Transmission Line block, the default value is `Frequency domain`.

Fitting options

This field displays when you select `Time domain (rationalfit)` as the **Modeling options**. Valid values are `Share all poles`, `Share poles by columns`, and `Fit individually`.

For the Amplifier block, the default value is `Fit individually`. For the S-parameters block, the default value is `Share all poles`.

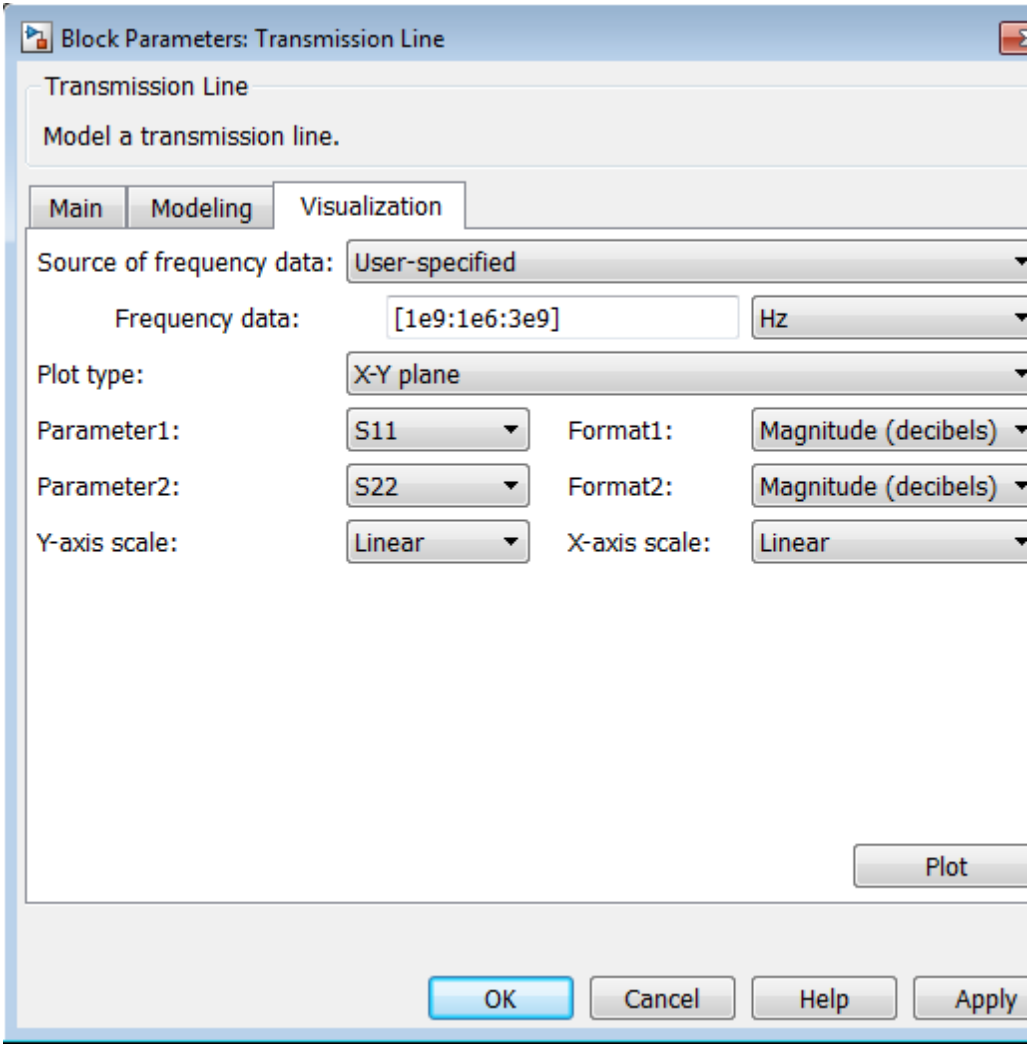
Relative error desired (dB)

This field displays when you select `Time domain (rationalfit)` as the **Modeling options**. Enter the desired relative error in decibels (dB). The default value is `-40`.

Rational fitting results

These fields display the results of rational fitting when you select `Time domain (rationalfit)` as the **Modeling options**.

Visualization Tab



Source of frequency data

Frequency data source. When **Source of frequency data** is `Extracted from data source`, the **Data source** must be set to `Data file`. Verify that the specified **Data file** contains frequency data.

When **Source of frequency data** is `User-specified`, specify a vector of frequencies in the **Frequency data** parameter. Also, specify units from the corresponding drop-down list.

For the Amplifier and S-parameters blocks, the default value is `Extracted from source data`. For the Transmission Line block, the default value is `User-specified`.

Plot type

Specify the type of plot that you want to produce with your data. The **Plot type** parameter provides the following options:

- `X-Y plane` — Generate a Cartesian plot of your data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- `Polar plane` — Generate a polar plot of your data. The block plots only the range of data corresponding to the specified frequencies.
- `Z smith chart`, `Y smith chart`, and `ZY smith chart` — Generate a Smith chart. The block plots only the range of data corresponding to the specified frequencies.

The default value is `X-Y plane`.

Parameter #

Specify the S-parameters to plot. From the **Parameter1** and **Parameter2** drop-down lists, select the S-parameters that you want to plot. If you specify two parameters, the block plots both parameters in a single window.

The default value for **Parameter1** is `S11`. For the Amplifier and S-parameters blocks, the default value for **Parameter2** is `None`. For the Transmission Line block, the default value for **Parameter2** is `S22`.

Format #

For *X-Y* plots, format the units of the parameters to plot from the **Format1** and **Format2** drop-down lists. For polar plots and Smith charts, the formats are set automatically.

The default value is `Magnitude (decibels)`.

Y-axis scale

Scale for the *Y*-axis.

The default value is `Linear`.

X-axis scale

Scale for the *X*-axis.

The default value is `Linear`.

Tips

- In general, blocks that model delay effects rely on signal history. You can minimize numerical error for these blocks that occurs because of a lack of signal history at the start of a simulation. To do so, in the Configuration Parameters dialog box Solver pane you can specify an **Initial step size**. For models with delay-based Transmission Line blocks, use an initial step size that is less than the value of the **Delay** parameter.

References

[1] Sussman-Fort, S. E., and J. C. Hantgan. "SPICE Implementation of Lossy Transmission Line and Schottky Diode Models." *IEEE Transactions on Microwave Theory and Techniques*.Vol. 36, No.1, January 1988.

[2] Pozar, David M., *Microwave Engineering*. Hoboken, NJ: John Wiley & Sons, Inc., 2005.

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

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

[5] True, Kenneth M. "Data Transmission Lines and Their Characteristics." *National Semiconductor Application Note 806*, April 1992.

See Also

S-Parameters

2

Blocks — Alphabetical List

Amplifier (Idealized Baseband)

Purpose

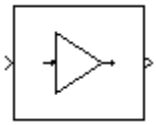
Complex baseband model of amplifier with noise

Library

Mathematical

Note To use this block, you must install DSP System Toolbox™ software. For more information, see the SimRF release notes.

Description



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

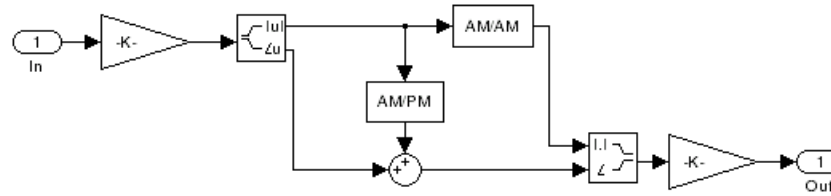
Note This block assumes a nominal impedance of 1 ohm.

Modeling Nonlinearity

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

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

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



Application of Nonlinearity

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

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

AM/AM and AM/PM Conversions

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

1. Right-click the Amplifier block.

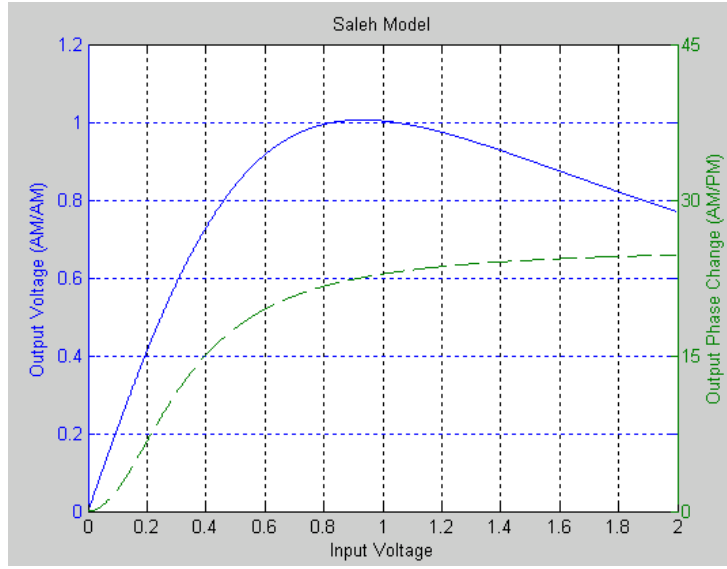
2 Select **Look under mask** in the pop-up menu. This displays the block's configuration underneath the mask. The block contains five subsystems corresponding to the five nonlinearity methods.

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

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

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

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



Model Parameters and Characteristics of Nonlinearity Modeling Methods

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

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

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

Cubic Polynomial Model

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

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

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

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

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

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

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

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

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

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

respectively.

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

Hyperbolic Tangent Model

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

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

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

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

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

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

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

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

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

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

respectively.

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

Saleh Model

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

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

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

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

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

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

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

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

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

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

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:

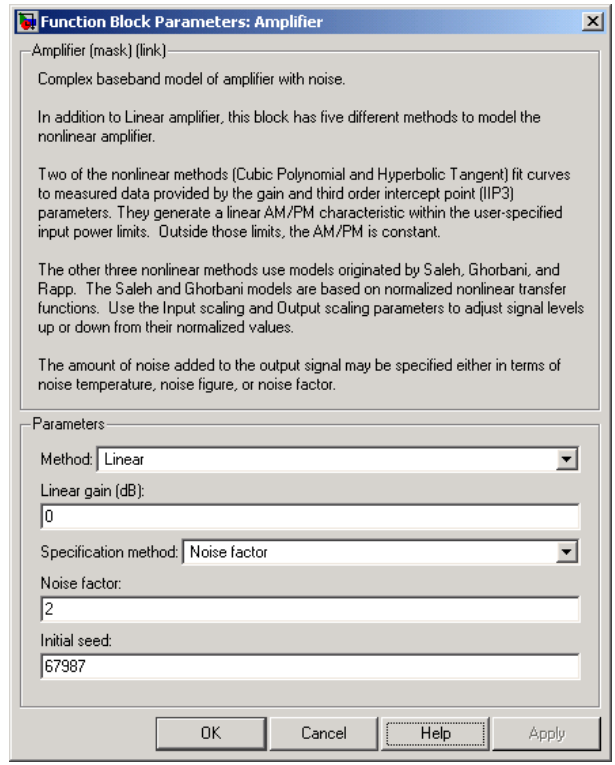
$$\text{Noise factor} = 1 + \frac{\text{Noise temperature}}{290}$$

- `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.

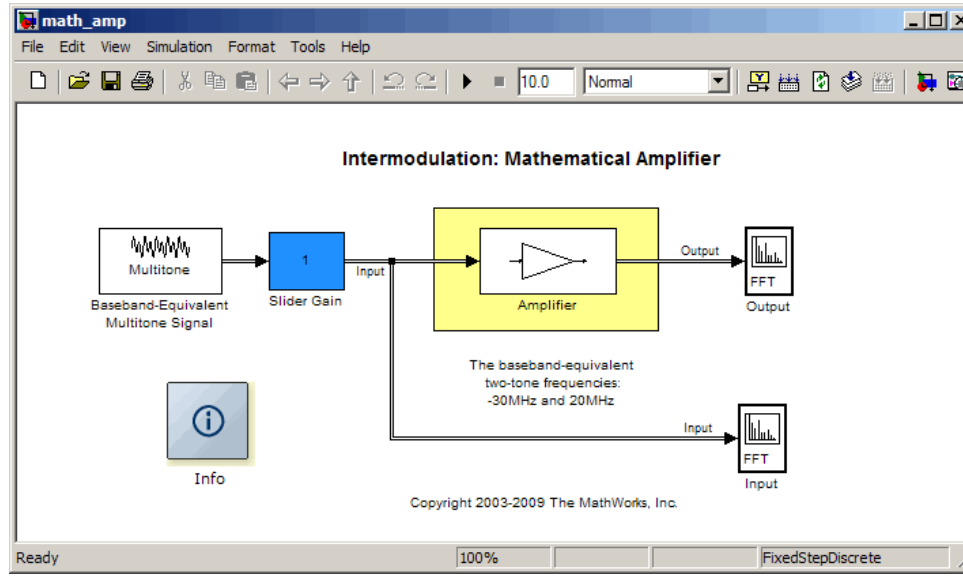
Noise figure (dB)
Scalar specifying the amount of noise relative to a noise temperature of 290 kelvin. A `Noise figure` setting of 0 decibels indicates a noiseless system. This field becomes visible if you select `Noise figure` as the **Specification method** parameter. Tunable.

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

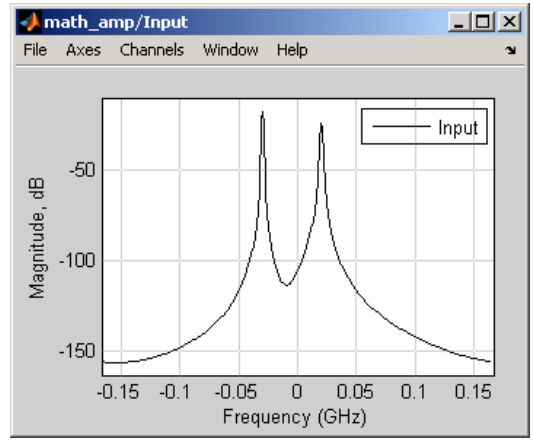
Initial seed
Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise.

Examples

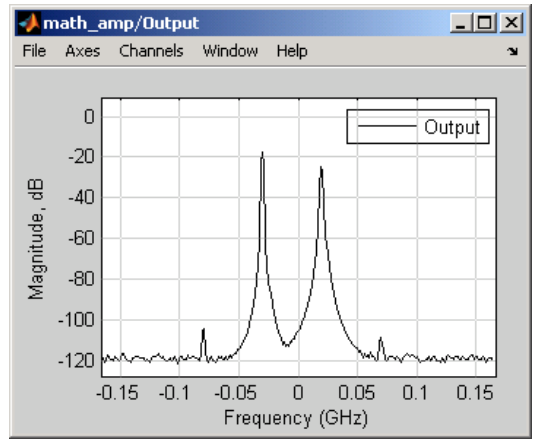
You can see the effect of the Amplifier block in the example Intermodulation: Mathematical Amplifier.



This example 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 example uses the default Amplifier block **IIP3 (dBm)** value of 30. It uses no AM/PM conversion. The example specifies thermal noise as `Noise figure`, for which it uses the default 3.01 dB.



References

[1] Ghorbani, A. and M. Sheikhan, "The Effect of Solid State Power Amplifiers (SSPAs) Nonlinearities on MPSK and M-QAM Signal Transmission," *Sixth Int'l Conference on Digital Processing of Signals in Comm.*, 1991, pp. 193-197.

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

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

See Also

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

Bandpass RF Filter

Purpose

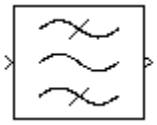
Standard bandpass RF filters in baseband-equivalent complex form

Library

Mathematical

Note To use this block, you must install DSP System Toolbox software. For more information, see the SimRF release notes.

Description



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

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

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

Note This block assumes a nominal impedance of 1 ohm.

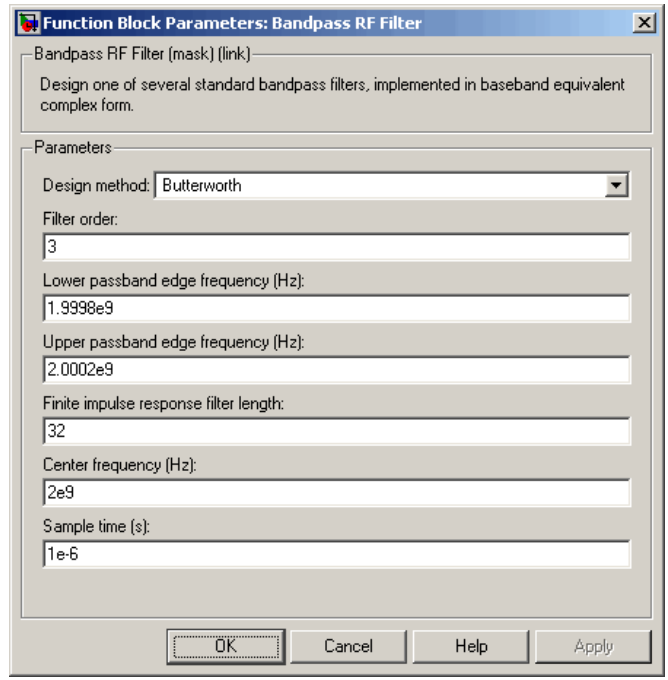
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

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

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

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

Dialog Box



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

You can change tunable parameters while the model is running.

Design method

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

Filter order

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

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

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

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

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

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

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

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

Center frequency (Hz)
Center of the modeling frequencies.

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

See Also

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

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

Bandstop RF Filter

Purpose

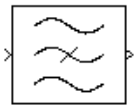
Standard bandstop RF filters in baseband-equivalent complex form

Library

Mathematical

Note To use this block, you must install DSP System Toolbox software. For more information, see the SimRF release notes.

Description



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

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

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

Note This block assumes a nominal impedance of 1 ohm.

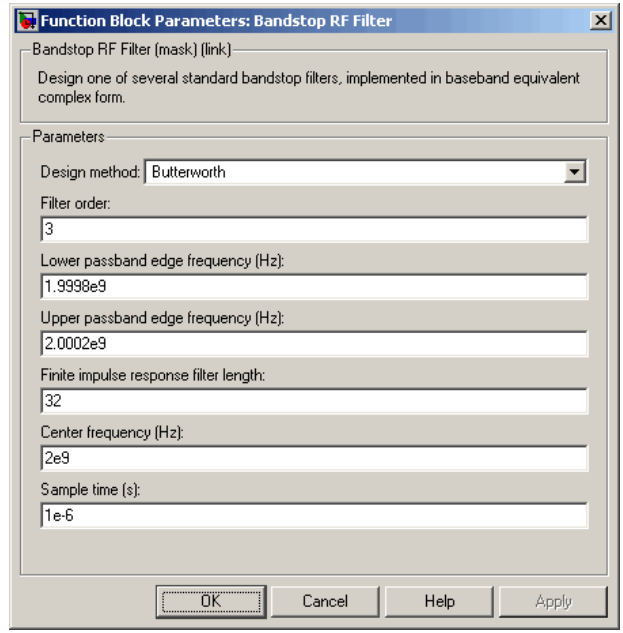
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

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

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

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

Dialog Box



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

You can change tunable parameters while the model is running.

Design method

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

Filter order

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

Lower passband edge frequency (Hz)

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

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

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

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

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

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

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

Center frequency (Hz)
Center of the modeling frequencies.

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

See Also

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

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

Coaxial Transmission Line

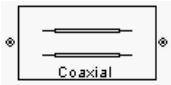
Purpose

Model coaxial transmission line

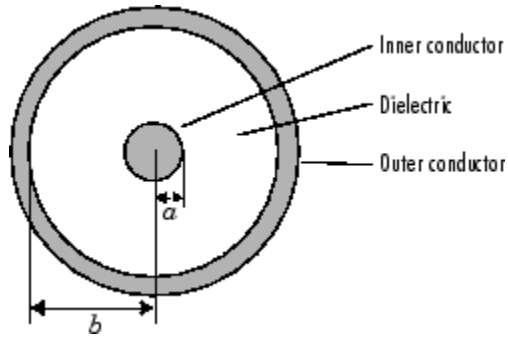
Library

Transmission Lines sublibrary of the Physical library

Description



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



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

Stubless Transmission Line

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

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

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

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

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

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

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

where

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

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

$$G = \frac{2\pi\omega\varepsilon''}{\ln\left(\frac{b}{a}\right)}$$

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

In these equations:

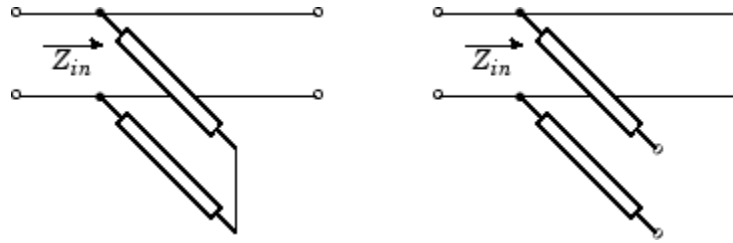
- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- σ_{cond} is the conductivity in the conductor.
- μ is the permeability of the dielectric.
- ε is the permittivity of the dielectric.
- ε'' is the imaginary part of ε, $\varepsilon'' = \varepsilon_0\varepsilon_r \tan\delta$, where:
  - ε_0 is the permittivity of free space.
  - ε_r is the **Relative permittivity constant** parameter value.
  - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.
- f is a vector of modeling frequencies determined by the Output Port block.

Shunt and Series Stubs

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

Shunt ABCD-Parameters

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

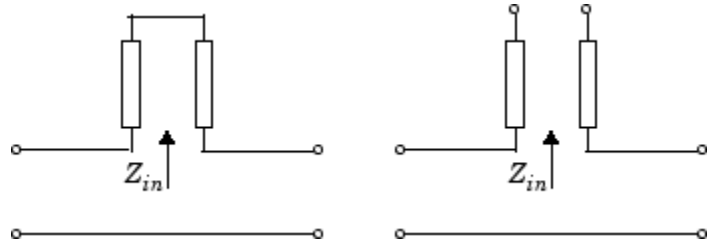


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

Series ABCD-Parameters

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

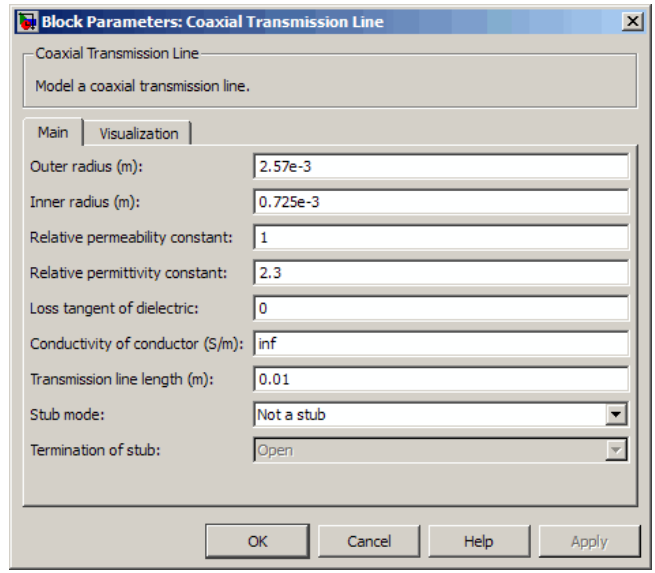


Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

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

Dialog Box

Main Tab



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

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

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

Loss tangent of dielectric
Loss angle tangent of the dielectric.

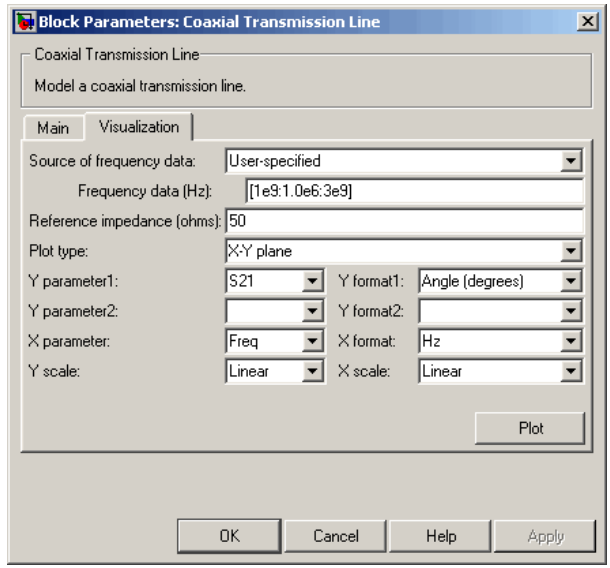
Conductivity of conductor (S/m)
Conductivity of the conductor in siemens per meter.

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

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

References

[1] Pozar, David M. *Microwave Engineering*, John Wiley & Sons, Inc., 2005.

See Also

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

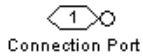
Purpose

Connection port for RF subsystem

Library

Input/Output Ports sublibrary of the Physical library

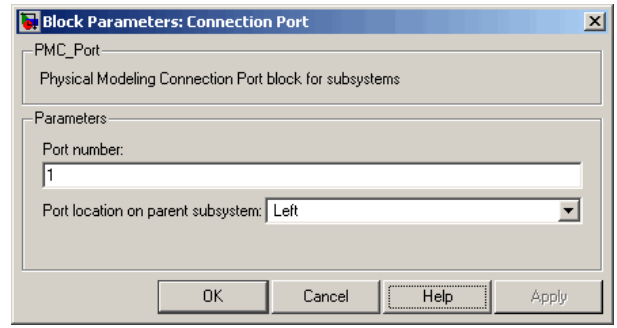
Description



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

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

Dialog Box



Port number

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

Port location on parent subsystem

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

See Also

See Creating Subsystems in the Simulink documentation.

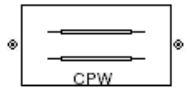
Purpose

Model coplanar waveguide transmission line

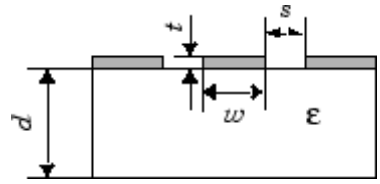
Library

Transmission Lines sublibrary of the Physical library

Description



The Coplanar Waveguide Transmission Line block models the coplanar waveguide transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A coplanar waveguide transmission line is shown in cross-section in the following figure. Its physical characteristics include the conductor width (w), the conductor thickness (t), the slot width (s), the substrate height (d), and the relative permittivity constant (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 first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

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

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

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

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

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

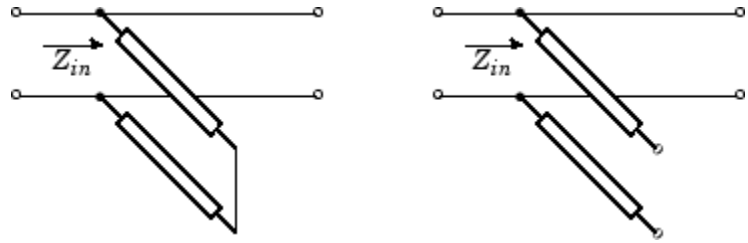
Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the specified conductor strip width, slot width, substrate height, conductor strip thickness, relative permittivity constant, conductivity and dielectric loss tangent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

Shunt and Series Stubs

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

Shunt ABCD-Parameters

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

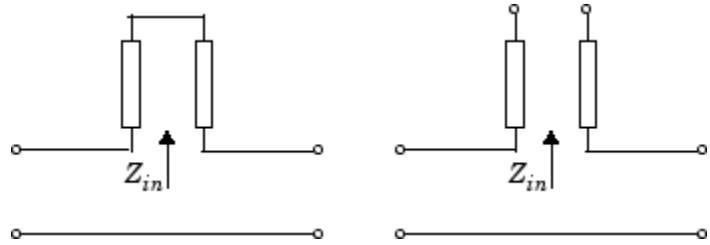


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

Series ABCD-Parameters

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

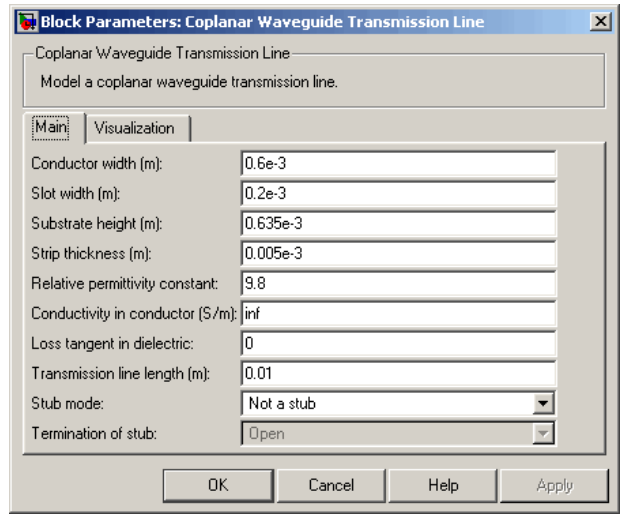


Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

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

Dialog Box

Main Tab



Conductor width (m)
Physical width of the conductor.

Slot width (m)
Physical width of the slot.

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

Strip thickness (m)
Physical thickness of the conductor.

Relative permittivity constant
Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space ε_0.

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

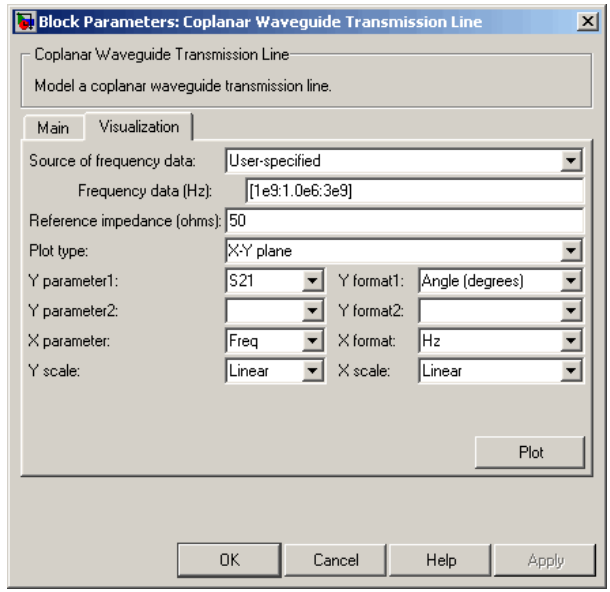
Loss tangent of dielectric
Loss angle tangent of the dielectric.

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

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

References

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

See Also

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

Purpose

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

Library

Amplifiers sublibrary of the Physical library

Description

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

Network Parameters

If network parameter data and corresponding frequencies exist as S-parameters in the data source, the General Amplifier block interpolates the S-parameters to determine their values at the modeling frequencies. If the network parameters are Y- or Z-parameters, the block first converts them to S-parameters. For more information, see "SimRF Equivalent Baseband Algorithms".

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from the power data.

If the data source contains no power data, then you can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the General Amplifier block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a linear amplifier. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1 The block checks whether you have specified a value other than `Inf` for:

- The third-order intercept point ($OIP3$ or $IIP3$).
- The output power at the 1-dB compression point ($P_{1dB,out}$).
- The output power at saturation ($P_{sat,out}$).

In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2 The block calculates a corresponding input or output value for the parameters that you have specified. In units of dB and dBm,

$$\begin{aligned} P_{sat,out} + GC_{sat} &= P_{sat,in} + G_{lin} \\ P_{1dB,out} + 1 &= P_{1dB,in} + G_{lin} \\ OIP3 &= IIP3 + G_{lin} \end{aligned}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations is equal to the number of parameters that you provide. For example, if you specify all three parameters, the block formulates the coefficients according to the following equations:

$$\begin{aligned} \sqrt{P_{sat,out}} &= c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7 \\ \sqrt{P_{1dB,out}} &= c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7 \\ 0 &= \frac{c_1}{IIP3} + c_3 \end{aligned}$$

The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Active Noise

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

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

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

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

Operating Conditions

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

Data Consistency

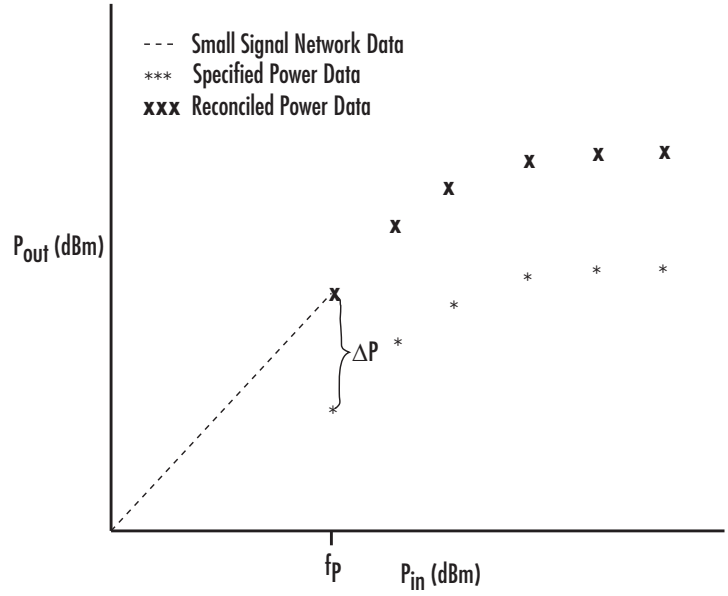
If the data source is a MathWorks™ AMP file or an Agilent S2D file that contains both network parameter data and power data, the blockset checks the data for consistency and reconciles it as necessary.

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

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

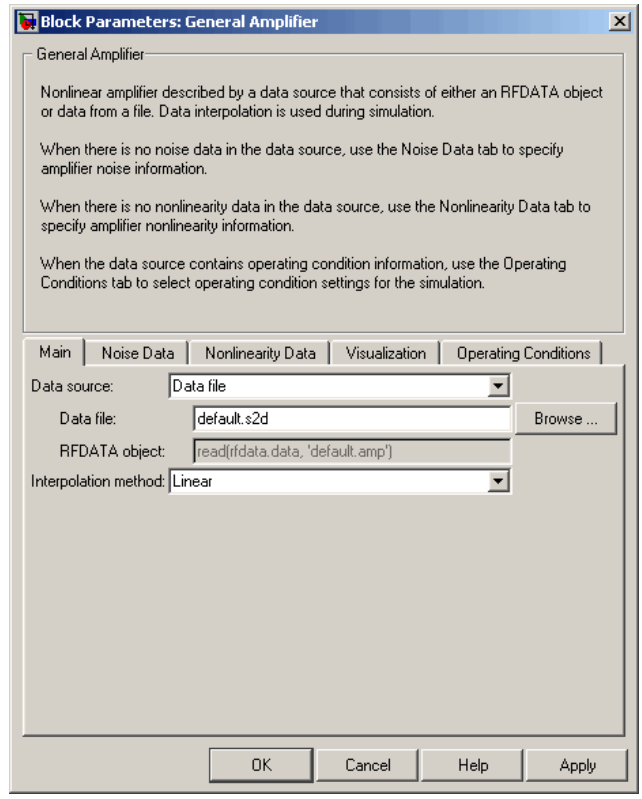
where f_P is the lowest frequency for which power data is specified.

If ΔP is more than 0.4 dB, a warning appears, and the blockset adds ΔP to the output power values at each specified input power value to resolve the discrepancy for simulation. The following graph shows this discrepancy.


--- Small Signal Network Data
*** Specified Power Data
xxx Reconciled Power Data
P_{out} (dBm)
ΔP
f_P
P_{in} (dBm)

Dialog Box

Main Tab



Data source

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

Data file

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

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

RFDATA object

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

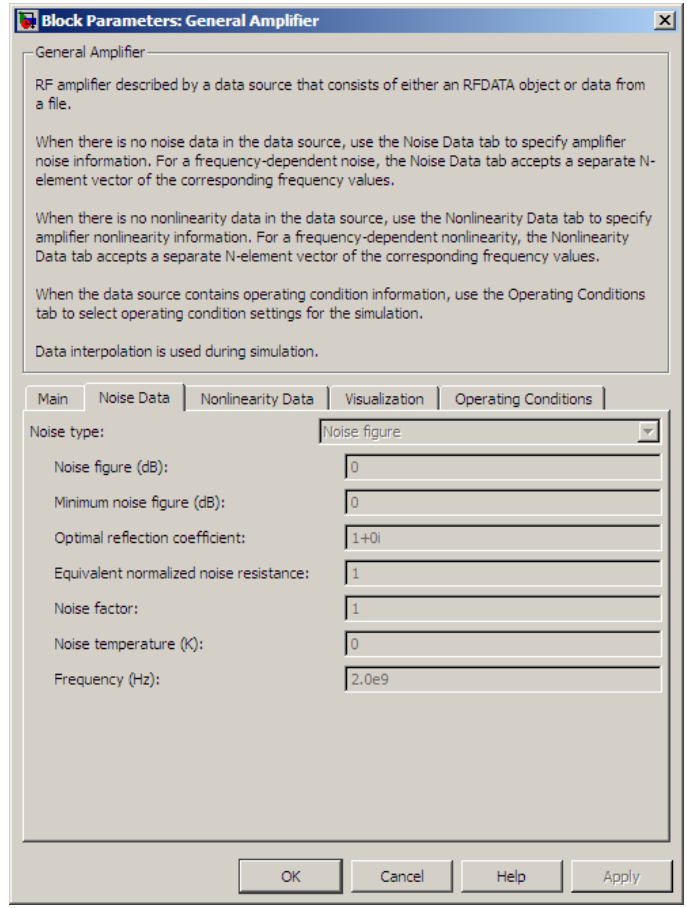
- The handle of a data object previously created using RF Toolbox software.
- An RF Toolbox command such as `rfdata.data('Freq',1e9,'S_Parameters',[0 0; 0.5 0])`, which creates a data object.
- A MATLAB expression that generates such an object.

Interpolation method

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

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

Noise Data Tab



Noise type

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

Noise figure (dB)

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

Minimum noise figure (dB)

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

Optimal reflection coefficient

Optimal amplifier source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance

Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

Noise factor

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

Noise temperature (K)

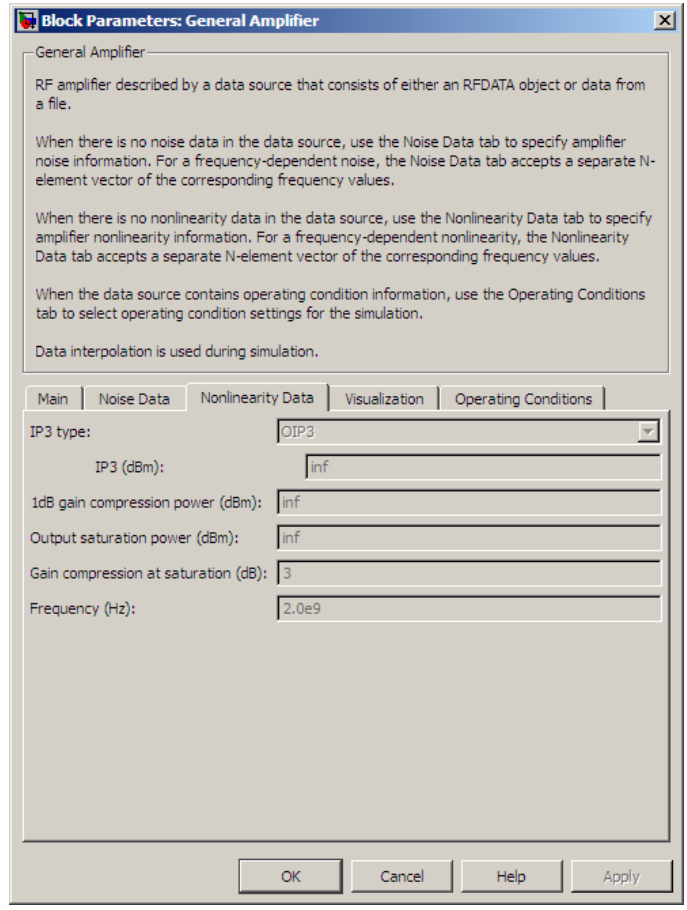
Equivalent temperature or vector of temperatures that produce the same amount of noise power as the amplifier. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the

Interpolation method specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

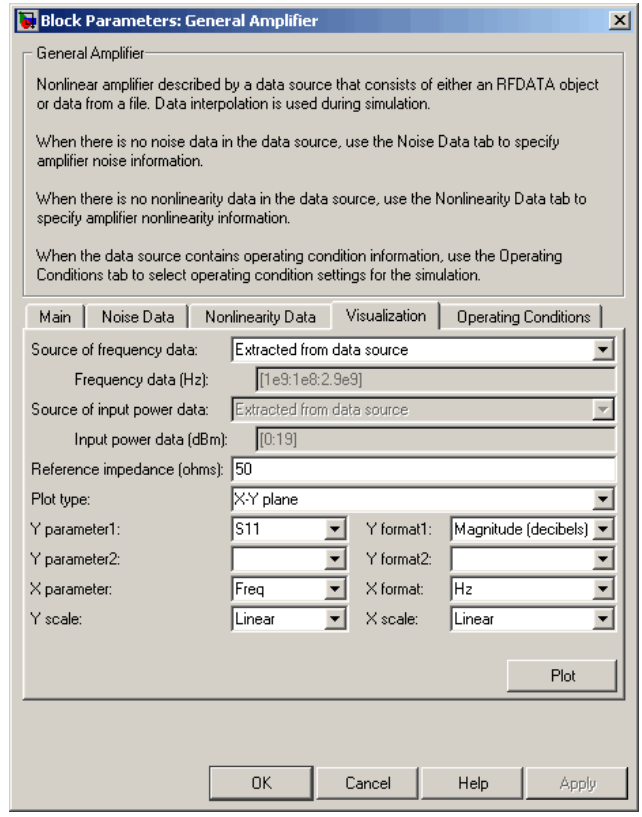
Gain compression at saturation (dB)

Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

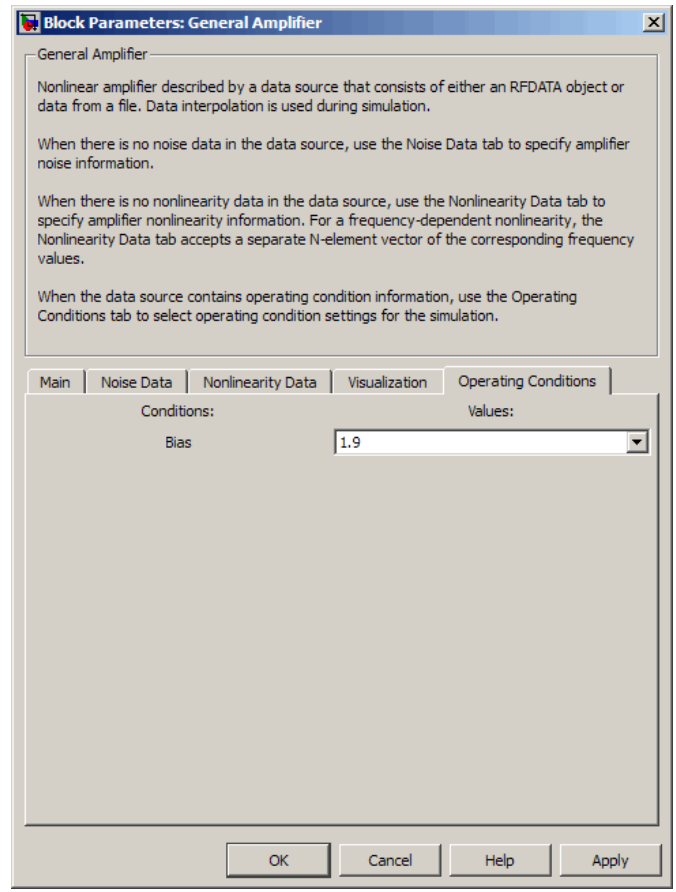
Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”. Use `rftool` or the RF Toolbox plotting functions to plot other data.

Operating Conditions Tab



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

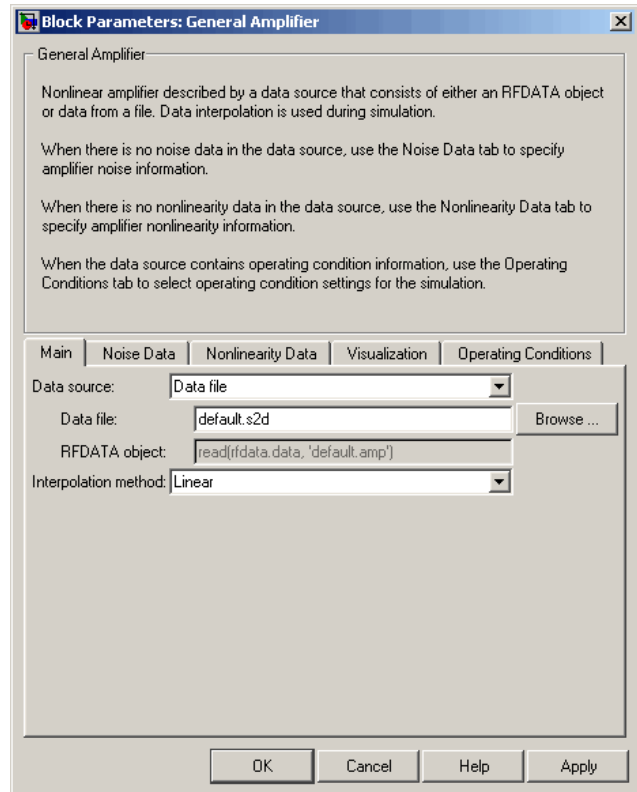
Use the drop-down lists to specify the operating condition values to use in simulation.

Examples

Creating a General Amplifier Block from File Data

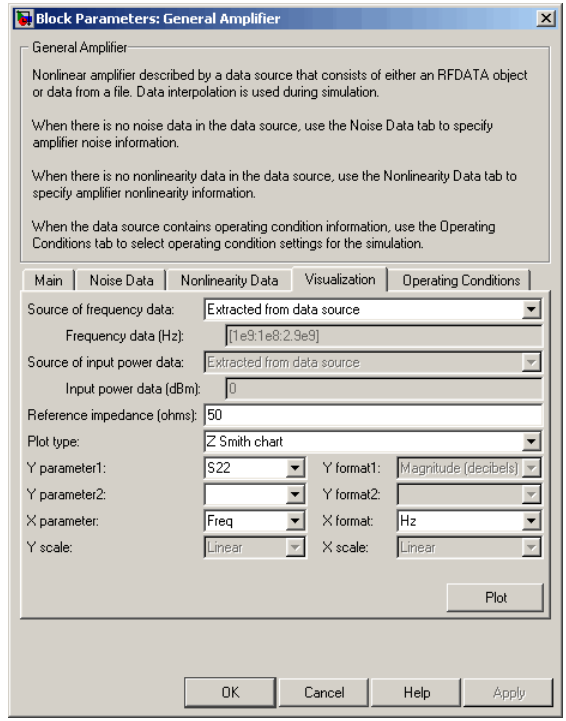
This example uses the default data source, which is the nonlinear amplifier in the file `default.s2d`. The file contains S-parameters for frequencies from 1.0 to 2.9 GHz at intervals of 0.01 GHz, power data at frequency 2.1 GHz, and active noise parameters. By default, the General Amplifier block uses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

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

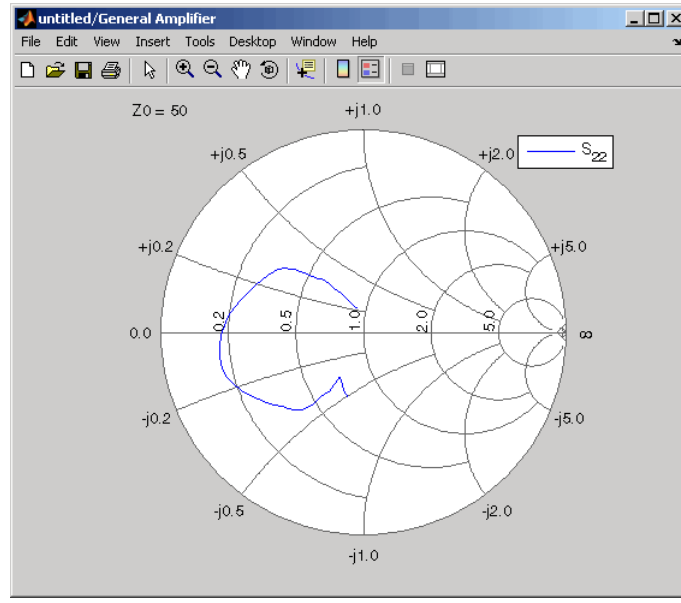


2 On the **Visualization** tab, set the parameters as follows:

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



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



For more about using an Agilent `.s2d` file in a Simulink model, see Effect of Nonlinear Amplifier on QPSK Modulation.

See Also

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

`rfdata.data` (RF Toolbox)

`interp1` (MATLAB)

General Circuit Element

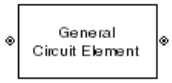
Purpose

Model two-port network described by `rfckt` object

Library

Black Box Elements sublibrary of the Physical library

Description

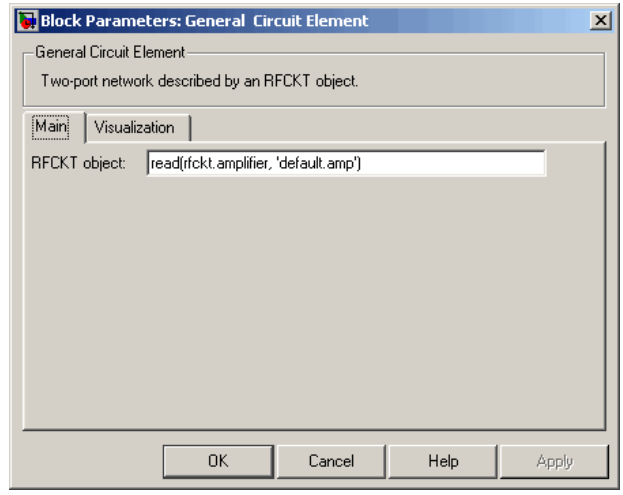


The General Circuit Element block models the two-port network described by an RF Toolbox circuit (`rfckt`) object.

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

Dialog Box

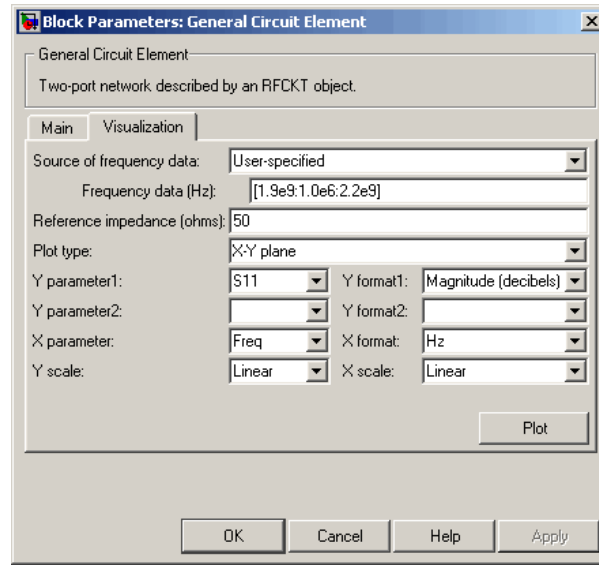
Main Tab



RFCKT object

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

Visualization Tab



For information about plotting, see “Create Plots”.

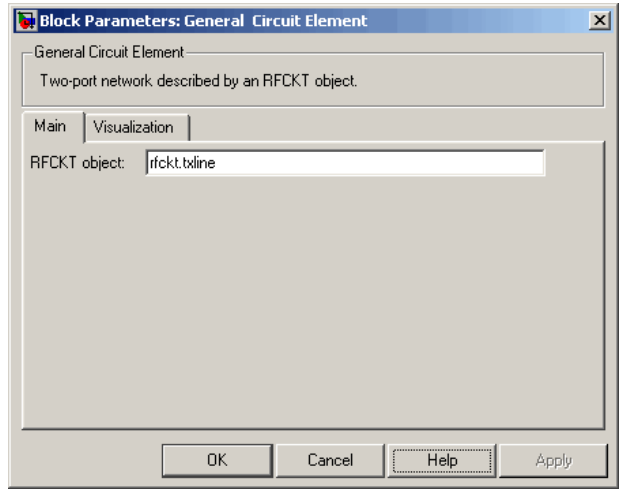
Examples

Creating a General Circuit Element from an RF Toolbox Object

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

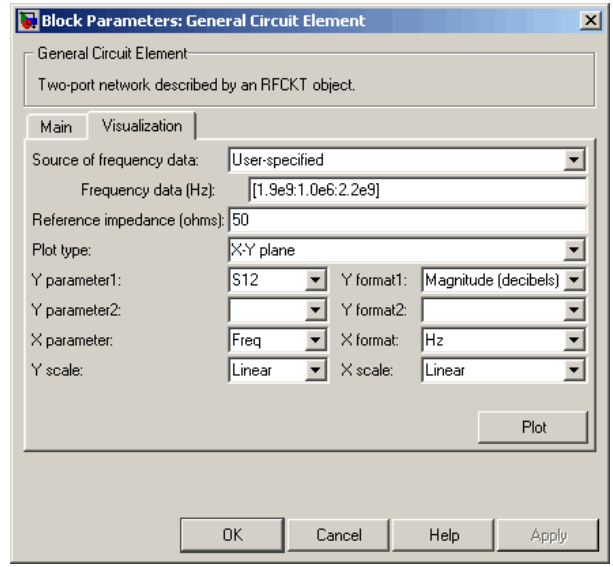
1 On the **Main** tab, set the **RFCKT object** parameter to `rfckt.txline`.

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

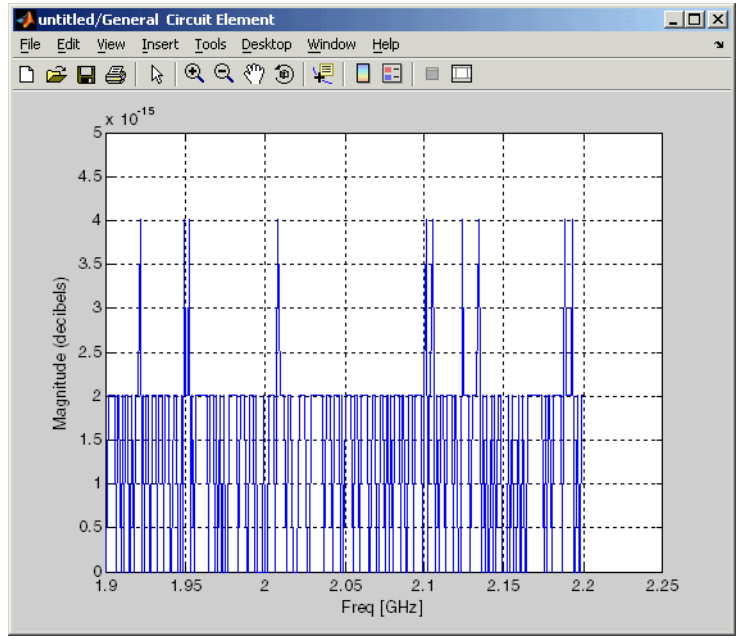


2 Set the General Circuit Element block parameters on the **Visualization** tab as follows:

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



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



See Also

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

`interp1` (MATLAB)

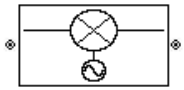
Purpose

Model mixer and local oscillator described by `rfdata` object

Library

Mixer sublibrary of the Physical library

Description



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

Network Parameters

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

SimRF Equivalent Baseband software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

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

where

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

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

Active Noise

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

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

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

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

Phase Noise

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

The blockset computes the digital filter by:

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

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

Nonlinearity

If power data exists in the data source, the block extracts the AMAM/AMPM nonlinearities from it.

If the data source contains no power data, then you can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the General Mixer block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a mixer with a linear gain. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1. The block checks whether you have specified a value other than `Inf` for:
   - The third-order intercept point ($OIP3$ or $IIP3$).
   - The output power at the 1-dB compression point ($P_{1dB,out}$).
   - The output power at saturation ($P_{sat,out}$).

   In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.
2. The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$P_{sat,out} + GC_{sat} = P_{sat,in} + G_{lin}$$
$$P_{1dB,out} + 1 = P_{1dB,in} + G_{lin}$$
$$OIP3 = IIP3 + G_{lin}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$
$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$
$$0 = \frac{c_1}{IIP3} + c_3$$

The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

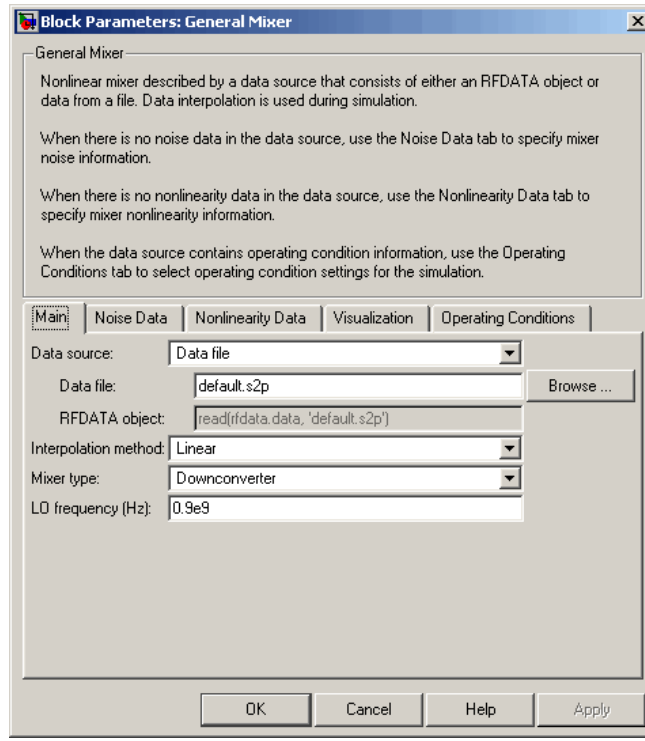
If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Operating Conditions

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

Dialog Box

Main Tab



Data source

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

Data file

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

specify the full path to the file or click the **Browse** button to find the file.

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

RFDATA object

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

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

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

Interpolation method

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

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

Mixer Type

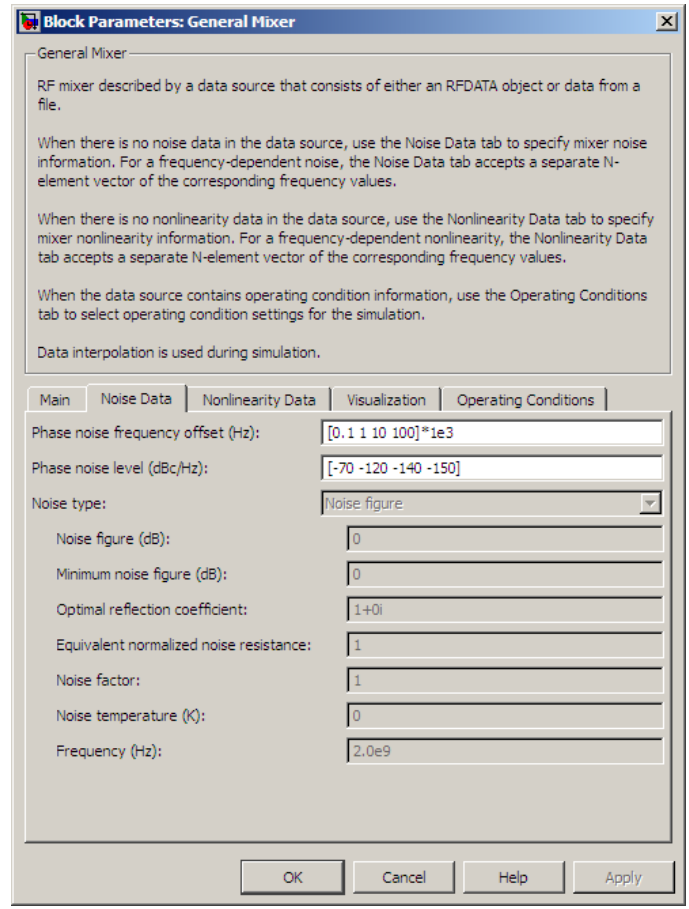
Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

LO frequency (Hz)

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

Note For a downconverting mixer, the local oscillator frequency must satisfy the condition $f_{in} - f_{lo} \geq 1/(2t_s)$, where t_s is the sample time specified in the Input Port block. Otherwise, an error appears.

Noise Data Tab



Phase noise frequency offset (Hz)

Vector specifying the frequency offset.

Phase noise level (dBc/Hz)

Vector specifying the phase noise level.

Noise type

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

Noise figure (dB)

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

Minimum noise figure (dB)

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

Optimal reflection coefficient

Optimal mixer source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance

Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

Noise factor

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

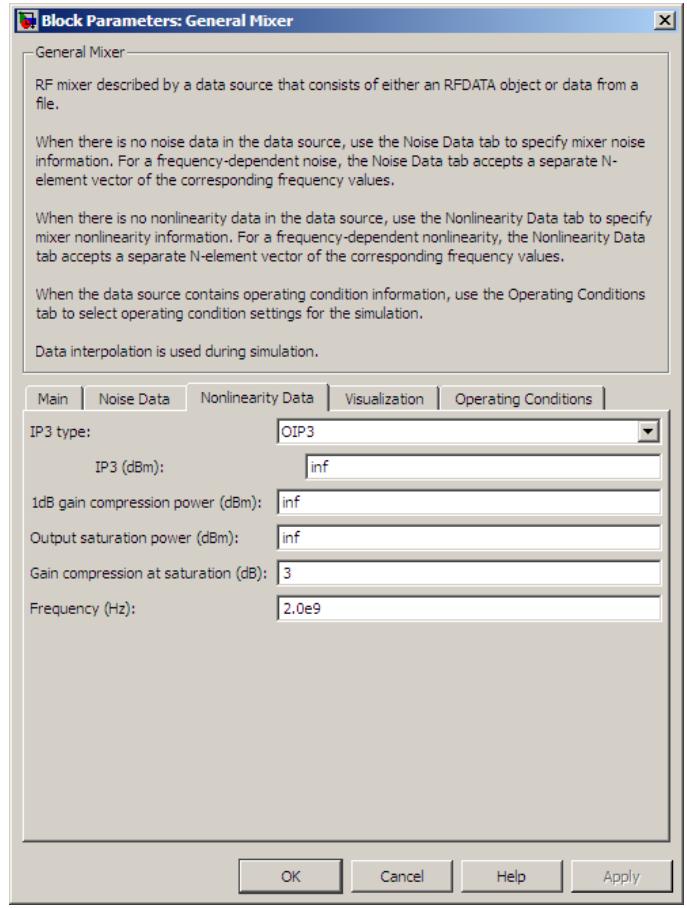
Noise temperature (K)

Equivalent temperature or vector of temperatures that produce the same amount of noise power as the mixer. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1 dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the mixer produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

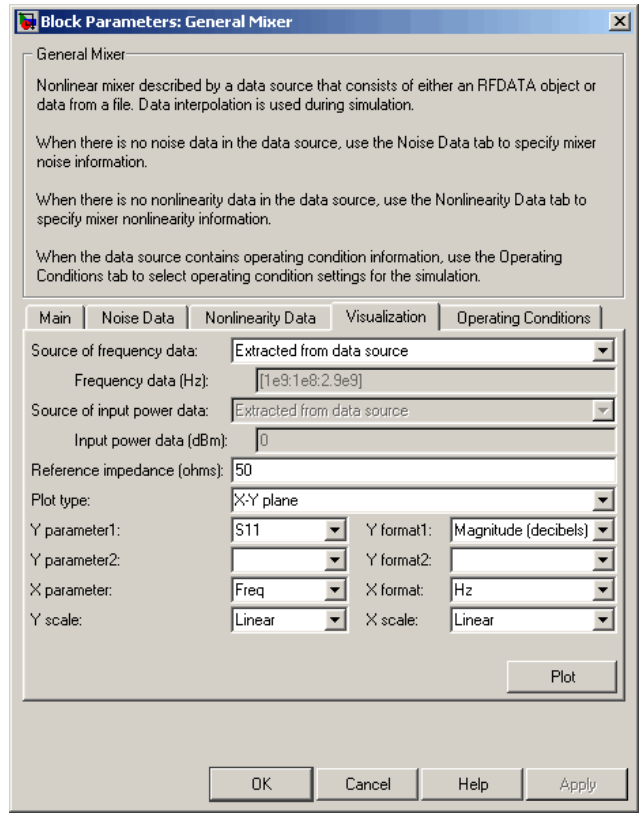
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

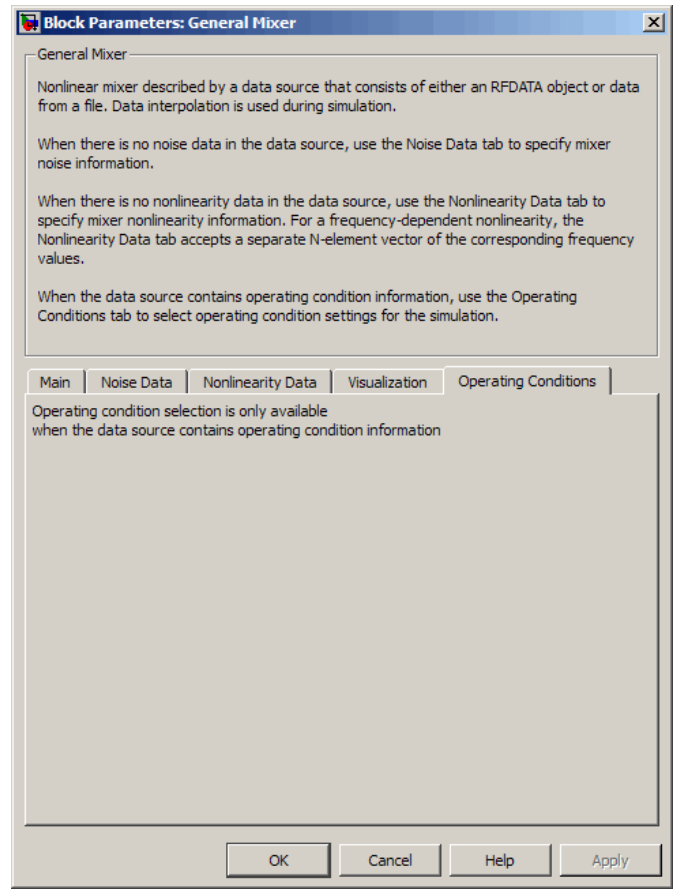
(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”. Use `rftool` or the RF Toolbox plotting functions to plot other data.

Operating Conditions Tab



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

See Also

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

`rfdata.data` (RF Toolbox)

Purpose

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

Library

Black Box Elements sublibrary of the Physical library

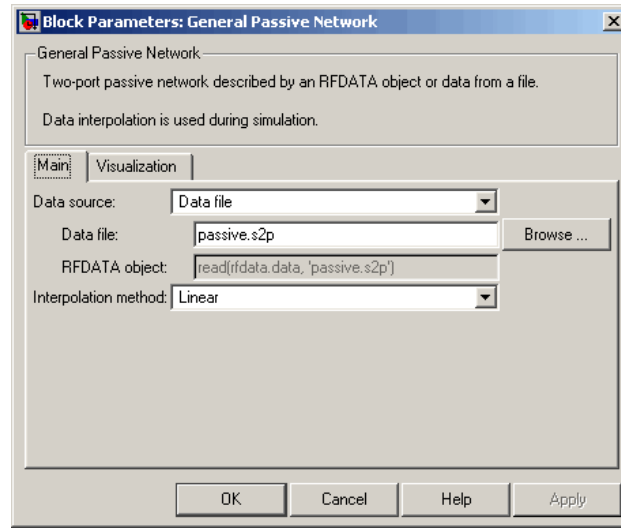
Description

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

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

Dialog Box

Main Tab



Data source

Determines the source of the data that describes the passive device behavior. The data source must contain network parameters and

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

RFDATA object

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

Data file

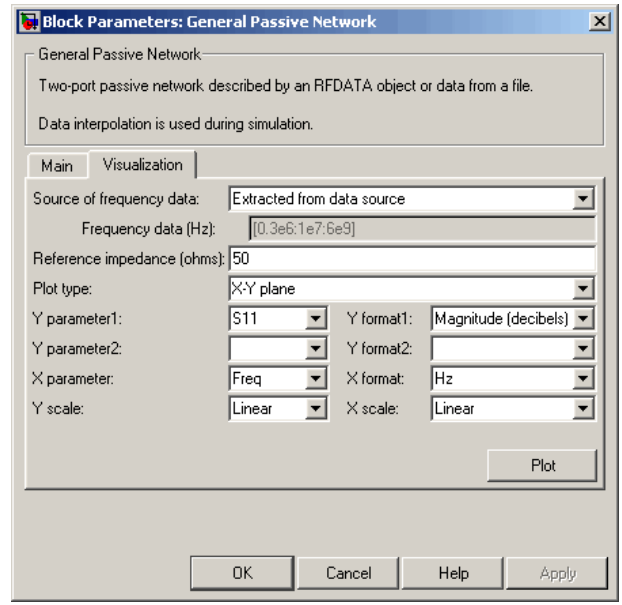
If **Data source** is set to `Data file`, use this field to specify the name of the file that contains the amplifier data. The file name must include the extension. If the file is not in your MATLAB path, specify the full path to the file or click the **Browse** button to find the file.

Interpolation method

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

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

Visualization Tab



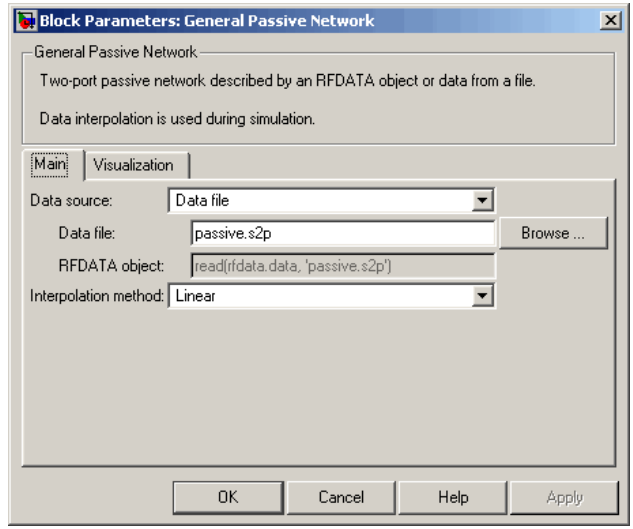
For information about plotting, see “Create Plots”.

Examples

Creating a General Passive Network Block from File Data

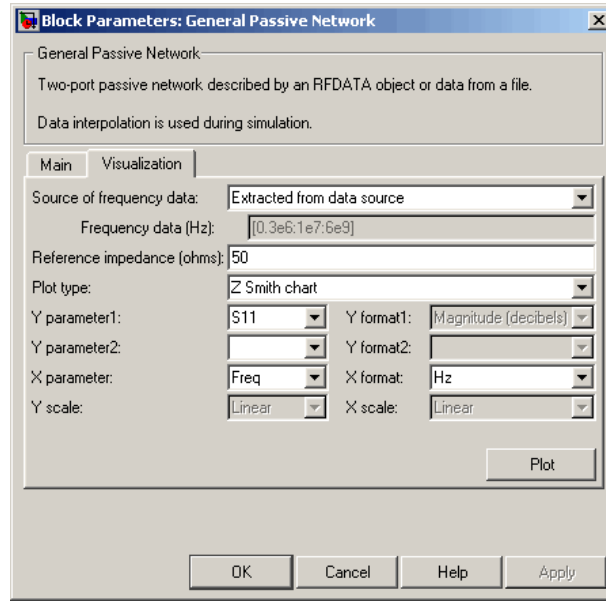
This example creates a two-port passive network from the data in the file `passive.s2p`. The file contains S-parameters for frequencies from about 0.315 MHz to 6.0 GHz. The General Passive Network block uses linear interpolation to model the network described in the object.

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

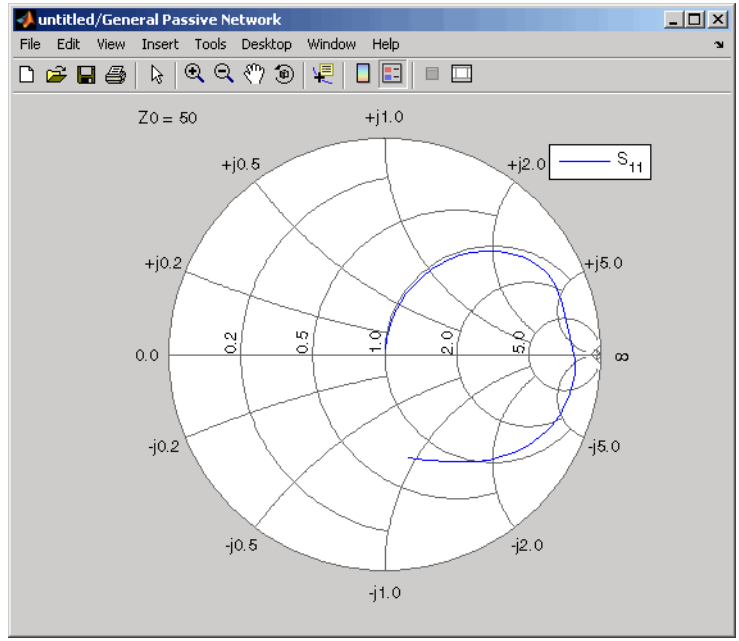


2 On the **Visualization** tab, set the parameters as follows:

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



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



See Also

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

`rfdata.data` (RF Toolbox)

`interp1` (MATLAB)

Purpose

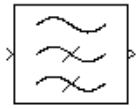
Standard highpass RF filters in baseband-equivalent complex form

Library

Mathematical

Note To use this block, you must install DSP System Toolbox software. For more information, see the SimRF release notes.

Description



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

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

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

Note This block assumes a nominal impedance of 1 ohm.

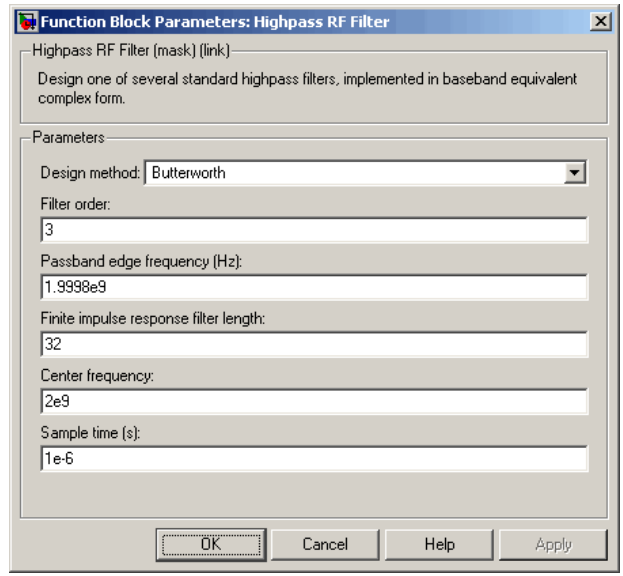
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block lets you specify the filter design parameters shown in the following table.

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

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

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

Dialog Box



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

You can change tunable parameters while the model is running.

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

Filter order
: Order of the filter.

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

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

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

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

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

Center frequency (Hz)
Center of the modeling frequencies.

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

See Also

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

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

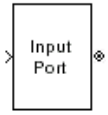
Purpose

Connection block from Simulink environment to RF physical blocks

Library

Input/Output Ports sublibrary of the Physical library

Description

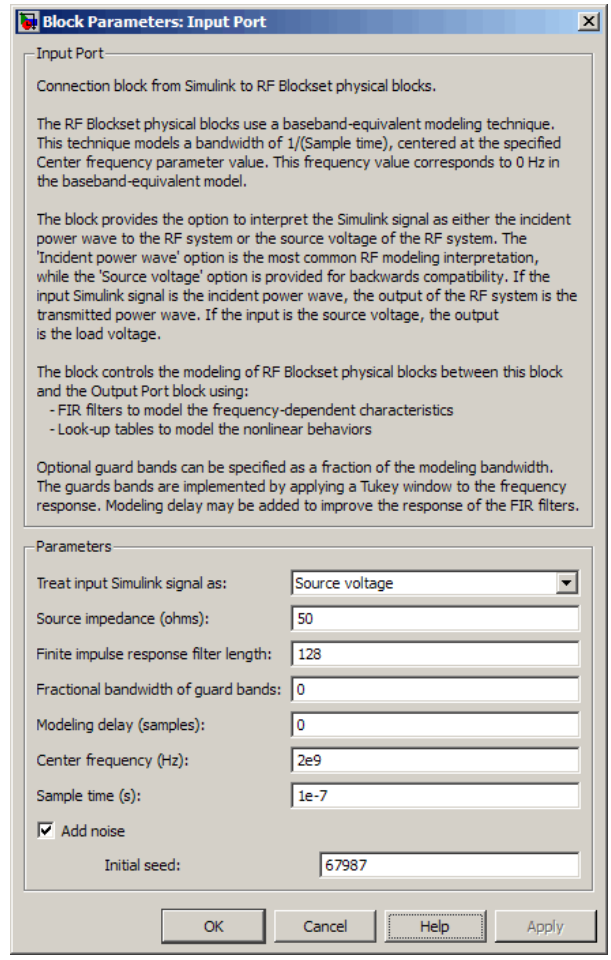


The Input Port block is a connecting port from the Simulink, or mathematical, part of the model to an RF physical part of the model. The Input Port block lets you provide the parameter data needed to calculate the modeling frequencies and the baseband-equivalent impulse response for the physical subsystem. It also lets you specify information about how to interpret the incoming Simulink signal.

For more information about how the Input Port block converts the mathematical Simulink signals to physical modeling environment signals, see “Convert to and from Simulink Signals”.

Note Some RF blocks use 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



Treat input Simulink signal as

Select one of the following options for interpreting the input Simulink signal:

- `Incident power wave` — Interpret the input signal as the incident power wave of the RF system described in the physical

model to which it connects. When you select this option, the output signal of the RF system is the transmitted power wave. This is the most common RF modeling interpretation.

- `Source voltage` — Interpret the input signal as the source voltage of the RF system described in the physical model to which it connects. As a result, the baseband-equivalent model includes the loss through the source impedance. When you select this option, the output signal of the RF system is the load voltage.

For more information about these options, see "Convert to and from Simulink Signals".

Source impedance (ohms)

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

Finite impulse response filter length

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

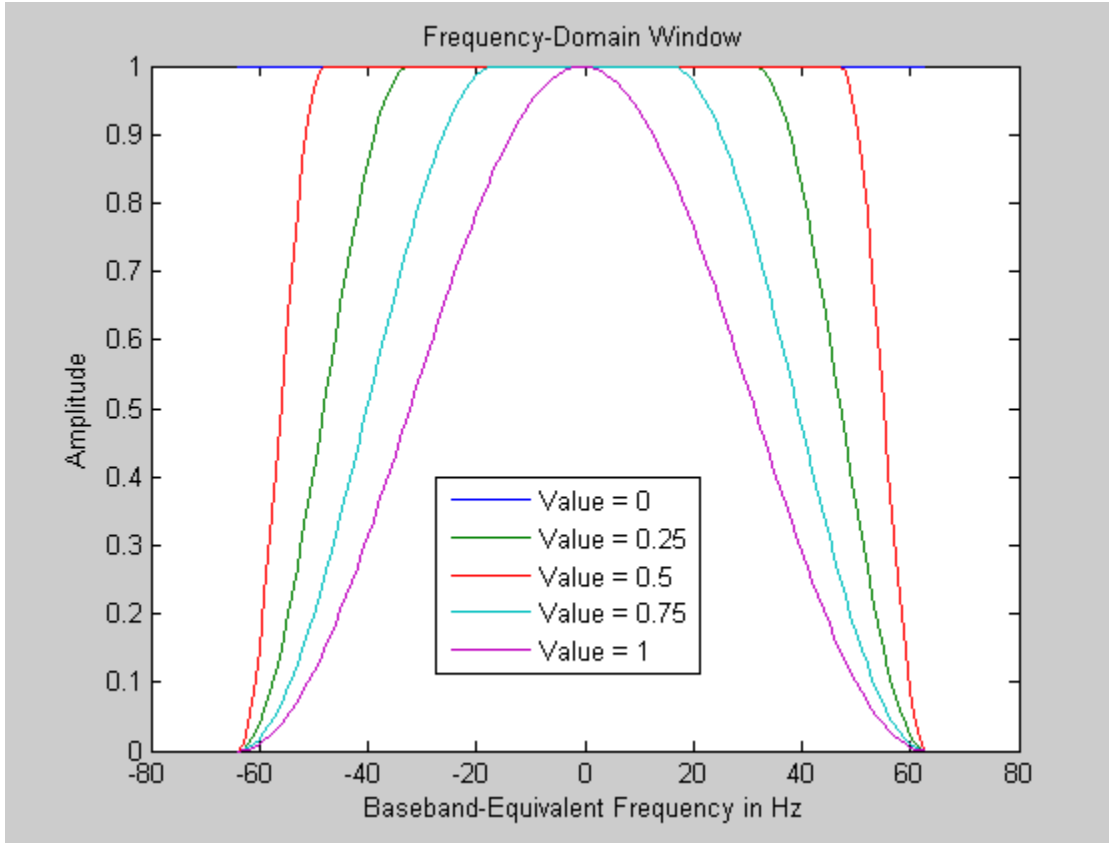
Note The equivalent-baseband simulation algorithm uses the next power of 2 greater than the specified filter length in its calculations and then truncates the impulse response to the specified length. As a result, you get different results when you set the **Finite impulse response filter length** parameter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

Fractional bandwidth of guard bands

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

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

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



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

Modeling delay (samples)

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

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

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

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

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

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

Initial seed
Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise. This parameter becomes visible if you select the **Add noise** parameter. If you specify the initial seed parameter with a variable, the initial seed changes each successive time you run a model.

See Also

Output Port

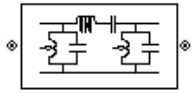
Purpose

Model LC bandpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description



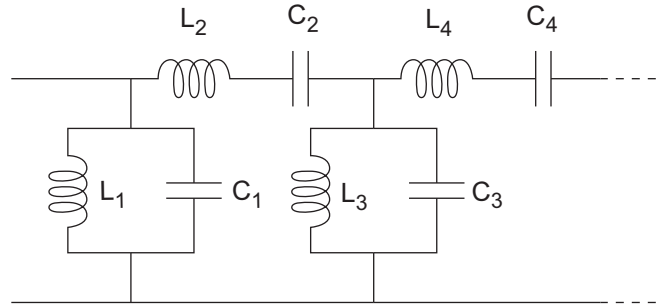
The LC Bandpass Pi block models the LC bandpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

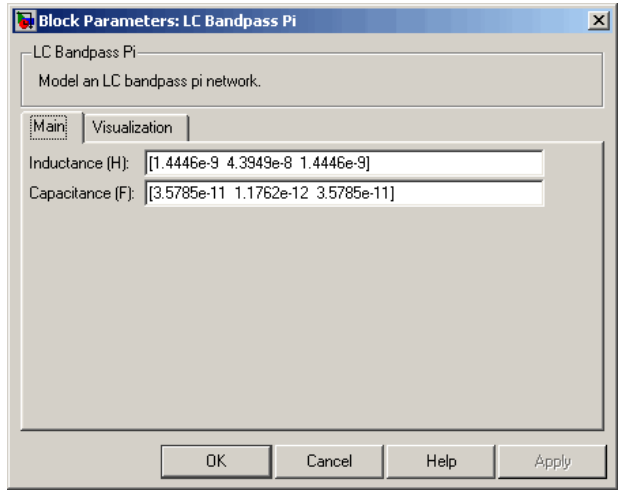
The LC bandpass pi network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



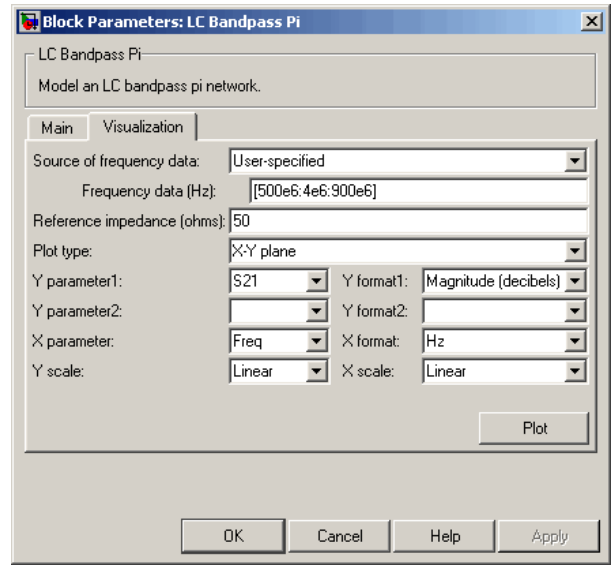
Inductance (H)

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

Capacitance (F)

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

Visualization Tab

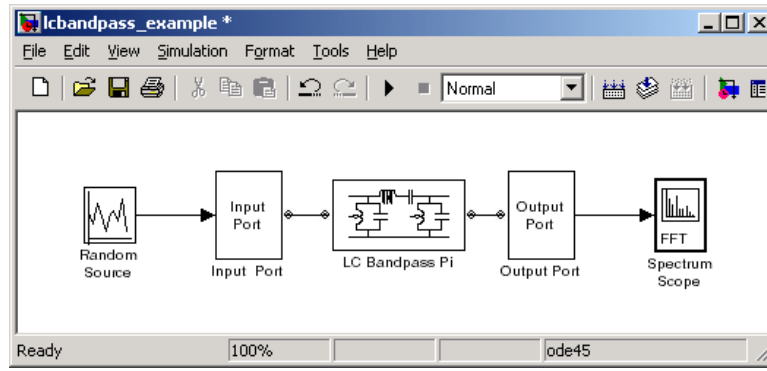


For information about plotting, see “Create Plots”.

Examples

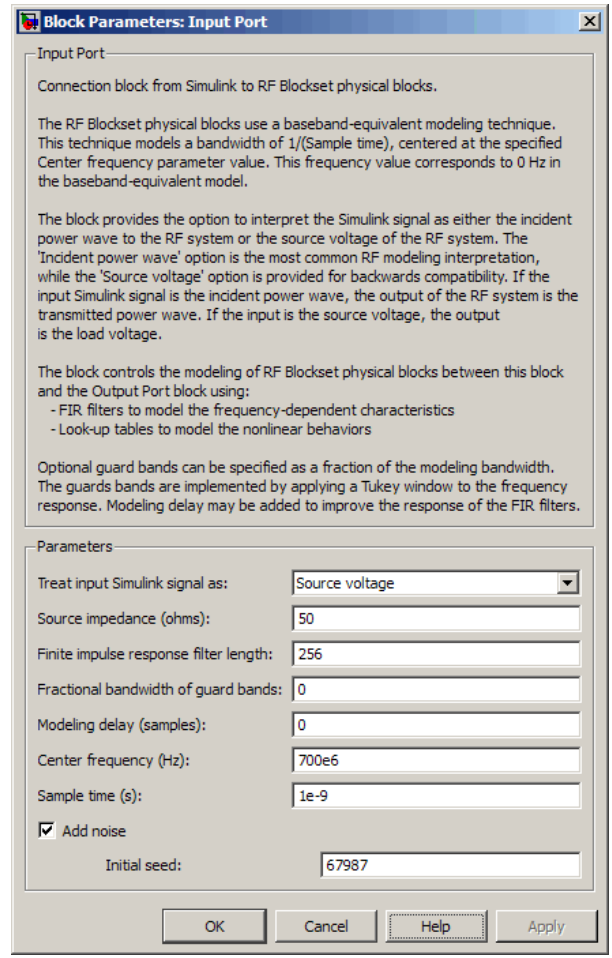
Using a Ladder Filter Block to Filter Gaussian Noise

This example provides complex random noise in Gaussian form as input to an LC Bandpass Pi block. A DSP System Toolbox Spectrum Scope block plots the filtered output.

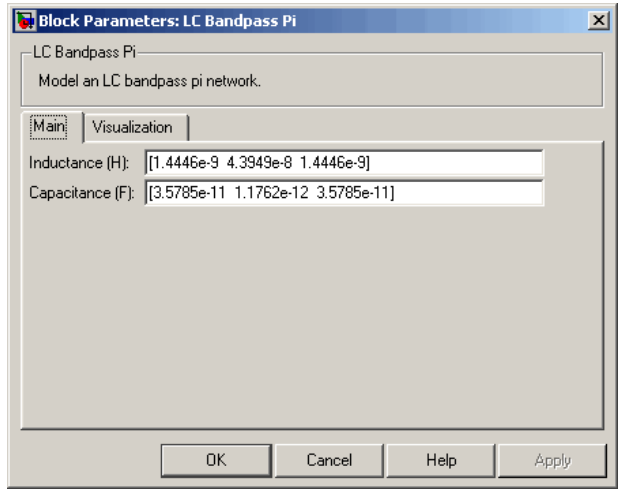


The DSP System Toolbox Random Source block produces frame-based output at 512 samples per frame. Its **Sample time** parameter is set to 1.0e-9. This sample time must match the sample time for the physical part of the model, which you provide in the Input Port block diagram.

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

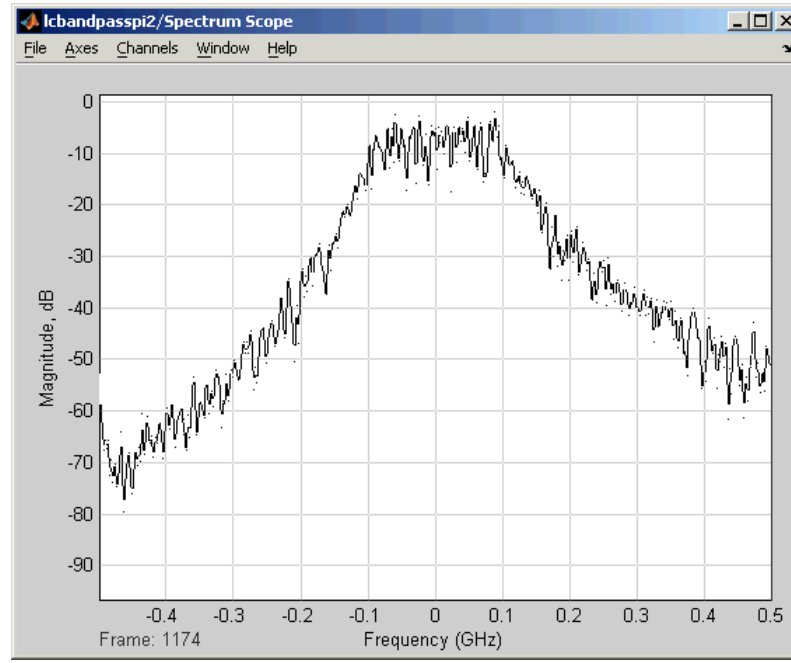


The LC Bandpass Pi block provides the inductances for three inductors, in order from source to load, [1.4446e-9, 4.3949e-8, 1.4446e-9]. Similarly, it provides the capacitances for three capacitors [3.5785e-11, 1.1762e-12, 3.5785e-11].



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

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



References

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

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

See Also

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

LC Bandpass Tee

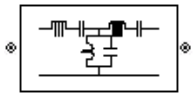
Purpose

Model LC bandpass tee network

Library

Ladder Filters sublibrary of the Physical library

Description



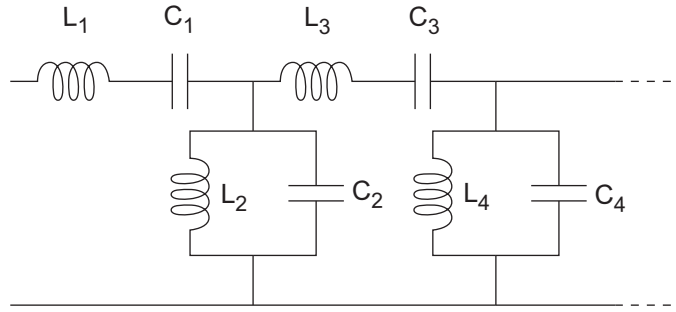
The LC Bandpass Tee block models the LC bandpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

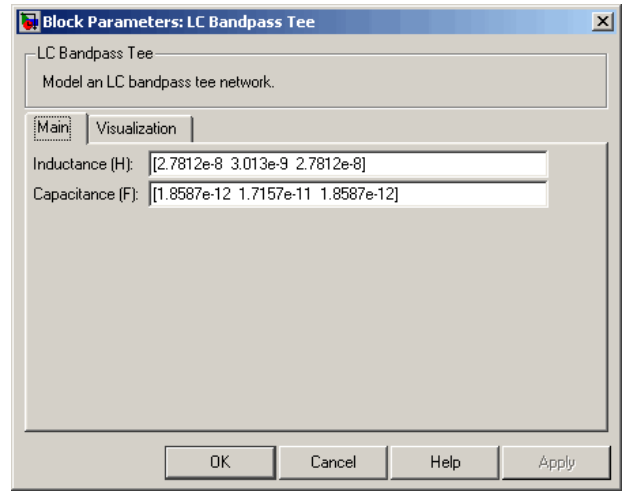
The LC bandpass tee network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



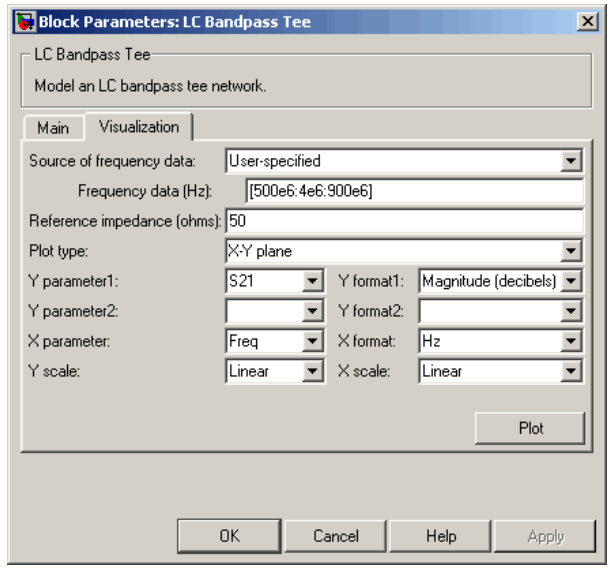
Inductance (H)

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

Capacitance (F)

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

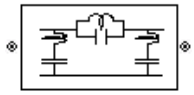
Purpose

Model LC bandstop pi network

Library

Ladder Filters sublibrary of the Physical library

Description



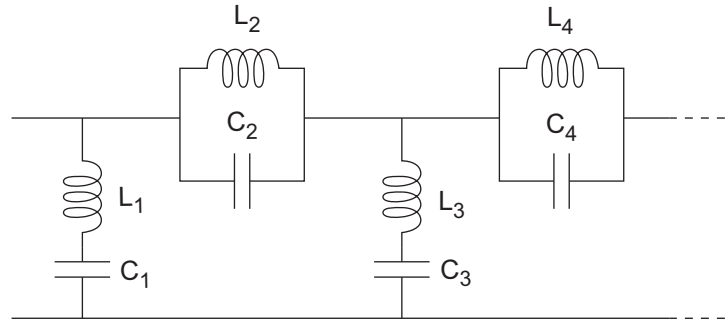
The LC Bandstop Pi block models the LC bandstop pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

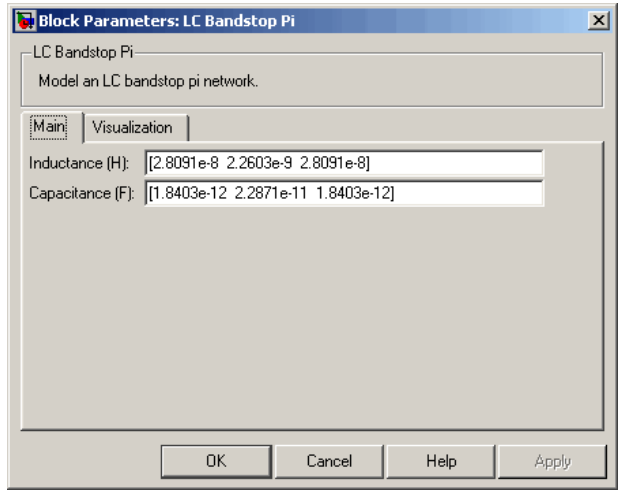
The LC bandstop pi network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



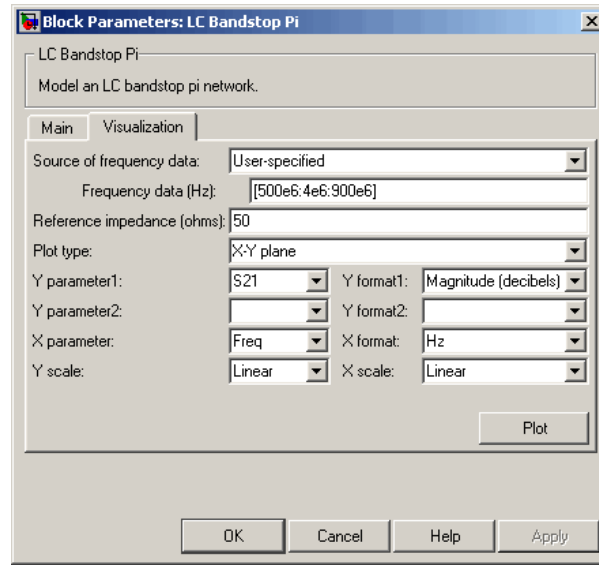
Inductance (H)

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

Capacitance (F)

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

LC Bandstop Tee

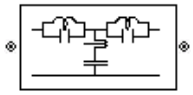
Purpose

Model LC bandstop tee network

Library

Ladder Filters sublibrary of the Physical library

Description



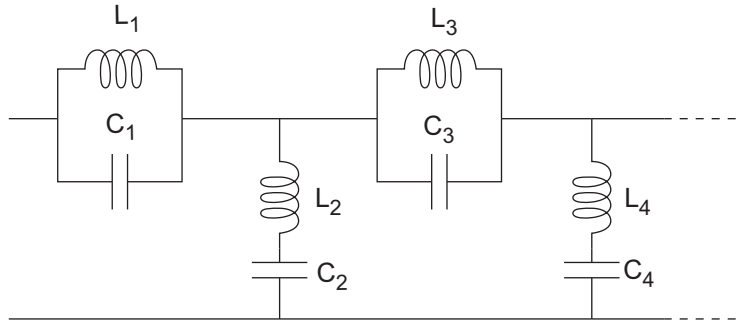
The LC Bandstop Tee block models the LC bandstop tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

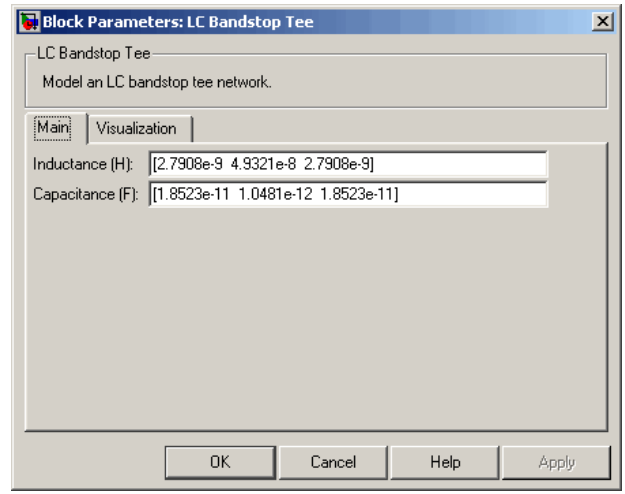
The LC bandstop tee network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



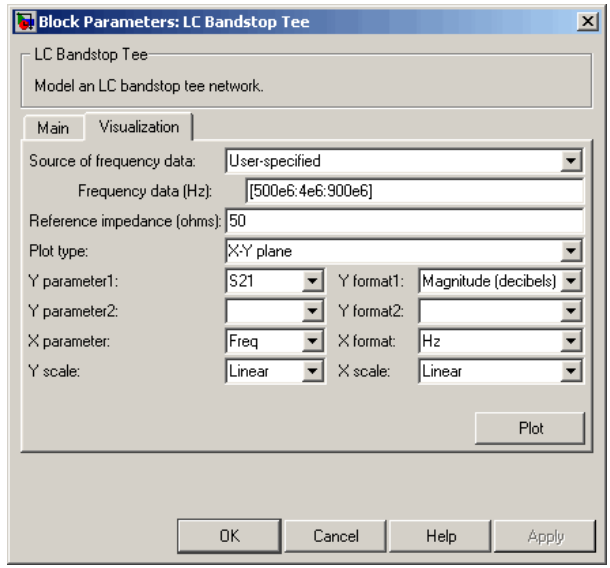
Inductance (H)

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

Capacitance (F)

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

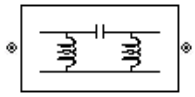
Purpose

Model LC highpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description



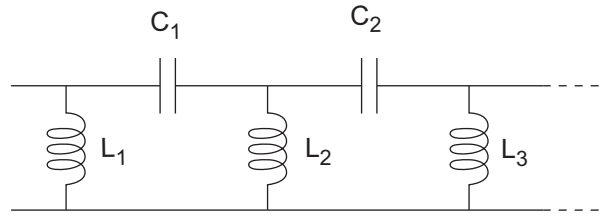
The LC Highpass Pi block models the LC highpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

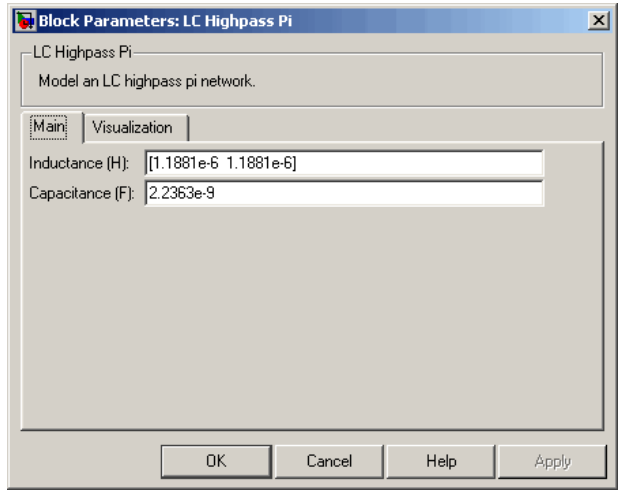
The LC highpass pi network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



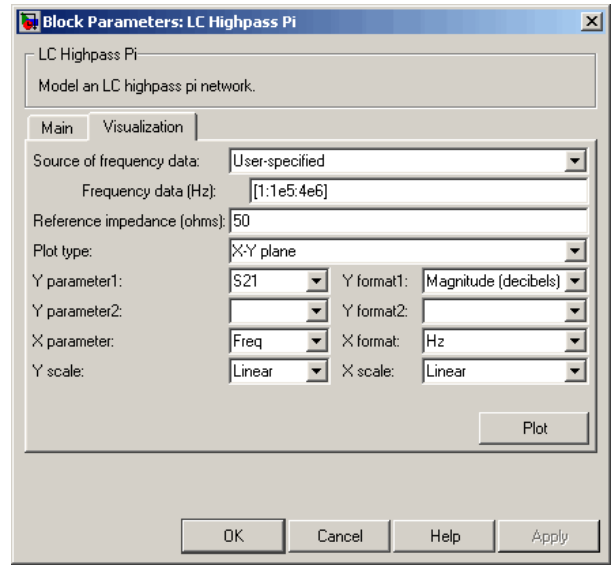
Inductance (H)

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

Capacitance (F)

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

LC Highpass Tee

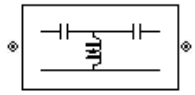
Purpose

Model LC highpass tee network

Library

Ladder Filters sublibrary of the Physical library

Description



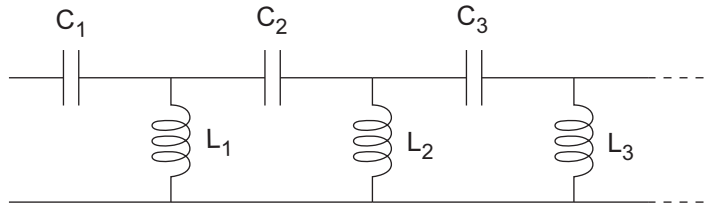
The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

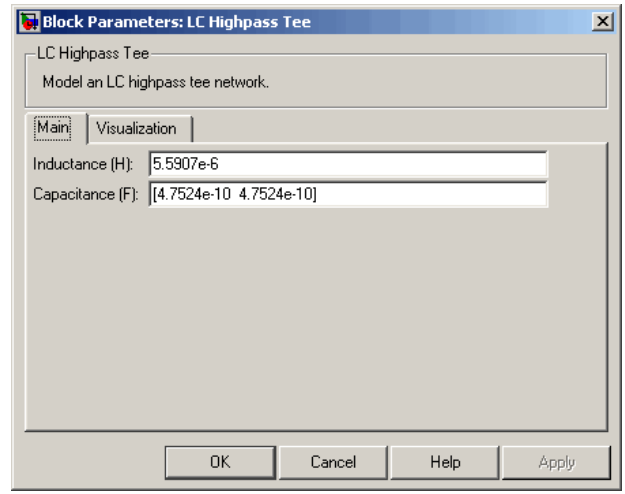
The LC highpass tee network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



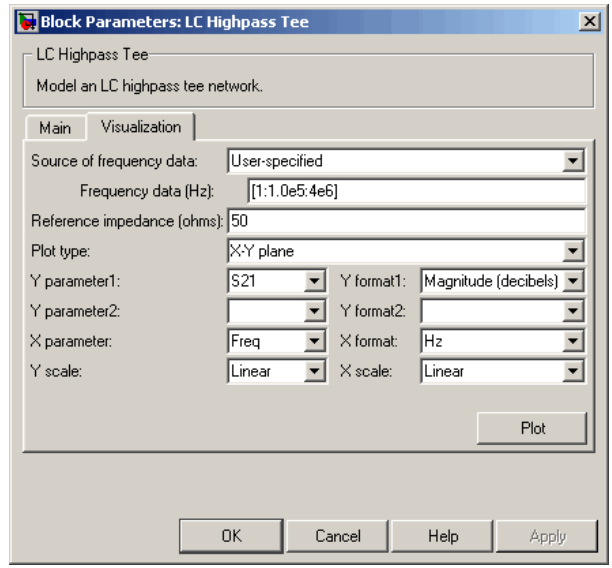
Inductance (H)

Vector containing the inductances, in order from source to load, of all inductors in the network. All values must be strictly positive. The vector cannot be empty.

Capacitance (F)

Vector containing the capacitances, in order from source to load, of all capacitors in the network. The capacitance vector must contain at least two elements. Its length must be equal to or one greater than the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

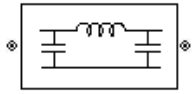
Purpose

Model LC lowpass pi network

Library

Ladder Filters sublibrary of the Physical library

Description



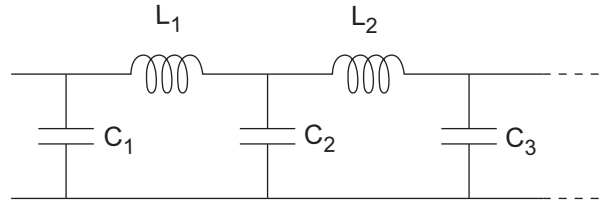
The LC Lowpass Pi block models the LC lowpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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

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

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

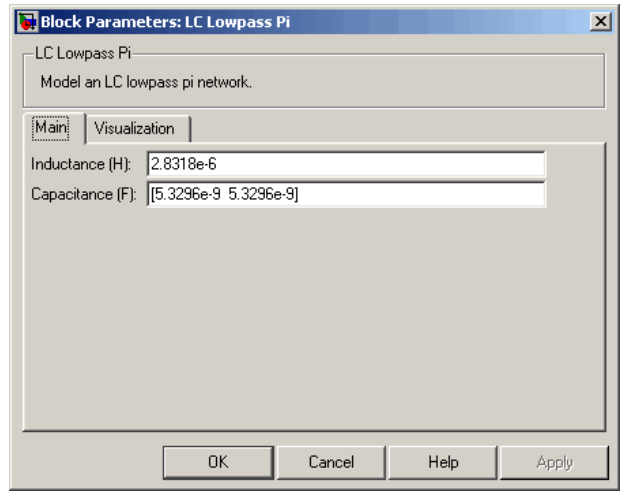
The LC lowpass pi network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



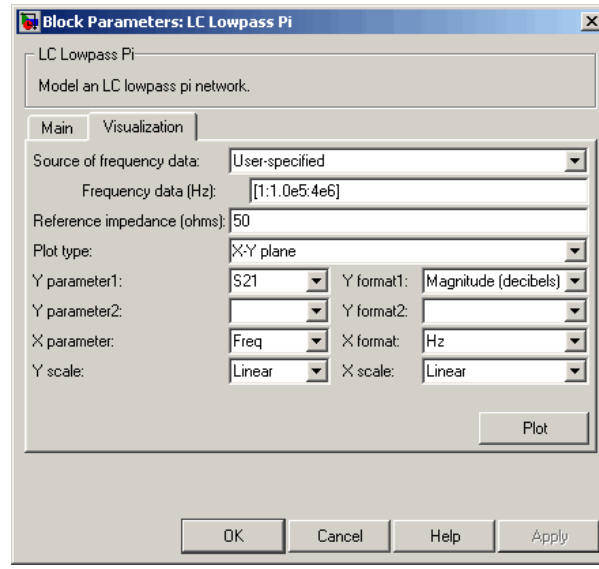
Inductance (H)

Vector containing the inductances, in order from source to load, of all inductors in the network. All values must be strictly positive. The vector cannot be empty.

Capacitance (F)

Vector containing the capacitances, in order from source to load, of all capacitors in the network. The capacitance vector must contain at least two elements. Its length must be equal to or one greater than the length of the vector you provide in the **Inductance** parameter. All values must be strictly positive.

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

LC Lowpass Tee

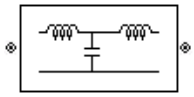
Purpose

Model LC lowpass tee network

Library

Ladders Filters sublibrary of the Physical library

Description



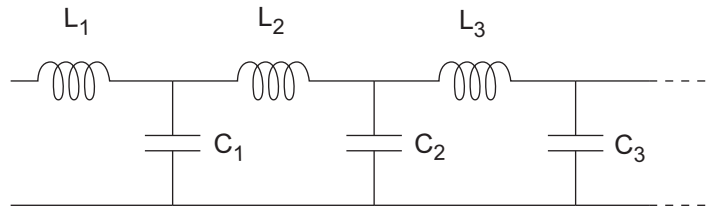
The LC Lowpass Tee block models the LC lowpass tee network described in the block dialog box in terms of its frequency-dependent S-parameters.

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

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

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

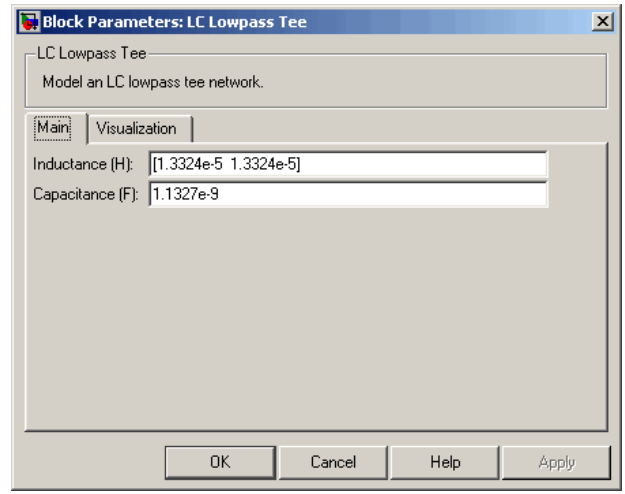
The LC lowpass tee network object is a two-port network as shown in the following circuit diagram.



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

Dialog Box

Main Tab



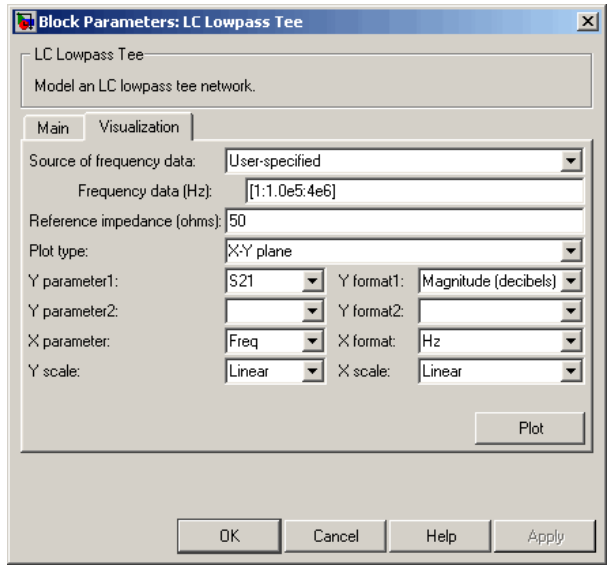
Inductance (H)

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

Capacitance (F)

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

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

References

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

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

See Also

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

Purpose

Standard lowpass RF filters in baseband-equivalent complex form

Library

Mathematical

Note To use this block, you must install DSP System Toolbox software. For more information, see the SimRF release notes.

Description

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

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

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

Note This block assumes a nominal impedance of 1 ohm.

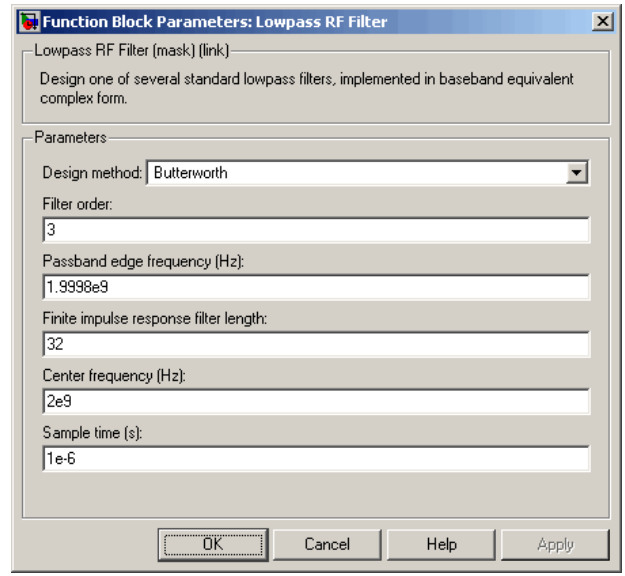
Select the design of the filter from the **Design method** list in the dialog box. For each design method, the block enables you to specify the filter design parameters shown in the following table.

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

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

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

Dialog Box



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

Parameters that are tunable can be changed while the model is running.

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

Filter order
: Order of the filter.

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

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

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

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

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

Center frequency (Hz)
Center of the modeling frequencies.

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

See Also

Amplifier, Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Mixer

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

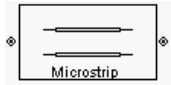
Purpose

Model microstrip transmission line

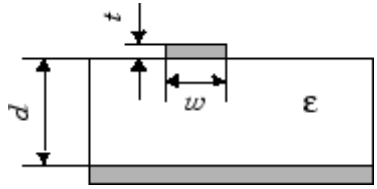
Library

Transmission Lines sublibrary of the Physical library

Description



The Microstrip Transmission Line block models the microstrip transmission line described in the block dialog in terms of its frequency-dependent S-parameters. A microstrip transmission line is shown in cross-section in the following figure. Its physical characteristics include the microstrip width (w), the microstrip thickness (t), the substrate height (d), and the relative permittivity constant (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 first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

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

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

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

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

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

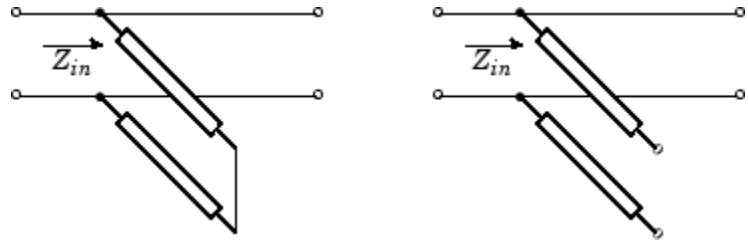
Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the specified conductor strip width, substrate height, conductor strip thickness, relative permittivity constant, conductivity, and dielectric loss tangent of the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

Shunt and Series Stubs

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

Shunt ABCD-Parameters

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

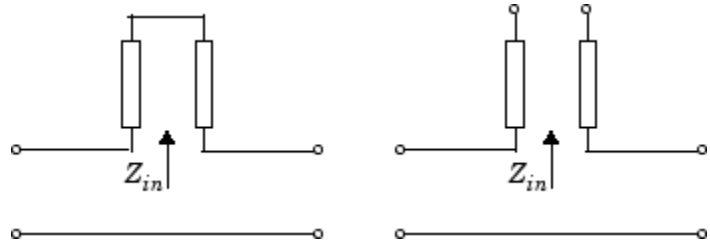


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

Series ABCD-Parameters

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

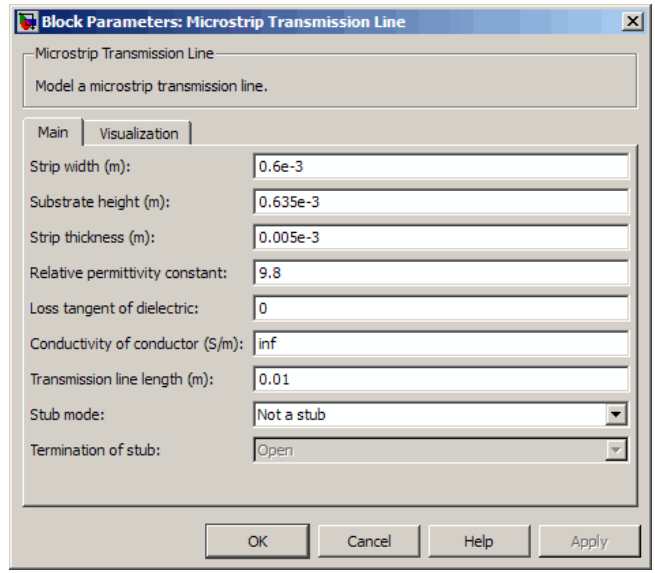


Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

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

Dialog Box

Main Tab



Strip width (m)
Width of the microstrip transmission line.

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

Strip thickness (m)
Physical thickness of the microstrip.

Relative permittivity constant
Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space ε_0.

Loss tangent in dielectric
Loss angle tangent of the dielectric.

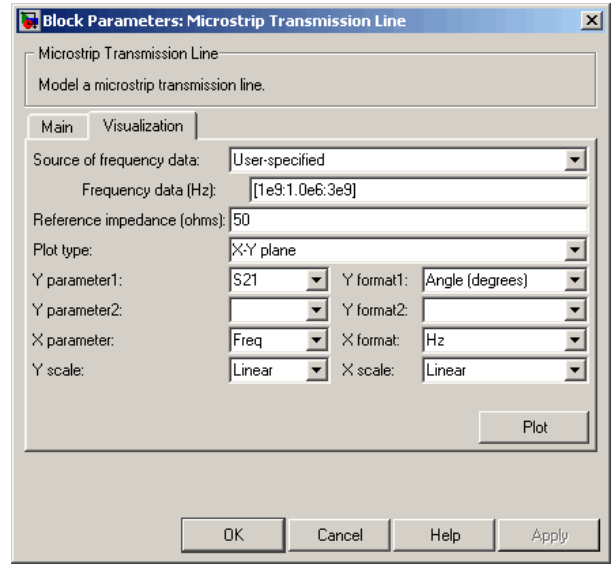
Conductivity in conductor (S/m)
Conductivity of the conductor in siemens per meter.

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

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

References

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

See Also

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

Purpose

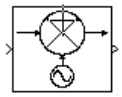
Complex baseband model of mixer and local oscillator with phase noise

Library

Mathematical

Note To use this block, you must install DSP System Toolbox software. For more information, see the SimRF release notes.

Description



The Mixer block generates a complex baseband model of the following:

- A mixer
- A local oscillator with phase noise whose spectrum is characterized by a 1/*f* slope

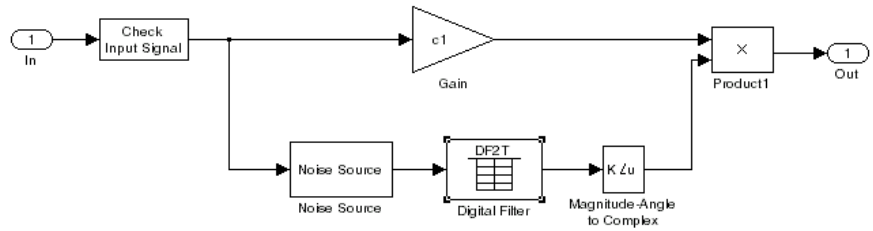
The Mixer block includes both the IF and RF signals as complex-baseband equivalent signals. Both the IF and RF center frequencies are represented as 0 hertz. The amplitude of the noise spectrum is specified by the noise power contained in a 1-hertz bandwidth offset from the carrier by a specified frequency.

Note This block assumes a nominal impedance of 1 ohm.

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

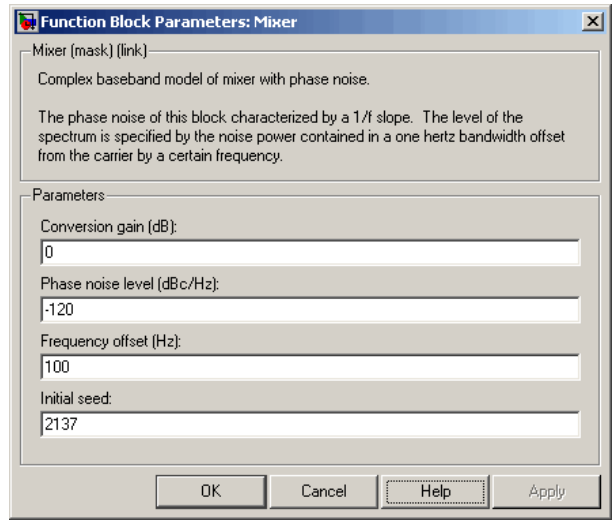
1. Generates additive white Gaussian noise (AWGN) and filters it with a digital filter.
2. Adds the resulting phase noise to the angle component of the input signal.

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



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

Dialog Box



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

Conversion gain (dB)

Scalar specifying the conversion gain for the mixer. Use a negative value to specify loss. Tunable.

Phase noise level (dBc/Hz)
Scalar specifying the phase noise level in decibels relative to the carrier, per hertz. Tunable.

Frequency offset (Hz)
Scalar specifying the frequency offset. Tunable.

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

References

[1] Kasdin, N.J., "Discrete Simulation of Colored Noise and Stochastic Processes and 1/(f^alpha); Power Law Noise Generation," The Proceedings of the IEEE, May, 1995, Vol. 83, No. 5.

See Also

Amplifier, Bandpass RF Filter, Bandstop RF Filter, Highpass RF Filter, Lowpass RF Filter

Output Port

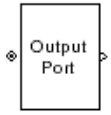
Purpose

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 how the Output Port block converts the physical modeling environment signals to mathematical Simulink signals, see “Convert to and from Simulink Signals”.

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

Linear Subsystem

For the linear subsystem, the Output Port block uses the Input Port block parameters and the interpolated S-parameters calculated by each of the cascaded physical blocks to calculate the baseband-equivalent impulse response. Specifically, it

1 Determines the modeling frequencies f as an N-element vector. The modeling frequencies are a function of the center frequency f_c, the sample time t_s, and the finite impulse response filter length N, all of which you specify in the Input Port block dialog box.

The nth element of f, f_n, is given by

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

where

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

2 Calculates the passband transfer function for the frequency range as

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

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

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

where

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

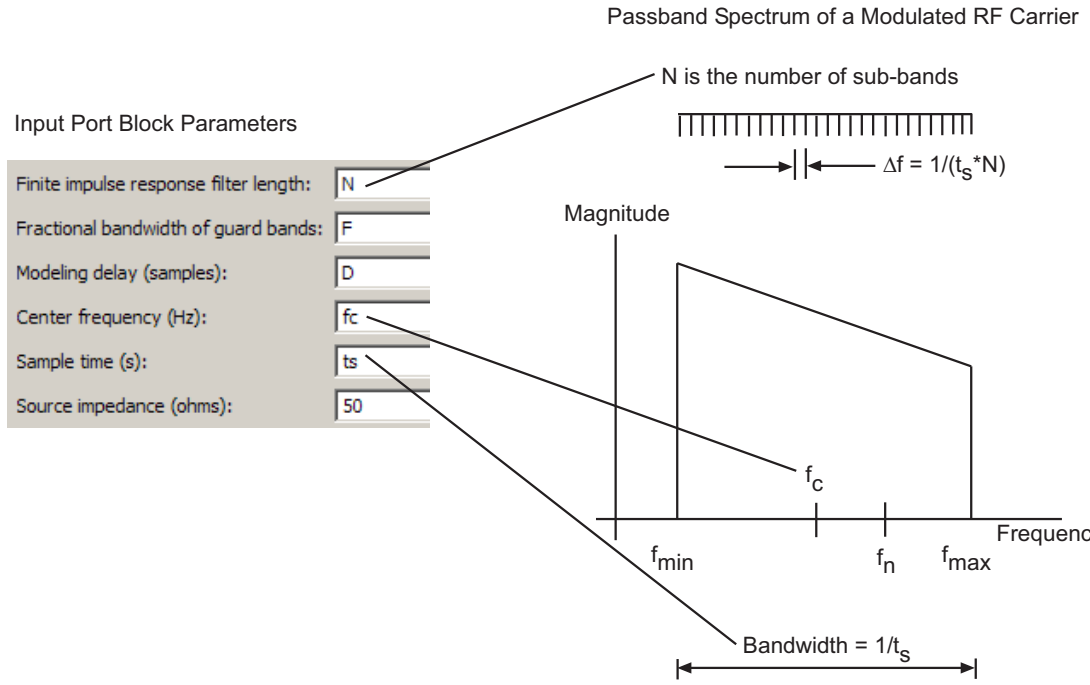
$$\Gamma_s = \frac{Z_s - Z_o}{Z_s + Z_o}$$

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

and

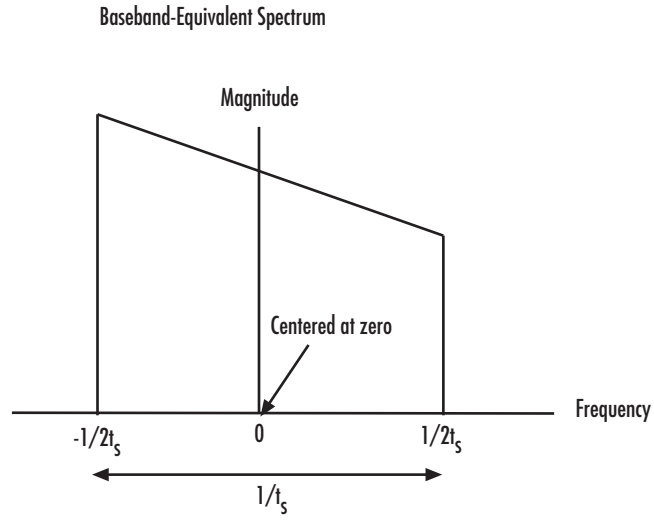
- Z_S is the source impedance.
- Z_L is the load impedance.
- S_{ij} are the S-parameters of a two-port network.

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



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

The baseband transfer function is shown in the following figure.

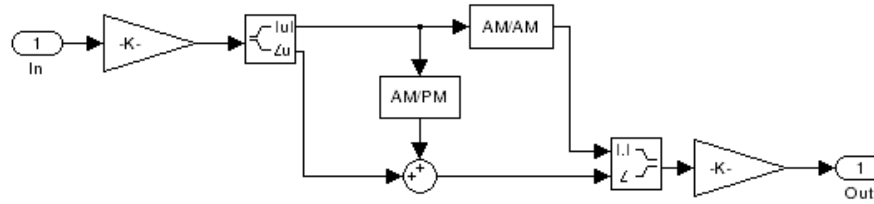


4 Obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. For faster simulation, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. Then, it truncates the impulse response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

For the linear subsystem, the Output Port block uses the calculated impulse response as input to the DSP System Toolbox Digital Filter block to determine the output.

Nonlinear Subsystem

The nonlinear subsystem is implemented by AM/AM and AM/PM nonlinear models, as shown in the following figure.



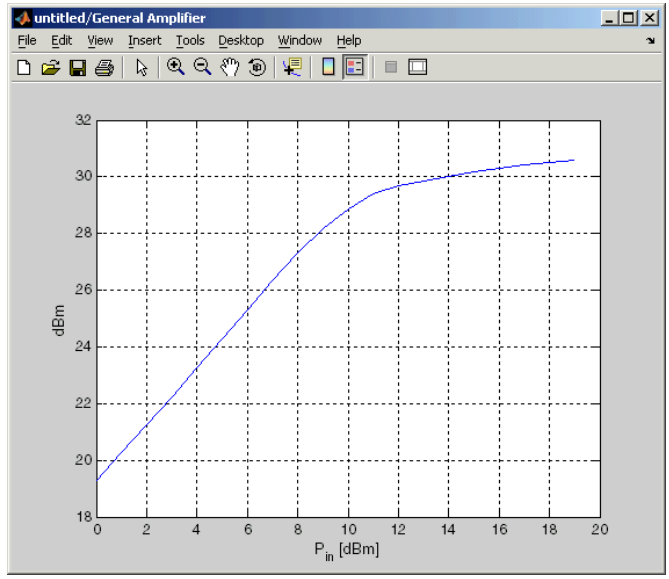
The nonlinearities of AM/AM and AM/PM conversions are extracted from the power data of an amplifier or mixer by the equations

$$AM_{out} = \sqrt{R_l P_{out}}$$
$$PM_{out} = \varphi$$
$$AM_{in} = \sqrt{R_s P_{in}}$$

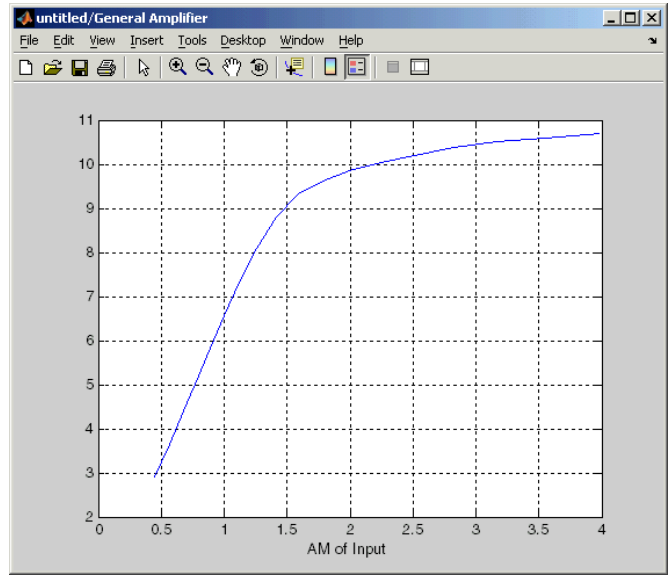
where AM_{in} is the AM of the input voltage, AM_{out} and PM_{out} are the AM and PM of the output voltage, R_s is the source resistance (50 ohms), R_l is the load resistance (50 ohms), P_{in} is the input power, P_{out} is the output power, and φ is the phase shift between the input and output voltage.

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

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

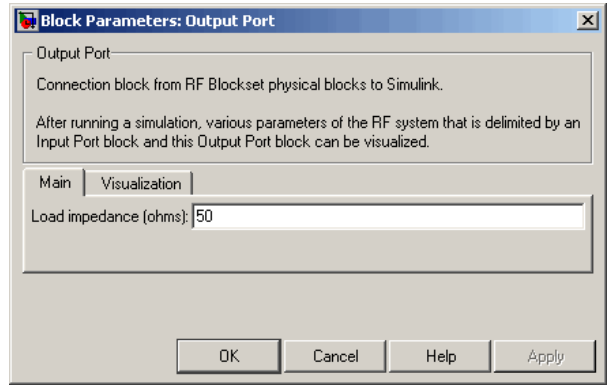


This figure shows the extracted AM/AM nonlinear conversion.



Dialog Box

Main Tab



Load impedance (ohms)

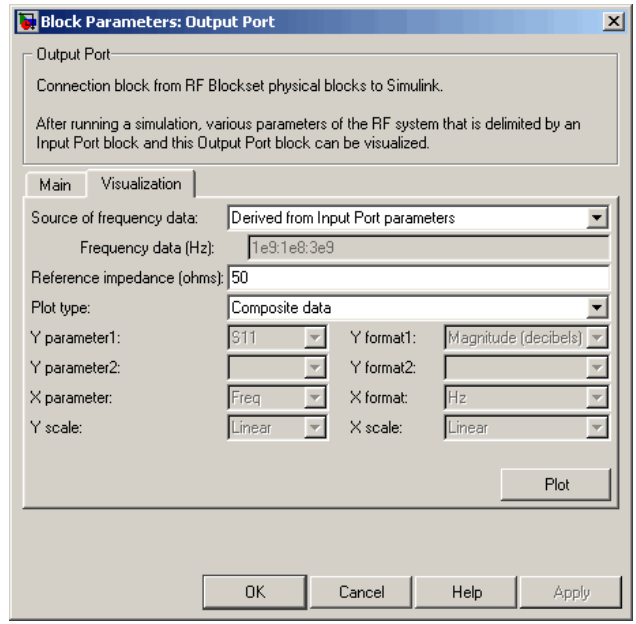
Load impedance of the RF network described in the physical model to which it connects.

Visualization Tab

This tab shows parameters for creating plots if you display the Output Port mask after you perform one or more of the following actions:

- Run a model with two or more blocks between the Input Port block and the Output Port block.
- Click the Update Diagram button to initialize a model with two or more blocks between the Input Port block and the Output Port block.

For information about plotting, see “Create Plots”.



See Also

Input Port

`s2y` (RF Toolbox)

Purpose

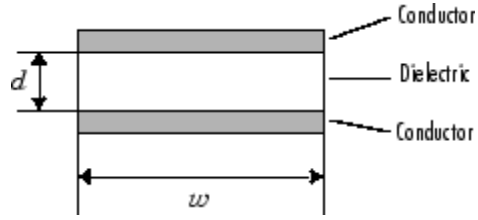
Model parallel-plate transmission line

Library

Transmission Lines sublibrary of the Physical library

Description

The Parallel-Plate Transmission Line block models the parallel-plate transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A parallel-plate transmission line is shown in cross-section in the following figure. Its physical characteristics include the plate width 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 first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

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

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

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

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

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

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

where

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

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

$$G = \omega\varepsilon''\frac{w}{d}$$

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

In these equations:

- σ_{cond} is the conductivity in the conductor.
- μ is the permeability of the dielectric.

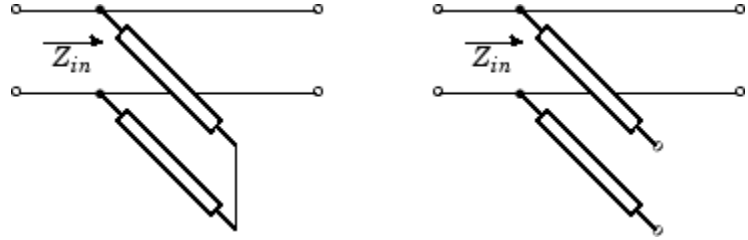
- ε is the permittivity of the dielectric.
- ε'' is the imaginary part of ε, $\varepsilon'' = \varepsilon_0\varepsilon_r \tan\delta$, where:
  - ε_0 is the permittivity of free space.
  - ε_r is the **Relative permittivity constant** parameter value.
  - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.
- f is a vector of modeling frequencies determined by the Output Port block.

Shunt and Series Stubs

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

Shunt ABCD-Parameters

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



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

$A = 1$

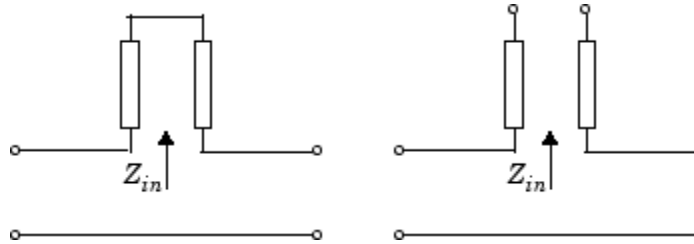
$B = 0$

$C = 1 / Z_{in}$

$D = 1$

Series ABCD-Parameters

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



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

$A = 1$

$B = Z_{in}$

$C = 0$

$D = 1$

Dialog Box

Main Tab

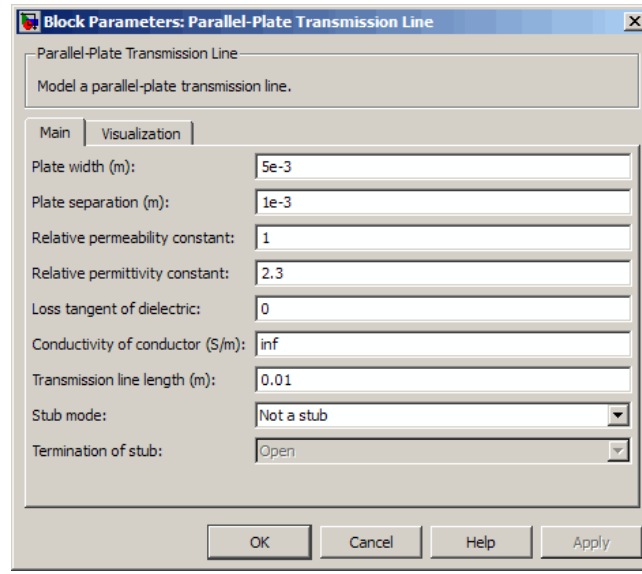


Plate width (m)
Physical width of the parallel-plate transmission line.

Plate separation (m)
Thickness of the dielectric separating the plates.

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

Loss tangent of dielectric
Loss angle tangent of the dielectric.

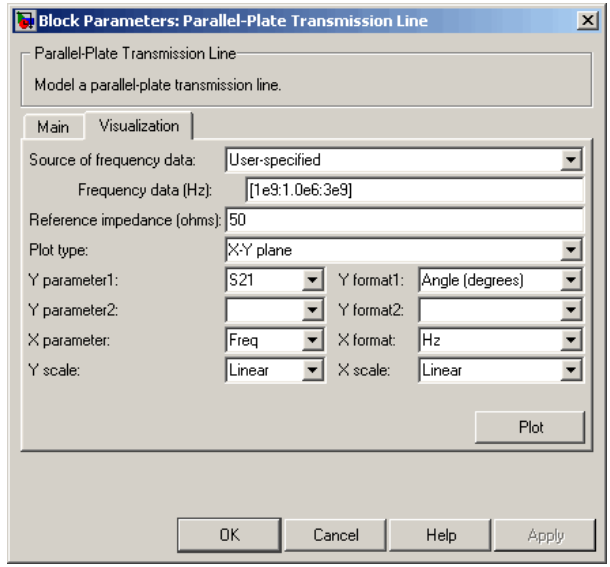
Conductivity of conductor (S/m)
Conductivity of the conductor in siemens per meter.

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

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

References

[1] Pozar, David M. *Microwave Engineering*, John Wiley & Sons, Inc., 2005.

See Also

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

RLCG Transmission Line

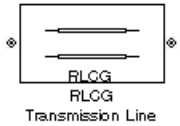
Purpose

Model RLCG transmission line

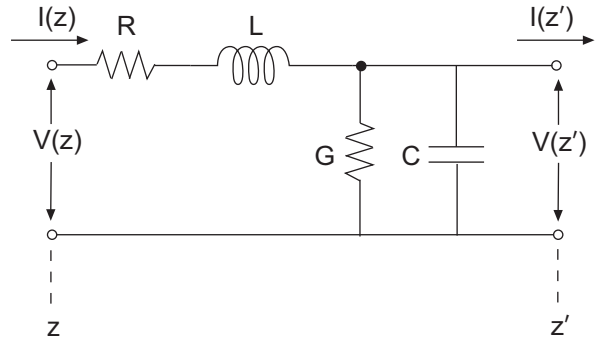
Library

Transmission Lines sublibrary of the Physical library

Description



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



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

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

Stubless Transmission Line

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

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

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

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

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

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

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

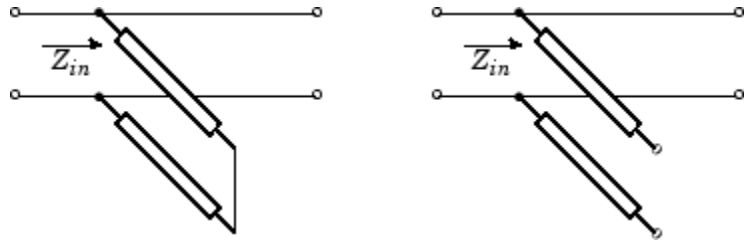
$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

Shunt and Series Stubs

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

Shunt ABCD-Parameters

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

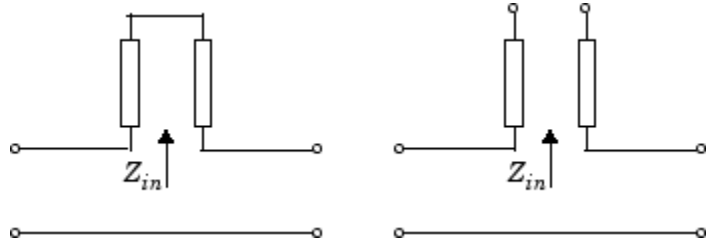


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

Series ABCD-Parameters

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

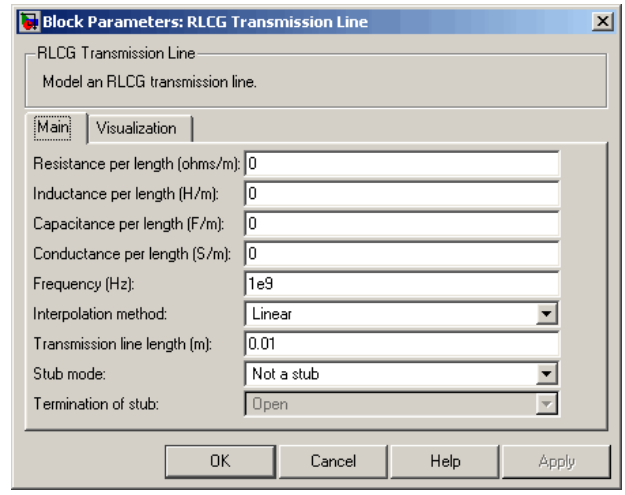


Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

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

Dialog Box

Main Tab



Resistance per length (ohms/m)
Vector of resistance values in ohms per meter.

Inductance per length (H/m)
Vector of inductance values in henries per meter.

Capacitance per length (F/m)
Vector of capacitance values in farads per meter.

Conductance per length (S/m)
Vector of conductance values in siemens per meter.

Frequency (Hz)
Vector of frequency values at which the resistance, inductance, capacitance, and conductance values are known.

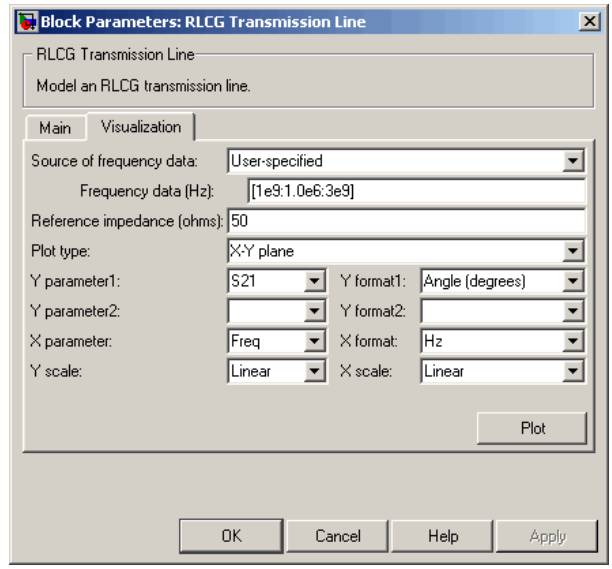
Interpolation method
Specify the interpolation method the block uses to calculate the parameter values at the modeling frequencies. Your choices are `Linear`, `Spline`, or `Cubic`.

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

Stub mode
Type of stub. Your choices are `Not a stub`, `Shunt`, or `Series`.

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

Visualization Tab



For information about plotting, see “Create Plots”.

References

[1] Pozar, David M. *Microwave Engineering*, John Wiley & Sons, Inc., 2005.

See Also

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

S-Parameters Amplifier

Purpose

Model nonlinear amplifier using S-parameters

Library

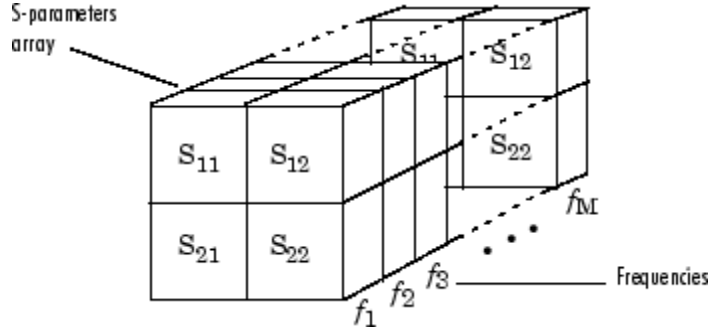
Amplifiers sublibrary of the Physical library

Description

The S-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent S-parameters, the frequencies and reference impedance of the S-parameters, noise data, and nonlinearity data.

Network Parameters

In the **S-parameters** field of the block dialog box, provide the S-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the S-parameters as an M-element vector. The elements of the frequencies vector must be in the same order as the S-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the S-parameters array and the vector of frequencies.



The S-Parameters Amplifier block interpolates the given S-parameters to determine their values at the modeling frequencies. See “SimRF Equivalent Baseband Algorithms” for more details.

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the S-Parameters Amplifier block dialog box. Depending on which of these parameters

you specify, the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a linear amplifier. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1 The block checks whether you have specified a value other than `Inf` for:

- The third-order intercept point ($OIP3$ or $IIP3$).
- The output power at the 1-dB compression point ($P_{1dB,out}$).
- The output power at saturation ($P_{sat,out}$).

In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2 The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$\begin{aligned} P_{sat,out} + GC_{sat} &= P_{sat,in} + G_{lin} \\ P_{1dB,out} + 1 &= P_{1dB,in} + G_{lin} \\ OIP3 &= IIP3 + G_{lin} \end{aligned}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$

$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$

$$0 = \frac{c_1}{IIP3} + c_3$$

The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Active Noise

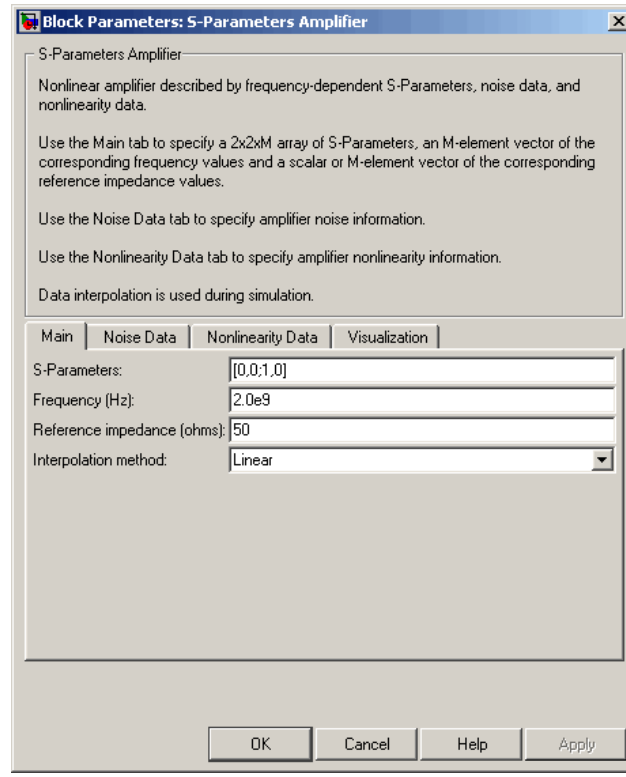
You can specify active block noise in one of the following ways:

- Spot noise data in the S-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the S-Parameters Amplifier block dialog box.

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

Dialog Box

Main Tab



S-Parameters

S-parameters for a nonlinear amplifier in a 2-by-2-by-M array. M is the number of S-parameters.

Frequency (Hz)

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

Reference impedance (ohms)

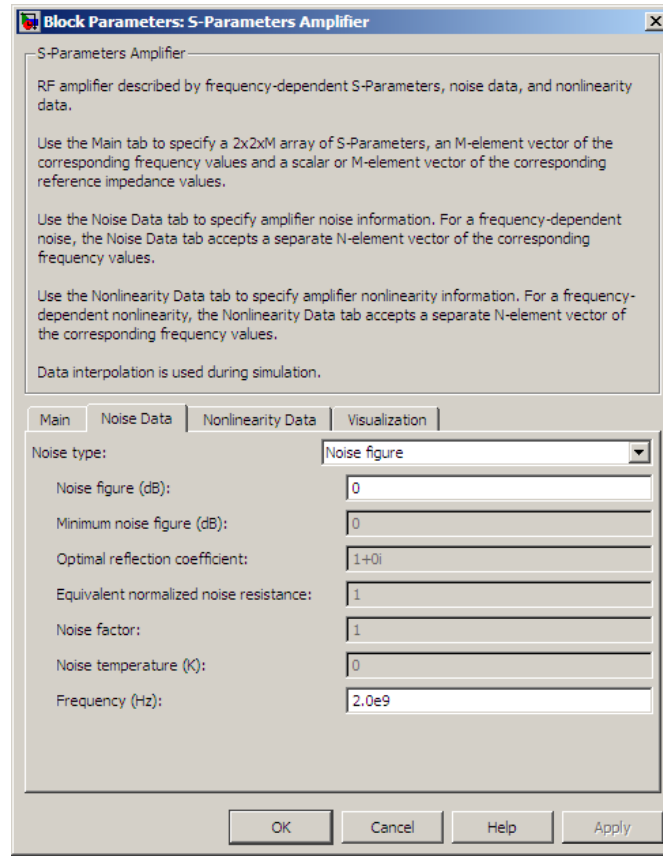
Reference impedance of the S-parameters as a scalar or a vector of length M. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method

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

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

Noise Data Tab



Noise type

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

Noise figure (dB)

Scalar ratio or vector of ratios, in decibels, of the available signal-to-noise power ratio at the input to the available

signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if **Noise type** is set to `Noise figure`.

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

Optimal reflection coefficient
Optimal amplifier source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

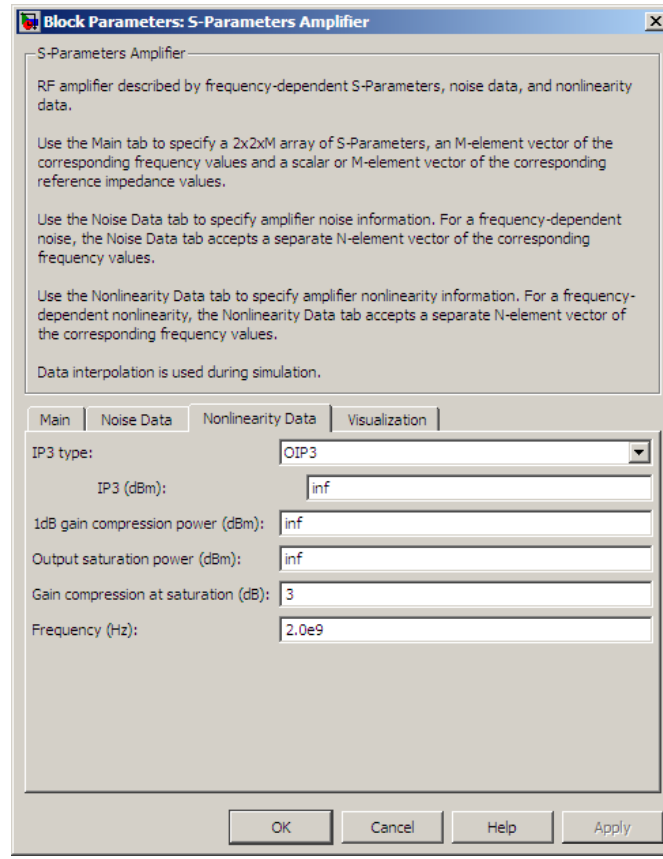
Equivalent normalized resistance
Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

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

Noise temperature (K)
Equivalent temperature or vector of temperatures that produce the same amount of noise power as the amplifier. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)
Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

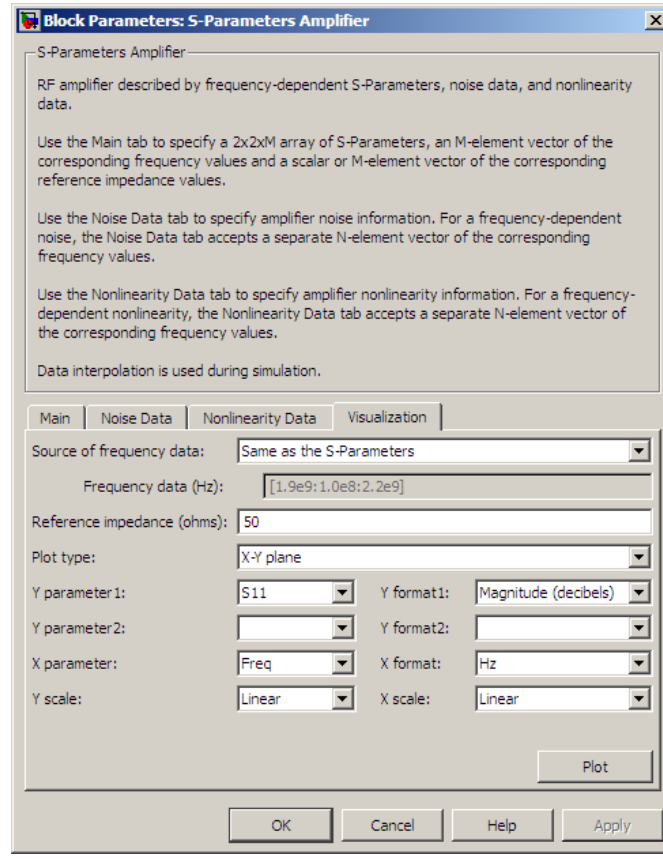
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

Plotting Parameters with the S-Parameters Amplifier Block

The following example specifies S-parameters [-.33+.71i, -.03i; 8.12-.02i, -.37-.37i] and [0.16+.20i, -.03-.04i; 7.71-8.04i, -.70-.12i] at frequencies 2.0 GHz and 2.1 GHz respectively, with a reference impedance of 50 ohms. The example uses the MATLAB `cat` function to create the 2-by-2-by-2 S-parameters array.

```
cat(3,[-.33+0.71i,     -.03i;  8.12-.02i, -.37-.37i],...
      [ .16+0.20i, -.03-.04i; 7.71-8.04i, -.70-.12i])
```

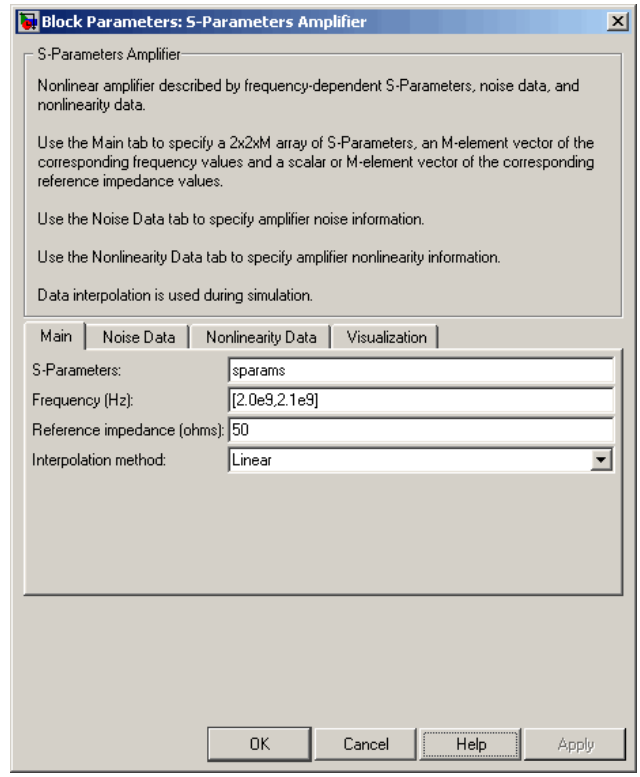
1. Type the following command at the MATLAB prompt to create a variable called `sparams` that stores the values of the S-parameters.

   ```
   sparams = cat(3,...
         [-.33+0.71i, -.03i;  8.12-.02i, -.37-.37i],...
         [ .16+0.20i, -.03-.04i; 7.71-8.04i, -.70-.12i])
   ```

2. Set the S-Parameters Amplifier block parameters on the **Main** tab as follows:

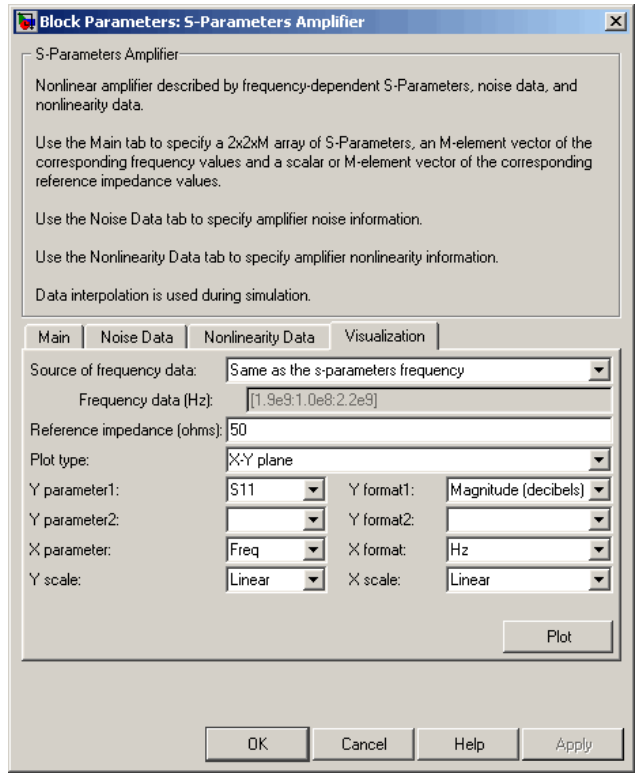
   - Set the **S-Parameters** parameter to `sparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

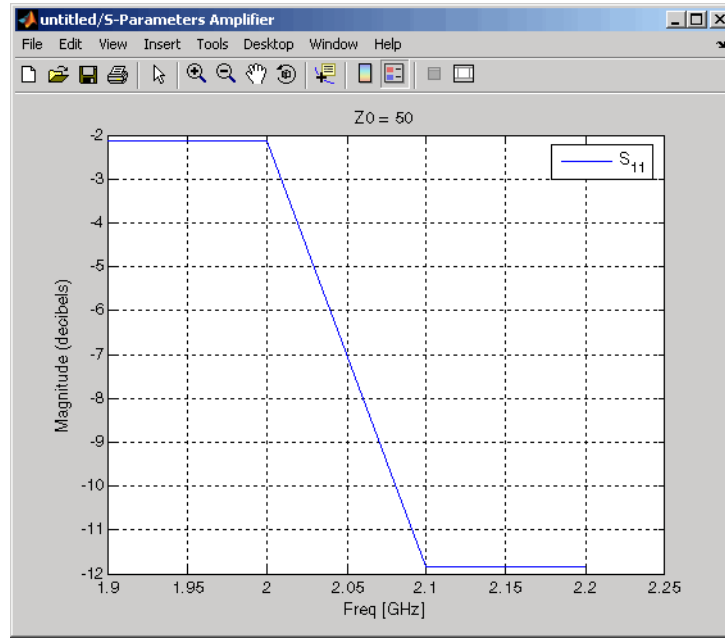


3 Set the S-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Plot type** list, select `X-Y plane`.
- In the **Y parameter1** list, select `S11`.



Click **Plot**. This action creates an X-Y Plane plot of the S_{11} parameters using the frequencies taken from the **Frequency (Hz)** parameter on the **Main** tab.



See Also

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

`interp1` (MATLAB)

S-Parameters Mixer

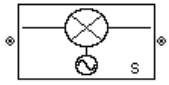
Purpose

Model mixer and local oscillator using S-parameters

Library

Mixer sublibrary of the Physical library

Description



The S-Parameters Mixer block models the nonlinear mixer described in the block dialog box, in terms of its frequency-dependent S-parameters, the frequencies and reference impedance of the S-parameters, noise data (including phase noise data), and nonlinearity data.

Network Parameters

The S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency. The other S-parameters also refer to the mixer input frequency.

The S-Parameters Mixer block interpolates the given S-parameters to determine their values at the modeling frequencies the Output Port block calculates. For more details about how the Output Port block calculates the modeling frequencies, see "SimRF Equivalent Baseband Algorithms".

SimRF Equivalent Baseband software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

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

where

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

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

Active Noise

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

- Spot noise data in the S-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the S-Parameters Mixer block dialog box.

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

Phase Noise

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

The blockset computes the digital filter by:

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

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

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the S-parameters Mixer block dialog box. Depending on which of these parameters you specify,

the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a mixer with a linear gain. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1 The block checks whether you have specified a value other than `Inf` for:

- The third-order intercept point ($OIP3$ or $IIP3$).
- The output power at the 1-dB compression point ($P_{1dB,out}$).
- The output power at saturation ($P_{sat,out}$).

In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2 The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$\begin{aligned} P_{sat,out} + GC_{sat} &= P_{sat,in} + G_{lin} \\ P_{1dB,out} + 1 &= P_{1dB,in} + G_{lin} \\ OIP3 &= IIP3 + G_{lin} \end{aligned}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$

$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$

$$0 = \frac{c_1}{IIP3} + c_3$$

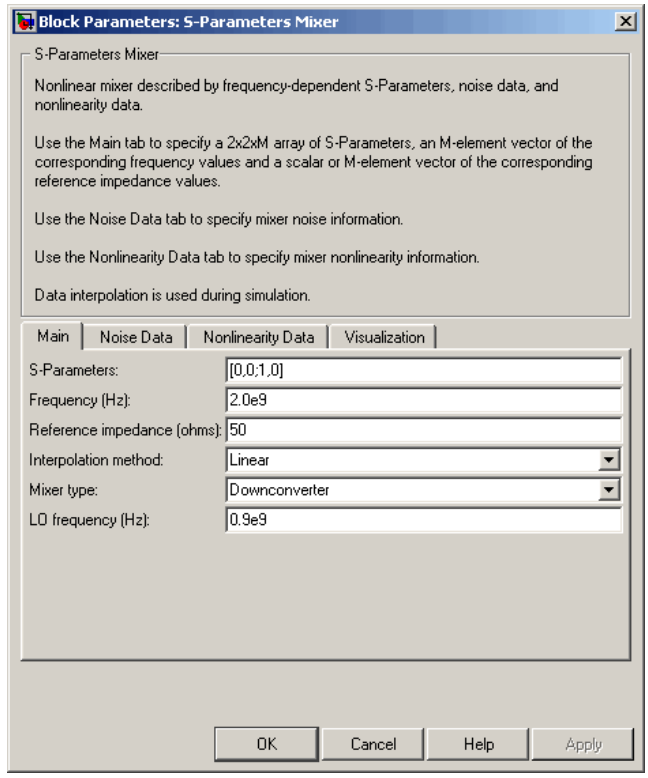
The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Dialog Box

Main Tab

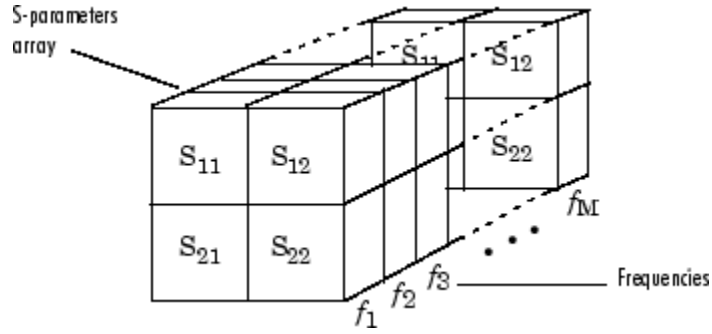


S-Parameters

S-parameters for a nonlinear mixer in a 2-by-2-by-M array. M is the number of S-parameters.

Frequency (Hz)

Frequencies of the S-parameters as an M-element vector. The order of the frequencies must correspond to the order of the S-parameters in **S-Parameters**. All frequencies must be positive. The following figure shows the correspondence between the S-parameters array and the vector of frequencies.



Reference impedance (ohms)

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

Interpolation method

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

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

Mixer Type

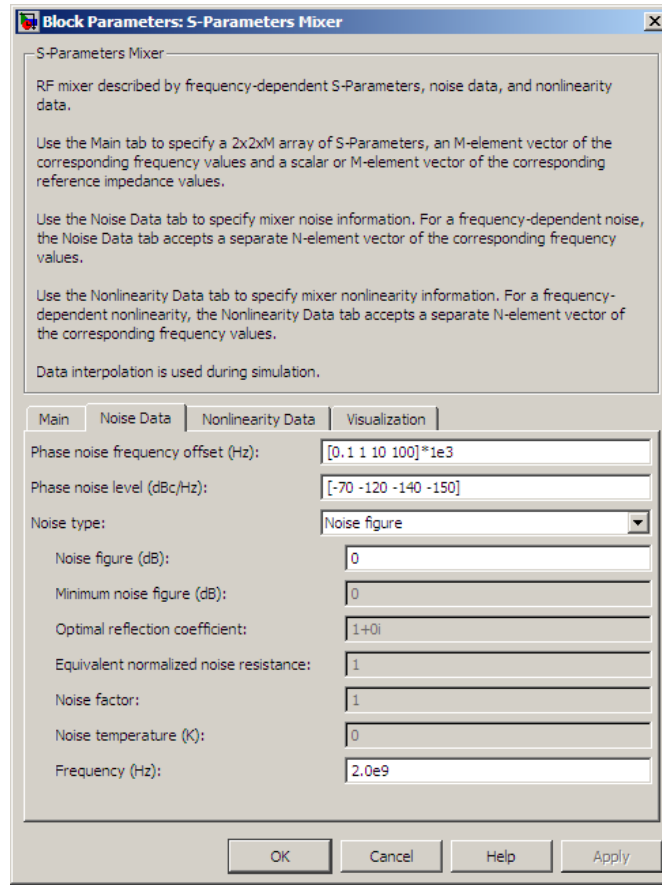
Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

LO frequency (Hz)

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

Note For a downconverting mixer, the local oscillator frequency must satisfy the condition $f_{in} - f_{lo} \geq 1/(2t_s)$, where t_s is the sample time specified in the Input Port block. Otherwise, an error appears.

Noise Data Tab



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

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

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

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

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

Optimal reflection coefficient
Optimal mixer source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance
Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

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

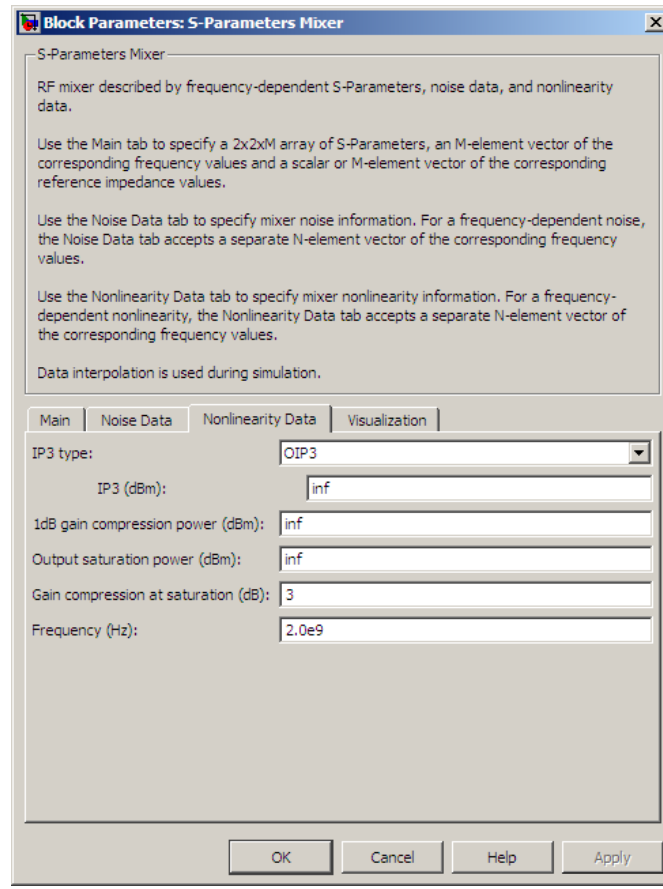
Noise temperature (K)

Equivalent temperature or vector of temperatures that produce the same amount of noise power as the mixer. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the mixer produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

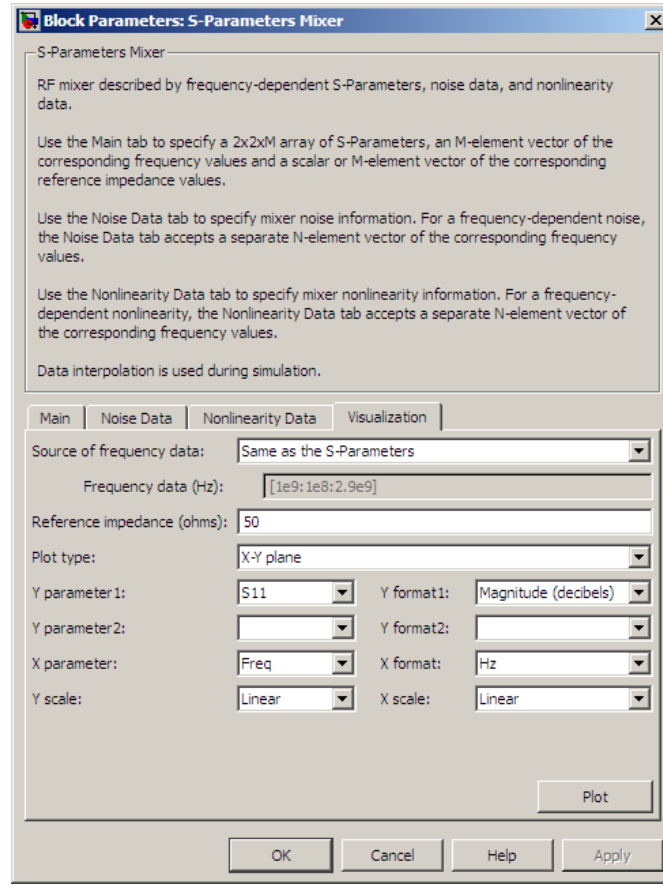
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”.

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

Purpose

Model passive network using S-parameters

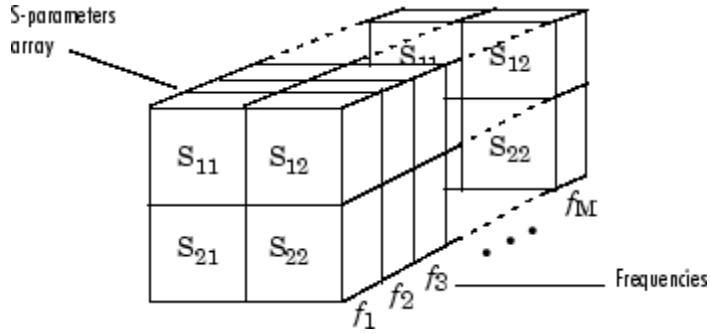
Library

Black Box Elements sublibrary of the Physical library

Description

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

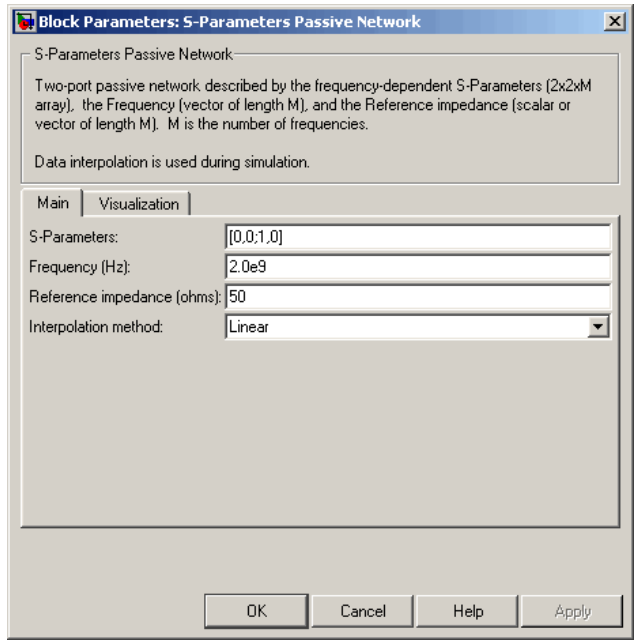
In the **S-Parameters** field of the block dialog box, provide the S-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the S-parameters as an M-element vector. The elements of the vector must be in the same order as the S-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the S-parameters array and the vector of frequencies.



The S-Parameters Passive Network block interpolates the given S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See "SimRF Equivalent Baseband Algorithms" for more details.

Dialog Box

Main Tab



S-Parameters

S-parameters for a two-port passive network in a 2-by-2-by-M array. M is the number of S-parameters.

Frequency (Hz)

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

Reference impedance (ohms)

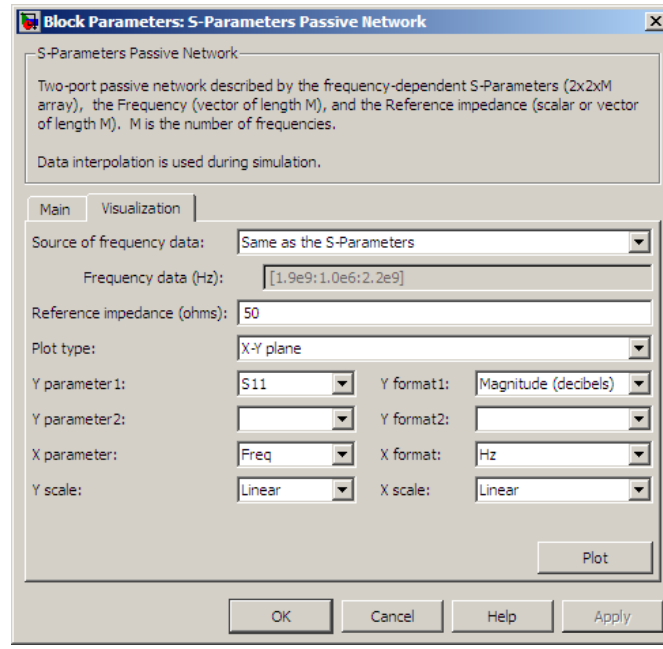
Reference impedance of the network as a scalar or a vector of length M. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

Plotting Parameters with the S-Parameters Passive Network Block

The following example specifies S-parameters [-.96-.23i, .03-.12i; .03-.12i, -.96-.23i] and [-.96-.11i, .02-.21i; .02-.21i, -.96-.11i] at frequencies 2.0 GHz and 2.1 GHz respectively. The example then plots these parameters.

The example first uses the MATLAB `cat` function to create the 2-by-2-by-2 S-parameters array.

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

You could also use the MATLAB `reshape` function. The following command produces the same result as previous command.

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

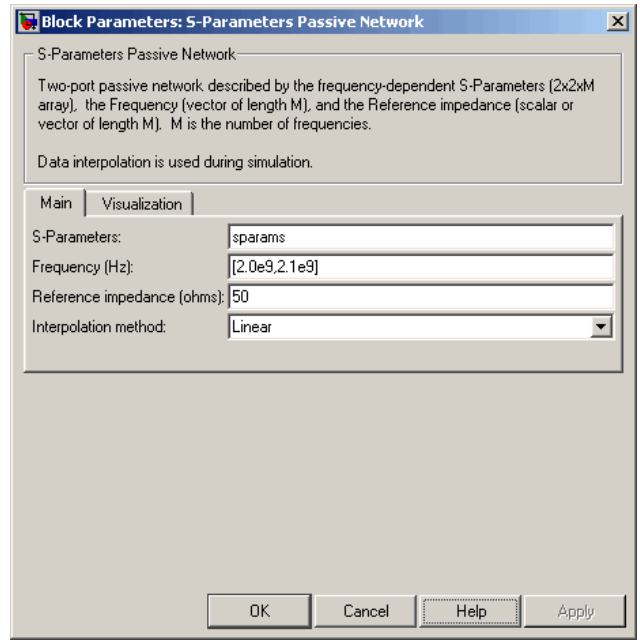
1. Type the following command at the MATLAB prompt to create a variable called `sparams` that stores the values of the S-parameters.

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

2. Set the S-Parameters Passive Network block parameters on the **Main** tab as follows:

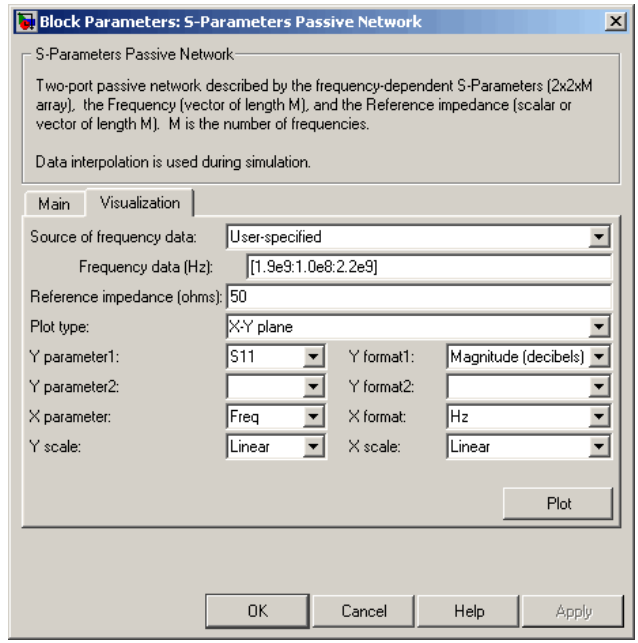
   - Set the **S-Parameters** parameter to `sparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

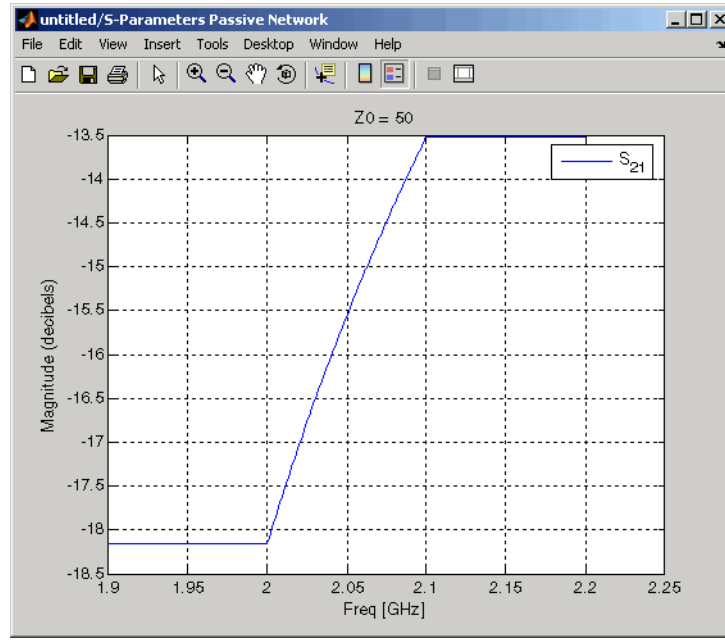


3 Set the S-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Y parameter1** list, select `S21`.



Click **Plot**. This action creates an X-Y Plane plot of the magnitude of the S_{21} parameters, in decibels, in the frequency range 1.9 to 2.2 GHz.



See Also

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

`interp1` (MATLAB)

Series C

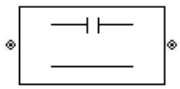
Purpose

Model series capacitor

Library

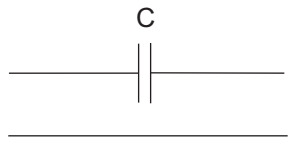
Ladders Filters sublibrary of the Physical library

Description



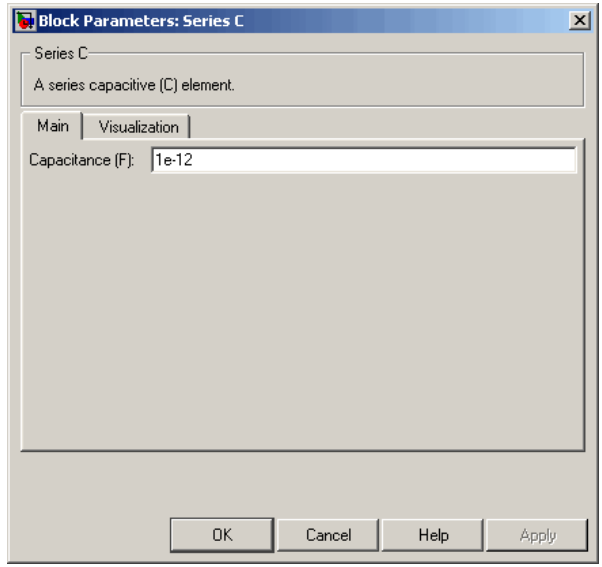
The Series C block models the series capacitor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series C object is a two-port network, as shown in the following circuit diagram.



Dialog Box

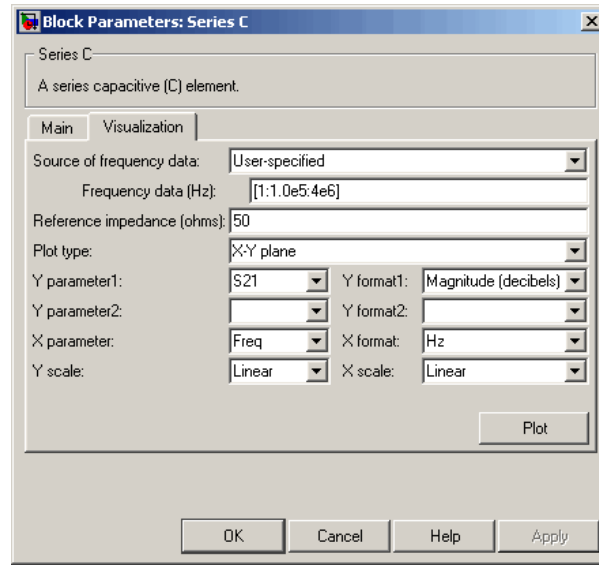
Main Tab



Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

Series L

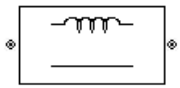
Purpose

Model series inductor

Library

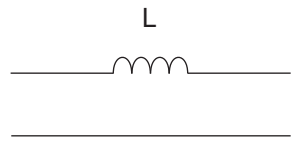
Ladders Filters sublibrary of the Physical library

Description



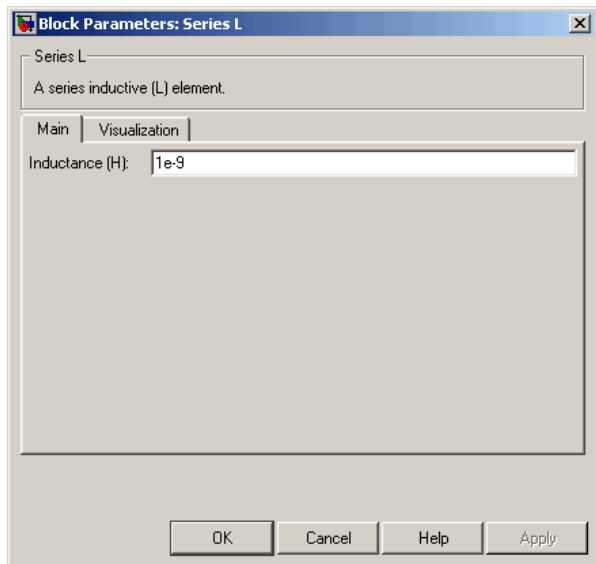
The Series L block models the series inductor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series L object is a two-port network, as shown in the following circuit diagram.



Dialog Box

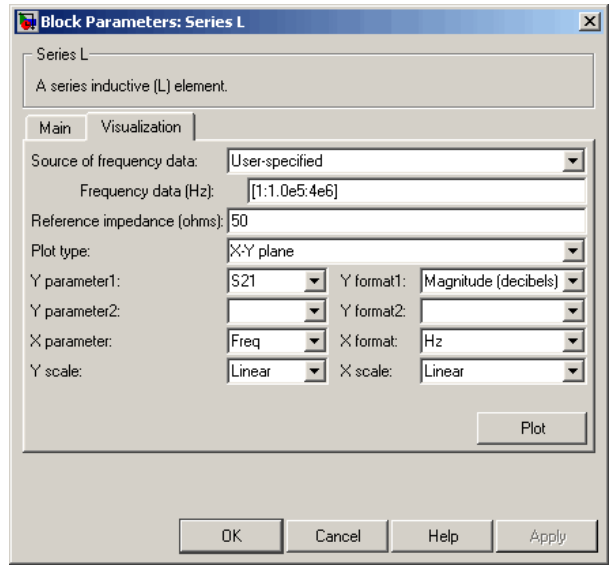
Main Tab



Inductance (H)

Scalar value for the inductance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

Series R

Purpose

Model series resistor

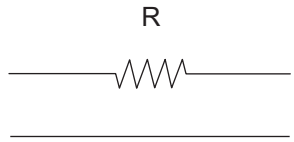
Library

Ladders Filters sublibrary of the Physical library

Description

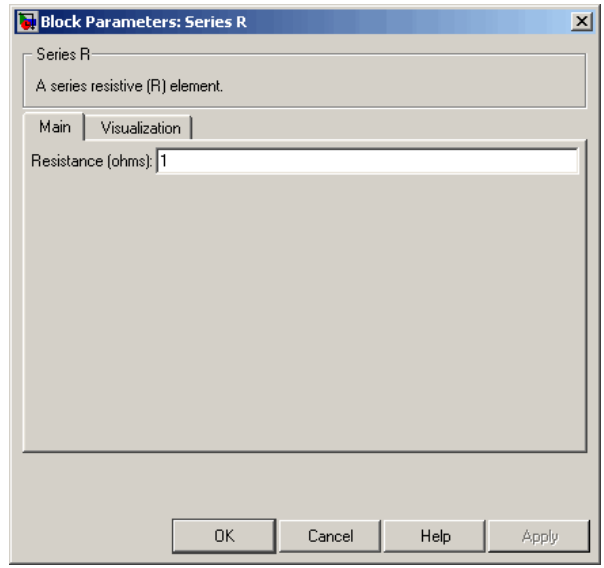
The Series R block models the series resistor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series R object is a two-port network, as shown in the following circuit diagram.



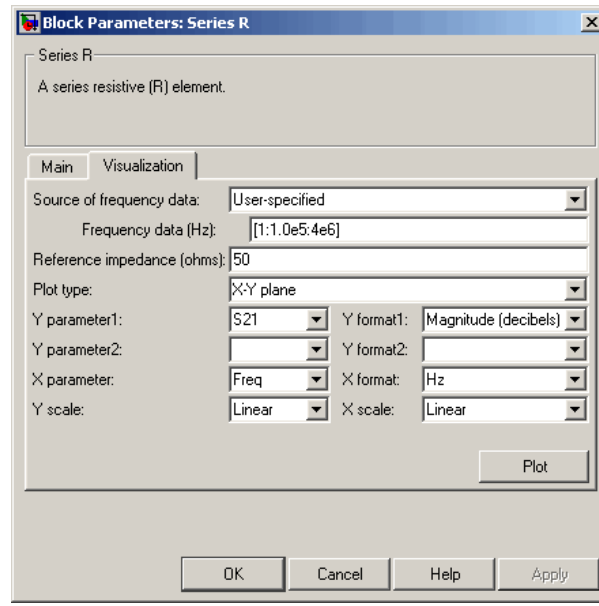
Dialog Box

Main Tab



Resistance (ohms)
Scalar value for the resistance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

Series RLC

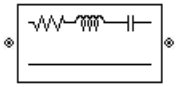
Purpose

Model series RLC network

Library

Ladders Filters sublibrary of the Physical library

Description



The Series RLC block models the series RLC network described in the block dialog box, in terms of its frequency-dependent S-parameters.

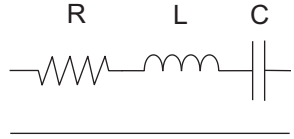
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference page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

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

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

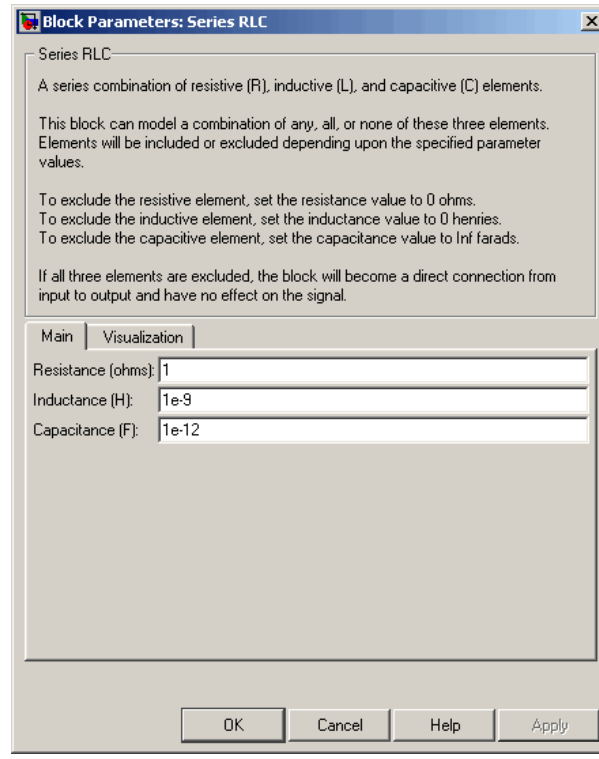
and $\omega = 2\pi f$.

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



Dialog Box

Main Tab



Resistance (ohms)

Scalar value for the resistance. The value must be nonnegative.

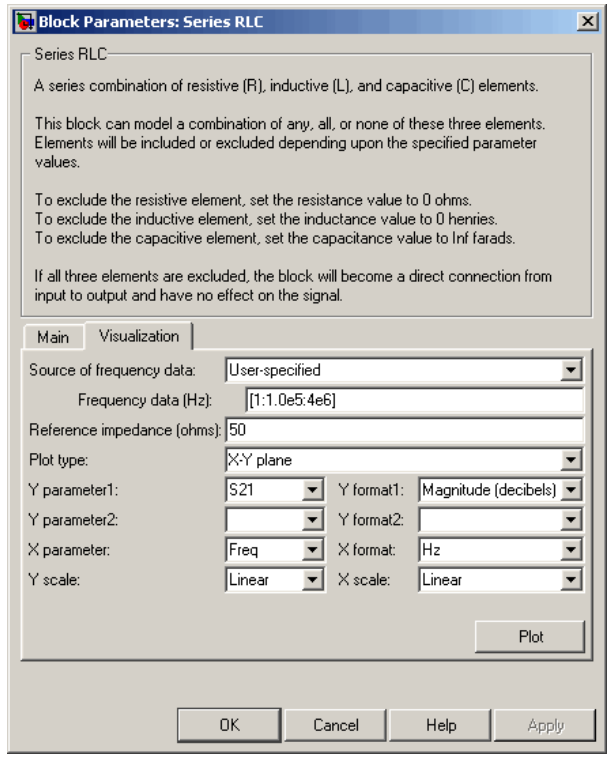
Inductance (H)

Scalar value for the inductance. The value must be nonnegative.

Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

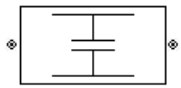
Purpose

Model shunt capacitor

Library

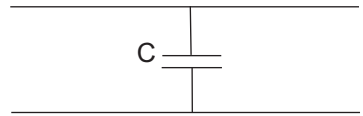
Ladders Filters sublibrary of the Physical library

Description



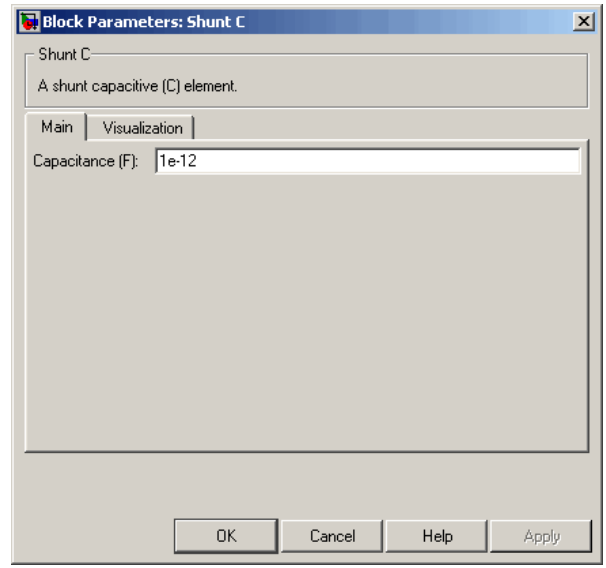
The Shunt C block models the shunt capacitor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt C object is a two-port network, as shown in the following circuit diagram.



Dialog Box

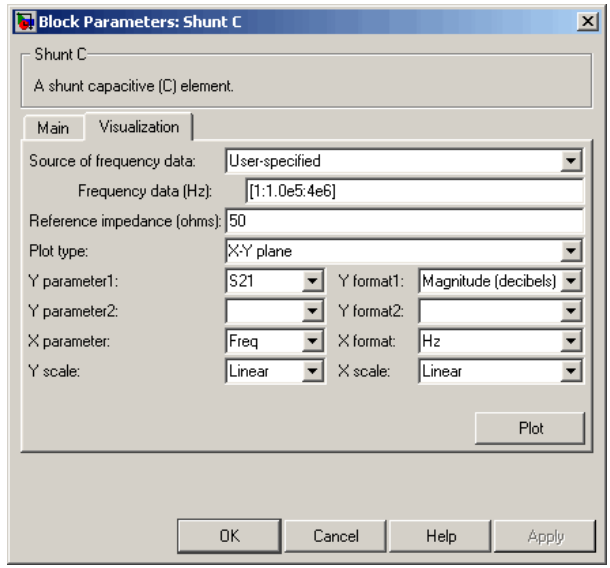
Main Tab



Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

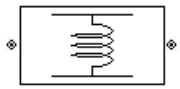
Purpose

Model shunt inductor

Library

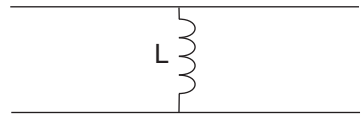
Ladders Filters sublibrary of the Physical library

Description



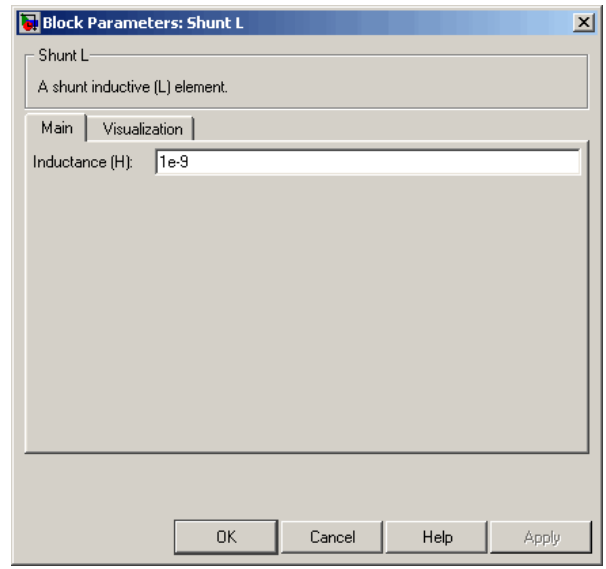
The Shunt L block models the shunt inductor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt L object is a two-port network, as shown in the following circuit diagram.



Dialog Box

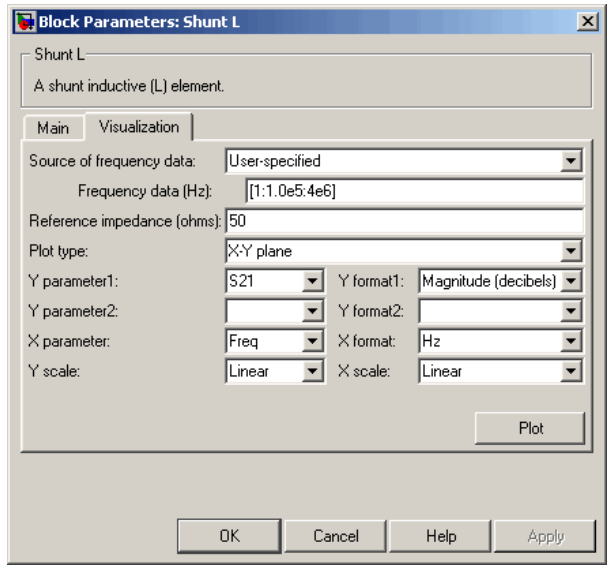
Main Tab



Inductance (H)

Scalar value for the inductance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

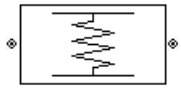
Purpose

Model shunt resistor

Library

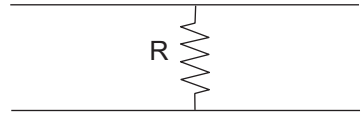
Ladders Filters sublibrary of the Physical library

Description



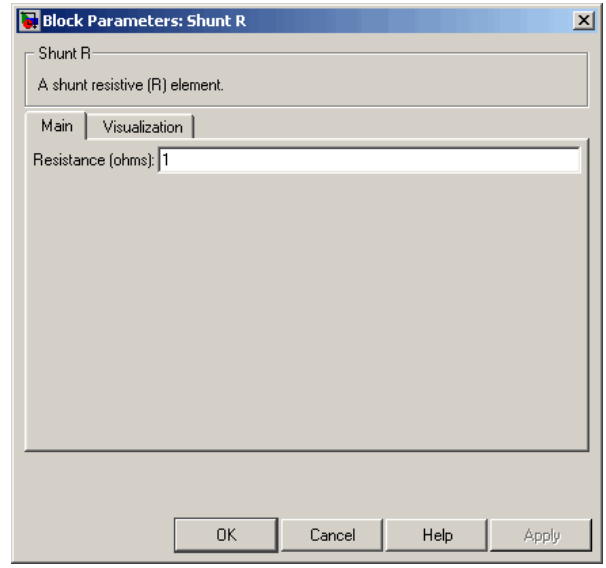
The Shunt R block models the shunt resistor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt R object is a two-port network, as shown in the following circuit diagram.



Dialog Box

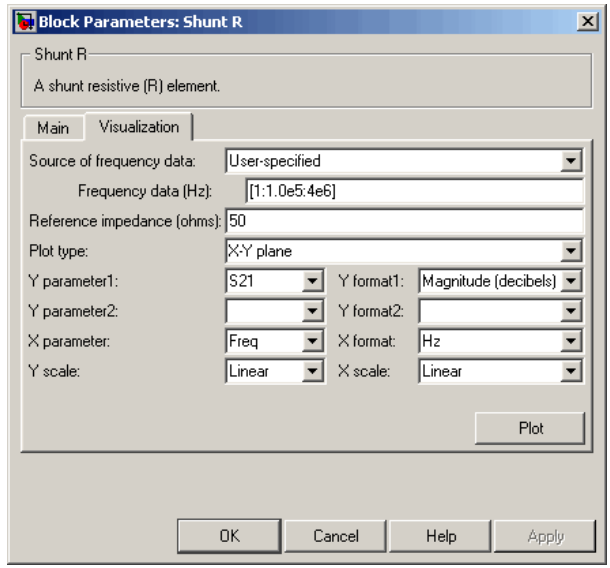
Main Tab



Resistance (ohms)

Scalar value for the resistance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

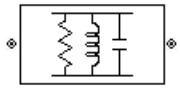
Purpose

Model shunt RLC network

Library

Ladders Filters sublibrary of the Physical library

Description



The Shunt RLC block models the shunt RLC network described in the block dialog box, in terms of its frequency-dependent S-parameters.

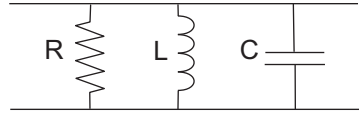
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference page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

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

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

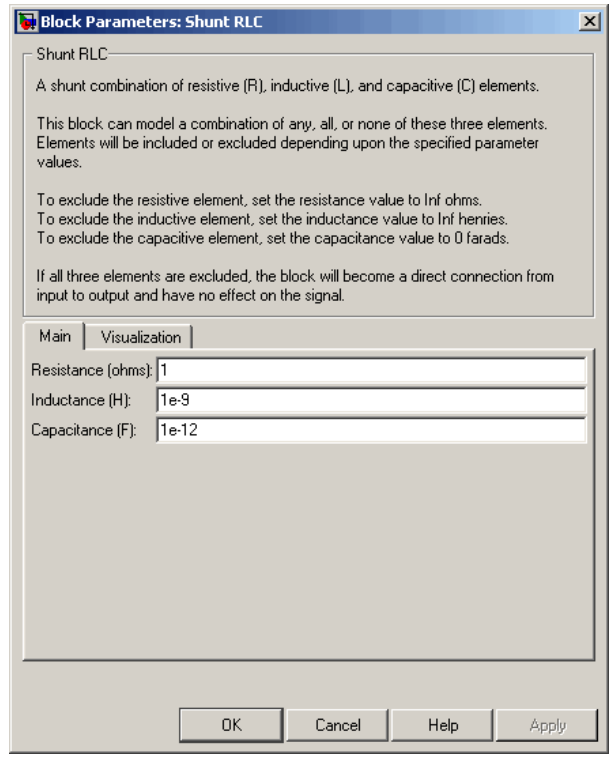
and $\omega = 2\pi f$.

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



Dialog Box

Main Tab

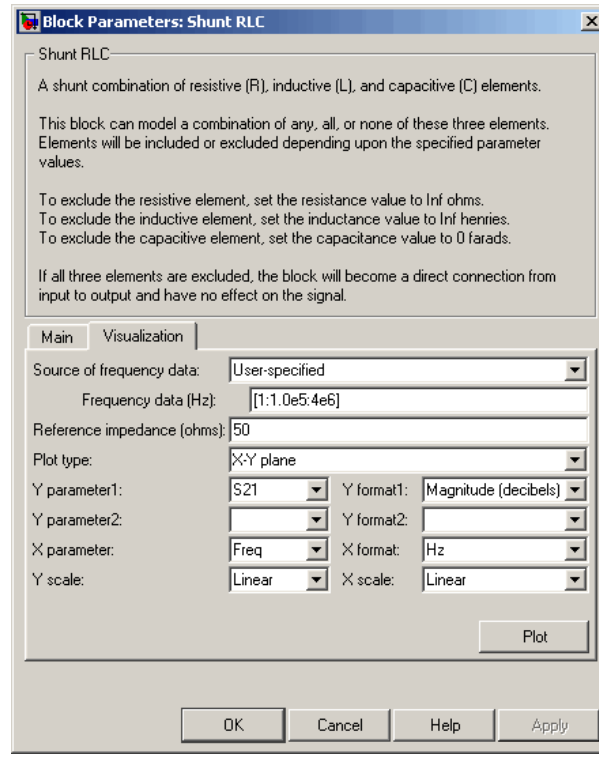


Resistance (ohms)
Scalar value for the resistance. The value must be nonnegative.

Inductance (H)
Scalar value for the inductance. The value must be nonnegative.

Capacitance (F)
Scalar value for the capacitance. The value must be nonnegative.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

Transmission Line (Equivalent Baseband)

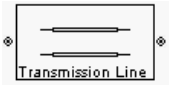
Purpose

Model general transmission line

Library

Transmission Lines sublibrary of the Physical library

Description



The Transmission Line block models the transmission line described in the block dialog box in terms of its physical parameters. The transmission line, which can be lossy or lossless, is treated as a two-port linear network.

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

Stubless Transmission Line

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

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

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

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

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

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

Z_0 is the specified characteristic impedance. k is a vector whose elements correspond to the elements of the input vector `freq`. The block calculates k from the specified parameters as $k = \alpha_a + i\beta$, where α_a is

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

$$\alpha_a = -\ln\left(10^{\alpha/20}\right)$$

The wave number β is related to the specified phase velocity, V_p, by

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

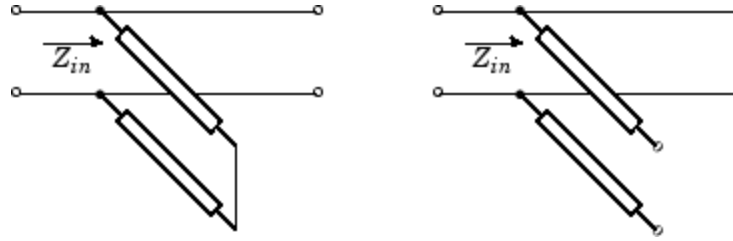
The phase velocity V_P is also known as the *wave propagation velocity*.

Shunt and Series Stubs

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

Shunt ABCD-Parameters

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



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

$A = 1$

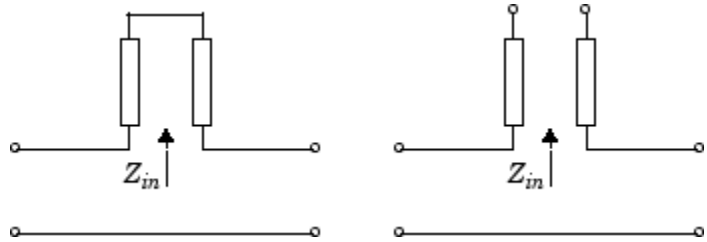
$B = 0$

$C = 1 / Z_{in}$

$D = 1$

Series ABCD-Parameters

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



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

$A = 1$

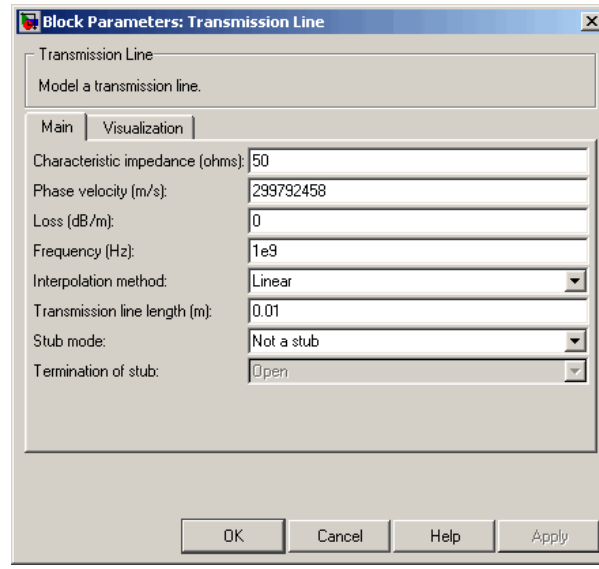
$B = Z_{in}$

$C = 0$

$D = 1$

Dialog Box

Main Tab



Characteristic impedance (ohms)

Characteristic impedance of the transmission line. The value can be complex.

Phase velocity (m/s)

Propagation velocity of a uniform plane wave on the transmission line.

Loss (dB/m)

Reduction in strength of the signal as it travels over the transmission line. Must be positive.

Frequency (Hz)

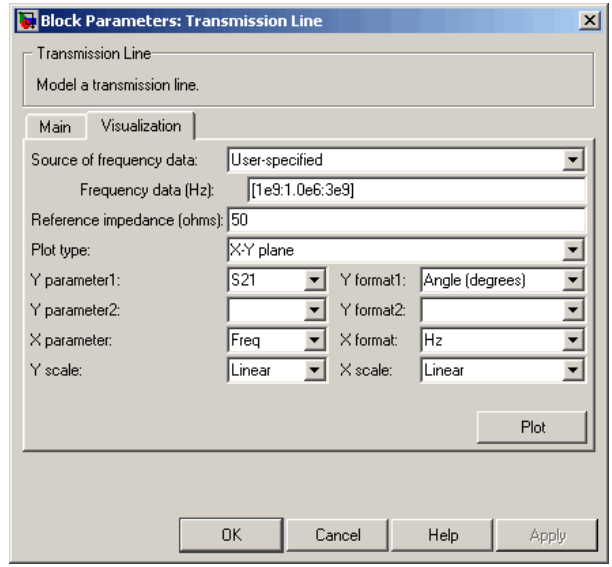
Vector of modeling frequencies. The block performs the calculations listed in the Description section at each frequency you provide.

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

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

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

Two-Wire Transmission Line

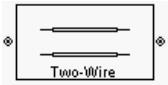
Purpose

Model two-wire transmission line

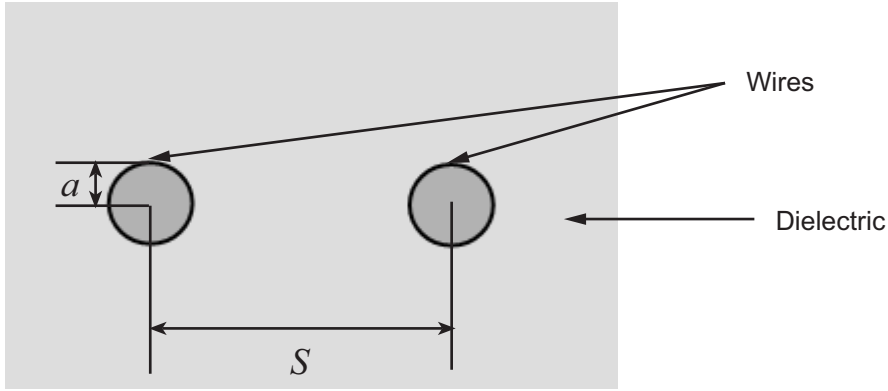
Library

Transmission Lines sublibrary of the Physical library

Description



The Two-Wire Transmission Line block models the two-wire transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A two-wire transmission line is shown in cross-section in the following figure. Its physical characteristics include the radius of the wires *a*, the separation or physical distance between the wire centers *S*, and the relative permittivity and permeability of the wires. SimRF Equivalent Baseband software assumes the relative permittivity and permeability are uniform.



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

Stubless Transmission Line

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

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

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

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

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

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

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

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

$$k = k_r + jk_i = \sqrt{(R + j\omega L)(G + j\omega C)}$$

where

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

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

$$G = \frac{\pi \omega \varepsilon''}{\operatorname{a\,cosh}\left(\frac{D}{2a}\right)}$$

$$C = \frac{\pi \varepsilon}{\operatorname{a\,cosh}\left(\frac{D}{2a}\right)}$$

and $\omega = 2\pi f$.

In these equations:

- σ_{cond} is the conductivity in the conductor.
- μ is the permeability of the dielectric.
- ε is the permittivity of the dielectric.
- ε'' is the imaginary part of ε, $\varepsilon'' = \varepsilon_0 \varepsilon_r \tan \delta$, where:
  - ε_0 is the permittivity of free space.
  - ε_r is the **Relative permittivity constant** parameter value.
  - $\tan \delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.
- f is a vector of modeling frequencies determined by the Output Port block.

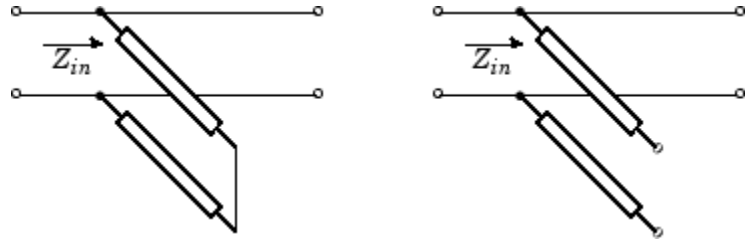
Shunt and Series Stubs

If you model the transmission line as a shunt or series stub, the Two-Wire Transmission Line block first calculates the ABCD-parameters at each frequency contained in the vector of

modeling frequencies. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters.

Shunt ABCD-Parameters

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

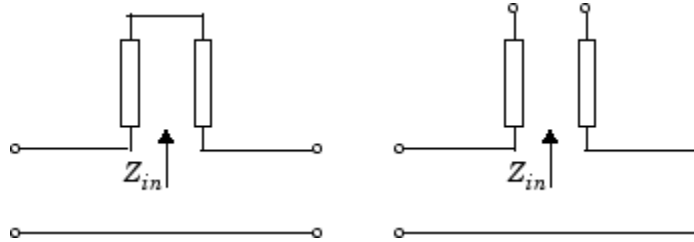


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

Series ABCD-Parameters

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

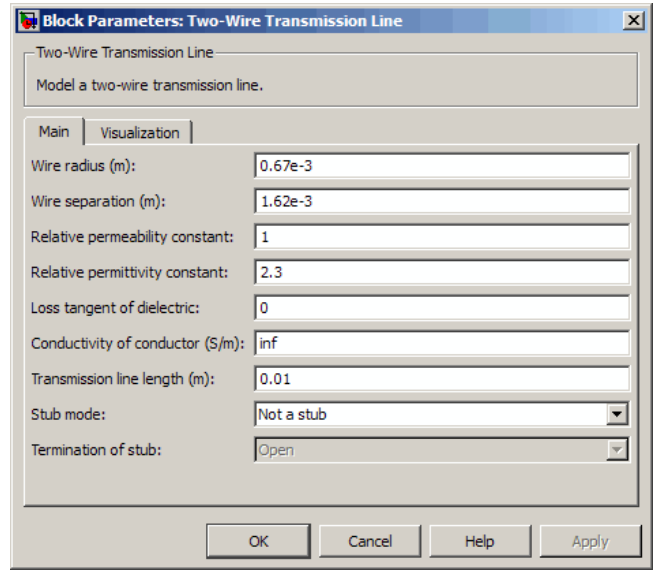


Z_{in} is the input impedance of the series circuit. The ABCD-parameters for the series stub are calculated as

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

Dialog Box

Main Tab



Wire radius (m)
Radius of the conducting wires of the two-wire transmission line.

Wire separation (m)
Physical distance between the wires.

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

Loss tangent of dielectric
Loss angle tangent of the dielectric.

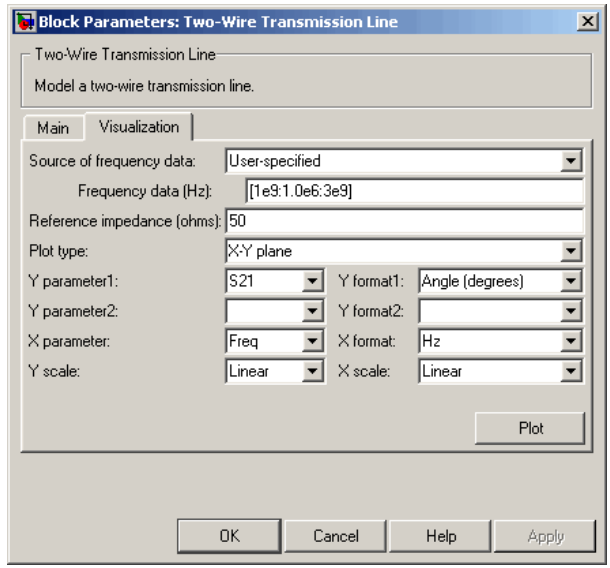
Conductivity of conductor (S/m)
Conductivity of the conductor in siemens per meter.

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

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

References

[1] Pozar, David M. *Microwave Engineering*, John Wiley & Sons, Inc., 2005.

See Also

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

Purpose

Model nonlinear amplifier using Y-parameters

Library

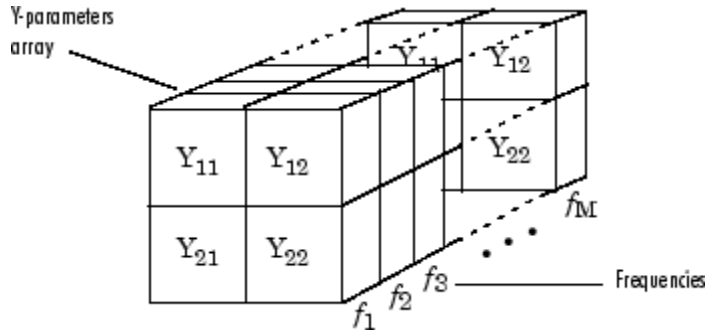
Amplifiers sublibrary of the Physical library

Description

The Y-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent, the frequencies of the Y-parameters, noise data, and nonlinearity data

Network Parameters

In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Y-parameters as an M-element vector. The elements of the frequencies vector must be in the same order as the Y-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Y-parameters array and the vector of frequencies.



The Y-Parameters Amplifier block uses the RF Toolbox `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See “SimRF Equivalent Baseband Algorithms” for more details.

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Y-Parameters

Amplifier block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a linear amplifier. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1. The block checks whether you have specified a value other than `Inf` for:

   - The third-order intercept point ($OIP3$ or $IIP3$).
   - The output power at the 1-dB compression point ($P_{1dB,out}$).
   - The output power at saturation ($P_{sat,out}$).

   In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2. The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$\begin{aligned} P_{sat,out} + GC_{sat} &= P_{sat,in} + G_{lin} \\ P_{1dB,out} + 1 &= P_{1dB,in} + G_{lin} \\ OIP3 &= IIP3 + G_{lin} \end{aligned}$$

   where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$

$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$

$$0 = \frac{c_1}{IIP3} + c_3$$

The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Active Noise

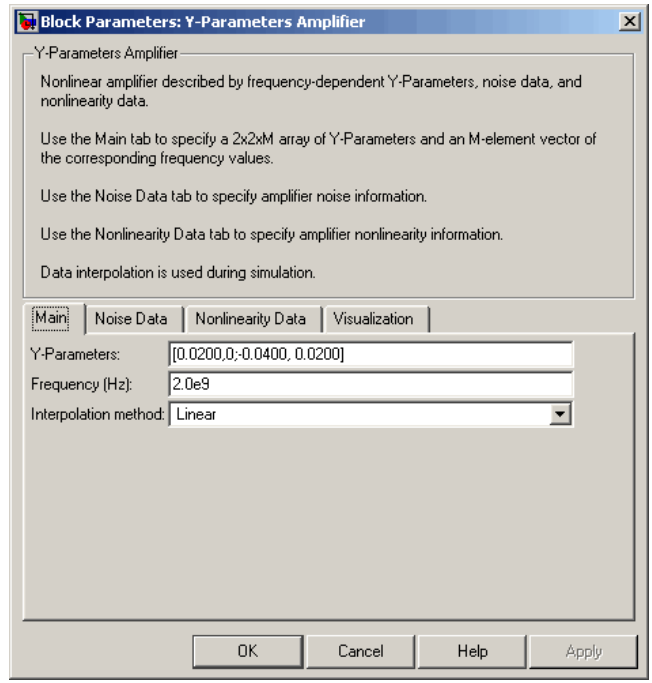
You can specify active block noise in one of the following ways:

- Spot noise data in the Y-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the Y-Parameters Amplifier block dialog box.

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

Dialog Box

Main Tab



Y-Parameters

Y-parameters for a nonlinear amplifier in a 2-by-2-by-M array. M is the number of Y-parameters.

Frequency (Hz)

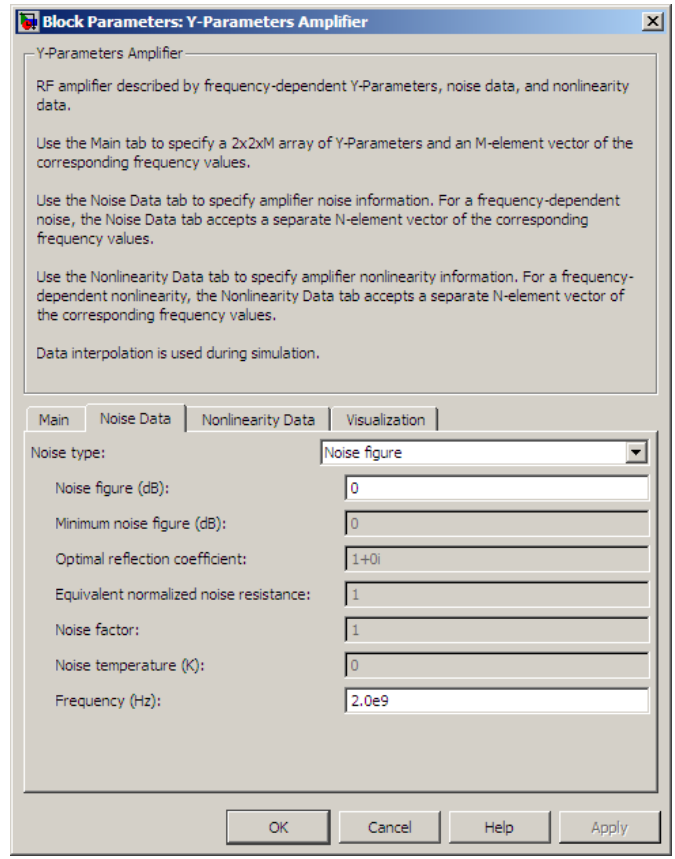
Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in **Y-Parameters**. All frequencies must be positive.

Interpolation method

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

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

Noise Data Tab



Noise type

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

Noise figure (dB)

Scalar ratio or vector of ratios, in decibels, of the available signal-to-noise power ratio at the input to the available

signal-to-noise power ratio at the output, $(S_i/N_i)/(S_o/N_o)$. This parameter is enabled if **Noise type** is set to `Noise figure`.

Minimum noise figure (dB)

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

Optimal reflection coefficient

Optimal amplifier source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance

Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

Noise factor

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

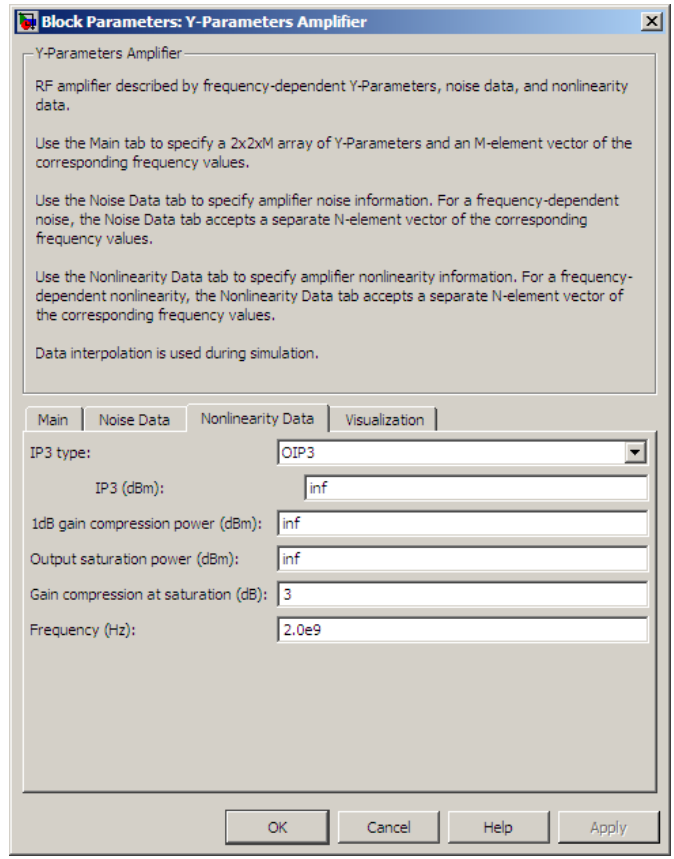
Noise temperature (K)

Equivalent temperature or vector of temperatures that produce the same amount of noise power as the amplifier. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

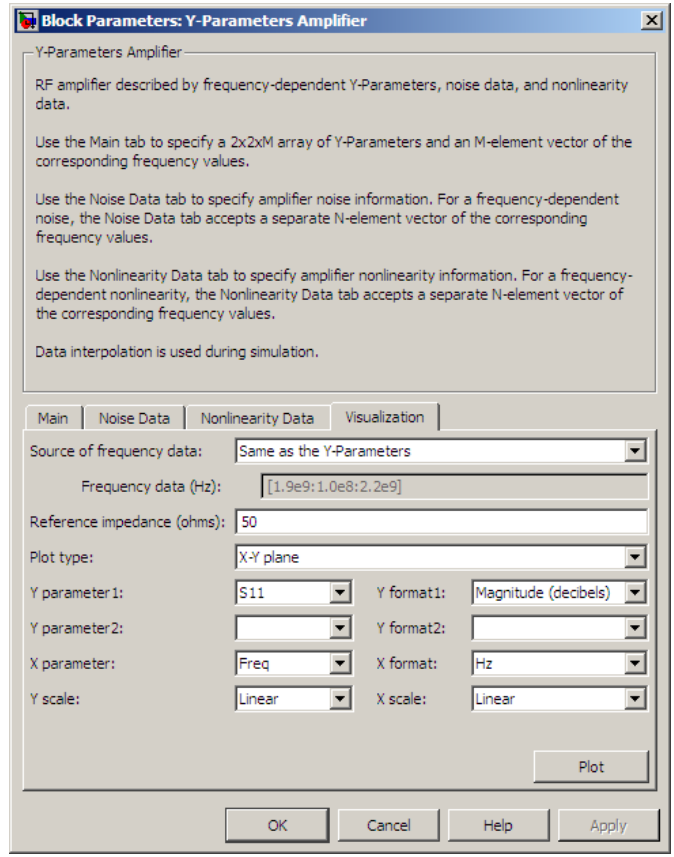
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

Plotting Parameters with the Y-Parameters Amplifier Block

The following example specifies Y-parameters [-.06+.58i, -.08i; 1.14-1.82i, -.07+.28i] and [.02-.21i, 0.03i; -.21+.72i, .03-.11i] at frequencies 2.0 GHz and 2.1 GHz respectively. It uses the MATLAB `cat` function to create the 2-by-2-by-2 Y-parameters array

```
cat(3,[-.06+.58i, -.08i; 1.14-1.82i, -.07+.28i],...
      [ .02-.21i, 0.03i;  -.21+.72i,  .03-.11i])
```

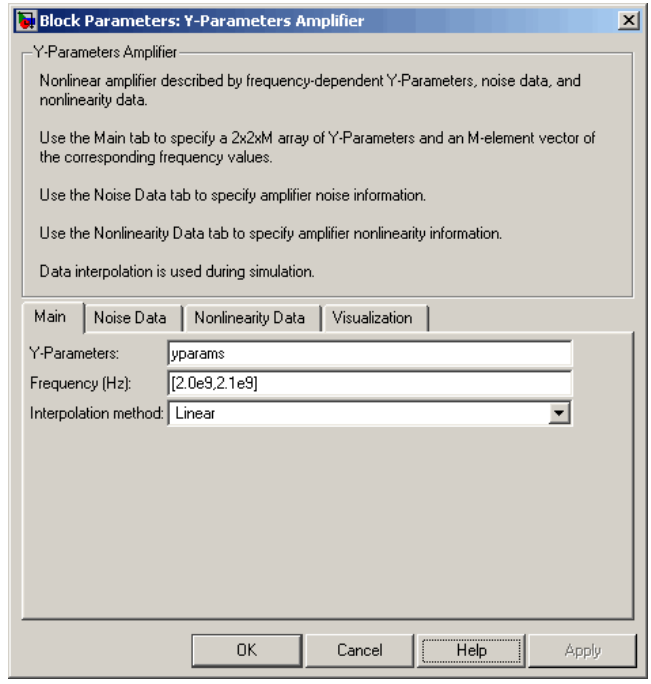
1. Type the following command at the MATLAB prompt to create a variable called `yparams` that stores the values of the Y-parameters.

   ```
   yparams = cat(3,...
         [-.06+.58i, -.08i; 1.14-1.82i, -.07+.28i],...
         [ .02-.21i, 0.03i;  -.21+.72i,  .03-.11i])
   ```

2. Set the Y-Parameters Amplifier block parameters on the **Main** tab as follows:

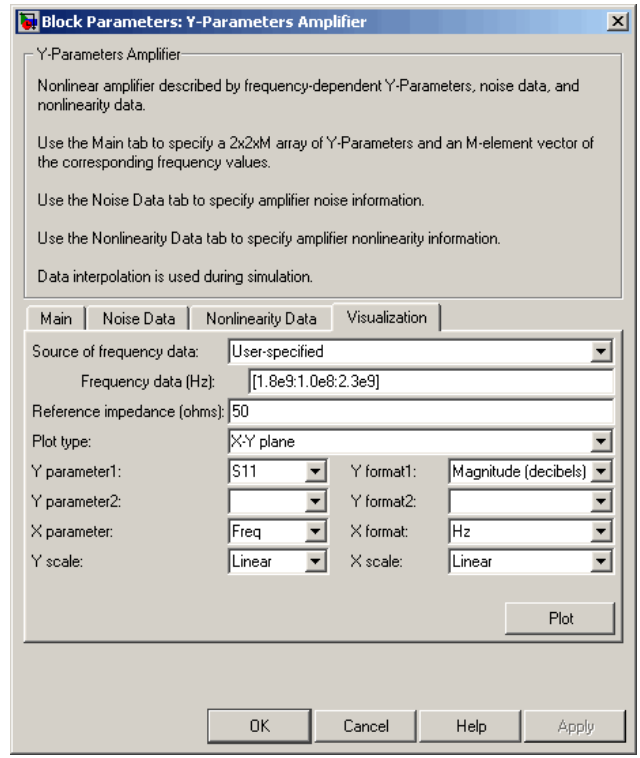
   - Set the **Y-Parameters** parameter to `yparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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



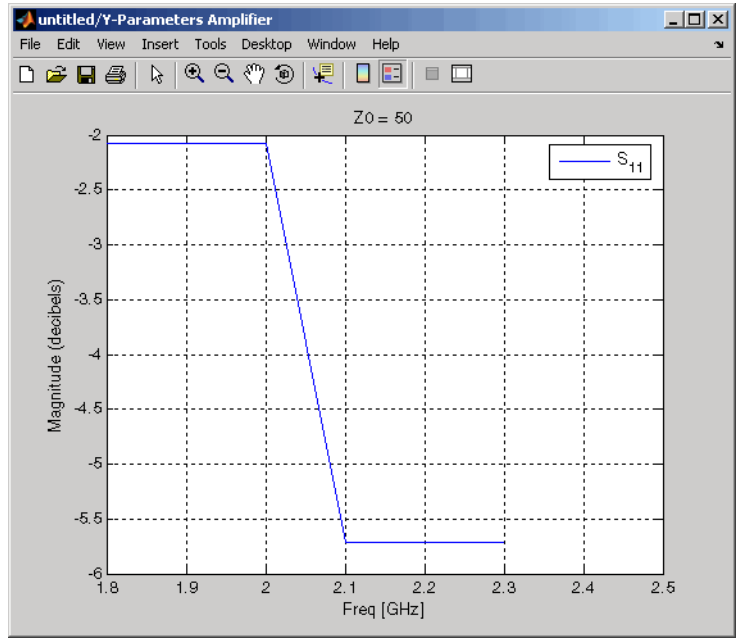
3 Set the Y-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.8e9:1.0e8:2.3e9]`.
- In the **Plot type** list, select `X-Y plane`.
- In the **Y parameter1** list, select `S11`.



Click **Plot**. This action creates an X-Y Plane plot of the S_{11} parameters in the frequency range 1.8 to 2.3 GHz.

Y-Parameters Amplifier



See Also

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

`y2s` (RF Toolbox)

`interp1` (MATLAB)

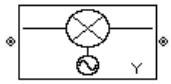
Purpose

Model mixer and local oscillator using Y-parameters

Library

Mixer sublibrary of the Physical library

Description



The Y-Parameters Mixer block models the nonlinear mixer described in the block dialog box in terms of its frequency-dependent Y-parameters, the frequencies of the Y-parameters, noise data (including phase noise data), and nonlinearity data.

Network Parameters

The Y-parameter values all refer to the mixer input frequency.

The Y-Parameters Mixer block uses the RF Toolbox `y2s` function to convert the Y-parameters to S-parameters and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See "SimRF Equivalent Baseband Algorithms" for more details.

SimRF Equivalent Baseband software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

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

where

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

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

Active Noise

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

- Spot noise data in the Y-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the Y-Parameters Mixer block dialog box.

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

Phase Noise

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

The blockset computes the digital filter by:

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

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

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Y-Parameters Mixer block dialog box. Depending on which of these parameters you specify,

the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a mixer with a linear gain. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1 The block checks whether you have specified a value other than `Inf` for:

- The third-order intercept point ($OIP3$ or $IIP3$).
- The output power at the 1-dB compression point ($P_{1dB,out}$).
- The output power at saturation ($P_{sat,out}$).

In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2 The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$\begin{aligned} P_{sat,out} + GC_{sat} &= P_{sat,in} + G_{lin} \\ P_{1dB,out} + 1 &= P_{1dB,in} + G_{lin} \\ OIP3 &= IIP3 + G_{lin} \end{aligned}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three

parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$

$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$

$$0 = \frac{c_1}{IIP3} + c_3$$

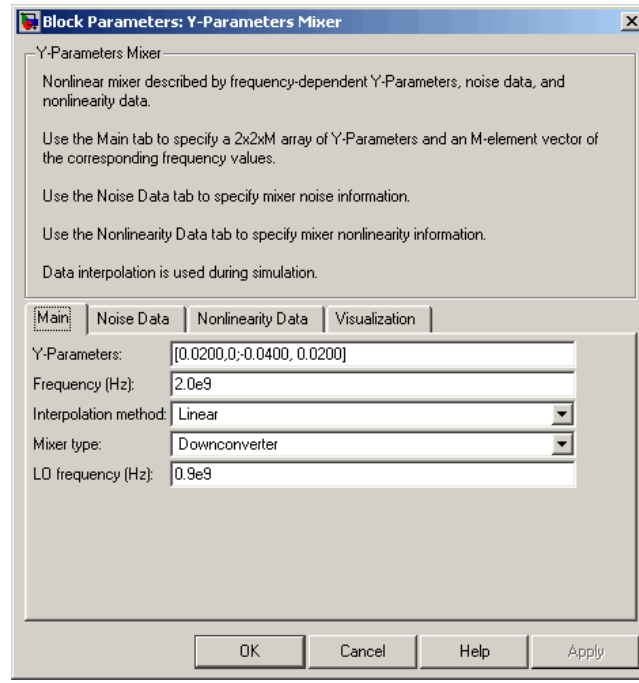
The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Dialog Box

Main Tab

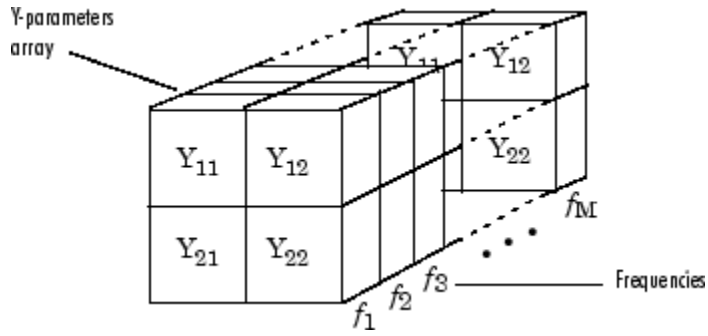


Y-Parameters

Y-parameters for a nonlinear mixer in a 2-by-2-by-M array. M is the number of Y-parameters.

Frequency (Hz)

Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in **Y-Parameters**. All frequencies must be positive. The following figure shows the correspondence between the Y-parameters array and the vector of frequencies.



Interpolation method

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

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

Mixer Type

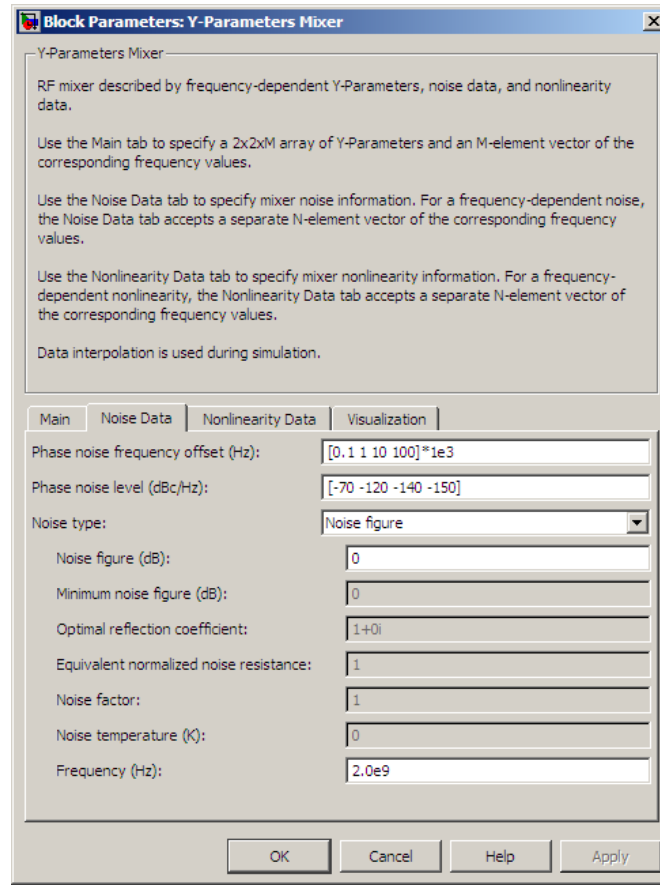
Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

LO frequency (Hz)

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

Note For a downconverting mixer, the local oscillator frequency must satisfy the condition $f_{in} - f_{lo} \geq 1/(2t_s)$, where t_s is the sample time specified in the Input Port block. Otherwise, an error appears.

Noise Data Tab



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

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

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

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

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

Optimal reflection coefficient
Optimal mixer source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance
Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

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

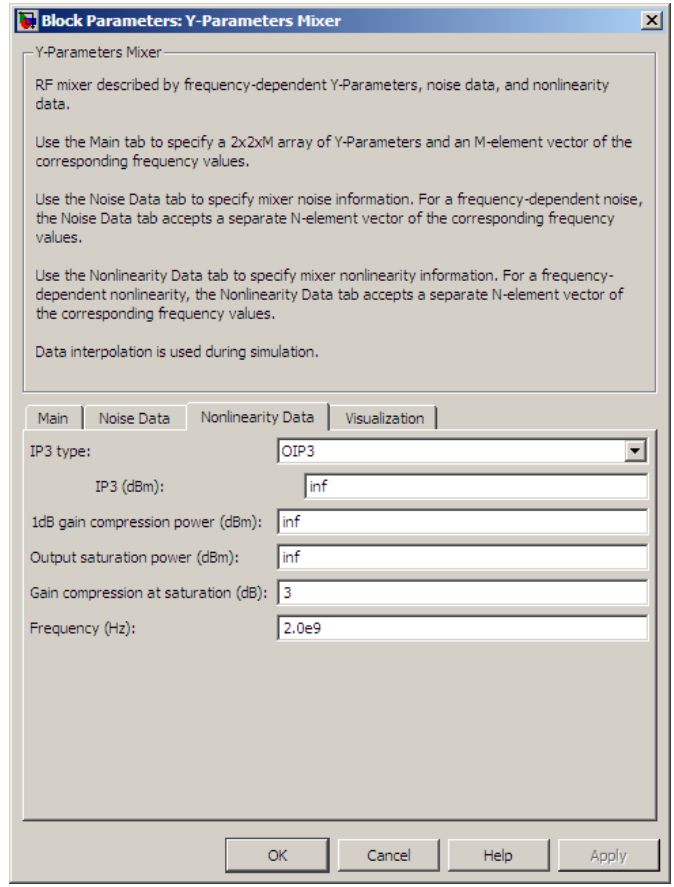
Noise temperature (K)

Equivalent temperature or vector of temperatures that produce the same amount of noise power as the mixer. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the mixer produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

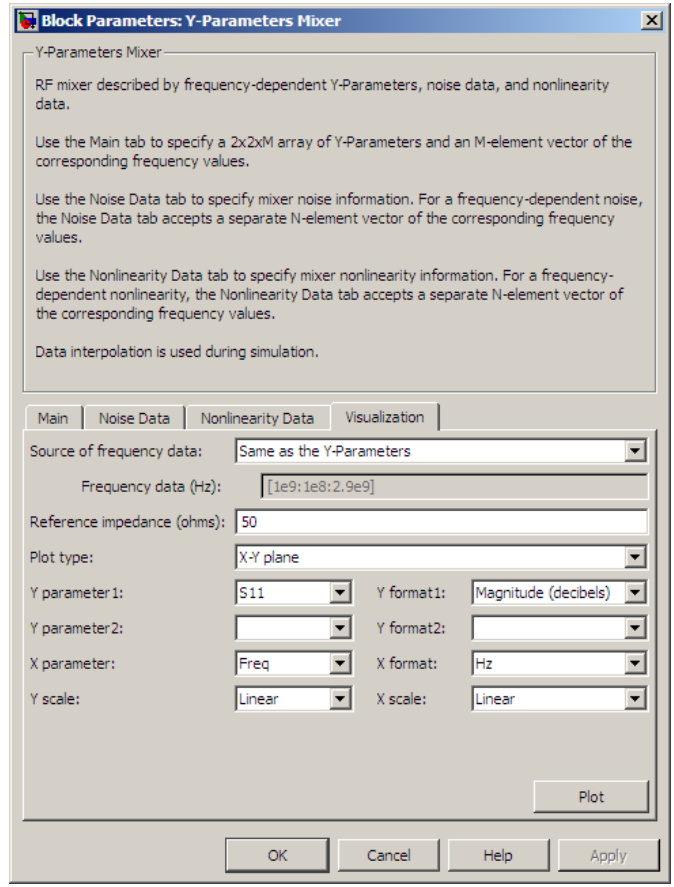
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

Y-Parameters Passive Network

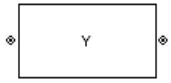
Purpose

Model passive network using Y-parameters

Library

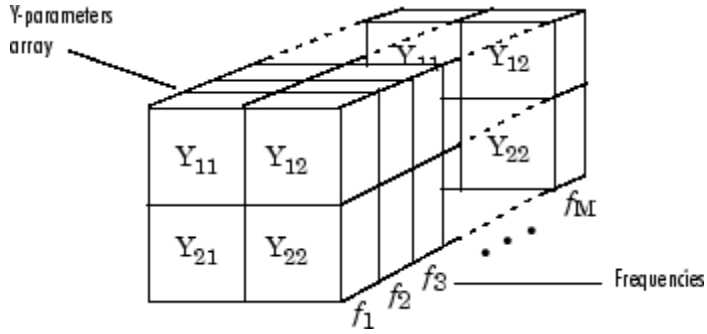
Black Box Elements sublibrary of the Physical library

Description



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

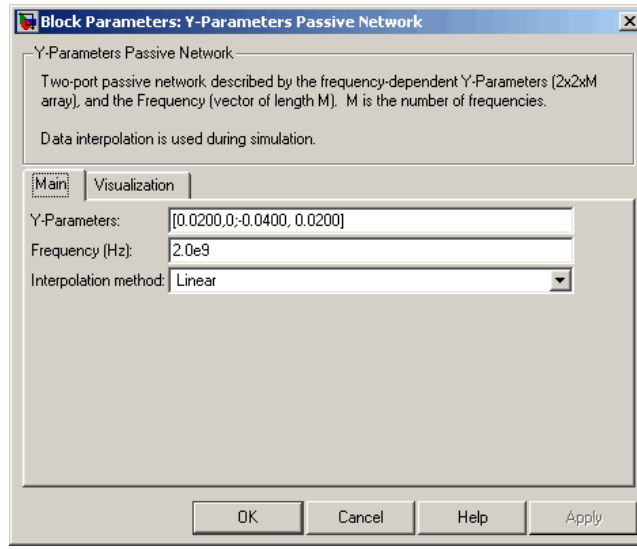
In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Y-parameters as an M-element vector. The elements of the vector must be in the same order as the Y-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Y-parameters array and the vector of frequencies.



The Y-Parameters Passive Network block uses the RF Toolbox `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See “SimRF Equivalent Baseband Algorithms” for more details.

Dialog Box

Main Tab



Y-Parameters

Y-parameters for a two-port passive network in a 2-by-2-by-M array. M is the number of Y-parameters.

Frequency (Hz)

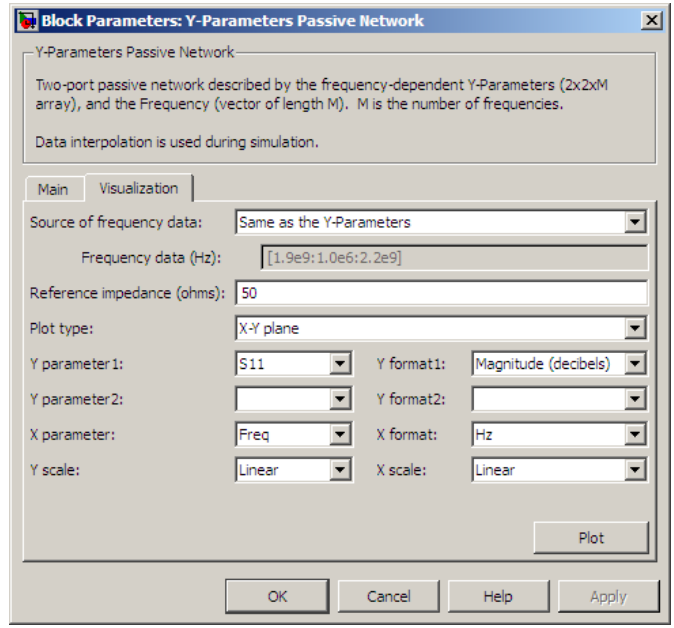
Frequencies of the Y-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Y-parameters in **Y-Parameters**. All frequencies must be positive.

Interpolation method

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

Plotting Parameters with the Y-Parameters Passive Network Block

The following example specifies Y-parameters [.23i, -.12i; -.12i, .23i] and [.02-.13i, -.02+.25i; -.02+.25i, .02-.13i] at frequencies 2.0 GHz and

2.1 GHz respectively. It uses the MATLAB `cat` function to create the 2-by-2-by-2 Y-parameters array.

```
cat(3,[.23i,-.12i;-.12i,.23i],...
      [.02-.13i,-.02+.25i;-.02+.25i, .02-.13i])
```

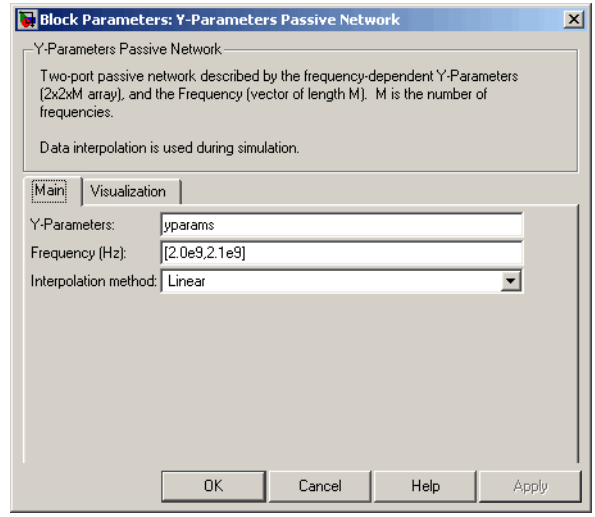
1. Type the following command at the MATLAB prompt to create a variable called `yparams` that stores the values of the Y-parameters.

   ```
   yparams = cat(3,[.23i,-.12i;-.12i,.23i],...
         [.02-.13i,-.02+.25i;-.02+.25i, .02-.13i])
   ```

2. Set the Y-Parameters Passive Network block parameters on the **Main** tab as follows:

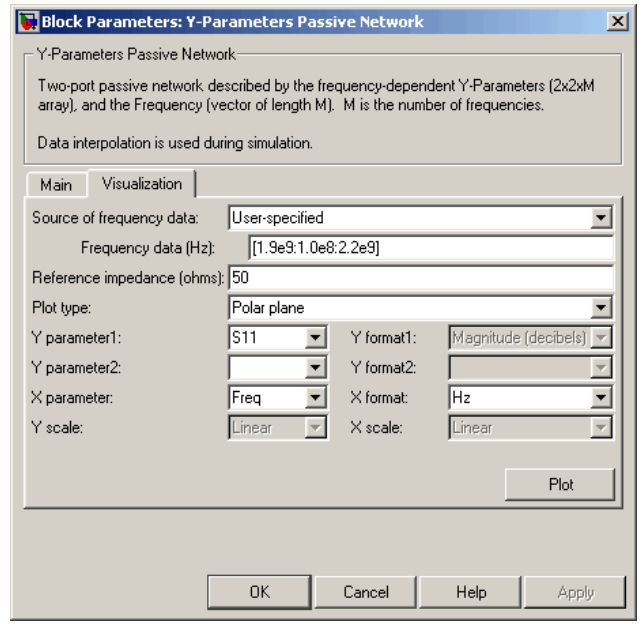
   - Set the **Y-Parameters** parameter to `yparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

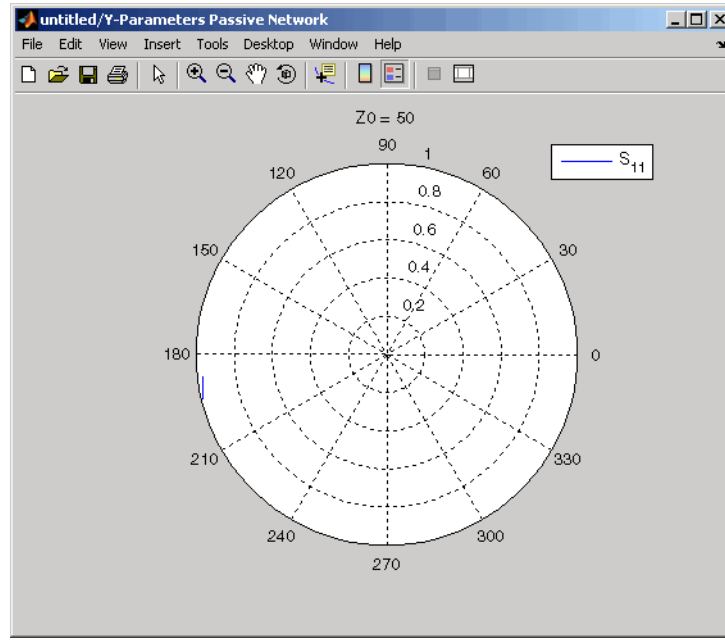


3 Set the Y-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Plot type** list, select `Polar plane`.



Click **Plot**. This action creates a polar plane plot of the S_{11} parameters in the frequency range 1.9 to 2.2 GHz.



See Also

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

`y2s` (RF Toolbox)

`interp1` (MATLAB)

Z-Parameters Amplifier

Purpose

Model nonlinear amplifier using Z-parameters

Library

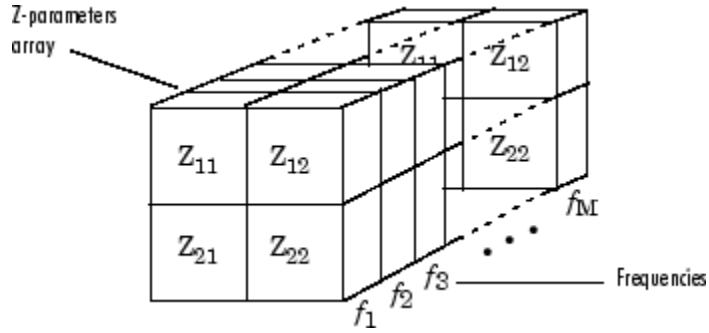
Amplifiers sublibrary of the Physical library

Description

The Z-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent Z-parameters, the frequencies of the Z-parameters, noise data, and nonlinearity data

Network Parameters

In the **Z-Parameters** field of the block dialog box, provide the Z-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Z-parameters as an M-element vector. The elements of the frequencies vector must be in the same order as the Z-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Z-parameters array and the vector of frequencies.



The Z-Parameters Amplifier block uses the RF Toolbox `z2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See “SimRF Equivalent Baseband Algorithms” for more details.

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Z-Parameters Amplifier block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a linear amplifier. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1 The block checks whether you have specified a value other than `Inf` for:

- The third-order intercept point ($OIP3$ or $IIP3$).
- The output power at the 1-dB compression point ($P_{1dB,out}$).
- The output power at saturation ($P_{sat,out}$).

In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2 The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$P_{sat,out} + GC_{sat} = P_{sat,in} + G_{lin}$$
$$P_{1dB,out} + 1 = P_{1dB,in} + G_{lin}$$
$$OIP3 = IIP3 + G_{lin}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$
$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$
$$0 = \frac{c_1}{IIP3} + c_3$$

The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Active Noise

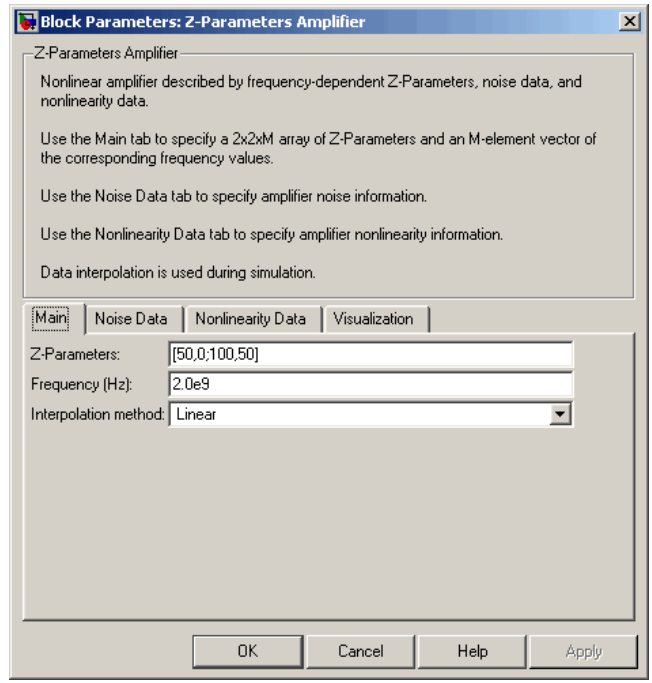
You can specify active block noise in one of the following ways:

- Spot noise data in the Z-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the Z-Parameters Amplifier block dialog box.

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

Dialog Box

Main Tab



Z-Parameters

Z-parameters for a nonlinear amplifier in a 2-by-2-by-M array. M is the number of Z-parameters.

Frequency (Hz)

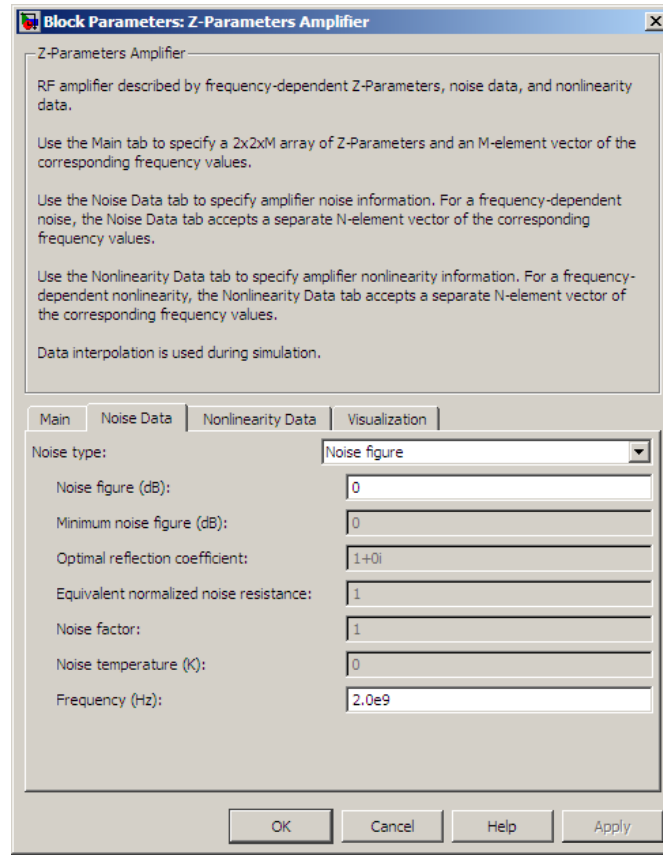
Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in **Z-Parameters**. All frequencies must be positive.

Interpolation method

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

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

Noise Data Tab



Noise type

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

Noise figure (dB)

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

Minimum noise figure (dB)

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

Optimal reflection coefficient

Optimal amplifier source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance

Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

Noise factor

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

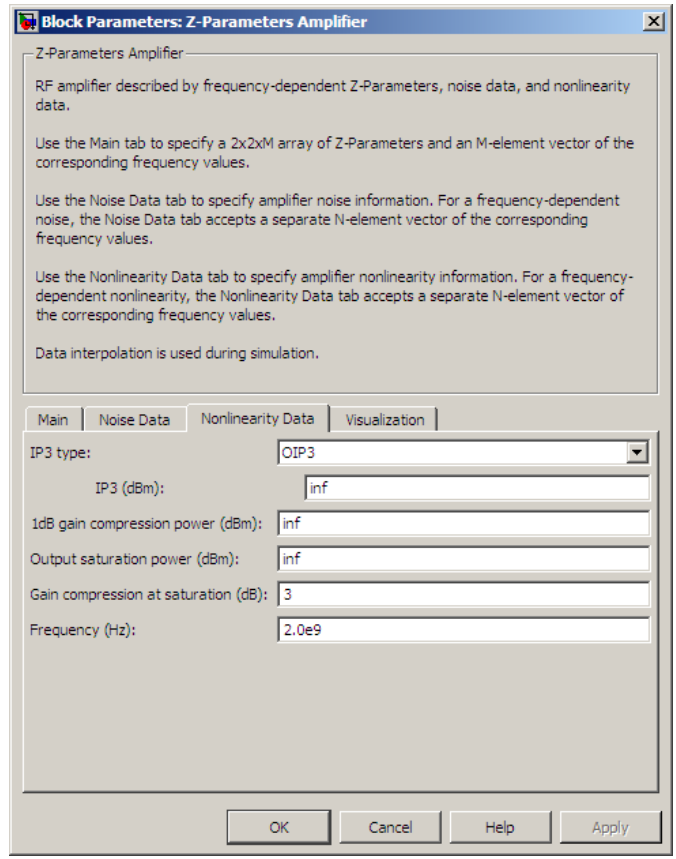
Noise temperature (K)

Equivalent temperature or vector of temperatures that produce the same amount of noise power as the amplifier. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the amplifier produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

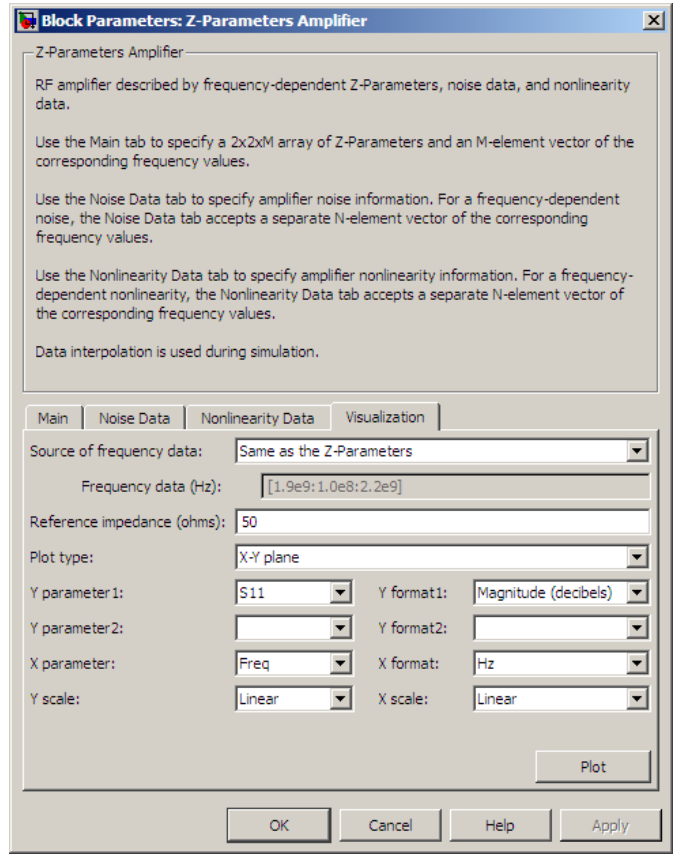
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

Plotting Parameters with the Z-Parameters Amplifier Block

The following example specifies Z-parameters [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i] and [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i] at frequencies 2.0 GHz and 2.1 GHz respectively. It uses the MATLAB `cat` function to create the 2-by-2-by-2 Z-parameters array.

```
cat(3,...
   [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i],...
   [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i])
```

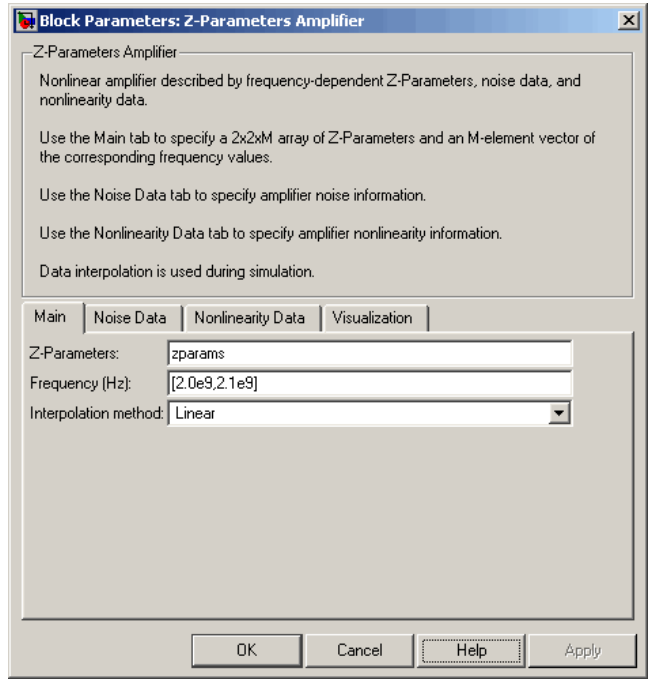
1 Type the following command at the MATLAB prompt to create a variable called `zparams` that stores the values of the Z-parameters.

```
zparams = cat(3,...
 [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i],...
 [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i])
```

2 Set the Z-Parameters Amplifier block parameters on the **Main** tab as follows:

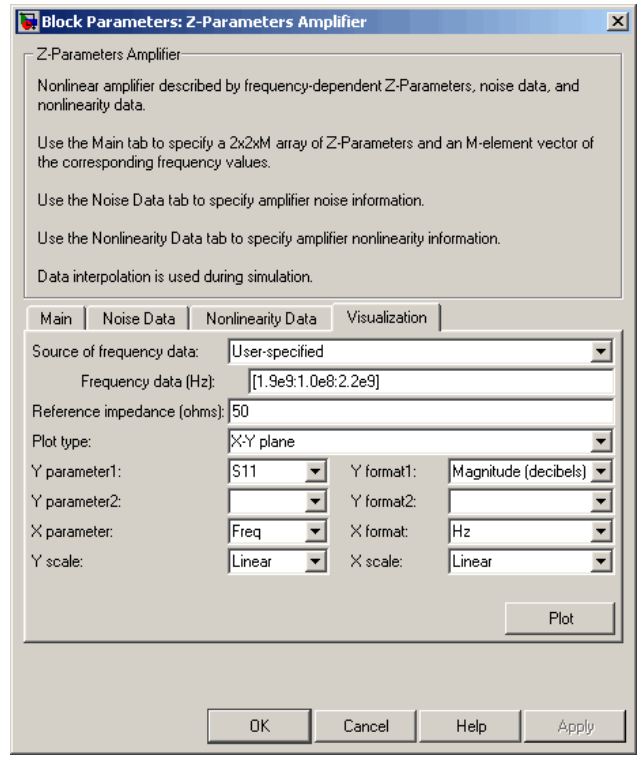
- Set the **Z-Parameters** parameter to `zparams`.
- Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

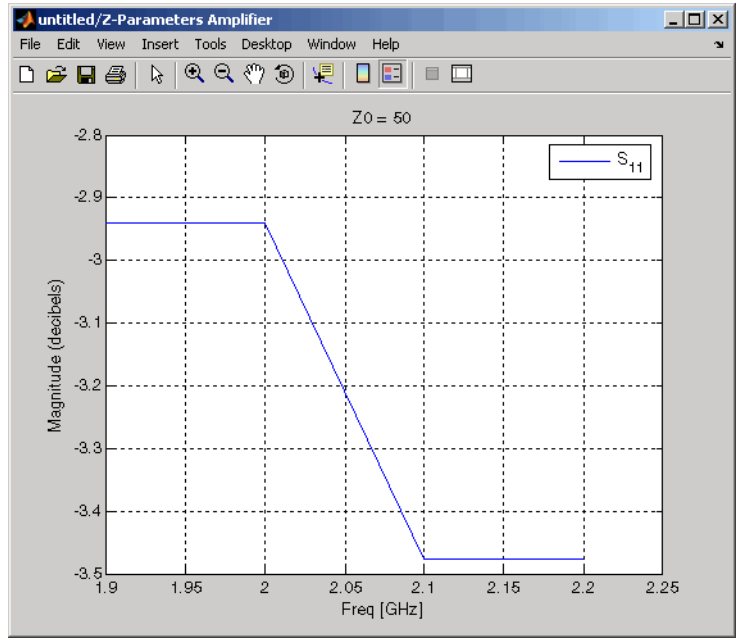


3 Set the Z-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Plot type** list, select `X-Y plane`.
- In the **Y parameter1** list, select `S11`.



Click **Plot**. This action creates an X-Y Plane plot of the S_{11} parameters in the frequency range 1.9 to 2.2 GHz.



See Also

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

`z2s` (RF Toolbox)

`interp1` (MATLAB)

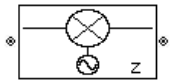
Purpose

Model mixer and local oscillator using Z-parameters

Library

Mixer sublibrary of the Physical library

Description



The Z-Parameters Mixer block models the nonlinear mixer described in the block dialog box in terms of its frequency-dependent Z-parameters, the frequencies of the Z-parameters, noise data (including phase noise data), and nonlinearity data.

Network Parameters

The Z-parameter values all refer to the mixer input frequency.

The Z-Parameters Mixer block uses the RF Toolbox `z2s` function to convert the Z-parameters to S-parameters and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See "SimRF Equivalent Baseband Algorithms" for more details.

SimRF Equivalent Baseband software computes the reflected wave at the mixer input (b_1) and at the mixer output (b_2) from the interpolated S-parameters as

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

where

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

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

Active Noise

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

- Spot noise data in the Z-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the Z-Parameters Mixer block dialog box.

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

Phase Noise

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

The blockset computes the digital filter by:

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

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

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Z-Parameters Mixer block dialog box. Depending on which of these parameters you specify,

the block computes up to four of the coefficients c_1, c_3, c_5, and c_7 of the polynomial

$$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$$

that determines the AM/AM conversion for the input signal s. The block automatically calculates c_1, the linear gain term. If you do not specify additional nonlinearity data, the block operates as a mixer with a linear gain. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method.

1 The block checks whether you have specified a value other than `Inf` for:

- The third-order intercept point ($OIP3$ or $IIP3$).
- The output power at the 1-dB compression point ($P_{1dB,out}$).
- The output power at saturation ($P_{sat,out}$).

In addition, if you have specified $P_{sat,out}$, the block uses the value for the gain compression at saturation (GC_{sat}). Otherwise, GC_{sat} is not used. You define each of these parameters in the block dialog box, on the **Nonlinearity Data** tab.

2 The block calculates a corresponding input or output value for the parameters you have specified. In units of dB and dBm,

$$\begin{aligned} P_{sat,out} + GC_{sat} &= P_{sat,in} + G_{lin} \\ P_{1dB,out} + 1 &= P_{1dB,in} + G_{lin} \\ OIP3 &= IIP3 + G_{lin} \end{aligned}$$

where G_{lin} is c_1 in units of dB.

3 The block formulates the coefficients c_3, c_5, and c_7, where applicable, as the solutions to a system of one, two, or three linear equations. The number of equations used is equal to the number of parameters you provide. For example, if you specify all three

parameters, the block formulates the coefficients according to the following equations:

$$\sqrt{P_{sat,out}} = c_1\sqrt{P_{sat,in}} + c_3\left(\sqrt{P_{sat,in}}\right)^3 + c_5\left(\sqrt{P_{sat,in}}\right)^5 + c_7\left(\sqrt{P_{sat,in}}\right)^7$$

$$\sqrt{P_{1dB,out}} = c_1\sqrt{P_{1dB,in}} + c_3\left(\sqrt{P_{1dB,in}}\right)^3 + c_5\left(\sqrt{P_{1dB,in}}\right)^5 + c_7\left(\sqrt{P_{1dB,in}}\right)^7$$

$$0 = \frac{c_1}{IIP3} + c_3$$

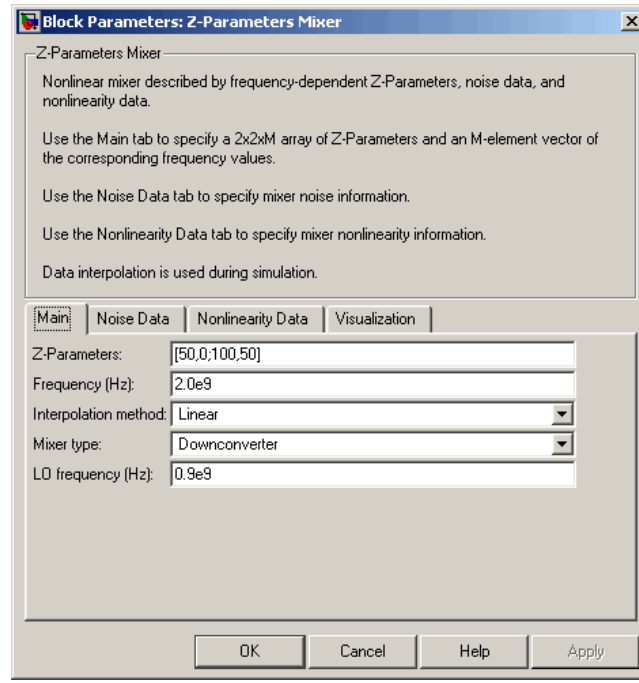
The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-Ω impedance. The third equation is the definition of the third-order intercept point.

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

If you provide vectors of nonlinearity and frequency data, the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

Dialog Box

Main Tab

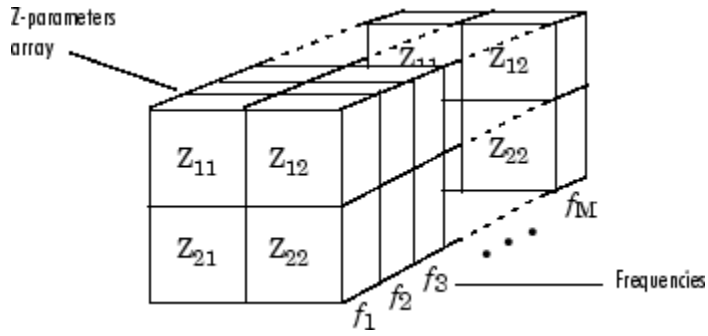


Z-Parameters

Z-parameters for a nonlinear mixer in a 2-by-2-by-M array. M is the number of Z-parameters.

Frequency (Hz)

Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in **Z-Parameters**. All frequencies must be positive. The following figure shows the correspondence between the Z-parameters array and the vector of frequencies.



Interpolation method

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

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

Mixer Type

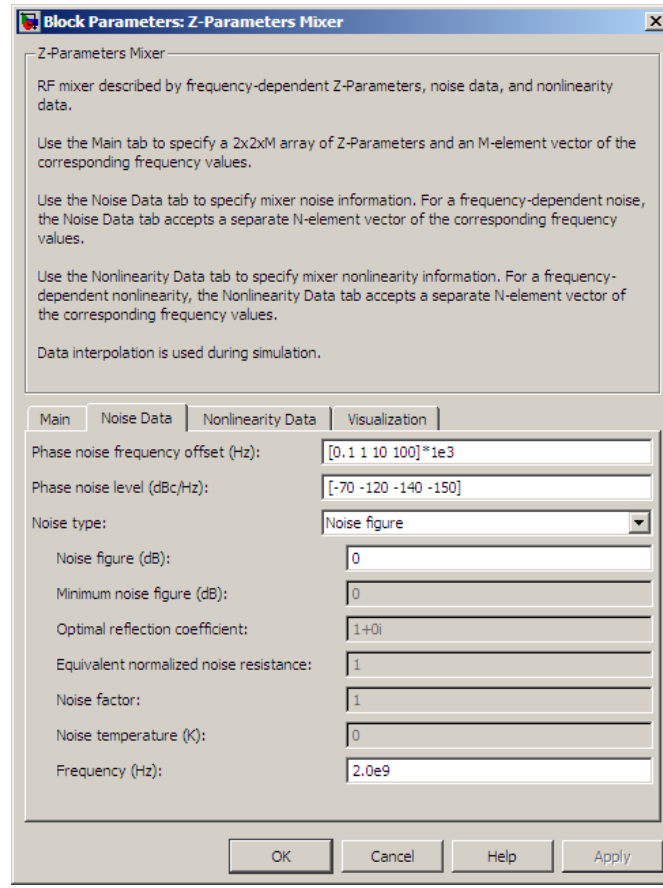
Type of mixer. Choices are `Downconverter` (default) and `Upconverter`.

LO frequency (Hz)

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

Note For a downconverting mixer, the local oscillator frequency must satisfy the condition $f_{in} - f_{lo} \geq 1/(2t_s)$, where t_s is the sample time specified in the Input Port block. Otherwise, an error appears.

Noise Data Tab



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

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

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

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

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

Optimal reflection coefficient
Optimal mixer source impedance. This parameter is enabled if **Noise type** is set to `Spot noise data`. The value can be a scalar or vector.

Equivalent normalized resistance
Resistance or vector of resistances normalized to the resistance value or values used to take the noise measurement. This parameter is enabled if **Noise type** is set to `Spot noise data`.

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

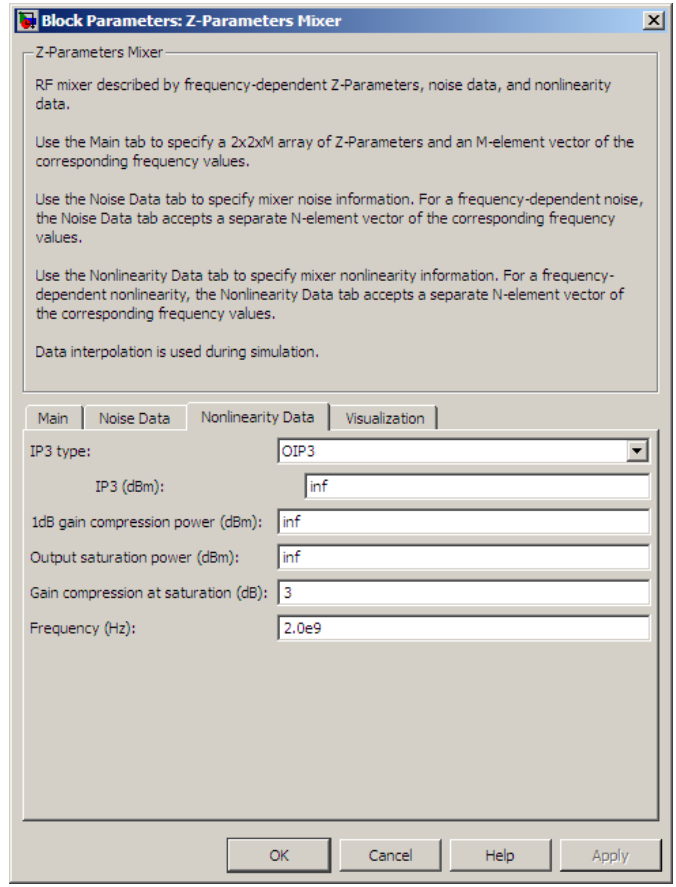
Noise temperature (K)

Equivalent temperature or vector of temperatures that produce the same amount of noise power as the mixer. This parameter is enabled if **Noise type** is set to `Noise temperature`.

Frequency (Hz)

Scalar value or vector corresponding to the domain of frequencies over which you are specifying the noise data. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the noise data for all frequencies. If you provide a vector of values for your noise data, it must be the same size as the vector of frequencies. The block uses the **Interpolation method** specified in the **Main** tab to interpolate noise data.

Nonlinearity Data Tab



IP3 type

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

IP3 (dBm)

Value of third-order intercept point. This parameter is disabled if the data source contains power data or IP3 data. Use the default value, `Inf`, if you do not know the IP3 value. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

1 dB gain compression power (dBm)

Output power value ($P_{1dB,out}$) at which gain has decreased by 1 dB. This parameter is disabled if the data source contains power data or 1-dB compression point data. Use the default value, `Inf`, if you do not know the 1-dB compression point. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Output saturation power (dBm)

Output power value ($P_{sat,out}$) that the mixer produces when fully saturated. This parameter is disabled if the data source contains output saturation power data. Use the default value, `Inf`, if you do not know the saturation power. If you specify this parameter, you must also specify the **Gain compression at saturation (dB)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Gain compression at saturation (dB)

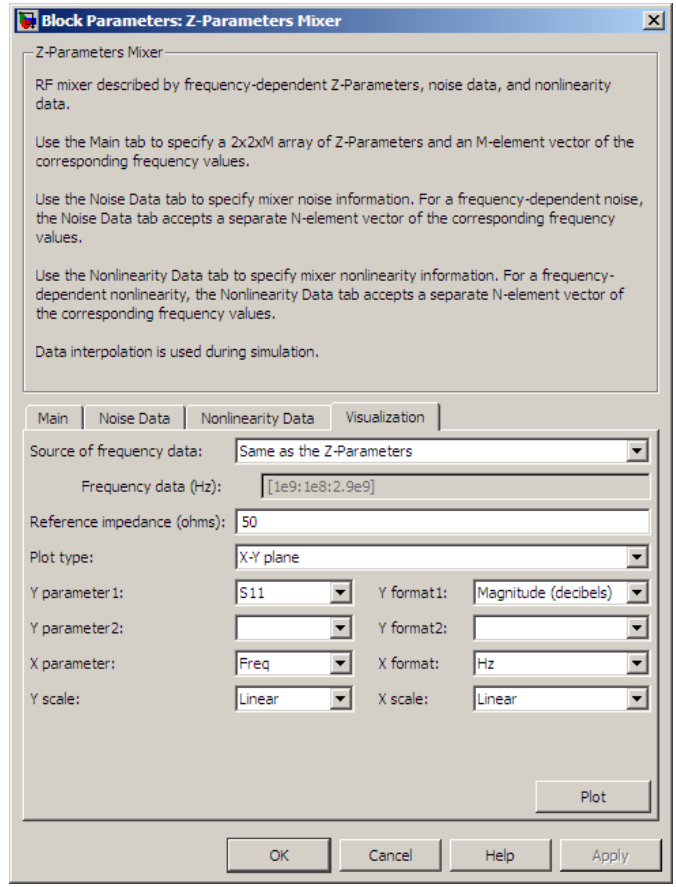
Decrease in gain (GC_{sat}) when the power is fully saturated. The block ignores this parameter if you do not specify the **Output saturation power (dBm)**. This parameter can be a scalar (to specify frequency-independent nonlinearity data) or a vector (to specify frequency-dependent nonlinearity data).

Frequency (Hz)

Scalar or vector value of frequency points corresponding to the third-order intercept and power data. This parameter is disabled if the data source contains power data or IP3 data. If you use a scalar value, the **IP3 (dBm)**, **1 dB gain compression power**

(dBm), and **Output saturation power (dBm)** parameters must all be scalars. If you use a vector value, one or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization Tab



For information about plotting, see “Create Plots”.

See Also

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

Z-Parameters Passive Network

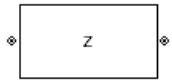
Purpose

Model passive network using Z-parameters

Library

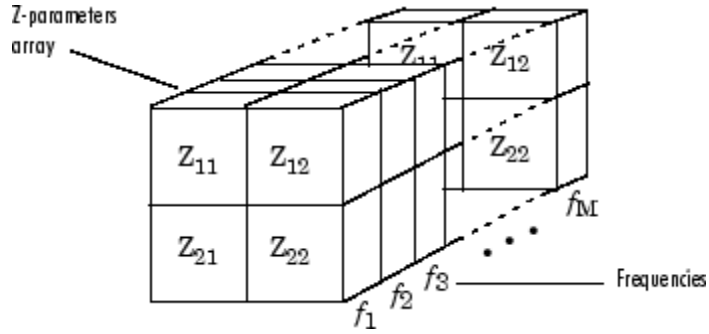
Black Box Elements sublibrary of the Physical library

Description



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

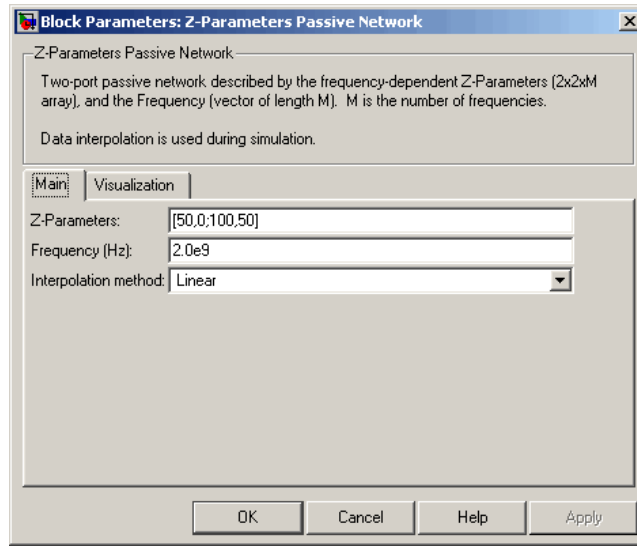
In the **Z-Parameters** field of the block dialog box, provide the Z-parameters for each of M frequencies as a 2-by-2-by-M array. In the **Frequency** field, specify the frequencies for the Z-parameters as an M-element vector. The elements of the vector must be in the same order as the Z-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Z-parameters array and the vector of frequencies.



The Z-Parameters Passive Network block uses the RF Toolbox `z2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See "SimRF Equivalent Baseband Algorithms" for more details.

Dialog Box

Main Tab



Z-Parameters

Z-parameters for a two-port passive network in a 2-by-2-by-M array. M is the number of Z-parameters.

Frequency (Hz)

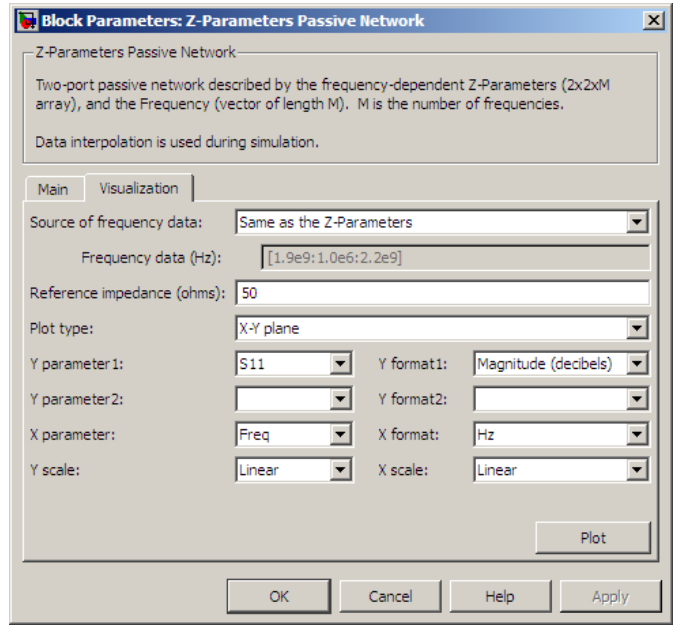
Frequencies of the Z-parameters as an M-element vector. The order of the frequencies must correspond to the order of the Z-parameters in **Z-Parameters**. All frequencies must be positive.

Interpolation method

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

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

Visualization Tab



For information about plotting, see “Create Plots”.

Examples

Plotting Parameters with the Z-Parameters Passive Network Block

The following example specifies Z-parameters [0.13 - 5.93i, .03-3.16i; 0.03-3.16i, .13-5.93i] and [0.27-2.86i, -.09-5.41i; -.09-5.41i, .27-2.86i] at

frequencies 2.0 GHz and 2.1 GHz respectively. It uses the MATLAB `cat` function to create the 2-by-2-by-2 Z-parameters array.

```
cat(3,[0.13-5.93i, .03-3.16i; 0.03-3.16i, .13-5.93i],...
      [0.27-2.86i,-.09-5.41i; -.09-5.41i, .27-2.86i])
```

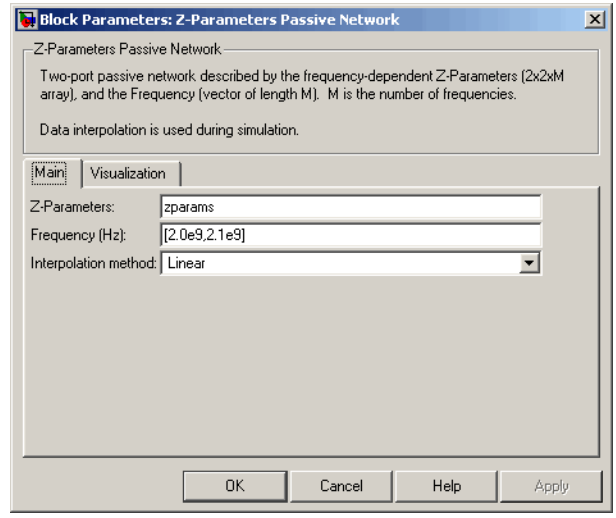
1. Type the following command at the MATLAB prompt to create a variable called `zparams` that stores the values of the Z-parameters.

   ```
   zparams = cat(3,...
         [0.13-5.93i,.03-3.16i; 0.03-3.16i,.13-5.93i],...
         [0.27-2.86i,-.09-5.41i; -.09-5.41i,.27-2.86i])
   ```

2. Set the Z-Parameters Passive Network block parameters on the **Main** tab as follows:

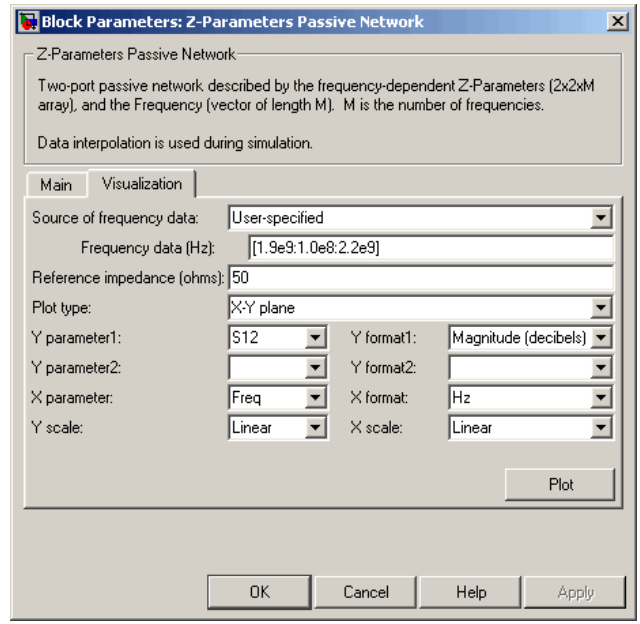
   - Set the **Z-Parameters** parameter to `zparams`.
   - Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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

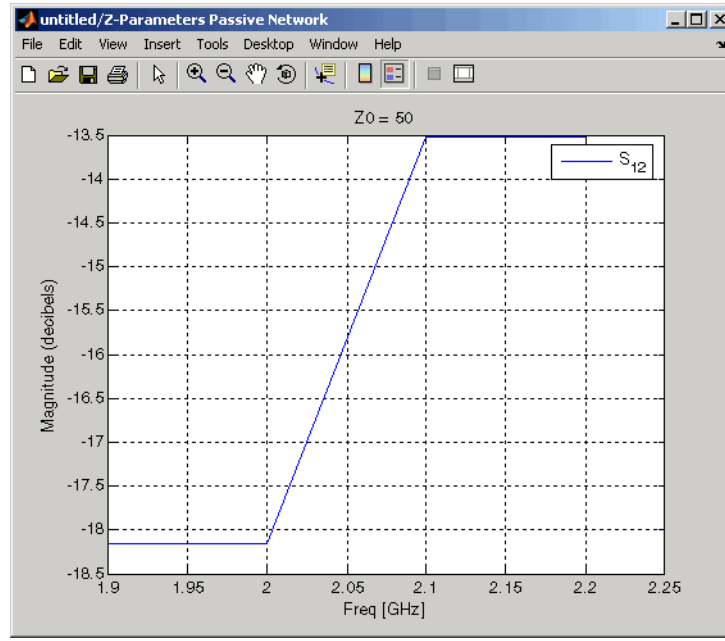


3 Set the Z-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- In the **Source of frequency data** list, select `User-specified`.
- Set the **Frequency data (Hz)** parameter to `[1.9e9:1.0e8:2.2e9]`.
- In the **Y parameter1** list, select `S12`.



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



See Also

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

`z2s` (RF Toolbox)

`interp1` (MATLAB)

Z-Parameters Passive Network

Index

A

active noise
  General Amplifier block 2-41
  General Mixer block 2-61
  S-Parameters Amplifier block 2-158
  S-Parameters Mixer block 2-171
  Y-Parameters Amplifier block 2-223
  Y-Parameters Mixer block 2-235
  Z-Parameters Amplifier block 2-257
  Z-Parameters Mixer block 2-269
amplifier
  modeling from file data 2-52
  modeling from `rfdata` object 2-39
Amplifier block 1-2 2-2
  example 2-13
  modeling nonlinearity 2-2
    methods 2-5
  sample time comparison 2-10
  thermal noise simulation 2-9
  See also General Amplifier block

B

Bandpass RF Filter block 2-16
  sample time comparison 2-17
Bandstop RF Filter block 2-20
  sample time comparison 2-21
baseband-equivalent time domain
  modeling 2-134
  linear subsystem 2-134
  nonlinear subsystem 2-138
Bessel filter design
  Bandpass RF Filter block 2-16
  Bandstop RF Filter block 2-20
  Highpass RF Filter block 2-83
  Lowpass RF Filter block 2-121
Butterworth filter design
  Bandpass RF Filter block 2-16
  Bandstop RF Filter block 2-20
  Highpass RF Filter block 2-83
  Lowpass RF Filter block 2-121

C

Capacitor block 1-16
Chebyshev I filter design
  Bandpass RF Filter block 2-16
  Bandstop RF Filter block 2-20
  Highpass RF Filter block 2-83
  Lowpass RF Filter block 2-121
Chebyshev II filter design
  Bandpass RF Filter block 2-16
  Bandstop RF Filter block 2-20
  Highpass RF Filter block 2-83
  Lowpass RF Filter block 2-121
circuit element
  modeling from `rfckt` object 2-57
Coaxial Transmission Line block 2-24
  shunt and series stubs 2-26
  stubless 2-24
Configuration block 1-18
Connection Port block 2-31
connector blocks
  Connection Port 2-31
  Input Port 2-87
  Output Port 2-134
Continuous Wave block 1-23
Coplanar Waveguide Transmission Line block 2-33
  shunt and series stubs 2-34
  stubless 2-33

D

data consistency
  General Amplifier block 2-42

E

Elliptic filter design
  Bandpass RF Filter block 2-16

Bandstop RF Filter block 2-20
Highpass RF Filter block 2-83
Lowpass RF Filter block 2-121

F

filter design
physical blocks
LC Lowpass Tee block 2-118
filter design mathematical blocks
Bandpass RF Filter block 2-16
Bandstop RF Filter block 2-20
Highpass RF Filter block 2-83
Lowpass RF Filter block 2-121
filter design physical blocks
LC Bandpass Pi block 2-93
LC Bandpass Tee block 2-100
LC Bandstop Pi block 2-103
LC Bandstop Tee block 2-106
LC Highpass Pi block 2-109
LC Highpass Tee block 2-112
LC Lowpass Pi block 2-115

G

General Amplifier block 2-39
active noise 2-41
generating from file data 2-52
nonlinearity 2-39
operating conditions 2-42
See also Amplifier block
General Circuit Element block 2-56
generating from `rfckt` object 2-57
General Mixer block 2-61
active noise 2-61
nonlinearity 2-63
operating conditions 2-65
phase noise 2-62
See also Mixer block
General Passive Network block 2-77
generating from file data 2-79
Ghorbani model 2-8
Ground block 1-27

H

Highpass RF Filter block 2-83
sample time comparison 2-84

I

Impedance block 1-28 1-33
Inductor block 1-35
Inport block 1-29
Input Port block 2-87
parameter input for physical subsystem 2-87
sample time comparison 2-87
See also Output Port block

L

ladder filters
example to filter Gaussian noise 2-95
LC Bandpass Pi block 2-93
example to filter Gaussian noise 2-95
LC Bandpass Tee block 2-100
LC Bandstop Pi block 2-103
LC Bandstop Tee block 2-106
LC Highpass Pi block 2-109
LC Highpass Tee block 2-112
LC Ladder block 1-37
LC Lowpass Pi block 2-115
LC Lowpass Tee block 2-118
Lowpass RF Filter block 2-121
sample time comparison 2-122

M

Microstrip Transmission Line block 2-125
shunt and series stubs 2-126
stubless 2-125

Mixer block 1-43 2-131
See also General Mixer block
modeling
amplifier from file data 2-52
amplifier from `rfdata` object 2-39
baseband-equivalent time domain 2-134
circuit element from `rfckt` object 2-57
passive network from file data 2-79
Mutual Inductor block 1-51

N

noise
active
General Amplifier block 2-41
General Mixer block 2-61
S-Parameters Amplifier block 2-158
S-Parameters Mixer block 2-171
Y-Parameters Amplifier block 2-223
Y-Parameters Mixer block 2-235
Z-Parameters Amplifier block 2-257
Z-Parameters Mixer block 2-269
phase
General Mixer block 2-62
S-Parameters Mixer block 2-171
Y-Parameters Mixer block 2-236
Z-Parameters Mixer block 2-270
Noise block 1-54
nonlinearity
Amplifier block 2-2
General Amplifier block 2-39
General Mixer block 2-63
S-Parameters Amplifier block 2-156
S-Parameters Mixer block 2-171
Y-Parameters Amplifier block 2-221
Y-Parameters Mixer block 2-236
Z-Parameters Amplifier block 2-255
Z-Parameters Mixer block 2-270

O

operating conditions
General Amplifier block 2-42
General Mixer block 2-65
Outport block 1-57
Output Port block 2-134
baseband-equivalent time domain modeling 2-134
plotting RF subsystem parameters 2-142
See also Input Port block

P

Parallel-Plate Transmission Line block 2-143
shunt and series stubs 2-145
stubless 2-143
parameter input
to physical subsystem 2-87
passive network
modeling from file data 2-79
phase noise
General Mixer block 2-62
S-Parameters Mixer block 2-171
Y-Parameters Mixer block 2-236
Z-Parameters Mixer block 2-270
Phase Shift block 1-61
physical subsystem
parameter input for 2-87
plots
RF subsystem parameters 2-142

R

Rapp model 2-9
Resistor block 1-63
`rfckt` object
modeling general circuit element 2-56
`rfdata` object
modeling nonlinear amplifier 2-39
RLCG Transmission Line block 2-150

shunt and series stubs 2-151
stubless 2-150

S

S-Parameters Amplifier block 2-156
active noise 2-158
nonlinearity 2-156
S-Parameters block 1-65
S-Parameters Mixer block 2-170
active noise 2-171
nonlinearity 2-171
phase noise 2-171
S-Parameters Passive Network block 2-183
Saleh model 2-7
Series C block 2-190
Series L block 2-192
Series R block 2-194
Series RLC block 2-196
Shunt C block 2-199
Shunt L block 2-201
Shunt R block 2-203
Shunt RLC block 2-205
Signal Combiner block 1-75
Sinusoid block 1-76
stubless transmission lines
Coaxial Transmission Line block 2-24
Coplanar Waveguide Transmission Line block 2-33
Microstrip Transmission Line block 2-125
Parallel-Plate Transmission Line block 2-143
RLCG Transmission Line block 2-150
Transmission Line block 2-208
Two-Wire Transmission Line block 2-214
stubs (shunt and series)
Coaxial Transmission Line block 2-26
Coplanar Waveguide Transmission Line block 2-34
Microstrip Transmission Line block 2-126
Parallel-Plate Transmission Line block 2-145
RLCG Transmission Line block 2-151
Transmission Line block 2-209
Two-Wire Transmission Line block 2-216

T

Three-Winding Transformer block 1-80
time-domain modeling
baseband-equivalent 2-134
Transmission Line block 1-83 2-208
shunt and series stubs 2-209
stubless 2-208
Two-Wire Transmission Line block 2-214
shunt and series stubs 2-216
stubless 2-214

Y

Y-Parameters Amplifier block 2-221
active noise 2-223
nonlinearity 2-221
Y-Parameters Mixer block 2-235
active noise 2-235
nonlinearity 2-236
phase noise 2-236
Y-Parameters Passive Network block 2-248

Z

Z-Parameters Amplifier block 2-254
active noise 2-257
nonlinearity 2-255
Z-Parameters Mixer block 2-269
active noise 2-269
nonlinearity 2-270
phase noise 2-270
Z-Parameters Passive Network block 2-282