SimRF™

Reference

MATLAB®&SIMULINK®

MathWorks®

How to Contact MathWorks

	Latest news:	www.mathworks.com
	Sales and services:	www.mathworks.com/sales_and_services
	User community:	www.mathworks.com/matlabcentral
	Technical support:	www.mathworks.com/support/contact_us
	Phone:	508-647-7000

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

Contents

Blocks — Alphabetical List

Amplifier

Model amplifier in RF systems
Library:　　　SimRF / Circuit Envelope / Elements
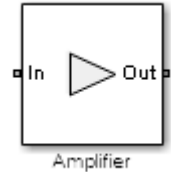
Amplifier

Description

Use the Amplifier block to model a linear or nonlinear amplifier, with or without noise. Defining the amplifier gain using a data source also defines input data visualization and modeling. Use the **Main** tab parameters to 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 also produces additional intermodulation frequencies.

Parameters

Main

Source of amplifier gain — Source parameter of the amplifier gain
Available power gain (default) | Open circuit voltage gain | Data source | Polynomial coefficients

Source parameter of the amplifier gain, specified as one of the following:

- `Available power gain` — **Available power gain** parameter is used 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` — **Open circuit voltage gain** parameter is used as the linear voltage gain term of the polynomial VCVS, a_1.

- `Data source` — Linear voltage gain term of the polynomial VCVS is calculated from the specified data source options:

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

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. The data sources are specified using either `Data file` or `Network-parameters` or `Rational model`, depending on the value of `Data source`.

- `Polynomial coefficients` — The block implements a nonlinear voltage gain according to the specified polynomial coefficients

`Available power gain` — Available power gain
0 `dB` (default) | scalar

Available power gain of amplifier, specified as a scalar in dB. Specify the units from the corresponding drop-down list.

Dependencies

To enable this parameter, choose `Available power gain` in the **Source of amplifier gain** tab.

`Open circuit voltage gain` — Open circuit voltage gain
0 `dB` (default) | scalar

Open circuit voltage of amplifier, specified as a scalar in dB. Specify the units from the corresponding drop-down list.

Dependencies

To enable this parameter, choose Open circuit voltage gain in the **Source of amplifier gain** tab.

Data source — Data source
Data File (default) | Network-parameters | Rational Model

Data source, specified as one of the following:

- Data file — Name of a Touchstone file with the extension .s2p. The block ignores noise and nonlinearity data in imported files.

- Network-parameters — Provide **Network parameter** data such as S-parameters, Y-parameters, and Z-parameters with corresponding **Frequency** and **Reference impedance (ohms)** for the amplifier.

- Rational model — Provide values for **Residues**, **Poles**, and **Direct feedthrough** parameters which correspond 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 rational model equation, each C_k is the residue of the pole A_k. If C_k is complex, a corresponding complex conjugate pole and residue must also be enumerated.This object has the properties C, A, and D. You can use these properties to specify the **Residues**, **Poles**, and **Direct feedthrough** parameters.

Dependencies

To enable this parameter, select Data source in **Source of amplifier gain** tab.

Polynomial coefficients — Order of polynomial
[0 1] (default) | vector

Order of polynomial, specified as a vector.

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, $[a_0, a_1, a_2, \ldots a_9]$, the polynomial it represents is:

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

where a_1 represents the linear gain term, and higher-order terms are modeled according to [1].

For example, the vector $[a_0, a_1, a_2, a_3]$ specifies the relation $V_o = a_0 + a_1 V_1 + a_2 V_1^2 + a_3 V_1^3$. 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 $V_o = V_i$.

Dependencies

To enable this parameter, select `Polynomial coefficients` in **Source of amplifier gain** tab.

`Network parameter type` — Network parameter type
`S-parameters` (default) | `Y-parameters` | `Z-parameters`

Network parameter type, specified as `S-parameters`, `Y-parameters`, or `Z-parameters`.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Network-parameters` in the **Data source** tab.

`Input impedance (Ohm)` — Input impedance
`50` (default) | scalar

Input impedance of amplifier, specified as a scalar.

Dependencies

To enable this parameter, select `Available power gain`, `Open circuit voltage gain`, or `Polynomial coefficients` in **Source of amplifier gain** tab.

`Output impedance (Ohm)` — Output impedance
`50` (default) | scalar

Output impedance of amplifier, specified as a scalar.

Dependencies

To enable this parameter, select `Available power gain`, `Open circuit voltage gain`, or `Polynomial coefficients` in **Source of amplifier gain** tab.

`Data file` — Name of network parameter data file
simrfV2_unitygain.s2p (default) | character vector

Name of network parameter data file, specified as a character vector.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Data file` in **Data source**.

`Noise figure (dB)` — Noise figure
0 (default) | scalar

Noise figure of amplifier, specified as a scalar in dB. 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).

If the noise figure is a part of the file specified in the `Data source`, then the following message is displayed `Noise specified in Data file`.

`Frequency (dB)` — Frequency of network parameters
1e9 Hz (default) | scalar | Hz | kHz | MHz | GHz

Frequency of network parameters, specified as a scalar in Hz.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Network-parameters` in **Data source**.

`Reference Impedance(Ohm)` — Reference impedance of network parameters
50 (default) | scalar

Reference impedance of network parameters, specified as a scalar.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Network-parameters` in **Data source**.

`Residues` — Residues in order of rational model
0 (default) | vector

Residues in order of rational model, specified as a vector.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Rational model` in **Data source**.

`Poles` — Residues in order of rational model
0 (default) | vector

Poles in order of rational model, specified as a vector.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Rational model` in **Data source**.

`Direct feedthrough` — Direct feedthrough
{0 0:1 0} (default) | array of vectors

Direct feedthrough, specified as an array vector.

Dependencies

To enable this parameter, first select `Data source` in **Source of amplifier gain** tab. Then, select `Rational model` in **Data source**.

`Polynomial coefficients` — Polynomial coefficients
[0 1] (default) | vector

Polynomial coefficients, specified as a vector.

Dependencies

To enable this parameter, select `Polynomial coefficients` in **Source of amplifier gain** tab.

`Ground and hide negative terminals` — Ground RF circuit terminals
on (default) | `off`

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

By default, this option is selected.

Nonlinearity

`Nonlinear polynomial type` — Type of nonlinearity
`Even and odd order` (default) | `Odd order`

Type of nonlinearity, specified as `Even and odd order` or `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 figure shows the graphical definition of the nonlinear amplifier parameters.

Intercept points convention — Intercept points convention
Output (default) | Input

Intercept points convention, specified a Input-referred, or Output-referred convention. Use this specification for the intercept points, 1-dB gain compression power, and saturation power.

IP2 — Second-order intercept point
infdBm (default) | scalar | W | mW | dBW | dBm

Second-order intercept point, specified as a scalar.

Dependencies

To set this parameter, select Even and odd order in **Nonlinear polynomial type**.

`IP3` — **Third-order intercept point**
`infdBm` (default) | scalar | W | mW | dBW | dBm

Third-order intercept point, specified as a scalar.

`1-dB gain compression power` — **1-dB gain compression power**
`infdBm` (default) | scalar | W | mW | dBW | dBm

1-dB gain compression power, specified as a scalar.

Dependencies

To set this parameter, select `Odd order` in **Nonlinear polynomial type**.

`Output saturation power` — **Output saturation power**
`infdBm` (default) | scalar | W | mW | dBW | dBm

Output saturation power, specified as scalar. 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.

Dependencies

To set this parameter, select `Odd order` in **Nonlinear polynomial type**.

`Gain compression at saturation` — **Gain compression at saturation**
`infdBm` (default) | scalar | W | mW | dBW | dBm

Gain compression at saturation, specified as scalar.

When **Nonlinear polynomial type** is `Odd order`, specify the gain compression at saturation.

Dependencies

To set this parameter, first select `Odd order` in **Nonlinear polynomial type**. Then, change the default value of **Output saturation power**

Modeling

`Modeling options` — Model S-parameters
`Time-domain (rationalfit)` (default) | `Frequency-domain`

Model S-parameters, specified as:

- Time-domain (rationalfit) technique creates an analytical rational model that approximates the whole range of the data. When modeling using `Time domain`, the **Plot** in `Visualization` tab plots the data defined in `Data Source` and the values in the `rationalfit` function.
- Frequency-domain computes the baseband impulse response for each carrier frequency independently. This technique is based on convolution. There is an option to specify the duration of the impulse response. For more information, see "Compare Time and Frequency Domain Simulation Options for S-parameters".
- 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`.

Dependencies

To set this parameter, first select `Data source` in **Source of amplifier gain**. This selection activates the **Modeling** Tab which contains **Modeling options**

`Fitting options` — Rationalfit fitting options
`Fit individually` (default) | `Share poles by column` | `Share all poles`

Rationalfit fitting options, specified as `Fit individually`, `Share poles by column`, or `Share all poles`.

Rational fitting results shows values of **Number of independent fits**, **Number of required poles**, and **Relative error achieved (dB)**.

Dependencies

To set this parameter, select `Time domain (rationalfit)` in **Modeling options**.

`Relative error desired (dB)` — Relative error acceptable for the rational fit
`-40` (default) | scalar

Relative error acceptable for the rational fit, specified as a scalar.

Dependencies

To set this parameter, select `Time domain (rationalfit)` in **Modeling options**.

Automatically estimate impulse response duration — Automatically calculate impulse response
`on | off`

Select this parameter to automatically calculate impulse response. Clear this parameter to manually specify the impulse response duration using **Impulse response duration**.

Dependencies

To set this parameter, select `Frequency domain` in **Modeling options**.

Impulse response duration — Impulse response duration
`1e-10` (default) | scalar

Impulse response duration, specified as a scalar.

Dependencies

To set this parameter, first select `Frequency domain` in **Modeling options**. Then, clear `Automatically estimate impulse response duration`.

Visualization

Source of frequency data — Frequency data source
`Extracted from data source` (default) | `User-defined`

Frequency data source, specified as:

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.

Dependencies

To set this parameter, first select Data source in **Source of amplifier gain**. This selection activates the **Visualization** Tab which contains **Source of frequency data**

Frequency data — Frequency data range
[1e9:1e6:3e9] (default) | vector | Hz | kHz | MHz | GHz

Frequency data range, specified as a vector

Plot type — Type of data plot
X-Y plane (default) | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot that you want to produce with your data specified as one of the following:

- 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.

Parameter 1 — Type of S-Parameters to plot
S11 (default) | S12 | S21 | S22

Type of S-Parameters to plot, specified as S11, S12, S21, or S22.

Parameter 2 — Type of S-Parameters to plot
None (default) | S11 | S12 | S21 | S22

Type of S-Parameters to plot, specified as S11, S12, S21, or S22.

Format1 — Plot format
Magnitude (decibels) (default) | Angle(degrees) | Real | Imaginary

Plot format, specified as `Magnitude (decibels)`, `Angle(degrees)`, `Real`, or `Imaginary`.

Format2 — Plot format
`Magnitude (decibels)` (default) | `Angle(degrees)` | `Real` | `Imaginary`

Plot format, specified as `Magnitude (decibels)`, `Angle(degrees)`, `Real`, or `Imaginary`.

Y-axis scale — Y-axis scale
`Linear` (default) | `Logarithmic`

Y-axis scale, specified as `Linear` or `Logarithmic`.

X-axis scale — X-axis scale
`Linear` (default) | `Logarithmic`

X-axis scale, specified as `Linear` or `Logarithmic`.

Plot — Plot specified data
button

Plot specified data using plot button.

Model Examples

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`

Introduced in R2010b

Attenuator

Model attenuator for RF circuit
Library: SimRF / Circuit Envelope / Elements

Attenuator

Description

The Attenuator block to attenuates the signal power by a given factor known as *Insertion Loss* in dB. Commonly, the block matches the impedance of the RF circuit at the input and output ports. You can use attenuators to dampen the power of the incoming signal to protect RF circuits.

Parameters

`Attenuation (dB)` — Level of insertion loss or attenuation
3 (default) | scalar

Level of insertion loss or attenuation to apply to the signal, specified as a scalar in dB.

`Input impedance (Ohm)` — Input impedance
50 (default) | scalar

Input impedance of the attenuator, specified as a scalar in ohms.

`Output impedance (Ohm)` — Output impedance
50 (default) | scalar

Output impedance of the attenuator, specified as a scalar in ohms.

`Simulate noise` — Simulate thermal noise
on (default) | off

Select this parameter to simulate thermal noise in the attenuator. You must select **Simulate noise** in the `Configuration` block.

This parameter inserts a current noise source with the single-sided power density of 4 $\mathbf{k}T/R$ A^2/Hz, where:

- T is the value of the **Temperature** parameter in the `Configuration` block. Units are in degrees Kelvin.

- $\mathbf{k}$ is the Boltzmann constant.

`Ground and hide negative terminals` — Ground RF circuit terminals
on (default) | off

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear this parameter. By exposing these terminals, you can connect them to other parts of your model.

Model Examples

See Also
`Variable Attenuator`

Introduced in R2016b

C (Capacitor)

Model capacitor for circuit envelope analysis

Library

Elements
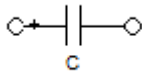
Description

The Capacitor block models a capacitor in circuit envelope environment.

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.

  Note: Harmonic balance method calculates the steady state response of SimRF systems. For further information please refer to Kenneth, S.Kundrt. " Introduction to RF Simulation." *The Designers Guide Community*, April 2003, http://www.designers-guide.org/analysis/rf-sim.pdf.

- 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

Circulator

Model ideal frequency-independent circulators with S-parameters
Library: SimRF / Circuit Envelope / Junctions
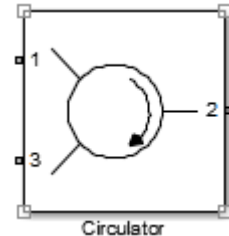
Circulator

Description

Use the Circulator block to model ideal passive three-port circulators in a circuit
envelope environment. Circulators are used to control signal direction and flow inside an
RF circuit. The Circulator block can circulate the signal in clockwise or counterclockwise
direction depending on the parameter selection.

Parameters

Select component — Circulator direction
Circulator clockwise (default) | Ciculator counter clockwise

Circulator direction, specified as:

• Circulator clockwise

 The default option is Circulator clockwise. The s-parameter matrix for
 Circulator clockwise is:

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

 Circulator counter clockwise

The s-parameter matrix for `Circulator counter clockwise` is:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

`Reference impedances (Ohm)` — Reference impedance of circulator
50 (default) | positive scalar | three-tuple vector

Reference impedance of circulator, specified as a scalar or three-tuple vector.

`Ground and hide negative terminals` — Ground RF circuit terminals
on (default) | `off`

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear this parameter. By exposing these terminals, you can connect them to other parts of your model.

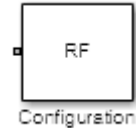By default, this option is selected.

See Also
`Coupler` | `Divider`

Introduced in R2014a

Configuration

Define system simulation settings
Library: SimRF / Circuit Envelope / 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 "Simulate High Frequency Components".

Parameters

`Automatically select fundamental tones and harmonic order` —
Automatically select fundamental tones and harmonic order
on (default) | off

Select this parameter to choose **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.

To manually set the **Fundamental tones** and **Harmonic order**, clear this paramter. A smaller set of simulation frequencies decreases simulation time and decreases memory requirements. However, a decrease in simulation frequencies can reduce accuracy.

`Fundamental tones` — Fundamental tones of set of simulation frequency
vector of positive integers in Hz

Fundamental tones of a set of simulation frequencies, specified as a vector of positive integers in Hz.

Dependencies

To enable this parameter, clear **Automatically select fundamental tones and harmonic order**.

`Harmonic order` — Harmonic order for each fundamental tone
vector of positive integers

Harmonic order for each fundamental tone, specified as a vector of positive integer. You can also specify a scalar and this value is applied to each **Fundamental tones**.

Dependencies

To enable this parameter, clear **Automatically select fundamental tones and harmonic order**.

`Total simulation frequencies: Computed at simulation time` — Displays number for simulation frequencies for linear model
button

Click **View** to open dialog box containing additional information about the simulation frequencies in your system. The `Configuration` block displays the number of simulation frequencies for a nonlinear model. For linear models, the actual number of frequencies are automatically optimized during simulation. Because the solver computes a solution to the network at each simulation frequency, computation time scales according to the size of this value.

By clicking a listed simulation frequency, you can see which linear or multiple combinations of fundamental tones represent that frequency. From the dialog box, you can also plot the simulation frequencies on a number line.

The block parameters define a set of simulation frequencies as combinations of fundamental tones: $[m*f_1 + n*f_2 + \ldots]$. In this case, represented as $[f_1, f_2, \ldots]$, and the integers m and n are bounded by the corresponding **Harmonic order**, $|m| \leqq h1$, $|n| \leqq h_2$, etc. Only positive frequencies are considered.

For example, suppose that you have a single fundamental tone $f_1 = 2$ GHz and corresponding harmonic order $h1 = 3$. The set of simulation frequencies $[0, f_1, 2f_1, 3f_1] = [0\text{GHz}, 2\text{ GHz}, 4\text{ GHz}, 6\text{GHz}]$.

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 $h_1 = h_2 = 1$. This setup results in five simulation frequencies with values $[0, f_2, f_1\text{-}f_2, f_1, f_1\text{+}f_2]$.

The set of simulation frequencies must include all carrier frequencies specified in the SimRF subsystem such as the carrier frequencies inside Inport, Outport, and source blocks.

Normalize Carrier Power — Normalize power of carrier signal
on (default) | off

Select this option to normalize the carrier power such that the average power of the signal is:

$$I^2 + Q^2$$

In this case, the equation gives the corresponding passband signal at ω:

$$s_k(t) = I(t)\sqrt{2}\cos(2\pi f_k t) - Q(t)\sqrt{2}\sin(2\pi f_k t)$$

where:

- $I(t)$ am the in-phase part of the carrier signal.
- $Q(t)$ is the quadrature part of the carrier signal.
- f_k are the carrier frequencies.

Clear this option so the average power of the carrier signal is:

$$\frac{I^2 + Q^2}{2}$$

In this case, the corresponding passband signal at ω represented by the equation

$$s_k(t) = I(t)\cos(2\pi f_k t) - Q(t)\sin(2\pi f_k t)$$

0 carrier frequency is a special case. Its passband representation is always I and average power I^2

`Solver` — Fixed-step solver of SimRF environment
Auto (default) | NDF2 | Trapezoidal Rule | Backward Euler

Fixed-step solver of SimRF environment, specified as one of the following:

- `Auto`: Set this parameter to `Auto`, when you are not sure which solver to use.
- `NDF2`: Set this parameter to `NDF2` to balance 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.
- `Trapezoidal Rule`: Set this parameter to `Trapezoidal Rule` for narrowband simulations. Frequency warping and the lack of damping effects make this method inappropriate for most wideband simulations.
- `Backward Euler`: Set this parameter to `Backward Euler` to simulate the largest class of systems and signals. Damping effects make this solver suitable for wideband simulation, but overall accuracy is low.

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` — Time step for fixed step solver configuration
1e-6 (default) | vector of integers in seconds

Time step for fixed step solver configuration, specified as a vector of integers in seconds. The default is sufficient for modeling envelope signals with bandwidths of up to $1/h$, or 1 MHz. But 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` — Globally enable or disable noise modeling
on (default) | off

Select this parameter to globally enable noise modeling in SimRF circuits. 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.

To disable noise modeling globally, clear this parameter.

`Use default ramdom number generator` — Default pseudorandom noise stream for SimRF sources
on (default) | off

Select this parameter to retain the default pseudorandom noise stream for SimRF sources. Clear this option to specify an independent pseudorandom number stream for the SimRF topological subsystem and determine the stream's seed

Dependencies

To expose this parameter, select **Simulate noise**.

`Noise seed` — Seed of the independent pseudorandom number stream
0 (default) | scalar positive integer

Seed of the independent pseudorandom number stream, specified as a scalar positive integer.

Dependencies

To expose this parameter, clear **Use default random number generator**.

`Temperature` — Global noise temperature
290.0K | scalar integer in kelvin

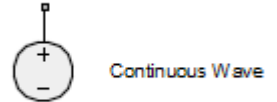Global noise temperature, specified as a scalar integer in kelvin.

Model Examples

See Also
Inport | Outport

Introduced in R2013a

Continuous Wave

Model constant envelope source
Library: SimRF / Circuit Envelope / Sources

Continuous Wave

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, "Simulate High Frequency Components".

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.

Parameters

Source type — Wave type
Ideal voltage (default) | Ideal current | Power

- 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

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

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.

Constant in-phase value — In-phase modulations of carrier frequencies
0 V (default) | vector of real numbers | V | mV | kV

In-phase modulations for each of the **Carrier frequencies**, specified as a vector of real numbers. Specify the units from the corresponding drop-down list.

Dependencies

To enable this parameter, on the **Source type** tab, choose Ideal voltage or Ideal current.

Constant quadrature value — Quadrature modulations of carrier frequencies
0 V (default) | vector of real numbers | V | mV | kV

Quadrature modulations for each of the RF circuit carrier frequencies, specified as a vector of real numbers. Specify the units from the corresponding drop-down list.

Dependencies

To enable this parameter, on the **Source type** tab, choose Ideal voltage or Ideal current.

Source impedance(Ohm) — Input impedance of source
50 (default) | positive real number | complex number

Input impedance of source, specified as a positive real number or a complex number. The complex number must contain real and imaginary parts greater than or equal to `1e-18` ohms.

Dependencies

To enable this parameter, on the **Source type** tab, choose `Power`.

`Available power` — Input impedance of source
`0 W` (default) | vector of real numbers | `W` | `mW` | `dBW` | `dBm`

Available power at the specified **Carrier frequencies**, specified as a vector of real numbers. Specify the units from the drop-down list. The default value is `0 W`.

Dependencies

To enable this parameter, on the **Source type** tab, choose `Power`. The default value is `0 W`

`Angle(degrees)` — Phase angle of power waves
`0` (default) | vector of real numbers

Phase angle of power waves at the specified **Carrier frequencies**, specified as a vector of real numbers. The default value is `0` degrees.

Dependencies

To enable this parameter, on the **Source type** tab, choose `Power`.

`Carrier frequencies` — Carrier frequencies
`0` (default) | vector of real positive numbers | `Hz` | `kHz` | `MHz` | `GHz`

Carrier frequencies, specified as a vector of real positive numbers. The elements in the carrier frequencies are combinations of fundamental tones and corresponding harmonics in the `Configuration` block. The default value is `0 Hz`.

`Ground and hide negative terminals` — Ground RF circuit terminals
`on` (default) | `off`

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

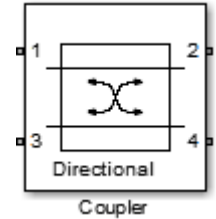By default, this option is selected.

See Also

Inport | Noise | Sinusoid

Introduced in R2010b

Coupler

Model ideal frequency-independent couplers with S-parameters
Library: SimRF / Circuit Envelope / Junctions

Description

The Coupler block models four port directional couplers in a circuit envelope environment as an ideal S-parameter model. The four ports of the coupler are `Input port` (Port 1), `Through port` (Port 2), `Isolated port` (Port 3), `Coupled port` (Port 4).

Directional couplers are used to sample forward and reflected waves propagating along a transmission line. Directional couplers find uses in many RF design applications such as line power sensors and transmitter automatic level controls.

Hybrid couplers are used to split or combine signals with specific phase relations.

Parameters

Select component — Coupler type
Directional coupler (default) | Coupler symmetrical | Coupler antisymmetrical | Hybrid quadrature (90 deg) | Hybrid rat-race

Coupler type, specified as:

• Directional coupler

   The default option is `Directional coupler`. The s-parameter matrix for `Directional coupler` is:

$$\begin{bmatrix} r_l & i_l & i_s & c \\ i_l & r_l & c & i_s \\ i_s & c & r_l & i_l \\ c & i_s & i_l & r_l \end{bmatrix}$$

where:

- $r_l = 10^{(\text{-ReturnLoss}/20)}$

- $i_l = 10^{(\text{-InsertionLoss}/20)}$

- $i_s = 10^{(\text{-(Coupling+Directivity)}/20)}$

- $c = 10^{(\text{-Coupling}/20)}$

- `Coupler symmetrical`

  The s-parameter matrix for `Coupler symmetrical` is:

  $$\begin{bmatrix} 0 & \alpha & 0 & j\beta \\ \alpha & 0 & j\beta & 0 \\ 0 & j\beta & 0 & \alpha \\ j\beta & 0 & \alpha & 0 \end{bmatrix}$$

  where:

  - $|\alpha| \leq 1$ = Power transmission coefficient
  - $\beta = \text{sqrt}(1 - \alpha^*\alpha)$

- `Coupler antisymmetrical`

  The s-parameter matrix for `Coupler antisymmetrical` is:

  $$\begin{bmatrix} 0 & \alpha & 0 & \beta \\ \alpha & 0 & -\beta & 0 \\ 0 & -\beta & 0 & \alpha \\ \beta & 0 & \alpha & 0 \end{bmatrix}$$

  where:

- $|\alpha| \leq 1$ = Power transmission coefficient.
- β = sqrt $(1 - \alpha^*\alpha)$

- `Hybrid quadrature (90deg)`

  The s-parameter matrix for `Hybrid quadrature(90deg)` is:

$$\begin{bmatrix} 0 & -j/\sqrt{2} & 0 & -1/\sqrt{2} \\ -j/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 0 & -j/\sqrt{2} \\ -1/\sqrt{2} & 0 & -j/\sqrt{2} & 0 \end{bmatrix}$$

- `Hybrid rat-race`

  The s-parameter matrix for `Hybrid rat-race` is:

$$\begin{bmatrix} 0 & -j/\sqrt{2} & 0 & -j/\sqrt{2} \\ -j/\sqrt{2} & 0 & j/\sqrt{2} & 0 \\ 0 & j/\sqrt{2} & 0 & -j/\sqrt{2} \\ -j/\sqrt{2} & 0 & -j/\sqrt{2} & 0 \end{bmatrix}$$

Coupling (dB) — Fraction of input signal power coupled to output port
0 (default) | nonnegative and real scalar

Fraction of input signal power coupled to output port of the `Directional coupler`, specified as a nonnegative and real scalar. The default value is 0 dB.

Dependencies

To enable this parameter, select `Directional coupler` in **Select component** tab.

Directivity (dB) — Ratio of power at coupled port to power at isolated port
inf (default) | nonnegative and real scalar

Ratio of power at coupled port to power at isolated port of the `Directional coupler`, specified as a nonnegative and real scalar. The default value is inf.

Dependencies

To enable this parameter, select `Directional coupler` in **Select component** tab.

`Insertion loss (dB)` — Loss of signal power between input and output ports
`inf` (default) | nonnegative and real scalar

Loss of signal power between input and output ports of the `Directional coupler`, specified as a nonnegative and real scalar. The default value is `inf`.

Dependencies

To enable this parameter, select `Directional coupler` in **Select component** tab.

`Return loss (dB)` — Loss of signal power due to impedance mismatch
`inf` (default) | nonnegative and real scalar

Loss of signal power due to impedance mismatch of the `Directional coupler`, specified as a nonnegative, and real scalar. The default value is `inf`.

Dependencies

To enable this parameter, select `Directional coupler` in **Select component** tab.

`Power transmission coeffecient` — Transmitted signal power
0 (default) | real scalar

Transmitted signal power of the `Directional coupler`, specified as a real scalar. The default value is 0.

Dependencies

To enable this parameter, select `Coupler symmetrical` or `Coupler antisymmetrical` in **Select component** tab.

`Reference impedances (Ohm)` — Reference impedance of coupler
50 (default) | positive scalar | three-tuple

Reference impedance of coupler, specified as a scalar or three-tuple. The default value is `50 Ohms`.

Ground and hide negative terminals — Ground RF circuit terminals
on (default) | off

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear this paramter. By exposing these terminals, you can connect them to other parts of your model.

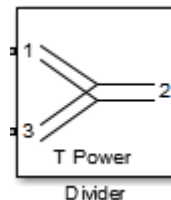By default, this option is selected.

See Also
`Circulator` | `Divider`

Introduced in R2014a

Divider

Model ideal frequency-independent dividers (combiners) with S-parameters
Library: SimRF / Circuit Envelope / Junctions

Description

The Divider block models power dividers (combiners) in a circuit envelope environment as an ideal s-parameter model.

Parameters

Select component — Divider type
T power divider (default) | Resistive power divider | Wlkinson power divider

Divider type, specified as:

- T power divider

  The s-parameter matrix for T-power divider is :

  $$\begin{bmatrix} s_{11} & s_{21} & s_{31} \\ s_{21} & s_{22} & s_{32} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}$$

  where:

  - $s_{11} = (z_{23} - z_1) / (z_{23} + z_1)$

  - $s_{22} = (z_{13} - z_2) / (z_{13} + z_2)$

  - $s_{33} = (z_{12} - z_3) / (z_{12} + z_3)$

- $s_{21} = (1 + s_{11}) * \text{sqrt}(z_1 / z_2)$
- $s_{31} = (1 + s_{11}) * \text{sqrt}(z_1 / z_3)$
- $s_{32} = (1 + s_{22}) * \text{sqrt}(z_2 / z_3)$
- $z_{12} = z_1 * z_2 / (z_1 + z_2)$
- $z_{13} = z_1 * z_3 / (z_1 + z_3)$
- $z_{23} = z_2 * z_3 / (z_2 + z_3)$
- $z_1 = Z0(1)$, $z_2 = Z0(2)$, $z_3 = Z0(3)$
- Reference Impedances: $Z0 = [z_1, z_2, z_3]$

- `Resistive power divider`

  The s-parameter matrix for `Resistive power divider` is :

  $$\begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

  When you select this option, the following parameter is available:

- `Wilkinson power divider`

  The s-parameter matrix for `Wilkinson power divider` is:

  $$\begin{bmatrix} 0 & -j/\sqrt{2} & -j/\sqrt{2} \\ -j/\sqrt{2} & 0 & 0 \\ -j/\sqrt{2} & 0 & 0 \end{bmatrix}$$

Note: For DC carrier (0 Hz) frequency, Wilkinson power divider is a zero matrix.

`Reference impedances (Ohm)` — Reference impedance of divider
`50` (default) | positive scalar | three-tuple vector

Reference impedance of divider, specified as a positive scalar or three-tuple vector.

`Ground and hide negative terminals` — Ground RF circuit terminals
`on` (default) | `off`

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear the parameter. By exposing these terminals, you can connect them to other parts of your model.

By default, this option is selected.
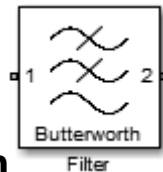
See Also
`Circulator` | `Coupler`

Introduced in R2014a

Filter

Model RF filter

Library

Elements

Description

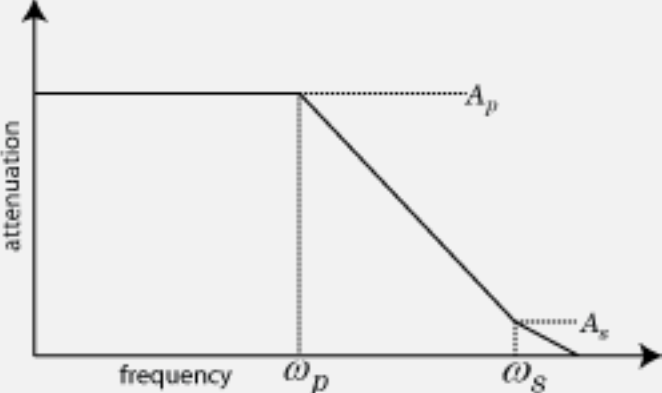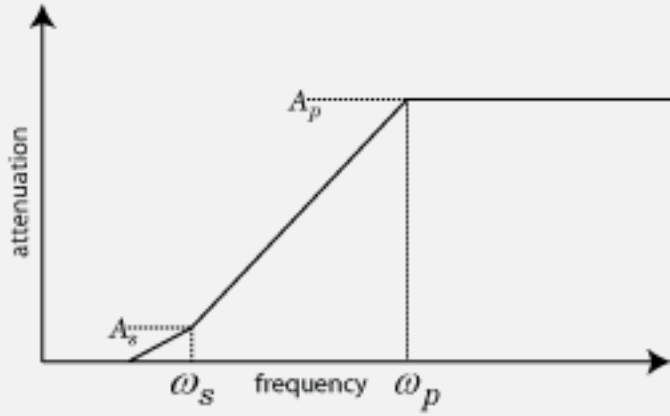The Filter block models RF filters of two designs:
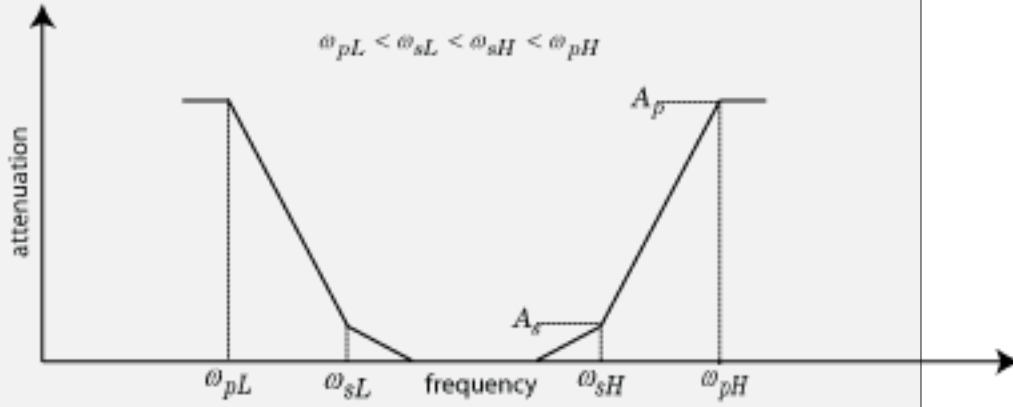
- Butterworth: Butterworth filters have a magnitude response that is maximally flat in the passband and monotonic overall. This smoothness comes at the price of decreased roll-off steepness.
- Chebyshev: Chebyshev Type I filters have equal ripple in the passband and monotonic in the stopband.

The frequency responses of the filter types are shown:

Filter Type	Frequency Response
Lowpass	 ω_p = passband frequency ω_s = stopband frequency A_p = passband attenuation @ ω_p A_s = stopband attenuation @ ω_s

Filter Type	Frequency Response
Highpass	ω_p = passband frequency ω_s = stopband frequency A_p = passband attenuation @ ω_p A_s = stopband attenuation @ ω_s

Filter Type	Frequency Response
Bandpass	$\omega_{sL} < \omega_{pL} < \omega_{pH} < \omega_{sH}$ (attenuation vs. frequency plot with A_p, A_s, ω_{sL}, ω_{pL}, ω_{pH}, ω_{sH}) ω_{pL}, ω_{pH} = passband frequencies ω_{sL}, ω_{sH} = stopband frequencies A_p = passband attenuation at specified passband frequencies A_s = stopband attenuation at specified stopband frequencies

Filter Type	Frequency Response
Bandstop	$\omega_{pL} < \omega_{sL} < \omega_{sH} < \omega_{pH}$ ω_{pL}, ω_{pH} = passband frequencies ω_{sL}, ω_{sH} = stopband frequencies A_p = passband attenuation at specified passband frequencies A_s = stopband attenuation at specified stopband frequencies

Parameters

Main

Design method

- Butterworth

  Simulates a butterworth filter of the type specified in **Filter type** and the model specified in **Implementation**.

- Chebyshev

  Simulates a chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

By default, the **Design method** is Butterworth.

Filter type

- `Lowpass`: Simulates a lowpass filter type of the design specified in **Design method**.
- `Highpass`: Simulates a highpass filter type of the design specified in **Design method**.
- `Bandpass`: Simulates a bandpass filter type of the design specified in **Design method**.
- `Bandstop`: Simulates a bandstop filter type of the design specified in **Design method**.

By default, the **Filter Type** is `Lowpass`.

Implementation

- `LC Tee`: Model an analog filter having a LC lumped Tee structure.
- `LC Pi`: Model an analog filter having a LC lumped Pi structure.
- `Transfer Function`: Model an analog filter using two-port S-parameters.

By default, the **Implementation** is `LC Tee`.

Implement using filter order

Select this check box to specify the **Filter order**.

Filter order

Specify the order of the filter. This order is the number of lumped storage elements in `lowpass` or `highpass`. In case of `bandpass` or `bandstop`, the number of lumped storage elements will be twice the value. This option is available only when **Implement using filter order** is selected. The default value of filter order is 3.

Note: For even order Chebyshev filters, the resistance ratio Rload/Rsrc > Rratio for Tee network implementation and Rload/Rsrc < 1/Rratio for Pi network implementation.

$$R_{ratio} = \frac{\sqrt{1+\epsilon^2} + \epsilon}{\sqrt{1+\epsilon^2} - \epsilon}$$

where:

- $\epsilon = \sqrt{10^{(0.1R_p)} - 1}$

- R_p is the passband ripple in dB.

Passband frequency

When the **Filter type** is set to `Lowpass` or `Highpass`, specify the passband frequency as a scalar in `Hz`, `kHz`, `MHz`, `GHz`. The default value is 1 GHz.

Passband frequencies

When the **Filter type** is set to `Bandpass`, specify the passband edge frequencies as a 2–tuple vector in `Hz`, `kHz`, `MHz`, `GHz`. The default value is [2 3] GHz.

Note: This tab is also available for `Bandstop` filters when **Implement using filter order** is not selected.

Passband attenuation (dB)

Specify the passband attenuation in `dB`. For bandpass filters this value is applied equally to both edges of the passband.

Stopband frequency

When the **Filter type** is set to `Lowpass` or `Highpass`, specify the stopband frequency as a scalar in `Hz`, `kHz`, `MHz`, `GHz`. The default value is 2 GHz.

Stopband frequencies

When the **Filter type** is set to `Bandstop`, specify the stopband edge frequencies as a 2–tuple vector in `Hz`, `kHz`, `MHz`, `GHz`. The default value is [2.1 2.9] GHz.

Note: This tab is also available for `Bandpass` filters when **Implement using filter order** is not selected.

Stopband attenuation (dB)

Specify the stopband attenuation in `dB`. For bandstop filters, this value is applied equally to both edges of the stopband. The default value is 40 dB.

Source impedance (Ohm)

Specify the value of input source resistance in ohms. The default value is 50 ohms.

Load impedance (Ohm)

Specify the value of output load resistance in ohms. The default value is 50 ohms.

Ground and hide negative terminals

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

By default, this option is selected.

Export

Save the filter design to a file. Valid file types are .mat and .txt .

Main Combination Options

	Lowpass	Highpass	Bandpass	Bandstop
Butterworth	Order, ω_p, A_p	Order, ω_p, A_p	Order, ω_{pL}, ω_{pH}, A_p	Order, ω_{sL}, A_p, ω_{sH}, A_s
	ω_p, A_p, ω_s, A_s	ω_p, A_p, ω_s, A_s	$(\omega_{pL}, \omega_{pH})$, A_p, $(\omega_{sL}, \omega_{sH})$, A_s	$(\omega_{pL}, \omega_{pH})$, A_p, $(\omega_{sL}, \omega_{sH})$, A_s
Chebyshev	Order, ω_p, A_p	Order, ω_p, A_p	Order, ω_{pL}, ω_{pH}, A_p	Order, ω_{sL}, A_p, ω_{sH}, A_s
	ω_p, A_p, ω_s, A_s	ω_p, A_p, ω_s, A_s	$(\omega_{pL}, \omega_{pH})$, A_p, $(\omega_{sL}, \omega_{sH})$, A_s	$(\omega_{pL}, \omega_{pH})$, A_p, $(\omega_{sL}, \omega_{sH})$, A_s
	Order = filter order, ω_p = passband frequency, A_p = passband attenuation in dB, ω_{pL}, ω_{pH} = passband frequencies, ω_{sL}, ω_{sH} = stopband frequencies, A_s = stopband attenuation			

Visualization Tab

Parameter1, Parameter2

For **Parameter1**, specify the plots on left y-axis from : `Voltage transfer`, `Phase delay` or `Group delay`. The default value `Voltage transfer`.

For **Parameter2**, specify the plots on left y-axis from : `Voltage transfer`, `Phase delay` or `Group delay`. If the model has only one parameter specify the plots as `None`. The default value `None`.

The options in **Parameter1** and **Parameter2** are mutually exclusive.

Format1, Format2

Specify the scaling of the y-axis for **Parameter1** and **Parameter2**.

- For `Voltage transfer` parameters, specify the format as `Magnitude(decibels)`, `Magnitude(linear)` or `Angle(degrees)`, `Real`, or `Imaginary`.
- For `Phase delay` or `Group delay` parameters, specify the format as `Magnitude(decibels)` or `Magnitude(linear)`.

The default value is `Magnitude (decibels)`.

Frequency points

Specify frequency points to plot on the x-axis in `Hz`, `kHz`, `MHz`, `GHz`.

Y-axis scale

Specify the scale for the y-axis. The default value is `Linear`.

X-axis scale

Specify the scale for the x-axis. The default value is `Linear`.
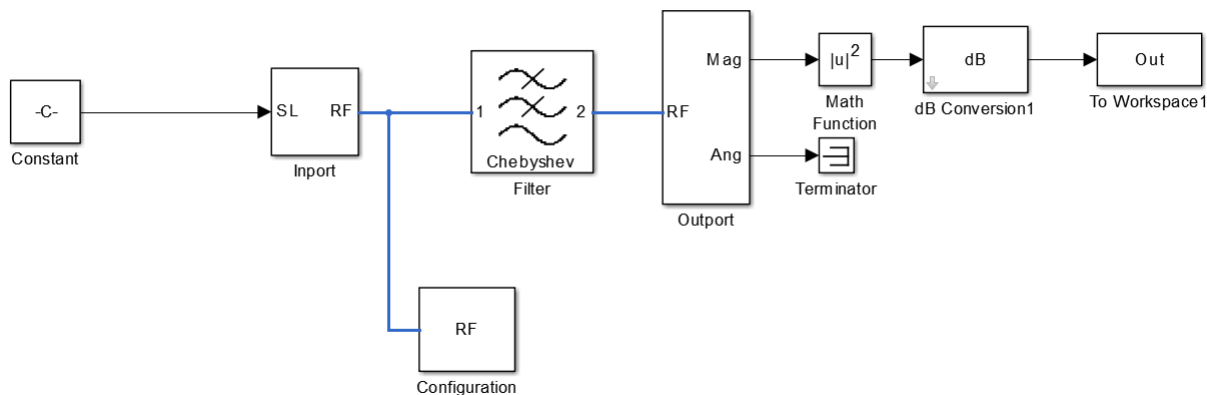
Examples

Frequency Response of Lowpass Chebyshev Filter

1 Use the Filter block to study the frequency response of a lowpass chebyshev filter.

2 From the MATLAB® command prompt, open the model `ex_filter_lowpass_cheby_resp`

```
addpath(fullfile(docroot,'toolbox','simrf','examples'))
ex_simrf_filter_lowpass_cheby_resp
```

3 The Constant block defines the amplitude of the 201 signals with `ones(1,201)`.

4 The Inport block defines the carrier frequencies of the signal. The carrier frequencies are `logspace(7,9,201)`. The Configuration block defines the step size.

5 In this example, the Filter block design method is a Chebyshev filter. This filter is a lowpass LC pi filter with a filter order of 3. The bandwidth of the filter is 200 MHz. The passband frequency is 600 MHz and stopband frequency is 700 MHz.

6 The Filter block output signal is fed into the Outport. The output block is configured to give both the magnitude and angle of the signal. The angle output is terminated with the terminator block.

7 To run the model, select `Simulation > Run`. The model creates an `Out` array in the MATLAB workspace.

8 To plot the frequency response, use the following commands in the MATLAB Workspace:

```
figure
freq = logspace(7,9,201);
h = semilogx(freq, Out,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
xlabel('Frequency[Hz]');
ylabel('Amplitude[dB]')
title('Frequency Response of Lowpass Chebyshev Filter')
```

9 Notice the ripples in the passband of the chebyshev filter.
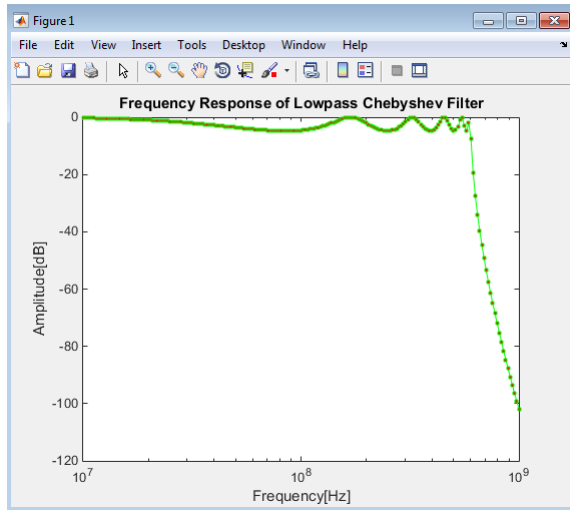
Frequency Response of Lowpass Butterworth Filter

1 In `ex_simrf_filter_lowpass_cheby_resp`, change the design method of Filter block to Butterworth.

2 To run the model, select **Simulation Run**. The model creates an Out array in the MATLAB workspace.

3 To plot the frequency response, use the following commands in the MATLAB Workspace:

```
figure
freq = logspace(7,9,201);
h = semilogx(freq, Out,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
xlabel('Frequency[Hz]');
ylabel('Amplitude[dB]')
title('Frequency Response of Lowpass Butterworth Filter')
```
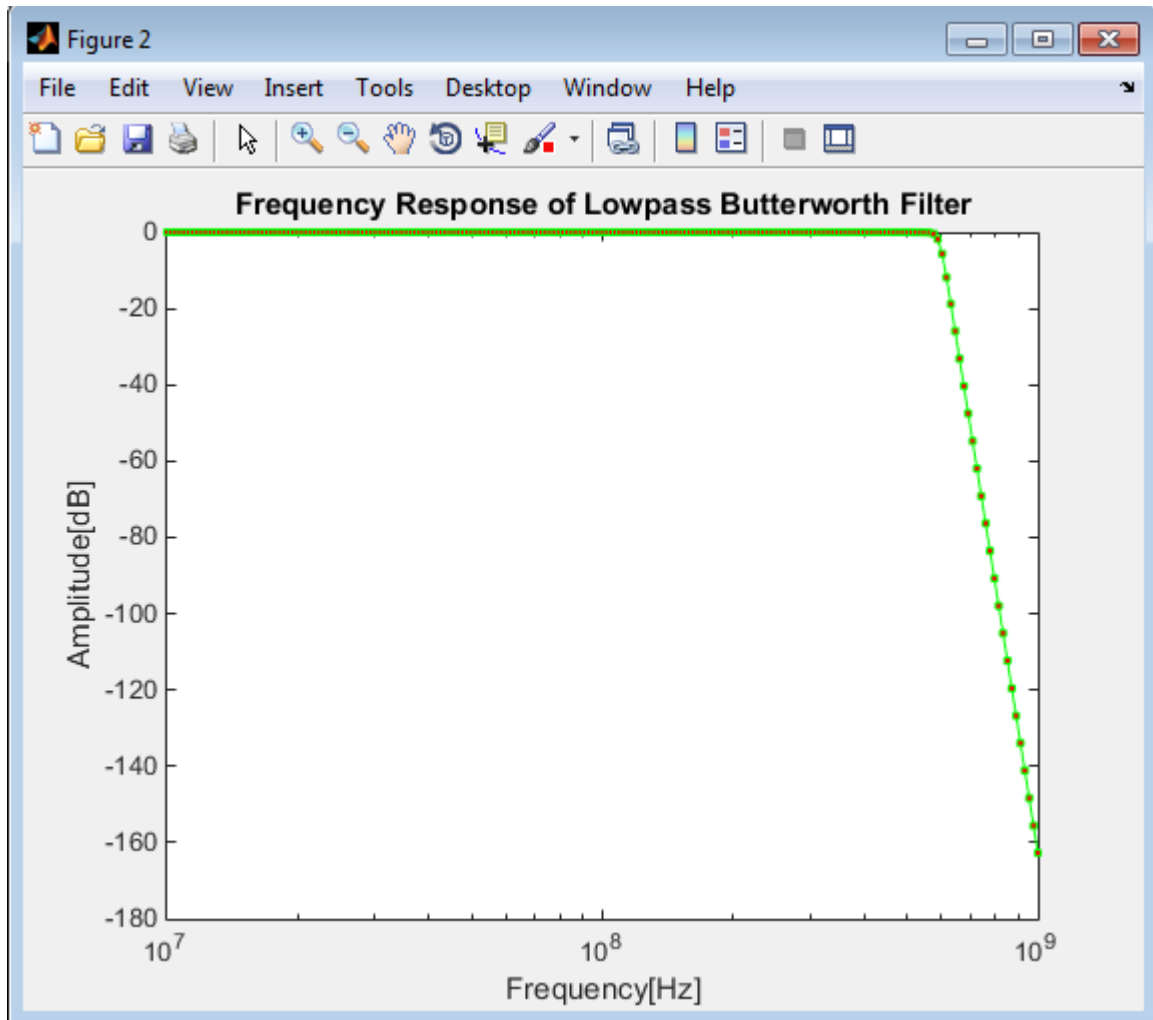
References

[1] Kendall Su, *Analog Filters, Second Edition*.

[2] Louis Weinberg, *Network Analysis and Synthesis*, Huntington, New York: Robert E. Krieger Publishing Company, 1975.

[3] Larry D. Paarmann, *Design and Analysis of Analog Filters, A Signal Processing Perspective with MATLAB Examples*, Kluwer Academic Publishers, 2001.

[4] Michael G. Ellis, SR., *Electronic Filter Analysis and Synthesis*, Norwood, MA: Artech House, 1994.

[5] Anatol I. Zverev, *Handbook of Filter Synthesis*, Hoboken, NJ: John Wiley & Sons, 2005.

Ground

Simulate connection to electrical ground
Library: SimRF / Circuit Envelope / Utilities

Gnd
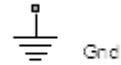
Description

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.

Parameters

The Ground block has no parameters.

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

Introduced in R2016b

Ideal Transformer

Model ideal transformer

Library

Elements
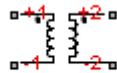
Ideal Transformer

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, "Simulate High Frequency Components".

Parameters

Winding ratio

Specify the winding ratio.

See Also

`Mutual Inductor | Three Winding Transformer`

Inport

Convert Simulink input signal to SimRF signal
Library: SimRF / Circuit Envelope / Utilities

Inport

Description

The Inport block imports Simulink® signals into the SimRF circuit envelope simulation environment. For an introduction to RF simulation, see the example, "Simulate High Frequency Components".

Complex-valued input signals $I_k(t) + j \cdot Q_k(t)$ are the modulations at the frequencies $\{f_k\}$ specified in the **Carrier frequencies** parameter of the block.

The **Source type** parameter specifies the Simulink signal as either current, or voltage, or power source.

Parameters

Source type — Inport block interpretation of Simulink signal
Ideal voltage (default) | Ideal current | Power

Inport block interpretation of Simulink signal, specified as:

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

1-55
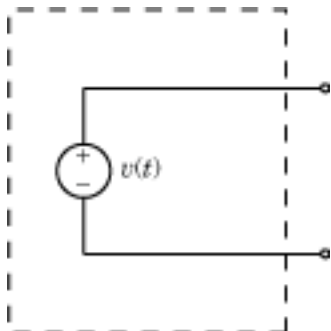
- Ideal current — The block outputs Simulink signals as current signals $i(t)$ in the SimRF environment. When you choose this option, the following figure illustrates the internal configuration of the block.
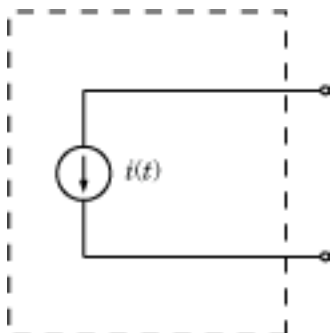
- Power — The block interprets the Simulink signals, $P_v(t)$, as available power and internally uses a voltage source, and series impedance as shown in the following figure.

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 real power to the load Z_l:

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

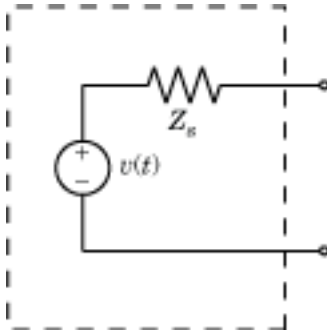When $Z_l = Z_s^*$, this generator delivers the available power $|v_{SL}(t)|^2$.

Source impedance (Ohm) — Source impedance for available power match
50 (default) | vector of positive integers in ohms

Fundamental tones of a set of simulation frequencies, specified as a vector of positive integers in ohms.

Dependencies

To enable this parameter, select Power in **Source type**.

Carrier frequencies — Carrier frequencies
0 Hz (default) | vector of positive integers in Hz

Carrier frequencies, specified as a vector of positive integers in Hz. In carrier frequencies, the elements are a combination of fundamental tones and corresponding harmonics in the `Configuration` block.

Ground and hide negative terminals — Ground RF circuit terminals
on (default) | off

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

By default, this option is selected.

Model Examples

See Also
`Configuration` | `Outport`

Introduced in R2010b

Z (Impedance)

Model complex impedance

Library

Elements
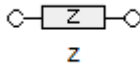
Description

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.

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 dependent`, 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`

L (Inductor)

Model inductor for circuit envelope analysis

Library

Elements
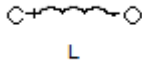
Description

The Inductor block models an inductor in circuit envelope environment.

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.

Note: Harmonic balance method calculates the steady state response of SimRF systems. For further information please refer to Kenneth, S.Kundrt. " Introduction to RF Simulation." *The Designers Guide Community*, April 2003, http://www.designers-guide.org/analysis/rf-sim.pdf.

- 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.
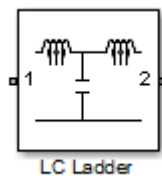
See Also

Capacitor | Resistor | Mutual Inductor | Three-Winding Transformer

LC Ladder

Model LC ladder networks

Library

Elements

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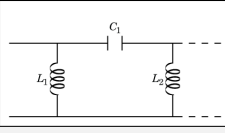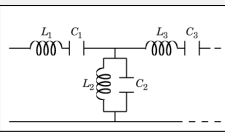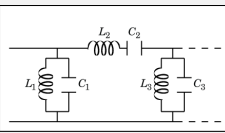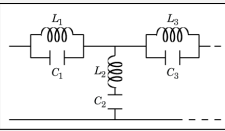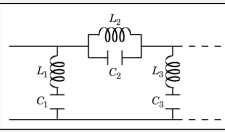
The LC Ladder block models common two-port LC lossless networks.

Parameters

Specify ladder topology from the following options:

Ladder Topology

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

By default, this option is selected.

Examples

- The section, "Filter Mixing Products", shows the use of a LC Ladder.

Mixer

Model mixer in RF systems
Library: SimRF / Circuit Envelope / Elements
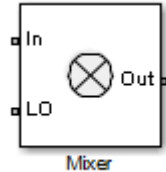
Mixer

Description

The Mixer block performs signal frequency translation and nonlinear amplification.

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.

Parameters

Main

Source of conversion gain — Source parameter of conversion gain
`Available power gain` (default) | `Open circuit voltage gain` | `Polynomial coefficients`

Source parameter of conversion gain, specified as one of the following:

- `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, $[a_0, a_1, a_2, ..., a_9]$, the polynomial it represents is $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \cdots + a_9 V_{in}^9$. 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 $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \cdots + a_3 V_{in}^3$.

  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 $V_o = V_i$.

Available power gain — Linear gain of mixer
`0 dB` (default) | scalar

Linear gain of mixer, specified as a scalar in dB. Specify the units from the corresponding drop-down list.

Dependencies

To enable this parameter, select `Available power gain` in **Source of conversion gain** tab.

Open circuit voltage gain — Open circuit voltage gain

0 dB (default) | scalar

Open circuit voltage of mixer, specified as a scalar in dB. Specify the units from the corresponding drop-down list.

Dependencies

To enable this parameter, select `Open circuit voltage gain` in **Source of conversion gain** tab.

Polynomial coefficients — Order of polynomial

[0 1] (default) | vector

Order of polynomial, specified as a vector.

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:

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

where a_1 represents the linear gain term, and higher-order terms are modeled according to [1].

For example, the vector $[a_0, a_1, a_2, a_3]$ specifies the relation $V_o = a_0 + a_1 V_1 + a_2 V_1^2 + a_3 V_1^3$. 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 $V_o = V_i$.

Dependencies

To enable this parameter, select `Polynomial coefficients` in **Source of conversion gain** tab.

`Input impedance (Ohm)` — Input impedance of mixer
50 (default) | scalar

Input impedance of mixer, specified as a scalar.

`Output impedance (Ohm)` — Output impedance of mixer
50 (default) | scalar

Output impedance of mixer, specified as a scalar.

`LO impedance (Ohm)` — Impedance at LO port of mixer
inf (default) | scalar

Output impedance of mixer, specified as a scalar.

`Noise figure (dB)` — Single-sideband noise figure of mixer
0 dB (default) | scalar

Single-sideband noise figure of mixer, specified as a scalar.

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 $\omega_{\mathrm{IM}} = \omega_{\mathrm{LO}} + (\omega_{\mathrm{LO}} - \omega_{\mathrm{RF}})$.

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` — Ground RF circuit terminals
on (default) | off

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

By default, this option is selected.

Nonlinearity

Nonlinear polynomial type — Polynomial nonlinearity
Even and odd order (default) | Odd order

Polynomial nonlinearity, specified as one of the following:

- 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.

- Odd order: 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 figure shows the graphical definition of the nonlinear amplifier parameters.

`Intercept points convention` — Intercept points convention
`Output` (default) | `Input`

Intercept points convention, specified as `Input`-referred or `Output`-referred. Use this specification for the intercept points, 1-dB gain compression power, and saturation power.

`IP2` — Second-order intercept point
`infdBm` (default) | scalar

Second-order intercept point. specified as a scalar. The default value `inf dBm` corresponds to an unspecified point.

Dependencies

To enable this parameter, select `Even and odd order` in **Nonlinear polynomial type** tab.

`IP3` — Third-order intercept point
`inf dBm` (default) | scalar

Third-order intercept point, specified as a scalar. The default value `inf dBm` corresponds to an unspecified point.

`1-dB gain compression power` — 1-dB gain compression power
`infdBm` (default) | scalar

1-dB gain compression power, specified as a scalar. The 1-dB gain compression point must be less than the output saturation power.

Dependencies

To enable this parameter, select `Odd order` in **Nonlinear polynomial type** tab.

`Output saturation power` — Output saturation power
`infdBm` (default) | scalar

Output saturation power, specified as a scalar. 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.

Dependencies

To enable this parameter, select `Odd order` in **Nonlinear polynomial type** tab.

Gain compression at saturation — Gain compression at saturation
`infdBm` (default) | scalar

Gain compression at saturation, specified as a scalar.

Dependencies

To enable this parameter, select `Odd order` in **Nonlinear polynomial type** tab and set **Output saturation power** .

Model Examples

References

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

See Also
`Amplifier | S-Parameters`

Introduced in R2010b

Mutual Inductor

Model two coupled inductors for circuit envelope analysis

Library

Elements

Mutual Inductor

Description

The Mutual Inductor block models an inductor within the SimRF circuit envelope simulation environment. For an introduction to RF simulation, see the example, "Simulate High Frequency Components".

The block implements the relations

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

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

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

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:

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

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

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

Model noise using current or voltage noise source in RF systems
Library: SimRF / Circuit Envelope / Sources
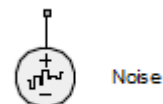
Description

Use the Noise block to model noise as a ideal current or voltage source for blocks. When you use a Noise to simulate noise in an RF model, also select the **Simulate noise** check box in the `Configuration` block. Otherwise, the model simulates without noise. The Noise block does not depend on the Temperature parameter in the `Configuration`.

Parameters

Source type — Noise type
Ideal voltage (default) | Ideal current

Noise type, specified as `Ideal voltage` or `Ideal current`.

Noise distribution — Noise distribution type
White (default) | Piece-wise linear | Colored

Noise distribution type, specified as `White`, `Piece-wise linear`, or `Colored`.

Noise power spectral density — Single-sided noise power spectral distribution (PSD)
0 (default)

Single-sided noise power spectral distribution (PSD), specified as:

- `White`, spectral density is a single non-negative value. The power value of the noise depends on the bandwidth of the carrier and the bandwidth depends on the time step. This is an uncorrelated noise source.

- `Piece-wise linear`, spectral density is a vector of values [p_i]. For each carrier, the noise source behaves like a white uncorrelated noise. The power of the noise source is carrier-dependent.
- `Colored`, depends on both carrier and bandwidth. This is a correlated noise source.

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 — Frequencies for piece-wise linear noise distribution
`0 Hz` (default) | vector of non-negative frequencies | `Hz` | `kHz` | `MHz` | `GHz`

Frequencies for piece-wise linear noise distribution, specified as vector of non-negative frequencies.

Ground and hide negative terminals — Ground RF circuit terminals
`on` (default) | `off`

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

By default, this option is selected.

Model Examples

See Also
`Continuous Wave` | `Sinusoid`

Introduced in R2010b

Outport

Convert SimRF signal to Simulink output signals
Library: SimRF / Circuit Envelope / Utilities
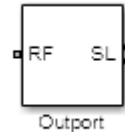
Description

The Outport block outputs carrier modulation signals in the SimRF circuit envelope simulation environment as Simulink signal. For an introduction to RF simulation, see the example, "Simulate High Frequency Components".

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

The **Sensor type** parameter determines which signal the block measures, and the **Output** parameter defines the format of the Simulink signal.

Parameters

Sensor type — Type of signal measured by sensor
`Ideal voltage` (default) | `Ideal current` | `Power`

Type of signal measured by sensor, specified as:

- `Ideal voltage` — The block outputs the modulations of the voltage signal at the specified **Carrier frequencies** in the format specified by the **Output** parameter.
- `Ideal current` — The block outputs the modulations of the current signal at the specified **Carrier frequencies** in the format specified by the **Output** parameter.
- `Power` — The block outputs the modulations of the voltage signal

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

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

Load impedance (Ohm) — Load impedance of RF circuit
inf (default) | vector of positive integers in ohms

Load impedance of RF circuit, specified as a vector of positive integers in ohms. 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.

Dependencies

To enable this parameter, select Power in **Sensor type**.

Output — Format of output signals
Complex Baseband (default) | In-phase and Quadrature Baseband | Magnitude and Angle Baseband | Real Passband

Format of output signals, specified as one of the following:

- Complex Baseband — The block outputs a vector of complex-valued signals $I_k(t) + j \cdot Q_k(t)$ at the port labeled **SL**. 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 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 real-valued vectors, 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 by combining envelope and carrier signals for all frequencies listed under **Carrier frequencies**. When using the Real Passband option, the solver takes time steps small enough to resolve the carrier. Thus, simulation speed improvements from envelope simulation may be lost.

`Automatically compute output step size` — Determine optimal time step to resolve highest listed carrier frequency
`on` (default) | `off`

Select this parameter to allow SimRF to determine the optimal time step to resolve the highest listed carrier frequency. Clear the parameter selection to enter a value for step size.

`Step size` — Time step
`1e-6 s` (default) | positive integer in seconds

Time step, specified as a positive integer in seconds. The step size should be small enough to resolve the fastest carrier signal. The size helps to avoid passband output undersampling and aliasing effects.

Set the time step value to `-1` to inherit the time step specified from `Step size` in Configuration block.

`Carrier frequencies` — Carrier frequencies
`0 Hz` (default) | vector of positive integers in Hz

Carrier frequencies, specified as a vector of positive integers in Hz. In carrier frequencies, the elements are a combination of fundamental tones and corresponding harmonics in the `Configuration` block.

`Ground and hide negative terminals` — Ground RF circuit terminals
`on` (default) | `off`

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

By default, this option is selected.

Model Examples

More About

Real Passband Formula

Normalized carrier power option in the `Configuration` block defines the passband formula:

- When this option is selected, SimRF interprets complex envelope I+jQ signal for the kth carrier as,

$$s_k(t) = I(t)\sqrt{2}\cos(2\pi f_k t) - Q(t)\sqrt{2}\sin(2\pi f_k t)$$

- When this option is not selected, the signal on the kth

$$s_k(t) = I(t)\cos(2\pi f_k t) - Q(t)\sin(2\pi f_k t)$$

- In both cases, the signal for zero-frequency (DC) carrier is $x(t) = I(t)$. The final output signal is computed as $s(t) = \text{sum}(s_k)$

Formula for Time Step

The formula for the time step selected is:

$$\min(h, ((1/2\pi f) \div 8))$$

- f is the largest listed carrier frequency.
- h is the time step listed in Configuration block.

See Also
`Configuration` | `Inport`

Introduced in R2010b

Phase Shift

Model phase shift in RF systems

Library

Elements

Phase Shift

Description

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

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

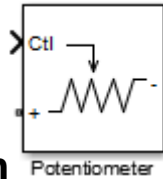By default, this option is selected.

Examples

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

Potentiometer

Simulink controlled potentiometer

Library

Junctions

Potentiometer

Description

The Potentiometer block models a variable resistor. The Potentiometer block uses Simulink signal as input control signal to vary the resistance between the positive and the negative terminals.

The equations for the potentiometer voltage–current relationships when the control signal varies between zero and one are:

Linear: $\quad V_{pot} = I_{pot}.(R_{off}.Ctl + R_{on}.(1 - Ctl))$

Logarithmic: $V_{pot} = I_{pot}.(C_1.\exp(C_2.Ctl) + C_3)$

Antilog: $\quad V_{pot} = I_{pot}.(C_1.\log(1 + C_2.Ctl) + C_3.Ctl + R_{on})$

- Linear, Logarithmic, Antilog — Electrical characteristic of the potentiometer.
- Ctl — Simulink signal
- R$_{on}$ — Minimum resistance of the potentiometer
- R$_{off}$ — Maximum resistance of the potentiometer

Parameters

Minimum resistance

Minimum input resistance of the potentiometer, specified as a positive scalar. The default value is `10` ohms.

Maximum resistance

Maximum input resistance of the potentiometer, specified as a positive scalar. The default value is `50` ohms. This value must be greater than the minimum resistance.

Potentiometer type

Curve type of the potentiometer: `Linear`, `Logarithmic`, or `Antilog`. The default is `Linear`.

Percentage of resistance at half wiper

Percentage of resistance at half wiper for logarithmic and antilog potentiometer types, specified as a positive scalar. The valid values depend on the potentiometer type:

- `Logarithmic` — Greater than 0 ohms and less than 50 ohms. The default is 20.
- `Antilog` — Greater than 50 ohms and less than 100 ohms. The default is 80.
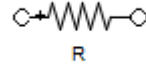
See Also

`Switch` | `SPST` | `SPDT`

Introduced in R2015b

Resistor

Model resistor for circuit envelope analysis
Library: SimRF / Circuit Envelope / Elements

Description

The Resistor block models a resistor within the SimRF circuit envelope simulation environment. For an introduction to RF simulation, see the example, "Simulate High Frequency Components".

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:

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

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

Parameters

`Resistance` — Resistance value
`50 Ohm` (default) | real number in ohms

Resistance value , specified as a real number in ohms greater than zero. Specify the units of the resistance from the corresponding drop-down menu.

`Simulate noise` — Simulate thermal noise
`on` (default) | `off`

Select this parameter, to simulate thermal noise in a resistor. 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 $\mathbf{k}T/R$ A^2/Hz, where:

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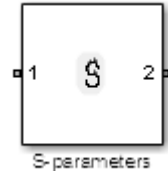- T is the value of the **Temperature** parameter, in degrees Kelvin. (Also located in the `Configuration` block.)

See Also
`Capacitor` | `Inductor`

S-Parameters

Model S-parameter network
Library: SimRF / Circuit Envelope / 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, "Simulate High Frequency Components".

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.

Parameters

Main

`Data source — Data source`
`Data file` (default) | `Network-parameters` | `Rational model`

Data source for S-parameters behavior, specified as one of the following:

- `Data file` — Name of a Touchstone file with the extension `.s2p`. The block ignores noise and nonlinearity data in imported files.

- `Network-parameters` — Provide **Network parameter** data such as `S-parameters`, `Y-parameters`, and `Z-parameters` with corresponding **Frequency** and **Reference impedance (ohms)** for the s-parameters.

- `Rational model` — Provide values for **Residues**, **Poles**, and **Direct feedthrough** parameters which correspond 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 rational model equation, each C_k is the residue of the pole A_k. If C_k is complex, a corresponding complex conjugate pole and residue must also be enumerated. This object has the properties `C`, `A`, and `D`. You can use these properties to specify the **Residues**, **Poles**, and **Direct feedthrough** parameters.

Data file — Name of network parameter data file
simrfV2_unitygain.s2p (default) | character vector

Name of network parameter data file, specified as a character vector.

Dependencies

To enable this parameter, select `Data file` in **Data source** tab.

Network parameter type — Network parameter type
S-parameters (default) | Y-parameters | Z-parameters

Network parameter type, specified as S-parameters, Y-parameters, or Z-parameters.

Dependencies

To enable this parameter, select `Network-parameters` in **Data source** tab.

Network-parameters — Network parameter values
[0 0;1 0] (default) | multidimensional array

Network parameter values specified as a multidimensional array. The third dimension of the S-parameter array must be the same length as the vector of frequencies specified by the **Frequency** parameter. The default values are different for S-parameters, Y-parameters, and Z-parameters respectively.

Dependencies

To enable this parameter, select `Network-parameters` in **Data source** tab.

`Frequency (dB)` — Frequency of network parameters
`1e9 Hz` (default) | scalar | `Hz` | `kHz` | `MHz` | `GHz`

Frequency of network parameters, specified as a scalar in Hz.

Dependencies

To enable this parameter, select `Network-parameters` in **Data source** tab.

`Reference Impedance(Ohm)` — Reference impedance of network parameters
`50` (default) | scalar

Reference impedance of network parameters, specified as a scalar.

Dependencies

To enable this parameter, select `Network-parameters` in **Data source** tab.

`Residues` — Residues in order of rational model
`0` (default) | vector

Residues in order of rational model, specified as a vector.

Dependencies

To enable this parameter, select `Rational model` in **Data source** tab.

`Poles` — Residues in order of rational model
`0` (default) | vector

Poles in order of rational model, specified as a vector.

Dependencies

To enable this parameter, select `Rational model` in **Data source** tab.

`Direct feedthrough` — Direct feedthrough
`{0 0:1 0}` (default) | array of vectors

Direct feedthrough, specified as an array vector.

Dependencies

To enable this parameter, select `Rational model` in **Data source** tab. .

`Simulate noise` — Generate thermal noise waves
`off` (default) | `on`

Choose this parameter to generate thermal noise waves [1]. Clear this parameter to stop simulating noise. For more information see, "Generate Thermal Noise" on page 1-93.

`Ground and hide negative terminals` — Ground RF circuit terminals
`on` (default) | `off`

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

By default, this option is selected.

Modeling

`Modeling options` — Model S-parameters
`Time-domain (rationalfit)` (default) | `Frequency-domain`

Model S-parameters, specified as:

• Time-domain (rationalfit) technique creates an analytical rational model that approximates the whole range of the data. When modeling using `Time domain`, the **Plot** in `Visualization` tab plots the data defined in `Data Source` and the values in the `rationalfit` function.

- Frequency-domain computes the baseband impulse response for each carrier frequency independently. This technique is based on convolution. There is an option to specify the duration of the impulse response. For more information, see "Compare Time and Frequency Domain Simulation Options for S-parameters".

- 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`.

Dependencies

To set this parameter, first select `Data file` or `Network-parameters` in **Data source**. This selection activates the **Visualization** Tab which contains **Source of frequency data**

Fitting options — Rationalfit fitting options
`Fit individually` (default) | `Share poles by column` | `Share all poles`

Rationalfit fitting options, specified as `Fit individually`, `Share poles by column`, or `Share all poles`.

Rational fitting results shows values of **Number of independent fits**, **Number of required poles**, and **Relative error achieved (dB)**.

Dependencies

To set this parameter, select `Time domain (rationalfit)` in **Modeling options**.

Relative error desired (dB) — Relative error acceptable for the rational fit
`-40` (default) | scalar

Relative error acceptable for the rational fit, specified as a scalar.

Dependencies

To set this parameter, select `Time domain (rationalfit)` in **Modeling options**.

`Automatically estimate impulse response duration` — **Automatically calculate impulse response**
`on` | `off`

Select this parameter to automatically calculate impulse response. Clear this parameter to manually specify the impulse response duration using **Impulse response duration**.

Dependencies

To set this parameter, select `Frequency domain` in **Modeling options**.

`Impulse response duration` — **Impulse response duration**
`1e-10` (default) | scalar

Impulse response duration, specified as a scalar.

Dependencies

To set this parameter, first select `Frequency domain` in **Modeling options**. Then, clear `Automatically estimate impulse response duration`.

Visualization

`Source of frequency data` — **Frequency data source**
`Extracted from data source` (default) | `User-defined`

Frequency data source, specified as:

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.

`Frequency data` — **Frequency data range**
`[1e9:1e6:3e9]` (default) | vector | `Hz` | `kHz` | `MHz` | `GHz`

1-91

Frequency data range, specified as a vector

`Plot type` — Type of data plot
X-Y plane (default) | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot that you want to produce with your data specified as one of the following:

- `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.

`Parameter 1` — Type of S-Parameters to plot
S11 (default) | S12 | S21 | S22

Type of S-Parameters to plot, specified as S11, S12, S21, or S22.

`Parameter 2` — Type of S-Parameters to plot
None (default) | S11 | S12 | S21 | S22

Type of S-Parameters to plot, specified as S11, S12, S21, or S22.

`Format1` — Plot format
Magnitude (decibels) (default) | Angle(degrees) | Real | Imaginary

Plot format, specified as Magnitude (decibels), Angle(degrees), Real, or Imaginary.

`Format2` — Plot format
Magnitude (decibels) (default) | Angle(degrees) | Real | Imaginary

Plot format, specified as Magnitude (decibels), Angle(degrees), Real, or Imaginary.

`Y-axis scale` — Y-axis scale
Linear (default) | Logarithmic

Y-axis scale, specified as Linear or Logarithmic.

X-axis scale — X-axis scale
Linear (default) | Logarithmic

X-axis scale, specified as Linear or Logarithmic.
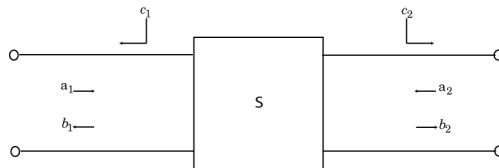
Plot — Plot specified data
button

Plot specified data using plot button.

Model Examples

More About

Generate Thermal Noise

You can only generate thermal noise if the given S-parameters multiport components are passive.

To include the noise waves, the block augments the S-parameters equation:

$$b = Sa + c$$

- a and b — Customary wave vectors
- c — Noise wave vector

The noise wave vector, noise correlation matrix, and S-parameters have these relationships:

$$C_s = K * T(I - SS^*) = \overline{cc}$$

where:

- K — Boltzmann's constant
- T — System temperature
- c — Noise wave vector

References

[1] Wedge, Scott & Rutledge, David. " Wave Techniques for Noise Modeling and Measurement" *IEEE Transactions on Microwave Theory and Techniques.* Vol. 40, Number 11, pp. 2004–2012, Nov. 1992.

See Also
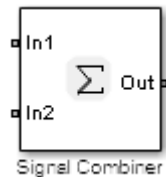
Amplifier | Mixer | Transmission Line

Introduced in R2010b

Signal Combiner

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 an introduction to RF simulation, see the example, "Simulate High Frequency Components"

Parameters

Ground and hide negative terminals

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

By default, this option 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

Model DC offset and sinusoidal modulation
Library: SimRF / Circuit Envelope / 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 kth 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.

Parameters

Source type — Wave type
Ideal voltage (default) | Ideal current

Wave type, specified 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**.

Offset in-phase — In-phase offset

0 V (default) | 0 A | vector of real or complex numbers | V | mV | kV | A | mA | uA | kA

In-phase offset for each of the RF circuit carrier frequencies, specified as a vector of real or complex numbers. Specify the units from the corresponding drop-down list. The units are in volts for `Ideal Voltage` and amperes for `Ideal Current`.

Offset quadrature — Quadrature offset

0 V (default) | 0 A | vector of real or complex numbers | V | mV | kV | A | mA | uA | kA

Quadrature offset for each of the RF circuit carrier frequencies, specified as a vector of real or complex numbers. Specify the units from the corresponding drop-down list. The units are in volts for `Ideal Voltage` and amperes for `Ideal Current`.

Sinusoidal amplitude in-phase — In-phase amplitude

0 (default) | vector of real or complex numbers | V | mV | kV | A | mA | uA | kA

In-phase amplitude, specified as a vector of real number or a complex number. Specify the units from the corresponding drop-down list. The units are in volts for `Ideal Voltage` and amperes for `Ideal Current`.

Sinusoidal amplitude quadrature — Quadrature amplitude

0 (default) | vector of real or complex numbers | V | mV | kV | A | mA | uA | kA

Quadrature amplitude, specified as a vector of real number or a complex number. Specify the units from the corresponding drop-down list. The units are in volts for `Ideal Voltage` and amperes for `Ideal Current`.

Sinusoidal modulation frequency — Modulation frequency

0 (default) | vector of real or complex numbers | Hz | kHz | MHz | GHz

Modulation frequency at each of the RF circuit carrier frequencies, specified as a vector of real number or a complex number. Specify the units from the corresponding drop-down list. Th

`Time delay` **— Time delay of modulation**
`0` (default) | vector of real | `s` | `ms` | `us` | `ns`

Time delay of modulation, specified as a vector of positive numbers. Specify the units from the corresponding drop-down list.

`Carrier frequencies` **— Carrier frequencies**
`0` (default) | vector of real positive numbers | `Hz` | `kHz` | `MHz` | `GHz`

Carrier frequencies, specified as a vector of real positive numbers. The elements in the carrier frequencies are combinations of fundamental tones and corresponding harmonics in the `Configuration` block. The default value is `0 Hz`.

`Ground and hide negative terminals` **— Ground RF circuit terminals**
`on` (default) | `off`

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear the parameter. By exposing these terminals, you can connect them to other parts of your model.

By default, this option is selected.

Model Examples
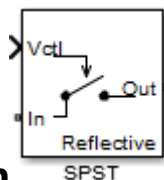
See Also
`Continuous Wave` | `Noise`

Introduced in R2010b

SPST

Single pole single throw switch

Library

Junctions

Description

The SPST block models a single pole single throw switch. The input Simulink signal, V_{ctl}, controls when the signal at the input port is transferred to the output port.

For resistance characterization:

- If the control voltage is less than the threshold voltage, the block attenuates the signal using the resistance value specified in **Off resistance**. The block places the resistor between the positive input and output terminals.
- If the control voltage is greater than or equal to the threshold voltage, the block attenuates the signal using the resistance value specified in **On resistance**.
- When the switch loading type is absorptive, and the control voltage is less than the threshold voltage, the block places internal shunt resistors at the input and output ports.

For insertion loss characterization:

- If the control voltage is less than the threshold voltage, calculate the off resistance value using the **Isolation (dB)** value.
- If the control voltage is greater than or equal to the threshold voltage, calculate the on resistance value using the **Insertion loss (dB)** value.
- When the switch loading type is absorptive, and the control voltage is less than the threshold voltage, the block places internal shunt resistors at the input and output ports.

The shunt resistor values provide matching terminations at the input and output ports.

The voltage–current relationship at the terminals depends on the relationship between control voltage, V_{ctl}, and the threshold voltage, $V_{threshold}$.

- If $V_{ctl} < V_{threshold}$

$$I_{in} = V_{inout}.G_{off} + V_{in}.G_{Z01}$$
$$I_{out} = -V_{inout}.G_{off} + V_{out}.G_{Z02}$$

- If $V_{ctl} \geq V_{threshold}$

$$I_{in} = V_{inout}.G_{off}$$
$$I_{out} = -V_{inout}.G_{on}$$

- G_{on}, G_{off} — On and off path conductances
- G_{Z01}, G_{Z02} — Shunt port conductances

Parameters

Threshold voltage

Threshold voltage of the switch, specified as a positive scalar. The default value is `0` volts.

Characterization

Characterization of the `SPST` switch: `Resistance` or `Insertion loss`. The default value is `Resistance`.

On resistance

On resistance value of the switch, specified as a positive scalar. The default value is `10` ohms. **On resistance** is available when you set **Characterization** to `Resistance`.

Off resistance

Off resistance value of the switch, specified as a positive scalar. The default value is `1e6` ohms. **Off resistance** is available when you set **Characterization** to `Resistance`.

Insertion loss (dB)

Insertion loss value of the switch, specified as a positive scalar. The default value is 1 dB. **Insertion loss (dB)** is available when you set **Characterization** to `Insertion loss`.

Isolation (dB)

Isolation value of the switch, specified as a positive scalar. The default value is 70 dB. **Isolation (dB)** is available when you set **Characterization** to `Insertion loss`

Loading type

Loading type of the SPST switch: `Absorptive` or `Reflective`. The default is `Reflective`.

Port terminations

Port termination that matches the impedance conditions of the SPST switch. The default value is 50 ohms.

Note: **Port terminations** is available when you set the SPST switch **Characterization** to `Insertion loss` or when you set **Loading type** to `Absorptive`.

Ground and hide reference terminals

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

By default, this option is selected.
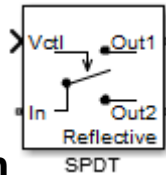
See Also
Potentiometer | Switch | SPDT |

Introduced in R2015b

SPDT

Single pole double throw switch

Library

Junctions

Description

The SPDT block models a single pole double throw switch. The input Simulink signal, V_{ctl}, controls the transfer of the signal at the input port, In, to one of the two output ports.

If the control voltage is less than the threshold voltage, the block transfers the input signal to output port 1. If control voltage is greater than or equal to the threshold voltage, the block transfers the input signal to output port 2. Two different resistors regulate the conduction path for transfer of input signal from input port to either output port.

The voltage–current relationship at the terminals depends on the relationship between control voltage, V_{ctl}, and the threshold voltage, $V_{threshold}$.

- If $V_{ctl} < V_{threshold}$:

$$I_{in} = V_{in1}.G_{on} + V_{in2}.G_{off}$$
$$I_1 = -V_{in1}.G_{on}$$
$$I_2 = -V_{in2}.G_{off} + V_2.G_{Z02}$$

- If $V_{ctl} \geq V_{threshold}$:

$$I_{in} = V_{inout}.G_{off}$$
$$I_{out} = -V_{inout}.G_{on}$$

- I_{in}, I_1 and I_2 — Currents into the input terminal and two output terminals

- V_{in1}, V_{in2} — Voltages between the input terminal and the output 1 and output 2 terminals, respectively

- V_1, V_2 — Voltages at output port 1 and output port 2, respectively

- G_{on}, G_{off} — On and off path conductances

- G_{Z01}, G_{Z02} — Shunt port conductances

Parameters

Threshold voltage

Threshold voltage of the switch, specified as a positive scalar. The default value is `0` volts.

Characterization

Characterization of the `SPDT` switch: `Resistance` or `Insertion loss`. The default value is `Resistance`.

On resistance

On resistance value of the switch, specified as a positive scalar. The default value is `10` ohms. **On resistance** is available when you set **Characterization** to `Resistance`.

Off resistance

Off resistance value of the switch, specified as a positive scalar. The default value is `1e6` ohms. **Off resistance** is available when you set **Characterization** to `Resistance`.

Insertion loss (dB)

Insertion loss value of the switch, specified as a positive scalar. The default value is `1` dB. **Insertion loss (dB)** is available when you set **Characterization** to `Insertion loss`.

Isolation (dB)

Isolation value of the switch, specified as a positive scalar. The default value is `70` dB. **Isolation (dB)** is available when you set **Characterization** to `Insertion loss`

Loading type

Loading type of the SPDT switch: `Absorptive` or `Reflective`. The default is `Reflective`.

Port terminations

Port termination that matches the impedance conditions of the SPDT switch. The default value is `50` ohms.

Note: **Port terminations** is available when you set the SPDT switch **Characterization** to `Insertion loss` or when you set **Loading type** to `Absorptive`.

Ground and hide reference terminals

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

By default, this option is selected.
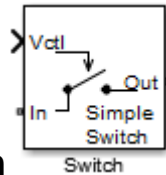
See Also
`Potentiometer` | `Switch` | `SPST` |

Introduced in R2015b

Switch

Simulink controlled two–terminal switch

Library

Junctions

Description

The switch block models a two terminal switch. The input Simulink signal, V_{ctl}, controls the transfer of the SimRF signal from the `In` terminal to the `Out` terminal.

If the control voltage is less than the threshold voltage, the block attenuates the signal using the resistance value specified in **Off resistance** between the input and output terminals. If the control voltage is greater than the threshold voltage, the block passes the signal using the resistance value specified in **On resistance** between the input and output terminals.

The voltage–current relationship for the switch, or controlled resistor:

- If $V_{ctl} < V_{thres}$:

$$I_{res}.R_{off} = V_{res}$$

- If $V_{ctl} \geq V_{thres}$:

$$I_{res}.R_{on} = V_{res}$$

R_{off} is the off resistance and R_{on} is the on resistance of the switch.

If R_{on} is less than R_{off}, and V_{ctl} is greater or equal to V_{thres}, the switch is on.

If R_{on} is greater than R_{off}, and V_{ctl} is greater or equal to V_{thres}, the switch is off.

Parameters

On resistance

> On resistance value of the switch, specified as a positive scalar. The default value is `10` ohms.

Off resistance

> Off resistance value of the switch, specified as a positive scalar. The default value is `1e6` ohms.

Threshold voltage

> Threshold voltage of the switch. The default value is `0` volts.

See Also
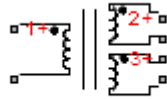`Potentiometer` | `SPST` | `SPDT` |

Introduced in R2015b

Three-Winding Transformer

Model three coupled inductors for circuit envelope analysis

Library

Elements

Three-Winding Transformer

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, "Simulate High Frequency Components".

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.

1-107

- $i_1(t)$, $i_2(t)$, 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:

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

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

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.
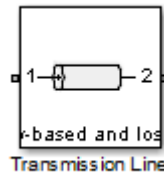

See Also

Inductor | Mutual Inductor

Transmission Line

Model transmission line

Library

Elements

Transmission Line

Description

Use the Transmission Line block to model delayed-based, lumped, and distributed transmission lines. Mask dialog options will change automatically to accommodate model type selection.

Parameters

Main Tab

Model type

- `Delay-based and lossless`

  Specify the transmission line as delay-based and lossless. When you select this option, the following parameters are available:
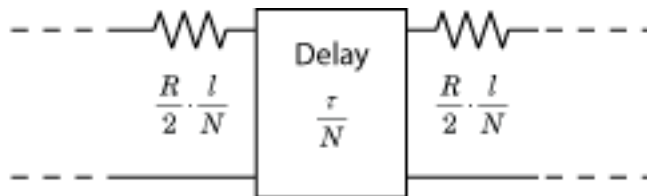
  - **Transmission delay**

    Specify the delay and select the units. The default value is `4.7e-9 s`.

  - **Characteristic impedance**

Specify the characteristic impedance of the transmission line and select the units. The default value is 50 Ohm.

- **Delay-based and lossy**

  Specify the transmission line as delay-based and lossy.

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

  - **Transmission delay**

    Specify the delay and select the units. The default value is 4.7e-9 s.

  - **Characteristic impedance**

    Specify the characteristic impedance of the transmission line and select the units. The default value is 50 Ohm.

  - **Resistance per unit length**

    Specify the resistance per unit length R and select the units. The default value is .3 Ohm/m.
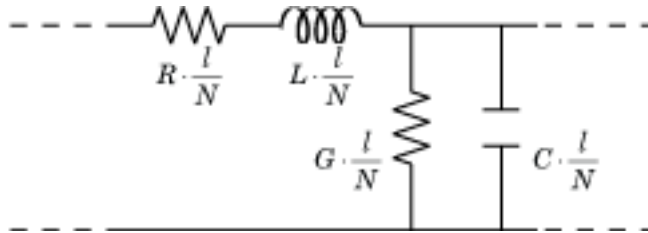
  - **Line length**

    Specify the physical length of the transmission line l and select the units. The default length is 1 cm.

  - **Number of segments**

    Specify the number of segments of the transmission line N. The default number is 10.

- **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 are available:

• **Parameterization**

 Choose a parameterization for the segments. The options are By characteristic impedance and capacitance and By inductance and capacitance.

 The default parameterization is By characteristic impedance and capacitance and the following parameters are available:

  • **Characteristic impedance**

   Specify the characteristic impedance of the transmission line and select the units. The default value is 50 Ohm.

  • **Resistance per unit length**

   Specify the resistance per unit length R and select the units. The default value is .3 Ohm/m.

  • **Capacitance per unit length**

   Specify the capacitance per unit length C and select the units. The default value is 94e-12 F/m.

  • **Conductance per unit length**

   Specify the conductance per unit length G and select the units. The default value is 5e-6 S/m.

  • **Line length**

   Specify the physical length of the transmission line l and select the units. The default length is 1 cm.

- **Number of segments**

  Specify the number of segments of the transmission line N. The default number of segments are `10`.

When you select the second option, the following parameters are available:

- **Resistance per unit length**

  Specify the resistance per unit length R and select the units. The default value is `.3 Ohm/m`.

- **Inductance per unit length**

  Specify the inductance per unit length G and select the units. The default value is `235e-9 H/m`.

- **Capacitance per unit length**

  Specify the capacitance per unit length C and select the units. The default value is `94e-12 F/m`.

- **Conductance per unit length**

  Specify the conductance per unit length G and select the units. The default value is `5e-6 S/m`.
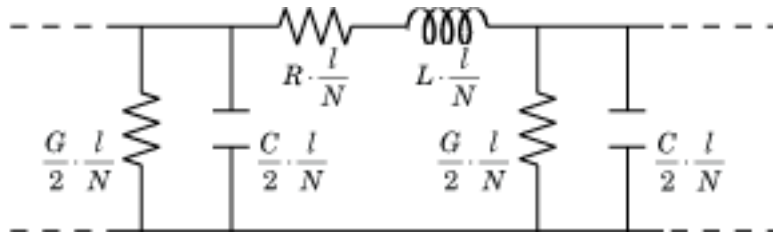
- **Line length**

  Specify the physical length of the transmission line l and select the units. The default length is `1 cm`.

- **Number of segments**

  Specify the number of segments of the transmission line N. The default number 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 are available:

- **Parameterization**

  Choose a parameterization for the segments. The options are `By characteristic impedance and capacitance` and `By inductance and capacitance`.

  The default parameterization is `By characteristic impedance and capacitance` and the following parameters are available:

  - **Characteristic impedance**

    Specify the characteristic impedance of the transmission line and select the units. The default value is `50 Ohm`.

  - **Resistance per unit length**

    Specify the resistance per unit length R and select the units. The default value is `.3 Ohm/m`.

  - **Capacitance per unit length**

    Specify the capacitance per unit length C and select the units. The default value is `94e-12 F/m`.

  - **Conductance per unit length**

    Specify the conductance per unit length G and select the units. The default value is `5e-6 S/m`.

  - **Line length**

    Specify the physical length of the transmission line l and select the units. The default length is `1 cm`.

1-114

- **Number of segments**

  Specify the number of segments of the transmission line N. The default is `10`.

When you select the second option, the following parameters are available:

- **Resistance per unit length**

  Specify the resistance per unit length R and select the units. The default value is `.3 Ohm/m`.

- **Inductance per unit length**

  Specify the inductance per unit length G and select the units. The default value is `235e-9 H/m`.

- **Capacitance per unit length**

  Specify the capacitance per unit length C and select the units. The default value is `94e-12 F/m`.

- **Conductance per unit length**

  Specify the conductance per unit length G and select the units. The default value is `5e-6 S/m`.

- **Line length**

  Specify the physical length of the transmission line l and select the units. The default length is `1 cm`.

- **Number of segments**

  Specify the number of segments of the transmission line N. The default number 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.
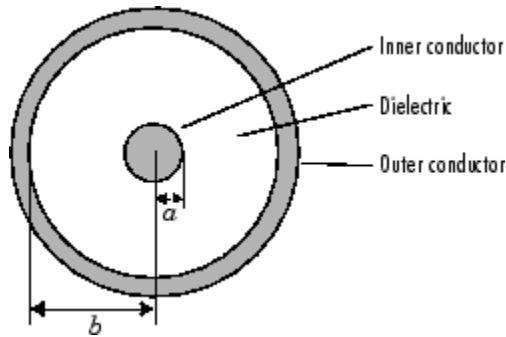
When you select this option, the following parameters are available:

- **Outer radius**

  Specify the radius of the outer conductor of the coaxial transmission line. The default value is `2.57 mm`.

- **Inner radius**

  Specify the radius of the inner conductor of the coaxial transmission line. The default value is `0.725 mm`.

- **Relative permeability constant**

  Specify the 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 is `1`.

- **Relative permittivity constant**

  Specify the 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 is `2.2`.

- **Loss tangent of dielectric**

  Specify the loss angle tangent of the dielectric. The default value is `0`.

- **Line length**

  Specify the physical length of the transmission line. The default value is `1 cm`.

- **Conductivity of conductor**

  Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

- **Stub mode**

  Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

  The following auxiliary equations are used for ABCD-parameter calculations. For general information, see .

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

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

where

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

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

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

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

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.

1-117

- 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 internal modeling frequencies.
- 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 $k = a_a + i\beta$, where a_a is the attenuation coefficient and β is the wave number. The attenuation coefficient a_a is related to the specified loss, a, 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*.

- **`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.
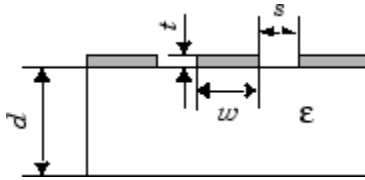
When you select this option, the following parameters are available:

- **Conductor width**

  Specify the physical width of the conductor. The default value is `0.6` mm.

- **Slot width**

  Specify the physical width of the slot. The default value is `0.2` mm.

- **Substrate height**

  Specify the thickness of the dielectric on which the conductor resides. The default is `0.635` mm.

- **Strip thickness**

  Specify the physical thickness of the conductor. The default value is `5` mm.

- **Relative permittivity constant**

  Specify the 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 is `2.2`.

- **Loss tangent of dielectric**

  Specify the loss angle tangent of the dielectric. The default value is `1`.

- **Line length**

  Specify the physical length of the transmission line. The default value is `1` cm.

- **Conductivity of conductor**

  Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

- **Stub mode**

Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

- **`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.
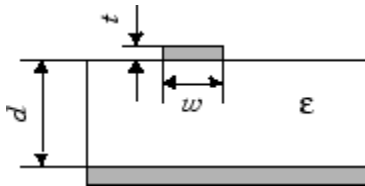
  When you select this option, the following parameters are available:

  - **Strip width**

    Specify the width of the microstrip transmission line. The default value is `0.6` mm.

  - **Substrate height**

    Specify the thickness of the dielectric on which the microstrip resides. The default value is `0.635` mm.

  - **Strip thickness**

    Specify the physical thickness of the microstrip. The default value is `5` mm.

  - **Relative permittivity constant**

    Specify the 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 is `2.2`.

  - **Loss tangent of dielectric**

    Specify the loss angle tangent of the dielectric. The default value is `0`.

  - **Line length**

Specify the physical length of the transmission line. The default value is `1 cm`.

- **Conductivity of conductor**

  Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

- **Stub mode**

  Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

- **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.
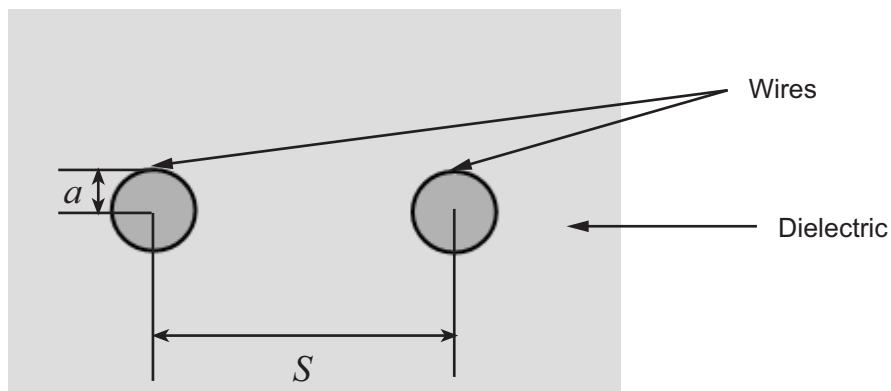
When you select this option, the following parameters are available:

- **Wire radius**

  Specify the radius of the conducting wires of the two-wire transmission line. The default value is `0.67 mm`.

- **Wire separation**

  Specify the physical distance between the wires. The default value is `1.62 mm`.

- **Relative permeability constant**

  Specify the 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 is `1`.

- **Relative permittivity constant**

  Specify the 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 is `2.2`.

- **Loss tangent of dielectric**

  Specify the loss angle tangent of the dielectric. The default value is `0`.

- **Line length**

  Specify the physical length of the transmission line. The default value is `1 cm`.

- **Conductivity of conductor**

  Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

- **Stub mode**

  Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

- **`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]
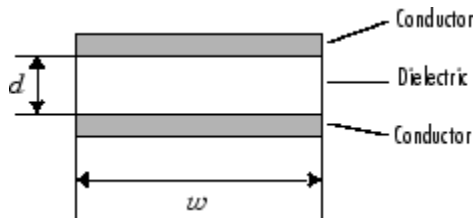
When you select this option, the following parameters are available:

- **Plate width**

  Specify the physical width of the parallel-plate transmission line. The default value is 5 mm.

- **Plate separation**

  Specify the thickness of the dielectric separating the plates. The default value is 1 mm.

- **Relative permeability constant**

  Specify the 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 is 1.

- **Relative permittivity constant**

  Specify the 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 is 2.2.

- **Loss tangent of dielectric**

  Specify the loss angle tangent of the dielectric. The default value is 0.

- **Line length**

  Specify the physical length of the transmission line. The default value is 1 cm.

- **Conductivity of conductor**

  Specify the conductivity of the conductor. The default value is inf S/m.

- **Stub mode**

Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

- **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 parameters are available:

  - **Phase velocity (m/s)**

    Specify the propagation velocity of a uniform plane wave on the transmission line. The default value is `299792458`.

  - **Characteristic impedance**

    Specify the characteristic impedance of the transmission line. The value can be complex. The default value is `50 Ohm`.

  - **Loss (dB/m)**

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

  - **Line length**

    Specify the physical length of the transmission line. The default value is `1 cm`.

  - **Frequency**

    Specify the vector of modeling frequencies. The block performs the calculations listed in the Description section at each frequency you provide. The default value 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 is `Linear`.

  - **Stub mode**

Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

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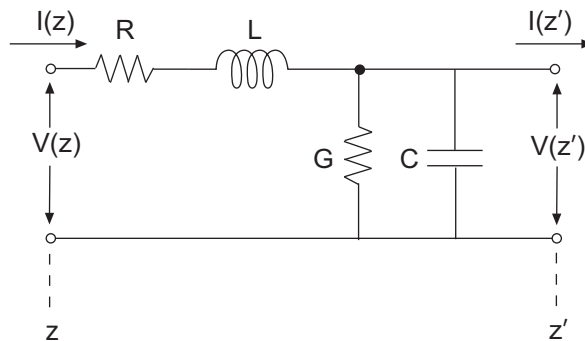
where $z' = z + \Delta z$.

When you select this option, the following parameters are available:

- **Resistance per unit length**

  Specify the vector of resistance values in ohms per meter. The default value is `0.3 Ohm/m`.

- **Inductance per unit length**

  Specify the vector of inductance values in henries per meter. The default value is `235e-9 H/m`.

- **Capacitance per unit length**

  Specify the vector of capacitance values in farads per meter. The default value is `94e-12 F/m`.

- **Conductance per unit length**

1-125

Specify the vector of conductance values in siemens per meter. The default value is `5e-6 S/m`.

- **Line length**

  Specify the physical length of the transmission line. The default value is `1 cm`.

- **Frequency**

  Specify the vector of frequency values at which the resistance, inductance, capacitance, and conductance values are known. The default value 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 is `Linear`.

- **Stub mode**

  Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See for more information.

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.

Parameter Calculations for Distributed Transmission Line

When modeling distributed transmission lines, the block first calculates ABCD-parameters at a set of internal frequencies. The ABCD-parameters are converted S-parameters for simulation.

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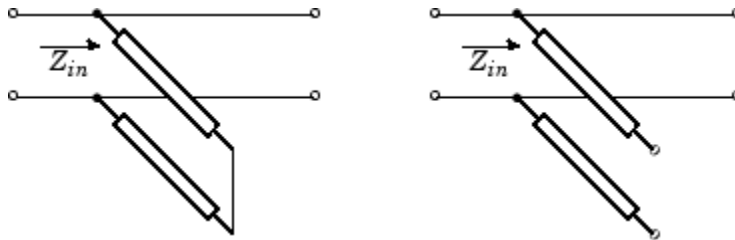
$$A = \frac{e^{kd} + e^{-kd}}{2}$$

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

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

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

Transmission Line with Stub

When you set the **Stub mode** parameter in the mask dialog box to Shunt, the two-port network consists of a transmission line in series with a stub. You can terminate the stub with 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
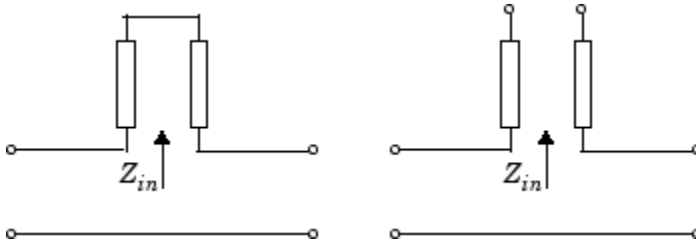
$$A = 1$$
$$B = 0$$
$$C = 1 / Z_{in}$$
$$D = 1$$

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:

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

Modeling Tab

Modeling options tab is activated for all transmission line options except `Delay-based and lossless`, `Delay-based and lossy`, `Lumped parameter L-section`, and `Lumped parameter pi-section`.

Modeling options

SimRF provides two different ways to model S-parameters:

- Time-domain (rationalfit) technique creates an analytical rational model that approximates the whole range of the data.
- Frequency-domain computes the baseband impulse response for each carrier frequency independently. This technique is based on convolution. There is an option to specify the duration of the impulse response. For more information, see "Compare Time and Frequency Domain Simulation Options for S-parameters".
- 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`.

Time domain
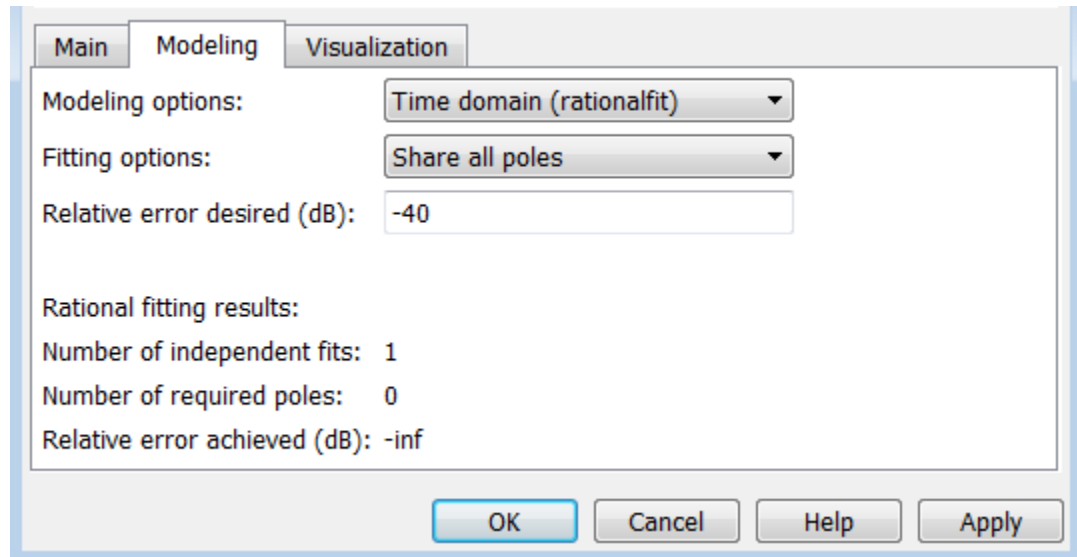
- **Fitting options**

  The fitting options are `Share all poles`, `Share poles by columns`, or `Fit individually`.

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

- **Relative error desired (dB)**

  Enter the desired relative error in decibels (dB). The default value is `-40`.

- **Rational fitting results**

  Shows values of **Number of independent fits**, **Number of required poles**, and **Relative error achieved (dB)**.

- When modeling using `Time domain`, the **Plot** in `Visualization` tab plots the data defined in `Data Source` and the values in the `rationalfit` function.

Frequency domain

- **Automatically estimate impulse response duration**

Select **Automatically estimate impulse response duration** to calculate impulse response duration automatically. Clear the selection to specify impulse response duration.

- When using `Frequency domain`, the **Plot** in `Visualization` tab plots the data defined in the `Data Source`.

Visualization Tab

Source of frequency data

The only option for **Source of frequency data** is `User-specified`. To plot, specify a vector of frequencies in the **Frequency data** parameter and select units.

Plot type

Specify the type of plot that you want to produce with your data. When you model using `Frequency domain`, Visualization tab plots only the data defined in `Data Source`. When you model using `Time domain`, Visualization tab plots the data defined in `Data Source` and the `rationalfit` values. 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 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`.

Tip

In general, blocks that model delay effects rely on signal history. You can minimize numerical error that occur due to 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.

Examples

The example, "Transmission Lines, Delay-based and Lumped Models", shows how to use `Delay-based` and `Lumped` Transmission Line blocks.

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.
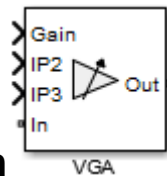
See Also

S-Parameters

VGA

Variable gain amplifier

Library

Elements

Description

The VGA block models a SimRF variable gain amplifier. The Simulink signal controls the non-linear gain. The VGA output voltage is a function of the input voltage:

$$V_{out} = g * V_{in} + c_2 * V_{in}^2 + c_3 * V_{in}^2$$

The table shows the formulas for the coefficients, g, c_2, and c_3

Coefficient Formulae

	Input Referred	Output Referred
g	$g = \dfrac{2\sqrt{G * Z_{out} * Z_{in}}}{|Z_{in}|}$	
c_2	$C_2 = \dfrac{g}{\sqrt{2 * IP2 * Z_{in}}} volts^{-1}$	$C_2 = \dfrac{g^2}{\sqrt{8 * IP2 * Z_{out}}} volts^{-1}$
c_3	$C_3 = \dfrac{4}{3} \cdot \dfrac{g}{2 * IP3 * Z_{in}} volts^{-2}$	$C_2 = \dfrac{4}{3} \cdot \dfrac{g^3}{8 * IP3 * Z_{out}} volts^{-2}$

The three Simulink input ports are `Gain`, `IP2` (second-order intercept point), and `IP3` (third-order intercept point). `In` is the SimRF voltage input and `OUT` is SimRF voltage output.

Parameters

Input impedance (Ohm)

Input impedance of the amplifier, specified as a scalar. The default value is `50` ohms.

Output impedance (Ohm)

Output impedance of the amplifier, specified as a scalar. The default value is `50` ohms.

Intercept points convention

Input-referred or output-referred convention for IP2 or IP3 intercept points, specified as `Input` or `Output`.

The default is `Output`.

Ground and hide negative terminals

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

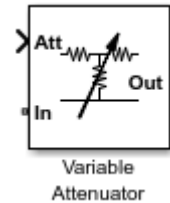By default, this option is selected.

See Also
`Amplifier`

Introduced in R2015b

Variable Attenuator

Model variable attenuator
Library: SimRF / Circuit Envelope / Elements

Description

The Variable Attenuator block attenuates the signal power by a given factor known as *Insertion Loss* in dB. Using the Variable Attenuator block, you can vary the attenuation of the input Simulink signal during simulation. Commonly, the block matches the impedance of the RF circuit at the input and output ports. You can use attenuators to dampen the power of the incoming signal to protect RF circuits.

Parameters

`Minimum attenuation (dB)` — Lowest value for insertion loss or attenuation
`0.001` (default) | scalar

Lowest value of insertion loss or attenuation to apply to the signal, specified as a scalar in dB.

`Maximum attenuation (dB)` — Highest value of insertion loss or attenuation
`1000` (default) | scalar

Highest value of insertion loss or attenuation to apply to the signal, specified as a scalar in dB.

`Input impedance (Ohm)` — Input impedance
`50` (default) | scalar

Input impedance of the attenuator, specified as a scalar in ohms.

`Output impedance (Ohm)` — Output impedance
50 (default) | scalar

Output impedance of the attenuator, specified as a scalar in ohms.

`Simulate noise` — Simulate thermal noise
on (default) | off

Select this parameter to simulate thermal noise in the attenuator. You must select **Simulate noise** in the `Configuration` block.

This parameter inserts a current noise source with the single-sided power density of 4 $\mathbf{k}T/R$ A^2/Hz, where:

- T is the value of the **Temperature** parameter in the `Configuration` block. Units are in degrees Kelvin.
- $\mathbf{k}$ is the Boltzmann constant.

`Ground and hide negative terminals` — Ground RF circuit terminals
on (default) | off

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear this parameter. By exposing these terminals, you can connect them to other parts of your model.
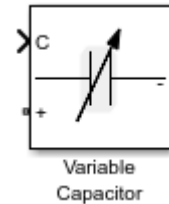
Model Examples

See Also
`Attenuator` | `Variable Inductor` | `Variable Capacitor` | `Variable Phase Shift`

Introduced in R2016b

Variable Capacitor

Model variable capacitor
Library: SimRF / Circuit Envelope / Elements

Description

The Variable Capacitor block controls the output of SimRF feedback circuits using Simulink controlled capacitance in farads. The minimum value of the capacitance (*Cmin*) is a SimRF defined constant independent of the Simulink control signal. The block has two electrical terminals. One terminal is for the Simulink control signal and one terminal is for the SimRF signal.

Parameters

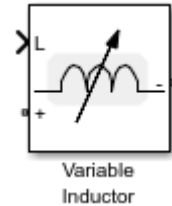The Variable Capacitor block has no parameters.

See Also
`Capacitor` | `Variable Inductor`

Introduced in R2016b

Variable Inductor

Model variable inductor
Library: SimRF / Circuit Envelope / Elements

Variable
Inductor

Description

The Variable Inductor block controls the output of SimRF feedback circuits using Simulink controlled inductance in henries. The minimum value of the inductance (*Lmin*) is a SimRF defined constant independent of the Simulink control signal. The block has two electrical terminals. One terminal is for the Simulink control signal and one terminal is for the SimRF signal.

Parameters

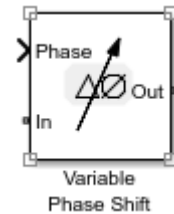The Variable Inductor block has no parameters.

See Also

Inductor | Variable Capacitor

Introduced in R2016b

Variable Phase Shift

Model variable phase device
Library: SimRF / Circuit Envelope / Elements

Description

The Variable Phase Shift block controls the phase of the output signal of a SimRF circuit. The block uses a Simulink signal to control the phase of the circuit. One input terminal is for the Simulink control signal and one input terminal is for the SimRF signal.

Phase shifter are commonly used in phased array antenna systems. In these systems, electronically controlled phase shifters steer the antenna beam in space. Phase shifters are also used in test and measurement systems.

Parameters

Phase shift unit — Angle units for controlling phase of circuit
rad (default) | deg

Angle units for controlling the phase of the signal, specified as rad for radians or deg for degrees.

Reference impedances (Ohm) — Reference impedance for phase shifter
50 (default) | vector of positive scalars

Reference impedance for phase shifter, specified as a vector of positive scalars.

Ground and hide negative terminals — Ground RF circuit terminals
on (default) | off

Select this parameter to ground and hide the negative terminals. To expose the negative terminals, clear this parameter. By exposing these terminals, you can connect them to other parts of your model.
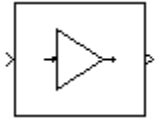
Model Examples

See Also
`PhaseShift` | `Variable Inductor` | `Variable Attenuator` | `Variable Capacitor`

Introduced in R2016b

Blocks — Alphabetical List

Amplifier (Idealized Baseband)

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.
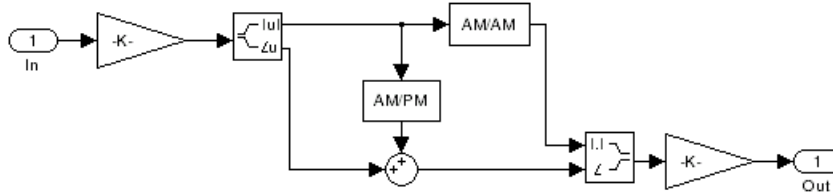
Application of Nonlinearity

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
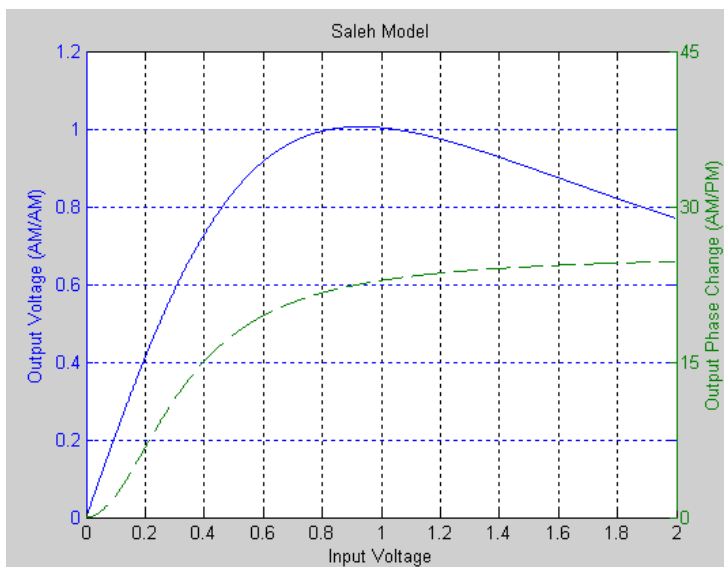
Model Parameters and Characteristics of Nonlinearity Modeling Methods

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:

-
-

-
-
-

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:

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

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:

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

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

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

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,
  Noise figure = 10log(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.

Parameters

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.
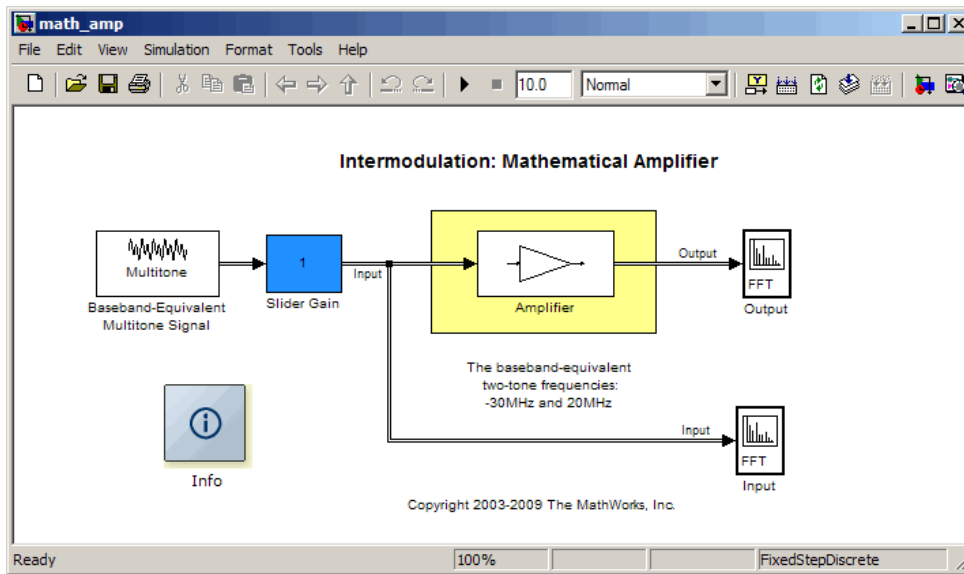
This example uses a baseband-equivalent multitone signal as input to the Amplifier block. A Simulink Slider Gain block enables you to vary the gain from 1 to 10. The following figure shows the input signal with gain set to the default 1.
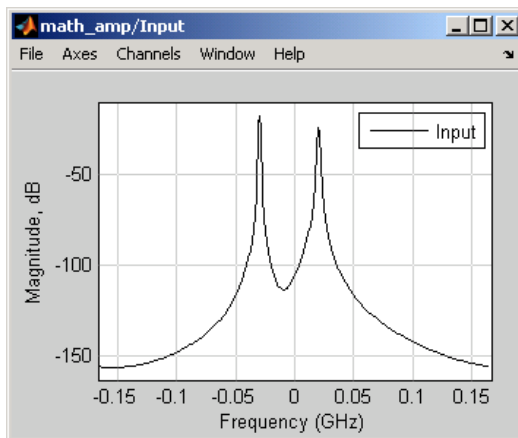
The next figure shows the same signal after it passes through the Amplifier block, with the **Method** parameter set to `Hyperbolic tangent`. The example uses the default

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.
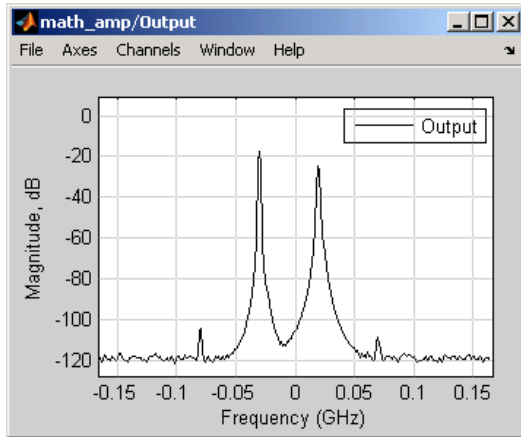

References

[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

Standard bandpass RF filters in baseband-equivalent complex form
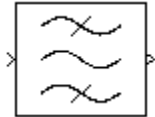
Library

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.

Parameters

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)

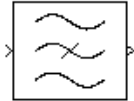Time interval between consecutive samples of the input signal.

See Also

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

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

Bandstop RF Filter

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.

Parameters

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)

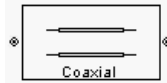Time interval between consecutive samples of the input signal.

See Also

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

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

Coaxial Transmission Line

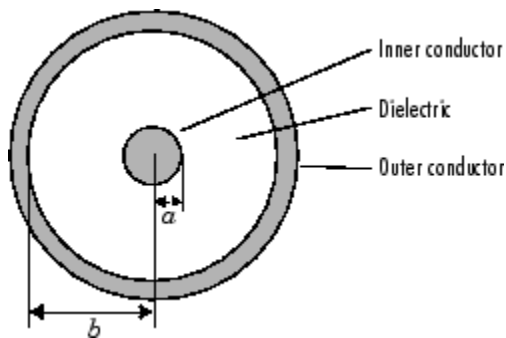Model coaxial transmission line

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.

2-21

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, determined by the Output Port block. 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.
- o_{cond} is the conductivity in the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$ where:
  - μ_0 is the permeability in free space.
  - μ_r is the **Relative permeability constant** parameter value.
- The is a complex dielectric constant given by $\varepsilon = \varepsilon' - j\varepsilon'' = \varepsilon' (1 - jtan\delta)$
- ε' is the real part of complex dielectric constant ε, $\varepsilon' = \varepsilon_0\varepsilon_r$. ε'' is the imaginary part of complex dielectric constant 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 $1 / \sqrt{\pi f \mu \sigma_{cond}}$.

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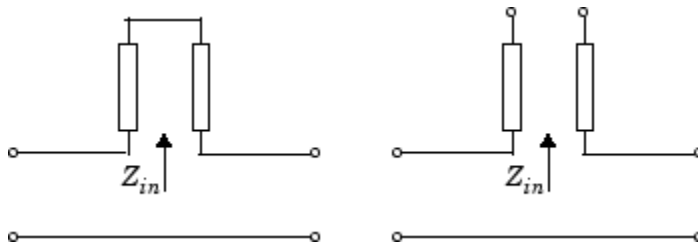
Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

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

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

Parameters

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`

Connection Port

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
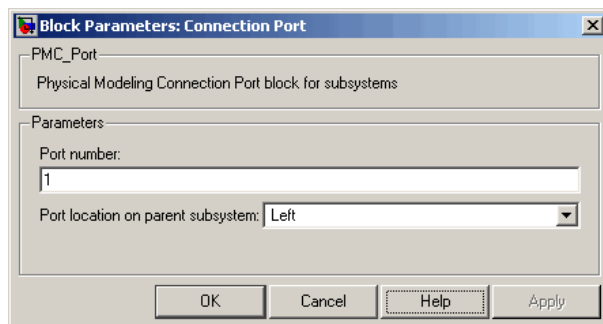
Port number

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.

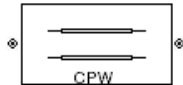**Port location on parent subsystem**

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.

Coplanar Waveguide Transmission Line
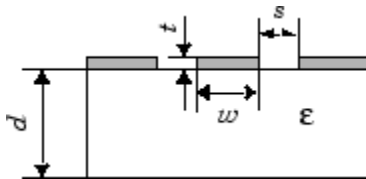
Model coplanar waveguide transmission line

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:

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

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

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

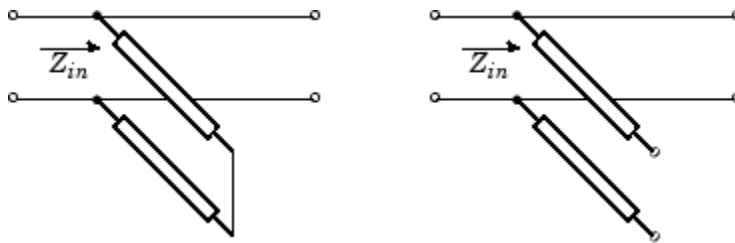$$D = \frac{e^{kd} + e^{-kd}}{2}$$

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.
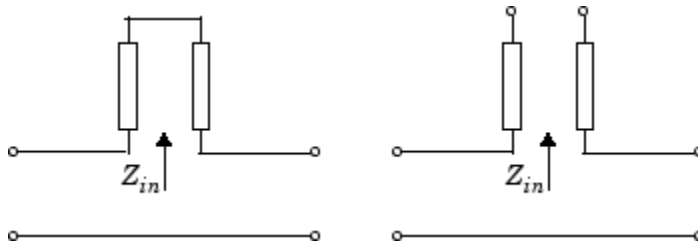

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

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

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

Parameters

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

General Amplifier

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,

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

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 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:

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

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

$$0 = \frac{c_1}{IIP3} + c_3$$

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 data source.
- Spot noise data in the block dialog box.
- Spot noise data (`rfdata.noise class`) 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)
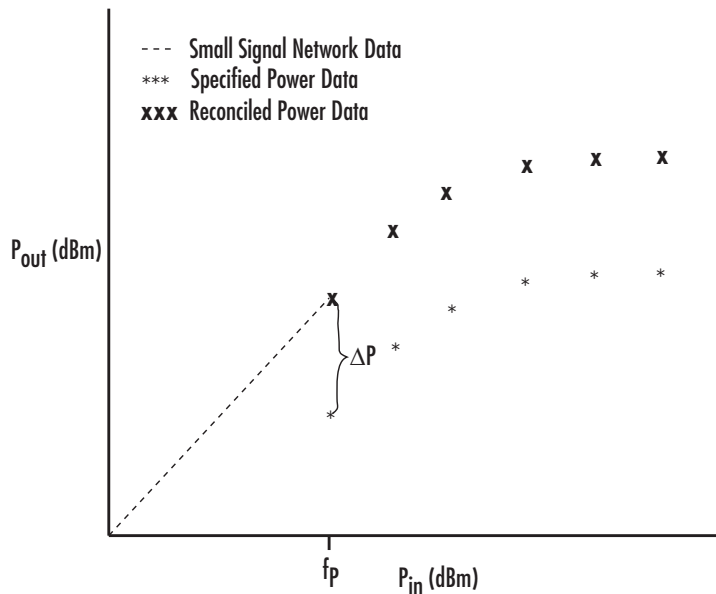
Parameters

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.
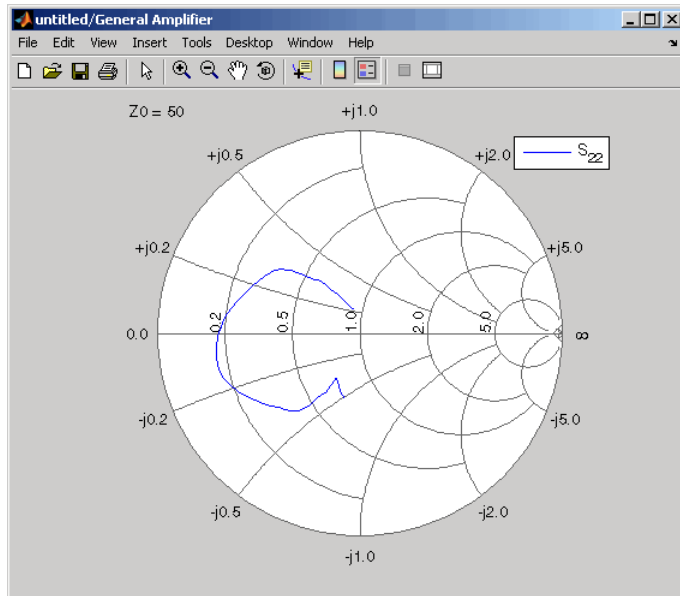
1 On the **Main** tab, accept the default settings.
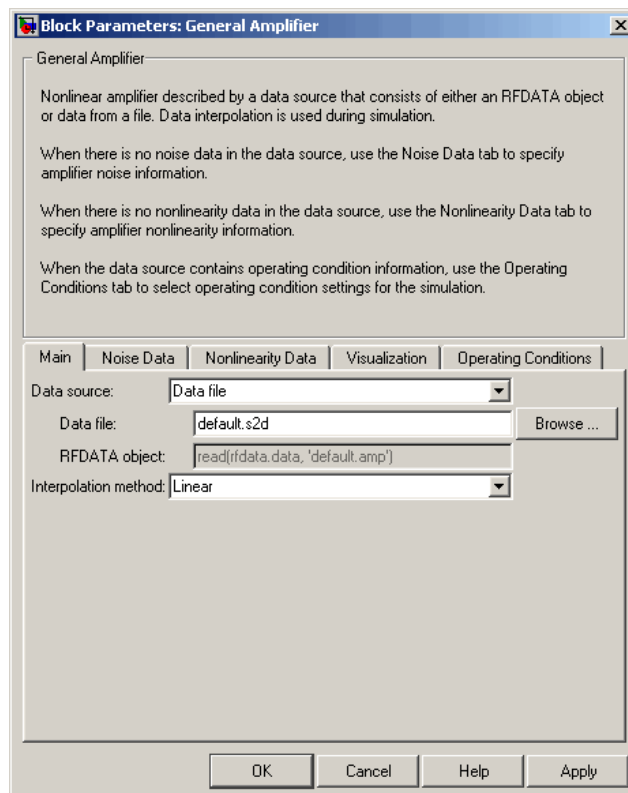
2 On the **Visualization** tab, set the parameters as follows:

- In the **Plot type** list, select `Z Smith chart`.
- In the **Y parameter1** list, select `S22`.
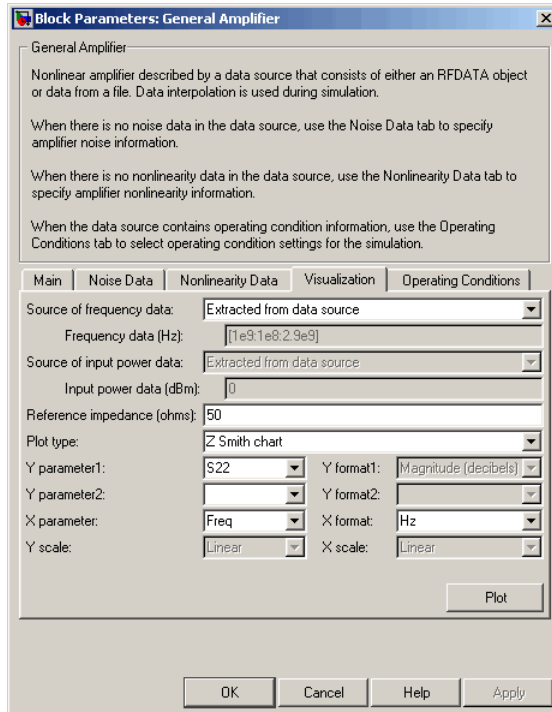
Click **Plot**. This action creates Z Smith chart of the S_{22} parameters using the frequencies taken from the data source.

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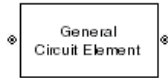
See Also

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

`rfdata.data` (RF Toolbox)

`interp1` (MATLAB)

General Circuit Element

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.

Parameters

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

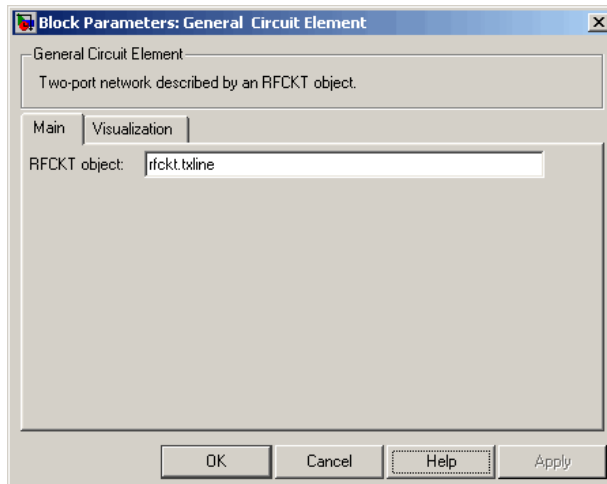For information about plotting, see "Create Plots".

Examples

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

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

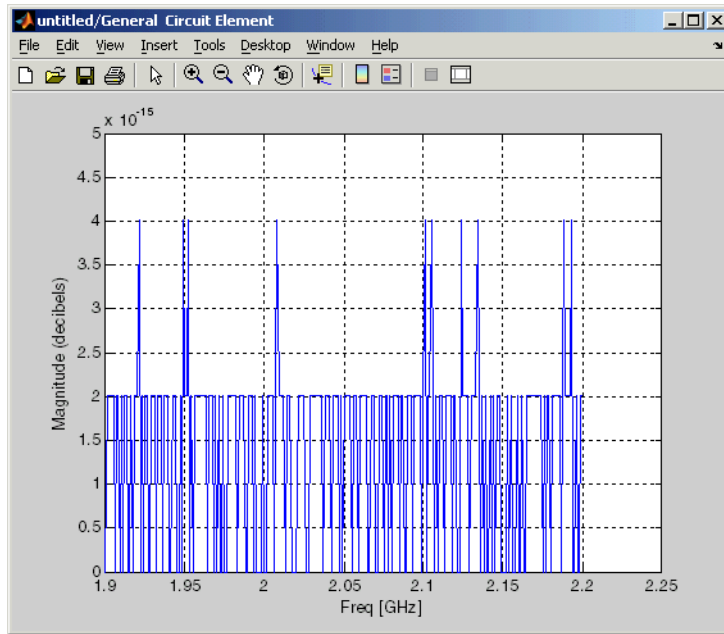1   On the **Main** tab, set the **RFCKT object** parameter to rfckt.txline.

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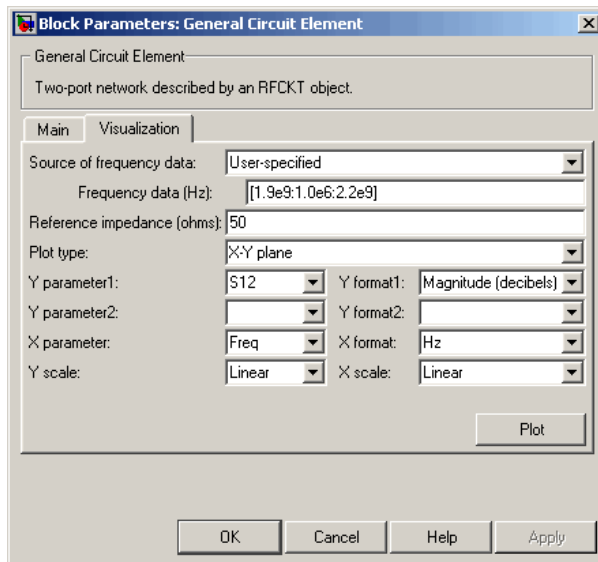
Click **Plot**. This action creates an X-Y Plane plot of the S_{12} parameters in the frequency range 1.9 to 2.2 GHz.
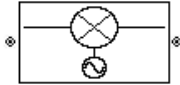
See Also

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

interp1 (MATLAB)

General Mixer

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 class`) 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.

Parameters

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 class` 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)

General Passive Network

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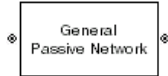
Library

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.

Parameters

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.
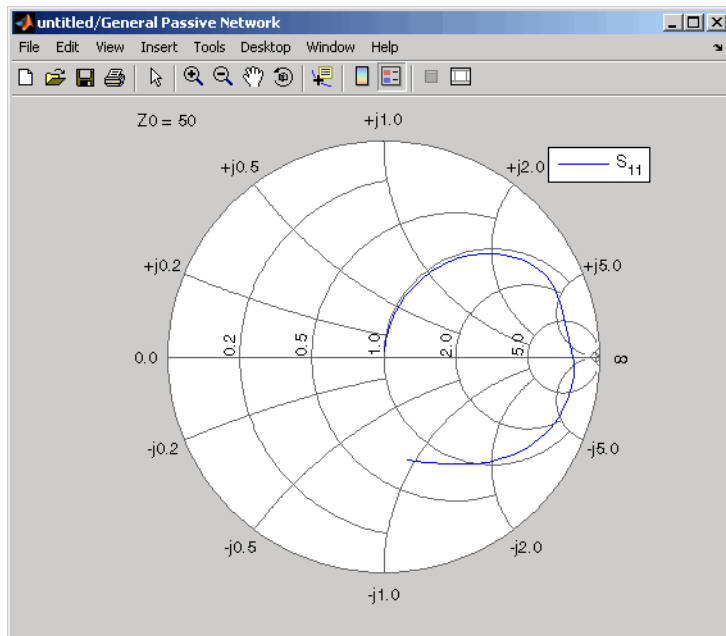
1   On the **Main** tab, accept the default settings.
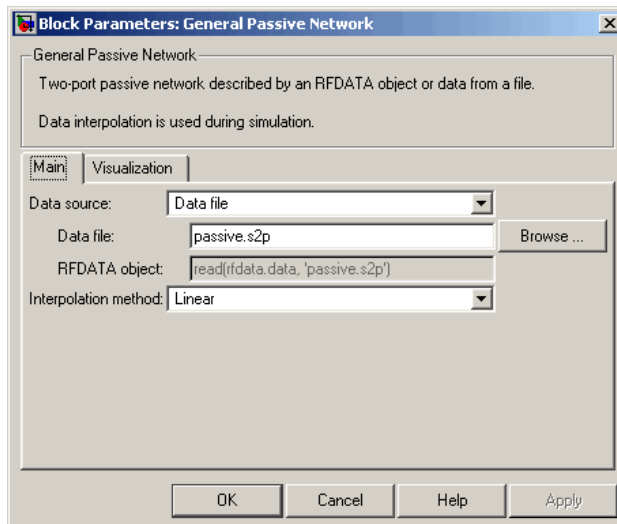
2   On the **Visualization** tab, set the parameters as follows:

- In the **Plot type** list, select `Z Smith chart`.
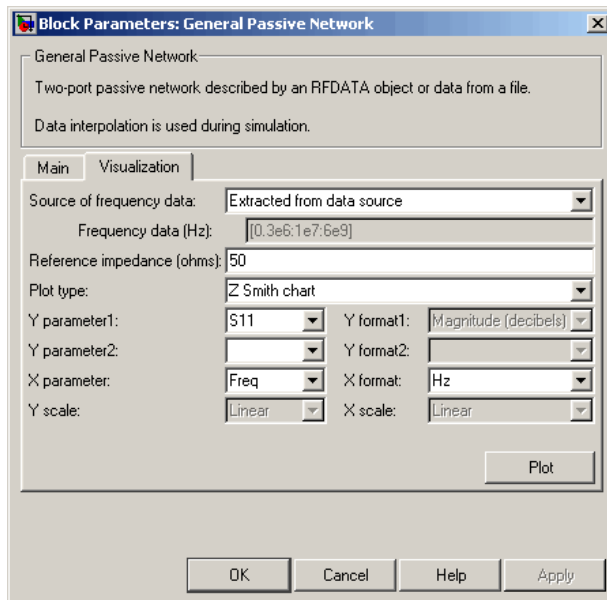
Click **Plot**. This action creates a Z Smith chart of the S_{11} parameters, using the frequencies taken from the **RFDATA object** parameter on the **Main** tab.
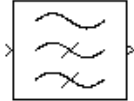
See Also

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

rfdata.data (RF Toolbox)

interp1 (MATLAB)

Highpass RF Filter

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.

Parameters

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)

Input Port

Connection block from Simulink environment to RF physical blocks
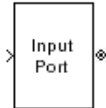
Library

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.

Parameters

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.
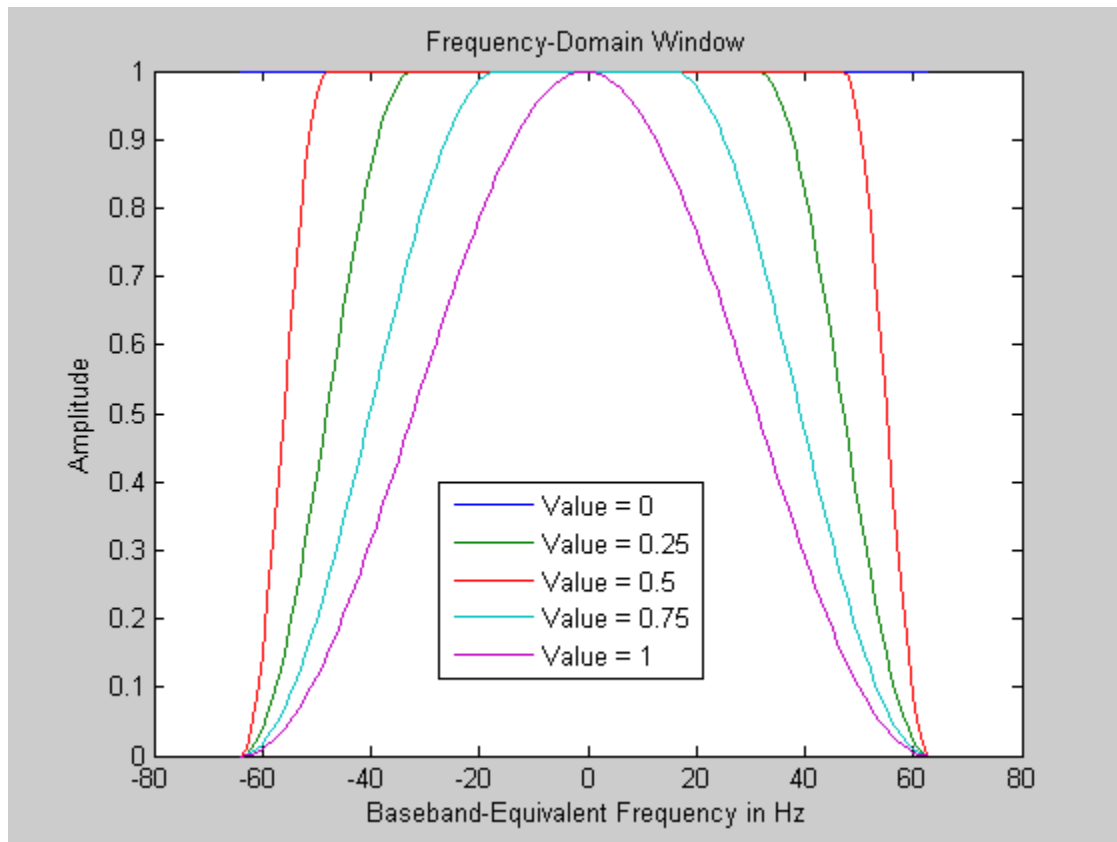
See "Create a Complex Baseband-Equivalent Model" for information about how the Input Port block applies this window to reduce the Gibbs phenomenon (also known as ringing), and other artifacts in the baseband-equivalent model of the system.

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`

LC Bandpass Pi

Model LC bandpass pi network
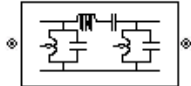
Library

Ladder Filters sublibrary of the Physical library
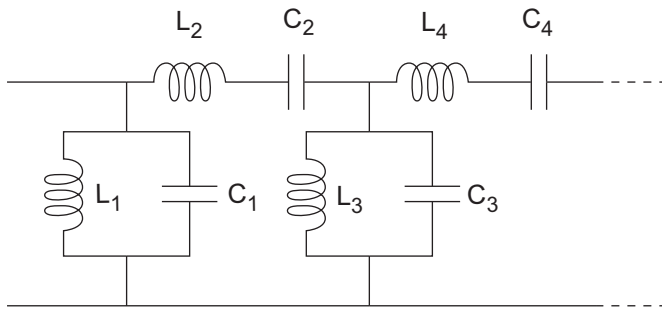
Description

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.

Parameters

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.
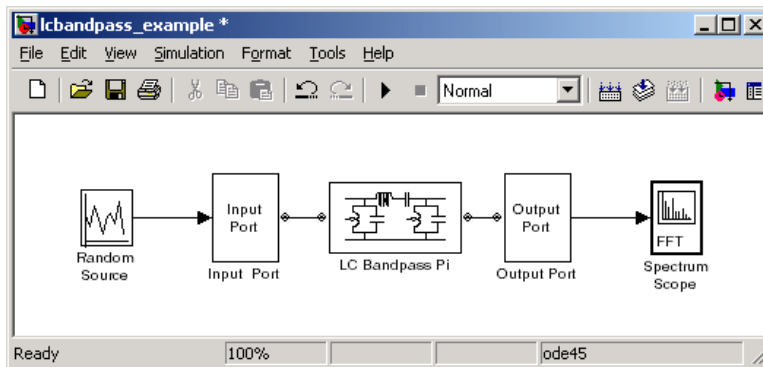

The DSP System Toolbox Random Source block produces frame-based output at 512 samples per frame. Its **Sample time** parameter is set to 1.0e-9. This sample time must match the sample time for the physical part of the model, which you provide in the Input Port block diagram.

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.

Block Parameters: Input Port

Input Port

Connection block from Simulink to RF Blockset physical blocks.

The RF Blockset physical blocks use a baseband-equivalent modeling technique. This technique models a bandwidth of 1/(Sample time), centered at the specified Center frequency parameter value. This frequency value corresponds to 0 Hz in the baseband-equivalent model.

The block provides the option to interpret the Simulink signal as either the incident power wave to the RF system or the source voltage of the RF system. The 'Incident power wave' option is the most common RF modeling interpretation, while the 'Source voltage' option is provided for backwards compatibility. If the input Simulink signal is the incident power wave, the output of the RF system is the transmitted power wave. If the input is the source voltage, the output is the load voltage.

The block controls the modeling of RF Blockset physical blocks between this block and the Output Port block using:
- FIR filters to model the frequency-dependent characteristics
- Look-up tables to model the nonlinear behaviors

Optional guard bands can be specified as a fraction of the modeling bandwidth. The guards bands are implemented by applying a Tukey window to the frequency response. Modeling delay may be added to improve the response of the FIR filters.

Parameters

Treat input Simulink signal as:	Source voltage
Source impedance (ohms):	50
Finite impulse response filter length:	256
Fractional bandwidth of guard bands:	0
Modeling delay (samples):	0
Center frequency (Hz):	700e6
Sample time (s):	1e-9

☑ Add noise

Initial seed:	67987

[OK]　[Cancel]　[Help]　[Apply]

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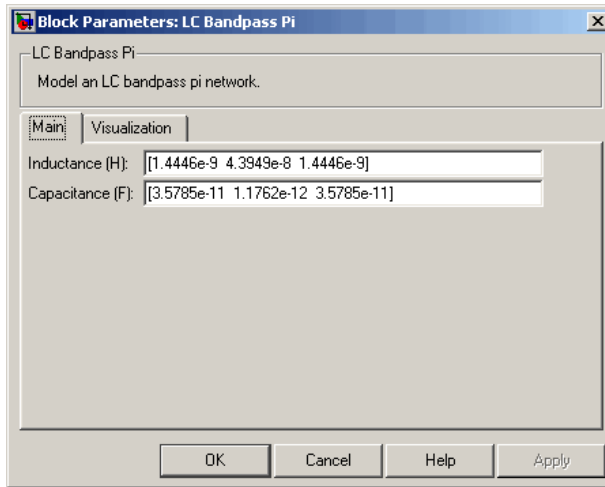
The following plot shows a sample of the baseband-equivalent RF signal generated by this LC Bandpass Pi block. Zero (0) on the frequency axis corresponds to the center frequency specified in the Input Port block. The bandwidth of the frequency spectrum is 1/sample time. You specify the **Sample time**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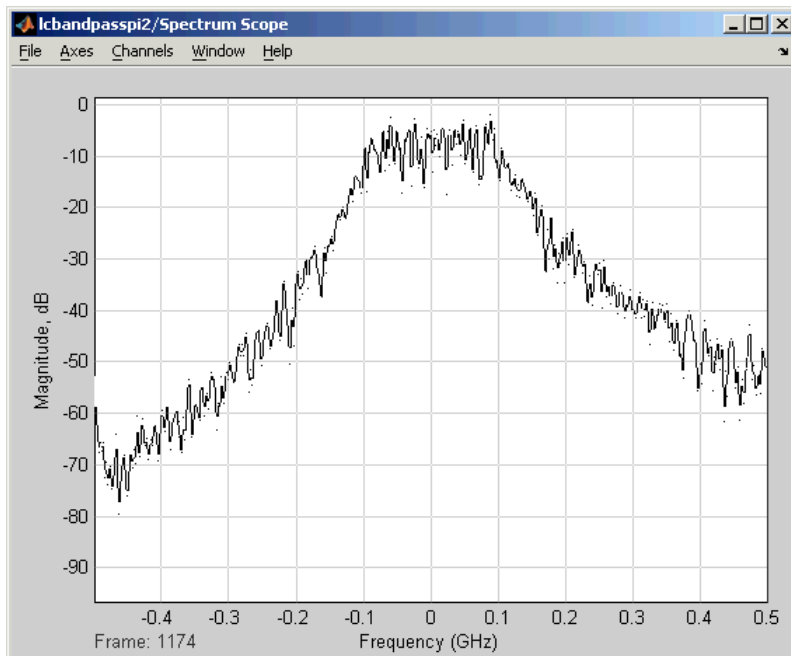
References

[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

Model LC bandpass tee network
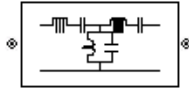
Library

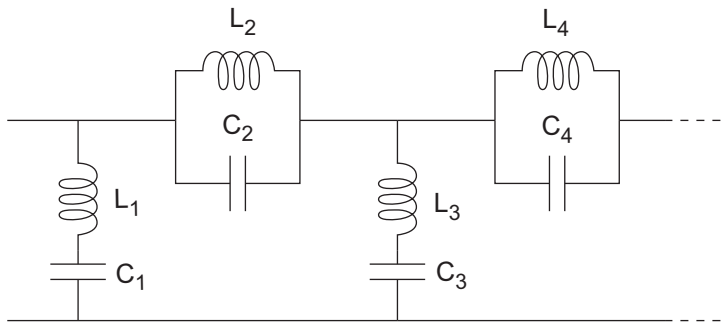Ladder Filters sublibrary of the Physical library

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.

Parameters

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

LC Bandstop Pi

Model LC bandstop pi network
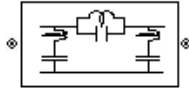
Library

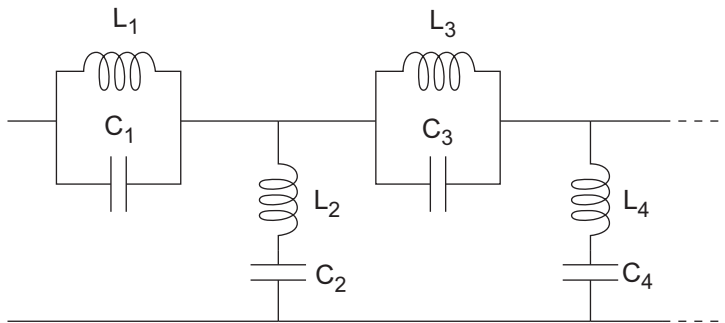Ladder Filters sublibrary of the Physical library

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.

Parameters

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

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.

Parameters

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

LC Highpass Pi

Model LC highpass pi network
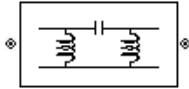
Library

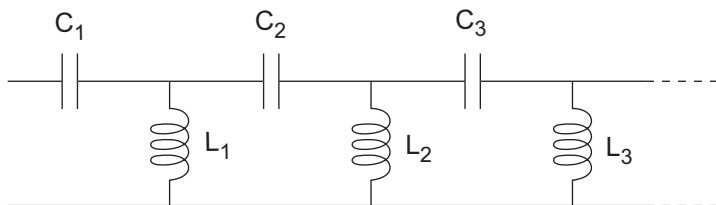Ladder Filters sublibrary of the Physical library

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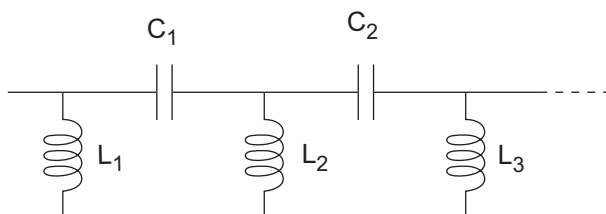
$[L_1, L_2, L_3, ...]$ is the value of the `'L'` property, and $[C_1, C_2, ...]$ is the value of the `'C'` property.

Parameters

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

Model LC highpass tee network
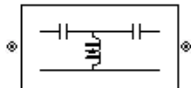
Library

Ladder Filters sublibrary of the Physical library

Description

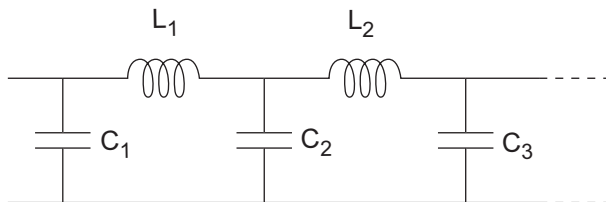The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

Parameters

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

LC Lowpass Pi

Model LC lowpass pi network
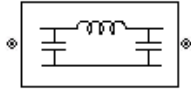
Library

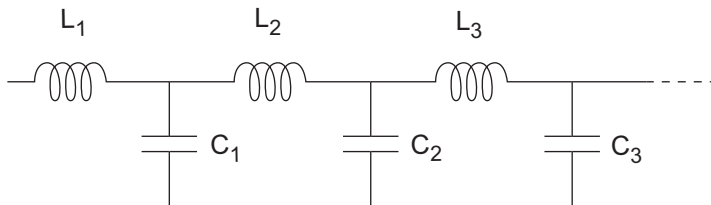Ladder Filters sublibrary of the Physical library

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.

Parameters

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

Model LC lowpass tee network
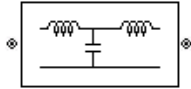
Library

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.

Parameters

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
```

Lowpass RF Filter

Standard lowpass RF filters in baseband-equivalent complex form
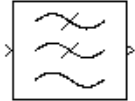
Library

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.

Parameters

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

Parameters that are tunable can be changed while the model is running.

Design method

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

Filter order

Order of the filter.

Passband edge frequency (Hz)

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

Stopband edge frequency (Hz)

Stopband edge frequency for Chebyshev II designs. Tunable.

Passband ripple in dB

Passband ripple for Chebyshev I and elliptic designs. Tunable.

Stopband attenuation in dB

Stopband attenuation for Chebyshev II and elliptic designs. Tunable.

Finite impulse response filter length

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

Center frequency (Hz)

Center of the modeling frequencies.

Sample time (s)

Time interval between consecutive samples of the input signal.

See Also

`Amplifier`, `Bandpass RF Filter`, `Bandstop RF Filter`, `Highpass RF Filter`, `Mixer`

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

Microstrip Transmission Line

Model microstrip transmission line
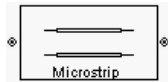
Library

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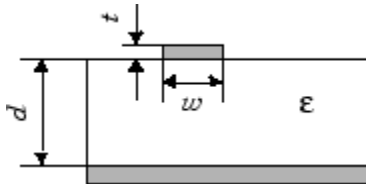
The block lets you model the transmission line as a stub or as a stubless line.

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

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

Parameters

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`

Mixer (Idealized Baseband)

Complex baseband model of mixer and local oscillator with phase noise
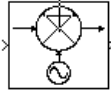
Library

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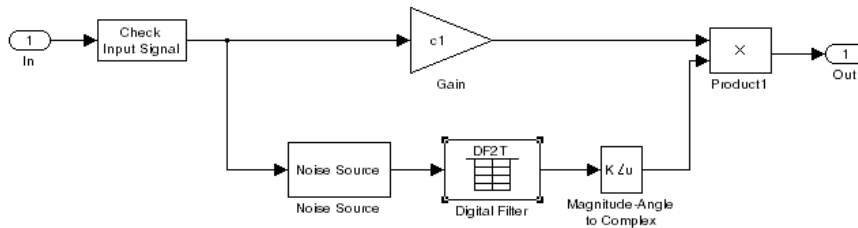
You can view the construction of the Noise Source subsystem by double-clicking it.

Parameters

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

Connection block from RF physical blocks to Simulink environment
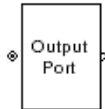
Library

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

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

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,

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

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:

Passband Spectrum of a Modulated RF Carrier

N is the number of sub-bands

Input Port Block Parameters

Finite impulse response filter length: N

Fractional bandwidth of guard bands: F

Modeling delay (samples): D

Center frequency (Hz): fc

Sample time (s): ts

Source impedance (ohms): 50

$\Delta f = 1/(t_s*N)$

Magnitude

f_c

f_{min} f_n f_{max} Frequency

Bandwidth = $1/t_s$
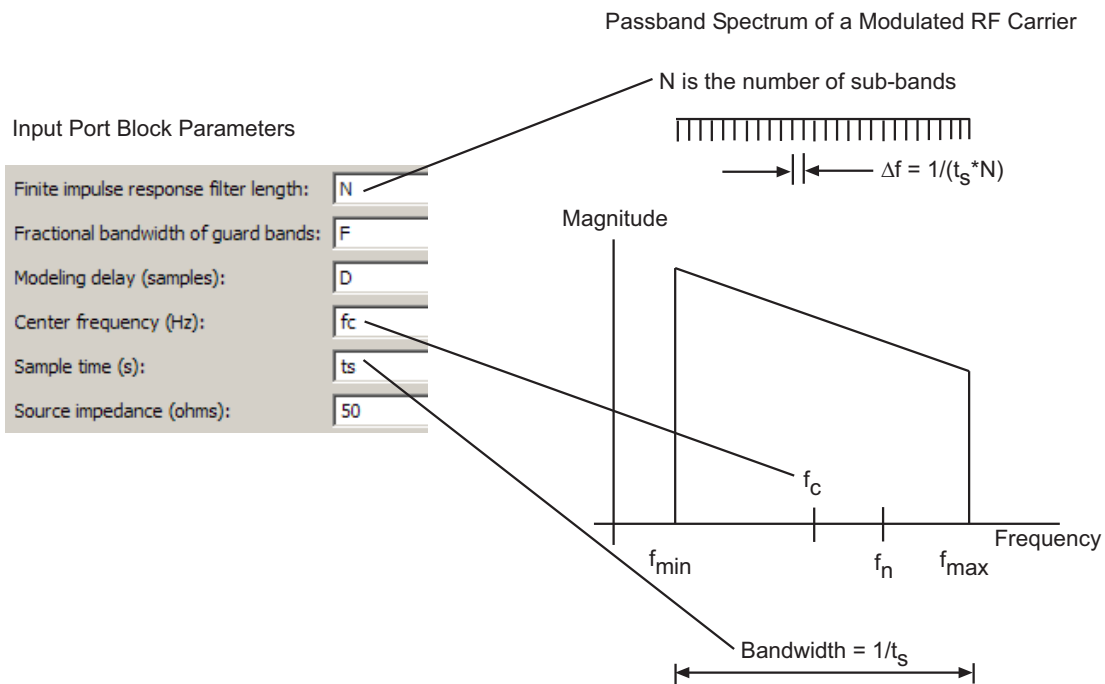
3   Translates the passband transfer function to baseband as $H(f - f_c)$, where f_c is the specified center frequency.

The baseband transfer function is shown in the following figure.

Baseband-Equivalent Spectrum
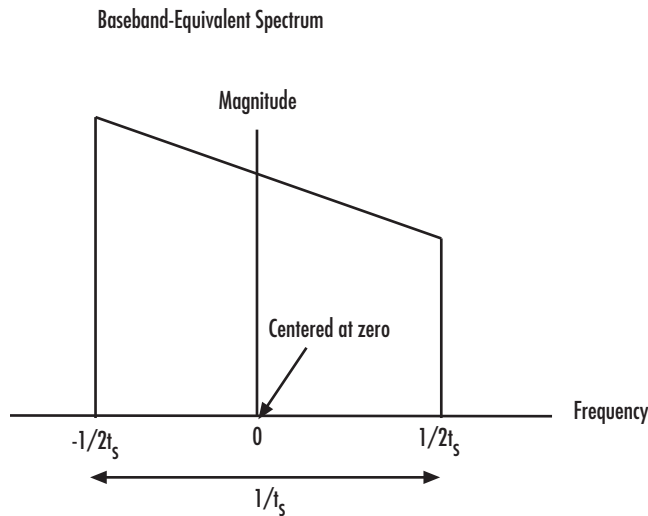
4 Obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. For faster simulation, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. Then, it truncates the impulse response 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 Design` 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.
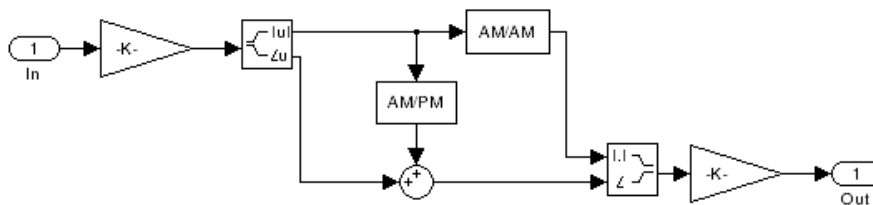
The nonlinearities of AM/AM and AM/PM conversions are extracted from the power data of an amplifier or mixer by the equations

$$AM_{out} = \sqrt{R_l P_{out}}$$
$$PM_{out} = \varphi$$
$$AM_{in} = \sqrt{R_s P_{in}}$$

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 Data Sections" in the RF Toolbox documentation for information about this format.

The following figure shows the original power data of an amplifier.
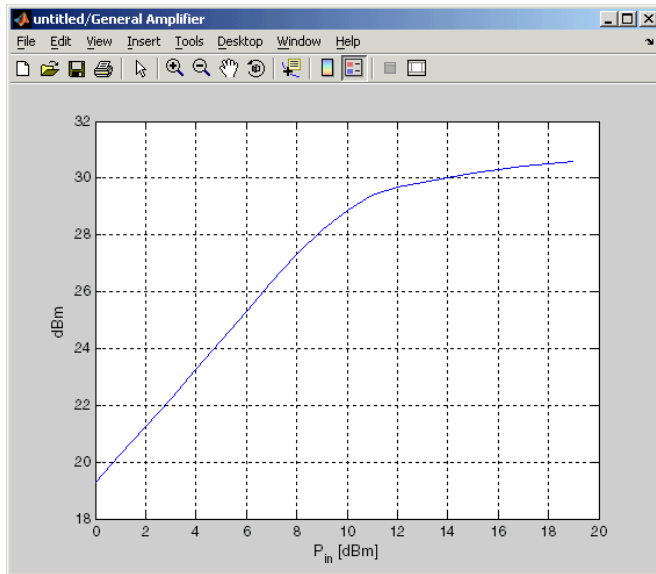
This figure shows the extracted AM/AM nonlinear conversion.
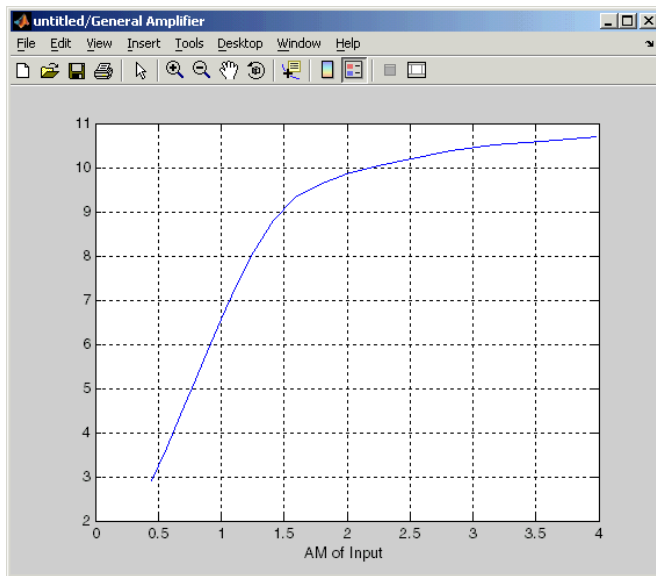
Parameters

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)

Parallel-Plate Transmission Line

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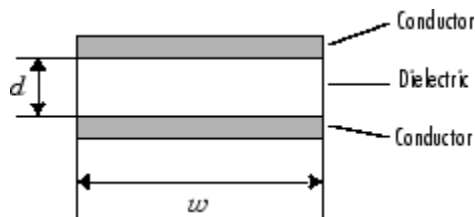
The block lets you model the transmission line as a stub or as a stubless line.

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

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

Parameters

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
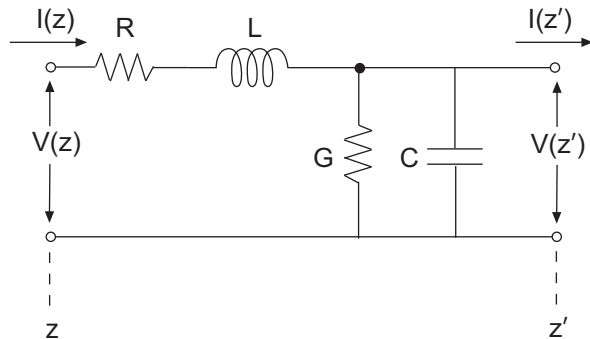
Model RLCG transmission line
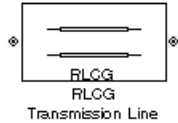
Library

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.
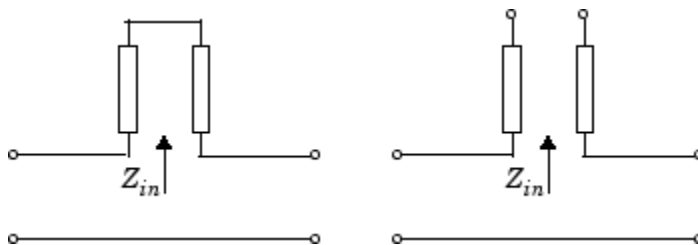

Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

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

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

Parameters

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

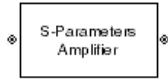For information about plotting, see "Create Plots".

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`

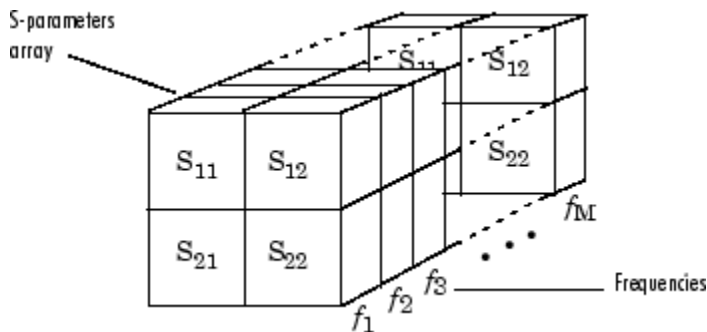# S-Parameters Amplifier

Model nonlinear amplifier using S-parameters

```
S-Parameters
Amplifier
```

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,

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

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 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.

Parameters

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]`.

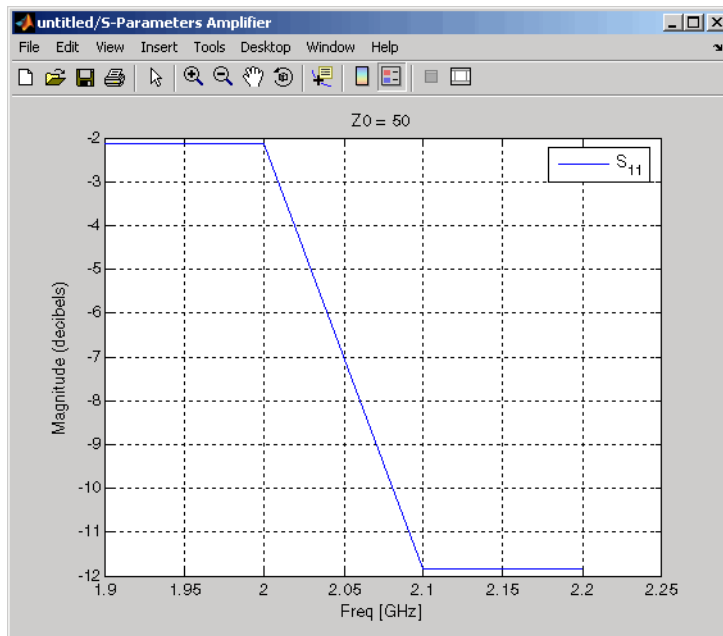Click **Apply**. This action applies the specified settings.
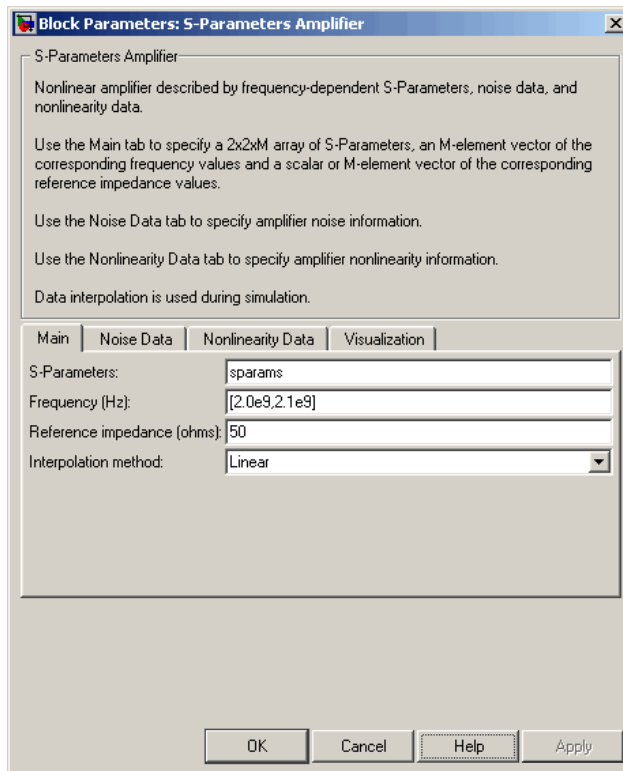
3 Set the S-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- In the **Plot type** list, select X-Y plane.
- In the **Y parameter1** list, select S11.
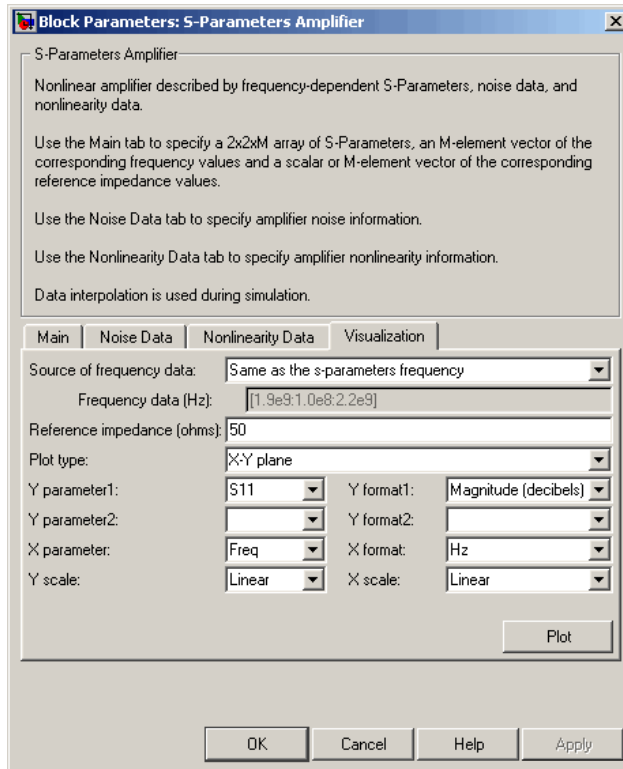
Click **Plot**. This action creates an X-Y Plane plot of the S_{11} parameters using the frequencies taken from the **Frequency (Hz)** parameter on the **Main** tab.
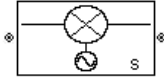
See Also

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

interp1 (MATLAB)

S-Parameters Mixer

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

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

2-133

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,

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

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.
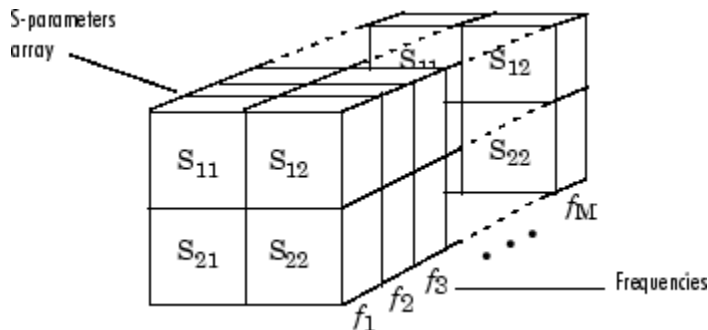
Parameters

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

S-Parameters Passive Network

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.

Parameters

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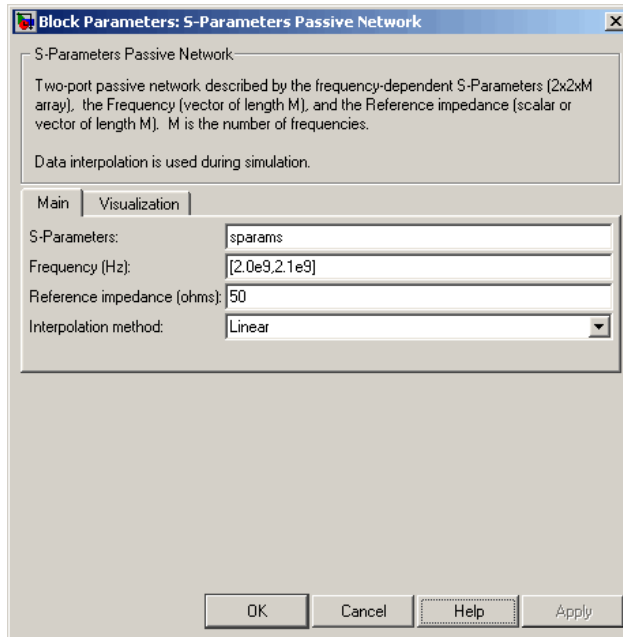
3 Set the S-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- 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.
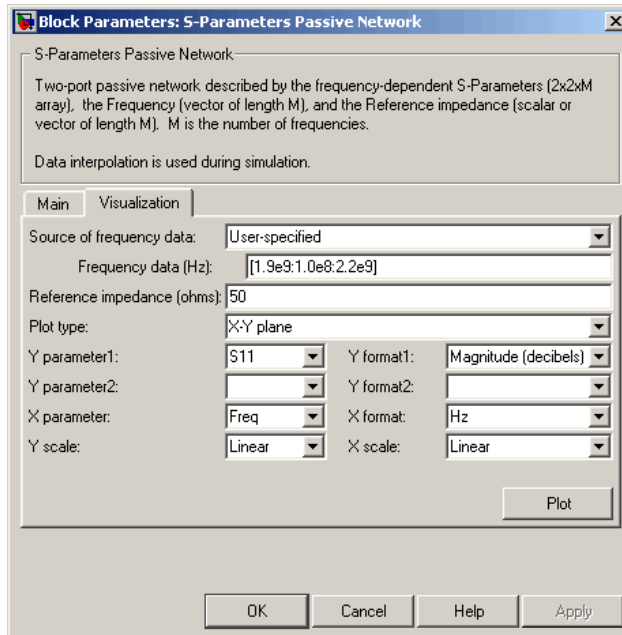
Click **Plot**. This action creates an X-Y Plane plot of the magnitude of the S_{21} parameters, in decibels, in the frequency range 1.9 to 2.2 GHz.
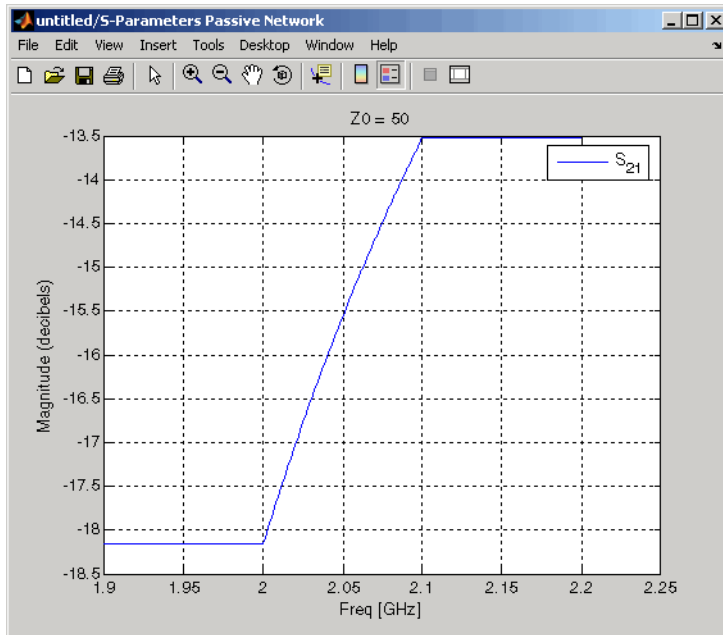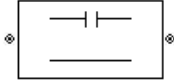
See Also

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

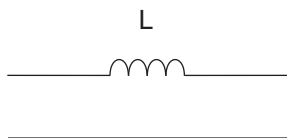interp1 (MATLAB)

Series C

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.
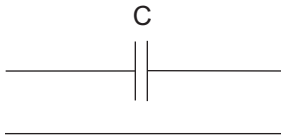
Parameters

Main Tab

Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.
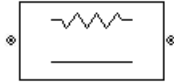
Visualization Tab

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
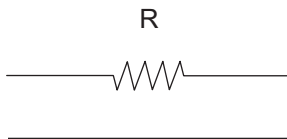
Series L

Model series inductor
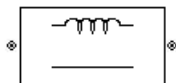
Library

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.

Parameters

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

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.

Parameters

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

Model series RLC network
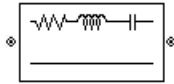
Library

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.
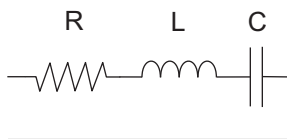
Parameters

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.
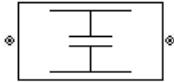
Visualization Tab

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
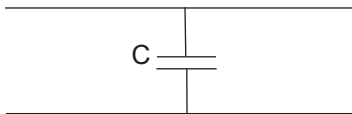
Shunt C

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.

Parameters

Main Tab

Capacitance (F)

Scalar value for the capacitance. The value must be nonnegative.
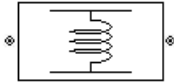
Visualization Tab

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
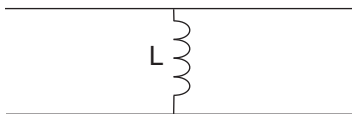
Shunt L

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.

Parameters

Main Tab

Inductance (H)

   Scalar value for the inductance. The value must be nonnegative.
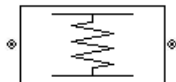
Visualization Tab

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
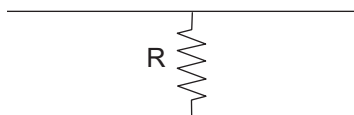
Shunt R

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.

Parameters

Main Tab

Resistance (ohms)

Scalar value for the resistance. The value must be nonnegative.
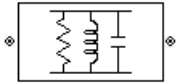
Visualization Tab

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

Shunt RLC

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.
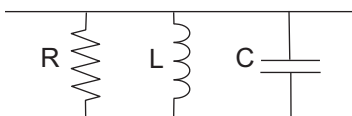
Parameters

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)

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

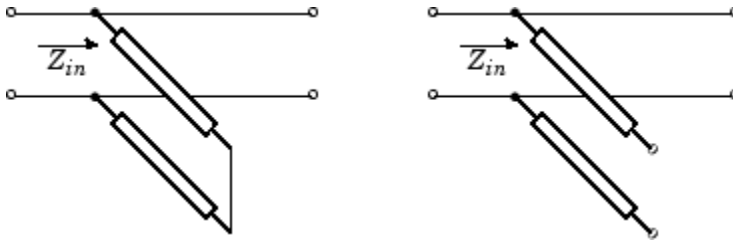$$\beta = \frac{2\pi f}{V_p}$$

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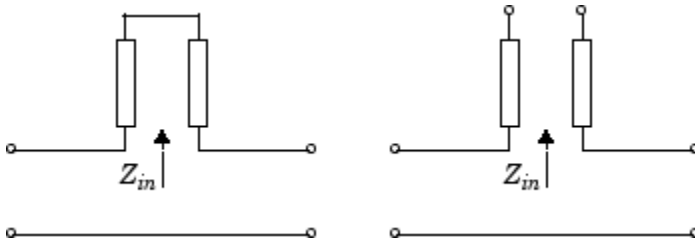
$$A = 1$$
$$B = Z_{in}$$
$$C = 0$$
$$D = 1$$

2-165

Parameters

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
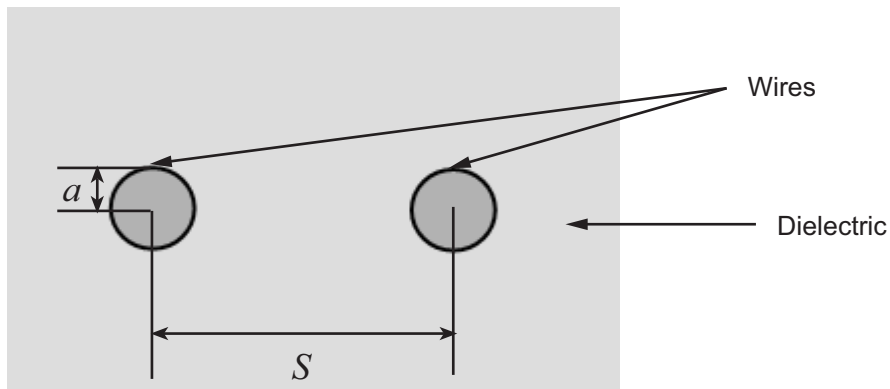
Model two-wire transmission line
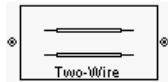
Library

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

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

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

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

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

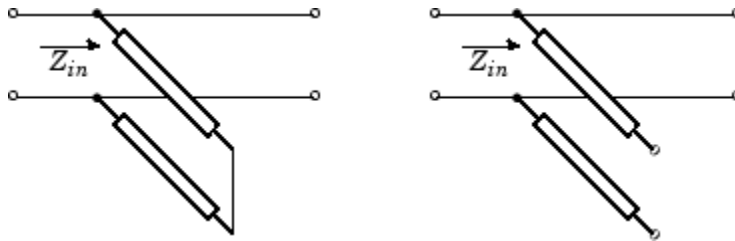and $\omega = 2\pi f$.

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

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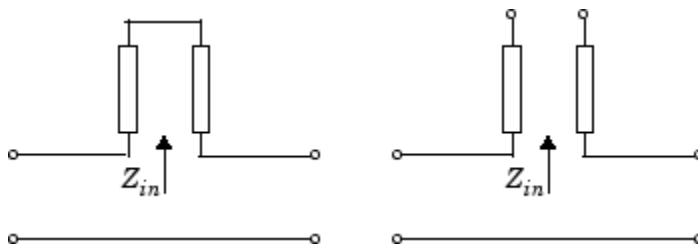$$C = 1 / Z_{in}$$
$$D = 1$$

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

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

Parameters

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".
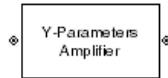
References

[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`

Y-Parameters Amplifier

Model nonlinear amplifier using Y-parameters
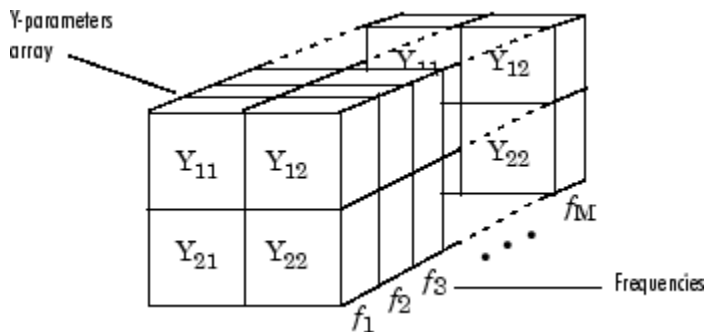
Y-Parameters
Amplifier

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,

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

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 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.

Parameters

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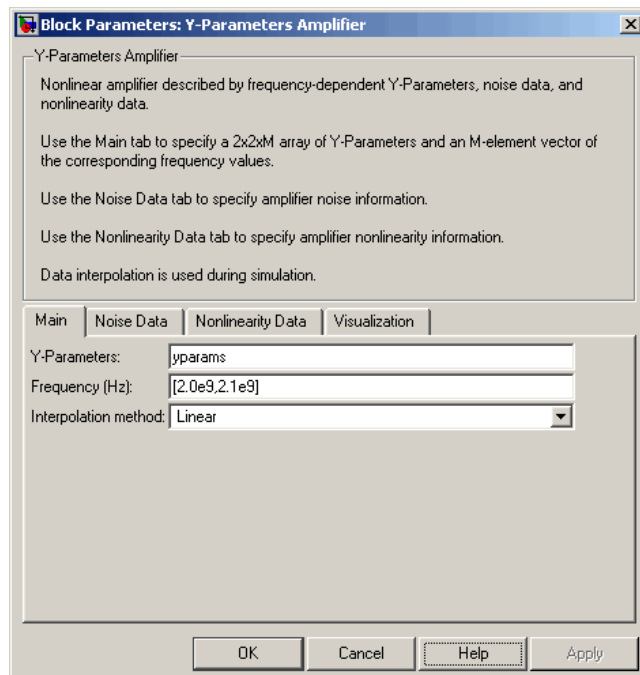
3 Set the Y-Parameters Amplifier block parameters on the **Visualization** tab as follows:

- 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.
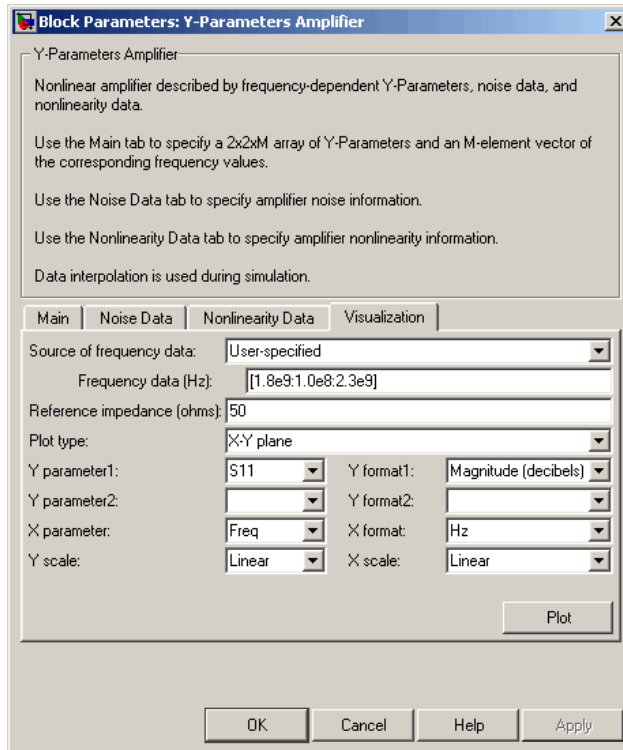
- In the **Y parameter1** list, select S11.

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.
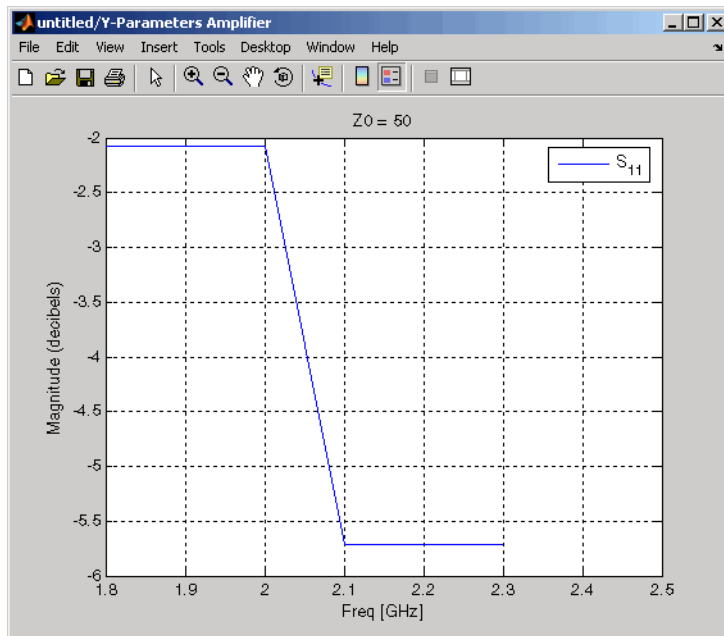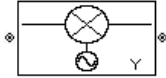
See Also

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

y2s (RF Toolbox)

interp1 (MATLAB)

Y-Parameters Mixer

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,

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

   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.
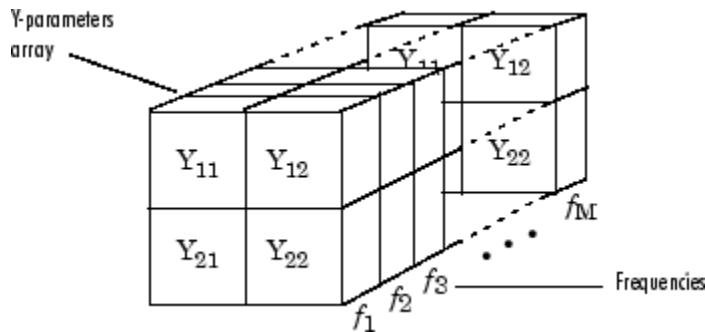
Parameters

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.
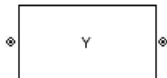
Visualization Tab

For information about plotting, see "Create Plots".

See Also

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

Y-Parameters Passive Network

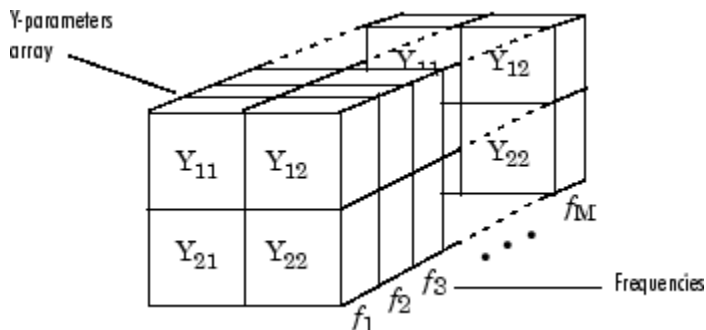Model passive network using Y-parameters

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.

Parameters

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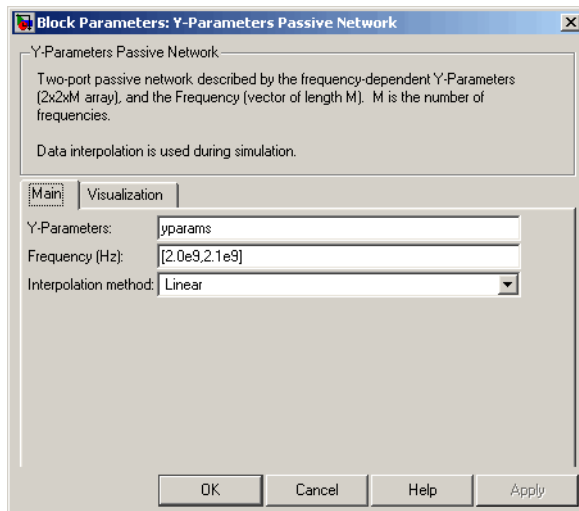
3 Set the Y-Parameters Passive Network block parameters on the **Visualization** tab as follows:

- 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`.
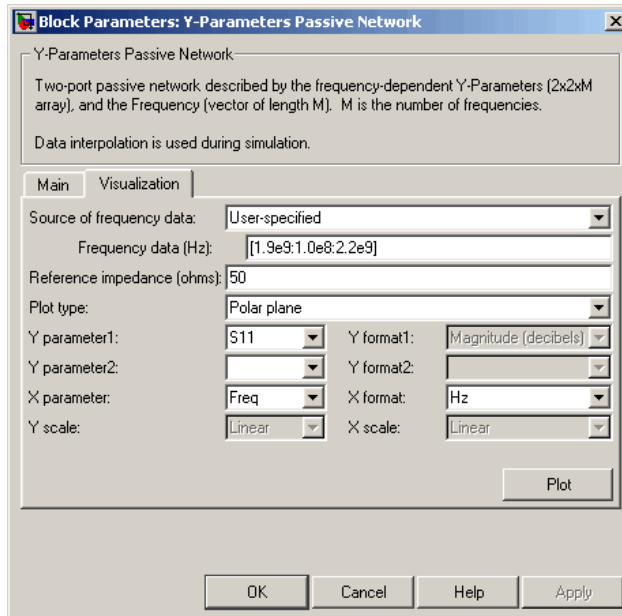
Click **Plot**. This action creates a polar plane plot of the S_{11} parameters in the frequency range 1.9 to 2.2 GHz.
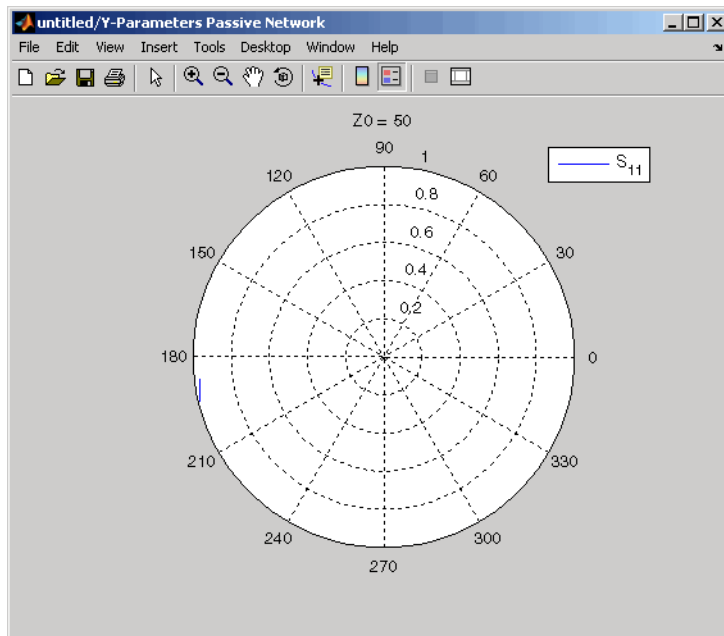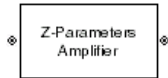
See Also

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

y2s (RF Toolbox)

interp1 (MATLAB)

Z-Parameters Amplifier

Model nonlinear amplifier using Z-parameters
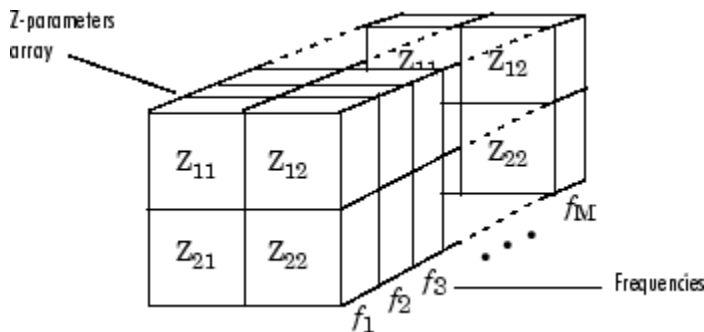

Library

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.

Parameters

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.
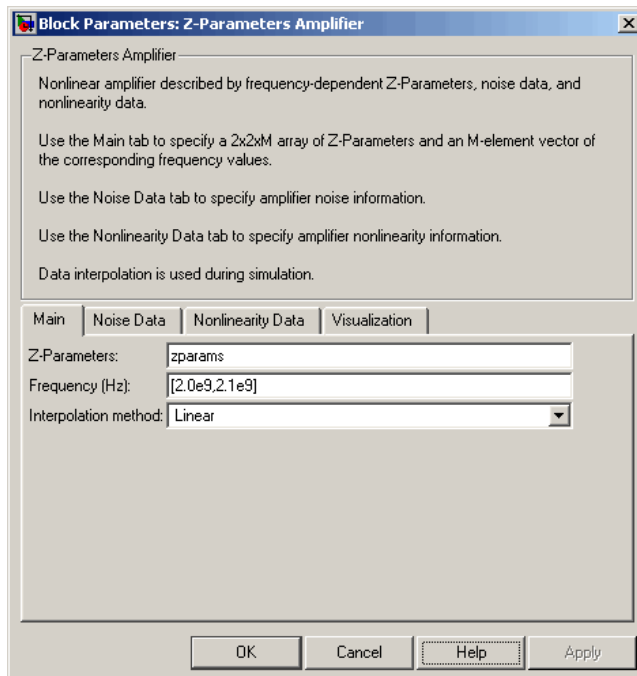
```
cat(3,...
   [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i],...
   [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i])
```

1   Type the following command at the MATLAB prompt to create a variable called `zparams` that stores the values of the Z-parameters.
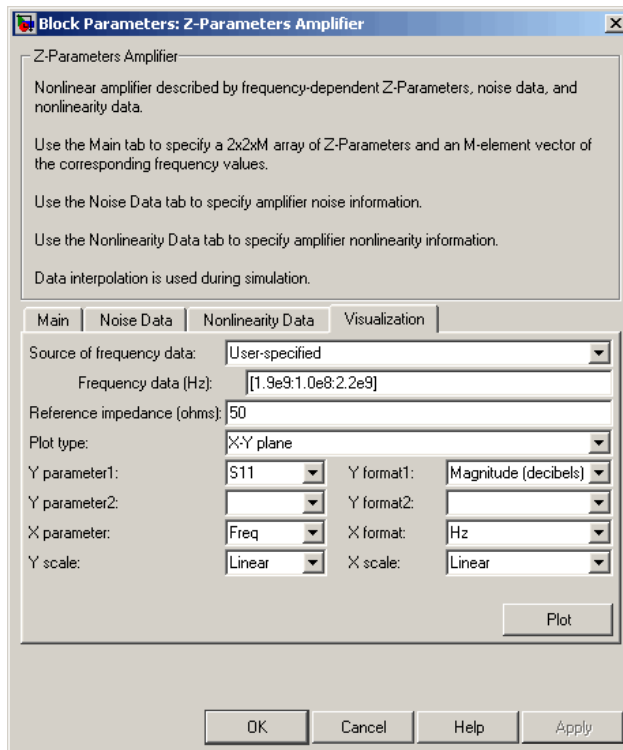
```
zparams = cat(3,...
 [12.60+3.80i, 3.77-0.17i; 80.02+54.68i, 26.02+3.84i],...
 [15.12+3.55i, 4.14-0.92i; 92.10+23.67i, 27.59+2.71i])
```

2   Set the Z-Parameters Amplifier block parameters on the **Main** tab as follows:

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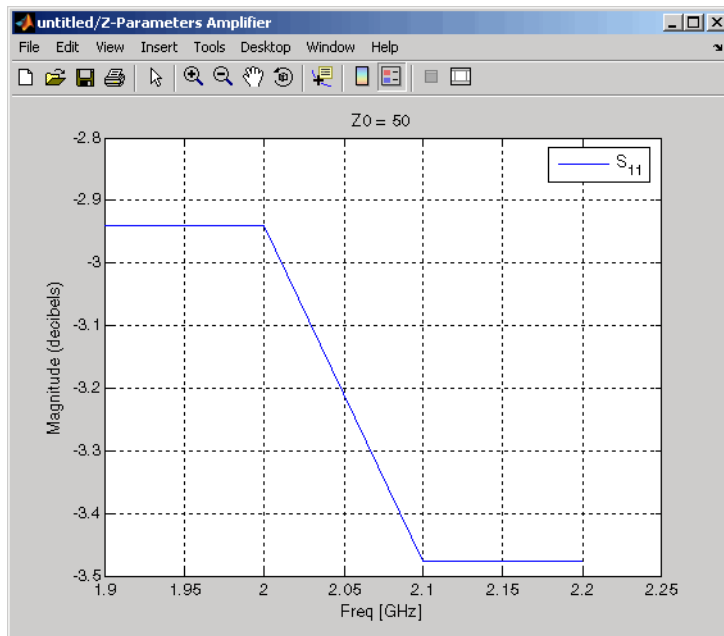- Set the **Frequency (Hz)** parameter to `[2.0e9,2.1e9]`.

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.
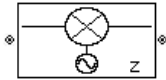
See Also

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

z2s (RF Toolbox)

interp1 (MATLAB)

Z-Parameters Mixer

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,

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

  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.

Parameters

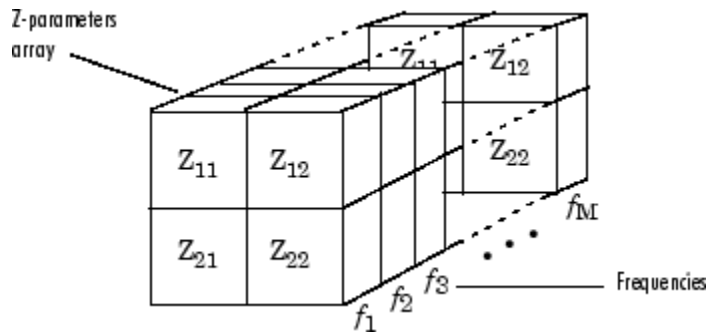### Main Tab

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.
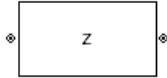
Visualization Tab

For information about plotting, see "Create Plots".

See Also

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

Z-Parameters Passive Network

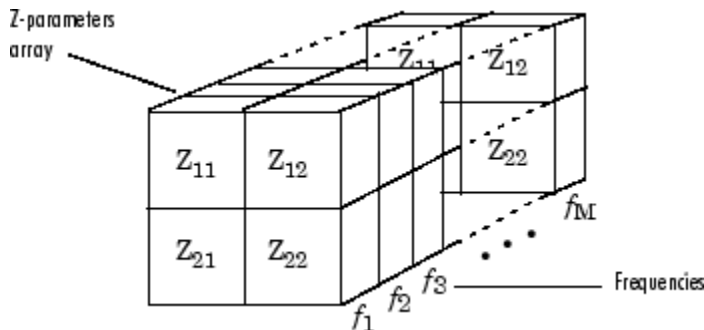Model passive network using Z-parameters

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.

Parameters

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

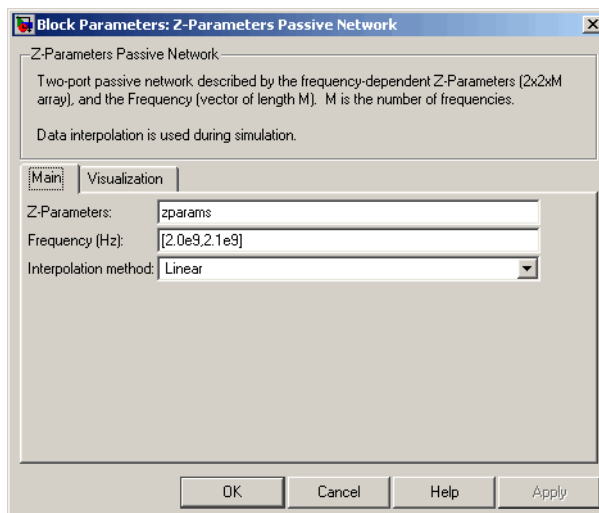respectively. It uses the MATLAB `cat` function to create the 2-by-2-by-2 Z-parameters array.

```
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])
```
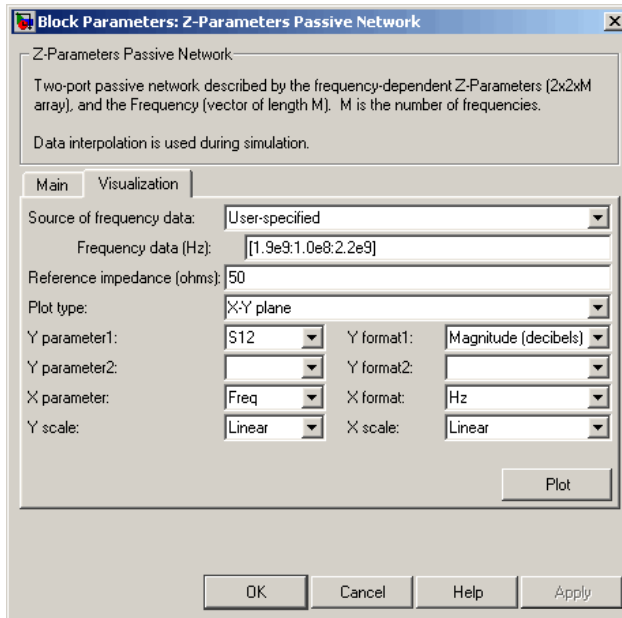
2 Set the Z-Parameters Passive Network block parameters on the **Main** tab as follows:

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

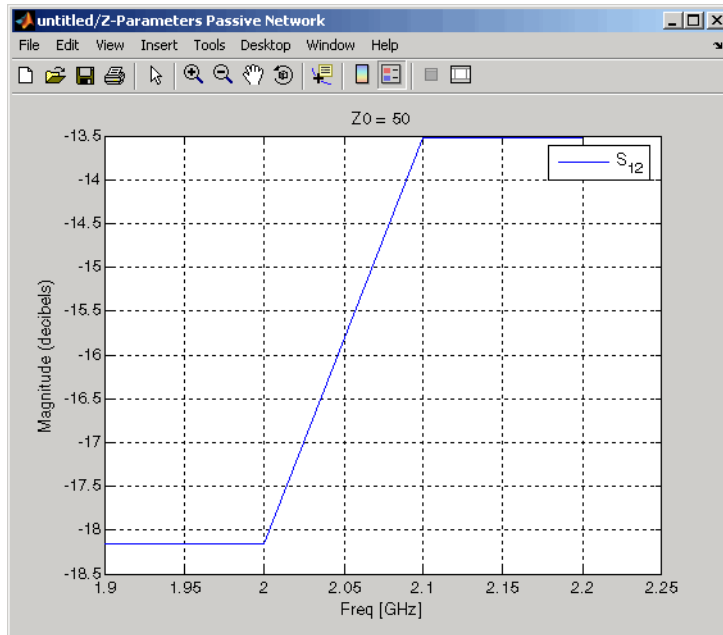Click **Apply**. This action applies the specified settings.

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]`.

2-217

- 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)

Functions — Alphabetical List

simrfSupportPackages

Download, install, or uninstall SimRF models and supporting software for third–party hardware

Syntax

```
simrfSupportPackages
```

Description

`simrfSupportPackages` opens the installer to download, install or uninstall SimRF models and supporting software for third–party hardware.

Examples

Install SimRF System Models

1 At the MATLAB command prompt, type:

   `simrfSupportPackages`
2 In the Support Package Installer, follow the instructions for installation. For more information about the options on a particular screen, click **Help**.
3 On the **Select a support package** screen, select `Analog Devices RF Transceivers`.
4 Accept or change the installation folder and click **Next**.

Note: You must have write privileges for the installation folder.

---

5 Complete the installation process. You can now access the following Analog Devices® models:

AD9361_TX Analog Devices Transmitter
AD9361_RX Analog Devices Receiver

> **Note:** To run the models, you require these additional licenses:
>
> - Communications System Toolbox™
> - Stateflow®
> - Fixed-Point Designer™

When you update your MATLAB software, you must repeat the installation process for the latest updates. You can also check for updates between releases.

> **Note:** The function `simrfSupportPackages` automatically installs AD9361 Filter Design Wizard. If you have an earlier version of the models installed, remove the filter design wizard before updating the AD9361 SimRF models. This ensures that the filter design wizard and the AD9361 models are in sync.

Introduced in R2014b