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

RF Blockset® Reference

É COPYRIGHT 2010ð2017 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

Revision History

September 2010	Online only	New for Version 3.0 (Release 2010b)
April 2011	Online only	Revised for Version 3.0.2 (Release 2011a)
September 2011	Online only	Revised for Version 3.1 (Release 2011b)
March 2012	Online only	Revised for Version 3.2 (Release 2012a)
September 2012	Online only	Revised for Version 3.3 (Release 2012b)
March 2013	Online only	Revised for Version 4.0 (Release 2013a)
September 2013	Online only	Revised for Version 4.1 (Release 2013b)
March 2014	Online only	Revised for Version 4.2 (Release 2014a)
October 2014	Online only	Revised for Version 4.3 (Release 2014b)
March 2015	Online only	Revised for Version 4.4 (Release 2015a)
September 2015	Online only	Revised for Version 4.5 (Release 2015b)
March 2016	Online only	Revised for Version 5.0 (Release 2016a)
September 2016	Online only	Revised for Version 5.1 (Release 2016b)
March 2017	Online only	Revised for Version 6.0 (Release 2017a)
September 2017	Online only	Revised for Version 6.1 (Release 2017b)

Contents

1 Circuit Envelope Blocks ñ Alphabetical List

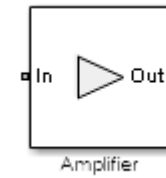
2 Equivalent Baseband Blocks ñ Alphabetical List

3 Functions ñ Alphabetical List

Circuit Envelope Blocks مم Alphabetical List

Amplifier

Model amplifier in RF systems
Library: RF Blockset / Circuit Envelope / Elements



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) except when the amplifier gain is obtained from a `Data source`. 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 $V_i(t) = A_i(t)e^{j\$t}$, and the output voltage is $V_o(t) = A_o(t)e^{j\$t}$ at each carrier $w = 2\hat{o}f$ in the RF Blockset environment.

In case the amplifier gain is obtained from a data source, amplifier implementation is based on S-parameter data.

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` —

  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`.
- `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.

When the amplifier is nonlinear, the nonlinearity applies only to the S21 term of the scattering parameters representing the 2-port element. In this case, S21 is frequency-independent with its constant value being either the maximal value of S21, or the S21 value at an Operation frequency specified by the user. The other scattering parameters, S11, S12, and S22 remain the same as in the linear case.

Dependencies

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

`Polynomial coefficients` ⇨ Polynomial coefficients

`[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, `[a0,a1,a2, ... a9]`, 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 `[a0,a1,a2,a3]` 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 `[a0,a1,a2]` defines the same polynomial as `[a0,a1,a2, 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.

`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.

`Specify operation frequency` Specify operation frequency

`on` (default) | `off`

Select this option to specify operation frequency.

By default, this option is not selected.

Dependencies

To enable this parameter, first you should specify nonlinear `Polynomial coefficients` in Source of amplifier gain. Then select `Piece-wise linear` or`Colored` in Noise distribution in the Noise pane.

`Operation frequency` Operation frequency

`0` (default) | scalar | vector

Operation frequency, specified as a scalar or vector in Hz.

Dependencies

To enable this parameter, first you should select Specify operation frequency.

`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

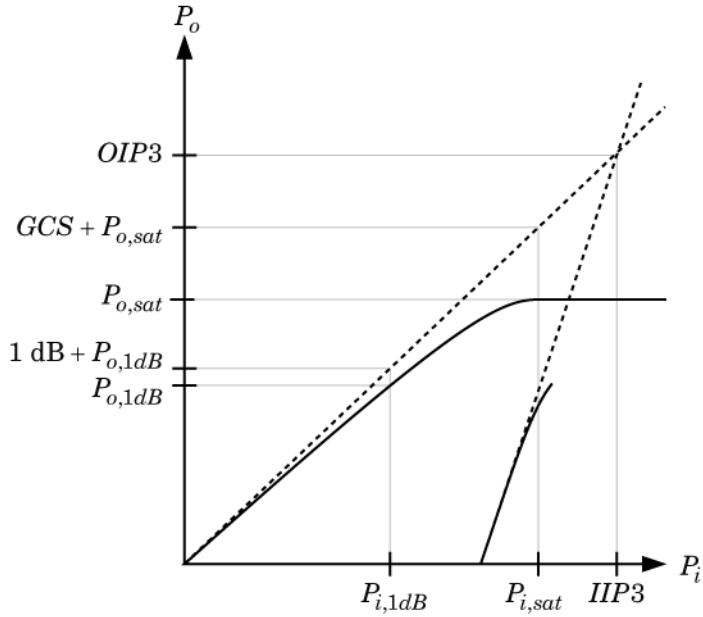
`Nonlinear polynomial type` ⌘ Type of nonlinearity

`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` م 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` م 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` م 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

`Specify operation frequency` م Specify operation frequency

`on` (default) | `off`

Select this option to specify operation frequency.

By default, this option is not selected.

Dependencies

To enable this parameter, the data source must be nonlinear or the noise should be colored.

`Operation frequency` Operation frequency
`0` (default) | scalar | vector

Operation frequency, specified as a scalar or vector in Hz.

Dependencies

To enable this parameter, first you should select Specify operation frequency.

Noise

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

Select this parameter, to simulate noise as specified in block parameters or on file.

If the noise is specified in an `.s2p` file, then it is used for simulation.

`Noise type` Noise type
`Noise figure` (default) | `Spot noise data`

Noise type, specified as `Noise figure` or `Spot noise data`.

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

Noise distribution, 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.

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

Noise figure, specified as a scalar in decibels.

`Frequencies` م Frequency data
`0Hz` (default) | scalar | vector

Frequency data, specified as a scalar or vector in hertz.

Dependencies

To set this parameter, first select `Piece-wise linear` or `Colored` in Noise distribution.

`Minimum noise figure (dB)` م Minimum noise figure
`0` (default) | scalar | vector

Minimum noise figure, specified as a scalar or vector in decibels.

Dependencies

To set this parameter, first select `Spot noise data` in Noise type.

`Optim reflection coefficient` م Optim reflection coefficient
`0` (default) | scalar | vector

Optim reflection coefficient, specified as a scalar or a vector.

Dependencies

To set this parameter, first select `Spot noise data` in Noise type.

`Equivalent normalized noise resistance` م Equivalent normalized noise resistance
`0` (default) | scalar | vector

Equivalent normalized noise resistance, specified as a scalar or vector.

Dependencies

To set this parameter, first select `Spot noise data` in Noise type.

`Automatically estimate impulse response duration` م Automatically estimate impulse response duration
`on` (default) | `off`

Select this parameter to automatically calculate impulse response for frequency dependent noises. Clear this parameter to manually specify the impulse response

duration using Impulse response duration. You cannot specify impulse response when amplifier is nonlinear, as in this case noise is simulated as white-noise.

Dependencies

To set this parameter, first select `Colored` in Noise distribution.

`Impulse response duration` Impulse response duration

`1e-10s` (default) | scalar

Impulse response duration used to simulate frequency dependant noise, specified as a scalar in seconds. You cannot specify impulse response if the amplifier is nonlinear.

Dependencies

To set this parameter, first clear Automatically estimate impulse response duration.

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 òCompare Time and Frequency Domain Simulation Options for S-parametersó.
- 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` مم 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)` مم 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` مم 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` مم 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`.

`Use only S-parameter magnitude with appropriate delay` Use only S-parameter magnitude with appropriate delay
`off` (default) | `on`

Select this parameter to s-parameter phase and delay the impulse response by half its length. This parameter is applicable only for S-parameter data modeled in time domain. You can use this to shape spectral content with filter effects by specifying only magnitude.

Note This parameter introduces an artificial delay to the system.

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` ñ 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` ñ 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` ñ 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` | `NF`

Type of S-Parameters to plot, specified as `S11`, `S12`, `S21`, or `S22`. When noise is spectral `NF` plotting is possible.

`Parameter 2` ⌘ Type of S-Parameters to plot

`None` (default) | `S11` | `S12` | `S21` | `S22` | `NF`

Type of S-Parameters to plot, specified as `S11`, `S12`, `S21`, or `S22`. When noise is spectral `NF` plotting is possible.

`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

Definitions

Noise Figure Data

Noise figure represents only a subset of the noise information (spot noise data) needed to fully describe the noise behavior of a two-port device. When only noise figure is specified, RF Blockset amplifier defines the spot noise parameters in the following manner:

$$NF_{\min} = NF \,(F_{\min} = 10^{NF/10}$$

$$R_n = Z_0 \frac{F_{\min} - 1}{4}, \quad Z_0 \in \mathrm{R}$$

$$Y_{opt} = \frac{1}{Z_0}$$

Amplifier exhibits specified noise figure when source impedance is matched to the reference impedance ($Z = Z_0, Z_0 \in \mathrm{R}$).

Spot Noise Data

Noise in RF Blockset amplifiers are represented as two correlated noise sources at the input port of a noiseless two-port:

The noise sources variance and correlation are governed by an ABCD-correlation matrix:

that is determined by measurable quantities:

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

- NF_{min} - Minimum noise figure
- R_n - Equivalent noise resistance
- Y_{opt} - Optimal source admittance
- k - Boltzman's constant
- T - Noise temperature in Kelvin

.

The above quantities are specified in the amplifier from the noise data section in the `.s2p` file or directly as masked parameters in the noise pane. In both cases:

- NF_{min} is specified in decibels
- R_n is specified as equivalent normalized resistance, R_N ($R_n = Z_0 R_N$).
- Y_{opt} is specified in terms of optimal reflection coefficient, Γ_{opt} ($Y_{opt} = Y_0(1-\Gamma_{opt})/(1+\Gamma_{opt})$).

In the above, $Z_0 = 1/Y_0$ is the reference impedance that is real. If the Source of amplifier gain is `Data source`, the reference impedance is specified in the `.s2p` file or in the amplifier mask. Other wise the reference impedance is 50 ohms.

The noise factor, F, of the amplifier is affected by the noisy source impedance, Z_s, and is determined from the ABCD-correlation matrix:

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

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

The noise figure, NF, is obtained from the noise factor using, `NF = 10log(F)`.

References

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

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

[3] Kundert, Ken. òAccurate and Rapid Measurement of IP_2 and IP_3ó, *The Designers Guide Community*, Version 1b, May 22, 2002. http://www.designers-guide.org/analysis/intercept-point.pdf.

[4] Pozar, David M. òMicrowave Engineeringó, Hoboken NJ: John Wiley & Sons, 2005.

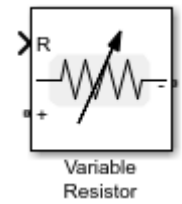
See Also

Mixer | S-Parameters

Introduced in R2010b

Variable Resistor

Model variable resistor
Library: RF Blockset / Circuit Envelope / Elements



Description

The Variable Resistor block controls the output of RF Blockset feedback circuits using Simulink® controlled resistance in ohms. The minimum value of the resistance (*Rmin*) is an RF Blockset 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 RF Blockset signal.

Parameters

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

Select this parameter, to simulate thermal noise in the variable 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 kT/R A^2/Hz, where:

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

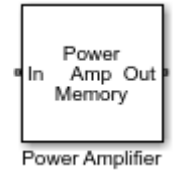
See Also

Resistor | Variable Capacitor | Variable Inductor

Introduced in R2017b

Power Amplifier

Model power amplifier with memory
Library: RF Blockset / Circuit Envelope / Elements



Description

The Power Amplifier block models two-terminal power amplifiers. A memory polynomial expression derived from the Volterra series models the nonlinear relationship between input and output signals. This power amplifier includes memory effects because the output response depends on the current input signal as well as the input signal at preceding times. These power amplifiers are useful while transmitting wideband or narrowband signals.

Parameters

`Model` ρ∞ Model type
`Memory polynomial` (default) | `Generalized Hammerstein`

Model type, specified as:

Å `Memory polynomial` ð This implementation operates on the in-phase (I) and the quadrature (Q) phase components of the envelope of the input signal. The narrowband memory polynomial does not generate new frequency components, but it captures in-band spectral regrowth. You can use this model to create a narrowband amplifier operating at a high frequency. The matrix Â represent the memory polynomial series:

$$V_{out}(t) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} \alpha_{km} V_{in}(t-m) \bullet \left| V_{in}(t-m) \right|^{k}$$

Å `Generalized Hammerstein` ð This implementation operates on the real passband input signal. This wideband model generates harmonics that are integral multiples of carrier frequencies. This formulation is equivalent to a static polynomial nonlinearity followed by a linear filter. You can use this model to create wideband amplifiers operating at low frequency. The matrix Å represent the memory polynomial series:

$$V_{out}(t) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} \alpha_{km} V_{in}^{k}(t-m)$$

`Coefficient Matrix [mem x deg]` ìí **Coefficient matrix**

1 (default) | *M*-by-*N* complex coefficient matrix | *M*-by-*N* real coefficient matrix

Coefficient matrix, specified as an *M*-by-*N* complex coefficient matrix for `Memory polynomial` and *M*-by-*N* real matrix for `Generalized Hammerstein` where, *M* is the Number of delays and *N* is the highest order of poly voltages in degrees..

For the `Memory polynomial` model, you can identify the complex coefficient matrix based on the measured complex (I,Q) output vs. input amplifier characteristic. Use the `fit_memory_poly_model` function given in òAlgorithmsó on page 1-24 as an example for identifying the coefficient matrix. When identifying the coefficients, make sure that the input-output data is aligned in time.

For the `Generalized Hammerstein` model, you can identify the real coefficient matrix based on the measured real passband output vs. input amplifier characteristic. Use the `fit_hammerstein_model` function given in òAlgorithmsó on page 1-24 as an example for identifying the coefficient matrix. When identifying the coefficients, make sure that the input-output data is aligned in time.

`Rin (Ohm)` ìí **Input resistance**

50 (default) | real scalar

Input resistance, specified as a real scalar greater than `0`.

`Rout (Ohm)` ìí **Output resistance**

50 (default) | real scalar

Output resistance, specified as a real scalar greater than `0`.

`Ground and hide negative terminals` ìí **Ground RF circuit terminals**

`on` (default) | `off`

Ground RF circuit terminals, specified as `on` or `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.

Algorithms

Function to Compute Fit Memory Polynomial Coefficients

```
function [coef]= fit_memory_poly_model(x,y,...
                 memorylen,degree)

% Note the memory polynomial in the paper is undefined for the first
% memorylen samples, as n - m < 0 and hence undefined.
% The function ignores the first memorylen samples.
% Note that the input vector x must be at least lengthVector+memorylen long.

% x_terms has lengthVector rows and number of columns corresponding to kernels
x_terms = zeros(lengthVector,degree*memorylen);

for n = 1:lengthVector %cycle through training vector
    for k = 1:degree %cycle through exponents
        %A=[a00 a01 a02 ... a0(M-1) , a10 a11 a12 ... a1(M-1 ), ... ,
             a(K-1)0 a(K-1)1 a(K-1)(M-1))]
        x_terms(n,(k - 1)*memorylen+(1:memorylen)) = ...
            (x(n:(n + memorylen-1)).*(abs(x(n:(n + memorylen -1))).^(k - 1))).';
    end
end

% Use MATLAB \ operator to generate least squares solution to overdetermined
% problem
y_terms = y(memorylen-1+(1:lengthVector));
a_coef = x_terms\y_terms;
coef = fliplr(transpose(reshape(a_coef, [memorylen degree])));
% force even coefficients to be 0!
% for j=2:2:2*floor(degree/2)
%     coef(j,:)=0;
% end
```

Function to Compute Fit Hammerstein Polynomial Coefficients

```
function [coef]= fit_hammerstein_model(x,y,lengthVector, ...
    memorylen, =degree)

% Note the memory polynomial in the paper is undefined for the first
% memorylen samples, as n - m < 0 and hence undefined.
% The function ignores the first memorylen samples.
% The input vector x must be at least lengthVector+memorylen long.

% x_terms has lengthVector rows and number of columns corresponding to kernels
x_terms = zeros(lengthVector, degree*memorylen);

for n = 1:lengthVector % cycle through training vector
    for k = 1:degree % cycle through exponents
        % A = [a00 a01 a02 ... a0(M-1) , a10 a11 a12 ... a1(M-1 ), ...
        % ..., a(K-1)0 a(K-1)1 a(K-1)(M-1))]
        x_terms(n,(k-1)*memorylen+(1:memorylen)) = ...
            (x(n:n+memorylen-1).*((x(n:n+memorylen-1)).^(k-1))).';
    end
end
% Use MATLAB \ operator to compute least squares solution to overdetermined
% (linear) problem
y_terms = y(memorylen-1+(1:lengthVector));
a_coef = x_terms\y_terms;
% Return the coefficients in matrix form [memorylen x degree]
coef = fliplr(transpose(reshape(a_coef, [memorylen degree])));
```

References

[1] Morgan,Dennis R., Zhengxiang Ma, Jaehyeong Kim, and Michael G.Zierdt. "A Generalized Memory Polynomial Model for Digital Predistortion of Power Amplifiers". *IEEE Transactions on Signal Processing* Vol.54, No.10, October 2006.

See Also

Amplifier

Introduced in R2017b

Attenuator

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

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 kT/R A^2/Hz, where:

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

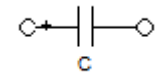
See Also

Variable Attenuator

Introduced in R2016b

Capacitor

Model capacitor for circuit envelope analysis
Library: RF Blockset / Circuit Envelope / Elements



Description

The Capacitor block models a capacitor in circuit envelope environment.

Parameters

`Capacitance` ⌘ Capacitance value
`1e-12 F` (default) | real number

Capacitance value, specified as a real number. Specify the units of the capacitance from the corresponding drop-down menu.

If you set this parameter value between `0` and `1e-18 F`, the block uses a value equal to `1e-18 F` during simulation. By default, the value is `1e-12 F`.

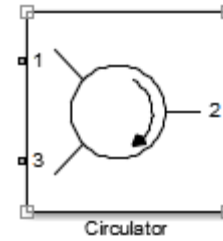
Model Examples

See Also

Inductor | Resistor

Circulator

Model ideal frequency-independent circulators with S-parameters
Library: RF Blockset / Circuit Envelope / Junctions



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.

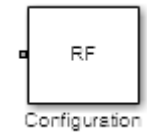
See Also

Coupler | Divider

Introduced in R2014a

Configuration

Define system simulation settings
Library: RF Blockset / 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 RF Blockset subsystem. Each Configuration block defines the parameters of the connected RF Blockset subsystem. To see an example of the Configuration block in a model, enter `RFNoiseExample` in the MATLAB Command Window.

For an introduction to RF simulation, see òSimulate High Frequency Componentsó.

Sample time in RF Blockset is

Parameters

Main

`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. This approach uses conservative number of simulatio frequencies to capture the non -linear behaviour of the system.

To set the Fundamental tones and Harmonic order, clear this parameter. 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

button

The block determines the simulation frequencies based on the fundamental tones and their respective harmonic order. The solver computes a solution to the network at each simulation frequency and the computation time scales according to the total number of simulation frequencies.

Combinations of fundamental tones determine the set of simulation frequencies: [$m*f1 + n*f2 +$ é]. In this case, the fundamental tones are represented by [f_s1,$f2$, é], and the integers m and n are integers bounded by the corresponding Harmonic order, $|m| \quad h1$, $|n| \quad h2$, etc. Only positive frequencies are considered.

Click View to open the 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.

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.

Consider a single fundamental tone $f1$ = 2 GHz and corresponding harmonic order $h1$ = 3. The set of simulation frequencies are: `[0, f1, 2f1, 3f1] = [0GHz, 2 GHz, 4 GHz, 6GHz]`.

Consider a circuit with two fundamental tones `[f1 = 2 GHz, f2 = 50 MHz]` and corresponding harmonic orders `h1 = h2 = 1`. This setup results in five simulation frequencies with values: `[0, f2, f1-f2, f1, f1+f2]`.

Consider a circuit with two fundamental tones `[f1 = 2 GHz, f2 = 3GHz]` and corresponding harmonic orders `h1 = 1`, and `h2 = 3`. This setup results in 11 simulation frequencies with values: `[0, f2, f1-f2, f1, f1+f2, -f1+2f2, 2f2, -f1+3f2, f1+2f2, 3f2, f1+3f2]`.

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

Dependencies

To enable this parameter, select Automatically select fundamental tones and harmonic order. If you clear Automatically select fundamental tones and harmonic order, the option becomes, Total simulation frequencies: N/A: Fundamental tones undefined.

`Step size` Time step for fixed step solver configuration

`1e-6` (default) | scalar in seconds

Time step for fixed step solver configuration, specified as a scalar in seconds. The inverse of the time step determines the simulation bandwidth of the signal envelope centred around each simulation frequency.

The time step of a circuit envelope simulation should be commensurate to relative signal bandwidth and not to the absolute value of the carrier frequency.

The default (1e-6s) is sufficient for modelling envelope signals with bandwidths of up to 1/*h*, or 1MHz. 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.

The simulation speed is inversely proportional to the simulation step size. A smaller simulation step size corresponds to a wider envelope bandwidth and to a slower simulation.

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

`Envelope bandwidth` ⁄ Maximum simulated envelope bandwidth

`1 MHz` (default) | scalar in Hz

Maximum simulated envelope bandwidth, returned as a scalar in Hz. Configuration block automatically calculates this value using the Step size parameter. The formula used is: $bandwidth = \frac{1}{(step\ size)}$.

`Simulate noise` ⁄ Globally enable or disable noise modeling

`on` (default) | `off`

Select this parameter to globally enable noise modeling in RF Blockset 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 random number generator` ⁄ Default pseudorandom noise stream for RF Blockset sources

`on` (default) | `off`

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

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.

`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 Ş:

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

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 Ş 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

Advanced

`Transient analysis` ⇒ Fixed-step solver of RF Blockset environment
`Auto` (default) | `NDF2` | `Trapezoidal Rule` | `Backward Euler`

Fixed-step solver of RF Blockset 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 RF Blockset solver is an extension of the Simscape local solver. For more information on the Simscape local solver, see the Solver Configuration block reference page.

`Relative tolerance` ⇒ Relative newton tolerance for system variables
`1e-3` (default) | real positive finite scalar

Relative newton tolerance for system variables, specified as a real positive finite scalar.

`Absolute tolerance` ⇒ Absolute newton tolerance for system variables
`1e-6` (default) | real positive finite scalar

Absolute newton tolerance for system variables, specified as a real positive finite scalar.

`Maximum iterations` ⇒ Number iterations required for convergence
`10` (default) | real positive integer scalar

Number iterations required for convergence, specified as a real positive integer scalar.

`Error estimation` Check for error of convergence in system variables
`2-norm over all variables` (default) | `Each variable separately`

Check for error of convergence in system variables, specified as:

- `2-norm over all variables`: Use this option to calculate the 2-norm of all the state variables and then check the error in convergence of state variables.
- `Each variable separately`: Use this option to check the error in convergence of each variable separately.

`Restore Default Settings` Restore newton solver to default values
button

Restore newton solver to default values, specified as a button.

Model Examples

Definitions

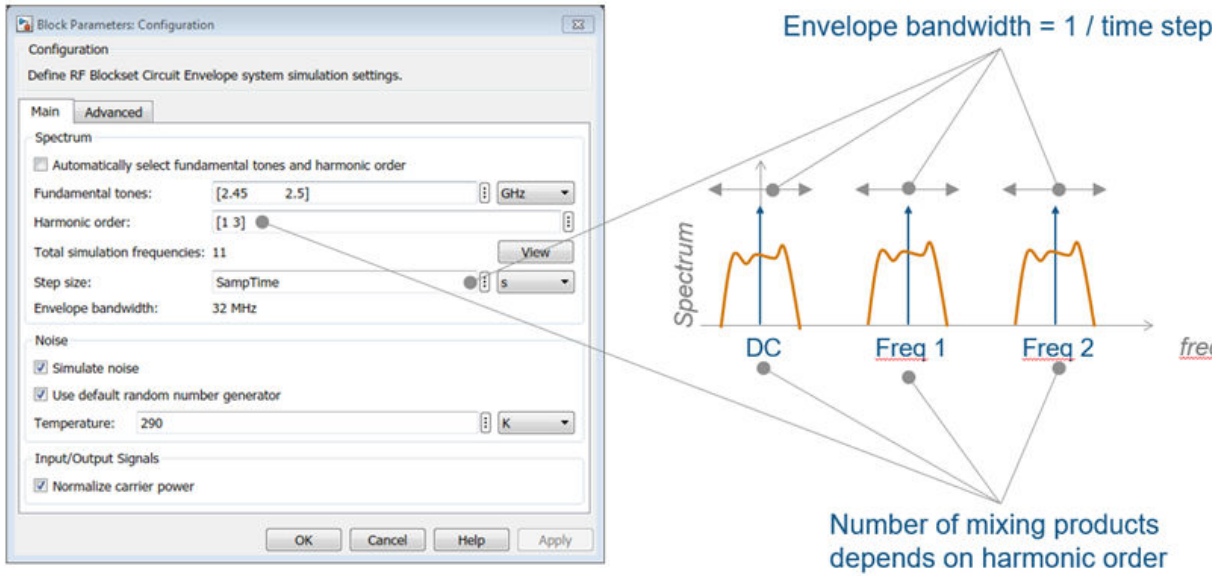
Simulation Setup and Complexity

The key parameters in setting up a Circuit Envelope simulation are the fundamental tones, the harmonic order, and the step size. To speed up simulation, you can trade off the simulation step size and the total number of simulation frequencies.

For example, suppose that you have two large inputs signals each with 100 MHz bandwidth, centred around 10 GHz, and 10.1 GHz respectively. You can simulate the two signals using two separate fundamental tones [10 10.1] GHz. Each tone has a harmonic order of 3 (for a total of 13 simulation frequencies), and a simulation step size equal to 1/200MHz = 5 ns.

You could also set up the RF subsystem so that both of the signals are within the same simulation bandwidth centred around 10.05 GHz. In this case, you set the harmonic order equal to 3 (for a total of 4 simulation frequencies), and a simulation step size equal to 1/400MHz = 2.5 ns. The latter configuration is faster as the number of simulation frequencies is smaller by a factor 3, and the simulation step size is only smaller by a factor 2.

When setting up a circuit envelope simulation, avoid overlapping envelopes. The thermal noise generated by passive components are accounted for separately in each subband thus allowing for overlap of separate envelopes.

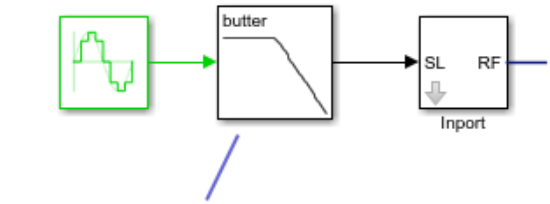


Criteria for Determining Simulation Step Size

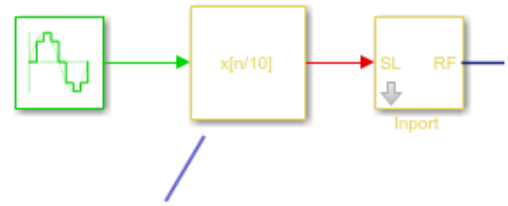
The simulation step size must be small enough to capture the signal bandwidth and in-band spectral regrowth.

For example, your complex input Simulink signal has a sample frequency equal to 10 MHz. The minimum time step required to simulate this signal is 1/20 MHz = 50 ns. You can use an oversampling factor from 4 through 8, corresponding to a simulation time step between 25 ns and 12.5 ns. This captures the spectral regrowth caused by non-linear effects.

It is recommended that the Configuration block,use the same step size as the input Simulink signals. The input port resamples the input signal with the step size specified in the Configuration block. Using the same step size avoids undesired aliasing effects. It is best to resample the Simulink signals before importing them in RF Blockset using either analog (continuous time) or digital (discrete time) interpolation filters.



Continuous time interpolation filter



Discrete time interpolation filter

Relative Tolerance and Absolute Tolerance

Circuit envelope solver in the RF Blockset is a solving a set of nonlinear equations from a set of system variables. These system variables are derived from the circuit topology and simulation frequencies. Relative tolerance and absolute tolerance are used to keep the error in convergence of the system variables to minimum. The number of iterations used at each time step dramatically affects the speed of the solutions and the tradeoff between accuracy and speed. The tradeoff is governed by the stopping criterion for the iterations. This stopping criterion is based on 3 sub criterion:

- Variable error convergence:

  $$|\Delta X| < \mathrm{Re}lTol.\max_{t}(|X|) + AbsTol_x$$

  where:

  - X- System variables
  - t- maximum iterations.
- Residue error convergence:

  $$|F(X)| < \mathrm{Re}lTol.\max_{t,n}(|F_n(X)| + AbsTol_F$$

  where:

  - $F_n(X)$- represents a part of $F(X)$ coming from the nth branch.
- Maximum number of iterations.

Stop the calculations if the first two sub criteria are filled or the last sub criterion is filled. If only one of the sub-criteria is filled, error out that the ' non-linear solver failed'.

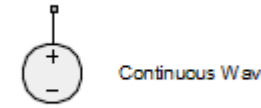
See Also

Inport | Outport

Introduced in R2013a

Continuous Wave

Model constant envelope source
Library: RF Blockset / Circuit Envelope / Sources



Description

The Continuous Wave block models a constant modulation on a carrier in the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

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 $\$_k$, depending on the source type.

Parameters

`Source type` **Wave type**
`Ideal voltage` (default) | `Ideal current` | `Power`

- `Ideal Voltage` ñ 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` ñ 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.

Å `Power` ñ The block simulates the constant voltage envelope

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

where:

- Å P_0 is the value of the Available power parameter
- Å Z_s is the value of the Source impedance (ohms) parameter.
- Å 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`.

`Add phase noise` — Add phase noise

`off` (default) | `on`

Select this parameter, to add phase noise to your system with continuous wave source.

`Phase noise frequency offset (Hz)` — Phase noise frequency offset

`0` (default) | scalar | vector | matrix

Phase noise frequency offset, specified as a scalar or vector or matrix with each element units in hertz.

If you specify a matrix, each column should correspond to a non-DC carrier frequency of the CW source. The frequency offset values must be bounded by the envelope bandwidth of the simulation. For more information see, Configuration.

Dependencies

To enable this parameter, select Add phase noise.

`Phase noise level (dBc/Hz)` مم **Phase noise level**

`0` (default) | scalar | vector | matrix

Phase noise level, specified as a scalar or vector or matrix with element unit in decibel per hertz.

If you specify a matrix, each column should correspond to a non-DC carrier frequency of the CW source. The frequency offset values must be bounded by the envelope bandwidth of the simulation. For more information see, Configuration.

Dependencies

To enable this parameter, select Add phase noise.

`Automatically estimate impulse response duration` مم **Automatically estimate impulse response duration**

`on` (default) | `off`

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

`Impulse response duration` مم **Impulse response duration**

`1e-10s` (default) | scalar

Impulse response duration used to simulate phase noise, specified as a scalar in seconds. You cannot specify impulse response if the amplifier is nonlinear.

Note The phase noise profile resolution in frequency is limited by the duration of the impulse response used to simulate it. Increase this duration to improve the accuracy of the phase noise profile. A warning message appears if the phase noise frequency offset resolution is too high for a given impulse response duration, specifying the minimum duration suitable for the required resolution

---

Dependencies

To set this parameter, first clear Automatically estimate impulse response duration.

`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

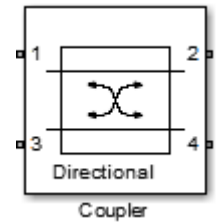
See Also

Inport | Noise | Sinusoid

Introduced in R2010b

Coupler

Model ideal frequency-independent couplers with S-parameters
Library: RF Blockset / 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:

Å `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:

- $r_l = 10^{(-\text{ReturnLoss}/20)}$
- $i_l = 10^{(-\text{InsertionLoss}/20)}$
- $i_s = 10^{(-(\text{Coupling}+\text{Directivity})/20)}$
- $c = 10^{(-\text{Coupling}/20)}$

- `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:

- $|\alpha| \le 1$ = 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:

Å |À| Ò 1 = Power transmission coefficient.

Å à = sqrt (1ð À*À)

Å `Hybrid quadrature (90deg)`

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

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

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

By default, this option is selected.

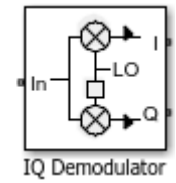
See Also

Circulator | Divider

Introduced in R2014a

IQ Demodulator

Convert RF signal to baseband signal
Library: RF Blockset / Circuit Envelope / Systems



Description

The `IQ Demodulator` converts an RF signal to baseband signal. `I` stands for the in-phase component of the signal and `Q` stands for the quadrature phase component of the signal. You can use the IQ Demodulator to design direct conversion receivers.

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` ñ Relates the power of a single sideband (SSB) at the output `I` branch to the input power. If there is no gain mismatch, the gain at the `Q` branch matches the gain at the `I` branch.
- `Open circuit voltage gain` ñ Value of the open circuit voltage gain parameter as the linear voltage gain term of the polynomial VCVS.
- `Polynomial coefficients` ñ Implements a nonlinear voltage gain according to the polynomial you specify.

`Available power gain` Power of SSB at output to input power at `I`
0 dB (default) | scalar in dB or a unitless ratio

Power of SSB at output to input power at `I`, specified as a scalar in dB or a unitless ratio, `None`.

Dependencies

To enable this parameter, set Source of conversion gain to `Available power gain`.

`Open circuit voltage gain` ⌘ Open circuit voltage gain

`0 dB` (default) | scalar

Open circuit voltage gain, specified as a scalar in dB.

Dependencies

To enable this parameter, set Source of conversion gain to `Open circuit voltage gain`.

`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 must be ordered in ascending powers. If a vector has 10 coefficients, $[a_0, a_1, a_2, \ldots a_9]$, the polynomial it represents is:

$V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \ldots + a_9 V_{in}^9$
a_1 represents the linear gain term, and higher-order terms are modeled according to [2].

For example, the vector $[a_0, a_1, a_2, a_3]$ specifies the relation
$V_o = a_0 + a_1 V_1 + a_2 V_1^2 + a_3 V_1^3$. Trailing zeros are omitted, if $a_3 = 0$, then $[a_0, a_1, a_2]$ defines the same polynomial as $[a_0, a_1, a_2, 0]$.

By default, the value is [0,1], corresponding to the linear relation $V_o = V_i$.

Dependencies

To enable this parameter, set Source of conversion gain to `Polynomial coefficients`.

`Local oscillator frequency` ⌘ Local oscillator (LO) frequency

`0 Hz` (default) | scalar

Local oscillator (LO) frequency, specified as a scalar in `Hz`, `kHz`, `MHz`, or`GHz`. `0` Hz is not allowed.

`Input impedance (Ohm)` Input impedance of demodulator
`50` (default) | scalar

Input impedance of demodulator, specified as a scalar.

`Output impedance (Ohm)` Output impedance of demodulator
`50` (default) | scalar

Output impedance of demodulator, specified as a scalar.

`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.

Impairments

`I/Q gain mismatch` Gain difference between `I` and `Q` branches
`0 dB` (default) | scalar

Gain difference between `I` and `Q` branches, specified as a scalar in dB. Gain mismatch is assumed to be forward-going, that is, the mismatch does not affect leakage from LO to RF).

If the gain mismatch is specified, the value $Available\ power\ gain + I/Q\ gain\ mismatch$ relates to the power of the single-sideband (SSB) at output the `Q` branch to the input power.

`I/Q phase mismatch` Phase difference between `I` and `Q` branches
`0 degrees` (default) | scalar in degrees or radians

Phase difference between `I` and `Q` branches, specified as a scalar in degrees or radians. The phase mismatch affects the LO to input RF leakage.

`LO to RF isolation` ⁄ Ratio of magnitude between LO voltage to leaked RF voltage
`inf` dB (default) | scalar

Ratio of magnitude between LO voltage to leaked RF voltage, specified as a scalar in dB. Phase accumulation in the path from LO input to the internal `I` and `Q` mixers (after phase shift and phase mismatch) and then to the RF is assumed to be zero.

`Noise figure (dB)` ⁄ Signal-to-noise ratio (SNR) between outputs and input
`0` (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, IQ Demodulator 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 $Ş_{IM} = Ş_{LO} + (Ş_{LO} ð Ş_{RF})$.

Noise Convention	Signal at RF Frequency	Signal at IM Frequency	IQ Demodulator 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.

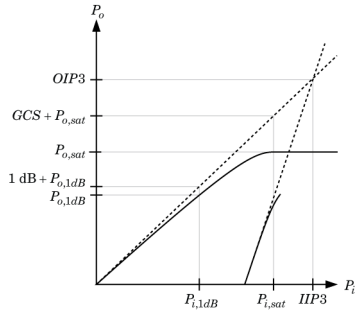
Nonlinearity

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

Polynomial nonlinearity, specified as one of the following:

- `Even and odd order`: The IQ Demodulator can produce second-order and third-order intermodulation frequencies, in addition to a linear term.
- `Odd order`: The IQ Demodulator generates only odd- order intermodulation frequencies.

The linear gain determines the linear a_1 term. The block calculates the remaining terms from the values specified in 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 demodulator parameters.



`Intercept points convention` Intercept points convention

`Output` (default) | `Input`

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

`IP2` Second-order intercept point

`inf dBm` (default) | scalar

Second-order intercept point, specified as a scalar in `dBm`, `W`, `mW`, or `dBW`. The default value `inf dBm` corresponds to an unspecified point.

Dependencies

To enable this parameter, set Nonlinear polynomial type to `Even and odd order`.

`IP3` Third-order intercept point

`inf dBm` (default) | scalar

Third-order intercept point, specified as a scalar in `dBm`, `W`, `mW`, or `dBW`. 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 .

References

[1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.

[2] Grob, Siegfried and Lindner, Jurgen, òPolynomial Model Derivation of Nonlinear Amplifiersó, *Department of Information Technology*, University of Ulm, Germany.

See Also

IQ Modulator | Mixer

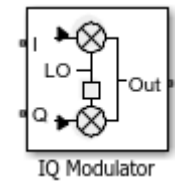
Topics

òDemodulate Two-Tone RF Signal Using IQ Demodulatoró

Introduced in R2017a

IQ Modulator

Convert baseband signal to RF signal
Library: RF Blockset / Circuit Envelope / Systems



Description

The `IQ Modulator` converts a baseband signal to RF signal and models an IQ modulator with impairments. `I` stands for the in-phase component of the signal and `Q` stands for the quadrature phase component of the signal. You can use the IQ Modulator to design direct conversion transmitters.

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` ñ Relates the power of a single-sideband (SSB) at the output `I` branch to the input power. If there is no gain mismatch, the gain at the `Q` branch matches the gain at the `I` branch.
- `Open circuit voltage gain` ñ Value of the open circuit voltage gain parameter as the linear voltage gain term of the polynomial VCVS.
- `Polynomial coefficients` ñ Implements a nonlinear voltage gain according to the polynomial you specify.

`Available power gain` — Power of SSB at output to input power at `I`

`0 dB` (default) | scalar in dB or a unitless ratio

Power of SSB at output to input power at `I`, specified as a scalar in dB or a unitless ratio, `None`.

Dependencies

To enable this parameter, set Source of conversion gain to `Available power gain`.

`Open circuit voltage gain` — Open circuit voltage gain

`0 dB` (default) | scalar in dB or a unitless ratio

Open circuit voltage of IQ modulator, specified as a scalar in dB or a unitless ratio, `None`.

Dependencies

To enable this parameter, set Source of conversion gain to `Open circuit voltage gain`.

`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 must be ordered in ascending powers. If a vector has 10 coefficients, `[`a_0`,`a_1`,`a_2`, ... `a_9`]`, the polynomial it represents is:

$V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + ... + a_9V_{in}^9$
a_1 represents the linear gain term, and higher-order terms are modeled according to [2].

For example, the vector `[`a_0`,`a_1`,`a_2`,`a_3`]` specifies the relation
$V_o = a_0 + a_1V_1 + a_2V_1^2 + a_3V_1^3$. Trailing zeros 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]`.

By default, the value is [0,1], corresponding to the linear relation $V_o = V_i$.

Dependencies

To enable this parameter, set Source of conversion gain to `Polynomial coefficients`.

`Local oscillator frequency` ممم Local oscillator (LO) frequency
`0 Hz` (default) | scalar

Local oscillator (LO) frequency, specified as a scalar in `Hz`, `kHz`, `MHz`, or`GHz`. `0` Hz is not allowed.

`Input impedance (Ohm)` ممم Input impedance of modulator
`50` (default) | scalar

Input impedance of modulator, specified as a scalar.

`Output impedance (Ohm)` ممم Output impedance of modulator
`50` (default) | scalar

Output impedance of modulator, specified as a scalar.

`Ground and hide negative terminals` ممم 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.

Impairments

`I/Q gain mismatch` ممم Gain difference between `I` and `Q` branches
`0 dB` (default) | scalar

Gain difference between `I` and `Q` branches, specified as a scalar in dB. Gain mismatch is assumed to be forward-going, that is, the mismatch does not affect leakage from LO to RF).

If the gain mismatch is specified, the value $Available\ power\ gain + I/Q\ gain\ mismatch$, relates to the power of the single-sideband (SSB) at input the `Q` branch to the output power.

`I/Q phase mismatch` ممم Phase difference between `I` and `Q` branches
`0 dB` (default) | scalar in degrees or radians

Phase difference between `I` and `Q` branches, specified as a scalar in degrees or radians. This mismatch affects the LO to input RF leakage.

`LO to RF isolation` — Ratio of magnitude between LO voltage to leaked RF voltage
`inf dB` (default) | scalar

Ratio of magnitude between LO voltage to leaked RF voltage, specified as a scalar in dB.

`Noise floor (dBm/Hz)` — Single-sided noise power spectral distribution
`0` (default) | scalar in dBm/Hz

Single-sided noise power spectral distribution, specified as a scalar in dBm/Hz. This block assumes zero noise input at `I` and `Q` branches.

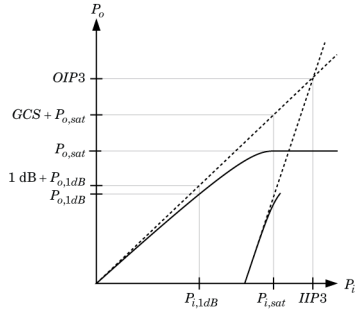
Nonlinearity

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

Polynomial nonlinearity, specified as one of the following:

- `Even and odd order`: The IQ Modulator can produce second-order and third-order intermodulation frequencies, in addition to a linear term.
- `Odd order`: The IQ Modulator generates only odd- order intermodulation frequencies.

The linear gain determines the linear a_1 term. The block calculates the remaining terms from the values specified in 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 modulator parameters.



`Intercept points convention` ⌘ Intercept points convention

`Output` (default) | `Input`

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

`IP2` ⌘ Second-order intercept point

`inf dBm` (default) | scalar

Second-order intercept point, specified as a scalar in `dBm`, `W`, `mW`, or `dBW`. The default value `inf dBm` corresponds to an unspecified point.

Dependencies

To enable this parameter, set Nonlinear polynomial type to `Even and odd order`.

`IP3` ⌘ Third-order intercept point

`inf dBm` (default) | scalar

Third-order intercept point, specified as a scalar in `dBm`, `W`, `mW`, or `dBW`. 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 .

References

[1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.

[2] Grob, Siegfried and Lindner, Jurgen, òPolynomial Model Derivation of Nonlinear Amplifiersó, *Department of Information Technology*, University of Ulm, Germany.

See Also

IQ Demodulator | Mixer

Topics

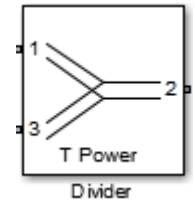
òModulate Two-Tone DC Signal Using IQ Modulatoró

Introduced in R2017a

Divider

Model ideal frequency-independent dividers (combiners) with S-parameters

Library: RF Blockset / 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:

Å `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:

Å $s_{11} = (z_{23\,\delta}z_1) / (z_{23} + z_1)$

- $s_{22} = (z_{13} \delta z_2) / (z_{13} + z_2)$
- $s_{33} = (z_{12} \delta z_3) / (z_{12} + z_3)$
- $s_{21} = (1+ s_{11}) * sqrt(z_1 / z_2)$
- $s_{31} = (1+ s_{11}) * sqrt(z_1 / z_3)$
- $s_{32} = (1+ s_{22}) * sqrt(z_2 / z_3)$
- $z_{12} = z_1 * z_2 / (z_1 + z_2)$
- $z_{13} = z_1 * z_3 / (z_1 + z_3)$
- $z_{23} = z_2 * z_3 / (z_2 + z_3)$
- 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.

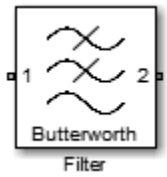
See Also

Circulator | Coupler

Introduced in R2014a

Filter

Model RF filter



Library

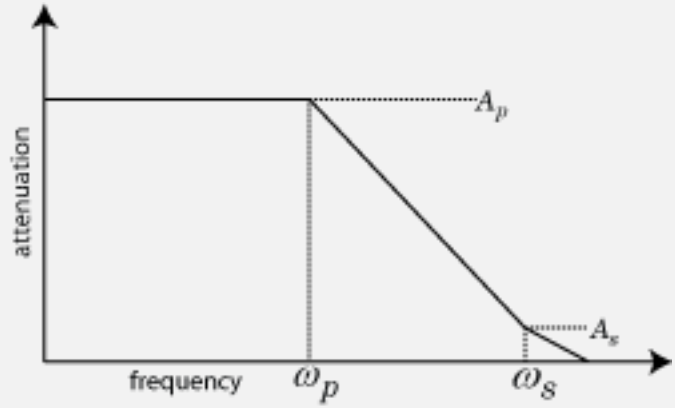
Elements

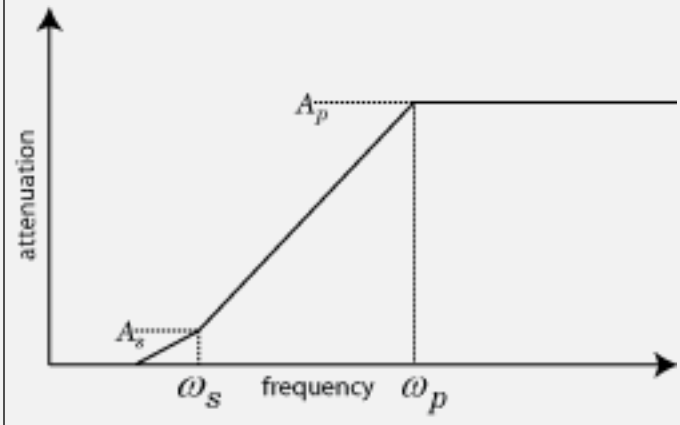
Description

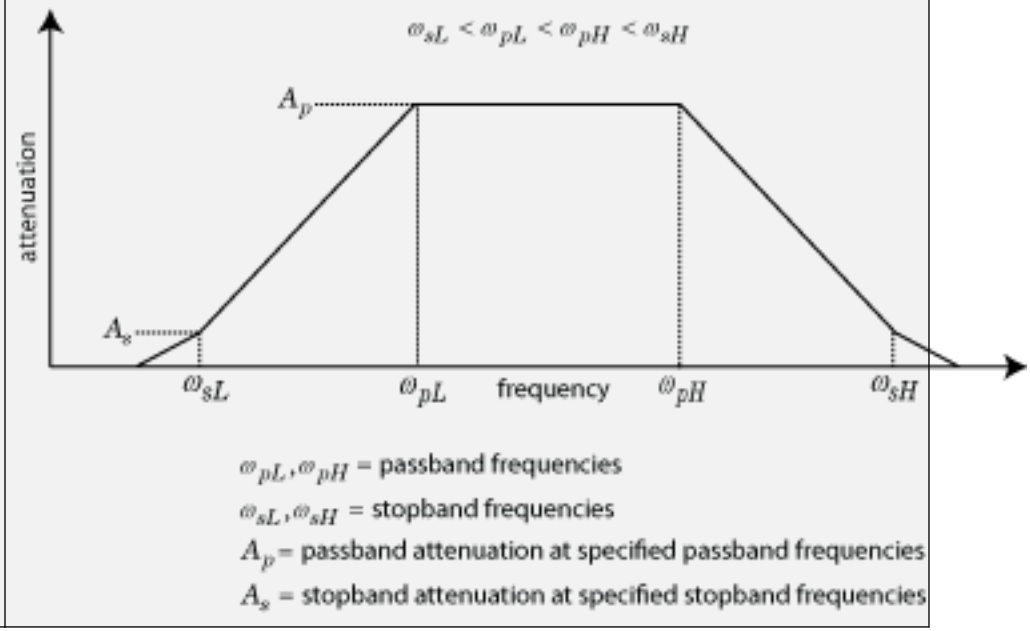
The Filter block models RF filters of two designs:

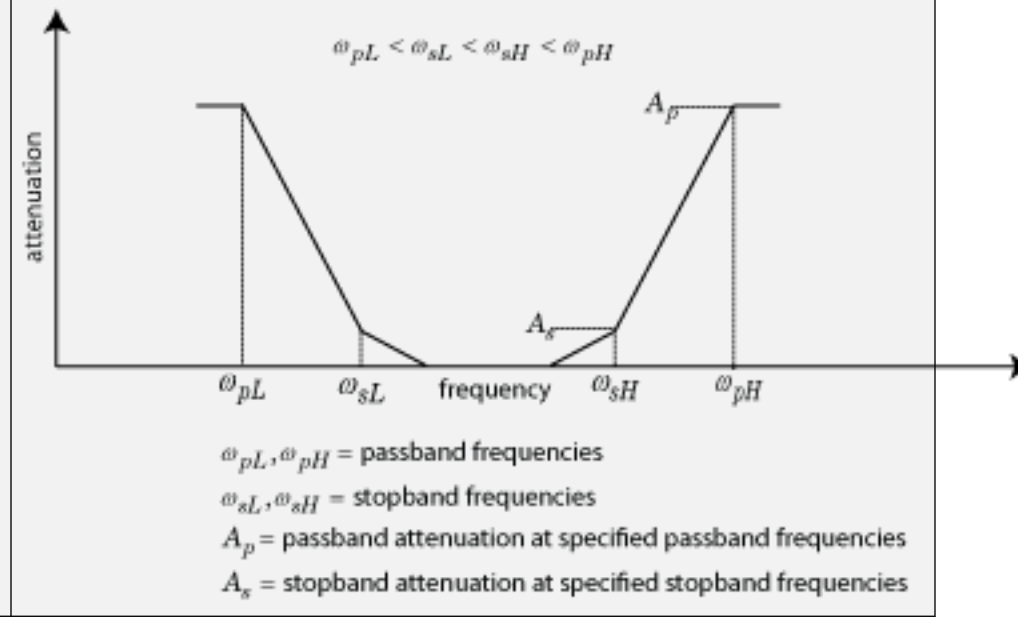
- Butterworth: Butterworth filters have a magnitude response that is maximally flat in the passband and monotonic overall. This smoothness comes at the price of decreased roll-off steepness.
- Chebyshev: Chebyshev Type I filters have equal ripple in the passband and monotonic in the stopband.

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	

Filter Type	Frequency Response
Bandstop	

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 2ðtuple 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 2ðtuple 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, f_p, A_p	Order, f_p, A_p	Order, f_{pL}, f_{pH}, A_p	Order, f_{sL}, A_p, f_{sH}, A_s
	f_p, A_p, f_s, A_s	f_p, A_p, f_s, A_s	(f_{pL}, f_{pH}), A_p, (f_{sL}, f_{sH}), A_s	(f_{pL}, f_{pH}), A_p, (f_{sL}, f_{sH}), A_s
Chebyshev	Order, f_p, A_p	Order, f_p, A_p	Order, f_{pL}, f_{pH}, A_p	Order, f_{sL}, A_p, f_{sH}, A_s
	f_p, A_p, f_s, A_s	f_p, A_p, f_s, A_s	(f_{pL}, f_{pH}), A_p, (f_{sL}, f_{sH}), A_s	(f_{pL}, f_{pH}), A_p, (f_{sL}, f_{sH}), A_s
	Order = filter order, f_p = passband frequency, A_p = passband attenuation in dB, f_{pL}, f_{pH} = passband frequencies, f_{sL}, f_{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

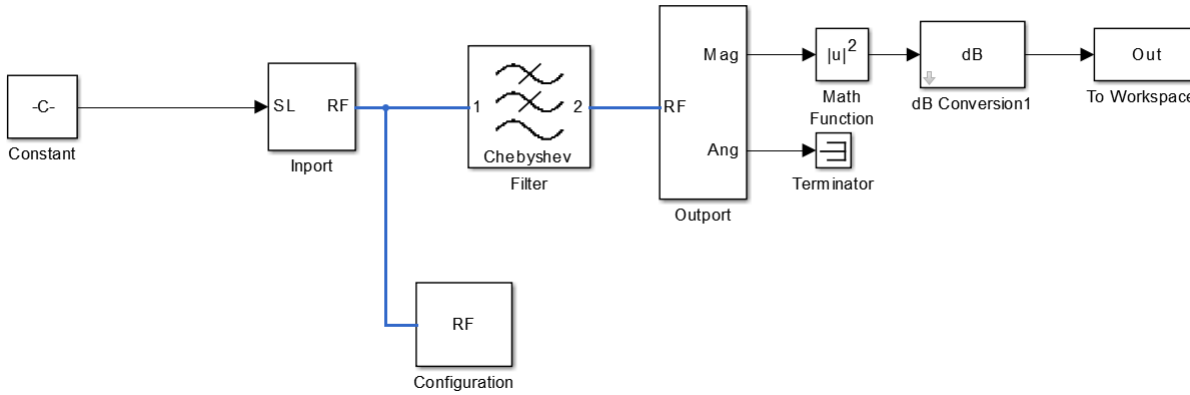
Specify the scale for the x-axis. The default value is `Linear`.

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.

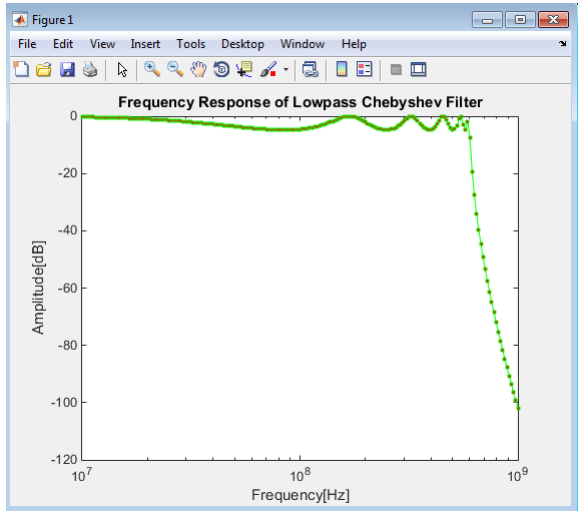
6 The Filter block output signal is fed into the Outport. The output block is configured to give both the magnitude and angle of the signal. The angle output is terminated with the terminator block.

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')
```

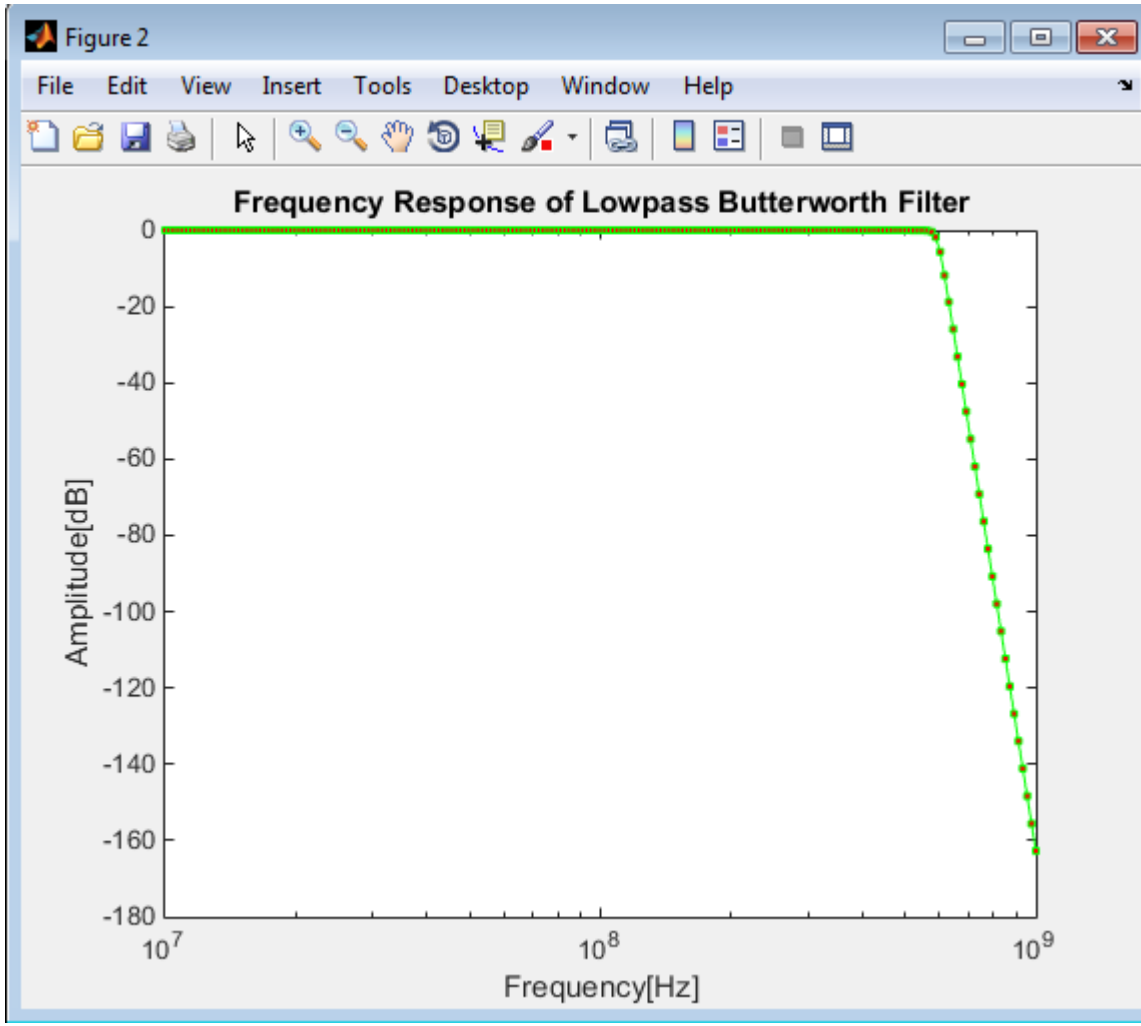
9 Notice the ripples in the passband of the chebyshev filter.



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.

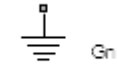
[3] Larry D. Paarmann, *Design and Analysis of Analog Filters, A Signal Processing Perspective with MATLAB Examples*, Kluwer Academic Publishers, 2001.

[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: RF Blockset / Circuit Envelope / Utilities



Description

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

Parameters

The Ground block has no parameters.

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

See Also

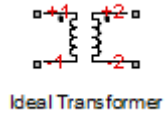
Introduced in R2016b

Ideal Transformer

Model ideal transformer

Library

Elements



Description

The Ideal Transformer block models a constant IV relationship within the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

Parameters

Winding ratio

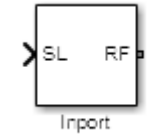
Specify the winding ratio.

See Also

Mutual Inductor | Three Winding Transformer

Inport

Convert Simulink input signal to RF Blockset signal
Library: RF Blockset / Circuit Envelope / Utilities



Description

The Inport block imports Simulink signals into the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

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 input port converts the complex simulink input signals into an Rf signal suitable for multicarrier simulation:

$$out = \mathrm{Re}\,al(\sum_{k=1}^{N}(I_k(t) + j \cdot Q_k(t)) \cdot e^{j 2\pi f_k t})$$

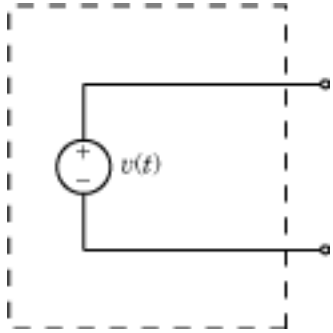
The Source type parameter specifies the Simulink signal as either current, or voltage, or power source.

Parameters

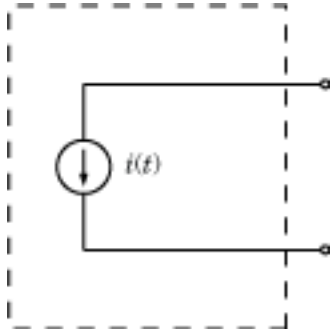
Source type ¬ Inport block interpretation of Simulink signal
`Ideal voltage` (default) | `Ideal current` | `Power`

Inport block interpretation of Simulink signal, specified as:

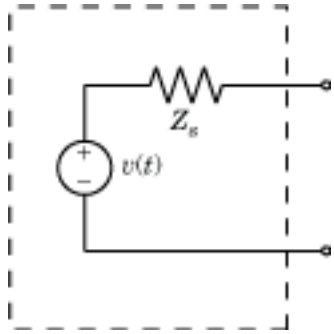
- `Ideal voltage` ñ The block outputs Simulink signals as voltage signals $v(t)$ in the RF Blockset environment. When you choose an ideal voltage input port you need to manually add a series of source impedance to match the blocks connected to the input port. The following figure illustrates the internal configuration of the block.



- `Ideal current` ñ The block outputs Simulink signals as current signals $i(t)$ in the RF Blockset environment. When you use an ideal current input port, you manually add a parallel source impedance to match the blocks connected to the input port. The following figure illustrates the internal configuration of the block.



- `Power` ñ The block interprets the Simulink signals, $P_v(t)$, as available power and internally uses a voltage source, and series impedance. You use this option to import a 50 ohm environment or different reference impedance signals created using Communications System Toolbox. When you select this option, the input port automatically inserts a source impedance in your circuit 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

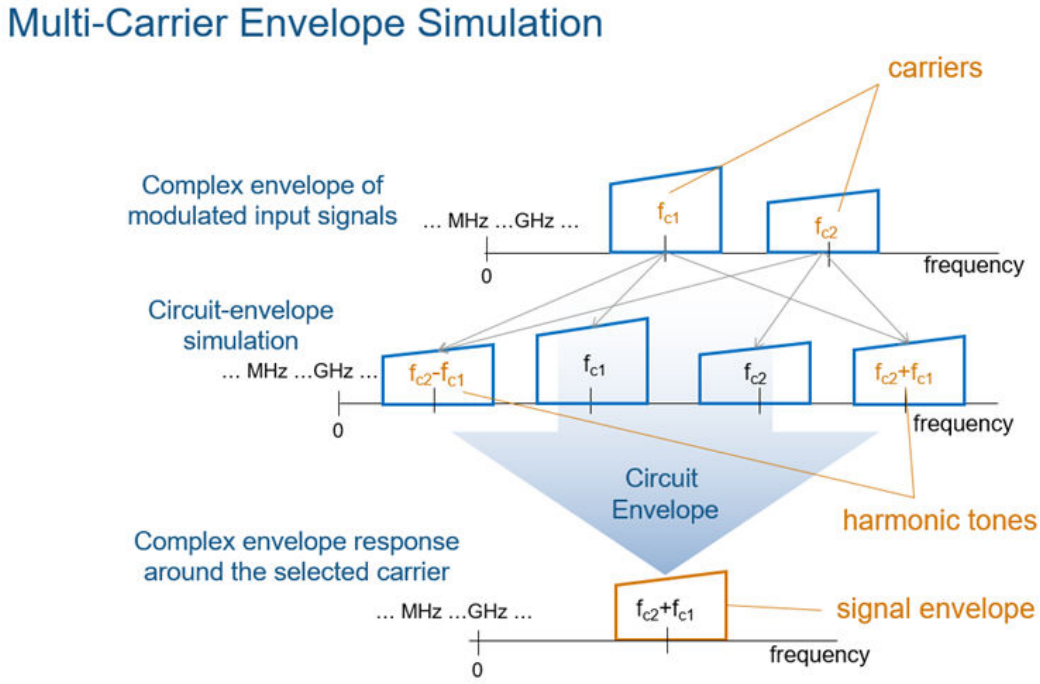
Definitions

Multi-Carrier Envelope Simulation

Using the Inport block you can specify the complex envelopes of your input signals and import them as RF signals for multi-carrier simulation.

The Configuration block automatically determines the fundamental tones specified in the input ports and proposes a suitable harmonic order to capture the non-linearity of the system. You can also manually specify the harmonic order for each fundamental tone in the simulation.

In the input port, you can specify as many carrier frequencies as you want. It is recommended that you trade off the simulation bandwidth (inversely proportional to the simulation time step) and the total number of simulation frequencies.

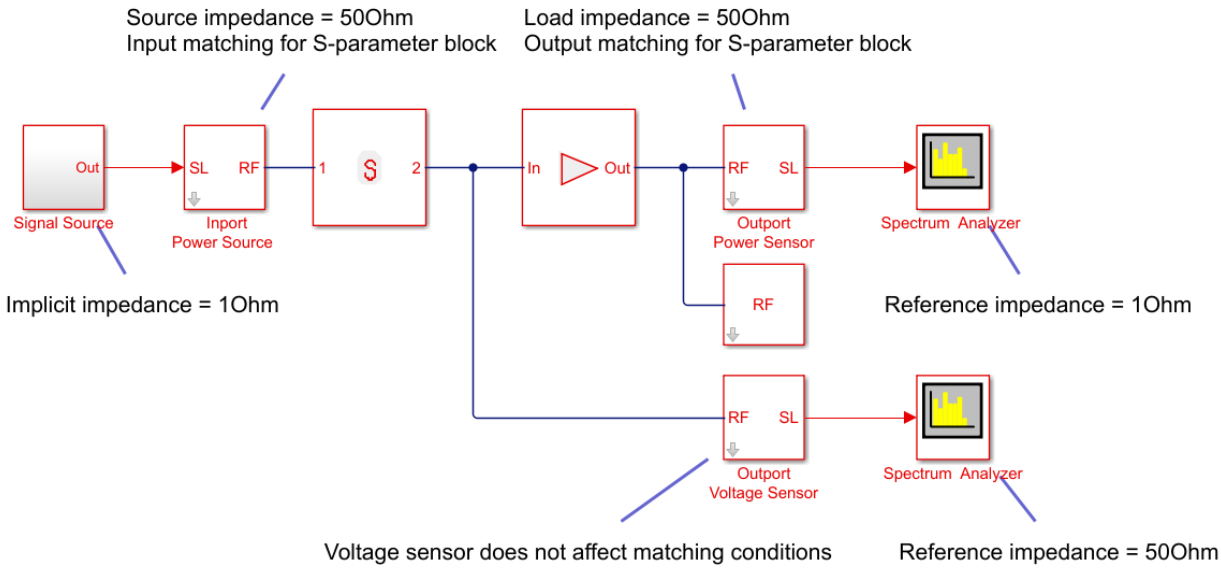


Algorithms

How to Use Inport Block

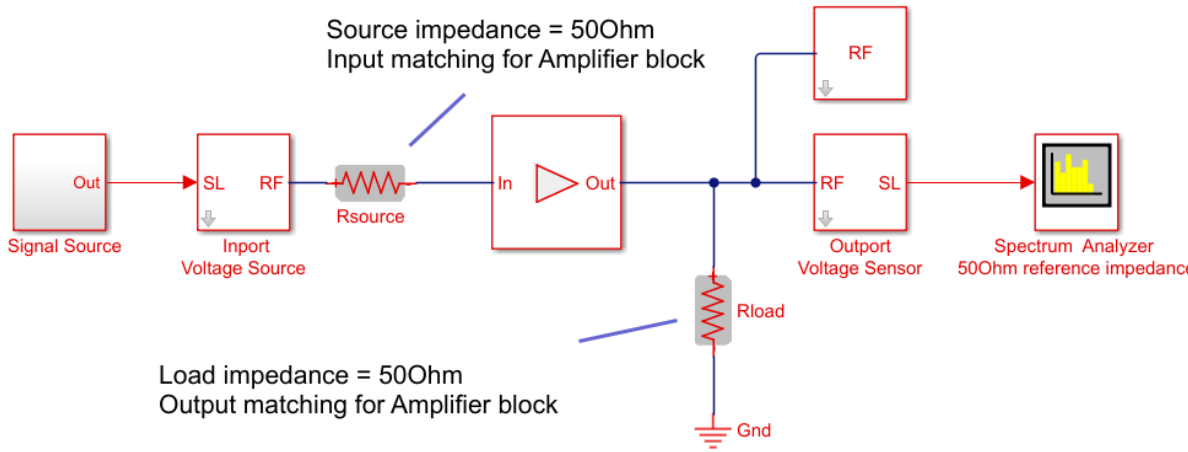
The Inport block allows you to specify the complex envelopes of your input signals and import them as RF signals for multi-carrier simulation.

The power option automatically insert a source or load impedance in your network, and normalizes the signal power with respect to the specified impedance. You do not need to manually insert source and load terminations, and your signals are automatically scaled between RF Blockset and the Simulink environment that assumes an implicit 1 Ohm reference impedance.



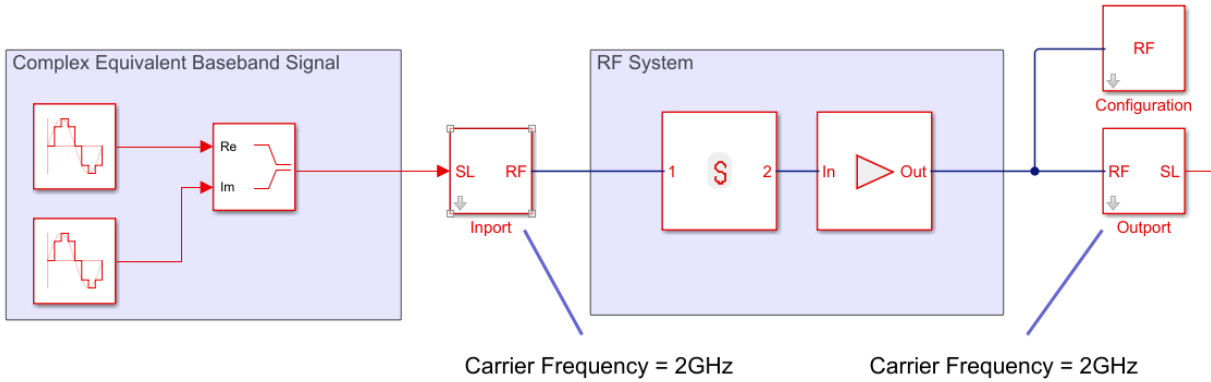
When using voltage sources and sensors, manually add source and load terminations, otherwise there might be an undesired impedance mismatch in your network. When you measure the power of a voltage signal, make sure that you use a 50Ohm reference impedance.

If you use an ideal voltage source and add a source impedance, in perfectly matched conditions, the actual voltage applied to the first block of the RF chain is half of the value of the input Simulink signal. The source impedance and the input impedance of the first block of the RF chain form a voltage divider network.



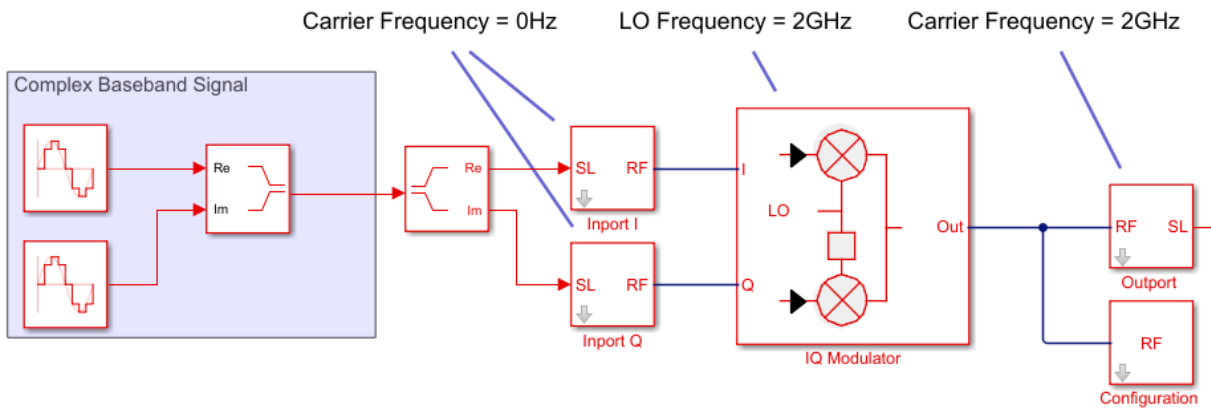
Input signal is a digital communication complex equivalent baseband signal (I,Q). You assume an implicit carrier for the system that is equal to the carrier frequency, F_c. You want to model RF effects such as amplifier nonlinearity and S-parameter filters using RF Blockset:

- Enter F_c in the Carrier Frequencies parameter.
- The simulation step size in the configuration block is the same as the sample time of the Simulink input signal, and it is not related to the carrier frequency.
- If the RF chain does not include any modulator or demodulator, use an Outport block at the end of the chain. You can use the `Outport` block to probe the complex equivalent signal centered on F_c.



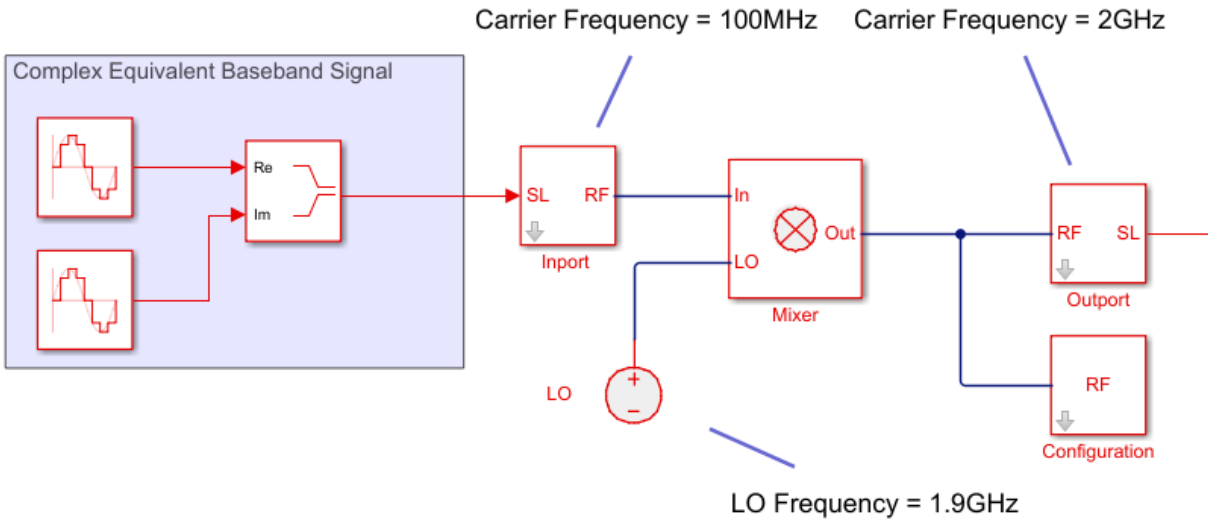
Input signal is a digital communication complex input baseband signal (I,Q). You assume that no carrier is associated with the input signal. You want to upconvert the signal to F_c and model RF effects such as amplifier nonlinearity and S-parameter filters:

- Use to two Inport blocks for the I and Q components of the input signal. Set the Carrier Frequencies parameter of each Inport block to 0
- To upconvert the signal, use an IQ Modulator block. Set the Local oscillator frequency to F_c.
- The simulation step size in the configuration block is the same as the sample time of the Simulink input signal, and it is not related to the Local Oscillator frequency.
- Use an Outport block at the end of the chain. and probe the signal at F_c.



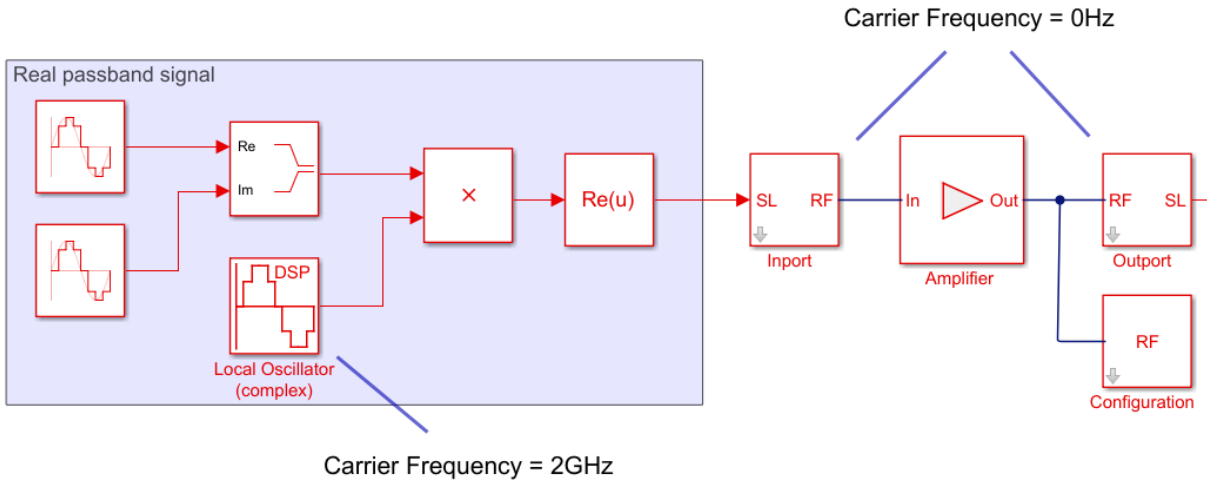
Input signal is a digital communication complex equivalent baseband signal (I,Q). You want to first upconvert the signal to intermediate frequency (IF), then to RF, and model RF imperfections:

- Set the Carrier Frequencies parameter of each Inport block to `IF`.
- Use the Mixer block. Set the LO carrier frequency to `RF-IF`.
- Use an Outport block at the end of the chain. and probe the signal at RF.



Input signal is a digital or analog real passband signal that is explicitly modulated to high frequency in Simulink domain:

- Set the Carrier Frequencies parameter of each Inport block to `0` and simulate RF effects.
- The simulation step size in the configuration block is the same as the sample time of the Simulink input signal, and it is proportional to the RF frequency.
- However there is no speed benefit in using RF Blockset for real-passband simulation. This option is not recommended



See Also

Configuration | Outport

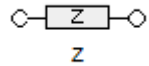
Introduced in R2010b

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, é]. 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, é]. 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

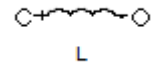
The example, òFrequency Response of an RF Transmit/Receive Duplex Filteró, simulates an analog RF filter comprising Capacitor, Inductor, Resistor, and Impedance blocks.

See Also

Capacitor | Inductor | Resistor

Inductor

Model inductor for circuit envelope analysis
Library: RF Blockset / Circuit Envelope / Elements



Description

The Inductor block models an inductor in circuit envelope environment.

Parameters

`Inductance` ⌘ Inductance value
`1e-9 H` (default) | real number

Inductance value, specified as a real number. Specify the units of the capacitance from the corresponding drop-down menu.

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. By default, the value is `1e-9 H`.

Model Examples

See Also

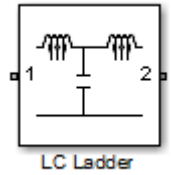
Capacitor | Resistor

LC Ladder

Model LC ladder networks

Library

Elements



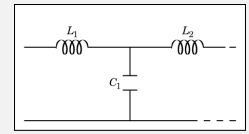
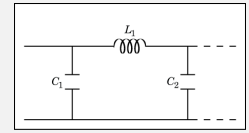
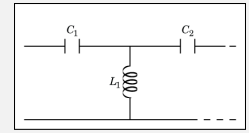
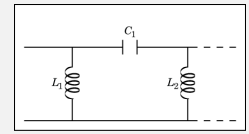
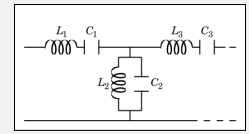
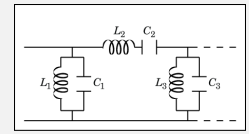
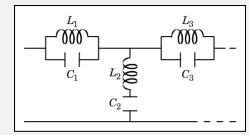
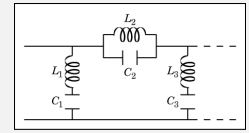
Description

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.

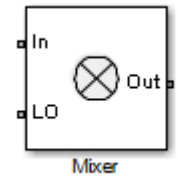
By default, this option is selected.

Examples

Å The section, òFilter Mixing Productsó, shows the use of a LC Ladder.

Mixer

Model mixer in RF systems
Library: RF Blockset / Circuit Envelope / Elements



Description

The Mixer block performs signal frequency translation and nonlinear amplification.

For a given RF input signal $V_{RF} = A_{RF}\cos(Ş_{RF}t)$ and an LO input signal $V_{LO} = A_{LO}\cos(Ş_{LO}t)$, 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 $Ş_{RF}$ + $Ş_{LO}$ and $Ş_{RF}$ ð $Ş_{LO}$. For the mixer to perform this operation correctly, you must include the frequencies $Ş_{RF}$ + $Ş_{LO}$ or $Ş_{RF}$ ð $Ş_{LO}$ 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` ñ 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` ñ 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` ñ 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, é , a_9], the polynomial it represents is $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \quad + a_9 V_{in}^9$. 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 $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + + a_3 V_{in}^3$.

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

$V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \ldots + a_9 V_{in}^9$
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 $f_{IM} = f_{LO} + (f_{LO} - f_{RF})$.

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.

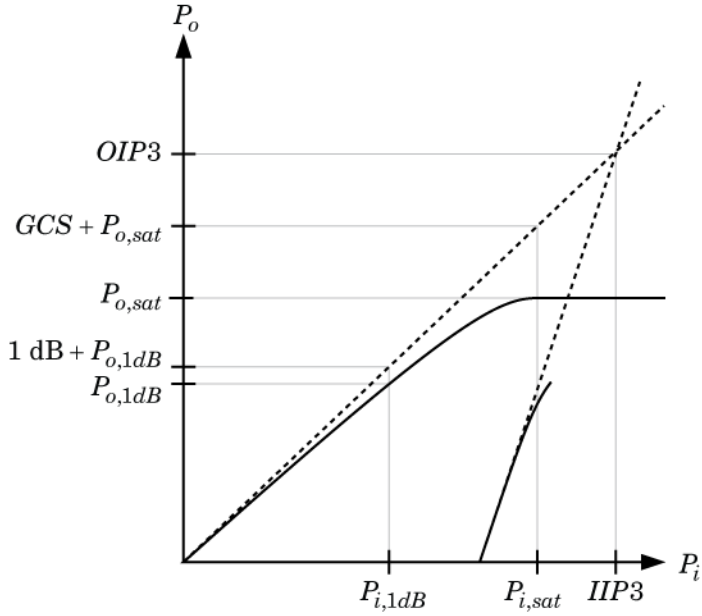
Nonlinearity

`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, òPolynomial Model Derivation of Nonlinear Amplifiersó, *Department of Information Technology*, University of Ulm, Germany.

See Also

Amplifier | S-Parameters

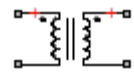
Introduced in R2010b

Mutual Inductor

Model two coupled inductors for circuit envelope analysis

Library

Elements



Mutual Inductor

Description

The Mutual Inductor block models an inductor within the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

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.

RF Blockset 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`.

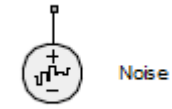
Note The minimum nonzero inductance value that the RF Blockset environment recognizes is `1e-18 H`. During simulation, the block uses a value of `1e-18 H` for any inductance and mutual inductance values specified between `0` and `1e-18 H`.

See Also

Inductor | Three-Winding Transformer

Noise

Model noise using current or voltage noise source in RF systems
Library: RF Blockset / 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

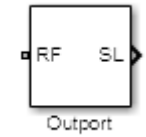
See Also

Continuous Wave | Sinusoid

Introduced in R2010b

Outport

Convert RF Blockset signal to Simulink output signals
Library: RF Blockset / Circuit Envelope / Utilities



Description

The Outport block outputs carrier modulation signals in the RF Blockset circuit envelope simulation environment as Simulink signal. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

The output port senses current and voltage complex envelope or real passband signals. Complex baseband signals consist of in-phase (I_k) and quadrature (Q_k) components centred around each specified centre frequency f_k.

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` ñ The block outputs the modulations of the voltage signal at the specified Carrier frequencies in the format specified by the Output parameter. This is the recommended option to sense a signal without adding a loading impedance and changing matching conditions.
- `Ideal current` ñ The block outputs the modulations of the current signal at the specified Carrier frequencies in the format specified by the Output parameter.

Å `Power` ñ The block outputs the modulations of the voltage signal at the specified Carrier frequencies and scaled with respect to the specified load impedance. This is the recommended option to sense a signal generated in RF Blockset 50 Ohm environment or a different reference impedance. When you use the power option , the output port automatically inserts a load impedance in your circuit.

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

where Z_l is the value of the Load impedance (ohms) parameter.

`Load impedance (Ohm)` ⌘ Load impedance of RF circuit

`inf` (default) | vector of positive integers in ohms

Load impedance of RF circuit used to measure signal power, specified as a vector of positive integers in ohms. When you use the power option , the output port automatically inserts a load impedance in your circuit. 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:

Å `Complex Baseband` ñ The block outputs a vector of complex-valued signals $I_k(t) + j \cdot Q_k(t)$ at the port labeled SL. The *k*th element of the vector is the *k*th frequency specified by the Carrier frequencies parameter.

Å `In-phase and Quadrature Baseband` ñ 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 *k*th element of the vector is the *k*th frequency specified by the Carrier frequencies parameter. The quadrature component of a signal with carrier frequency equal to 0 Hz is zero.

Å `Magnitude and Angle Baseband` ñ 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 limited.

$$out = \text{Re}\,al(\sum_{k=1}^{N}(I_k(t) + j \cdot Q_k(t)) \cdot e^{j 2\pi f_k t})$$

where t is the value of Load impedance (ohms) parameter.

`Automatically compute output step size` — Determine optimal time step to resolve highest listed carrier frequency

`on` (default) | `off`

Select this parameter to allow RF Blockset 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

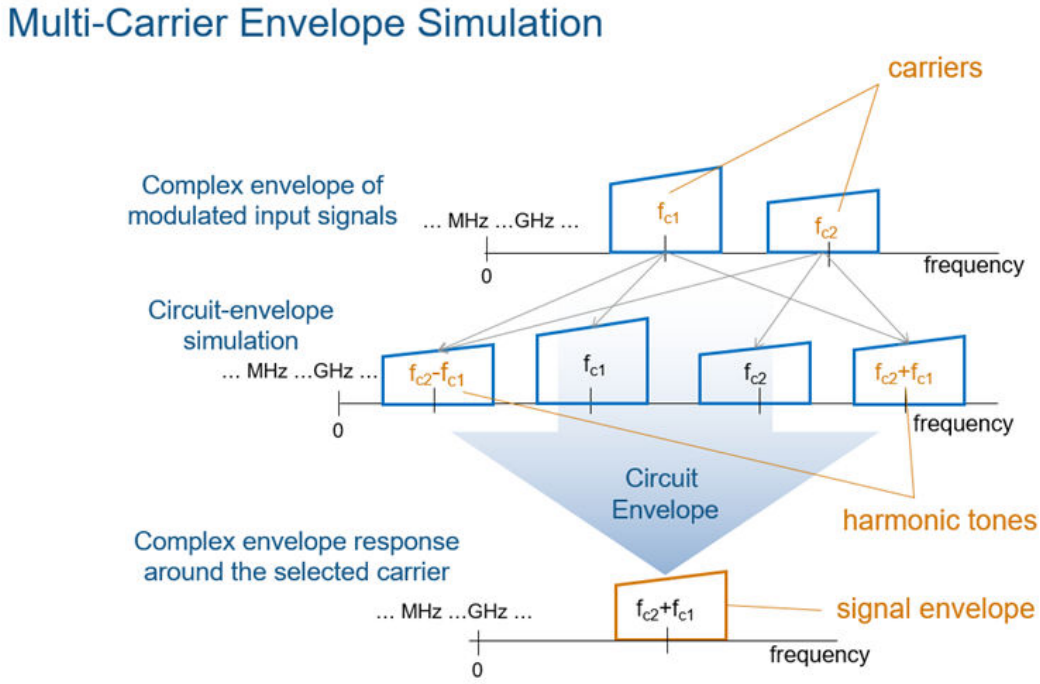
Definitions

Multi-Carrier Envelope Simulation

Using the Inport block you can specify the complex envelopes of your input signals and import them as RF signals for multi-carrier simulation.

The Configuration block automatically determines the fundamental tones specified in the input ports and proposes a suitable harmonic order to capture the non-linearity of the system. You can also manually specify the harmonic order for each fundamental tone in the simulation.

In the input port, you can specify as many carrier frequencies as you want. It is recommended that you trade off the simulation bandwidth (inversely proportional to the simulation time step) and the total number of simulation frequencies.



Real Passband Formula

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

- When this option is selected, RF Blockset interprets complex envelope I+jQ signal for the k^{th} 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 k^{th}

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

- 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.

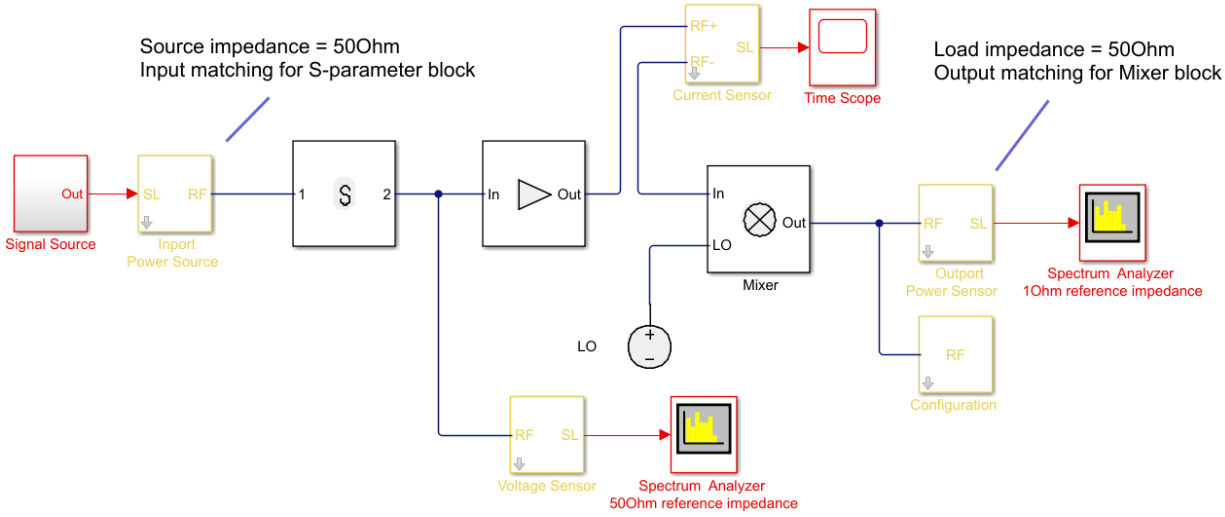
Algorithms

How to Use Outport Block

Consider an RF chain composed by several stages. You want to sense and inspect the signal behavior at different intermediate nodes.

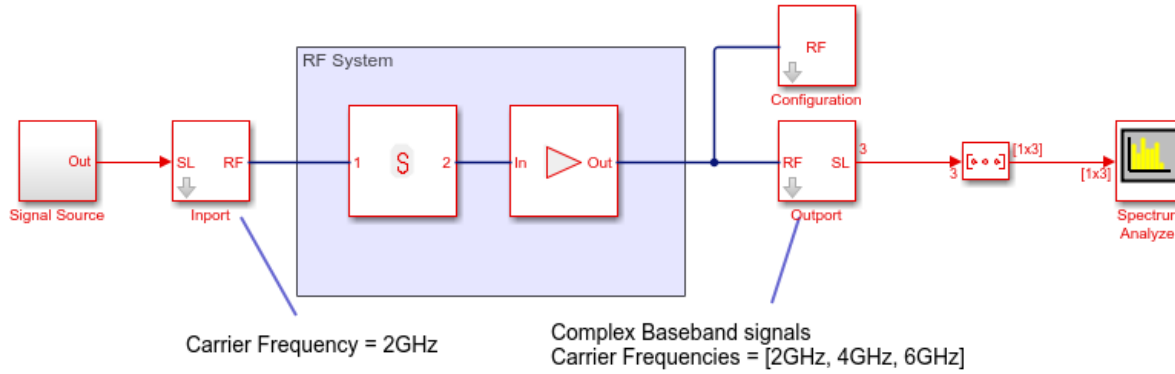
Use the power input port and power sensor port at the input and output of your chain, as gateways between Simulink (reference 1Ohm) and the RF domain.

Use the voltage or current output ports connected to the intermediate nodes and branches that you want to inspect. To measure the power of a voltage signal, change the default spectrum analyzer reference impedance to 50Ohm.



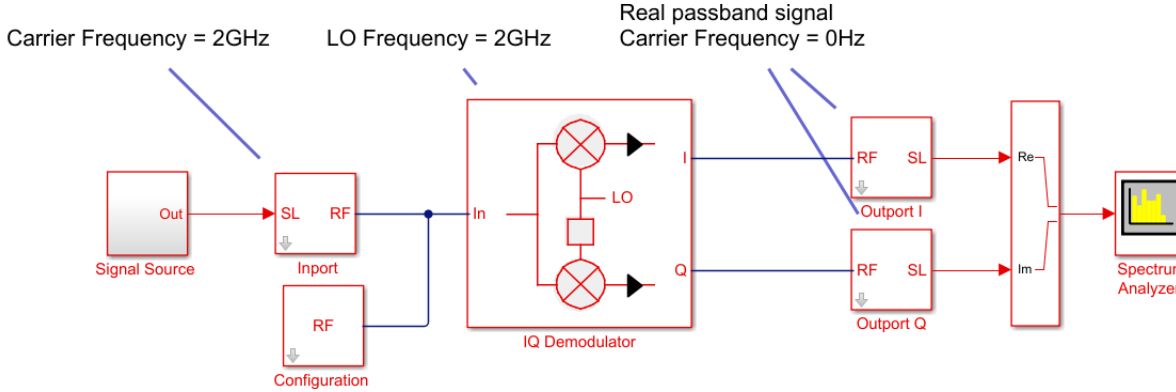
Output signal is a combination of (digital communication) complex equivalent baseband signals (I,Q). For each envelope, assume an implicit centre frequency for the signal that is equal to the carrier frequency,F_c.

- Enter the array F_c in the Carrier Frequencies parameter corresponding to the centre frequency of the envelopes that you want to sense. The output signals is an array of complex (I,Q) envelopes.
- The output port of sensor type power provides a termination (by default 50Ohm) at the end of the chain.
- The simulation step size in the configurationConfiguration block is the same as the sample time of the Simulink signal, and it is not related to the carrier frequency.



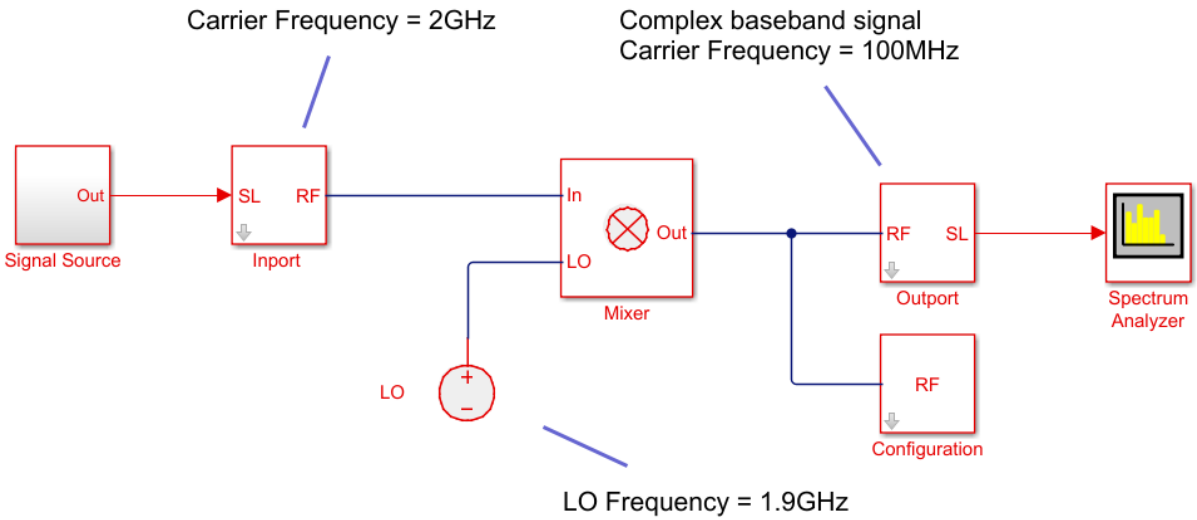
Output signal is a (digital communication) complex baseband signal (I,Q) resulted from a direct conversion receiver. Assume that no carrier is associated with the output signal.

- Use two Outport blocks for the I and Q components of the signal. Set the Carrier Frequencies parameter of each Outport block to `0`. Use the real passband option to sense a real signal rather than a complex signal with quadrature component equal to `0`.
- The output ports of sensor type power provide a termination (by default 50Ohm) at the end of the chain.
- To down-convert the signal, use an IQ Demodulator block. Set the Local oscillator frequency to F_c equal to the centre frequency of the input signal (direct conversion).
- The simulation step size in the configuration block is the same as the sample time of the Simulink signal, and it is not related to the Local Oscillator frequency.



Output signal is a (digital communication) complex equivalent baseband signal (I,Q) down-converted to intermediate frequency (IF). Apply digital signal processing techniques for processing the complex equivalent baseband information of the signal.

- Set the Carrier frequencies parameter of the Outport block to `IF`. Use the complex baseband option.
- Use the Mixer block for down-conversion. Set the LO carrier frequency to `LO = RF-IF`. The output port will behave as an ideal filter and select only the down-converted signal. The envelope at `RF+LO = 2RF-IF` frequency is simulated but it is not sensed by the output port.
- The simulation step size in the configuration block is the same as the sample time of the Simulink input signal, and it is not related to the IF, RF or LO frequencies.



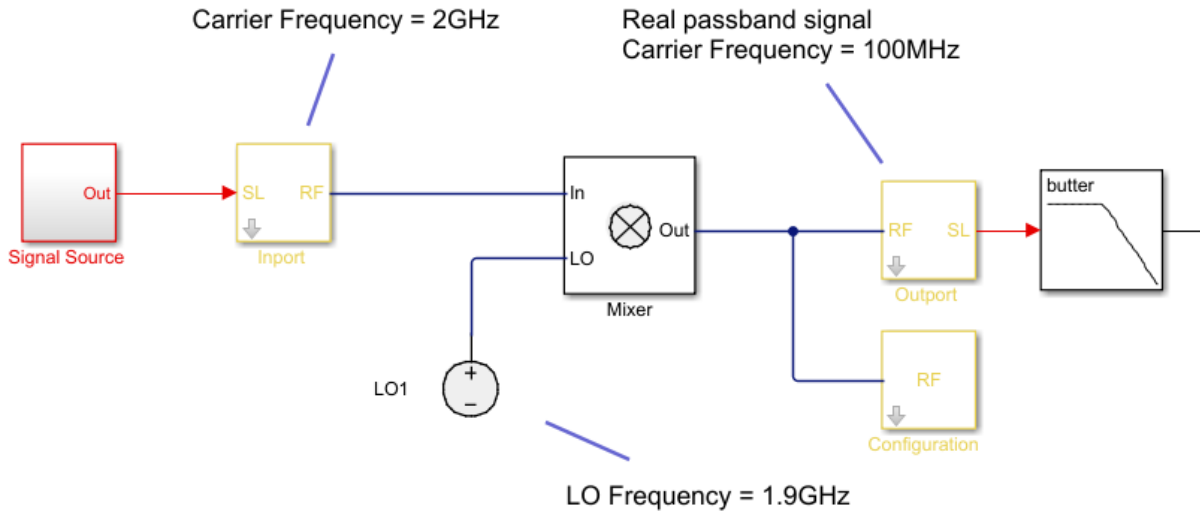
Output signal is a (digital communication) signal (I,Q) down-converted to intermediate frequency (IF). Apply analog signal processing techniques to the output signal.

- Set the Carrier frequencies parameter of the Output block to `IF`. Use the real passband option.

If the intermediate frequency is within the simulation bandwidth defined in the configuration block, Use the same step size in the output without the need for resampling the signal.

If the intermediate frequency is not within the simulation bandwidth defined in the configuration block, you need to resample the signal (as described in the real-passband formula) to correctly resample the carrier.

- Use the Mixer block. Set the LO carrier frequency to `LO = RF-IF`. The output port will behave as an ideal filter, and select only the down-converted signal. The envelope at `RF+LO = 2RF-IF` frequency is simulated but it is not sensed by the output port.
- The simulation step size in the configuration block is the same as the sample time of the Simulink input signal. All the blocks in the RF Blockset network connected to the configuration block are executed with the same step size. The time step of the outport, when the real passband option is selected, might differ from the time step of the input Simulink signal and of the time step set in the configuration block.



See Also

Configuration | Inport

Introduced in R2010b

Phase Shift

Model phase shift in RF systems

Library

Elements

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

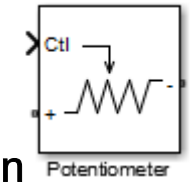
The example, òMeasuring Image Rejection Ratio in Receiversó, shows how to use Phase Shift blocks to model an LO phase offset in a receiver architecture.

Potentiometer

Simulink controlled potentiometer

Library

Junctions



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 voltageðcurrent relationships when the control signal varies between zero and one are:

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

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

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

- Linear, Logarithmic, Antilog ñ Electrical characteristic of the potentiometer.
- `Ctl` ñ Simulink signal
- R_{on} ñ Minimum resistance of the potentiometer
- R_{off} ñ 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` ñ Greater than 0 ohms and less than 50 ohms. The default is 20.
- `Antilog` ñ Greater than 50 ohms and less than 100 ohms. The default is 80.

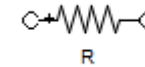
See Also

SPDT | SPST | Switch

Introduced in R2015b

Resistor

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



Description

The Resistor block models a resistor within the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

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.

RF Blockset 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 kT/R A^2/Hz, where:

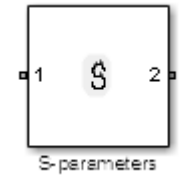
- k is the Boltzmann constant
- T is the value of the Temperature parameter, in degrees Kelvin. (Also located in the Configuration block.)

See Also

Capacitor | Inductor

S-Parameters

Model S-parameter network
Library: RF Blockset / Circuit Envelope / Elements



Description

The S-parameters block models a network defined by S-parameters in the RF Blockset circuit envelope simulation environment. The device can have up to eight ports. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

The block models S-parameter data in the RF Blockset 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` ñ Name of a Touchstone file with the extension `.s2p`. The block ignores noise and nonlinearity data in imported files.
- `Network-parameters` ñ Provide Network parameter data such as `S-parameters`, `Y-parameters`, and `Z-parameters` with corresponding Frequency and Reference impedance (ohms) for the s-parameters.

Å `Rational model` ñ 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)` ממ 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` ממ 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` ממ 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` ממ 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` ממ 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, òGenerate Thermal Noiseó on page 1-133.

`Ground and hide negative terminals` ם **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` ם **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 òCompare Time and Frequency Domain Simulation Options for S-parametersó.
- 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` ם **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)` مم 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` مم 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` مم 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`.

`Use only S-parameter magnitude with appropriate delay` مم Use only S-parameter magnitude with appropriate delay

`off` (default) | `on`

Select this parameter to ignore the s-parameter phase and delay the impulse response by half its length. This parameter is applicable only for S-parameter data modeled in time

domain. You can use this to shape spectral content with filter effects by specifying only magnitude.

Note This parameter introduces an artificial delay to the system.

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`

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` مم Type of S-Parameters to plot
`S11` (default) | `SNN`

Type of S-Parameters to plot, specified as `SNN`, where *N* is the number of ports in the s-parameters block.

`Parameter 2` مم Type of S-Parameters to plot
`None` (default) | `SNN`

Type of S-Parameters to plot, specified as `SNN`, where *N* is the number of ports in the s-parameters block.

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

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

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

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

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

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

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

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

`Plot` مم Plot specified data
button

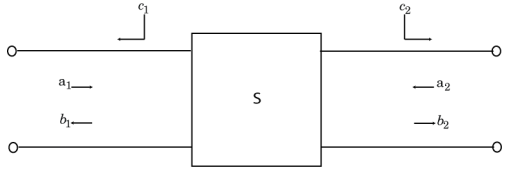
Plot specified data using plot button.

Model Examples

Definitions

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. 2004ð2012, Nov. 1992.

See Also

Amplifier | Mixer | Transmission Line

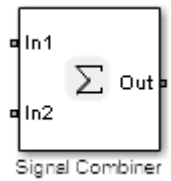
Introduced in R2010b

Signal Combiner

Compute sum of RF signals

Library

Elements



Description

Use the Signal Combiner block to sum signals across each carrier frequency in the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó

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

Å The example, òMeasuring Image Rejection Ratio in Receiversó shows how to use a signal combiner to perform image rejection.

Å The example, òCarrier To Interference Performance of Weaver Receiveró uses a Signal Combiner block as part of a realization of the Weaver receiver architecture.

See Also

Sinusoid

Model DC offset and sinusoidal modulation
Library: RF Blockset / 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 k^{th} listed block carrier:

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

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

where:

- D_i and D_q are DC offsets.
- A_i and A_q are in-phase and quadrature amplitudes.
- $\hat{r}$ is the time delay.
- $Ş_k$ 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` ñ The block simulates a voltage envelope $v(t)$ at the specified Carrier frequencies.
- `Ideal Current` ñ The block simulates a current envelope $i(t)$ at the specified Carrier frequencies.

`Offset in-phase` ם 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` ם 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` ם 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` ם 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` ם 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` مم **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` مم **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` مم **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

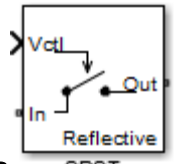
Introduced in R2010b

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 voltageðcurrent 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} Ó$V_{threshold}$

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

- G_{on}, G_{off} ñ On and off path conductances
- G_{Z01}, G_{Z02} ñ 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 | SPDT | Switch

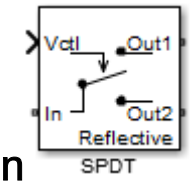
Introduced in R2015b

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

Å If $V_{ctl} < V_{threshold}$:

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

Å If V_{ctl} Ó $V_{threshold}$:

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

- I_{in}, I_1 and I_2 ñ Currents into the input terminal and two output terminals
- V_{in1}, V_{in2} ñ Voltages between the input terminal and the output 1 and output 2 terminals, respectively
- V_1, V_2 ñ Voltages at output port 1 and output port 2, respectively
- G_{on}, G_{off} ñ On and off path conductances
- G_{Z01}, G_{Z02} ñ 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 | SPST | Switch

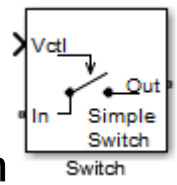
Introduced in R2015b

Switch

Simulink controlled twoðterminal switch

Library

Junctions



Description

The switch block models a two terminal switch. The input Simulink signal, V_{ctl} , controls the transfer of the RF Blockset 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 voltageðcurrent relationship for the switch, or controlled resistor:

Å If $V_{ctl} < V_{thres}$:
$$I_{res}.R_{off} = V_{res}$$

Å If V_{ctl} ÓV_{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 | SPDT | SPST

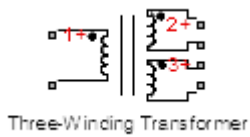
Introduced in R2015b

Three-Winding Transformer

Model three coupled inductors for circuit envelope analysis

Library

Elements



Description

The Three-Winding Transformer block models three coupled inductors within the RF Blockset circuit envelope simulation environment. For an introduction to RF simulation, see the example, òSimulate High Frequency Componentsó.

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.

Å $i_1(t)$, $i_2(t)$, 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.

RF Blockset 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 RF Blockset 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

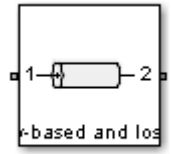
Inductor | Mutual Inductor

Transmission Line

Model transmission line

Library

Elements



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

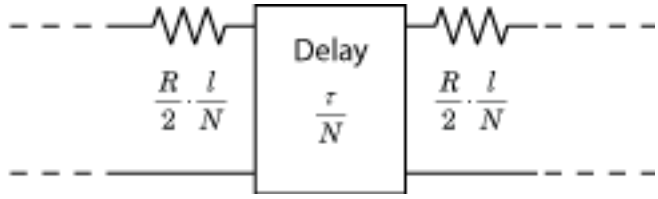
    Specify the delay and select the units. The default value is `4.7e-9 s`.

- 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

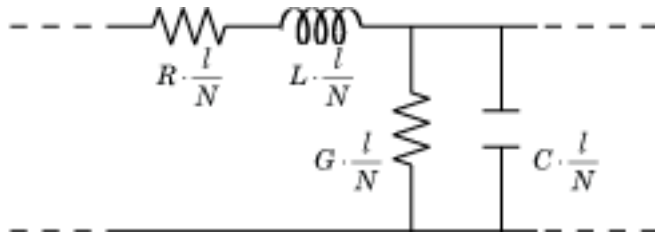
  Specify the physical length of the transmission line *l* and select the units. The default length is `1 cm`.

- 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:

Å 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:

Å Characteristic impedance

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

Å Resistance per unit length

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

Å Capacitance per unit length

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

Å Conductance per unit length

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

Å 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

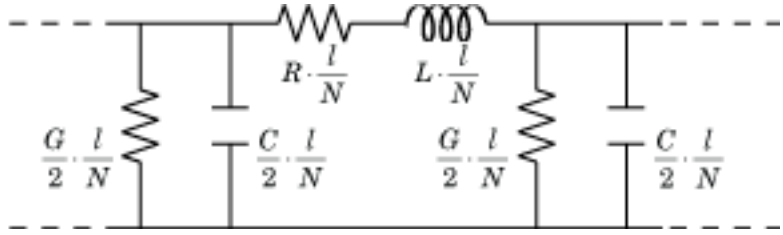
  Specify the physical length of the transmission line *l* and select the units. The default length is `1 cm`.

- 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`.

  - 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

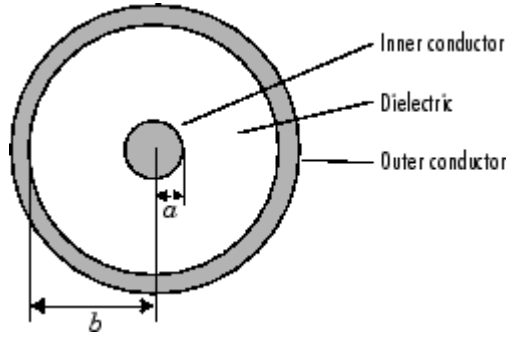
  Specify the physical length of the transmission line *l* and select the units. The default length is `1 cm`.

- 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, $\hat{\imath}_0$. 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, $\ddot{E}_0$. 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 òTransmission Line with Stubó on page 1-168 for more information.

  The following auxiliary equations are used for ABCD-parameter calculations. For general information, see òParameter Calculations for Distributed Transmission Lineó on page 1-168.

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

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

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.
- $\hat{R}_{cond}$ is the conductivity in the conductor.
- $\hat{\imath}$ is the permeability of the dielectric.
- $\grave{E}$ is the permittivity of the dielectric.
- $\grave{E}\acute{z}$ is the imaginary part of $\grave{E}$, $\grave{E}\acute{z} = \grave{E}_0\grave{E}_r \tan \hat{a}$, where:
  - $\grave{E}_0$ is the permittivity of free space.
  - $\grave{E}_r$ is the Relative permittivity constant parameter value.
  - $\tan \hat{a}$ is the Loss tangent of dielectric parameter value.
- $\hat{a}_{cond}$ 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 $k = \grave{A}_a + i\grave{a}$, where $\grave{A}_a$ is the attenuation coefficient and $\grave{a}$ is the wave number. The attenuation coefficient $\grave{A}_a$ is related to the specified loss, $\grave{A}$, by

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

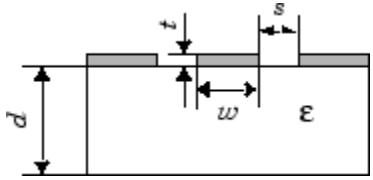
  The wave number $\grave{a}$ is related to the specified phase velocity, V_p, by

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

  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, $\grave{E}$.



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

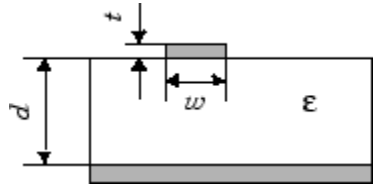
  Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

- Stub mode

  Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See ÒTransmission Line with StubÓ on page 1-168 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

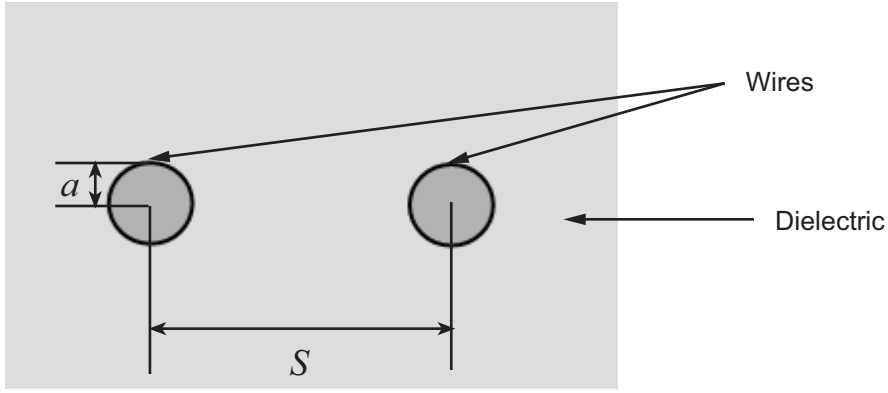
  Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

- Stub mode

  Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See ÒTransmission Line with StubÓ on page 1-168 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] RF Blockset 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, $\hat{\imath}_0$. 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, $\grave{E}_0$. 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`.

Å Conductivity of conductor

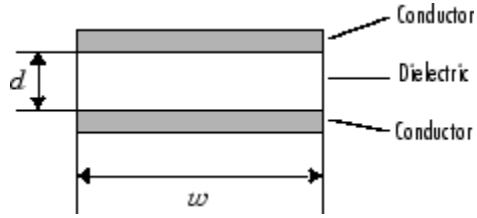
Specify the conductivity of the conductor in siemens per meter. The default value is `inf S/m`.

Å Stub mode

Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See òTransmission Line with Stubó on page 1-168 for more information.

Å **`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:

Å Plate width

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

Å Plate separation

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

Å 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, $\hat{\imath}_0$. The default value is `1`.

Å 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, $\ddot{E}_0$. 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 òTransmission Line with Stubó on page 1-168 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`.

Å 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`.

Å Interpolation method

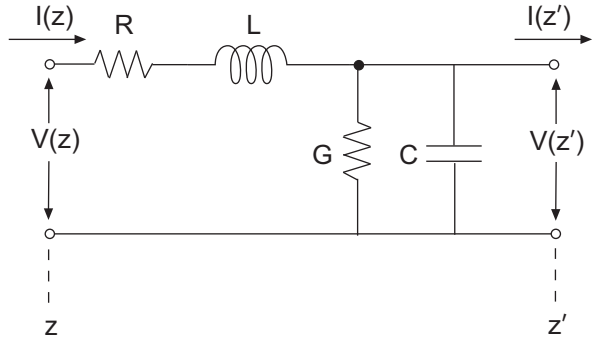
Specify the interpolation method the block uses to calculate the parameter values at the modeling frequencies. Your choices are `Linear`, `Spline`, or `Cubic`. The default is `Linear`.

Å Stub mode

Specify the type of stub. Choices are `Not a stub`, `Shunt`, or `Series`. The default value is `Not a stub`. See òTransmission Line with Stubó on page 1-168 for more information.

Å **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 $z' = z + Gz$.

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

Å 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

  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 ÒTransmission Line with StubÓ on page 1-168 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.

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:

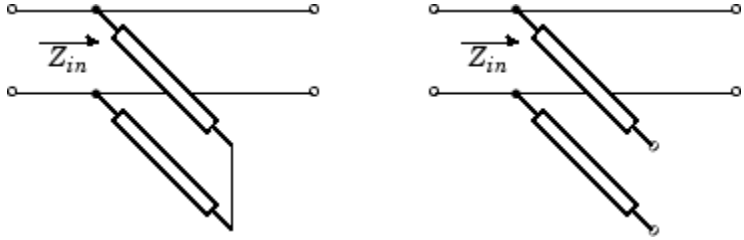
$$A = \frac{e^{kd} + e^{-kd}}{2}$$

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

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

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

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

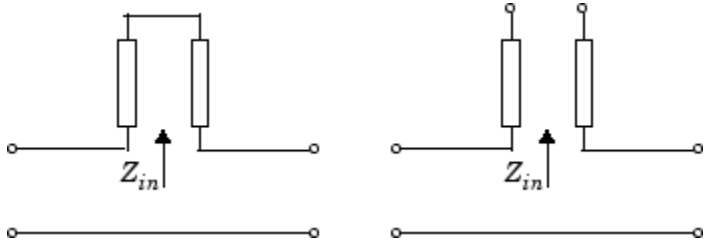
$$A = 1$$

$$B = 0$$

$$C = 1 / Z_{in}$$

$$D = 1$$

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

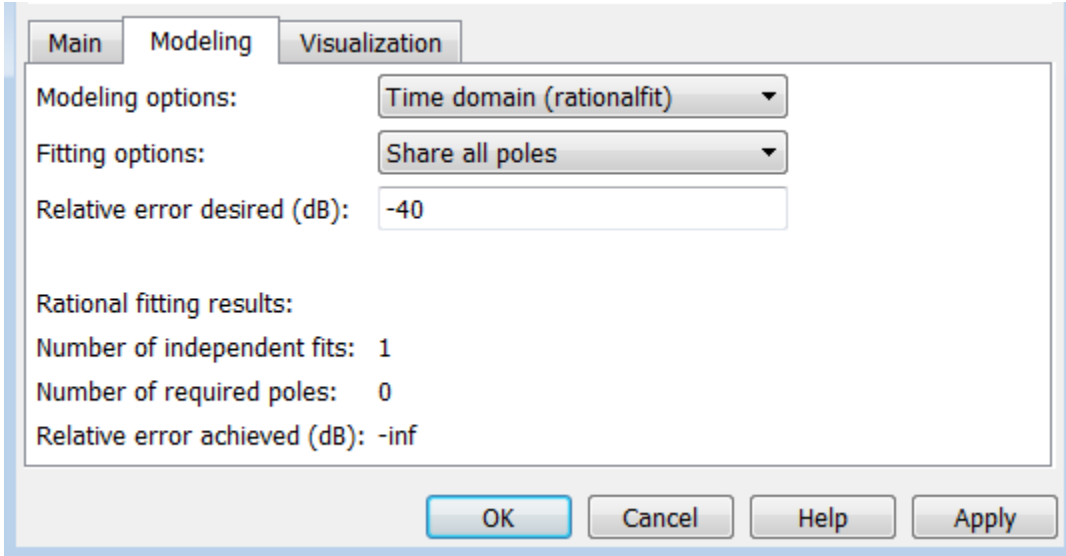
Modeling options tab is activated for all transmission line options except `Delay-based and lossless`, `Delay-based and lossy`, `Lumped parameter L-section`, and `Lumped parameter pi-section`.

Modeling options

RF Blockset 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 òCompare Time and Frequency Domain Simulation Options for S-parametersó.
- 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` ñ 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` ñ 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` ñ 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.

Examples

The example, òTransmission Lines, Delay-based and Lumped Modelsó, shows how to use Delay-based and Lumped Transmission Line blocks.

References

[1] Sussman-Fort, S. E., and J. C. Hantgan. òSPICE Implementation of Lossy Transmission Line and Schottky Diode Models.ó *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. òData Transmission Lines and Their Characteristics.ó *National Semiconductor Application Note 806*, April 1992.

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.

See Also

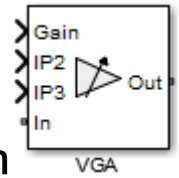
S-Parameters

VGA

Variable gain amplifier

Library

Elements



Description

The VGA block models a RF Blockset 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	$g = \frac{2\sqrt{G * Z_{out} * Z_{in}}}{\lvert Z_{in} \rvert}$	
c_2	$C_2 = \frac{g}{\sqrt{2 * IP2 * Z_{in}}} volts^{-1}$	$C_2 = \frac{g^2}{\sqrt{8 * IP2 * Z_{out}}} volts^{-1}$
c_3	$C_3 = \frac{4}{3} \cdot \frac{g}{2 * IP3 * Z_{in}} volts^{-2}$	$C_2 = \frac{4}{3} \cdot \frac{g^3}{8 * IP3 * Z_{out}} volts^{-2}$

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

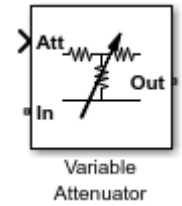
See Also

Amplifier

Introduced in R2015b

Variable Attenuator

Model variable attenuator
Library: RF Blockset / 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 kT/R A^2/Hz, where:

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

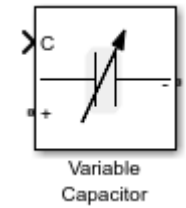
See Also

Attenuator | Variable Inductor | Variable Capacitor | Variable Phase Shift

Introduced in R2016b

Variable Capacitor

Model variable capacitor
Library: RF Blockset / Circuit Envelope / Elements



Description

The Variable Capacitor block controls the output of RF Blockset feedback circuits using Simulink controlled capacitance in farads. The minimum value of the capacitance (*Cmin*) is a RF Blockset 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 RF Blockset signal.

Parameters

The Variable Capacitor block has no parameters.

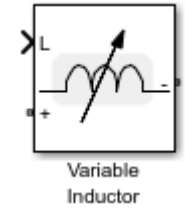
See Also

Capacitor | Variable Inductor

Introduced in R2016b

Variable Inductor

Model variable inductor
Library: RF Blockset / Circuit Envelope / Elements



Description

The Variable Inductor block controls the output of RF Blockset feedback circuits using Simulink controlled inductance in henries. The minimum value of the inductance (*Lmin*) is a RF Blockset 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 RF Blockset signal.

Parameters

The Variable Inductor block has no parameters.

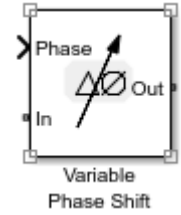
See Also

Inductor | Variable Capacitor

Introduced in R2016b

Variable Phase Shift

Model variable phase device
Library: RF Blockset / Circuit Envelope / Elements



Description

The Variable Phase Shift block controls the phase of the output signal of a RF Blockset 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 RF Blockset 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

Spectrum Analyzer

Display frequency spectrum of time-domain signals
Library: RF Blockset / Circuit Envelope / Utilities

Description

Note The Spectrum Analyzer block in the Simscape product contains a subset of functionality of the DSP System Toolbox block with the same name. This page describes the block configuration and functionality available with a Simscape license. If you also have a DSP System Toolbox license, then the Spectrum Analyzer block in the Simscape > Utilities library is identical to the block in the DSP System Toolbox > Sinks library. For more information, see Spectrum Analyzer in the DSP System Toolbox documentation.

The Spectrum Analyzer block accepts input signals with discrete sample times and displays frequency spectra of these signals.

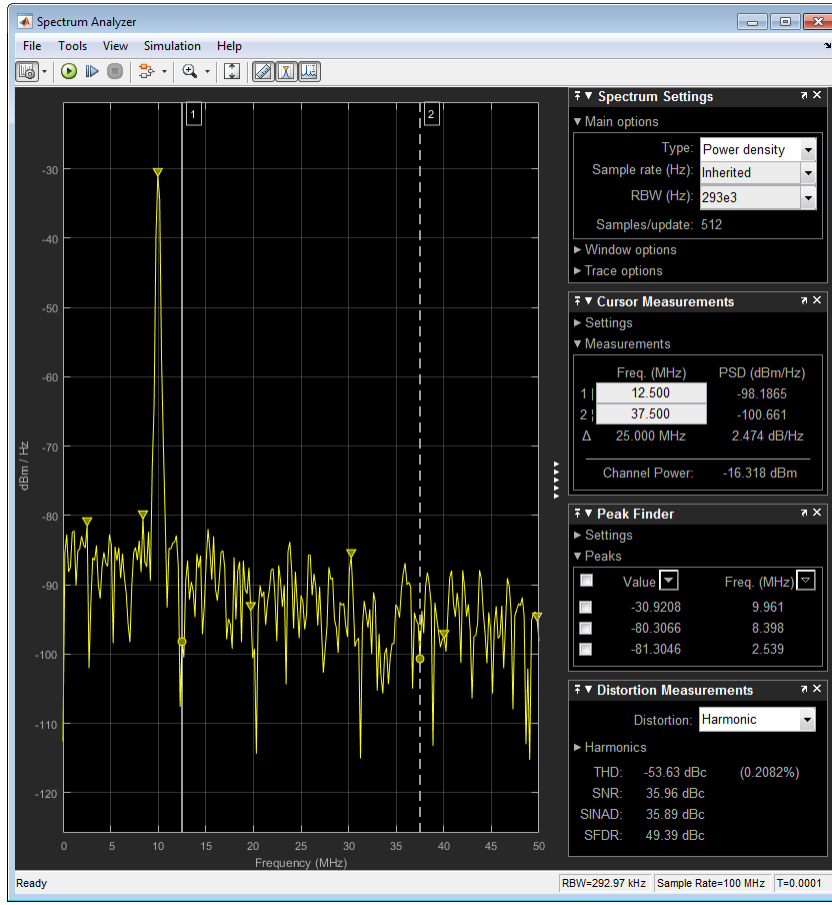
To use a Spectrum Analyzer block, instead of a regular scope, in a Simscape model:

1. Add a Spectrum Analyzer block to your block diagram.
2. If your model uses a variable-step solver, also add a Rate Transition block and connect it to the input of the Spectrum Analyzer, setting the Output port sample time to the sample time you wish the Spectrum Analyzer to use.

   If your model uses a local solver, then it produces output physical signals with discrete sample times and you do not need to add a Rate Transition block. However, if you need to down-sample from the solver fixed step size, you can also use a Rate Transition block. For more information on using local solvers, see òMaking Optimal Solver Choices for Physical Simulationó (Simscape).
3. Use a PS-Simulink Converter block to connect the output physical signal of interest to the input of the Spectrum Analyzer block (or to the input of the Rate Transition

block, if using one). For more information, see òConnecting Simscape Diagrams to Simulink Sources and Scopesó (Simscape). You can also use additional signal processing blocks between the PS-Simulink Converter and the Spectrum Analyzer to enhance signal quality.

4 Run the simulation. The Spectrum Analyzer, referred to here as the scope, opens and displays the frequency spectrum of the signal.



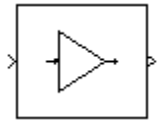
See Also

Introduced in R2017b

Equivalent Baseband Blocks مم Alphabetical List

Amplifier (Idealized Baseband)

Complex baseband model of amplifier with noise



Library

Mathematical

Description

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

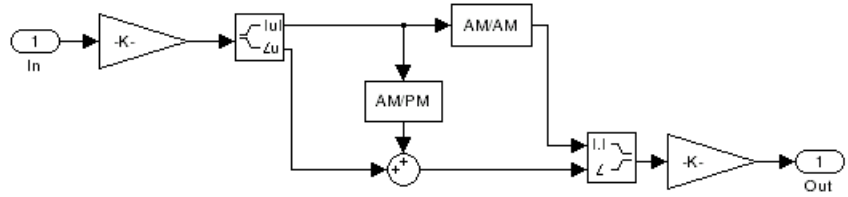
Note This block assumes a nominal impedance of 1 ohm.

Modeling Nonlinearity

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

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

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



Application of Nonlinearity

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

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

AM/AM and AM/PM Conversions

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

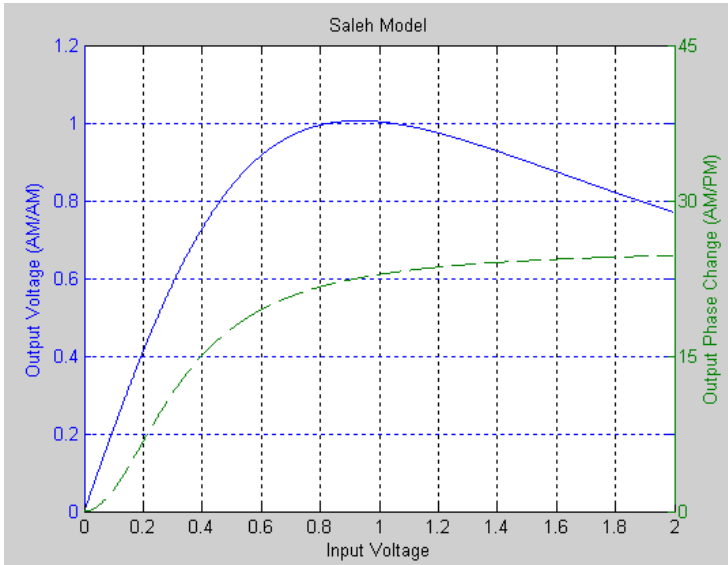
1. Right-click the Amplifier block.
2. Select Look under mask in the pop-up menu. This displays the block's configuration underneath the mask. The block contains five subsystems corresponding to the five nonlinearity methods.

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

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

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

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



Model Parameters and Characteristics of Nonlinearity Modeling Methods

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

- òCubic Polynomial Modeló on page 2-5
- òHyperbolic Tangent Modeló on page 2-5
- òSaleh Modeló on page 2-6
- òGhorbani Modeló on page 2-7

Å òRapp Modeló on page 2-7

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

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

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

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

2. Computing the scaled input signal by multiplying the amplifier input signal by *f*.
3. Limiting the scaled input signal to a maximum value of 1.
4. Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

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

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

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

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

respectively.

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

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

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

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

2. Computing the scaled input signal by multiplying the amplifier input signal by *f*.
3. Limiting the scaled input signal to a maximum value of 1.
4. Applying an AM/AM conversion to the amplifier gain, according to the following cubic polynomial equation:

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

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

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

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

respectively.

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

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.

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.

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

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

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

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

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

Thermal Noise Simulation

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

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

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

- `Noise figure` ñ 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

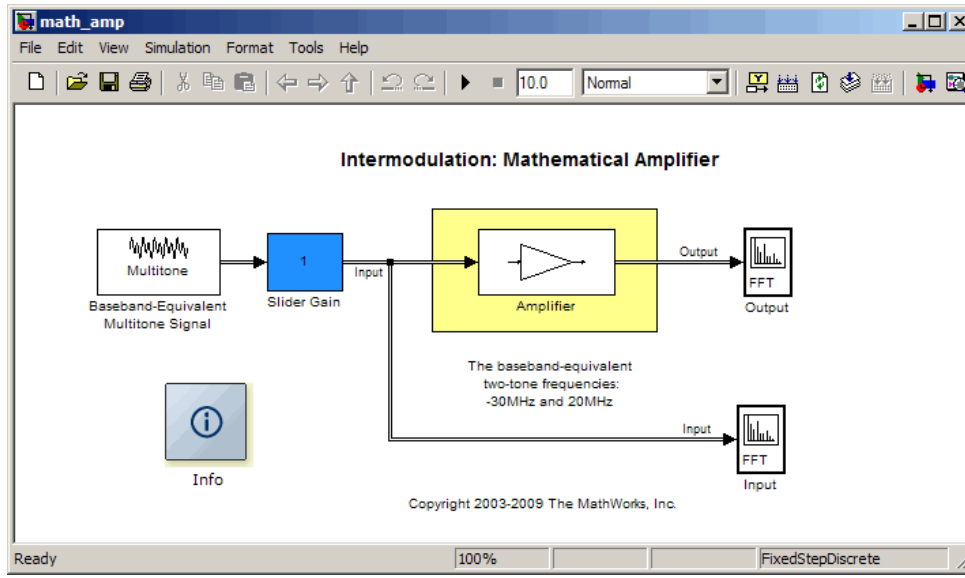
Scalar specifying the amount of noise relative to a noise temperature of 290 kelvin. This field becomes visible if you select `Noise factor` as the Specification method parameter. Tunable.

Initial seed

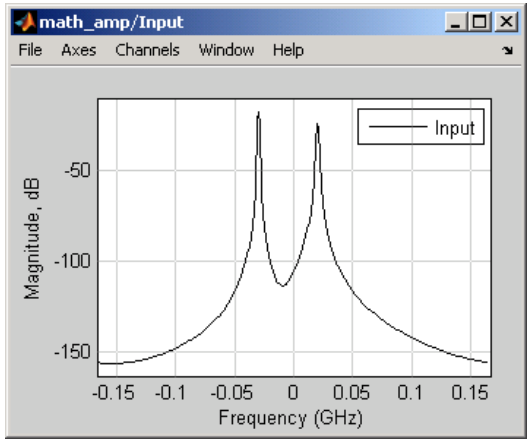
Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise.

Examples

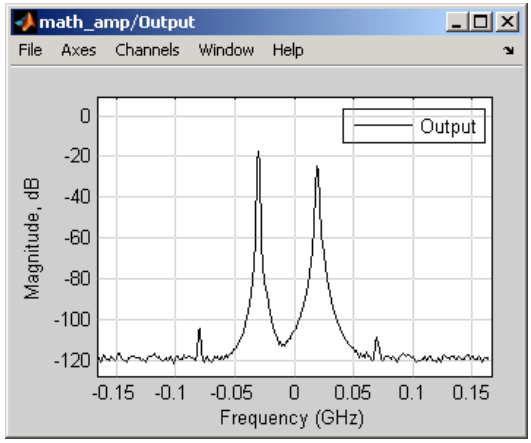
You can see the effect of the Amplifier block in the example Intermodulation: Mathematical Amplifier.



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, òThe Effect of Solid State Power Amplifiers (SSPAs) Nonlinearities on MPSK and M-QAM Signal Transmission,ó *Sixth Int'l Conference on Digital Processing of Signals in Comm.*, 1991, pp. 193-197.

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

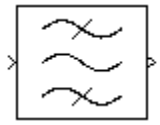
[3] Saleh, A.A.M., òFrequency-independent and frequency-dependent nonlinear models of TWT amplifiers,ó *IEEE Trans. Communications*, vol. COM-29, pp.1715-1720, November 1981.

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 RF Blockset 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.

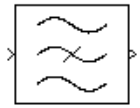
See Also

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 RF Blockset 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.

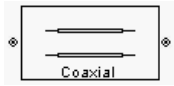
See Also

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

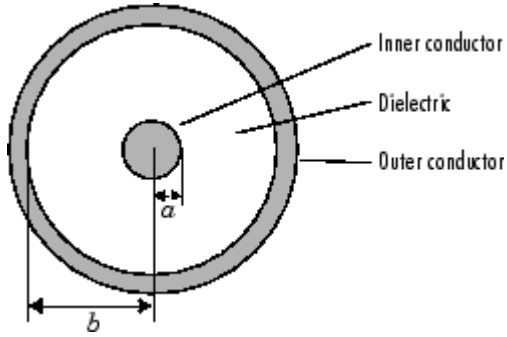


Library

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.

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:

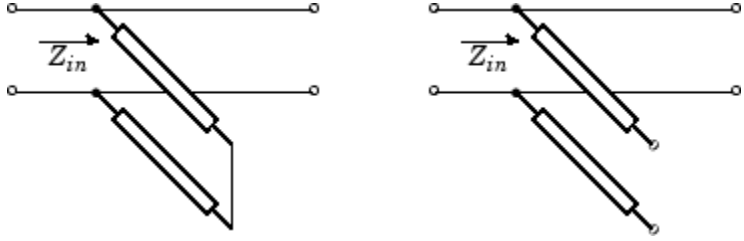
- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- $\hat{R}_{cond}$ is the conductivity in the conductor.
- $\hat{\imath}$ is the permeability of the dielectric. $\hat{\imath} = \hat{\imath}_0\ \hat{\imath}_r$ where:
  - $\hat{\imath}_0$ is the permeability in free space.
  - $\hat{\imath}_r$ is the Relative permeability constant parameter value.
- The is a complex dielectric constant given by $\grave{E} = \grave{E}\acute{z} \check{\imath}\ \eta\grave{E}z = \grave{E}\acute{z}(1 \check{\imath}\ \eta tan\hat{a})$
- $\grave{E}\acute{z}$ is the real part of complex dielectric constant $\grave{E}$, $\grave{E}\acute{z} = \grave{E}_0\grave{E}_r$. $\grave{E}z$ is the imaginary part of complex dielectric constant $\grave{E}$, $\grave{E}z = \grave{E}_0\grave{E}_r \tan\hat{a}$ where :
  - $\grave{E}_0$ is the permittivity of free space.
  - $\grave{E}_r$ is the Relative permittivity constant parameter value.
  - $\tan\hat{a}$ is the Loss tangent of dielectric parameter value.
- $\hat{a}_{cond}$ 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

$A = 1$

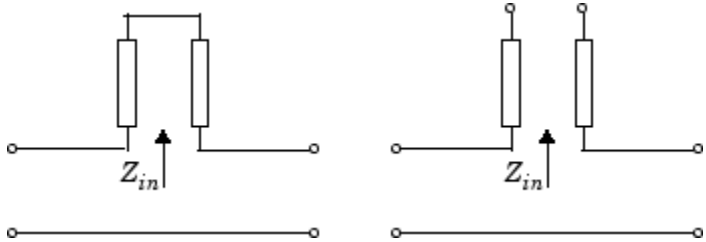
$B = 0$

$C = 1 / Z_{in}$

$D = 1$

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 $\hat{\imath}_0$.

Relative permittivity constant

Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space $\ddot{E}_0$.

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 òCreate Plotsó.

References

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

See Also

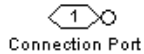
Coplanar Waveguide Transmission Line, General Passive Network, Transmission Line, Microstrip Transmission Line, Parallel-Plate Transmission Line, Two-Wire Transmission Line

Connection Port

Connection port for RF subsystem

Library

Input/Output Ports sublibrary of the Physical library

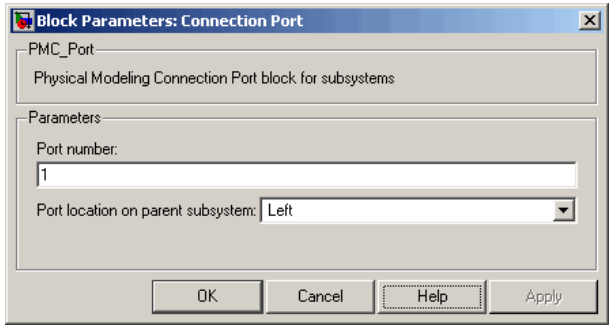


Description

The Connection Port block, placed inside a subsystem composed of RF Blockset 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 RF Blockset 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 RF Blockset 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

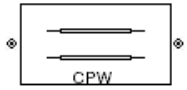
Use this parameter to choose on which side of the parent subsystem boundary the Port is placed. The choices are `Left` or `Right`. The default choice is `Left`.

See Also

Creating Subsystems (Simulink)

Coplanar Waveguide Transmission Line

Model coplanar waveguide transmission line

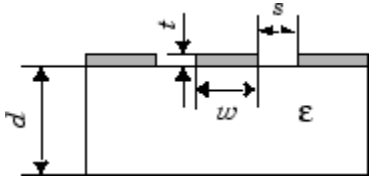


Library

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 (*È*).



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:

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

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

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

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

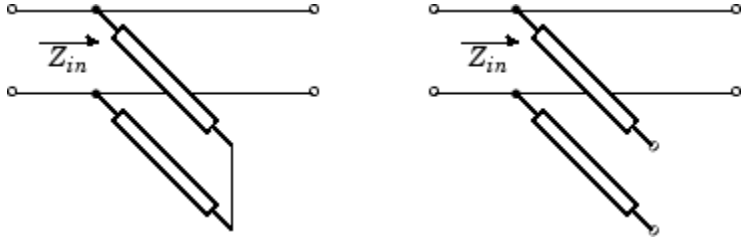
Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the specified conductor strip width, slot width, substrate height, conductor strip thickness, relative permittivity constant, conductivity and dielectric loss tangent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

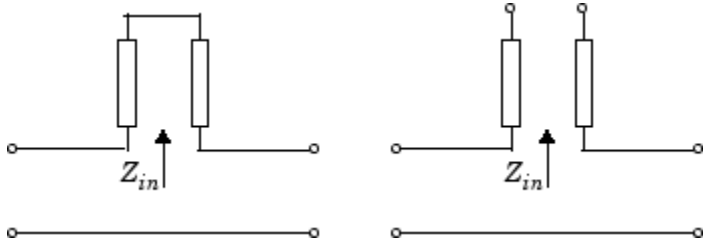


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

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 $\ddot{E}_0$.

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 òCreate Plotsó.

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 Appendix A.

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

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

If you specify block noise as spot noise data, the block uses the data to calculate noise figure. The block first interpolates the noise data for the modeling frequencies, using the

specified Interpolation method. It then calculates the noise figure using the resulting values.

Operating Conditions

Agilent® P2D and S2D files define block parameters for several operating conditions. Operating conditions are the independent parameter settings that are used when creating the file data. By default, RF Blockset 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

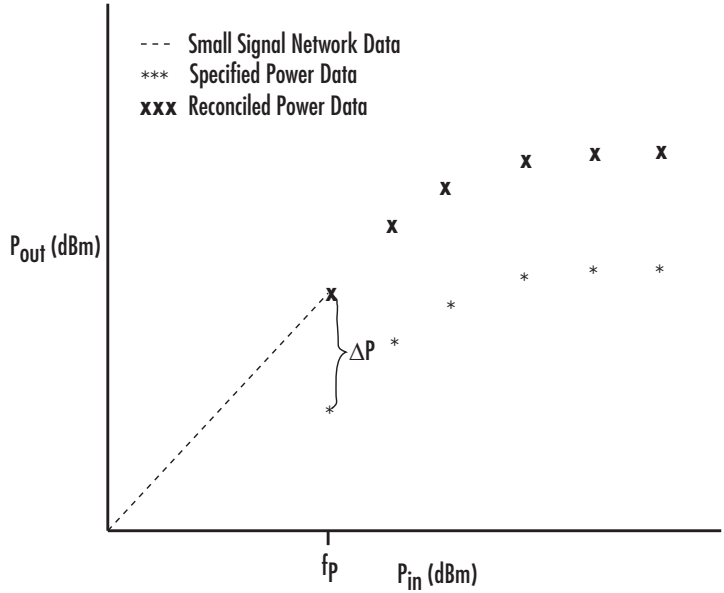
If the data source is a MathWorkso AMP file or an Agilent S2D file that contains both network parameter data and power data, the blockset checks the data for consistency and reconciles it as necessary.

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



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 òCreate Plotsó. Use `rftool` or the RF Toolbox plotting functions to plot other data.

Operating Conditions Tab

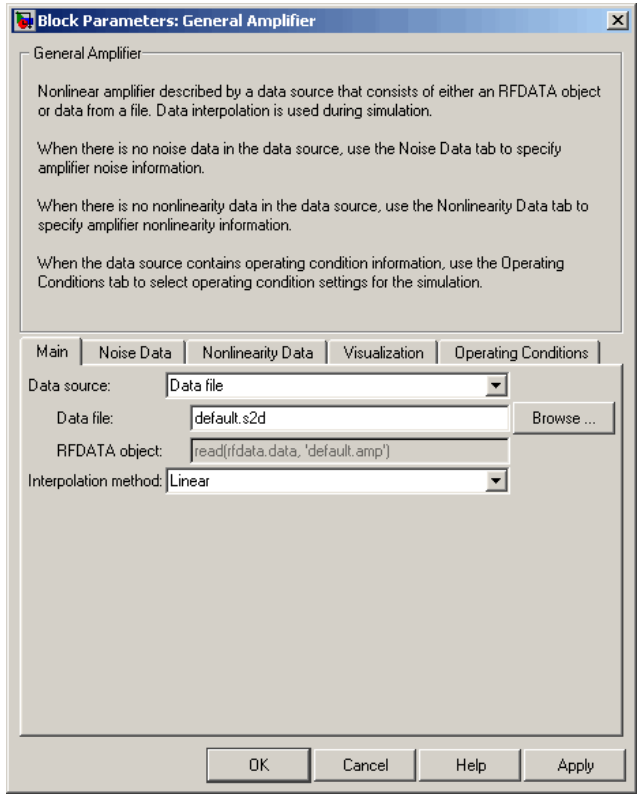
If the data source contains data at multiple operating conditions, the Operating Conditions tab contains two columns. The Conditions column shows the available conditions, and the Values column contains a drop-down list of the available values for the corresponding condition. Use the drop-down lists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

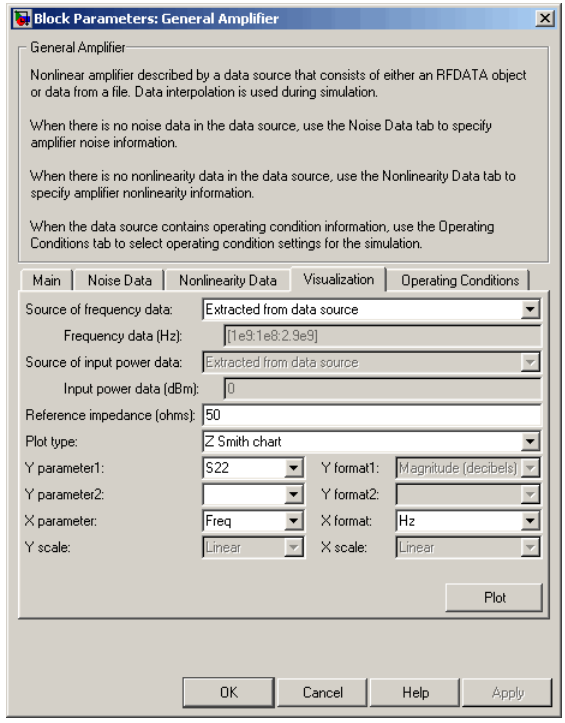
1 On the Main tab, accept the default settings.



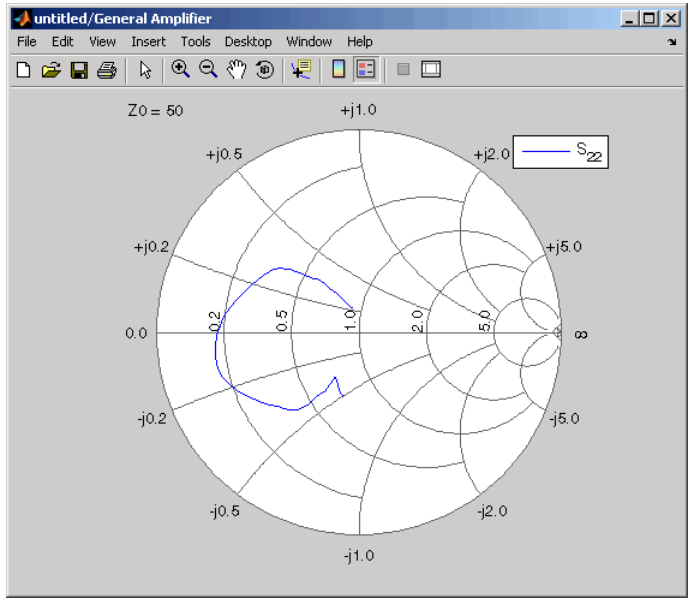
2 On the Visualization tab, set the parameters as follows:

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

Å 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

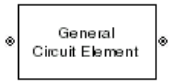
Output Port, S-Parameters Amplifier, Y-Parameters Amplifier, Z-Parameters Amplifier

`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 òRF Circuit Objectsó (RF Toolbox) for more information about circuit objects.

Visualization Tab

For information about plotting, see òCreate Plotsó.

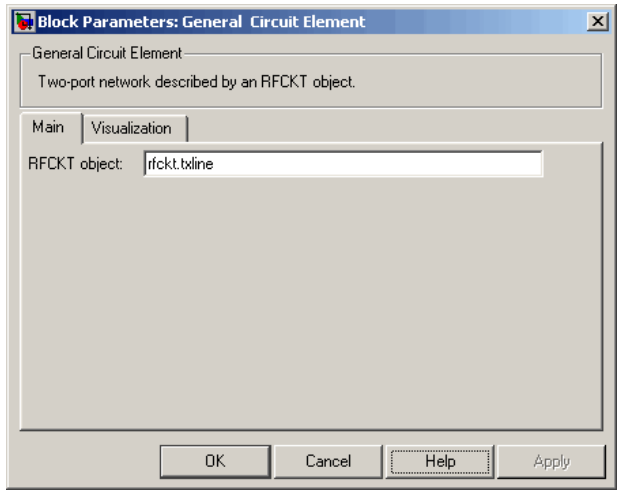
Examples

Creating a General Circuit Element from an RF Toolbox Object

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

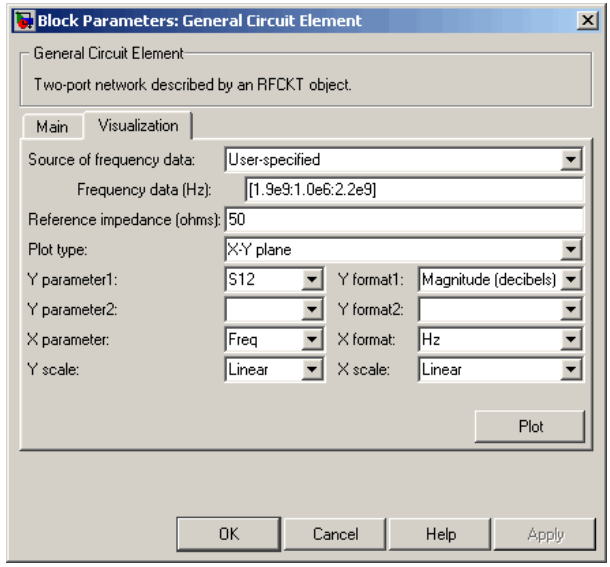
1 On the Main tab, set the RFCKT object parameter to `rfckt.txline`.

Click Apply. This action applies the specified settings.

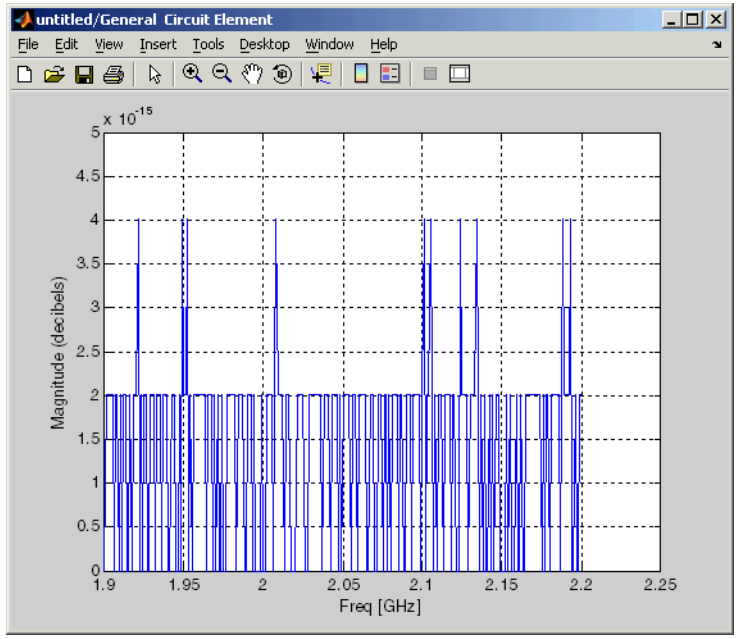


2 Set the General Circuit Element block parameters on the Visualization tab as follows:

Å 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.



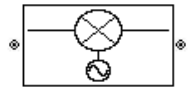
See Also

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 òMap Network Parameters to Modeling Frequenciesó for more details.

RF Blockset 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

Å f_{in} and f_{out} are the mixer input and output frequencies, respectively.

- a_1 and a_2 are the incident waves at the mixer input and output, respectively.

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

Active Noise

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

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

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

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

Phase Noise

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

The blockset computes the digital filter by:

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

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

Nonlinearity

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

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

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

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

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

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

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

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

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

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

3

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

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

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

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

The first two equations are the evaluation of the polynomial $F_{AM/AM}(s)$ at the points $(\sqrt{P_{sat,in}}, \sqrt{P_{sat,out}})$ and $(\sqrt{P_{1dB,in}}, \sqrt{P_{1dB,out}})$, expressed in linear units (such as W or mW) and normalized to a 1-N 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` 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 $f_{out} = f_{in}$ ð 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} ð f_{lo} Ó $1/(2t_s)$, 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 òCreate Plotsó. 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 òMap Network Parameters to Modeling Frequenciesó 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

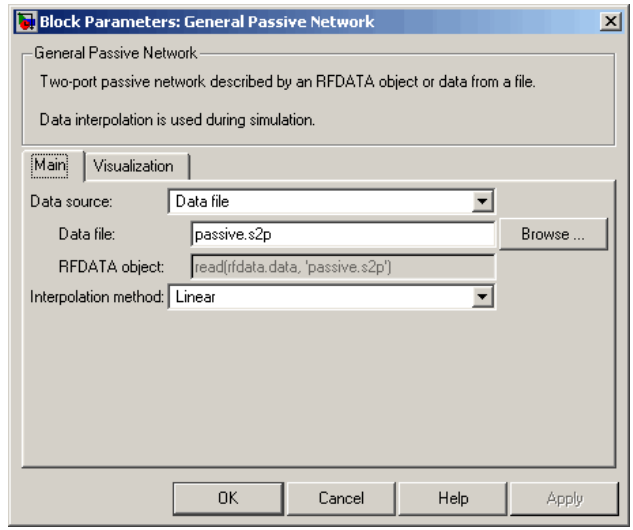
For information about plotting, see òCreate Plotsó.

Examples

Creating a General Passive Network Block from File Data

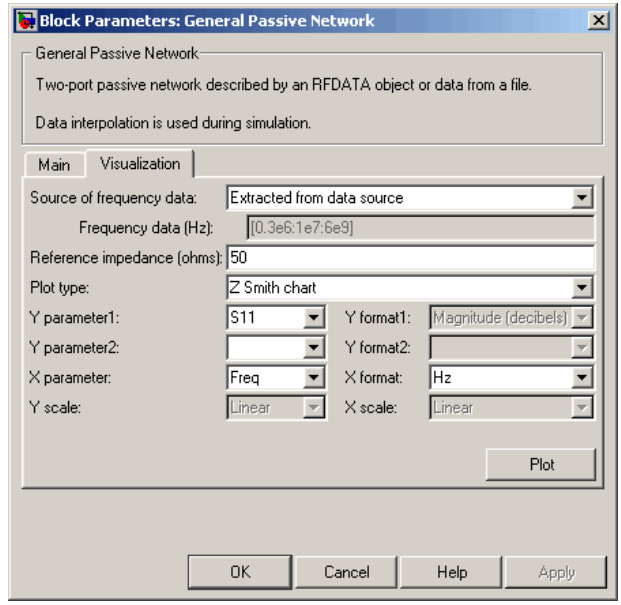
This example creates a two-port passive network from the data in the file `passive.s2p`. The file contains S-parameters for frequencies from about 0.315 MHz to 6.0 GHz. The General Passive Network block uses linear interpolation to model the network described in the object.

1 On the Main tab, accept the default settings.

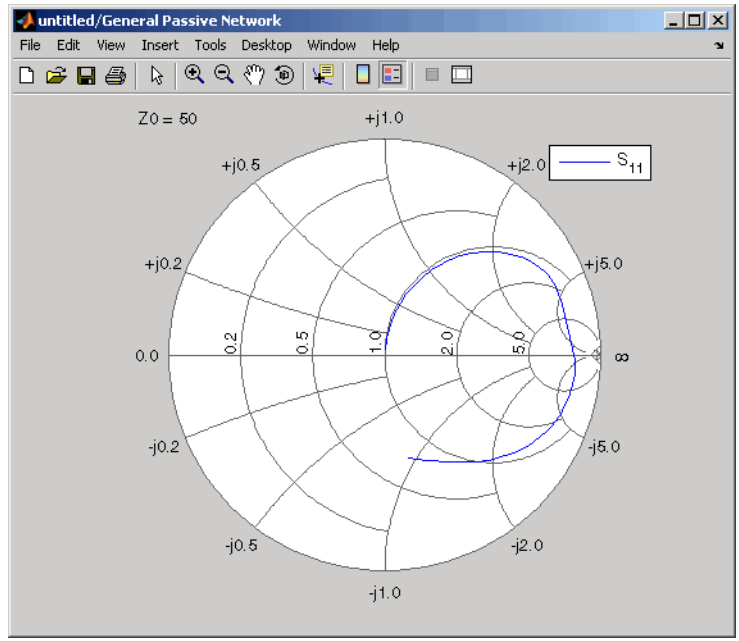


2 On the Visualization tab, set the parameters as follows:

Å 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

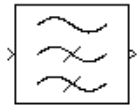
General Circuit Element, Output Port, S-Parameters Passive Network, Y-Parameters Passive Network, Z-Parameters Passive Network

`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 RF Blockset 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.

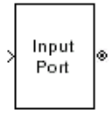
See Also

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 òConvert to and from Simulink Signalsó.

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` ñ 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` ñ 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 òConvert to and from Simulink Signalsó.

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 òBaseband-Equivalent Modelingó.

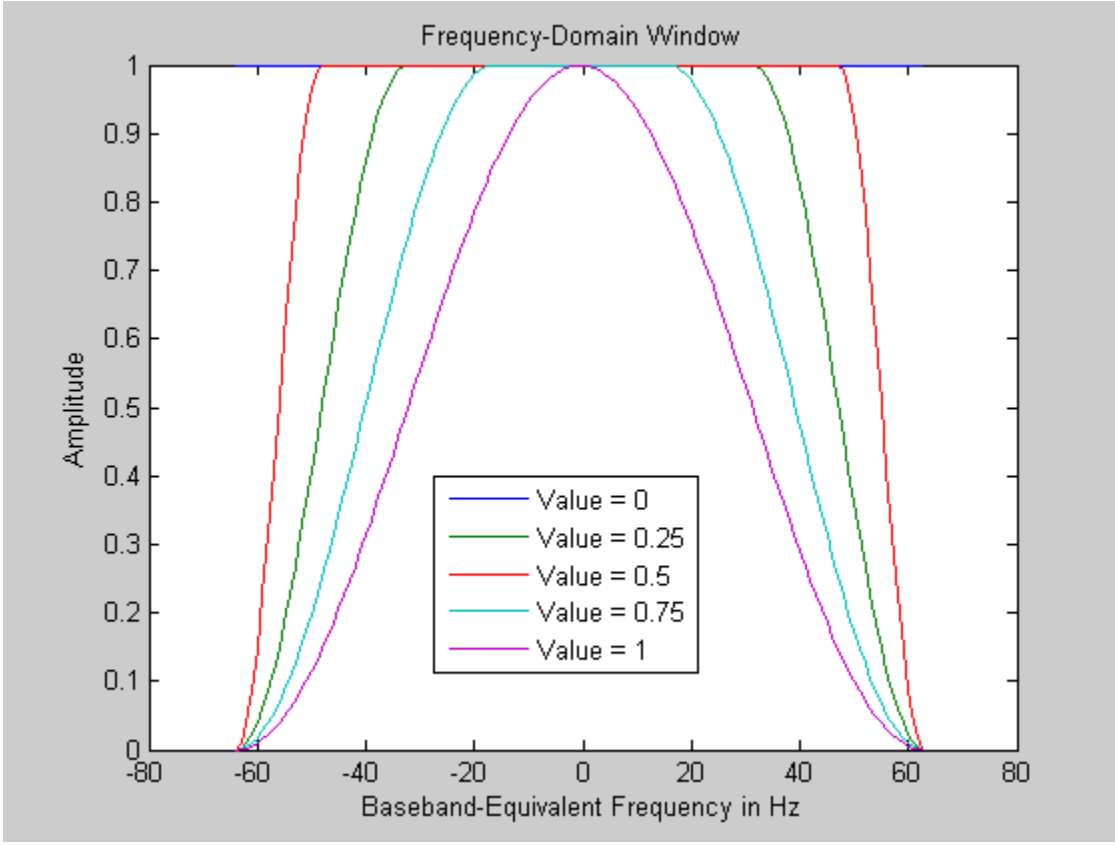
Note The equivalent-baseband simulation algorithm uses the next power of 2 greater than the specified filter length in its calculations and then truncates the impulse response to the specified length. As a result, you get different results when you set the Finite impulse response filter length parameter to a number that is not a power of 2. For more information, see òCalculate the Baseband-Equivalent Impulse Responseó.

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 òCreate a Complex Baseband-Equivalent Modelô 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 òCreate a Complex Baseband-Equivalent Modeló 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

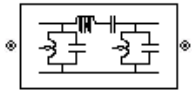
Nonnegative integer specifying the initial seed for the random number generator the block uses to generate noise. This parameter becomes visible if you select the Add noise parameter. If you specify the initial seed parameter with a variable, the initial seed changes each successive time you run a model.

See Also

Output Port

LC Bandpass Pi

Model LC bandpass pi network



Library

Ladder Filters sublibrary of the Physical library

Description

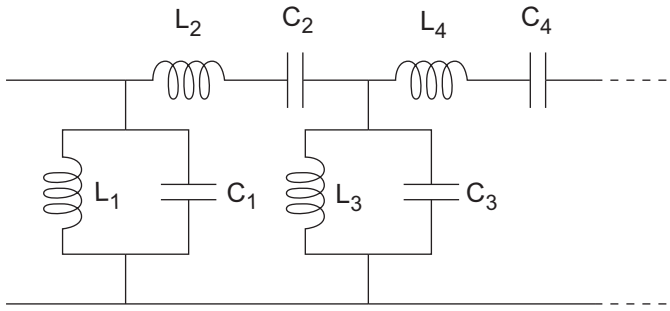
The LC Bandpass Pi block models the LC bandpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.

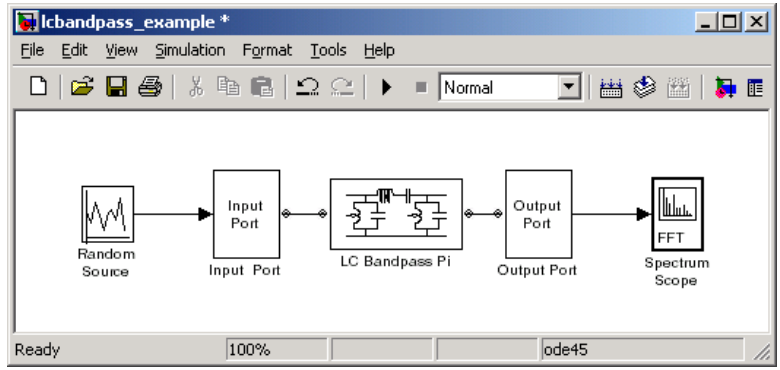
Visualization Tab

For information about plotting, see òCreate Plotsó.

Examples

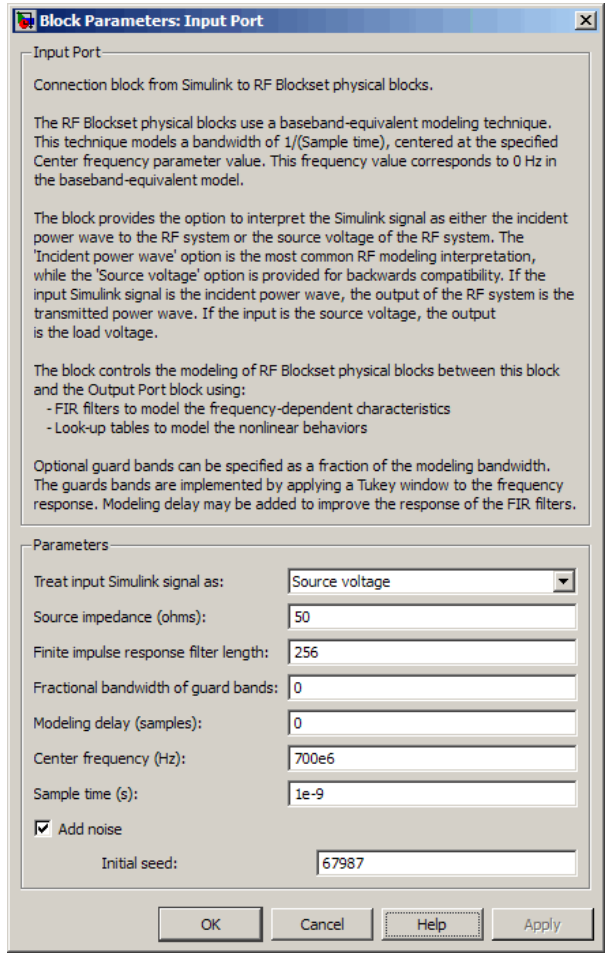
Using a Ladder Filter Block to Filter Gaussian Noise

This example provides complex random noise in Gaussian form as input to an LC Bandpass Pi block. A DSP System Toolbox Spectrum Scope block plots the filtered output.

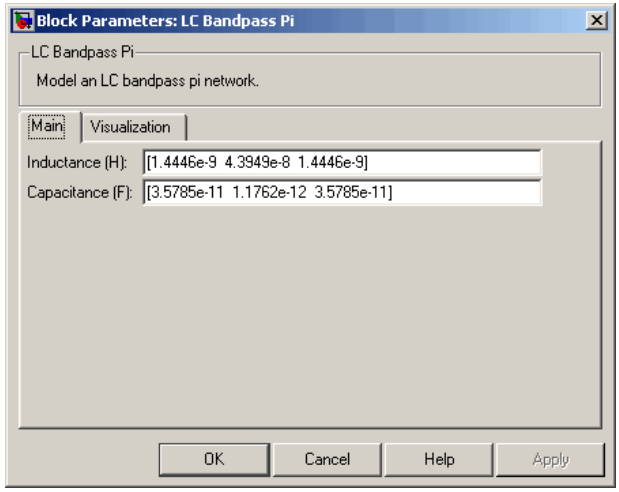


The DSP System Toolbox Random Source block produces frame-based output at 512 samples per frame. Its Sample time parameter is set to 1.0e-9. This sample time must match the sample time for the physical part of the model, which you provide in the Input Port block diagram.

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

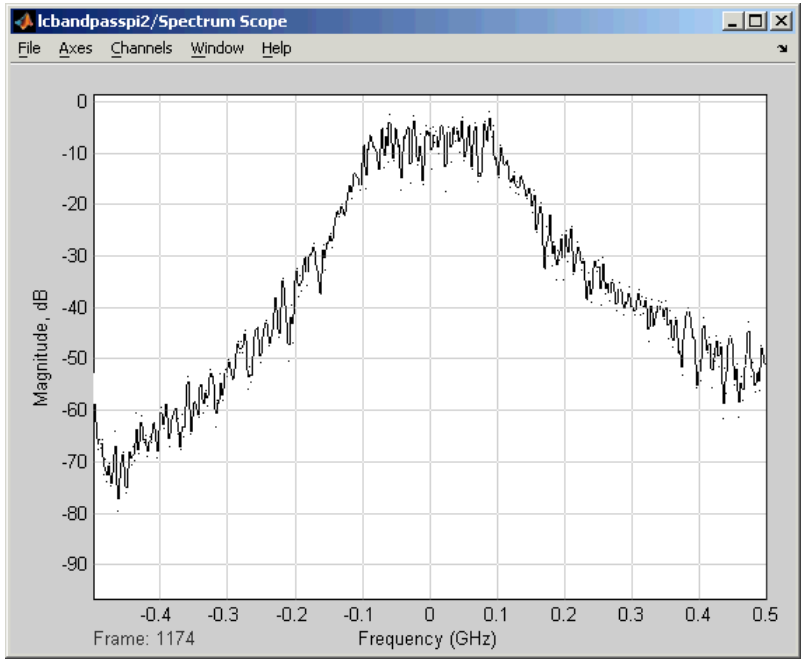


The LC Bandpass Pi block provides the inductances for three inductors, in order from source to load, [1.4446e-9, 4.3949e-8, 1.4446e-9]. Similarly, it provides the capacitances for three capacitors [3.5785e-11, 1.1762e-12, 3.5785e-11].



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

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



References

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

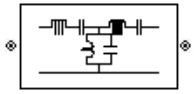
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

See Also

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

LC Bandpass Tee

Model LC bandpass tee network



Library

Ladder Filters sublibrary of the Physical library

Description

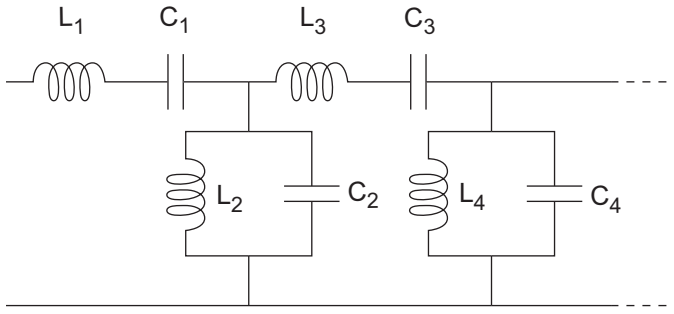
The LC Bandpass Tee block models the LC bandpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

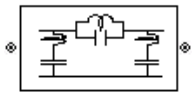
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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

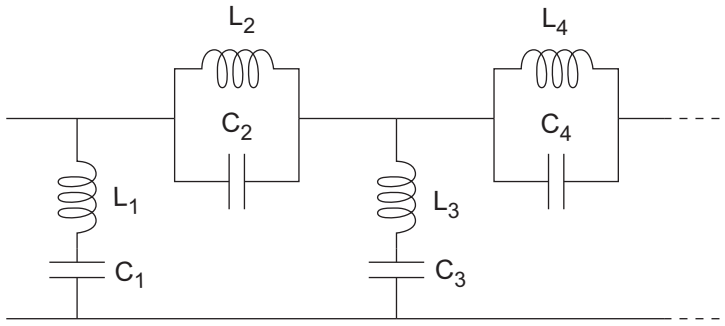
The LC Bandstop Pi block models the LC bandstop pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

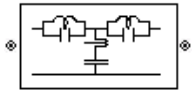
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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

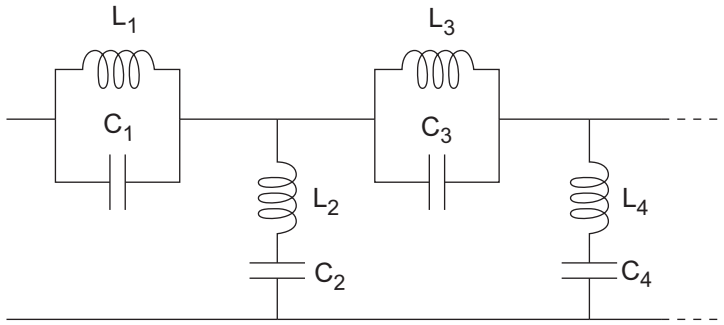
The LC Bandstop Tee block models the LC bandstop tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

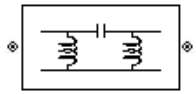
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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

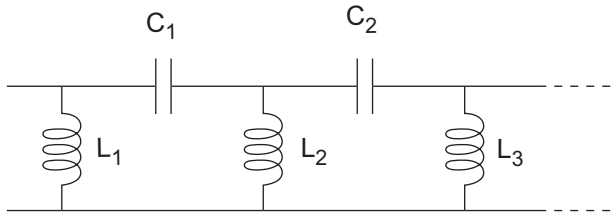
The LC Highpass Pi block models the LC highpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

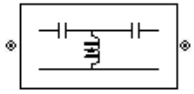
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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

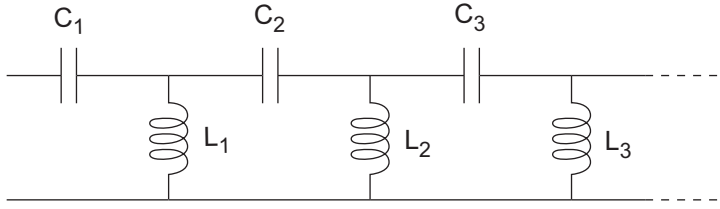
The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

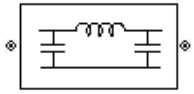
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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

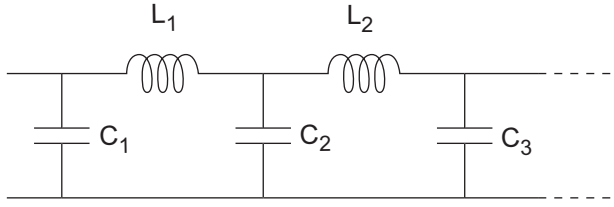
The LC Lowpass Pi block models the LC lowpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

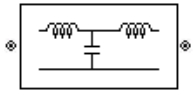
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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

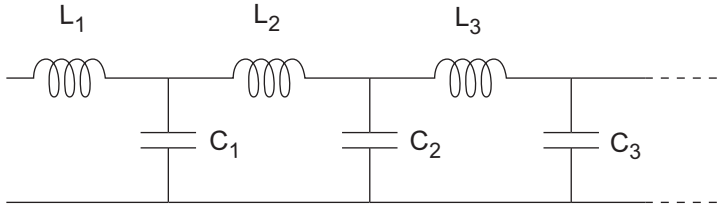
The LC Lowpass Tee block models the LC lowpass tee network described in the block dialog box in terms of its frequency-dependent S-parameters.

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 òCreate Plotsó.

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.

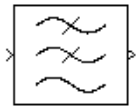
[2] Zverev, Anatol I., *Handbook of Filter Synthesis*, John Wiley & Sons, 1967.

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 RF Blockset 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.

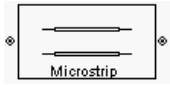
See Also

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

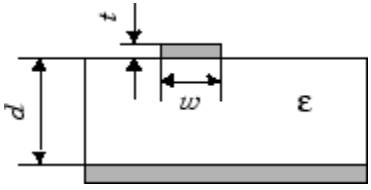


Library

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 (*Ɛ*).



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:

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

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

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

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

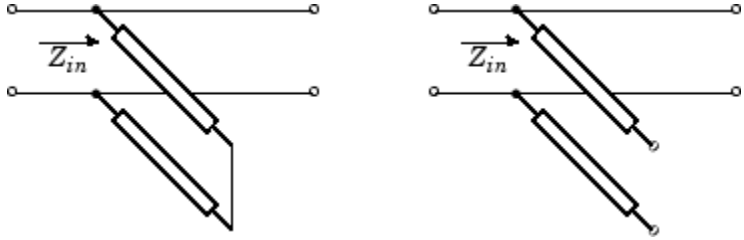
Z_0 and k are vectors whose elements correspond to the elements of f, a vector of modeling frequencies. Both can be expressed in terms of the specified conductor strip width, substrate height, conductor strip thickness, relative permittivity constant, conductivity, and dielectric loss tangent of the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

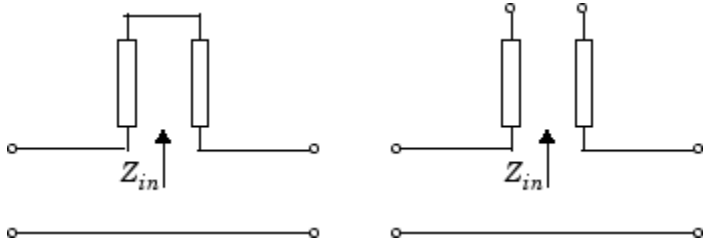


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

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

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 $\ddot{E}_0$.

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 òCreate Plotsó.

References

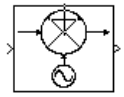
[1] Gupta, K.C., G. Ramesh, I. Bahl, and P. Bhartia, *Microstrip Lines and Slotlines*, Second Edition, Artech House, 1996. pp. 102-109.

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

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

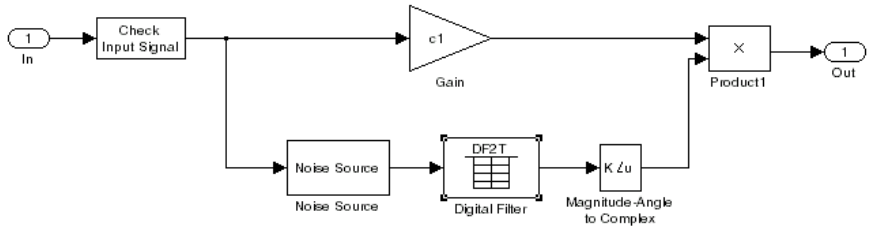
The Mixer block includes both the IF and RF signals as complex-baseband equivalent signals. Both the IF and RF center frequencies are represented as 0 hertz. The amplitude of the noise spectrum is specified by the noise power contained in a 1-hertz bandwidth offset from the carrier by a specified frequency.

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

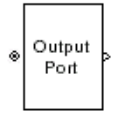
[1] Kasdin, N.J., òDiscrete Simulation of Colored Noise and Stochastic Processes and 1/(f^alpha); Power Law Noise Generation,ó The Proceedings of the IEEE, May, 1995, Vol. 83, No. 5.

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 òConvert to and from Simulink Signalsó.

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

Linear Subsystem

For the linear subsystem, the Output Port block uses the Input Port block parameters and the interpolated S-parameters calculated by each of the cascaded physical blocks to calculate the baseband-equivalent impulse response. Specifically, it

1 Determines the modeling frequencies f as an N-element vector. The modeling frequencies are a function of the center frequency f_c, the sample time t_s, and the finite impulse response filter length N, all of which you specify in the Input Port block dialog box.

The nth element of f, f_n, is given by

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

where

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

2 Calculates the passband transfer function for the frequency range as

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

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

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

where

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

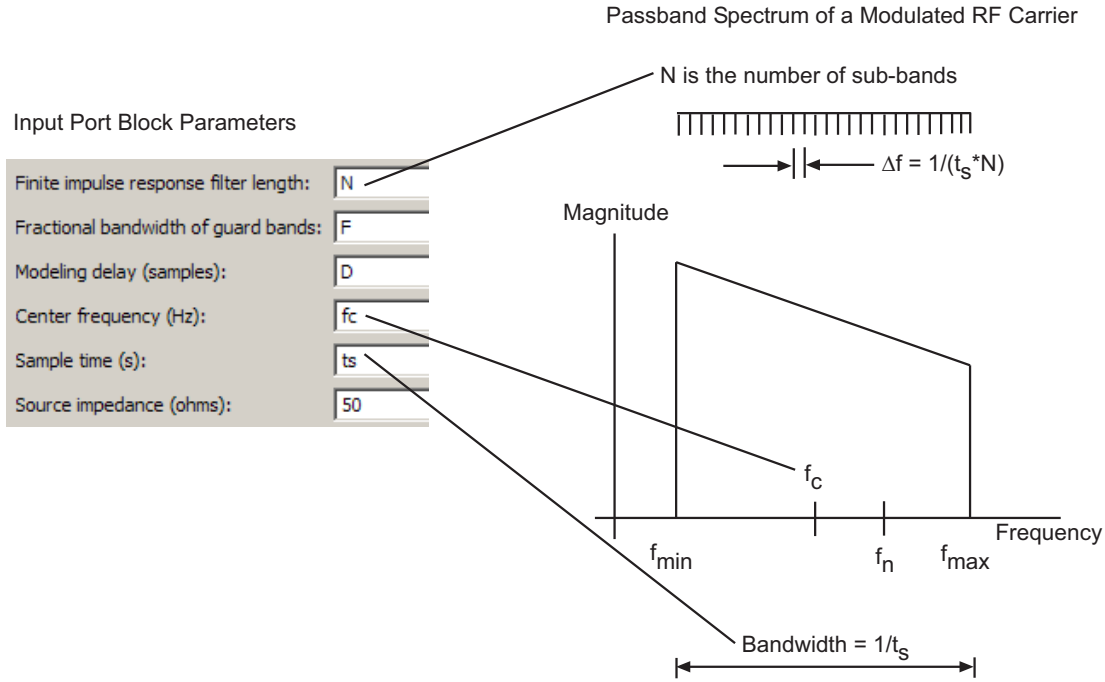
$$\Gamma_s = \frac{Z_s - Z_o}{Z_s + Z_o}$$

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

and

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

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



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

The baseband transfer function is shown in the following figure.