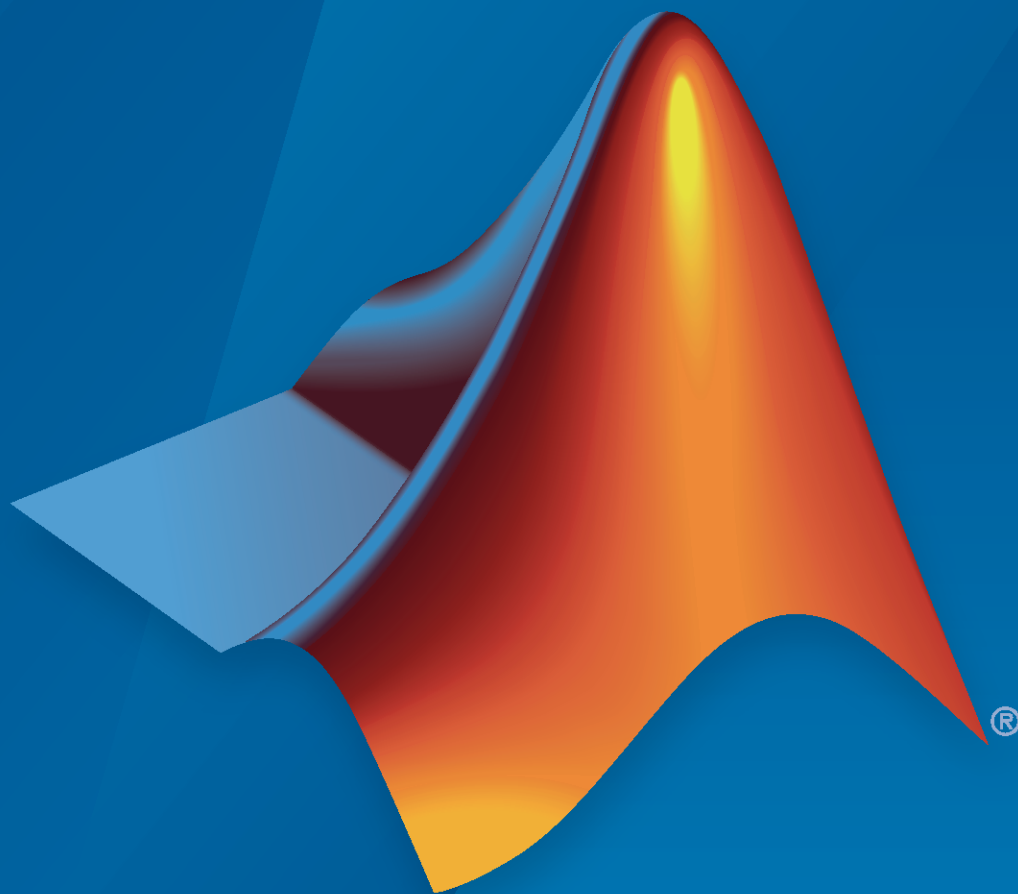


RF Blockset™

Reference



MATLAB® & SIMULINK®

R2023a



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.
1 Apple Hill Drive
Natick, MA 01760-2098

RF Blockset™ Reference

© COPYRIGHT 2010–2023 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)
March 2018	Online only	Revised for Version 7.0 (Release 2018a)
September 2018	Online only	Revised for Version 7.1 (Release 2018b)
March 2019	Online only	Revised for Version 7.2 (Release 2019a)
September 2019	Online only	Revised for Version 7.3 (Release 2019b)
March 2020	Online only	Revised for Version 7.4 (Release 2020a)
September 2020	Online only	Revised for Version 8.0 (Release 2020b)
March 2021	Online only	Revised for Version 8.1 (Release 2021a)
September 2021	Online only	Revised for Version 8.2 (Release 2021b)
March 2022	Online only	Revised for Version 8.3 (Release 2022a)
September 2022	Online only	Revised for Version 8.4 (Release 2022b)
March 2023	Online only	Revised for Version 8.5 (Release 2023a)

1	Circuit Envelope Blocks
2	Equivalent Baseband Blocks
3	Functions
4	Idealized Baseband

Circuit Envelope Blocks

Amplifier

Model amplifier in RF systems



Libraries:
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 $V_i(t) = A_i(t)e^{j\omega t}$, and the output voltage is $V_o(t) = A_o(t)e^{j\omega t}$ at each carrier $w = 2\pi f$ in the 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.

Amplifier block mask icons are dynamic and show the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Noise figure (dB)** parameter on the block.

Noise Figure (dB): 10	Noise Figure (dB): 0

Parameters

Main

Source of amplifier gain — Amplifier gain source

Available power gain (default) | Open circuit voltage gain | Data source | Polynomial coefficients | AM/AM-AM/PM table

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

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

- **Data source** — The block uses the **Data source** parameter to calculate the linear voltage gain. When you use the data source option, the block uses S_{11} and S_{22} as the input and output impedances.
- **Polynomial coefficients** — The block uses the **Polynomial coefficients** parameter to calculate the nonlinear voltage gain.
- **AM/AM-AM/PM table** — The block uses the **AM/AM-AM/PM table** parameter to determine the power characteristics of the amplifier.

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

$$V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + \dots + a_9V_{in}^9$$

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 $V_{out} = a_0 + a_1V_1 + a_2V_1^2 + a_3V_1^3$. Trailing zeroes are omitted. So, $[a_0, a_1, a_2]$ 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 $V_{out} = V_{in}$.

Dependencies

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

AM/AM-AM/PM table — Lookup table

[-25, 5, -1; -10, 20, -2; 0, 27, 5; 5, 28, 12] (default) | *M*-by-3 real matrix

Table lookup entries specified as a real *M*-by-3 matrix. The second column of the matrix represents the AM/AM-AM/PM model output power in dBm. The third column represents the model phase change in degrees. The second and third column values are related to the absolute value of the power of the input signal represented in the first column of matrix. The input power in the first column must increase monotonically.

Dependencies

To enable this parameter, set **Source of amplifier gain** to **AM/AM - AM/PM table**.

Network-parameter object — Network parameter objects as data source

off (default) | on

Select this parameter to use a network parameter object as the source of amplifier gain.

Once you select this parameter, in the text box that opens, specify one of these:

- Network parameter object.
- MATLAB® base workspace variable or a Simulink® model workspace variable. The block evaluates if the specified variable is a network parameter object.

The network parameter objects that you can specify include:

- sparameters
- yparameters
- zparameters
- gparameters
- hparameters

- abcdparameters
- tparameters

Dependencies

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

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** and then, select `Network-parameters` in the **Data source**.

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**.

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

Output impedance of amplifier, specified as a scalar in ohms.

Dependencies

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

Data file — Name of network parameter data file
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** and 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** and 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 in ohms.

Dependencies

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

Use rational object — Rational object as data source
off (default) | on

Select this parameter to use a `rational` object as the data source to model amplifier gain.

Once you select this parameter, in the text box that opens, specify one of these:

- `rational` object.
- MATLAB base workspace variable or a Simulink model workspace variable. The block evaluates if the specified variable is a rational object."

The `rational` objects that you can specify include:

- `rational`
- `rationalfit`
- `rfmodel.rational`

Dependencies

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

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.

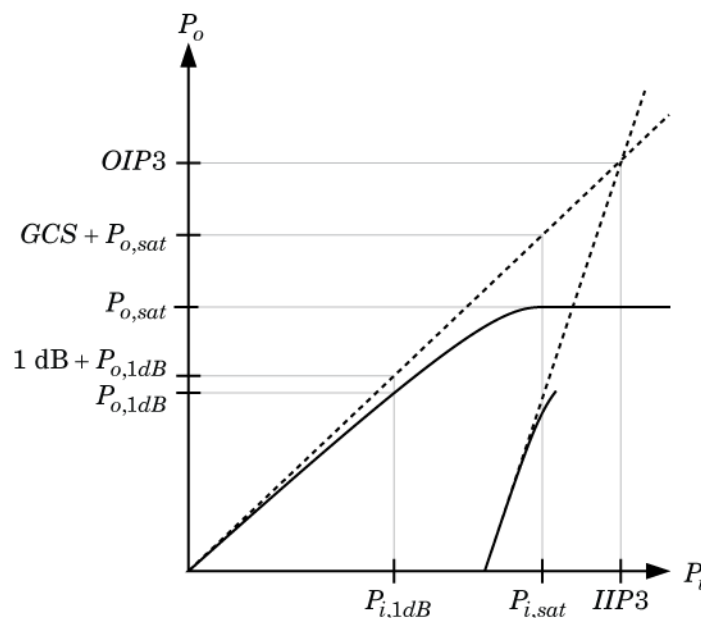
Nonlinearity

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

inf dBm (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

inf dBm (default) | scalar | W | mW | dBW | dBm

Third-order intercept point, specified as a scalar.

1-dB gain compression power — 1-dB gain compression power

inf dBm (default) | scalar | W | mW | dBW | dBm

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

Dependencies

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

Output saturation power — Output saturation power

inf dBm (default) | scalar | W | mW | dBW | dBm

Output saturation power, specified as scalar. The block uses this value to calculate the voltage saturation point used in the nonlinear model. In this case, the first derivative of the polynomial is zero, and the second derivative is negative.

Dependencies

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

Gain compression at saturation — Gain compression at saturation

inf dBm (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**

Use operation frequency at maximum S21 magnitude — Specify operation frequency

off (default) | on

Select this parameter to specify operation frequency at the maximum magnitude of S21.

Dependencies

To enable this parameter, first set **Source of amplifier gain** in the **Main** tab to **Data** source and then specify the IP2 value in **IP2** in the **Nonlinearity** tab.

Operation frequency — Operation frequency

1 GHz (default) | scalar | finite vector | Hz | KHz | MHz | GHz

Operation frequency, specified as a scalar or finite vector.

Dependencies

To enable this parameter, first choose either Available power gain, Open circuit voltage gain, or Data source in **Source of amplifier gain** in the **Main** tab. Second specify an IP2 value in **IP2** in the **Nonlinearity** tab. And third clear the **Pick operation frequency at maximum S21 magnitude** check box.

Use constant S21 and nonlinearity — Operation frequency

off (default) | on

Select this parameter to use the constant magnitude of S21 and nonlinearity in your amplifier model.

Dependencies

To enable this parameter, first set **Source of amplifier gain** in the **Main** tab to Data source and then, specify an IP2 value in **IP2** in the **Nonlinearity** tab.

Plot Power Characteristics — Amplifier nonlinear power curve visualization

button (default)

Use this button to plot the power characteristics of the amplifier based on the nonlinearity parameters specified on the **Nonlinearity** tab. When you clear the **Use constant S21 and nonlinearity** parameter, the power curve reflects the amplifier power measured at the frequency specified in the **Operation frequency** parameter.

Dependencies

To enable this parameter, set **Source of amplifier gain** in the **Main** tab to Available power gain, Open circuit voltage gain, or Data source.

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 — Frequency data

0 Hz (default) | real scalar | finite vector

Frequency data, specified as a real scalar or finite vector in Hz, kHz, MHz, or GHz.

DependenciesTo set this parameter, first select **Piece-wise linear** or **Colored** in **Noise distribution**.**Minimum noise figure (dB)** — Minimum noise figure

0 (default) | scalar | finite vector

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

DependenciesTo set this parameter, first select **Spot noise data** in **Noise type**.**Optimal reflection coefficient** — Optimal reflection coefficient

0 (default) | scalar | finite vector

Optimal reflection coefficient, specified as a scalar or a finite vector.

DependenciesTo set this parameter, first select **Spot noise data** in **Noise type**.**Equivalent normalized noise resistance** — Equivalent normalized noise resistance

0 (default) | scalar | finite vector

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

DependenciesTo set this parameter, first select **Spot noise data** in **Noise type**.**Automatically estimate impulse response duration** — 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.

DependenciesTo 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 dependent 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 “Compare Time and Frequency Domain Simulation Options for S-parameters”.
- For the Amplifier, Antenna, and S-Parameters blocks, the default value is Time domain (rationalfit). For the Transmission Line block, the default value is Frequency domain.

Dependencies

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

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

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

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

Dependencies

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

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

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

Dependencies

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

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

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

Dependencies

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

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

Impulse response duration, specified as a scalar in seconds.

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 parameter 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) | finite vector | Hz | kHz | MHz | GHz

Frequency data range, specified as a finite vector.

Dependencies

To enable this parameter, set **Source of frequency data** to User-specified.

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

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

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

Parameter 1 — Type of S-Parameters to plot

S11 (default) | S12 | S21 | S22 | NF

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

Dependencies

To enable NF, set **Noise type** to Noise figure and select **Apply**.

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.

Dependencies

To enable NF, set **Noise distribution** to Piece-wise linear or Colored and select **Apply**.

Format1 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Real | Imaginary

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

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Format2 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Real | Imaginary

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

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Y-axis scale — Y-axis scale

Linear (default) | Logarithmic

Y-axis scale, specified as Linear or Logarithmic.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X-axis scale — X-axis scale

Linear (default) | Logarithmic

X-axis scale, specified as Linear or Logarithmic.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot specified data using plot button.

More About

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:

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

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

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

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

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:

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

- 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, $\Gamma_{\text{opt}} (Y_{\text{opt}} = Y_0 (1 - \Gamma_{\text{opt}}) / (1 + \Gamma_{\text{opt}}))$.

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. Otherwise 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:

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

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

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

Version History

Introduced in R2010b

R2021b: Amplifier block icon updated

Behavior changed in R2021b

Starting in R2021b, the Amplifier block icon has updated. The block icon is now dynamic and shows the current state of the noise parameter.

When you open a model created before R2021b containing an Amplifier block, the software replaces the block icon with the R2021b version.

R2022a: Modeling noise in nonlinear circuit envelope amplifier

Behavior changed in R2022a

The Amplifier block now enables you to model frequency-dependent noise as colored noise when your amplifier operates in a nonlinear region.

When you open a model created before R2022a containing the Amplifier block operating in a nonlinear region with frequency-dependent noise, the software models the noise as specified in the block.

R2022b: Three new parameters added

The Amplifier block has three new parameters.

- **Network-parameter object** — Use this parameter to specify a network parameter object as the source of amplifier gain.
- **Use rational object** — Use this parameter to specify a `rational` object as the source of amplifier gain.

- **Plot power characteristics** — Use this button to plot the power characteristics of an amplifier based on the specified nonlinearities.

R2022b: Two block parameters renamed

Behavior changed in R2022b

The **Pick operation frequency at maximum S21 magnitude** and **Frequency data** parameters have been renamed to **Use operation frequency at maximum S21 magnitude** and **Source of frequency data**, respectively .

When you open a model created before R2022b containing the Amplifier block, the software replaces the old parameter names with the new names.

R2022a: Two new parameters added

The Amplifier block has two new parameters.

- **AM/AM-AM/PM table** — Use this parameter to specify amplifier power characteristics using AM/AM-AM/PM table.
- **Use constant S21 and nonlinearity** — Use this parameter to model frequency-dependent gain (S21) when operating in a nonlinear region.

References

- [1] Gonzalez, Guillermo. “Microwave Transistor Amplifiers: Analysis and Design”, Englewood Cliffs, N.J.: Prentice-Hall, 1984.
- [2] Grob, Siegfried and Juergen Lindner. “Polynomial Model Derivation of Nonlinear Amplifiers, *Department of Information Technology*, University of Ulm, Germany.
- [3] Kundert, Ken. “Accurate and Rapid Measurement of IP_2 and IP_3 ”, *The Designers Guide Community*, Version 1b, May 22, 2002. <http://www.designers-guide.org/analysis/intercept-point.pdf>.
- [4] Pozar, David M. “Microwave Engineering”, Hoboken NJ: John Wiley & Sons, 2005.

See Also

Mixer | S-Parameters | Power Amplifier

Variable Resistor

Model variable resistor



Libraries:
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 (R_{min}) 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.

Variable Resistor block mask icons are dynamic and show the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Simulate noise** parameter on the block.

Simulate noise: on	Simulate noise: off

Ports

Input

R — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: double

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 $4 kT/R$ A²/Hz, where:

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

Version History

Introduced in R2017b

R2021b: Variable Resistor block icon updated

Behavior changed in R2021b

Starting in R2021b, the Variable Resistor block icon is now updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Variable Resistor block, the software replaces the block icon with the R2021b version.

See Also

Variable Inductor | Resistor | Variable Capacitor

Power Amplifier

Model power amplifier with memory



Libraries:

RF Blockset / Circuit Envelope / Elements

Description

The Power Amplifier block models two-port 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 and the input signal at previous times. These power amplifiers are useful when transmitting wideband or narrowband signals.

Parameters

Model — Model type

Memory polynomial (default) | Generalized Hammerstein | Cross-Term Memory | Cross-Term Hammerstein

Model type, specified as `Memory polynomial`, `Generalized Hammerstein`, `Cross-Term Memory`, or `Cross-Term Hammerstein`. The following table summarizes the characteristics of the different models:

Model	Characterization Data	Type of Coefficients	In-Band Spectral Regrowth	Out-of-Band Harmonic Generation
Memory polynomial (default)	Bandpass (I,Q)	Complex	Yes	No
Generalized Hammerstein	True passband	Real	Yes	Yes
Cross-Term Memory	Bandpass (I,Q)	Complex	Yes	No
Cross-Term Hammerstein	True passband	Real	Yes	Yes

- `Memory polynomial` - This narrowband memory polynomial implementation (equation (19) of [1]) operates on the envelope of the input signal, does not generate new frequency components, and captures in-band spectral regrowth. Use this model to create a narrowband amplifier operating at high frequency.

The output signal, at any instant of time, is the sum of all the elements of the following complex matrix of dimensions `Memory Depth (mem)×Voltage Order (deg)`:

$$\begin{bmatrix} C_{11}V_0 & C_{12}V_0|V_0| & \dots & C_{1,\text{deg}}V_0|V_0|^{\text{deg}-1} \\ C_{21}V_1 & C_{22}V_1|V_1| & \dots & C_{2,\text{deg}}V_1|V_1|^{\text{deg}-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{\text{mem},1}V_{\text{mem}-1} & C_{\text{mem},2}V_{\text{mem}-1}|V_{\text{mem}-1}| & \dots & C_{\text{mem},\text{deg}}V_{\text{mem}-1}|V_{\text{mem}-1}|^{\text{deg}-1} \end{bmatrix}.$$

In the matrix, the number of rows equals the number of memory terms, and the number of columns equals the degree of the nonlinearity. The signal subscript represents amount of delay.

- **Generalized Hammerstein** - This wideband memory polynomial implementation (equation (18) of [1]) operates on the envelope of the input signal, generates frequency components that are integral multiples of carrier frequencies, and captures in-band spectral regrowth. Increasing the degree of the nonlinearity increases the number of out-of-band frequencies generated. Use this model to create wideband amplifiers operating at low frequency.

The output signal, at any instant of time, is the sum of all the elements of the following real matrix of dimensions Memory Depth (mem)×Voltage Order (deg):

$$\begin{bmatrix} C_{11}V_0 & C_{12}V_0^2 & \dots & C_{1,\text{deg}}V_0^{\text{deg}} \\ C_{21}V_1 & C_{22}V_1^2 & \dots & C_{2,\text{deg}}V_1^{\text{deg}} \\ \vdots & \vdots & \ddots & \vdots \\ C_{\text{mem},1}V_{\text{mem}-1} & C_{\text{mem},2}V_{\text{mem}-1}^2 & \dots & C_{\text{mem},\text{deg}}V_{\text{mem}-1}^{\text{deg}} \end{bmatrix}.$$

In the matrix, the number of rows equals the number of memory terms, and the number of columns equals the degree of the nonlinearity. The signal subscript represents amount of delay.

- **Cross-Term Memory** - This narrowband memory polynomial implementation (equation (23) of [1]) operates on the envelope of the input signal, does not generate new frequency components, and captures in-band spectral regrowth. Use this model to create a narrowband amplifier operating at high frequency. The model includes leading and lagging memory terms and provides a generalized implementation of the memory polynomial model.

The output signal, at any instant of time, is the sum of all the elements of a matrix specified by the element-by-element product

$$\mathbf{C}.*\mathbf{M}_{\text{CTM}},$$

where \mathbf{C} is a complex coefficient matrix of dimensions

Memory Depth (mem) × {Memory Depth (mem) · (Voltage Order (deg) – 1) + 1} and

$$\mathbf{M}_{\text{CTM}} = \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{\text{mem}-1} \end{bmatrix} \begin{bmatrix} 1 & |V_0| & |V_1| & \dots & |V_{\text{mem}-1}| & |V_0|^2 & \dots & |V_{\text{mem}-1}|^2 & \dots & |V_0|^{\text{deg}-1} & \dots & |V_{\text{mem}-1}|^{\text{deg}-1} \end{bmatrix}$$

$$= \begin{bmatrix} V_0 & V_0|V_0| & V_0|V_1| & \dots & V_0|V_{\text{mem}-1}| & V_0|V_0|^2 & \dots & V_0|V_{\text{mem}-1}|^2 & \dots & V_0|V_0|^{\text{deg}-1} & \dots & V_0|V_{\text{mem}-1}|^{\text{deg}-1} \\ V_1 & V_1|V_0| & V_1|V_1| & \dots & V_1|V_{\text{mem}-1}| & V_1|V_0|^2 & \dots & V_1|V_{\text{mem}-1}|^2 & \dots & V_1|V_0|^{\text{deg}-1} & \dots & V_1|V_{\text{mem}-1}|^{\text{deg}-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ V_{\text{mem}-1} & V_{\text{mem}-1}|V_0| & V_{\text{mem}-1}|V_1| & \dots & V_{\text{mem}-1}|V_{\text{mem}-1}| & V_{\text{mem}-1}|V_0|^2 & \dots & V_{\text{mem}-1}|V_{\text{mem}-1}|^2 & \dots & V_{\text{mem}-1}|V_0|^{\text{deg}-1} & \dots & V_{\text{mem}-1}|V_{\text{mem}-1}|^{\text{deg}-1} \end{bmatrix}$$

In the matrix, the number of rows equals the number of memory terms, and the number of columns is proportional to the degree of the nonlinearity and the number of memory terms. The

signal subscript represents amount of delay. The additional columns that do not appear in the Memory polynomial model represent the cross terms.

- **Cross-Term Hammerstein** - This wideband memory polynomial implementation operates on the envelope of the input signal, generates frequency components that are integral multiples of carrier frequencies, and captures in-band spectral regrowth. Increasing the order of the nonlinearity increases the number of out-of-band frequencies generated. Use this model to create wideband amplifiers operating at low frequency.

The output signal, at any instant of time, is the sum of all the elements of a matrix specified by the element-by-element product

$$\mathbf{C} .* \mathbf{M}_{\text{CTH}},$$

where \mathbf{C} is a complex coefficient matrix of dimensions

Memory Depth (mem) \times {Memory Depth (mem) \cdot (Voltage Order (deg) - 1) + 1} and

$$\mathbf{M}_{\text{CTH}} = \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{\text{mem}-1} \end{bmatrix} \begin{bmatrix} 1 & V_0 & V_1 & \dots & V_{\text{mem}-1} & V_0^2 & \dots & V_{\text{mem}-1}^2 & \dots & V_0^{\text{deg}-1} & \dots & V_{\text{mem}-1}^{\text{deg}-1} \end{bmatrix}$$

$$= \begin{bmatrix} V_0 & V_0^2 & V_0 V_1 & \dots & V_0 V_{\text{mem}-1} & V_0^3 & \dots & V_0 V_{\text{mem}-1}^2 & \dots & V_0^{\text{deg}} & \dots & V_0 V_{\text{mem}-1}^{\text{deg}-1} \\ V_1 & V_1 V_0 & V_1^2 & \dots & V_1 V_{\text{mem}-1} & V_1 V_0^2 & \dots & V_1 V_{\text{mem}-1}^2 & \dots & V_1 V_0^{\text{deg}-1} & \dots & V_1 V_{\text{mem}-1}^{\text{deg}-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ V_{\text{mem}-1} & V_{\text{mem}-1} V_0 & V_{\text{mem}-1} V_1 & \dots & V_{\text{mem}-1}^2 & V_{\text{mem}-1} V_0^2 & \dots & V_{\text{mem}-1}^3 & \dots & V_{\text{mem}-1} V_0^{\text{deg}-1} & \dots & V_{\text{mem}-1}^{\text{deg}} \end{bmatrix}.$$

In the matrix, the number of rows equals the number of memory terms, and the number of columns is proportional to the degree of the nonlinearity and the number of memory terms. The signal subscript represents amount of delay. The additional columns that do not appear in the Generalized Hammerstein model represent the cross terms.

Coefficient Matrix — Coefficient matrix

1 (default) | complex matrix | real matrix

Coefficient matrix, specified as a complex matrix for the Memory polynomial and Cross-Term Memory models and as a real matrix for the Generalized Hammerstein and Cross-Term Hammerstein models.

- For the Memory polynomial and Cross-Term Memory models, you can identify the complex coefficient matrix based on the measured complex (I,Q) output-vs.-input amplifier characteristic. As an example, see the helper function in “Coefficient Matrix Computation” on page 1-22.
- For the Generalized Hammerstein and Cross-Term Hammerstein models, you can identify the real coefficient matrix based on the measured real passband output-vs.-input amplifier characteristic.

The size of the matrix depends on the number of delays and the degree of the system nonlinearity.

- For the Memory polynomial and Generalized Hammerstein models, the matrix has dimensions Memory Depth (mem) \times Voltage Order (deg).
- For the Cross-Term Memory and Cross-Term Hammerstein models, the matrix has dimensions Memory Depth (mem) \times {Memory Depth (mem) \cdot (Voltage Order (deg) - 1) + 1}.

Coefficient Sample Time (s) — Sample interval of input-output data
 1e-6 (default) | real positive scalar

Sample interval of input-output data to define the coefficient matrix, specified as a real positive scalar.

Note To improve model accuracy, specify the **Coefficient Sample Time (s)** parameter to be equal or larger than the **Step size** parameter in the Configuration block.

Rin (ohm) — Input resistance
 50 (default) | real positive scalar

Input resistance, specified as a real positive scalar.

Rout (ohm) — Output resistance
 50 (default) | real positive scalar

Output resistance, specified as a real positive scalar.

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.

Tips

- To avoid the nonlinear power amplifier to operate in an undesirable region, the Simulink® input signal must be scaled. This happens when a nonlinear power amplifier in the RF domain is used to amplify a Simulink signal.

Algorithms

Coefficient Matrix Computation

To compute coefficient matrices, the block solves an overdetermined linear system of equations. Consider the **Memory polynomial** model for the case where the memory length is 2 and the system nonlinearity is of third degree. The matrix that describes the system is

$$\begin{bmatrix} C_{11}V_0 & C_{12}V_0|V_0| & C_{13}V_0|V_0|^2 \\ C_{21}V_1 & C_{22}V_1|V_1| & C_{23}V_1|V_1|^2 \end{bmatrix}'$$

and the sum of its elements is equivalent to the inner product

$$\begin{bmatrix} V_0 & V_1 & V_0|V_0| & V_1|V_1| & V_0|V_0|^2 & V_1|V_1|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \end{bmatrix}.$$

If the input to the amplifier is the five-sample signal $[x(1) x(2) x(3) x(4) x(5)]$ and the corresponding output is $[y(1) y(2) y(3) y(4) y(5)]$, then the solution to

$$\begin{bmatrix} x(2) & x(1) & x(2)|x(2)| & x(1)|x(1)| & x(2)|x(2)|^2 & x(1)|x(1)|^2 \\ x(3) & x(2) & x(3)|x(3)| & x(2)|x(2)| & x(3)|x(3)|^2 & x(2)|x(2)|^2 \\ x(4) & x(3) & x(4)|x(4)| & x(3)|x(3)| & x(4)|x(4)|^2 & x(3)|x(3)|^2 \\ x(5) & x(4) & x(5)|x(5)| & x(4)|x(4)| & x(5)|x(5)|^2 & x(4)|x(4)|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \end{bmatrix} = \begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix},$$

which can be found using the MATLAB backslash operator, provides an estimate of the coefficient matrix.

The treatment of the Cross - Term Memory model is similar. The matrix that describes the system is

$$\begin{bmatrix} C_{11}V_0 & C_{12}V_0|V_0| & C_{13}V_0|V_1| & C_{14}V_0|V_0|^2 & C_{15}V_0|V_1|^2 \\ C_{21}V_1 & C_{22}V_1|V_0| & C_{23}V_1|V_1| & C_{24}V_1|V_0|^2 & C_{25}V_1|V_1|^2 \end{bmatrix},$$

and the sum of its elements is equivalent to the inner product

$$\begin{bmatrix} V_0 & V_1 & V_0|V_0| & V_1|V_0| & V_0|V_1| & V_1|V_1| & V_0|V_0|^2 & V_1|V_0|^2 & V_0|V_1|^2 & V_1|V_1|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \\ C_{14} \\ C_{24} \\ C_{15} \\ C_{25} \end{bmatrix}.$$

If the input to the amplifier is the five-sample signal $[x(1) x(2) x(3) x(4) x(5)]$ and the corresponding output is $[y(1) y(2) y(3) y(4) y(5)]$, then the solution to

$$\begin{bmatrix} x(2) & x(1) & x(2)|x(2)| & x(1)|x(2)| & x(2)|x(1)| & x(1)|x(1)| & x(2)|x(2)|^2 & x(1)|x(2)|^2 & x(2)|x(1)|^2 & x(1)|x(1)|^2 \\ x(3) & x(2) & x(3)|x(3)| & x(2)|x(3)| & x(3)|x(2)| & x(2)|x(2)| & x(3)|x(3)|^2 & x(2)|x(3)|^2 & x(3)|x(2)|^2 & x(2)|x(2)|^2 \\ x(4) & x(3) & x(4)|x(4)| & x(3)|x(4)| & x(4)|x(3)| & x(3)|x(3)| & x(4)|x(4)|^2 & x(3)|x(4)|^2 & x(4)|x(3)|^2 & x(3)|x(3)|^2 \\ x(5) & x(4) & x(5)|x(5)| & x(4)|x(5)| & x(5)|x(4)| & x(4)|x(4)| & x(5)|x(5)|^2 & x(4)|x(5)|^2 & x(5)|x(4)|^2 & x(4)|x(4)|^2 \end{bmatrix}$$

$$\begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \\ C_{14} \\ C_{24} \\ C_{15} \\ C_{25} \end{bmatrix} = \begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix}$$

provides an estimate of the coefficient matrix.

Use this helper function to compute coefficient matrices for the Memory polynomial and Cross-Term Memory models. The inputs to the function are the given input and output signals, the memory length, the degree of nonlinearity, and the absence or presence of cross terms.

```
function a_coef = fit_memory_poly_model(x,y,memLen,degLen,modType)
% FIT_MEMORY_POLY_MODEL
% Procedure to compute a coefficient matrix given input and output
% signals, memory length, nonlinearity degree, and model type.
%
% Copyright 2017 MathWorks, Inc.

x = x(:);
y = y(:);
xLen = length(x);

switch modType
case 'memPoly' % Memory polynomial
    xrow = reshape((memLen:-1:1)' + (0:xLen:xLen*(degLen-1)),1,[]);
    xVec = (0:xLen-memLen)' + xrow;
    xPow = x.*(abs(x).^(0:degLen-1));
    xVec = xPow(xVec);
case 'ctMemPoly' % Cross-term memory polynomial
    absPow = (abs(x).^(1:degLen-1));
    partTop1 = reshape((memLen:-1:1)' + (0:xLen:xLen*(degLen-2)),1,[]);
    topPlane = reshape(
    ...
```

```





        [ones(xLen-memLen+1,1),absPow((0:xLen-memLen)' + partTop1)].', ...
        1,memLen*(degLen-1)+1,xLen-memLen+1);
    sidePlane = reshape(x((0:xLen-memLen)' + (memLen:-1:1)).', ...
        memLen,1,xLen-memLen+1);
    cube = sidePlane.*topPlane;
    xVec = reshape(cube,memLen*(memLen*(degLen-1)+1),xLen-memLen+1).';
end

coef = xVec\y(memLen:xLen);
a_coef = reshape(coef,memLen,numel(coef)/memLen);

```

Power Amplifier Block Icons

Power Amplifier block mask icons are dynamic and show the current type of model. This table shows you how the icons on this block vary based on the model type you set on the **Model** parameter on the block.

Model: Generalized Hammerstein	Output: Memory polynomial	Output: Cross-Term Memory	Output: Cross-Term Hammerstein
			

Version History

Introduced in R2017b

R2021b: Power Amplifier block icon updated

Behavior changed in R2021b

Starting in R2021b, the Power Amplifier block icon is now updated. When you open a model created before R2021b containing a Power Amplifier block, the software replaces the block icon with the R2021b version.

References

- [1] Morgan, Dennis R., Zhengxiang Ma, Jaehyeong Kim, Michael G. Zierdt, and John Pastalan. "A Generalized Memory Polynomial Model for Digital Predistortion of Power Amplifiers." *IEEE® Transactions on Signal Processing*. Vol. 54, No. 10, October 2006, pp. 3852-3860.
- [2] Gan, Li, and Emad Abd-Elrady. "Digital Predistortion of Memory Polynomial Systems using Direct and Indirect Learning Architectures." In *Proceedings of the Eleventh IASTED International Conference on Signal and Image Processing (SIP)* (F. Cruz-Roldán and N. B. Smith, eds.), No. 654-802. Calgary, AB: ACTA Press, 2009.

See Also

Amplifier

Topics

"Power in Simulink Sources and Signals"

"Power Ports and Signal Power Measurement in RF Blockset"

Attenuator

Model attenuator for RF circuit



Libraries:
RF Blockset / Circuit Envelope / Elements

Description

The Attenuator block 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.

Attenuator block mask icons are dynamic and indicate the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Simulate noise** parameter on the block.

Simulate noise: on	Simulate noise: off

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 $4 kT/R$ A²/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.

Version History

Introduced in R2016b**R2021b: Attenuator block icon updated***Behavior changed in R2021b*

Starting in R2021b, the Attenuator block icon is now updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing an Attenuator block, the software replaces the block icon with the R2021b version.

See Also

Variable Attenuator

Capacitor

Model capacitor for circuit envelope analysis



Libraries:
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.

Version History

R2021b: Capacitor block icon updated

Behavior changed in R2021b

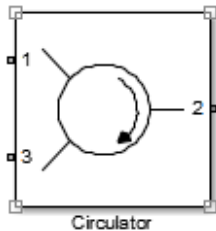
Starting in R2021b, the Capacitor block icon has updated. When you open a model created before R2021b containing a Capacitor block, the software replaces the block icon with the R2021b version.

See Also

Resistor | Inductor

Circulator

Model ideal frequency-independent circulators with S-parameters



Libraries:

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) | Circulator counter clockwise | Tee H-plane (S11=0) | Reciprocal phase shifter

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:

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

- Tee H-plane (S11=0)

The s-parameter matrix for `Tee H-plane (S11=0)` is:

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

- Reciprocal phase shifter

The s-parameter matrix for Reciprocal phase shifter is:

$$\begin{pmatrix} 0 & phase_{12} & 0 \\ phase_{12} & 0 & 0 \\ 0 & 0 & phase_{33} \end{pmatrix} \left(\frac{-j}{\sqrt{2}} \right)$$

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.

Phase shift, ports 1 and 2 (rad) — Angle to shift phase of signal in port 1 and port 2
0 (default) | positive scalar

Angle to shift the phase of the signal in port 1 and port 2, specified as a positive scalar in radians.

Phase shift, ports 3 (rad) — Angle to shift phase of signal in port 3
0 (default) | positive scalar

Angle to shift the phase of the signal in port 3, specified as a positive scalar in radians.

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.

Version History

Introduced in R2014a

See Also

Coupler | Divider

Configuration

Define system simulation settings



Libraries:
RF Blockset / Circuit Envelope / Utilities

Description

Use the Configuration block to set the model conditions for a circuit envelope simulation. The block parameter defines RF and solver attributes. The RF attributes include properties such as simulation frequencies, harmonic order, envelope bandwidth, and thermal noise. The solver attributes include transient analysis types, tolerances, and small signal approximation.

The small signal transient simulation performs a full non-linear harmonic balance steady state solution to determine an operation point for a subsequent linear transient analysis. This option allows you to capture the right spectral behavior of a small signal affected by large constant (over carrier) signals.

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 “Simulate High Frequency Components”.

Configuration block mask icons are dynamic and show the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Simulate noise** parameter on the block.

Simulate noise: on	Simulate noise: off

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 simulation frequencies to capture the non-linear behavior 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: $[m*f1 + n*f2 + \dots]$. In this case, the fundamental tones are represented by $[f_s1, f2, \dots]$, and the integers m and n are integers bounded by the corresponding **Harmonic order**, $|m| \leq h1$, $|n| \leq h2$, 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: $[0, f1, 2f1, 3f1] = [0\text{GHz}, 2\text{ GHz}, 4\text{ GHz}, 6\text{GHz}]$.

Consider a circuit with two fundamental tones $[f1 = 2\text{ GHz}, f2 = 50\text{ MHz}]$ 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 $[f1 = 2\text{ GHz}, f2 = 3\text{GHz}]$ 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 centered 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 pseudo random noise stream for RF Blockset sources. Clear this option to specify an independent pseudo random 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 pseudo random number stream
0 (default) | scalar positive integer

Seed of the independent pseudo random 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.

Samples per frame — Number of samples in each channel of input signal to Inport block
1 | positive scalar integer

Number of samples in each channel of input signal to Inport block, specified as a positive scalar integer less than or equal to 1024. A channel corresponds to an input frequency in the Inport block.

Note To improve system performance, the recommended maximum number of **Samples per frame** is 1024. To work with larger frames, consider one of the following:

- Use `rfsystem` System object™, which supports vector inputs for frame-based processing without any frame-size limits.
- Use a Buffer block before and after the Circuit Envelope Inport and Outport blocks. For more information, see “Buffering and Frame-Based Processing” (DSP System Toolbox).

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

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

$$I^2 + Q^2$$

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

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

where:

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

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

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

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

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

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

Enable input interpolation filter — Automatic interpolation of lower rate baseband signal to higher rate RF signal

on (default) | off

Select this option to enable input interpolation filter to up sample the input signal rates to fit sample rate of the RF solver. You can now directly use baseband communication signals using a lower sample rate in a wider band circuit. This filter introduces a delay in the RF signal. **Filter delay (in samples)** shows the delay introduced after you simulate the model.

Note When you enable this filter,

- The RF-to-baseband sample rate ratio must be 2, 4, 6, or 8.
 - The RF Blockset model can have only one Inport block.
-

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.

Approximate transient as small signal — Choose small subset of frequencies for transient small signal analysis

off (default) | on

Select this option to choose a small subset of frequencies for transient small signal analysis.

Use all steady-state simulation frequencies for small signal analysis — Choose all steady-state simulation frequencies
on (default) | off

Select this option to choose all steady-state simulation frequencies. Clear this option to specify the frequencies for small signal transient simulation.

Dependencies

To expose this parameter, check **Approximate transient as small signal**.

Small signal frequencies — Frequencies for small signal transient simulation
scalar | vector

Frequencies for small signal transient simulation, specified as a scalar or vector. The frequencies specified are contained in the entire set of simulation frequencies determined from **Fundamental tones** and **Harmonic order** in the **Main** tab.

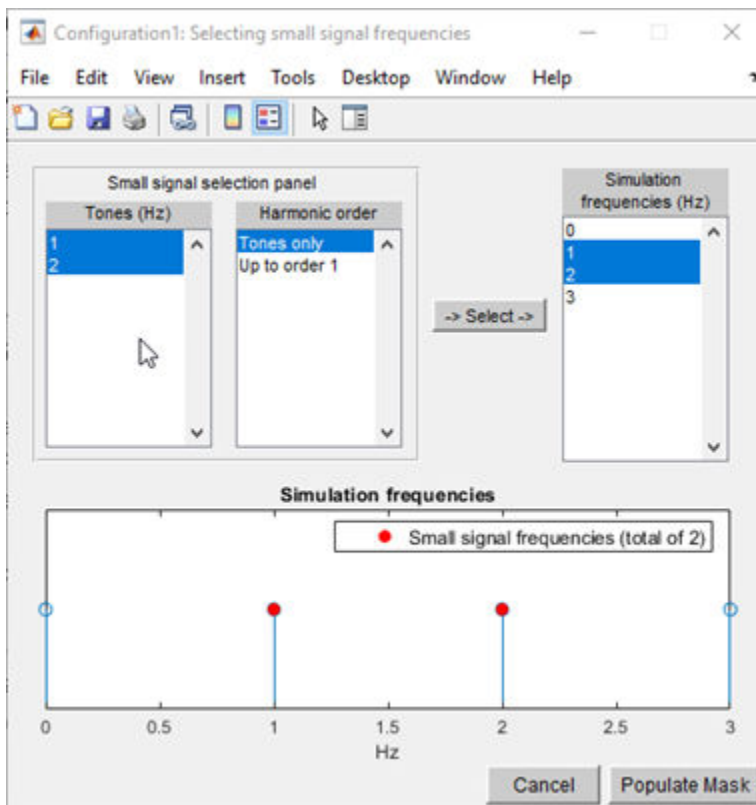
The default values in this box and the corresponding units are not constants. The values depend on the state of the configuration dialog box when **Use all steady-state simulation frequencies for small signal analysis** is first cleared.

Dependencies

To expose this parameter, clear **Use all steady-state simulation frequencies for small signal analysis**.

Populate Frequencies — Tool to choose small signal transient frequencies
button

Tool to choose small signal transient frequencies to populate **Small signal frequencies**. The selected frequencies are a subset of the simulation frequencies determined from **Fundamental tones** and **Harmonic order** inputs in the **Main** tab. The entire set of simulation frequencies are given in the combo box on the right-hand side of the dialog box and the selected frequencies are highlighted. You can select by directly choosing the frequencies in the selection box, or by choosing the desired tones and harmonic order in the **Small signal selection panel** and pressing **Select**. The **Tones(Hz)** and **Harmonic order** values in the combo boxes are also populated using **Fundamental tones** and **Harmonic order** inputs in the **Main** tab.



Dependencies

To expose this parameter, clear **Use all steady-state simulation frequencies for small signal analysis**.

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.

More About

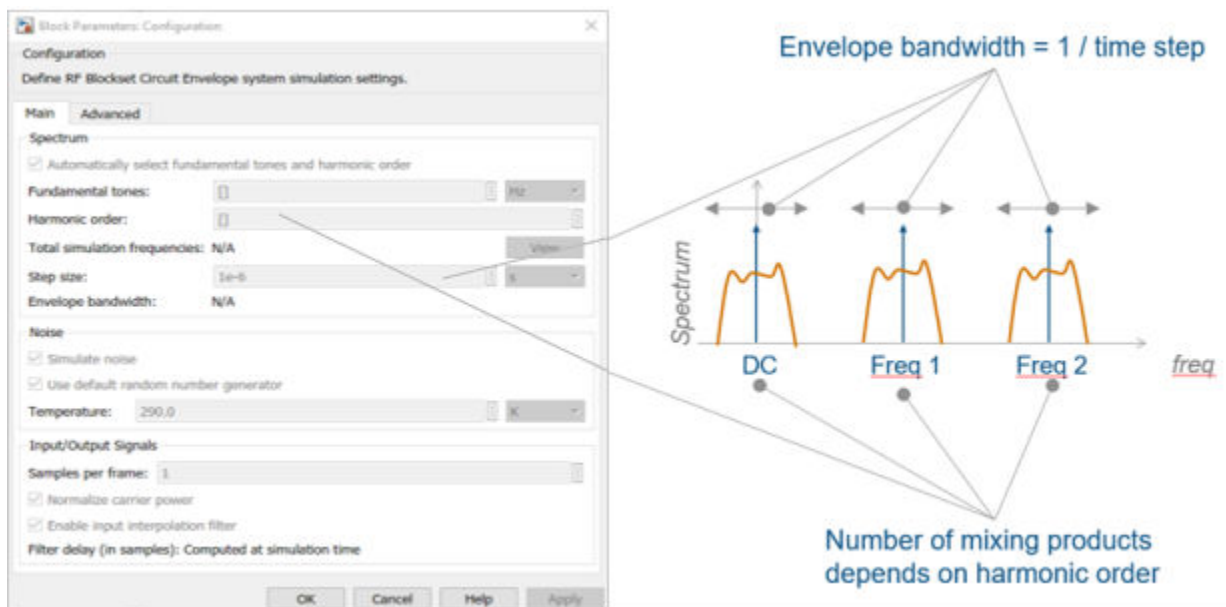
Simulation Setup and Complexity

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, centered 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 25 simulation frequencies), and a simulation step size equal to $1/200\text{MHz} = 5 \text{ ns}$.

You could also set up the RF subsystem so that both of the signals are within the same simulation bandwidth centered 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 $1/400\text{MHz} = 2.5 \text{ ns}$. 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 $1/20 \text{ MHz} = 50 \text{ ns}$. 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.

By default, the configuration block allows automatic interpolation of lower rate baseband signal to higher rate RF signal. If you disable this property, it is recommended that you 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 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 criteria:

- Variable error convergence:

$$|\Delta X| < \text{RelTol} \cdot \max_t(|X|) + \text{AbsTol}_x$$

where:

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

$$|F(X)| < \text{RelTol} \cdot \max_{t,n}(|F_n(X)|) + \text{AbsTol}_F$$

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'.

Version History

Introduced in R2013a

R2021b: Configuration block icon updated

Behavior changed in R2021b

Starting in R2021b, the Configuration block icon is now updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Configuration block, the software replaces the block icon with the R2021b version.

See Also

Inport | Outport

Continuous Wave

Model constant envelope source



Libraries:
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, “Simulate High Frequency Components”.

The block implements the relation

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

or

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

at the carrier ω_k , depending on the source type.

Continuous Wave block mask icons are dynamic and show the current state of the applied noise parameter. For more information, see “Continuous Wave Block Icons” on page 1-44.

Parameters

Source type — Wave type

Ideal voltage (default) | Ideal current | Power

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

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

where:

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

Constant in-phase value — In-phase modulations of carrier frequencies

0 V (default) | vector of real numbers | V | mV | kV

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

Dependencies

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

Constant quadrature value — Quadrature modulations of carrier frequencies

0 V (default) | vector of real numbers | V | mV | kV

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

Dependencies

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

Source impedance(Ohm) — Input impedance of source

50 (default) | positive real number | complex number

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

Dependencies

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

Available power — Input impedance of source

0 W (default) | vector of real numbers | W | mW | dBW | dBm

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

Dependencies

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

Angle(degrees) — Phase angle of power waves

0 (default) | vector of real numbers

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

Dependencies

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

Carrier frequencies — Carrier frequencies

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

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

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) — 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 — 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**.

Dependencies

To enable this parameter, select **Add phase noise**.

Impulse response duration — 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**.

Plot phase noise characteristics — Phase noise magnitude response button

The block plots the phase noise characteristics based on the parameters specified in the **Impairments** tab and either the **Envelope bandwidth** parameter in the Configuration block when available or the value specified in the **Phase noise frequency offset (Hz)** parameter.

Dependencies

To enable this parameter, select **Add phase noise**.

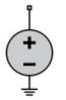
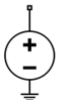
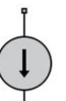
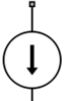
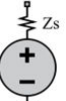
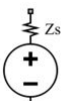
Ground and hide negative terminals — Ground RF circuit terminals on (default) | off

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.

Algorithms

Continuous Wave Block Icons

This table shows you how the icons on this block vary based on the state of the **Simulate noise** and **Source type** parameters on the block.

Source type	Simulate noise: on	Simulate noise: off
Ideal voltage		
Ideal current		
Power		

Version History

Introduced in R2010b

R2021b: Continuous Wave block icon updated

Behavior changed in R2021b

Starting in R2021b, the Continuous Wave block icon is now updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Continuous Wave block, the software replaces the block icon with the R2021b version.

R2022b: Estimate impulse response duration using phase noise offset frequencies

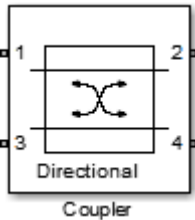
The Continuous Wave block estimates the impulse response duration using the **Phase noise frequency offset (Hz)** parameter and visualizes the phase noise characteristics when you click the **Plot phase noise characteristics** button.

See Also

Inport | Sinusoid | Noise

Coupler

Model ideal frequency-independent couplers with S-parameters



Libraries:

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 | Magic tee

Coupler type, specified as:

- Directional coupler

The default option is Directional coupler. The S-parameters matrix for the 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:

- $r_l = 10^{(-\text{ReturnLoss}/20)}$
- $i_l = j10^{(-\text{InsertionLoss}/20)}$
- $i_s = j10^{(-(\text{Coupling} + \text{Directivity})/20)}$
- $c = 10^{(-\text{Coupling}/20)}$

Note In release 2019a, the S-parameters representation of the Directional Coupler block is altered to permit a valid implementation for all specified mask values. This implementation adds a 90 degree phase shift to s12, s13, s21, s24, s31, s34, s42, and s43 terms. In addition, a -180 phase shift is added to the s14, s23, s32 and s41 terms.

Use this option to model coupler parameters from data sheets.

- Coupler symmetrical

The S-parameters matrix for the Coupler symmetrical is:

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

where:

- $|\alpha| \leq 1$ = Power transmission coefficient
- $\beta = \sqrt{1 - \alpha^2}$
- Coupler antisymmetrical

The S-parameters matrix for the Coupler antisymmetrical is:

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

where:

- $|\alpha| \leq 1$ = Power transmission coefficient.
- $\beta = \sqrt{1 - \alpha^2}$
- Hybrid quadrature (90deg)

The S-parameters matrix for the Hybrid quadrature(90deg) is:

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

- Hybrid rat-race

The S-parameters matrix for the Hybrid rat-race is:

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

- Magic tee

The S-parameters matrix for the Magic tee is:

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

The Divider block uses the `ispassive` function to test the passivity of the S-parameters matrix.

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 coefficient — 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.

Version History

Introduced in R2014a

See Also

Circulator | Divider

IQ Demodulator

Convert RF signal to baseband signal



Libraries:
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.

The IQ Demodulator block mask icons are dynamic and indicate the current state of the applied noise parameter. For more information, see “IQ Demodulator Icons” on page 1-63.

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** — Relates the ratio of the power of a single sideband (SSB) of 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** — Value of the open circuit voltage gain parameter as the linear voltage gain term of the polynomial voltage controlled voltage-source (VCVS).
- **Polynomial coefficients** — Implements a nonlinear voltage gain according to the polynomial you specify.

Available power gain — Ratio of power of SSB at output I branch to input power

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

Ratio of power of SSB at output I branch to input power, specified as a scalar in dB or a unitless ratio. For a unitless ratio, select 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 — Coefficients of polynomial specifying voltage gain
[0 1] (default) | vector

Polynomial coefficients, 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, [$a_0, a_1, a_2, \dots, a_9$], the polynomial it represents is:

$$V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + \dots + a_9V_{in}^9$$

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 $V_{out} = a_0 + a_1V_1 + a_2V_1^2 + a_3V_1^3$. Trailing zeros are omitted. So [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 $V_{out} = V_{in}$.

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.

Input impedance (Ohm) — Input impedance of IQ demodulator
50 (default) | scalar

Input impedance of IQ demodulator, specified as a scalar in Ohms.

Output impedance (Ohm) — Output impedance of IQ demodulator
50 (default) | scalar

Output impedance of IQ demodulator, specified as a scalar in Ohms.

Add Image Reject filter — Image reject (IR) filter parameters
off (default) | on

Select to add the **IR filter** parameter tab. Clear to remove the tab.

Add Channel Select filters — Channel select (CS) filter parameters
off (default) | on

Select to add the **CS filter** parameter tab. Clear to remove the tab.

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

Select to internally ground and hide the negative terminals. Clear to expose the negative terminals. When the terminals are exposed, you can connect them to other parts of your model.

Edit System — Break IQ demodulator block links and replace internal variables by appropriate values
button

Use this button to break IQ modulator links to the library. The internal variables are replaced by their values which are estimated using IQ modulator parameters. The IQ Modulator becomes a simple subsystem masked only to keep the icon.

Use **Edit System** to edit the internal variables without expanding the subsystem. Use **Expand System** to expand the subsystem in the Simulink canvas and to edit the subsystem.

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 the ratio of 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 $\omega_{IM} = \omega_{LO} + (\omega_{LO} - \omega_{RF})$.

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

Noise Convention	Signal at RF Frequency	Signal at IM Frequency	IQ Demodulator Block Supports This Model?
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.

Add phase noise — Add phase noise
off (default) | on

Select this parameter to add phase noise to your IQ demodulator system.

Phase noise frequency offset (Hz) — Phase noise frequency offset
1 (default) | scalar | vector | matrix

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

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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) — Phase noise level
-Inf (default) | scalar | vector | matrix

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

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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 — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, select **Add phase noise**.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns.

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. This message also specifies the minimum duration suitable for the required resolution.

Dependencies

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

Plot phase noise characteristics — Phase noise magnitude response button

The block plots the phase noise characteristics based in the parameters specified on the **Impairments** tab and either the **Envelope bandwidth** parameter in the Configuration block when available or the value specified in the **Phase noise frequency offset (Hz)** parameter.

Dependencies

To enable this parameter, select **Add phase noise**.

Nonlinearity

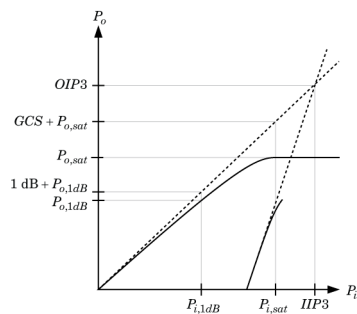
Selecting Polynomial coefficients for **Source of conversion gain** in the **Main** tab removes the **Nonlinearity** parameters.

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 IQ demodulator parameters.



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

Intercept points convention, specified as **Input** (input-referred) or **Output** (output-referred). Use this specification for the intercept points **IP2**, **IP3**, the **1-dB gain compression power**, and the **Output 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.

Dependencies

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

1-dB gain compression power — 1-dB gain compression power

`inf dBm (default) | scalar`

1-dB gain compression power, specified as a scalar in dBm, W, mW, or dBW. The 1-dB gain compression point must be less than the output saturation power.

Dependencies

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

Output saturation power — Output saturation power

`inf dBm (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, set **Odd** order in **Nonlinear polynomial type** tab.

Gain compression at saturation — Gain compression at saturation

`inf dBm (default) | scalar`

Gain compression at saturation, specified as a scalar.

Dependencies

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

IR Filter

Select **Add Image Reject filter** in the **Main** tab to see the **IR Filter** parameters tab.

Design method — Simulation type

Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**

Implementation — Implementation

LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Filter Domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is `Constant per carrier` for an ideal filter and `LC Tee` for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to `Ideal` and **Filter type** to `Lowpass` or `Highpass`.

Implement using filter order — Implement using filter order

on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order

3 (default) | scalar

Filter order, specified as a scalar. For a **Filter type** of Lowpass or Highpass, the filter order is the number of lumped storage elements. For a **Filter type** of Bandpass or Bandstop, the number of lumped storage elements is twice the filter order.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{\text{load}}}{R_{\text{source}}} > R_{\text{ratio}}$ for Tee network implementation and $\frac{R_{\text{load}}}{R_{\text{source}}} < \frac{1}{R_{\text{ratio}}}$ for Pi network implementation.

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

where:

- $\varepsilon = \sqrt{10^{(0.1R_p)} - 1}$
 - R_p is the passband ripple in dB.
-

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters

scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. The default value is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters

[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation

$10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation
40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance
50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance
50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to manually specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns. 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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

To enable this parameter, clear **Automatically estimate impulse response duration**.

Export — Save filter design to a file
button

Use this button to save filter design to a file. Valid file types are .mat and .txt.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

CS Filter

Select **Add Channel Select filters** in the **Main** tab to see the **CS Filter** parameters.

Design method — Simulation type
Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type
Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**.

Implementation — Implementation
LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Filter Domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is **Constant per carrier** for an ideal filter and **LC Tee** for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to **Ideal**.

Implement using filter order — Implement using filter order

on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order

3 (default) | scalar

Filter order, specified as a scalar. This order is the number of lumped storage elements in lowpass or highpass. In bandpass or bandstop, the number of lumped storage elements are twice the value.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{load}}{R_{source}} > R_{ratio}$ for Tee network

implementation and $\frac{R_{load}}{R_{source}} < \frac{1}{R_{ratio}}$ for Pi network implementation.

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

where:

- $\varepsilon = \sqrt{10^{(0.1R_p)} - 1}$
- R_p is the passband ripple in dB.

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters
scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. By default, the passband frequency is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters
[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation
 $10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation
40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance
50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance
50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

set **Design method** to Ideal and **Implementation** to Frequency domain.

Impulse response duration — 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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

To enable this parameter, clear **Automatically estimate impulse response duration**.

Export — Save filter design to a file
button

Use this button to save filter design to a file. Valid file types are `.mat` and `.txt`.





Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Algorithms

IQ Demodulator Icons

This table shows you how the icons on this block will vary based on how you set the **Noise figure (dB)** and **Add LO phase noise** parameters on the block.

Noise figure (dB)	Add LO phase noise: off	Add LO phase noise: on
0		
10		

Version History

Introduced in R2017a

R2021b: IQ Demodulator block icon updated

Behavior changed in R2021b

Starting in R2021b, the IQ Demodulator block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a IQ Demodulator block, the software replaces the block icon with the R2021b version.

R2022b: Estimate impulse response duration using phase noise offset frequencies

The IQ Demodulator block estimates the impulse response duration using the **Phase noise frequency offset (Hz)** parameter and visualizes the phase noise characteristics when you click the **Plot phase noise characteristics** button.

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and Lindner, Jurgen, "Polynomial Model Derivation of Nonlinear Amplifiers", *Department of Information Technology, University of Ulm, Germany*.

See Also

IQ Modulator | Mixer

IQ Modulator

Convert baseband signal to RF signal



Libraries:
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.

The IQ Modulator block mask icons are dynamic and indicate the current state of the applied noise parameter. For more information, see “IQ Modulator Icons” on page 1-78.

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** — Relates the ratio of the power of a single sideband (SSB) of the output to the input power at the I branch. This assumes no gain mismatch and that the input at the Q branch is $Q_{in} = -jI_{in}$
- **Open circuit voltage gain** — Value of the open circuit voltage gain parameter as the linear voltage gain term of the polynomial voltage-controlled voltage source (VCVS).
- **Polynomial coefficients** — Implements a nonlinear voltage gain according to the polynomial you specify.

Available power gain — Ratio of power of SSB at output to input power at I

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

Ratio of the power of SSB at the output to input power at I branch, specified as a scalar in dB or a unitless ratio. For a unitless ratio, select 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. For a unitless ratio, select None.

Dependencies

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

Polynomial coefficients — Coefficients of polynomial specifying voltage gain

[0 1] (default) | vector

Polynomial coefficients, 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, [$a_0, a_1, a_2, \dots, a_9$], the polynomial it represents is:

$$V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + \dots + a_9V_{in}^9$$

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 $V_{out} = a_0 + a_1V_1 + a_2V_1^2 + a_3V_1^3$. Trailing zeros are omitted. So [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 $V_{out} = V_{in}$.

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.

Input impedance (Ohm) — Input impedance of IQ modulator

50 (default) | scalar

Input impedance of IQ modulator, specified as a scalar in Ohms.

Output impedance (Ohm) — Output impedance of IQ modulator

50 (default) | scalar

Output impedance of IQ modulator, specified as a scalar in Ohms.

Add Image Reject filters — Image reject (IR) filter parameters

off (default) | on

Select to add the **IR filter** parameter tab. Clear to remove the tab.

Add Channel Select filter — Channel select (CS) filter parameters

off (default) | on

Select to add the **CS filter** parameter tab. Clear to remove the tab.

Ground and hide negative terminals — Ground and hide terminals

on (default) | off

Select to internally ground and hide the negative terminals. Clear to expose the negative terminals. When the terminals are exposed, you can connect them to other parts of your model.

Edit System — Break IQ modulator block links and replace internal variables by appropriate values
button

Use this button to break IQ modulator links to the library. The internal variables are replaced by their values which are estimated using IQ modulator parameters. The IQ Modulator becomes a simple subsystem masked only to keep the icon.

Use **Edit System** to edit the internal variables without expanding the subsystem. Use **Expand System** to expand the subsystem in the Simulink canvas and to edit the subsystem.

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, or a unitless ratio. 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 the ratio of power of the single-sideband (SSB) at the Q input branch to the output 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. 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, or a unitless ratio. For a unitless ratio, select None.

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

Single-sided noise power spectral distribution, specified as a scalar in dBm/Hz. This block assumes -174dBm/Hz noise input at both I and Q branches.

Add phase noise — Add phase noise
off (default) | on

Select this parameter to add phase noise to your IQ modulator system.

Phase noise frequency offset (Hz) — Phase noise frequency offset
1 (default) | scalar | vector | matrix

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

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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) — Phase noise level

- Inf (default) | scalar | vector | matrix

Phase noise level, specified as a scalar, vector, or matrix with element unit in dBc/Hz.

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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 — Automatically estimate impulse response duration

on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, select **Add phase noise**.

Impulse response duration — Impulse response duration

1e-10 s (default) | scalar

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns.

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. This message also specifies the minimum duration suitable for the required resolution.

Dependencies

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

Plot phase noise characteristics — Phase noise magnitude response button

The block plots the phase noise characteristics based in the parameters specified on the **Impairments** tab and either the **Envelope bandwidth** parameter in the Configuration block when available or the value specified in the **Phase noise frequency offset (Hz)** parameter.

Dependencies

To enable this parameter, select **Add phase noise**.

Nonlinearity

Selecting Polynomial coefficients for **Source of conversion gain** in the **Main** tab removes the **Nonlinearity** parameters.

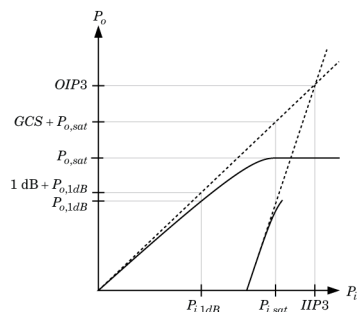
Nonlinear polynomial type — Polynomial nonlinearity

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 IQ 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 **IP2**, **IP3**, the **1-dB gain compression power**, and the **Output 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.

Dependencies

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

1-dB gain compression power — 1-dB gain compression power
inf dBm (default) | scalar

1-dB gain compression power, specified as a scalar in dBm, W, mW, or dBW. The 1-dB gain compression point must be less than the output saturation power.

Dependencies

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

Output saturation power — Output saturation power
inf dBm (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, set **Odd order** in **Nonlinear polynomial type** tab.

Gain compression at saturation — Gain compression at saturation
inf dBm (default) | scalar

Gain compression at saturation, specified as a scalar.

Dependencies

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

IR Filter

Select **Add Image Reject filters** in the **Main** tab to see the **IR Filter** parameters tab.

Design method — Simulation type
Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type
Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**

Implementation — Implementation
LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.

- **Transfer Function:** Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier:** Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Filter Domain:** Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is Constant per carrier for an ideal filter and LC Tee for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Lowpass or Highpass.

Implement using filter order — Implement using filter order

on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order

3 (default) | scalar

Filter order, specified as a scalar. For a **Filter type** of Lowpass or Highpass, the filter order is the number of lumped storage elements. For a **Filter type** of Bandpass or Bandstop, the number of lumped storage elements is twice the filter order.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{\text{load}}}{R_{\text{source}}} > R_{\text{ratio}}$ for Tee network

implementation and $\frac{R_{\text{load}}}{R_{\text{source}}} < \frac{1}{R_{\text{ratio}}}$ for Pi network implementation.

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

where:

- $\varepsilon = \sqrt{10^{(0.1R_p)} - 1}$
 - R_p is the passband ripple in dB.
-

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters
scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. The default value is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters
[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation
 $10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation
40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance
50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance
50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to manually specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns. 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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

To enable this parameter, clear **Automatically estimate impulse response duration**.

Export — Save filter design to a file
button

Use this button to save filter design to a file. Valid file types are .mat and .txt.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

CS Filter

Select **Add Channel Select filter** in the **Main** tab to see the **CS Filter** parameters.

Design method — Simulation type
Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type
Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**.

Implementation — Implementation
LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Filter Domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is Constant per carrier for an ideal filter and LC Tee for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to Ideal.

Implement using filter order — Implement using filter order

on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order

3 (default) | scalar

Filter order, specified as a scalar. This order is the number of lumped storage elements in lowpass or highpass. In bandpass or bandstop, the number of lumped storage elements are twice the value.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{\text{load}}}{R_{\text{source}}} > R_{\text{ratio}}$ for Tee network implementation and $\frac{R_{\text{load}}}{R_{\text{source}}} < \frac{1}{R_{\text{ratio}}}$ for Pi network implementation.

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

where:

- $\varepsilon = \sqrt{10^{(0.1R_p)} - 1}$
 - R_p is the passband ripple in dB.
-

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters

scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. By default, the passband frequency is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters

[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation

$10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation

40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance
50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance
50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns. 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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

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

Export — Save filter design to a file
button

Use this button to save filter design to a file. Valid file types are .mat and .txt.





Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Algorithms

IQ Modulator Icons

This table shows you how the icons on this block will vary based on how you set the **Noise figure (dB)** and **Add LO phase noise** parameters on the block.

Noise figure (dB)	Add LO phase noise: off	Add LO phase noise: on
0		
10		

Version History

Introduced in R2017a

R2021b: IQ Modulator block icon updated

Behavior changed in R2021b

Starting in R2021b, the IQ Modulator block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a IQ Modulator block, the software replaces the block icon with the R2021b version.

R2022b: Estimate impulse response duration using phase noise offset frequencies

The IQ Modulator block estimates the impulse response duration using the **Phase noise frequency offset (Hz)** parameter and visualizes the phase noise characteristics when you click the **Plot phase noise characteristics** button.

References

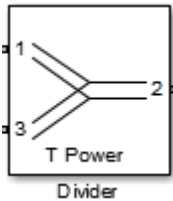
- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and Lindner, Jurgen, "Polynomial Model Derivation of Nonlinear Amplifiers", *Department of Information Technology, University of Ulm, Germany*.

See Also

Modulator | IQ Demodulator | Mixer

Divider

Model ideal frequency-independent dividers or combiners with S-parameters



Libraries:
RF Blockset / Circuit Envelope / Junctions

Description

The Divider block models power dividers or combiners in a circuit envelope environment as an S-parameter model. In a power divider, the input power enters at a port, splits, and exits at different ports.

To use a power divider to model and simulate a 66 GHz QPSK RF transmit and receive system with a 32-element hybrid beamforming antenna, see the Transmit Subarrays section in the “Modeling RF mmWave Transmitter with Hybrid Beamforming” example.

Parameters

Select component — Divider type

T power divider (default) | Resistive power divider | Wilkinson power divider | Tee H-plane (S33=0) | Tee E-plane

Divider type, specified as:

- T power divider

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

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

where:

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

- $z_{13} = z_1 * z_3 / (z_1 + z_3)$
- $z_{23} = z_2 * z_3 / (z_2 + z_3)$
- $z_1 = Z0(1), z_2 = Z0(2), z_3 = Z0(3)$
- Reference Impedances: $Z0 = [z_1, z_2, z_3]$
- Resistive power divider

The S-parameter matrix for Resistive power divider is :

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

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

- Wilkinson power divider

The S-parameter matrix for Wilkinson power divider is:

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

Note

- For DC carrier (0 Hz) frequency, Wilkinson power divider is a zero matrix.
 - Starting R2021a, port positions in the Divider block when you select the Wilkinson power divider component have changed. For more information, see “Compatibility Considerations” on page 1-81.
-

- Tee H-plane (S33=0)

The Tee H-plane (S33=0) is symmetrical, lossless, and reciprocal. The S-parameter matrix is:

$$\frac{\begin{pmatrix} -1 & 1 & \sqrt{2} \\ 1 & -1 & \sqrt{2} \\ \sqrt{2} & \sqrt{2} & 0 \end{pmatrix}}{2}$$

- Tee E-plane

The Tee E-plane is symmetrical, lossless, and reciprocal. The S-parameter matrix is:

$$\frac{\begin{pmatrix} 1 & 1 & \sqrt{2} \\ 1 & 1 & -\sqrt{2} \\ \sqrt{2} & -\sqrt{2} & 0 \end{pmatrix}}{2}$$

Number of divider outputs — Number of Wilkinson power divider outputs
 2 (default) | nonnegative integer in the range (1, 65]

Number of Wilkinson power divider outputs, specified as a nonnegative integer in the range (1, 65].

Dependencies

To enable this parameter, set **Select Component** to `Wilkinson power divider`.

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.

Version History

Introduced in R2014a

R2022b: Model Wilkinson power divider with up to 65 ports

Use the Divider block to model a Wilkinson power divider with more than two and up to 65 outputs. Use the **Number of divider outputs** to set the number of outputs.

R2021a: Port positions of Wilkinson power divider component in Divider block changed

Behavior changed in R2021a

The port positions in the Divider block when you select the `Wilkinson power divider` component have changed. Prior to this release, Ports 1 and 3 were on the same side opposite that of Port 2. This release, Ports 2 and 3 appear on one side together opposite the side of Port 1.

Existing models using the Divider block with the `Wilkinson power divider` component might need to be updated to ensure that there are no broken connections. Use the RF Blockset check in the Simulink® Upgrade Advisor to identify and rewire the broken port connections in your existing models. For more information, see “Check model for RF Blockset Divider blocks using Wilkinson power divider component with broken connections”.

See Also

Circulator | Coupler

Ground

Simulate connection to electrical ground



Libraries:
RF Blockset / Circuit Envelope / Elements

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.

Version History

Introduced in R2016b

R2021b: Ground block icon updated

Behavior changed in R2021b

Starting in R2021b, the Ground block icon has updated. When you open a model created before R2021b containing a Ground block, the software replaces the block icon with the R2021b version.

Ideal Transformer

Model ideal transformer

**Libraries:**

RF Blockset / Circuit Envelope / 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, “Simulate High Frequency Components”.

Parameters

Winding ratio n — Winding ratio of ideal transformer

1 (default) | positive scalar

Winding ratio of the ideal transformer, specified as a positive scalar.

Version History

Introduced in R2013a

See Also

Mutual Inductor | Three-Winding Transformer

Inport

Convert Simulink input signal to RF Blockset signal



Libraries:
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, “Simulate High Frequency Components”. You can also modulate an input signal onto a square carrier wave in the RF Blockset circuit envelope simulation environment.

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-valued input signals into an RF signal suitable for multicarrier simulation:

$$out = Real\left(\sum_{k=1}^N (I_k(t) + j \cdot Q_k(t)) \cdot e^{j2\pi f_k t}\right)$$

The **Source type** parameter specifies the Simulink signal as either current, or voltage, or power source. For more information, see “How to Use Inport Block” on page 1-88. Inport block mask icons are dynamic and show the current type of source. This table shows you how the icons on this block vary based on the type of source you set on the **Source type** parameter on the block.

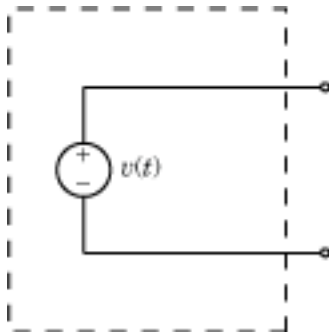
Source type: Ideal voltage	Source type: Ideal current	Source type: Power

Parameters

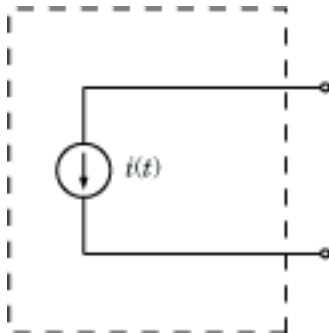
Source type — Inport block interpretation of Simulink signal
Ideal voltage (default) | Ideal current | Power

Inport block interpretation of a Simulink signal, specified as:

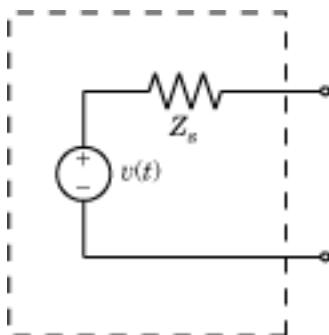
- **Ideal voltage** — The block outputs Simulink signals as voltage signals $v(t)$ in the RF Blockset environment. When you choose an ideal voltage input port you need to manually add a series of source impedance to match the blocks connected to the input port. The following figure illustrates the internal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



- **Ideal current** — 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** — 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 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 $v_{SL}(t)$:

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

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 :

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

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

Source impedance for available power match, 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 | scalar

Carrier frequencies, specified as a vector or scalar in Hz, kHz, MHz, or GHz. Specify **Carrier frequencies** as a positive scalar greater than 0 when you select the **Use Square Wave** parameter. In carrier frequencies, the elements are a combination of fundamental tones and corresponding harmonics in the Configuration block.

Use Square Wave — Use square carrier wave
off (default) | on

Select this parameter to modulate your input signal onto a square carrier wave.

View — Plot square wave
button

Plot the simulation frequencies of the square carrier wave and the approximate square wave with RSM error.

Dependencies

To enable this parameter, select **Use Square Wave**.

Number of Fourier Coefficients — Number of Fourier coefficients
10 (default) | integer greater than or equal to 1

Number of Fourier coefficients required to modulate an input signal onto a square carrier wave, specified as an integer greater than or equal to 1.

Dependencies

To enable this parameter, select **Use Square Wave**.

DC Bias — DC bias
0 (default) | real number

DC bias, specified as a real number.

Dependencies

To enable this parameter, select **Use Square Wave**.

Duty Cycle (%) — Duty cycle of square carrier wave

50 (default) | positive integer between 0 and 100

Duty cycle of a square carrier wave, specified as a positive integer between 0 and 100 in %.

Dependencies

To enable this parameter, select **Use Square Wave**.

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.

More About

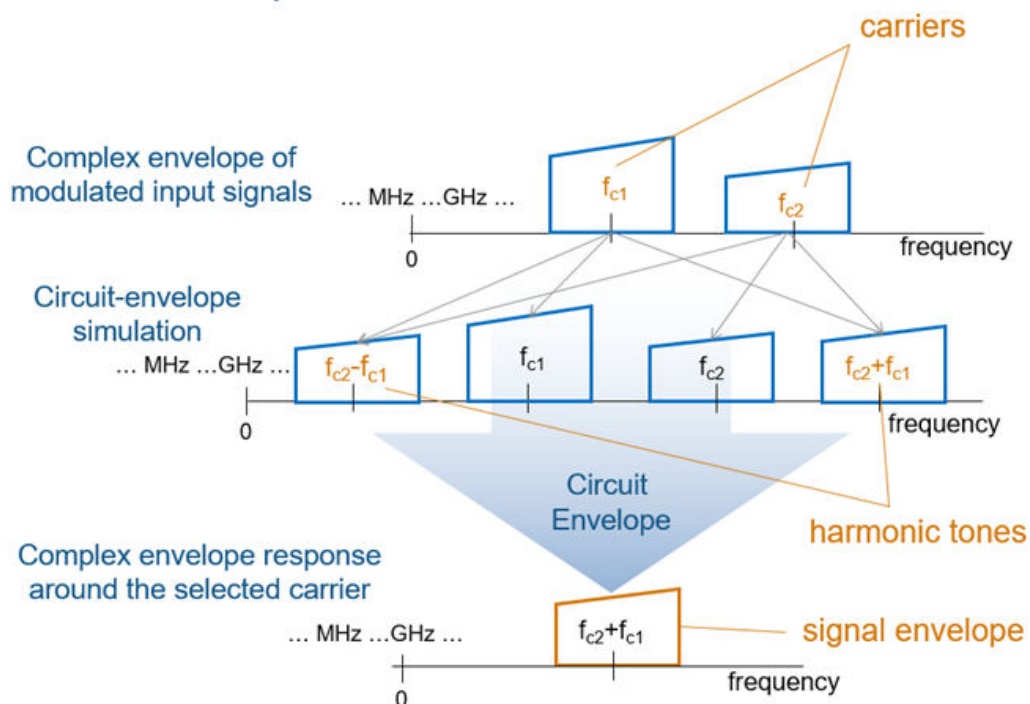
Multi-Carrier Envelope Simulation

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.

Multi-Carrier Envelope Simulation



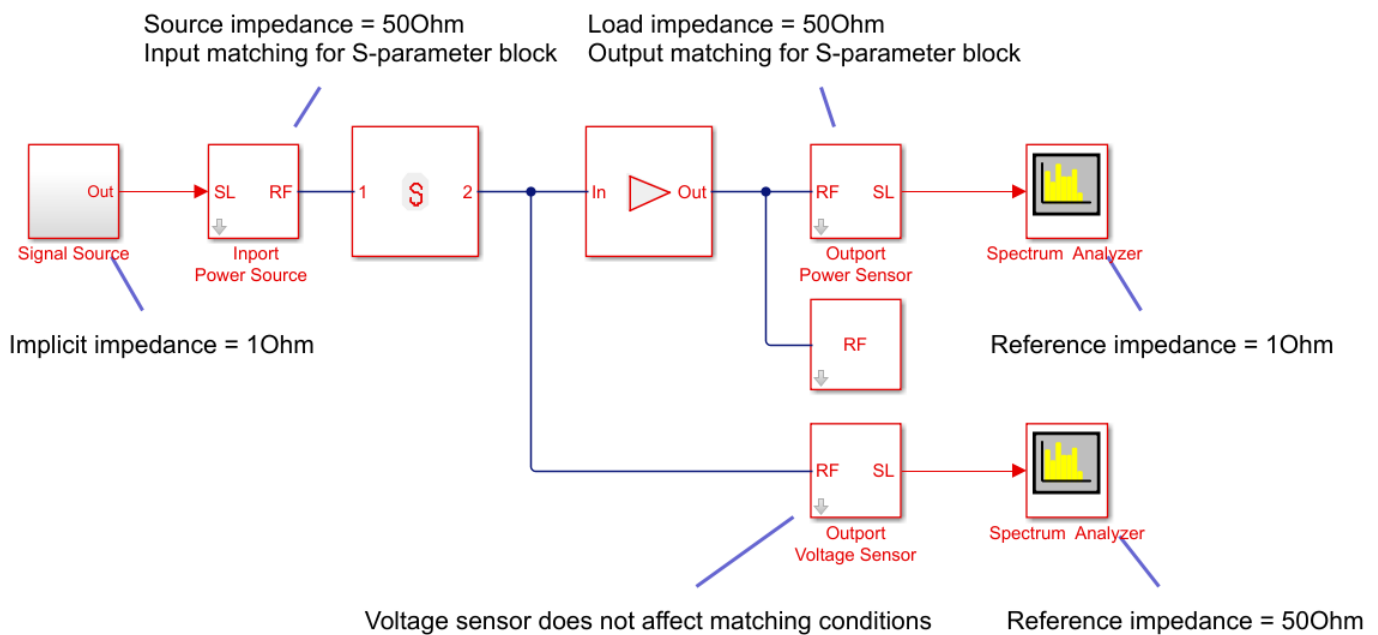
Algorithms

How to Use Inport Block

Physical Interpretation of Simulink Signals

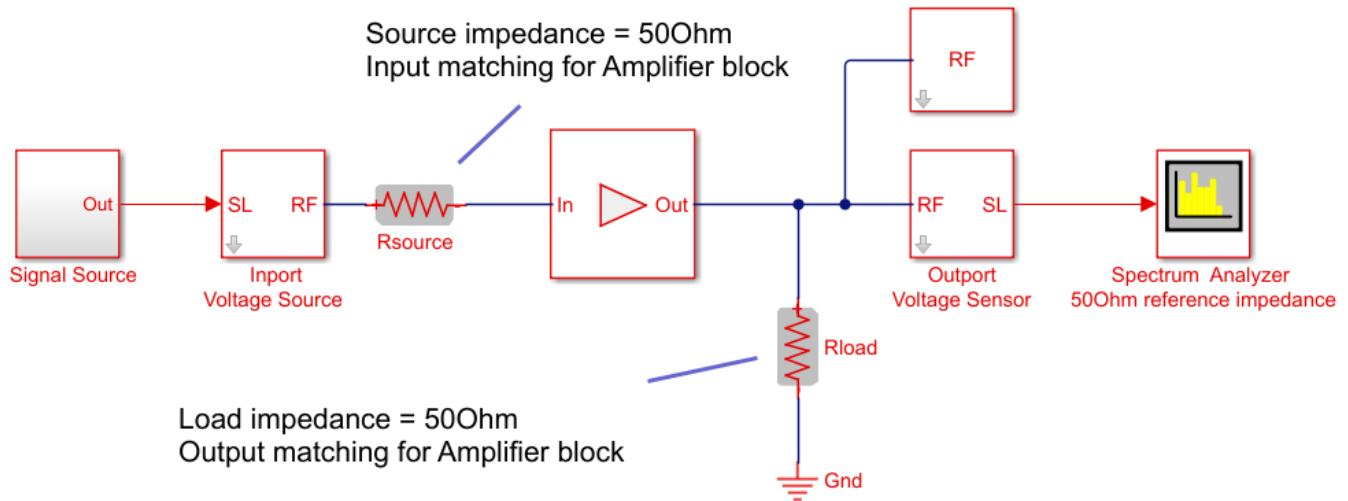
The Inport block allows you to specify the complex envelopes of your input signals and import them as RF signals for multi-carrier simulation.

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.

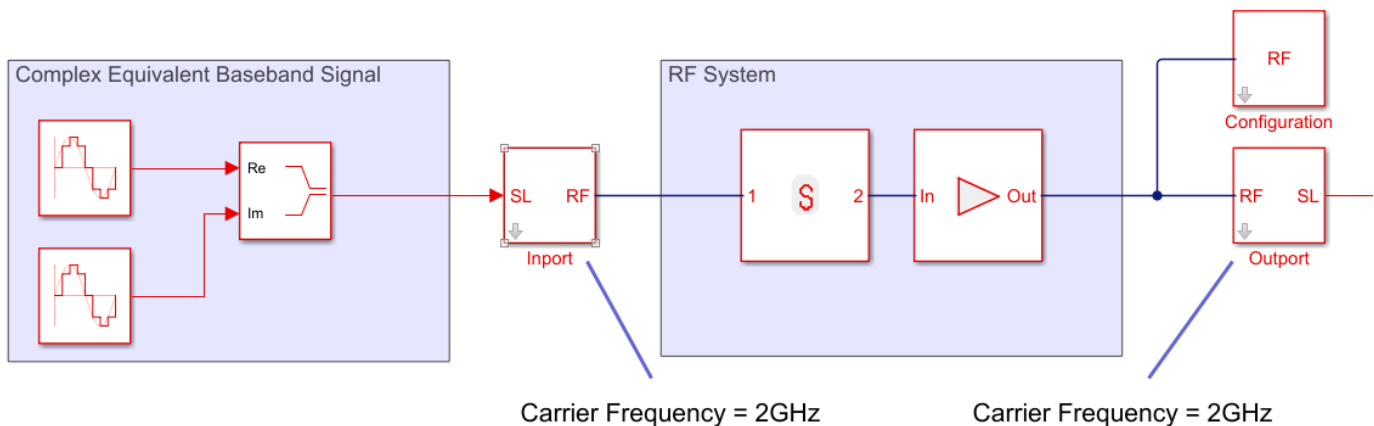
If you use an ideal voltage source and add a source impedance, in perfectly matched conditions, the actual voltage applied to the first block of the RF chain is half of the value of the input Simulink signal. The source impedance and the input impedance of the first block of the RF chain form a voltage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



Implicit Carrier for Complex Equivalent Baseband Signal

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 .

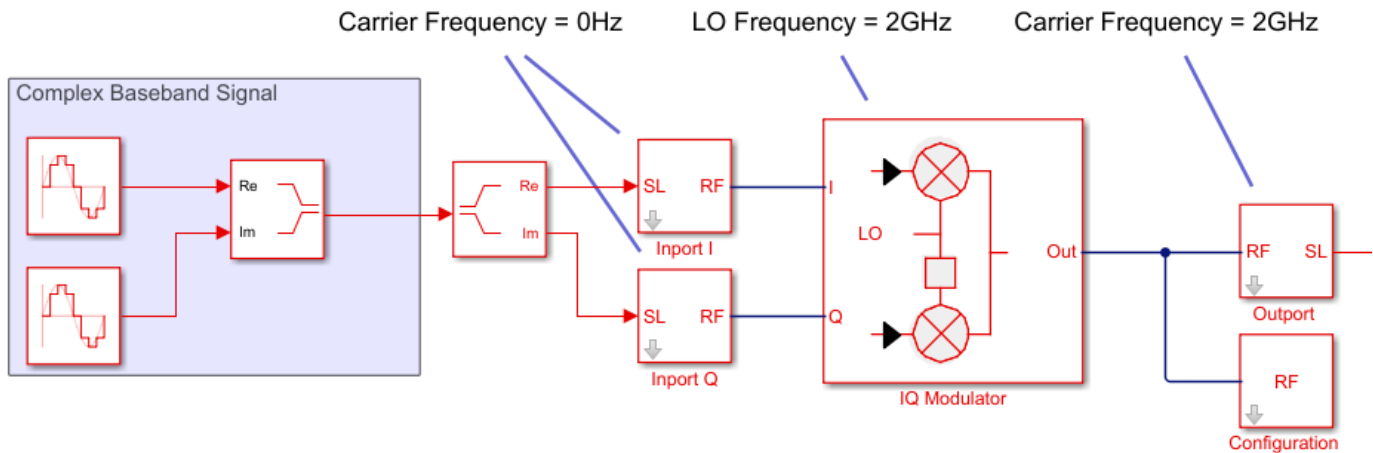


No Carrier for Baseband Signal

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 two **Inport** blocks for the I and Q components of the input signal. Set the **Carrier Frequencies** parameter of each **Inport** block to 0

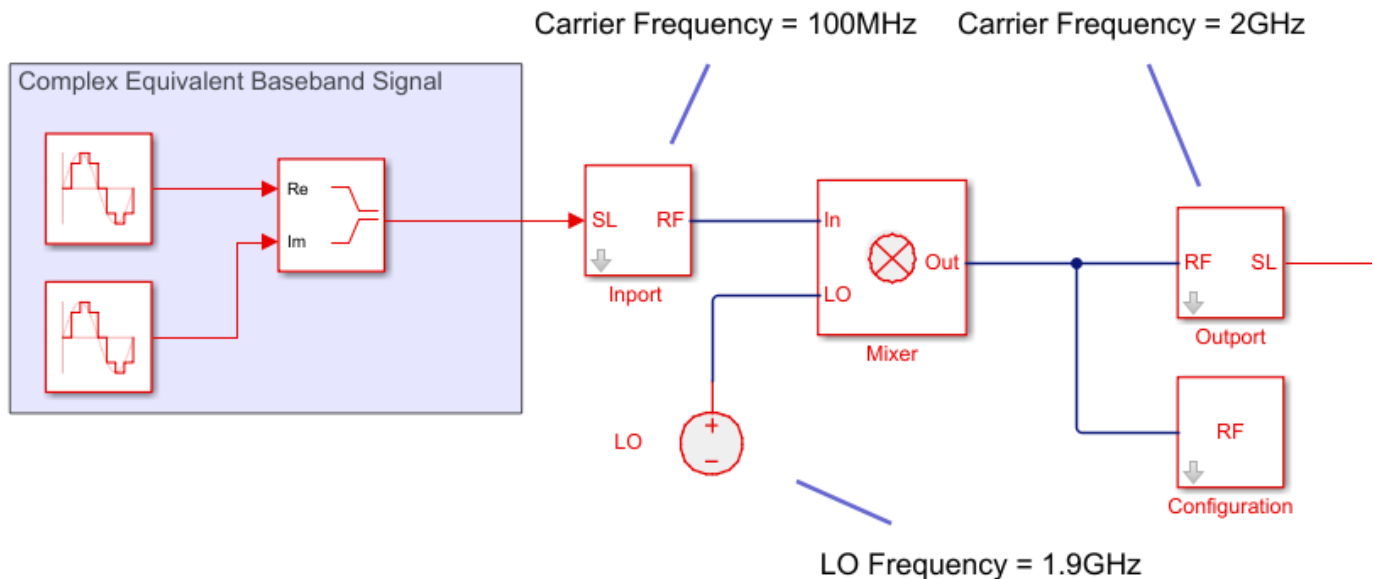
- To upconvert the signal, use an IQ Modulator block. Set the **Local oscillator frequency** to F_c .
- The simulation step size in the configuration block is the same as the sample time of the Simulink input signal, and it is not related to the Local Oscillator frequency.
- Use an Output block at the end of the chain. and probe the signal at F_c .



Upconvert Signal to IF and Then RF

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:

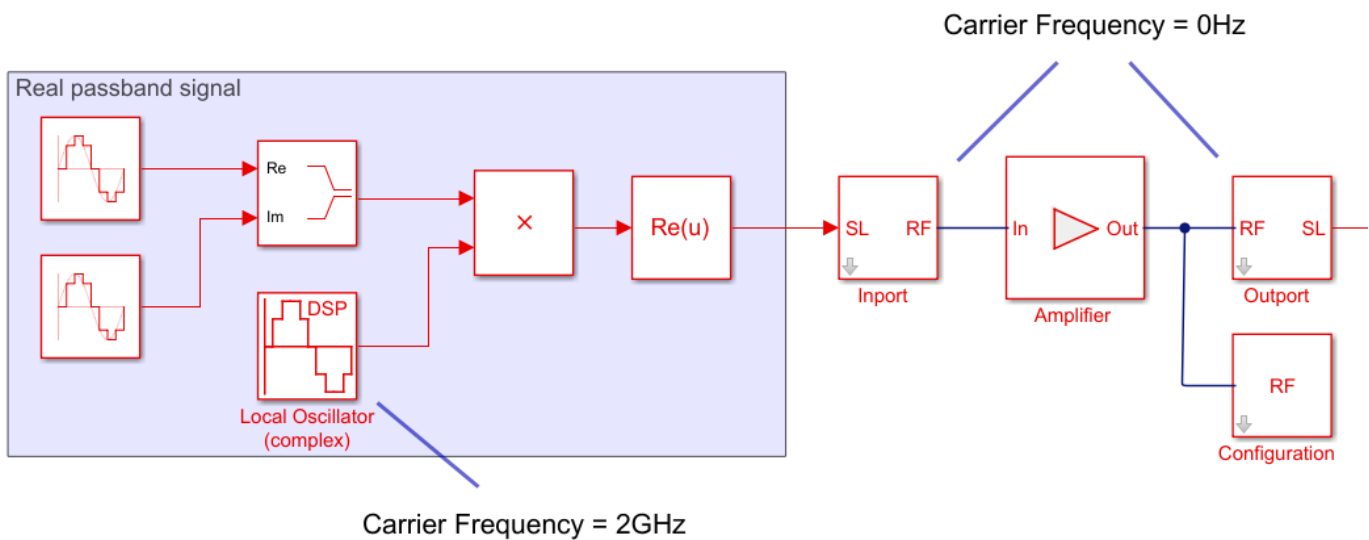
- Set the **Carrier Frequencies** parameter of each Inport block to IF.
- Use the Mixer block. Set the LO carrier frequency to $RF - IF$.
- Use an Output block at the end of the chain. and probe the signal at RF.



Real Passband Signal

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



Version History

Introduced in R2010b

R2021b: Inport block icon updated

Behavior changed in R2021b

Starting in R2021b, the Inport block icon is now updated. When you open a model created before R2021b containing a Inport block, the software replaces the block icon with the R2021b version.

R2022b: Modulate input signal onto square carrier wave in circuit envelope simulation environment

Select the new **Use Square Wave** parameter to modulate an input signal onto a square carrier wave in the circuit envelope simulation environment.

See Also

Output | Configuration

Inductor

Model inductor for circuit envelope analysis



Libraries:
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.

Version History

R2021b: Inductor block icon updated

Behavior changed in R2021b

Starting in R2021b, the Inductor block icon has updated. When you open a model created before R2021b containing an Inductor block, the software replaces the block icon with the R2021b version.

See Also

Resistor | Capacitor

Mixer

Model mixer in RF systems



Libraries:
RF Blockset / Circuit Envelope / Elements

Description

The Mixer block performs signal frequency translation and nonlinear amplification.

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

$$V_{in}V_{LO} = A_{in}\cos(\omega_{in}t)A_{LO}\cos(\omega_{LO}t)$$

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

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

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

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

Mixer block mask icons are dynamic and indicate the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Noise figure (dB)** parameter on the block.

Noise figure (dB): 10	Noise figure (dB): 0

Parameters

Main

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

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

- **Available power gain** — The block uses the value of the Available power gain parameter to calculate the linear voltage gain term of the polynomial VCVS, a_1 . This calculation assumes a matched load termination for the mixer.
- **Open circuit voltage gain** — The block uses the value of the Open circuit voltage gain parameter as the linear voltage gain term of the polynomial VCVS, a_1 .
- **Polynomial coefficients** — The block implements a nonlinear voltage gain according to the polynomial you specify. The order of the polynomial must be less than or equal to 9 and the coefficients are ordered in ascending powers. If a vector a has 10 coefficients, $[a_0, a_1, a_2, \dots, a_9]$, the polynomial it represents is $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \dots + a_9 V_{in}^9$. 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 $V_{out} = a_0 + a_1 V_{in} + a_2 V_{in}^2 + \dots + a_3 V_{in}^3$.

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

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

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

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 IEEE noise figure of mixer
0 dB (default) | scalar

Single-sideband noise figure of mixer, specified as a scalar according to the IEEE definition.

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 IEEE SSB definition assumes that the noise in the image bandwidth at the input of the mixer is perfectly rejected, while the mixer internally generates noise in both the image bandwidth and the signal bandwidth. As a result, the noise at the output of the mixer is the sum of two contributions:

$$N_{\text{out}} = N_{\text{in}}G_{\text{mix}} + 2N_{\text{mixer}}G_{\text{mix}}$$

where:

- N_{out} is the noise at the output of the mixer.
- N_{in} is the noise at the input of the mixer (assuming that the noise in the image bandwidth is perfectly rejected).
- N_{mixer} is the noise internally generated by the mixer in both the signal and the image bandwidths.
- G_{mix} is the mixer gain.

As a result, the noise factor according to the IEEE SSB definition can be expressed as

$$F_{\text{SSB-IEEE}} = 1 + 2N_{\text{mixer}}/N_{\text{in}}$$

which is related to other commonly used definitions through

$$F_{\text{SSB}} = 2 + 2N_{\text{mixer}}/N_{\text{in}} = 1 + F_{\text{SSB-IEEE}},$$

$$F_{\text{DSB}} = 1 + N_{\text{mixer}}/N_{\text{in}} = \frac{1}{2}(1 + F_{\text{SSB-IEEE}}).$$

You can apply the IEEE SSB definition directly to describe mixer stages when using the Friis formulas for link budget analysis. Using the other definitions requires changing the Friis formulas. Both the Mixer block in RF Blockset and the **RF Budget Analyzer** app in RF Toolbox™ use the IEEE definition.

The analytic relationships between the three definitions allow you to simulate the noise level at the output of the mixer. For example,

$$F_{\text{SSB}} = 4 \text{ dB} \Rightarrow F_{\text{SSB-IEEE}} = 10 \log_{10}(10^{F_{\text{SSB}}/10} - 1) = 1.79 \text{ dB},$$
$$F_{\text{DSB}} = 4 \text{ dB} \Rightarrow F_{\text{SSB-IEEE}} = 10 \log_{10}(2 \times 10^{F_{\text{DSB}}/10} - 1) = 6.04 \text{ dB}.$$

If you simulate an RF Blockset mixer without including an ideal image rejection filter, then the noise at the output of the mixer is larger than that predicted by the noise figure, because the noise in the image bandwidth is effectively folded onto the output signal.

For this reason, when generating models, both the Modulator and Demodulator blocks insert an ideal image-rejection filter automatically. (You can remove the filtering within the block mask.)

The Noise Figure Testbench block measures the SSB noise figure and enables you to verify that the simulated noise figure has the expected value.

- If you add an ideal image rejection filter to your model, then the effective noise figure is consistent with the analytic value.
- If you remove the image-rejection filter, or if you use a filter with partial rejection, then the measured noise figure is larger than the analytic value.

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

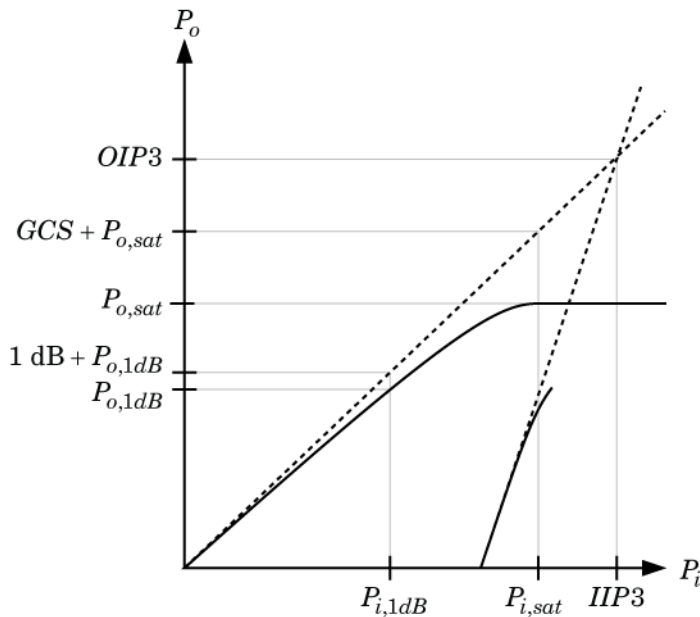
Nonlinear polynomial type — Polynomial nonlinearity

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 mixer 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
inf dBm (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
inf dBm (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
inf dBm (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** .

Version History

Introduced in R2010b**R2021b: Mixer block icon updated**

Behavior changed in R2021b

Starting in R2021b, the Mixer block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Mixer block, the software replaces the block icon with the R2021b version.

References

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

See Also

Amplifier | S-Parameters

Noise

Model noise using current or voltage noise source in RF systems




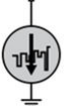
Libraries:

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.

Noise block mask icons are dynamic and show the current type of source. This table shows you how the icons on this block vary based on the type of source you set on the **Source type** parameter on the block.

Source type: Ideal voltage	Source type: Ideal current
	

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.

Version History

Introduced in R2010b

R2021b: Noise block icon updated

Behavior changed in R2021b

Starting in R2021b, the Noise block icon is now updated. When you open a model created before R2021b containing a Noise block, the software replaces the block icon with the R2021b version.

See Also

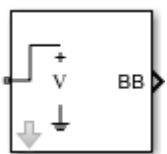
Continuous Wave | Sinusoid

Topics

“Model System Noise Figure”

Outport

Convert RF Blockset signal to Simulink output signals



Libraries:
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, “Simulate High Frequency Components”.

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 centered around each specified center frequency f_k . The **Sensor type** parameter determines which signal the block measures, and the **Output** parameter defines the format of the Simulink signal. For more information, see “How to Use Outport Block” on page 1-106.

Outport block icons are dynamic and show the current format of the output signal. For more information, see “Outport Block Icons” on page 1-105.

Note Outport block Real passband output does not support frame-based processing (supported by the Configuration blocks). This block errors if the **Samples per frame** is more than 1 in the Configuration block.

Parameters

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

Type of signal measured by sensor, specified as:

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

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

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 Outputport blocks as power sources at the same node in a given circuit, the resulting load is the parallel combination of the specified load impedances.

Dependencies

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

Output — Format of output signals

Complex Baseband (default) | In-phase and Quadrature Baseband | Magnitude and Angle Baseband | Real Passband

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

- **Complex Baseband** — The block outputs a vector of complex-valued signals $I_k(t) + j \cdot Q_k(t)$ at the port labeled **SL**. The k th element of the vector is the k th frequency specified by the **Carrier frequencies** parameter.
- **In-phase and Quadrature Baseband** — The block outputs two vectors of real-valued signals $I_k(t)$ and $Q_k(t)$ at the **I** port and **Q** port, respectively. The signal at the **I** port contains the in-phase components, and the signal at the **Q** port contains the quadrature components. The 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.
- **Magnitude and Angle Baseband** — The block outputs two real-valued vectors, whose elements are the magnitude and phase angle of the modulation. The **Mag** port outputs $|I_k(t) + j \cdot Q_k(t)|$ and the **Ang** port outputs $\text{Arg}[I_k(t) + j \cdot Q_k(t)]$. The k th element of the vector is the k th 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.

$$out = \text{Real} \left(\sum_{k=1}^N (I_k(t) + j \cdot Q_k(t)) \cdot e^{j2\pi f_k t} \right)$$

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

Determine optimal time step to resolve highest listed carrier frequency, specified as a on or 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.

Interpolation Filter Half Polyphase Length — Half-polyphase length of filter

12 (default) | nonnegative integer

Half-polyphase length of filter, specified as a nonnegative integer. Set the half-polyphase length to a value greater than zero to mitigate undesired spectral images on a real passband output. If you set this parameter to 0, then the block does not filter.

Dependencies

To enable this parameter, set **Output** to **Real Passband**.

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

Ground RF circuit terminals, specified as a **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.

More About

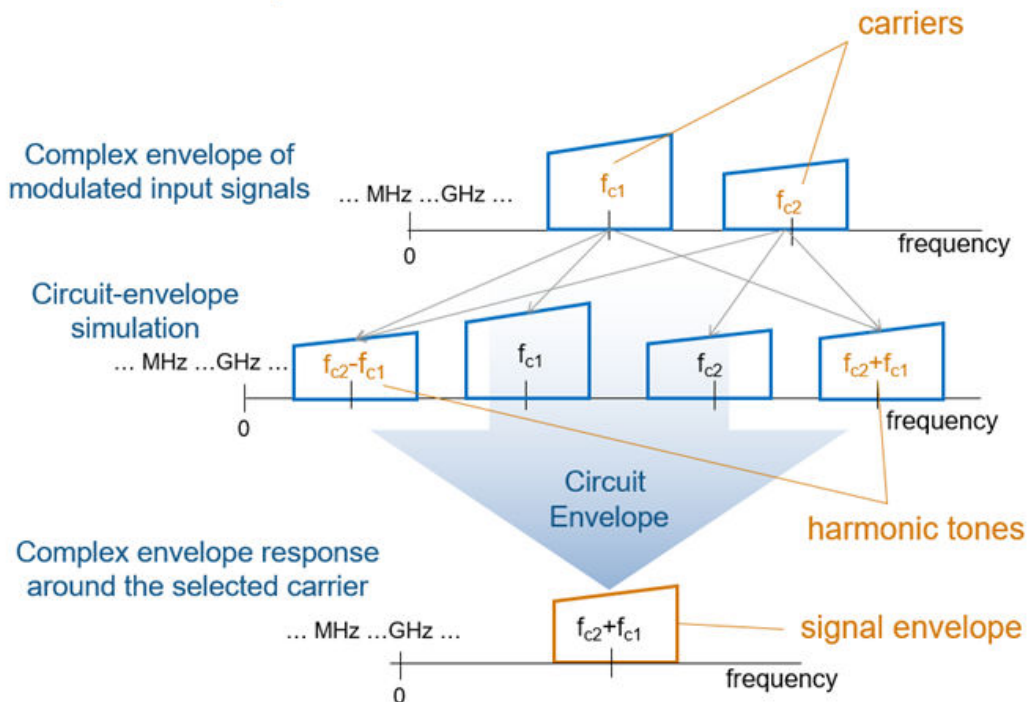
Multi-Carrier Envelope Simulation

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.

Multi-Carrier Envelope Simulation



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.

Output Block Icons

Outport block icons are dynamic and show the current format of the output signal. This table shows how the block icons vary based on how you set the **Sensor type**, **Ground and hide negative terminals**, and **Output** parameters.

Sensor type	Ground and hide negative terminal	Output: Complex Baseband	Output: In-phase and Quadrature Baseband	Output: Magnitude and Angle Baseband	Output: Real Passband
Ideal voltage	On				
	Off				
Ideal current	On				
	Off				
Power	On				
	Off				

Algorithms

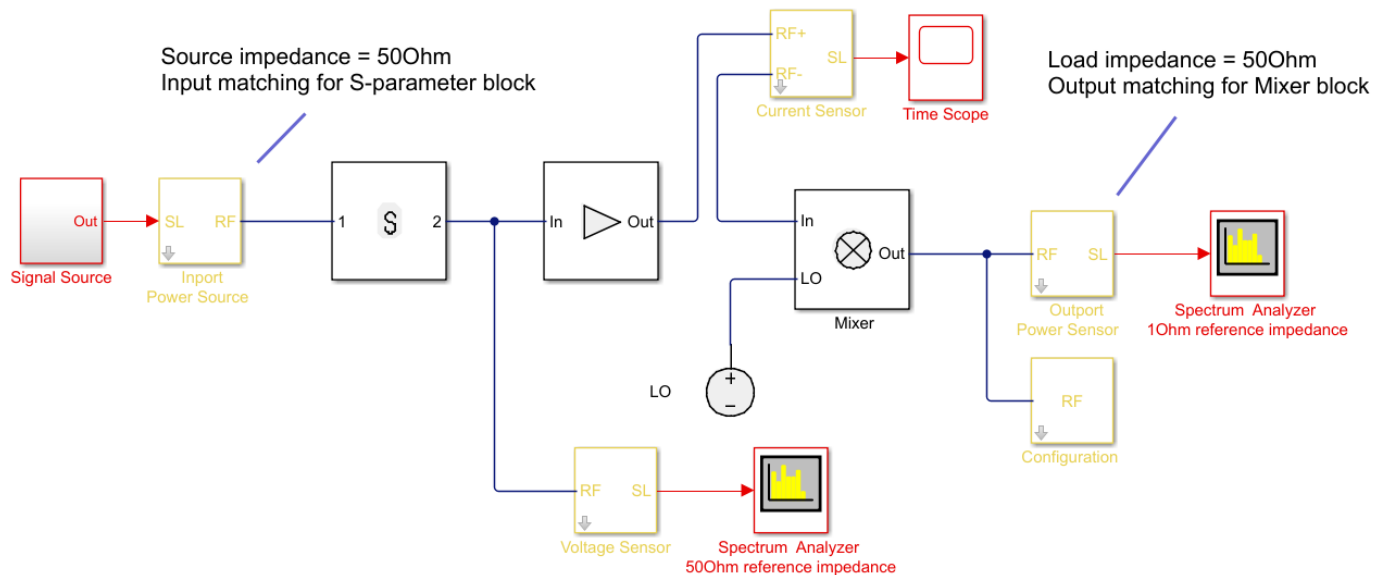
How to Use Output Block

Sensing Signals at Different Nodes of RF Chain

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 1 ohm) and the RF domain.

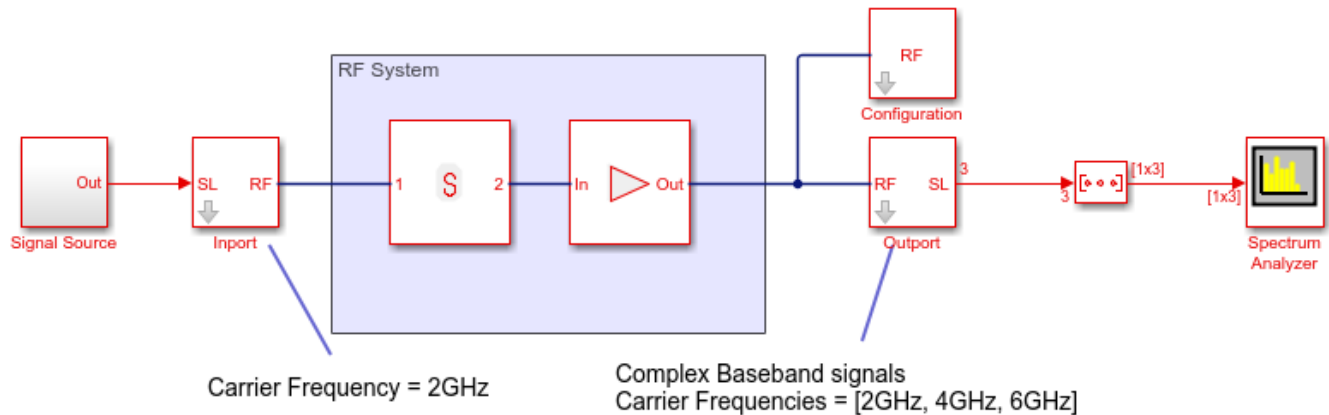
Use the voltage or current output ports connected to the intermediate nodes and branches that you want to inspect. To measure the power of a voltage signal, change the default spectrum analyzer reference impedance to 50 ohm.



Sense Complex Equivalent Baseband Signal for Digital Signal Processing

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

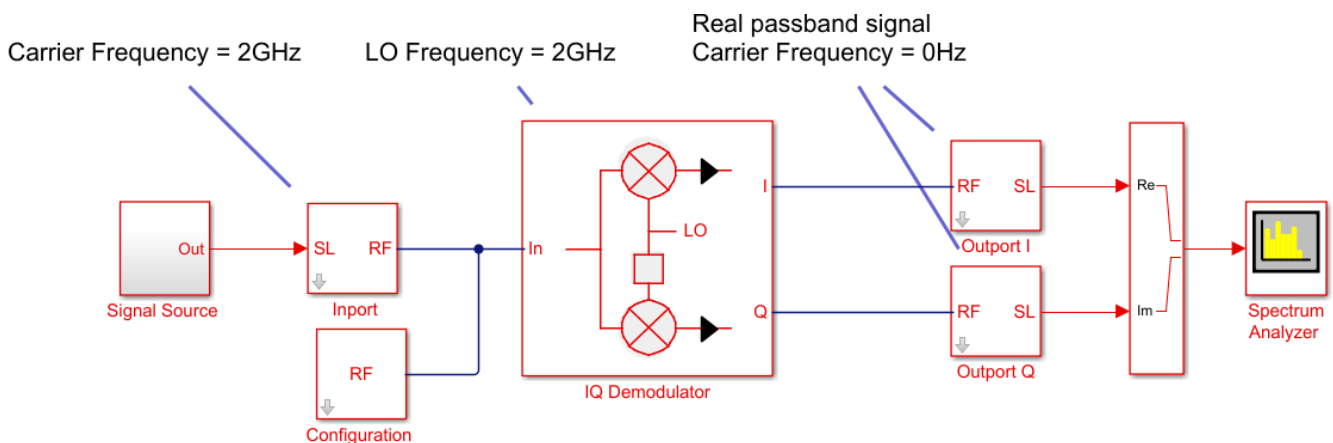
- Enter the array F_c in the **Carrier Frequencies** parameter corresponding to the center frequency of the envelopes that you want to sense. The output signals is an array of complex (I,Q) envelopes.
- The output port of sensor type power provides a termination (by default 50Ohm) at the end of the chain.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



Sense Real Passband Signal

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 center 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.

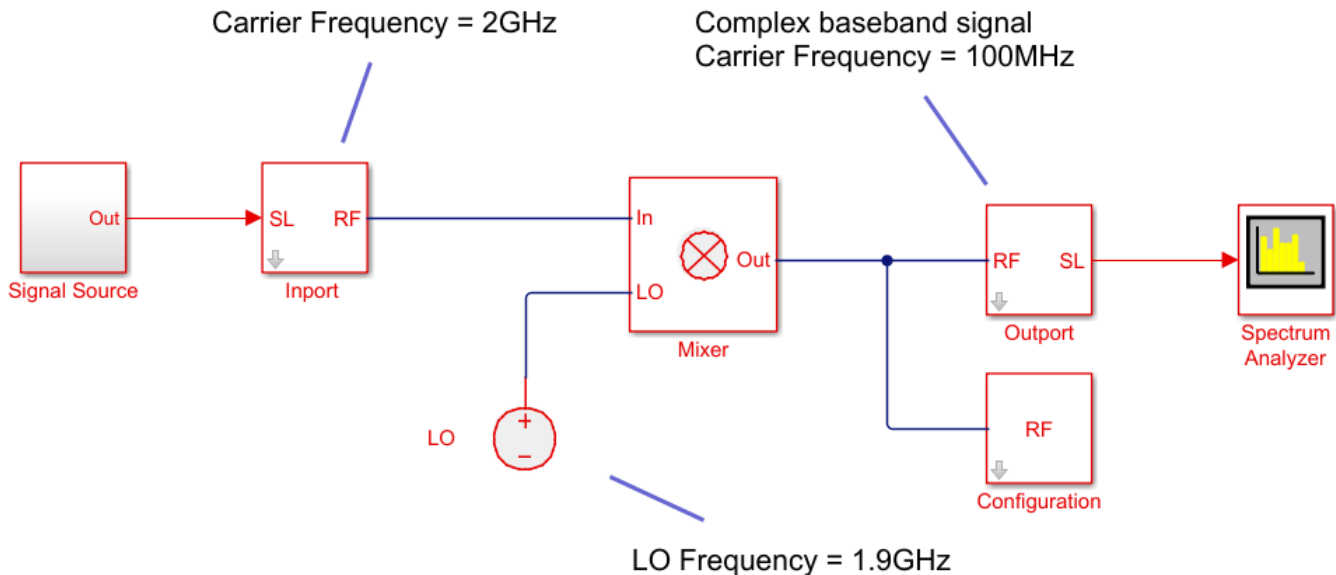


Down Convert Signal to IF and Sense Complex Equivalent Baseband Signal

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.

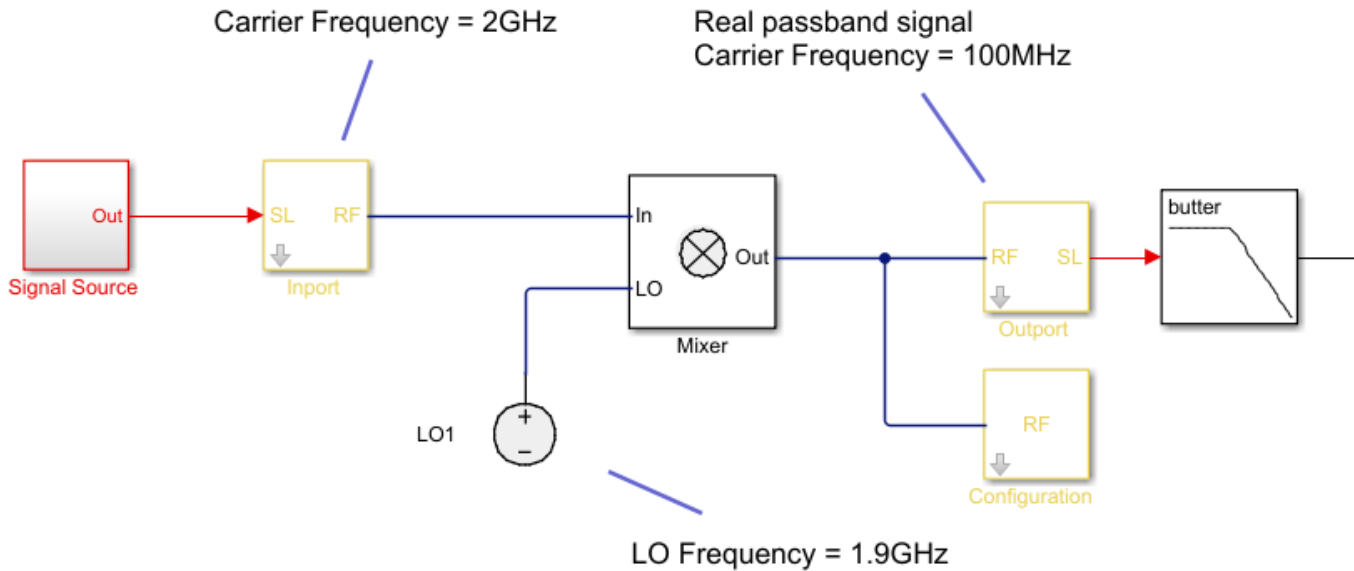
- Use the Mixer block for down-conversion. Set the LO carrier frequency to $L0 = RF - IF$. The output port will behave as an ideal filter and select only the down-converted signal. The envelope at $RF+L0 = 2RF - IF$ frequency is simulated but it is not sensed by the output port.
- The simulation step 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.



Down Convert Signal to IF and Sense Real Passband Signal

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 $L0 = RF - IF$. The output port will behave as an ideal filter, and select only the down-converted signal. The envelope at $RF+L0 = 2RF - IF$ 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.



Version History

Introduced in R2010b

R2021b: Outport block icon updated

Behavior changed in R2021b

The Outport block icon was updated in R2021b. When you open a model created before R2021b containing an Outport block, the software replaces the block icon with the R2021b version.

R2022b: Outport block icon updated

Behavior changed in R2022b

The Outport block icon now changes based on how you set the **Sensor type**, **Ground and hide negative terminals**, and **Output** parameters. When you open a model created before R2022b containing an Outport block, the software replaces the block icon with the R2022b version.

R2023a: FIR interpolation filter for real passband output in Outport block

The Outport block now supports an FIR interpolation filter for real passband output to mitigate undesired spectral images.

See Also

Inport | Configuration

Resistor

Model resistor for circuit envelope analysis



Libraries:
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, “Simulate High Frequency Components”.

The block implements the relation

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

where:

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

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:

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

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

Resistor block mask icons are dynamic and indicate the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Simulate noise** parameter on the block.

Simulate noise: on	Simulate noise: off

To learn how you can use the Resistor block to design output matching network, open *RF Receiver Front End* subsystem and select *Linear → Output Matching* in “Radar System Modeling” example.

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 $4 kT/R$ A²/Hz, where:

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

Version History

R2021b: Resistor block icon updated

Behavior changed in R2021b

Starting in R2021b, the Resistor block icon is now updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Resistor block, the software replaces the block icon with the R2021b version.

See Also

Capacitor | Inductor

S-Parameters

Model S-parameter network



Libraries:
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 65 ports. For an introduction to RF simulation, see the example, “Simulate High Frequency Components”.

Note If you specify more than eight ports, the block does not simulate noise.

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 `rational` function.

Note During the fitting procedure, the block sets the `TendsToZero` and `Qlimit` properties of the rational object to `false` and `inf`, respectively.

S-parameters block mask icons are dynamic and indicate the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Simulate noise** parameter on the block.

Simulate noise: on	Simulate noise: off

To model RF complex baseband S-parameters in Simulink, use the Idealized Baseband S-parameters block.

Parameters

Main

Data source — Data source

Data file (default) | Network-parameters | Rational model

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

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

- **Network-parameters** — Provide **Network parameter** data such as S-parameters, Y-parameters, and Z-parameters with corresponding **Frequency** and **Reference impedance (ohms)** for the s-parameters.
- **Rational model** — Provide values for **Residues**, **Poles**, and **Direct feedthrough** parameters which correspond to the equation for a rational model

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

.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 object — Network parameter objects as data source
off (default) | on

Select this parameter to use a network parameter object as the data source to model the S-parameters.

Once you select this parameter, in the text box that opens, specify one of these:

- Network parameter object.
- MATLAB base workspace variable or a Simulink model workspace variable. The block evaluates if the specified variable is a network parameter object.

`sparameters('default.s2p')` is the default network parameter object when you select this parameter.

The network parameter objects that you can specify include:

- `sparameters`
- `yparameters`
- `zparameters`
- `gparameters`
- `hparameters`
- `abcdparameters`
- `tparameters`

Dependencies

To enable this parameter, set **Data source** to **Network-parameters**.

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, set **Data source** to Network-parameters.

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

Network parameter values specified as a multidimensional array or a scalar. When you specify **Network parameters** as a multi-dimensional array, the third dimension of the S-parameter array must be of the same length as the frequencies vector specified in the **Frequency (dB)** parameter.

Dependencies

To enable this parameter, set **Data source** to Network-parameters.

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

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

Dependencies

To enable this parameter, set **Data source** to Network-parameters.

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, set **Data source** to Network-parameters.

Use rational object — Rational object as data source
off (default) | on

Select this parameter to use a `rational` object as the data source to model S-parameter.

Once you select this parameter, in the text box that opens, specify one of these:

- `rational` object.
- MATLAB base workspace variable or a Simulink model workspace variable. The block evaluates if the specified variable is a rational object."

`rational('default.s2p')` is the default `rational` object when you select this parameter.

The `rational` objects that you can specify include:

- `rational`
- `rationalfit`
- `rfmodel.rational`

Dependencies

To enable this parameter, set **Data source** to `Rational` model.

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, set **Data source** to `Rational` model.

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, set **Data source** to `Rational` model.

Direct feedthrough — Direct feedthrough

{0 0:1 0} (default) | array of vectors

Direct feedthrough, specified as an array vector.

Dependencies

To enable this parameter, set **Data source** to `Rational` model.

Simulate noise — Generate thermal noise waves

off (default) | on

Choose this parameter to generate thermal noise waves [1]. Clear this parameter to stop simulating noise. For more information see, “Generate Thermal Noise” on page 1-118.

Note This parameter is cleared when you specify more than eight ports.

Ground and hide negative terminals — Ground RF circuit terminals

on (default) | off

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

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 `rational` function.

- **Frequency domain** computes the baseband impulse response for each carrier frequency independently. This technique is based on convolution. There is an option to specify the duration of the impulse response. For more information, see “Compare Time and Frequency Domain Simulation Options for S-parameters”.
- For the Amplifier, Antenna, and S-parameters blocks, the default value is **Time domain (rationalfit)**. For the Transmission Line block, the default value is **Frequency domain**.

Dependencies

To enable this parameter, set **Data source** to either **Data file** or **Network-parameters**.

Fitting options — Rational fit fitting options

`Fit individually (default) | Share poles by column | Share all poles`

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

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

Dependencies

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

Relative error desired (dB) — Relative error acceptable for the rational fit

`-40 (default) | scalar`

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

Dependencies

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

Maximum number of poles — Maximum number of poles

`48 (default) | positive scalar`

Maximum number of poles to control over the fitting procedure, specified as a positive scalar. You can specify up to 99 poles.

Dependencies

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

Automatically estimate impulse response duration — Automatically calculate impulse response

`on | off`

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

Dependencies

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

Impulse response duration — Impulse response duration

`1e-10 (default) | scalar`

Impulse response duration, specified as a scalar.

Dependencies

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

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

Select this parameter to ignore the S-parameter phase and to delay the impulse response by half its length. You can use this parameter to shape spectral content with filter effects by specifying only magnitude.

This parameter uses the magnitude of the original data to obtain the impulse response of the block. This impulse response is centered about the time sequence and adds an inherent delay of half the duration of the impulse response. In the frequency domain, this translates to a linear monotonic decrease in phase throughout the given spectrum.

Note

- This parameter is applicable only for S-parameter data modeled in the frequency domain and introduces an artificial delay to the system.
 - Adding a time delay (or frequency phase variation) to the system can result in unexpected behaviors. For example, in an unmatched system, delayed signals are bounced back and added to the original signals.
-

Dependencies

To enable this parameter, set **Modeling options** to `Frequency` domain.

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] Hz (default) | vector | scalar`

Frequency data range, specified as a vector or a scalar in Hz, kHz, MHz, or GHz. Specify **Frequency data** as a scalar when you want to visualize a single frequency S-parameter data.

Dependencies

To enable this parameter, set **Source of frequency data** to `User-specified`.

Plot type — Type of data plot

X-Y plane (default) | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

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

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

Parameter 1 — Type of S-Parameters to plot

S11 (default) | SNN

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

Parameter 2 — 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 — 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

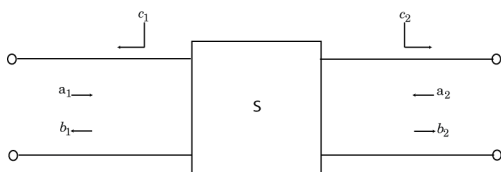
button

Plot specified data using plot button.

More About

Generate Thermal Noise

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



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

$$b = Sa + c$$

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

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

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

where:

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

Version History

Introduced in R2010b

R2021b: S-parameter network data construction in S-parameter block

Behavior changed in R2021b

Starting in R2021b, the Circuit Envelope library S-Parameters block uses the `rational` function to perform rational fitting to complex frequency-dependent data. The `rational` object provides improved fitting speed. The block also has a new parameter, **Maximum number of poles**, to provide additional control over the fitting procedure. You can specify up to 99 poles.

When you open a model created before R2021b containing the Circuit Envelope library S-parameter block, the software updates the extraction method to perform a rational fit with the new **Maximum number of poles** parameter. The output of the model gets refitted and the refitted output might differ from the output in previous releases.

R2021b: S-parameter block icon updated

Behavior changed in R2021b

Starting in R2021b, the S-parameters block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a S-parameters block, the software replaces the block icon with the R2021b version.

R2022b: Use Network parameter and rational objects to model S-parameters

Use network parameter and `rational` objects created at the MATLAB command line as the data source to model S-parameters by setting **Data Source** to `Network parameters` and `Rational model`. For more information, see **Network-parameter object** and **Use rational object**.

R2022b: Visualize single-frequency S-parameter data

Visualize single-frequency S-parameter data by specifying single-frequency S-parameter data in the **Frequency data** parameter.

R2022b: Block parameter renamed

Behavior changed in R2022b

The **Frequency data** parameter in the S-parameter block has been renamed to **Source of frequency data**.

When you open a model created before R2022b containing the S-parameter block, the software replaces the **Frequency data** parameter with the **Source of frequency data** parameter.

References

- [1] Wedge, Scott, and David Rutledge. "Wave Techniques for Noise Modeling and Measurement." IEEE Transactions on Microwave Theory and Techniques 40, no. 11 (November 1992): 2004-12. <https://doi.org/10.1109/22.168757>.

See Also

Transmission Line | Amplifier | Mixer

Sinusoid

Model DC offset and sinusoidal modulation



Libraries:
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:

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

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

where:

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

Note Sinusoid block does not support frame-based processing (supported by the Configuration blocks) as it uses SL Sine wave block. This block errors if the **Samples per frame** is more than 1 in the configuration block.

Sinusoid block mask icons are dynamic and show the current type of source. This table shows you how the icons on this block vary based on the type of source you set on the **Source type** parameter on the block.

Source type: Ideal voltage	Source type: Ideal current

Parameters

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

Wave type, specified as:

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

Offset in-phase — In-phase offset

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

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

Offset quadrature — Quadrature offset

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

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

Sinusoidal amplitude in-phase — In-phase amplitude

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

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

Sinusoidal amplitude quadrature — Quadrature amplitude

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

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

Sinusoidal modulation frequency — Modulation frequency

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

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

Time delay — Time delay of modulation

0 (default) | vector of real | s | ms | us | ns

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

Carrier frequencies — Carrier frequencies

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

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

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

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

By default, this option is selected.

Version History

Introduced in R2010b

R2021b: Sinusoid block icon updated

Behavior changed in R2021b

Starting in R2021b, the Sinusoid block icon is now updated. When you open a model created before R2021b containing a Sinusoid block, the software replaces the block icon with the R2021b version.

See Also

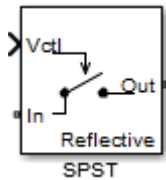
Noise | Continuous Wave

Topics

“Model System Noise Figure”

SPST

Single pole single throw switch



Libraries:

RF Blockset / Circuit Envelope / Junctions

Description

The SPST block models a single pole single throw switch. This block supports both reflective and absorptive switches. The control signal is a Simulink signal connected to the `Vctl` port of the block. The control signal specifies when the output port, `Out`, transmits the input signal at the input port, `In`.

Ports

Input

Vctl — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: `double`

Parameters

Threshold voltage — Threshold voltage
0 (default) | positive scalar

Threshold voltage of the switch, specified as a positive scalar in volts.

Characterization — Characterization
Resistance (default) | Insertion loss

Characterization of the switch, specified as `Resistance` or `Insertion loss`.

On resistance — On resistance
10 (default) | positive scalar

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

Off resistance — Off resistance
1e-6 (default) | positive scalar

Off resistance value of the switch, specified as a positive scalar in ohms. **Off resistance** is available when you set **Characterization** to Resistance.

Insertion loss (dB) — Insertion loss

1 (default) | scalar

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

Isolation (dB) — Isolation value

70 (default) | scalar

Isolation value of the switch, specified as a scalar in decibels. **Isolation (dB)** is available when you set **Characterization** to Insertion loss.

Loading type — Loading type

Reflective (default) | Absorptive

Loading type of the switch, specified as Reflective or Absorptive.

Port terminations (Ohm) — Port terminations

50 (default) | scalar | vector of size 1-by-2

Port terminations that match the impedance conditions of the SPDT switch, specified as a scalar in ohms or a vector of size 1-by-2 with each element unit in ohms. If the value is a scalar, then the input port and the output port impedances are the same. If the value is a vector, the first value is the matching required at the inport of the block and the second values is the matching required at the output port.

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 — Ground RF circuit terminals

on (default) | off

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

More About

Resistance Characterization

- If the control voltage is less than the threshold voltage, then 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, then 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. The shunt resistor values provide matching terminations at the input and output ports.

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.

Voltage-Current Relationship

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

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

$$I_{in} = V_{inout} \cdot G_{off} + V_{in} \cdot G_{Z01}$$

$$I_{out} = -V_{inout} \cdot G_{off} + V_{out} \cdot G_{Z02}$$

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

$$I_{in} = V_{in1}G_{off} + V_{in2}G_{on}$$

$$I_1 = -V_{in1}G_{off} + V_1G_{Z01}$$

$$I_2 = -V_{in2}G_{on}$$

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

Version History

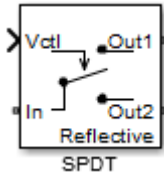
Introduced in R2015b

See Also

Potentiometer | Switch | SPDT

SPDT

Single pole double throw switch



Libraries:

RF Blockset / Circuit Envelope / Junctions

Description

The SPDT block models a single pole double throw switch. This block supports both reflective and absorptive switches. The control signal is a Simulink signal connected to the `Vctl` port of the block. The control signal specifies which output port transmits the input signal, `In`.

- If the control voltage is less than the threshold voltage, then the block transfers the input signal to the first output port, `Out1`.
- If control voltage is greater than or equal to the threshold voltage, then the block transfers the input signal to the second output port, `Out2`.

Two different resistors regulate the conduction path for transferring the input signal from the input port to either output port.

Ports

Input

Vctl — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: double

Parameters

Threshold voltage — Threshold voltage
0 (default) | positive scalar

Threshold voltage of the switch, specified as a positive scalar in volts.

Characterization — Characterization
Resistance (default) | Insertion loss

Characterization of the switch, specified as Resistance or Insertion loss.

On resistance — On resistance
10 (default) | positive scalar

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

Off resistance — Off resistance
1e-6 (default) | positive scalar

Off resistance value of the switch, specified as a positive scalar in ohms. **Off resistance** is available when you set **Characterization** to Resistance.

Insertion loss (dB) — Insertion loss
1 (default) | scalar

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

Isolation (dB) — Isolation value
70 (default) | scalar

Isolation value of the switch, specified as a scalar in decibels. **Isolation (dB)** is available when you set **Characterization** to Insertion loss.

Loading type — Loading type
Reflective (default) | Absorptive

Loading type of the switch, specified as Reflective or Absorptive.

Port terminations (Ohm) — Port terminations
50 (default) | scalar | vector of size 1-by-3

Port terminations that match the impedance conditions of the SPDT switch, specified as a scalar in ohms or a vector of size 1-by-3 with each element unit in ohms. If the value is a scalar, then all three (input port and the two output port) impedances are same. If the value is a vector, the first value is the matching required at the input of the block and the other two values must be identical and represent matching required at the two output ports.

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 — Ground RF circuit terminals
on (default) | off

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

More About

Voltage-Current Relationship

The voltage-current relationship at the terminals depends on the relationship between the control voltage of the input Simulink signal, V_{ctl} , and the “Threshold voltage” on page 1-0 of the switch, $V_{threshold}$.

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

$$I_{in} = V_{in1} \cdot G_{on} + V_{in2} \cdot G_{off}$$

$$I_1 = -V_{in1} \cdot G_{on}$$

$$I_2 = -V_{in2} \cdot G_{off} + V_2 \cdot G_{Z02}$$

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

$$I_{in} = V_{in1} G_{off} + V_{in2} G_{on}$$

$$I_1 = -V_{in1} G_{off} + V_1 G_{Z01}$$

$$I_2 = -V_{in2} G_{on}$$

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

Old content from switch block

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

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

$$I_{res} \cdot R_{off} = V_{res}$$

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

$$I_{res} \cdot R_{on} = V_{res}$$

Where, R_{off} is the off resistance, R_{on} is the on resistance of the switch, V_{ctl} is the input Simulink signal, and V_{thres} is the “Threshold voltage” on page 1-0 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.

Version History

Introduced in R2015b

See Also

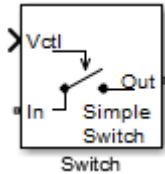
Potentiometer | Switch | SPST

Topics

“Single Pole Triple Throw Switch”

Switch

Simulink controlled two-terminal switch



Libraries:

RF Blockset / Circuit Envelope / 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.

Ports

Input

Vctl — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: double

Parameters

On resistance — On resistance
10 (default) | positive scalar

On resistance value of the switch, specified as a positive scalar in ohms.

Off resistance — Off resistance
1e-6 (default) | positive scalar

Off resistance value of the switch, specified as a positive scalar in ohms.

Threshold voltage — Threshold voltage
0 (default) | scalar

Threshold voltage of the switch, specified as a scalar in volts.

More About

Voltage-Current Relationship

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

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

$$I_{res} \cdot R_{off} = V_{res}$$

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

$$I_{res} \cdot R_{on} = V_{res}$$

Where, R_{off} is the off resistance, R_{on} is the on resistance of the switch, V_{ctl} is the input Simulink signal, and V_{thres} is the “Threshold voltage” on page 1-0 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.

Version History

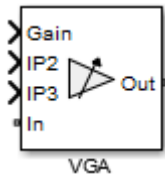
Introduced in R2015b

See Also

Potentiometer | SPST | SPDT

VGA

Model variable gain amplifier



Libraries:

RF Blockset / Circuit Envelope / Elements

Description

The VGA block models a RF Blockset variable gain amplifier (VGA). The Simulink signal controls the non-linear gain.

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

Ports

Input

Gain — Power gain signal
scalar

Power gain signal, specified as a scalar in dB. The power gain signal is a Simulink signal.

Data Types: double

IP2 — Second-order intercept point signal
scalar

Second-order intercept point signal, specified as a scalar in dBm. The second-order intercept point signal is a Simulink signal.

Data Types: double

IP3 — Third-order intercept point signal
scalar

Third-order intercept point signal, specified as a scalar in dBm. The third-order intercept point signal is a Simulink signal.

Data Types: double

Parameters

Input impedance (Ohm) — Input impedance of VGA
50 (default) | scalar

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

Output impedance (Ohm) — Output impedance of VGA
50 (default) | positive scalar

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

Intercept points convention — Input-referred or output-referred convention
Output (default) | Input

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

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

Ground and hide negative terminals, specified as on or off. Select this parameter 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.

More About

Coefficient Formulae

The VGA output voltage is a function of the input voltage:

$$V_{out} = g * V_{in} + c_2 * V_{in}^2 + c_3 * V_{in}^3$$

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

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

Version History

Introduced in R2015b

See Also

Amplifier

Variable Attenuator

Model variable attenuator



Libraries:
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.

Variable Attenuator block mask icons are dynamic and show the current state of the applied noise parameter. This table shows you how the icons on this block vary based on the state of the **Simulate noise** parameter on the block.

Simulate noise: on	Simulate noise: off

Ports

Input

Att — Simulink control signal
scalar

Simulink control signal, specified as a scalar. This control signal is an Insertion Loss in dB.

Data Types: double

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 $4 kT/R$ A²/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.

Version History

Introduced in R2016b

R2021b: Variable Attenuator block icon updated

Behavior changed in R2021b

Starting in R2021b, the Variable Attenuator block icon is now updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Variable Attenuator block, the software replaces the block icon with the R2021b version.

See Also

Attenuator | Variable Capacitor | Variable Inductor | Variable Phase Shift

Variable Capacitor

Model variable capacitor



Libraries:
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 (C_{min}) 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.

Ports

Input

C — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: `double`

Parameters

The Variable Capacitor block has no parameters.

Version History

Introduced in R2016b

R2021b: Variable Capacitor block icon updated

Behavior changed in R2021b

Starting in R2021b, the Variable Capacitor block icon is now updated. When you open a model created before R2021b containing a Variable Capacitor block, the software replaces the block icon with the R2021b version.

See Also

Variable Inductor | Capacitor

Variable Inductor

Model variable inductor

**Libraries:**

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 (L_{min}) 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.

Ports

Input

L — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: double

Parameters

The Variable Inductor block has no parameters.

Version History

Introduced in R2016b

R2021b: Variable Inductor block icon updated

Behavior changed in R2021b

Starting in R2021b, the Variable Inductor block icon is now updated. When you open a model created before R2021b containing a Variable Capacitor block, the software replaces the block icon with the R2021b version.

See Also

Variable Capacitor | Inductor

Variable Phase Shift

Model variable phase device



Libraries:
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.

Ports

Input

Phase — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: double

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.

Version History

Introduced in R2016b

R2021b: Variable Phase Shift block icon updated*Behavior changed in R2021b*

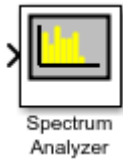
Starting in R2021b, the Variable Phase Shift block icon is now updated. When you open a model created before R2021b containing a Variable Phase Shift block, the software replaces the block icon with the R2021b version.

See Also

Variable Attenuator | Variable Capacitor | Variable Inductor | Phase Shift

Spectrum Analyzer

Display frequency spectrum of time-domain signals

**Libraries:**

RF Blockset / Circuit Envelope / Utilities

Description

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

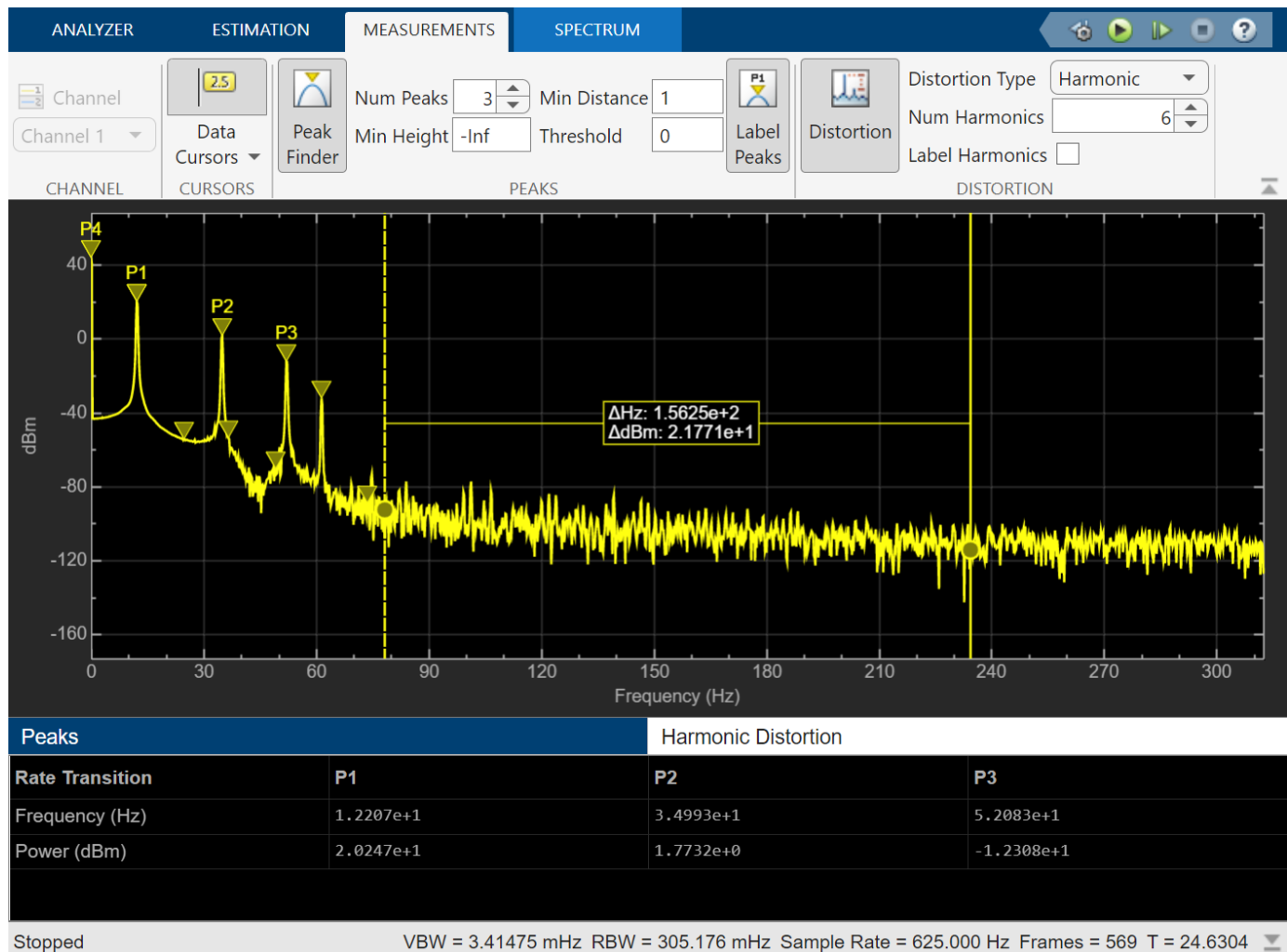
The Spectrum Analyzer block displays frequency spectra of input signals with discrete sample times.

In a RF Blockset model, to use a Spectrum Analyzer block instead of a regular scope:

- 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 port of the Spectrum Analyzer block. Set the **Output port sample time** parameter to the sample time you want the Spectrum Analyzer to use.

If your model uses a local solver, then it outputs physical signals with discrete sample times and you do not need to add a Rate Transition block. However, if you need to downsample from the solver fixed step size, you can use a Rate Transition block. For more information on using local solvers, see “Making Optimal Solver Choices for Physical Simulation”.

- 3 Use a PS-Simulink Converter block to connect the physical output signal of interest to the input port of the Spectrum Analyzer block (or to the input port of the Rate Transition block, if you are using one). For more information, see “Connecting Simscape Diagrams to Simulink Sources and Scopes”. You can also use additional signal processing blocks between the PS-Simulink Converter and the Spectrum Analyzer blocks to enhance signal quality.
- 4 Run the simulation. Once the simulation is complete, the Spectrum Analyzer, referred to here as the scope, opens and displays the frequency spectrum of the signal.



Programmatic Control

You can configure and display the Spectrum Analyzer settings from the command line with the `SpectrumAnalyzerConfiguration` object.

Ports

Input

Signal — Signals to visualize
 scalar | vector | matrix | array

Connect the signals you want to visualize. You can have up to 96 input ports. Input signals must have these characteristics:

- **Signal Domain** — Frequency or time signals.
- **Type** — Discrete signals.
- **Data type** — Any data type that Simulink supports. See “Data Types Supported by Simulink”.

- **Dimension** — One dimensional (vector), two dimensional (matrix), or multidimensional (array) signals. Input signal must have a fixed number of channels. See “Signal Dimensions” and “Determine Signal Dimensions”.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64` | `fixed point`

RBW — RBW Value
positive scalar

Specify the resolution bandwidth in Hz through this port. RBW defines the smallest positive frequency that can be resolved by the scope. You can also specify the RBW value using the **RBW (Hz)** parameter on the **Analyzer** tab.

Dependency

To enable this port, set the **RBW (Hz)** parameter on the **Analyzer** tab to `Input` port.

Data Types: `single` | `double` | `int8` | `int16` | `int32` | `int64` | `uint8` | `uint16` | `uint32` | `uint64` | `Boolean` | `fixed point`

Parameters

This section lists the block parameters for the Spectrum Analyzer block in Simscape and RF Blockset. For a complete list of parameters, see Spectrum Analyzer.

Analyzer Tab

Views

Spectrum — Type of spectrum to display
`Power` (default) | `Power Density` | `RMS`

Set the type of spectrum to display as one of these values:

- `Power` — Spectrum Analyzer shows the power spectrum.
- `Power Density` — Spectrum Analyzer shows the power spectral density. The power spectral density is the squared magnitude of the spectrum normalized to a bandwidth of 1 Hz.
- `RMS` — Spectrum Analyzer shows the root mean squared spectrum. Use this option to view the frequency of voltage or current signals.

Tunable: Yes

Programmatic Use

Block Parameter: `SpectrumType`
Type: character vector or string scalar

Data Types: `char` | `string`

Bandwidth

Sample Rate (Hz) — Sample rate the scope uses in Hz
`Inherited` (default) | positive scalar

Specify the sample rate the scope uses in Hz as one of the following:

- **Inherited** -- Use this option to specify the same sample rate as the input signal.
- **Positive scalar** -- The sample rate you specify must be at least twice the sample rate of the input signal. Otherwise, you might see unexpected behavior when visualizing your signal in the scope due to aliasing.

To display this property on the status bar, right-click the status bar at the bottom of the Spectrum Analyzer window and select **Sample Rate**.

Tunable: Yes

Programmatic Use

Block Parameter: `SampleRate`, `SampleRateSource`

Type: `double`

RBW (Hz) — Resolution bandwidth in Hz
 Auto (default) | `Input port` | positive scalar

Specify the resolution bandwidth in Hz. This parameter defines the smallest positive frequency that can be resolved by the scope. By default, this parameter is set to `Auto`. In this case, the Spectrum Analyzer determines the appropriate value to ensure that there are 1024 RBW intervals over the specified frequency span.

If you set this parameter to `Input port`, you can specify the RBW value through an input port on the block.

If you set this parameter to a numeric value, the value must allow at least two RBW intervals over the specified frequency span. In other words, the ratio of the overall frequency span to RBW must be greater than two:

$$\frac{span}{RBW} > 2$$

To display this property on the status bar, right-click the status bar at the bottom of the Spectrum Analyzer window and select **RBW**.

Tunable: Yes

Programmatic Use

Block Parameter: `RBWSource`, `RBW`

Type: character vector, string scalar, `double`

Offset (Hz) — Offset to apply to frequency axis
 0 (default) | scalar | vector

Specify the offset to apply to the frequency axis (x-axis) in units of Hz as one of the following:

- **Scalar** — Apply the same frequency offset to all channels.
- **Vector** — Apply a specific frequency offset for each channel. The vector length must be equal to the number of input channels.

The overall span must fall within the “Nyquist Frequency Interval” on page 1-163. You can control the overall span in different ways based on how you set the **Span (Hz)** parameter.

Tunable: Yes

Programmatic Use**Block Parameter:** FrequencyOffset**Type:** double**Configuration > Spectrum Analyzer Settings ()****Num Inputs** — Number of input ports

1 (default) | integer between 1 and 96

The number of input ports to the block, specified as an integer between 1 and 96. To change the number of input ports, drag a new input signal line to the block and the block automatically creates new ports.

Programmatic Use**Block Parameter:** NumInputPorts**Type:** character vector or string scalar**Values:** scalar between 1 and 96**Open at Simulation Start** — Automatically open scope when simulation starts

on (default) | off

Select this parameter to automatically open the Spectrum Analyzer window when you run the simulation.

Programmatic Use**Block Parameter:** OpenAtSimulationStart**Type:** logical**Font Size** — Font size of labels

Small (default) | Medium | Large | Extra Large

Specify the font size of labels in the display as Small, Medium, Large, and Extra Large.

Tunable: Yes**Programmatic Use****Block Parameter:** -**Type:** character vector or string scalar**Plot Type** — Plot type

Line (default) | Stem

Specify whether to display a Line or Stem plot in the Spectrum display.

Tunable: Yes**Programmatic Use****Block Parameter:** PlotType**Type:** character vector or string scalar**Y-Label** — y-axis label

character vector | string scalar

Specify the y-axis label in the Spectrum display as a character vector or a string scalar. To display signal units, add (%<SignalUnits>) to the label. When simulation starts, Simulink replaces (%<SignalUnits>) with the units associated with the signals. For example, for a velocity signal with units of m/s enter

Velocity (%<SignalUnits>)

Tunable: Yes

Programmatic Use

Block Parameter: YLabel

Type: character vector or string scalar

Y-Axis Limits — y-axis limits

[–80 20] | [ymin ymax]

Specify the y-axis limits in the Spectrum Analyzer display as a two-element numeric vector of the form [ymin ymax]. The units of the y-axis limits depend on the **Spectrum Unit** in the **Spectrum** tab.

Tunable: Yes

Programmatic Use

Block Parameter: YLimits

Type: double

Title — Display title

character vector | string

Specify the display title. Enter %<SignalLabel> to use the signal labels in the Simulink model as the axes titles.

Tunable: Yes

Programmatic Use

Block Parameter: Title

Type: character vector or string

Show Grid — Show internal grid lines

on (default) | off

Select this check box to show the grid in the Spectrum Analyzer display.

Tunable: Yes

Programmatic Use

Block Parameter: ShowGrid

Type: logical

Preserve colors for copy to clipboard — Preserve colors when copying to clipboard

off (default) | on

When you select this parameter, the scope preserves the colors when you copy the display to the clipboard using the **Copy Display** and the **Print** options in the **Analyzer** tab > **Share** section.

Tunable: Yes

Background — Window background
black (default) | color picker

Specify the background color in the scope figure.

Tunable: Yes

Axes — Axes background color
black (default) | color picker

Specify the background color of the axes.

Tunable: Yes

Labels — Color of labels
gray (default) | color picker

Specify the color of the labels, grid, and the channel names in the legend.

Tunable: Yes

Line — Channel for line property settings
channel names

Specify the channel for which you want to modify the visibility, line color, style, width, and marker properties.

Tunable: Yes

Visible — Channel visibility
on (default) | off

Select this check box to display the channel you have selected. If you clear this check box, the selected channel is no longer visible. You can also click the signal name in the legend to control its visibility. For more details, see “Legend” on page 1-0 .

Tunable: Yes

Style — Line style
- (default) | : | -. | -- | None

Specify the line style for the selected channel.

Tunable: Yes

Width — Line width
1.5 (default) | 0.5 | 1 | 2 | ...

Specify the line width for the selected channel.

Tunable: Yes

Marker — Data point markers
None (default) | + | x | ...

Specify a data point marker for the selected channel. This parameter is similar to the 'Marker' property for plots. You can choose any of the marker symbols from the drop-down list.

Tunable: Yes

Color — Line color

yellow (default) | color picker

Specify the line color for the selected channel.

Tunable: Yes

Configuration

Legend — Show or hide signal legend

button

Click the **Legend** button to enable the Spectrum Analyzer to display. The legend displays the signal names from the model. For signals with multiple channels, the scope appends a channel index after the signal name. Continuous signals have straight lines before their names and discrete signals have step-shaped lines.

You can control which signals are visible using the legend. To hide a signal in the scope legend, click the signal name. To display the signal, click the signal name again. Alternatively, you can control

which signal is visible using the **Visible** parameter in the **Spectrum Analyzer Settings** ().

To display only one signal and hide all other signals, right-click the name of the signal you want the scope to display. To show all signals, press **Esc**.

Note The legend displays only the first 20 signals. You cannot view or control any additional signals from the legend.

Tunable: Yes

Programmatic Use

Block Parameter: ShowLegend

Type: logical

Share

Copy Display — Send display to clipboard

button

Click this button to copy the scope display to the clipboard. You can preserve the color in the display by selecting the “Preserve colors for copy to clipboard” on page 1-0 parameter.

Print — Print display

button

Click this button to save the scope display as an image or a PDF or to print the display.

Estimation Tab

Averaging

Averaging Method — Smoothing method

Exponential (default) | VBW

Specify the smoothing method as one of the following:

- **Exponential** — The block computes the average over samples weighted by an exponentially decaying forgetting factor. Use the **Forgetting Factor** parameter to specify the weighted forgetting factor.
- **VBW** — The block uses a lowpass filter to smooth the trace and decrease the noise. Use the **VBW (Hz)** parameter to specify the video bandwidth (VBW) value.

For more information on the averaging methods, see “Averaging Method” on page 1-165.

Tunable: Yes

Programmatic Use

Block Parameter: AveragingMethod

Type: character vector or string scalar

VBW (Hz) — Video bandwidth

Auto (default) | Input port | positive scalar

Specify the video bandwidth as one of the following:

- **Auto** -- The Spectrum Analyzer adjusts the VBW such that the equivalent forgetting factor is 0.9.
- **Input port** -- An input port appears on the scope and you can specify the VBW value through this port.
- **Positive scalar** -- You specify a positive scalar. The Spectrum Analyzer adjusts the VBW using this value. The value you specify must be less than or equal to **Sample Rate (Hz)/2**.

For more details on the video bandwidth method, see “Averaging Method” on page 1-165.

The Spectrum Analyzer shows the VBW value in the status bar at the bottom of the display. To display the VBW value, right-click the status bar and select VBW.

Tunable: Yes

Dependency

To enable this parameter, set **Averaging Method** to VBW.

Programmatic Use

Block Parameter: VBWSource, VBW

Type: double

Forgetting Factor — Forgetting factor of weighted average method

0.9 (default) | scalar in the range [0,1]

Specify the forgetting factor of the exponential weighted averaging method as a scalar in the range [0,1].

Tunable: Yes

Dependency

To enable this parameter, set **Averaging Method** to Exponential.

Programmatic Use

Block Parameter: ForgettingFactor

Type: double

Window Options

Window — Windowing method

Hann (default) | Rectangular

Specify the windowing method to apply to the spectrum. Windowing is used to control the effect of sidelobes in spectral estimation. The window you specify affects the window length required to achieve a resolution bandwidth and the required number of samples per update. For more information about windowing, see “Windows” (Signal Processing Toolbox).

You can access additional windowing algorithms if you install DSP System Toolbox.

Tunable: Yes

Programmatic Use

Block Parameter: Window

Type: character vector or string scalar

Overlap (%) — Percentage of overlap

0 (default) | scalar in the range [0 100)

Specify the percentage of overlap between the previous and the current buffered data segments as a scalar in the range [0 100). The overlap creates a window segment that the scope uses to compute a spectral estimate. The value must be greater than or equal to zero and less than 100.

Tunable: Yes

Programmatic Use

Block Parameter: OverlapPercent

Type: double

Measurements Tab

Channel

Channel — Channel for which to obtain measurements

1 (default) | positive integer

The channel for which you need to obtain measurements, specified as a positive integer in the range [1 N], where N is the number of input channels.

Tunable: Yes

Dependency

To enable this parameter, pass some data through the scope.

Programmatic Use

See MeasurementChannel.

Cursors**Data Cursors** — Enable cursor measurements
button

Click the **Data Cursors** button to enable data cursor measurements. Each cursor tracks a vertical line along the signal. The scope displays the difference between x- and y-values of the signal at the two cursors in the box between the cursors.

Tunable: Yes

Programmatic Use

See Enabled.

Snap to data — Snap cursors to data
off (default) | on

Select this parameter to position the cursors on the signal data points.

Tunable: Yes

Programmatic Use

See SnapToData.

Lock cursor spacing — Lock cursor spacing
off (default) | on

Select this parameter to lock the frequency difference between the two cursors.

Tunable: Yes

Programmatic Use

See LockSpacing.

Peaks**Peak Finder** — Enable peak finder measurements
button

Click the **Peak Finder** button to enable peak finder measurements. An arrow appears on the plot at each maxima and a Peaks panel appears at the bottom of the scope window.

Tunable: Yes

Programmatic Use

See Enabled.

Num Peaks — Maximum number of peaks to show
3 (default) | positive integer less than 100

Specify the maximum number of peaks to show as a positive integer less than 100.

Tunable: Yes

Programmatic Use

See NumPeaks.

Min Height — Level above which scope detects peaks
- Inf (default) | real scalar value

Specify the level above which the scope detects peaks as a real scalar.

Tunable: Yes

Programmatic Use

See MinHeight.

Min Distance — Minimum number of samples between adjacent peaks
1 (default) | positive integer

Specify the minimum number of samples between adjacent peaks as a positive integer.

Tunable: Yes

Programmatic Use

See MinDistance.

Threshold — Minimum difference between height of peak and its neighboring samples
0 (default) | nonnegative scalar

Specify the minimum difference between the height of the peak and its neighboring samples as a nonnegative scalar.

Tunable: Yes

Programmatic Use

See Threshold.

Label Peaks — Label peaks
button

Click the **Label Peaks** button to label the peaks. The scope displays the labels (**P1**, **P2**, ...) above the arrows in the plot.

Tunable: Yes

Programmatic Use

See LabelPeaks.

Distortion

Distortion — Enable distortion measurements
button

Click the **Distortion** button to enable distortion measurements. A Distortion panel appears at the bottom of the scope window when you click this button.

Tunable: Yes

Programmatic Use

See Enabled.

Distortion Type — Type of measurement to display

Harmonic (default) | Intermodulation

Specify the type of measurement data to display as **Harmonic** or **Intermodulation**. For more details, see “Distortion Measurements” on page 1-156.

Tunable: Yes

Programmatic Use

See Type.

Num Harmonics — Number of harmonics to measure

6 (default) | positive integer

Specify the number of harmonics to measure as a positive integer less than or equal to 99.

Tunable: Yes

Dependency

To enable this parameter, set **Distortion Type** to **Harmonic**.

Programmatic Use

See NumHarmonics.

Label Harmonics — Label harmonics

off (default) | on

When you select this parameter, the Spectrum Analyzer adds numerical labels to harmonics in the spectrum display.

Tunable: Yes

Programmatic Use

See LabelValues.

Label Frequencies — Label frequencies

off (default) | on

When you select this parameter, the Spectrum Analyzer adds numerical labels to the first-order intermodulation product and third-order frequencies in the Spectrum Analyzer display.

Tunable: Yes

Programmatic Use

See LabelValues.

Spectrum Tab

Trace Options

Two-Sided Spectrum — Enable two-sided spectrum view
on (default) | off

Select this check box to enable a two-sided spectrum view. In this view, the Spectrum Analyzer shows both negative and positive frequencies. When the input signal is complex-valued, you must select this parameter. If you clear this check box, the Spectrum Analyzer shows a one-sided spectrum with positive frequencies only. In this case, the input signal data must be real valued.

When you clear this check box, the Spectrum Analyzer uses power folding. The y-axis values are twice the amplitude that they would be if you were to select this parameter, except at 0 and the Nyquist frequency. A one-sided power spectral density (PSD) contains the total power of the signal in the frequency interval from DC to half the Nyquist rate. For more information, see `pwelch`.

Tunable: Yes

Programmatic Use

Block Parameter: `PlotAsTwoSidedSpectrum`
Type: logical

Scale

Frequency Scale — Scale of frequency axis
Linear (default) | Log

Specify the scale to display frequencies as `Linear` or `Log`. When the frequency span contains negative frequency values, you cannot choose the logarithmic option.

Tunable: Yes

Dependency

To set the **Frequency Scale** to `Log`, clear the **Two-Sided Spectrum** check box in the **Trace Options** section in the **Spectrum** tab. If you select the **Two-Sided Spectrum** check box, then the **Frequency Scale** parameter is set to `Linear`.

Programmatic Use

Block Parameter: `FrequencyScale`
Type: character vector or string scalar

Reference load (Ω) — Reference load in Ω
1 (default) | positive real scalar

Specify the reference load in ohms that the Spectrum Analyzer uses as a reference to compute the power values.

Tunable: Yes

Dependency

To enable this parameter, set:

- **Spectrum** type to Power or Power Density.

Programmatic Use

Block Parameter: ReferenceLoad

Type: double

Spectrum Unit — Units of the spectrum

dBm (default) | dBm/Hz | dBV | Vrms

Specify the units in which the Spectrum Analyzer displays the power values as one of the following:

- dBm
- dBV
- Vrms
- dBm/Hz

Tunable: Yes

Dependency

The units available depend on the value you choose for the **Spectrum** parameter in the **Analyzer** tab.

Analyzer tab > Spectrum option	Available Units
Power	dBm
Power Density	dBm/Hz
RMS	dBV, Vrms

Programmatic Use

Block Parameter: SpectrumUnits

Type: character vector or string scalar

Property Inspector Only

Channel Names — Input channel names

[] (default) | character vector | string | array of strings or character vectors.

Input channel names, specified as a character vector, string, or array. The names appear in the legend, **Settings**, and **Measurements** panels. If you do not specify the names, the scope labels the channels as Channel 1, Channel 2, etc.

Example: ["A","B"]

Dependency

To see channel names, select **Legend** in the **Analyzer** tab.

Programmatic Use


Block Parameter: ChannelNames

Type: cell array of character vectors or string array

Maximize Axes — Maximize size of plots

Auto (default) | Off | On

- **Auto** — If you have not specified **Title** and **Y-Label**, the scope maximizes all plots.
- **On** — The scope maximizes all plots and hides all values in **Title** and **Y-label**.
- **Off** — The scope does not maximize plots.

Hover over the Spectrum Analyzer to see the maximize axes button .

Tunable: Yes

Programmatic Use

Block Parameter: MaximizeAxes

Type: character vector or string scalar

Axes Scaling — Y-axis scaling mode

OnceAtStop (default) | Manual | Auto | Updates

- **OnceAtStop** — Scale y-axis after simulation completes.
- **Manual** — Manually scale y-axis range with the **Scale Y-axis Limits** toolbar button.
- **Auto** — Scale y-axis range during and after simulation.
- **Updates** — Scale y-axis after the number of time steps specified in the “Number of Updates” on page 1-0 text box (100 by default). Scaling occurs only once during each run.

Tunable: Yes

Programmatic Use

Block Parameter: AxesScaling

Type: character vector or string scalar

Number of Updates — Number of updates before scaling

100 (default) | integer

Set this property to delay auto scaling the y-axis.

Tunable: Yes

Dependency

To enable this property, set “Axes Scaling” on page 1-0 to Updates.

Programmatic Use

Block Parameter: AxesScalingNumUpdates

Type: character vector or string scalar

Values: scalar

More About

Cursor Measurements

Measure signal values using vertical waveform cursors that track along the signal.

When you click the **Data Cursors** button in the **Measurements** tab of the Spectrum Analyzer, the spectrum display shows vertical cursors on each signal. Each cursor tracks a vertical line along the

signal. The scope displays the difference between x- and y-values of the signal at the two cursors in the box between the cursors.

To enable cursor measurements, click the **Data Cursors** button in the **Measurements** tab. The cursors appear only when the Spectrum Analyzer has at least one signal in its display.

You can use the mouse to move the vertical cursors left and right.

In the **Measurements** tab, click the **Data Cursors** drop-down arrow to select one of these options:

- **Snap to Data** — To position the cursors on the signal data points.
- **Lock Cursor Spacing** — To lock the frequency difference between the two cursors.

For modifying the cursor measurements programmatically, see the `CursorMeasurementsConfiguration` object.

Peak Finder Measurements

Compute and display peak values in the scope display.

When you click on the **Peak Finder** button in the **Measurements** tab of the Spectrum Analyzer, an arrow appears on the plot at each maxima and a **Peaks** panel appears at the bottom of the scope window. The Spectrum Analyzer computes peaks from the portion of the input signal that is currently on display in the scope, and the **Peaks** panel shows the peak values and the frequencies at which they occur.

The **Peaks** section in the **Measurements** tab allows you to specify the number of peaks you want the scope to display, the minimum height above which you want the scope to detect peaks, the minimum distance between peaks, and label the peaks.

The Spectrum Analyzer algorithm defines a peak as a local maximum with lower values present on either side of the peak. It does not consider end points as peaks. For more information on the algorithm, see the `findpeaks` function.

The peaks are valid for any units of the input signal. The letter after the value associated with each measurement indicates the abbreviation for the appropriate International System of Units (SI) prefix, such as *m* for *milli-*. For example, if the input signal is measured in volts, an *m* next to a measurement value indicates that this value is in units of millivolts.

For modifying the peak finder measurements programmatically, see the `PeakFinderConfiguration` object. For more information on these settings in the UI, see “Peaks” on page 1-0 .

Distortion Measurements

Measure harmonic distortion and intermodulation distortion.

When you click the **Distortion** button in the **Distortion** section of the **Measurements** tab, a distortion panel opens at the bottom of the Spectrum Analyzer window. This panel shows the harmonic and distortion measurement values for the input signal currently on display in the scope. The **Distortion** section in the **Measurements** tab allows you to specify the distortion type, number of harmonics, and even label the harmonics.

Note For an accurate measurement, ensure that the fundamental signal (for harmonics) or primary tones (for intermodulation) is larger than any spurious or harmonic content. To do so, you may need

to adjust the resolution bandwidth (RBW) of the Spectrum Analyzer. Make sure that the bandwidth is low enough to isolate the signal and harmonics from spurious noise content. In general, you should set the RBW value such that there is at least a 10 dB separation between the peaks of the sinusoids and the noise floor. You also might need to select a different spectral window to obtain a valid measurement.

You can set the **Distortion Type** parameter to one of these values:

- **Harmonic** -- Select **Harmonic** if your input is a single sinusoid.
- **Intermodulation** -- Select **Intermodulation** if your input is two equal-amplitude sinusoids. Intermodulation can help you determine distortion when the scope uses only a small portion of the available bandwidth.

See “Distortion Measurements” on page 1-163 for information on how distortion measurements are calculated.

Harmonic Distortion

When you set the **Distortion Type** to **Harmonic**, these fields appear in the **Harmonic Distortion** panel at the bottom of the Spectrum Analyzer window.

- **H1** — Fundamental frequency in Hz and its power in decibels of the measured power referenced to 1 milliwatt (dBm).
- **H2, H3, ...** — Harmonics frequencies in Hz and their power in decibels relative to the carrier (dBc). If the harmonics are at the same level or exceed the fundamental frequency, reduce the input power.
- **THD** — Total harmonic distortion. This value represents the ratio of the power in the harmonics D to the power in the fundamental frequency S . If the noise power is too high in relation to the harmonics, the THD value is not accurate. In this case, lower the resolution bandwidth or select a different spectral window.

$$THD = 10 \cdot \log_{10}(D/S)$$

- **SNR** — Signal-to-noise ratio (SNR). This value represents the ratio of the power in the fundamental frequency S to the power of all nonharmonic content N , including spurious signals, in decibels relative to the carrier (dBc).

$$SNR = 10 \cdot \log_{10}(S/N)$$

If you see — as the reported SNR, the total nonharmonic content of your signal is less than 30% of the total signal.

- **SINAD** — Signal-to-noise-and-distortion ratio. This value represents the ratio of the power in the fundamental frequency S to all other content (including noise N and harmonic distortion D) in decibels relative to the carrier (dBc).

$$SINAD = 10 \cdot \log_{10}\left(\frac{S}{N+D}\right)$$

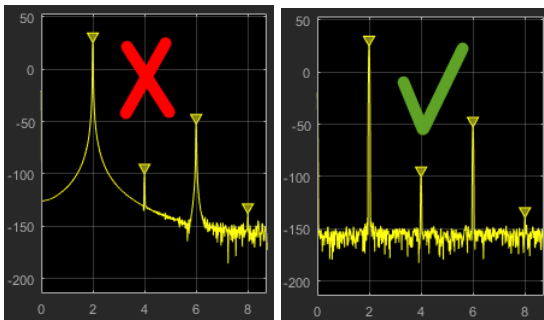
- **SFDR** — Spurious-free dynamic range (SFDR). This value represents the ratio of the power in the fundamental frequency S to power of the largest spurious signal R regardless of where it falls in the frequency spectrum. The worst spurious signal might or might not be a harmonic of the original signal. SFDR represents the smallest value of a signal that can be distinguished from a large interfering signal. SFDR includes harmonics.

$$SNR = 10 \cdot \log_{10}(S/R)$$

The harmonic distortion measurement automatically locates the largest sinusoidal component (fundamental signal frequency). It then computes the harmonic frequencies and power in each harmonic in your signal and ignores any DC component. The measurement does not include any harmonics that are outside the Spectrum Analyzer frequency span. Adjust your frequency span so that it includes all the desired harmonics.

Note To view the best harmonics, make sure that your fundamental frequency is set high enough to resolve the harmonics. However, this frequency should not be so high that aliasing occurs. For the best display of harmonic distortion, your plot should not show skirts, which indicate frequency leakage. The noise floor should be visible.

For a better display, try a Kaiser window with a large sidelobe attenuation (e.g. between 100–300 db).



Intermodulation Distortion

When you set the **Distortion Type** to **Intermodulation**, the following fields appear in the **Intermodulation Distortion** panel at the bottom of the Spectrum Analyzer window.


- **F1** — Lower fundamental first-order frequency.
- **F2** — Upper fundamental first-order frequency.
- **2F1 - F2** — Lower intermodulation product from third-order harmonics.
- **2F2 - F1** — Upper intermodulation product from third-order harmonics.
- **TOI** — Third-order intercept point. If the noise power is too high in relation to the harmonics, the TOI value will not be accurate. In this case, you should lower the resolution bandwidth or select a different spectral window. If the TOI has the same amplitude as the input two-tone signal, reduce the power of that input signal.

The intermodulation distortion measurement automatically locates the fundamental and the first-order frequencies ($F1$ and $F2$). It then computes the frequencies of the third-order intermodulation products ($2F1 - F2$ and $2F2 - F1$).

For modifying the distortion measurements programmatically, see the `DistortionMeasurementsConfiguration` object. For more information on these settings in the UI, see “Distortion” on page 1-0 .

Customize Visualization

Set configuration and style settings in the Spectrum Analyzer.

To control the settings of the display and labels, color and styling, click on **Settings** () in the **Analyzer** tab of the Spectrum Analyzer toolbar.





In the dialog box that opens, you can customize the font size, plot type, y-axis properties of the spectrum plot, and color map properties of the spectrogram plot. You can change the color of the spectrum plot, background, axes, and labels and also change the line properties.

When you view the spectrum or the spectrogram, you see only the relevant options. For more details about these options, see “Configuration > Spectrum Analyzer Settings ()” on page 1-0 .

Display Controls

Zoom and pan axes using display controls.

To scale the plot axes, use the mouse to pan around the axes and the scroll button on your mouse to zoom in and out of the plot. Additionally, you can use the buttons that appear when you hover over the plot window.

-  — Maximize the axes, hide all labels and inset the axes values.
-  — Zoom in on the plot.
-  — Pan the plot.
-  — Autoscale the axes to fit the shown data.

Algorithms

Spectrum Estimation — Welch's Method

When you select the Welch method, the power spectrum estimate is the averaged modified periodograms.

The algorithm in the Spectrum Analyzer consists of these steps:

- 1 The block buffers the input into N -point data segments. Each data segment is split into P overlapping data segments, each of length M , overlapping by D points. The data segments can be represented as:

$$x_i(n) = x(n + iD), \quad n = 0, 1, \dots, M - 1$$

$$i = 0, 1, \dots, P - 1$$

- If $D = M/2$, the overlap is 50%.
 - If $D = 0$, the overlap is 0%.
- 2 Apply a window to each of the P overlapping data segments in the time domain.

The Spectrum Analyzer uses **RBW (Hz)** in the **Analyzer** tab to determine the data window length N_{window} internally. Then, it partitions the input signal into a number of windowed data segments.

Most window functions afford more influence to the data at the center of the set than to the data at the edges, which represents a loss of information. To mitigate that loss, the individual data sets are commonly overlapped in time. For each windowed segment, compute the periodogram by computing the discrete Fourier transform. Then compute the squared magnitude of the result and divide the result by M .

$$P_{xx}^i(f) = \frac{1}{MU} \left| \sum_{n=0}^{M-1} x_i(n)w(n)e^{-j2\pi fn} \right|^2, \quad i = 0, 1, \dots, P-1$$

where U is the normalization factor for the power in the window function and is given by

$$U = \frac{1}{M} \sum_{n=0}^{M-1} w^2(n)$$

You can specify the window using the **Window** parameter in the **Estimation** tab of the Spectrum Analyzer toolstrip.

- 3** The Spectrum Analyzer calculates and plots the power spectrum, power spectrum density, and RMS using the modified Periodogram estimator. For more information about the Periodogram method, see [periodogram](#).

To determine the power spectrum estimate for Welch's method, the Spectrum Analyzer averages the result of the periodograms for the last P data segments. The averaging reduces the variance, compared to the original N -point data segment. For more details on the averaging, see "Averaging Method" on page 1-165.

$$\text{PSD}(f) = \frac{1}{P} \sum_{i=0}^{P-1} P_{xx}^i(f)$$

- 4** The Spectrum Analyzer computes the power spectral density using:

$$\text{PSD}(f) = \frac{1}{P * F_s} \sum_{i=0}^{P-1} P_{xx}^i(f)$$

- 5** The power spectrum is the product of the power spectral density and the resolution bandwidth, as given by this equation.

$$P_{\text{spectrum}}(f) = \text{PSD}(f) \times \text{RBW} = \text{PSD}(f) \times \frac{F_s \times \text{NENBW}}{N_{\text{window}}}$$

- 6**

The Spectrum Analyzer requires a minimum number of samples to compute a spectral estimate. This value is directly related to the resolution bandwidth (RBW).

$$N_{\text{samples}} = \frac{\left(1 - \frac{O_p}{100}\right) \times \text{NENBW} \times F_s}{\text{RBW}}$$

where O_p on page 1-161 is the overlap percentage, NENBW on page 1-161 is the normalized effective noise bandwidth, F_s on page 1-161 is the input sample rate, and RBW on page 1-162 is the resolution bandwidth.

The Spectrum Analyzer shows the number of samples per update in the Spectrum Analyzer status bar.

You can enable Samples/Update in the status bar only when you set **Input Domain** to Time and **Method** to Welch in the **Estimation** tab on the Spectrum Analyzer toolstrip.

Overlap Percentage (O_p)

The overlap percentage O_p is the value you specify in the **Overlap %** property. To view the **Overlap %** in the scope, click the **Estimation** tab on the Spectrum Analyzer toolstrip and navigate to the **Window Options** section.

When you increase the overlap percentage, the Spectrum Analyzer needs fewer new input samples to compute a new spectral update.

O_p	$N_{samples}$
0%	100
50%	50
80%	20

Normalized Effective Noise Bandwidth (NENBW)

The normalized effective noise bandwidth $NENBW$ is a window parameter that measures the noise performance of the window. $NENBW$ is determined using the window length and the window coefficients, and is given by the following equation:

$$NENBW = N_{window} \times \frac{\sum_{n=1}^{N_{window}} w^2(n)}{\left[\sum_{n=1}^{N_{window}} w(n) \right]^2}$$

$w(n)$ denotes the vector of window coefficients (calculated internally). N_{window} is the window length the Spectrum Analyzer needs to compute one spectral update, and is directly related to the resolution bandwidth and normalized effective noise bandwidth.

$$N_{window} = \frac{NENBW \times F_s}{RBW}$$

The rectangular window has the smallest $NENBW$, with a value of 1. All other windows have a larger $NENBW$ value. For example, the Hann window has an $NENBW$ value of approximately 1.5.

The Spectrum Analyzer shows the value of $NENBW$ in the Spectrum Analyzer status bar.

You can enable $NENBW$ only when you set **Input Domain** to Time and **Method** to Welch in the **Estimation** tab on the Spectrum Analyzer toolstrip.

Input Sample Rate (F_s)

F_s is the sample rate of the input signal. To view the **Sample Rate (Hz)** in the scope, click the **Analyzer** tab on the Spectrum Analyzer toolstrip and navigate to the **Bandwidth** section. You can enable this property in the status bar at the bottom of the Spectrum Analyzer window. Right-click the status bar and select **Sample Rate**.

Resolution Bandwidth (RBW)

Resolution bandwidth controls the spectral resolution of the displayed signal. The RBW value determines the spacing between frequencies that the scope can resolve. A smaller value gives a higher spectral resolution and lowers the noise floor, that is, the Spectrum Analyzer can resolve frequencies that are closer to each other. However, this comes at the cost of a longer sweep time.

You can set the resolution bandwidth through the **RBW (Hz)** property. To view **RBW (Hz)** in the scope, click the **Analyzer** tab on the Spectrum Analyzer toolstrip and navigate to the **Bandwidth** section.

When you set **RBW (Hz)** to:

- **Auto** -- The Spectrum Analyzer requires 1024 samples to update the display. The Spectrum Analyzer determines the appropriate resolution bandwidth to ensure that there are 1024 RBW intervals over the specified frequency span. When you set **RBW (Hz)** to Auto, the Spectrum Analyzer calculates using this equation.

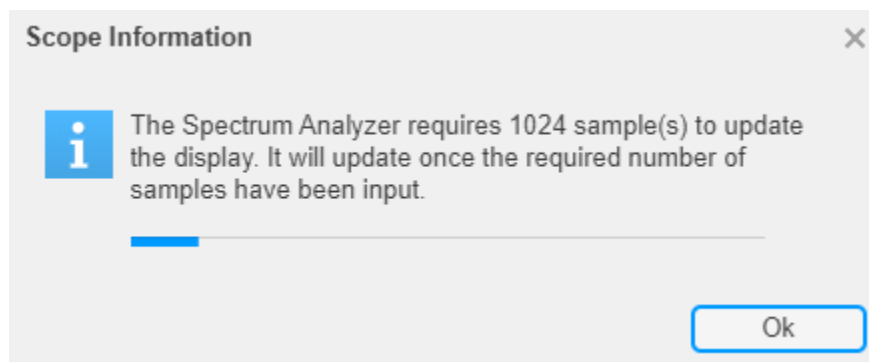
$$RBW_{auto} = \frac{span}{1024}$$

- **scalar value** -- Specify a value such that there are at least two RBW intervals over the specified frequency span. The ratio of the overall span to RBW must be greater than two:

$$\frac{span}{RBW} > 2$$

span is the frequency span over which the Spectrum Analyzer computes and plots the spectrum. Spectrum Analyzer shows the span through the **Span (Hz)** property. To view the **Span (Hz)** in the scope, click the **Estimation** tab on the Spectrum Analyzer toolstrip, navigate to the **Frequency Options** section, and set **Frequency Span** to Span and Center Frequency.

When the number of input samples is not sufficient to achieve the specified resolution bandwidth, the Spectrum Analyzer displays a message similar to this one.



The Spectrum Analyzer removes this message and displays a spectral estimate once you provide enough input samples.

You can enable this property in the status bar at the bottom of the Spectrum Analyzer window. Right-click the status bar and select **RBW**.

Nyquist Frequency Interval

When you plot the two-sided spectrum by selecting **Two-Sided Spectrum** in the **Spectrum** tab, the Nyquist frequency interval is $\left[-\frac{SampleRate}{2}, \frac{SampleRate}{2}\right] + FrequencyOffset$ Hz.

When you clear the **Two-Sided Spectrum**, the Nyquist frequency interval is $\left[0, \frac{SampleRate}{2}\right] + FrequencyOffset$ Hz.

Frequency Vector

When you set **Frequency (Hz)** to Auto, the software calculates the frequency vector for the frequency-domain input.

When you plot the two-sided spectrum by selecting **Two-Sided Spectrum** in the **Spectrum** or **Spectrogram** tab, the frequency vector is:

$$\left[-\frac{SampleRate}{2}, \frac{SampleRate}{2}\right]$$

When you clear the **Two-Sided Spectrum**, the frequency vector is:

$$\left[0, \frac{SampleRate}{2}\right]$$

Occupied BW

The Spectrum Analyzer calculates *Occupied BW* using these steps.

- 1 Calculate the total power in the measured frequency range.
- 2 Determine the lower frequency value. Starting at the lowest frequency in the range and moving upward, sum the power distributed in each frequency until the result is

$$\frac{100 - OccupiedBW\%}{2}$$

of the total power.

- 3 Determine the upper frequency value. Starting at the highest frequency in the range and moving downward, sum the power distributed in each frequency until the result reaches

$$\frac{100 - OccupiedBW\%}{2}$$

of the total power.

- 4 The bandwidth between the lower and upper power frequency values is the occupied bandwidth.
- 5 The frequency halfway between the lower and upper frequency values is the center frequency.

Distortion Measurements

The Spectrum Analyzer calculates Distortion Measurements using these steps.

- 1 Estimate spectral content by finding peaks in the spectrum. When the algorithm detects a peak, it records the width of the peak and clears all monotonically decreasing values by treating all these values as if they belong to the peak. Using this method, the algorithm removes all spectral

content centered at DC (0 Hz) from the spectrum and records the amount of bandwidth cleared (W_0).

- 2 Determine the fundamental power (P_1) from the remaining maximum value of the displayed spectrum. Create a local estimate (Fe_1) of the fundamental frequency by computing the central moment of the power near the peak. Record the bandwidth of the fundamental power content (W_1). Then remove the power from the fundamental as in step 1.
- 3 Determine the power and width of the higher-order harmonics (P_2, W_2, P_3, W_3 , etc.) in succession by examining the frequencies closest to the appropriate multiple of the local estimate (Fe_1). Remove any spectral content that decreases monotonically about the harmonic frequency from the spectrum before proceeding to the next harmonic.
- 4 After removing the DC, fundamental, and harmonic content from the spectrum, examine the power of the remaining spectrum for its sum ($P_{remaining}$), peak value ($P_{maxspur}$), and median value ($P_{estnoise}$).
- 5 Compute the sum of all the removed bandwidth as $W_{sum} = W_0 + W_1 + W_2 + \dots + W_n$.

Compute the sum of powers of the second and higher-order harmonics as $P_{harmonic} = P_2 + P_3 + P_4 + \dots + P_n$.

- 6 Estimate the sum of the noise power as:

$$P_{noise} = (P_{remaining} \cdot dF + P_{est.noise} \cdot W_{sum})/RBW$$

Where dF is the absolute difference between frequency bins, and RBW is the resolution bandwidth of the window.

- 7 Then compute the metrics for SNR, THD, SINAD, and SFDR from the estimates.

$$THD = 10 \cdot \log_{10} \left(\frac{P_{harmonic}}{P_1} \right)$$

$$SINAD = 10 \cdot \log_{10} \left(\frac{P_1}{P_{harmonic} + P_{noise}} \right)$$

$$SNR = 10 \cdot \log_{10} \left(\frac{P_1}{P_{noise}} \right)$$

$$SFDR = 10 \cdot \log_{10} \left(\frac{P_1}{\max(P_{maxspur}, \max(P_2, P_3, \dots, P_n))} \right)$$

Harmonic Measurements

- 1 The harmonic distortion measurements use the spectrum trace shown in the display as the input to the measurements. The default Hann window setting of the Spectrum Analyzer might exhibit leakage that can completely mask the noise floor of the measured signal.

The harmonic measurements attempt to correct for leakage by ignoring all frequency content that decreases monotonically away from the maximum of harmonic peaks. If the window leakage covers more than 70% of the frequency bandwidth in your spectrum, you may see a blank reading (-) reported for **SNR** and **SINAD**. If your application can tolerate the increased equivalent noise bandwidth (ENBW), consider using a Kaiser window with a high attenuation (up to 330 dB) to minimize spectral leakage.

- 2 Ignore the DC component.
- 3 After windowing, the width of each harmonic component masks the noise power in the neighborhood of the fundamental frequency and harmonics. To estimate the noise power in each

region, the Spectrum Analyzer computes the median noise level in the nonharmonic areas of the spectrum. It then extrapolates that value into each region.

- 4 N^{th} order intermodulation products occur at $A*F1 + B*F2$,

where $F1$ and $F2$ are the sinusoid input frequencies and $|A| + |B| = N$. A and B are integer values.

- 5 For intermodulation measurements, compute the third-order intercept (TOI) point as follows.

- $TOI_{lower} = P_{F1} + (P_{F2} - P_{(2F1-F2)})/2$
- $TOI_{upper} = P_{F2} + (P_{F1} - P_{(2F2-F1)})/2$
- $TOI = (TOI_{lower} + TOI_{upper})/2$

Where P is power in decibels of the measured power referenced to 1 milliwatt (dBm).

Averaging Method

The Spectrum Analyzer can calculate the moving average using two methods:

- Video bandwidth — The Spectrum Analyzer uses a time-domain lowpass filter to smooth the noise in the signal. The video bandwidth (VBW) filter smoothes the trace and decreases noise, and the Spectrum Analyzer applies the filter to the data before displaying it.

Video bandwidth is the bandwidth of the lowpass filter that Spectrum Analyzer uses to average or smooth the noise in the signal before displaying it in the scope. The Spectrum Analyzer computes the video bandwidth using this equation:

$$VBW = \frac{(1 - \lambda)RBW}{2\pi\lambda NENBW}$$

where,

- λ is the forgetting factor.
- RBW on page 1-162 is the resolution bandwidth.
- NENBW on page 1-161 is the normalized effective noise bandwidth.

Video bandwidth does not affect the level of the noise (noise floor), but only increases the signal-to-noise ratio and smoothes the trace of the noise. When you decrease the value of VBW, the signal-to-noise ratio improves.

The cutoff frequency of the video bandwidth filter is given by:

$$\omega_c = \frac{2\pi VBW}{F_s/NFFT}$$

where F_s is the input sample rate and NFFT is the number of FFT points.

The Spectrum Analyzer shows the values of sample rate, VBW, and NFFT in the status bar at the bottom of the display. To enable, right-click the status bar and select **Sample Rate**, **VBW**, and **NFFT**.

- Exponential — The moving average algorithm uses the exponential weighting method to update the weights and compute the moving average recursively for each Z vector that comes in by using the following recursive equations:

$$w_N = \lambda w_{N-1} + 1$$

$$\bar{z}_N = \left(1 - \frac{1}{w_N}\right)\bar{z}_{N-1} + \left(\frac{1}{w_N}\right)z_N$$

- λ — Forgetting factor
- w_N — Weighting factor applied to the current Z vector
- z_N — Current Z vector
- \bar{z}_{N-1} — Moving average until the previous Z vector
- $\left(1 - \frac{1}{w_N}\right)\bar{z}_{N-1}$ — Effect of the previous Z vectors on the average
- \bar{z}_N — Moving average including the current Z vector

Version History

Introduced in R2017b

R2023a: Spectrum Analyzer with improved responsiveness and toolstrip interface in Simulink

In R2023a, the Spectrum Analyzer block is more responsive and its toolstrip interface is improved to provide you easy access to spectral analysis, estimation, and measurements. You can configure and display Spectrum Analyzer settings from the command line with the `SpectrumAnalyzerConfiguration` object.

Extended Capabilities

C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

This block can be used for simulation visibility in systems that generate code. However, the block is not included in the generated code.

See Also

Objects

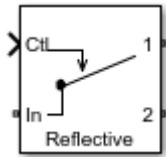
`SpectrumAnalyzerConfiguration` | `spectrumAnalyzer`

Blocks

Spectrum Analyzer

SPnT

Single pole multiple throw switch



Libraries:

RF Blockset / Circuit Envelope / Junctions

Description

The SPnT block models a single pole multiple throw switch. This block supports both reflective and absorptive switches. You can specify two to eight output ports for the switch. The control signal is a Simulink signal connected to the Ctl port of the block. The control signal specifies which output port transmits the input signal. When the input to the Ctl port is between one and the number of output ports, the Ctl signal value is rounded to the nearest integer. The output port that corresponds to the integer value transmits the data. If the rounded Ctl signal value does not correspond to any of the output ports, then all the output ports are turned off.

Ports

Input

Ctl — Simulink control signal
scalar

Simulink control signal, specified as a scalar.

Data Types: double

Parameters

Main

Insertion loss (dB) — Insertion loss of switch
1 (default) | scalar

Insertion loss value of switch, specified as a finite scalar in decibels. Specify an insertion loss value greater than zero but less than the isolation value of the switch. For the equation, see “Insertion Loss” on page 1-169.

Isolation (dB) — Isolation value of switch
70 (default) | scalar

Isolation value of switch, specified as a scalar in decibels. Specify an isolation value greater than the insertion loss value of the switch. For the equation, see “Isolation” on page 1-169.

Loading type — Switch type
Absorptive (default) | Reflective

Switch type, specified as **Absorptive** or **Reflective**.

Number of outputs — Number of outputs for switch
2 (default) | scalar

Number of outputs for the switch, specified as a scalar. The number of outputs must be greater than or equal to two or less than or equal to eight.

Port terminations (Ohm) — Port terminations
50 (default) | scalar

Port terminations representing matching required at all input and output ports, specified as a scalar in ohms. If you set the **Load type** is **Absorptive**, the impedance seen at all disconnected output ports is the value specified in **Port terminations**.

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

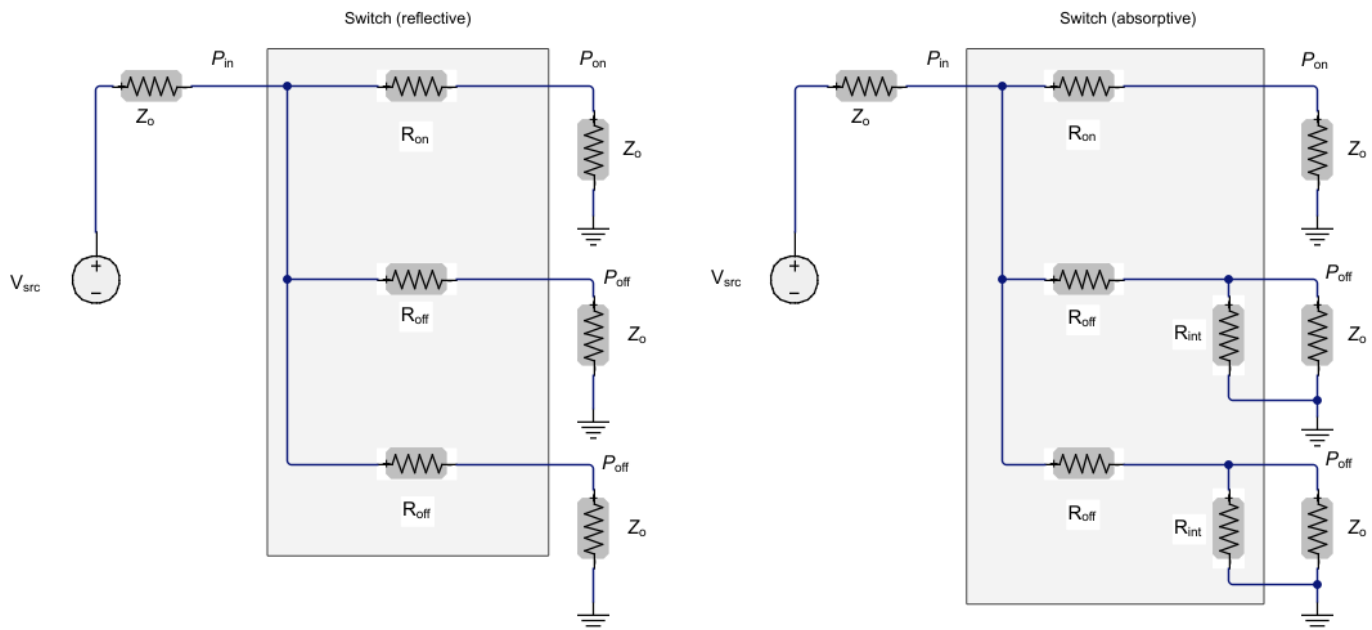
Select this parameter to internally ground and hide the reference terminals. To expose the reference terminals, clear this parameter. By exposing this terminal, you can connect it to other parts of your model.

More About

Reflective and Absorptive SPnT Switches

If there is no internal resistor connected in parallel to the load when the output terminal is turned off, any incident signal at the output side is reflected back to its source. This type of switch is known as a *reflective switch*.

If there is a matching internal resistor connected in parallel to the load when the output terminal is turned off, any incident signal at the output side is absorbed by the switch. This type of switch is known as an *absorptive switch*.



Insertion Loss

Insertion loss is given by the equation:

$$I_{\text{loss}}(\text{dB}) = -10\log_{10}\left(\frac{P_{\text{on}}}{P_{\text{in}}}\right)$$

Isolation

Isolation is given by the equation:

$$I_{\text{iso}}(\text{dB}) = -10\log_{10}\left(\frac{P_{\text{off}}}{P_{\text{in}}}\right)$$

Version History

Introduced in R2018a

References

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

See Also

Switch | SPST | SPDT

Modulator

Model RF to RF modulator



Libraries:
RF Blockset / Circuit Envelope / Systems

Description

The Modulator block models an RF to RF modulator. The Modulator block mask icons are dynamic and indicate the current state of the applied noise parameter. For more information, see “Modulator Icons” on page 1-182.

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** — Relates the ratio of power of a single sideband (SSB) at the output to the input power. This calculation assumes a matched load and source termination.
- **Open circuit voltage gain** — Value of the open circuit voltage gain parameter as the linear voltage gain term of the polynomial voltage-controlled voltage source (VCVS).
- **Polynomial coefficients** — Implements a nonlinear voltage gain according to the polynomial you specify.

Available power gain — Ratio of power of SSB at the output to input power

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

Ratio of power of SSB at output to input power, specified as a scalar in dB or a unitless ratio. For a unitless ratio, select **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 — Coefficients of polynomial specifying voltage gain

[0 1] (default) | vector

Polynomial coefficients, 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, $[a_0, a_1, a_2, \dots, a_9]$, the polynomial it represents is:

$$V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + \dots + a_9V_{in}^9$$

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 $V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + a_3V_{in}^3$. Trailing zeros are omitted. So, $[a_0, a_1, a_2]$ defines the same polynomial as $[a_0, a_1, a_2, 0]$.

The default value is $[0, 1]$, corresponding to the linear relation $V_{out} = V_{in}$.

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.

Input impedance (Ohm) — Input impedance of modulator

50 (default) | scalar

Input impedance of modulator, specified as a scalar in Ohms.

Output impedance (Ohm) — Output impedance of modulator

50 (default) | scalar

Output impedance of modulator, specified as a scalar in Ohms.

Add Image Reject filter — Image reject (IR) filter parameters

off (default) | on

Select to add the **IR filter** parameter tab. Clear to remove the tab.

Add Channel Select filter — Channel select (CS) filter parameters

off (default) | on

Select to add the **CS filter** parameter tab. Clear to remove the tab.

Ground and hide negative terminals — Ground and hide negative terminals

on (default) | off

Select to internally ground and hide the negative terminals. Clear to expose the negative terminals. When the terminals are exposed, you can connect them to other parts of your model.

Edit System — Break modulator block links and replace internal variables by appropriate values

button

Use this button to break modulator links to the library. The internal variables are replaced by their values which are estimated using modulator parameters. The Modulator becomes a simple subsystem masked only to keep the icon.

Use **Edit System** to edit the internal variables without expanding the subsystem. Use **Expand System** to expand the subsystem in Simulink canvas and to edit the subsystem.

Impairments

LO to Out isolation — Ratio of magnitude of LO voltage to leaked voltage at output port (RF)
inf dB (default) | scalar

Ratio of magnitude of LO voltage to leaked voltage at output port (RF), specified as a scalar in dB, or a unitless ratio. For a unitless ratio, select **None**.

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 in dB.

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

Add phase noise — Add phase noise
off (default) | on

Select this parameter to add phase noise to your modulator system.

Phase noise frequency offset (Hz) — Phase noise frequency offset
1 (default) | scalar | vector | matrix

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

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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) — Phase noise level
-Inf (default) | scalar | vector | matrix

Phase noise level, specified as a scalar, vector, or matrix with each element in dBc/Hz.

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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 — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, select **Add phase noise**.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns.

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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

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

Plot phase noise characteristics — Phase noise magnitude response
button

The block plots the phase noise characteristics based in the parameters specified on the **Impairments** tab and either the **Envelope bandwidth** parameter in the Configuration block when available or the value specified in the **Phase noise frequency offset (Hz)** parameter.

Dependencies

To enable this parameter, select **Add phase noise**.

Nonlinearity

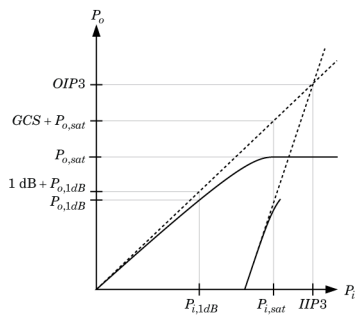
Selecting Polynomial coefficients for **Source of conversion gain** in the **Main** tab removes the **Nonlinearity** parameters.

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 Modulator can produce second-order and third-order intermodulation frequencies, in addition to a linear term.
- **Odd order**: The 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 **IP2**, **IP3**, the **1-dB gain compression power**, and the **Output 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
 inf dBm (default) | scalar

1-dB gain compression power, specified as a scalar in dBm, W, mW, or dBW.

Dependencies

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

1-dB gain compression power — 1-dB gain compression power
 inf dBm (default) | scalar

1-dB gain compression power, specified as a scalar in dBm, W, mW, or dBW.

Dependencies

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

Gain compression at saturation — Gain compression at saturation
 inf dBm (default) | scalar

Gain compression at saturation, specified as scalar in dBm, W, mW, or dBW.

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**

IR Filter

Select **Add Image Reject filter** in the **Main** tab to see the **IR Filter** parameters tab.

Design method — Simulation type

Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**.

Implementation — Implementation

LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Frequency domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is `Constant per carrier` for an ideal filter and `LC Tee` for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Lowpass or Highpass.

Implement using filter order — Implement using filter order
on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order
3 (default) | scalar

Filter order, specified as a scalar. For a **Filter type** of Lowpass or Highpass, the filter order is the number of lumped storage elements. For a **Filter type** of Bandpass or Bandstop, the number of lumped storage elements is twice the filter order.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{load}}{R_{source}} > R_{ratio}$ for Tee network implementation and $\frac{R_{load}}{R_{source}} < \frac{1}{R_{ratio}}$ 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.
-

Dependencies

To enable this parameter, select **Implement using filter order** and set **Design method** to Butterworth or Chebyshev.

Passband frequency — Passband frequency for lowpass and highpass filters
scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. The default value is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters

[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation10*log₁₀(2) (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters

[2.1 2.9]GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters

[2.1 2.9]GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation

40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance

50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance

50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration

on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to manually specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

Impulse response duration — Impulse response duration

1e-10 s (default) | scalar

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us or ns. 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. The message also specifies the minimum duration suitable for the required resolution

Dependencies

To enable this parameter, clear **Automatically estimate impulse response duration**.

Export — Save filter design to a file

button

Use this button to save filter design to a file. Valid file types are .mat and .txt.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

CS Filter

Select **Add Channel Select filter** in the **Main** tab to see the **CS Filter** parameters.

Design method — Simulation type

Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**.

Implementation — Implementation

LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Frequency domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is **Constant per carrier** for an ideal filter and **LC Tee** for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to **Ideal**.

Implement using filter order — Implement using filter order

on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order

3 (default) | scalar

Filter order, specified as a scalar. This order is the number of lumped storage elements in lowpass or highpass. In bandpass or bandstop, the number of lumped storage elements are twice the value.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{load}}{R_{source}} > R_{ratio}$ for Tee network implementation and $\frac{R_{load}}{R_{source}} < \frac{1}{R_{ratio}}$ 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.
-

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters

scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. By default, the passband frequency is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters

[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation

$10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation

40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance

50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance

50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration

on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us or ns. 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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

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

Export — Save filter design to a file
button

Use this button to save filter design to a file. Valid file types are .mat and .txt.





Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Algorithms

Modulator Icons

This table shows you how the icons on this block will vary based on how you set the **Noise figure (dB)** and **Add LO phase noise** parameters on the block.

Noise figure (dB)	Add LO phase noise: off	Add LO phase noise: on
0		
10		

Version History

Introduced in R2018a

R2021b: Modulator block icon updated*Behavior changed in R2021b*

Starting in R2021b, the Modulator block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Modulator block, the software replaces the block icon with the R2021b version.

R2022b: Estimate impulse response duration using phase noise offset frequencies

The Modulator block estimates the impulse response duration using the **Phase noise frequency offset (Hz)** parameter and visualizes the phase noise characteristics when you click the **Plot phase noise characteristics** button.

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried, and Lindner, Jurgen. "Polynomial Model Derivation of Nonlinear Amplifiers." *Department of Information Technology, University of Ulm, Germany*.

See Also

Demodulator | IQ Modulator | Mixer

Demodulator

Model RF to RF demodulator



Libraries:
RF Blockset / Circuit Envelope / Systems

Description

The Demodulator block models an RF to RF demodulator. The Demodulator block mask icons are dynamic and indicate the current state of the applied noise parameter. For more information, see “Demodulator Icons” on page 1-196 .

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** — Relates the ratio of power of a single sideband (SSB) at the output to the input power. This calculation assumes a matched load and source termination.
- **Open circuit voltage gain** — Value of the open circuit voltage gain parameter as the linear voltage gain term of the polynomial voltage-controlled voltage source (VCVS).
- **Polynomial coefficients** — Implements a nonlinear voltage gain according to the polynomial you specify.

Available power gain — Ratio of power of SSB at the output to input power

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

Ratio of power of SSB at output to input power, specified as a scalar in dB or a unitless ratio. For a unitless ratio, select **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 — Coefficients of polynomial specifying voltage gain

[0 1] (default) | vector

Polynomial coefficients, 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, $[a_0, a_1, a_2, \dots, a_9]$, the polynomial it represents is:

$$V_{out} = a_0 + a_1V_{in} + a_2V_{in}^2 + \dots + a_9V_{in}^9$$

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 $V_{out} = a_0 + a_1V_1 + a_2V_1^2 + a_3V_1^3$. Trailing zeros are omitted. So, $[a_0, a_1, a_2]$ defines the same polynomial as $[a_0, a_1, a_2, 0]$.

The default value is $[0, 1]$, corresponding to the linear relation $V_{out} = V_{in}$.

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.

Input impedance (Ohm) — Input impedance of demodulator

50 (default) | scalar

Input impedance of demodulator, specified as a scalar in Ohms.

Output impedance (Ohm) — Output impedance of demodulator

50 (default) | scalar

Output impedance of demodulator, specified as a scalar in Ohms.

Add Image Reject filter — Image reject (IR) filter parameters

off (default) | on

Select to add the **IR filter** parameter tab. Clear to remove the tab.

Add Channel Select filters — Channel select (CS) filter parameters

off (default) | on

Select to add the **CS filter** parameter tab. Clear to remove the tab.

Ground and hide negative terminals — Ground and hide negative circuit terminals

on (default) | off

Select to internally ground and hide the negative terminals. Clear to expose the negative terminals. When the terminals are exposed, you can connect them to other parts of your model.

Edit System — Break demodulator block links and replace internal variables by appropriate values
button

Use this button to break demodulator links to the library. The internal variables are replaced by their values which are estimated using demodulator parameters. The Demodulator becomes a simple subsystem masked only to keep the icon.

Use **Edit System** to edit the internal variables without expanding the subsystem. Use **Expand System** to expand the subsystem in Simulink canvas and to edit the subsystem.

Impairments

LO to In isolation — Ratio of magnitude between LO voltage to leaked voltage at input port (RF)
inf dB (default) | scalar

Ratio of magnitude of LO voltage to leaked voltage at input port (RF), specified as a scalar in dB.

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 in dB.

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

Add phase noise — Add phase noise
off (default) | on

Select this parameter to add phase noise to your demodulator system.

Phase noise frequency offset (Hz) — Phase noise frequency offset
1 (default) | scalar | vector | matrix

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

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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) — Phase noise level
-Inf (default) | scalar | vector | matrix

Phase noise level, specified as a scalar, vector, or matrix with element in dBc/Hz.

If you specify a matrix, each column corresponds to a non-DC carrier frequency of the CW source. The frequency offset values bind 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 — Automatically estimate impulse response duration
on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, select **Add phase noise**.

Impulse response duration — Impulse response duration

1e-10 s (default) | scalar

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns.

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. This message also specifies the minimum duration suitable for the required resolution

Dependencies

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

Plot phase noise characteristics — Phase noise magnitude response

button

The block plots the phase noise characteristics based in the parameters specified on the **Impairments** tab and either the **Envelope bandwidth** parameter in the Configuration block when available or the value specified in the **Phase noise frequency offset (Hz)** parameter.

Dependencies

To enable this parameter, select **Add phase noise**.

Nonlinearity

Selecting Polynomial coefficients for **Source of conversion gain** in the **Main** tab removes the **Nonlinearity** parameters.

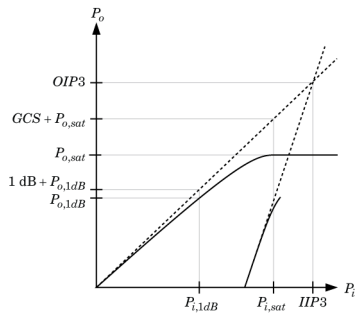
Nonlinear polynomial type — Polynomial nonlinearity

Even and odd order (default) | Odd order

Polynomial nonlinearity, specified as one of the following:

- **Even and odd order**: The Demodulator can produce second-order and third-order intermodulation frequencies, in addition to a linear term.
- **Odd order**: The 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
 Input (default) | Output

Intercept points convention, specified as Input (input referred) or Output (output referred). Use this specification for the intercept points **IP2**, **IP3**, the **1-dB gain compression power**, and the **Output 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
 inf dBm (default) | scalar

1-dB gain compression power, specified as a scalar in dBm, W, mW, or dBW.

Dependencies

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

1-dB gain compression power — 1-dB gain compression power
 inf dBm (default) | scalar

1-dB gain compression power, specified as a scalar in dBm, W, mW, or dBW.

Dependencies

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

Gain compression at saturation — Gain compression at saturation
 inf dBm (default) | scalar

Gain compression at saturation, specified as scalar in dBm, W, mW, or dBW.

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**

IR Filter

Select **Add Image Reject filter** in the **Main** tab to see the **IR Filter** parameters tab.

Design method — Simulation type

Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**.

Implementation — Implementation

LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Frequency Domain**: Model a ideal filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is Constant per carrier for an ideal filter and LC Tee for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Lowpass or Highpass.

Implement using filter order — Implement using filter order
on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order
3 (default) | scalar

Filter order, specified as a scalar. For a **Filter type** of Lowpass or Highpass, the filter order is the number of lumped storage elements. For a **Filter type** of Bandpass or Bandstop, the number of lumped storage elements is twice the filter order.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{load}}{R_{source}} > R_{ratio}$ for Tee network implementation and $\frac{R_{load}}{R_{source}} < \frac{1}{R_{ratio}}$ 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.
-

Dependencies

To enable this parameter, select **Implement using filter order** and set **Design method** to Butterworth or Chebyshev.

Passband frequency — Passband frequency for lowpass and highpass filters
scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. The default passband frequency is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters

[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation

$10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters

[2.1 2.9]GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation

40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance

50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance

50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Automatically estimate impulse response duration — Automatically estimate impulse response duration

on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to manually specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

Impulse response duration — Impulse response duration

1e-10s (default) | scalar

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns. 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. The message also specifies the minimum duration suitable for the required resolution

Dependencies

To enable this parameter, clear **Automatically estimate impulse response duration**.

Export — Save filter design to a file

button

Use this button to save filter design to a file. Valid file types are .mat and .txt.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

CS Filter

Select **Add Channel Select filters** in the **Main** tab to see the **CS Filter** parameters.

Design method — Simulation type

Ideal (default) | Butterworth | Chebyshev

Simulation type. Simulates an ideal, Butterworth, or Chebyshev filter of the type specified in **Filter type** and the model specified in **Implementation**.

Filter type — Filter type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter. Simulates a lowpass, highpass, bandpass, or bandstop filter type of the design specified in **Design method**.

Implementation — Implementation

LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Frequency Domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay impairs the filter performance when the Source and Load resistances differ from the values specified in filter parameters.

By default, the **Implementation** is **Constant per carrier** for an ideal filter and **LC Tee** for Butterworth or Chebyshev.

Passband edge frequency — Passband edge frequency

2 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to **Ideal**.

Implement using filter order — Implement using filter order

on (default) | off

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order

3 (default) | scalar

Filter order, specified as a scalar. This order is the number of lumped storage elements in lowpass or highpass. In bandpass or bandstop, the number of lumped storage elements are twice the value.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{load}}{R_{source}} > R_{ratio}$ for Tee network implementation and $\frac{R_{load}}{R_{source}} < \frac{1}{R_{ratio}}$ for Pi network implementation.

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

where:

- $\varepsilon = \sqrt{10^{(0.1R_p)} - 1}$
 - R_p is the passband ripple in dB.
-

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters

scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. By default, the passband frequency is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters

[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation

$10 \cdot \log_{10}(2)$ (default) | scalar

Passband attenuation, specified as a scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband edge frequencies — Stopband edge frequencies for ideal bandstop filters

[2.1 2.9] GHz (default) | 2-tuple vector

Stopband edge frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for ideal bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation

40 (default) | scalar

Stopband attenuation, specified as a scalar in dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance

50 (default) | scalar

Input source resistance, specified as a scalar in Ohms.

Load impedance (Ohm) — Output load resistance

50 (default) | scalar

Output load resistance, specified as a scalar in Ohms.

Automatically estimate impulse response duration — Automatically estimate impulse response duration

on (default) | off

Select to automatically estimate impulse response for phase noise. Clear to specify the impulse response duration using **Impulse response duration**.

Dependencies

To enable this parameter, set **Design method** to Ideal and **Implementation** to Frequency domain.

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

Impulse response duration used to simulate phase noise, specified as a scalar in s, ms, us, or ns. 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. The message also specifies the minimum duration suitable for the required resolution

Dependencies

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

Export — Save filter design to a file
 button

Use this button to save filter design to a file. Valid file types are .mat and .txt.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Algorithms

Demodulator Icons

This table shows you how the icons on this block will vary based on how you set the **Noise figure (dB)** and **Add LO phase noise** parameters on the block.

Noise figure (dB)	Add LO phase noise: off	Add LO phase noise: on
0		
10		

Version History

Introduced in R2018a

R2021b: Demodulator block icon updated

Behavior changed in R2021b

Starting in R2021b, the Demodulator block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a Demodulator block, the software replaces the block icon with the R2021b version.

R2022b: Estimate impulse response duration using phase noise offset frequencies

The Demodulator block estimates the impulse response duration using the **Phase noise frequency offset (Hz)** parameter and visualizes the phase noise characteristics when you click the **Plot phase noise characteristics** button.

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried, and Lindner, Jurgen. "Polynomial Model Derivation of Nonlinear Amplifiers." *Department of Information Technology, University of Ulm, Germany*.

See Also

Modulator | IQ Demodulator | Mixer

IIP2 Testbench

Measure input second intercept point of system



Libraries:
RF Blockset / Circuit Envelope / Testbenches

Description

Use the IIP2 Testbench to measure the input second intercept point (IIP2) of an RF device under test (DUT).

Parameters

Parameters

Use Internal Configuration block — Use testbench internal configuration block
on (default) | on

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Simulate noise (both stimulus and DUT internal) — Enable noise modeling in stimulus signal
on (default) | off

Select to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.

Dependencies

To enable this parameter, select **Use internal Configuration block**.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | scalar

Input power to DUT, specified as a scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are between -90 dBm and 60 dBm.

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. Input frequency must be greater than baseband bandwidth.

Output frequency (Hz) — Output frequency of DUT
2.1e9 (default) | scalar

Output frequency of DUT, specified as a scalar in Hz. Output frequency must be greater than baseband bandwidth.

Baseband bandwidth (Hz) — Baseband bandwidth of input signal

10e6 (default) | scalar

Baseband bandwidth of input signal, specified as a scalar in Hz. The value must be greater than zero.

Ratio of test tone frequency to baseband bandwidth — Position of the test tones

1/8 (default) | scalar

Position of the test tones, specified as a scalar.

Source resistance (Ohm) — Source resistance to measure DUT

50 (default) | positive finite scalar

Source resistance to measure DUT, specified as a positive finite scalar in ohms.

Load resistance (Ohm) — Load resistance to measure DUT

50 (default) | positive finite scalar

Load resistance to measure DUT, specified as a positive real scalar in ohms.

Show Response spectrum — View response spectrum using a spectrum scope during simulation

on (default) | off

Select to view response spectrum using a spectrum scope during simulation.

Note To view the response spectrum using a Spectrum Analyzer, you need a DSP System Toolbox license.

Ground and hide negative terminals — Internally ground and hide RF circuit terminals

on (default) | off

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

Version History

Introduced in R2018a

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and, Jurgen Lindner. "Polynomial Model Derivation of Nonlinear Amplifiers". *Department of Information Technology, University of Ulm, Germany*.
- [3] Kundert, Ken. "Accurate and Rapid Measurements of IP2 and IP3", *Designer's Guide Consulting, Inc.*.

See Also

Transducer Gain Testbench | Noise Figure Testbench | OIP3 Testbench | IIP3 Testbench | OIP2 Testbench

IIP3 Testbench

Measure input third intercept point of system



Libraries:

RF Blockset / Circuit Envelope / Testbenches

Description

Use the IIP3 Testbench to measure the input third intercept point (IIP3) of an RF device under test (DUT).

Parameters

Parameters

Use Internal Configuration block — Use testbench internal configuration block
on (default) | off

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Simulate noise (both stimulus and DUT internal) — Enable noise modeling in stimulus signal
on (default) | off

Select to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.

Dependencies

To enable this parameter, select **Use internal Configuration block**.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | scalar

Input power to DUT, specified as a scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are between -90 dBm and 60 dBm.

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. Input frequency must be greater than baseband bandwidth.

Output frequency (Hz) — Output frequency of DUT
2.1e9 (default) | scalar

Output frequency of DUT, specified as a scalar in Hz. Output frequency must be greater than baseband bandwidth.

Baseband bandwidth (Hz) — Baseband bandwidth of input signal
10e6 (default) | scalar

Baseband bandwidth of input signal, specified as a scalar in Hz. The value must be greater than zero.

Ratio of test tone frequency to baseband bandwidth — Position of the test tones
1/8 (default) | scalar

Position of the test tones, specified as a scalar.

Source resistance (Ohm) — Source resistance to measure DUT
50 (default) | positive finite scalar

Source resistance to measure DUT, specified as a positive finite scalar in ohms.

Load resistance (Ohm) — Load resistance to measure DUT
50 (default) | positive finite scalar

Load resistance to measure DUT, specified as a positive finite scalar in ohms.

Show Response spectrum — View response spectrum using a spectrum scope during simulation
on (default) | off

Select to view response spectrum using a spectrum scope during simulation.

Note To view the response spectrum using a Spectrum Analyzer, you need a DSP System Toolbox license.

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

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

Version History

Introduced in R2018a

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and, Jurgen Lindner. "Polynomial Model Derivation of Nonlinear Amplifiers". *Department of Information Technology, University of Ulm, Germany*.
- [3] Kundert, Ken. "Accurate and Rapid Measurements of IP2 and IP3", *Designer's Guide Consulting, Inc.*.

See Also

Transducer Gain Testbench | Noise Figure Testbench | OIP3 Testbench | OIP2 Testbench | IIP2 Testbench

Noise Figure Testbench

Measures noise figure of system



Libraries:

RF Blockset / Circuit Envelope / Testbenches

Description

Use the Noise Figure Testbench to measure the noise figure (NF) of an RF device under test (DUT).

Parameters

Parameters

Use Internal Configuration block — Use testbench internal configuration block
on (default) | off

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Simulate noise (both stimulus and DUT internal) — Enable noise modeling in stimulus signal
on (default) | off

Select to enable noise modeling in the stimulus signal entering the DUT and inside the DUT. In the NF testbench, this parameter is permanently disabled because stimulus noise is required for meaningful noise calculation.

Dependencies

To enable this parameter, select **Use internal Configuration block**.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | scalar

Input power to DUT, specified as a scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are between -90 dBm and 60 dBm

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. Input frequency must greater than baseband bandwidth.

Output frequency (Hz) — Output frequency of DUT

2.1e9 (default) | scalar

Output frequency of DUT, specified as a scalar in Hz. Output frequency must be greater than baseband bandwidth.

Baseband bandwidth (Hz) — Baseband bandwidth of input signal

10e6 (default) | scalar

Baseband bandwidth of input signal, specified as a scalar in Hz. The value must be greater than zero.

Source impedance (Ohm) — Source impedance to measure DUT

50 (default) | complex finite scalar

Source impedance to measure DUT, specified as a complex finite scalar in ohms. The real part of the impedance must be positive.

Clear noise history — Clear noise history

button (default)

Use this button to clear noise history used for internal noise measurement. When a DUT has an initial settling time duration, it is recommended to clear noise history after this duration. This improves the accuracy and convergence of noise measurement.

Show Response spectrum — View response spectrum using a spectrum scope during simulation

on (default) | off

Select to view response spectrum using a spectrum scope during simulation.

Ground and hide negative terminals — Internally ground and hide RF circuit terminals

on (default) | off

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

Version History

Introduced in R2018a

References

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

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

See Also

Transducer Gain Testbench | OIP3 Testbench | IIP3 Testbench | OIP2 Testbench | IIP2 Testbench

OIP2 Testbench

Measure output second intercept point of system



Libraries:
RF Blockset / Circuit Envelope / Testbenches

Description

Use the OIP2 Testbench block to measure the output second intercept point (OIP2) of an RF device under test (DUT).

Parameters

Parameters

Use Internal Configuration block — Use testbench internal configuration block
on (default) | off

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Simulate noise (both stimulus and DUT internal) — Enable noise modeling in stimulus signal
on (default) | off

Select to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.

Dependencies

To enable this parameter, select **Use internal Configuration block**.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | scalar

Input power to DUT, specified as a scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are between -90 dBm and 60 dBm

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. Input frequency must be greater than baseband bandwidth.

Output frequency (Hz) — Output frequency of DUT
2.1e9 (default) | scalar

Output frequency of DUT, specified as a scalar in Hz. Output frequency must be greater than baseband bandwidth.

Baseband bandwidth (Hz) — Baseband bandwidth of input signal
10e6 (default) | scalar

Baseband bandwidth of input signal, specified as a scalar in Hz. The value must be greater than zero.

Ratio of test tone frequency to baseband bandwidth — Position of the test tones
1/8 (default) | scalar

Position of the test tones, specified as a scalar.

Source resistance (Ohm) — Source resistance to measure DUT
50 (default) | positive finite scalar

Source resistance to measure DUT, specified as a positive finite scalar in ohms.

Load resistance (Ohm) — Load resistance to measure DUT
50 (default) | positive finite scalar

Load resistance to measure DUT, specified as a positive finite scalar in ohms.

Show Response spectrum — View response spectrum using a spectrum scope during simulation
on (default) | off

Select to view response spectrum using a spectrum scope during simulation.

Note To view the response spectrum using a Spectrum Analyzer, you need a DSP System Toolbox license.

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

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

Version History

Introduced in R2018a

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and, Jurgen Lindner. "Polynomial Model Derivation of Nonlinear Amplifiers". *Department of Information Technology, University of Ulm, Germany*.
- [3] Kundert, Ken. "Accurate and Rapid Measurements of IP2 and IP3", *Designer's Guide Consulting, Inc.*.

See Also

Transducer Gain Testbench | Noise Figure Testbench | OIP3 Testbench | IIP3 Testbench | IIP2 Testbench

OIP3 Testbench

Measure output third intercept point of system



Libraries:

RF Blockset / Circuit Envelope / Testbenches

Description

Use the OIP3 Testbench to measure the output third intercept point (OIP3) of an RF device under test (DUT).

Parameters

Parameters

Use Internal Configuration block — Use testbench internal configuration block
on (default) | off

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Simulate noise (both stimulus and DUT internal) — Enable noise modeling in stimulus signal
on (default) | off

Select to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.

Dependencies

To enable this parameter, select **Use internal Configuration block**.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | scalar

Input power to DUT, specified as a scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are between -90 dBm and 60 dBm

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. Input frequency must be greater than baseband bandwidth.

Output frequency (Hz) — Output frequency of DUT
2.1e9 (default) | scalar

Output frequency of DUT, specified as a scalar in Hz. Output frequency must be greater than baseband bandwidth.

Baseband bandwidth (Hz) — Baseband bandwidth of input signal
10e6 (default) | scalar

Baseband bandwidth of input signal, specified as a scalar in Hz. The value must be greater than zero.

Ratio of test tone frequency to baseband bandwidth — Position of the test tones
1/8 (default) | scalar

Position of the test tones, specified as a scalar.

Source resistance (Ohm) — Source resistance to measure DUT
50 (default) | positive finite scalar

Source resistance to measure DUT, specified as a positive finite scalar in ohms.

Load resistance (Ohm) — Load resistance to measure DUT
50 (default) | positive finite scalar

Load resistance to measure DUT, specified as a positive finite scalar in ohms.

Show Response spectrum — View response spectrum using a spectrum scope during simulation
on (default) | off

Select to view response spectrum using a spectrum scope during simulation.

Note To view the response spectrum using a Spectrum Analyzer, you need a DSP System Toolbox license.

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

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

Version History

Introduced in R2018a

References

- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and, Jurgen Lindner. "Polynomial Model Derivation of Nonlinear Amplifiers". *Department of Information Technology, University of Ulm, Germany*.
- [3] Kundert, Ken. "Accurate and Rapid Measurements of IP2 and IP3", *Designer's Guide Consulting, Inc.*.

See Also

Transducer Gain Testbench | Noise Figure Testbench | IIP3 Testbench | OIP2 Testbench | IIP2 Testbench

Transducer Gain Testbench

Measures transducer gain of system



Libraries:

RF Blockset / Circuit Envelope / Testbenches

Description

Use the Transducer Gain Testbench to measure the transducer gain (G_T) of an RF device under test (DUT).

Parameters

Parameters

Use Internal Configuration block — Use testbench internal configuration block
on (default) | off

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Simulate noise (both stimulus and DUT internal) — Enable noise modeling in stimulus signal
on (default) | off

Select to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.

Dependencies

To enable this parameter, select **Use internal Configuration block**.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | scalar

Input power to DUT, specified as a scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are between -90 dBm and 60 dBm

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. Input frequency must be greater than baseband bandwidth.

Output frequency (Hz) — Output frequency of DUT
2.1e9 (default) | scalar

Output frequency of DUT, specified as a scalar in Hz. Output frequency must be greater than baseband bandwidth.

Baseband bandwidth (Hz) — Baseband bandwidth of input signal

10e6 (default) | scalar

Baseband bandwidth of input signal, specified as a scalar in Hz. The value must be greater than zero.

Source resistance (Ohm) — Source resistance to measure DUT

50 (default) | positive finite scalar

Source resistance to measure DUT, specified as a positive finite scalar in ohms.

Load resistance (Ohm) — Load resistance to measure DUT

50 (default) | positive finite scalar

Load resistance to measure DUT, specified as a positive finite scalar in ohms.

Show Response spectrum — View response spectrum using a spectrum scope during simulation

on (default) | off

Select to view response spectrum using a spectrum scope during simulation.

Note To view the response spectrum using a Spectrum Analyzer, you need a DSP System Toolbox license.

Ground and hide negative terminals — Internally ground and hide RF circuit terminals

on (default) | off

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

Version History

Introduced in R2018a

References

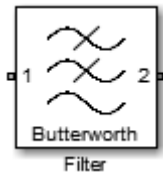
- [1] Razavi, Behzad. *RF Microelectronics*. Upper Saddle River, NJ: Prentice Hall, 2011.
- [2] Grob, Siegfried and, Jurgen Lindner. "Polynomial Model Derivation of Nonlinear Amplifiers". *Department of Information Technology, University of Ulm, Germany*.

See Also

Noise Figure Testbench | OIP3 Testbench | IIP3 Testbench | OIP2 Testbench | IIP2 Testbench

Filter

Model RF Filter



Libraries:

RF Blockset / Circuit Envelope / Elements

Description

The Filter block models RF filters of three 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 ripples of equal magnitude in the passband and monotonic in the stopband.
- **Inverse Chebyshev:** Chebyshev Type II filters have ripples of equal magnitude in the stopband and monotonic in the passband.
- **Ideal:** Ideal filters perfectly allow frequencies in the passband and completely reject frequencies in the stopband.

To filter RF complex baseband signals in Simulink, use the Idealized Baseband Filter block.

Parameters

Main

Design method — Simulation type

Butterworth (default) | Chebyshev | Inverse Chebyshev | Ideal

Simulation type, specified as one of the following:

- Ideal

Simulates an ideal filter of the type specified in **Filter type** and the model specified in **Implementation**.

- 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**.

- Inverse Chebyshev

Simulates an inverse Chebyshev filter of the type specified in **Filter type** and the Transfer function model specified in **Implementation**.

Filter type — Filter type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter type, specified as one of the following:

- **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**.

Implementation — Implementation

LC Tee | LC Pi | Transfer function | Constant per carrier | Frequency Domain

Implementation, specified as one of the following:

- **LC Tee**: Model an analog filter with an LC lumped Tee structure when the **Design method** is Butterworth or Chebyshev.
- **LC Pi**: Model an analog filter with an LC lumped Pi structure when the **Design method** is Butterworth or Chebyshev.
- **Transfer Function**: Model an analog filter using two-port S-parameters when the **Design method** is Butterworth or Chebyshev.
- **Constant per carrier**: Model a filter with either full transmission or full reflection set as constant throughout the entire envelope band around each carrier. The **Design method** is specified as ideal.
- **Frequency Domain**: Model a filter using convolution with an impulse response. The **Design method** is specified as ideal. The impulse response is computed independently for each carrier frequency to capture the ideal filtering response. When a transition between full transmission and full reflection of the ideal filter occurs within the envelope band around a carrier, the frequency-domain implementation captures this transition correctly up to a frequency resolution specified in **Impulse response duration**.

By default, the **Implementation** is **Constant per carrier** for an ideal filter and **LC Tee** for Butterworth or Chebyshev.

Note Due to causality, a delay of half the impulse response duration is included for both reflected and transmitted signals. This delay will impair the filter performance when the source and load resistances differ from the values specified as filter parameters.

Passband edge frequency — Passband edge frequency

1 GHz (default) | scalar

Passband edge frequency, specified as a scalar in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, set **Design method** to **Ideal**.

Implement using filter order — Implement using filter order
off (default) | on

Select this parameter to implement the filter order manually.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Filter order — Filter order
3 (default) | scalar

Filter order, specified as a scalar. This order is the number of lumped storage elements in lowpass or highpass. In bandpass or bandstop, the number of lumped storage elements are twice the value.

Note For even order Chebyshev filters, the resistance ratio $\frac{R_{\text{load}}}{R_{\text{source}}} > R_{\text{ratio}}$ for Tee network implementation and $\frac{R_{\text{load}}}{R_{\text{source}}} < \frac{1}{R_{\text{ratio}}}$ for Pi network implementation.

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

where:

- $\varepsilon = \sqrt{10^{(0.1R_p)} - 1}$
- R_p is the passband ripple in dB.

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency — Passband frequency for lowpass and highpass filters
scalar

Passband frequency for lowpass and highpass filters, specified as a scalar in Hz, kHz, MHz, or GHz. The default value is 1 GHz for Lowpass filters and 2 GHz for Highpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Lowpass or Highpass.

Passband frequencies — Passband frequencies for bandpass filters
[2 3] GHz (default) | 2-tuple vector

Passband frequencies for bandpass filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandstop filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandpass.

Passband attenuation (dB) — Passband attenuation
10*log₁₀(2) (default) | scalar

Passband attenuation, specified as a scalar dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Stopband frequencies — Stopband frequencies for bandstop filters
[2.1 2.9] GHz (default) | 2-tuple vector

Stopband frequencies for bandstop filters, specified as a 2-tuple vector in Hz, kHz, MHz, or GHz. This option is not available for bandpass filters.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Stopband attenuation (dB) — Stopband attenuation
40 (default) | scalar

Stopband attenuation, specified as a scalar dB. For bandstop filters, this value is applied equally to both edges of the stopband.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev and **Filter type** to Bandstop.

Source impedance (Ohm) — Input source resistance
50 (default) | scalar

Input source resistance, specified as a scalar in ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

Load impedance (Ohm) — Output load resistance
50 (default) | scalar

Output load resistance, specified as a scalar in ohms.

Dependencies

To enable this parameter, set **Design method** to Butterworth or Chebyshev.

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

Select to internally ground and hide the negative terminals. Clear to expose the negative terminals. When the terminals are exposed, you can connect them to other parts of your model.

Export — Save filter design to a file

button (default)

Use this button to save filter design to a file. Valid file types are `.mat` and `.txt`.

Visualization

Parameter 1 — Type of plots on y-axis

Voltage transfer (default) | Phase delay | Group delay

Type of plots, specified as Voltage transfer, Phase delay, or Group delay.

Parameter 2 — Type of plots

None (default) | Voltage transfer | Phase delay | Group delay

Type of plots, specified as None, Voltage transfer, Phase delay, or Group delay.

Format 1 — Scaling of y-axis

Magnitude (decibels) (default) | Magnitude (linear) | Angle (degrees) | Real | Imaginary

Scaling of y-axis, specified as,

- Magnitude(decibels), Magnitude(linear) or Angle(degrees), Real, or Imaginary for Voltage transfer parameters.
- Magnitude(decibels) or Magnitude(linear) for Phase delay or Group delay parameters.

Format 2 — Scaling of y-axis

Magnitude (decibels) (default) | Magnitude (linear) | Angle (degrees) | Real | Imaginary

Scaling of y-axis, specified as,

- Magnitude(decibels), Magnitude(linear) or Angle(degrees), Real, or Imaginary for Voltage transfer parameters.
- Magnitude(decibels) or Magnitude(linear) for Phase delay or Group delay parameters.

Frequency points — Frequency points to plot on x-axis

logspace(0,10,101) Hz (default) | vector

Frequency points to plot on x-axis, specified as a vector with each element units in Hz, kHz, MHz, or GHz.

X-axis scale — X-axis scale

Linear (default) | Logarithmic

X-axis scale, specified as Linear or Logarithmic.

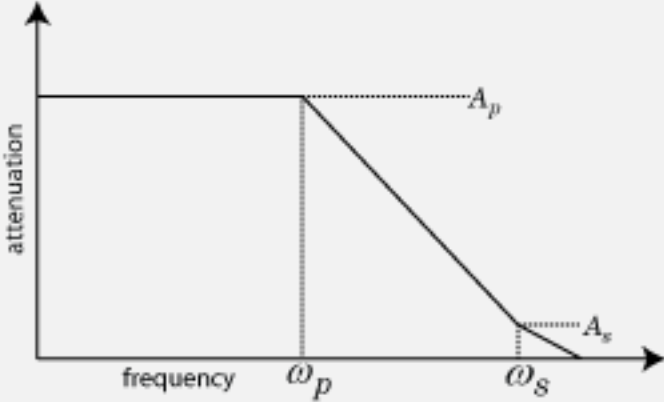
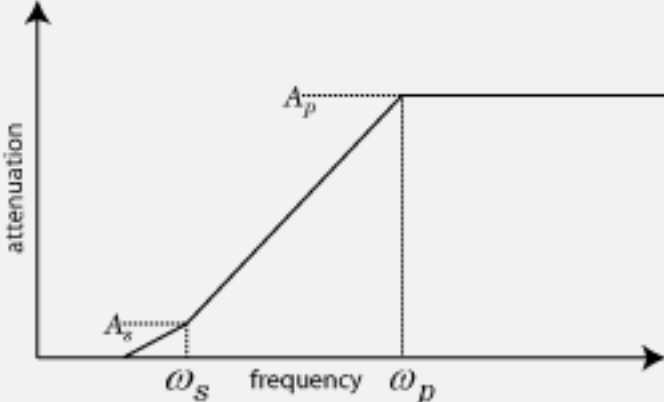
Y-axis scale — Y-axis scale

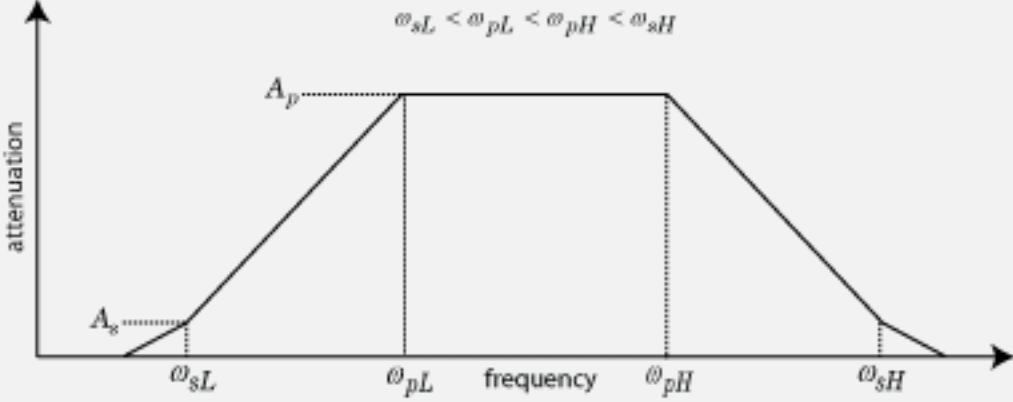
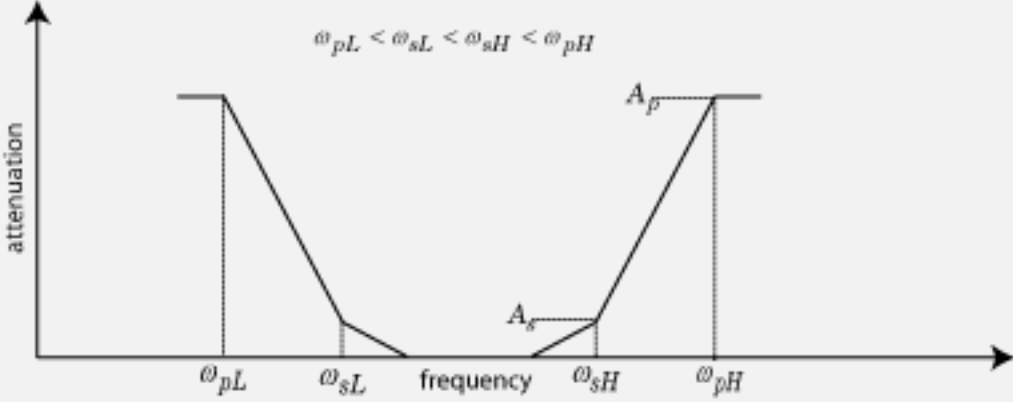
Linear (default) | Logarithmic

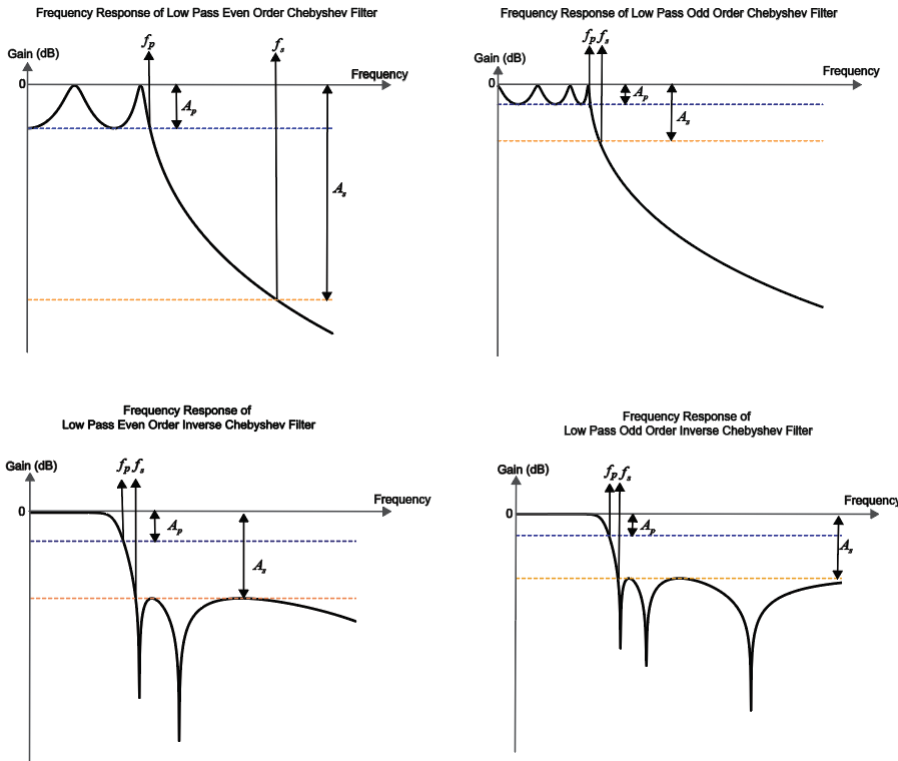
Y-axis scale, specified as Linear or Logarithmic.

More About

Frequency Responses

Filter Type	Frequency Response
Lowpass	 <p>The graph shows attenuation on the y-axis and frequency on the x-axis. The response is constant at a level A_p up to the passband frequency ω_p. After ω_p, the attenuation decreases linearly until it reaches a level A_s at the stopband frequency ω_s. Beyond ω_s, the attenuation continues to decrease at a shallower rate.</p> <p> ω_p = passband frequency ω_s = stopband frequency A_p = passband attenuation @ ω_p A_s = stopband attenuation @ ω_s </p>
Highpass	 <p>The graph shows attenuation on the y-axis and frequency on the x-axis. The response is low (attenuation A_s) at the stopband frequency ω_s and increases linearly until it reaches a level A_p at the passband frequency ω_p. Beyond ω_p, the attenuation remains constant at the level A_p.</p> <p> ω_p = passband frequency ω_s = stopband frequency A_p = passband attenuation @ ω_p A_s = stopband attenuation @ ω_s </p>

Filter Type	Frequency Response
Bandpass	 <p style="text-align: center;">$\omega_{sL} < \omega_{pL} < \omega_{pH} < \omega_{sH}$</p> <p>The graph shows attenuation on the vertical axis and frequency on the horizontal axis. The passband is a flat region between ω_{pL} and ω_{pH} with a constant attenuation A_p. The stopbands are regions where attenuation increases as frequency moves away from the passband. At the lower stopband frequency ω_{sL}, the attenuation is A_s. At the upper stopband frequency ω_{sH}, the attenuation is also A_s.</p> <p>ω_{pL}, ω_{pH} = passband frequencies ω_{sL}, ω_{sH} = stopband frequencies A_p = passband attenuation at specified passband frequencies A_s = stopband attenuation at specified stopband frequencies</p>
Bandstop	 <p style="text-align: center;">$\omega_{pL} < \omega_{sL} < \omega_{sH} < \omega_{pH}$</p> <p>The graph shows attenuation on the vertical axis and frequency on the horizontal axis. The passband is a flat region between ω_{pL} and ω_{pH} with a constant attenuation A_p. The stopband is a region where attenuation decreases as frequency moves away from the passband. At the lower stopband frequency ω_{sL}, the attenuation is A_s. At the upper stopband frequency ω_{sH}, the attenuation is also A_s.</p> <p>ω_{pL}, ω_{pH} = passband frequencies ω_{sL}, ω_{sH} = stopband frequencies A_p = passband attenuation at specified passband frequencies A_s = stopband attenuation at specified stopband frequencies</p>



Parameters To Define Filter and Design Tips

This table shows all the parameters required to design each filter correctly:

	Low-pass	High-pass	Band-pass	Band-stop
Butterworth	Order, f_p , A_p	Order, f_p , A_p	Order, f_p , A_p	Order, f_s , A_s
	f_p , f_s , A_p , A_s	f_p , f_s , A_p , A_s	f_p , f_s , A_p , A_s	f_p , f_s , A_p , A_s
Chebyshev	Order, f_p , R_p	Order, f_p , R_p	Order, f_p , R_p	Order, f_s , R_p , A_s
	f_p , f_s , R_p , A_s	f_p , f_s , R_p , A_s	f_p , f_s , R_p , A_s	f_p , f_s , R_p , A_s
Chebyshev Inverse	Order, f_p , A_p , A_s	Order, f_p , A_p , A_s	Order, f_p , A_p , A_s	Order, f_s , A_s
	f_p , f_s , A_s , A_p	f_p , f_s , A_s , A_p	f_p , f_s , A_s , A_p	f_p , f_s , A_s , A_p
	Order, f_s , A_s	Order, f_s , A_s	Order, f_s , A_s	
Legend	passband frequency - f_p , passband attenuation/passband ripple - A_p/R_p Note: Passband ripple is parsed in as passband attenuation.		stopband frequency - f_s , stopband attenuation/ stopband ripple - A_s/R_s Note: Stopband ripple is parsed in as stopband attenuation.	

Additional Design Tips

Some additional design tips:

	Low-pass	High-pass	Band-pass	Band-stop
Butterworth	Order, f_p (3dB), Auxiliary (Polynomial of Numerator21)		Order, f_p , Auxiliary (Polynomial of Numerator21, Wx)	Order, f_s , Auxiliary (Wx)
Chebyshev	Order, f_p , Auxiliary (Polynomial of Numerator21)			Order, f_s , Auxiliary (Quartics of Numerator21, Wx)
Chebyshev Inverse		Order, f_s , Auxiliary (Wx)		Order, f_s

Version History

Introduced in R2016b

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.

See Also

Inport | Outport | Configuration

S-Parameter Testbench

Measure S-parameters of system



Libraries:

RF Blockset / Circuit Envelope / Testbenches

Description

Use the S-Parameter Testbench block to measure the S-parameter data of a general RF system. The S-parameter testbench sequentially injects a stimulus signal into each port and measures the response at all ports to obtain the scattering matrix of the RF system. While the stimulus should be small signal for meaningful measurement, the testbench allows for large steady state external signals.

Parameters

Main

Use Internal Configuration block — Use testbench internal configuration block
on (default) | off

Select to use testbench internal configuration block. Clear this parameter to specify your own configuration block. Clearing this check box removes the **Approximate transient as small signal** option in the **Advanced** tab.

Note When using your own configuration block, parameters such as step size, fundamental tones, harmonic order, and simulate noise may affect the measured results.

Input power amplitude (dBm) — Input power to DUT
-30 (default) | real-valued scalar

Input power to device under test (DUT), specified as a real valued scalar in dBm. You can change the input power by entering the value in the text box or selecting a value using the knob. The specified input power represents the power available at the input ports of the DUT. The valid values are from -90 through 60 dBm.

This parameter is disabled while simulation is running when the **Approximate transient as small signal** option is checked in the **Advanced** tab.

Data Types: double

Number of ports — Number of measured ports
2 (default) | scalar integer

Number of measured ports, specified as a scalar integer limited to the range 1:128. Once you change the number of ports, new ports appear on the block.

Data Types: double

Measure all S-parameters — Measure entire S-parameters matrix
on (default) | off

Select to measure entire S-parameters matrix. This measurement is done by sequentially exciting each port and measuring all the ports for the output. When you select this box, **Save measurement result to .s2p** button appears below this parameter. In this case, the output signal S-parameters are of dimension N -by- N -by- F . N is the number of ports and F is the number of frequencies.

Clear this parameter to manually specify the S-parameters elements. In this case, the output signal S-parameters are of dimension M -by- F . M is the total number of S-parameter elements and F is the number of frequencies.

Export measurement results to sNp — Save S-parameters data to Touchstone file
button

Click to save the measured S-parameters data to a Touchstone file. This button opens a standard dialog box for browsing and choosing a file. The only file type suggested is `.sNp`, where N is the number of measured ports. This button is disabled before the first simulation takes place and while simulations are running or initializing. It captures the results of the previous simulations. The N value in `.sNp` corresponds to the number of ports previously measured.

Dependencies

To enable this parameter, select **Measure all S-parameters**.

S-Parameter elements — S-parameter elements to measure
[1 1] (default) | two-column matrix

S-parameter elements to measure, specified as a two-column matrix. Each row represents an S-parameter element. The first column represents the incident wave port and the second column represents the scattered wave port. For example, `[[2 1]; [1 1]]` indicates a two element measurement: S21 and S11. You can choose any elements from the matrix in any order, but the elements must be unique.

Dependencies

To enable this parameter, clear **Measure all S-parameters**.

Data Types: double

Workspace variable name — Variable name to store the measurement data in MATLAB workspace
'SparamObjOut' (default) | character vector

Variable name to store the measurement data in MATLAB workspace, specified as a character vector. The data is stored as an RF Toolbox `sparameters` object.

Data Types: char

Input frequency (Hz) — Carrier frequency of DUT
2.1e9 (default) | scalar

Carrier frequency of the DUT, specified as a scalar in Hz. By default, output frequency is equal to the input frequency because S-parameters are measured to quantify linear systems.

For small-signal measurements over large constant external signals, you can specify an output frequency in the **Advanced** tab when **Adjust for steady-state external signals** is selected.

Data Types: double

Baseband bandwidth (Hz) — Baseband bandwidth of input signal
10e6 (default) | positive finite scalar

Baseband bandwidth of input signal, specified as a positive finite scalar in Hz. The measured frequencies reside on this band around the input carrier frequency. This band is by default narrower than the solver envelope bandwidth to keep simulation artifacts outside of the measured results. The ratio of the two bands can be controlled using **Ratio of Envelope to Baseband bandwidths** in the **Advanced** tab.

Data Types: double

Reference Impedance (Ohm) — Impedance for S-parameter measurement
50 (default) | positive finite scalar

Impedance for S-parameter measurement, specified as a positive finite scalar in ohms. All ports are measured using the same reference impedance.

Data Types: double

Show S-Parameter spectrum — View measured values of specified S-parameters over the specified frequency bandwidth
off (default) | on

Select to view measured values of specified S-parameters over the specified frequency bandwidth using a spectrum analyzer. You can view two types of curves in the plot: **Magnitude** or **Real & Imag**. The **Magnitude** plot shows the s-parameter data magnitudes in dBm. **Real & Imag** shows both the real and imaginary parts of the S-parameters. The real and imaginary plot of the S-parameters has no units.

Note To view the S-parameters spectrum using a Spectrum Analyzer, you need a DSP System Toolbox license.

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

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

Advanced

FFT Length — Number of FFT bins
128 (default) | scalar integer power of 2

Number of FFT bins used for measurements, specified as a scalar integer power of 2. The value must be an integer power of 2. This value controls the spectral resolution of measurements over the specified bandwidths.

Measurement Time — Total time duration
12.8 μ s (default) | scalar integer

This parameter is read-only.

Total time duration in which each port output is measured to get the spectral result, specified as a scalar integer. The value is the ratio of **FFT Length** over the **Baseband bandwidth** specified in the **Main** tab. It lets you know that any system response beyond this value is not included in the spectral result. This bound is an outcome of the spectral resolution limitation.

Ratio of wait time to measurement time — Intermission between sequential measurements
0 (default) | scalar integer

Intermission between sequential measurements where the port excitation varies, specified as a scalar integer number of **Measurement time** instances. Similar to real measurements, the DUT cannot be reset to clear its internal states. By switching between measurements too fast, outputs that extend beyond the **Measurement time** can be collected in the next measurement. This value helps to avoid the contamination of the next measurement.

Ratio of Envelope to Baseband bandwidths — Ratio of internal solver envelope bandwidth to measured bandwidth
8 (default) | scalar integer power of 2

Ratio between internal solver envelope bandwidth and measured bandwidth, specified as a scalar integer power of 2.

Adjust for steady state external signals — Adjust for large steady state signals
off (default) | on

Select to adjust for large steady-state external signals and measure only the linear effect of the DUT for small signal stimulus injected by the testbench. This means that the DUT can contain elements that mix with large signals that are constant-in-time over the envelope at each carrier. When you select this parameter, you can specify an output frequency different from the input frequency.

Clear this if there are no signals injected to the DUT external to the testbench.

Note The DUT should not contain any external signals that are time varying over the envelope at each carrier.

Output frequency (Hz) — Output carrier frequency
2.1e9 (default) | positive finite scalar

Output carrier frequency, specified as a positive finite scalar.

Dependencies

To enable this parameter, select **Adjust for steady state external signals**.

Approximate transient as small signal — Choose small subset of frequencies for transient small signal analysis
off (default) | on

Select this option to choose a small subset of frequencies for transient small signal analysis. Use this parameter to accelerate the measurement of S-parameters of the large nonlinear systems around a given operation point specified by large external steady-state signals.

Use all steady-state simulation frequencies for small signal analysis — Use all frequencies automatically chosen for full-harmonic balance nonlinear solution
on (default) | off

Select this option to use all frequencies automatically chosen for a full-harmonic balance nonlinear solution. Clear to specify the frequencies that carry the small-signal transient.

Dependencies

To enable this parameter, check **Approximate transient as small signal**.

Small signal frequencies (Hz) — Frequencies that carry small signal transient

2.1e9 (default) | scalar | vector

Frequencies that carry the small signal transient, specified as a real scalar integer or vector. The specified frequencies should be contained in the entire set of simulation frequencies. If some frequencies are not contained, a warning message appears. If all frequencies are not contained, an error message appears.

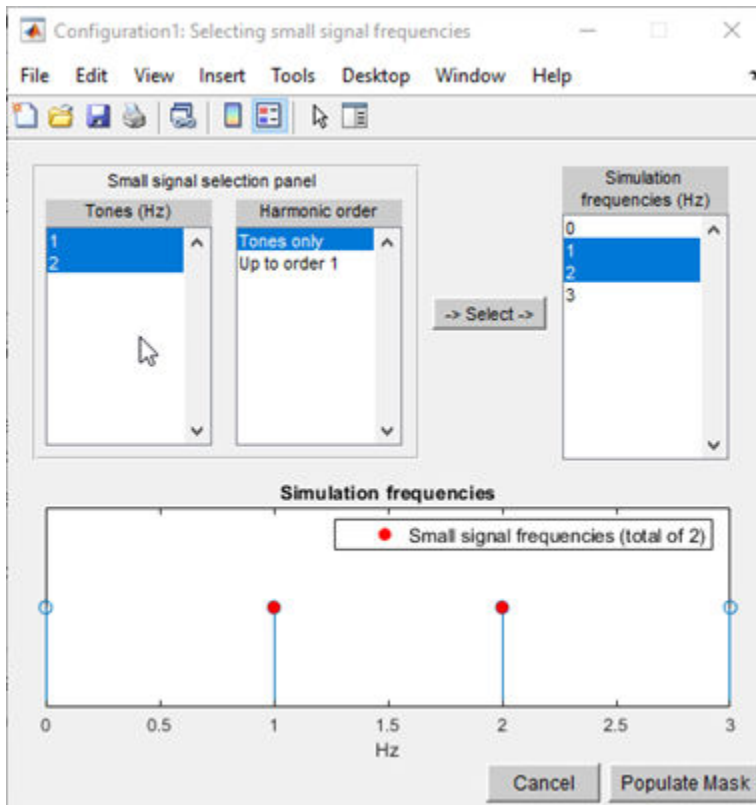
Dependencies

To enable this parameter, clear **Use all steady-state simulation frequencies for small signal analysis**.

Populate Frequencies — Tool to choose small signal transient frequencies

button

Open the tool to choose small signal transient frequencies to populate **Small signal frequencies**. The selected frequencies are a subset of the simulation frequencies determined from **Fundamental tones** and **Harmonic order** used in simulation. The entire set of simulation frequencies are given in the combo box on the right side of the dialog box, and the selected frequencies are highlighted. You can select by directly choosing the frequencies in the selection box, or by choosing the desired tones and harmonic order in the **Small signal selection panel** and clicking **Select**. The **Tones (Hz)** and **Harmonic order** values in the combo boxes are also populated using **Fundamental tones** and **Harmonic order** used in simulation.



Dependencies

To expose this parameter, clear **Use all steady-state simulation frequencies for small signal analysis**.

Version History

Introduced in R2019b

References

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

See Also

Transducer Gain Testbench | Noise Figure Testbench | OIP3 Testbench | IIP3 Testbench | IIP2 Testbench

IMT Mixer

Model mixer using intermodulation table (IMT)



Libraries:
RF Blockset / Circuit Envelope / Elements

Description

Use the IMT Mixer to perform frequency translation defined in an intermodulation table (see [1], [2], [3], and [4]) for a single tone carrier mixed with a local oscillator (LO) signal. The block includes nonlinear amplification, device and phase noise, and mixer spur visualization. For a single tone carrier F_{car} nonlinearly modulated with an LO signal of frequency F_{LO} , the mixer output intermodulation products occur at frequencies:

$$F_{out}(M, N) = |M \times F_{car} \pm N \times F_{LO}|$$

where:

- F_{car} - input RF signal carrier frequency
- F_{LO} - local oscillator frequency
- M and N are nonnegative integers (0,1,..., order of nonlinearity)

For a downconverter, the desired output tone is $|F_{car} - F_{LO}|$, and for an upconverter it is $|F_{car} + F_{LO}|$. All other combinations of M and N represent the spurious intermodulation products.

IMT Mixer block mask icons are dynamic and indicate the current state of the applied noise parameter. For more information, see "IMT Mixer Icons" on page 1-237.

Parameters

Main

Carrier frequency (Hz) — Carrier frequency

1e9 (default) | scalar

Carrier frequency, specified as a scalar in hertz. When multiple carriers exist on the input connection, this carrier frequency is selected as the RF signal input. Distance between adjacent carriers must be greater than $(1 - 1e^{-8})F_c$.

Data Types: double

Local oscillator frequency (Hz) — Local oscillator (LO) frequency

0.9e9 (default) | scalar

Local oscillator (LO) frequency, specified as a scalar in hertz.

Reference input power (dBm) — Reference input power

-10 (default) | scalar

Reference input power, specified as a scalar in dBm. The expression for the normalized input signal for the specified reference input power is:

$$10^{\left(\frac{-P_{\text{rf}} + 20\log_{10}\left(\frac{1}{\sqrt{R_{\text{in}}}}\right) + 30}{20}\right)}$$

Intermodulation tables assume that the spur levels are measured for a reference input and output power. For example, if you increase the input power by 10 dB, the spurs due to second order non-linearity will increase by 20 dB, the spurs due to third order non-linearity will increase by 30 dB, and so forth.

Nominal output power (dbm) — Nominal LO power
-20 (default) | scalar

Nominal output power, specified as a scalar in dBm. The expression for the normalized output signal for the specified reference input power is:

$$2 \times 10^{\left(\frac{P_{\text{if}} - 20\log_{10}\left(\frac{1}{\sqrt{R_{\text{out}}}}\right) - 30}{20}\right)}$$

Use data file — Specify data file to use
off (default) | on

Select this parameter to specify the data file you want to use to extract the spur table. Clear to specify your own spur values. The data file may contain any combination of IMT table, AC data, and colored spot noise in S2D or P2D format. See [4].

Data file — Data file
samplespur1.s2d (default)

Data file, specified as IMT data, AC data, and colored spot noise. To learn about the combination of inputs a data file can have and how this block extracts and calculates the parameters, see “Block Parameters When Using Data File” on page 1-234.

Dependencies

To set this parameter, first select **Use data file**.

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

Input impedance of mixer, specified as a real scalar.

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

Output impedance of mixer, specified as a real scalar.

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

Select this parameter to internally ground and hide the negative terminals. To expose the negative terminals, clear this parameter. If the terminals are exposed, the input signal is not referenced to the ground.

IMT**IMT table** — IMT spur visualization

[99 99 99; 99 0 99; 99 99 99] (default) | square matrix

IMT spur visualization, specified as a square matrix.

Output signal power (dBm) — Signal power of desired output tone

0 (default) | scalar

Signal power of the desired output tone when plotting intermodulation products, specified as a scalar.

Mixer type — Mixer type

Upconverter (default) | Downconverter

Mixer type, specified as Upconverter or Downconverter.

For a single tone carrier F_{car} nonlinearly modulated with an LO signal of frequency F_{LO} , the mixer output intermodulation products occur at frequencies:

$$F_{out}(M, N) = |M \times F_{car} \pm N \times F_{LO}|$$

where:

- F_{car} - input RF signal carrier frequency
- F_{LO} - local oscillator frequency
- M and N are nonnegative integers (0,1,..., order of nonlinearity)

For a downconverter, the desired output tone is $|F_{car} - F_{LO}|$, and for an upconverter it is $|F_{car} + F_{LO}|$. All other combinations of M and N represent the spurious intermodulation products.**Plot** — Visualize IMT table values using specified signal power and mixer type
button

Visualize IMT table values using specified signal power and mixer type.

Noise**Simulate noise** — Simulate device or phase noise

on (default) | off

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

DependenciesTo set this parameter, first select **Simulate noise**.**Noise distribution** — Noise distribution

White (default) | Piece-wise linear | Colored

Noise distribution, specified as:

- **White** – Spectral density is a single nonnegative 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.

Dependencies

To set this parameter, first select **Simulate noise**.

Noise figure (dB) — Noise figure

0 (default) | scalar

Noise figure, specified as a scalar in decibels.

Dependencies

To set this parameter, first select **Simulate noise**.

Frequencies (Hz) — Frequency data

0 (default) | scalar | vector

Frequency data, specified as a scalar for white noise or vector for piece-wise linear or colored noise in hertz.

Dependencies

To set this parameter, first select **Select noise** then select **Piece-wise linear** or **Colored** in **Noise distribution**.

Minimum noise figure (dB) — 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**.

Dependencies

To set this parameter, first select **Select noise** then select **Spot noise data** in **Noise Type**.

Optimal reflection coefficient — Optimal reflection coefficient

0 (default) | scalar | vector

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

Dependencies

To set this parameter, first select **Select noise** then select **Spot noise data** in **Noise Type**.

Equivalent normalized noise resistance — 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 **Select noise** then select **Spot noise data** in **Noise Type**.

Add LO phase noise — Add phase noise

off (default) | on

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

Dependencies

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

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

1 (default) | scalar | vector

Phase noise frequency offset with respect to LO signal, specified as a scalar or vector with each element unit in hertz.

The frequency offset values must be bounded by the envelope bandwidth of the simulation. For more information see Configuration.

Dependencies

To enable this parameter, first select **Simulate noise** then select **Add phase noise**.

Phase noise level (dBc/Hz) — Phase noise level

-Inf (default) | scalar | vector | matrix

Phase noise level, specified as a scalar or vector or matrix with elements 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, first select **Simulate noise** then select **Add phase noise**.

Automatically estimate impulse response duration — 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**.

Dependencies

To enable this parameter, first select **Simulate noise** then select **Add phase noise**.

Impulse response duration — Impulse response duration

1e-10s (default) | scalar

Impulse response duration used to simulate frequency-dependent noise, specified as a scalar in seconds. The time should be an integer multiple of the step size in the configuration block, $T_{duration} = NT_{step}$.

Dependencies

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

Algorithms

Block Parameters When Using Data File

This table shows you how the block extracts the block parameters if you choose to use a data file. The way the block extracts the parameters depends on the contents of the data file.

Contents of Data File	Reference input power (dBm)	Nominal output power (dBm)	IMT Table	Noise
IMT, AC, and noise data	Block extracts value from IMT header of data file	Block extracts value from data file	Block extracts value from data file	Block extracts value from data file
IMT data	Block extracts value from IMT header of data file	User provides value or block uses default value	Block extracts value from data file	User provides value or block uses default value
AC and IMT data	Block extracts value from IMT header of data file	Block extracts value from data file	Block extracts value from data file	User provides value or block uses default value
Noise and IMT	Block extracts value from IMT header of data file	User provides value or block uses default value	Block extracts value from data file	Block extracts value from data file
AC Data	User provides value or block uses default value	Extracted from the data file	User provides value or block uses default value	User provides value or block uses default value
AC and Noise	User provides value or block uses default value	Extracted from the data file	User provides value or block uses default value	Extracted from the data file
Noise	User provides value or block uses default value	User provides value or block uses default value	User provides value or block uses default value	Extracted from the data file

Consider the default data file `samplespur1.s2d`. This data file contains IMT, AC, and noise data. When you specify the block to use this data file by selecting the **Use data file** parameter, the block extracts the reference input power, nominal output power, IMT, and noise data from the data file. This image shows the block mask with the extracted **Reference input power (dBm)** and **Nominal output power (dbm)** parameters.

Block Parameters: IMT Mixer

IMT Mixer (mask) (link)

Mixer model using an Intermodulation Table (IMT)

Parameters

Main IMT Noise

Carrier frequency (Hz) 1e9

Local oscillator frequency (Hz) .1e9

Reference input power (dBm) -10

Nominal output power (dbm) -4.3793

Use data file

Data file samplespur1.s2d Browse ...

Input impedance (Ohm) 50

Output impedance (Ohm) 50

Ground and hide negative terminals.

OK Cancel Help Apply

To verify the extracted data, inspect the data file using this command

```
edit(samplespur1.s2d)
```

This image shows the real value of the mixer gain at the **Carrier frequency (Hz)** of 1 GHz and the **Reference input power (dBm)**.

```

BEGIN ACDATA
#AC( GHZ S MA R 50.0 FC 1 0 )
% F n11x n11y n21x n21y n12x n12y n22x n22y
1.00 0.87 -146.41 1.91 111.04 0.01 25.05 0.33 -102.98

BEGIN ACDATA
#AC( GHZ S MA R 50.0 FC 1 0 )
% F n11x n11y n21x n21y n12x n12y n22x n22y
1.00 0.87 -146.41 1.91 111.04 0.01 25.05 0.33 -102.98
1.10 0.88 -154.22 1.96 101.32 0.01 15.73 0.31 -107.00
1.20 0.90 -161.58 2.03 92.14 0.01 6.95 0.29 -110.60
1.30 0.91 -168.64 2.14 83.34 0.01 -1.44 0.27 -113.40
1.40 0.93 -175.54 2.29 74.76 0.01 -9.62 0.26 -115.72
1.50 0.93 177.53 2.52 66.22 0.01 -17.76 0.25 -117.24
1.60 0.94 179.38 2.84 57.47 0.01 -26.11 0.25 -118.00
1.70 0.95 162.34 3.33 48.00 0.01 -35.00 0.26 -118.50
1.80 0.94 152.73 4.12 37.20 0.02 -45.40 0.30 -119.27
1.90 0.91 139.27 5.50 23.10 0.02 -59.28 0.37 -122.50
2.00 0.79 115.17 8.13 -0.21 0.04 -82.19 0.53 -134.75
2.10 0.26 51.21 11.15 -46.21 0.06 -127.79 0.72 -170.05
2.20 0.51 -142.70 8.38 -96.70 0.04 -177.07 0.55 140.12
2.30 0.77 -172.64 5.28 -122.79 0.03 156.43 0.36 131.29
2.40 0.86 172.85 3.67 -137.46 0.02 142.17 0.26 123.44
2.50 0.90 163.47 2.77 -147.80 0.02 132.22 0.21 118.50
2.60 0.92 156.36 2.21 -156.17 0.01 124.26 0.17 114.57
2.70 0.93 150.44 1.84 -163.47 0.01 117.35 0.15 110.68
2.80 0.94 145.25 1.58 -170.14 0.01 111.00 0.13 106.64
2.90 0.95 140.51 1.38 -176.39 0.01 105.24 0.12 102.37
END
BEGIN NDATA
# AC( GHZ S RI R 50.0 )
% F rfmn r11x r11y r11z
1.0000 2.0000 0.3026 -0.1211 .4
2.0000 2.5000 0.3517 -0.3054 .45
3.0000 3.0000 0.0430 -0.5016 .5
4.0000 3.5000 0.4071 -0.2756 .55
5.0000 4.0000 0.2041 0.2000 .6
6.0000 4.5000 0.5665 0.0343 .65
7.0000 5.0000 0.0430 0.6016 .7
8.0000 5.5000 0.3050 0.5659 .75
9.0000 6.0000 0.3071 0.4145 .8
10.0000 6.5000 0.3419 0.3336 .85
END
BEGIN IMTDATA
! Signal Level (dbm) LO Level (dbm)
# -10 7




! N x LO ( Horizontal ) N x Signal ( Vertical )
%R1 2 3 4 5 6 7 8 9 1011121314 15
!
90 26 35 30 50 41 53 40 51 42 62 51 60 47 77 50
24 0 35 13 40 24 45 28 40 33 53 42 60 47 63
73 73 74 70 71 64 60 64 60 62 74 62 72 60
67 64 60 50 77 47 74 44 74 47 75 44 70
86 90 86 88 88 85 86 85 90 85 85 85
90 80 90 71 90 68 90 65 88 65 85
90 90 90 90 90 90 90 90 90
90 90 90 90 87 90 90 90
90 95 90 95 90 95 90 95
90 95 90 90 90 90
90 90 90 90 90
90 90 90
90 90 90
90 90
90
END

```

The block extracts the **Nominal output power (dbm)** from the data file using this equation.

IMT Mixer Icons

This table shows you how the icons on this block will vary based on how you set the **Simulate noise** and **Add LO phase noise** parameters on the block.

Simulate noise	Add LO phase noise: off	Add LO phase noise: on
off		Not applicable. For more information, see "Dependencies" on page 1-0 .
on		

Version History

Introduced in R2019b

R2021b: IMT Mixer block icon updated

Behavior changed in R2021b

Starting in R2021b, the IMT Mixer block icon has updated. The block icons are now dynamic and show the current state of the noise parameter.

When you open a model created before R2021b containing a IMT Mixer block, the software replaces the block icon with the R2021b version.

R2022a: Extract reference input power and nominal output power from data file

You can now extract reference input power and nominal output power from a data file. For more information, see "Block Parameters When Using Data File" on page 1-234.

References

- [1] <https://www.mathworks.com/help/rf/examples/visualizing-mixer-spurs.html>
- [2] <https://www.microwavejournal.com/articles/3430-the-use-of-intermodulation-tables-for-mixer-simulations>
- [3] Faria, Daniel., Lawrence Dunleavy, and Terje Svensen. "The Use of Intermodulation Tables for Mixer Simulations". Microwave Journal, April 2002. <https://www.microwavejournal.com/articles/3430-the-use-of-intermodulation-tables-for-mixer-simulations>.
- [4] "RF Mixing / Multiplication: Frequency Mixers". Electronic Notes. <https://www.electronic-notes.com/articles/radio/rf-mixer/rf-mixing-basics.php>.

See Also

Amplifier | S-Parameters

Transmission Line

Model transmission line



Libraries:
RF Blockset / Circuit Envelope / Elements

Description

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

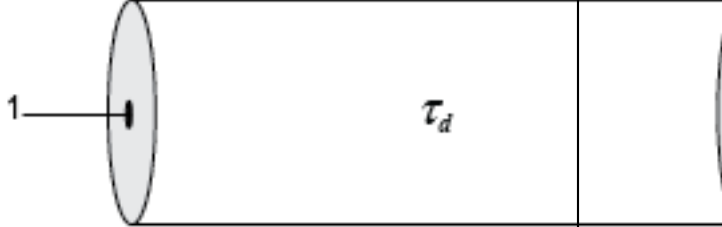
Parameters

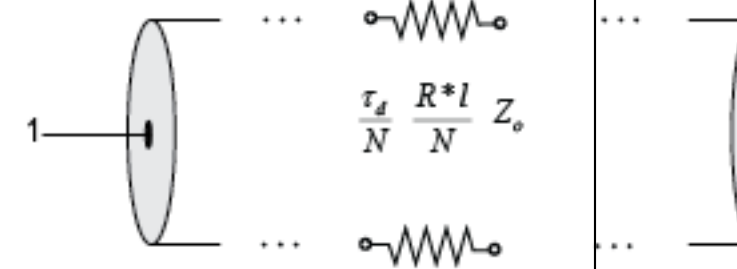
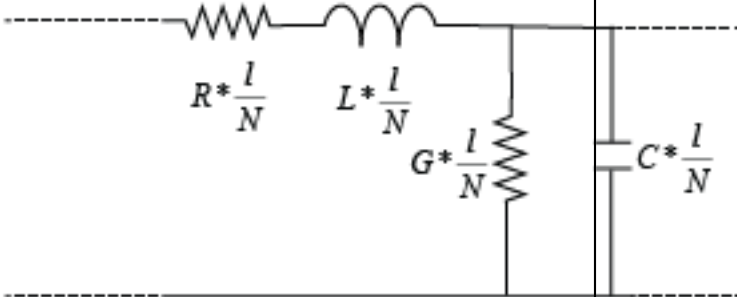
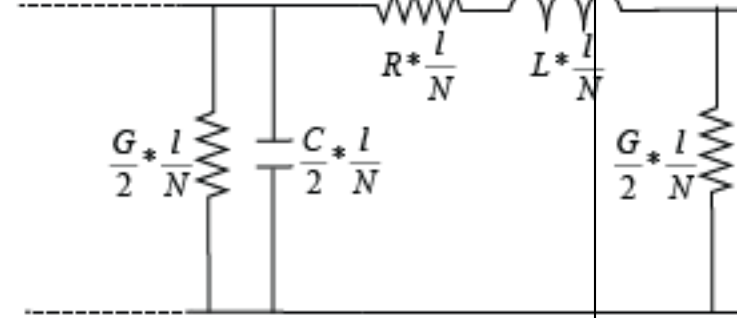
Main

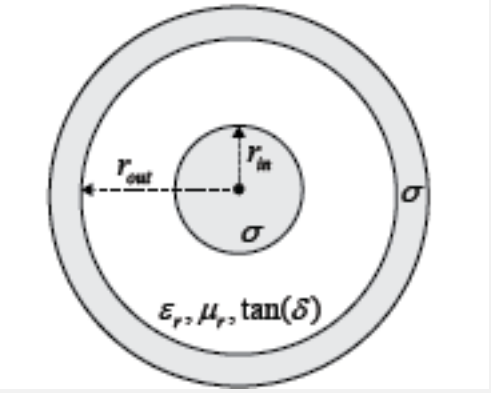
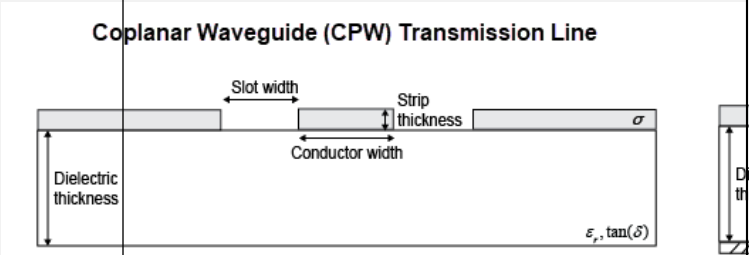
Model type — Transmission line model

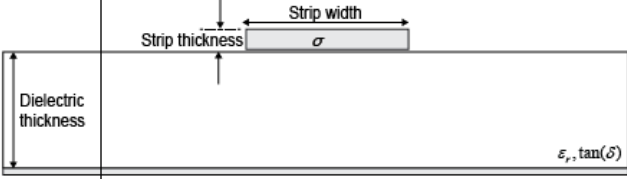
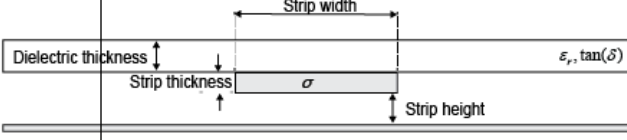
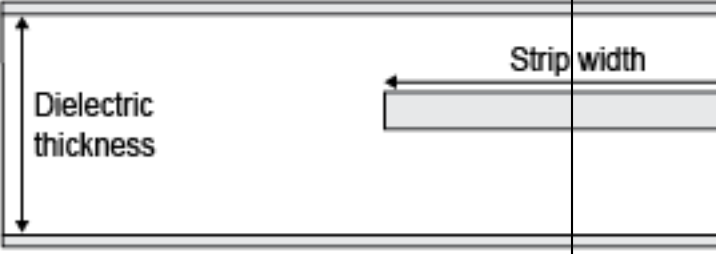
Delay-based and lossless (default) | Delay-based and lossy | Lumped parameter L-section | Lumped parameter Pi-section | Coaxial | Coplanar waveguide | Microstrip | Stripline | Two-wire | Parallel-plate | Equation-based | RLCG | ...

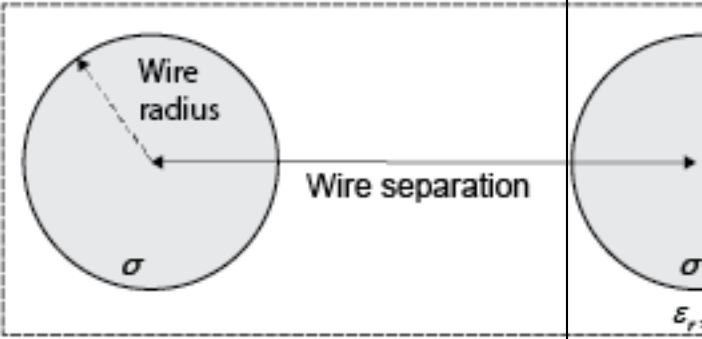
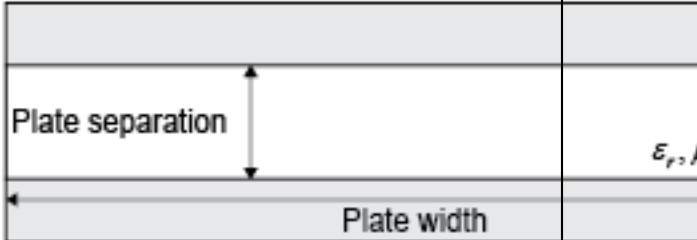
Type of transmission line model, specified as one of these.

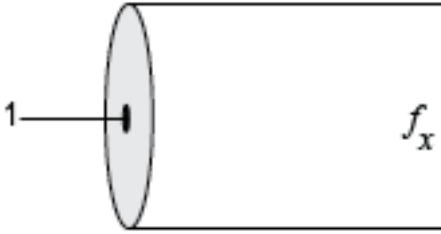
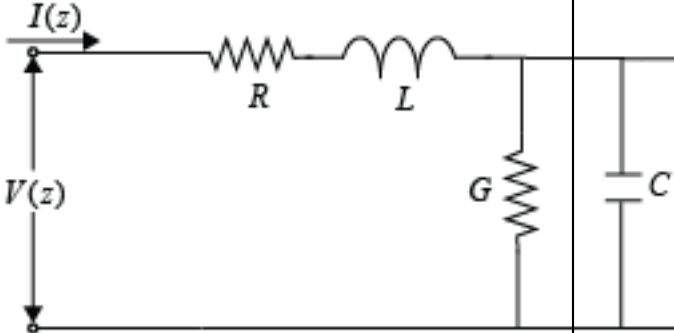
Transmission Line Types	Description
Delay-based and lossless	Model transmission line with delay but no loss. 

Transmission Line Types	Description
Delay-based and lossy -	Model transmission line with delay and loss. 
Lumped parameter L-section	Model transmission line with RLGC L-sections. 
Lumped parameter Pi-section	Model transmission line with RLGC pi-sections. 

Transmission Line Types	Description
Coaxial	<p>Model transmission line as a coaxial transmission line. The cross section of a coaxial transmission line is shown in the following figure. Its physical characteristics include the radius of the inner conductor a, and the radius of the outer conductor b.</p> 
Coplanar waveguide	<p>Model transmission line as a coplanar waveguide. The cross section of a coplanar waveguide transmission line is shown in the following figure. Its physical characteristics include conductor width w, conductor thickness t, slot width s, substrate height d, and relative permittivity constant ϵ.</p> 

Transmission Line Types	Description
<p>Microstrip</p>	<p>Model transmission line as a standard, embedded, inverted, or suspended microstrip transmission line. The cross-sections of standard, embedded, inverted, and suspended microstrip transmission lines are shown here. The physical characteristics of such a transmission line include microstrip width w, microstrip thickness t, dielectric thickness d, and relative permittivity constant ϵ.</p> <div style="text-align: center;"> <p>Standard Microstrip Transmission Line</p>  <p>Inverted Microstrip Transmission Line</p>  </div>
<p>Stripline</p>	<p>Model transmission line as stripline transmission line. The cross-section of a stripline transmission line is shown in this figure. Its physical characteristics include strip width w, strip thickness t, dielectric thickness h, and relative permittivity constant ϵ.</p> <div style="text-align: center;">  </div>

Transmission Line Types	Description
Two-wire	<p>Model transmission line as two-wire transmission line. The cross-section of a two-wire transmission line is shown in the following figure. Its physical characteristics include the radius of wires a, separation or physical distance between the wire centers S, and relative permittivity and permeability of the wires.</p> 
Parallel plate	<p>Model transmission line as a parallel-plate transmission line. The cross-section of a parallel-plate transmission line is shown in the following figure. Its physical characteristics include plate width w, and plate separation d.</p> 

Transmission Line Types	Description
Equation based	<p>Model 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.</p> 
RLCG	<p>Model transmission line as an RLCG transmission line. This line is defined 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.</p> 

Transmission delay — Delay in transmission line

4.7e-9 s (default) | real scalar | s | ms | us | ns

Delay in the transmission line, specified as a real scalar in s, ms, us, or ns.

Dependencies

To enable this parameter, choose one of the following:

- Delay-based and lossless in **Model type**.
- Delay-based and lossy in **Model type**.

Characteristic impedance — Impedance of transmission line

50 ohm (default) | real scalar | Ohm | kOhm | MOhm | GOhm

Impedance of the transmission line, specified as a real scalar in Ohm, kOhm, MOhm, or GOhm.

Dependencies

To enable this parameter, choose one of the following:

- Delay-based and lossless, Delay-based and lossy, or Equation-based in **Model type**.
- Lumped parameter L-section or Lumped parameter Pi-section in **Model type** and By characterisitic impedance and capacitance in **Parameterization**.

Resistance per unit length — Resistance per unit length of transmission line

0.3 Ohm/m (default) | positive scalar | Ohm/m | kOhm/m | MOhm/m | GOhm/m

Resistance per unit length of the transmission line, specified as a positive scalar in Ohm/m, kOhm/m, MOhm/m, or GOhm/m.

Dependencies

To enable this parameter, choose one of the following:

- Delay-based and lossy or RLCG in **Model type**.
- Lumped parameter L-section or Lumped parameter Pi-section in **Model type** and By characterisitic impedance and capacitance in **Parameterization**.

Number of segments — Number of segments in transmission line

10 (default) | positive scalar

Number of segments in the transmission line, specified as a positive scalar.

Dependencies

To enable this parameter, choose one of the following:

- Delay-based and lossy in **Model type**.
- Lumped parameter L-section or Lumped parameter Pi-section in **Model type** and By characterisitic impedance and capacitance or By inductance and capacitance in **Parameterization**.

Parameterization — Type of parameters to model segments in transmission line

By characterisitic impedance and capacitance (default) | By inductance and capacitance

Type of parameters to model segments in transmission line, specified as By characterisitic impedance and capacitance or By inductance and capacitance.

Dependencies

To enable this parameter, select Lumped parameter L-section or Lumped parameter Pi-section in **Model type**.

Capacitance per unit length — Capacitance per unit length of transmission line

94e-12 F/m (default) | positive scalar

Capacitance per unit length of the transmission line, specified as a positive scalar in F/m, mF/m, uF/m, nF/m, or pF/m.

Dependencies

To enable this parameter, choose Lumped parameter L-section, Lumped parameter Pi-section, or RLCG in **Model type**.

Conductance per unit length — Conductance per unit length of transmission line

5e-6 S/m (default) | positive scalar | S/m | mS/m | uS/m | nS/m

Conductance per unit length of the transmission line, specified as a positive scalar in S/m, mS/m, uS/m, or nS/m.

Dependencies

To enable this parameter, choose Lumped parameter L-section, Lumped parameter Pi-section, or RLCG in **Model type**.

Inductance per unit length — Inductance per unit length of transmission line

235e-9 H/m (default) | positive scalar | H/m | mH/m | uH/m | nH/m

Inductance per unit length of the transmission line, specified as a positive scalar in H/m, mH/m, uH/m, or nH/m.

Dependencies

To enable this parameter, choose one of the following:

- Lumped parameter L-section, or Lumped parameter Pi-section in **Model type** and By inductance and capacitance in **Parameterization**.
- RLCG in **Model type**

Outer radius — Outer radius of coaxial transmission line

2.57 mm (default) | positive scalar | m | cm | mm | um | in | ft

Outer radius of coaxial transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Coaxial in **Model type**.

Inner radius — Inner radius of coaxial transmission line

2.57 mm (default) | positive scalar | m | cm | mm | um | in | ft

Inner radius of coaxial transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Coaxial in **Model type**.

Conductor width — Physical width of conductor

0.6 mm (default) | positive scalar | m | cm | mm | um | in | ft

Physical width of the conductor, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Coplanar waveguide in **Model type**.

Slot width — Physical width of slot

0.2 mm (default) | positive scalar | m | cm | mm | um | in | ft

Physical width of the slot, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Coplanar waveguide in **Model type**.

Conductor backed — Infinite bottom conductor

off (default) | on

Select this parameter to add an infinite-bottom conductor to your coplanar waveguide transmission line.

Dependencies

To enable this parameter, set **Model type** to Coplanar waveguide.

Structure — Type of microstrip transmission line

Standard (default) | Embedded | Inverted | Suspended

Type of microstrip transmission line, specified as Standard, Embedded, Inverted, or Suspended.

Dependencies

To enable this parameter, set **Model type** to Microstrip.

Strip Width — Width of microstrip transmission line

0.6 mm (default) | positive scalar | m | cm | mm | um | in | ft

Width of the microstrip transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, set **Model type** to Microstrip.

Strip thickness — Physical thickness of conductor

5 um (default) | nonnegative scalar | m | cm | mm | um | in | ft

Physical thickness of the conductor, specified as a nonnegative scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Coplanar waveguide or Microstrip in **Model type**.

Strip Height — Strip height of microstrip transmission line

0.635 mm (default) | positive scalar | m | cm | mm | um | in | ft

Strip height of the inverted, suspended, or embedded microstrip transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Microstrip in **Model type** and choose Inverted, Suspended, or Embedded in **Structure**.

Wire radius — Radius of conducting wires of two-wire transmission line

0.67 mm (default) | positive scalar | m | cm | mm | um | in | ft

Radius of the conducting wires of the two-wire transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Two-wire in **Model type**.

Wire separation — Physical distance between conducting wires of two-wire transmission line
1.62 mm (default) | positive scalar | m | cm | mm | um | in | ft

Physical distance between the conducting wires of the two-wire transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Two-wire in **Model type**.

Plate width — Width of parallel-plate transmission line
5 mm (default) | positive scalar | m | cm | mm | um | in | ft

Width of the parallel-plate transmission line, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Parallel-plate in **Model type**.

Plate separation — Thickness of dielectric separating plates
1 mm (default) | positive scalar | m | cm | mm | um | in | ft

Thickness of the dielectric separating the plates, specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose Parallel-plate in **Model type**.

Phase velocity (m/s) — Propagation velocity of a uniform plane wave on transmission line
299792458 (default) | positive scalar

Propagation velocity of a uniform plane wave on the transmission line, specified as a positive scalar in meters per second

Dependencies

To enable this parameter, choose Equation-based in **Model type**.

Loss (dB/m) — Reduction in strength of signal as it travels over transmission line
0 (default) | positive scalar

Reduction in strength of the signal as it travels over the transmission line, specified as a positive scalar in meters per second

Dependencies

To enable this parameter, choose Equation-based in **Model type**.

Frequency — Modeling frequencies
1e9 Hz (default) | positive scalar | Hz | kHz | MHz | GHz

Modeling frequencies, specified as a positive scalar or vector in Hz, kHz, MHz, or GHz.

Dependencies

To enable this parameter, choose Equation-based or RLCG in **Model type**.

Interpolation method — Interpolation method used to calculate values at the modeling frequencies
Linear (default) | Spline | Cubic

Interpolation method used to calculate the values at the modeling frequencies, specified as Linear, Spline, or Cubic.

Dependencies

To enable this parameter, choose Equation-based or RLCG in **Model type**.

Conductivity of conductor — Conductivity of conductor
inf (default) | scalar | S/m | mS/m | uS/m | nS/m

Conductivity of conductor, specified as a scalar in S/m, mS/m, uS/m, or nS/m.

Dependencies

To enable this parameter, choose Coaxial, Coplanar waveguide, Microstrip, Two-wire or Parallel-plate in **Model type**.

Dielectric Thickness — Thickness of dielectric on which conductor resides
0.635 mm (default) | positive scalar | m | cm | mm | um | in | ft

Thickness of the dielectric on which the conductor resides, specified as a positive scalar in m, cm, mm, um, in, or ft.

Default values of the dielectric thickness of coplanar waveguide, standard, embedded, inverted, and suspended microstrip transmission lines are listed in this table.

Model Type	Structure	Default Dielectric Thickness in mm
Coplanar waveguide	N.A.	0.635
Microstrip	Standard and Inverted	0.635
	'Suspended'	0.3175
	'Embedded'	1.37

Dependencies

To enable this parameter, set **Model type** to Microstrip or Coplanar waveguide.

Relative permeability of dielectric — Relative permeability of dielectric
1 (default) | scalar

Relative permeability of the dielectric, specified as a scalar.

Dependencies

To enable this parameter, set **Model type** to Coaxial, Two-wire, or Parallel-plate.

Relative permittivity of dielectric — Relative permittivity of dielectric
2.2 (default) | scalar

Relative permittivity of the dielectric, specified as a scalar.

Dependencies

To enable this parameter, set **Model type** to Coaxial, Coplanar waveguide, Microstrip, Two-wire, or Parallel-plate.

Loss tangent of dielectric — Loss tangent of dielectric
0 (default) | scalar

Loss tangent of the dielectric, specified as a scalar.

Dependencies

To enable this parameter, set **Model type** to Coaxial, Coplanar waveguide, Microstrip, Two-wire, or Parallel-plate.

Line length — Physical length of transmission line
1 cm (default) | positive scalar | m | cm | mm | um | in | ft

Physical length of the transmission line or l , specified as a positive scalar in m, cm, mm, um, in, or ft.

Dependencies

To enable this parameter, choose one of the following:

- Delay-based and lossy, Coaxial, Coplanar waveguide, Microstrip, or Two-wire, Parallel-plate, Equation-based, or RLCG in **Model type**.
- Lumped parameter L-section or Lumped parameter Pi-section in **Model type** and By characterisitc impedance and capacitance or By inductance and capacitance in **Parameterization**.

Stub mode — Type of stub
Not a stub (default) | Shunt | Series

Type of stub, specified as Not a stub, Shunt, or Series. See “Parameter Calculations for Transmission Line with Stub” on page 1-252.

Dependencies

To enable this parameter, choose Coaxial, Coplanar waveguide, Microstrip Two-wire, Parallel-plate, Equation-based, or RLCG in **Model type**.

Termination of stub — Type of termination for stub
Open (default) | Short

Type of termination for stub, specified as Open or Short.

Dependencies

To enable this parameter, choose Series or Shunt in **Stub mode**.

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.

Modeling

Modeling Options — Options to model S-parameters

Frequency domain (default) | Time domain (rationalfit)

Options to model S-parameters, specified as:

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

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

Automatically estimate impulse response duration — Calculate impulse response duration automatically

off (default) | on

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

Dependencies

To enable this parameter, choose **Frequency domain** in **Modeling options**.

Impulse response duration — Manually specify impulse response duration

0 s (default) | positive scalar | s | ms | us | ns

Manually specify impulse response duration, specified as a positive scalar in s, ms, us, or ns.

Dependencies

To enable this parameter, clear **Automatically estimate impulse response duration**.

Fitting options — Fitting options for rationalfit

Share all poles (default) | Share poles by columns | Fit individually

Fitting options for rationalfit, specified as **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**.

Dependencies

To enable this parameter, choose **Time domain (rationalfit)** in **Modeling options**.

Relative error desired (dB) — Relative error acceptable in rationalfit output
-40 (default) | real scalar

Relative error acceptable in rationalfit output, specified as a real scalar in decibels.

Dependencies

To enable this parameter, choose `Time domain (rationalfit)` in **Modeling options**.

Rational fitting results — Values of rationalfit calculations
read-only (default)

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.

Dependencies

To enable this parameter, choose `Time domain (rationalfit)` in **Modeling options**.

Note Modeling tab is enabled for all transmission line types except Delay-based and lossless, Delay-based and lossy, Lumped parameter L-section, and Lumped parameter pi-section.

More About

Equations for ABCD Parameter Calculations

The following auxiliary equations are used for ABCD-parameter calculations.

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

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

where

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

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

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

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

In these equations:

- a is the radius of the inner conductor.

- b is the radius of the outer conductor.
- σ_{cond} is the conductivity in the conductor.
- μ is the permeability of the dielectric.
- ϵ is the permittivity of the dielectric.
- ϵ'' is the imaginary part of ϵ , $\epsilon'' = \epsilon_0 \epsilon_r \tan \delta$, where:
 - ϵ_0 is the permittivity of free space.
 - ϵ_r is the **Relative permittivity constant** parameter value.
 - $\tan \delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.
- f is a vector of internal modeling frequencies.
- Z_0 is the specified characteristic impedance.
- k is a vector whose elements correspond to the elements of the input vector, `freq`. The block calculates k from the specified parameters as $k = \alpha_a + i\beta$, where α_a is the attenuation coefficient and β is the wave number. The attenuation coefficient α_a is related to the specified loss, α , by

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

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

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

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

Parameter Calculations for Distributed Transmission Line

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

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

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

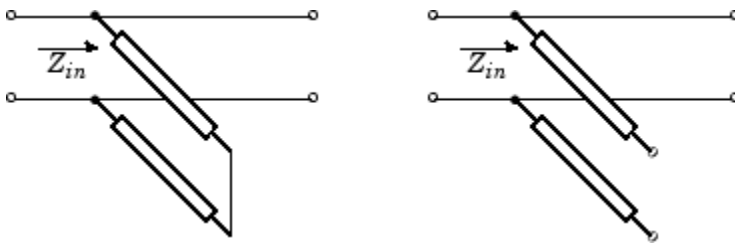
$$B = \frac{Z_0 * (e^{kd} - e^{-kd})}{2}$$

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

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

Parameter Calculations for Transmission Line with Stub

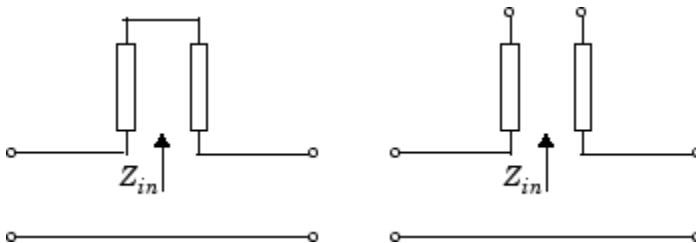
When you set the **Stub mode** parameter in the mask dialog box to **Shunt**, the two-port network consists of a transmission line in series with a stub. You can terminate the stub with a short circuit or an open circuit as shown in the following figure.



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

$$\begin{aligned} A &= 1 \\ B &= 0 \\ C &= 1/Z_{in} \\ D &= 1 \end{aligned}$$

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:

$$\begin{aligned} A &= 1 \\ B &= Z_{in} \\ C &= 0 \\ D &= 1 \end{aligned}$$

Tips

- 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.

Version History

Introduced in R2012a

R2022a: Three block parameters renamed

Behavior changed in R2022a

The **Substrate height**, **Relative permittivity constant**, and **Relative permeability constant** parameters in the Transmission Line block have been renamed to **Dielectric thickness**, **Relative permittivity of dielectric**, and **Relative permeability of dielectric**, respectively.

When you open a model created before R2022a containing the Transmission Line block, the software replaces the old parameters names with the new ones.

R2022a: Model transmission lines

Use the Transmission Line block to model:

- Embedded, inverted, and suspended microstrip transmission lines
- Conductor-backed coplanar waveguide transmission lines
- Stripline transmission lines

References

- [1] Sussman-Fort, S. E., and J. C. Hantgan. "SPICE Implementation of Lossy Transmission Line and Schottky Diode Models." *IEEE Transactions on Microwave Theory and Techniques*. Vol. 36, No.1, January 1988.
- [2] Pozar, David M. *Microwave Engineering*. Hoboken, NJ: John Wiley & Sons, Inc., 2005.
- [3] Gupta, K. C., Ramesh Garg, Inder Bahl, and Prakash Bhartia. *Microstrip Lines and Slotlines*, 2nd Edition, Norwood, MA: Artech House, Inc., 1996.
- [4] Ludwig, Reinhold and Pavel Bretchko. *RF Circuit Design: Theory and Applications*. Englewood Cliffs: NJ: Prentice-Hall, 2000.
- [5] True, Kenneth M. "Data Transmission Lines and Their Characteristics." *National Semiconductor Application Note 806*, April 1992.

See Also

Amplifier | S-Parameters

Mutual Inductor

Model two coupled inductors for circuit envelope analysis



Libraries:
RF Blockset / Circuit Envelope / Elements

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, “Simulate High Frequency Components”.

The block implements the relations

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

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

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

where:

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

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:

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

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

Parameters

Inductance L1 — Inductance of first inductor

1e-6 H (default) | positive scalar

Inductance of the first inductor, specified as a positive scalar in H, nH, μ H, mH.

Inductance L2 — Inductance of second inductor

1e-6 H (default) | positive scalar

Inductance of the second inductor, specified as a positive scalar in H, nH, μ H, mH.

Coefficient of coupling — Coefficient of coupling for mutual inductance
0.9 (default) | scalar

Coefficient of coupling for the mutual inductance of the two inductors, specified as a scalar value from 0 through 1.

Version History

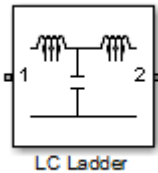
Introduced in R2012a

See Also

Inductor | Three-Winding Transformer

LC Ladder

Model LC ladder networks

**Libraries:**

RF Blockset / Circuit Envelope / Elements

Description

The LC Ladder block models common two-port LC lossless networks. To learn how you can use LC Ladder to compare the input and output signal amplitudes to study the signal attenuation, see “Model RF Filter Using Circuit Envelope”.

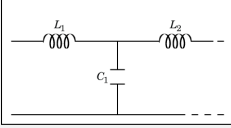
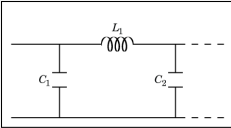
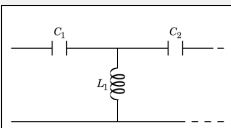
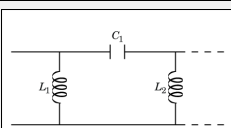
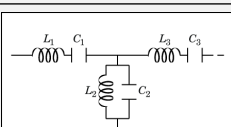
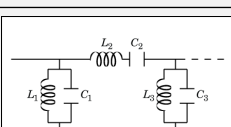
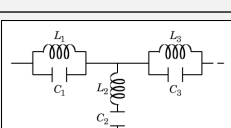
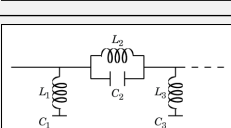
Parameters

Ladder topology — Type of ladder topology

LC Lowpass Tee (default)

Type of ladder topology, specified as one of the following:

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 — inductance values
 [1.3324e-5 1.3324e-5] H (default) | real number

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

Data Types: double

Capacitance — Capacitance value
 1.1327e-9 F (default) | real number

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

Ground and hide negative terminals — Internally ground and hide the negative terminals
on (default) | off

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.

Version History

Introduced in R2015b

See Also

Inductor | Capacitor

Topics

“Model RF Filter Using Circuit Envelope”

“Filter Mixing Products”

Phase Shift

Model phase shift in RF systems



Libraries:

RF Blockset / Circuit Envelope / Elements

Description

The Phase Shift block models an ideal phase shift in the circuit envelope environment. To learn how to use Phase Shift block to measure image rejection ratio in receiver, see “Measuring Image Rejection Ratio in Receivers” example.

Parameters

Phase shift — Phase shift angle

90 deg (default) | non-negative scalar integer

Phase shift angle, specified as a nonnegative scalar integer in degrees. The phase shift is applied to all nonzero simulation frequencies. For zero (DC) frequency, the shift is always zero.

Data Types: double

<varlistentry version="5.0-variant tmwbook5.0" xml:base="../../shared/ground_and_hide.xml">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.

</varlistentry>

Version History

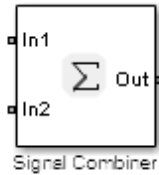
Introduced in R2015b

See Also

Variable Phase Shift

Signal Combiner

Compute sum of RF signals



Libraries:

RF Blockset / Circuit Envelope / Elements

Description

The Signal Combiner block sums signals across each carrier frequency in the circuit envelope simulation environment.

Parameters

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.

Version History

Introduced before R2006a

See Also

“Simulate High Frequency Components”

Three-Winding Transformer

Compute sum of RF signals



Libraries:
RF Blockset / Circuit Envelope / Elements

Description

The Three-Winding Transformer block models three coupled inductors within the circuit envelope simulation environment.

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 p th and q th inductors, with coefficient of coupling K_{pq} .
- $v_1(t)$, $v_2(t)$, and $v_3(t)$ represent the voltage across the terminals of the inductors at time t .
- $i_1(t)$, $i_2(t)$, and $i_3(t)$ represent the current through the inductors at time t . The block uses standard dot notation to indicate the direction of positive current flow relative to a positive voltage.

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:

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

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

Parameters

Inductance L1 — Inductance value of first inductor
1e-6 H (default) | nonnegative scalar integer

Inductance value of the first inductor, specified as a nonnegative scalar integer in henries. Specify the units of the inductance from the corresponding drop-down menu.

Data Types: double

Inductance L2 — Inductance value of second inductor

1e-6 H (default) | nonnegative scalar integer

Inductance value of the second inductor, specified as a nonnegative scalar integer in henries. Specify the units of the inductance from the corresponding drop-down menu.

Data Types: double

Inductance L3 — Inductance value of third inductor

1e-6 H (default) | nonnegative scalar integer

Inductance value of the third inductor, specified as a nonnegative scalar integer in henries. Specify the units of the inductance from the corresponding drop-down menu.

Data Types: double

Inductance L3 — Inductance value of third inductor

1e-6 H (default) | nonnegative scalar integer

Inductance value of the third inductor, specified as a nonnegative scalar integer in henries. Specify the units of the inductance from the corresponding drop-down menu.

Data Types: double

Coefficient of coupling K12 — Coupling coefficient of first and second inductors

0.9 (default) | nonnegative scalar integer

Coupling coefficient for the mutual inductance of the first and second inductors, specified as a nonnegative scalar integer between 0 and 1

Data Types: double

Coefficient of coupling K13 — Coupling coefficient of first and third inductors

0.9 (default) | nonnegative scalar integer

Coupling coefficient for the mutual inductance of the first and third inductors, specified as a nonnegative scalar integer between 0 and 1

Data Types: double

Coefficient of coupling K23 — Coupling coefficient of second and third inductors

0.9 (default) | nonnegative scalar integer

Coupling coefficient for the mutual inductance of the second and third inductors, specified as a nonnegative scalar integer between 0 and 1

Data Types: double

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.

Version History

Introduced before R2006a

See Also

Inductor | Mutual Inductor

Z (Impedance)

Model complex impedance



Libraries:

RF Blockset / Circuit Envelope / 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 — Frequency dependency

Frequency independent (default) | Frequency dependent

Frequency dependency for the impedance, specified as Frequency independent or Frequency dependent.

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

Data Types: double

Complex impedance (Ohm) — Impedance value

50 (default) | scalar complex number | vector of complex numbers

Impedance value, specified as a scalar complex number in ohms or a vector for complex numbers with each element unit in ohms.

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, $[Z_1, Z_2, Z_3, \dots]$. This vector is linearly interpolated for all simulation frequencies.

Data Types: double

Frequency — Frequency values for frequency dependent impedance

1e9 (default) | nonnegative scalar number | nonnegative vector numbers

Frequency values for frequency dependent impedance, specified as a nonnegative scalar number in Hz or nonnegative vector number with each element unit in Hz.

When **Impedance type** is set to **Frequency dependent**, you can specify a vector of nonnegative frequencies, $[f_1, f_2, f_3, \dots]$. 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.

Data Types: double

Version History

Introduced in R2010b

R2021b: Impedance block icon updated

Behavior changed in R2021b

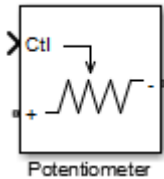
Starting in R2021b, the Impedance block icon has updated. When you open a model created before R2021b containing an Impedance block, the software replaces the block icon with the R2021b version.

See Also

Inductor | Capacitor | Resistor

Potentiometer

Simulink controlled potentiometer



Libraries:

RF Blockset / Circuit Envelope / 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 voltage-current relationships when the control signal varies between zero and one are:

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

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

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

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

Parameters

Minimum resistance — Minimum input resistance of potentiometer

10 0hm (default) | positive scalar

Minimum input resistance of the potentiometer, specified as a positive scalar in ohms.

Data Types: double

Maximum resistance — Maximum input resistance of potentiometer

1e6 0hm (default) | positive scalar

Maximum input resistance of potentiometer, specified as a positive scalar in ohms. This value must be greater than the minimum resistance.

Data Types: double

Potentiometer type — Curve type of potentiometer

Linear (default) | Logarithmic | Antilog

Curve type of the potentiometer, specified as Linear, Logarithmic, Antilog.

Data Types: char

Percentage of resistance at half wiper — Maximum input resistance of potentiometer positive scalar

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

- Logarithmic - Greater than 0 and less than 50. The default is 20.
- Antilog - Greater than 50 and less than 100. The default is 80.

Dependencies

To enable this parameter, select Logarithmic or Antilog in **Potentiometer type**.

Data Types: double

Version History

Introduced in R2015b

See Also

SPST | Switch | SPDT

Antenna

Model antenna or antenna array accounting for incident power wave (RX) and radiated power wave (TX)



Libraries:
RF Blockset / Circuit Envelope / Elements

Description

Model an antenna or antenna array using the Antenna block. Use this block to:

- Convert a Simulink input of an incident power wave vector into RF Blockset voltage signal at the antenna or antenna array ports.
- Convert current at RF Blockset antenna or antenna array ports to a Simulink output of a radiated power wave vector.
- Introduce antenna impedance into an RF system.

By default, the antenna block is an isotropic radiator producing a Simulink output signal. For an isotropic radiator, specify the gain and impedance of the antenna in the block parameters. The **Radiated carrier frequencies** parameter is a set of carrier frequencies over-which the Antenna block creates the radiated power wave. For more information, see “Radiated Wave” on page 1-273 and “Incident Wave” on page 1-275.

The Antenna block mask icons are dynamic: The icons show the current state of the noise parameter, indicate if you have specified an **antennaarray** object in the **Antenna object** parameter, and update the number of ports based on the number of elements in the **antennaarray** object. This table shows how the block icons vary based on the state of the **Simulate noise** and **Antenna object** parameters.

Antenna object	Simulate noise: on	Simulate noise: off
Antenna Catalog Objects		
Antenna Array Catalog Objects		

Ports

Input

RX — Received signal

scalar | vector | matrix | m -by- n -by- k array

Received signal, specified as a scalar, vector, matrix, or an array of size m -by- n -by- k . In the array:

- m — Represents the frame size

- n — Represents the number of interferers
- k — Represents horizontal or vertical polarization. The value of the k must be 2.

Data Types: double

Output

TX — Transmitted signal

scalar | vector | matrix | m -by- n -by- k array

Transmitted signal, specified as a scalar, vector, matrix, or an array of size m -by- n -by- k . IN the array:

- m — Represents the frame size
- n — Represents the number of interferers
- k — Represents horizontal or vertical polarization. The value of the k must be 2.

Data Types: double

Parameters

Main

Source of antenna model — Antenna model

Isotropic radiator (default) | Antenna Designer | Antenna object

Antenna model, specified as one of the following:

- Isotropic radiator
- Antenna Designer
- Antenna object

Note To use Antenna Designer and Antenna object options you will need Antenna Toolbox™.

Create Antenna — Open Antenna Toolbox **Antenna Designer** app button

Open the **Antenna Designer** app from the Antenna Toolbox to create an antenna.

Dependencies

To enable this parameter, set the **Source of antenna model** to Antenna Designer.

Antenna object — Antenna or array element input from workspace

antenna catalog object | antenna array catalog object | pcbstack object

Antenna or array element input from the workspace, specified as a single port antenna element or up to 65 port antenna array elements, or `pcbStack` object created using the Antenna Toolbox. Analyze the antenna, array, or PCB stack objects in the workspace for at least one frequency before using them in the block.

For a complete list of the type of antenna and array objects that you can use in this block, see “Antenna Catalog” (Antenna Toolbox) and “Array Catalog” (Antenna Toolbox).

Dependencies

To enable this parameter, set **Source of antenna model** to `Antenna` object.

Antenna Gain — Antenna gain

0 dBi (default) | real scalar or vector | positive scalar or vector

Antenna gain, specified as real scalar or vector if you set units to `dBi` or positive scalar or vector if you set units are `None`. If the antenna gain is a vector, the vector length must be equal to the vector length of **Incident carrier frequencies** and **Radiated carrier frequencies**.

Dependencies

To enable this parameter, set **Source of antenna model** to `Isotropic radiator` and check **Input incident wave** or **Output radiated wave** or both

Impedance (Ohm) — Input impedance

50 (default) | complex-valued scalar or vector

Input impedance, specified as a complex-valued scalar or vector in ohms. If the impedance is a vector, the vector length must be equal to the length of **Incident carrier frequencies** and **Radiated carrier frequencies**.

Dependencies

To enable this parameter, set **Source of antenna model** to `Isotropic radiator`.

Data Types: `double`

Complex Number Support: `Yes`

Input incident wave — Input incident wave for simulating receiving antenna

'off' (default) | 'on'

Select this parameter if you want to simulate a receiving antenna.

Output radiated wave — Output radiated wave for transmitting antenna

'on' (default) | 'off'

Select this parameter if you want a simulate a transmitting antenna.

Incident carrier frequencies — Carrier frequencies for receiving signal

2.1 GHz (default) | nonnegative scalar or row vector

Carrier frequencies for a receiving signal, specified as a nonnegative scalar in hertz or a row vector with each element unit in hertz. If the value of **Antenna gain** or **Impedance** is a vector, then the values of **Incident carrier frequencies** and **Radiated carrier frequencies** must be identical.

Dependencies

To enable this parameter, select **Input incident wave**.

Radiated carrier frequencies — Carrier frequencies for transmitting signal

2.1 GHz (default) | nonnegative scalar or row vector

Carrier frequencies for a transmitting signal, specified as a nonnegative scalar in hertz or a row vector with each element unit in hertz. If the value of **Antenna gain** or **Impedance** is a vector, then the values of **Incident carrier frequencies** and **Radiated carrier frequencies** must be identical.

Dependencies

To enable this parameter, select **Output radiated wave**.

Direction of departure — Azimuth and elevation angles towards which output signal power wave radiates

[0 0] deg (default) | finite real row vector of length 1-by-2 | finite real matrix of length m -by-2

Azimuth and elevation angles towards which the output signal power wave radiates, specified as a finite real row vector of length 1-by-2 or finite real matrix of length m -by-2 with each element unit in degrees or radians. m is an integer between in the range [2, 100].

Dependencies

To enable this parameter, set **Source of antenna model** to Antenna Designer or Antenna object and select **Output radiated wave**.

Direction of arrival — Azimuth and elevation angles from which input signal power wave arrives

[180 0] deg (default) | finite real row vector of length 1-by-2 | finite real matrix of length m -by-2

Azimuth and elevation angles towards which the input signal power wave arrives, specified as a finite real row vector of length 1-by-2 or finite real matrix of length m -by-2 with each element unit in degrees or radians. m is an integer between in the range [2, 100].

Dependencies

To enable this parameter, set **Source of antenna model** to Antenna Designer or Antenna object and select **Input incident wave**.

Simulate noise — Simulate thermal noise

'on' (default) | 'off'

Select this parameter to simulate thermal noise in the antenna due to the real part of the impedance see at the antenna terminals. You must select **Simulate noise** in the Configuration block also.

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.

Modeling

Modeling options — Model frequency-dependent antenna parameters

Time domain (rationalfit) (default) | Frequency domain

Model frequency-dependent antenna parameters, specified as:

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

The frequency-dependent parameters are:

- **Antenna impedance** — The input impedance at the antenna terminals. This is used in RF system simulation.
- **Normalized vector effective length** — A property used that ties between the current flowing at the antenna terminals and the radiated far-field at a given direction. Due to reciprocity, the effective length also ties between the incident field and the induced open-circuit voltage on the antenna terminals.

Dependencies

To set source **Source of antenna model** of Antenna Designer or Antenna object to activate the **Modeling** Tab that contains the **Modeling options** parameters.

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

Relative error acceptable for the rational fit, specified as a scalar. Applies to time domain modeling of both antenna impedance and normalized vector effective length. The corresponding rational fitting results for each property are displayed on the block mask.

Dependencies

To set **Modeling options** to Time domain (rationalfit) in .

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

Select this parameter to automatically calculate the impulse response duration. Clear this parameter to manually specify the impulse response duration using **Impulse response duration**. Applies to frequency-domain modeling of both antenna impedance and normalized vector effective length.

Dependencies

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

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

Impulse response duration, specified as a scalar. Applies to frequency-domain modeling of both antenna impedance and normalized vector effective length.

Dependencies

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

More About

Radiated Wave

The antenna block produces a Simulink signal representing a normalized power wave similar to power waves in circuits. Since an antenna radiates two independent field components in the far field, the signal is extended into the third dimension:

$$TX(:, :, 1) = TX_\theta = \frac{E_\theta}{\sqrt{\eta_0}} \cdot \sqrt{4\pi} \cdot R \cdot e^{\gamma R}$$

$$TX(:, :, 2) = TX_\phi = \frac{E_\phi}{\sqrt{\eta_0}} \cdot \sqrt{4\pi} \cdot R \cdot e^{\gamma R}$$

where

- E_θ and E_ϕ are the electric field components radiated from the antenna and measured at a far-field location in the direction of departure.
- η_0 is the free-space intrinsic impedance
- R is the distance to the far-field measurement location.
- $\gamma = \frac{j\omega}{c}$ where ω is the angular frequency and c is the speed of light in free space.

The above definition makes the transmit (TX) signal independent of the distance R . The total power carried by this normalized radiated power wave is the equivalent isotropically radiated power wave (EIRP) of the transmitter in the direction of departure:

$$\|TX\|^2 = |TX_\theta|^2 + |TX_\phi|^2 = EIRP = P_t G_t$$

where total

- P_t is the input power at the antenna terminals
- G_t is transmitter antenna gain at the direction of departure.

The EIRP is a commonly used concept in communication systems. This value represents the amount of power radiated from an isotropic antenna such that the same power density is obtained in the direction of departure.

In case of an isotropic radiator, you resolve the ambiguity in polarization by assuming that the antenna is radiating a single field component. You also assume that this field component is always aligned for full reception by a receiving antenna. Thus, the TX signal and the expected RX in a receiving antenna is two dimensional. Following the above definitions, the transmit signal for an isotropic radiator is:

$$TX(:, :) = \sqrt{G_t R_e\{Z_{in}\}} \cdot I_{in}$$

where:

- Z_{in} is the input impedance of the antenna.
- I_{in} is the current at antenna terminals.

In all definitions of TX, the array elements are arranged in the first two dimensions in a manner similar to the of the output signal of an RF Blockset Outport block. If the signal is framed, the column size corresponds to the number of frame bits and the row size corresponds to the number of carrier frequencies. If the signal is not framed, then the column size corresponds to the number of carrier frequencies and the row size is one.

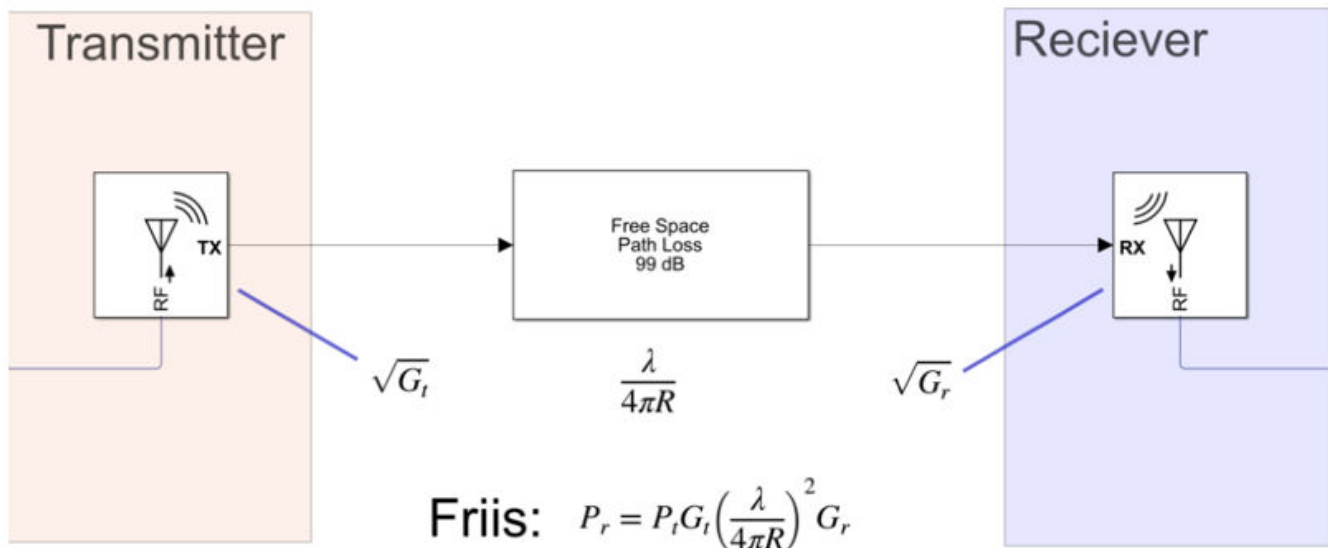
Effect of Free Space Channel

The effect of free space channel between the antennas is not captured by antenna block. You can model it externally using Simulink blocks. For a free-space channel the effect is given by the transfer function:

$$pl = \frac{\lambda}{4\pi R} \cdot e^{-\gamma R}$$

where

- λ is the wavelength modeled outside the antenna.
- R is the distance between the antennas
- Exponential term at the end of the equation represents the time delay occurring over the distance R .
- pl is free-space path loss.



You can model a free-space channel using the Communications Toolbox Free Space Path Loss. The effect of the power wave is described using the Friis equation. The Free Space Path Loss block operates for a single carrier frequency and is narrow band. For multiple carriers with narrow bands, the signal must be split and passed through multiple Free Space Path Loss blocks, with carrier frequencies specified in the Antenna block. When the Antenna blocks are not isotropic radiators, the output signal is a 3D array and needs to be split and reshaped before being sent to the Free Space Path Loss

Incident Wave

The antenna block can also accept a Simulink signal representing a normalized incident power wave. Since an antenna also receives two independent field components, the signal is extended in the third dimension:

$$RX(:, :, 1) = RX_\theta = TX_\theta \cdot pl = \frac{E_\theta}{\sqrt{\eta_0}} \cdot \frac{\lambda}{\sqrt{4\pi}}$$

$$RX(:, :, 2) = RX_\phi = TX_\phi \cdot pl = \frac{E_\phi}{\sqrt{\eta_0}} \cdot \frac{\lambda}{\sqrt{4\pi}}$$

where

- TX_θ and TX_ϕ are signals from transmitting antenna.
- pl free-space channel transfer function.
- E_θ and E_ϕ are the electrical field components measured from transmitting antenna.
- η_0 is the free-space intrinsic impedance.
- λ is the wavelength.
- RX is the incident power wave normalized such that is the power received by the isotropic antenna.

Using the above equations, the total power carried by the normalized incident power wave, $\|RX\|^2 = |RX_\theta|^2 + |RX_\phi|^2$ is available power received by ideal isotropic receiving antenna. The available power received by a true antenna is:

$$P_r = \|RX\|^2 G_r$$

where G_r is the receiver antenna gain at the direction of arrival.

In case the receiving antenna is an isotropic radiator, you can resolve the ambiguity in polarization by assuming that the antenna is receiving a single field component that it is aligned for full reception. Thus, the RX signal is expected to be two dimensional. In all definitions of the RX signal, the array elements are arranged in the first two dimensions in a manner similar to that of the input signal of an RF Blockset Inport block.

Version History

Introduced in R2020b

R2021b: Antenna block icon updated

Behavior changed in R2021b

The Antenna block icon was updated in R2021b. The block icon is now dynamic and shows the current state of the noise parameter.

When you open a model created before R2021b containing an Antenna block, the software replaces the block icon with the R2021b version.

R2022a: Model transmit and receive antenna array

To model a transmit and receive antenna, specify an antenna array from the “Array Catalog” (Antenna Toolbox) in the **Antenna object** parameter.

R2023a: Multiple directions of arrival and departure in Antenna block

You can now use the Antenna block to transmit and receive signals in multiple directions using the **Direction of arrival** and **Direction of departure** properties.

References

- [1] Stutzman, Warren L., and Gary A. Thiele. *Antenna Theory and Design*. 3rd ed. Hoboken, NJ: Wiley, 2013
- [2] Farr, Everett G. "Characterizing Antennas in the Time and Frequency Domains [Education Corner]." *IEEE Antennas and Propagation Magazine* 60, no. 1 (February 2018): 106-10. <https://doi.org/10.1109/MAP.2017.2774200>.

See Also

Amplifier | S-Parameters | Free Space Path Loss

Connection Label

Virtual connection between circuit envelope ports

**Libraries:**

RF Blockset / Circuit Envelope / Utilities
Simscape / Utilities


Description

The Connection Label block lets you specify virtual connections between circuit envelope ports, similar to the Goto and From blocks in Simulink diagrams. These connection lines are nondirectional, which means that all sides of the virtual connection use the same type of block.

Virtual connectivity is established by the label name. If two or more Connection Label blocks in a model subsystem have the same label, then the circuit envelope ports of other blocks connected to them behave as though they were physically connected. Virtual connections cannot cross subsystem boundaries. The label name is displayed next to the block icon on the model canvas.

You specify the label name by using the **Label** parameter.

Use the Connection Label block to reduce diagram clutter by breaking off tangled connection lines.

You can select a Connection Label block to highlight blocks related to it. To show a related block in an open diagram or new tab, pause on the ellipsis. Then, select **Related Blocks**  from the action bar. When multiple blocks correspond to the selected block, a list of related blocks opens. You can filter the list of related blocks by entering a search term in the text box. After you select a related block from the list, window focus changes to the open diagram or new tab that shows the related block.

Ports

Input

Port_1 — Connection port

untyped circuit envelope port

Circuit envelope connection port. By default, this port is untyped.

You define the port type by connecting this port to a circuit envelope port of another block in the subsystem. Once you establish a connection, other Connection Label blocks that are in the same subsystem and have the same label can be connected only to the same type of port.

Parameters

Label — Name of connection label

Label1 (default) | character vector | string

Label name, which establishes the virtual connection. If two or more Connection Label blocks in the same subsystem of a model have the same label, then the circuit envelope ports of the blocks connected to them behave as though they are physically connected.

The value of this parameter appears as a label next to the block icon on the model canvas.

Version History

Introduced in R2022a

See Also

Configuration | Inport | Outport

Equivalent Baseband Blocks

Connection Port



Connection port for RF subsystem




Libraries:

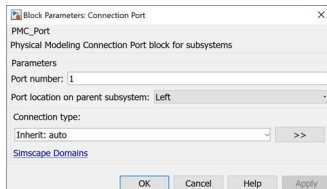
RF Blockset / Equivalent Baseband / Input
RF Blockset / Circuit Envelope / Utilities

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.

Parameters



PMC_Port: Physical Modelling Connection Port block for subsystems

Port number — Connector port number
1 (default) | scalar

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.

Port location on parent subsystem — Port location on the parent subsystem. Choices are:
left (default) | Right

Use this parameter to choose on which side of the parent subsystem boundary the Port is placed.

Version History

Introduced in R2007a

R2021b: Connection Port block icon updated
Behavior changed in R2021b

Starting in R2021b, the Connection Port block icon has updated. When you open a model created before R2021b containing a Connection Port block, the software replaces the block icon with the R2021b version.

Extended Capabilities

C/C++ Code Generation

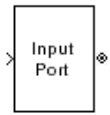
Generate C and C++ code using Simulink® Coder™.

See Also

Creating Subsystems

Input Port

Connection block from Simulink environment to RF physical blocks



Libraries:

RF Blockset / Equivalent Baseband / Input / Output Ports

Description

The Input Port block is a connecting port from the Simulink, or mathematical, part of the model to an RF physical part of the model. The Input Port block lets you provide the parameter data needed to calculate the modeling frequencies and the baseband-equivalent impulse response for the physical subsystem. It also lets you specify information about how to interpret the incoming Simulink signal.

For more information about how the Input Port block converts the mathematical Simulink signals to physical modeling environment signals, see “Convert to and from Simulink Signals”.

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

Parameters

Treat input Simulink signal as — Options for interpreting the input

Source voltage (default) | Incident power wave | scalar

Options for interpreting the input Simulink signal, specified as one of the following:

- **Incident power wave** — Interpret the input signal as the incident power wave of the RF system described in the physical model to which it connects. When you select this option, the output signal of the RF system is the transmitted power wave. This is the most common RF modeling interpretation.
- **Source voltage** — Interpret the input signal as the source voltage of the RF system described in the physical model to which it connects. As a result, the baseband-equivalent model includes the loss through the source impedance. When you select this option, the output signal of the RF system is the load voltage.

For more information about these options, see “Convert to and from Simulink Signals”.

Source impedance (ohms) — Source impedance of the RF network

50 (default) | scalar

Source impedance of the RF network described in the physical model to which it connects, specified as a scalar in ohms.

Finite impulse response filter length — Desired length of the baseband-equivalent impulse response

128 (default) | scalar

Desired length of the baseband-equivalent impulse response for the physical model, specified as a scalar. 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 $1/(\text{Finite impulse response filter length} \times \text{Sample time (s)})$. For a graphical representation of this parameter, see “Baseband-Equivalent Modeling”.

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

Fractional bandwidth of guard bands — Fraction of modeling bandwidth

0 (default) | scalar

Fraction of modeling bandwidth, specified as a scalar. 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.

Modeling delay (samples) — Number of time samples by which to delay the impulse response of baseband-equivalent model

0 (default) | scalar

Number of time samples by which to delay the impulse response of the baseband-equivalent model, specified as a scalar. Setting **Modeling delay (samples)** ensure that the baseband-equivalent model has a causal response.

See “Create Complex Baseband-Equivalent Model” for information on how the Input Port block applies this delay to ensure a causal response.

Center frequency (Hz) — Center of modeling frequencies

2e9 (default) | scalar

Center of the modeling frequencies, specified as a scalar. See the Output Port block reference page for information about calculating the modeling frequencies.

Sample time (s) — Time interval between consecutive samples

1e-7 (default) | scalar

Time interval between consecutive samples of the input signal, specified as a scalar.

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.

Input Processing — Specify type of processing mode

Elements as channels (sample based) (default) | Columns as channels (frame based)

Specifying the type of processing mode the block performs as one of the following:

- **Elements as channels (sample based)**: Set the **Input processing** to **Elements as channels (sample based)**, if your block needs to process based on sample-based model. In sample-based processing, blocks process signals one sample at a time. Each element of the input signal represents one sample in a distinct channel.
- **Columns as channels (frame based)**: Set the **Input processing** to **Columns as channels (frame based)**, if your block needs to process based on frame-based model. In frame-based processing, blocks process data one frame at a time. Each frame of data contains sequential samples from an independent channel. Each channel is represented by a column of the input signal.

Add noise — Add noise data to block

on (default) | off

Add noise data to block, specified as a **on** or **off**. 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 — Initial seed for random number generator block

67987 (default) | nonnegative integer

Initial seed for random number generator block to generate noise, specified as a nonnegative integer 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.

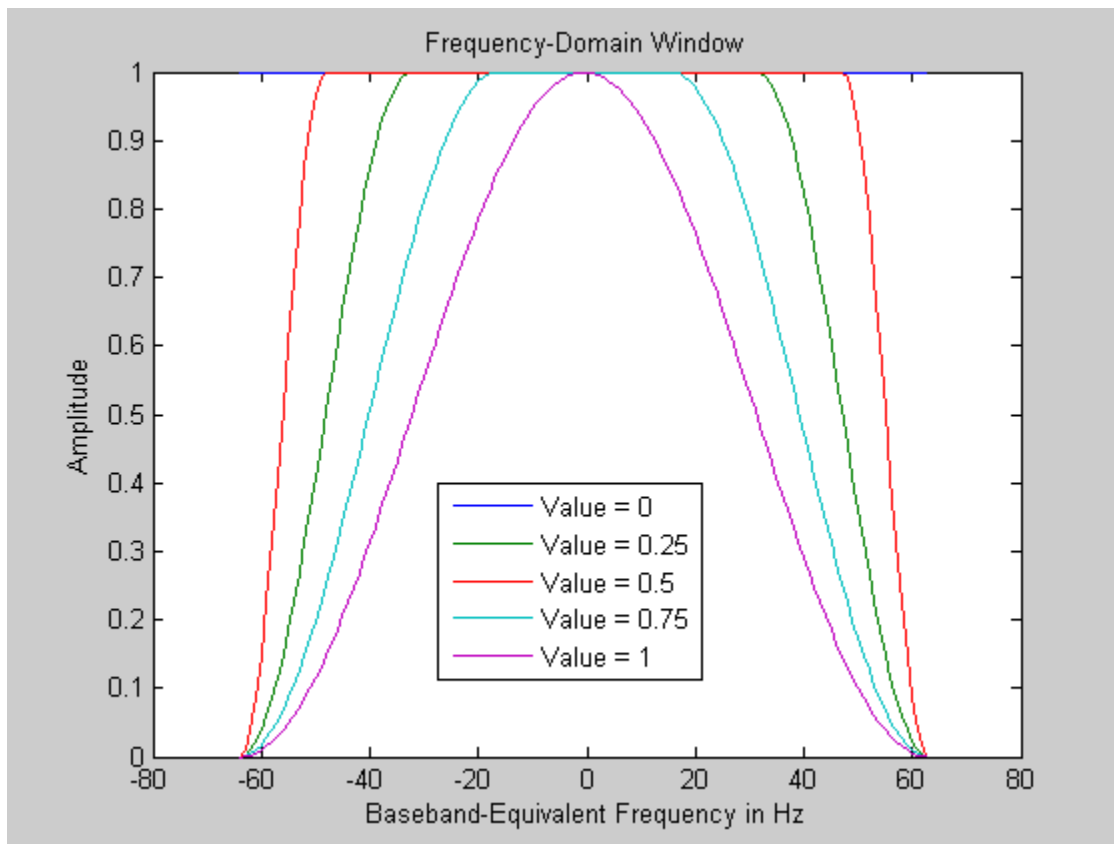
Dependencies

To enable this parameter, select **Add noise** parameter.

More About

Fractional Bandwidth of Guard Bands

The blockset uses the Signal Processing Toolbox™ `tukeywin` function to generate the window. The following figure shows the resulting frequency-domain window for several values of the **Fractional bandwidth of guard bands** parameter.



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

Version History

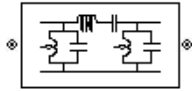
Introduced before R2006a

See Also

Output Port

LC Bandpass Pi

Model LC bandpass pi network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

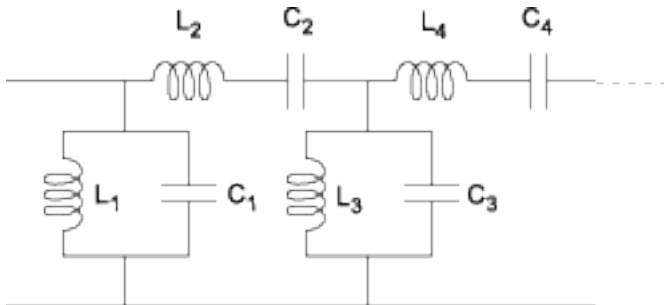
The LC Bandpass Pi block models the LC bandpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



$[L_1, L_2, L_3, L_4, \dots]$ is the value of the 'L' property, and $[C_1, C_2, C_3, C_4, \dots]$ is the value of the 'C' property.

Parameters

Main

Inductance (H) — Inductance of the LC Bandpass Pi
 $[1.4446e-9 \ 4.3949e-8 \ 1.4446e-9]$ (default) | vector

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) — Capacitance of the LC Bandpass Pi
[3.5785e-11 1.1762e-12 3.5785e-11] (default) | vector

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

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[500e6:4e6:900e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None

Y parameter2	Y format2
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced in R2013a

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

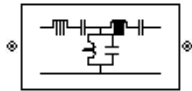
General Passive Network | LC Bandpass Tee | LC Bandstop Pi | LC Bandstop Tee | LC Highpass Pi | LC Highpass Tee | LC Lowpass Pi | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Bandpass Tee

Model LC bandpass tee network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

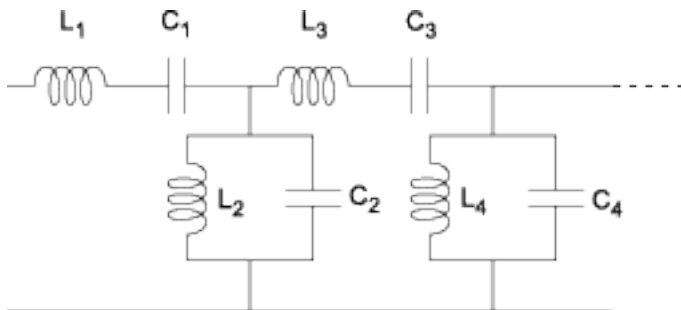
The LC Bandpass Tee block models the LC bandpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



$[L_1, L_2, L_3, L_4, \dots]$ is the value of the 'L' property, and $[C_1, C_2, C_3, C_4, \dots]$ is the value of the 'C' property.

Parameters

Main

Inductance (H) — Inductance of the LC Bandpass Tee
 $[2.7812e-8 \ 3.013e-9 \ 2.7812e-8]$ (default) | vector

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) — Capacitance of the LC Bandpass Tee

[1.8587e-12 1.7157e-11 1.8587e-12] (default) | vector

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

Source of frequency data — Frequency data source

User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range

[500e6:4e6:900e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

DependenciesTo enable **Y format1**, set **Plot type** to X-Y plane.**Y format2** — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None

Y parameter2	Y format2
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced in R2013a

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

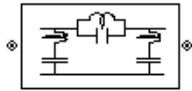
General Passive Network | LC Bandstop Pi | LC Bandstop Tee | LC Highpass Pi | LC Highpass Tee | LC Lowpass Pi | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Bandstop Pi

Model LC bandstop pi network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

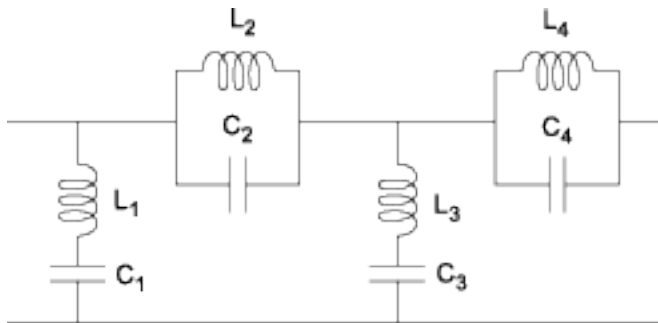
The LC Bandstop Pi block models the LC bandstop pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



$[L_1, L_2, L_3, L_4, \dots]$ is the value of the 'L' property, and $[C_1, C_2, C_3, C_4, \dots]$ is the value of the 'C' property.

Parameters

Main

Inductance (H) — Inductance of the LC Bandstop Pi
 $[2.8091e-8 \ 2.2603e-9 \ 2.8091e-8]$ (default) | vector

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) — Capacitance of the LC Bandstop Pi
[1.8403e-12 2.2871e-11 1.8403e-12] (default) | vector

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

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[500e6:4e6:900e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None

Y parameter2	Y format2
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced in R2013a

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

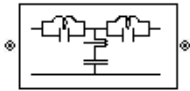
General Passive Network | LC Bandpass Tee | LC Bandstop Tee | LC Highpass Pi | LC Highpass Tee | LC Lowpass Pi | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Bandstop Tee

Model LC bandstop tee network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

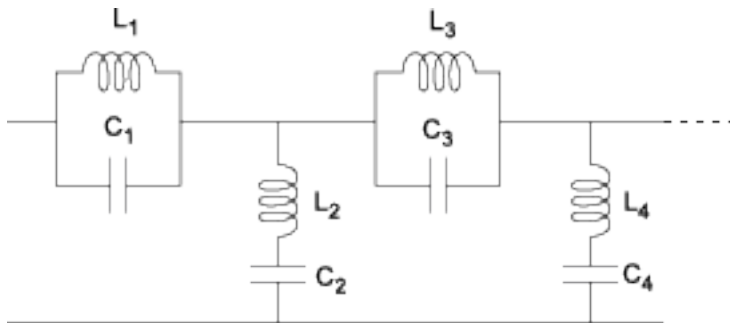
The LC Bandstop Tee block models the LC bandstop tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



$[L_1, L_2, L_3, L_4, \dots]$ is the value of the 'L' property, and $[C_1, C_2, C_3, C_4, \dots]$ is the value of the 'C' property.

Parameters

Main

Inductance (H) — Inductance of the LC Bandstop Tee

`[2.7908e-9 4.9321e-8 2.7908e-9]` (default) | vector

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) — Capacitance of the LC Bandstop Tee

[1.8523e-11 1.0481e-12 1.8523e-11] (default) | vector

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

Source of frequency data — Frequency data source

User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range

[500e6:4e6:900e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

DependenciesTo enable **Y format1**, set **Plot type** to X-Y plane.**Y format2** — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None

Y parameter2	Y format2
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced in R2013a

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

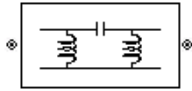
General Passive Network | LC Bandpass Tee | LC Bandstop Pi | LC Highpass Pi | LC Highpass Tee | LC Lowpass Pi | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Highpass Pi

Model LC highpass pi network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

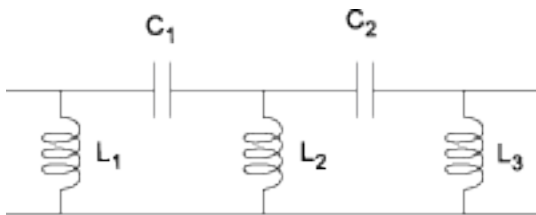
The LC Highpass Pi block models the LC highpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



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

Parameters

Main

Inductance (H) — Inductance of the LC Highpass Pi
 $[1.1881e-6 \ 1.1881e-6]$ (default) | vector

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) — Capacitance of the LC Highpass Pi
 $2.2363e-9$ (default) | vector

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

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced in R2013a

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

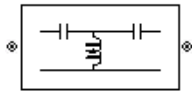
General Passive Network | LC Bandpass Pi | LC Bandpass Tee | LC Bandstop Pi | LC Bandstop Tee | LC Highpass Tee | LC Lowpass Pi | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Highpass Tee

Model LC highpass tee network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

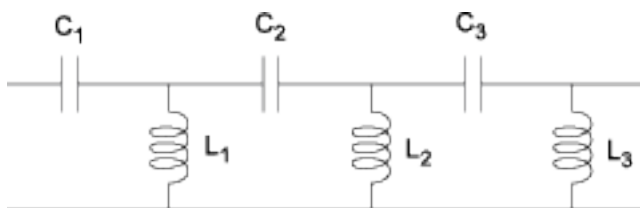
The LC Highpass Tee block models the LC highpass tee network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



$[L_1, L_2, L_3, \dots]$ is the value of the 'L' property, and $[C_1, C_2, C_3, \dots]$ is the value of the 'C' property.

Parameters

Main

Inductance (H) — Inductance of the LC Highpass Tee
 $5.5907e-6$ (default) | vector

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) — Capacitance of the LC Highpass Tee
 $[4.7524e-10 \ 4.7524e-10]$ (default) | vector

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

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History
Introduced in R2013a

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

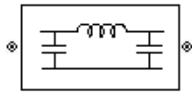
General Passive Network | LC Bandpass Pi | LC Bandpass Tee | LC Bandstop Pi | LC Bandstop Tee | LC Highpass Pi | LC Lowpass Pi | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Lowpass Pi

Model LC lowpass pi network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

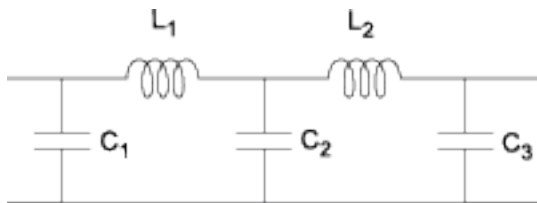
The LC Lowpass Pi block models the LC lowpass pi network described in the block dialog box, in terms of its frequency-dependent S-parameters.

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.



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

Parameters

Main

Inductance (H) — Inductance of the LC Lowpass Pi
 $2.8318e-6$ (default) | vector

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) — Capacitance of the LC Lowpass Pi
 $[5.3296e-9 \ 5.3296e-9]$ (default) | vector

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

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced in R2013a

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

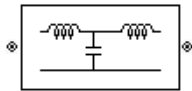
General Passive Network | LC Bandpass Pi | LC Bandpass Tee | LC Bandstop Pi | LC Bandstop Tee | LC Highpass Pi | LC Highpass Tee | LC Lowpass Tee | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

LC Lowpass Tee

Model LC lowpass tee network



Libraries:

RF Blockset / Equivalent Baseband / Ladder Filters

Description

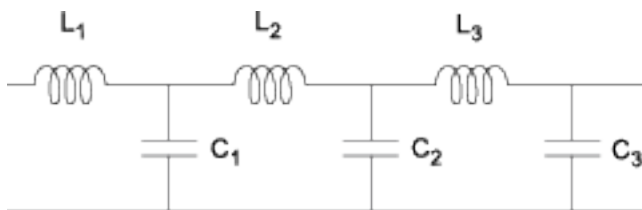
The LC Lowpass Tee block models the LC lowpass tee network described in the block dialog box in terms of its frequency-dependent S-parameters.

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.



$[L_1, L_2, L_3, \dots]$ is the value of the 'L' property, and $[C_1, C_2, C_3, \dots]$ is the value of the 'C' property.

Parameters

Main

Inductance (H) — Inductance of the LC Lowpass Tee
 $[1.3324e-5 \ 1.3324e-5]$ (default) | vector

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) — Capacitance of the LC Lowpass Tee
 $1.1327e-9$ (default) | vector

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

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

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

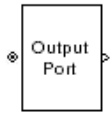
General Passive Network | LC Bandpass Pi | LC Bandpass Tee | LC Bandstop Pi | LC Bandstop Tee | LC Highpass Pi | LC Highpass Tee | LC Lowpass Pi | Series C | Series L | Series R | Series RLC | Shunt C | Shunt L | Shunt R | Shunt RLC

Topics

“Model RF Filter Using Equivalent Baseband”

Output Port

Connection block from RF physical blocks to Simulink environment



Libraries:

RF Blockset / Equivalent Baseband / Input / Output Ports

Description

The Output Port block produces the baseband-equivalent time-domain response of an input signal traveling through a series of RF physical components. The Output Port block

- 1 Partitions the RF physical components into linear and nonlinear subsystems.
- 2 Extracts the complex impulse response of the linear subsystem for baseband-equivalent modeling of the RF linear system.
- 3 Extracts the nonlinear AMAM/AMPM modeling for RF nonlinearity.

The Output Port block also serves as a connecting port from an RF physical part of the model to the Simulink, or mathematical, part of the model. For more information about how the Output Port block converts the physical modeling environment signals to mathematical Simulink signals, see “Convert to and from Simulink Signals”.

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

Parameters

Main Tab

Load impedance(ohms) — Load impedance of RF network

50 (default) | scalar

Load impedance of the RF network described in the physical model to which it connects, specified as scalar in ohms.

Visualization Tab

Source of frequency data — Frequency data source

Derived from Input Port parameter (default) | User-specified

Frequency data source in specified based on one of the following:

- When **Source of frequency data** is Derived from Input Port parameter, frequency data source will be derived from the parameters set on the Input Port.

- When **Source of frequency data** is User-specified, specify as a vector of frequencies in the **Frequency data** parameter.

Frequency data (Hz) — Frequency data range

1e9:1e8:3e9 (default) | vector

Frequency data range, specified as a vector in hertz.

Dependencies

To enable this parameter, set User-specified in **Source of amplifier gain**.

Reference impedance (ohms) — Reference impedance

50 (default) | scalar

Reference impedance of the coaxial transmission line, specified as a scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | 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:

- **X-Y plane** — Generate a Cartesian plot of your data versus frequency. To create linear, semi-log, or log-log plots, set the **Y scale** and **X scale** accordingly.
- **Composite data**—The composite data plot automatically generates four separate plots in one figure window, showing the frequency dependence of several parameters.
- **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 of your data. The block plots only the range of data corresponding to the specified frequencies.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, Gt, GroupDelay, GammaIn, GammaOut, VSWRIn, VSWRout, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, S22, GammaIn, and GammaOut
Z Smith chart	S11, S22, GammaIn, and GammaOut.
Y Smith chart	S11, S22, GammaIn, and GammaOut.
ZY smith chart	S11, S22, GammaIn, and GammaOut.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, Gt, GroupDelay, GammaIn, GammaOut, VSWRIn, VSWRout, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, S22, GammaIn, and GammaOut
Z Smith chart	S11, S22, GammaIn, and GammaOut.
Y Smith chart	S11, S22, GammaIn, and GammaOut.
ZY smith chart	S11, S22, GammaIn, and GammaOut.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, and GammaOut.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn, VSWRout, and Gt.	Magnitude (decibels) and None.
OIP3	dBm, dBW, W, and mW.
NF	Magnitude (decibels).
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, and GammaOut.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn, VSWRout, and Gt.	Magnitude (decibels) and None.
OIP3	dBm, dBW, W, and mW.
NF	Magnitude (decibels).

Y parameter2	Y format2
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — X parameter
Freq (default)

Parameter, specified as Freq. This parameter determines the data for x-axes on the X-Y plane plot.

X format — Plot format
Hz (default) | Auto | KHz | MHz | GHz | THz

Plot format, specified as one of the following Hz, Auto, KHz, MHz, GHz or THz.

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

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

X-axis scale, specified as Linear or Log.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Dependencies The **Visualization** tab shows parameters for creating plots if you display the Output Port mask after you perform one or more of the following actions:

- Run a model with two or more blocks between the Input Port block and the Output Port block.
- Click the Update Diagram button to initialize a model with two or more blocks between the Input Port block and the Output Port block.

For information about plotting, see “Create Plots”.

More About

Linear Subsystem

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

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

The n th element of f , f_n , is given by

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

where

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

- 2 Calculates the passband transfer function for the frequency range as

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

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

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

where

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

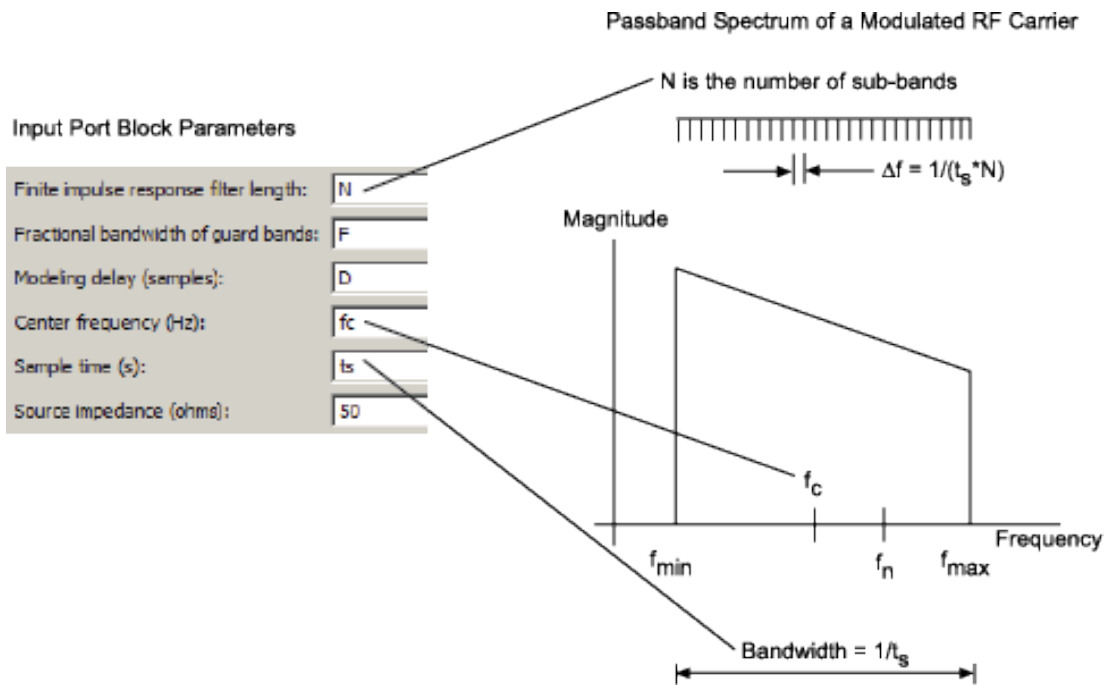
$$\Gamma_s = \frac{Z_s - Z_o}{Z_s + Z_o}$$

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

and

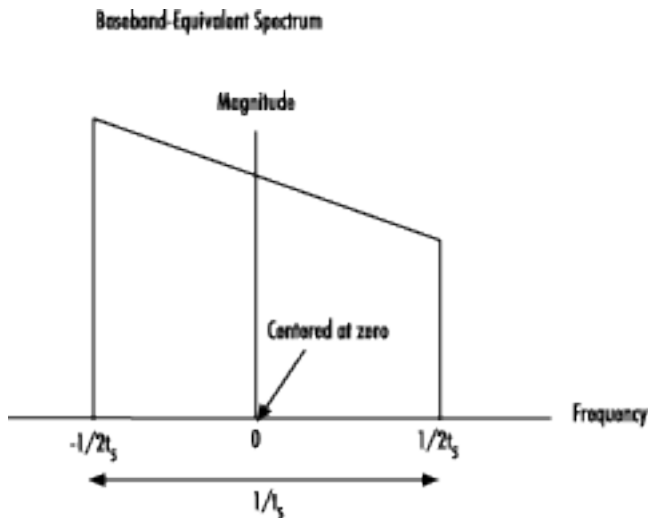
- Z_S is the source impedance.
- Z_L is the load impedance.
- S_{ij} are the S-parameters of a two-port network.

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



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

The baseband transfer function is shown in the following figure.

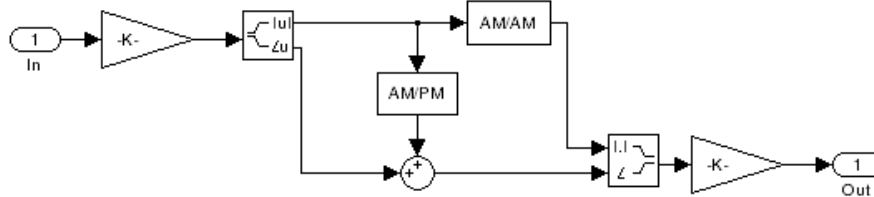


- 4 Obtains the baseband-equivalent impulse response by calculating the inverse FFT of the baseband transfer function. For faster simulation, the block calculates the IFFT using the next power of 2 greater than the specified finite impulse response filter length. Then, it truncates the impulse response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

For the linear subsystem, the Output Port block uses the calculated impulse response as input to the DSP System Toolbox Digital Filter Design block to determine the output.

Nonlinear Subsystem

The nonlinear subsystem is implemented by AM/AM and AM/PM nonlinear models, as shown in the following figure.



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

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

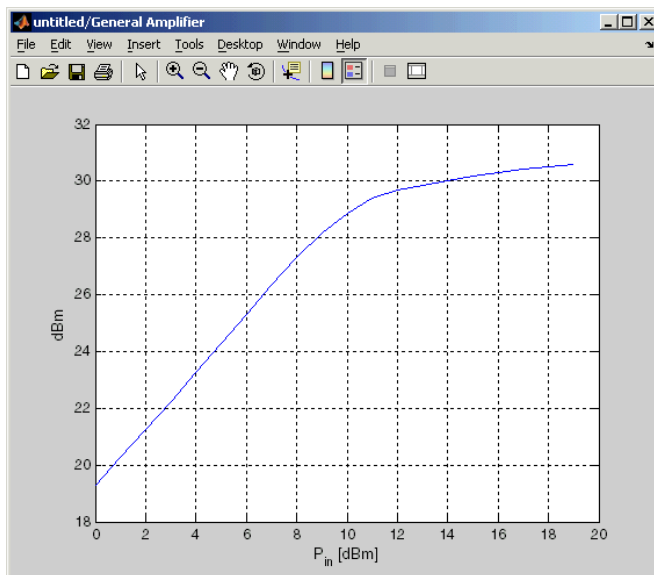
$$PM_{out} = \phi$$

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

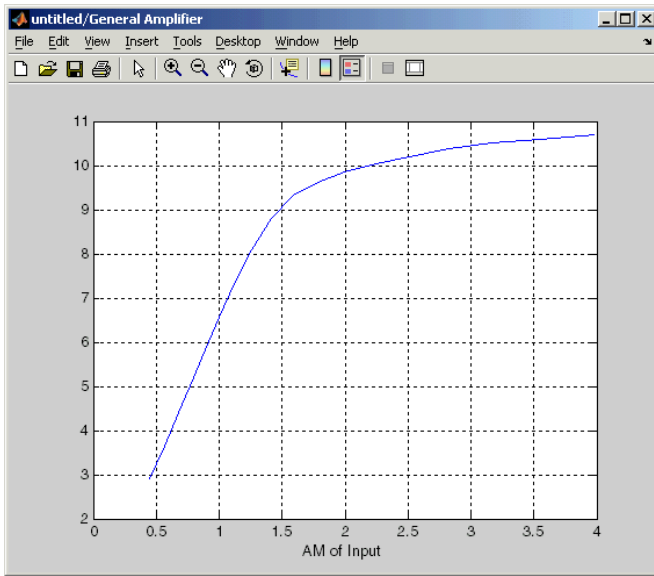
where AM_{in} is the AM of the input voltage, AM_{out} and PM_{out} are the AM and PM of the output voltage, R_s is the source resistance (50 ohms), R_l is the load resistance (50 ohms), P_{in} is the input power, P_{out} is the output power, and ϕ is the phase shift between the input and output voltage.

Note You can provide power data via a .amp file. See “AMP File Data Sections” for information about this format.

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



This figure shows the extracted AM/AM nonlinear conversion.



Version History

Introduced before R2006a

See Also

Input Port

Parallel-Plate Transmission Line

Model parallel-plate transmission line

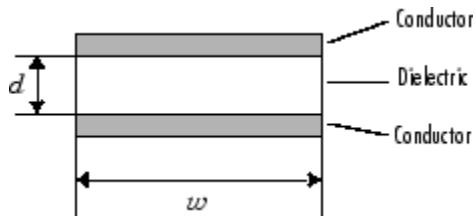


Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

The Parallel-Plate Transmission Line block models the parallel-plate transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A parallel-plate transmission line is shown in cross-section in the following figure. Its physical characteristics include the plate width w and the plate separation d .



Parameters

Main

Plate width (m) — Width of parallel-plate transmission line

5e-3 (default) | scalar

Physical width of the parallel-plate transmission line, specified as a scalar in meters.

Plate separation (m) — Dielectric spacing in parallel-plate transmission line

1e-3 (default) | scalar

Thickness of the dielectric separating the plates, specified as a scalar in meters.

Relative permeability constant — Relative permeability of the dielectric material in the parallel-plate transmission line

1 (default) | scalar

Relative permeability of the dielectric expressed as the ratio of the permeability of the dielectric to permeability in free space μ_0 , specified as a scalar.

Relative permittivity constant — Relative permittivity of the dielectric material in the parallel-plate transmission line

2.3 (default) | scalar

Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space ϵ_0 , specified as a scalar.

Loss tangent of dielectric — Loss tangent of dielectric in the parallel-plate transmission line
 0 (default) | scalar

Loss angle tangent of the dielectric, specified as a scalar.

Conductivity of the conductor (S/m) — Conductivity of conductor
 inf (default) | scalar

Conductivity of the conductor, specified as a scalar in siemens per meter. Conductivity is a metric to measure the flow of current in a conductor.

Transmission line length(m) — Physical length of the transmission line
 0.01 (default) | positive scalar

Physical length of the transmission line in meters, specified as positive scalar in meters.

Stub mode — Type of stub
`Not a stub` (default) | `Shunt` | `Series`

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

Stubless Transmission Line

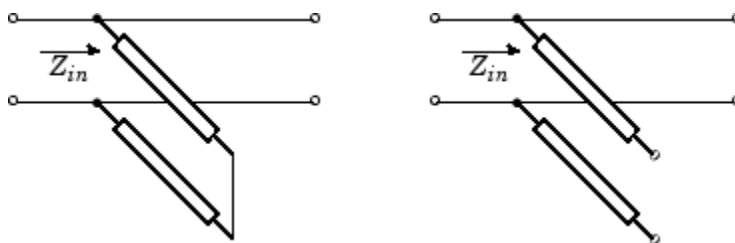
- `Not a stub`—**Not a stub**

If you model a parallel-plate transmission line as stubless line, the Parallel-Plate Transmission Line block first calculates the ABCD-parameters at each frequency contained in the modeling frequencies vector. It then uses the `abcd2s` function to convert the ABCD-parameters to S-parameters. For more information, see “Stub Mode- Not a stub” on page 2-124.

Shunt and Series Stubs

- The stub mode of the parallel-plate transmission line is classified as following:

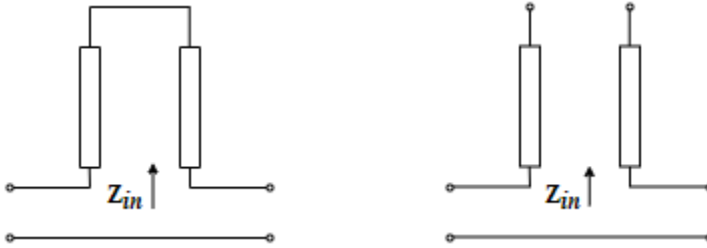
Shunt—**Shunt** parameter provides the users with the two-port network consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown here.



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

$$\begin{aligned}
 A &= 1 \\
 B &= 0 \\
 C &= 1/Z_{in} \\
 D &= 1
 \end{aligned}$$

- **Series—Series** parameter provides the user with 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

$$A = 1$$

$$B = Z_{in}$$

$$C = 0$$

$$D = 1$$

Termination of stub — Stub termination

Open (default)

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt, or Series, or in **Stub mode**

Visualization

Source of frequency data — Frequency data source

User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range

[1e9:1e6:3e9] (default) | vector

Frequency data range, specified as a vector in Hz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.

- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot specified data using plot button.

More About

Stub Mode - Not a Stub

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:

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

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

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

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

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.
- σ_{cond} is the conductivity in the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$ where:
 - μ_0 is the permeability in free space.
 - μ_r is the **Relative permeability constant** parameter value.
- The is a complex dielectric constant given by $\varepsilon = \varepsilon' - j\varepsilon'' = \varepsilon' (1 - j\tan\delta)$
- ε' is the real part of complex dielectric constant ε , $\varepsilon' = \varepsilon_0\varepsilon_r$. ε'' is the imaginary part of complex dielectric constant ε , $\varepsilon'' = \varepsilon_0\varepsilon_r\tan\delta$ where :
 - ε_0 is the permittivity of free space.
 - ε_r is the **Relative permittivity constant** parameter value.
 - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.

Version History

Introduced in R2009a

References

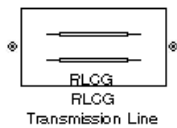
[1] Pozar, David M “Microwave Engineering”, John Wiley & Sons, Inc., 2005.

See Also

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

RLCG Transmission Line

Model RLCG transmission line

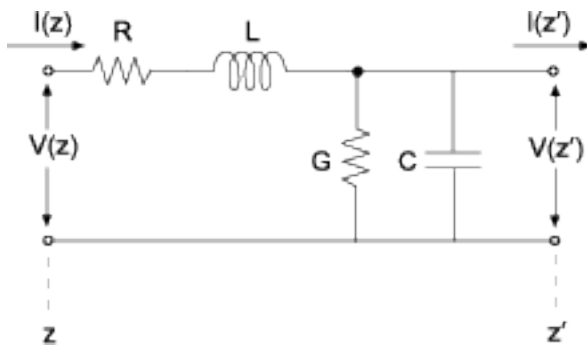


Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

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



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

To learn how you can use RLCG Transmission Line block to create baseband equivalent model, see “Create Complex Baseband-Equivalent Model”.

Parameters

Main

Resistance per length (ohms/m) — Resistance of RLGC transmission lines

0 (default) | scalar

Resistance of the RLGC transmission lines, specified as a scalar in ohms/meters.

Inductance per length (H/m) — Inductance of RLGC transmission lines

0 (default) | scalar

Inductance of the RLGC transmission lines, specified as a scalar in henries/meters.

Capacitance per length (F/m) — Capacitance of RLGC transmission lines

0 (default)

Capacitance of the RLGC transmission lines, specified as a scalar in farad/meters

Conductance per length(S/m) — Conductance of RLGC transmission lines

0 (default) | scalar

Conductance of the RLGC transmission lines, specified as a scalar in siemens per meter.

Frequency (Hz) — Frequency at which RLGC transmission line operate

1e-9 (default)

Vector of frequency values at which the resistance, inductance, capacitance, and conductance values are known.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Transmission line length (m) — Physical length of transmission line

0.01 (default) | scalar

Physical length of the transmission line, specified as a scalar in meters.

Stub mode — Type of stub

Not a stub (default) | Shunt | Series

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

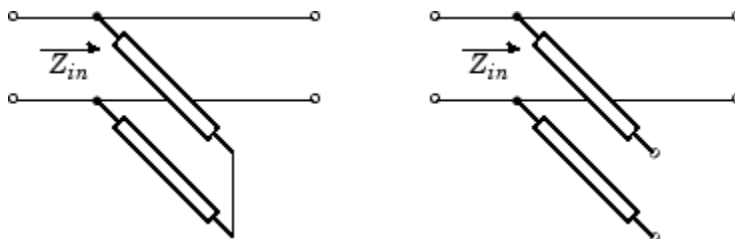
Stubless transmission Line

- Not a stub—**Not a stub**

If you model a coaxial transmission line as 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. For more information, see “Stub Mode - Not a Stub” on page 2-67.

Shunt Transmission Line

- Shunt—This parameter provides a two-port network that consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown in these diagrams.

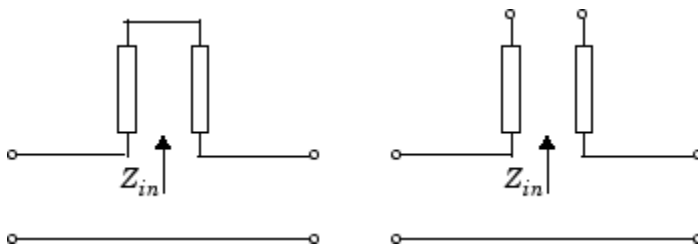


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$\begin{aligned} A &= 1 \\ B &= 0 \\ C &= 1/Z_{in} \\ D &= 1 \end{aligned}$$

Series Transmission Line

- **Series**—This mode parameter provides a two-port network that consists of a series transmission line that you can terminate with either a short circuit or an open circuit as show in these diagrams.



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

$$\begin{aligned} A &= 1 \\ B &= Z_{in} \\ C &= 0 \\ D &= 1 \end{aligned}$$

Termination of stub — Stub termination
Open (default) | Short

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt or Series in **Stub mode**

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

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

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**X scale** — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**Plot** — Plot specified data

button

Plot the specified data using the plot button.

More About**Stub Mode - Not a Stub**

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

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

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

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

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

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)

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

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

where

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

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

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

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

In these equations:

- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- σ_{cond} is the conductivity of the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$, where:
 - μ_0 is the permeability in free space.
 - μ_r is the **Relative permeability constant**
- The is a complex dielectric constant given by $\epsilon = \epsilon' - j\epsilon'' = \epsilon' (1 - j\tan\delta)$
- ϵ' is the real part of complex dielectric constant ϵ , $\epsilon' = \epsilon_0\epsilon_r$.
- ϵ'' is the imaginary part of complex dielectric constant ϵ , $\epsilon'' = \epsilon_0\epsilon_r\tan\delta$ where :
 - ϵ_0 is the permittivity of free space.
 - ϵ_r is the **Relative permittivity constant** parameter value.
 - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f\mu\sigma_{cond}}$.

Version History

Introduced in R2009a

References

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

See Also

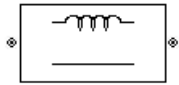
Coaxial Transmission Line | Coplanar Waveguide Transmission Line | General Passive Network | Parallel-Plate Transmission Line | Transmission Line | Microstrip Transmission Line | Two-Wire Transmission Line

Topics

“Create Complex Baseband-Equivalent Model”

Series L

Model series inductor



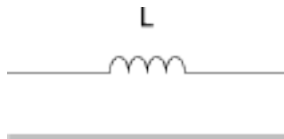
Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Series L block models the series inductor described in the block dialog box, in terms of its frequency-dependent S-parameters

The series L object is a two-port network, as shown in the following circuit diagram.



Parameters

Main

Inductance (H) — Inductance of the series element
1e-9 (default) | scalar

Scalar value for the inductance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.

Y parameter1	Y format1
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

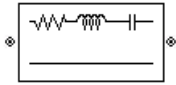
Plot the specified data using the plot button.

See Also

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

Series RLC

Model series RLC network



Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Series RLC block models the series RLC network described in the block dialog box, in terms of its frequency-dependent S-parameters.

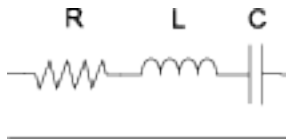
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference page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

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

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

and $\omega = 2\pi f$.

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



Parameters

Main

Resistance (Ohms) — Resistance of the series RLC network
1 (default) | scalar

Scalar value for the resistance. The value must be non-negative.

Inductance (H) — Inductance of the series RLC network
1e-9 (default) | scalar

Scalar value for the inductance. The value must be non-negative.

Capacitance (F) — Capacitance of the series RLC network
1e-12 (default) | scalar

Scalar value for the capacitance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

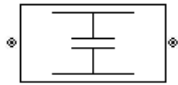
Introduced before R2006a

See Also

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

Shunt C

Model shunt capacitor



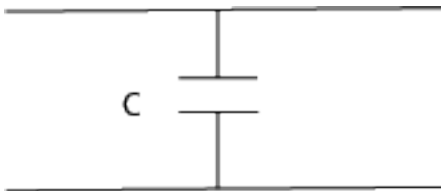
Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Shunt C block models the series inductor described in the block dialog box, in terms of its frequency-dependent S-parameters

The shunt C object is a two-port network, as shown in the following circuit diagram.



Parameters

Main

Capacitance (F) — Capacitance of the shunt element
1e-12 (default) | scalar

Scalar value for the capacitance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.

Y parameter1	Y format1
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot the specified data using the plot button.

Version History

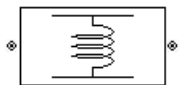
Introduced before R2006a

See Also

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

Shunt L

Model shunt inductor



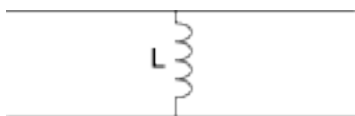
Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Shunt L block models the shunt inductor described in the block dialog box, in terms of its frequency-dependent S-parameters

The shunt L object is a two-port network, as shown in the following circuit diagram.



Parameters

Main

Inductance (H) — Inductance of the shunt element
1e-9 (default) | scalar

Scalar value for the inductance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | 0IP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, 0IP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | 0IP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, 0IP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.

Y parameter1	Y format1
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot the specified data using the plot button.

Version History

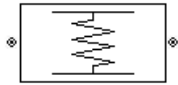
Introduced before R2006a

See Also

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

Shunt R

Model shunt resistor



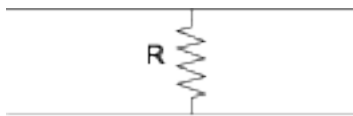
Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Shunt R block models the shunt resistor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The shunt R object is a two-port network, as shown in the following circuit diagram.



Parameters

Main

Resistance (Ohms) — Resistance of the shunt element
1 (default) | scalar

Scalar value for the resistance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.

Y parameter1	Y format1
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot the specified data using the plot button.

Version History

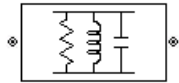
Introduced before R2006a

See Also

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

Shunt RLC

Model shunt RLC network



Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Shunt RLC block models the shunt RLC network described in the block dialog box, in terms of its frequency-dependent S-parameters.

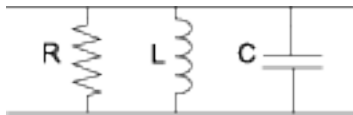
For the given resistance, inductance, and capacitance, the block first calculates the ABCD-parameters at each frequency contained in the vector of modeling frequencies, and then converts the ABCD-parameters to S-parameters using the RF Toolbox `abcd2s` function. See the Output Port block reference page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

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

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

and $\omega = 2\pi f$.

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



Parameters

Main

Resistance (Ohms) — Resistance of shunt RLC network

1 (default) | scalar

Scalar value for the resistance. The value must be non-negative.

Inductance (H) — Inductance of shunt RLC network

1e-9 (default) | scalar

Scalar value for the inductance. The value must be non-negative.

Capacitance (F) — Capacitance of shunt RLC network

1e-12 (default) | scalar

Scalar value for the capacitance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced before R2006a

See Also

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

Transmission Line (Equivalent Baseband)

Model general transmission line



Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

The Transmission Line block models the transmission line described in the block dialog box in terms of its physical parameters. The transmission line, which can be lossy or lossless, is treated as a two-port linear network. The block enables you to model the transmission line as a stub or as a stubless line.

Parameters

Main

Characteristic impedance (ohms) — Characteristic Impedance of transmission line
50 (default) | scalar | complex scalar

Characteristic impedance of the transmission line, specified as a scalar or complex scalar in ohms.

Phase velocity (m/s) — Propagation velocity of uniform plane wave on transmission line
299792458 (default) | scalar

Propagation velocity of a uniform plane wave on the transmission line, specified as a scalar in meters per seconds.

Loss (dB/m) — Reduction in strength of signal
0 (default) | positive scalar

Reduction in strength of the signal as it travels over the transmission line, specified as a positive scalar in dB/m..

Frequency (Hz) — modeling frequencies
1e9 (default) | positive vector

Modeling frequencies, specified as a positive vector in Hz.

Interpolation method — Method to interpolate network parameters
Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation

Method	Description
Cubic	Piecewise cubic Hermite interpolation

Transmission line length (m) — Physical length of transmission line

0.01 (default) | positive scalar

Physical length of the transmission line, specified as a positive scalar in meters.

Stub mode — Type of stub

Not a stub (default) | Shunt | Series

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

Stubless Transmission Line

- **Not a stub—Not a stub**

If you model a coaxial transmission line as 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. For more information, see “Stub Mode - Not a Stub” on page 2-116.

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:

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

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

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

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

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

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

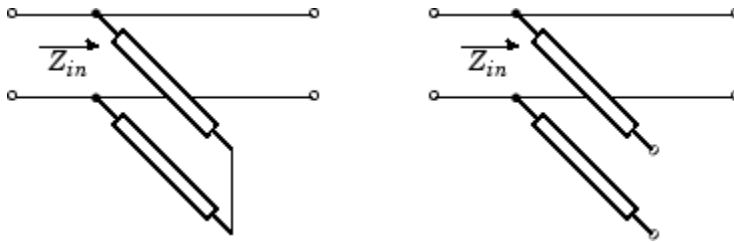
The wave number β is related to the specified phase velocity, V_p , by

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

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

Shunt Transmission Line

- **Shunt**—This parameter provides a two-port network that consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown in these diagrams.



Z_{in} is the input impedance of the shunt circuit and it is calculated as follows:

for a short circuited transmission line

and for the open circuited transmission line

where

- Z_0 is the characteristic impedance and
- γ is the propagation constant

The ABCD-parameters for the shunt stub are calculated as

$$A = 1$$

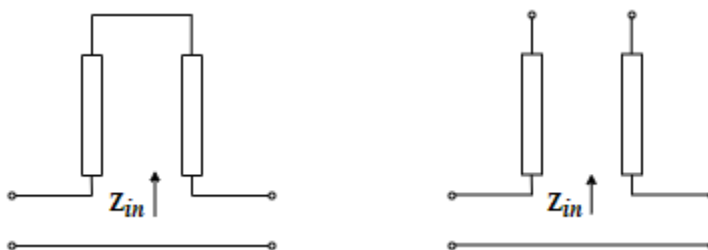
$$B = 0$$

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

$$D = 1$$

Series Transmission Line

- **Series**—This mode parameter provides a two-port network that consists of a series transmission line that you can terminate with either a short circuit or an open circuit as show in these diagrams.



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

$$A = 1$$

$$B = Z_{in}$$

$$C = 0$$

$$D = 1$$

Termination of stub — Stub termination
Open (default) | Short

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt or Series in **Stub mode**

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1e9:1.0e6:3e9] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note **Y parameter2** is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWRout.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

DependenciesTo enable **Y format1**, set **Plot type** to X-Y plane.**Y format2** — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

Introduced before R2006a

References

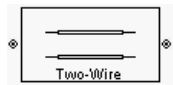
[1] Ludwig, Reinhold and Pavel Bretchko, *RF Circuit Design: Theory and Applications*, Prentice-Hall, 2000.

See Also

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

Two-Wire Transmission Line

Model two-wire transmission line

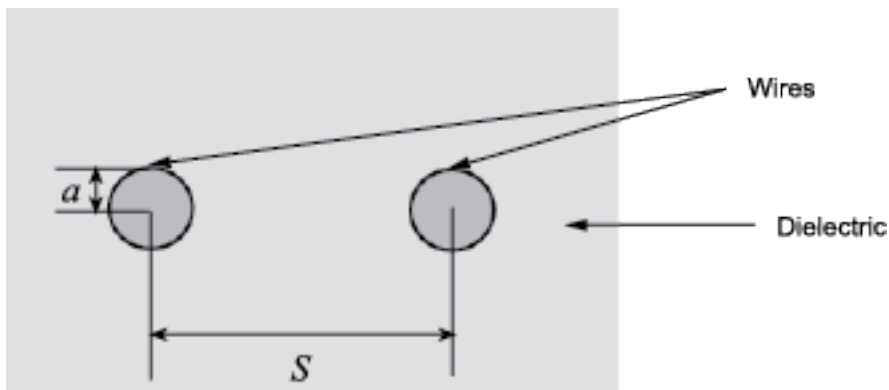


Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

The Two-Wire Transmission Line block models the two-wire transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A two-wire transmission line is shown in cross-section in the following figure. Its physical characteristics include the radius of the wires a , the separation or physical distance between the wire centers S , and the relative permittivity and permeability of the wires. RF Blockset Equivalent Baseband software assumes the relative permittivity and permeability are uniform.



Parameters

Main

Wire radius (m) — Radius of conducting wires

0.67e-3 (default) | scalar

Radius of the conducting wires of the two-wire transmission line, specified as a scalar in meters.

Wire separation (m) — Separation in two-wire transmission line

1.62e-3 (default) | scalar

Physical distance between the wires, specified as a scalar in meters.

Relative permeability constant — Relative permeability of dielectric

1 (default) | scalar

Relative permeability of the dielectric, expressed as the ratio of the permeability of the dielectric to permeability in free space μ_0 , specified as a scalar.

Relative permittivity constant — Relative permittivity of dielectric

2.3 (default) | scalar

Relative permittivity of the dielectric, expressed as the ratio of the permittivity of the dielectric to permittivity in free space ϵ_0 , specified as a scalar.

Loss tangent of dielectric — Dielectric loss tangent

0 (default) | scalar

Loss tangent of the dielectric, specified as a scalar.

Conductivity of the conductor (S/m) — Conductivity of conductor

inf (default) | scalar

Conductivity of the conductor, specified as a scalar in siemens per meter. Conductivity measures the ease with which current flows in the conductor.

Transmission line length (m) — Physical length of transmission line

0.01 (default) | scalar

Physical length of the transmission line, specified as a scalar in meters.

Stub mode — Type of stub

Not a stub (default) | Shunt | Series

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

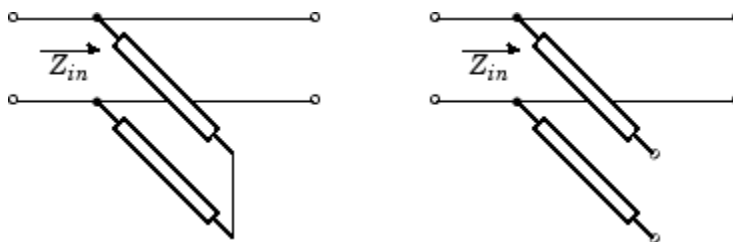
Stubless Transmission Line

- Not a stub—**Not a stub**

If you model a coaxial transmission line as 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. For more information, see “Stub Mode- Not a stub” on page 2-105.

Shunt Transmission Line

- Shunt—This parameter provides a two-port network that consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown in these diagrams.

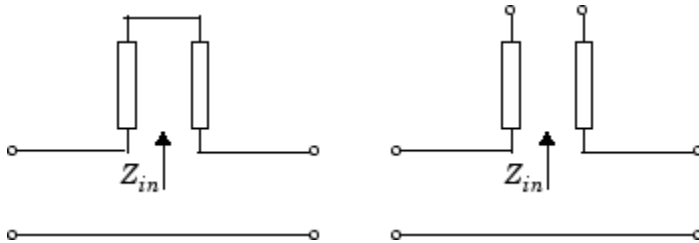


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$\begin{aligned} A &= 1 \\ B &= 0 \\ C &= 1/Z_{in} \\ D &= 1 \end{aligned}$$

Series Transmission Line

- **Series**—This mode parameter provides a two-port network that consists of a series transmission line that you can terminate with either a short circuit or an open circuit as show in these diagrams.



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

$$\begin{aligned}
 A &= 1 \\
 B &= Z_{in} \\
 C &= 0 \\
 D &= 1
 \end{aligned}$$

Termination of stub — Stub termination
 Open (default) | Short

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt or Series in **Stub mode**

Visualization

Source of frequency data — Frequency data source
 User-specified (default)

Frequency data source, specified as User-specified.

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

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
 50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to **Composite data**.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to **Composite data**.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**X scale** — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**Plot** — Plot specified data

button

Plot the specified data using the plot button.

More About**Stub Mode- Not a stub**

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

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

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

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

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

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)

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

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

where

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

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

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

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

In these equations:

- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- σ_{cond} is the conductivity of the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$, where:
 - μ_0 is the permeability in free space.
 - μ_r is the **Relative permeability constant**
- The is a complex dielectric constant given by $\varepsilon = \varepsilon' - j\varepsilon'' = \varepsilon' (1 - j\tan\delta)$
- ε' is the real part of complex dielectric constant ε , $\varepsilon' = \varepsilon_0\varepsilon_r$.
- ε'' is the imaginary part of complex dielectric constant ε , $\varepsilon'' = \varepsilon_0\varepsilon_r\tan\delta$ where :
 - ε_0 is the permittivity of free space.
 - ε_r is the **Relative permittivity constant** parameter value.
 - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f\mu\sigma_{cond}}$.

Version History

Introduced in R2009a

References

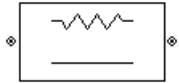
[1] Pozar, David M. *Microwave Engineering*. Hobken, NJ, John Wiley & Sons, Inc., 2005.

See Also

Coaxial Transmission Line | Coplanar Waveguide Transmission Line | General Passive Network | Transmission Line | Microstrip Transmission Line | Parallel-Plate Transmission Line | RLCG Transmission Line

Series R

Model series resistor



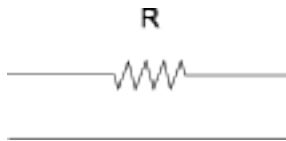
Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Series R block models the series resistor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series R object is a two-port network, as shown in the following circuit diagram.



Parameters

Main

Resistance (ohms) — Resistance value
1 (default) | scalar

Resistance value, specified as a scalar.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.

Y parameter1	Y format1
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

Version History

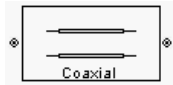
Introduced before R2006a

See Also

Inductor | Three-Winding Transformer

Coaxial Transmission Line

Model coaxial transmission line

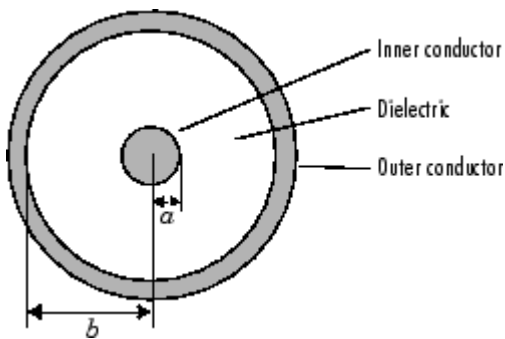


Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

The Coaxial Transmission Line block uses S-parameters to model its frequency behavior. A cross-section of the Coaxial Transmission Line is shown in the following figure. Its physical characteristics include the radius of the inner conductor a and the radius of the outer conductor b .



Parameters

Main

Outer radius (m) — Radius of outer conductor
2.57e-3 (default) | scalar

Radius of the outer conductor of the coaxial transmission line, specified as a scalar in meters.

Inner radius(m) — Radius of Inner conductor
0.725e-3 (default) | scalar

Radius of the inner conductor of the coaxial transmission line, specified as a scalar in meters.

Relative permeability constant — Relative permeability of dielectric
1 (default) | real scalar

Relative permeability of the dielectric, expressed as the ratio of the permeability of the dielectric to permeability in free space μ_0 , specified as a real scalar.

Relative permittivity constant — Relative permittivity of dielectric
2.3 (default) | real scalar

Relative permittivity of the dielectric, expressed as the ratio of the permittivity of the dielectric to permittivity in free space ϵ_0 , specified as a real scalar.

Loss tangent of dielectric — Dielectric loss tangent

0 (default) | nonnegative scalar

Loss tangent of the dielectric, specified as a nonnegative scalar.

Conductivity of the conductor (S/m) — Conductivity of conductor

inf (default) | positive scalar

Conductivity of the conductor, specified as a positive scalar in Siemens/meters.

Transmission line length (m) — Physical length of transmission line

0.01 (default) | positive scalar

Physical length of the transmission line, specified as a positive scalar in meters.

Stub mode — Type of stub

Not a stub (default) | Shunt | Series

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

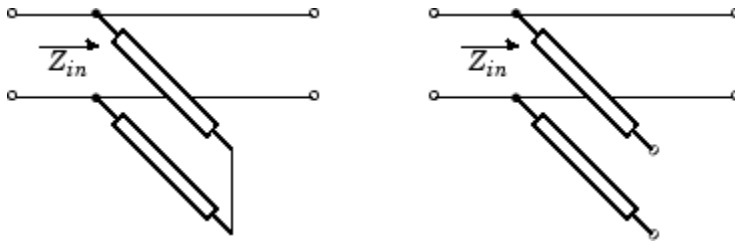
Stubless Transmission Line

- Not a stub—**Not a stub**

If you model a coaxial transmission line as 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. For more information, see “Stub Mode - Not a Stub” on page 2-116.

Shunt Transmission Line

- Shunt—This parameter provides a two-port network that consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown in these diagrams.



Z_{in} is the input impedance of the shunt circuit and it is calculated as follows:

for a short circuited transmission line

$$Z_{in} = jZ_0 \tan \gamma l$$

and for the open circuited transmission line

$$Z_{in} = -jZ_0 \cot \gamma l$$

where

- Z_0 is the characteristic impedance and

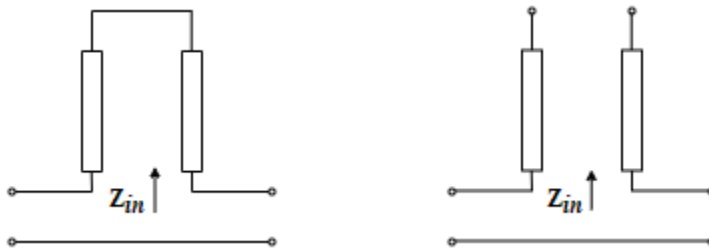
- γ is the propagation constant

The ABCD-parameters for the shunt stub are calculated as

$$\begin{aligned} A &= 1 \\ B &= 0 \\ C &= 1/Z_{in} \\ D &= 1 \end{aligned}$$

Series Transmission Line

- **Series**—This mode parameter provides a two-port network that consists of a series transmission line that you can terminate with either a short circuit or an open circuit as show in these diagrams.



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

$$\begin{aligned} A &= 1 \\ B &= Z_{in} \\ C &= 0 \\ D &= 1 \end{aligned}$$

Termination of stub — Stub termination
Open (default) | Short

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt or Series in **Stub mode**

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

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

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to **Composite data**.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note **Y parameter2** is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot the specified data using the plot button.

More About

Stub Mode - Not a Stub

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

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

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

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

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

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)

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

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

where

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

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

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

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

In these equations:

- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- σ_{cond} is the conductivity of the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$, where:
 - μ_0 is the permeability in free space.
 - μ_r is the **Relative permeability constant**
- The is a complex dielectric constant given by $\epsilon = \epsilon' - j\epsilon'' = \epsilon'(1 - j\tan\delta)$
- ϵ' is the real part of complex dielectric constant ϵ , $\epsilon' = \epsilon_0\epsilon_r$.
- ϵ'' is the imaginary part of complex dielectric constant ϵ , $\epsilon'' = \epsilon_0\epsilon_r\tan\delta$ where :
 - ϵ_0 is the permittivity of free space.
 - ϵ_r is the **Relative permittivity constant** parameter value.
 - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f\mu\sigma_{cond}}$.

Version History

Introduced in R2009a

References

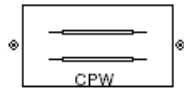
- [1] Pozar, David M. *Microwave Engineering*. Hobken, NJ, John Wiley & Sons, Inc., 2005.

See Also

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

Coplanar Waveguide Transmission Line

Model coplanar waveguide transmission line

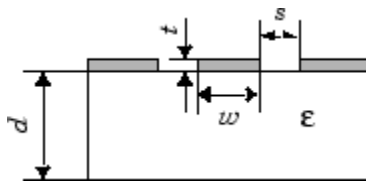


Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

The Coplanar Waveguide Transmission Line block models the coplanar waveguide transmission line described in the block dialog box in terms of its frequency-dependent S-parameters. A coplanar waveguide transmission line is shown in cross-section in the following figure. Its physical characteristics include the conductor width (w), the conductor thickness (t), the slot width (s), the substrate height (d), and the relative permittivity constant (ϵ).



Parameters

Main

Conductor width (m) — Physical width of conductor

0.6e-3 (default) | scalar

Physical width of the conductor of the coplanar waveguide, specified as a scalar in meters.

Slot width (m) — Physical width of slot

0.2e-3 (default) | scalar

Physical width of the slot of the coplanar waveguide, specified as a scalar in meters.

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

0.635e-3 (default) | scalar

Thickness of the dielectric on which the conductor resides in the coplanar waveguide, specified as a scalar in meters.

Strip thickness (m) — Physical thickness of conductor

0.005e-3 (default) | scalar

Physical thickness of the conductor in which wave propagates.

Relative permittivity constant — Relative permittivity of dielectric

9.8 (default) | scalar

Relative permittivity of the dielectric, expressed as the ratio of the permittivity of the dielectric to permittivity in free space ϵ_0 , specified as a scalar.

Loss tangent of dielectric — Dielectric loss tangent

0 (default) | scalar

Loss tangent of the dielectric, specified as a scalar.

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

inf (default) | scalar

Conductivity of the conductor, specified as a scalar in siemens per meter. Conductivity measures the ease with which current flows in the conductor.

Transmission line length (m) — Physical length of transmission line

0.01 (default) | scalar

Physical length of the transmission line, specified in meters.

Stub mode — Type of stub

Not a stub (default) | Shunt | Series

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

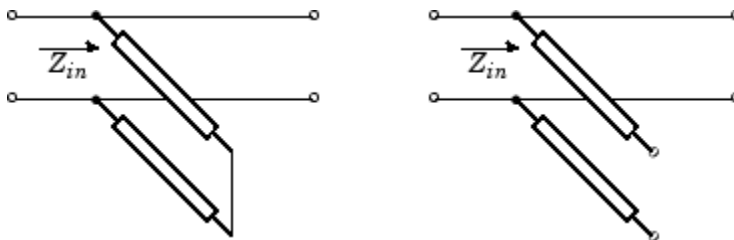
Stubless Transmission Line

- Not a stub—**Not a stub**

If you model a coaxial transmission line as 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. For more info, see “Stub Mode- Not a stub” on page 2-124.

Shunt Transmission Line

- Shunt—This parameter provides a two-port network that consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown in these diagrams.

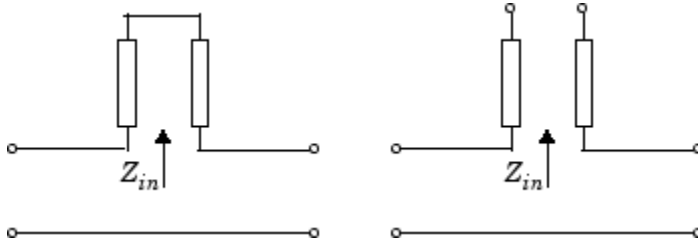


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$\begin{aligned}
 A &= 1 \\
 B &= 0 \\
 C &= 1/Z_{in} \\
 D &= 1
 \end{aligned}$$

Series Transmission Line

- **Series**—This mode parameter provides a two-port network that consists of a series transmission line that you can terminate with either a short circuit or an open circuit as show in these diagrams.



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

$$A = 1$$

$$B = Z_{in}$$

$$C = 0$$

$$D = 1$$

Termination of stub — Stub termination

Open (default) | Short

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt or Series in **Stub mode**

Visualization

Source of frequency data — Frequency data source

User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range

[1e9:1e6:3e9] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot the specified data using the plot button.

More About

Stub Mode- Not a stub

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

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

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

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

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

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)

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

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

where

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

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

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

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

In these equations:

- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- σ_{cond} is the conductivity of the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$, where:
 - μ_0 is the permeability in free space.
 - μ_r is the **Relative permeability constant**
- The is a complex dielectric constant given by $\varepsilon = \varepsilon' - j\varepsilon'' = \varepsilon' (1 - j\tan\delta)$
- ε' is the real part of complex dielectric constant ε , $\varepsilon' = \varepsilon_0\varepsilon_r$.
- ε'' is the imaginary part of complex dielectric constant ε , $\varepsilon'' = \varepsilon_0\varepsilon_r\tan\delta$ where :
 - ε_0 is the permittivity of free space.
 - ε_r is the **Relative permittivity constant** parameter value.
 - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f \mu \sigma_{cond}}$.

Version History

Introduced in R2009a

References

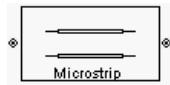
- [1] Gupta, K. C., Ramesh Garg, Inder Bahl, and Prakash Bhartia. *Microstrip Lines and Slotlines*. 2nd Edition, Artech House, Inc., Norwood, MA, 1996.

See Also

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

Microstrip Transmission Line

Model microstrip transmission line

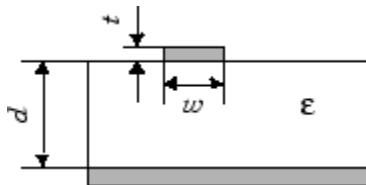


Libraries:

RF Blockset / Equivalent Baseband / Transmission Lines

Description

The Microstrip Transmission Line block models the microstrip transmission line described in the block dialog in terms of its frequency-dependent S-parameters. A microstrip transmission line is shown in cross-section in the following figure. Its physical characteristics include the microstrip width (w), the microstrip thickness (t), the substrate height (d), and the relative permittivity constant (ϵ).



Parameters

Main

Strip width (m) — Width of microstrip transmission line

0.6e-3 (default) | scalar

Width of the microstrip transmission line, specified as a scalar in meters.

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

0.635e-3 (default) | scalar

Thickness of the dielectric on which the conductor resides in the coplanar waveguide, specified as a scalar in meters.

Strip thickness (m) — Physical thickness of conductor

0.005e-3 (default) | positive scalar

Physical thickness of the conductor in which wave propagates, specified as a positive scalar in meters.

Relative permittivity constant — Relative permittivity of dielectric material in the microstrip transmission line

9.8 (default) | scalar

Relative permittivity of the dielectric expressed as the ratio of the permittivity of the dielectric to permittivity in free space ϵ_0 , specified as a scalar.

Loss tangent of dielectric — Dielectric loss tangent θ (default) | scalar

Loss tangent of the dielectric, specified as a scalar.

Conductivity of the conductor (S/m) — Conductivity of conductor

inf (default) | scalar

Conductivity of the conductor, specified as a positive scalar in Siemens/meters. Conductivity measures the ease with which current flows in the conductor.

Transmission line length (m) — Physical length of transmission line $\theta.01$ (default) | positive scalar

Physical length of the transmission line, specified as a positive scalar in meters.

Stub mode — Type of stub

Not a stub (default) | Shunt | Series

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

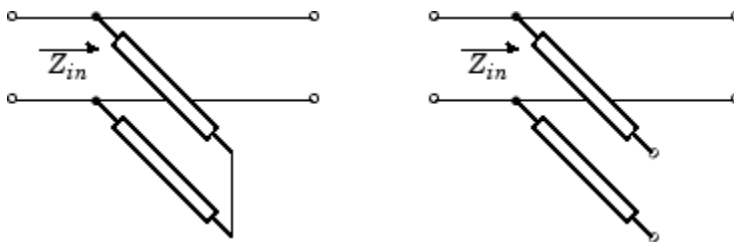
Stubless Transmission Line

- Not a stub—**Not a stub**

If you model a coaxial transmission line as 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. For more information, see “Stub Mode - Not a Stub” on page 2-131.

Shunt Transmission Line

- Shunt—This parameter provides a two-port network that consists of a stub transmission line that you can terminate with either a short circuit or an open circuit as shown in these diagrams.

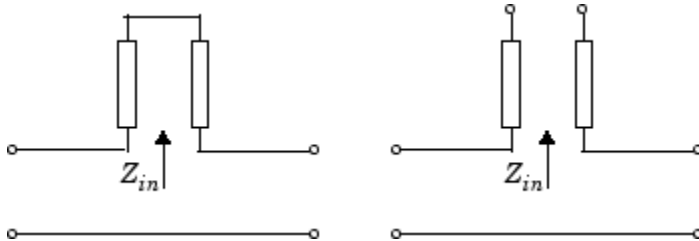


Z_{in} is the input impedance of the shunt circuit. The ABCD-parameters for the shunt stub are calculated as

$$\begin{aligned} A &= 1 \\ B &= 0 \\ C &= 1/Z_{in} \\ D &= 1 \end{aligned}$$

Series Transmission Line

- **Series**—This mode parameter provides a two-port network that consists of a series transmission line that you can terminate with either a short circuit or an open circuit as show in these diagrams.



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

$$\begin{aligned}
 A &= 1 \\
 B &= Z_{in} \\
 C &= 0 \\
 D &= 1
 \end{aligned}$$

Termination of stub — Stub termination

Open (default) | Short

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt or Series in **Stub mode**

Termination of stub — Stub termination

Open (default)

Stub termination for stub modes Shunt and Series. Choices are Open or Short

Dependencies

To enable this parameter, select Shunt, or Series, or in **Stub mode**

Visualization

Source of frequency data — Frequency data source

User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range

[1e9:1e6:3e9] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**X scale** — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**Plot** — Plot specified data

button

Plot the specified data using the plot button.

More About**Stub Mode - Not a Stub**

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

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

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

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

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

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)

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

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

where

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

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

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

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

In these equations:

- a is the radius of the inner conductor.
- b is the radius of the outer conductor.
- σ_{cond} is the conductivity of the conductor.
- μ is the permeability of the dielectric. $\mu = \mu_0 \mu_r$, where:
 - μ_0 is the permeability in free space.
 - μ_r is the **Relative permeability constant**
- The is a complex dielectric constant given by $\varepsilon = \varepsilon' - j\varepsilon'' = \varepsilon' (1 - j\tan\delta)$
- ε' is the real part of complex dielectric constant ε , $\varepsilon' = \varepsilon_0\varepsilon_r$.
- ε'' is the imaginary part of complex dielectric constant ε , $\varepsilon'' = \varepsilon_0\varepsilon_r\tan\delta$ where :
 - ε_0 is the permittivity of free space.
 - ε_r is the **Relative permittivity constant** parameter value.
 - $\tan\delta$ is the **Loss tangent of dielectric** parameter value.
- δ_{cond} is the skin depth of the conductor, which the block calculates as $1/\sqrt{\pi f\mu\sigma_{cond}}$.

Version History

Introduced in R2009a

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

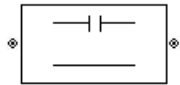
Coplanar Waveguide Transmission Line | General Passive Network | Transmission Line | Coaxial Transmission Line | Parallel-Plate Transmission Line | Two-Wire Transmission Line | RLCG Transmission Line

Topics

“Model RF Components”

Series C

Model series capacitor



Libraries:

RF Blockset / Equivalent Baseband / Series / Shunt RLC

Description

The Series C block models the series capacitor described in the block dialog box, in terms of its frequency-dependent S-parameters.

The series C object is a two-port network, as shown in the following circuit diagram.



Parameters

Main

Capacitance (F) — Capacitance of the series element
1e-12 (default) | scalar

Scalar value for the capacitance. The value must be non-negative.

Visualization

Source of frequency data — Frequency data source
User-specified (default)

Frequency data source, specified as User-specified.

Frequency data (Hz) — Frequency data range
[1:1.0e5:4e6] (default) | vector

Frequency data range, specified as a vector in hertz.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, and NTemp.
Composite data	No Y parameter2 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.

Y parameter1	Y format1
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Auto (default) | Hz | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot the specified data using the plot button.

Version History

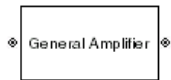
Introduced before R2006a

See Also

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

General Amplifier

Model nonlinear amplifier described by object or file data



Libraries:

RF Blockset / Equivalent Baseband / Amplifiers

Description

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

Parameters

Main

Data source — Data source that describes the amplifier behavior

Data file (default) | RFDATA object

Data source that describes the amplifier behavior, specified as a Data file or an RFDATA object.

Data file — Name of file that contains amplifier data

default.s2d (default) | string | character vector

Name of file that contains the amplifier data, specified as a string or a character vector. 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.

Dependencies

To enable this parameter, choose Data file in **Data source**.

RFDATA object — RF data object that contains amplifier data

read(rfddata.data, 'default.amp') (default) | rfddata.data object

RF data object that contains the amplifier data, specified as an RF Toolbox `rfddata.data` object, an RF toolbox command that creates the `rfddata.data` object, or a MATLAB expression that generates such an object.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data**Noise type** — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor
- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

DependenciesTo enable this parameter, select Noise figure in **Noise type**.**Minimum noise figure** — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

DependenciesTo enable this parameter, select Spot noise data in **Noise type**.**Optimal reflection coefficient** — Optimal source impedance

1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

DependenciesTo enable this parameter, select Spot noise data in **Noise type**.**Equivalent normalized noise resistance** — Normalized resistance values used to take noise measurement

1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

DependenciesTo enable this parameter, select Spot noise data in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output

1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select **Noise factor** in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces the same amount of noise as amplifier

0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the amplifier, specified as a nonnegative scalar in kelvins or nonnegative vector with each element unit in kelvins.

Dependencies

To enable this parameter, select **Noise temperature** in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data

2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

OIP3 (default) | IIP3

Type of third order intercept, specified as OIP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB,out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that amplifier produces when fully saturated
inf (default) | scalar | vector

Output power value ($P_{sat, out}$) that the amplifier produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value
3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data
2.0e9 (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.
- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

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

Frequency data source, specified as `Extracted from data source` or `User-specified`.

Frequency data (Hz) — Frequency data range
[1e9:1e8:2.9e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to `User specified`.

Source of input power — Input power source
User-specified (default) | Extracted from data source

Input power source, specified as `Extracted from data source` or `User-specified`. This parameter is disabled if the data source contains power data.

Input power data (dBm) — Input power data range
[0:19] (default) | vector

Input power data range, specified as a vector with each element unit in dBm. This parameter is disabled if the data source contains power data.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | 0IP3 | IIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	For X-Y plane data plot, Y parameter1 varies based on Data source you select. For more information, see “Parameters to plot in X-Y pane” on page 2-146.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | 0IP3 | IIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter2
X-Y plane	For X-Y plane data plot, Y parameter2 varies based on Data source you select. For more information, see “Parameters to plot in X-Y pane” on page 2-146.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.

Plot type	Y parameter2
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, GammaOPT, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
AM/AM, VSWRIn, and VSWROut.	dB, Magnitude (decibels), and None.
Pout, OIP3, and IIP3.	dBm, dBW, W, and mW.
NF	dB and Magnitude (decibels).
RN, NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Fmin, Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.
Phase and AM/PM	Angle, Angle(degrees) , and Angle(radians).

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, GammaOPT, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
AM/AM, VSWRIn, and VSWROut.	dB, Magnitude (decibels), and None.
Pout, OIP3, and IIP3.	dBm, dBW, W, and mW.
NF	dB and Magnitude (decibels).
RN, NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Fmin, Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Y parameter2	Y format2
Phase and AM/PM	Angle, Angle(degrees) , and Angle(radians).

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default) | AM

Frequency plot, specified as Freq or AM.

Dependencies

To set **X parameter** to AM, you must set **Y parameter1** to AM/AM or AM/PM.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz | ...

Frequency plot format, specified as one of the following.

X parameter	X format
Freq	Auto, Hz, kHz, MHz, GHz, and THz.
AM	Magnitude (decibels), Magnitude (linear), dB, Mag, and None.

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

Plot specified data using the plot button.

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.

Bias — Amplifier bias

1.9 (default) | scalar

Amplifier bias, specified as a scalar in volts. This parameter is set to 1.9 volts when the **Data file** is `default.s2p`.

More About

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 “Map Network Parameters to Modeling Frequencies”.

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

$$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$$

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

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

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

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

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

$$P_{sat, out} + GC_{sat} = P_{sat, in} + G_{lin}$$

$$P_{1dB, out} + 1 = P_{1dB, in} + G_{lin}$$

$$OIP3 = IIP3 + G_{lin}$$

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

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

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

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

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

If you provide vectors of nonlinearity and frequency data, then 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.

Data Consistency

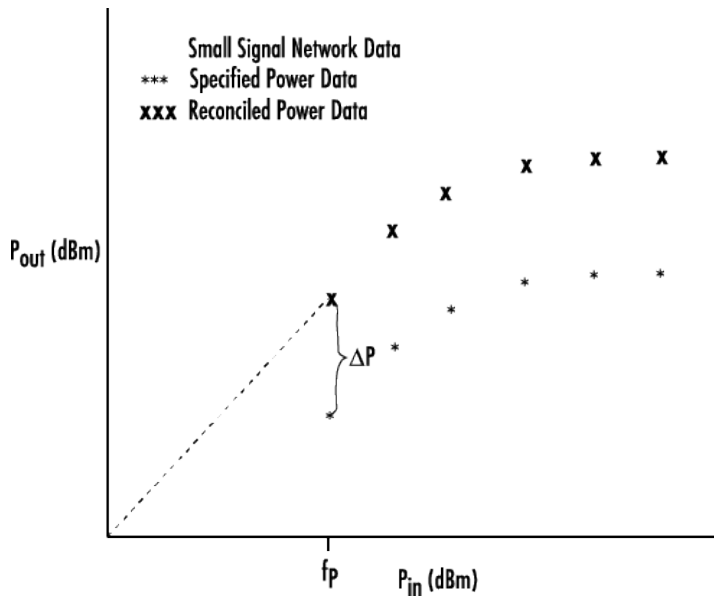
If the data source is a MathWorks AMP file or an Agilent S2D file that contains both network parameter data and power data, the blockset checks the data for consistency and reconciles it as necessary.

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

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

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

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



Parameters to plot in X-Y pane

When you select **Data source** to Data file, you can plot the parameters listed in this table in X-Y plane by selecting the **Y parameters1** and **Y parameters2** on the Visualization tab.

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWROut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime
Fmin	GammaOPT	RN	Pout
Phase	AM/AM	AM/PM	

When you select **Data source** to RFDATA object, you can plot the parameters listed in this table in X-Y plane by selecting the **Y parameters1** and **Y parameters2** on the Visualization tab.

S11	S12	S21	S22
GroupDelay	OIP3	NF	NFactor
NTemp	Fmin	GammaOPT	RN
Pout	Phase	AM/AM	AM/PM

Version History

Introduced before R2006a

See Also

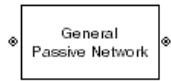
S-Parameters Amplifier | Y-Parameters Amplifier | Z-Parameters Amplifier

Topics

“Create Plots”

General Passive Network

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



Libraries:

RF Blockset / Equivalent Baseband / Black Box Elements

Description

The General Passive Network block models the two-port passive network described by an RF Toolbox data (`rfddata.data`) object.

If network parameter data and their corresponding frequencies exist as S-parameters in the `rfddata.data` object, the General Passive Network block interpolates the S-parameters to determine their values at the modeling frequencies. If the block contains network Y- or Z-parameters, the block first converts them to S-parameters. See “Map Network Parameters to Modeling Frequencies” for more details.

To learn how you can use General Passive Network block to model 2-port bandpass filter, see *Import Data Files into RF Blocks* section of “Specify or Import Component Data” example.

Parameters

Main

Data source — Data source that describes the passive device behaviour

Data file (default) | RFDATA object

Data source that describes the passive device behaviour, specified as a Data file or an RFDATA object. The data source must contain network parameters and may also include noise data, nonlinearity data, or both.

Data file — Name of file that contains passive device data

passive.s2p (default) | string | character vector

Name of file that contains the passive device data, specified as a string or a character vector. 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.

Dependencies

To enable this parameter, choose Data file in **Data source**.

RFDATA object — RF data object that contains passive device data

read(rfddata.data, 'passive.s2p') (default) | rfddata.data object |

RF data object that contains passive device data, specified as an RF Toolbox `rfddata.data` object. You can also use an expression such as the default value to create an `rfddata.data` object from an `.s2p` file.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization

Source of frequency data — Frequency data source

Extracted from data source (default) | User-specified

Frequency data source, specified as `Extracted from data source` or `User-specified`.

Frequency data — Frequency data range

[0.3e6:1e:6e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- `X-Y plane` — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- `Composite data` — Plot the composite data. For more information, see “Create Plots”.
- `Polar plane` — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2

TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

DependenciesTo enable **Y format2**, set **Plot type** to X-Y plane.**X parameter** — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**X scale** — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot specified data using the plot button.

Version History

Introduced before R2006a

See Also

General Circuit Element | S-Parameters Passive Network | Y-Parameters Passive Network | Z-Parameters Passive Network | `interp1` | `rfddata.data`

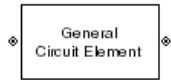
Topics

“Specify or Import Component Data”

“Create Plots”

General Circuit Element

Model two-port network described by `rfckt` object



Libraries:

RF Blockset / Equivalent Baseband / Black Box Elements

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.

To learn how you can use General Circuit Element block to realize cascaded filters, see “Multiple Realizations of Cascaded Filters” example.

Parameters

Main

RFCKT object — RFCKT object that contains circuit element device data
`read(rfckt.passive, 'passive.s2p')` (default) | any `rfckt` object |

RFCKT object that contains circuit element device data, specified as an any RF Toolbox `rfckt` object that creates a default circuit object of the specified type, or a MATLAB expression that generates such an object. See “RF Circuit Objects” for more information about circuit objects.

Visualization

Source of frequency data — Frequency data source
 User-specified (default)

Frequency data source, specified as `User-specified`.

Frequency data (Hz) — Frequency data range
`[1.9e9:1.0e6:2.2e9]` (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Reference impedance (ohms) — Reference impedance
`50` (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**X scale** — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

DependenciesTo enable this parameter, set **Plot type** to X-Y plane.**Plot** — Plot specified data

button

Plot specified data using the plot button.

Version History

Introduced before R2006a

See Also

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

Topics

"Create Plots"

S-Parameters Amplifier

Model nonlinear amplifier using S-parameters



Libraries:

RF Blockset / Equivalent Baseband / Amplifiers

Description

The S-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent S-parameters, the frequencies and reference impedance of the S-parameters, noise data, and nonlinearity data.

Parameters

Main

S-Parameters — S-parameters for nonlinear amplifier

[0, 0; 1, 0] (default) | 2-by-2-by- M array

S-parameters for a nonlinear amplifier, specified as a 2-by-2-by- M array where M is the number of S-parameters.

Frequency (Hz) — Frequency of S-parameters

2.0e9 (default) | M -element vector

Frequency of S-parameters, specified as M -element vector with each element unit in hertz.

Reference Impedance (ohms) — Reference impedance of network

50 (default) | nonnegative scalar | vector of length M

Reference impedance of network, specified as a nonnegative scalar in ohms or a vector of length M with each element unit in ohms. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor
- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise figure` in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance

1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement

1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output

1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select **Noise factor** in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces the same amount of noise as amplifier

0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the amplifier, specified as a nonnegative scalar in kelvins or nonnegative vector with each element unit in kelvins.

Dependencies

To enable this parameter, select **Noise temperature** in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data

2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

OIP3 (default) | IIP3

Type of third order intercept, specified as OIP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB, out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that amplifier produces when fully saturated

inf (default) | scalar | vector

Output power value ($P_{sat, out}$) that the amplifier produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value
3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data
 2.0×10^9 (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.
- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

Source of frequency data — Frequency data source
Same as the S-Parameters (default) | User-specified

Frequency data source, specified as Same as the S-Parameters or User-specified.

Frequency data (Hz) — Frequency data range
[1.9×10^9 : 1.0×10^8 : 2.2×10^9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to User specified.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.

Y parameter1	Y format1
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

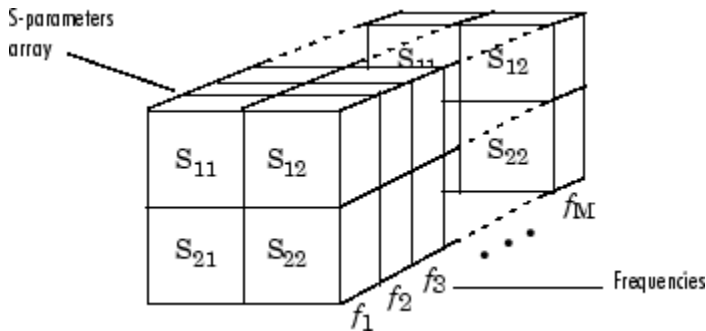
button

Plot specified data using the plot button.

More About

Network Parameters

In the **S-parameters** field of the block dialog box, provide the S-parameters for each of the M frequencies as a 2-by-2-by- M array. In the **Frequency** field, specify the frequencies for the S-parameters as an M -element vector. The elements of the frequencies vector must be in the same order as the S-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the S-parameters array and the vector of frequencies.



The S-Parameters Amplifier block interpolates the given S-parameters to determine their values at the modeling frequencies. See “Map Network Parameters to Modeling Frequencies” for more details.

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the S-Parameters Amplifier block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1 , c_3 , c_5 , and c_7 of the polynomial

$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$ 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. If you provide vectors of nonlinearity and frequency data, then the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

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

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

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

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

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

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

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

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

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

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

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

Active Noise

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

- Spot noise data in the S-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the S-Parameters Amplifier block dialog box.

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

Version History

Introduced before R2006a

See Also

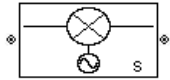
General Amplifier | Y-Parameters Amplifier | Z-Parameters Amplifier | Output Port | `interp1`

Topics

“Create Plots”

S-Parameters Mixer

Model mixer and local oscillator using S-parameters



Libraries:

RF Blockset / Equivalent Baseband / Mixers

Description

The S-Parameters Mixer block models the nonlinear mixer described in the block dialog box, in terms of its frequency-dependent S-parameters, the frequencies and reference impedance of the S-parameters, noise data (including phase noise data), and nonlinearity data.

Parameters

Main

S-Parameters — S-parameters for nonlinear mixer

[0, 0; 1, 0] (default) | 2-by-2-by- M array

S-parameters for the nonlinear mixer, specified as a 2-by-2-by- M array where M is the number of S-parameters.

Frequency (Hz) — Frequency of S-parameters

2.0e9 (default) | M -element vector

Frequency of the S-parameters, specified as an M -element vector with each element unit in hertz. All frequencies must be positive.

Reference Impedance (ohms) — Reference impedance of S-parameters

50 (default) | numeric scalar | numeric vector of length M

Reference impedance of the S-parameters as a numeric scalar or a numeric vector of length M . If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type — Type of mixer

Downconverter (default) | Upconverter

Type of mixer, specified as Downconverter or Upconverter.

LO frequency (Hz) — Local oscillator frequency

0.9e9 (default) | M -element vector

Local oscillator frequency, specified as an M -element vector with each element unit in hertz.

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

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

Noise Data**Phase noise frequency offset (Hz)** — Phase noise frequency offset

[0.1 1 10 100]*1e3 (default) | numeric vector

Phase noise frequency offset, specified as a numeric vector with units in hertz.

Data Types: double

Phase noise level (dBc/Hz) — Phase noise level

[-70 -120 -140 -150] (default) | numeric vector

Phase noise level, specified as a numeric vector with units in decibels relative to the carrier per hertz.

Data Types: double

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor
- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select **Noise figure** in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance

1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement

1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output

1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise factor` in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces same amount of noise as mixer

0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the mixer, specified as a nonnegative scalar or nonnegative vector with units in kelvins.

Dependencies

To enable this parameter, select `Noise temperature` in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data

2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data,

the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

0IP3 (default) | IIP3

Type of third order intercept, specified as 0IP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB, out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that mixer produces when fully saturated

inf (default) | scalar | vector

Output power value ($P_{sat, out}$) that the mixer produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value

3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data

2.0×10^9 (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.

- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

Source of frequency data — Frequency data source
 Same as the S-Parameters (default) | User-specified

Frequency data source, specified as Same as the S-Parameters or User-specified.

Frequency data (Hz) — Frequency data range
 [1e9:1e8:2.9e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to User specified.

Reference impedance (ohms) — Reference impedance
 50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
 S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | NFactor | NTemp | PhaseNoise | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 | S12 | S21 | S22 | GroupDelay | OIP3 | NF | NFactor | NTemp | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin
PhaseNoise	dBc/Hz

DependenciesTo enable **Y format1**, set **Plot type** to X-Y plane.**Y format2** — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.

Y parameter1	Y format1
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

button

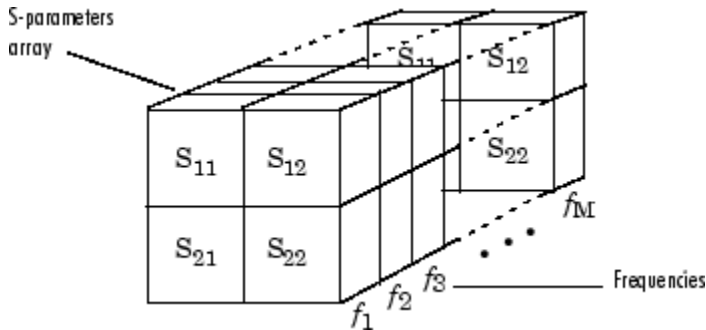
Plot specified data using the plot button.

More About

Network Parameters

The S_{21} parameter values describe the conversion gain as a function of frequency, referred to the mixer input frequency. The other S-parameters also refer to the mixer input frequency.

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



The S-Parameters Mixer block interpolates the given S-parameters to determine their values at the modeling frequencies the Output Port block calculates. For more details about how the Output Port block calculates the modeling frequencies, see “Map Network Parameters to Modeling Frequencies”.

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

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

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

Active Noise

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

- Spot noise data in the S-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the S-Parameters Mixer block dialog box.

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

Phase Noise

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

The blockset computes the digital filter by:

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

Note If you specify phase noise as a scalar value, then 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

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the S-Parameters Mixer block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1 , c_3 , c_5 , and c_7 of the polynomial

$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$ 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:

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

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

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

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

Version History

Introduced before R2006a

See Also

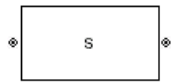
General Mixer | Output Port | Y-Parameters Mixer | Z-Parameters Mixer

Topics

“Create Plots”

S-Parameters Passive Network

Model passive network using S-parameters



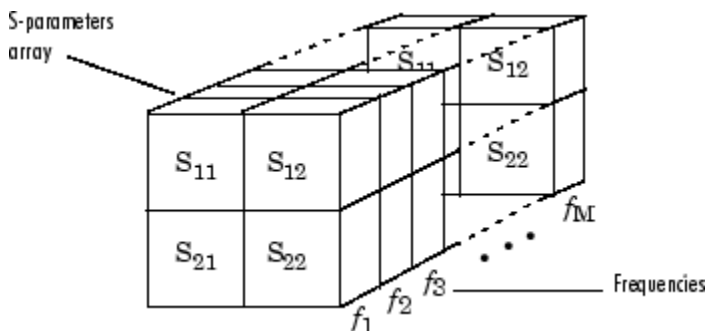
Libraries:

RF Blockset / Equivalent Baseband / Black Box Elements

Description

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

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



The S-Parameters Passive Network block interpolates the given S-parameters to determine their values at the modeling frequencies. The Output Port block determines the modeling frequencies. See “Map Network Parameters to Modeling Frequencies” for more details.

Parameters

Main

S-Parameters — S-parameters for two-port passive network

[0, 0; 1, 0] (default) | 2-by-2-by- M array

S-parameters for a two-port passive network, specified as a 2-by-2-by- M array where M is the number of S-parameters.

Frequency (Hz) — Frequency of S-parameters

2.0e9 (default) | M -element vector

Frequency of S-parameters, specified as an M -element vector with each element unit in hertz. All frequencies must be positive.

Reference Impedance (ohms) — Reference impedance of network

50 (default) | nonnegative scalar | vector of length M

Reference impedance of network, specified as a nonnegative scalar in ohms or vector of length M with each element unit in ohms. The value of this parameter can be real or complex. If you provide a scalar value, then that value is applied to all frequencies.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization

Source of frequency data — Frequency data source

Same as the S-Parameters (default) | User-specified

Frequency data source, specified as Same as the S-Parameters or User-specified.

Frequency data (Hz) — Frequency data range

[1.9e9:1.0e6:2.2e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin

Y parameter1	Y format1
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWRout.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

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

Dependencies

To enable this parameter, set **Plot type** to `X-Y plane`.

Plot — Plot specified data

button

Plot specified data using the plot button.

Version History

Introduced before R2006a

See Also

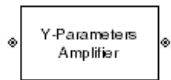
General Circuit Element | General Passive Network | Y-Parameters Passive Network | Output Port | Z-Parameters Passive Network | `interp1` | `rfddata.data`

Topics

“Create Plots”

Y-Parameters Amplifier

Model nonlinear amplifier using Y-parameters



Libraries:

RF Blockset / Equivalent Baseband / Amplifiers

Description

The Y-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent Y-parameters, the frequencies of the Y-parameters, noise data, and nonlinearity data

Parameters

Main

Y-Parameters — Y-parameters for nonlinear amplifier

[0.0200, 0; -0.0400, 0.0200] (default) | 2-by-2-by- M array

Y-parameters for a nonlinear amplifier, specified as a 2-by-2-by- M array where M is the number of Y-parameters.

Frequency (Hz) — Frequency of Y-parameters

2.0e9 (default) | M -element vector

Frequency of Y-parameters, specified as an M -element vector with each element unit in hertz.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor

- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise figure` in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance

1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement

1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output

1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise factor` in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces the same amount of noise as amplifier

0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the amplifier, specified as a nonnegative scalar in kelvins or nonnegative vector with each element unit in kelvins.

Dependencies

To enable this parameter, select **Noise temperature** in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data

2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

0IP3 (default) | IIP3

Type of third order intercept, specified as 0IP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB, out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that amplifier produces when fully saturated

inf (default) | scalar | vector

Output power value ($P_{sat, out}$) that the amplifier produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value

3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data
2.0e9 (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.
- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

Source of frequency data — Frequency data source
Same as the Y-Parameters (default) | User-specified

Frequency data source, specified as Same as the Y-Parameters or User-specified.

Frequency data (Hz) — Frequency data range
[1.9e9:1.0e8:2.2e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to User specified.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None

Y parameter1	Y format1
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWROut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

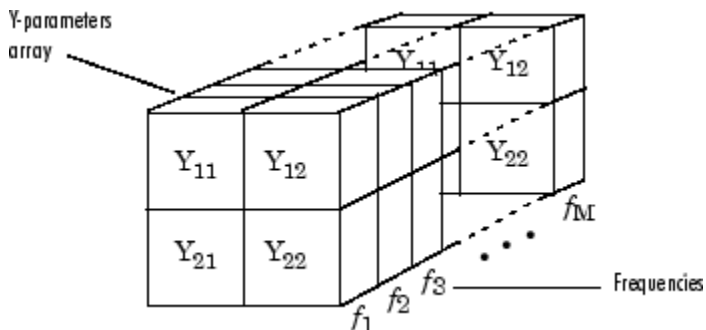
button

Plot specified data using the plot button.

More About

Network Parameters

In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of the M frequencies as a 2-by-2-by- M array. In the **Frequency** field, specify the frequencies for the Y-parameters as an M -element vector. The elements of the frequencies vector must be in the same order as the Y-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Y-parameters array and the vector of frequencies.



The Y-Parameters Amplifier block uses the RF Toolbox `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See “Map Network Parameters to Modeling Frequencies” for more details.

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Y-Parameters Amplifier block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1 , c_3 , c_5 , and c_7 of the polynomial

$F_{AM/AM}(s) = c_1 s + c_3 |s|^2 s + c_5 |s|^4 s + c_7 |s|^6 s$ that determines the AM/AM conversion for the input signal s . The block automatically calculates c_1 , the linear gain term. If you do not specify additional nonlinearity data, the block operates as a linear amplifier. If you do, the block calculates one or more of the remaining coefficients as the solution to a system of linear equations, determined by the following method. If you provide vectors of nonlinearity and frequency data, then the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

- 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:

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

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

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

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

Active Noise

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

- Spot noise data in the Y-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the Y-Parameters Amplifier block dialog box.

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

Version History

Introduced before R2006a

See Also

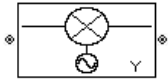
General Amplifier | S-Parameters Amplifier | Z-Parameters Amplifier | Output Port | `interp1`

Topics

“Create Plots”

Y-Parameters Mixer

Model mixer and local oscillator using Y-parameters



Libraries:

RF Blockset / Equivalent Baseband / Mixers

Description

The Y-Parameters Mixer block models the nonlinear mixer described in the block dialog box in terms of its frequency-dependent Y-parameters, the frequencies of the Y-parameters, noise data (including phase noise data), and nonlinearity data.

Parameters

Main

Y-Parameters — Y-parameters for nonlinear mixer

[0.0200, 0; -0.0400, 0.0200] (default) | 2-by-2-by- M array

Y-parameters for the nonlinear mixer, specified as a 2-by-2-by- M array where M is the number of Y-parameters.

Frequency (Hz) — Frequency of Y-parameters

2.0e9 (default) | M -element vector

Frequency of the Y-parameters, specified as an M -element vector with each element unit in hertz. All frequencies must be positive.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type — Type of mixer

Downconverter (default) | Upconverter

Type of mixer, specified as Downconverter or Upconverter.

LO frequency (Hz) — Local oscillator frequency

0.9e9 (default) | M -element vector

Local oscillator frequency, specified as an M -element vector with each element unit in hertz.

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

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

Noise Data

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

[0.1 1 10 100]*1e3 (default) | numeric vector

Phase noise frequency offset, specified as a numeric vector with units in hertz.

Data Types: double

Phase noise level (dBc/Hz) — Phase noise level

[-70 -120 -140 -150] (default) | numeric vector

Phase noise level, specified as a numeric vector with units in decibels relative to the carrier per hertz.

Data Types: double

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor
- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select Noise figure in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance
1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement
1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output
1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise factor` in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces same amount of noise as mixer
0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the mixer, specified as a nonnegative scalar or nonnegative vector with units in kelvins.

Dependencies

To enable this parameter, select `Noise temperature` in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data
2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

0IP3 (default) | IIP3

Type of third order intercept, specified as 0IP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB,out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that mixer produces when fully saturated

inf (default) | scalar | vector

Output power value ($P_{sat,out}$) that the mixer produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value

3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data

$2.0e9$ (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.
- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

Source of frequency data — Frequency data source

Same as the Y-Parameters (default) | User-specified

Frequency data source, specified as Same as the Y-Parameters or User-specified.

Frequency data (Hz) — Frequency data range

[1e9:1e8:2.9e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to User specified.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | NFactor | NTemp | PhaseNoise | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 | S12 | S21 | S22 | GroupDelay | OIP3 | NF | NFactor | NTemp | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None

Y parameter1	Y format1
NTemp	Kelvin
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

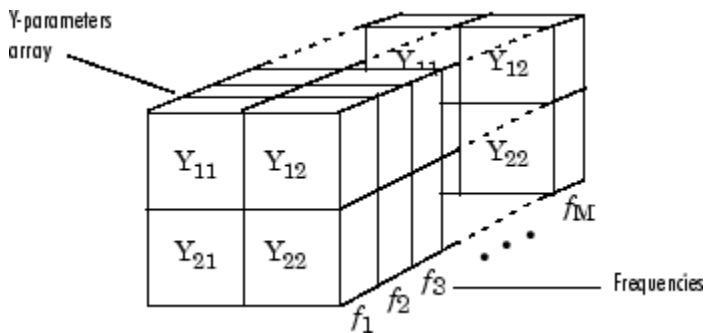
Plot — Plot specified data
button

Plot specified data using the plot button.

More About

Network Parameters

In the **Y-Parameters** field of the block dialog box, provide the Y-parameters for each of M frequencies as a 2-by-2-by- M array. The Y-parameter values all refer to the mixer input frequency. In the **Frequency** field, specify the frequencies for the Y-parameters as an M -element vector. The elements of the vector must be in the same order as the Y-parameters. The figure shows the correspondence between the Y-parameters array and the vector of frequencies.



The Y-Parameters Mixer block uses the RF Toolbox `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See “Map Network Parameters to Modeling Frequencies” 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

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

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

Active Noise

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

- Spot noise data in the Y-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the Y-Parameters Mixer block dialog box.

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

Phase Noise

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

The blockset computes the digital filter by:

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

Note If you specify phase noise as a scalar value, then 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

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Y-Parameters Mixer block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1 , c_3 , c_5 , and c_7 of the polynomial

$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$ 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:

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

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

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

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

The calculation omits higher-order terms according to the available degrees of freedom of the system. If you specify only two of the three parameters, the block does not use the equation

involving the parameter you did not specify, and eliminates any c_7 terms from the remaining equations. Similarly, if you provide only one of the parameters, the block uses only the solution to the equation involving that parameter and omits any c_5 or c_7 terms.

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

Version History

Introduced before R2006a

See Also

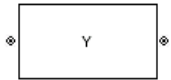
General Mixer | Output Port | S-Parameters Mixer | Z-Parameters Mixer

Topics

“Create Plots”

Y-Parameters Passive Network

Model passive network using Y-parameters



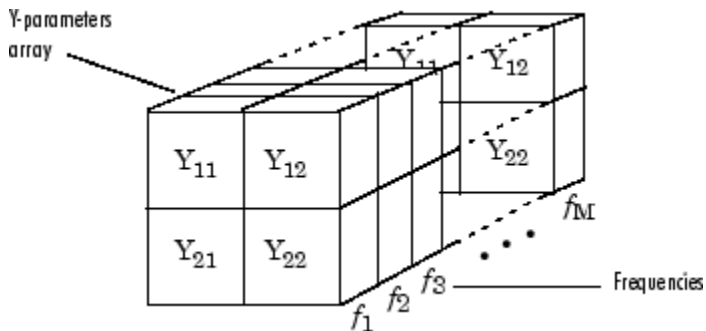
Libraries:

RF Blockset / Equivalent Baseband / Black Box Elements

Description

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

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



The Y-Parameters Passive Network block uses the RF Toolbox `y2s` function to convert the Y-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See “Map Network Parameters to Modeling Frequencies” for more details.

Parameters

Main

Y-Parameters — Y-parameters for two-port passive network
`[0.0200, 0; -0.0400, 0.0200]` (default) | 2-by-2-by- M array

Y-parameters for a two-port passive network, specified as a 2-by-2-by- M array where M is the number of Y-parameters.

Frequency (Hz) — Frequency of Y-parameters
`2.0e9` (default) | M -element vector

Frequency of Y-parameters, specified as an M -element vector with each element unit in hertz. All frequencies must be positive.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization

Source of frequency data — Frequency data source

Same as the Y-Parameters (default) | User-specified

Frequency data source, specified as Same as the Y-Parameters or User-specified.

Frequency data (Hz) — Frequency data range

[1.9e9:1.0e6:2.2e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWROut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWROut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot specified data using the plot button.

Version History

Introduced before R2006a

See Also

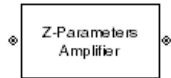
General Circuit Element | General Passive Network | Output Port | S-Parameters Passive Network | Z-Parameters Passive Network | `interp1` | `rfddata.data`

Topics

“Create Plots”

Z-Parameters Amplifier

Model nonlinear amplifier using Z-parameters



Libraries:

RF Blockset / Equivalent Baseband / Amplifiers

Description

The Z-Parameters Amplifier block models the nonlinear amplifier described in the block dialog box, in terms of its frequency-dependent Z-parameters, the frequencies of the Z-parameters, noise data, and nonlinearity data

Parameters

Main

Z-Parameters — Z-parameters for nonlinear amplifier

[0.0200, 0; -0.0400, 0.0200] (default) | 2-by-2-by- M array

Z-parameters for a nonlinear amplifier, specified as a 2-by-2-by- M array where M is the number of Z-parameters.

Frequency (Hz) — Frequency of Z-parameters

2.0e9 (default) | M -element vector

Frequency of Z-parameters, specified as M -element vector with each element unit in hertz.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Noise Data

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor

- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise figure` in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance

1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement

1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output

1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise factor` in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces the same amount of noise as amplifier

0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the amplifier, specified as a nonnegative scalar in kelvins or nonnegative vector with each element unit in kelvins.

Dependencies

To enable this parameter, select **Noise temperature** in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data

2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

OIP3 (default) | IIP3

Type of third order intercept, specified as OIP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB, out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that amplifier produces when fully saturated

inf (default) | scalar | vector

Output power value ($P_{sat, out}$) that the amplifier produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value

3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data
2.0e9 (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.
- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

Source of frequency data — Frequency data source
Same as the Z-Parameters (default) | User-specified

Frequency data source, specified as Same as the Z-Parameters or User-specified.

Frequency data (Hz) — Frequency data range
[1.9e9:1.0e8:2.2e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to User specified.

Reference impedance (ohms) — Reference impedance
50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- **X-Y plane** — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot
S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None

Y parameter1	Y format1
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWRout.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

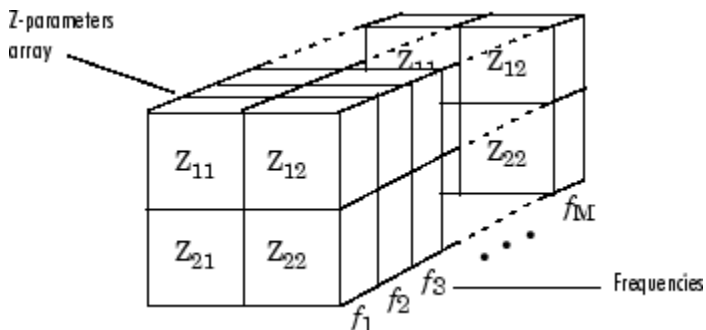
button

Plot specified data using the plot button.

More About

Network Parameters

In the **Z-Parameters** field of the block dialog box, provide the Z-parameters for each of the M frequencies as a 2-by-2-by- M array. In the **Frequency** field, specify the frequencies for the Z-parameters as an M -element vector. The elements of the frequencies vector must be in the same order as the Z-parameters. All frequencies must be positive. For example, the following figure shows the correspondence between the Z-parameters array and the vector of frequencies.



The Z-Parameters Amplifier block uses the RF Toolbox `z2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See “Map Network Parameters to Modeling Frequencies” for more details..

Nonlinearity

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Z-Parameters Amplifier block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1 , c_3 , c_5 , and c_7 of the polynomial

$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$ 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. If you provide vectors of nonlinearity and frequency data, then the block calculates the polynomial coefficients using values for the parameters interpolated at the center frequency.

- 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:

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

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

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

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

Active Noise

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

- Spot noise data in the Z-Parameters Amplifier block dialog box.
- Noise figure, noise factor, or noise temperature value in the Z-Parameters Amplifier block dialog box.

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

Version History

Introduced before R2006a

See Also

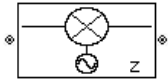
General Amplifier | S-Parameters Amplifier | Y-Parameters Amplifier | Output Port | `interp1`

Topics

“Create Plots”

Z-Parameters Mixer

Model mixer and local oscillator using Z-parameters



Libraries:

RF Blockset / Equivalent Baseband / Mixers

Description

The Z-Parameters Mixer block models the nonlinear mixer described in the block dialog box in terms of its frequency-dependent Z-parameters, the frequencies of the Z-parameters, noise data (including phase noise data), and nonlinearity data.

Parameters

Main

Z-Parameters — Z-parameters for nonlinear mixer

[0.0200, 0; -0.0400, 0.0200] (default) | 2-by-2-by- M array

Z-parameters for the nonlinear mixer, specified as a 2-by-2-by- M array where M is the number of Z-parameters.

Frequency (Hz) — Frequency of Z-parameters

2.0e9 (default) | M -element vector

Frequency of the Z-parameters, specified as an M -element vector with each element unit in hertz. All frequencies must be positive.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type — Type of mixer

Downconverter (default) | Upconverter

Type of mixer, specified as Downconverter or Upconverter.

LO frequency (Hz) — Local oscillator frequency

0.9e9 (default) | M -element vector

Local oscillator frequency, specified as an M -element vector with each element unit in hertz.

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

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

Noise Data

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

[0.1 1 10 100]*1e3 (default) | numeric vector

Phase noise frequency offset, specified as a numeric vector with units in hertz.

Data Types: double

Phase noise level (dBc/Hz) — Phase noise level

[-70 -120 -140 -150] (default) | numeric vector

Phase noise level, specified as a numeric vector with units in decibels relative to the carrier per hertz.

Data Types: double

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor
- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise figure` in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance
1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement
1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output
1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise factor` in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces same amount of noise as mixer
0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the mixer, specified as a nonnegative scalar or nonnegative vector with units in kelvins.

Dependencies

To enable this parameter, select `Noise temperature` in **Noise type**.

Nonlinearity Data

IP3 type — Type of third order intercept
0IP3 (default) | IIP3

Type of third order intercept, specified as 0IP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value
inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB
 inf (default) | scalar | vector

Output power value ($P_{1dB, out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that mixer produces when fully saturated
 inf (default) | scalar | vector

Output power value ($P_{sat, out}$) that the mixer produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value
 3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Visualization

Source of frequency data — Frequency data source
 Same as the Z-Parameters (default) | User-specified

Frequency data source, specified as Same as the Z-Parameters or User-specified.

Frequency data (Hz) — Frequency data range
 [1e9:1e8:2.9e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to User specified.

Reference impedance (ohms) — Reference impedance
 50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

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

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.

- **Composite data** — Plot the composite data. For more information, see “Create Plots”.
- **Polar plane** — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | NFactor | NTemp | PhaseNoise | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 | S12 | S21 | S22 | GroupDelay | OIP3 | NF | NFactor | NTemp | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.

Y parameter1	Y format1
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, and S22.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor	None
NTemp	Kelvin
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data

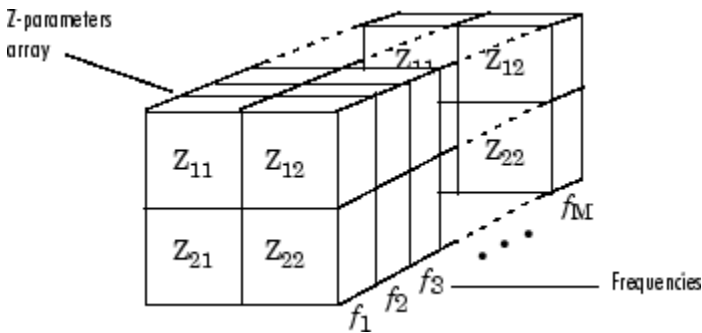
button

Plot specified data using the plot button.

More About

Network Parameters

In the **Z-Parameters** field of the block dialog box, provide the Z-parameters for each of M frequencies as a 2-by-2-by- M array. The Z-parameter values all refer to the mixer input frequency. In the **Frequency** field, specify the frequencies for the Z-parameters as an M -element vector. The elements of the vector must be in the same order as the Z-parameters. The figure shows the correspondence between the Z-parameters array and the vector of frequencies.



The Z-Parameters Mixer block uses the RF Toolbox `z2s` function to convert the Z-parameters to S-parameters and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. See “Map Network Parameters to Modeling Frequencies” 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

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

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

Active Noise

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

- Spot noise data in the Z-Parameters Mixer block dialog box.
- Noise figure, noise factor, or noise temperature value in the Z-Parameters Mixer block dialog box.

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

Phase Noise

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

The blockset computes the digital filter by:

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

Note If you specify phase noise as a scalar value, then 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

You can introduce nonlinearities into your model by specifying parameters in the **Nonlinearity Data** tab of the Z-Parameters Mixer block dialog box. Depending on which of these parameters you specify, the block computes up to four of the coefficients c_1 , c_3 , c_5 , and c_7 of the polynomial

$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$ 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:

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

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

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

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

Version History

Introduced before R2006a

See Also

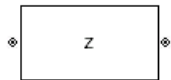
General Mixer | Output Port | S-Parameters Mixer | Y-Parameters Mixer

Topics

“Create Plots”

Z-Parameters Passive Network

Model passive network using Z-parameters



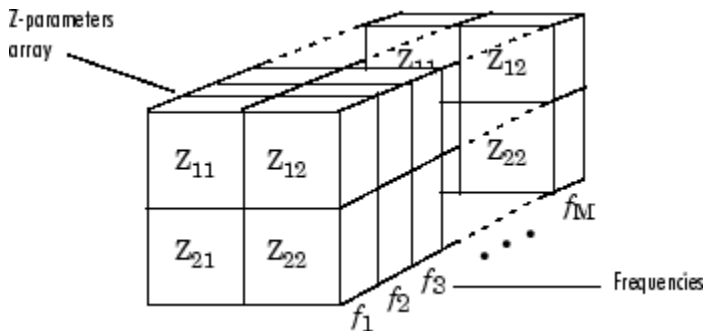
Libraries:

RF Blockset / Equivalent Baseband / Black Box Elements

Description

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

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



The Z-Parameters Passive Network block uses the RF Toolbox `z2s` function to convert the Z-parameters to S-parameters, and then interpolates the resulting S-parameters to determine their values at the modeling frequencies. The modeling frequencies are determined by the Output Port block. See “Map Network Parameters to Modeling Frequencies” for more details.

Parameters

Main

Z-Parameters — Z-parameters for two-port passive network
`[50, 0; 100, 50]` (default) | 2-by-2-by- M array

Z-parameters for a two-port passive network, specified as a 2-by-2-by- M array where M is the number of Z-parameters.

Frequency (Hz) — Frequency of Z-parameters
`2.0e9` (default) | M -element vector

Frequency of Z-parameters, specified as an M -element vector with each element unit in hertz. All frequencies must be positive.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Visualization

Source of frequency data — Frequency data source

Same as the Z-Parameters (default) | User-specified

Frequency data source, specified as Same as the Z-Parameters or User-specified.

Frequency data (Hz) — Frequency data range

[1.9e9:1.0e6:2.2e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- X-Y plane — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- Composite data — Plot the composite data. For more information, see “Create Plots”.
- Polar plane — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWRout	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter1 is disabled when you select **Plot type** to Composite data.

Y parameter2 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | IIP3 | NF | ...

Type of parameters to plot, specified as one of the following:

S11	S12	S21	S22
GroupDelay	GammaIn	GammaOut	VSWRIn
VSWR0ut	OIP3	IIP3	NF
NFactor	NTemp	TF1	TF2
TF3	Gt	Ga	Gp
Gmag	Gmsg	GammaMS	GammaML
K	Delta	Mu	MuPrime

Note Y parameter2 is disabled when you select **Plot type** to Composite data.

Y format1 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Mag | Magnitude (linear) | Angle | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, GammaIn, GammaOut, TF1, TF2, TF3, GammaMS, GammaML, and Delta.	dB, Magnitude (decibels), Abs, Mag, Magnitude (linear), Angle, Angle(degrees), Angle(radians), Real, Imag, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
VSWRIn and VSWR0ut.	Magnitude (decibels) and None.
OIP3 and IIP3.	dBm, W, and mW.
NF	dB and Magnitude (decibels).
NFactor, K, Mu, and MuPrime.	None
NTemp	Kelvin
Gt, Ga, Gp, Gmag, and Gmsg.	dB, Magnitude (decibels), and None.

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot
Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format
Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale
Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

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

X-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

Plot — Plot specified data
button

Plot specified data using the plot button.

Version History

Introduced before R2006a

See Also

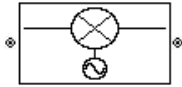
General Circuit Element | General Passive Network | Output Port | S-Parameters Passive Network | Y-Parameters Passive Network | `interp1` | `rfddata.data`

Topics

“Create Plots”

General Mixer

Model mixer and local oscillator described by `rfddata` object



Libraries:

RF Blockset / Equivalent Baseband / Mixers

Description

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

Parameters

Main

Data source — Data source that describes the mixer behavior

Data file (default) | RFDATA object

Data source that describes the mixer behavior, specified as a Data file or an RFDATA object.

Data Types: char

Data file — Name of file that contains mixer data

default.s2d (default) | string | character vector

Name of file that contains the mixer data, specified as a string or a character vector. 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.

Dependencies

To enable this parameter, choose Data file in **Data source**.

Data Types: char | string

RFDATA object — RF data object that contains mixer data

read(rfddata.data, 'default.s2p') (default) | rfddata.data object

RF data object that contains the mixer data, specified as an RF Toolbox `rfddata.data` object, an RF Toolbox command that creates the `rfddata.data` object, or a MATLAB expression that generates such an object.

Interpolation method — Method to interpolate network parameters

Linear (default) | Spline | Cubic

Method to interpolate the network parameters, specified as one of the following:

Method	Description
Linear	Linear interpolation
Spline	Cubic spline interpolation
Cubic	Piecewise cubic Hermite interpolation

Mixer Type — Type of mixer

Downconverter (default) | Upconverter

Type of mixer, specified as Downconverter or Upconverter.

LO frequency (Hz) — Local oscillator frequency0.9e9 (default) | M -element vectorLocal oscillator frequency, specified as an M -element vector with each element unit in hertz.

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

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

Noise Data**Phase noise frequency offset (Hz)** — Phase noise frequency offset

[0.1 1 10 100]*1e3 (default) | numeric vector

Phase noise frequency offset, specified as a numeric vector with units in hertz.

Data Types: double

Phase noise level (dBc/Hz) — Phase noise level

[-70 -120 -140 -150] (default) | numeric vector

Phase noise level, specified as a numeric vector with units in decibels relative to the carrier per hertz.

Data Types: double

Noise type — Type of noise data

Noise figure (default) | Spot noise data | Noise factor | Noise temperature

Type of noise data, specified as one of the following:

- Noise figure
- Spot noise data
- Noise factor
- Noise temperature

This parameter is disabled if the data source contains noise data.

Noise figure (dB) — Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output

0 (default) | scalar ratio | vector of ratios

Available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise figure` in **Noise type**.

Minimum noise figure — Minimum ratio of available signal-to-noise ratio at input to available signal-to-noise ratio at output

0 (default) | scalar ratio | vector of ratios

Minimum ratio of available signal-to-noise ratio at the input to available signal-to-noise ratio at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Optimal reflection coefficient — Optimal source impedance

1+0i (default) | complex scalar | complex vector

Optimal source impedance, specified as a complex scalar or a complex vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Equivalent normalized noise resistance — Normalized resistance values used to take noise measurement

1 (default) | positive scalar | positive vector

Normalized resistance values used to take noise measurement, specified as a positive scalar or a positive vector.

Dependencies

To enable this parameter, select `Spot noise data` in **Noise type**.

Noise factor — Ratio of available signal-to-noise power at input to available signal-to-noise power at output

1 (default) | scalar ratio | vector of ratios

Ratio of available signal-to-noise power at the input to available signal-to-noise power at the output, specified as a scalar ratio or a vector of ratios.

Dependencies

To enable this parameter, select `Noise factor` in **Noise type**.

Noise temperature (K) — Equivalent temperature that produces same amount of noise as mixer

0 (default) | nonnegative scalar | nonnegative vector

Equivalent temperature that produces the same amount of noise as the mixer, specified as a nonnegative scalar or nonnegative vector with units in kelvins.

Dependencies

To enable this parameter, select **Noise temperature** in **Noise type**.

Frequency (Hz) — Domain of frequencies to express noise data

2.0e9 (default) | nonnegative scalar | nonnegative vector

Domain of frequencies to express noise data, specified as a nonnegative scalar in hertz or nonnegative vector with each element unit in hertz. If you provide a scalar value for your noise data, the block ignores the **Frequency (Hz)** parameter and uses the same 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

IP3 type — Type of third order intercept

0IP3 (default) | IIP3

Type of third order intercept, specified as 0IP3 (output intercept point) or IIP3 (input intercept point). This parameter is disabled if the data source contains power data or IP3 data.

IP3 (dBm) — IP3 value

inf (default) | scalar | vector

IP3 value, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

1dB gain compression power (dBm) — Output power value at which gain has decreased by 1 dB

inf (default) | scalar | vector

Output power value ($P_{1dB,out}$) at which the gain has decreased by 1 dB, specified as a scalar in dBm for frequency independent nonlinear data or vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains power data or IP3 data.

Output saturation power (dBm) — Output power value that mixer produces when fully saturated

inf (default) | scalar | vector

Output power value ($P_{sat,out}$) that the mixer produces when fully saturated, specified as a scalar in dBm for frequency independent nonlinear data or a vector with each element unit in dBm for frequency dependent nonlinear data. This parameter is disabled if the data source contains output saturation power data.

Gain compression at saturation (dB) — Decrease in gain value

3 (default) | scalar | vector

Decrease in gain value (GC_{sat}) when the power is fully saturated, specified as a scalar in dB for frequency independent nonlinear data or a vector with each element unit in dB for frequency dependent nonlinear data.

Frequency (Hz) — Frequency points corresponding to third-order intercept or power data

2.0e9 (default) | positive scalar | positive vector

Frequency points corresponding to third-order intercept or power data, specified as a positive scalar or positive vector in units of hertz. This parameter is disabled if the data source contains power data or IP3 data.

Dependencies

- If you specify the frequency as a scalar, then the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must all be scalars.
- If you specify the frequency as a vector, then or more of the **IP3 (dBm)**, **1 dB gain compression power (dBm)**, and **Output saturation power (dBm)** parameters must also be a vector.

Visualization

Source of frequency data — Frequency data source

Extracted from data source (default) | User-specified

Frequency data source, specified as `Extracted from data source` or `User-specified`.

Frequency data (Hz) — Frequency data range

[1e9:1e8:2.9e9] (default) | vector

Frequency data range, specified as a vector with each element unit in hertz.

Dependencies

To enable this parameter, set **Source of frequency data** to `User specified`.

Source of input power data — Input power data source

Extracted from data source (default)

Input power data source, specified as `Extracted from data source`.

Input power data (dBm) — Input power data

[0:19] (default) | vector

Input power data, specified as a vector with each element unit in dBm.

Reference impedance (ohms) — Reference impedance

50 (default) | nonnegative scalar

Reference impedance, specified as a nonnegative scalar in ohms.

Plot type — Type of data plot

X-Y plane (default) | Composite data | Polar plane | Z Smith chart | Y Smith chart | ZY Smith chart

Type of data plot to visualize using the given data, specified as one of the following:

- `X-Y plane` — Generate a Cartesian plot of the data versus frequency. To create linear, semilog, or log-log plots, set the **Y-axis scale** and **X-axis scale** accordingly.
- `Composite data` — Plot the composite data. For more information, see “Create Plots”.
- `Polar plane` — Generate a polar plot of the 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.

Y parameter1 — Type of parameters to plot

S11 (default) | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, Fmin, GaammaOPT, RN, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y parameter2 — Type of parameters to plot

S11 | S12 | S21 | S22 | GroupDelay | OIP3 | NF | ...

Type of parameters to plot based on the **Plot type** you set, specified as one of the following.

Plot type	Y parameter1
X-Y plane	S11, S12, S21, S22, GroupDelay, OIP3, NF, NFactor, NTemp, Fmin, GaammaOPT, RN, and PhaseNoise.
Composite data	No Y parameter1 to set.
Polar plane	S11, S12, S21, and S22
Z Smith chart	S11 and S22.
Y Smith chart	S11 and S22.
ZY smith chart	S11 and S22.

Y format1 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter1	Y format1
S11, S12, S21, S22, and GaammaOPT.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor and RN.	None
NTemp	Kelvin

Y parameter1	Y format1
Fmin	None and Magnitude (decibels)
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format1**, set **Plot type** to X-Y plane.

Y format2 — Plot format

Magnitude (decibels) (default) | Magnitude (linear) | Angle(degrees) | Angle(radians) | Real | Imaginary | ...

Plot format, specified as one of the following.

Y parameter2	Y format2
S11, S12, S21, S22, and GaammaOPT.	Magnitude (decibels), Magnitude (linear), Angle(degrees), Angle(radians), Real, and Imaginary.
GroupDelay	ns, us, ms, s, and ps.
OIP3	dB, dBm, W, and mW.
NF	Magnitude (decibels)
NFactor and RN.	None
NTemp	Kelvin
Fmin	None and Magnitude (decibels)
PhaseNoise	dBc/Hz

Dependencies

To enable **Y format2**, set **Plot type** to X-Y plane.

X parameter — Frequency plot

Freq (default)

Frequency plot, specified as Freq.

X format — Frequency plot format

Hz (default) | Auto | kHz | MHz | GHz | THz

Frequency plot format, specified as one of the following.

Auto	Hz	kHz	MHz
GHz	THz		

Y scale — Y-axis scale

Linear (default) | Log

Y-axis scale, specified as Linear or Log.

Dependencies

To enable this parameter, set **Plot type** to X-Y plane.

X scale — X-axis scale

Linear (default) | Log

X-axis scale, specified as **Linear** or **Log**.

Dependencies

To enable this parameter, set **Plot type** to **X-Y plane**.

Plot — Plot specified data

button

Plot specified data using the plot button.

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.

More About**Network Parameters**

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

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

- 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 (`rfddata.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 (`rfddata.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, then 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

$$F_{AM/AM}(s) = c_1s + c_3|s|^2s + c_5|s|^4s + c_7|s|^6s$$

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:

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

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

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

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

Version History

Introduced before R2006a

See Also

rfddata.data | S-Parameters Mixer | Y-Parameters Mixer | Z-Parameters Mixer

Topics

“Create Plots”

Functions

simrfSupportPackages

Start Add-On Explorer to download, install, or uninstall RF Blockset models and supporting software for third-party hardware

Syntax

```
simrfSupportPackages
```

Description

`simrfSupportPackages` opens the Add-On Explorer to install the Analog Devices models and testbenches.

Examples

Install RF Blockset System Models

- 1 Start Add-On Explorer using:

```
simrfSupportPackages
```
- 2 Click **SimRF Models for Analog Devices Rf Transceivers**.
- 3 Click **Install** and follow the instructions to install the models.
- 4 Complete the installation process. You can now access the following Analog Devices® models:

AD9361 models
AD9361 testbenches
AD9371 models
AD9371 testbenches

Note To run the models, you require these additional licenses:

- Communications Toolbox
- Stateflow®
- Fixed-Point Designer™
- DSP System Toolbox
- Simscape

When you update your MATLAB software, you must repeat the installation process for the latest updates. You can also check for updates between releases.

Version History

Introduced in R2014b

SpectrumAnalyzerConfiguration

Configure Spectrum Analyzer for programmatic access

Description

The `spbscopes.SpectrumAnalyzerConfiguration` object contains the scope configuration information for the Spectrum Analyzer block.

Creation

`MyScopeConfiguration = get_param(gcbh, 'ScopeConfiguration')` constructs a new spectrum analyzer configuration object. You must first select the block in the model or provide the full path to the block.

Properties

Frequently Used

NumInputPorts — Number of input ports

"1" (default) | character vector | string scalar

Number of input ports on a scope block, specified as a character vector or string scalar. The maximum number of input ports that you can specify is 96.

Scope Window Use

This parameter is available only from the **Property Inspector**. To launch the **Property Inspector**, click the **Modeling** tab of the Simulink model window that contains the Spectrum Analyzer block. In the **Design** section, click the drop-down arrow and select **Property Inspector**. The **Property Inspector** window opens on the right. If you select the Spectrum Analyzer block, the **Property Inspector** window shows the corresponding parameters. In the parameters that show up, specify the **Number of input ports** parameter.

Data Types: char | string

SpectrumType — Spectrum type

"Power" (default) | "Power density" | "RMS"

Spectrum type, specified as one of these:

"Power" — Power spectrum

"Power density" — Power spectral density. The power spectral density is the magnitude squared of the spectrum normalized to a bandwidth of 1 Hz.

"RMS" — Root mean square. The root mean square shows the square root of the mean square. Use this option to view the frequency of voltage or current signals.

Tunable: Yes

Scope Window Use

In the **Analyzer** tab on the Spectrum Analyzer toolstrip, select **Spectrum**. Click **Spectrum** to select Power, Power Density, or RMS.

Data Types: char | string

SampleRateSource — Source of input sample rate

"Inherited" (default) | "Property"

Source of the input sample rate, specified as one of these:

- "Inherited" — Spectrum Analyzer inherits the input sample rate from the model.
- "Property" — Specify the sample rate input directly using the `SampleRate` property.

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Bandwidth** section, set **Sample Rate (Hz)** to Inherited or specify a finite scalar.

Data Types: char | string

SampleRate — Sample rate of input

"10e3" (default) | character vector of a positive scalar | string scalar of a positive scalar

Sample rate of the input signal in Hz, specified as a character vector or a string scalar of a positive scalar.

Dependency

To enable this property, set `SampleRateSource` to "Property".

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Bandwidth** section, specify **Sample Rate (Hz)** to a finite scalar.

Data Types: char | string

PlotAsTwoSidedSpectrum — Option to plot a two-sided spectrum

false (default) | true

Option to plot a two-sided spectrum, specified as one of the following:

- `false` — Compute and plot one-sided spectral estimates. If you set this property to `false`, then the input signal must be real valued.

When you set this property to `false`, the Spectrum Analyzer uses power-folding. The y-axis values are twice the amplitude that they would be if you were to set this property to `true`, except at 0 and the Nyquist frequency. A one-sided power spectral density (PSD) contains the total power of the signal in the frequency interval from DC to half the Nyquist rate. For more information, see `pwelch`.

- `true` — Compute and plot two-sided spectral estimates. When the input signal is complex valued, you must set this property to `true`.

Scope Window Use

Click the **Spectrum** tab of the Spectrum Analyzer toolstrip. In the **Trace Options** section, select **Two-Sided Spectrum** to compute and plot two-sided spectral estimates.

Data Types: logical

FrequencyScale — Scale to display frequency

"Linear" (default) | "Log"

Scale to display frequency, specified as one of the following:

- "Linear" — Use a linear scale to display frequencies on the x-axis. To use the "Linear" option, you must also set the `PlotAsTwoSidedSpectrum` property to true.
- "Log" — Use a logarithmic scale to display frequencies on the x-axis. To use the "Log" option, you must also set the `PlotAsTwoSidedSpectrum` property to false.

Tunable: Yes

Scope Window Use

Click the **Spectrum** tab on the Spectrum Analyzer toolstrip. In the **Scale** section, set the **Frequency Scale** to Linear or Log.

To set the **Frequency Scale** to Log, clear the **Two-Sided Spectrum** check box in the **Trace Options** section in the **Spectrum** tab. If you select the **Two-Sided Spectrum** check box, then you must set the **Frequency Scale** to Linear.

Data Types: char | string

PlotType — Plot type to display normal traces

"Line" (default) | "Stem"

Plot type to display normal traces, specified as "Line" or "Stem". Normal traces are traces that display free-running spectral estimates.

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip, navigate to the **Configuration** section and click **Settings**. In the Spectrum Analyzer Settings window, under **Display and Labels**, set **Plot Type** to Line or Stem.

Data Types: char | string

AxesScaling — Axes scaling mode

"Auto" (default) | "Manual" | "OnceAtStop" | "Updates"

Axes scaling mode, specified as one of these:

- "Auto" — The scope scales the axes to fit the data, both during and after simulation.
- "Manual" — The scope does not scale the axes automatically.
- "OnceAtStop" — The scope scales the axes when the simulation stops.
- "Updates" — The scope scales the axes after a specific number of visual updates. It determines the number of updates using the `AxesScalingNumUpdates` property.

Tunable: Yes

Data Types: char | string

AxesScalingNumUpdates — Number of updates before scaling

"100" (default) | character vector of a positive integer | string scalar of a positive integer

Number of updates before scaling, specified as a character vector or a string scalar of a positive integer.

Tunable: Yes

Dependency

To enable this property, set `AxesScaling` to "Updates".

Data Types: char | string

Advanced

RBWSource — Source of resolution bandwidth value

"Auto" (default) | "Property" | "InputPort"

Source of the resolution bandwidth (RBW) value, specified as one of these:

- "Auto" — The Spectrum Analyzer adjusts the spectral estimation resolution to ensure that there are 1024 RBW intervals over the defined frequency span.
- "Property" — Specify the resolution bandwidth directly using the RBW property.
- "InputPort" — An input port appears on the Spectrum Analyzer block through which you specify the RBW. You can use this option only for frequency inputs.

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Bandwidth** section, set **RBW (Hz)** to Auto, Input port or to a positive scalar.

Data Types: char | string

RBW — Resolution bandwidth

"9.76" (default) | character vector of a positive scalar | string scalar of a positive scalar

Resolution bandwidth (RBW) in Hz, specified as a character vector or a string scalar of a positive scalar. Specify the value to ensure that there are at least two RBW intervals over the specified frequency span. The ratio of the overall span to RBW satisfies this condition:

$$\frac{span}{RBW} > 2$$

You can specify the overall span in different ways based on how you set the `FrequencySpan` property.

RBW controls the spectral resolution of the displayed signal. The RBW value determines the spacing between frequencies that can be resolved. A smaller value gives a higher spectral resolution and lowers the noise floor, that is, the Spectrum Analyzer can resolve frequencies that are closer to each other. However, this comes at the cost of a longer sweep time.

Tunable: Yes

Dependency

To enable this property, set `RBWSource` to "Property".

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Bandwidth** section, set **RBW (Hz)** to a positive scalar.

Data Types: char | string

OverlapPercent — Overlap percentage

"0" (default) | character vector of a real scalar in the range [0 100] | string scalar of a real scalar in the range [0 100]

Overlap percentage between the previous and current buffered data segments, specified as a character vector or string scalar of a real scalar in the range [0 100]. The overlap creates a window segment that the Spectrum Analyzer uses to compute a spectral estimate.

Tunable: Yes

Scope Window Use

Click the **Estimation** tab on the Spectrum Analyzer toolstrip. In the **Window Options** section, set the **Overlap (%)**.

Data Types: char | string

Window — Window function

"Hann" (default) | "Rectangular"

Specify a window function for the spectral estimation. You can select one of the window options in the following table. For more information on the window functions, click the link to the function reference page in the Signal Processing Toolbox documentation.

Window Option	Corresponding Signal Processing Toolbox Function
"Rectangular"	rectwin
"Hann"	hann

Scope Window Use

Click the **Estimation** tab on the Spectrum Analyzer toolstrip. In the **Window Options** section, set the **Window**.

Data Types: char | string

AveragingMethod — Smoothing method

"Exponential" (default) | "VBW"

Averaging method, specified as one of the following:

- "Exponential" — Weighted average of samples. The object computes the average over samples weighted by an exponentially decaying forgetting factor. Use the `ForgettingFactor` property to specify the weighted forgetting factor.

- "VBW" — Video bandwidth method. The object uses a lowpass filter to smooth the trace and decrease noise. Use the VBWSource and VBW properties to specify the VBW value.

For more information, see "Averaging Method" on page 1-165.

Tunable: Yes

Scope Window Use

Click the **Estimation** tab on the Spectrum Analyzer toolstrip. In the **Averaging** section, set **Averaging Method** to VBW or Exponential.

Data Types: char | string

ForgettingFactor — Weighting forgetting factor

"0.9" (default) | string scalar of scalar in the range (0,1] | character vector of scalar in the range (0,1]

Forgetting factor of the exponential weighted averaging method, specified as a character vector or string scalar of a real scalar in the range (0,1].

Tunable: Yes

Dependency

To enable this property, set AveragingMethod to "Exponential".

Scope Window Use

Click the **Estimation** tab on the Spectrum Analyzer toolstrip. In the **Averaging** section, adjust the **Forgetting Factor** slider.

To enable the **Forgetting Factor**, set **Averaging Method** to Exponential.

Data Types: char | string

VBWSource — Source of video bandwidth

"Auto" (default) | "Input port" | "Property"

Source of the video bandwidth (VBW), specified as one of these:

- "Auto" — The Spectrum Analyzer adjusts the VBW such that the equivalent forgetting factor is 0.9.
- "Input port" -- An input port appears on the scope and you can specify the VBW value through this port.
- "Property" — The Spectrum Analyzer adjusts the VBW using the value specified in the VBW property.

For more details on the video bandwidth method, see "Averaging Method" on page 1-165.

Tunable: Yes

Dependency

To enable this property, set AveragingMethod to "VBW".

Scope Window Use

Click the **Estimation** tab on the Spectrum Analyzer toolstrip. In the **Averaging** section, set **VBW (Hz)** to either Auto or a positive real scalar less than or equal to **Sample Rate (Hz)/2**.

To enable the **VBW (Hz)**, set **Averaging Method** to VBW.

Data Types: char | string

VBW – Video bandwidth

"9.76" (default) | character vector of a positive scalar | string scalar of a positive scalar

Video bandwidth, specified as a character vector or a string scalar of a positive scalar less than or equal to $\text{SampleRate}/2$. For more information on the video bandwidth method, see "Averaging Method" on page 1-165.

Tunable: Yes

Dependency

To enable this property, set VBWSource to "Property".

Scope Window Use

Click the **Estimation** tab on the Spectrum Analyzer toolstrip. In the **Averaging** section, set **VBW (Hz)** to Auto or enter a positive real scalar that is less than or equal to **Sample Rate (Hz)/2**.

To enable the **VBW (Hz)**, set **Averaging Method** to VBW.

Data Types: double

SpectrumUnits – Units of the spectrum

"dBm" (default) | "dBm/Hz" | "dBV" | "Vrms"

Units of the spectrum, specified as one of these:

- "dBm"
- "dBm/Hz"
- "dBV"
- "Vrms"

The Spectrum Analyzer displays power values in the units you select in this property.

Tunable: Yes

Dependency

The available spectrum units depend on the value you specify in the SpectrumType property.

SpectrumType	Allowed SpectrumUnits
"Power"	"dBm"
"Power density"	"dBm/Hz"
"RMS"	"dBV", "Vrms"

Scope Window Use

Click the **Spectrum** tab on the Spectrum Analyzer toolstrip. In the **Scale** section, set **Spectrum Unit**.

Data Types: char | string

ReferenceLoad — Reference load to compute power levels

"1" (default) | character vector of a real positive scalar | string scalar of a real positive scalar

Reference load to compute the power levels, specified as a character vector or a string scalar of a positive scalar in Ohms.

Tunable: Yes

Scope Window Use

Click the **Spectrum** tab on the Spectrum Analyzer toolstrip. In the **Scale** section, set **Reference Load (Ω)**.

Data Types: char | string

FrequencyOffset — Frequency offset

"0" (default) | numeric scalar character vector | numeric vector character vector | numeric scalar string scalar | numeric vector string scalar

Frequency offset, specified as one of these:

- Numeric scalar — Apply the same frequency offset to all channels, specified as a character vector or a string scalar in Hz.
- Numeric vector — Apply a specific frequency offset for each channel, specified as a character vector or a string scalar in Hz. The vector length must be equal to the number of input channels.

The frequency-axis values are offset by the values you specify in this property. The overall span must fall within the "Nyquist Frequency Interval" on page 1-163.

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Bandwidth** section, set **Offset (Hz)**.

Data Types: char | string

TreatMby1SignalsAsOneChannel — Option to treat unoriented input signal as a column vector

true (default) | false

Option to treat unoriented input signal as a column vector, specified as true or false. Set this property to true to treat M -by-1 and unoriented inputs as a column vector or one channel. Set this property to false to treat M -by-1 and unoriented inputs as a 1-by- M row vector.

Data Types: logical

Measurements

MeasurementChannel — Channel for which to obtain measurements

"1" (default) | character vector | string scalar

Channel over which the measurements are obtained, specified as a character vector or a string scalar which evaluates to a positive integer less than or equal to 100. The highest number you can specify is equal to the number of channels (columns) in the input signal.

Tunable: Yes

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip. In the **Channel** section, select a **Channel**.

Data Types: char | string

CursorMeasurements — Cursor measurements

CursorMeasurementsConfiguration object

Cursor measurements, specified as a CursorMeasurementsConfiguration object. Enable cursor measurements to display waveform cursors. All CursorMeasurementsConfiguration properties are tunable.

Tunable: Yes

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip and modify the cursor measurements in the **Cursors** section.

DistortionMeasurements — Distortion measurements

DistortionMeasurementsConfiguration object

Distortion measurements, specified as a DistortionMeasurementsConfiguration object. Enable distortion measurements to compute and display the harmonic distortion and intermodulation distortion. All DistortionMeasurementsConfiguration properties are tunable.

Tunable: Yes

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip and modify the distortion measurements in the **Distortion** section.

PeakFinder — Peak finder measurement

PeakFinderConfiguration object

Peak finder measurement, specified as a PeakFinderConfiguration object. Enable peak finder to compute and display the largest calculated peak values. All PeakFinderConfiguration properties are tunable.

Tunable: Yes

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip and modify the peak finder measurements in the **Peaks** section.

Visualization**Name — Caption to display in spectrum Analyzer window**

"Spectrum Analyzer" (default) | character vector | string scalar

Caption to display in the scope window, specified as a character vector or string scalar.

Tunable: Yes

Data Types: char | string

Position — Window position

screen center (default) | [left bottom width height]

Spectrum Analyzer window position in pixels, specified by the size and location of the scope window as a four-element double vector of the form [left bottom width height]. You can place the scope window in a specific position on your screen by modifying the values to this property.

By default, the window appears in the center of your screen with a width of **800** pixels and height of **450** pixels. The exact center coordinates depend on your screen resolution.

Tunable: Yes

MaximizeAxes — Maximize axes control

"Auto" (default) | "On" | "Off"

Maximize axes control, specified as one of the following:

- "Auto" -- The Spectrum Analyzer maximizes axes only if the display does not contain any labels or title annotations.
- "On" -- The Spectrum Analyzer maximizes axes in all displays.
- "Off" -- The Spectrum Analyzer does not maximize axes in any display.

Tunable: Yes

Scope Window Use

Click the  control on the Spectrum Analyzer display to maximize the axes.

Data Types: char | string

Title — Display title

' ' (default) | character vector | string scalar

Display title, specified as a character vector or a string scalar.

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Configuration** section, click **Settings**. In the Spectrum Analyzer Settings window that opens up, under **Display and labels**, enter **Title**.

Data Types: char | string

YLabel — Y-axis label

' ' (default) | character vector | string scalar

y-axis label, specified as a character vector or a string scalar. The Spectrum Analyzer displays the label to the left of the y-axis.

Regardless of the value of this property, Spectrum Analyzer always displays power units as one of the `SpectrumUnits` values.

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Configuration** section, click **Settings**. In the Spectrum Analyzer Settings window that opens up, under **Display and labels**, enter **Y-Label**.

Data Types: char | string

YLimits — Y-axis limits

[-80, 20] (default) | [ymin ymax]

y-axis limits, specified as a two-element numeric vector of the form [ymin ymax]. The units of the y-axis limits depend on the `SpectrumUnits` property.

Example: `scope.YLimits = [-10,20]`

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Configuration** section, click **Settings**. In the Spectrum Analyzer Settings window that opens up, under **Display and Labels**, enter **Y-Axis Limits**.

ShowGrid — Flag to show grid

true (default) | false

Flag to show the grid, specified as true or false. Set this property to true to show grid lines in the plot.

Tunable: Yes

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. In the **Configuration** section, click **Settings**. In the Spectrum Analyzer Settings window that appears, under **Display and Labels**, select **Show Grid**.

Data Types: logical

ChannelNames — Channel names

empty cell (default) | cell array of character vectors | array of strings

Channel names in the input data, specified as a cell array of character vectors or an array of strings. The names you specify in this property appear in the following locations:

- Legend
- **Spectrum Analyzer Settings > Color and styling** section
- **Measurements** and **Channel Measurements** tabs

If you do not specify channel names, the Spectrum Analyzer names the channels as Channel 1, Channel 2, and so on.

Tunable: Yes**Dependency**

To see the channel names, set ShowLegend to true.

Scope Window Use

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. To see the legend, click **Legend** in the **Configuration** section.

Data Types: char

ShowLegend — Flag to show legend

false (default) | true

Flag to show the legend, specified as true or false. To show a legend with the input names, set this property to true.

Use the legend to control which signals are visible. In the scope legend, click a signal name to hide the signal in the scope. To show the signal, click the signal name again. To show only one signal, right-click the signal name. To show all signals, press **Esc**.

Tunable: Yes**Scope Window Use**

Click the **Analyzer** tab on the Spectrum Analyzer toolstrip. To see the legend, click **Legend** in the **Configuration** section.

Data Types: logical

OpenAtSimulationStart — Flag to open scope when simulation starts

true (default) | false

Flag to open scope when simulation starts, specified as true or false.

Set this property to true to open the scope when the simulation starts. Set this property to false to prevent the scope from opening when you simulate the model.

Scope Window Use

This parameter is available only from the **Property Inspector**. To launch the **Property Inspector**, click the **Modeling** tab of the Simulink model window that contains the Spectrum Analyzer block. In

the **Design** section, click the drop-down arrow and select **Property Inspector**. The **Property Inspector** window opens on the right. If you select the Spectrum Analyzer block, the **Property Inspector** window shows the corresponding parameters. In the parameters that show up, select or clear the **Open at simulation start** parameter.

Data Types: logical

Visible – Visibility of the Spectrum Analyzer

false (default) | true

Set this property to `true` to display the spectrum analyzer window, and to `false` to hide the spectrum analyzer window.

Data Types: logical

Examples

Construct Spectrum Analyzer Configuration Object

Create the configuration object for a Spectrum Analyzer block.

Create a new Simulink® model with a randomly generated name.

```
sysname=char(randi(26,1,7)+96);
new_system(sysname);
```

Add a new Spectrum Analyzer block to the model.

```
add_block('built-in/SpectrumAnalyzer',[sysname,'/SpectrumAnalyzer'])
```

Call the `get_param` function to retrieve the default Spectrum Analyzer block configuration properties.

```
config = get_param([sysname,'/SpectrumAnalyzer'],'ScopeConfiguration')
```

```
config =
```

```
  SpectrumAnalyzerBlockConfiguration with properties:
```

```
      NumInputPorts: '1'
      SpectrumType: 'Power'
      SampleRateSource: 'Inherited'
      PlotAsTwoSidedSpectrum: 1
      FrequencyScale: 'Linear'
      PlotType: 'Line'
      AxesScaling: 'Auto'
```

```
  Advanced
```

```
      RBWSource: 'Auto'
      AveragingMethod: 'VBW'
      VBWSource: 'Auto'
      SpectrumUnits: 'dBm'
      ReferenceLoad: '1'
      FrequencyOffset: '0'
```

```
  TreatMby1SignalsAsOneChannel: 1
```

```
  Spectrogram
```

```
  No properties.
```

```
  Measurements
```

```
      MeasurementChannel: '1'
      CursorMeasurements: [1x1 CursorMeasurementsConfiguration]
      DistortionMeasurements: [1x1 DistortionMeasurementsConfiguration]
      PeakFinder: [1x1 PeakFinderConfiguration]
```

```
Visualization
    Name: 'SpectrumAnalyzer'
    Position: [880 470 800 500]
    MaximizeAxes: 'Auto'
    Title: ''
    YLabel: ''
    YLimits: [-80 20]
    ShowGrid: 1
    ChannelNames: {''}
    ShowLegend: 0
    OpenAtSimulationStart: 1
    Visible: 0
```

Version History

Introduced in R2016b

R2023a: Spectrum Analyzer with better responsiveness and toolstrip interface in Simulink

In R2023a, the Spectrum Analyzer block comes with better responsiveness and a toolstrip interface that gives you easy access to spectral analysis, estimation, and measurements.

See Also

Blocks

Spectrum Analyzer

Topics

“Control Scope Blocks Programmatically”

CursorMeasurementsConfiguration

Measure signal values using vertical waveform cursors that track along the signal

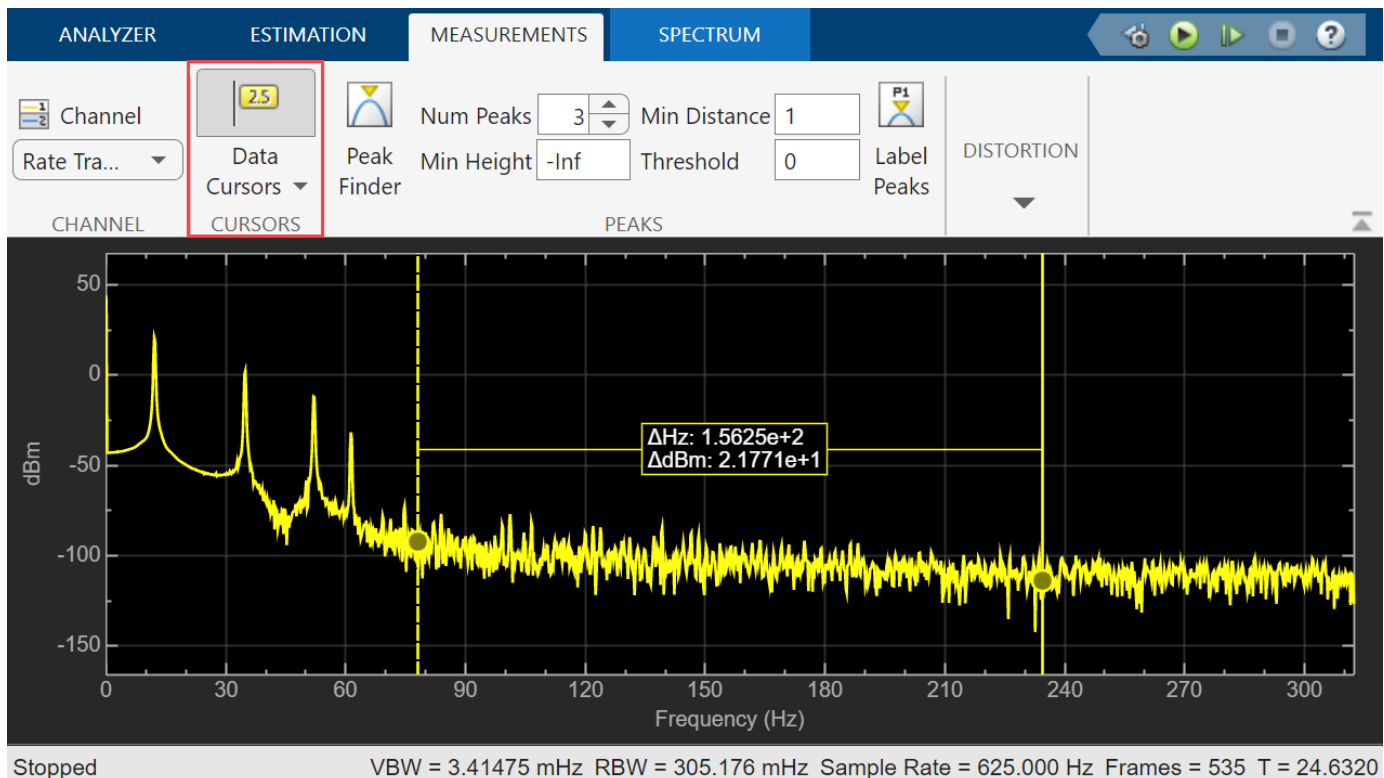
Description

Use the `CursorMeasurementsConfiguration` object to enable waveform cursors. You can control the cursor settings from the toolbar of the scope or from the command line.

To display the vertical cursors on each signal and to modify the cursor settings in the scope UI, click the **Measurements** tab and enable **Data Cursors**. Each cursor tracks a vertical line along the signal. The scope displays the difference between x- and y-values of the signal at the two cursors in the box between the cursors. The cursors appear only when the Spectrum Analyzer has at least one signal in its display. You can use the mouse to move the vertical cursors left and right.

The `SpectrumAnalyzerConfiguration` object supports the `CursorMeasurementsConfiguration` object in the command line.

Spectrum Analyzer Toolstrip



Creation

Syntax

```
cursormeas = CursorMeasurementsConfiguration()
```

Description

`cursormeas = CursorMeasurementsConfiguration()` creates a cursor measurements configuration object.

Properties

All properties are tunable.

XLocation — x-coordinates of the cursors

[2 8] (default) | two-element vector

x-coordinates of the cursors, specified as a two-element vector of real numbers.

Scope Window Use

On the **Measurements** tab, select **Data Cursors** and then click **Data Cursors** to specify the two elements in X location properties.

Data Types: `double`

SnapToData — Position cursors on signal data points

`false` (default) | `true`

Position cursors on the signal data points, specified as `true` or `false`.

Scope Window Use

On the **Measurements** tab, select **Data Cursors** and then click **Data Cursors** to select **Snap to data**.

Data Types: `logical`

LockSpacing — Lock spacing between cursors

`false` (default) | `true`

Lock spacing between cursors, specified as `true` or `false`. Set this property to `true` to lock the frequency difference between the cursors.

Scope Window Use

On the **Measurements** tab, select **Data Cursors** and then click **Data Cursors** to select **Lock cursor spacing**.

Data Types: `logical`

Enabled — Enable cursor measurements

`false` (default) | `true`

Enable cursor measurements, specified as `true` or `false`. Set this property to `true` to enable cursor measurements.

Scope Window Use

On the **Measurements** tab, select **Data Cursors**.

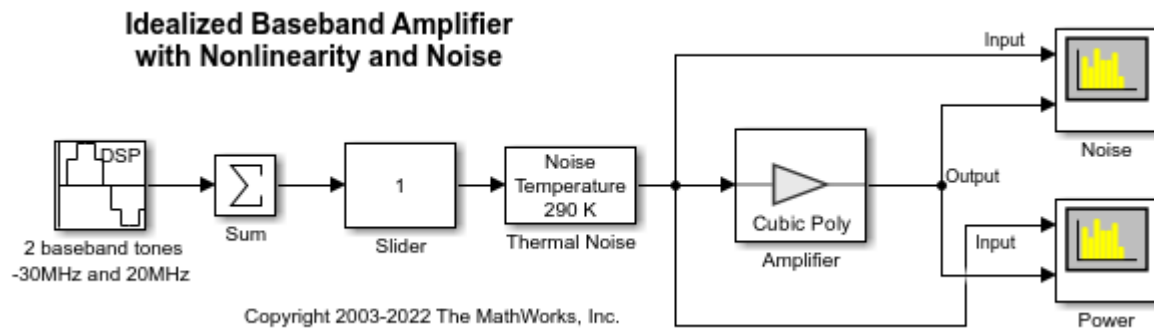
Data Types: `logical`

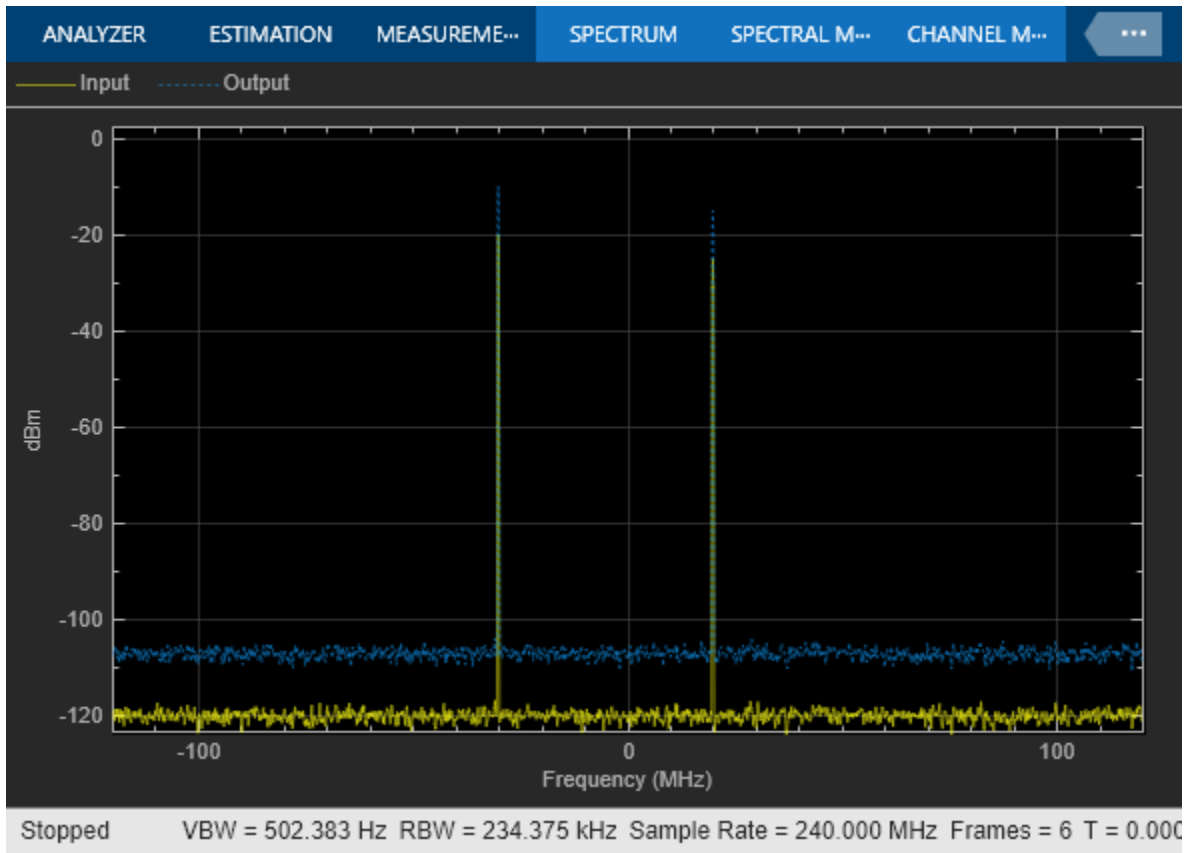
Examples

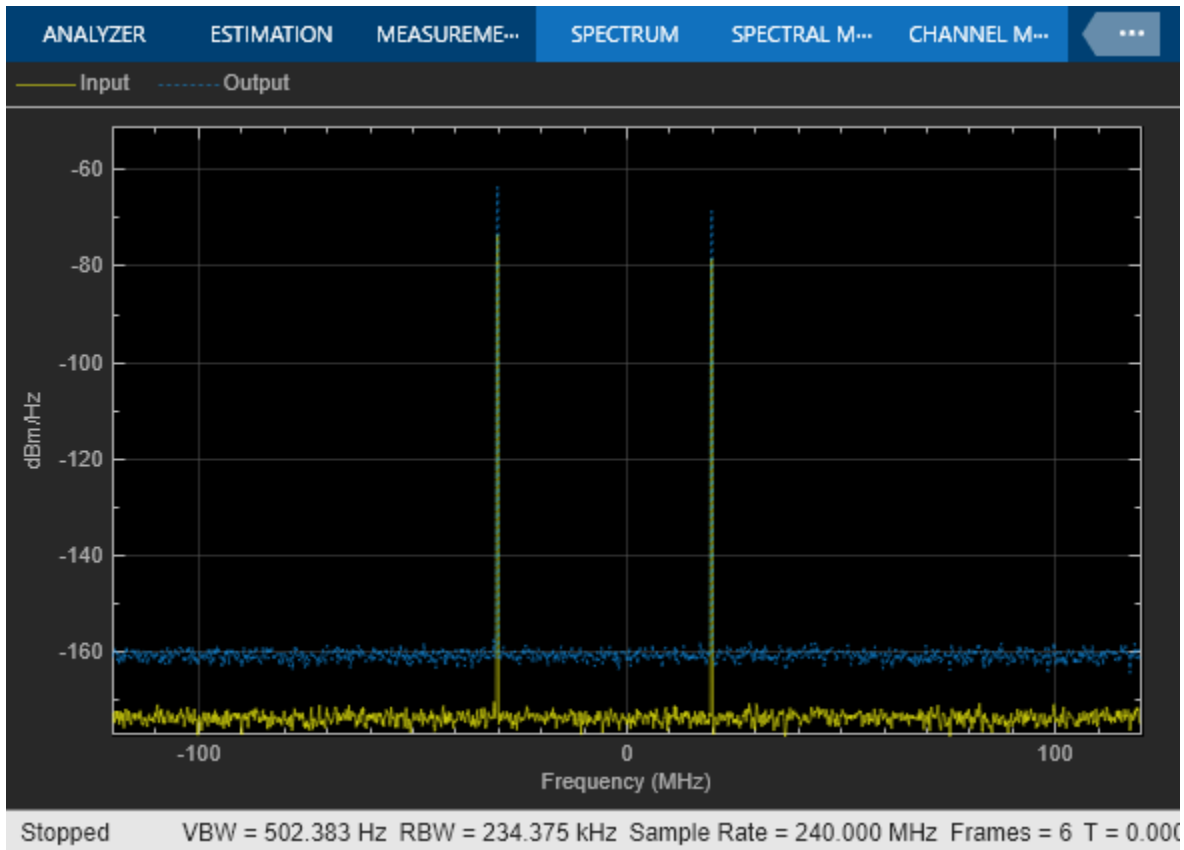
Obtain Measurements Data Programmatically for Spectrum Analyzer Block

Open and run the signal amplifier model.

```
open_system('amplifySigNoise.slx')  
sim('amplifySigNoise.slx')
```







The Spectrum Analyzer blocks in the model show the amplified noise and the signal power plots, respectively. For more details on this model, see the “Analysis of Frequency Response of RF System” example.

You can obtain or modify the peak, cursor, and distortion measurements in the spectra programmatically by enabling these properties in the `SpectrumAnalyzerConfiguration` object:

- `PeakFinder`
- `CursorMeasurements`
- `DistortionMeasurements`

Access the configuration properties of the Spectrum Analyzer block using the `get_param` function.

```
sablockNoise = 'amplifySigNoise/Noise';
cfgNoise = get_param(sablockNoise, 'ScopeConfiguration');
sablockPwr = 'amplifySigNoise/Power';
cfgPwr = get_param(sablockPwr, 'ScopeConfiguration');
```

To enable these measurements, set the `Enabled` property to `true`.

```
cfgNoise.CursorMeasurements.Enabled = true;
cfgNoise.PeakFinder.Enabled = true;
cfgNoise.DistortionMeasurements.Enabled = true;
cfgPwr.CursorMeasurements.Enabled = true;
cfgPwr.PeakFinder.Enabled = true;
cfgPwr.DistortionMeasurements.Enabled = true;
```

You can now modify these measurements programmatically. As an example, display the first two peaks on the spectrum display. By default, the spectrum shows three peaks. To modify the number of peaks it shows, change `NumPeaks` to 2. You can even label the peaks using the `LabelPeaks` property and control the minimum distance between adjacent peaks using the `MinDistance` property. The changes you make to these properties update the spectra accordingly.

```
cfgNoise.PeakFinder.NumPeaks = 2;  
cfgPwr.PeakFinder.NumPeaks = 2;
```

Version History

Introduced in R2023a

See Also

Blocks

Spectrum Analyzer

Objects

SpectrumAnalyzerConfiguration

DistortionMeasurementsConfiguration

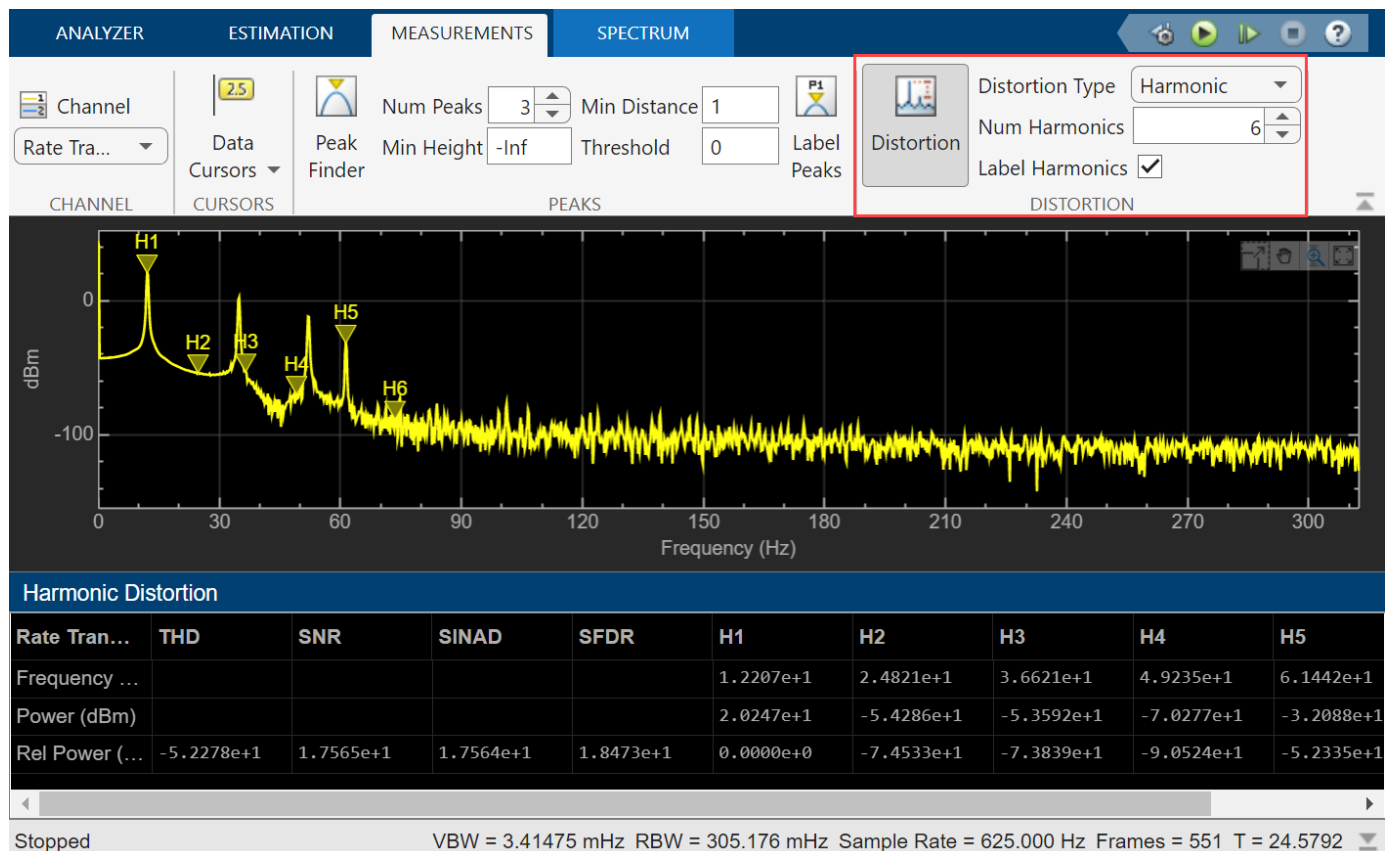
Compute and display harmonic and intermodulation distortion

Description

Use the `DistortionMeasurementsConfiguration` object to compute and display harmonic and intermodulation distortion.

You can specify the distortion type, number of harmonics, and label the harmonics. You can control the distortion settings from the Spectrum Analyzer toolstrip or from the command line. The `SpectrumAnalyzerConfiguration` object supports the `DistortionMeasurementsConfiguration` object in the command line.

To modify the distortion settings in the Spectrum Analyzer toolstrip, click the **Measurements** tab and edit the settings in the **Distortion** section.



Creation

Syntax

```
distmeas = DistortionMeasurementsConfiguration()
```

Description

`distmeas = DistortionMeasurementsConfiguration()` creates a distortion measurements configuration object `distmeas`.

Properties

All properties are tunable.

Type — Type of measurement to display

"harmonic" (default) | "intermodulation"

Type of measurement data to display, specified as "harmonic" or "intermodulation".

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip. In the **Distortion** section, set **Distortion Type** to Harmonic or Intermodulation.

The **Measurements** tab appears when you select **Spectrum** in the **Analyzer** tab.

Data Types: char | string

NumHarmonics — Number of harmonics to measure

6 (default) | positive integer

Number of harmonics to measure, specified as a positive integer less than or equal to 99.

Dependency

To enable this property, set Type to "harmonic".

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip. In the **Distortion** section, set **Num Harmonics** to a positive integer less than or equal to 99.

The **Measurements** tab appears when you select **Spectrum** in the **Analyzer** tab.

Data Types: double

LabelValues — Label values

false (default) | true

Label harmonics (for harmonic distortion) and frequencies (for intermodulation distortion), specified as true or false.

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip. In the **Distortion** section, select **Label Harmonics**.

The **Measurements** tab appears when you select **Spectrum** in the **Analyzer** tab.

Data Types: `logical`

Enabled — Enable distortion measurements

`false` (default) | `true`

Enable distortion measurements, specified as `true` or `false`.

Scope Window Use

Click the **Measurements** tab on the Spectrum Analyzer toolstrip. In the **Distortion** section, select **Distortion**.

The **Measurements** tab appears when you select **Spectrum** in the **Analyzer** tab.

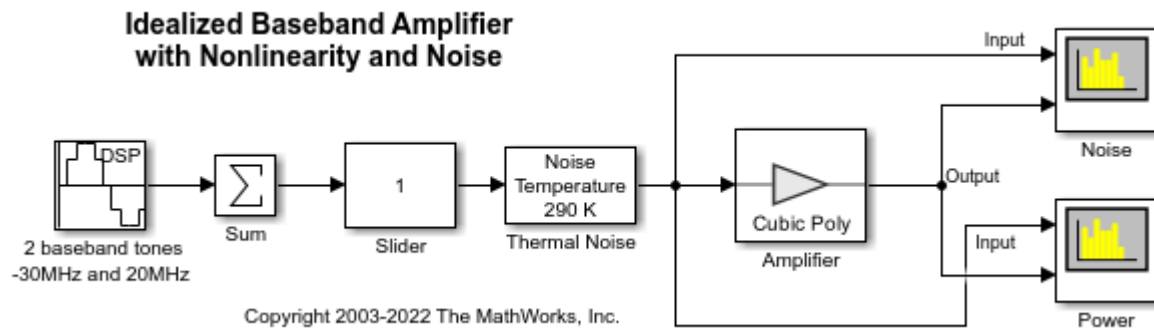
Data Types: `logical`

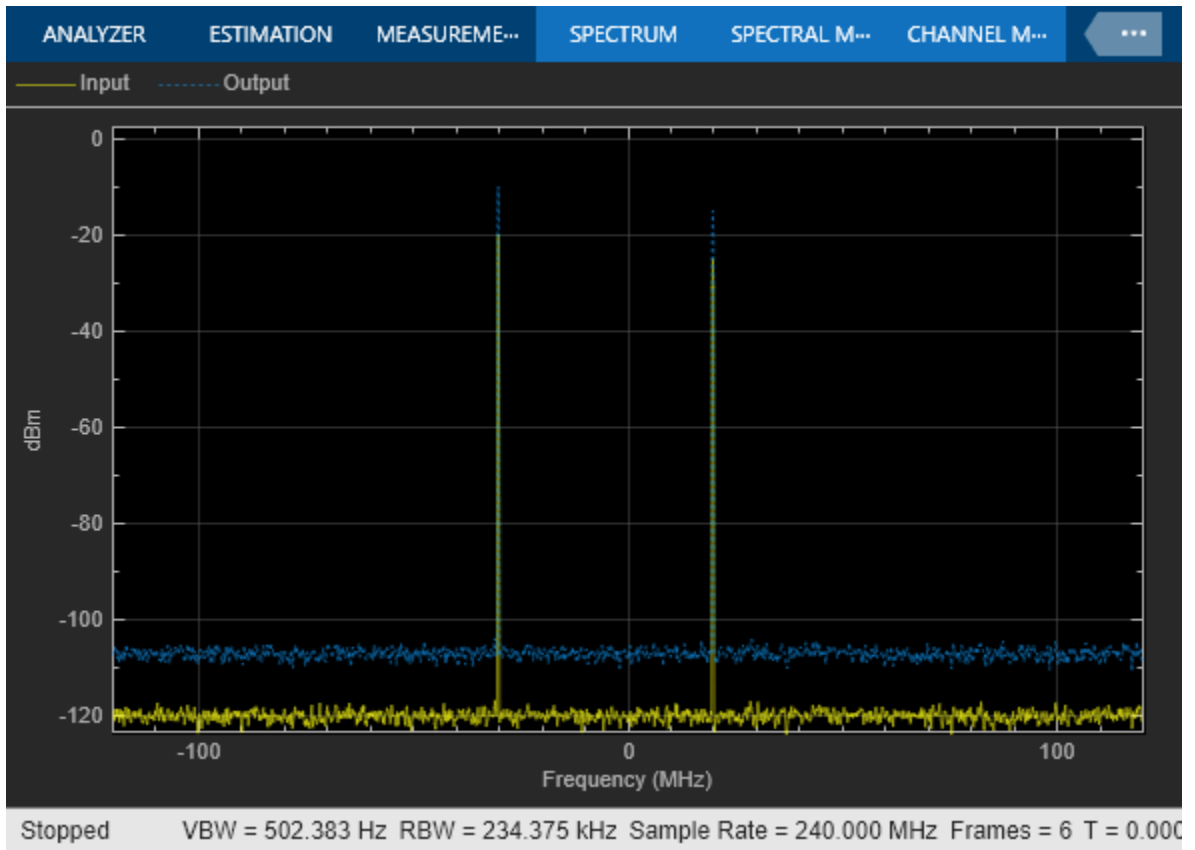
Examples

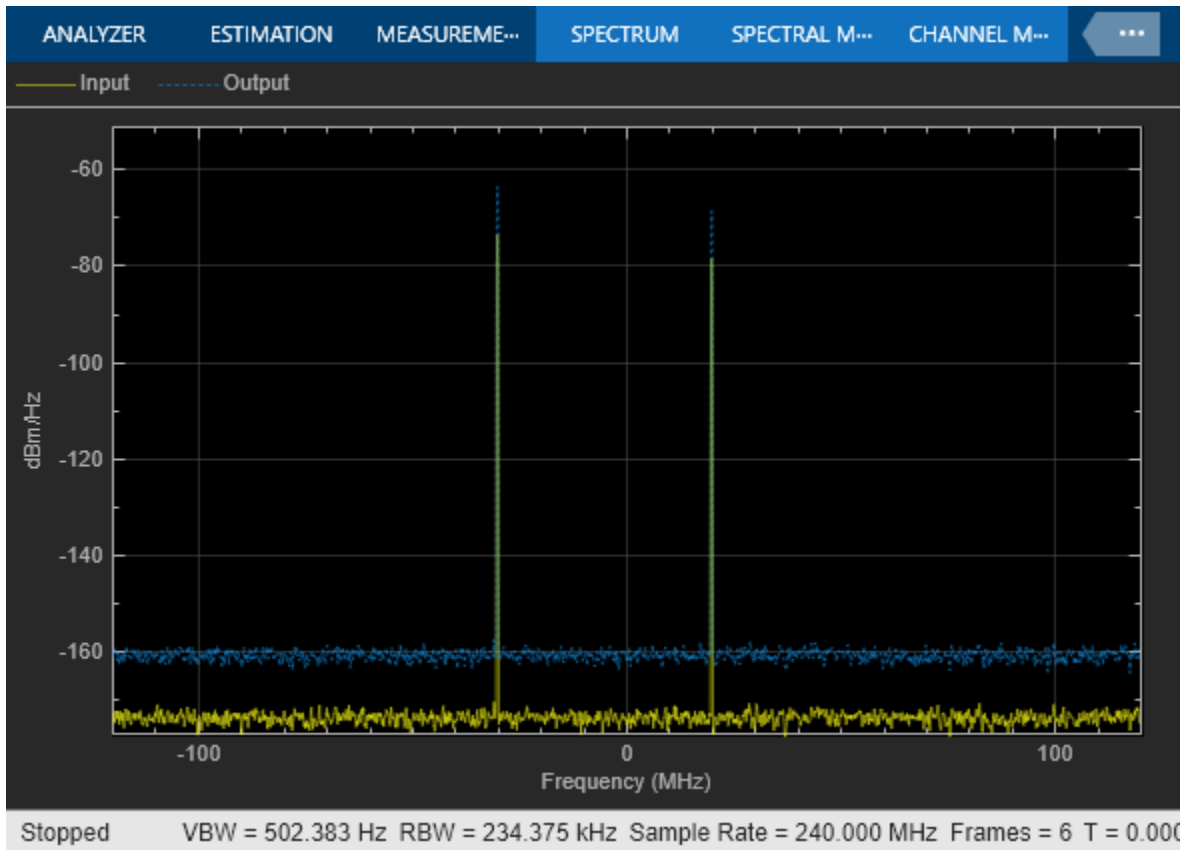
Obtain Measurements Data Programmatically for Spectrum Analyzer Block

Open and run the signal amplifier model.

```
open_system('amplifySigNoise.slx')
sim('amplifySigNoise.slx')
```







The Spectrum Analyzer blocks in the model show the amplified noise and the signal power plots, respectively. For more details on this model, see the “Analysis of Frequency Response of RF System” example.

You can obtain or modify the peak, cursor, and distortion measurements in the spectra programmatically by enabling these properties in the `SpectrumAnalyzerConfiguration` object:

- `PeakFinder`
- `CursorMeasurements`
- `DistortionMeasurements`

Access the configuration properties of the Spectrum Analyzer block using the `get_param` function.

```
sablockNoise = 'amplifySigNoise/Noise';
cfgNoise = get_param(sablockNoise, 'ScopeConfiguration');
sablockPwr = 'amplifySigNoise/Power';
cfgPwr = get_param(sablockPwr, 'ScopeConfiguration');
```

To enable these measurements, set the `Enabled` property to `true`.

```
cfgNoise.CursorMeasurements.Enabled = true;
cfgNoise.PeakFinder.Enabled = true;
cfgNoise.DistortionMeasurements.Enabled = true;
cfgPwr.CursorMeasurements.Enabled = true;
cfgPwr.PeakFinder.Enabled = true;
cfgPwr.DistortionMeasurements.Enabled = true;
```

You can now modify these measurements programmatically. As an example, display the first two peaks on the spectrum display. By default, the spectrum shows three peaks. To modify the number of peaks it shows, change `NumPeaks` to 2. You can even label the peaks using the `LabelPeaks` property and control the minimum distance between adjacent peaks using the `MinDistance` property. The changes you make to these properties update the spectra accordingly.

```
cfgNoise.PeakFinder.NumPeaks = 2;  
cfgPwr.PeakFinder.NumPeaks = 2;
```

Version History

Introduced in R2023a

See Also

Blocks

Spectrum Analyzer

Objects

SpectrumAnalyzerConfiguration

PeakFinderConfiguration

Compute and display the largest calculated peak values on the scope display

Description

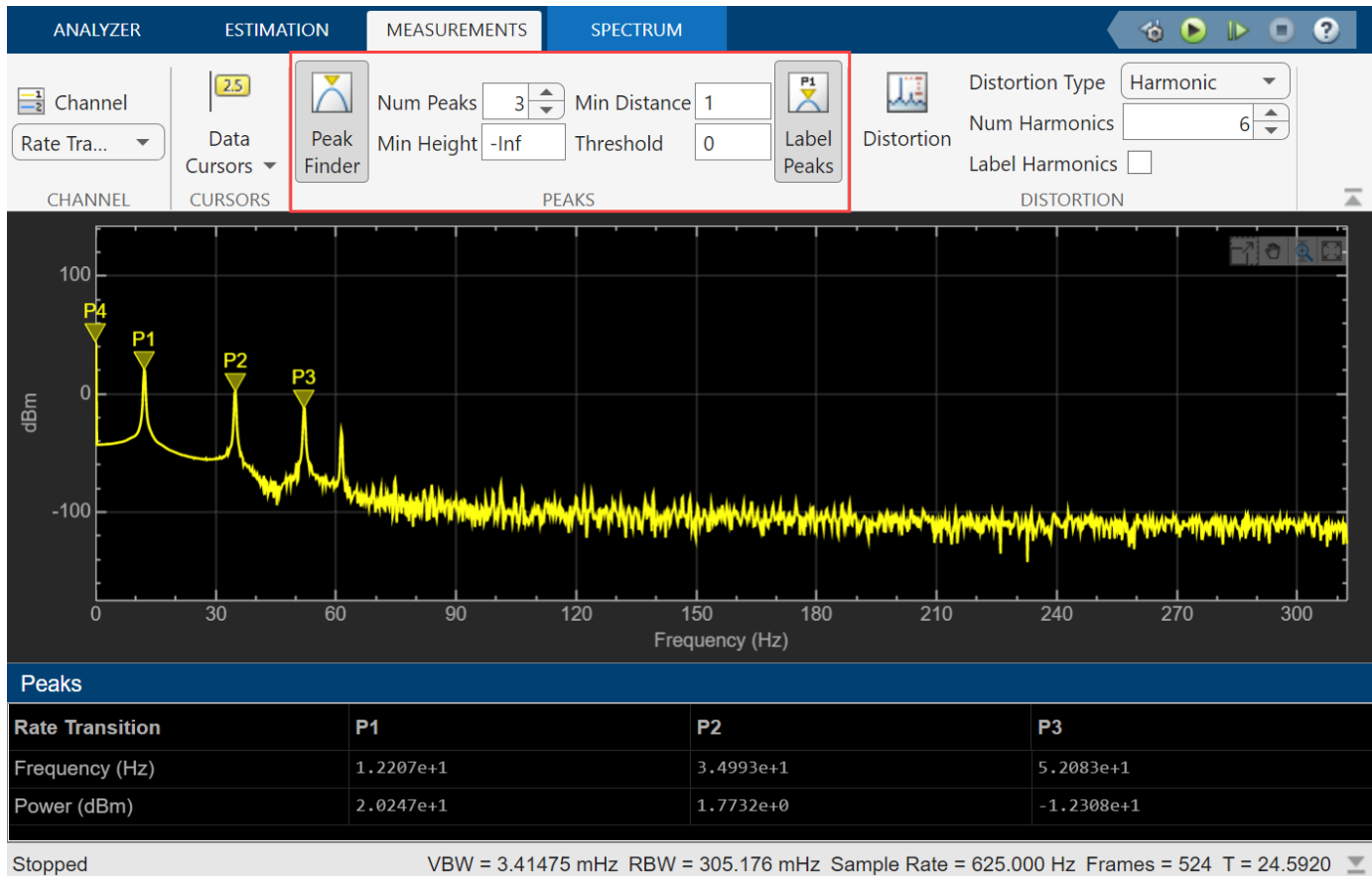
Use the `PeakFinderConfiguration` object to compute and display peaks in the scope. The scope computes and displays peaks for only the portion of the input signal that is currently on display in the scope.

You can specify the number of peaks you want the scope to display, the minimum height above which you want the scope to detect peaks, the minimum distance between peaks, and label the peaks. You can control the peak finder settings from the scope toolstrip or from the command line. The algorithm defines a peak as a local maximum with lower values present on either side of the peak. It does not consider end points as peaks. For more information on the algorithm, see the `findpeaks` function.

To modify the peak finder settings in the scope interface, click the **Measurements** tab and enable **Peak Finder** in the **Peaks** section. Once you enable the **Peak Finder**, an arrow appears on the plot at each maxima and a Peaks panel appears at the bottom of the scope window.

The `SpectrumAnalyzerConfiguration` object supports the `PeakFinderConfiguration` object in the command line.

Spectrum Analyzer Toolstrip



Creation

Syntax

```
pkfinder = PeakFinderConfiguration()
```

Description

`pkfinder = PeakFinderConfiguration()` creates a peak finder configuration object.

Properties

All properties are tunable.

MinHeight — Level above which scope detects peaks

-Inf (default) | real scalar value

Level above which the scope detects peaks, specified as a real scalar.

Scope Window Use

On the **Measurements** tab, select **Peak Finder**. In the peak finder settings, specify a real scalar in the **Min Height** box.

Data Types: double

NumPeaks — Maximum number of peaks to show

3 (default) | positive integer less than 100

Maximum number of peaks to show, specified as a positive integer less than 100.

Scope Window Use

On the **Measurements** tab, select **Peak Finder**. In the peak finder settings, specify a positive integer less than 100 in the **Num Peaks** box.

Data Types: double

MinDistance — Minimum number of samples between adjacent peaks

1 (default) | positive integer

Minimum number of samples between adjacent peaks, specified as a positive integer.

Scope Window Use

On the **Measurements** tab, select **Peak Finder**. In the peak finder settings, specify a positive integer in the **Min Distance** box.

Data Types: double

Threshold — Minimum difference in height of peak and its neighboring samples

0 (default) | nonnegative scalar

Minimum difference in the height of the peak and its neighboring samples, specified as a nonnegative scalar.

Scope Window Use

On the **Measurements** tab, select **Peak Finder**. In the peak finder settings, specify a nonnegative scalar in the **Threshold** box.

Data Types: double

LabelPeaks — Label found peaks

false (default) | true

Label found peaks, specified as true or false. The scope displays the labels (**P1**, **P2**, ...) above the arrows in the plot.

Scope Window Use

On the **Measurements** tab, select **Peak Finder**. In the peak finder settings, select **Label Peaks**.

Data Types: logical

LabelFormat — Coordinates to display

"x + y" (default) | "x" | "y"

Coordinates to display next to the calculated peak value, specified as "x", "y", or "x + y".

Data Types: char | string

Enabled — Enable peak finder measurements

false (default) | true

Enable peak finder measurements, specified as `true` or `false`. Set this property to `true` to enable the peak finder measurements.

Scope Window Use

On the **Measurements** tab, select **Peak Finder**.

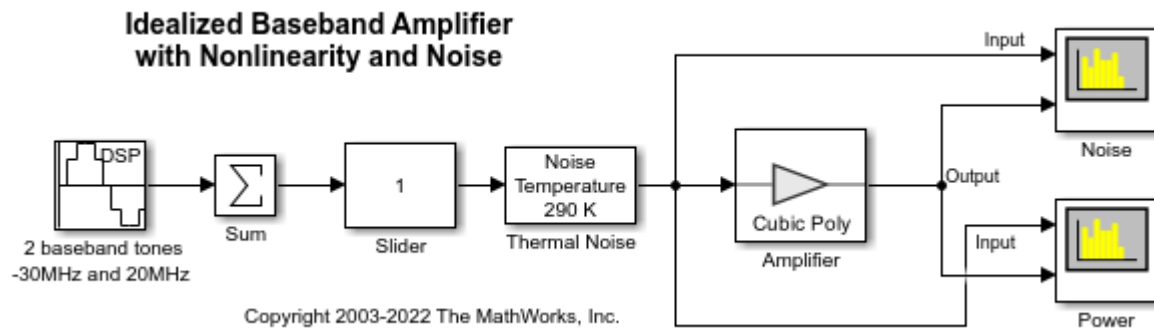
Data Types: `logical`

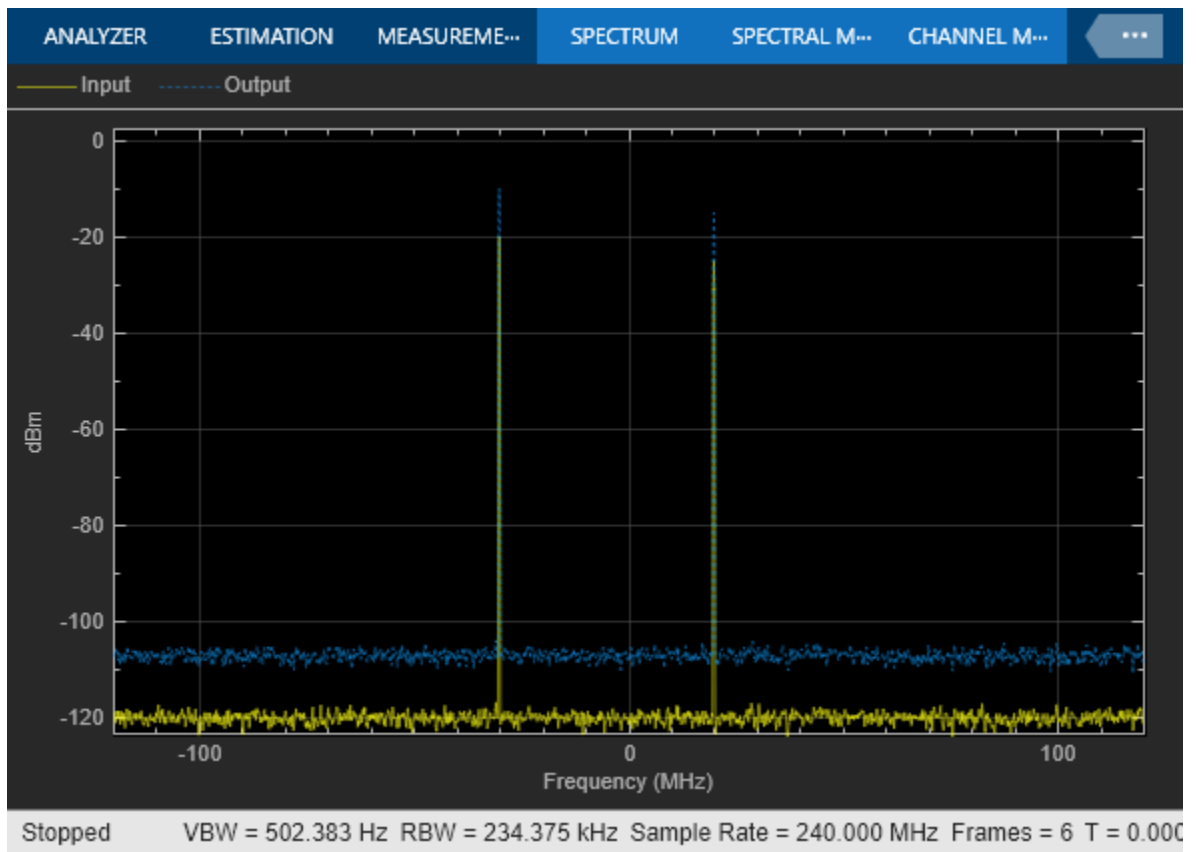
Examples

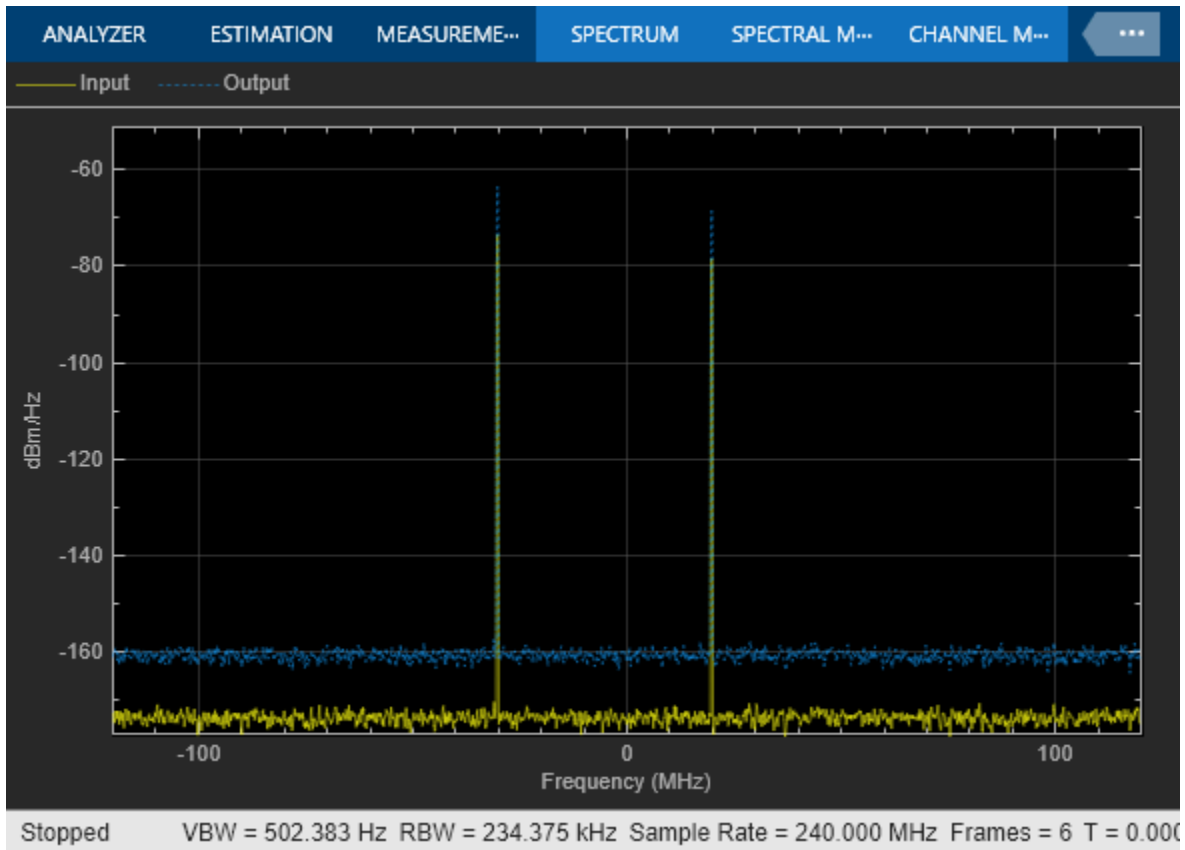
Obtain Measurements Data Programmatically for Spectrum Analyzer Block

Open and run the signal amplifier model.

```
open_system('amplifySigNoise.slx')  
sim('amplifySigNoise.slx')
```







The Spectrum Analyzer blocks in the model show the amplified noise and the signal power plots, respectively. For more details on this model, see the “Analysis of Frequency Response of RF System” example.

You can obtain or modify the peak, cursor, and distortion measurements in the spectra programmatically by enabling these properties in the `SpectrumAnalyzerConfiguration` object:

- `PeakFinder`
- `CursorMeasurements`
- `DistortionMeasurements`

Access the configuration properties of the Spectrum Analyzer block using the `get_param` function.

```
sablockNoise = 'amplifySigNoise/Noise';
cfgNoise = get_param(sablockNoise, 'ScopeConfiguration');
sablockPwr = 'amplifySigNoise/Power';
cfgPwr = get_param(sablockPwr, 'ScopeConfiguration');
```

To enable these measurements, set the `Enabled` property to `true`.

```
cfgNoise.CursorMeasurements.Enabled = true;
cfgNoise.PeakFinder.Enabled = true;
cfgNoise.DistortionMeasurements.Enabled = true;
cfgPwr.CursorMeasurements.Enabled = true;
cfgPwr.PeakFinder.Enabled = true;
cfgPwr.DistortionMeasurements.Enabled = true;
```

You can now modify these measurements programmatically. As an example, display the first two peaks on the spectrum display. By default, the spectrum shows three peaks. To modify the number of peaks it shows, change `NumPeaks` to 2. You can even label the peaks using the `LabelPeaks` property and control the minimum distance between adjacent peaks using the `MinDistance` property. The changes you make to these properties update the spectra accordingly.

```
cfgNoise.PeakFinder.NumPeaks = 2;  
cfgPwr.PeakFinder.NumPeaks = 2;
```

Version History

Introduced in R2023a

See Also

Blocks

Spectrum Analyzer

Objects

SpectrumAnalyzerConfiguration

Idealized Baseband

Amplifier

Complex baseband model of amplifier with noise and nonlinearities



Libraries:
RF Blockset / Idealized Baseband

Description

The Amplifier block generates a complex baseband model of an amplifier with thermal noise. This block provides four nonlinearity models and three options to specify noise representation.

Ports

Input

Port_1 — Input baseband signal

real scalar | real column | complex scalar | complex column

Input baseband signal, specified as a real scalar, real column, complex scalar, or complex column.

Data Types: double | single

Output

Port_1 — Output baseband signal

real scalar | real column | complex scalar | complex column

Output baseband signal, specified as a real scalar, real column, complex scalar, or complex column. The output port mimics the properties of the input port. For example, if the input baseband signal is specified as a real scalar with a data type double, then the output baseband signal is also specified as a real signal with the data type double.

Data Types: double | single

Parameters

Main Tab

Model — Amplifier nonlinearity model

Cubic polynomial (default) | AM/AM - AM/PM | Modified Rapp | Saleh

Specify the amplifier nonlinearity model as one of the following:

- Cubic polynomial
- AM/AM - AM/PM
- Modified Rapp
- Saleh

For more information, see “Nonlinearity Models in Idealized Amplifier Block”.

Linear power gain (dB) — Linear gain of amplifier

0 (default) | real scalar

Linear gain, specified as a scalar in dB.

Type of Non-Linearity — Third - order nonlinearity type

IIP3 (default) | OIP3 | IP1dB | OP1dB | IPsat | OPsat

Third order nonlinearity type, specified as IIP3, OIP3, IP1dB, OP1dB, IPsat, or OPsat.

IIP3 (dBm) — Input third-order intercept point

Inf (default) | real positive number

Input third-order intercept point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Model** to `Cubic polynomial` and **Type of Non-Linearity** to IIP3.

OIP3 (dBm) — Output third-order intercept point

Inf (default) | real positive number

Output third-order intercept point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Model** to `Cubic polynomial` and **Type of Non-Linearity** to OIP3.

IP1dB (dBm) — Input 1 dB compression point

Inf (default) | real positive number

Input 1 dB compression point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Model** to `Cubic polynomial` and **Type of Non-Linearity** to IP1dB.

OP1dB (dBm) — Output 1 dB compression point

Inf (default) | real positive number

Output 1 dB compression point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Model** to `Cubic polynomial` and **Type of Non-Linearity** to OP1dB.

IPsat (dBm) — Input saturation point

Inf (default) | real positive number

Input saturation point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Model** to `Cubic polynomial` and **Type of Non-Linearity** to IPsat.

OPsat (dBm) — Output saturation point

Inf (default) | real positive number

Output saturation point, specified as a positive real number in dBm.

Dependencies

To enable this parameter, set **Model** to `Cubic polynomial` and **Type of Non-Linearity** to `OPsat`.

Reference load (ohm) — Reference load

1 (default) | positive scalar

Reference load value in ohms, specified as a positive scalar. This value is used to convert between the voltage levels and the signal and noise power levels.

Tunable: Yes

Simulate using — Specify type of simulation to run

`Code generation` (default) | `Interpreted execution`

- `Code generation` - Simulate model using generated C code. The first time you run a simulation, Simulink generates C code for the block. The C code is reused for subsequent simulations, as long as the model does not change. This option requires additional startup time, but the speed of the subsequent simulations is faster than `Interpreted execution`.
- `Interpreted execution` - Simulate model using the MATLAB interpreter. This option shortens startup time speed, but the speed of the subsequent simulations is slower than `Code generation`. In this mode, you can debug the source code of the block.

Plot power characteristics — Plot power characteristics

`button` (default)

This button plots the power characteristics based on the parameters specified on the **Main** tab.

For more information, see “Plot Power Characteristics”.

Lookup table (Pin(dBm), Pout(dBm), deg) — Lookup table

[-25, 5, -1; -10, 20, -2; 0, 27, 5; 5, 28, 12] (default) | *M*-by-3 real matrix

Table lookup entries specified as a real *M*-by-3 matrix. This table expresses the model output power dBm level in matrix column 2 and the model phase change in degrees in matrix column 3 as related to the absolute value of the input signal power of matrix column 1 for the AM/AM - AM/PM model. The column 1 input power must increase monotonically.

Dependencies

To enable this parameter, set **Model** to `AM/AM - AM/PM`.

Output saturation level (V) — Output saturation level

1 (default) | real positive number

Voltage output saturation level, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Model** to `Modified Rapp`.

Magnitude smoothness factor — Magnitude smoothness factor

2 (default) | real positive number

Magnitude smoothness factor for the Modified Rapp amplifier model AM/AM calculations, specified as a positive real number.

Dependencies

To enable this parameter, set **Model** to Modified Rapp.

Phase gain (rad) — Phase gain

-0.45 (default) | real scalar

Phase gain for the Modified Rapp amplifier model AM/PM calculations, specified as a real scalar in radians.

Dependencies

To enable this parameter, set **Model** to Modified Rapp.

Phase saturation — Phase saturation

0.88 (default) | real positive number

Phase saturation for the Modified Rapp amplifier model AM/PM calculations, specified as a positive real number.

Dependencies

To enable this parameter, set **Model** to Modified Rapp.

Phase smoothness factor — Phase smoothness factor

3.43 (default) | real positive number

Phase smoothness factor for the Modified Rapp amplifier model AM/PM calculations, specified as a positive real number.

Dependencies

To enable this parameter, set **Model** to Modified Rapp.

Input scaling (dB) — Scaling factor for input signal level

0 (default) | nonnegative real number

Scaling factor for input signal level for the Saleh amplifier model, specified as a nonnegative real number in dB.

Dependencies

To enable this parameter, set **Model** to Saleh.

AM / AM parameters [alpha beta] — AM/AM conversion parameters

[2.1587, 1.1517] (default) | two-element vector

AM/AM two-tuple conversion parameters for Saleh amplifier model, specified as a two-element vector of nonnegative real numbers.

Dependencies

To enable this parameter, set **Model** to Saleh.

AM / PM parameters [alpha beta] — AM/PM conversion parameters

[4.0033, 9.1040] (default) | two-element vector

AM/PM two-tuple conversion parameters for Saleh amplifier model, specified as a two-element vector of nonnegative real numbers.

Dependencies

To enable this parameter, set **Model** to Saleh.

Output scaling (dB) — Scaling factor for output signal level
0 (default) | nonnegative real number

Scaling factor for output signal level for Saleh amplifier model, specified as nonnegative real number in dB.

Dependencies

To enable this parameter, set **Model** to Saleh.

Noise Tab

Include Noise — Add noise to system
off (default) | on

Select this parameter to add system noise to the input signal. Once you select this parameter, the parameters associated with the **Noise** tab are displayed.

Specify noise type — Noise representation
Noise temperature (default) | Noise figure | Noise factor

Noise descriptive type, specified as `Noise temperature`, `Noise figure`, or `Noise factor`.

For more information, see “Thermal Noise Simulations in Idealized Amplifier Block”.

Dependencies

To enable this parameter, select **Include Noise**.

Noise temperature (K) — Noise temperature to model noises in amplifier
290 (default) | nonnegative real number

Noise temperature to model noise in the amplifier, specified as a nonnegative real number in degrees (K).

Dependencies

To enable this parameter, select **Include Noise** and set **Specify noise type** to `Noise temperature`.

Noise figure (dB) — Noise figure to model noise in amplifier
 $10 * \log_{10}(2)$ (default) | nonnegative real number

Noise figure to model noise in the amplifier, specified as a nonnegative real number in dB.

Dependencies

To enable this parameter, select **Include Noise** and set **Specify noise type** to `Noise figure`.

Noise factor — Noise factor to model noise in amplifier
2 (default) | positive integer scalar greater than or equal to 1

Noise factor to model noise in the amplifier, specified as a positive integer scalar greater than or equal to 1.

Dependencies

To enable this parameter, select **Include Noise** and set **Specify noise type** to `Noise factor`.

Seed source — Source of initial seed

`Auto` (default) | `User specified`

Source of initial seed used to prepare the Gaussian random number noise generator, specified as one of the following:

- `Auto` - When **Seed source** is set to `Auto`, seeds for each amplifier instance are generated using a random number generator. The reset method of the instance has no effect.
- `User specified` - When **Seed source** is set to `User specified`, the value provided in the **Seed** is used to initialize the random number generator and the reset method resets the random number generator using the **Seed** property value.

Seed — Seed for random number generator

67987 (default) | nonnegative integer

Seed for the random number generator, specified as a nonnegative integer less than 2^{32} . Use this value to initialize the random number generator.

Dependencies

To enable this parameter, click **Include Noise** check box and choose `User specified` in the **Seed source** parameter.

Version History

Introduced in R2020a

R2022b: Reference load parameter added to the block

You can now specify load resistance in ohms using the `Reference load` parameter.

References

- [1] Razavi, Behzad. "Basic Concepts " in *RF Microelectronics*, 2nd edition, Prentice Hall, 2012.
- [2] Rapp, C., "Effects of HPA-Nonlinearity on a 4-DPSK/OFDM-Signal for a Digital Sound Broadcasting System." *Proceedings of the Second European Conference on Satellite Communications*, Liege, Belgium, Oct. 22-24, 1991, pp. 179-184.
- [3] Saleh, A.A.M., "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers." *IEEE Trans. Communications*, vol. COM-29, pp.1715-1720, November 1981.
- [4] IEEE 802.11-09/0296r16. "TGad Evaluation Methodology." Institute of Electrical and Electronics Engineers.<https://www.ieee.org/>
- [5] Kundert, Ken. "Accurate and Rapid Measurement of IP_2 and IP_3 ," *The Designer Guide Community*, May 22, 2002.

Extended Capabilities

C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

See Also

Blocks

Mixer

Topics

“Nonlinearities and Noise in Idealized Baseband Amplifier Block”

Amplifier

Complex baseband model of amplifier with noise

Description

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

Note This block assumes a nominal impedance of 1 ohm.

Parameters

Method — Method used to model nonlinearity

Linear (default) | Cubic polynomial | Hyperbolic tangent | Saleh model | Ghorbani model | Rapp model

Method used to model the amplifier nonlinearity, specified as one of the following:

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

Tunable: Yes

Linear gain (dB) — Linear gain for output function

0 (default) | real scalar

Linear gain for the output function of the Amplifier block, specified as a real scalar in decibels.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to Linear, Cubic polynomial, Hyperbolic tangent, or Rapp model.

IIP3 (dBm) — Third-order input power intercept point

30 (default) | real positive number

Third-order input power intercept point for the cubic polynomial and the hyperbolic tangent amplifier model, specified as a real positive number in dBm.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Cubic polynomial` or `Hyperbolic tangent`.

AM/PM conversion (degrees per dB) — AM/PM conversion

10 (default) | scalar

AM/PM conversion for the cubic polynomial and the hyperbolic tangent amplifier model, specified as a scalar in degrees per decibel.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Cubic polynomial` or `Hyperbolic tangent`.

Lower input power limit for AM/PM conversion (dBm) — Minimum input power for which AM/PM conversion scales linearly with input power value

10 (default) | scalar

Minimum input power for which AM/PM conversion scales linearly with the input power value, specified as a scalar. Below this value, the phase shift resulting from AM/PM conversion is zero.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Cubic polynomial` or `Hyperbolic tangent`.

Upper input power limit for AM/PM conversion (dBm) — Maximum input power for which AM/PM conversion scales linearly with input power value

inf (default) | positive scalar

Maximum input power for which AM/PM conversion scales linearly with the input power value, specified as a positive scalar. Above this value, the phase shift resulting from AM/PM conversion is constant. The value of this maximum shift is given by:

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

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Cubic polynomial` or `Hyperbolic tangent`.

Input scaling (dB) — Number that scales input signal level

0 (default) | scalar

Number that scales the input signal level for the Saleh and Ghorbani amplifier model, specified as scalar in decibels.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Saleh model` or `Ghorbani model`.

Output scaling (dB) — Number that scales output signal level
0 (default) | scalar

Number that scales the output signal level for the Saleh and Ghorbani amplifier model, specified as scalar in decibels.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Saleh model` or `Ghorbani model`.

AM/AM parameters [alpha beta] — AM/AM parameters for Saleh amplifier model
[2.1587 1.1517] (default) | vector

AM/AM parameters for the Saleh amplifier model, specified as a vector.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Saleh model`.

AM/PM parameters [alpha beta] — AM/AM parameters for Saleh amplifier model
[4.0033 9.1040] (default) | vector

AM/PM parameters for the Saleh amplifier model, specified as a vector.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Saleh model`.

AM/AM parameters [x1 x2 x3 x4] — AM/AM parameters for Ghorbani
[8.1081 1.5413 6.5202 -0.0718] (default) | vector

AM/AM parameters for the Ghorbani amplifier model, specified as a vector.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Ghorbani model`.

AM/PM parameters [y1 y2 y3 y4] — AM/PM parameters for Ghorbani
[4.6645 2.0965 10.88 -0.003] (default) | vector

AM/PM parameters for the Ghorbani amplifier model, specified as a vector.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Ghorbani model`.

Smoothness factor — Magnitude smoothness factor
0.5 (default) | positive scalar

Magnitude smoothness factor for the Rapp amplifier model, specified as a positive scalar.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Rapp_model`.

Output saturation level — Output saturation level
1 (default) | real positive number

Output saturation level for the Rapp amplifier model, specified as a real positive number. Scalar specifying the output saturation level.

Tunable: Yes

Dependencies

To enable this parameter, set **Method** to `Rapp_model`.

Specification method — Type of noise
`Noise_factor` (default) | `Noise_temperature` | `Noise_figure`

Type of noise, specified as `Noise_temperature`, `Noise_figure`, and `Noise_factor`.

Tunable: Yes

Noise temperature (K) — Noise temperature to model the amplifier noise
290 (default) | nonnegative real number

Noise temperature to model the amplifier noise, specified as a nonnegative real number in kelvin.

Tunable: Yes

Dependencies

To enable this parameter, set **Specification method** to `Noise_temperature`.

Noise figure (dB) — Noise figure to model amplifier noise
3.01 (default) | nonnegative real number

Noise figure to model the amplifier noise, specified as a nonnegative real number in decibels. Setting `Noise_figure` to 0 decibels indicates a noiseless system.

Tunable: Yes

Dependencies

To enable this parameter, set **Specification method** to `Noise_figure`.

Noise factor — Noise factor to model amplifier noise
2 (default) | positive integer scalar greater than or equal to 1

Noise factor to model amplifier noise relative to a noise temperature, specified as a positive integer scalar greater than or equal to 1.

To enable this parameter, set **Specification method** to `Noise_factor`.

Tunable: Yes

Initial seed — Seed for random number generator
67987 (default) | nonnegative integer less than 2^{32}

Seed for the random number generator, specified as a nonnegative integer less than 2^{32} . Use this value to initialize the random number generator the block uses to generate noise.

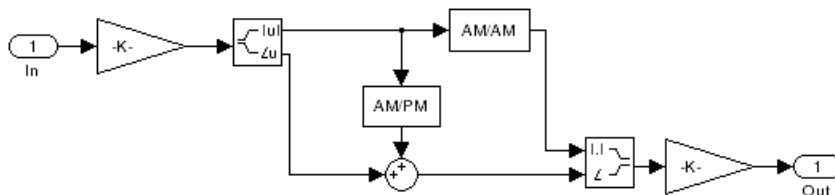
Algorithms

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.



Model Parameters and Characteristics of Nonlinearity Modeling Methods

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

- “Cubic Polynomial Model” on page 4-14
- “Hyperbolic Tangent Model” on page 4-14
- “Saleh Model” on page 4-15
- “Ghorbani Model” on page 4-15
- “Rapp Model” on page 4-16

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

Cubic Polynomial Model

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

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

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

- 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 unit less normalized input voltage.

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

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

respectively.

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

Hyperbolic Tangent Model

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

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

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

- 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 unit less 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

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

respectively.

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

Saleh Model

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

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

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

where u is the magnitude of the scaled signal.

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

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

where u is the magnitude of the input signal. Note that the AM/AM and AM/PM parameters, although similarly named alpha and beta, are distinct.

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

Ghorbani Model

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

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

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

where u is the magnitude of the scaled signal.

The AM/PM parameters, $[y_1 \ y_2 \ y_3 \ y_4]$, are used to compute the phase change for an input signal using the following function

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

where u is the magnitude of the scaled signal.

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

Rapp Model

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

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

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

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

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

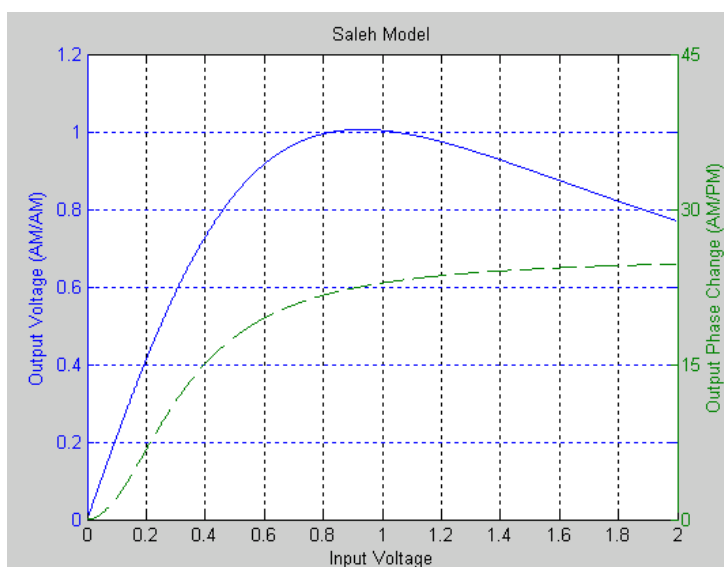
AM/AM and AM/PM Conversions

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



Thermal Noise Simulation

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

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

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

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

$$\text{Noise figure} = 10\log(\text{Noise factor}) \quad (4-1)$$

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.

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 (dB)** parameter.

Version History

Introduced before R2006a

References

- [1] Ghorbani, A. and M. Sheikhan, "The Effect of Solid State Power Amplifiers (SSPAs) Nonlinearities on MPSK and M-QAM Signal Transmission," *Sixth Int'l Conference on Digital Processing of Signals in Comm.*, 1991, pp. 193-197.
- [2] Rapp, C., "Effects of HPA-Nonlinearity on a 4-DPSK/OFDM-Signal for a Digital Sound Broadcasting System," in *Proceedings of the Second European Conference on Satellite Communications*, Liege, Belgium, Oct. 22-24, 1991, pp. 179-184.
- [3] Saleh, A.A.M., "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Communications*, vol. COM-29, pp.1715-1720, November 1981.

See Also

Amplifier

Mixer

Model RF and IQ modulator and RF and IQ demodulator with impairments and noise



Libraries:

RF Blockset / Idealized Baseband

Description

The Mixer block models four complex baseband mixers and with impairments and noise. The four mixer types that the block models are Modulator, Demodulator, IQ Modulator, and IQ Demodulator. Impairments include IQ gain and phase mismatch where appropriate, while noise includes both system and LO phase noise.

Note

- Idealized Baseband library blocks assume input and output ports are matched. For more information on port signal power, see “Power Ports and Signal Power Measurement in RF Blockset”.
 - Idealized Baseband library blocks are single carrier with assumed carrier frequency value. Therefore the Ideal Baseband Mixer block can produce only a single sideband output.
 - Mixer block mask icons are dynamic and indicate the current set of applied noise parameters. For more information, see “Mixer Block Icons” on page 4-29.
-

Ports

Input

Port_1 — Time-dependent input signal

real scalar | real column | complex scalar | complex column

Time-dependent input signal, specified as a real scalar, real column, complex scalar, or complex column. A column represents consecutive time points.

Data Types: double | single

Output

Port_1 — Time-dependent output signal

complex scalar | complex column

Time-dependent output signal, returned as a complex scalar or complex column. The size of the output time dependent signal is equal in size to the input time dependent signal.

Data Types: double | single

Parameters

Main Tab

Mixer type — Type of mixer

Modulator (default) | Demodulator | IQ Modulator | IQ Demodulator

Mixers available in the Mixer block, specified as one of the following:

- Modulator
- Demodulator
- IQ Modulator
- IQ Demodulator

For more information, see “Mixer Architectures and Design Equations” on page 4-26.

Mixer sideband — Sidebands of mixer

Upper (default) | Lower

Sidebands of the mixer, specified as one of the one of the following:

- Lower
- Upper

For more information, see “Mixer Sidebands” on page 4-29.

Dependencies

To enable this parameter, set **Mixer type** to Modulator.

Input carrier frequency greater than LO frequency — Specify input carrier frequency as greater than LO frequency

on (default) | off

Specify whether the mixer baseband input carrier frequency is greater than the mixer LO frequency, specified as a logical.

Dependencies

To enable this parameter, set **Mixer type** to Modulator and **Mixer sideband** to Lower or **Mixer type** to Demodulator.

Conversion gain (dB) — Conversion gain for mixer

0 (default) | real number

Conversion gain for the mixer model, specified as a real number in dB.

Simulate using — Specify type of simulation to run

Interpreted execution (default) | Code generation

Type of simulation to run, specified as one of the following:

- Code generation — Simulate model using generated C code. The first time you run a simulation, Simulink generates C code for the block. The C code is reused for subsequent

simulations, as long as the model does not change. This option requires additional startup time, but the speed of the subsequent simulations is faster than `Interpreted` execution.

- `Interpreted` execution — Simulate model using the MATLAB interpreter. This option shortens startup time speed, but the speed of the subsequent simulations is slower than `Code generation`. In this mode, you can debug the source code of the block.

Impairments Tab

LO phase offset (deg) — LO phase offset

0 (default) | real number

LO phase offset, specified as a real number in degrees.

Dependencies

To enable this parameter, set **Mixer type** to `Modulator` or `Demodulator`.

I/Q gain imbalance (dB) — IQ gain imbalance

0 (default) | nonnegative real numbers

IQ gain imbalance, specified as a nonnegative real numbers in decibels.

Dependencies

To enable this parameter, set **Mixer type** to `IQ Modulator` or `IQ Demodulator`.

I/Q phase imbalance (deg) — IQ phase imbalance

0 (default) | real numbers

IQ phase imbalance, specified as a real number in degrees.

Dependencies

To enable this parameter, set **Mixer type** to `IQ Modulator` or `IQ Demodulator`.

Type of Non-Linearity — Third-order nonlinearity type

IIP3 (default) | OIP3 | IP1dB | OP1dB | IPsat | OPsat

Type of third-order nonlinearity type in the cubic polynomial model, specified as IIP3, OIP3, IP1dB, OP1dB, IPsat, or OPsat.

For more information, see “Nonlinearities in Idealized Baseband Mixer Block”.

IIP3 (dBm) — Input third-order intercept point

Inf (default) | real positive number

Input third-order intercept point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Type of Non-Linearity** to IIP3.

OIP3 (dBm) — Output third-order intercept point

Inf (default) | real positive number

Output third-order intercept point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Type of Non-Linearity** to OIP3.

IP1dB (dBm) — Input 1 dB compression point

Inf (default) | real positive number

Input 1 dB compression point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Type of Non-Linearity** to IP1dB.

OP1dB (dBm) — Output 1 dB compression point

Inf (default) | real positive number

Output 1 dB compression point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Type of Non-Linearity** to OP1dB.

IPsat (dBm) — Input saturation point

Inf (default) | real positive number

Input saturation point, specified as a real positive number in dBm.

Dependencies

To enable this parameter, set **Type of Non-Linearity** to IPsat.

OPsat (dBm) — Output saturation point

Inf (default) | real positive number

Output saturation point, specified as a positive real number in dBm.

Dependencies

To enable this parameter, set **Type of Non-Linearity** to OPsat.

Plot power characteristics — Plot power characteristics

button

This button plots the power characteristics based on the value you specify in the “Conversion gain (dB)” on page 4-0 parameter in the **Main** tab and the “Type of Non-Linearity” on page 4-0 in the **Impairments** tab. When plotting power characteristics, the block ignores all other impairment values.

For more information, see “Plot Power Characteristics”.

Noise Tab

Include mixer noise — Add mixer noise to system

off (default) | on

Select this parameter to add mixer noise to the input signal. Once you select this parameter, the parameters associated with **Include mixer noise** are displayed and the mixer components in the block icon are shaded gray.

For more information, see “Mixer (System) Noise Simulations”.

Mixer noise type — Noise representation

Noise temperature (default) | Noise figure | Noise factor

Type of noise, specified as a Noise temperature, Noise figure, or Noise factor.

For more information, see “Mixer (System) Noise Simulations”.

Dependencies

To enable this parameter, select **Include mixer noise**.

Noise temperature (K) — Noise temperature to model mixer noise

290 (default) | nonnegative real number

Noise temperature to model the mixer noise, specified as a nonnegative real number in kelvin.

Dependencies

To enable this parameter, select **Include mixer noise** and set **Mixer noise type** to Noise temperature.

Noise figure (dB) — Noise figure to model mixer noise

$10 * \log_{10}(2)$ (default) | nonnegative real number

Noise figure to model the mixer noise, specified as a nonnegative real number in decibels.

Dependencies

To enable this parameter, select **Include mixer noise** and set **Mixer noise type** to Noise figure.

Noise factor — Noise factor to model mixer noise

2 (default) | positive integer scalar greater than or equal to 1

Noise factor to model mixer noise, specified as a positive integer scalar greater than or equal to 1

Dependencies

To enable this parameter, select **Include mixer noise** and set **Mixer noise type** to Noise factor.

Seed source, mixer noise — Source of initial seed

Auto (default) | User specified

Source of initial seed used to prepare the Gaussian random number noise generator, specified as one of the following:

- Auto — When you set **Seed source, mixer noise** to Auto, seeds for each mixer instance are generated using a random number generator. The reset method of the instance has no effect.
- User specified — When you set **Seed source, mixer noise** to User specified, the value provided in the **Seed for mixer noise** is used to initialize the random number generator and the reset method resets the random number generator using the **Seed for mixer noise** property value.

Dependencies

To enable this parameter, select **Include mixer noise**.

Seed for mixer noise — Seed for random number generator
67987 (default) | nonnegative integer less than 2^{32}

Seed for the random number generator, specified as a nonnegative integer less than 2^{32} . Use this value to initialize the random number generator.

Dependencies

To enable this parameter, select **Include mixer noise** and set the **Seed source, mixer noise** parameter to `User specified`.

Include phase noise — Add LO phase noise to LO signal
`off` (default) | `on`

Select this parameter to add frequency-dependent LO phase noise to the LO signal. Once you select this parameter, the parameters associated with the **Include phase noise** are displayed and the LO source inside the block icon is shaded gray.

For more information, see “Phase Noise in Mixer Block”.

Phase noise level (dBc/Hz) — Phase noise level relative to carrier
[-145 -150] (default) | negative real scalar | negative real vector

Phase noise level relative to carrier, specified as negative real scalar or vector in dBc/Hz.

Note The number of terms listed in the **Phase noise level (dBc/Hz)** parameter must equal the number of terms in the “Frequency offset (Hz)” on page 4-0 field.

Dependencies

To enable this parameter, select the **Include phase noise** parameter.

Data Types: `double`

Frequency offset (Hz) — Phase frequency offset
[1000 50000] (default) | positive real scalar | vector of positive increasing real values

Specify the frequency offset as a positive real scalar or vector of positive increasing real values of type `double` in Hz.

Note The number of terms listed in the **Frequency offset (Hz)** must equal the number of terms in the “Phase noise level (dBc/Hz)” on page 4-0 field.

Dependencies

To enable this parameter, select the **Include phase noise** parameter.

Automatic frequency resolution — Automatically determine number of frequency bins
`on` (default) | `off`

Select this parameter to automatically determine number of frequency bins used in a two-sided phase noise spectrum. You can also set the number of frequency bins using the **Number of signal samples**

and **Sample rate (Hz)** parameters when you set the **Automatic frequency resolution** parameter is set to off.

Dependencies

To enable this parameter, click **Include phase noise**.

Number of signal samples — Number of samples in time-domain signal
0 (default) | real nonnegative integer less than or equal to 65536

Number of samples in the time-domain signal for the blocks sample time or the number of frequency lines (bins) in the signals two-sided frequency spectrum to achieve the required frequency resolution for a specified Frequency offset, specified as a real nonnegative integer less than or equal to 65536. Frequency resolution increases as the value of number of signal samples increases.

Note The value of this parameter must be set to a power of two.

Dependencies

To enable this parameter, select the **Include phase noise** parameter and deselect **Automatic frequency resolution**.

Seed source, phase noise — Source of initial seed
Auto (default) | User specified

Source of initial seed used to prepare the Gaussian random number LO phase noise generator, specified as one of the following:

- **Auto** — When you set the **Seed source, phase noise** to Auto, seeds for each mixer instance are generated using a random number generator. The reset method of the instance has no effect.
- **User specified** — When you set the **Seed source, phase noise** to User specified, the value provided in the **Seed for phase noise** is used to initialize the random number generator and the reset method resets the random number generator using the **Seed for phase noise** property value.

Dependencies

To enable this parameter, select the **Include phase noise** parameter.

Seed for phase noise — Seed for random number generator
67987 (default) | nonnegative integer less than 2^{32}

Seed for the random number generator, specified as a nonnegative integer less than 2^{32} . Use this value to initialize the random number generator.

Dependencies

To enable this parameter, select **Include phase noise** and set the **Seed source, phase noise** parameter to User specified.

Plot phase characteristics — Plot phase characteristics
button

This button plots the phase characteristics based on the parameters specified on the **Noise** tab and either the block sample time when a simulation has been performed or estimated from the “Frequency offset (Hz)” on page 4-0 parameter values.

For more information, see “Plot Phase Noise Characteristics”.

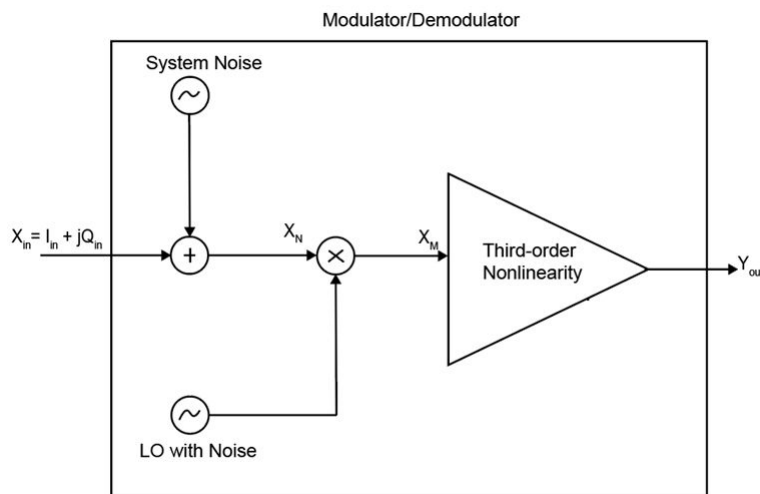
Algorithms

Mixer Architectures and Design Equations

Architectural models for the Mixer block are shown here. Mixer and phase noise for all cases are included here.

Modulator and Demodulator Architectures and Equations

Modulator and demodulator architectures includes system noise, phase noise, and nonlinear polynomials to translates the carrier. A random number generator is used as an input to generate phase noise.



The output of the ideal modulator and demodulator circuit with nonlinearities, Y_{out} , is given by this equation.

$$Y_{out} = \left(c_1 + \left(3 \times \frac{c_3}{4} \right) \times |X_M|^2 \right) \times |X_M| \times \exp(i \times \text{angle}(X_M))$$

where,

$$X_M = (I_N \times c_{Ph} - Q_N \times s_{Ph}) + j (Q_N \times c_{Ph} - I_N \times s_{Ph})$$

$$c_{Ph} = \cosd(\text{phaseOffset}(\text{deg}) + \text{phaseNoise}(\text{deg}))$$

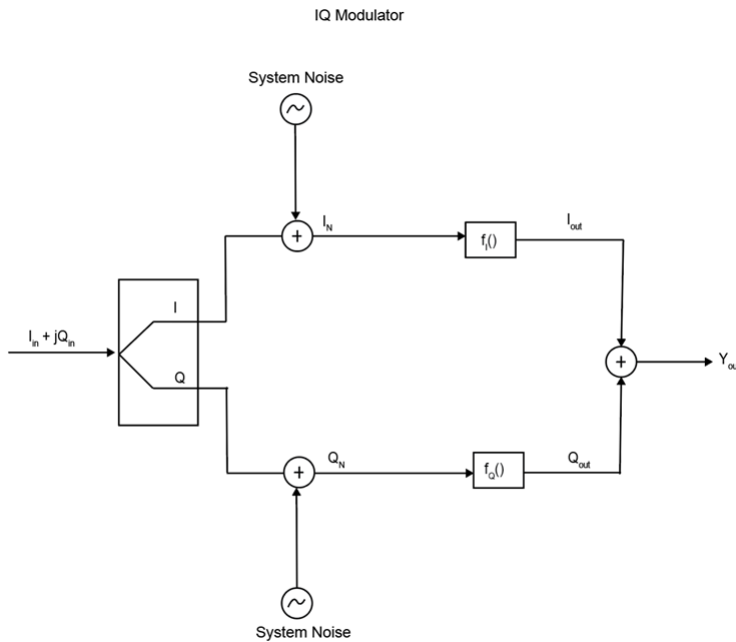
$$s_{Ph} = \text{sind}(\text{phaseOffset}(\text{deg}) + \text{phaseNoise}(\text{deg}))$$

$$I_N + jQ_N = (I_{in} + jQ_{in}) + \text{System Noise}$$

The nonlinear polynomial coefficients, c_1 and c_3 , are provided in “Nonlinearities in Idealized Baseband Mixer Block”.

IQ Modulator Architecture and Equations

The IQ modulator primary consists of two mixers, $f_I()$ and $f_Q()$. The Mixers convert baseband signals to RF signals and are commonly used in direct conversion architectures. The $f_I()$ and $f_Q()$, are responsible for introducing gain, gain imbalance, phase imbalance, and nonlinearities into the IQ demodulator.



The output of IQ Modulator, Y_{out} , is given as

$$Y_{out} = (I_{out} \times \cosd(I_{Offset}) + Q_{out} \times \sind(Q_{Offset})) - i(I_{out} \times \sind(I_{Offset}) - Q_{out} \times \cosd(Q_{Offset}))$$

where,

$$I_{out} = (c_1 + \frac{3c_3}{4} \times I_N^2) \times I_N$$

$$Q_{out} = (c_{1Q} + \frac{3c_{3Q}}{4} \times Q_N^2) \times Q_N$$

$$I_{offset} = \frac{\text{phaseNoise(deg)} + \text{phaseImbalance (deg)}}{2}$$

$$Q_{offset} = \frac{\text{phaseNoise(deg)} - \text{phaseImbalance (deg)}}{2}$$

The nonlinear polynomial coefficient c_3 are provided in "Nonlinearities in Idealized Baseband Mixer Block". The modulators linear gains are c_1 and c_{1Q} are provided in this equation.

$$c_1 = (1 + \delta) \times 10^{(\text{ConversionGainDB}/20)}$$

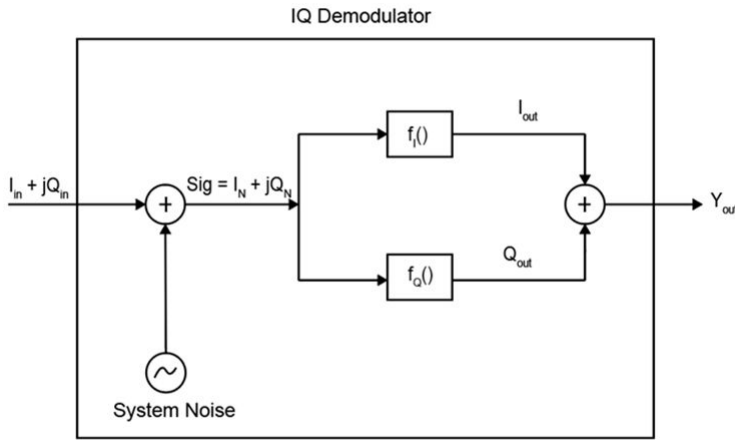
$$c_{1Q} = (1 - \delta) \times 10^{(\text{ConversionGainDB}/20)}$$

where,

$$\delta = \frac{10^{(\text{GainImbalanceDB} / 20)} - 1}{10^{(\text{GainImbalanceDB} / 20)} + 1}$$

IQ Demodulator Architecture and Equations

The architecture of an IQ demodulator is given below. The in-phase, I_{out} , and quadrature component, Q_{out} , of the modulated signal are the output of the $f_i()$ and $f_q()$, respectively. The mixers, $f_i()$ and $f_q()$ are responsible for introducing gain, gain imbalance, phase imbalance, and nonlinearities into the IQ demodulator.



The output of the IQ Demodulator, Y_{out} , is given as

$$Y_{\text{out}} = I_{\alpha} \times \text{sqrt}(P_I) \times \text{cosd}(\text{Theta} - I_{\text{offset}}) + iQ_{\alpha} \times \text{sqrt}(P_Q) \times \text{sind}(\text{Theta} - Q_{\text{offset}})$$

where,

$$I_{\alpha} = (c_1 + \frac{3c_3}{4} \times P_{IN})$$

$$Q_{\alpha} = (c_{1Q} + \frac{3c_{3Q}}{4} \times P_Q)$$

$$I_{\text{offset}} = \frac{\text{phaseNoise}(\text{deg}) + \text{phaseImbalance}(\text{deg})}{2}$$

$$Q_{\text{offset}} = \frac{\text{phaseNoise}(\text{deg}) - \text{phaseImbalance}(\text{deg})}{2}$$

$$\text{Theta} = \text{angle}(\text{Sig})$$

$$P_{IN} = |\text{Sig}|^2$$

$$P_I = f(P_{IN}, (1 + \text{GainImbalance}))$$

$$P_Q = f(P_{IN}, (1 - \text{GainImbalance}))$$

The nonlinear polynomial coefficient c_3 are provided in “Nonlinearities in Idealized Baseband Mixer Block”. The modulators linear gains c_1 and c_{1Q} are provided in this equation.

$$c_1 = (1 + \delta) \times 10^{(\text{ConversionGainDB}/20)}$$

$$c_{1Q} = (1 - \delta) \times 10^{(\text{ConversionGainDB}/20)}$$

where,

$$\delta = \frac{10^{(\text{GainImbalanceDB} / 20)} - 1}{10^{(\text{GainImbalanceDB} / 20)} + 1}$$

Mixer Sidebands

Upper and Lower Sidebands

The expression for S_{out} shows the production of upper and lower sidebands, $(\omega_{lo} + \omega_{in})$ and $(\omega_{lo} - \omega_{in})$, and the effect of the difference values of the input carrier and LO signal on the sine function. Applying the trigonometric identity

$$\cos(\alpha + \beta) = \cos(\alpha) \times \cos(\beta) - \sin(\alpha) \times \sin(\beta)$$

to the a mixer product expression, $S_{out} = S_{in} \times S_{lo}$ yields

$$\begin{aligned} S_{out} &= S_{in} \times S_{lo} \\ &= (A_{in}(t) \times \cos(\omega_{in}t)) \times (A_{lo}(t) \times \cos(\omega_{lo}t + \phi(t))) \\ &= \left(\frac{A_{in}(t) \times A_{lo}(t)}{2} \right) \times \left\{ \begin{aligned} &\cos(\phi(t)) \times \cos(\omega_{lo} + \omega_{in})t - \sin(\phi(t)) \times \sin(\omega_{lo} + \omega_{in})t \\ &+ \cos(\phi(t)) \times \cos(\omega_{lo} - \omega_{in})t - \sin(\phi(t)) \times \sin(\omega_{lo} - \omega_{in})t \end{aligned} \right\} \end{aligned}$$

where, the term connected to the higher output frequency, $\omega_{in} + \omega_{lo}$, is the upper sideband and $|\omega_{lo} - \omega_{in}|$ is the lower sideband. Since Ideal Baseband library blocks only support a single carrier signal, the desired output sideband must be selected. Set the **Mixer sideband** parameter to **Modulator**.

For demodulators, only the lower sideband can be used as an output. Finally select the **Input carrier frequency greater than LO frequency** parameter when the term $\sin(\omega_{lo} - \omega_{in})$ contributes to the output signal.

















Mixer Selections by Type

When selecting a Mixer type, this table provides the available sideband options **LO phase offset (deg)**, **I/Q gain imbalance (dB)**, and **I/Q phase imbalance (deg)**. In addition, specify the relative size of RF to LO when the output frequency depends on the sign of (RF-LO).

Mixer Type	Sideband	If RF > LO	Impairments
Modulator	Higher	N.A	LO phase offset
Modulator	Lower	Select the Input carrier frequency greater than LO frequency parameter	LO phase offset
Demodulator	N.A	N.A	LO phase offset

Mixer Block Icons

This table shows you how the icons on this block will vary based on how you set the “Parameters” on page 4-20 on the block.

Mixer Type	Include Mixer Noise	Include Phase Noise: off	Include Phase Noise: on
Modulator	off		
	on		
Demodulator	off		
	on		
IQ Modulator	off		
	on		
IQ Demodulator	off		
	on		

Note After you set the block parameters you must click the **Apply** button to see the icon change.

Version History

Introduced before R2006a

R2021a: Mixer models and noise supported for Mixer block in Idealized Baseband library changed

Behavior changed in R2021a

Starting in R2021a, the Mixer block supports modulator, demodulator, IQ modulator, and IQ demodulator mixers with impairments and noise. The new Mixer block enables you to use pre-R2021a functionality when the **Mixer type** parameter is set to **Modulator**.

When you open a model from a previous release containing the Mixer block from the Idealized Baseband library, the pre-R2021a version of this block is automatically replaced with the R2021a version of this block with impairments and noise.

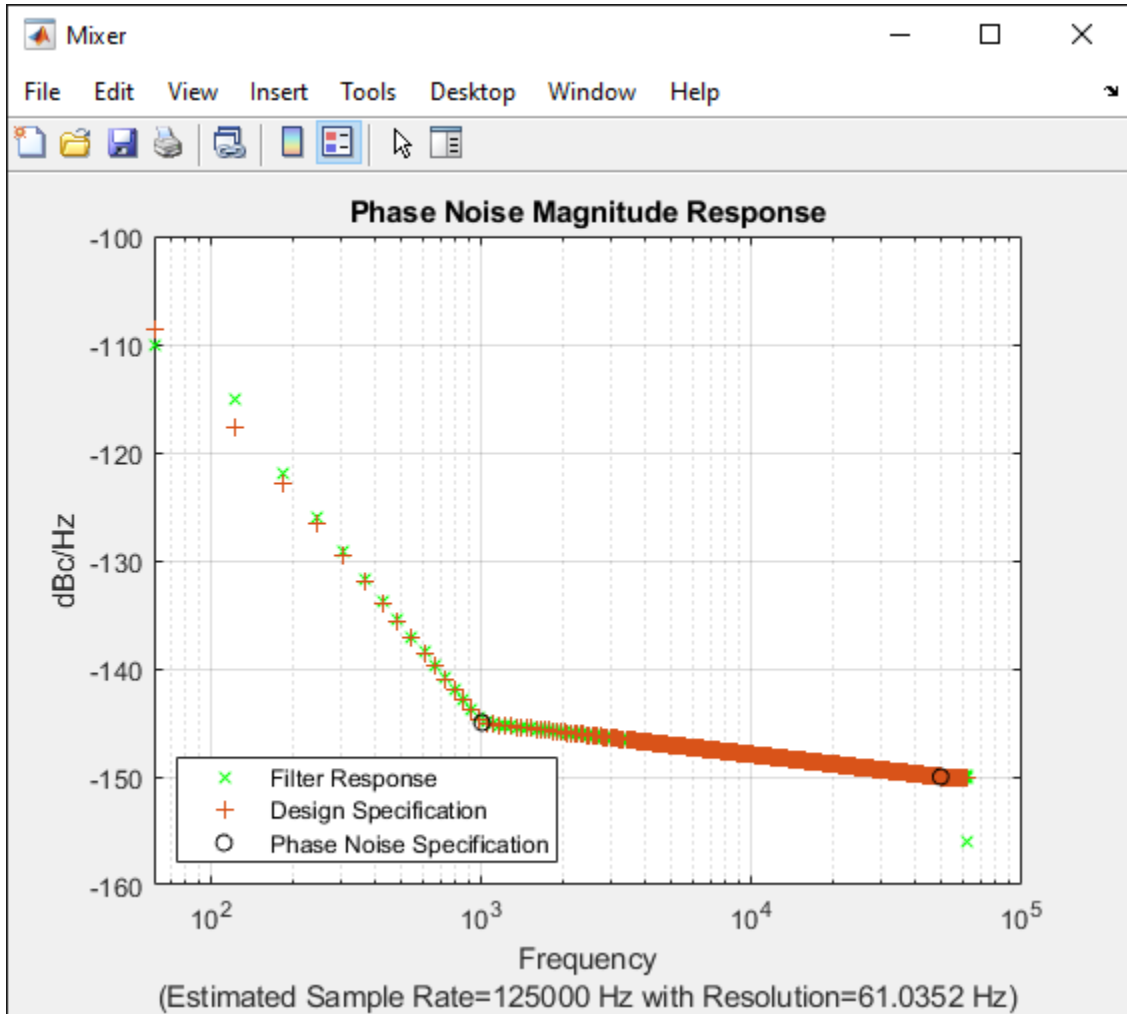
R2023a: Phase noise characteristics of Mixer block changed

Behavior changed in R2023a

Starting in this release, the extrapolated low-frequency phase noise below the first frequency point that specifies the $1/f^3$ in the Mixer block is now 30 dB/decade instead of 3 dB/decade.

When you open a model created before R2023a containing the Mixer block and click the **Plot Phase Characteristics** button, the software scales the phase noise level in dBc/Hz on the y-axis in a phase noise magnitude response plot.

For example, open the Mixer block and select **Include Phase Noise**. Set the **Number of signal samples** to 2^{10} and click the **Apply** button. Click the **Plot Phase Characteristics** button to plot the phase noise magnitude response and observe that the first frequency point that specifies the $1/f^3$ in the Mixer block is scaled to 30 dB/decade.



References

- [1] Razavi, Behzad. "Basic Concepts " in *RF Microelectronics*, 2nd edition, Prentice Hall, 2012.
- [2] Kundert, Ken. "Accurate and Rapid Measurement of IP_2 and IP_3 ," *The Designer Guide Community*, May 22, 2002.
- [3] Kasdin, N.J. "Discrete Simulation of Colored Noise and Stochastic Processes and $1/f^\alpha$ Power Law Noise Generation." *Proceedings of the IEEE* 83, no. 5 (May 1995): 802-27. <https://doi.org/10.1109/5.381848>.

Extended Capabilities

C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

See Also

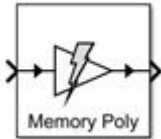
Amplifier

Topics

“Nonlinearities and Noise in Idealized Baseband Mixer Block”

Power Amplifier

Model narrowband power amplifier with memory



Libraries:
RF Blockset / Idealized Baseband

Description

The Power Amplifier block models a two-port power amplifier using a memory polynomial expression derived from the Volterra series. The Volterra series models the nonlinear relationship between input and output signals. This block includes memory effects: the output response depends on the current input signal and the input signal at previous times. Use this block when transmitting narrowband signals into your RF system.

The Power Amplifier block mask icons are dynamic and show the model specified in the **Model Type** parameter.

Model Type: Memory polynomial	Model Type: Cross-Term Memory

Ports

Input

Port_1 — Time-dependent input signal
real scalar | real column | complex scalar | complex column

Time-dependent input signal, specified as a real scalar, real column, complex scalar, or complex column. A column represents consecutive points in time.

Data Types: double | single

Output

Port_1 — Time-dependent output signal
complex scalar | complex column

Time-dependent output signal, returned as a complex scalar or complex column. The output time-dependent signal is equal in size to the input time dependent signal.

Data Types: double | single

Parameters

Model type — Model to design amplifier

Memory polynomial (default) | Cross-Term Memory

Use either the Memory polynomial model or the Cross-Term Memory model to design the power amplifier. The following table summarizes the characteristics of the two models.

Model	Characterization Data	Type of Coefficients	In-Band Spectral Regrowth	Out-of-Band Harmonic Generation
Memory polynomial (default)	Bandpass (I,Q)	Complex	Yes	No
Cross-Term Memory	Bandpass (I,Q)	Complex	Yes	No

For more information, see “Model Type” on page 4-35.

Coefficient matrix — Coefficient matrix

1 (default) | complex matrix | real matrix

Coefficient matrix, specified as a complex matrix for the Memory polynomial and Cross-Term Memory models.

For the Memory polynomial and Cross-Term Memory models, you can identify the complex coefficient matrix based on the measured complex (I,Q) output-vs.-input amplifier characteristic. As an example, see the helper function in “Coefficient Matrix Computation” on page 4-36 .

The size of the matrix depends on the number of delays and the degree of the system nonlinearity.

- For the Memory polynomial model, the matrix has dimensions Memory Depth (mem)×Voltage Order (deg).
- For the Cross-Term Memory model, the matrix has dimensions Memory Depth (mem) × {Memory Depth (mem) · (Voltage Order (deg) – 1) + 1}.

Time interval of measured PA data (s) — Sample time of measured input-output data

1e-6 (default) | positive scalar

Sample time of the measured input-output data used to construct the coefficient matrix, specified in seconds.

Simulate using — Type of simulation to run

Code generation (default) | Interpreted execution

Type of simulation to run, specified as one of the following:

- Code generation — Simulate model using generated C code. The first time you run a simulation, Simulink generates C code for the block. The C code is reused for subsequent simulations, as long as the model does not change. This option requires additional startup time, but the speed of the subsequent simulations is faster than Interpreted execution.

- **Interpreted execution** — Simulate model using the MATLAB interpreter. This option shortens startup time speed, but the speed of the subsequent simulations is slower than Code generation. In this mode, you can debug the source code of the block.

Algorithms

Model Type

The Power Amplifier block in the Idealized Baseband library supports two model types.

- **Memory polynomial** — The narrowband memory polynomial implementation (equation (19) of [1]) used in this model operates on the envelope of the input signal, does not generate new frequency components, and captures in-band spectral regrowth. Use this model to create a narrowband amplifier operating at high frequency.

The output signal, at any instant of time, is the sum of all the elements of the complex matrix of dimensions Memory Depth (mem)×Voltage Order (deg):

$$\begin{bmatrix} C_{11}V_0 & C_{12}V_0|V_0| & \cdots & C_{1,\text{deg}}V_0|V_0|^{\text{deg}-1} \\ C_{21}V_1 & C_{22}V_1|V_1| & \cdots & C_{2,\text{deg}}V_1|V_1|^{\text{deg}-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{\text{mem},1}V_{\text{mem}-1} & C_{\text{mem},2}V_{\text{mem}-1}|V_{\text{mem}-1}| & \cdots & C_{\text{mem},\text{deg}}V_{\text{mem}-1}|V_{\text{mem}-1}|^{\text{deg}-1} \end{bmatrix}.$$

In the matrix, the number of rows equals the number of memory terms, and the number of columns equals the degree of the nonlinearity. The signal subscript represents the amount of delay.

- **Cross-Term Memory** — The narrowband memory polynomial implementation (equation (23) of [1]) used in this model also operates on the envelope of the input signal, does not generate new frequency components, and captures in-band spectral regrowth. Use this model to create a narrowband amplifier operating at high frequency. This model includes leading and lagging memory terms and provides a generalized implementation of the memory polynomial model.

The output signal, at any instant of time, is the sum of all the elements of the matrix specified by the element-by-element product

$$\mathbf{C} .* \mathbf{M}_{\text{CTM}},$$

where \mathbf{C} is a complex coefficient matrix of dimensions

Memory Depth (mem) × {Memory Depth (mem) · (Voltage Order (deg) – 1) + 1} and

$$\mathbf{M}_{\text{CTM}} = \begin{bmatrix} V_0 \\ V_1 \\ \vdots \\ V_{\text{mem}-1} \end{bmatrix} \begin{bmatrix} 1 & |V_0| & |V_1| & \cdots & |V_{\text{mem}-1}| & |V_0|^2 & \cdots & |V_{\text{mem}-1}|^2 & \cdots & |V_0|^{\text{deg}-1} & \cdots & |V_{\text{mem}-1}|^{\text{deg}-1} \end{bmatrix}$$

$$= \begin{bmatrix} V_0 & V_0|V_0| & V_0|V_1| & \cdots & V_0|V_{\text{mem}-1}| & V_0|V_0|^2 & \cdots & V_0|V_{\text{mem}-1}|^2 & \cdots & V_0|V_0|^{\text{deg}-1} & \cdots & V_0|V_{\text{mem}-1}|^{\text{deg}-1} \\ V_1 & V_1|V_0| & V_1|V_1| & \cdots & V_1|V_{\text{mem}-1}| & V_1|V_0|^2 & \cdots & V_1|V_{\text{mem}-1}|^2 & \cdots & V_1|V_0|^{\text{deg}-1} & \cdots & V_1|V_{\text{mem}-1}|^{\text{deg}-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ V_{\text{mem}-1} & V_{\text{mem}-1}|V_0| & V_{\text{mem}-1}|V_1| & \cdots & V_{\text{mem}-1}|V_{\text{mem}-1}| & V_{\text{mem}-1}|V_0|^2 & \cdots & V_{\text{mem}-1}|V_{\text{mem}-1}|^2 & \cdots & V_{\text{mem}-1}|V_0|^{\text{deg}-1} & \cdots & V_{\text{mem}-1}|V_{\text{mem}-1}|^{\text{deg}-1} \end{bmatrix}$$

In the matrix, the number of rows equals the number of memory terms, and the number of columns is proportional to the degree of the nonlinearity and the number of memory terms. The signal subscript represents the amount of delay. The additional columns that do not appear in the Memory polynomial model represent the cross terms.

Coefficient Matrix Computation

To compute coefficient matrices, the block solves an overdetermined linear system of equations. Consider the Memory polynomial model for the case where the memory length is 2 and the system

nonlinearity is of third degree. The matrix that describes the system is $\begin{bmatrix} C_{11}V_0 & C_{12}V_0|V_0| & C_{13}V_0|V_0|^2 \\ C_{21}V_1 & C_{22}V_1|V_1| & C_{23}V_1|V_1|^2 \end{bmatrix}$, and the sum of its elements is equivalent to the inner product

$$\begin{bmatrix} V_0 & V_1 & V_0|V_0| & V_1|V_1| & V_0|V_0|^2 & V_1|V_1|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \end{bmatrix}.$$

If the input to the amplifier is the five-sample signal $[x(1) \ x(2) \ x(3) \ x(4) \ x(5)]$ and the corresponding output is $[y(1) \ y(2) \ y(3) \ y(4) \ y(5)]$, then the solution to

$$\begin{bmatrix} x(2) & x(1) & x(2)|x(2)| & x(1)|x(1)| & x(2)|x(2)|^2 & x(1)|x(1)|^2 \\ x(3) & x(2) & x(3)|x(3)| & x(2)|x(2)| & x(3)|x(3)|^2 & x(2)|x(2)|^2 \\ x(4) & x(3) & x(4)|x(4)| & x(3)|x(3)| & x(4)|x(4)|^2 & x(3)|x(3)|^2 \\ x(5) & x(4) & x(5)|x(5)| & x(4)|x(4)| & x(5)|x(5)|^2 & x(4)|x(4)|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \end{bmatrix} = \begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix},$$

which can be found using the

MATLAB backslash operator, provides an estimate of the coefficient matrix.

The treatment of the Cross-Term Memory model is similar. The matrix that describes the system is

$\begin{bmatrix} C_{11}V_0 & C_{12}V_0|V_0| & C_{13}V_0|V_1| & C_{14}V_0|V_0|^2 & C_{15}V_0|V_1|^2 \\ C_{21}V_1 & C_{22}V_1|V_0| & C_{23}V_1|V_1| & C_{24}V_1|V_0|^2 & C_{25}V_1|V_1|^2 \end{bmatrix}$, and the sum of its elements is equivalent to the

$$\text{inner product} \begin{bmatrix} V_0 & V_1 & V_0|V_0| & V_1|V_0| & V_0|V_1| & V_1|V_1| & V_0|V_0|^2 & V_1|V_0|^2 & V_0|V_1|^2 & V_1|V_1|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \\ C_{14} \\ C_{24} \\ C_{15} \\ C_{25} \end{bmatrix}.$$

If the input to the amplifier is the five-sample signal $[x(1) \ x(2) \ x(3) \ x(4) \ x(5)]$ and the corresponding output is $[y(1) \ y(2) \ y(3) \ y(4) \ y(5)]$,

then the solution to provides an estimate of the coefficient matrix.

$$\begin{bmatrix} x(2) & x(1) & x(2)|x(2)| & x(1)|x(2)| & x(2)|x(1)| & x(1)|x(1)| & x(2)|x(2)|^2 & x(1)|x(2)|^2 & x(2)|x(1)|^2 & x(1)|x(1)|^2 \\ x(3) & x(2) & x(3)|x(3)| & x(2)|x(3)| & x(3)|x(2)| & x(2)|x(2)| & x(3)|x(3)|^2 & x(2)|x(3)|^2 & x(3)|x(2)|^2 & x(2)|x(2)|^2 \\ x(4) & x(3) & x(4)|x(4)| & x(3)|x(4)| & x(4)|x(3)| & x(3)|x(3)| & x(4)|x(4)|^2 & x(3)|x(4)|^2 & x(4)|x(3)|^2 & x(3)|x(3)|^2 \\ x(5) & x(4) & x(5)|x(5)| & x(4)|x(5)| & x(5)|x(4)| & x(4)|x(4)| & x(5)|x(5)|^2 & x(4)|x(5)|^2 & x(5)|x(4)|^2 & x(4)|x(4)|^2 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{21} \\ C_{12} \\ C_{22} \\ C_{13} \\ C_{23} \\ C_{14} \\ C_{24} \\ C_{15} \\ C_{25} \end{bmatrix}$$

$$= \begin{bmatrix} y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix}$$

Use this helper function to compute coefficient matrices for the Memory polynomial and Cross-Term Memory models. The inputs to the function are the given input and output signals, the memory length, the degree of nonlinearity, and the absence or presence of cross terms.

```
function a_coef = fit_memory_poly_model(x,y,memLen,degLen,modType)
% FIT_MEMORY_POLY_MODEL
% Procedure to compute a coefficient matrix given input and output
% signals, memory length, nonlinearity degree, and model type.
%
% Copyright 2017 MathWorks, Inc.

x = x(:);
y = y(:);
xLen = length(x);

switch modType
case 'memPoly' % Memory polynomial
    xrow = reshape((memLen:-1:1)' + (0:xLen:xLen*(degLen-1)),1,[]);
    xVec = (0:xLen-memLen)' + xrow;
    xPow = x.*(abs(x).^((0:degLen-1)));
    xVec = xPow(xVec);
case 'ctMemPoly' % Cross-term memory polynomial
    absPow = (abs(x).^((1:degLen-1)));
```

```
partTop1 = reshape((memLen:-1:1)' + (0:xLen:xLen*(degLen-2)),1,[]);
topPlane = reshape(
    [ones(xLen-memLen+1,1),absPow((0:xLen-memLen)' + partTop1)].', ...
    1,memLen*(degLen-1)+1,xLen-memLen+1);
sidePlane = reshape(x((0:xLen-memLen)' + (memLen:-1:1)).', ...
    memLen,1,xLen-memLen+1);
cube = sidePlane.*topPlane;
xVec = reshape(cube,memLen*(memLen*(degLen-1)+1),xLen-memLen+1).';
end

coef = xVec\y(memLen:xLen);
a_coef = reshape(coef,memLen,numel(coef)/memLen);
```

Version History

Introduced in R2022a

References

- [1] Morgan, Dennis R., Zhengxiang Ma, Jaehyeong Kim, Michael G. Zierdt, and John Pastalan. "A Generalized Memory Polynomial Model for Digital Predistortion of Power Amplifiers." *IEEE Transactions on Signal Processing*. Vol. 54, No. 10, October 2006, pp. 3852-3860.
- [2] Gan, Li, and Emad Abd-Elrady. "Digital Predistortion of Memory Polynomial Systems using Direct and Indirect Learning Architectures". *Proceedings of the Eleventh IASTED International Conference on Signal and Image Processing (SIP)* (F. Cruz-Roldán and N. B. Smith, eds.), No. 654-802. Calgary, AB: ACTA Press, 2009.

Extended Capabilities

C/C++ Code Generation

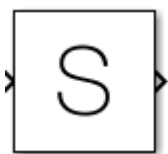
Generate C and C++ code using Simulink® Coder™.

See Also

Amplifier | Mixer | Power Amplifier

S-parameters

Model RF complex baseband S-parameters in Simulink



Libraries:
RF Blockset / Idealized Baseband

Description

Use the Idealized Baseband S-parameter block to model RF complex baseband S-parameters in Simulink®. You can input your S-parameters as a Touchstone file, an `sparameters` object, or as a `rational` object. If you describe the S-parameters using a Touchstone file or an `sparameters` object, the software uses rational fitting to derive a `rational` object. You can also model your S-parameters data in either the time or frequency domain and plot the S-parameters characteristics.

If you want to design S-parameters with impedance mismatches, use the Circuit Envelope S-Parameters block.

Ports

Input

Port_1 — Time-dependent input signal
real scalar | real column | complex scalar | complex column

Time-dependent input signal, specified as a real scalar, real column, complex scalar, or complex column. A column represents consecutive points in time.

Data Types: `double` | `single`

Output

Port_1 — Time-dependent output signal
complex scalar | complex column

Time-dependent output signal, returned as a complex scalar or complex column. The output time-dependent signal is equal in size to the input time-dependent signal.

Data Types: `double` | `single`

Parameters

Main

Data source — Data source to model RF complex baseband S-parameters
Data file (default) | `S-parameters` object | `Rational` object

Data source to model RF complex baseband S-parameters, specified as one of the following:

- **Data file** — Select this option to specify a Touchstone file with the extension `.s2p`. The block ignores noise and nonlinearity data in imported files.
- **S-parameters object** — Select this option to specify an `sparameters` object.
- **Rational object** — Select this option to specify a `rational` object.

Data file — Two-port Touchstone data file
`unitygain.s2p` (default) | character vector

Name of the two-port Touchstone file, specified as a character vector.

Dependencies

To enable this parameter, set **Data source** to `Data file`.

Sparameter object — Use `sparameters` object as data source
`sparameters('unitygain.s2p')` (default)

Select this parameter to use an `sparameters` object as the data source to model the S-parameters.

Dependencies

To enable this parameter, set **Data source** to `S-parameters object`.

Rational object — Use `rational` object as data source
`rational('unitygain.s2p','TendsToZero',false)` (default)

Select this parameter to use a `rational` object as the data source to model the S-parameters.

Dependencies

To enable this parameter, set **Data source** to `Rational object`.

Carrier frequency (Hz) — Carrier frequency at which S-parameters are modeled
`1e9` Hz (default) | positive real scalar

Carrier frequency at which the S-parameters are modeled, specified as a positive real scalar in Hz.

Simulate using — Type of simulation to run
`Interpreted execution` (default) | `Code generation`

Type of simulation to run, specified as one of the following:

- **Interpreted execution** — Simulate the model using the MATLAB interpreter. This option shortens the startup time, but the speed of the subsequent simulations is slower than when you use the `Code generation` option. Use this mode to debug the source code of the block.
- **Code generation** — Simulate the model using generated C code. The first time you run a simulation, Simulink generates C code for the block. The C code is reused for subsequent simulations as long as the model does not change. This option requires additional startup time, but the speed of the subsequent simulations is faster than when you use the `Interpreted execution` option.

Modeling

Modeling domain — Modeling domain
`Time domain (Fixed step)` (default) | `Time domain (Continuous)` | `Frequency domain (Digital filter)`

Modeling domain, specified as one of the following:

- Time domain (Fixed step) — Model using fixed-step solvers (NDF2, Trapezoidal, and Backward Euler)
- Time domain (Continuous) — Model using continuous or stiff solvers (ode15s, ode23s, ode23t, and ode23tb)
- Frequency domain (Digital filter) — model using a 1-D digital filter.

Solver — Time-domain solvers

NDF2 (default) | Trapezoidal | Backward Euler | ode15s | ode23s | ode23t (trap) | ode23tb (trap+BE)

Fixed-step and continuous time-domain solvers, specified as one of the following:

Fixed-step Solvers

- NDF2 — 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 — Perform narrowband simulations. Frequency warping and the lack of damping effects make this method inappropriate for most wideband simulations.
- Backward Euler — Simulate the largest class of systems and signals. Damping effects make this solver suitable for wideband simulation, but overall accuracy is low.

Continuous Solvers

- ode15s — Solve stiff differential equations and DAEs using a variable order method
- ode23s — Solve stiff differential equations using a low-order method
- ode23t (trap) — Solve moderately stiff ODEs and DAEs using a trapezoidal rule
- ode23tb (trap+BE) — Solve stiff differential equations using a trapezoidal rule and backward differentiation formula

Dependencies

To enable:

- Fixed-step time-domain solvers, set **Modeling domain** to Time domain (Fixed step).
- Continuous time-domain solvers, set **Modeling domain** to Time domain (Continuous).

Pole fitting options — Rational fit fitting options

Share all poles (default) | Fit individually | Share poles by columns

Rational fit fitting options to control the complex frequency-dependent data fitting process, specified as Share all poles, Fit individually, or Share poles by columns.

Dependencies

To set this parameter, set **Data source** to either Data file or S-parameters object.

Rational arguments — Rational fit arguments to control the fit

{ 'Tolerance', -40, 'TendsToZero', false, 'MaxPoles', 48, 'Qlimit', inf } (default) | cell array

Rational fit fitting arguments to control the fit, specified as a cell array.

Dependencies

To set this parameter, set **Data source** to either `Data file` or `S-parameters object`.

FIR filter length — Filter length of 1-D digital filter

128 (default) | real positive integer

Filter length of the 1-D digital filter or impulse response duration, specified as a real positive integer.

Dependencies

To set this parameter, set **Modeling domain** to `Frequency domain (Digital filter)`.

Visualization

First plot type — Plot type

S21 (default) | Phase delay | Group delay | Impulse response | Step response

Plot type of the first plot, specified as one of the following:

- S21
- Phase delay
- Group delay
- Impulse response
- Step response

Impulse response arguments — Input arguments to plot impulse response

[1e-11, 1e4] (default) | two-element vector

Input arguments to plot the impulse response, specified as a two-element vector. Plot the impulse response by specifying the sample time of the input signal and the number of samples in this parameter. For more information, see `impulse`.

Dependencies

To set this parameter, set **First plot type** or **Second plot type** to `Impulse response`.

Step response arguments — Input arguments to plot step response

[1e-11, 1e4, 1e-10] (default) | three-element vector

Input arguments to plot the step response, specified as a three-element vector. Plot the step response by specifying the sample time of the input signal, the number of samples, and the amount of time required for the step signal to reach the maximum value in this parameter. For more information, see `stepresp`.

Dependencies

To set this parameter, set **First plot type** or **Second plot type** to `Step response`.

First plot y-axis units — Y-axis units of first plot

Magnitude (dB) (default) | Magnitude (linear) | Angle (degrees) | Real | Imaginary

Y-axis units of the first plot, specified as one of the following:

- Magnitude (dB)

- Magnitude (linear)
- Angle (degrees)
- Real
- Imaginary

Dependencies

To set this parameter, set **First plot type** to S21.

Second Plot type — Plot type of second plot

None (default) | S21 | Phase delay | Group delay | Impulse response | Step response

Plot type of the second plot, specified as one of the following:

- None
- S21
- Phase delay
- Group delay
- Impulse response
- Step response

Second plot y-axis units — Y-axis units of second plot

Magnitude (dB) (default) | Magnitude (linear) | Angle (degrees) | Real | Imaginary

Y-axis units of the second plot, specified as one of the following:

- Magnitude (dB)
- Magnitude (linear)
- Angle (degrees)
- Real
- Imaginary

Dependencies

To set this parameter, set **Second plot type** to S21.

X-axis frequency points (Hz) — Frequency points to plot S-parameter data

logspace(8,10,2001) (default) | positive real vector

Frequency points to plot the S-parameter data, specified as a positive real vector in Hz.

X-axis scale — Linear or logarithmic X-axis scale

Linear (default) | Logarithmic

Select this parameter to plot your data on a linear or logarithmic X-axis scale. Specify **Linear** or **Logarithmic**.

Y-axis scale — Linear or logarithmic Y-axis scale

Linear (default) | Logarithmic

Select this parameter to plot your data in a linear or logarithmic Y-axis scale. Specify **Linear** or **Logarithmic**.

Plot Sparameter characteristics — Plot S-parameters data
button

Select this button to plot the characteristics of the S-parameters data.

Version History

Introduced in R2023a

Extended Capabilities

C/C++ Code Generation

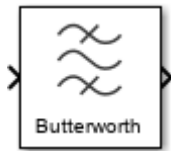
Generate C and C++ code using Simulink® Coder™.

See Also

[Amplifier](#) | [Mixer](#) | [Power Amplifier](#) | [Filter](#)

Filter

Filter RF complex baseband signals in Simulink



Libraries:

RF Blockset / Idealized Baseband

Description

Use the Idealized Baseband Filter block to filter RF complex baseband signals in Simulink. You can use the Butterworth, Chebyshev, or inverse Chebyshev design methods to design your filter. You can also model your filter either in the time- or frequency-domain and plot the filter characteristics.

To design an RF filter with impedance mismatches, use the Circuit Envelope Filter block.

Note The Idealized Baseband Filter block replaces the Lowpass RF Filter, Highpass RF Filter, Bandpass RF Filter, and Bandstop RF Filter blocks.

Ports

Input

Port_1 — Time-dependent input signal

real scalar | real column | complex scalar | complex column

Time-dependent input signal, specified as a real scalar, real column, complex scalar, or complex column. A column represents consecutive points in time.

Data Types: double | single

Output

Port_1 — Time-dependent output signal

complex scalar | complex column

Time-dependent output signal, returned as a complex scalar or complex column. The output time-dependent signal is equal in size to the input time-dependent signal.

Data Types: double | single

Parameters

Main

Design method — Filter design method

Butterworth (default) | Chebyshev | Inverse Chebyshev

Filter design method, specified as one of the following:

- Butterworth
- Chebyshev
- Inverse Chebyshev

Filter type — Filter response type

Lowpass (default) | Highpass | Bandpass | Bandstop

Filter response type, specified as one of the following:

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

Implement using filter order — Implement using filter order

off (default) | on

Select this parameter to implement the filter order manually.

Filter order — Filter order

3 (default) | integer greater than 2

Filter order, specified as a integer greater than 2. If you set **Filter design** to Lowpass or Highpass, specify the number of lumped storage elements. If you set **Filter design** to Bandpass or Bandstop, specify twice the number of lumped storage elements.

Dependencies

To enable this parameter, select **Implement using filter order**.

Passband frequency (Hz) — Passband frequency

1e9 (default) | positive real scalar | positive ascending two-tuple vector

Passband frequency for lowpass, highpass, and bandpass filter types, specified as a positive real scalar or positive ascending two-tuple vector in Hz. Based on the **Filter Type**, accepted value types and default values differ as follows.

Filter Type	Value Type	Default Values (Hz)
Lowpass	Positive real scalar	1e9
Highpass	Positive real scalar	2000000000
Bandpass	Positive ascending two-tuple vector	[2000000000 3000000000]

Dependencies

To enable this parameter, set **Filter type** to Lowpass, Highpass, or Bandpass.

Passband attenuation (dB) — Passband attenuation

$10 \cdot \log_{10}(2)$ (default) | positive real scalar

Passband attenuation of the filter, specified as a positive real scalar in dB. For bandpass filters, this value is applied equally to both edges of the passband.

Dependencies

To enable this parameter, set **Filter type** to Lowpass, Highpass, or Bandpass.

Stopband frequencies (Hz) — Stopband frequencies for bandstop filters
2000000000 (default) | positive real scalar | positive ascending two-tuple vector

Stopband frequencies for bandstop filters, specified as either a positive real scalar or positive ascending two-tuple vector in Hz. Based on the **Filter Type** and **Implement using filter order** parameters, accepted value types and default values differ as follows.

Filter Type	Value Type	Default Values (Hz)	To enable this parameter
Lowpass	Positive real scalar	2000000000	Set Filter type to Lowpass and clear Implement using filter order .
Highpass	Positive real scalar	1000000000	Set Filter type to Highpass and clear Implement using filter order .
Bandpass	Positive ascending two-tuple vector	[1500000000 3500000000]	Set Filter type to Bandpass and clear Implement using filter order .
Bandstop	Positive ascending two-tuple vector	[2100000000 2900000000]	Set Filter type to Bandstop and clear or select Implement using filter order .

Stopband attenuation (dB) — Stopband attenuation
40 (default) | positive real scalar

Stopband attenuation, specified as a positive real scalar greater than the **Passband attenuation (dB)** value in dB.

Dependencies

To enable this parameter, you can either:

- Set **Filter type** to Lowpass, Highpass, or Bandpass and clear **Implement using filter order**.
- Set **Filter type** to Bandstop, select **Implement using filter order**.

Carrier frequency (Hz) — Center of signal bandwidth
1e9 Hz (default) | positive real scalar

Center of the signal bandwidth with respect to the filter transfer function, specified as a positive real scalar in Hz. The signal bandwidth is $\frac{1}{\text{Simulink Step Size}}$.

Simulate using — Type of simulation to run
Interpreted execution (default) | Code generation

Type of simulation to run, specified as one of the following:

- **Interpreted execution** — Simulate the model using the MATLAB interpreter. This option shortens the startup time, but the speed of the subsequent simulations is slower than when you use the **Code generation** option. Use this mode to debug the source code of the block.
- **Code generation** — Simulate the model using generated C code. The first time you run a simulation, Simulink generates C code for the block. The C code is reused for subsequent simulations as long as the model does not change. This option requires additional startup time, but the speed of the subsequent simulations is faster than when you use the **Interpreted execution** option.

Modeling

Modeling domain — Modeling domain
Time domain (Fixed step) (default) | Time domain (Continuous) | Frequency domain (Digital filter)

Modeling domain, specified as one of the following:

- **Time domain (Fixed step)** — Model using fixed-step solvers (NDF2, Trapezoidal, and Backward Euler)
- **Time domain (Continuous)** — Model using continuous or stiff solvers (ode15s, ode23s, ode23t, and ode23tb)
- **Frequency domain (Digital filter)** — model using a 1-D digital filter.

Solver — Time-domain solvers
NDF2 (default) | Trapezoidal | Backward Euler | ode15s | ode23s | ode23t (trap) | ode23tb (trap+BE)

Fixed-step and continuous time-domain solvers, specified as one of the following:

Fixed-step Solvers

- **NDF2** — 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** — Perform narrowband simulations. Frequency warping and the lack of damping effects make this method inappropriate for most wideband simulations.
- **Backward Euler** — Simulate the largest class of systems and signals. Damping effects make this solver suitable for wideband simulation, but overall accuracy is low.

Continuous Solvers

- **ode15s** — Solve stiff differential equations and DAEs using a variable order method
- **ode23s** — Solve stiff differential equations using a low-order method
- **ode23t (trap)** — Solve moderately stiff ODEs and DAEs using a trapezoidal rule
- **ode23tb (trap+BE)** — Solve stiff differential equations using a trapezoidal rule and backward differentiation formula

Dependencies

To enable:

- Fixed-step time-domain solvers, set **Modeling domain** to Time domain (Fixed step).
- Continuous time-domain solvers, set **Modeling domain** to Time domain (Continuous).

FIR filter length — Filter length of 1-D digital filter

128 (default) | real positive integer

Filter length of the 1-D digital filter or impulse response duration, specified as a real positive integer.

Dependencies

To set this parameter, set **Modeling domain** to Frequency domain (Digital filter).

Visualization

First plot type — Plot type

S21 (default) | Phase delay | Group delay | Impulse response | Step response

Plot type of the first plot, specified as one of the following:

- S21
- Phase delay
- Group delay
- Impulse response
- Step response

Impulse response arguments — Input arguments to plot impulse response

[1e-11, 1e4] (default) | two-element vector

Input arguments to plot the impulse response, specified as a two-element vector. Plot the impulse response by specifying the sample time of the input signal and the number of samples in this parameter. For more information, see `impz`.

Dependencies

To set this parameter, set **First plot type** or **Second plot type** to Impulse response.

Step response arguments — Input arguments to plot step response

[1e-11, 1e4, 1e-10] (default) | three-element vector

Input arguments to plot the step response, specified as a three-element vector. Plot the step response by specifying the sample time of the input signal, the number of samples, and the amount of time required for the step signal to reach the maximum value in this parameter. For more information, see `stepresp`.

Dependencies

To set this parameter, set **First plot type** or **Second plot type** to Step response.

First plot y-axis units — Y-axis units of first plot

Magnitude (dB) (default) | Magnitude (linear) | Angle (degrees) | Real | Imaginary

Y-axis units of the first plot, specified as one of the following:

- Magnitude (dB)
- Magnitude (linear)
- Angle (degrees)
- Real
- Imaginary

Dependencies

To set this parameter, set **First plot type** to S21.

Second Plot type — Plot type of second plot

None (default) | S21 | Phase delay | Group delay | Impulse response | Step response

Plot type of the second plot, specified as one of the following:

- None
- S21
- Phase delay
- Group delay
- Impulse response
- Step response

Second plot y-axis units — Y-axis units of second plot

Magnitude (dB) (default) | Magnitude (linear) | Angle (degrees) | Real | Imaginary

Y-axis units of the second plot, specified as one of the following:

- Magnitude (dB)
- Magnitude (linear)
- Angle (degrees)
- Real
- Imaginary

Dependencies

To set this parameter, set **Second plot type** to S21.

X-axis frequency points (Hz) — Frequency points to plot S-parameter data

logspace(8,10,2001) (default) | positive real vector

Frequency points to plot the S-parameter data, specified as a positive real vector in Hz.

X-axis scale — Linear or logarithmic X-axis scale

Linear (default) | Logarithmic

Select this parameter to plot your data on a linear or logarithmic X-axis scale. Specify **Linear** or **Logarithmic**.

Y-axis scale — Linear or logarithmic Y-axis scale

Linear (default) | Logarithmic

Select this parameter to plot your data in a linear or logarithmic Y-axis scale. Specify **Linear** or **Logarithmic**.

Plot Sparameter characteristics — Plot S-parameters data button

Select this button to plot the characteristics of the S-parameters data.

Algorithms

Determine FIR Filter Coefficients

The software calculates the discrete FIR filter coefficients using the **Carrier frequency (Hz)** and **FIR filter length** parameters and the transfer function specified using the derived filter poles and zeros.

The following steps are followed to determine the **Discrete FIR Filter** block direct form coefficients:

- 1 Determine the frequency points contained in the bandwidth centered around the carrier frequency using this formula.

$$Freqs = \left[\left(\left(\frac{-fftLen}{2} \right) : \left(\frac{fftLen}{2} - 1 \right) \right) \times \left(\frac{1}{t_{step} \times fftLen} \right) \right] + f_{carrier}$$

where

$f_{carrier}$ – Carrier frequency in Hz

$fftLen$ – FIR filter length

t_{step} – Time step of the filter

- 2 Use the `rffilter` object and its functions to determine the rational analog filter transfer function from the mask parameters in [z,p,k] form.
- 3 Determine transfer function values for the frequency points in step one using this formula.

$$Transf(Freqs) = k \times \frac{\prod(z - 2\pi j \times Freqs)}{\prod(p - 2\pi j \times Freqs)}$$

- 4 Determine the discrete FIR filter coefficients using this formula.

$$Coeffs = \left(-1^{FilterOrder} \right) \times \text{ifft}(fftshift(Transf(Freqs)))$$

Version History

Introduced in R2023a

R2023a: Lowpass RF, Highpass RF, Bandpass RF, and Bandstop RF Filter blocks have been replaced

Starting in R2023a, the idealized baseband Filter block replaces the Lowpass RF Filter, Highpass RF Filter, Bandpass RF Filter, and Bandstop RF Filter blocks.

When you open a model created before R2023a containing the Lowpass RF Filter, Highpass RF Filter, Bandpass RF Filter, and Bandstop RF Filter Filter blocks, the software replace the old blocks with the new idealized baseband Filter block.

Extended Capabilities

C/C++ Code Generation

Generate C and C++ code using Simulink® Coder™.

See Also

Amplifier | Mixer | Power Amplifier | S-Parameters