RF Blockset™

Getting Started Guide

MATLAB&SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

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

Revision History

Contents

Circuit Envelope

Introduction to Circuit Envelope Simulation

1

Frequency Conversion

2

Equivalent Baseband

Circuit Envelope

Introduction to Circuit Envelope Simulation

RF Blockset Product Description

Design and simulate RF systems

RF Blockset (formerly SimRF™) provides a Simulink® model library and simulation engine for designing RF communications and radar systems.

RF Blockset lets you simulate RF transceivers and front-ends. You can model nonlinear RF amplifiers to estimate gain, noise, even-order, and odd-order intermodulation distortion, including memory effects. For RF mixers, you can predict image rejection, reciprocal mixing, local oscillator phase noise, and DC offset. RF models can be characterized using data sheet specifications or measured data such as multiport S-parameters. They can be used to accurately model adaptive architectures, including automatic gain control (AGC), digital predistortion (DPD) algorithms, and beamforming.

The RF Budget Analyzer app lets you automatically generate transceiver models and measurement test benches to validate performance and set up a circuit envelope multicarrier simulation.

With RF Blockset you can simulate RF systems at different levels of abstraction. Circuit envelope simulation enables high-fidelity, multicarrier simulation of networks with arbitrary topologies. The Equivalent Baseband library enables fast, discrete-time simulation of single-carrier cascaded systems.

Circuit Envelope Library

Use Circuit Envelope Library

The RF Blockset Circuit Envelope library provides components to model RF systems within the Simulink environment. This library extends your Simulink modeling environment with a library of blocks, including Amplifier, Mixer, Filter, and Antenna, to model RF systems. Use blocks from the Circuit Envelope library to model electrical components by specifying RF properties or by importing measured data.

The RF Blockset Circuit Envelope library enables simulation of arbitrary and cascaded RF networks with multicarrier frequencies, noise, and nonlinearities. The Elements, Junctions, Sources, Systems, Testbenches, and Utilities sublibraries enable you to model noise, nonlinear elements, and RF architecture. You can import S-parameter data files directly into RF Blockset models to specify linear networks, and you can visualize their frequency response using the S-Parameters block. Use the Testbenches sublibrary to measure and verify gain, noise figure, and intercept point. Additionally, using this library you can also author custom blocks using Simscape™ language.

Open Circuit Envelope Library

To open the main library window, type this at the MATLAB® Command Line:

```
rflib
```

Each yellow block in the window represents a library.
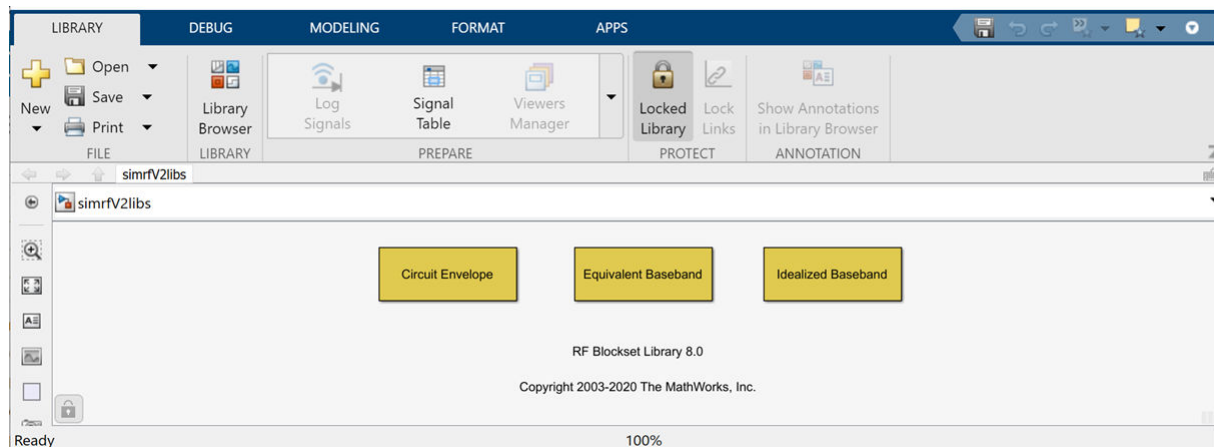
Double-click the **Circuit Envelope** block to open the library.
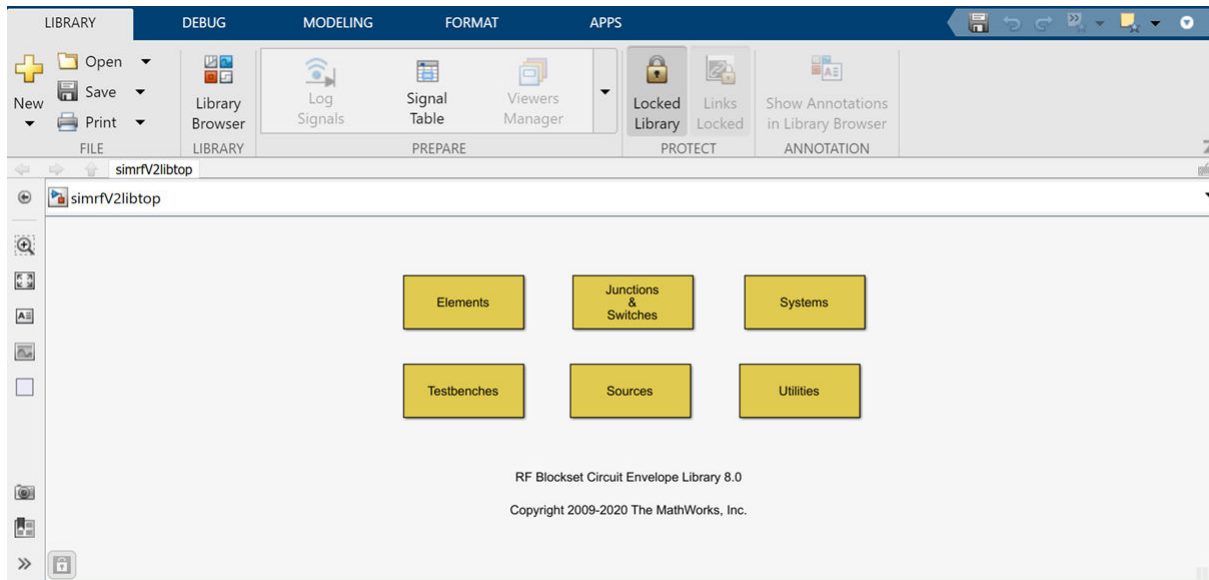
This table describes the sublibraries and the available models.

Sublibrary	Use
Elements	To model antenna, amplifier, filter, mixers, and other RF elements.
Junctions and Switches	To model ideal frequency-independent circulators, couplers, and dividers and to use switches and potentiometers.
Systems	To model a modulator/demodulator and an IQ modulator/demodulator.
Testbenches	To measure and verify gain, noise figure, intercept point, and S-parameters.
Sources	To use current or voltage in noise, sinusoidal, and continuous wave sources and power in sinusoidal and continuous wave sources.
Utilities	To define system simulation settings, convert input signal to RF Blockset signal, convert RF Blockset signal to Simulink output signal, and connect ports.

Understanding Circuit Envelope Environment

The Circuit Envelope environment comprises of groups of interconnected Circuit Envelope library blocks and algorithms that model an RF system. In the Circuit Envelope environment, all blocks fall into one of the three following categories.

Blocks Operating Within Circuit Envelope Environment

These blocks contribute to the physical representation of an RF system. Most Circuit Envelope library blocks fall into this category, including all blocks in the Elements and Sources sublibraries. For

example, a Resistor block can model a source impedance or part of a matching network, and an Amplifier block can model a physical RF amplifier. Both of these blocks model physical components.

Blocks with variable properties, such as a Variable Resistor or a VGA block, operate within the Circuit Envelope environment, although the control signals come directly from Simulink and you do not require to use cross-domain blocks.

Blocks Operating Between Circuit Envelope and Simulink Environments

These blocks, also called cross-domain blocks, provide an interface from an RF system to a larger design. The Inport and Outport blocks from the Circuit Envelope library fall in this category. For example, you can construct a signal using blocks from Communications Toolbox™ or DSP System Toolbox™ libraries, and you can input that signal into the RF Blockset environment using an RF Blockset Inport block.

Environment Independent Block

The Configuration block manipulates the environment and determines the simulation configuration. To use this block, connect it to any part of the RF system. Because the block is not a part of the physical representation of the system, it has the same effect regardless of where you connect it.

To run models containing other Circuit Envelope blocks, you must connect a Configuration block to the network.

See Also

More About

- "Circuit Envelope Basics" on page 1-6

Circuit Envelope Basics

In this section...

Introduction

The circuit envelope technique speeds up RF system simulation. In Simulink, simulating high-frequency signals requires a time step proportional to the highest frequency present in the RF system. However, the modulation frequency or envelope of the RF signals can be several orders of magnitude smaller than the highest frequency. The circuit envelope technique takes advantage of this condition to model RF signals accurately while reducing simulation time.

For example, the following figure shows the benefit of using circuit envelope with a signal $x(t)$:
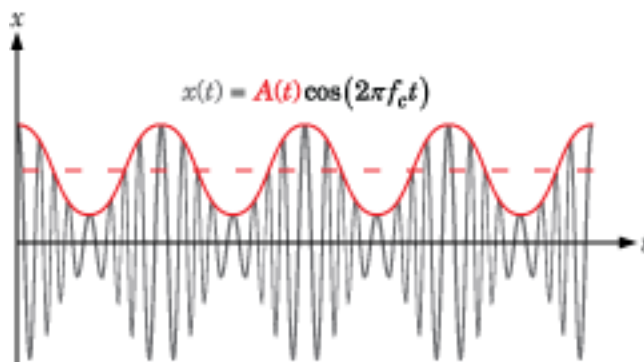
The signal consists of a time-varying modulating signal on a high-frequency carrier or center frequency. In many RF applications, the frequency of the modulating signal, $A(t)$, is smaller than the frequency of the carrier, f_c.

RF Blockset software handles the carrier $\cos(2\pi f_c t)$ analytically, so only simulates the modulating signal. RF Blockset uses two methods to handle this simulation, equivalent baseband and circuit envelope. In both methods, the simulation engine takes time steps on the scale of the modulating signal instead of the carrier.

Compared to equivalent baseband simulation, circuit envelope allows including additional non-linear effects beyond in-band spectral regrowth and is suitable for multicarrier simulation.

Using circuit envelope simulation, you can model:

- Even and odd order nonlinear effects, generating in-band, and out-of-band harmonics and spectral regrowth
- Multiport and wideband filtering (frequency selective) effects, like the effects introduced by S-parameters, including impedance mismatches

- In-band and out-of-band interfering and spur signals, including mixing effects
- DC conversion and DC offsets
- Arbitrary local oscillator signals, including phase noise
- Thermal noise generation
- Tunable RF elements controlled by Simulink signals (VGA, switches, RLC, attenuator, phase shifter, etc.)

Assumptions for Circuit Envelope Simulation

In circuit envelope simulation, you assume that the modulation envelope of the signal varies slowly with respect to the carrier. This type of simulation assumes the signal envelopes to be constant over a single cycle of the carrier (quasi-static assumption). You can assume that a signal envelope is narrowband when it has a frequency at least one order of magnitude smaller than the carrier frequency.

For real passband and wide-band modulated signals, circuit envelope provides correct results. However, the simulation can be slower than traditional time-domain (transient) simulation techniques, like the technique supported by Simulink. If your input is an ultra-wideband signal, or if you deal with many modulated signals filling the simulation spectrum, use time-domain passband simulation.

When considering the quasi-static assumption, you need to consider the effect of the nonlinearities that increase the bandwidth of the signal envelope. Signal envelopes, including in-band spectral regrowth, should be narrowband compared to the carrier frequency. Therefore, circuit envelope is less suitable for simulating hard non-linearities, like nonlinearities resulting from clipping or saturating effects.

In multicarrier signals, overlapping envelopes are not recommended. Overlapping envelopes occur when carrier frequencies are spaced at a distance smaller than the envelope bandwidth. In this case, you can reduce the simulation time step and accommodate the signal information within a single envelope. For more information, see Configuration.

What Is Circuit Envelope?

Circuit envelope is a time domain (transient) simulation superimposed on a harmonic balance analysis. The analysis is performed at discrete points in time.

Harmonic balance is a frequency-domain method for calculating the steady-state response of non-linear circuits when excited with a finite number of harmonic tones. This analysis solves the system of equations in the frequency domain and is suitable for simulating frequency-defined components such as S-parameters or transmission lines.

Harmonic balance is used by circuit envelope to analyze the system response at every time step. The simulation derives the analysis frequencies from the signal carriers. The harmonic tone coefficients are time varying and processed using transient simulation. This process provides the time-varying envelopes around the harmonic tones.

In the figure below, you see a schematic representation of a circuit envelope simulation. A modulated sinusoidal signal centered around f_{c1} is input to a nonlinear system. The output of the system has multiple harmonics, each with a time-varying envelope.

$$V_{in} = \text{Re}\{V(t)e^{j\omega_c t}\}$$

$$V_{out} = \text{Re}\{\sum_{k=0}^{N} V_k(t)e^{j\omega_{ck} t}\}$$
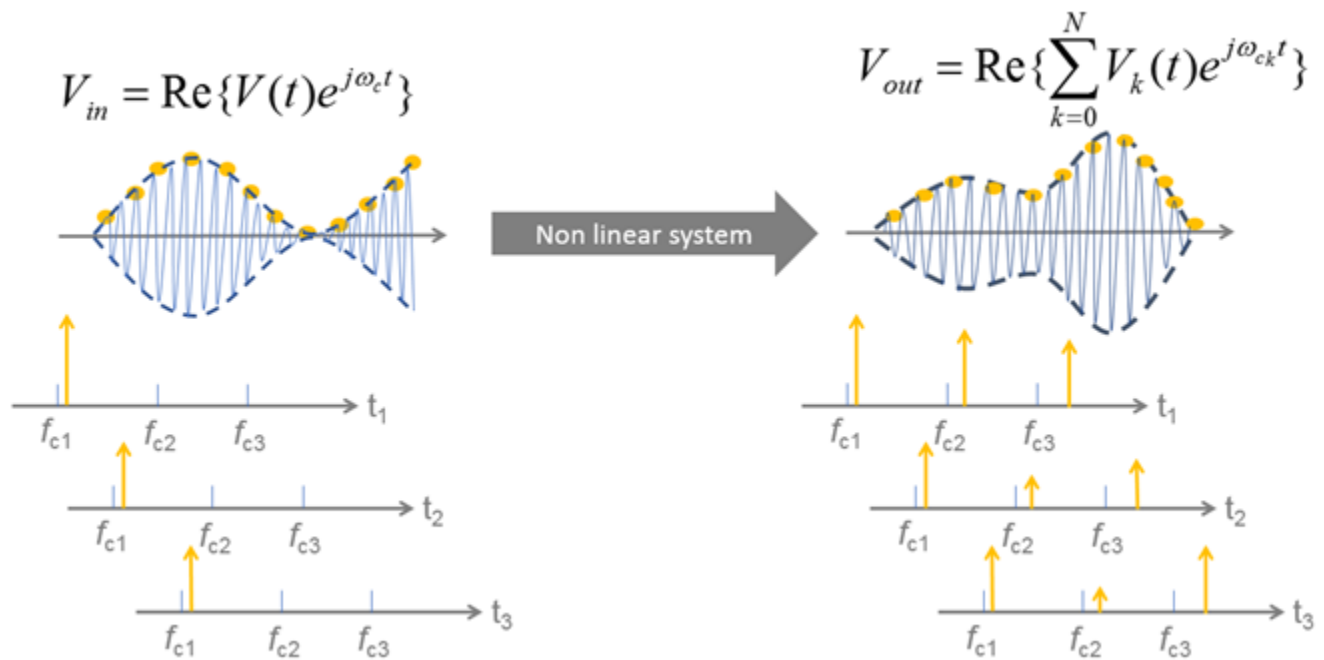
In circuit envelope, the time step should be small enough to capture the bandwidth of the envelope and not the maximum frequency (carrier) of the signal. A smaller simulation time step corresponds to a larger simulation (envelope) bandwidth, and thus a slower simulation.

The circuit envelope simulation time step is set in the Configuration block. The simulation time step must be sufficiently small to capture the signal modulation (bandwidth) and the in-band spectral regrowth caused by the system nonlinearity. At the same time, the simulation time step should be as large as possible to increase simulation speed. You can find a tradeoff between accuracy and simulation speed, by using a simulation time step within this range of values

- Use a simulation time step value less than 1/(2*bandwidth) to fulfill Nyquist criterion and correctly sample your signal modulation.

- Use a simulation time step value greater than or equal to 1/(8*bandwidth) to have maximum accuracy at the envelope edges. You can use these ranges of values when simulating S-parameters blocks, filters, and frequency defined components. This time step value also captures the in-band spectral regrowth caused by odd order nonlinearity.

Time Domain Simulation and Frequency Domain Analysis

Circuit envelope combines time-domain simulation with frequency-domain analysis, there are two edge cases of particular interest.

- If the carrier frequency of all signal sources is 0, then the simulation is reduced to a purely transient (real-passband) simulation with fixed time step. No harmonic balance is performed. This configuration does not speed up the RF system simulation.

- If the simulation stop time is equal to 0, then the simulation is reduced to a purely static nonlinear analysis (harmonic balance) of the system. No time-domain simulation is performed. This configuration is beneficial for the steady-state analysis of the RF system, for example to understand the resulting energy allocation from many signals, or for AC system analysis.

How Circuit Envelope Simulation Works in RF Blockset

Consider a RF Blockset circuit. You can divide this circuit into three sections: input signal generation, RF subsystem, output signal visualization.
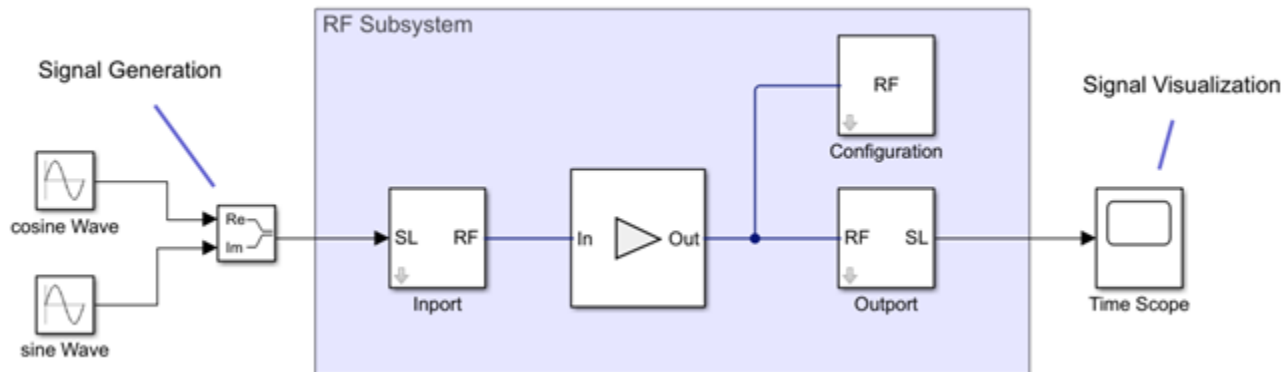
Input Signal Generation

You can use two kinds of input signal sources:

- Simulink signal source
- RF Blockset signal source

If you are using a Simulink signal source, you need a gateway into the RF subsystem (Inport block). The Simulink input signal represents the modulation of your RF signal. The signal can be complex or real based on the information it carries. To model a constant modulation on a carrier in the Circuit Envelope simulation environment use Continuous Wave block. You can also model noise using current or voltage noise source in RF systems using Noise block.

If the input signal is a vector, each element of the vector represents the envelope signal to be modulated around a certain carrier frequency. You specify the carrier frequencies of the input signal in the Inport block.

RF Subsystem

The RF subsystem (highlighted in blue) consists of three main blocks: Inport, Configuration, and Outport. You can include as many Inport or Outport blocks as you need in your RF subsystem.

- `Inport`: The Inport block performs an ideal frequency shift of the input signal around the carrier frequency by implementing a complex frequency multiplication. If your input signal is a vector with each element representing a separate envelope, you can specify separate carrier frequencies for each of them.

  You can also specify a zero-carrier frequency corresponding to a real passband signal. In this case, the imaginary part of the input signal is neglected.

  For more information, see Inport.

- `Configuration`: You use the Configuration block to specify the following:

- The simulation time step, which determines the bandwidth of the envelope simulation. All signal envelopes have the same bandwidth. To avoid resampling of the input signal and aliasing, use the same time step as the signal inputs from Simulink for multiple inport systems. For single inport systems, use slower input signals with interpolation filter checkbox.

- The harmonic order of simulation, which determines the total number of simulation frequencies used to perform the harmonic balance analysis. The auto option gives a conservative choice of harmonic frequencies. The simulation time is directly proportional to the total number of simulation frequencies. If your circuit operates in mildly nonlinear conditions, you can speed up the simulation by reducing the harmonic order.

- Normalization of the carrier power, used to scale the average signal power with respect to its root mean squared value. Use this option to scale the average power of the signal envelope with respect to the root mean square power of its real passband representation.

- The temperature and the seed used for generating thermal noise.

  For more information, see Configuration.

- `Outport`: The Outport block is the return gateway from circuit envelope to the Simulink environment. If you specify a carrier frequency other than zero, the block returns the complex envelope of the signal around the specified carrier. Outport blocks are used to probe modulated signals on the specified carriers for viewing using the Spectrum Analyzer or for further signal processing.

  For more information see, Outport.

Output Signal Visualization

You can use Simulink sinks to visualize the RF signals. To access circuit envelope signals in Simulink, you need an outport block to act as a gateway out of the RF subsystem. The Outport block probes the signal envelopes centered around the specified carrier frequencies.

By using a time scope, you can inspect the time-varying content of the modulated signals without plotting the respective carrier frequency.

By using a spectrum analyzer, you can inspect the spectral content of the modulated signals, which are implicitly centered around the different carrier frequencies.

See Also

More About

- "Circuit Envelope Library" on page 1-3
- "Simulate High Frequency Components" on page 1-18

Using RF Measurement Testbench

In this section...

Use the RF Measurement testbench to verify the cumulative gain, noise figure, and nonlinearity (IP3) values of an RF-to-RF system. To use the testbench, create a system in the **RF Budget Analyzer** app and click `Export > Export to Measurement Testbench`.
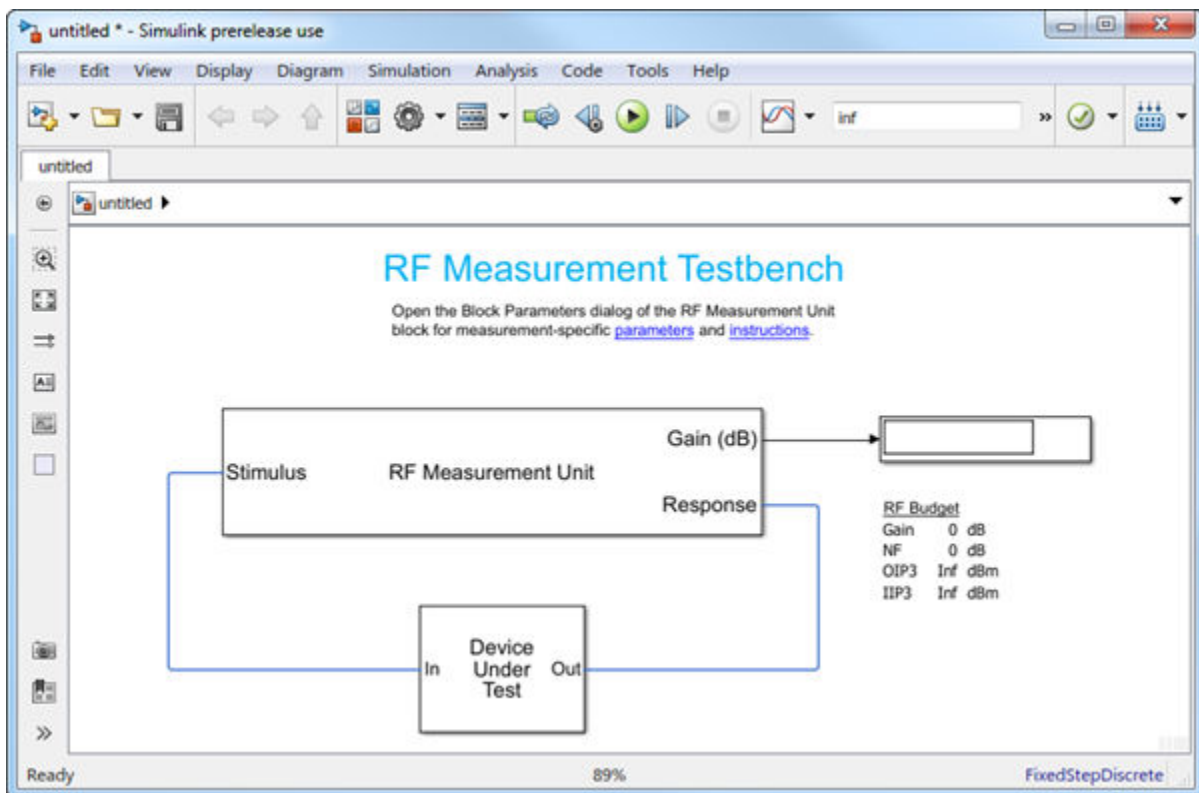
The testbench is made up of two subsystems:

- `RF Measurement Unit`
- `Device Under Test`

The testbench display shows the verified output values of gain, NF (noise figure), and IP3 (third-order intercept).

Device Under Test
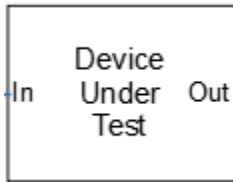
The Device Under Test subsystem contains the RF system exported from the app.
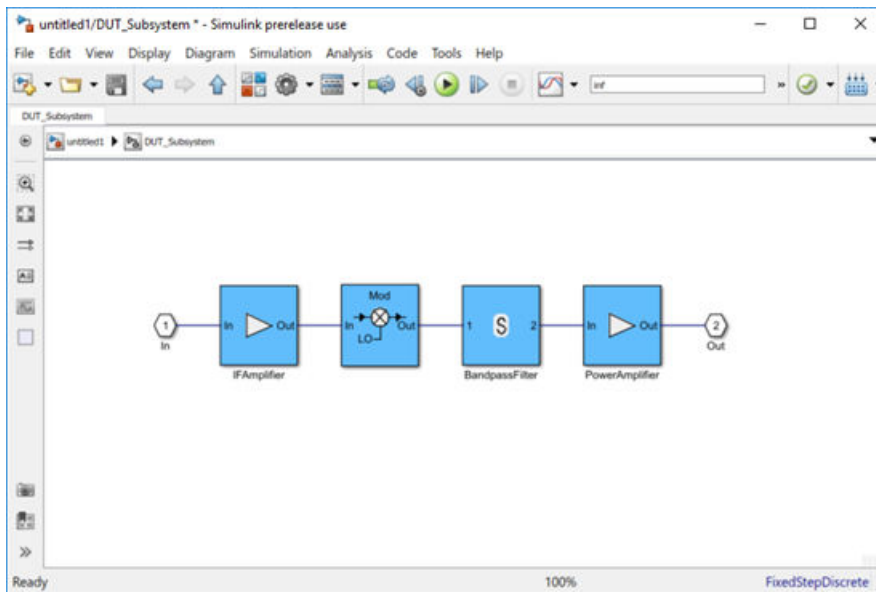
RF Measurement Unit
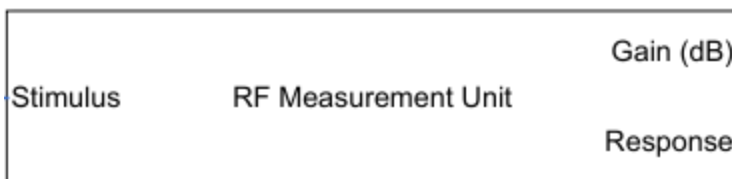
The RF Measurement Unit subsystem consists of a Simulink Controller and RF Blockset Circuit Envelope interface. The RF Blockset interface is used as input and output from the DUT.
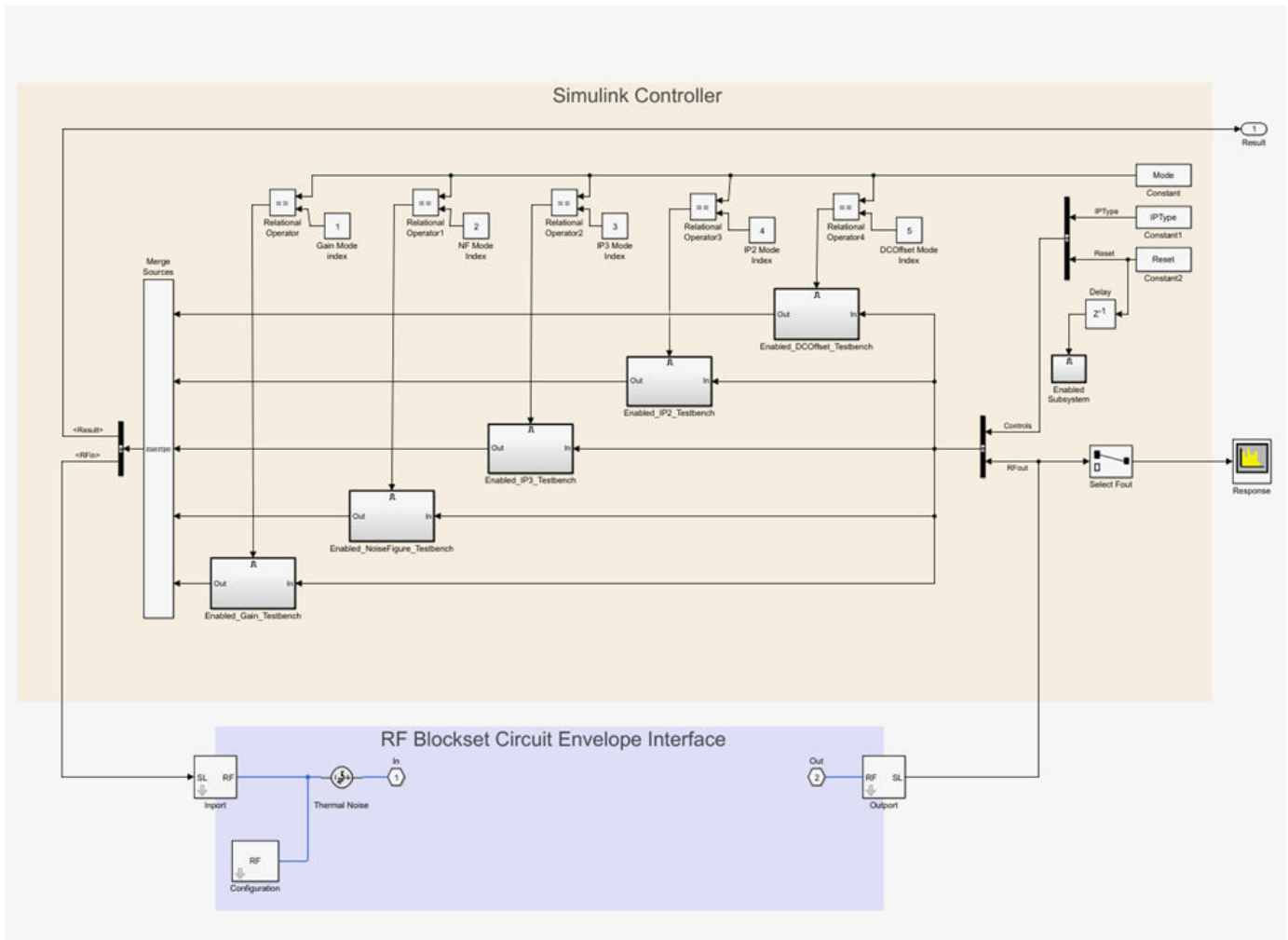
RF Measurement Unit Parameters
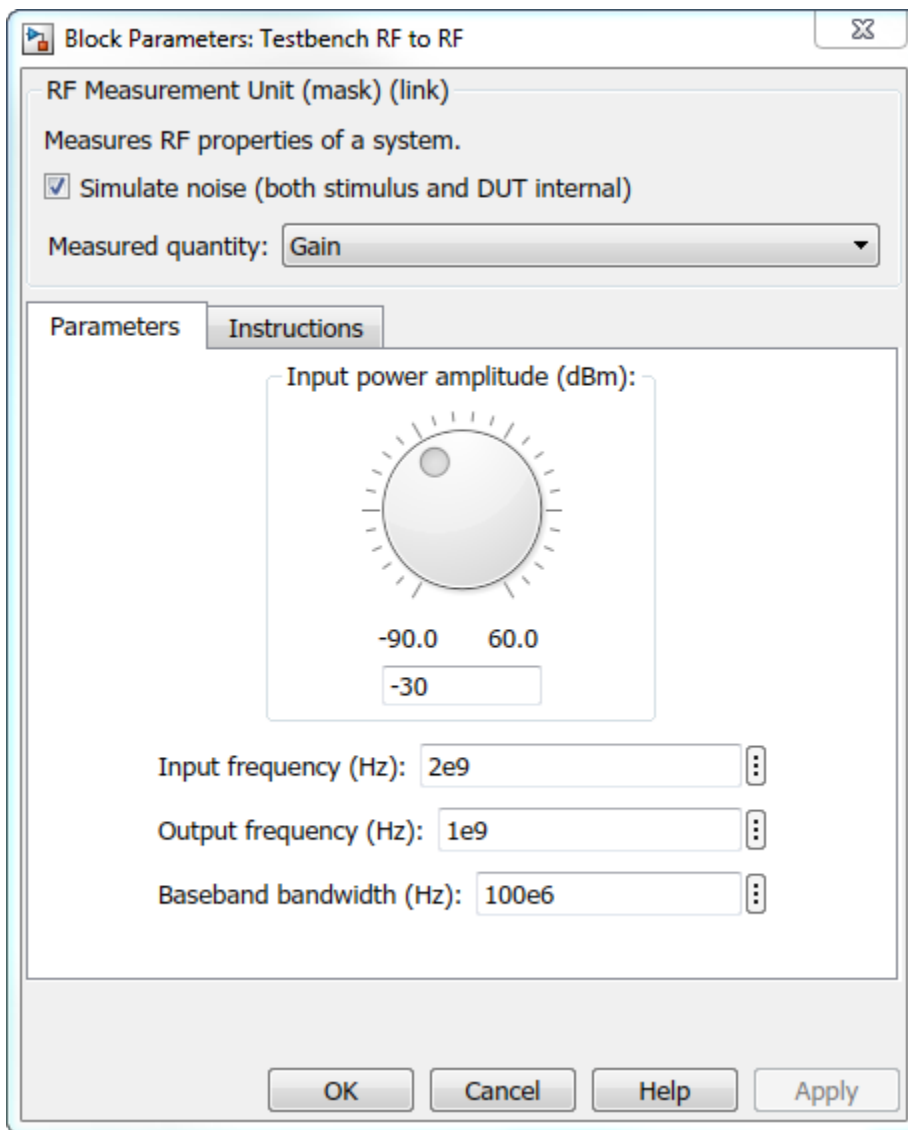
Double-click the RF Measurement Unit subsystem block to open the parameters.

- **Simulate noise (both stimulus and DUT internal)** — Select this check box to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.
- **Measured quantity** — Choose the quantity you want to verify from:

  - Gain – Measure the transducer gain of the converter, assuming a load of 50 ohm. If you choose only I or only Q from **Response branch**, you see only half the value of the measured gain.
  - NF – Measure the noise figure value at the output of the converter.
  - IP3 – Measure the output or input third-order intercept (IP3).
  - IP2 – Measure the output or input second-order intercept (IP2).
  - DC Offset – Measure the DC level interference centered on the desired signal due to LO leakage mixing with input signal.

The contents in the **Instructions** tab changes according to the **Measured quantity**.

- **IP Type** — Choose the type of intercept points (IP) to measure: `Output referred` or `Input referred`.

  By default, the testbench measures `Output referred`. This option is available when you set the **Measured quantity** to IP2 or IP3.

Parameters Tab

- **Input power amplitude (dBm)** — Input power to the DUT. You can change the input power by manually specifying or by turning the knob. When measuring `DC Offset`, this input field is **Input RMS voltage (dBmV)**, because the Offset is measured in voltage units. The specified voltage represents the voltage falling on the input ports of the DUT.

- **Input frequency (Hz)** — Carrier frequency of the DUT.

- **Output frequency (Hz)** — Output frequency of the DUT.

- **Baseband bandwidth (Hz)** — Bandwidth of the input signal.

- **Ratio of test tone frequency to baseband bandwidth** — Position of the test tones used for IP3 measurements. By default, the value is `1/8`.

  This option is available when you set the **Measured quantity** to IP2, IP3, or `DC Offset`.

Instructions

```
┌─────────────────────────────────────────────────────────┐
│ 🔲 Block Parameters: Testbench RF to RF              ☒   │
├─────────────────────────────────────────────────────────┤
│ ┌─ RF Measurement Unit (mask) (link) ──────────────────┐ │
│ │ Measures RF properties of a system.                   │ │
│ │ ☑ Simulate noise (both stimulus and DUT internal)    │ │
│ │                                                        │ │
│ │ Measured quantity: │ Gain                      ▼ │   │ │
│ │                                                        │ │
│ │ ┌──────────┬────────────┐                            │ │
│ │ │Parameters│Instructions│                            │ │
│ │ ├──────────┴────────────┴───────────────────────────┐│ │
│ │ │1. For accurate gain measurement, please uncheck    ││ │
│ │ │the 'Simulate noise' checkbox.                       ││ │
│ │ │                                                     ││ │
│ │ │2. For high input power, the measured gain may be   ││ │
│ │ │affected by nonlinearities of the Device Under Test ││ │
│ │ │(DUT) and differ from the gain calculated in the RF ││ │
│ │ │budget app. In this case, use the knob to reduce the││ │
│ │ │input power amplitude value until the resulting gain ││ │
│ │ │value settles down.                                  ││ │
│ │ │                                                     ││ │
│ │ │3. Other discrepancies between the measured gain and ││ │
│ │ │that calculated in the RF budget app may originate   ││ │
│ │ │from the more realistic account of the DUT           ││ │
│ │ │performance obtained using the SimRF simulation. In  ││ │
│ │ │this case, verify that the DUT performance is        ││ │
│ │ │evaluated correctly using RF budget calculations. For││ │
│ │ │more details, see the RF budget app documentation.   ││ │
│ │ └─────────────────────────────────────────────────────┘│ │
│ │                                                        │ │
│ │         [ OK ]  [ Cancel ]  [ Help ]  [ Apply ]       │ │
│ └───────────────────────────────────────────────────────┘ │
└─────────────────────────────────────────────────────────┘
```

Instructions for Gain Verification

• Clear **Simulate noise (both stimulus and DUT)** for accurate gain verification. Select the check box for account for noise.

• Change the **Input power amplitude (dBm)** or turn the knob to reduce the input power amplitude. For high input power, nonlinearities in the DUT can affect the gain measurements.

Instructions for NF Verification

• The testbench verifies the spot NF calculated. This calculation assumes a frequency-independent system within a given bandwidth. To simulate a frequency-independent system and calculate the correct NF value, reduce the baseband bandwidth until this condition is fulfilled. In common RF systems, the bandwidth should be reduced below 1 kHz for NF testing.

- Change **Input power amplitude (dBm)** or turn the knob to reduce or increase the input power amplitude. For high input power, nonlinearities in the DUT can affect the NF measurements. For low input power, the signal is too close or below the noise floor of the system. As a result, the NF fails to converge.

Instructions for OIP3 and IIP3 Verification

- Clear **Simulate noise (both stimulus and DUT)** for accurate OIP3 and IIP3 verification.
- Change **Input power amplitude (dBm)** or turn the knob to reduce the input power amplitude. For high input power, higher-order nonlinearities in the DUT can affect the OIP3 and IIP3 measurements.

For all measurement verifications using the testbench, you cannot correct result discrepancies using the **RF Budget Analyzer** app. The RF Blockset testbench provides true RF circuit simulation that incorporates RF phenomena including saturation and interaction between multiple tones and harmonics in nonlinear devices. These RF phenomena are not yet incorporated in **RF Budget Analyzer**, leading to some differences in the values between the testbench and the app.

Instructions for DC Offset Measurement

- Clear **Simulate noise (both stimulus and DUT)** for accurate DC offset measurement.
- Correct calculation of the DC offset assumes a frequency-independent system in the frequencies surrounding the test tones. Reduce the frequency separation between the test tones or reduce the baseband bandwidth until this condition is fulfilled. In common RF systems, the bandwidth is reduced below 1 KHz for DC offset testing.
- . Change **Input RMS voltage amplitude (dBmV)** or turn the knob to reduce the input RMS voltage amplitude. For high input RMS voltage, higher-order nonlinearities in the DUT can affect the DC offset measurements

See Also

IIP3 Testbench | Noise Figure Testbench | Transducer Gain Testbench | **RF Budget Analyzer** | "Using RF Measurement Testbench for IQ-to-RF Converter" | "Use RF Measurement Testbench for RF-to-IQ Converter"

Simulate High Frequency Components

This example shows how to use RF Blockset™ circuit envelope simulation to simulate high frequency components while reducing simulation time.

Simulate Passband Signal in Simulink Software

The model, `ex_simrf_tut_passband_model`, shows how to modulate a real passband signal with in-phase and quadrature components.
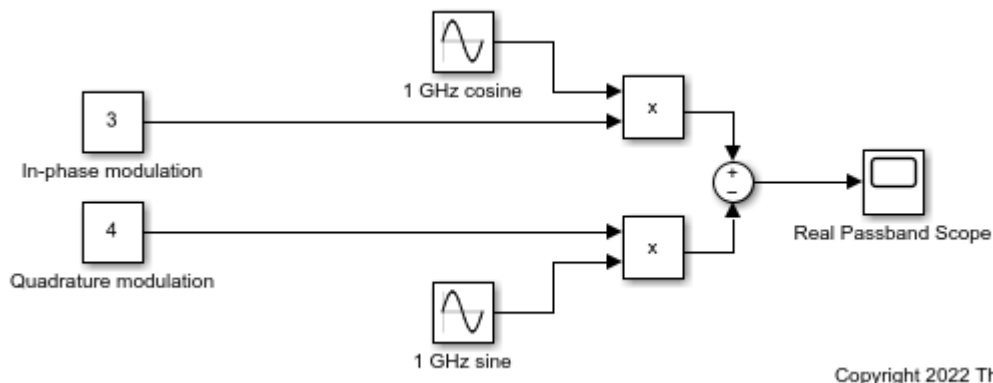
The system specifies a real passband signal x(t) according to the formula

$$x(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t)$$

where:

- I(t) is the in-phase part of the modulating signal and it is set to 3. This is modeled by the Constant block labeled In-phase modulation.
- Q(t) is the quadrature part of the modulating and it is set to 4. This is modeled by the Constant block labeled Quadrature modulation.
- fc is the carrier frequency and it is set to 1 GHz.

Running the model produces the following output on the scope.
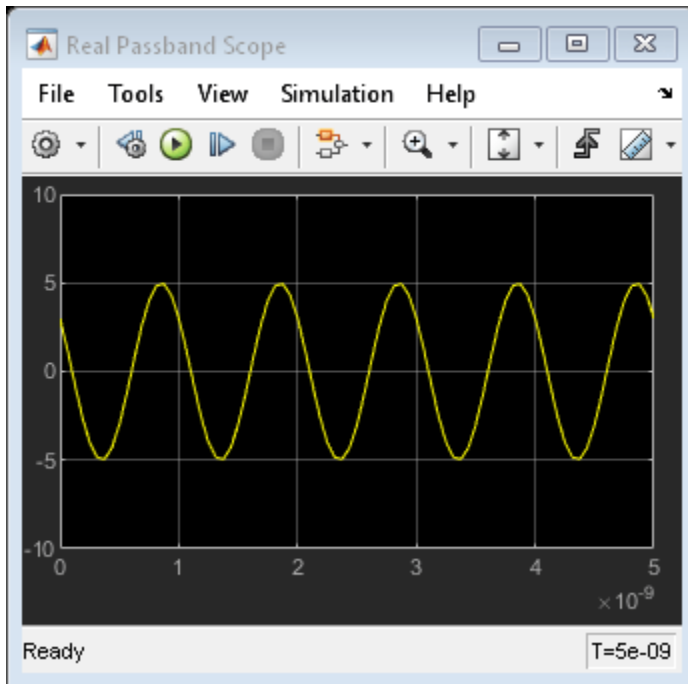
The output signal at the Real Passband Scope has a magnitude of 5 and a phase shift of atan2d(3,-4) or about 143 degrees.

In the Configuration block, the **Fixed-step size (fundamental sample time)** parameter has been set to `1/16*1e-9`. This value is on the order of the wavelength of the carrier. The simulation takes a total of `81` samples – 16 per cycle.

Compare Passband and Baseband Signals in RF Blockset Software

The model, `ex_simrf_tut_compare_model`, shows how to compare passband and baseband signals. This section builds on the results of the previous section, Simulate a Passband Signal in Simulink Software.
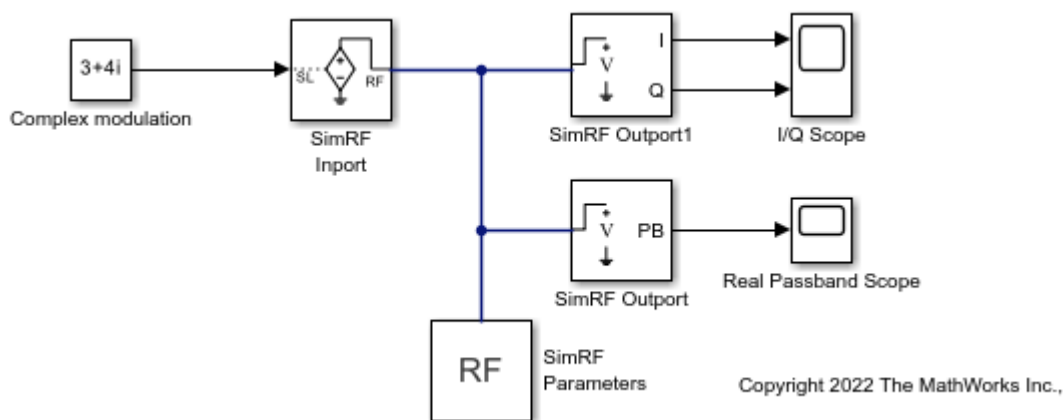
The system simulates a real passband signal as the real part of a complex passband signal according to the formula

$$x(t) = \mathrm{Re}[I(t) + jQ(t)e^{j2\pi f_c t}] = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t)$$

where:

- I(t) + jQ(t) is the complex-valued modulation, equal to 3 + 4j.
- fc is the angular frequency, equal to 1 GHz.

Contrary to the Simulink® passband implementation in the previous section, the complex baseband signal driving the RF Blockset system does not include the carrier. Instead, the RF Blockset environment handles the carrier analytically. The carrier appears in four different blocks in the RF Blockset environment:

- In the Inport block, the Carrier frequencies parameter defines the carrier frequencies of the modulations entering from outside the RF Blockset environment. In this example, there is only one input signal, and only one carrier (1 GHz, specified as 1e9 Hz).
- In the Outport block, the Carrier frequencies parameter specifies the signal on the 1e9 Hz carrier (1 GHz) as Simulink signals. These signals appear at the I and Q ports. The Output parameter is set to Real Passband, so this signal represents a real passband signal on the 1-GHz carrier.
- In the block labeled RF Blockset Outport1 block, also an Outport block, the Carrier frequencies parameter specifies the signal outputted on the 1e9 Hz carrier (1 GHz) as Simulink signals. These signals appear at the I and Q ports. The Output parameter is set to In-phase and Quadrature Baseband, so these signals represent the in-phase and quadrature modulations of the signal on the 1-GHz carrier.
- In the Configuration block, the Carrier frequencies parameter specifies all of the carriers to be modeled in the RF Blockset circuit envelope simulation environment. In this example, only one carrier is specified. For more options, refer to Configuration block.

Running the model produces the following output on the scopes.
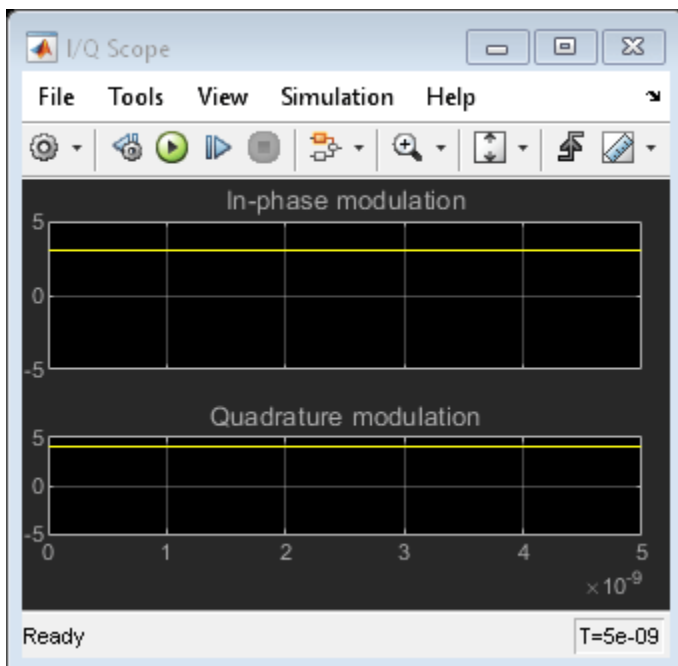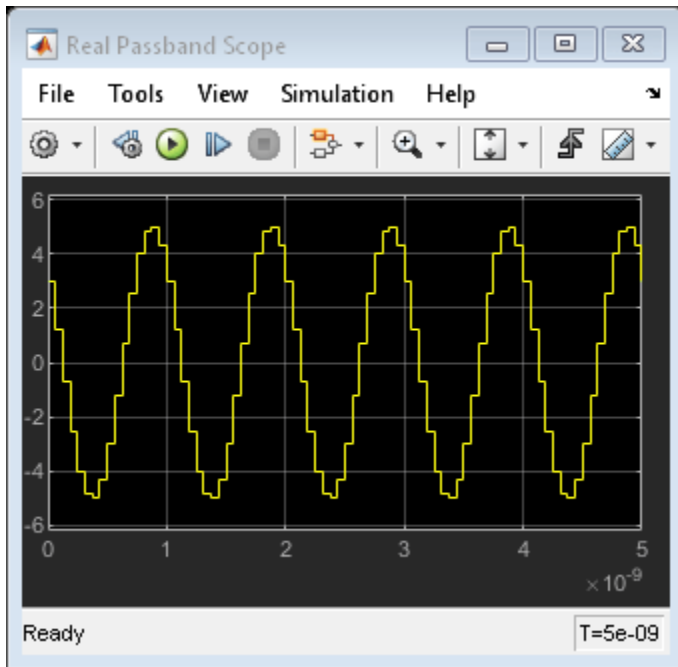
The Real Passband Scope displays the same output as the example in the previous section, Simulate a Passband Signal in Simulink Software. The signal has a magnitude of 5 and a phase shift consistent with the specified in-phase and quadrature amplitudes.

The 1-GHz carrier itself does not appear in the output. The results correspond to the real and imaginary parts of the Complex modulation at the input of the system. They also correspond to the In-phase modulation and Quadrature modulation blocks in Simulate a Passband Signal in Simulink Software.

In the Configuration Parameters dialog box, the **Fixed-step size (fundamental sample time)** parameter has been set to `1/16*1e-9`. This value is on the order of the wavelength of the carrier. The simulation takes a total of `81` samples – 16 per cycle.

Simulate Envelope of Baseband Signal

The model, `ex_simrf_tut_envelope_model`, shows how to simulate the envelope of a sine wave using RF Blockset blocks. This section builds on the results of the previous section, Compare Passband and Baseband Signals in RF Blockset Software.
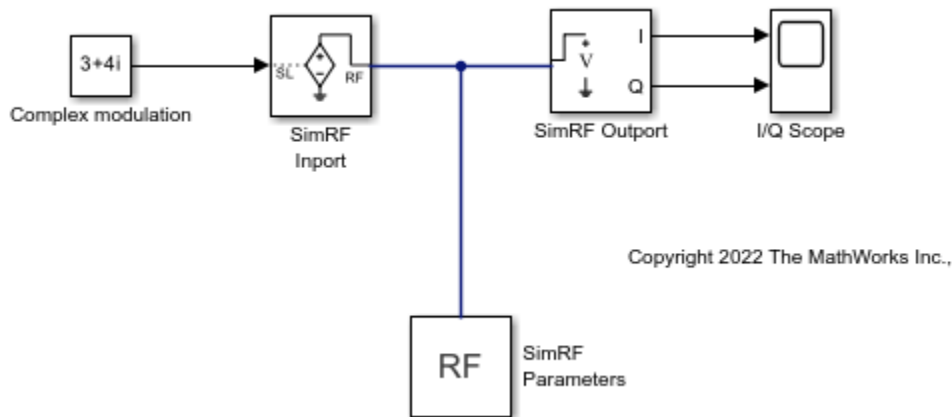
The system is almost identical to the system in the previous section, except:

*   The model contains only one RF Blockset Outport block and only one scope. The RF Blockset environment outputs in-phase and quadrature modulations of the 1-GHz signal. In the RF Blockset Outport block, the Output parameter is set to In-phase and quadrature baseband. Since the system is not configured to output a real passband signal, the carrier is not simulated.

*   In the Configuration Parameters dialog box, the **Fixed-step size (fundamental sample time)** parameter is greater. Its value is `5e-9` instead of `1/16*1e-9`.

Running the model produces the following output at the scope.
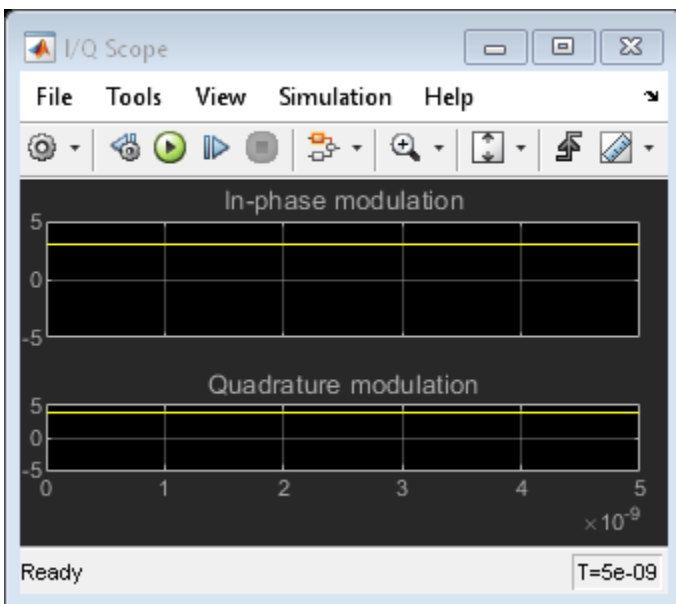
The I/Q Scope displays the in-phase and quadrature baseband components of the 1-GHz signal. The 1-GHz carrier itself does not appear in the output. The results correspond to the real and imaginary parts of the Complex modulation at the input of the system.

In contrast to the models in the previous two sections, Simulink works differently in this model. Because the modulating signals are constant in this example, only two sample points are needed. To

simulate a time-varying modulating signal, Simulink can use a fixed time step on the order of the reciprocal of its bandwidth.

The model uses a value of `5e-9` for the **Fixed-step size (fundamental sample time)** parameter. This value equals the **Stop time** because, in this case, the modulating signals are constant. Compared to the preceding examples, which use a sample time of `1/16*1e-9`, this model simulates accurately with a time step 80 times larger. This step size results in a reduction of total sample time by a factor of 80, excluding the initial time step at time `0`.

See Also
Inport | Outport | Configuration

More About
- "Model RF Mixer" on page 2-2

Using RF Blockset for the First Time

In this section...
"Expected Background" on page 1-24
"Circuit Envelope, Equivalent Baseband, and Idealized Baseband Features" on page 1-24
"Understanding RF Blockset Environment" on page 1-26

Expected Background

Topics in the RF Blockset documentation assume that you are already familiar with:

- Using MATLAB to write and execute scripts and functions.
- Using Simulink to create and simulate block diagrams.

Circuit Envelope, Equivalent Baseband, and Idealized Baseband Features

RF Blockset software offers Circuit Envelope, Equivalent Baseband, and Idealized Baseband libraries to model RF networks. Each library represents a distinct simulation paradigm. For certain applications, one library can offer an advantage over another.

- Use the "Circuit Envelope Library" on page 1-3 for multicarrier simulation of RF networks with arbitrary topologies.
- Use the "Equivalent Baseband Library" on page 3-19 for single-carrier simulation of cascaded RF networks.
- Use the "Idealized Baseband Library" on page 5-2 for discrete-time simulation of RF networks without impedance mismatches.

You can also use this decision matrix to decide which library to use to design your RF systems. The Circuit envelope library enables high-fidelity, multicarrier simulation of any network topology with impedance mismatches. The Equivalent Baseband library enables fast, discrete-time simulation of single-carrier single-chain systems with impedance mismatches. And the Idealized Baseband library enables fast, single-carrier simulation of perfectly matched networks with any network topologies.
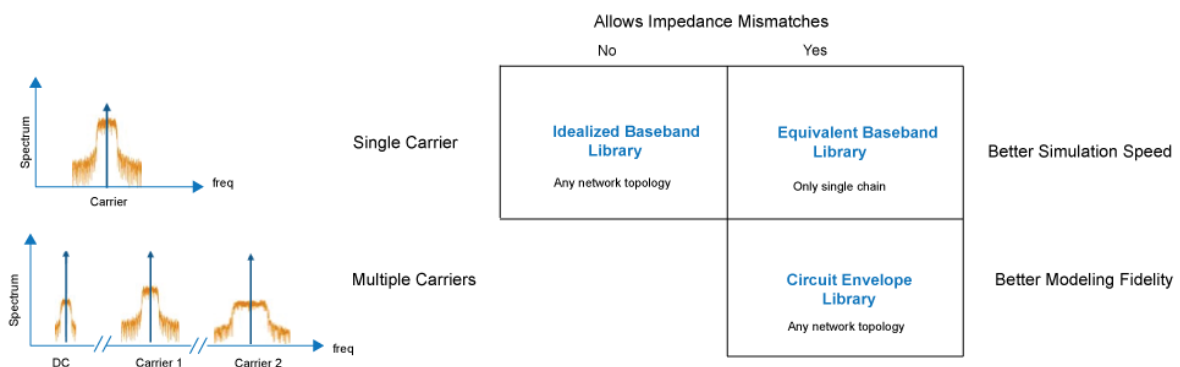
All blocks from the Circuit Envelope and Equivalent Baseband libraries except Inport, Outport, Input Port, and Output Port blocks do not connect to each other. Therefore, to avoid redesigning your model later, choose which library to use based on your application. Consult this table for a summary of the features of the three libraries.

Summary of Features

Feature	Circuit Envelope Library	Equivalent Baseband Library	Idealized Baseband Library
Connect directly to Simulink blocks	No, except the cross-domain blocks	No, except the cross-domain blocks	Yes
Support single-carrier simulation	Yes	Yes	Yes
Support multicarrier simulation	Yes	No	No
Support nonlinear elements	Yes	Yes	Yes
Support simulation of cascaded networks	Yes	Yes	Yes
Support simulation of networks with arbitrary topologies	Yes	No	Yes
Support signal probing between input and output	Yes	No	Yes
Support noise simulation	Yes	Yes	Yes
Support custom authoring	Yes	No	No
Support specification using network parameter data	Yes	Yes	Yes
Model input/output impedance mismatches	Yes	Yes	No
Generate C code with Simulink Coder™	No	Yes	Yes
Support Rapid Accelerator Simulation Mode	Yes	Yes	Yes
Support Model Reference Accelerator Simulation Mode	Yes	Yes	Yes
Support Model Protection	Yes	Yes	Yes

Note Rapid Accelerator is not supported to a subset of circuit envelope blocks. For more information, see Limitation.

Understanding RF Blockset Environment

The RF Blockset environment comprises of interconnected blocks and algorithms that model an RF system. In the RF Blockset environment, all blocks fall into one of the three categories.

Blocks Operating Within RF Blockset Environment

These blocks contribute to the physical representation of an RF system. Most RF Blockset Circuit Envelope, Equivalent Baseband and Idealized Baseband library blocks fall into this category, including all blocks in the Elements and Sources sublibraries. For example, a Resistor block can model a source impedance or part of a matching network, and an Amplifier block can model a physical RF amplifier. Both these blocks model physical components.

Blocks Operating Between RF Blockset and Simulink Environments

These blocks, also called cross-domain blocks, provide an interface from an RF system to a larger design. The Inport, Outport, Input Port, and Output Port blocks fall into this category. For example, you can construct a signal using blocks from Communications Toolbox or DSP System Toolbox libraries, and you can input that signal into the RF Blockset environment using a RF Blockset Inport block. You can also input that signal into an RF chain using an RF Blockset Input Port block.

Environment Independent Block

The Configuration block manipulates the environment and determines the simulation configuration. To use this block, connect it to any part of the RF system. Because this block is not a part of the physical representation of the system, it has the same effect regardless of where you connect it.

To run models containing other RF Blockset blocks, you must connect a Configuration block to the RF Blockset environment.

See Also

More About
- "Getting Started with RF Modeling"
- "Circuit Envelope Basics" on page 1-6

Model RF Filter Using Circuit Envelope

This example shows how to model an RF filter using Circuit Envelope Library. In this example, you compare the input and output signal amplitudes to study the signal attenuation.

Model Overview

This example uses an LC bandpass filter designed to have a bandwidth of 200 MHz. The filter uses a three tone input signal to demonstrate the filter attenuation property for in-band and out-band frequencies. The input signal tones are:

- 700 MHz – Center frequency of the filter passband
- 600 MHz – Lower edge frequency of the filter passband
- 900 MHz – Frequency outside the filter passband

Define Model Variables and Settings

Define model variables for blocks that share parameter values using the `InitFcn`.

- In Simulink® editor, select **MODELING** > **SETUP** > **Model Settings** > `Model Properties`.
- In the `Model Properties` dialog box, on the **Callbacks** tab in **Model callbacks** pane, select `InitFcn`.
- In the **Model initialization function** pane, enter: `amp = ones(1,3); freq = [600 700 900]*1e6; stepsize = 1/500e6;`
- Click **OK**.
- In the Simulink tool bar, set the simulation **stop time** to `0`.
- In Simulink editor, select **MODELING** > **SETUP** > **Model Settings** > `Model Settings Ctrl +E`
- In **Configuration Parameters** dialog box, set **Solver** to `discrete (no continuous states)` and in **Data Import/Export** pane, clear `Single simulation output`.

Required Blocks

The filter system consists of LC Ladder, Inport, Outport and Configuration blocks. The physical part of the model uses bidirectional RF signals. The blocks used in the system are:

Block	Library Path	Use	Quantity
Constant	Simulink > Sources	To generate real and complex constant value	1
Inport	RF Blockset > Circuit Envelope > Utilities	To convert Simulink input signal to RF Blockset™ input signal	1
Configuration	RF Blockset > Circuit Envelope > Utilities	To set the system wide parameters for RF Blockset simulation	1
LC Ladder	RF Blockset > Circuit Envelope > Elements	To models the signal attenuation	1
Outport	RF Blockset > Circuit Envelope > Utilities	To convert RF Blockset signals to Simulink signals	1
db Conversion	DSP System Toolbox > Math Functions > Math Operations	To convert magnitude data to decibels	2
Math Function	Simulink > Math Operations	To perform mathematical functions	2
To workspace	Simulink > Sinks	To write data to a MATLAB® workspace for plotting	2
Terminator	Simulink > Commonly Used Blocks > Terminator	To terminate the angle baseband output of the Outport block	1

Connect the blocks as shown in the model or open the model attached in this example.

```
open_system("rffilter_ce.slx")
```


Copyright 2022 The MathWorks, Inc.
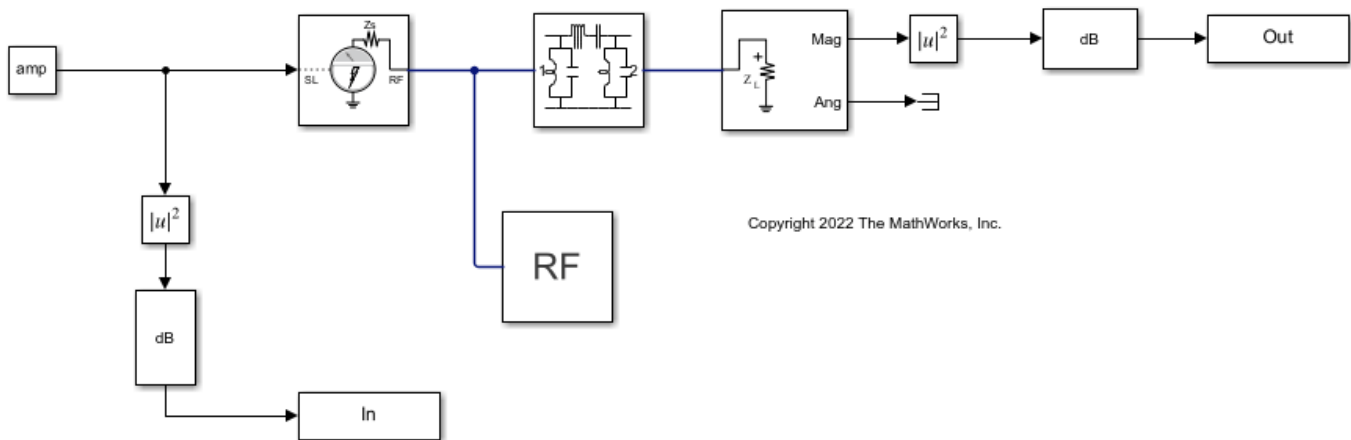
Configure Input Signal

Generate the three-tone input signal using these blocks:

- Constant block specifies the amplitude of the signal.
- Inport block configures the frequencies of the three tones.
- Configuration block specifies the step size.

In the Constant block:

- Set the **Constant** value to amp, as defined in the InitFcn.

In the Inport block:

- Set **Source type** to Power.

- Set **Carrier frequencies** to `freq` , as defined in `InitFcn`. The `freq` variable sets the frequency of the three tones to 600 MHz, 700 MHz, and 900 MHz, respectively.
- Click **OK**.

In the Configuration block:

- Set **Step size** to `stepsize`, as defined in `InitFcn`.
- Clear **Simulate noise**.
- Click **OK**.

The fundamental tones and harmonics are updated automatically when you run the model.

Configure RF Filter

In the LC Ladder block:

- Set **Ladder topology** to `LC Bandpass Pi`.
- Click **Apply** and then click **OK**.

Configure Output Settings

In the Outport block :

- Set **Sensor type** to `Power`.
- Set **Output** to `Magnitude and Angle Baseband`.
- Set **Carrier frequencies** to `freq`, as defined in `InitFcn`.
- Click **OK**.

Configure Other Blocks

- In the Simulink editor, connect the `Ang` port of the Outport block to Terminator block to terminate the angle baseband output.
- In Math Function and Math Function 1 blocks, set **Function** to `magnitude^2` and click **OK**. The block squares the magnitude of the input and output signal.
- In dB Conversion and dB Conversion 1 blocks, set **Input Signal** to `Power` and click **OK**. The block converts the input and output signals to dB.
- In To Workspace, change the **Variable name** to In. In To Workspace 1, change the **Variable name** to `Out`. In both blocks, set **Save format** to **Array** and click **OK**.

Run Model

Select **Simulation > Run** to run the model.

```
sim("rffilter_ce.slx")
```

Plot and Analyze Attenuated Output Signals

Display the input and output signals using `semilogx` function, in dB.

- Transfer the input and output dB values to the MATLAB® workspace using To Workspace block.
- To view the input signal to RF Filter, plot `In` array from the MATLAB workspace.

```
figure
h = semilogx(freq, In,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
```

```
xlim([5.5e8,9.5e8]);
xlabel('Frequency[Hz]');
ylabel('Amplitude[dB]');
title('Input Signal');
```
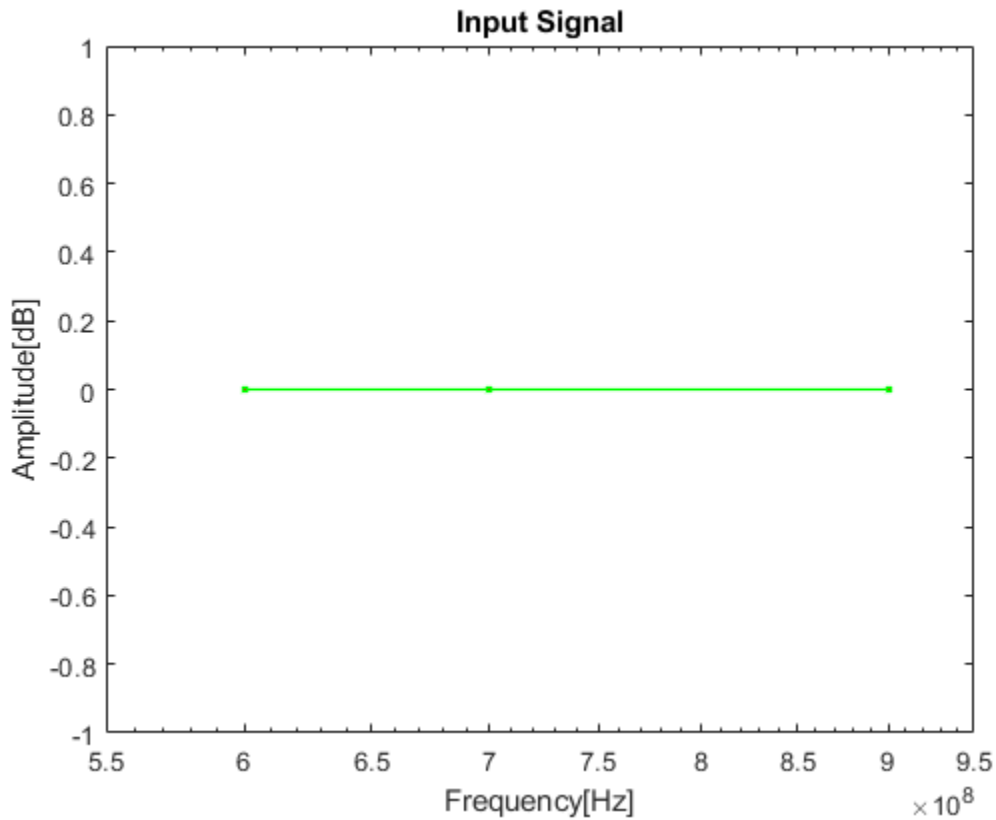
Input Signal

To view the output signal (filtered attenuated signal), plot `Out` array from the MATLAB workspace.

```
figure
h1 = semilogx(freq, Out,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
xlim([5.5e8,9.5e8]);
ylim([-25,1]);
xlabel('Frequency[Hz]');
ylabel('Amplitude[dB]');
title('Attenuated Output Signal');
```

Compare the input and output signal plots to verify the attenuation caused by the filter. Notice that the RF filter does not attenuate the signal at the center frequency of 700 MHz.

Analyze LC Bandpass Filter Response

Plot more points to better understand the response of the LC bandpass filter. Change the defined variables in Model Properties to: `amp = ones(1,201); freq = logspace (8,10,201); stepsize = 1/500e6;`.

Run the model. Notice that the signal is not attenuated within the 200 MHz range of the LC bandpass filter. Alternatively, you can run `rffilter_ce3.slx`. Note that `rffilter_ce3.slx` is loaded with the updated defined variables.

```
sim("rffilter_ce3.slx")
```

Plot the LC bandpass filter frequency response.

```
figure
h3 = semilogx(freq, Out,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
xlim([5.5e8,9.5e8]);
ylim([-25,1]);
xlabel('Frequency[Hz]');
ylabel('Amplitude[dB]');
title('LC Bandpass Filter Frequency Response');
```
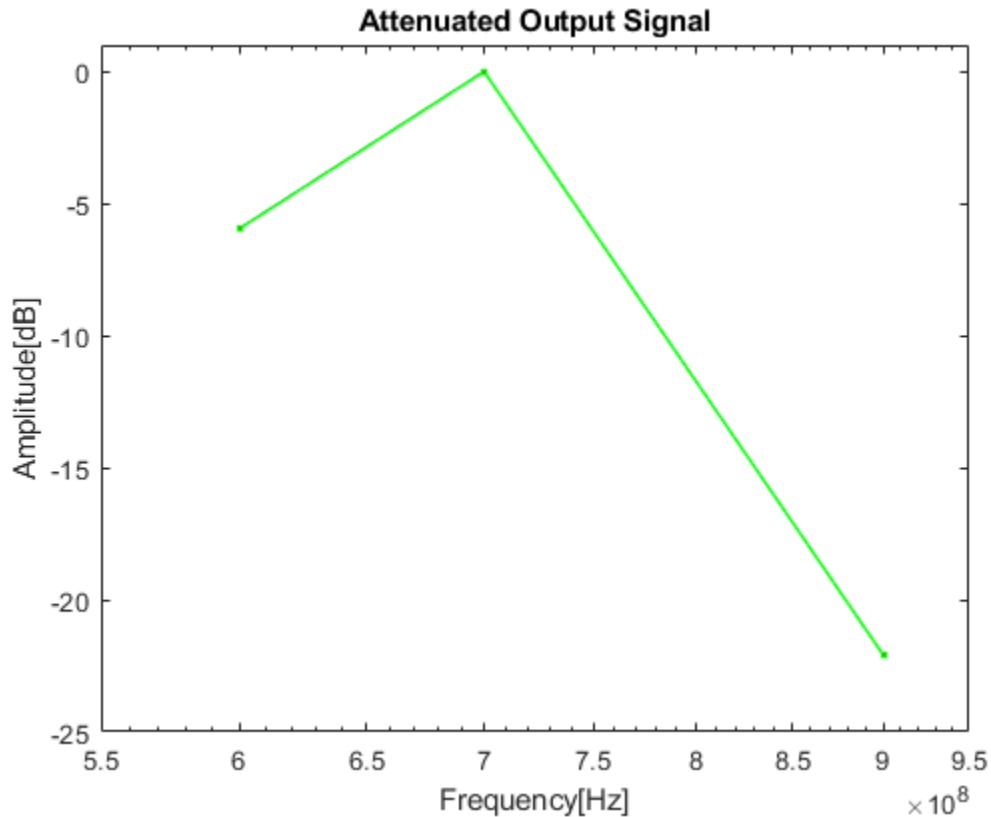
LC Bandpass Filter Frequency Response
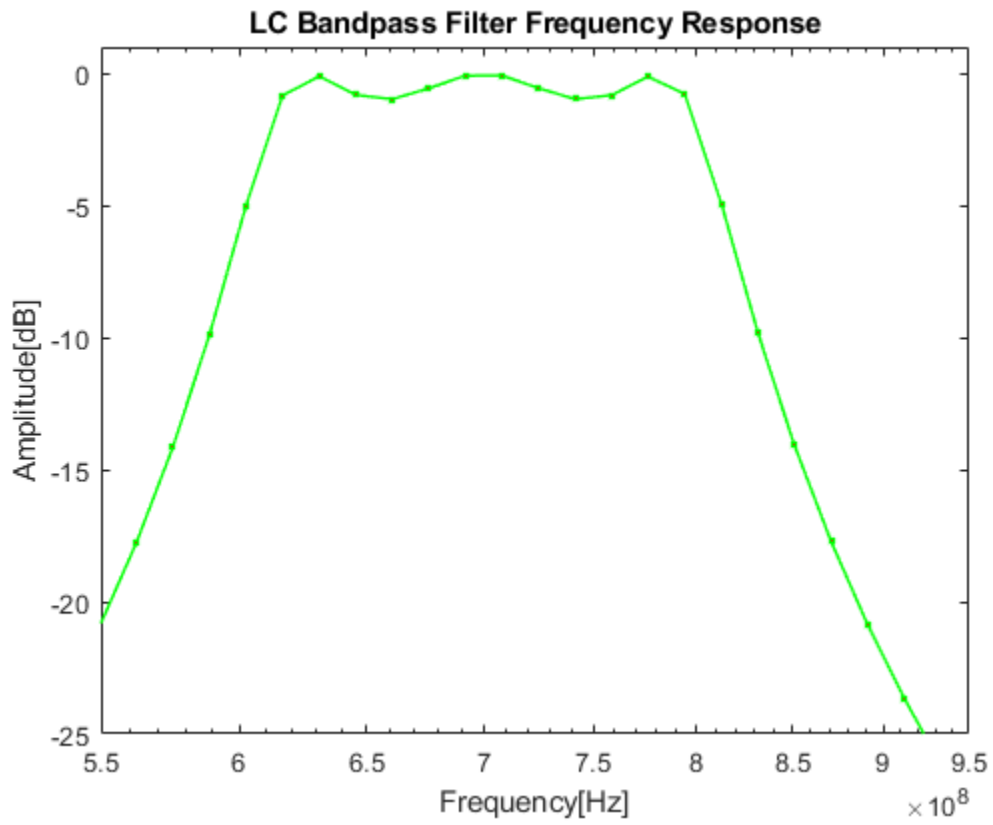
See Also

Filter

More About

- "Model RF Filter Using Equivalent Baseband" on page 3-7

Frequency Conversion

Model RF Mixer

This example shows how to

- Use a Sinusoid block to model the envelope of an amplitude-modulated (AM) waveform.
- Use a Continuous Wave block to model the constant envelope of an ideal local oscillator (LO).
- Use a Mixer block to downconvert the AM waveform to an intermediate frequency (IF).
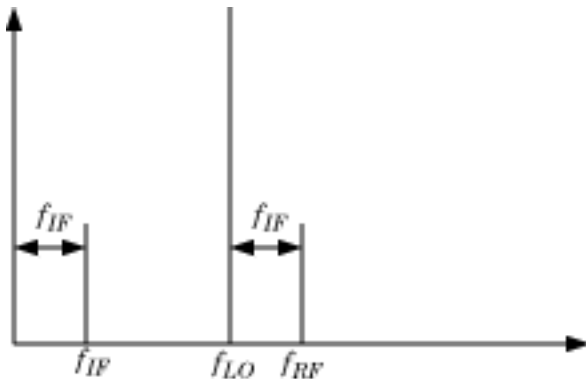
The preceding figure illustrates frequency conversion. A signal modulated on the carrier fRF mixes with a local oscillator at fLO. The operation downconverts fRF to fIF = fRF – fLO. The upper mixing product fRF + fLO is not modeled.

Build RF System

This model specifies an AM waveform at the input port of the mixer, an LO at the LO port of the mixer, and a 50 ohms termination at the output of a mixer.

To build this model, open the RF Blockset library by typing `simrfV2libs` in the MATLAB® Command Window.

Double-click the block labeled **Circuit Envelope** to open the Circuit Envelope library.

From the **Elements**, **Sources**, and **Utilities** sublibraries, add the following blocks to your model.

- From the **Elements** library, add a Resistor, a Mixer, and a Ground blocks.
- From the **Sources** library, add a Continuous Wave and a Sinusoid block.
- From the **Utilities** library, add a Configuration block.

Connect the blocks in the same configuration as shown in this image.
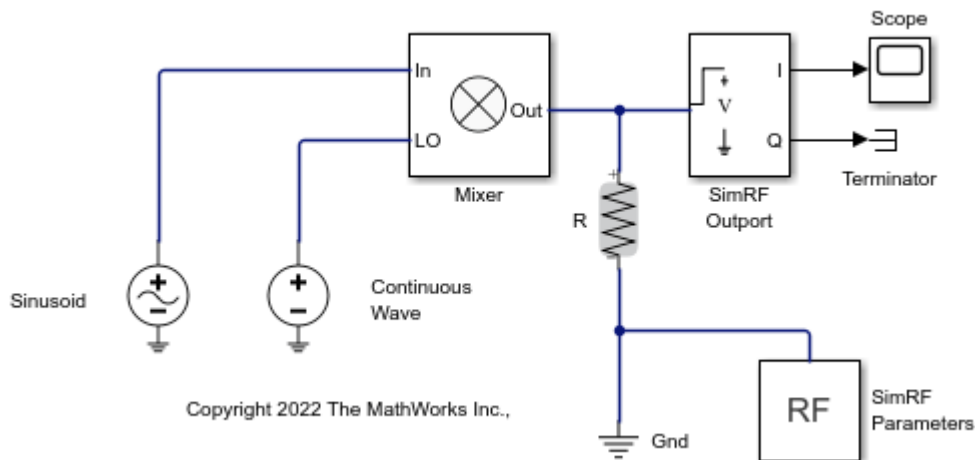
Define Model Variables

For models with blocks that share parameter values, specifying parameters values using variables saves time and effort.

Simulink® models run MATLAB code stored in the initialization function (**InitFcn**) each time the model starts. The MATLAB code runs in the base workspace. If the initialization function stores variables in the MATLAB workspace, the variables are overwritten every time the model executes the initialization function.

1   In Simulink editor, click **Modeling**. In the toolstrip, in **SETUP> Model Settings**, click **Model Properties** .
2   In the **Callbacks** tab, within the **Model callbacks** pane, select the **InitFcn** node.
3   In the **Model initialization function** pane, enter the following MATLAB code:

```
modulationAmplitude = 1; modulationFrequency = 5e5; LOAmplitude = 1;
LOFrequency = .95e9; RFCarrier = 1e9;
```

Specify Block Parameters for RF Simulation

In this section you will learn how to:

*   Configure the RF Blockset(M) environment for multi-frequency circuit envelope simulation by specifying the Simulation frequencies parameter in the Configuration block dialog box
*   Specify the attributes of the waveforms *Configure global simulation settings

Using block dialog boxes to specify the parameters for simulation:

In Simulink editor, click **Simulation** in the **PREPARE** , click **Model Settings** in **Configuration and Simulation**. Specify the following parameters:

*   Set **Stop time** to 1e-5. You can also set the stop time directly on the Simulink canvas.
*   Set **Solver** to ode23t (mod. stiff/Trapezoidal). The RF Blockset environment does not use the ode23t solver. However, since oscillating signals can be stiff, the solver is a good choice for the Simulink environment when using RF Blockset blocks.

Double-click the Sinusoid block to open the Sinusoid Block Parameters dialog box. Specify the following parameters:

- Set **Sinusoidal amplitude in-phase** to `modulationAmplitude`.
- Set **Sinusoidal modulation frequency** to `modulationFrequency`.
- Set **Carrier frequencies** to `RFCarrier`.

Double-click the Continuous Wave block to open the block dialog box. Specify the following parameters:

- Set **Constant in-phase value** to `LOAmplitude`.
- Set **Carrier frequencies** to `LOFrequency`.

Double-click the Configuration block to open the block dialog box. Set **Step size** to `1e-7`.

Probe Circuit Envelopes Waveforms

At the output of the mixer:

From the RF Blockset **Utilities** library, drag and drop an Outport block onto your model. In the block dialog:

- Set **Output** to `In-phase and quadrature`.
- Set **Carrier frequencies** to `RFCarrier − LOFrequency`.

From the Simulink **Commonly Used Blocks** library, drag and drop a Scope block and a Terminator block onto your model.

This configuration uses a RF Blockset Outport block as a voltage sensor at the output port of the mixer. The Simulink signal at the outport is the envelope of the carrier or carriers specified in the block. A scope attached to the outport plots the envelope. The **Output** parameter controls how signals are presented at the output ports. To change the appearance of the block, follow one of the workflows in the Specify Block Parameters for RF Simulation section of this tutorial.

Observe Downconverted Envelope Signal at Output Port

Click **Run** to run the model. To view the results of the simulation, double-click the scope, and click the **Autoscale** button.

The Sinusoid specifies a 1-V amplitude for the modulation on fRF, which the mixer downconverts to fIF. The RF Blockset Outport block probes the intermediate frequency and recovers the 1-V modulation amplitude. This value agrees with a specified conversion gain of 0 dB in the mixer.
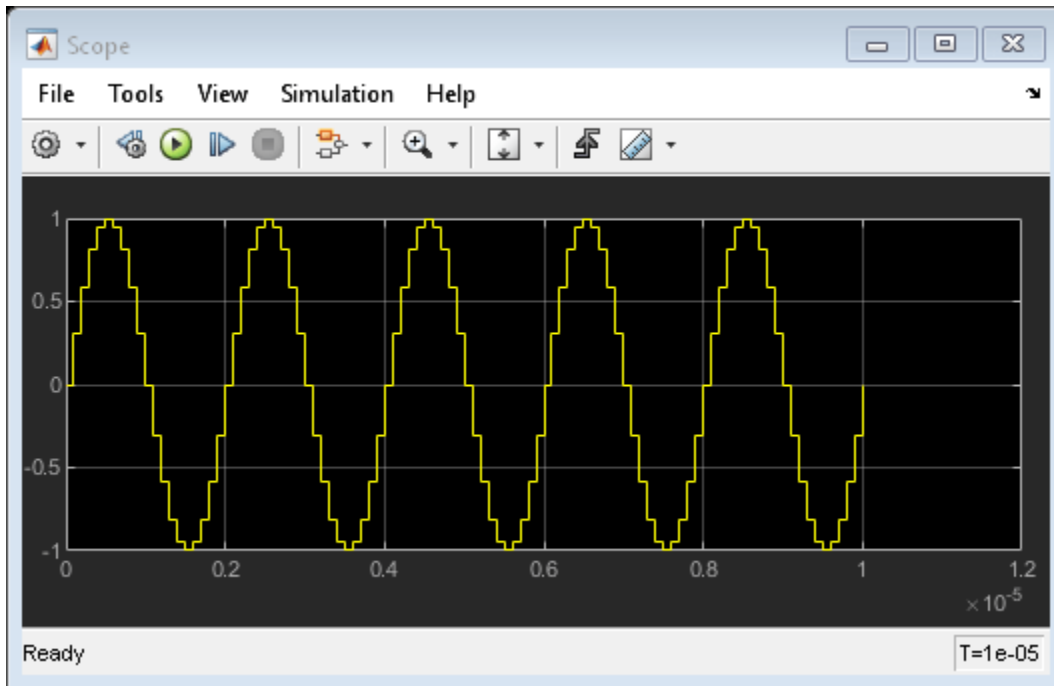
See Also
Mixer

More About
*   "Simulate High Frequency Components" on page 1-18

Filter Mixing Products

This example shows you how to:

- Configure a RF Blockset Outport block to probe multiple carrier frequencies simultaneously.
- Model an analog filter in the RF Blockset™ environment using RF Blockset Capacitor and Inductor blocks.
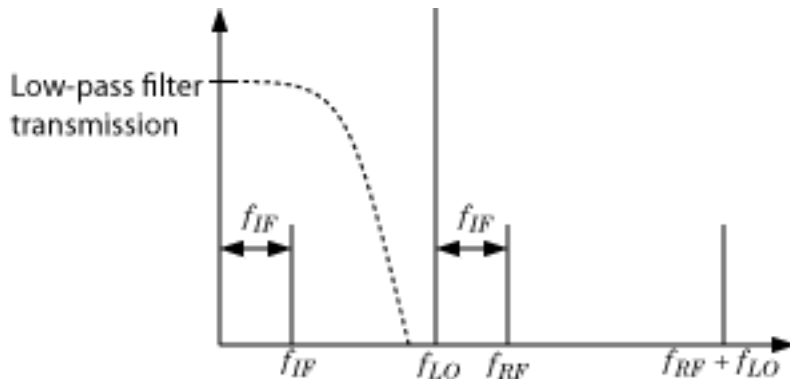
The preceding figure illustrates low-pass filtering of a low-side injection system. Mixing fRF and fLO produces signals on the carriers fIF = fRF – fLO and fRF + fLO. Adding a low-pass filter to the model reduces the power present in the high-frequency signal.

Probe Multiple RF Carriers

To begin, open the model that you created in the Model an RF Mixer example.
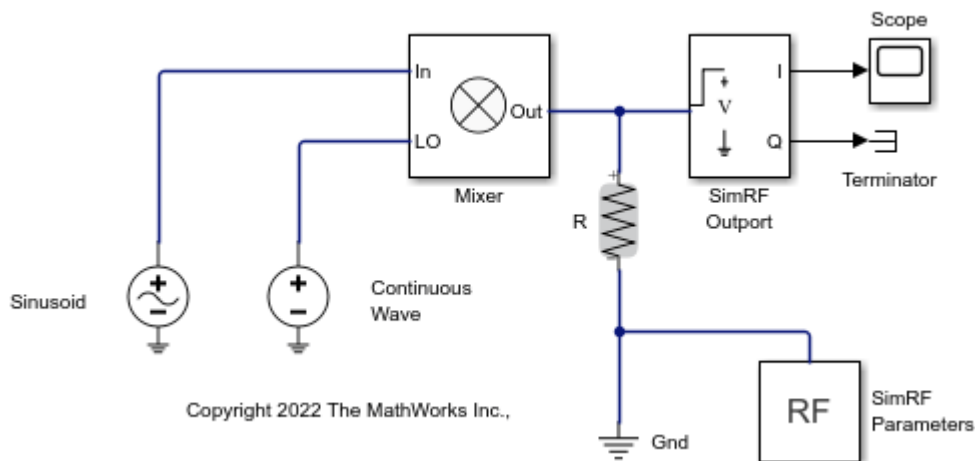
The RF Blockset environment specifies the carriers fRF – fLO, fRF, fLO, and fRF + fLO, but the RF Blockset Outport block probes only the carrier fRF – fLO. Examine the carriers fRF – fLO and fRF + fLO by changing the model according to the following workflow:

- From Simulink® Commonly Used Blocks library, add a Demux block to your model.

- In the Scope block dialog box, click the Parameters button, then set **Number of Axes** to 2. **In the Outport block dialog box, set *Carrier frequencies** to [RFCarrier − LOFrequency, RFCarrier + LOFrequency].
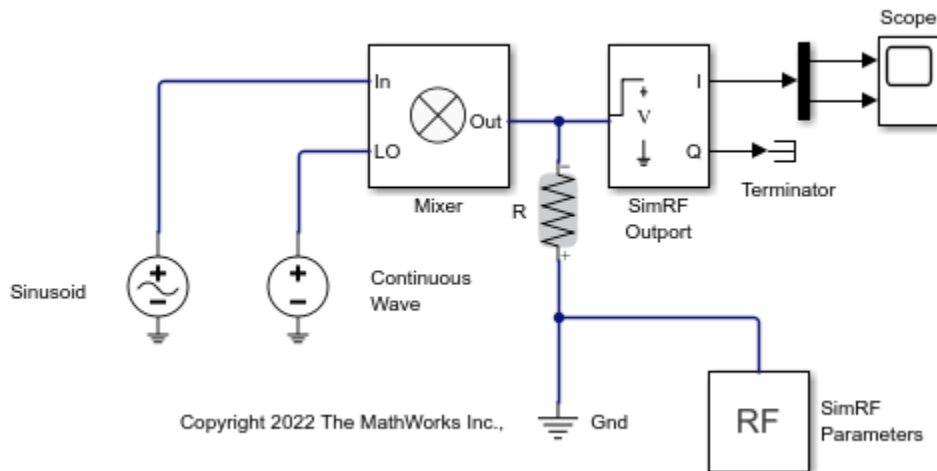
Run to run the model and to view the results of the simulation, double-click the scope.
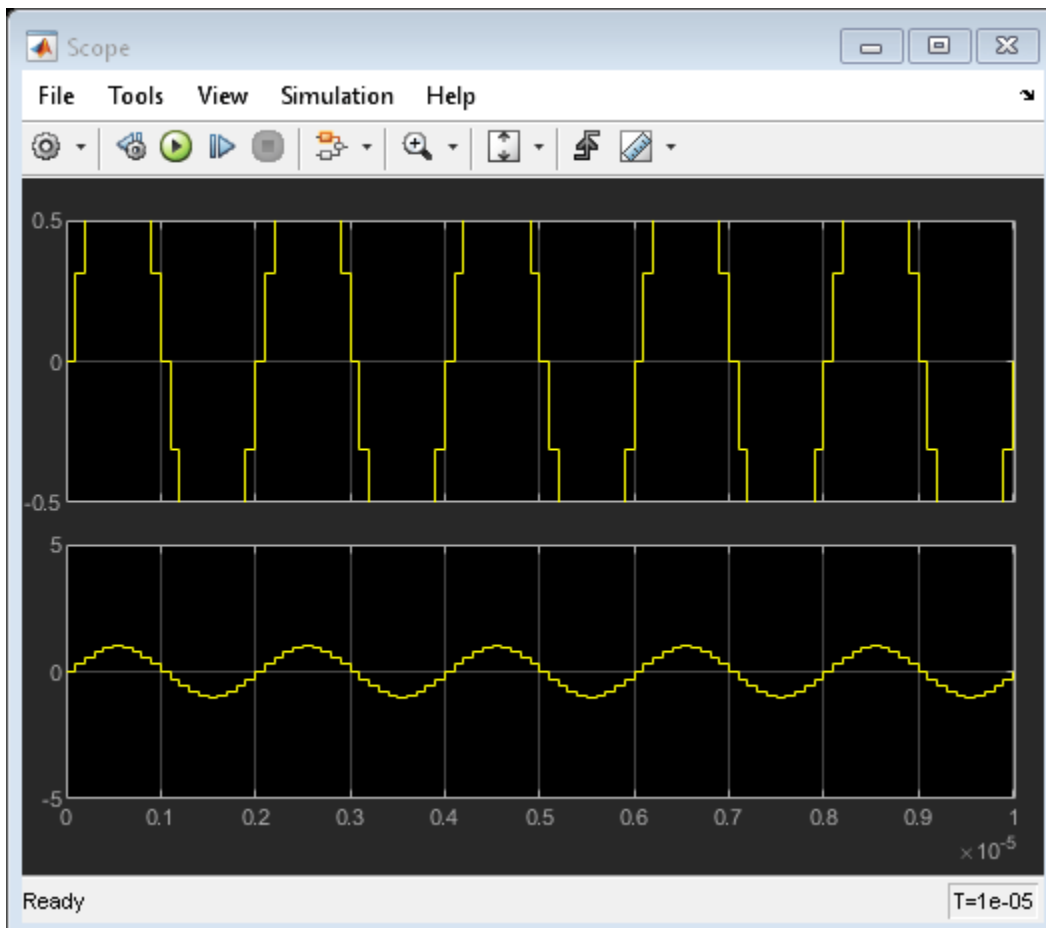
The first set of axes displays the modulation of specified carrier fIF = fRF – fLO. This carrier appears on the first set of axes because the Carrier frequencies parameter of the outport specifies it first. The second set of axes displays the modulation at fRF + fLO. The modulation of the upper mixing product has the same amplitude as the modulation of the downconverted signal.

Model RF Filter

To begin, open the model that you created in the Probe Multiple RF Carriers section.

- From the Elements library, drag and drop one Capacitor and two Inductor blocks onto your model
- Set the **Capacitance** parameter of the Capacitor block to `40e-12` F.
- Set the **Inductance** parameter of both Inductor blocks to `50e-9` H.
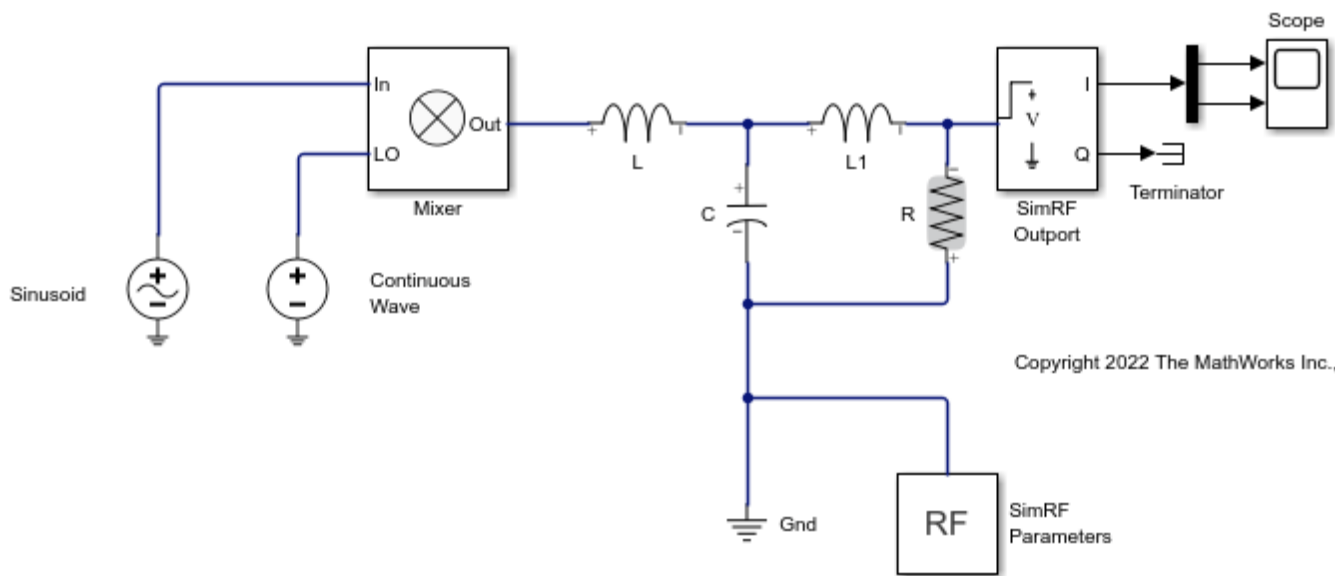
This configuration models a third-order low-pass Butterworth filter with a cutoff frequency of 1 rad/ns, or about 0.159 GHz. The high-frequency mixing product is in the stopband of the filter. The low-frequency product is in the passband. You can also use an LC Ladder block to model this filter.

Select **Run** to simulate Filtering of RF Signals.
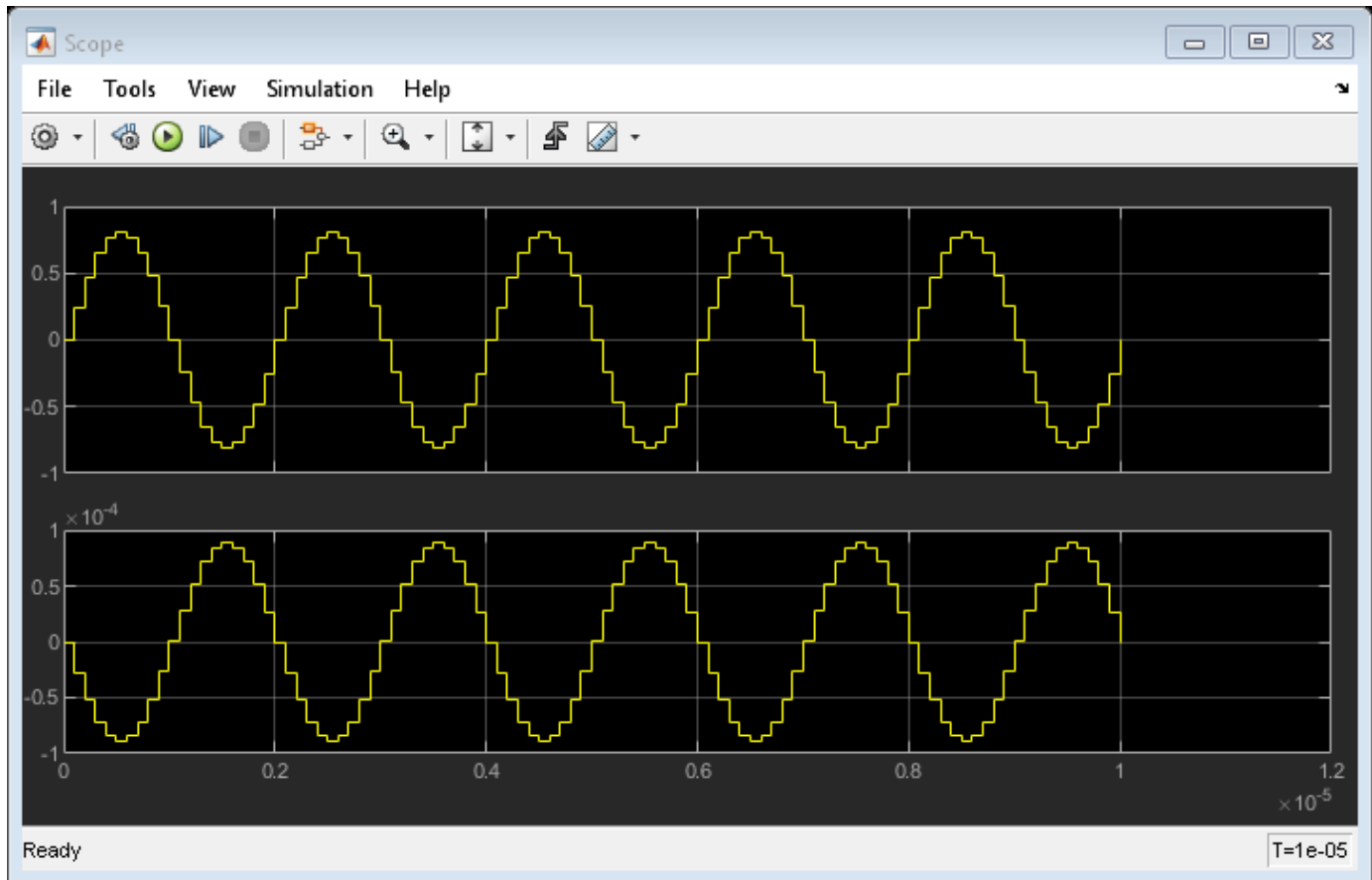
The filter attenuates the high-frequency signal at fRF + fLO and transmits the signal at fRF – fLO with minimal loss.

Improve Performance by Reducing Total Simulation Frequencies

The models in the preceding examples use the **Automatically select fundamental tones and harmonic order** setting in the Configuration block. This setting sacrifices performance for compatibility by guaranteeing that every specified **Carrier Frequency** parameter appears in the set of RF Blockset simulation frequencies. The performance of the `ex_simrf_tut_filter_model` can be improved by reducing the size of the set of simulation frequencies, as measured by the **Total simulation frequencies** value displayed in the Configuration block dialog.

To enhance performance of the `ex_simrf_tut_filter_model`, follow the procedure below.

1   Open and simulate the `ex_simrf_tut_filter_model`.

2   Double-click the Configuration block to open the block dialog box. Note that the block dialog displays a **Total simulation frequencies** value of 121. This value indicates that the environment is running 121 separate simulations.

3   Clear the **Automatically select fundamental tones and harmonic order** check box.

4   Set **Fundamental tones** to [RFCarrier, LOFrequency]. This step is not strictly necessary, but setting these values clarifies the meaning of the parameters. Because the mixer has fRF and fLO at its input ports, all signals at the output of the mixer have carrier frequencies that are linear combinations of these fundamental tones.

5 Set **Harmonic order** to 1. The frequencies fRF + fLO and fRF – fLO are the only output carriers of interest. Modeling signals at a higher harmonic order than one is not necessary for this system.

6 Click **Apply** . Note that the block dialog displays a Total simulation frequencies value of 9.

7 Simulate the model.

The result of the simulation has not changed because every frequency of importance appears in the new set of simulation frequencies. However, this procedure reduces overall compatibility. If you make modifications to the model, such as adding nonlinear amplification, the resulting signals of interest may not appear in the set of simulation frequencies. You can restore compatibility by restoring the Automatically select fundamental tones and harmonic order check box to its default.

Equivalent Baseband

Introduction to Equivalent Baseband Simulation

RF Blockset Equivalent Baseband Libraries

In this section...
"Overview of RF Blockset Equivalent Baseband Libraries" on page 3-2
"Open RF Blockset Equivalent Baseband Libraries" on page 3-2
"Equivalent Baseband Library" on page 3-3
"Idealized Baseband Library" on page 3-5

Overview of RF Blockset Equivalent Baseband Libraries

The RF Blockset Equivalent Baseband libraries consist of the Equivalent Baseband (physical) and Idealized Baseband (mathematical) libraries of components for modeling RF systems within the Simulink environment. An RF model can contain blocks from both the physical and mathematical libraries. It can also include Simulink blocks and blocks from other blocksets.

RF Blockset Equivalent Baseband software extends your Simulink modeling environment with a library of blocks for modeling RF systems that include RF filters, transmission lines, amplifiers, and mixers.

You use RF Blockset Equivalent Baseband library blocks to represent the components of your RF system in a Simulink model. The blockset provides several types of component representations using network parameters (S, Y, Z, ABCD, H, and T format), mathematical descriptions, and physical properties.

In the Simulink model, you cascade the components to represent your RF architecture and run the simulation. During the simulation, the model computes a time-domain, complex-baseband representation. This method results in fast simulation of the quadrature modeling schemes used in modern communication systems and enables compatibility with other Simulink blocks.

The blocks let you visualize their specified network parameters using plots and Smith® Charts.

A validated Simulink model of an RF system can provide an executable specification for RF circuit design for wireless communication systems.

You can also use the blockset with Simulink Coder™ software to generate embeddable C code for real-time execution.

Open RF Blockset Equivalent Baseband Libraries

To open the main library window, type the following at the MATLAB prompt:

```
rflib
```

Each yellow icon in the window represents a library. Double-click an icon to open the corresponding library.
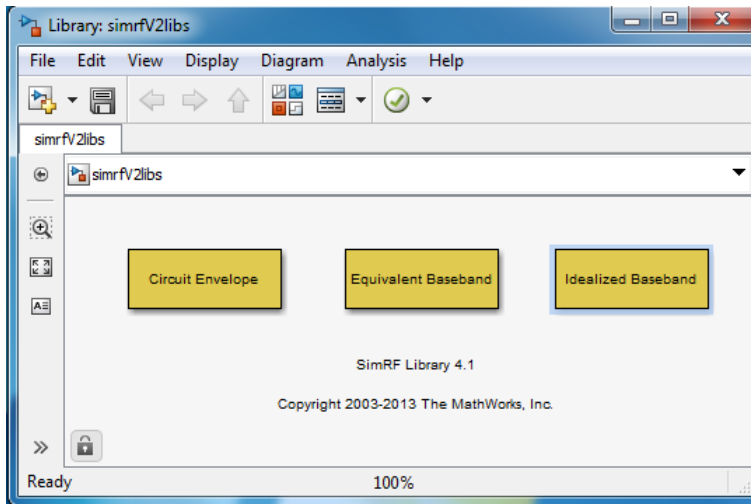
For a discussion of the Equivalent and Idealized Baseband libraries, see the following sections.

Note The blue icons take you to the MATLAB Help browser.

- Double-click the Demos icon to open the RF Blockset Equivalent Baseband examples.
- Double-click the Info icon to open the RF Blockset Equivalent Baseband documentation.

Equivalent Baseband Library

Use blocks from the Equivalent Baseband library to model physical and electrical components by specifying physical properties or by importing measured data. This library includes several sublibraries, as shown in the following figure.
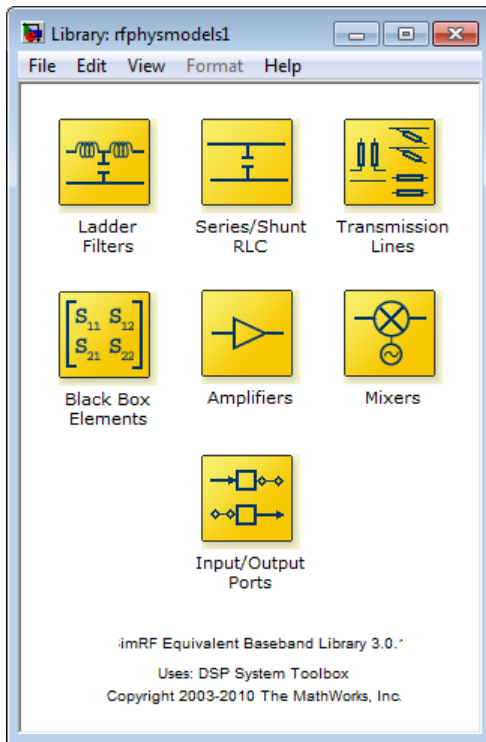
The following table describes the sublibraries and how to use them.

Sublibrary	Description
Amplifiers	RF amplifiers, specified using network parameters, noise data, and nonlinearity data, or a data file containing these parameters.
Ladder Filters	RF filters, specified using LC parameters. The software calculates the network parameters and noise data of the blocks from the topology of the filter and the LC values.
Series/Shunt RLC	Series and shunt RLC components for designing lumped element cascades, specified using RLC parameters. The software calculates the network parameters and noise data of the blocks from the topology of the components and the RLC values.
Mixers	RF mixers that contain local oscillators, specified using network parameters, noise data, and nonlinearity data, or a data file containing these parameters.
Transmission Lines	RF filters, specified using physical dimensions and electrical characteristics. The software calculates the network parameters and noise data of the blocks from the specified data.

Sublibrary	Description
Black Box	Passive RF components, specified using network parameters, or a data file containing these parameters. The software calculates the network parameters and noise data of the blocks from the specified data.
Input/Output Ports	Blocks for specifying simulation information that pertains to all blocks in a physical *subsystem*, such as center frequency and sample time. A physical subsystem is a collection of one or more physical blocks bracketed by an Input Port block and an Output Port block that bridge the physical and mathematical parts of the model.

For more information on defining components, see "Specify or Import Component Data".

Idealized Baseband Library

The Idealized Baseband library contains mathematical representations of the amplifier, mixer, and filter blocks. Use a block from this library to model an RF component in terms of mathematical equations that describe how the block operates on an input signal.

Idealized Baseband blocks assume perfect impedance matching and a nominal impedance of 1 ohm. This means there is no loading and the power flow is unidirectional. As such, they are similar to standard Simulink blocks. In contrast, the Equivalent Baseband blocks do not assume perfect matching—these blocks model the reflections that occur between blocks. Physical blocks model bidirectional power flow, and include loading effects. For these blocks, you can specify the source and load impedances using the Input Port and Output Port blocks.

The mathematical library is shown in the following figure.
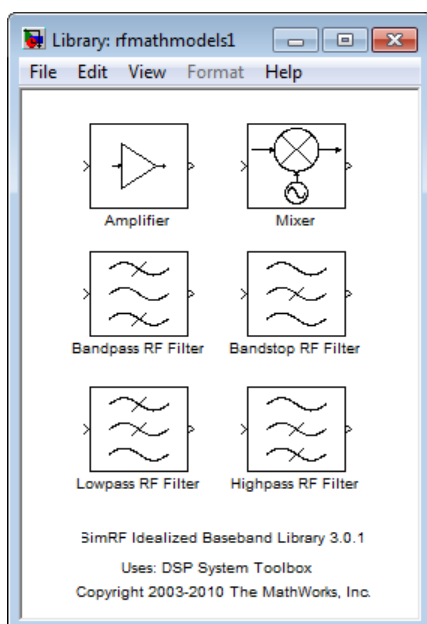
Equivalent Baseband Workflow

When you analyze an RF system using RF Blockset Equivalent Baseband software, your workflow might include the following tasks:

1 Create a Simulink model that includes RF components.

For more information , see "Model RF Components".

2 Define component data by

- Specifying network parameters, mathematical relationships, or physical properties
- Importing data from an industry-standard Touchstone file, a MathWorks AMP file, an Agilent® P2D or S2D file, or the MATLAB workspace

The product lets you access component data in Touchstone SnP, YnP, ZnP, HnP, and GnP formats. You can also import amplifier network parameters and power data from a MathWorks AMP file.

For more information, see "Specify or Import Component Data".

3 Where applicable, add the following information to the component definition:

- Operating condition values (see "Specify Operating Conditions").
- Nonlinearity data (see "Model Nonlinearity").
- Noise data (see "Model Noise").

4 Validate the behavior of individual blocks by plotting component data.

Note You can plot data for individual blocks from the Equivalent Baseband library that model physical components either before or after you run a simulation.

For more information, see "Create Plots".

5 Run the simulation.

For more information on how the product performs time-domain simulation of an RF system, see "Simulate an RF Model".

6 Generate plots to gain insight into system behavior.

For more information, see "Create and Modify Subsystem Plots".

The following plots and charts are available:

- Rectangular plots
- Polar plots
- Smith Charts
- Composite plots
- Budget plots

Model RF Filter Using Equivalent Baseband

In this section...
"Overview of LC Bandpass Filter Example" on page 3-7
"Select Blocks to Represent System Components" on page 3-7
"Build the Model" on page 3-8
"Specify Model Parameters" on page 3-8
"Validate Filter Components and Run the Simulation" on page 3-14
"Analyze the Simulation Results" on page 3-15

Overview of LC Bandpass Filter Example

In this example, you model the signal attenuation caused by an RF filter by comparing the signals at the input and output of the filter.

The RF filter you use in this example is an LC bandpass filter with a bandwidth of 200 MHz, centered at 700 MHz. You use a three-tone input signal to stimulate a range of in-band and out-of-band frequencies of the filter. The input signal has the following tones:

- 700 MHz — Center of the filter
- 600 MHz — Lower edge of the filter passband
- 900 MHz — Outside the filter passband

You simulate the effects of the filter over a bandwidth of 500 MHz.

Select Blocks to Represent System Components

In this part of the example, you select the blocks to represent the input signal, the RF filter, and the signal displays.

You model the RF filter using a physical *subsystem*, which is a collection of one or more physical blocks bracketed by an Input Port block and an Output Port block. The RF filter subsystem consists of an LC Bandpass Pi block, and the Input Port and Output Port blocks. The function of the Input Port and Output Port blocks is to bridge the physical part of the model, which uses bidirectional RF signals, and the rest of the model, which uses unidirectional Simulink signals.

The following table lists the blocks that represent the system components and a description of the role of each block.

Block	Description
Sine Wave	Generates a three-channel signal.
Matrix Sum	Combines the three channel signal into a single three-tone source signal.
Input Port	Establishes parameters that are common to all blocks in the RF filter subsystem, including the source impedance of the subsystem that is used to convert Simulink signals to the RF Blockset Equivalent Baseband physical modeling environment.

Block	Description
LC Bandpass Pi	Models the signal attenuation caused by the RF filter which, in this example, is the LC Bandpass Pi filter.
Output Port	Establishes parameters that are common to all blocks in the RF filter subsystem. These parameters include the load impedance of the subsystem, which is used to convert RF signals to Simulink signals.
Spectrum Analyzer	Displays signals at the input to and output of the filter.

Build the Model

In this part of the example, you create a Simulink model, add blocks to the model, and connect the blocks.

1 Create a model with the blocks shown in the following table. The Library column of the table specifies the hierarchical path to each block.

Block	Library Path	Quantity
Sine Wave	**DSP System Toolbox > Sources**	1
Matrix Sum	**DSP System Toolbox > Math Functions > Matrices and Linear Algebra > Matrix Operations**	1
Spectrum Analyzer	**DSP System Toolbox > Sinks**	2
Input Port	**RF Blockset > Equivalent Baseband > Equivalent Baseband > Input/Output Ports**	1
LC Bandpass Pi	**RF Blockset > Equivalent Baseband > Ladder Filters**	1
Output Port	**RF Blockset > Equivalent Baseband > Input/Output Ports**	1

2 Connect the blocks as shown in the following figure.

For more information on connecting physical and mathematical blocks, see "Connect Model Blocks".
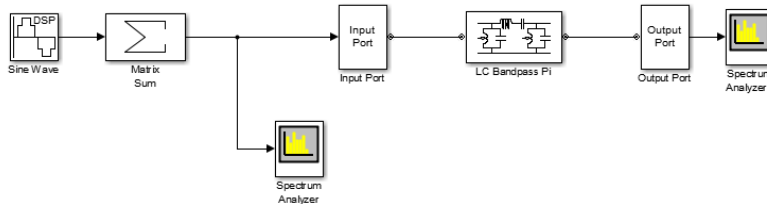
Now you are ready to specify block parameters.

Specify Model Parameters

In this part of the example, you specify the following parameters to represent the behavior of the system components:

- "Input Signal Parameters" on page 3-9
- "Filter Subsystem Parameters" on page 3-11
- "Signal Display Parameters" on page 3-13

Input Signal Parameters

You generate the three-tone source signal using two blocks. You use the Sine Wave block to generate a complex three-channel signal, where each channel corresponds to a different frequency. Then, you use the Matrix Sum block to combine the channels into a single three-tone source signal. Without this block, the signal in all subsequent blocks would have three independent channels.

The RF Blockset Equivalent Baseband simulation algorithm requires you to shift the frequencies of the input signal. The software simulates the filter subsystem using a complex-baseband modeling technique, which automatically shifts the filter response and centers it at zero. You must shift the frequencies of the signals outside the physical subsystem by the same amount.

For more information on complex-baseband modeling, see "Create Complex Baseband-Equivalent Model".

Note All signals in the RF model must be complex to match the signals in the physical subsystem, so you create a complex input signal.

The center frequency of the LC bandpass filter is 700 MHz, so you use a three-tone source signal with tones that are 700 MHz below the actual tones, at -100 MHz, 0 MHz, and 200 MHz, respectively.

1 In Sine Wave block dialog box:

- Set the **Amplitude** parameter to `1e-6`.
- Set the **Frequency (Hz)** parameter to `[-100 0 200]*1e6`.
- Set the **Output complexity** parameter to `Complex`.
- Set the **Sample time** parameter to `1/500e6`.
- Set the **Samples per frame** parameter to `128` .

Source Block Parameters: Sine Wave

Sine Wave (mask) (link)

Output samples of a sinusoid. To generate more than one sinusoid simultaneously, enter a vector of values for the Amplitude, Frequency, and Phase offset parameters.

Main   Data Types

Amplitude:

1e-6

Frequency (Hz):

[-100 0 200]*1e6

Phase offset (rad):

0

Sample mode: Discrete

Output complexity: Complex

Computation method: Trigonometric fcn

Sample time:

1/500e6

Samples per frame:

128

Resetting states when re-enabled: Restart at time zero

OK   Cancel   Help   Apply

2   In the Matrix Sum block dialog box:

- Set the **Sum over** parameter to `Specified dimension`.
- Set the **Dimension** parameter to 2.
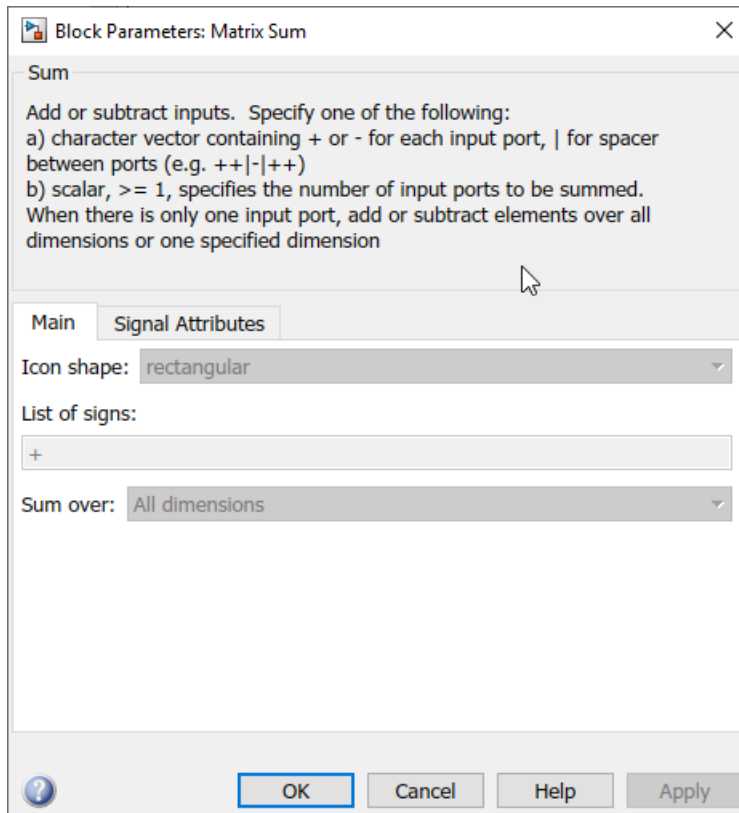
Filter Subsystem Parameters

In this part of the example, you configure the blocks that model the RF filter subsystem—the Input Port, LC Bandpass Pi, and Output Port blocks.

1   Set the Input Port block parameters as follows:

- **Treat input Simulink signal as** = `Incident power wave`

    This option tells the blockset to interpret the input signal as the incident power wave to the RF subsystem, and not the source voltage of the RF subsystem.

    Note If you use the default value for this parameter, the software interprets the input Simulink signal as the source voltage. As a result, the source and the load that model the Input Port and Output Port blocks, respectively, introduce 6 dB of loss into the physical system at all frequencies. For more information on why this loss occurs, see the note in "Convert to and from Simulink Signals".

- **Center frequency** = `700e6`
- **Sample time (s)** = `1/500e6`
- **Input Processing** = `Columns as channels (frame based)`
- Clear the **Add noise** check box so the software does not include noise in the simulation. To learn how to model noise, see "Model Noise".
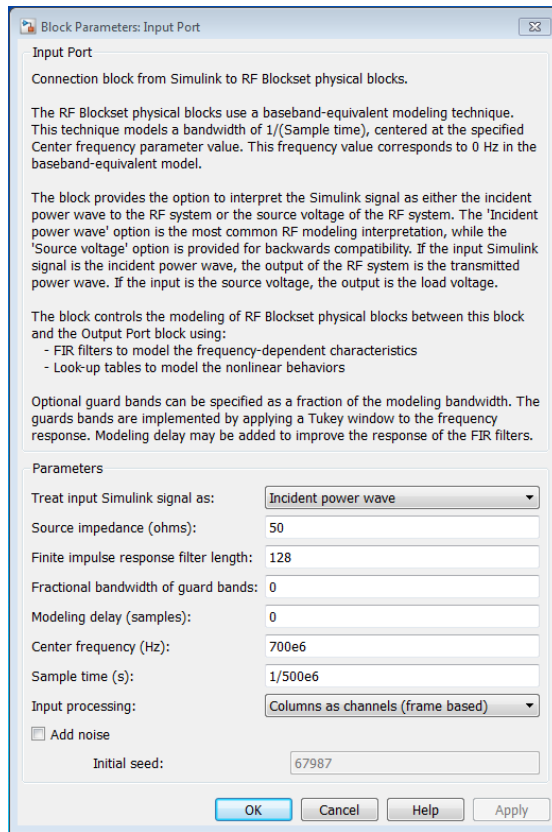
Note You must enter the **Sample time (s)** because the Input Port block does not inherit a sample time from the input signal. The specified sample time must match the sample time of the input signal. The **Sample time (s)** of `1/500e6` second used in this example is equivalent to a bandwidth of 500 MHz.

2   Accept default parameters for inductance and capacitance in the LC Bandpass Pi block. These parameters create a filter with the desired bandwidth of 200 MHz, centered at 700 MHz.
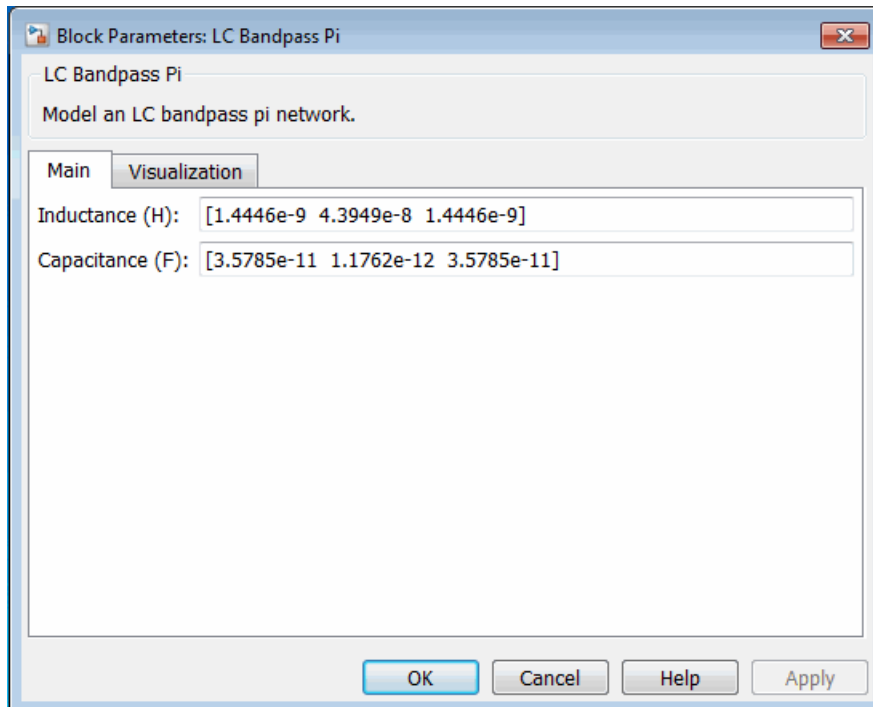
3 Accept the default parameters for the Output Port block to use a load impedance of 50 ohms.
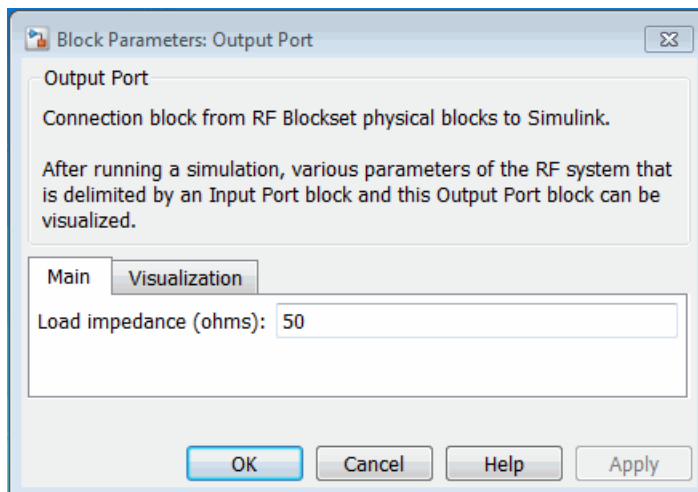
Signal Display Parameters

In this part of the example, you specify:

1 Set the Spectrum Analyzer parameters as follows:

- In the **View** tab check **Spectrum Settings**. In `Trace options` set the **Units** parameter to dBm.

- In the **View** tab open **Configuration Properties**. Set the **Minimum Y-limit** parameter to `-291` and the **Maximum Y-limit** parameter to `-67`. Also, set the **Y-axis label** parameter to dBm.

2   Set the Spectrum Analyzer 1 parameters as follows:

- In the **View** tab check **Spectrum Settings**. In `Trace options` set the **Units** parameter to dBm.

- In the **View** tab open **Configuration Properties**. Set the **Minimum Y-limit** parameter to `-291` and the **Maximum Y-limit** parameter to `-67`. Also, set the **Y-axis label** parameter to dBm.

Validate Filter Components and Run the Simulation

In this part of the example, you validate the behavior of the LC Bandpass Pi filter block by plotting its frequency response and then run the simulation.

Note When you plot information about a physical block, the plot displays the actual frequency response of the block at the selected passband (i.e., the response at the unshifted frequencies), not the response at the shifted frequencies. For more information on this shift, see "Input Signal Parameters" on page 3-9.

1   Double-click the LC Bandpass Pi block to open the block dialog box.

2   Select the **Visualization** tab and click **Plot** to plot the frequency response of the filter. This plots the magnitude of S_{21} as a function of frequency, which represents the gain of the filter.
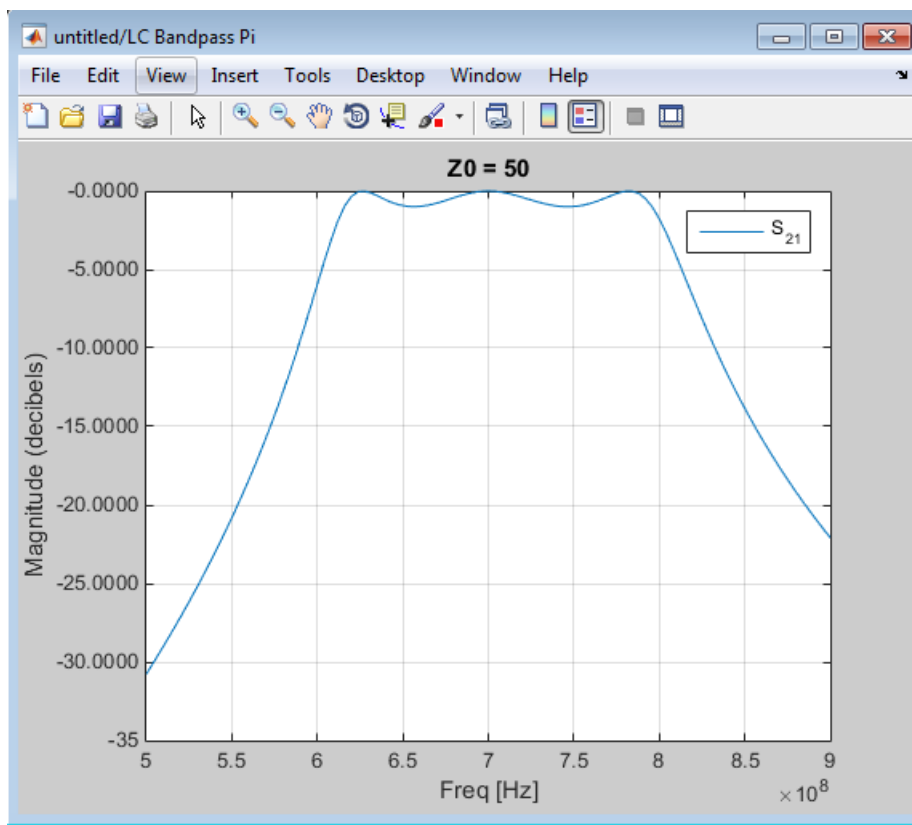
Filter Gain

> **Note** The physical blocks only model a band of frequencies around the center frequency of the physical subsystem. You must choose the sample time and center frequency such that all important frequency characteristics of your physical subsystem fall in this band of frequencies. The plot shows the frequency response of the filter for the portion of the RF spectrum that the physical blocks model. In this example, the physical blocks model a 500-MHz band centered at 700 MHz, as defined by the Input Port block.

3 In the model window, click **Run** to run the simulation.

Analyze the Simulation Results

In this part of the example, you analyze the results of the simulation. This section contains the following topics:

- "Compare the Input and Output Signals of the RF Filter" on page 3-15
- "Plot Model Parameters of the Filter Subsystem" on page 3-17

Compare the Input and Output Signals of the RF Filter

You can view the source signal and the filtered signal in the Spectrum Analyzer windows while the model is running. These windows appear automatically when you start the simulation.

The `Spectrum Analyzer` blocks display the signals at the shifted (baseband-equivalent) frequencies, not at the selected passband frequencies. You can relabel the *x*-axes of the `Spectrum Analyzer` windows to display the passband signal by entering the **Center frequency** parameter value of `700e6` (from the Input Port block) for the **Frequency display offset (Hz)** parameter in the **Axis Properties** tab of the `Spectrum Analyzer` block dialog boxes. For more information on complex-baseband modeling, see "Create Complex Baseband-Equivalent Model".

The `Spectrum Analyzer` blocks display power spectral density normalized to unit sampling frequency. To display power per channel, insert a Gain block with the **Gain** parameter set to `1/sqrt(N)` before each `Spectrum Analyzer` block. `N` is the number of channels. The Gain block is in the **Simulink > Commonly Used Blocks** library.

In this example, `N` is `128` (the value of the **Samples per frame** parameter of the Sine Wave block, `128`).

> **Note** RF Blockset Equivalent Baseband signals represent amplitudes, not voltages. This means that in the product, power is defined as:
>
> $$Power\,(in\,watts)\ =\ [Amplitude\,(in\,volts/sqrt(ohm))]^2$$

The following plot shows the RF filter input signal you specified in the Sine Wave block.
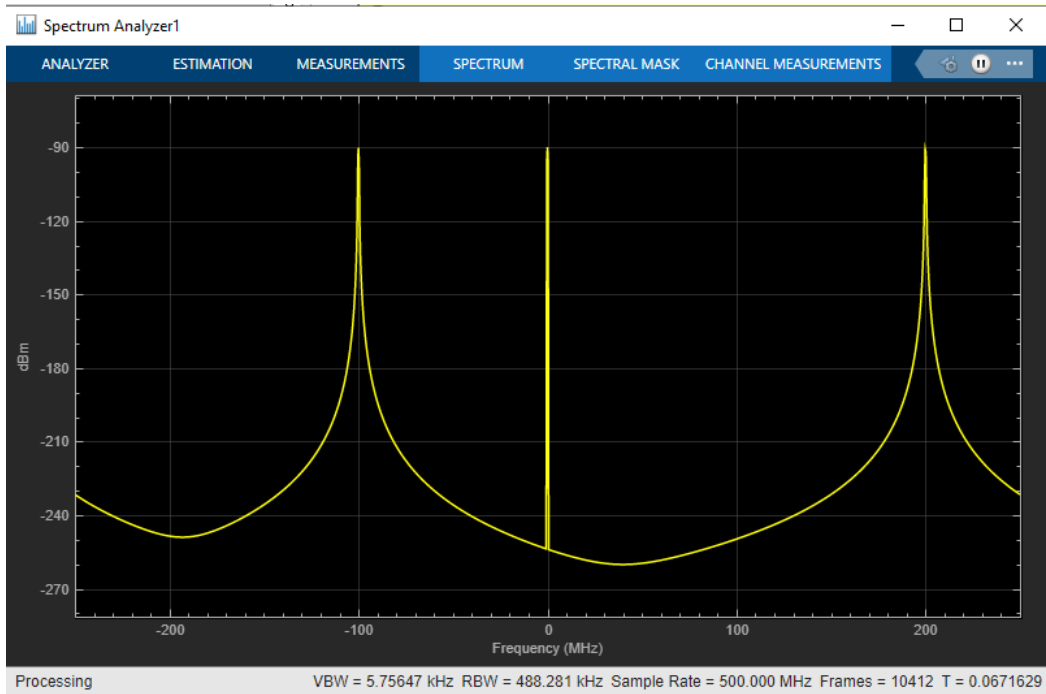
Input to RF Filter

The next plot shows the filtered signal. Notice the RF filter does not attenuate the signal at the center frequency.

Attenuated Output of RF Filter

Plot Model Parameters of the Filter Subsystem

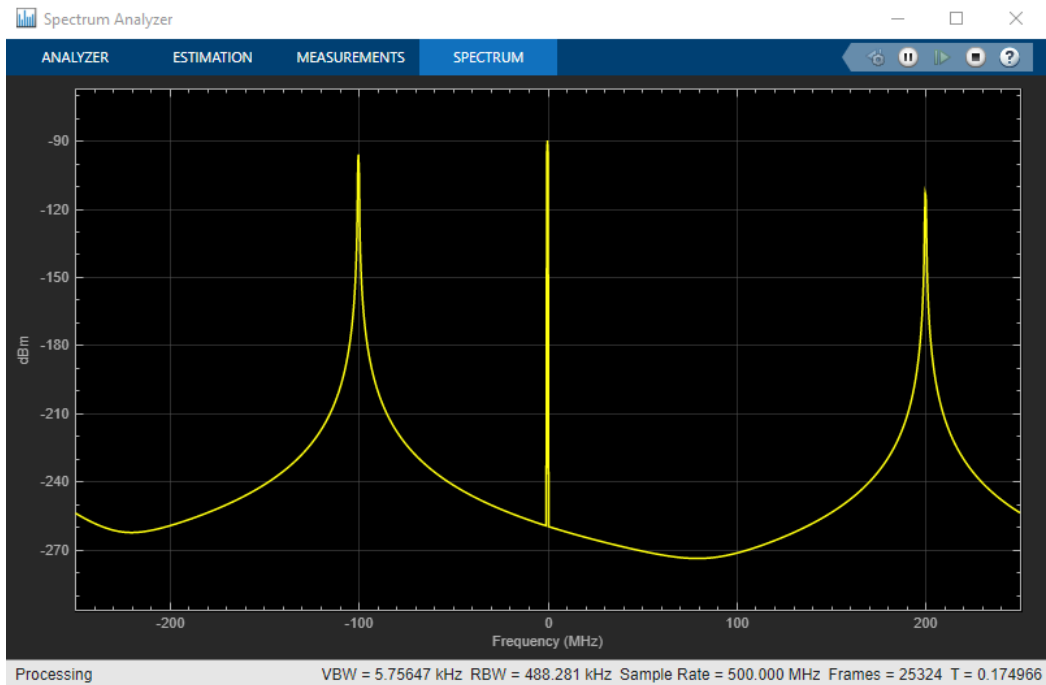After you simulate an RF model, you can evaluate the behavior of the physical subsystem by plotting the network parameters of the Output Port block.

Note When you plot information about a physical subsystem, the plot displays the actual frequency response of the subsystem at the selected passband (i.e. the response at the unshifted frequencies), not the response at the shifted frequencies.

To understand the frequency response of the filter, examine the S-parameters as a function of frequency for the RF filter subsystem on a composite plot.

1 Open the dialog box of the Output Port block by double-clicking the block.

2 Select the **Visualization** tab, and click **Plot**.

The composite plot, shown in the following figure, contains four separate plots in one figure. For the Output Port block, the composite plot shows the following as a function of frequency (counterclockwise from the upper-left plot):

- An X-Y plane plot of the magnitude of the filter gain, S_{21}, in decibels.
- An X-Y plane plot of the phase of the filter gain, S_{21}, in degrees.
- A Z Smith chart showing the real and imaginary parts of the filter reflection coefficient, S_{11}.
- A Polar plane plot showing the magnitude and phase of the filter reflection coefficient, S_{11}.

Note In this example, the response of the filter subsystem is the same as the response of the filter block because the subsystem contains only a filter block.
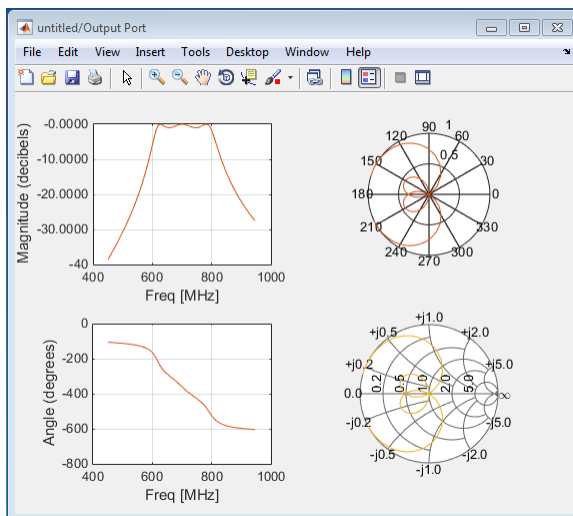
See Also
LC Bandpass Pi

More About

- "Model RF Filter Using Circuit Envelope" on page 1-27

Equivalent Baseband Library

Use Equivalent Baseband Library

The RF Blockset Equivalent Baseband library provides components to model RF systems within the Simulink environment. This library extends your Simulink modeling environment with a library of blocks that can be used to model RF systems that include RF filters, transmission lines, amplifiers, and mixers. The blocks from this library model physical and electrical components when you specify physical properties or import measured data.

The Equivalent Baseband library blocks represents the components of your RF system in a Simulink model. The blockset provides several types of component representations using network parameters (S, Y, Z, ABCD, H, and T format), mathematical descriptions, and physical properties. In the Simulink model, components are cascaded to represent an RF architecture and the model is simulated. During simulation, the model computes a time-domain complex-baseband representation. This method results in fast simulation of the quadrature modeling schemes used in communication systems and enables compatibility with other Simulink blocks.

A validated Simulink model of an RF system can provide an executable specification for RF circuit design of wireless communication systems. The blocks from this library let you visualize their specified network parameters using plots and Smith® charts or noise and power budget curves. Additionally, using this library you can generate embeddable C code for real-time execution using Simulink Coder™.

Open Equivalent Baseband Library

To open the main library window, type this MATLAB Command Window.

```
rflib
```

Each yellow block in the window represents a library.
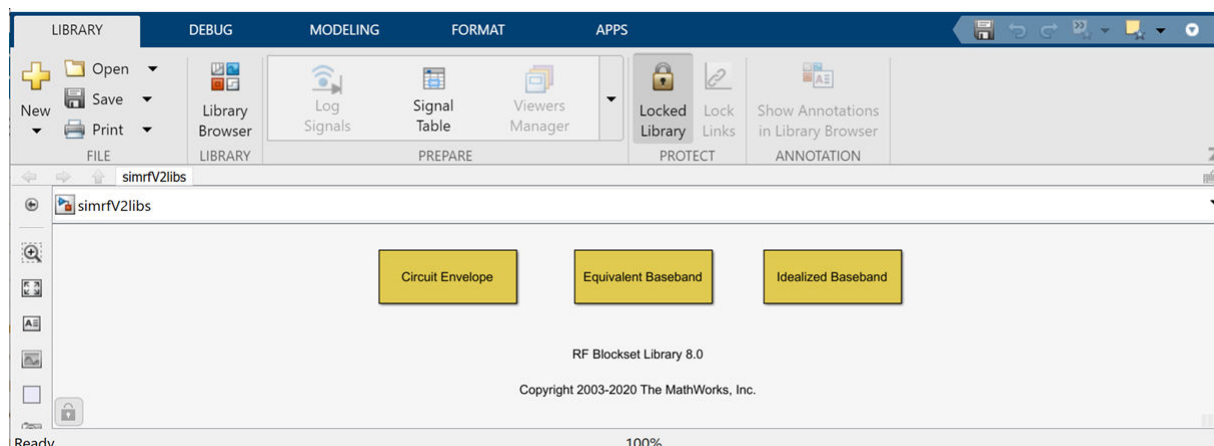
Double-click the **Equivalent Baseband** block to open the library.

This table describes the sublibraries and the types of available models.

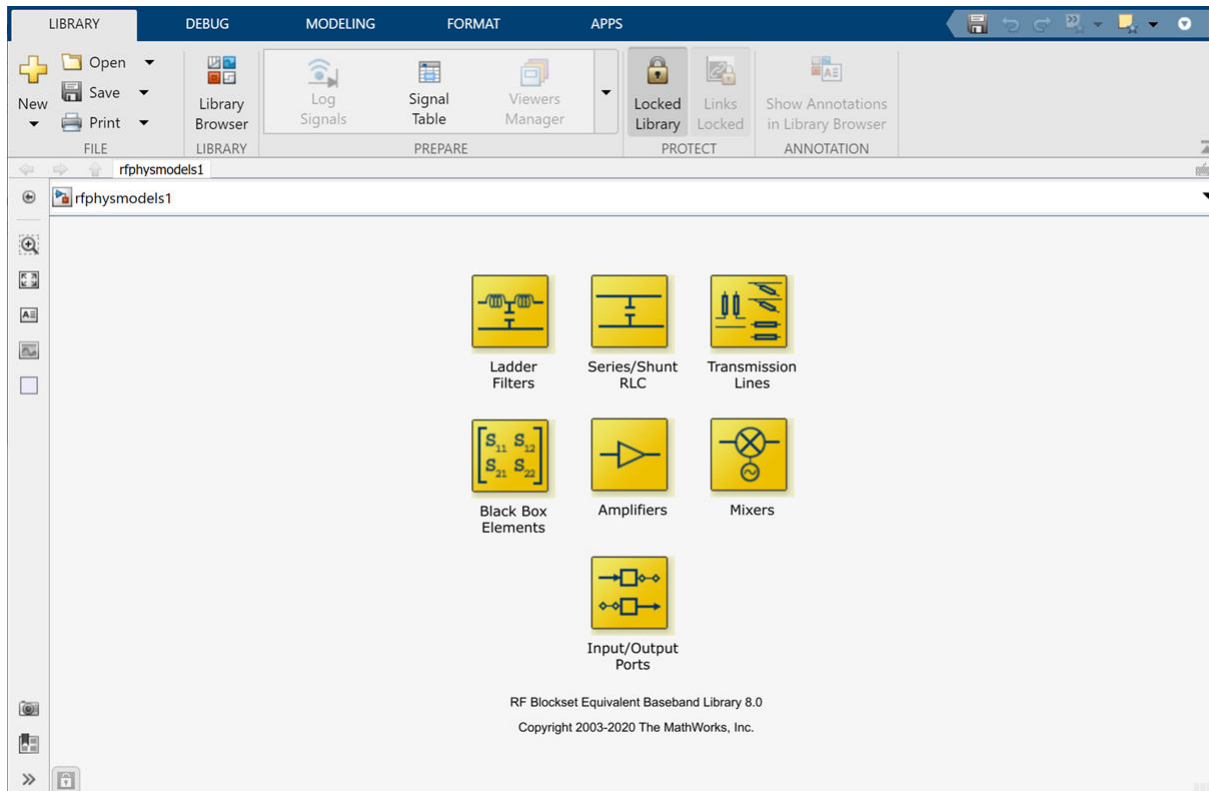| Sublibrary	Use
Ladder Filters	To design RF filters specified using LC parameters. The software calculates the network parameters and noise data of the blocks from the topology of the filter and the LC values.
Series/Shunt RLC	To model series and shunt RLC components for designing the lumped element cascades specified using RLC parameters.
Transmission Lines	To design various types of transmission lines specified using physical dimensions and electrical characteristics.
Black Box	To design passive RF components specified using network parameters, or a data file containing these parameters.
Amplifiers	To design RF amplifiers specified using network parameters, noise data, and nonlinearity data or a data file containing these parameters.
Mixers	To design RF mixers that contain local oscillators specified using network parameters, noise data, and nonlinearity data or a data file containing these parameters.

Sublibrary	Use
Input/Output Ports	To specify simulation information pertaining to all blocks in a physical subsystem, such as center frequency and sample time. A physical subsystem is a collection of one or more physical blocks bracketed by an Input Port block and an Output Port block that bridge the physical and mathematical parts of the model.

For more information on defining components, see "Specify or Import Component Data".

Understanding Equivalent Baseband Environment

The Equivalent Baseband environment comprises of groups of interconnected Equivalent Baseband library blocks and algorithms that model an RF system. In the Equivalent Baseband environment, all blocks fall into one of the two categories.

Blocks Operating Within Equivalent Baseband Environment

These blocks contribute to the physical representation of an RF system. Most Equivalent Baseband library blocks fall into this category. For example, a RLC network can model a source impedance or part of a matching network, and a General Amplifier block can model a physical RF amplifier. Both blocks model physical components. Each of these blocks have two ports and can be exclusively connected to model a chain configuration.

Blocks Operating Between Equivalent Baseband and Simulink Environments

These blocks, also called cross-domain blocks, provide an interface from an RF system to a larger design. The Input Port and the Output Port blocks fall into this category. For example, you can construct a signal using blocks from Communications Toolbox or DSP System Toolbox libraries, and you can input that signal into an RF chain using the Input Port block. The Input Port block also controls the simulation parameters as described in the "Create Complex Baseband-Equivalent Model" section.

See Also

More About

- "Create Complex Baseband-Equivalent Model"

RF Simulation Basics

RF Simulation Glossary

In this section...
"Introduction" on page 4-2
"Terms and Definitions" on page 4-2

Introduction

Using RF Blockset you can model and simulate radar and wireless communications systems with different architectures and used in multiple applications. RF or Radio Frequency deals with signals that work at very high frequency. RF signals change very rapidly and the simulation of such signals require very small time steps and long simulation runs. This combination allows you to capture data in milliseconds or seconds. RF Blockset provides you with two kinds of simulation libraries:

- Circuit Envelope
- Equivalent Baseband

Terms and Definitions

Equivalent Baseband

Equivalent Baseband simulation approach is for 2-port single carrier cascaded networks. This is a discrete time simulation and support frame-based processing of superheterodyne systems You can model filters, S-parameters, and in-band spectral regrowth caused by odd order nonlinearity.

Circuit Envelope

Circuit Envelope simulation approach is for multicarrier simulation of arbitrary network topologies. You can model filters, S-parameters, and in-band spectral regrowth caused by odd order nonlinearity. You can also model interferers, spurs, arbitrary LO signals, DC offset and even order nonlinearity.

Modulation

The process of superimposing a low frequency signal on to a high frequency signal also known as carrier (ω_c). The modulation signal can be complex, and described in terms of in-phase (I) and quadrature-phase (Q) components. The modulation process is equivalent to the complex multiplication:

$$Modulated\ Signal = (I(t) + jQ(t)) * e^{j2\pi\omega_c t}$$

Baseband Signals

Signals that are at or near DC frequencies. This is the original frequency range of the transmitted signal before it is modulated. Baseband signals are limited mainly to wired communications.

Passband Signals

Signals that are at a higher frequency range after modulation. Passband signals are used for wireless transmissions.

Linear Circuits

In linear circuits, amplitude of the output signal is proportional to the input signal amplitude. The basic assumption is that passive devices described by lumped elements such as resistor, capacitor, inductor, or distributed elements such as transmission lines are linear. S-parameter blocks are also linear.

Simulation Time Step

Simulation time step in RF Blockset determines the bandwidth of the envelope signals for both Equivalent Baseband (single carrier envelope) and Circuit Envelope (multicarrier envelope) signals. In order to avoid aliasing, it is recommended that you use the same time step used to define the signal inputs in Simulink.

Mixers

Mixers translate signals from one frequency range to another. They have two inputs and one output. One input is information signal and other is the timing signal also known as Local Oscillator frequency or LO frequency input. Mixers are used for up conversion and down conversion of signals from baseband to RF and vice versa.

Local Oscillators

Local oscillators (LO) generate the timing or reference signals for the mixers.

Introduction to Idealized Baseband Simulation

Idealized Baseband Library

Use Idealized Baseband Library

The RF Blockset Idealized Baseband library extends your Simulink environment with a library of blocks that model single-carrier complex-baseband systems with discrete-time signals. You can directly couple blocks from this library with DSP System Toolbox blocks or other Simulink blocks to estimate the impact of RF phenomena on overall system performance.
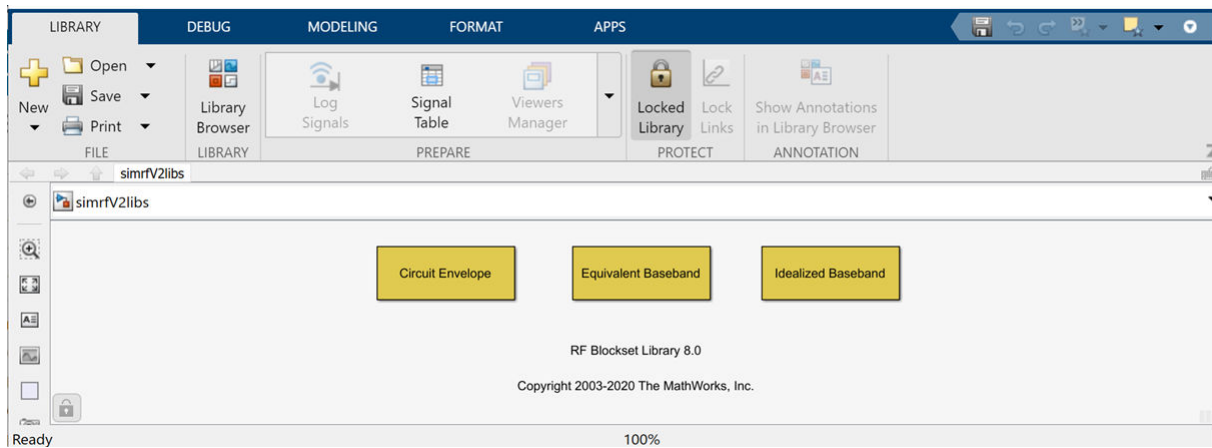
The RF Blockset Idealized Baseband library includes RF amplifiers, filters, and mixers. In the Simulink environment, you connect these Idealized Baseband library blocks to other Simulink blocks to represent a higher abstract level for your RF architecture.

Additionally, you can use this library with Simulink Coder™ software to generate embeddable C code for real-time execution.
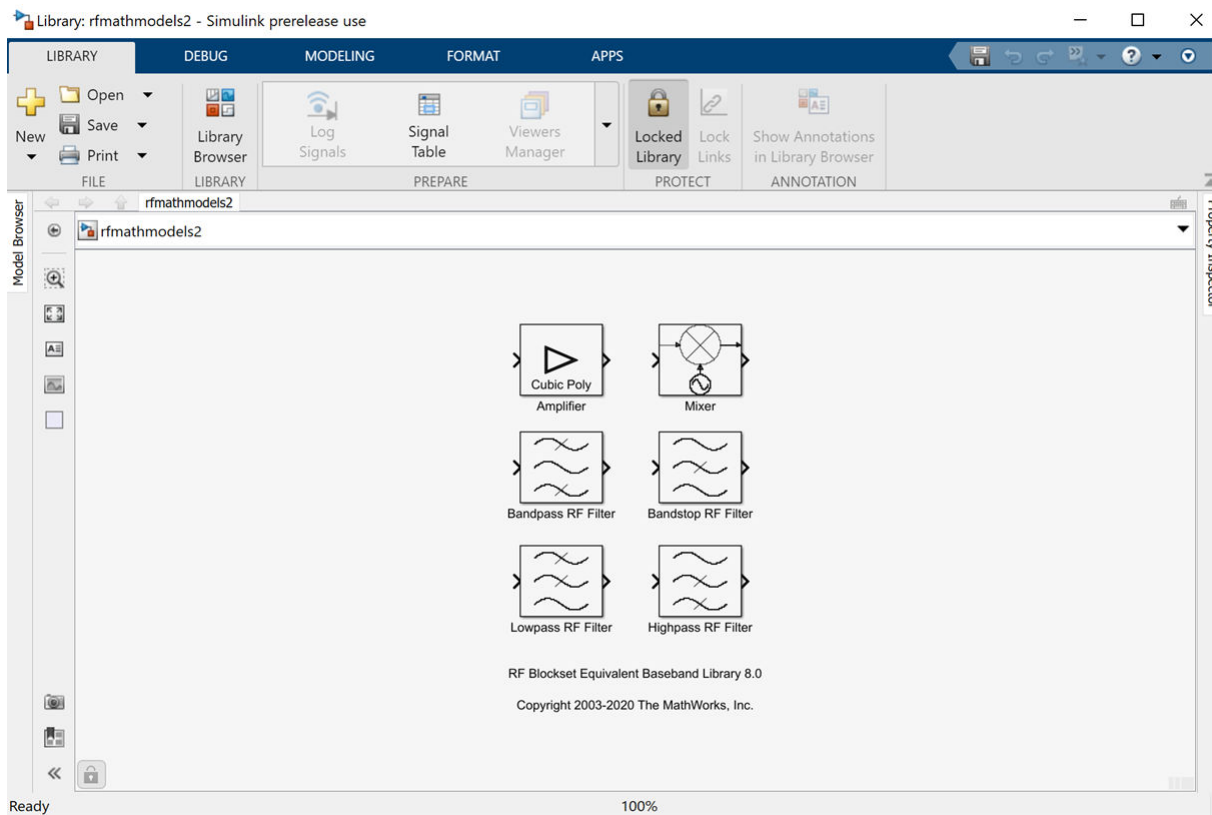
Open Idealized Baseband Library

To open the top-level RF Blockset library window, type this at the MATLAB command line:

```
rflib
```


Each yellow block in the window represents a time-domain complex-baseband library with different capabilities. Double-click the **Idealized Baseband** icon to open the library or type this at the MATLAB command line.

```
rfmathmodels2
```

Idealized Baseband library blocks assume perfect impedance matching. This implies no loading effects and that the signal and power flows are unidirectional. The blocks in this library perform as standard Simulink blocks. In contrast, the Equivalent Baseband and Circuit Envelope blocks do not assume perfect matching. These blocks are physical, model bidirectional power flow, and include loading effects and reflections at each block boundary. The Equivalent Baseband or Circuit Envelope blocks can connect only to blocks from their respective libraries. To interface these two libraries with Simulink blocks, the Input Port and Output Port blocks are required.

See Also

Related Examples

- "Idealized Baseband Amplifier with Nonlinearity and Noise"

Nonlinearities and Noise in Idealized Baseband Amplifier Block

Use the Idealized Baseband Amplifier block to simulate nonlinearities and noise in your RF system design. The Amplifier block provides four nonlinearity models and three options to represent noise.

Nonlinearity Models in Idealized Amplifier Block

Cubic Polynomial

The `Cubic polynomial` model uses linear power gain to determine the linear coefficient of a third-order polynomial and either IP3, P1dB, or Psat to determine the third - order coefficient of the polynomial. The general form of cubic nonlinearity models the AM/AM characteristics as

$$F_{AM/AM}(|u|) = c_1 \times |u| + \frac{3}{4}c_3 \times |u|^3$$

where $F_{AM/AM}(|u|)$ is the magnitude of the output signal, $|u|$ is the magnitude of the input signal, c_1 is the coefficient of the linear gain term, and c_3 is the coefficient of the cubic gain term. The results for IIP3, OIP3, IP1dB, OP1dB, IPsat and OPsat are taken from [1]. The coefficient c_3 are given in this table.

Nonlinearity Type	Equations
Input third-order intercept point, "IIP3 (dBm)"	$$c_3 = -\frac{4c_1}{3 \times 10^{[(IIP3 - 30)/10]}}$$ IIP3 is given in dBm.
Output third-order intercept point, "OIP3 (dBm)"	$$c_3 = -\frac{4c_1^3}{3 \times 10^{[(OIP3 - 30)/10]}}$$ OIP3 is given in dBm.
Input 1 dB gain compression power, "IP1dB (dBm)"	$$c_3 = -\frac{2c_1(10^{19/20} - 10)}{15 \times 10^{[(IP1dB - 30)/10]}}$$ IP1dB is given in dBm.
Output 1 dB gain compression power, "OP1dB (dBm)"	$$c_3 = -\frac{2c_1^3(10^{19/20} - 10)}{15 \times 10^{[(OP1dB - 30 - LGdB + 1)/10]}}$$ OP1dB is given in dBm, and LGdB is the linear gain in dB.
Input saturation power, "IPsat (dBm)"	$$c_3 = -\frac{4c_1}{9 \times 10^{[(IPsat - 30)/10]}}$$ IPsat is given in dBm.

Nonlinearity Type	Equations
Output saturation power, "OPsat (dBm)"	$$c_3 = -\frac{16c_1^3}{81 \times 10^{[(OPsat - 30)/10]}}$$ OPsat is given in dBm.

AM/AM-AM/PM

The `AM/AM-AM/PM` model uses a lookup table to specify the amplifier power characteristics. The table returns interpolated or extrapolated values using linear interpolation. Each row in the table expresses the relationship between output power or phase change as a function of input power.

$$u_{\text{out}} = Table_{AM/AM}(u) \times e^{((Table_{AM/PM}(u) + \angle u) \times i)}$$

where u_{out} is the output signal and u is the magnitude of input signal.

Saleh

The `Saleh` model is based on normalized transfer function. Use the input / output scaling parameters to adjust signal levels from their normalized values. For `Saleh`, the AM/AM parameters $alpha_{AM/AM}$ and $beta_{AM/AM}$ are used to compute the amplitude gain for an input signal using the following equation:

$$F_{AM/AM}(u) = \frac{alpha_{AM/AM} \times |u|}{1 + beta_{AM/AM} \times |u|^2}$$

where $|u|$ is the magnitude of the scaled signal and `u` is calculated as:

$$u = InputScale \times u_{\text{in}}$$

For `Saleh`, the AM/PM parameters $alpha_{AM/PM}$ and $beta_{AM/PM}$ are used to compute the phase change for an input signal using the following equation:

$$F_{AM/PM}(u) = \frac{alpha_{AM/PM} \times |u|^2}{1 + beta_{AM/PM} \times |u|^2} + angle(u)$$

where $|u|$ is the magnitude of the scaled signal and *angle* is a MATLAB function that returns the phase angle of *u*.

The scaled output signal, u_{out} is calculated as:

$$u_{\text{out}} = F_{AM/AM} \times e^{(F_{AM/PM} \times i)} \times OutputScale$$

Modified Rapp

The `Modified Rapp` model is based on normalized transfer functions. Use the input and output scaling parameters to adjust the signal levels from their normalized values. The AM/AM characteristics for `Modified Rapp` are given by:

$$F_{AM/AM}(u) = \frac{g_{lin} \times |u|}{(1 + \left|\frac{g_{lin} \times u}{V_{sat}}\right|^{2p})^{1/2p}}$$

where $|u|$ is the magnitude of input signal, g_{lin} is $10^{\text{(Linear Gain (dB)/20)}}$, and is the amplitude gain of the amplifier, V_{sat} is "Output saturation level (V)", and p is "Magnitude smoothness factor".

The AM/PM characteristics for `Modified Rapp` is given by

$$F_{AM/PM}(u) = \frac{A \times |u|^q}{[1 + (\frac{|u|}{B})]^q} + angle(u)$$

where u is the input signal, A is the **Phase gain (rad)**, B is **Phase saturation**, q is **Phase smoothness factor** and *angle* is a MATLAB function which returns phase angle of u.

The output signal u_{out} is calculated as:

$$u_{\text{out}} = F_{AM/AM}(u) \times e^{(F_{AM/PM}(u) \times i)}$$

Plot Power Characteristics

To visualize the functionality of **Plot power characteristics** button, you can set the parameters of the Amplifier block as listed in the table.

Model	Parameters	Power Characteristics Plot
Cubic Polynomial	**Main** tab: • **Linear power gain (dB)**: 7 • **Type of Non-linearity**: IIP3 • **IIP3 (dBm)**: 33 • **Simulate using**: Code generation **Noise** tab: • **Specify noise type**: Noise temperature • **Noise temperature**: 290 • **Seed source**: Auto	

Model	Parameters	Power Characteristics Plot
AM/AM - AM/PM	**Main** tab: • **Lookup table (Pin(dBm), Pout(dBm), deg)**: `[-25, 5, -1; -10, 20, -2; 0, 27, 5; 5, 28, 12]` • **Simulate using**: `Code generation` **Noise** tab: • **Specify noise type**: `Noise temperature` • **Noise temperature**: `290` • **Seed source**: `User specified` • **Seed** : `67987`	
Modified Rapp	**Main** tab: • **Linear power gain (dB)**: `7` • **Output saturation level (V)**: `1` • **Magnitude smoothness factor**: `2` • **Phase gain (rad)**: `-.45` • **Phase saturation**: `0.88` • **Phase smoothness factor**: `3.43` • **Simulate using**: `Code generation` **Noise** tab: • **Specify noise type**: `Noise temperature` • **Noise temperature**: `290` • **Seed source**: `User specified` • **Seed** : `67987`	

Model	Parameters	Power Characteristics Plot
Saleh	**Main** tab: • **Input scaling (dB)**: `0` • **AM/AM parameters [alpha beta]**: `[ 2.1587, 1.1517 ]` • **AM/PM parameters [alpha beta]**: `[ 4.0033, 9.1040 ]` • **Output scaling (dB)**: `0` • **Simulate using**: `Interpreted execution` **Noise** tab: • **Specify noise type**: `Noise figure` • **Noise figure**: `10 * log10( 2 )` • **Seed source**: `Auto`	

Application of Nonlinearities

All four subsystems for the amplifier nonlinearity models apply a memoryless nonlinearity to the complex baseband input signal. Each model

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 model, 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 model, and adds the result to the angle of the signal to produce the angle of the output signal.

Thermal Noise Simulations in Idealized Amplifier Block

According to the **Specify noise type** parameter, you can specify the amount of thermal noise in three ways,

• `Noise temperature` — Specifies the noise in kelvin.
• `Noise factor` — Specifies the noise by using the equation:

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

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

Noise figure = 10×log(Noise factor)

Note Some RF Blockset blocks require the sample time to perform baseband modeling calculations. To ensure accuracy in 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 these times do not match, or if the input sample time is missing because the blocks are not connected, an error message appears.

To learn how to use the idealized baseband library Amplifier block to amplify a signal with nonlinearity and noise, see "Idealized Baseband Amplifier with Nonlinearity and Noise".

References

[1] Kundert, Ken." Accurate and Rapid Measurement of IP$_2$ and IP$_3$," *The Designer Guide Community*, May 22, 2002.

See Also

Amplifier | Mixer

Related Examples

•   "Nonlinearities and Noise in Idealized Baseband Mixer Block" on page 5-10

Nonlinearities and Noise in Idealized Baseband Mixer Block

Use the Idealized Baseband Mixer block to simulate nonlinearities and noise in your RF system design. The block uses a cubic polynomial model to simulate nonlinearities. You can plot power characteristics, Pout (dBm) vs. Pin (dBm) using **Plot power characteristics** button. For more information, see "Plot Power Characteristics" on page 5-11.

The Mixer block supports both phase and system noise. Frequency dependent LO phase noise vs. frequency offset is modeled in this block. This block allows you to plot phase noise magnitude response (dBc/Hz vs. Frequency) using **Plot phase characteristics** button. For more information, see "Phase Noise in Mixer Block" on page 5-11 and "Plot Phase Noise Characteristics" on page 5-12.

"Noise temperature (K)", "Noise figure (dB)", and "Noise factor" parameters allows you to set the amount of system noise added to the input signal. For more information, see "Mixer (System) Noise Simulations" on page 5-13.

Nonlinearities in Idealized Baseband Mixer Block

The Mixer block uses third-order cubic polynomials to simulate nonlinearities. The model uses a linear power gain to determine the linear coefficient of a third-order polynomial and IP3, P1dB, or Psat to determine the third-order coefficient of the polynomial. The general form of cubic nonlinearity models the AM/AM characteristics as

$$F_{AM/AM}(|u|) = c_1 \times |u| + \frac{3}{4} c_3 \times |u|^3$$

$F_{AM/AM}(|u|)$ is the magnitude of the output signal, $|u|$ is the magnitude of the input signal, c_1 is the coefficient of the linear gain term, and c_3 is the coefficient of the cubic gain term. The equations for IIP3, OIP3, IP1dB, OP1dB, IPsat, and OPsat are taken from [1]. The c_3 coefficient is determined as follows.

Type of Nonlinearity	Equations
Input third-order intercept point, "IIP3 (dBm)"	$c_3 = -\dfrac{4c_1}{3 \times 10^{[(IIP3 - 30)/10]}}$ IIP3 is given in dBm.
Output third-order intercept point, "OIP3 (dBm)"	$c_3 = -\dfrac{4c_1^3}{3 \times 10^{[(OIP3 - 30)/10]}}$ OIP3 is given in dBm.
Input 1 dB gain compression power, "IP1dB (dBm)"	$c_3 = -\dfrac{2c_1(10^{19/20} - 10)}{15 \times 10^{[(IP1dB - 30)/10]}}$ IP1dB is given in dBm.

Type of Nonlinearity	Equations
Output 1 dB gain compression power, "OP1dB (dBm)"	$$c_3 = -\frac{2c_1^3(10^{19/20} - 10)}{15 \times 10^{[(OP1dB - 30 - LGdB + 1)/10]}}$$ OP1dB is given in dBm and LGdB is the linear gain in dB.
Input saturation power, "IPsat (dBm)"	$$c_3 = -\frac{4c_1}{9 \times 10^{[(IPsat - 30)/10]}}$$ IPsat is given in dBm.
Output saturation power, "OPsat (dBm)"	$$c_3 = -\frac{16c_1^3}{81 \times 10^{[(OPsat - 30)/10]}}$$ OPsat is given in dBm.

Plot Power Characteristics

Visualize the power characteristics of your system design by using the Mixer block to plot the Pout vs. Pin curves. For example, plot the power characteristics for a system with a conversion gain of 10 dB and IIP3 nonlinearity. On the **Main** tab, set **Conversion gain (dB)** to 10. On the **Impairments** tab, set **Type of non-linearity** to IIP3 and set **IIP3 (dBm)** to 33. Click **Plot power characteristics** on the **Impairments** tab.
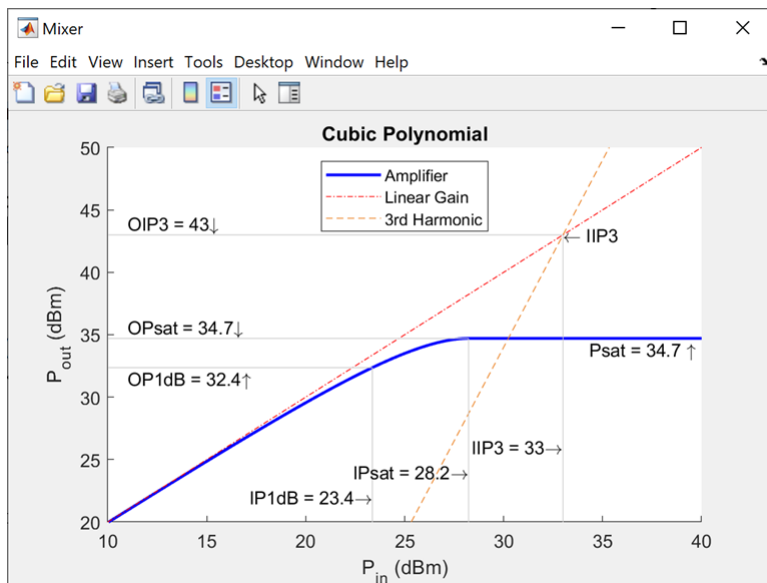
While plotting the power characteristics in this example, set all other parameters on the Impairments tab to default values.

Phase Noise in Mixer Block

Frequency dependent LO phase noise vs. frequency offset is modeled in this block using the MATLAB digital `filter` function where the white noise input is generated from the MATLAB random number

generator `randn` with the specified generator *twister*. The numerator and denominator `filter` coefficients are derived using two methods to model the phase noise. You can set the LO phase noise and frequency offset using the "Phase noise level (dBc/Hz)" and "Frequency offset (Hz)" parameters, respectively.

Method one is employed for a scalar **Phase noise level (dBc/Hz)** parameter and has an initial `-10` dBm change in phase noise level per frequency decade for frequencies greater than the specified **Frequency offset (Hz)** [2]. Using this method, an IIR digital filter is constructed. This is because the rational transfer function is defined with a constant numerator coefficient and *N* denominator coefficients. The number of denominator coefficients, *N*, is proportional to the **block sample rate / Frequency offset**.

Method two is employed for vector **Phase noise level (dBc/Hz)** parameter values. For modeling purposes, when the frequency is less than the smallest specified **Frequency offset (Hz)** parameter value, extrapolated phase noise values have a `1/f`^{3} dependence. If the frequency is greater than the largest **Frequency offset (Hz)** parameter value, the extrapolated phase noise values are set equal to the final **Phase noise level (dBc/Hz)** vector value. Using this method, an FIR digital filter is constructed. FIR digital filter is constructed because the rational transfer function is defined with a constant denominator coefficient and *N* numerator coefficients. The number of numerator coefficients, *N*, is proportional to **block sample rate** / **Frequency offset**. To reduce spectral leakage when simulating, an additional step is performed using an Hanning filter while deriving the filter coefficients.

When the LO phase noise parameter, "Automatic frequency resolution", is enabled, the block sample rate and frequency resolution are derived from the **Frequency offset (Hz)** parameter and they are used to determine the required number of filter coefficients. The number of filter coefficients can be determined using the equation

$$N = \frac{1}{\delta_f \times \delta_t} = \frac{\text{Bandwidth}}{\delta_f}$$

where,

δ_f is the frequency resolution

δ_t is the block sample time

The frequency resolution is chosen to ensure that a minimum of two modeling points exist between any two specified phase noise level points in the design. This choice for modeling points often leads to many filter coefficients with an adverse effect on the simulation speed. The model automatically limits the number of filter coefficients in the range [2^5,2^16].

To improve the simulation speed, either increase the minimum distance between the **Frequency offset (Hz)** parameter values or disable **Automatic frequency resolution** and specify the "Number of signal samples".

**Plot Phase Noise Characteristics**

Plot the phase noise magnitude response by clicking the **Plot phase characteristics** button. The plot displays the phase noise specification, the design specification, and the filter response of the last simulation.

If a simulation has not been performed, the sample rate is estimated from the **Frequency offset (Hz)** parameter. For the plot shown in the table, the number of frequency bins is `4096`. The number of bins can be calculated using this equation

$$\text{Number of Frequency Bins} = \frac{\text{Sample Rate (Hz)}}{\text{Resolution (Hz)}}$$

For example, plot the phase noise characteristics of a Mixer block. On the "Noise Tab" tab, set the following parameters and click **Plot phase characteristics** button.

- **Include phase noise**: On
- **Phase noise level (dBc/Hz)**: -150
- **Frequency offset (Hz)**: 1024
- **Automatic frequency resolution**: On
- **Seed source**: Auto
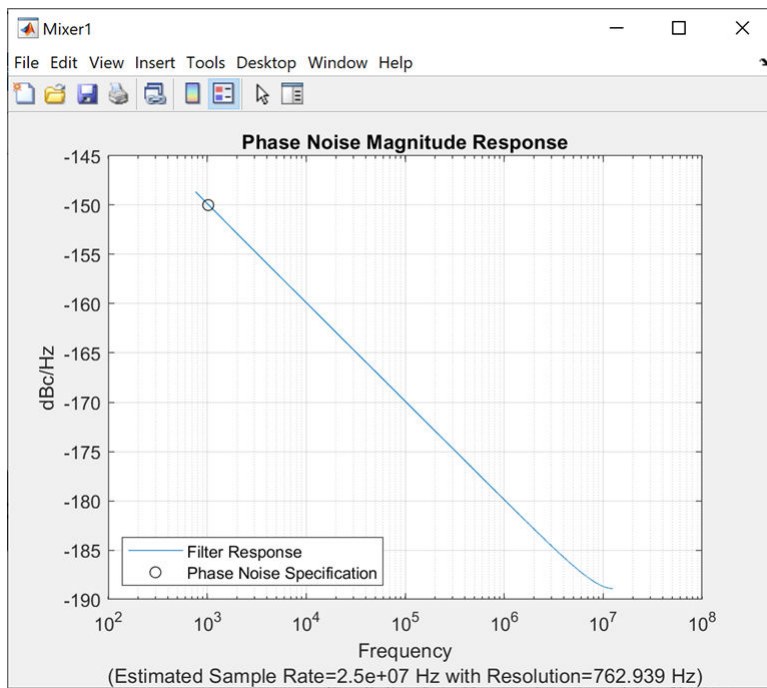
While plotting the power characteristics in this example, set all other parameters on the Impairments tab to default values.

Mixer (System) Noise Simulations

Specify the amount of system noise added to the input signal by setting the **Mixer noise type**. Set one of the following:

- Noise temperature — Specify the noise in kelvin. The noise added to the system is proportional to the square root of Noise temperature. The Noise temperature is calculated using this equation

  $$\text{Noise temperature} = 290 \times (10^{(\text{Noisefigure}/10)} - 1)$$

- Noise factor — Specify the noise by using the equation:

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

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

  Noise figure = 10×log(Noise factor)

References

[1] Kundert, Ken. Accurate and Rapid Measurement of IP$_2$ and IP$_3$. *The Designer Guide Community*, May 22, 2002.

[2] Kasdin, N.J. Discrete Simulation of Colored Noise and Stochastic Processes and 1/f/Sup α/ Power Law Noise Generation. *Proceedings of the IEEE* 83, no. 5 (May 1995): 802–27. https://doi.org/10.1109/5.381848.

See Also

Amplifier | Mixer

Related Examples

- "Nonlinearities and Noise in Idealized Baseband Amplifier Block" on page 5-4