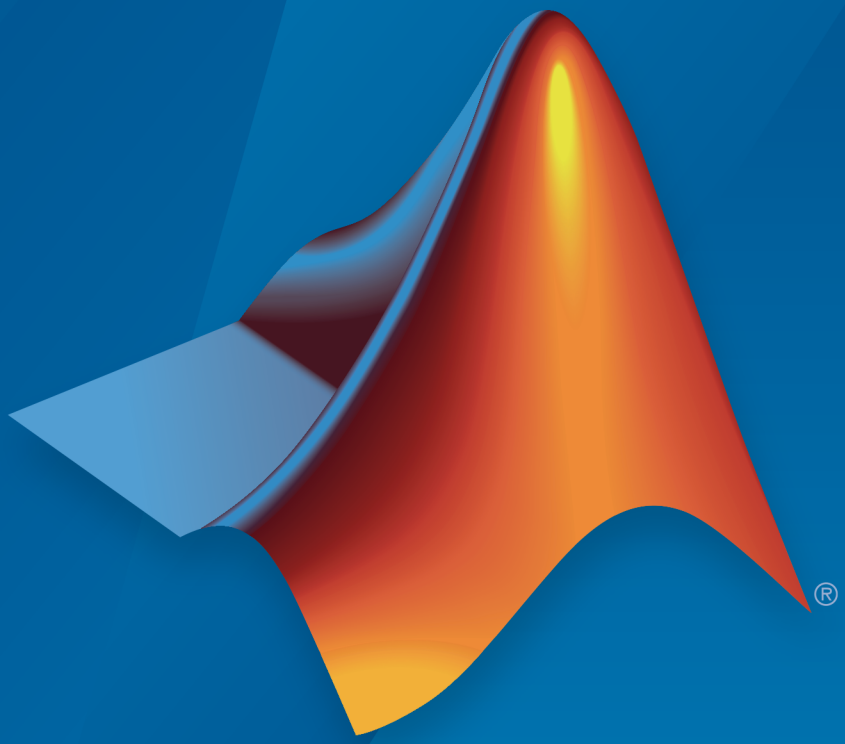
SimRF™

Getting Started Guide



MATLAB® & SIMULINK®

R2016b

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

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

Phone: 508-647-7000

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

SimRF™ Getting Started Guide

© COPYRIGHT 2010–2016 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

Revision History

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

Contents

Circuit Envelope

1 Introduction to Circuit Envelope Simulation

2 Frequency Conversion

Equivalent Baseband

3 Introduction to Equivalent Baseband Simulation

Circuit Envelope

1

Introduction to Circuit Envelope Simulation

SimRF Product Description

Design and simulate RF systems

SimRF provides a component library and simulation engine for designing RF systems. It includes amplifiers, mixers, S-parameter blocks, and other basic blocks for designing architectures for wireless transmitters and receivers in communication and radar systems. You can connect blocks arbitrarily to form diverse architectures and simulate the system-level behavior of the RF front-end.

SimRF lets you simulate RF amplifiers to estimate gain, noise, even-order, and odd-order intermodulation distortion. You can simulate mixers to predict image rejection, reciprocal mixing, local oscillator phase offsets, and DC conversion. You can also simulate frequency-dependent mismatches between linear and nonlinear components in the time and frequency domains. The RF Budget Analyzer app lets you generate test benches to validate your transmitter or receiver performance and automatically set up a circuit envelope simulation.

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

Key Features

- Circuit envelope simulation of multiple carrier-frequency models
- Test bench generation from the RF Budget Analyzer app
- General N-port models and S-parameter data files for time-domain and frequency-domain simulation
- Passive components, including RLC elements, transmission lines, filters, switches, junctions, and general impedance blocks
- Enhanced highly-nonlinear models of 3-port mixers and 2-port amplifiers specified by noise figure, IP2, IP3, and data files
- Model authoring using the Simscape™ language
- Equivalent baseband technology for discrete-time simulation of single-carrier cascaded systems

Circuit Envelope Library

Elements

`Amplifier`	Model amplifier in RF systems
`Capacitor`	Model capacitor for circuit envelope analysis
`Ideal Transformer`	Model ideal transformer
`Filter`	Model an RF filter
`Impedance`	Model complex impedance
`Inductor`	Model inductor for circuit envelope analysis
`LC Ladder`	Model LC ladder networks
`Mixer`	Model mixer in RF systems
`Mutual Inductor`	Model two coupled inductors for circuit envelope analysis
`Phase Shift`	Model phase shift in RF systems
`Resistor`	Model resistor for circuit envelope analysis
`S-Parameters`	Model S-parameter network
`Signal Combiner`	Compute sum of RF signals
`Three-Winding Transformer`	Model three coupled inductors for circuit envelope analysis
`Transmission Line`	Model transmission line
`VGA`	Model VGA (variable gain amplifier)

Junctions

`Circulator`	Model ideal frequency-independent circulators with S-parameters
`Coupler`	Model ideal frequency-independent couplers with S-parameters
`Divider`	Model ideal frequency-independent dividers (combiners) with S-parameters
`Potentiometer`	Model Simulink® controlled potentiometer
`Switch`	Model Simulink controlled two terminal switch

SPST	Model single pole single throw (SPST) switch
SPDT	Model single pole double throw (SPDT) switch

Sources

Continuous Wave	Model constant envelope source
Noise	Model noise using current or voltage noise source in RF systems
Sinusoid	Model DC offset and sinusoidal modulation

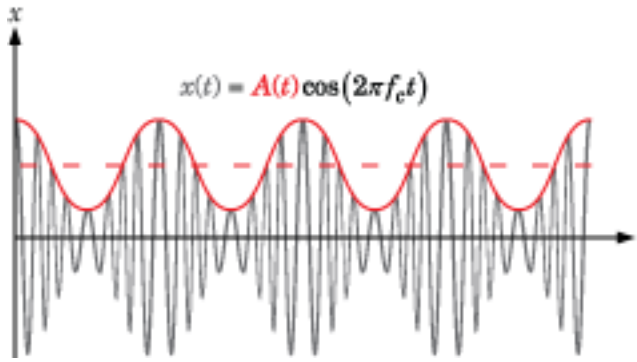
Utilities

Configuration	Specify system-wide parameters for circuit envelope analysis
Inport	Convert Simulink input signal to SimRF signal
Outport	Convert SimRF signal to Simulink output signals
Connection Port	Connection port for RF subsystem
Ground	Simulate connection to electrical ground

Circuit Envelope Basics

In Simulink, simulating signals with high-frequency components requires choosing a proportionately small time step. However, the *envelope* of such signals often varies on a much larger time scale. The circuit envelope environment takes advantage of this condition in order to model RF signals accurately, while also reducing simulation time.

The following figure illustrates a signal $x(t)$ for which circuit envelope simulation is ideal:



The signal consists of a time-varying modulating signal on a high-frequency carrier. In many RF applications, the frequency of the modulating signal $A(t)$ is much less than the frequency of the carrier, f_c.

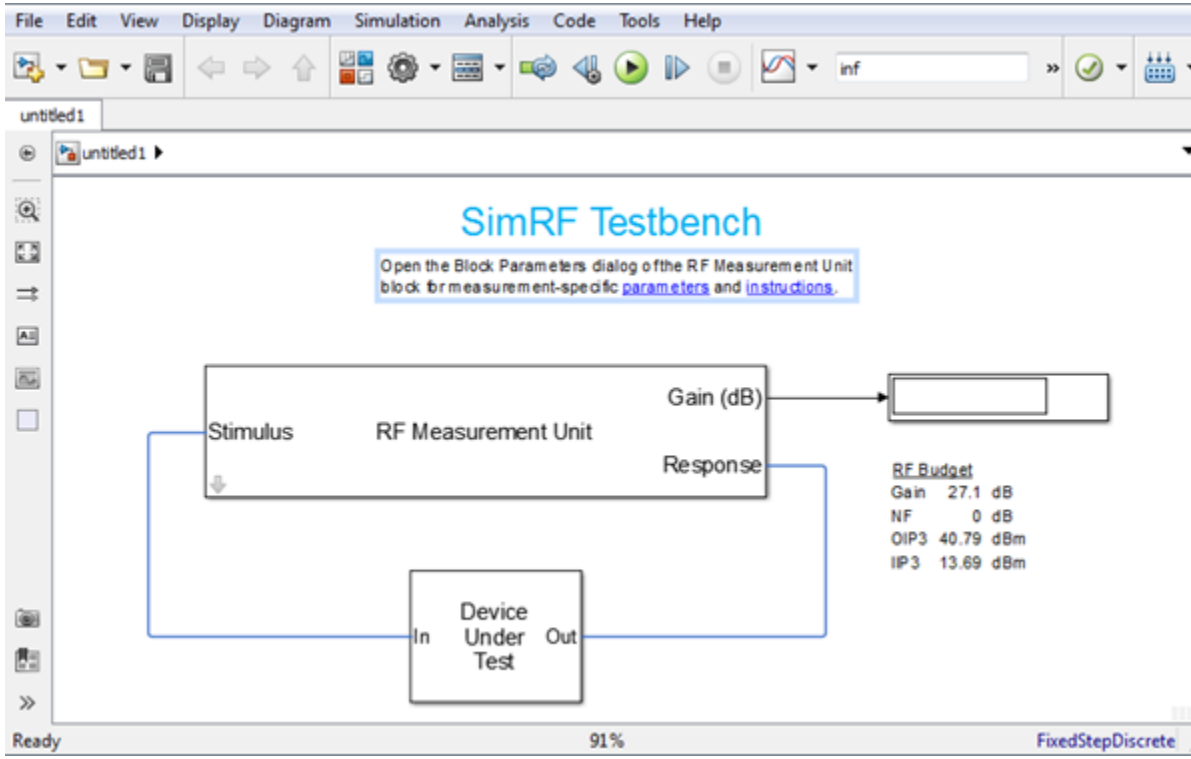
SimRF software handles the carrier $\cos(2\pi f_c t)$ analytically, so it only simulates the modulating signal. The redefined system that the software uses is mathematically equivalent to the original. However, by taking time steps on the scale of the modulating signal instead of the carrier, the software produces equivalent results in less time.

Using SimRF Testbench

In this section...

Introduction

Use the SimRF testbench to verify the cumulative gain, noise figure, and nonlinearity (IP3) values of an RF system generated using the **RF Budget Analyzer** app. To use the testbench, create a system in the **RF Budget Analyzer** app and click `Export` > `Export to SimRF Testbench`.

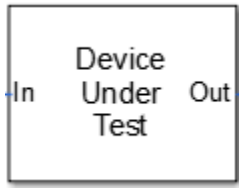


The testbench is made up of two subsystems:

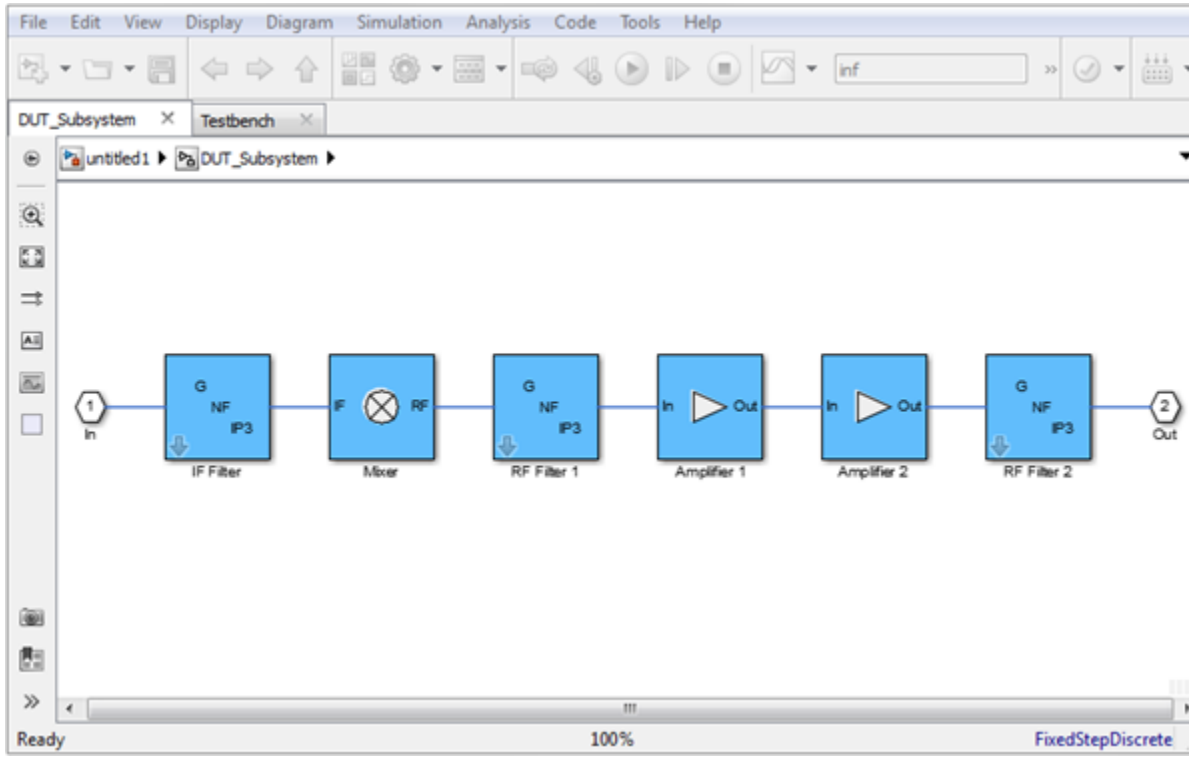
- `RF Measurement Unit`
- `Device Under Test`

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

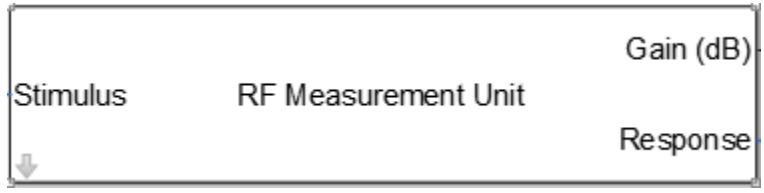
Device Under Test



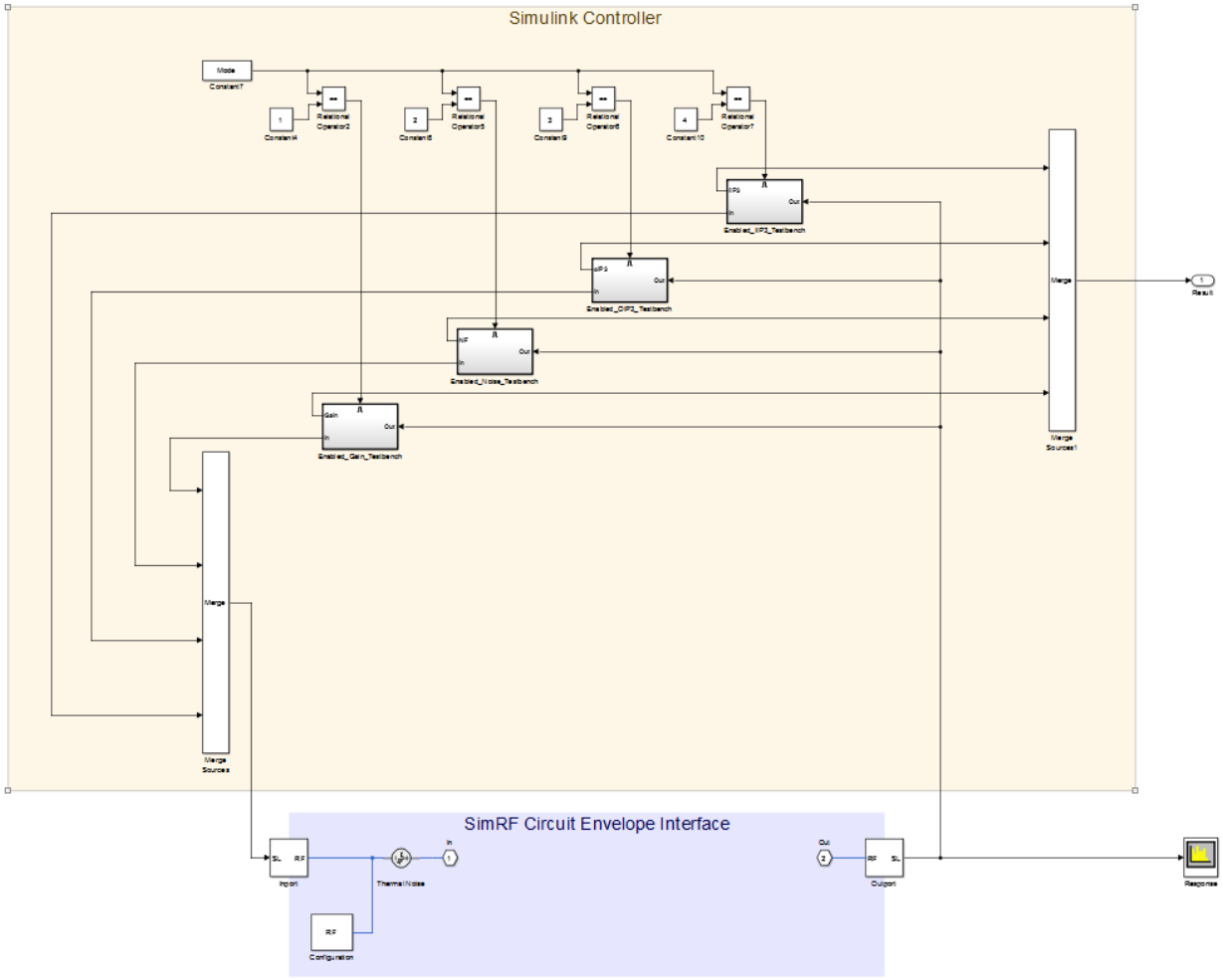
The Device Under Test subsystem contains the RF system exported from the app. To see the RF system, double-click the subsystem.



RF Measurement Unit



The RF Measurement Uni subsystem consists of a Simulink Controller and SimRF Circuit Envelope interface. The SimRF interface is used as input and output from the DUT. To look inside the `RF Measurement Unit` subsystem, click the arrow below **Stimulus**.



RF Measurement Unit Parameters

Double-click the RF Measurement Uni subsystem block to open the parameters.

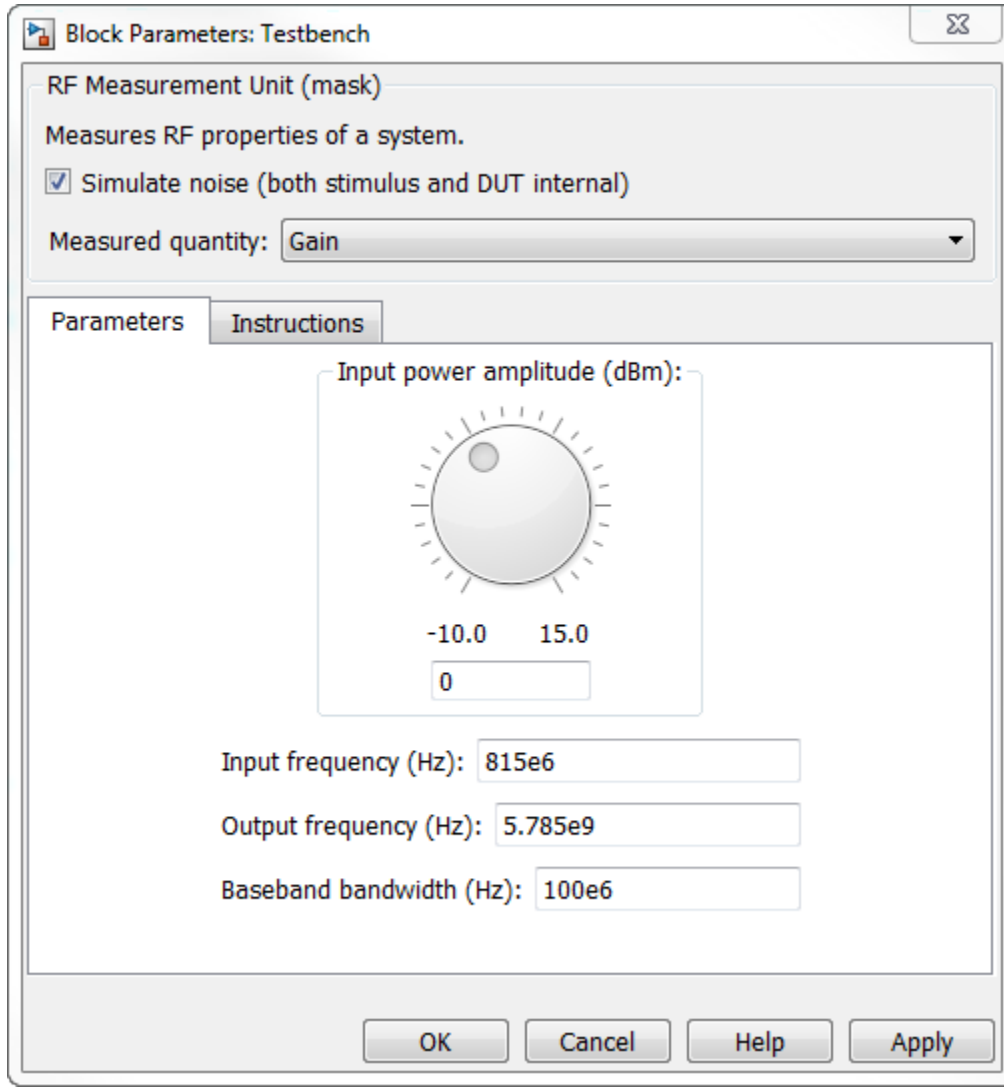
- **Simulate noise (both stimulus and DUT)** — Select this check box to enable noise modeling in the stimulus signal entering the DUT and inside the DUT.

- **Measured Quantity** — Choose the quantity you want to verify from: `Gain`, `NF`, `OIP3`, `IIP3`. By default, the testbench verifies `Gain`.

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

The two tabs are: **Parameters** and **Instructions**.

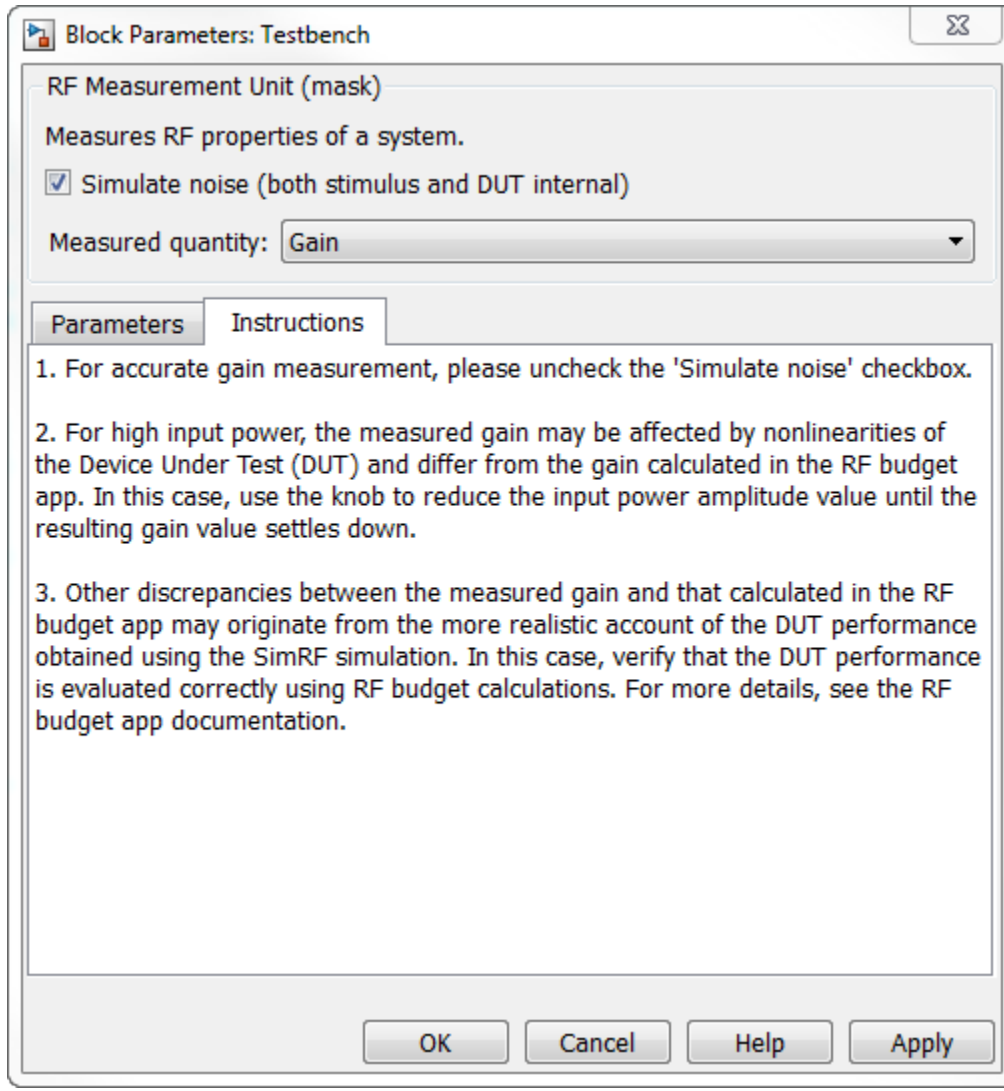
Parameters



- **Input power amplitude (dBm)** — Input power to the DUT. You can change the input power by manually specifying or by turning the knob.
- **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`.

Instructions



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 SimRF 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.

See Also

RF Budget Analyzer

Simulate High Frequency Components

In this section...

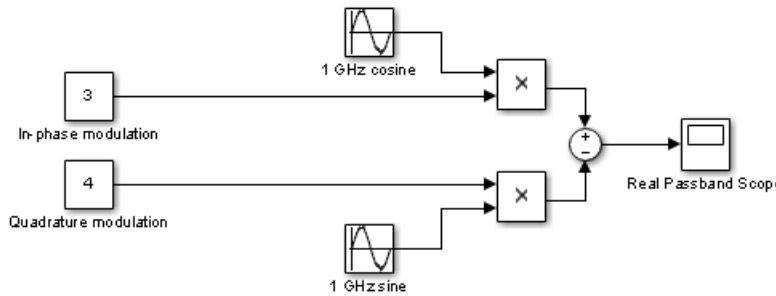
This example shows how to use SimRF circuit envelope simulation to simulate high frequency components while reducing simulation time.

Simulate a Passband Signal in Simulink Software

The model, `ex_simrf_tut_passband`, shows how to modulate a real passband signal with in-phase and quadrature components.

To open this model, at MATLAB® command line, enter:

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



The system specifies a real passband signal $x(t)$ according to the formula

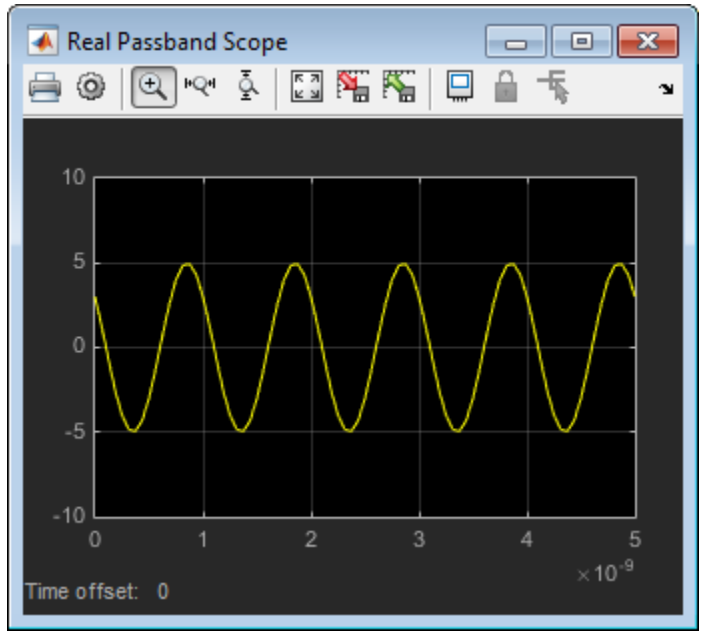
$$x(t) = I(t)\cos\left(2\pi f_c t\right) - Q(t)\sin\left(2\pi f_c t\right)$$

where:

- $I(t)$ is the in-phase part of the modulating signal, equal to 3 in this example, modeled by the Constant block labeled In-phase modulation.

- $Q(t)$ is the quadrature part of the modulating, equal to 4 in this example, modeled by the Constant block labeled Quadrature modulation.
- f_c is the carrier frequency, equal to 1 GHz in this example.

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°.

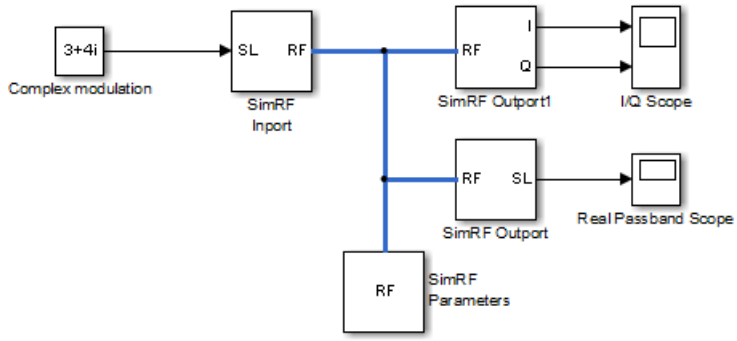
In the Configuration Parameters dialog box, the **Fixed-step size (fundamental sample time)** parameter has been set to `1/16*1e-9`. This value is on the order of the wavelength of the carrier. The simulation takes a total of 81 samples — 16 per cycle.

Compare Passband and Baseband Signals in SimRF Software

The model, `ex_simrf_tut_compare`, 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” on page 1-15.

To open this model, at MATLAB command line, enter:

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



The system simulates a real passband signal as the real part of a complex passband signal according to the formula

$$x(t) = \mathrm{Re}\left[\left(I(t) + jQ(t)\right)\exp\left(j \cdot 2\pi f_c t\right)\right] = I(t)\cos\left(2\pi f_c t\right) - Q(t)\sin\left(2\pi f_c t\right)$$

where:

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

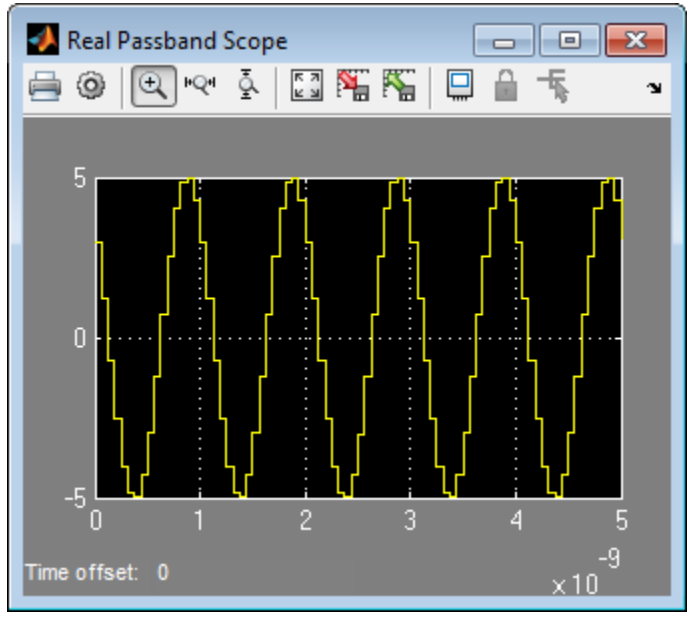
Contrary to the Simulink passband implementation in the previous section, the complex baseband signal driving the SimRF system does not include the carrier. Instead, the SimRF environment handles the carrier analytically. The carrier appears in four different blocks in the SimRF environment:

- In the `Inport` block, the **Carrier frequencies** parameter defines the carrier frequencies of the modulations entering from outside the SimRF 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

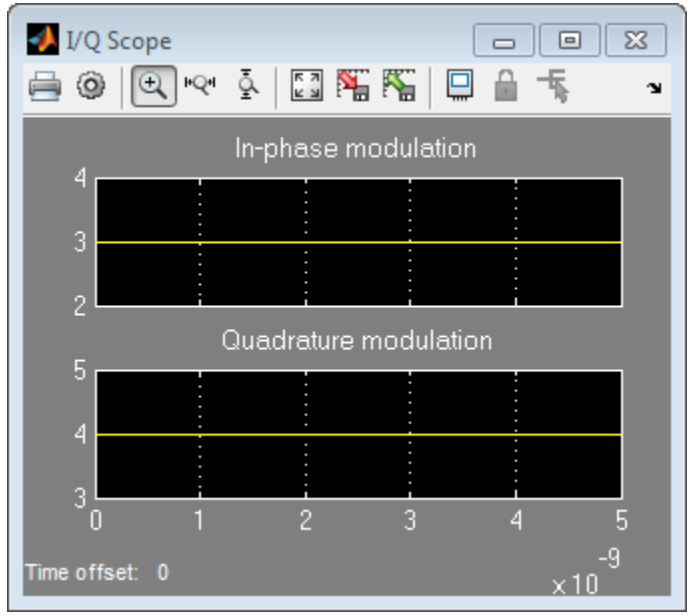
ports. The **Output** parameter is set to `Real Passband`, so this signal represents a real passband signal on the 1-GHz carrier.

- In the block labeled SimRF 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 SimRF 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” on page 1-15. 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” on page 1-15.

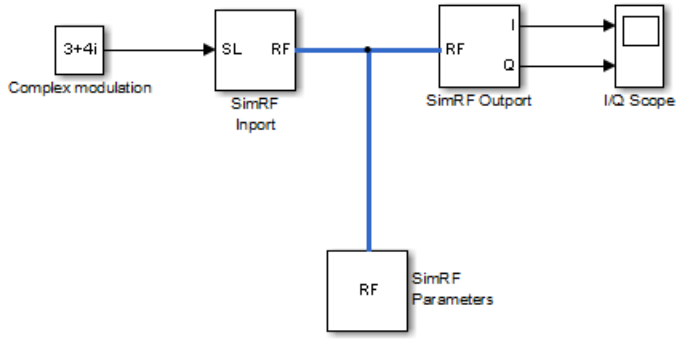
In the Configuration Parameters dialog box, the **Fixed-step size (fundamental sample time)** parameter has been set to `1/16*1e-9`. This value is on the order of the wavelength of the carrier. The simulation takes a total of 81 samples — 16 per cycle.

Simulate the Envelope of a Baseband Signal

The model, `ex_simrf_tut_envelope`, shows how to simulate the envelope of a sine wave using SimRF blocks. This section builds on the results of the previous section, “Compare Passband and Baseband Signals in SimRF Software” on page 1-16.

To open this model, at MATLAB command line, enter:

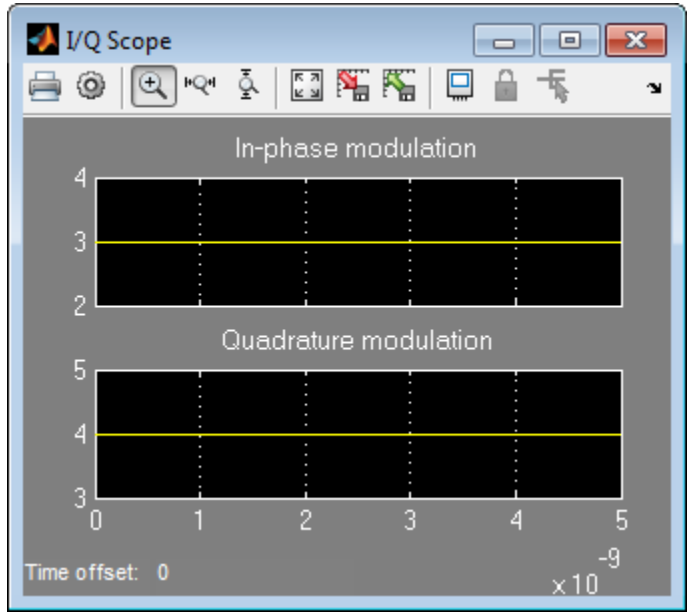
```
addpath(fullfile(docroot,'toolbox','simrf','examples'))
ex_simrf_tut_envelope
```



The system is almost identical to the system in the previous section, except:

- The model contains only one SimRF Outport block and only one scope. The SimRF environment outputs in-phase and quadrature modulations of the 1-GHz signal. In the SimRF 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.

Using SimRF Software for the First Time

In this section...

Expected Background

Topics in the SimRF 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 and Equivalent Baseband Features

SimRF software offers Circuit Envelope and Equivalent Baseband libraries for modeling RF networks. Each library represents a distinct simulation paradigm. For certain applications, one library may offer an advantage over another.

- Use the Circuit Envelope library for multicarrier simulation of RF networks with arbitrary topologies.
- Use the Equivalent Baseband library for single-carrier simulation of cascaded RF networks.

Except for the cross-domain SimRF Inport, SimRF Outport, Input Port, and Output Port blocks, blocks from the three libraries do not connect to each other. Therefore, to avoid redesigning your model later, choose which library to use based on your application. You can consult the following table for a summary of the features of the two libraries.

Summary Of Features

Do blocks in this library...	Circuit Envelope Library	Equivalent Baseband Library
Connect directly to Simulink blocks?	No, except for cross-domain blocks	No, except for cross-domain blocks
Support single-carrier simulation?	Yes	Yes

Do blocks in this library...	Circuit Envelope Library	Equivalent Baseband Library
Support multicarrier simulation?	Yes	No
Support nonlinear elements?	Yes	Yes
Support simulation of cascaded networks?	Yes	Yes
Support simulation of networks with arbitrary topologies?	Yes	No
Support signal probing between input and output?	Yes	No
Support noise simulation?	Yes	Yes
Support Simscape platform features, such as local solvers?	Yes	No
Support specification using network parameter data?	Yes	Yes

Understanding the SimRF Environment

Groups of interconnected SimRF Circuit Envelope library blocks and the algorithms that model the RF system that they represent comprise the *SimRF environment*. In the SimRF environment, all blocks fall into one of the three following categories.

Blocks that Operate Within the SimRF Environment

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

SimRF blocks connect via Simscape electrical terminals and lines. For an introduction to Simscape software and physical networks, see Basic Principles of Modeling Physical Networks.

Blocks that Convert Between the SimRF and Simulink Environments

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

The Configuration block

This blocks manipulates the environment itself. To use this block, connect it to any part of the RF system. Because it is not part of the physical representation of the system, it has the same effect regardless of where you connect it.

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

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`:

1. In Simulink editor, select **File** > **Model Properties** > **Model Properties**.
2. In the **Model Properties** dialog box, on the **Callbacks** tab in **Model callbacks** pane, select `InitFcn`.
3. In the **Model initialization function** pane, enter:

```
amp = ones(1,3)
freq = [600 700 900]*1e6
stepsize = 1/500e6
```

4. Click **OK**.
5. In the Simulink tool bar, change the Simulation stop time to 0.
6. In Simulink editor, select **Simulation** > `Model configuration Parameters`. In the **Configuration Parameters** dialog box, on the **Solver** tab, in **Solver options** change the **Solver** to `discrete (no continuous states)`.

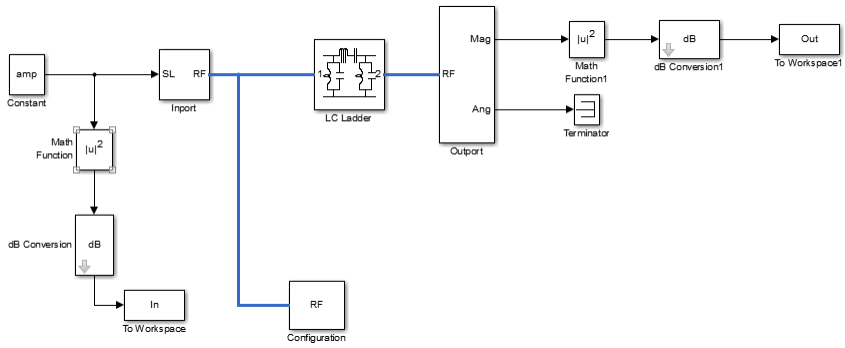
Required Blocks

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	Description	Quantity
Constant	**Simulink > Sources**	Generates real and complex constant value	1
Inport	**SimRF > Circuit Envelope > Utilities**	Convert Simulink input signal to SimRF input signal	1
Configuration	**SimRF > Circuit Envelope > Utilities**	Set the system wide parameters for SimRF simulation	1
LC Ladder	**SimRF > Circuit Envelope > Elements**	Models the signal attenuation caused by the LC Ban	1
Outport	**SimRF > Circuit Envelope > Utilities**	Convert SimRF signals to Simulink Signals	1
db Conversion	**DSP System Toolbox > Math Functions > Math Operations**	Convert magnitude data to decibels	2
Math Function	**Simulink > Math Operations**	Performs mathematical function.	2
To workspace	**Simulink > Sinks**	Write data to a MATLAB workspace for plotting	2
Terminator	**Simulink > Commonly Used Blocks > Terminator**	Terminate the angle baseband output of the Outport block	1

Connect the blocks as shown in the figure:



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 stepsize.

1. In the Constant block dialog box set the **Constant** value to `amp`, as defined in the `InitFcn`.
2. 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**.
3. In the Configuration block dialog box:
   - 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

1 In the LC Ladder block dialogue box:

- Set **Ladder topology** to `LC Bandpass Pi`.

  Click **Apply** and then click **OK**.

Configure Output Settings

1 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**.

2 In the Simulink editor, connect the Ang port of the Outport block to Terminator block to terminate the angle baseband output.

3 In **Math Function** and **Math Function 1** block dialog boxes, set **Function** to `magnitude^2` and click **OK**. The block squares the magnitude of the input and output signal.

4 In **dB Conversion** and **dB Conversion 1** block dialog boxes, set **Input Signal** to `Power` and click **OK**. The block converts the input and output signals to dB.

5 In **To Workspace**, change the **Variable name** to `In`. In **To Workspace 1**, change the **Variable name** to `Out`. In both the block dialog boxes, change the set **Save format** to `Array` and click **OK**.

6 Use `Simulation > Run` to run the model.

Plot and Analyze Attenuated Output Signals

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

1 Transfer the input and output dB values to the MATLAB workspace using To Workspace block.

2 To view the input signal, plot `In` array from the MATLAB workspace :

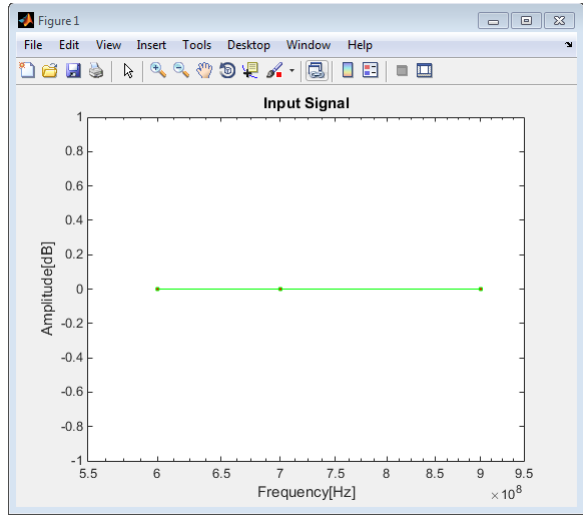
```
figure
h = semilogx(freq, In,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
```

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

3 To view the output signal, plot `Out` array from the MATLAB workspace :

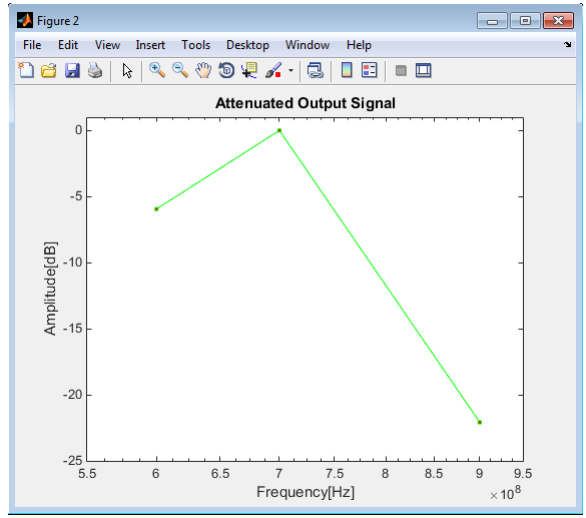
```
figure
h = semilogx(freq, Out,'-gs','LineWidth',1,'MarkerSize',3,'MarkerFaceColor','r');
xlim([5.5e8,9.5e8])
ylim([-25,1])
xlabel('Frequency[Hz]')
ylabel('Amplitude[dB]')
title('Attenuated Output Signal')
```

4 Compare the input and output signal plots to verify the attenuation caused by the filter.



Input Signal to RF Filter

5 The following plot shows the filtered attenuated signal.



Attenuated Output Signal

Notice that the RF filter does not attenuate the signal at the center frequency of 700 MHz.

Analyze LC Bandpass Filter Response

1. 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
```

2. Run the model. Notice that the signal is not attenuated within the 200 MHz range of the LC bandpass filter.
3. Plot the attenuated output:

```
figure
h = 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')
```

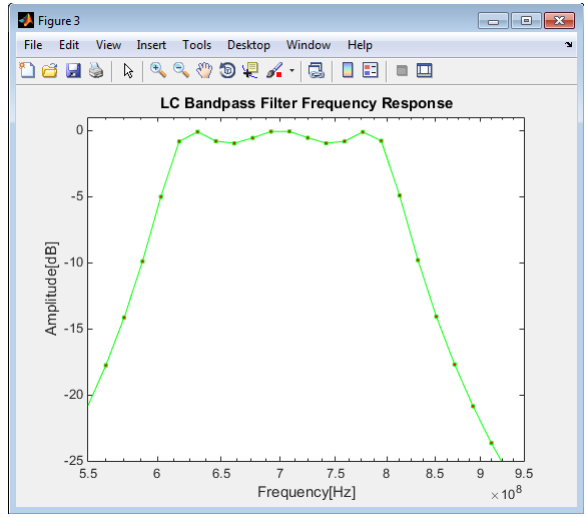


Figure 3
File Edit View Insert Tools Desktop Window Help
LC Bandpass Filter Frequency Response
Amplitude[dB]
0
-5
-10
-15
-20
-25
5.5
6
6.5
7
7.5
8
8.5
9
9.5
Frequency[Hz]
×10^8

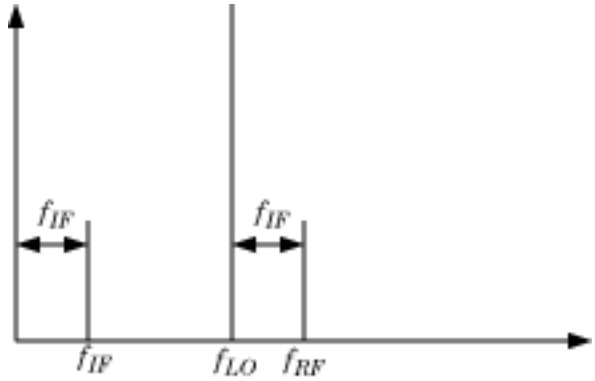
Frequency Conversion

Model an RF Mixer

In this section...

This example shows how to:

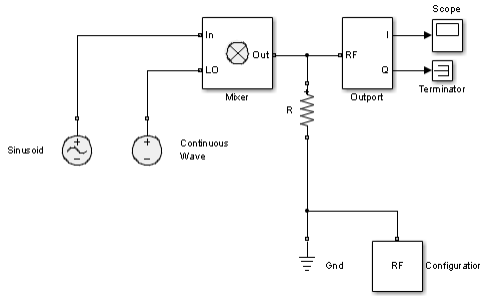
- Use a Sinusoid block to model the envelope of an amplitude-modulated (AM) waveform.
- Use a Continuous Wave block to model the constant envelope of an ideal local oscillator (LO).
- Use a Mixer block to downconvert the AM waveform to an intermediate frequency (IF).



The preceding figure illustrates frequency conversion. A signal modulated on the carrier f_{RF} mixes with a local oscillator at f_{LO}. The operation downconverts f_{RF} to $f_{IF} = f_{RF} - f_{LO}$. The upper mixing product $f_{RF} + f_{LO}$ is not modeled.

Build RF System

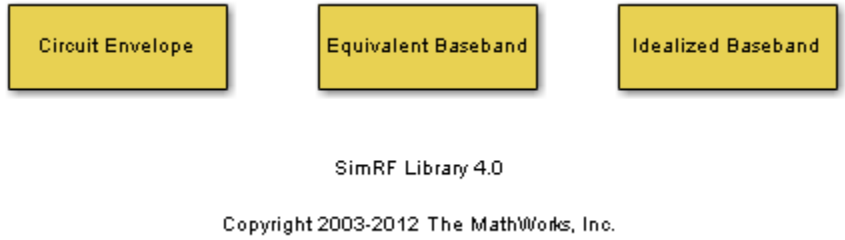
This example shows how to build the following SimRF model from a blank canvas.



To skip this section and start with the completed model, at MATLAB command line, enter:

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

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-Ω termination at the output of a mixer. To build this model, open the SimRF 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 block.
- From the Elements library, add a Mixer block.
- From the Elements library, add a Ground block.
- From the Sources library, add a Continuous Wave block.
- From the Sources library, add a Sinusoid block.
- From the Utilities library, add a Configuration block.

Connect the blocks in the same configuration as the `ex_simrf_tut_mixer` model.

Define Model Variables

For models with blocks that share parameter values, specifying parameters values using variables saves time and effort. Most models in the *SimRF User's Guide* use the `InitFcn` to define model variables.

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. Open the Model Properties dialog box by selecting **File** > **Model Properties** > **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:

- Configure the SimRF 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:

1. Select **Simulation** > **Model Configuration Parameters** to open the Configuration Parameters dialog box. 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 SimRF 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 SimRF blocks.
2. 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`.
3. 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`.
4. 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 SimRF Utilities library, drag and drop an `Outport` block onto your model. In the block dialog:
  - Set **Output** to `In-phase and quadrature`.

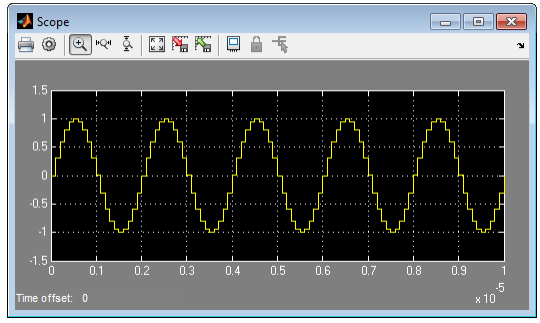
  - Set **Carrier frequencies** to `RFCarrier - LOFrequency`.
- From the Simulink Commonly Used Blocks library, drag and drop a `Scope` block and a `Terminator` block onto your model.

This configuration uses a SimRF 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" on page 2-4 section of this tutorial.

Observe Downconverted Envelope Signal at Output Port

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

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



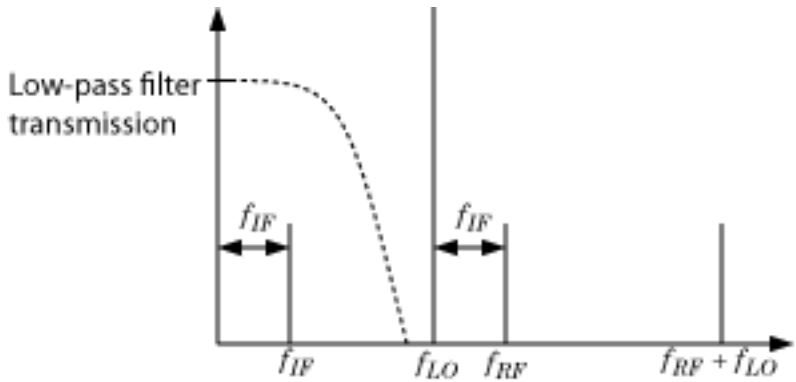
The Sinusoid specifies a 1-V amplitude for the modulation on f_{RF}, which the mixer downconverts to f_{IF}. The SimRF Outport block probes the intermediate frequency and recovers the 1-V modulation amplitude. This value agrees with a specified conversion gain of 0 dB in the mixer.

Filter Mixing Products

In this section...

This section of the tutorial shows you how to:

- Configure a SimRF Outport block to probe multiple carrier frequencies simultaneously.
- Model an analog filter in the SimRF environment using SimRF Capacitor and Inductor blocks.



The preceding figure illustrates low-pass filtering of a low-side injection system. Mixing f_{RF} and f_{LO} produces signals on the carriers $f_{IF} = f_{RF} - f_{LO}$ and $f_{RF} + f_{LO}$. Adding a low-pass filter to the model reduces the power present in the high-frequency signal.

To begin, open the model that you created in the “Model an RF Mixer” on page 2-2 section, or at MATLAB command line, enter:

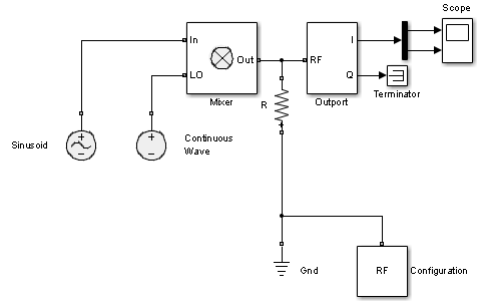
```
addpath(fullfile(docroot,'toolbox','simrf','examples'))
ex_simrf_tut_mixer
```

at the Command Window prompt.

Probe Multiple RF Carriers

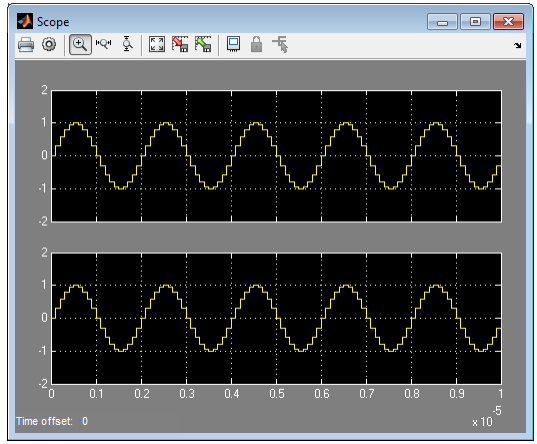
The SimRF environment specifies the carriers $f_{RF} - f_{LO}$, f_{RF}, f_{LO}, and $f_{RF} + f_{LO}$, but the SimRF Outport block probes only the carrier $f_{RF} - f_{LO}$. Examine the carriers $f_{RF} - f_{LO}$ and $f_{RF} + f_{LO}$ by changing the model according to the following workflow:

1. From the Simulink Commonly Used Blocks library, add a `Demux` block to your model.
2. In the Scope block dialog box, click the **Parameters** button, then set **Number of Axes** to `2`.
3. In the Outport block dialog box, set **Carrier frequencies** to `[RFCarrier - LOFrequency, RFCarrier + LOFrequency]`
4. Connect the blocks as shown in the following figure.



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

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



The first set of axes displays the modulation of specified carrier $f_{IF} = f_{RF} - f_{LO}$. 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 $f_{RF} + f_{LO}$. The modulation of the upper mixing product has the same amplitude as the modulation of the downconverted signal.

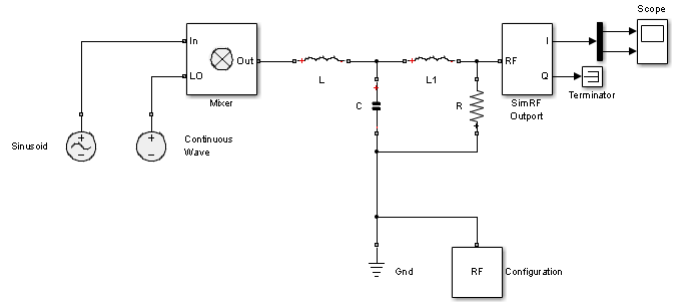
Model an RF Filter

To begin, open the model that you created in the "Probe Multiple RF Carriers" on page 2-8 section, or at MATLAB command line, enter:

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

at the MATLAB Command Window prompt.

1. From the Elements library, drag and drop one `Capacitor` and two `Inductor` blocks onto your model
2. Set the **Capacitance** parameter of the capacitor to `40e-12 F`.
3. Set the **Inductance** parameter of both inductors to `50e-9 H`.
4. Connect the blocks as shown in the following figure.



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.

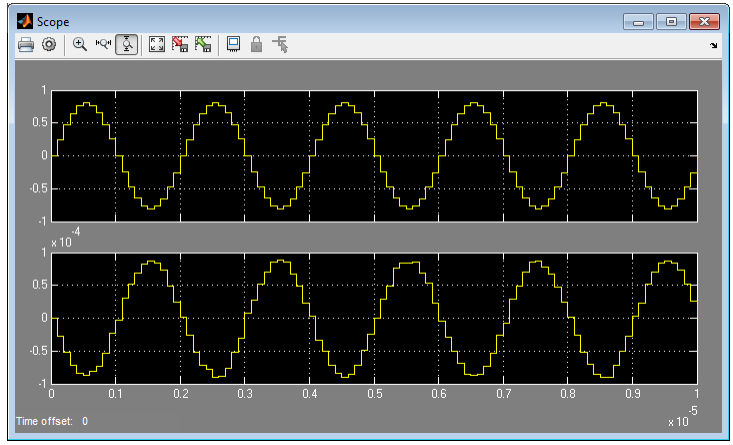
Simulate Filtering of RF Signals

To begin, open the model that you created in the previous section, “Model an RF Filter” on page 2-9, or at MATLAB command line, enter:

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

at the MATLAB Command Window prompt. Select **Simulation** > **Run** to run the model.

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



The filter attenuates the high-frequency signal at $f_{RF} + f_{LO}$ and transmits the signal at $f_{RF} - f_{LO}$ with minimal loss.

The SimRF documentation contains additional RF filter analysis and simulation examples like, the example“AC Analysis of an RF System” calculates the voltage transfer function of an RF filter using harmonic balance analysis. such as

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 SimRF simulation frequencies. The performance of the `ex_simrf_tut_filter` 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.

At the MATLAB command line, enter:

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

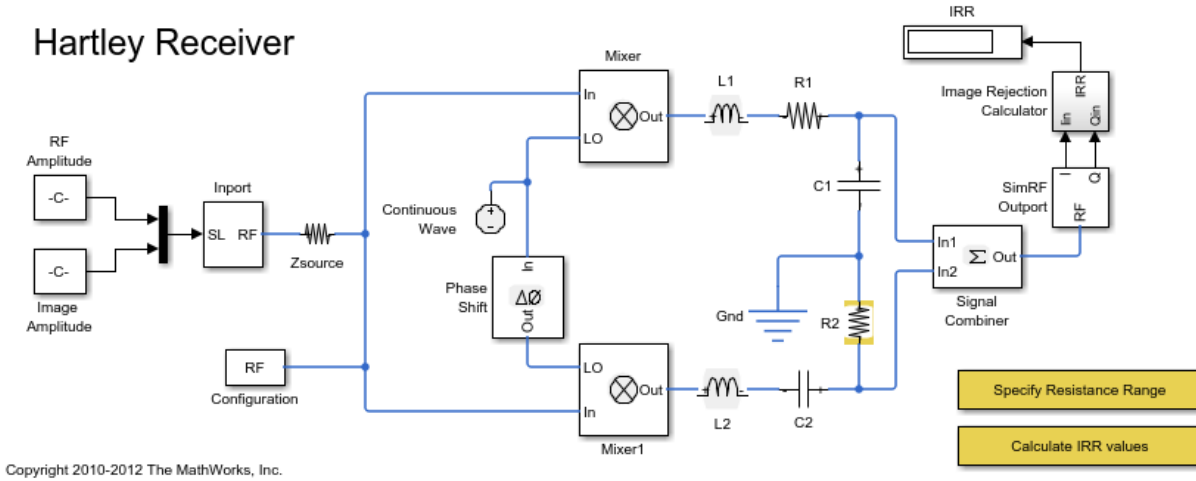
2. Double-click the Configuration block to open the block dialog box. Note that the block dialog displays a **Total simulation frequencies** value of 121. This value indicates that the environment is running 121 separate simulations.
3. Clear the **Automatically select fundamental tones and harmonic 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 f_{RF} and f_{LO} 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 $f_{RF} + f_{LO}$ and $f_{RF} - f_{LO}$ 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.

Measuring Image Rejection Ratio in Receivers

This example shows how to use the SimRF™ Circuit Envelope library to calculate the image rejection ratio (IRR) for high-side-injection in Weaver and Hartley receivers. The Weaver receiver shows the effect of phase offset on IRR, and the Hartley receiver shows a similar effect for resistor variation.

```
model1 = 'simrfV2_hartley';
open_system(model1);
```



System Architecture

The RF system consists of:

- An Inport block that assigns multiplexed outputs of the RF and Image, by using Simulink® Constant blocks, to the carriers `fc_RF` and `fc_IM` respectively. Real and imaginary values of the Constant blocks are matched to in-phase and quadrature carrier components.
- First stage mixers that mix the input signal with a local oscillator modeled by a Continuous Wave block with frequency `fc_LO`. The LO frequency is the average of the RF and image frequencies, so both signals are mixed down to the same frequency, `fc_IF`. The LO phase is shifted 90 degrees in one mixer relative to the other.

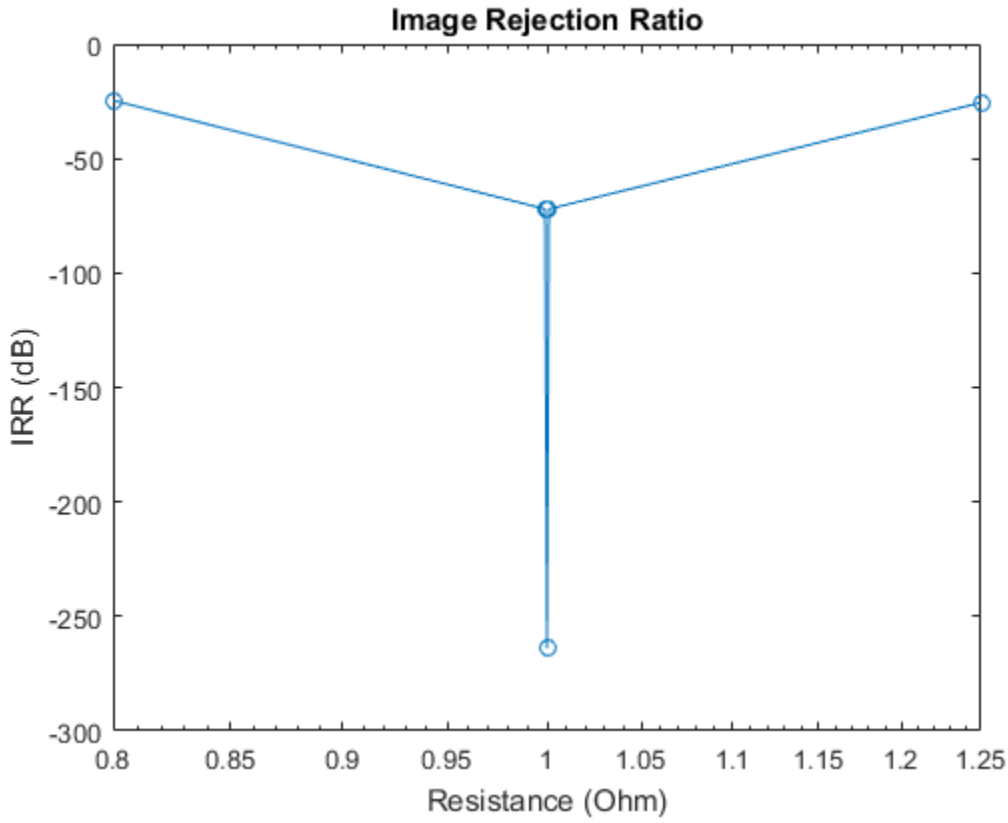
- The second stage of the Hartley uses a frequency independent RC-CR network to produce an additional 90 degree phase shift between the two signal paths, while the Weaver employs two additional mixers for channel selection.
- A Signal Combiner block that sums the voltage signals at its two inputs to yield the RF signal. If the Signal Combiner block is used to perform subtraction, the image can be obtained instead of the RF signal at its output. For low-side-injection, the Signal Combiner block needs to perform subtraction.
- The values of the in-phase and quadrature components of the RF and image signals are chosen to reduce the number of IRR calculations and facilitate reuse of the Image Rejection Calculator.

Simulating the Hartley Receiver

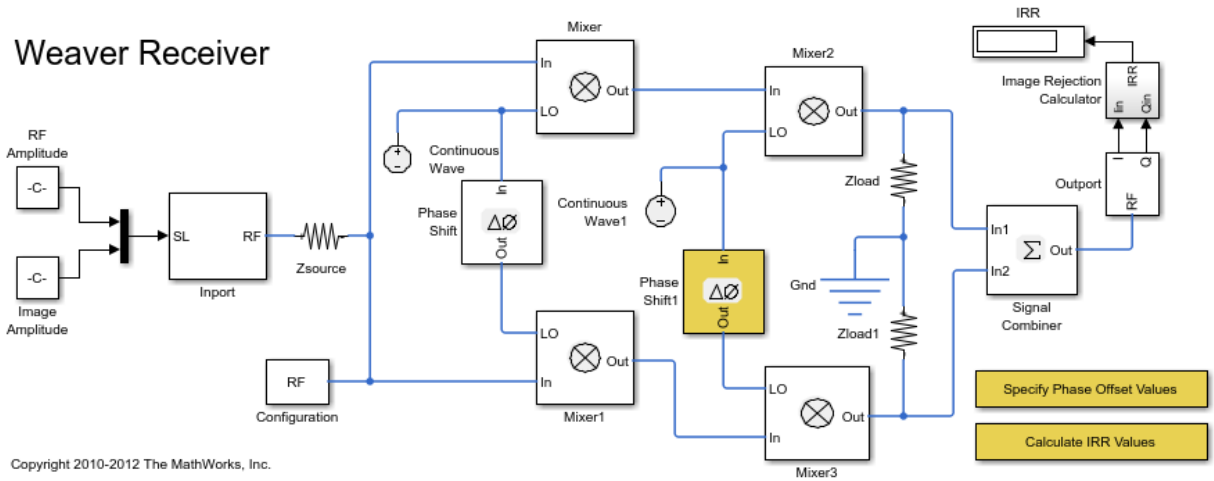
1. Type `open_system('simrfV2_hartley')` at the Command Window prompt.
2. Double-click 'Specify Resistance Range' and specify a set of resistance values for the highlighted resistor.
3. Double-click 'Calculate IRR values' to execute a script, `simrfV2_hartley_callback`, that simulates the model once for each specified resistance value and generates a plot.

The sensitivity of the architecture to the component variation is shown by simulating the system multiple times, varying the resistance of the highlighted Resistor block at each iteration. When the highlighted resistor has a resistance of 1 Ohm, the images sum to zero in the Signal Combiner block and the IRR is minus infinity.

```
evalc('simrfV2_hartley_callback');
```




```
bdclose(model1);
model2 = 'simrfV2_weaver';
open_system(model2);
```

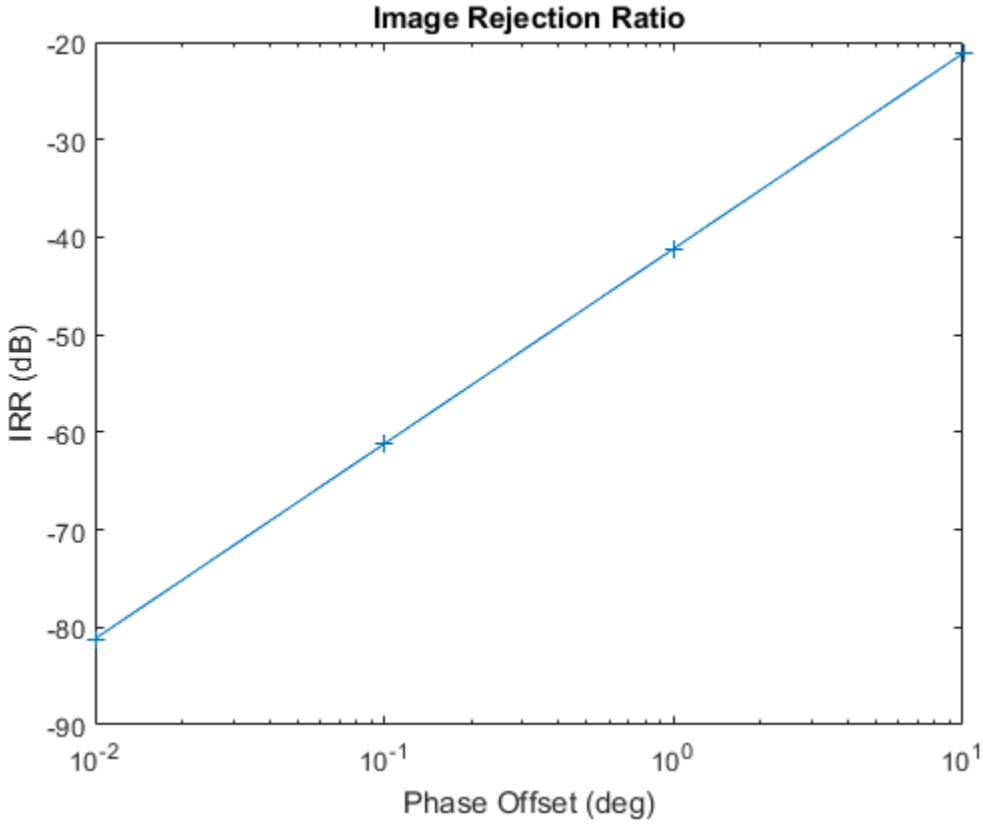


Simulating the Weaver Receiver

1. Type `open_system('simrfV2_weaver')` at the Command Window prompt.
2. Double-click 'Specify Phase Offset Values' and specify a set of phase offset values.
3. Double-click 'Calculate IRR values' to execute a script, `simrfV2_weaver_callback`, that simulates the model once for each specified offset and generates a plot.

The sensitivity of the architecture to LO phase offset is shown by simulating the system multiple times, varying the phase offset of the highlighted Phase Shift block at each iteration. When the phase offset of the highlighted Phase Shift block is zero, the images sum to zero in the Signal Combiner block and the IRR is minus infinity.

```
evalc('simrfV2_weaver_callback');
```




```
bdclose(model2)
```

Equivalent Baseband

3

Introduction to Equivalent Baseband Simulation

SimRF Equivalent Baseband Libraries

In this section...

Overview of SimRF Equivalent Baseband Libraries

The SimRF 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.

SimRF 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 SimRF 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.

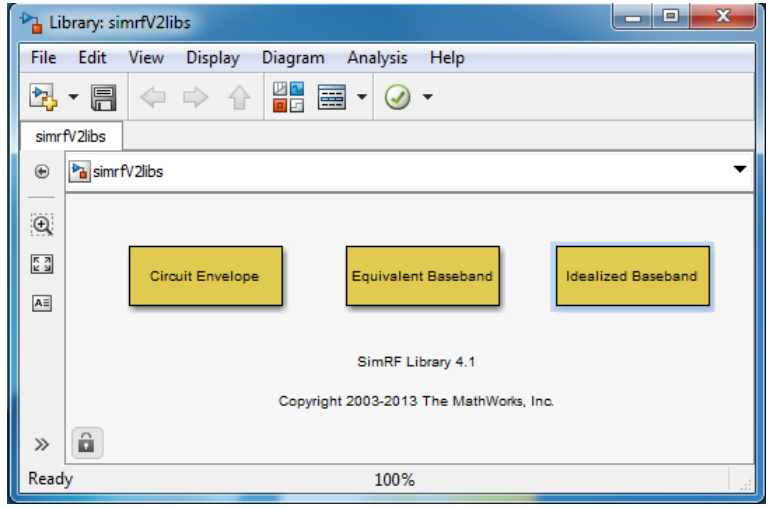
You can also use the blockset with Simulink Coder™ software to generate embeddable C code for real-time execution.

Open SimRF 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.



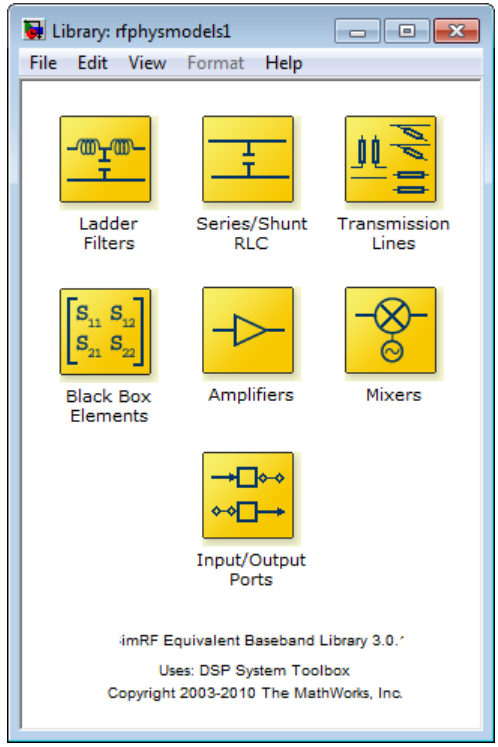
For a discussion of the Equivalent and Idealized Baseband libraries, see the following sections.

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

- Double-click the Demos icon to open the SimRF Equivalent Baseband examples.
- Double-click the Info icon to open the SimRF 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,

Sublibrary	Description
	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.
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. **Note:** 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.

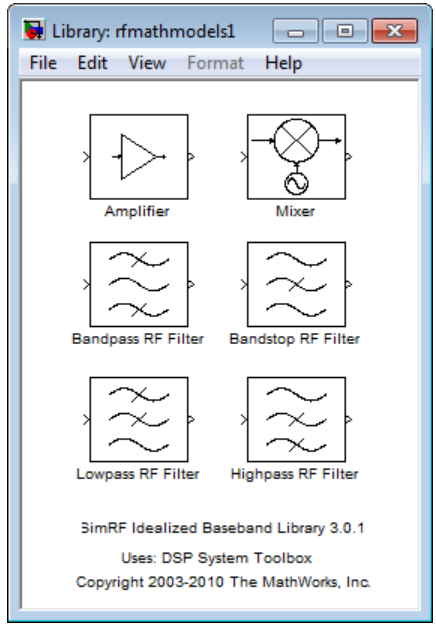
For more information on defining components, see “Specify or Import Component Data”.

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 SimRF 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

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 SimRF Equivalent Baseband physical modeling environment.
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.

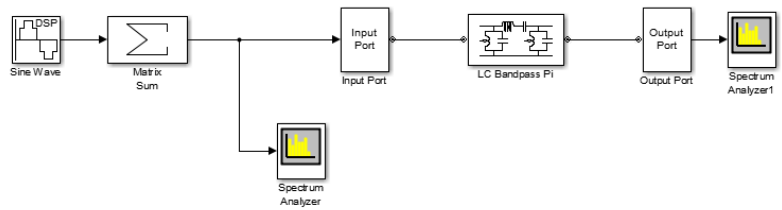
1 Create a model with the blocks shown in the following table. The Library column of the table specifies the hierarchical path to each block.

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	**SimRF > Equivalent Baseband > Equivalent Baseband > Input/Output Ports**	1
LC Bandpass Pi	**SimRF > Equivalent Baseband > Ladder Filters**	1

Block	Library Path	Quantity
Output Port	**SimRF > 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 3-11
- "Filter Subsystem Parameters" on page 3-14
- "Signal Display Parameters" on page 3-18

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 SimRF 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.

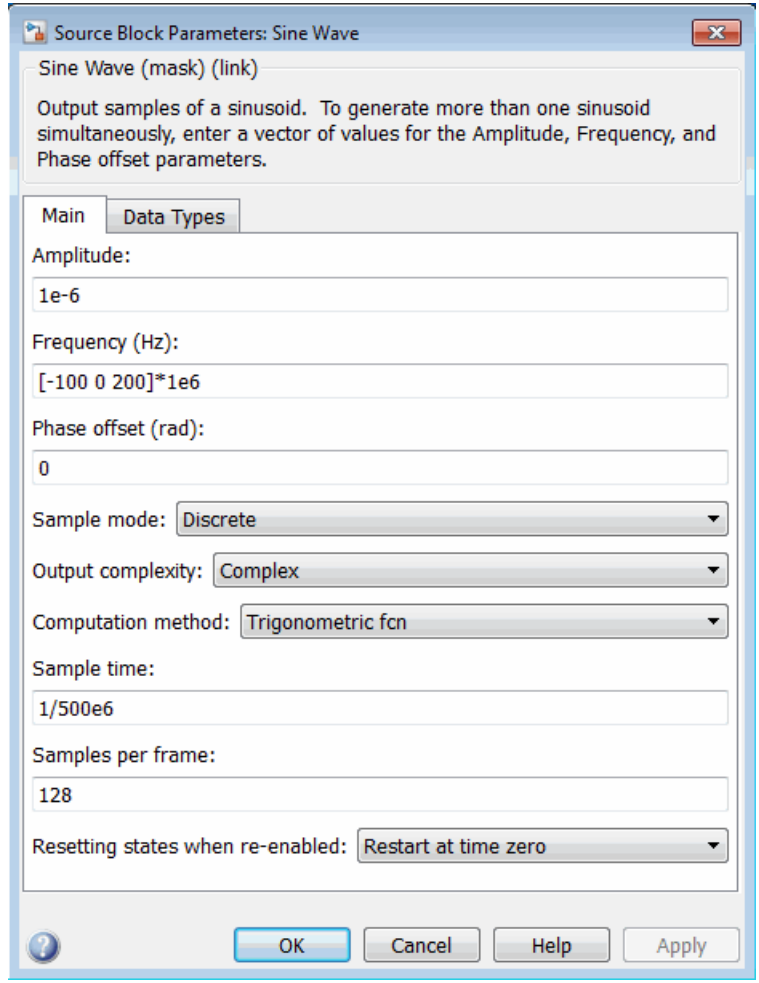
For more information on complex-baseband modeling, see "Create a Complex Baseband-Equivalent Model".

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.

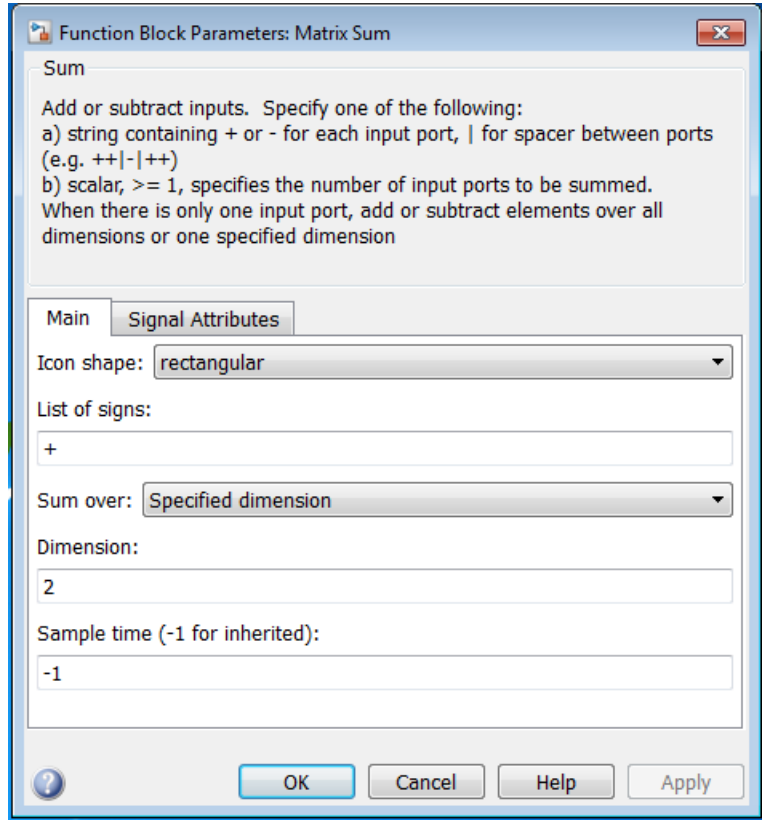
1 In Sine Wave block dialog box:

- 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` .



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.

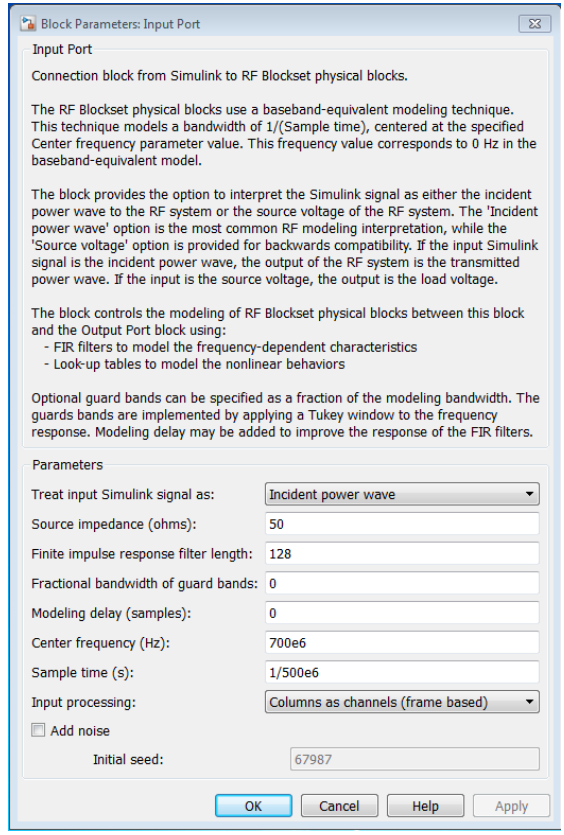
1 Set the Input Port block parameters as follows:

- **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".

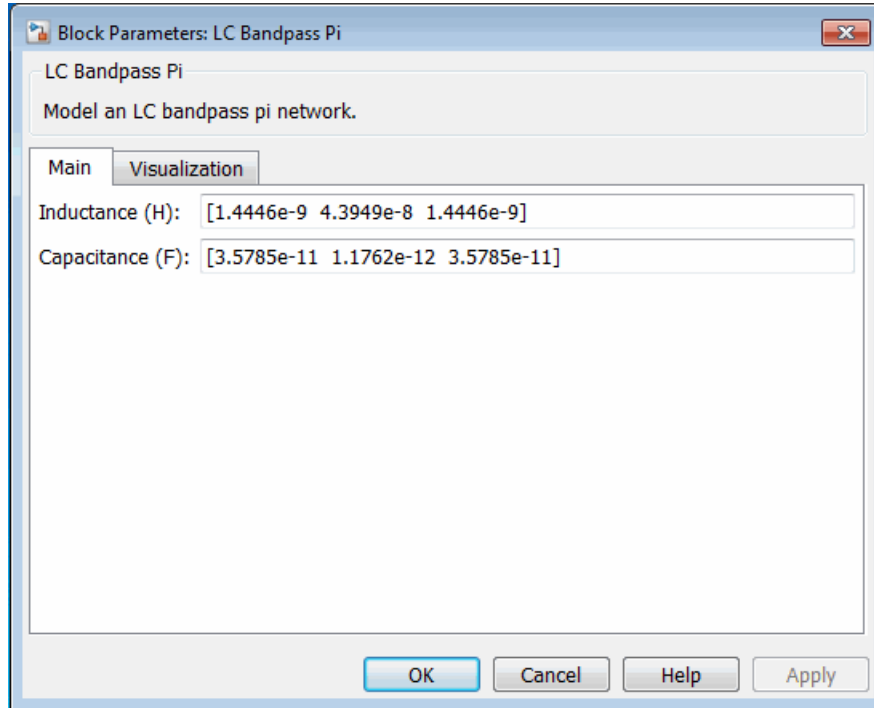
- **Center frequency** = `700e6`
- **Sample time (s)** = `1/500e6`
- **Input Processing** = `Columns as channels (frame based)`
- Clear the **Add noise** check box so the software does not include noise in the simulation. To learn how to model noise, see "Model Noise".



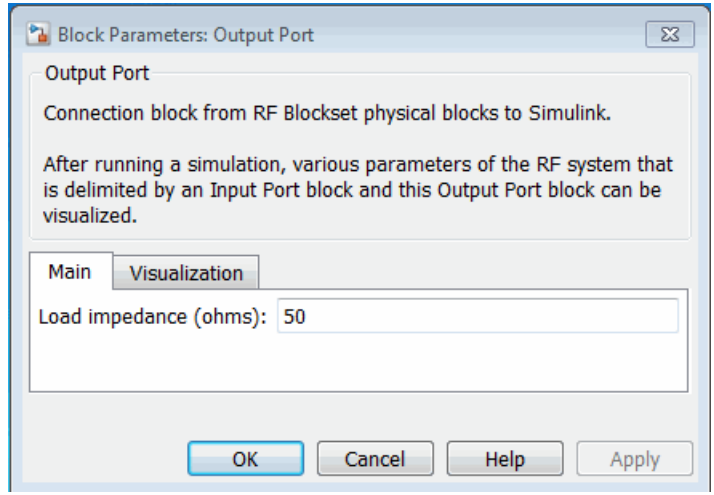
Note: You must enter the **Sample time (s)** because the Input Port block does not inherit a sample time from the input signal. The specified sample time must match the sample time of the input signal. The **Sample time (s)** of `1/500e6` second used in this example is equivalent to a bandwidth of 500 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:

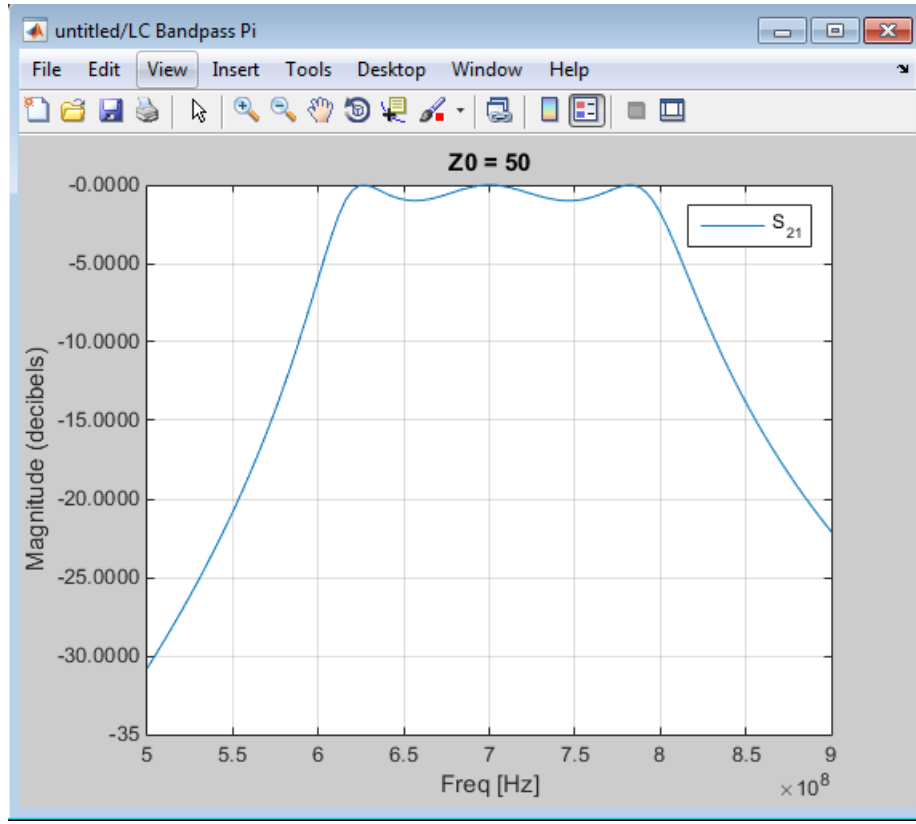
1. Set the Spectrum Analyzer parameters as follows:
   - In the **View** tab check **Spectrum Settings**. In `Trace options` set the **Units** parameter to `dBm`.
   - In the **View** tab open **Configuration Properties**. Set the **Minimum Y-limit** parameter to `-291` and the **Maximum Y-limit** parameter 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 3-11.

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, select **Simulation > Start** 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

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 a Complex Baseband-Equivalent Model".

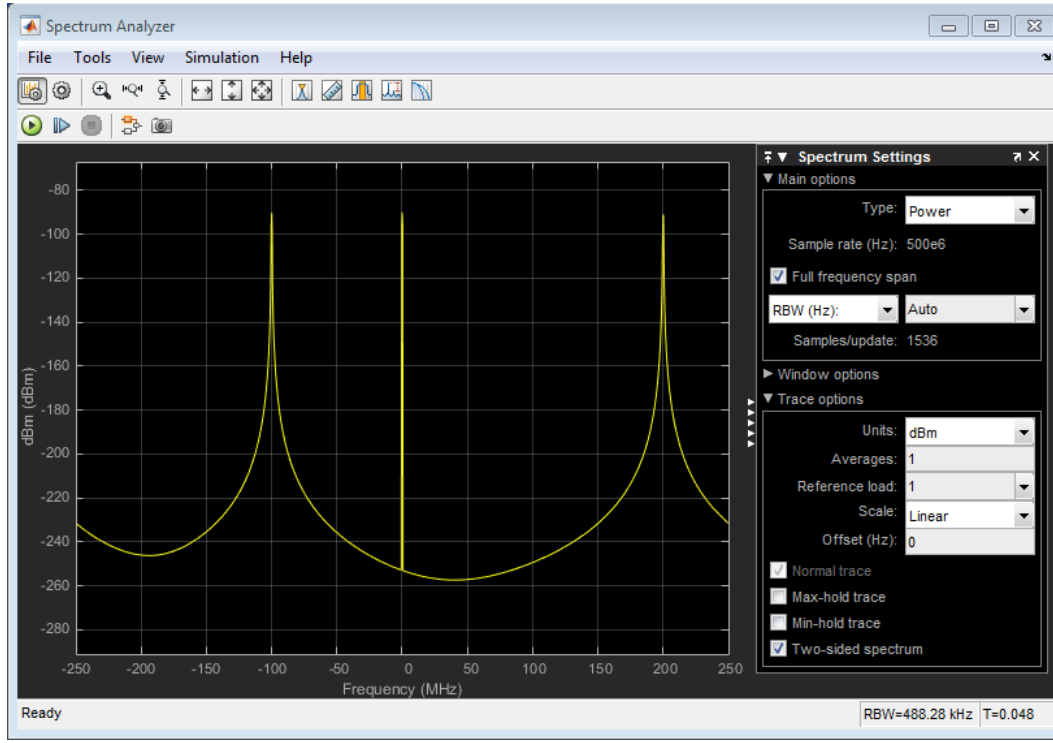
The `Spectrum Analyzer` blocks display power spectral density normalized to unit sampling frequency. To display power per channel, insert a Gain block with the **Gain** parameter set to `1/sqrt(N)` before each `Spectrum Analyzer` block. `N` is the number of channels. The Gain block is in the **Simulink > 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: SimRF Equivalent Baseband signals represent amplitudes, not voltages. This means that in the product, power is defined as:

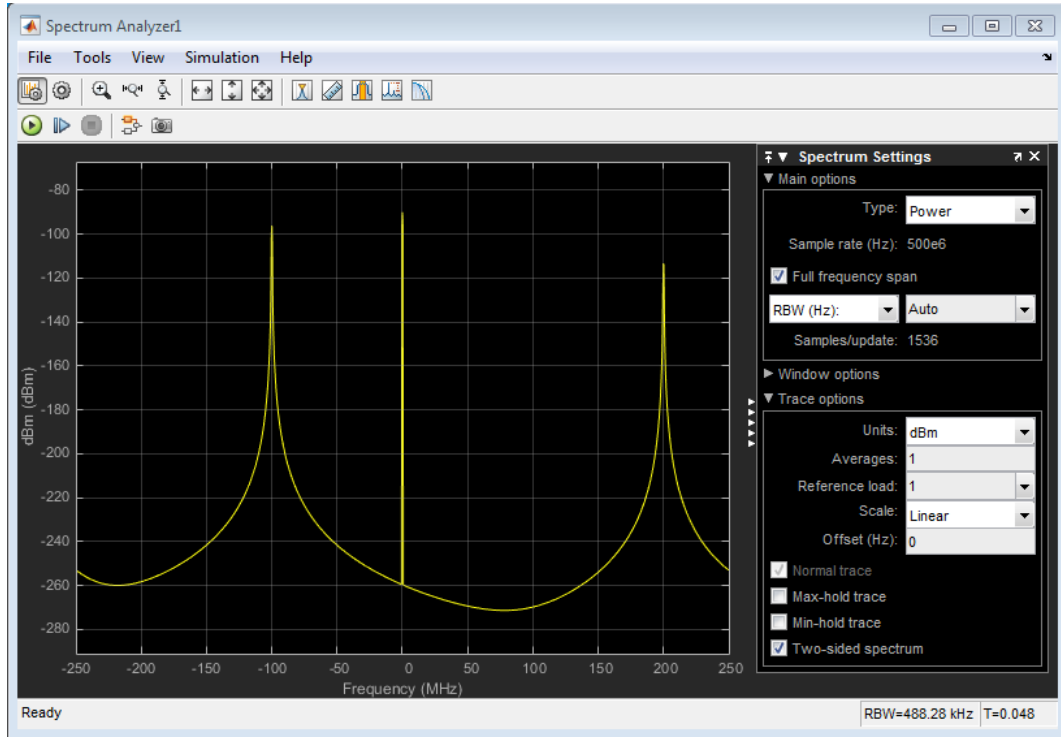
$$Power\,(in\,watts) = \left[Amplitude\,(in\,volts\,/\,sqrt(ohm))\right]^2$$

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.