RF Toolbox™ Release Notes

MATLAB®

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

RF Toolbox™ Release Notes

Contents

R2021a

R2020b

R2020a

R2019b

R2019a

R2018b

R2018a

R2017b

R2017a

R2016b

No New Features or Changes

R2016a

R2015b

R2015a

R2014b

R2014a

R2013b

R2013a

R2012b

R2012a

R2011b

R2011a

R2010b

R2010a

R2009b

R2009a

R2008b

R2008a

R2007b

R2007a

R2006b

R2006a

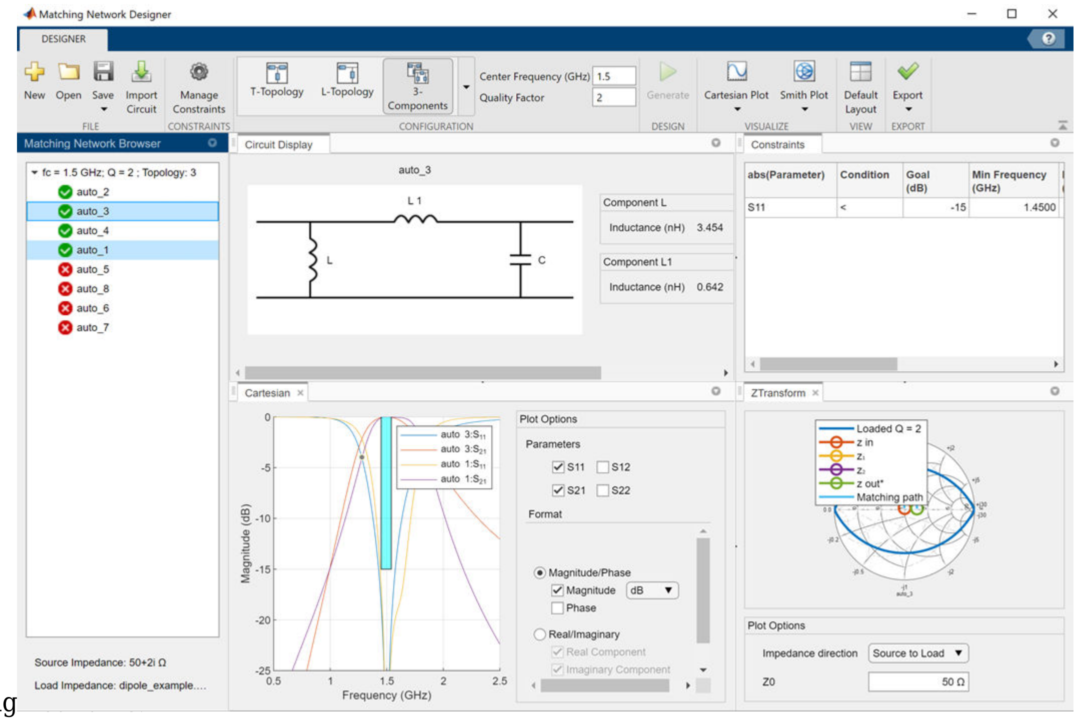
R14SP3

R14SP2

R2021a

Matching Network Designer App: Design, visualize, and compare matching networks for one-port load

Use the **Matching Network Designer** app to design, visualize, and compare matching networks for one-port load. You can also use this app to design two- and three-component lumped element



matching

RF System Object: Create System object for circuit envelope simulation in MATLAB

Use the `rfsystem` System object™ to perform a circuit envelope simulation of an RF system designed using an `rfbudget` object.

To use this System object, you must have an RF Blockset™ license.

Piecewise Linear Response: Calculate time responses of rational or rfmodel.rational objects to piecewise linear input signal

Use the `pwlresp` function to calculate time responses of a `rational` or an `rfmodel.rational` object to a piecewise linear input signal.

Relative Error Metrics and Noise Floor: Improve quality of rational fitting with relative error metrics and noise floor specification

Set the `'ErrorMetric'` property to `'Relative'` in a `rational` object to perform rational fitting to fit both peaks and valleys, and use the `'NoiseFloor'` property to ignore low-level noise in the data of the fitter.

Series and Shunt RLC Elements: Design series and shunt RLC networks using circuit objects or RF Budget Analyzer app

Use the `seriesRLC` and `shuntRLC` circuit objects to design series and shunt RLC networks at the command line. You can also add series and shunt RLC elements in the RF system using the **RF Budget Analyzer** app and export the circuits to RF Blockset.

Attenuator Element: Design attenuator with thermal noise using circuit objects or RF Budget Analyzer app

Use the `attenuator` object to design an attenuator element at the command line. You can also add an attenuator element in the RF system using the **RF Budget Analyzer** app and export the circuits to RF Blockset.

RF Antenna Element: Design transmit antenna using circuit objects or RF Budget Analyzer app

Use the `rfantenna` object to design an RF antenna element at the command line. You can also import your antenna design from the **Antenna Designer** app in the Antenna Toolbox™. You can also add an RF antenna element in the system designed in the **RF Budget Analyzer** app to design an antenna element modeled as a transmitter.

To import your antenna design from the Antenna Designer app, you must have an Antenna Toolbox license.

Transmission Line Elements: Model equation-based lossless and lossy transmission line elements using circuit objects or RF Budget Analyzer app

Use these circuit objects to model different types of transmission lines at the command line.

- `txlineEquationBased` — Equation-based transmission line
- `txlineDelayLossless` — Transmission line without time delay
- `txlineDelayLossy` — Transmission line with time delay

You can also add your transmission line elements to the RF system created in the **RF Budget Analyzer** app to design cascaded systems.

Force Overwrite: Suppress warning message when overwriting existing files with rfwrite function

If you want to suppress the warning message while overwriting the files using the `“filename”` argument, set the “'ForceOverwrite'” name-value argument in the `rfwrite` function to `true`.

Extract S-Parameters Examples: Extract S-parameters from circuits or mutual inductors

This release introduces two examples that show how to extract S-parameters using the Symbolic Math Toolbox™.

- The "Extract S-Parameters from Circuit" example shows how to derive analytical expressions for S-parameters of circuits using Symbolic Math Toolbox™.
- The "Extract S-Parameters from Mutual Inductor" example shows how to build user-defined element or a mutual inductor from S-parameters and add it to an `rfbudget` object for link budget analysis using the Symbolic Math Toolbox.

To run both the examples, you must have a Symbolic Math Toolbox license.

R2020b

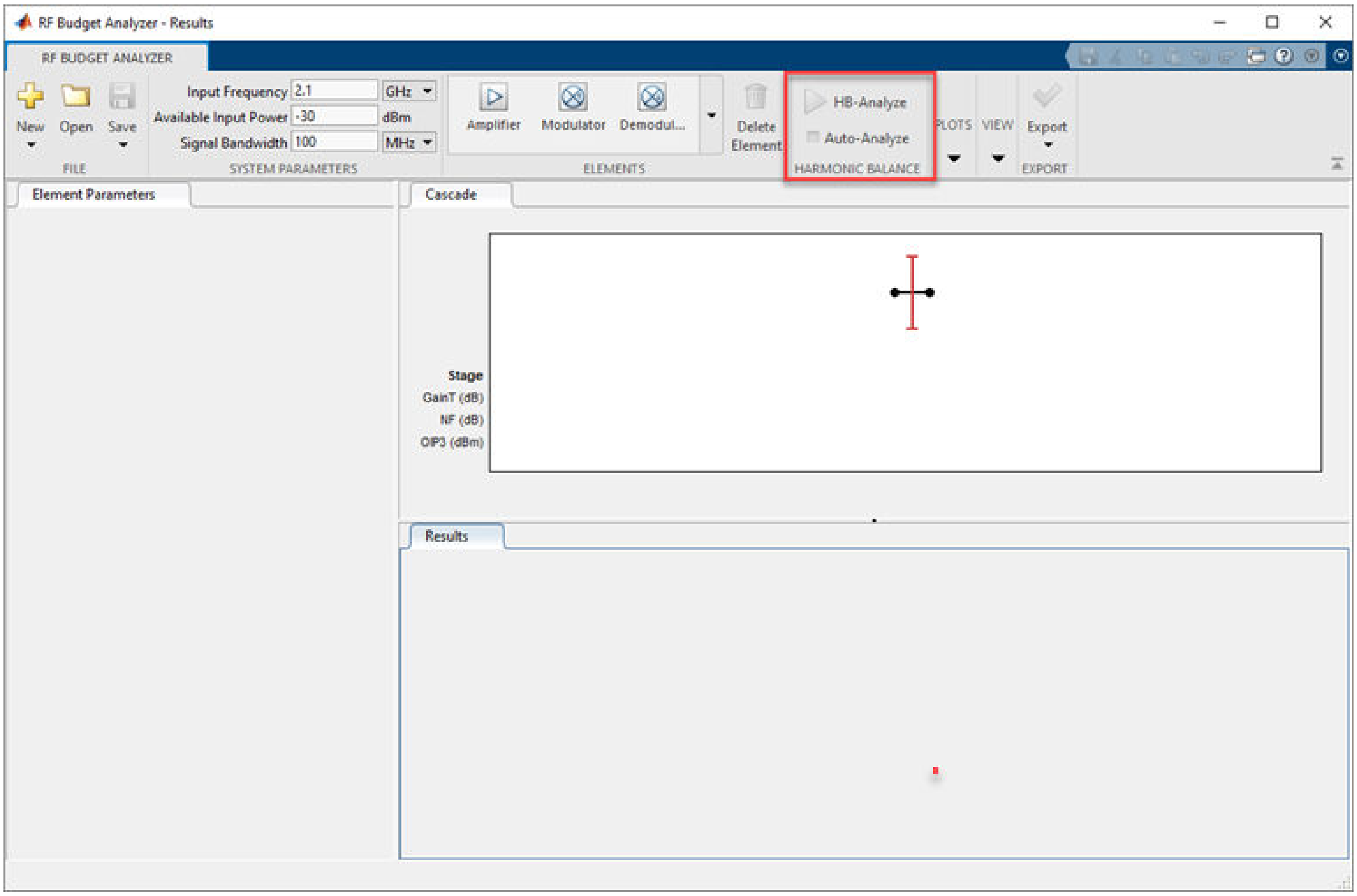
Version: 4.0

New Features

Compatibility Considerations

Harmonic Balance in RF Budget Analyzer App: Compute output power, IP2, NF, and SNR with nonlinear analysis

Use the **RF Budget Analyzer** to compute nonlinear effects using harmonic balance analysis with the **HB analyze** button.

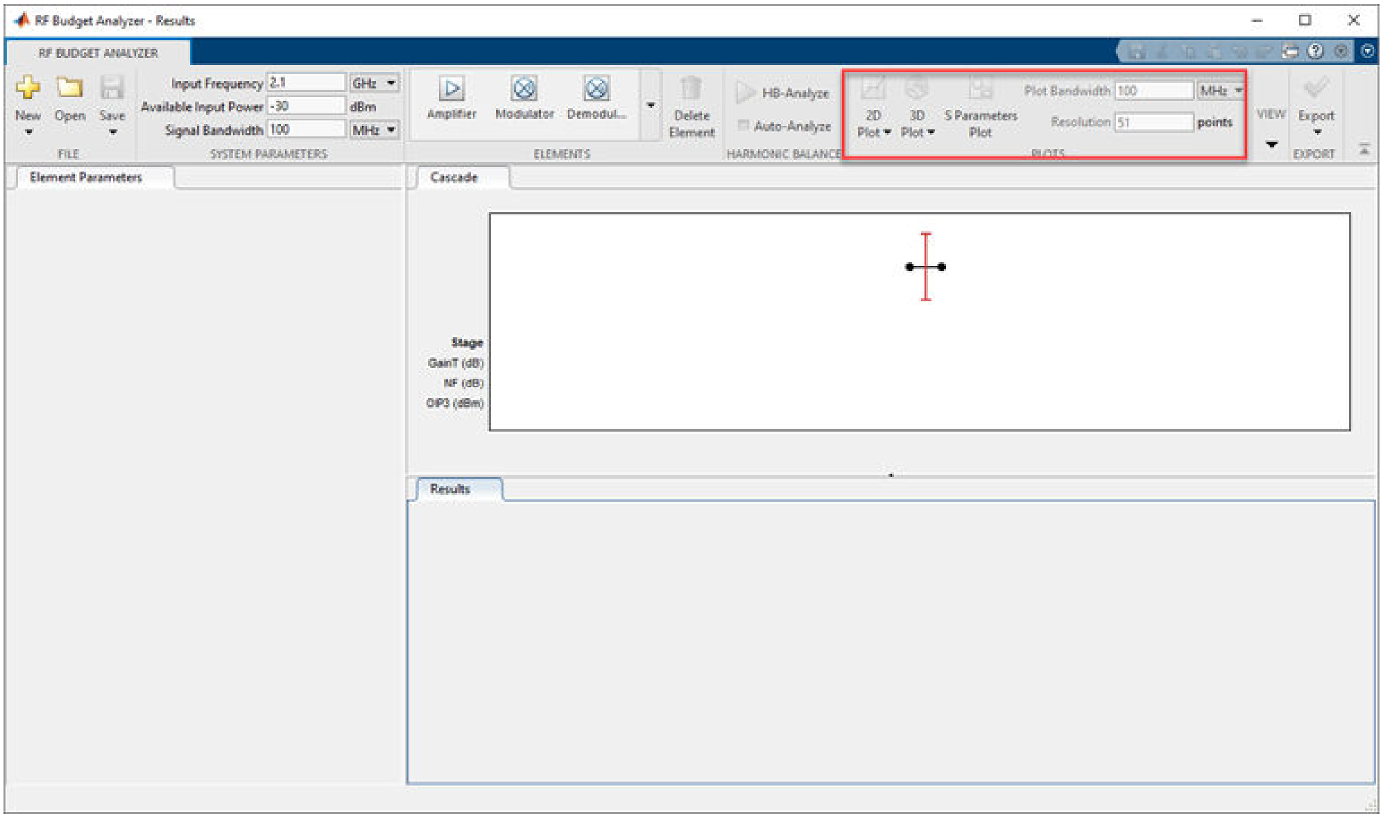


Plots in RF Budget Analyzer App: Visualize budget results and S-parameters over stages and frequencies

Use the updated plots tab in the **RF Budget Analyzer** to visualize:

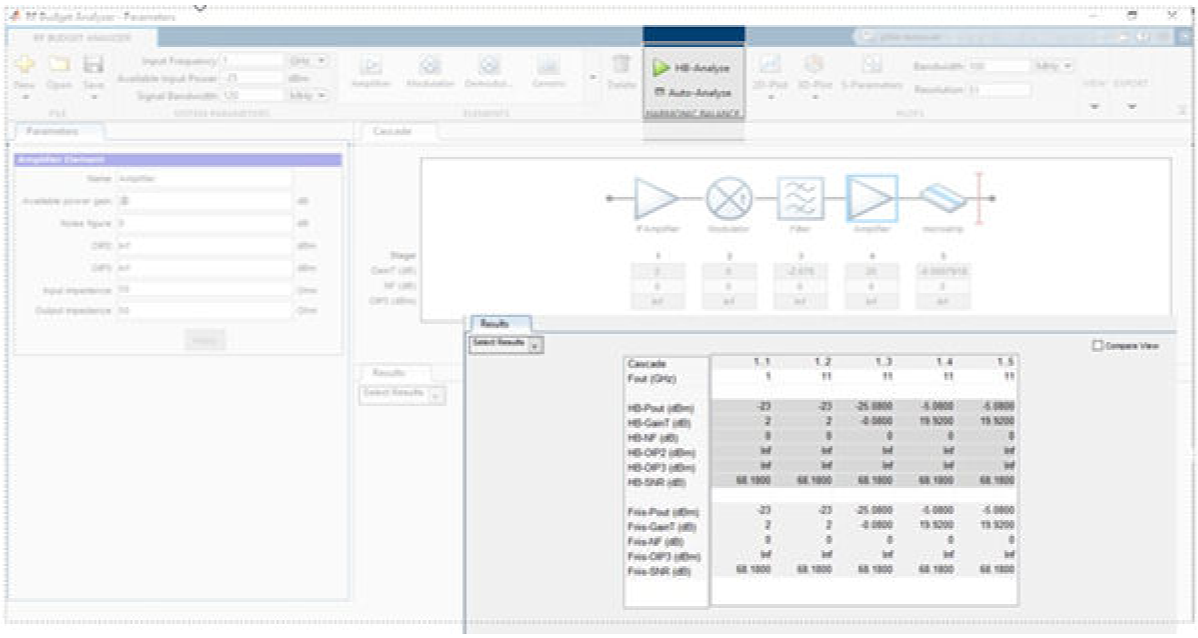
- 2-D and 3-D plots.
- S-parameter plots
- Frequency range and resolution field

Plots open docked tabs in the **Results** section of the app window. To restore the default app window layout, use the **Default Layout** button in the **View** tab.



Comparison Reports in RF Budget Analyzer App: Compare Friis and Harmonic Balance budget results

Use the updated **RF Budget Analyzer** app to compare Friis and harmonic balance analysis using the **Results** section of the app window.

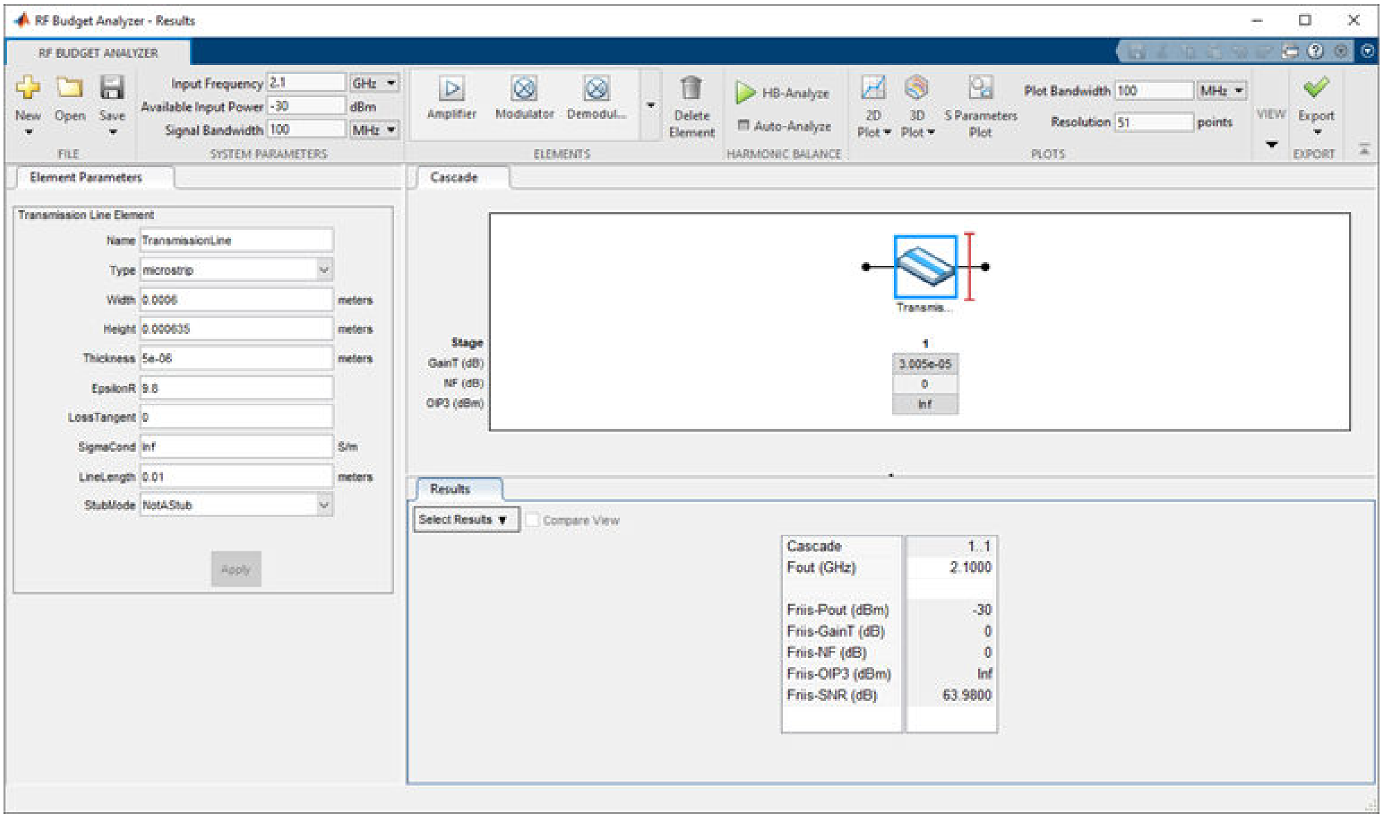


Transmission Line Elements: Model distributed elements in circuit objects and RF Budget Analyzer app

Use the following objects to create different types of transmission lines at the command line:

- `txlineMicrostrip` — Microstrip transmission line
- `txlineCoaxial` — Coaxial transmission line
- `txlineCPW` — Coplanar waveguide transmission line
- `txlineParallelPlate` — Parallel plate transmission line
- `txlineTwoWire` — Two wire transmission line
- `txlineRLCGLine` — RLCG line transmission line

You can also add transmission lines to systems created in the **RF Budget Analyzer** app.



Functionality being removed or changed

Phase Delay and Group Delay plots not supported in RF Budget Analyzer App

Behavior change

Phase and Group Delay plotting not supported from R2020b in **RF Budget Analyzer** app.

R2020a

Version: 3.8

New Features

New Rational Fitting Object: Fast and accurate time-domain modeling of S-parameters

Use the `rational` function to perform rational fitting to complex frequency-dependent data.

Harmonic Balance in rfbudget Object: Nonlinear analysis of rfbudget objects, including IP2 calculations

You can now use the `rfbudget` object to compute nonlinear effects such as IP2 and output power with saturation.

R2019b

Version: 3.7

New Features

Inverse Chebyshev Filter: Design inverse Chebyshev filters using command line or RF Budget Analyzer app

Use the `rffilter` object to design an inverse Chebyshev filter using the command line. You can also use the RF Budget Analyzer app to design the inverse Chebyshev filter and export the filter to RF Blockset™ or a measurement testbench.

generateSPICE function for rationalfit: Generate SPICE subcircuits for S-parameters rational fitting

Use the `generateSPICE` function to convert rationalfit output of S-parameters into SPICE subcircuits for simulation.

setrfplot function for rfplot: Toggle the scale X-axis of rfplot between scalar and engineering units

Use the `setrfplot` function to toggle the scale of `rfplot` X-axis between scalar units and engineering units (k, M, G, and so on).

R2019a

Version: 3.6

New Features

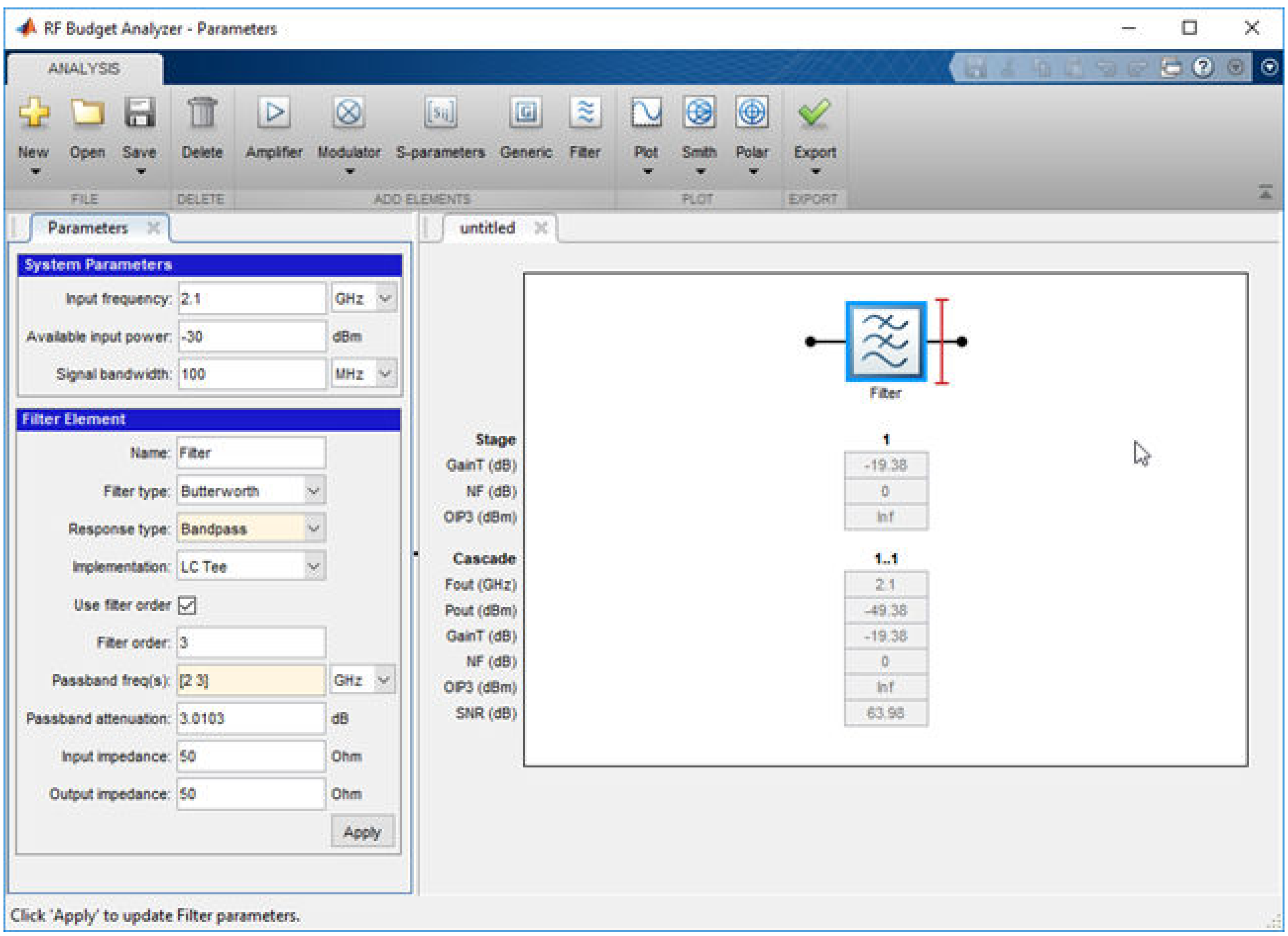
Test, visualize, and enforce passivity of rationalfit output

The `rationalfit` function converts frequency-domain S-parameters into rational functions that can be used for time-domain simulations. These simulations require the fit output to be passive. So, you can use the `ispassive` function to test the passivity of the `rationalfit` output. Use the `passivity` function to visualize the passivity of the `rationalfit` output. Use the `makepassive` function to enforce the passivity of the `rationalfit` output.

RF Filter element in RF Budget Analyzer app

In previous releases, to include an RF Filter element for budget analysis, you had to model it using the `rffilter` object at the command line and export its S-parameters as an `nport` object. You could then import the `nport` object into the app.

In this release, RF filter is now a budget element and can be directly added to the budget chain. You can also export the filter element into the RF Blockset, RF measurement testbench, a MATLAB® script, or the MATLAB workspace.



Design narrowband matching networks for RF circuits

Use the `matchingnetwork` object to design a set of circuits that match the impedance of a given source to the impedance of a given load at the specified center frequency. You can also export the matching network as `circuit` elements.

```
m = matchingnetwork % Default matching network
```

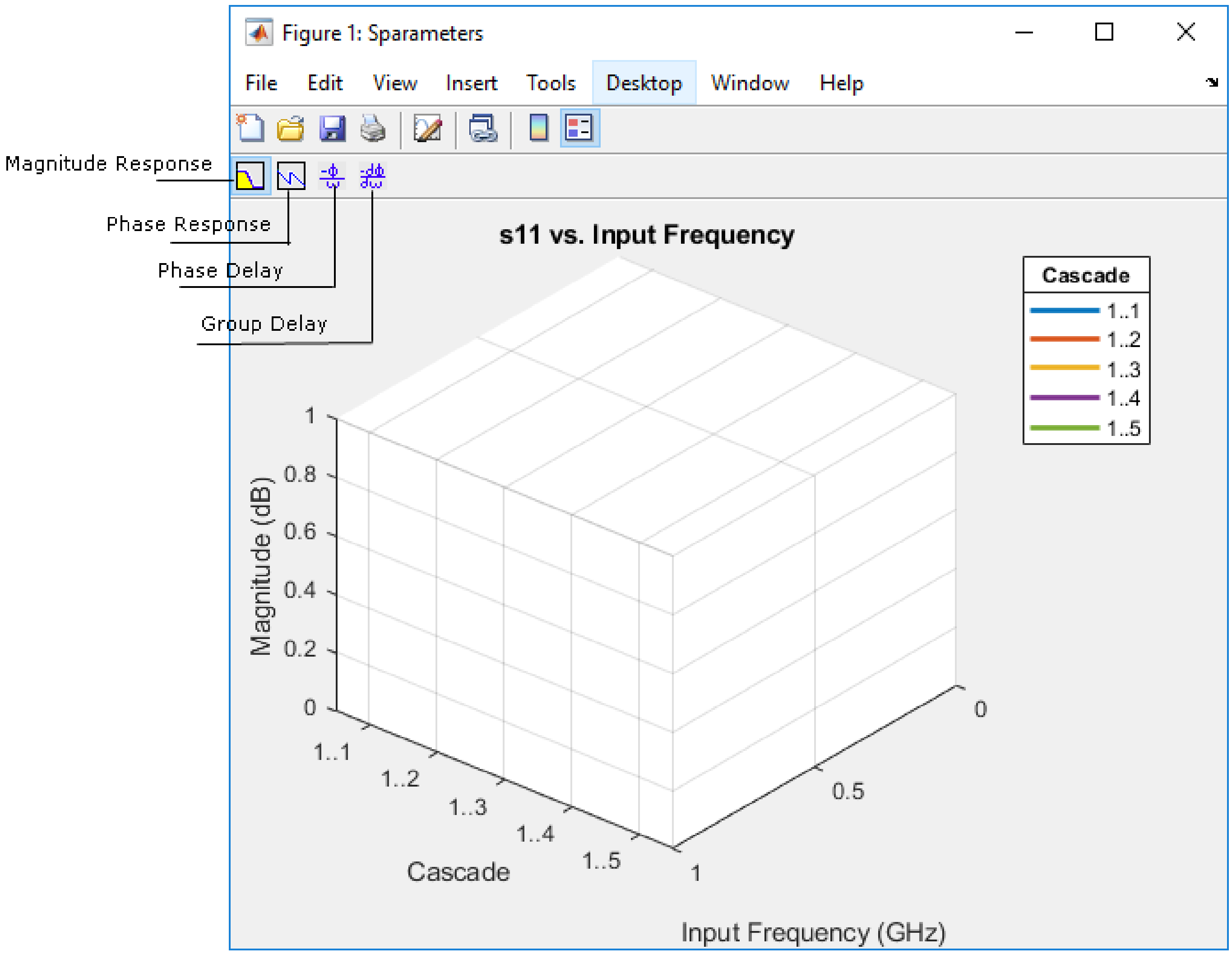
```
matchingnetwork with properties:

       SourceImpedance: 50 Ohms
         LoadImpedance: 50 Ohms
       CenterFrequency: 1 GHz
            Components: 2
               Circuit: [1x2 circuit]
```

R2018b

S-parameters Plotting in RFBudgetAnalyzer App: Plot S-parameter magnitude, phase, and group delay for the stages of a budget object

You can now use the **RF Budget Analyzer** app plot button S-Parameters option to plot the magnitude, phase, and group delay of an RF system.



RF Filter Element: Design a behavioral model of an RF filter and add it to an RF circuit

Use the `rffilter` object to build lowpass, highpass, bandpass or bandstop Butterworth and Chebyshev filters. You can also use the `rffilter` object as a circuit element.

Functionality being removed or changed

smithchart will be removed
Still runs

smithchart will be removed in a future release. Use smithplot instead.

Function Name	What Happens When You Use This Function	Use This Function Instead	Compatibility Considerations
smithchart	Still runs	smithplot	Replace all instances of smithchart function with smithplot function using smithplot.

R2018a

Version: 3.4

New Features

Smith and Polar Plots in RF Budget Analyzer App: Visualize input and output impedances, forward gain budget, and reverse isolation across signal bandwidth

Use the **Smith** and **Polar** buttons in the **RF Budget Analyzer** app to plot output impedances, forward gain budget, RF system S-parameters, and reverse isolation across signal bandwidths.

Improved smithplot Function: Visualize network parameters for rfckt, rfdata, nport, and sparameters objects and support the circle function

Use the smithplot function to visualize network parameters for `rfckt`, `rfdata`, `rfbudget`, and `nport` objects.

R2017b

Version: 3.3

New Features

smithplot Function: Visualize network parameters, data, and markers on a Smith chart

Use the `smithplot` function to visualize data and network parameters with improved markers.

Plot button in RF Budget Analyzer App: Visualize budget results across signal bandwidth

Use the **Plot** button to visualize cumulative RF budget results versus cascade input frequency from within the **RF Budget Analyzer** app.

R2017a

Version: 3.2

New Features

Zero IF Support in rfBudgetAnalyzer App: Analyze direct conversion transmitters and receivers

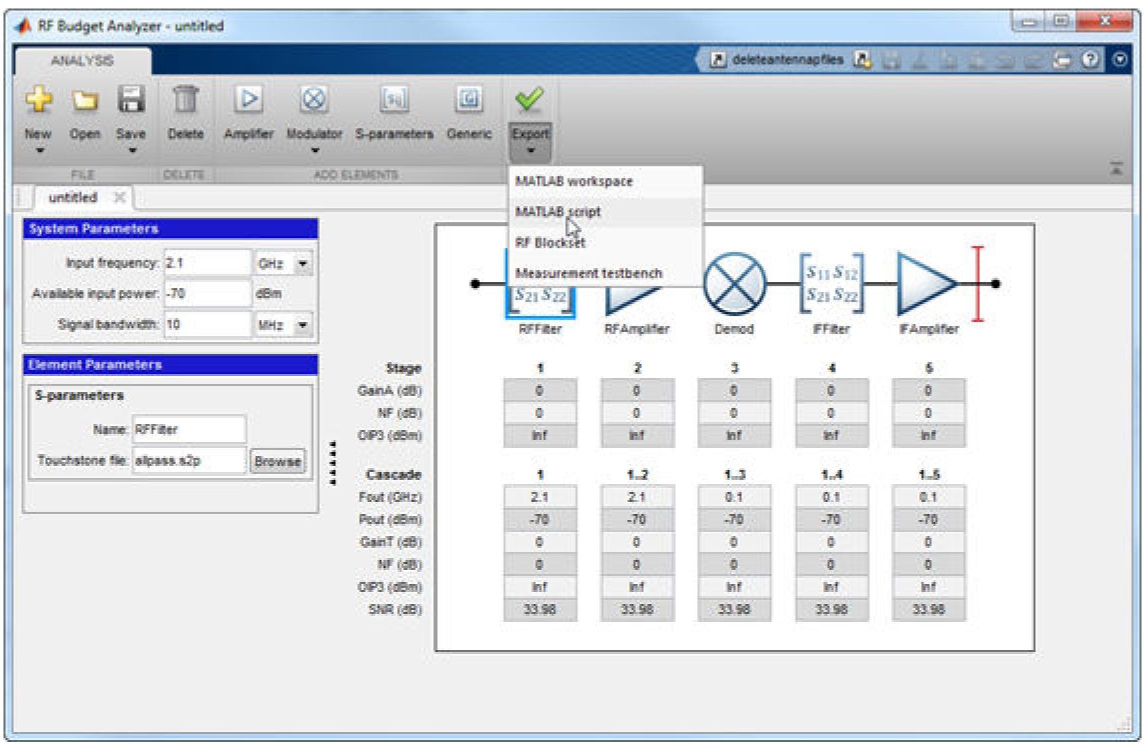
You can now use the **RF Budget Analyzer** app to analyze direct conversion and low-IF transmitters and receivers.

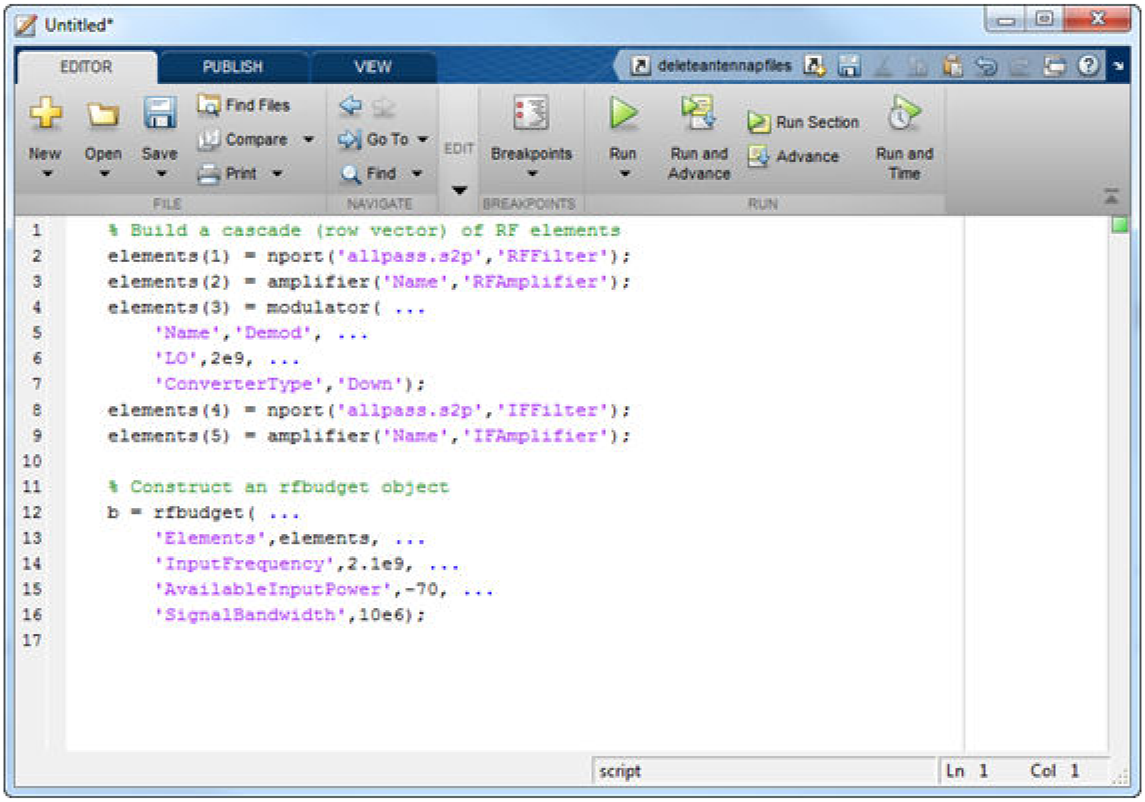
MATLAB Functionality for RF Budget Analysis: Script and automate RF budget analysis

Use the `rfbudget` class to compute the cumulative gain, noise figure, and third intercept point (IP3) for an RF circuit created using RF elements. You can also use the `show` function to view the analysis on the **RF Budget Analyzer** app.

Generation of MATLAB script from rfBudgetAnalyzer App: Automate scripting for RF budget analysis

In the **RF Budget Analyzer**, under **Export** tab, click `MATLAB Script` to create a MATLAB script of the RF circuit and RF budget analysis. The figure shows you how to create the script of an RF circuit and its budget analysis.





Amplifier and Modulator Objects: Build networks of active and passive components using circuit objects

Use the `amplifier`, `modulator`, and `rfelement` to build RF circuits. You can also analyze the RF budget of the circuit by using the `rfbudget` class.

R2016b

Version: 3.1

No New Features or Changes

R2016a

Version: 3.0

New Features

RF Budget Analyzer: Analytically compute gain, noise figure, and IP3 for cascaded RF components

You can use the RF Budget Analyzer app to compute the per-stage and cumulative gain, noise figure, and third intercept point (IP3) for a system of RF elements. Export the computed values to the MATLAB workspace. Simulate the system using SimRF™. Verify the gain, noise figure, and IP3 using the SimRF testbench simulation. To learn how to use the app, seeSuperheterodyne Receiver Using RF Budget Analyzer App.

R2015b

Version: 2.17

New Features

LC Ladder Objects: Build different ladder configurations of LC elements and include them in arbitrary networks

You can use the `lcladder` class to create LC filters, calculate the S-parameters of the filters, and add the filters to an RF Toolbox circuit object.

Group Delay Calculation: Compute the group delay for networks with arbitrary topologies and for any S-parameter data

You can use the `groupdelay` function to calculate the group delay of the S-parameters object and these RF network elements: nport, circuit, and LC ladder.

rationalfit Function Examples: Learn best practices for fitting S-parameter data

Use these examples to understand how to improve the quality of rationalfit output:

- Using the 'NPoles' Parameter with rationalfit
- Using the 'Weight' Parameter with rationalfit
- Using the 'DelayFactor' Parameter with rationalfit

2015a

Version: 2.16

New Features

N-Port S-Parameters Circuit Element: Analyze RF networks containing S-parameter components

You can use `nport` to create a linear nport circuit element from Touchstone files and s-parameter objects. Use the `add` function to add n-port element to circuit element.

T-Parameters Data Object: Store and convert T-parameters data from other formats

You can create tparameter objects from touchstone files, network parameters, and rfdata.network objects using the `tparameters` data object function.

rfplot Improvements: Plot real, imaginary, magnitude, or angle data

You can now use the `rfplot` function to specify the types of plot such as decibels (default), real, imaginary, absolute, or angle.

R2014b

Gain, noise figure, and IP3 cascade analysis for general RF stages

You can now compute the `gain`, `noise figure`, `oip3`, and `iip3` of cascaded networks using the rfchain object. Display the stage-by-stage results in a spreadsheet format using the `worksheet` method. Visualize the results using the `plot` method.

Network parameter object improvements for conversion of any network data object

You can now convert `rfckt` or `rfdata.data` or `rfdata.network` objects into any network parameter objects including S-parameters, Y-parameters, ABCD-parameters objects.

R2014a

General de-embedding function for 2N-port S-parameters

You can now use the deembedsparams function to de-embed 2N-port fixture effects from 2N-port measurements. It supports both three-dimensional S-parameters data and S-parameter objects.

N-port Touchstone file writing from any network data object or raw data

You can use the rfwrite function to write Touchstone files from three-dimensional network parameter data or any network parameter object (S-parameters, Y-parameters, Z-parameters, ABCD-parameters, etc.)

rationalfit function at least six times faster

The rationalfit function now fits a rational model to S-parameter data at least six times faster than previous releases. This responsiveness improves both RF Toolbox command-line behavior and SimRF simulation of S-parameter blocks.

R2013b

Additional functions for constructing arbitrary RLC networks

In R2013b, the following new functions are available:

- resistor, capacitor, inductor, and circuit — Use the basic building functions of an RF circuit to construct RLC networks.
- add — Insert basic RF elements to a circuit.
- clone — Duplicate any existing RF elements or circuits.
- setports — Define node pairs as ports of a circuit.
- setterminals — Map the nodes of a circuit to terminals.

S-parameter extraction for arbitrary RLC networks

The sparameters function now includes added functionality that you can use to calculate the S-parameters of RLC networks.

R2013a

Improved rationalfit function

This release introduces additional pole-searching optimizations to the `rationalfit` function algorithm. Models that the function returns in this release tend to have fewer poles than those in previous releases.

Compatibility Considerations

To constrain the function results across releases and machine architectures, explicitly specify optional parameters such as error tolerance and number of poles when you call the function. Given a data set and corresponding frequencies, the function attempts to calculate a rational function approximation to within a given specification. However, the exact model that the function returns can differ between releases and machines, as the algorithm improves.

R2012b

Network parameter interface improvements for faster reading of Touchstone files, plotting, and converting parameters

New network parameter objects and functions are available, with support for:

- Reading Touchstone files
- Converting network parameters
- Plotting network parameters

Additionally, some functions have been updated to support the new interface. For more information, see RF Network Parameter Objects.

Name-value pair syntax for rationalfit function

The `rationalfit` function now supports using name-value pairs for optional input arguments. Name-value pair arguments can be specified in any order and improve readability of code.

R2012a

New Visualization Option for Intermediate-Frequency (IF) Planning Object

The `OpenIF` object supports a new interactive, graphical interface for assessing spur information in a mixer chain. You access this interface using the new `show` method.

Enhanced S-Parameter Conversion Functions

S-parameter conversion functions have been enhanced to support larger data sets. The following functions now support conversion between parameter sets of 2*N*-port networks.

- `abcd2s`
- `abcd2y`
- `abcd2z`
- `s2abcd`
- `y2abcd`
- `z2abcd`

The `s2smm` function now supports mixed-mode conversions for *N*-port devices.

The following mixed-mode S-parameter functions now support mixed-mode conversions for 2*N*-port devices:

- `s2scc`
- `s2scd`
- `s2sdc`
- `s2sdd`
- `smm2s`

New Demos for Signal Integrity Applications

Two new signal-integrity demos are available in this version.

- The `Bandpass Filter Response` demo describes a procedure for designing and analyzing a simple bandpass filter using `rfckt` objects.
- The `MOS Interconnect and Crosstalk` demo reproduces Pillage and Rohrer's classic result from "Waveform Evaluation for Timing Analysis".

Enhanced Rational Fitting

The `rationalfit` function has improved robustness, speed, and accuracy in this version.

R2011b

New Intermediate Frequency (IF) Planning Object

The OpenIF object supports a new partial workflow for multiband transmitter or receiver design. Use these objects to analyze intermediate frequencies (IFs) that do not produce interference (spurs) in operating bands.

New Functions for Calculating Transmission Line RLGC Parameters

The rlgc2s and s2rlgc functions allow you to calculate the per-unit-length RLGC parameters of a transmission line from transmission-line S-parameters.

Enhanced Rational Fitting

The rationalfit function has improved robustness, speed, and accuracy in this version.

Compatibility Considerations

Some default values of rationalfit have changed. For more information, see the function reference page.

Conversion of Error and Warning Message Identifiers

For R2011b, error and warning messages identifiers have changed in RF Toolbox software.

Compatibility Considerations

If you have scripts or functions that use message identifiers that changed, you must update the code to use the new identifiers. Typically, message identifiers are used to turn off specific warning messages, or in code that uses a try/catch statement and performs an action based on a specific error identifier.

For example, the `rf:rfckt:seriesrlc:setpositive:NotAPositive` identifier has changed to `rf:rfbase:rfbase:setpositive:NotAPositive`. If your code checks for `rf:rfckt:seriesrlc:setpositive:NotAPositive`, you must update it to check for `rf:rfbase:rfbase:setpositive:NotAPositive` instead.

To determine the identifier for a warning, run the following command just after you see the warning:

```
[MSG,MSGID] = lastwarn;
```

This command saves the message identifier to the variable MSGID.

To determine the identifier for an error, run the following command just after you see the error:

```
exception = MException.last;
MSGID = exception.identifier;
```

Note Warning messages indicate a potential issue with your code. While you can turn off a warning, a suggested alternative is to change your code so it runs warning-free.

R2011a

Version: 2.8.1

Bug Fixes

R2010b

Enhanced Rational Function Modeling

An improved algorithm for the `rationalfit` function fits an accurate rational model to passive S-parameter data in less time than in previous versions. In addition, a new parameter specifies the number of iterations `rationalfit` attempts at each value for the number of poles.

Compatibility Considerations

Default behavior for some parameters have changed:

- The number-of-poles argument `npoles` defaults to a minimum value of `0` in version 2.8, instead of `4`, as in previous versions.
- `rationalfit` does not display a wait bar by default in this version. A new `showwaitbar` parameter allows you to specify whether `rationalfit` displays a wait bar.

For more information on using this function, see the `rationalfit` reference page.

Extended Methods and Parameters for RF Object Visualization

RF Toolbox version 2.8 extends the Plots and Charts methods to include:

- Support for third-order intercept point and transducer power gain parameters, `IIP3` and `Gt`.
- A new method, `table`, for visualizing network data in the Variable Editor.

R2010a

Version: 2.7

New Features

Function Added

The makepassive function creates passive S-Parameters from any S-parameter array. Use this function to enforce strict numerical passivity on an array of S-parameters that represents a passive device.

Methods Added

Two new methods for rfmodel.rational objects are available:

- The ispassive method tests global passivity of an rfmodel.rational object.
- The stepresp method calculates the response of an rfmodel.rational object to a step signal. You can use this function to perform time-domain reflectometry (TDR) and time-domain transmission (TDT) analysis.

Demo Added

The Modeling a High-Speed Backplane (4-Port S-Parameters to Differential TDR and TDT) demo shows how to perform time-domain reflectometry (TDR) and time-domain transmission (TDT) analysis on network data.

R2009b

New Function for Testing Passivity of S-Parameters

The ispassive function checks the passivity of N-port S-parameter matrices.

Expanded Port-Ordering Schemes for S-Parameter Conversion Functions

The functions s2scc, s2scd, s2sdc, s2sdd, s2smm, smm2s now support a third commonly-used port-ordering. For more information on using this feature, see the corresponding function reference page.

Support for Calculation of Power-Wave Gain for Transfer Functions

The s2tf function can now calculate the power-wave gain of 2-port S-parameters. Calculation in terms of voltage is still the default option.

R2009a

New Functions for Converting 4N-Port S-Parameter Matrices

There are two new functions for converting between 4N-port single-ended S-parameter matrices and 2N-port mixed-mode S-parameter matrices:

- The `s2smm` function lets you convert 4N-port single-ended S-parameters to 2N-port mixed-mode S-parameters. You can view the 2N-port output data to see interactions, such as crosstalk, that are not apparent in the single-ended data. This lets you easily select the ports of interest for further analysis.
- The `smm2s` function lets you convert 2N-port mixed-mode S-parameters to 4N-port single-ended S-parameters.

Enhanced Dielectric Loss Model in Three Transmission Line Objects

The following objects now provide a more realistic model for dielectric loss:

- `rfckt.coaxial`
- `rfckt.twowire`
- `rfckt.parallelplate`

To specify dielectric loss, you use a new property, `LossTangent`. This property replaces the `SigmaDiel` parameter.

Compatibility Considerations

Your existing objects with a nonzero value for the `SigmaDiel` parameter no longer model dielectric loss. Instead, the objects issue a warning message and use the default value of zero for the `LossTangent` property when you use the `analyze` method.

Demos Added

Two new demos show how to design broadband impedance matching networks for RF components:

- `Designing Broadband Matching Networks for Antennas` shows how to design a matching network for an antenna.
- `Designing Broadband Matching Networks (Part2:Amplifier)` shows how to design a matching network for an amplifier.

R2008b

Version: 2.4

New Features

cascadesparams Function now supports N-port S-parameters

You can now use the cascadesparams function to cascade the S-parameters of an arbitrary number of N-port devices to form a network. The function lets you specify how to connect the ports of each N-port device to the ports of the subsequent N-port device in the cascade. For more information about the function, see the cascadesparams reference page.

Improvements to the plotyy Method

The plotyy method now uses a more intuitive approach when determining how to plot the specified parameters if you do not specify the plot format. For more information about the function, see the plotyy reference page.

R2008a

Version: 2.3

New Features

Calculation and Plotting Metrics Added

You can now compute and visualize group delay, voltage gain, and stability factor using the `calculate` and `plot` methods.

Network Parameter Conversion Functions Enhanced

You can now use the `s2sdd`, `s2sdc`, `s2scd`, and `s2scc` functions to perform conversions on network parameters with alternate port arrangements.

gammams and gammaml Functions Added

Use the new `gammams` and `gammaml` functions to compute source and load reflection coefficients required for simultaneous conjugate match.

z2gamma Function Added

Use the new `z2gamma` function to convert impedance values to reflection coefficients.

Demos Added and Updated

A new demo, `Writing a Touchstone File`, shows how to write `rfckt` object data to an industry-standard Touchstone data file.

`Modeling a High-Speed Backplane (4-Port S-Parameters to a Rational Function Model)` now uses the new Communications Toolbox™ eye diagram scope, `commscope.eyediagram`, to plot the eye diagram.

Command-Line Help Updated

The `help` function returns additional information for objects and packages. The function now includes descriptions of all properties and links to all relevant methods.

R2007b

Version: 2.2

New Features

snp2smp Function Added

Use the new `snp2smp` function to convert N-port S-parameter data and termination impedances to M-port S-parameters.

circle Method Added

Use the new `circle` method to place circles on a Smith® Chart to depict stability regions and display constant gain, noise figure, reflection, and immitance circles.

powergain Function Added

Use the new `powergain` function to compute various power gains of a 2-port network.

Smith Chart Enhanced

The `smith` method now lets you plot the network parameters of devices with more than two ports on a Smith Chart.

Demos Added and Updated

`Modeling a High-Speed Backplane (Measured 16-Port S-parameters to 4-Port S-Parameters)` is the new first part of a four-part demo on "Modeling a High-Speed Backplane." The new demo shows how to extract 4-port S-parameter data from 16-port S-parameter data. The original three parts of the demo are now parts 2, 3, and 4.

The following demos replace the "Designing Impedance Matching Networks" and "Placing Circles on a Smith Chart" demos, respectively, and show how to use the new `circle` method:

- `Designing Matching Networks (Part1: Networks with LNA and Lumped Elements)` uses the available gain design technique to design a low-noise amplifier for a wireless communication system.
- `Designing Matching Networks (Part2: Single Stub Transmission Lines)` shows how to design input and output matching networks for an amplifier.

R2007a

Version: 2.1

New Features

Agilent P2D and S2D System-Level Verification Model Support Added

The `rfckt.amplifier` and `rfckt.mixer` objects now let you import system-level verification models of amplifiers and mixers, respectively, using data from Agilent® P2D and S2D files.

Use P2D files to specify the following data for multiple operating conditions, such as temperature and bias values:

- Small-signal network parameters
- Power-dependent network parameters
- Noise data
- Intermodulation tables

Use S2D files to specify the following data for multiple operating conditions:

- Small-signal network parameters
- Gain compression (1 dB)
- Third-order intercept point (IP3)
- Power-dependent S_{21} parameters
- Noise data
- Intermodulation tables

Use the following methods to work with operating condition data after you import a P2D or S2D file into an RF object:

- `setop` — Use this method to set operating condition values or to list all available values.
- `getop` — Use this method to display the selected operating condition values.

Mixer Spur Analysis Added

You can import an intermodulation table into an `rfckt.mixer` object. The object's `plot` method has a new option for plotting mixer spur data.

timeresp Method Added

Use the new `timeresp` method of the `rfmodel.rational` object to compute the time response of an `rfmodel` object to a specified input signal. Use this method rather than computing impulse response with the `impulse` method and then convolving that response with the input signal because the `timeresp` method generally gives a more accurate output signal for a given input signal.

Plotting Methods Added

Four new plotting methods provide additional plotting options:

- Use the `plotyy` method of the `rfckt` class to create a plot that contains RF circuit object data on both the left and right Y-axes.
- Use the `loglog` method of the `rfckt` class to plot RF circuit object data on a log-log scale.
- Use the `semilogx` method of the `rfckt` class to plot RF circuit object data using a logarithmic scale for the X-axis.

- Use the semilogy method of the rfckt class to plot RF circuit object data using a logarithmic scale for the Y-axis.

gamma2z Function Added

Use the new gamma2z function to compute input impedance from a reflection coefficient.

Tab Completion Added

Tab completion is now available in the MATLAB command window for all functions and methods. For more information on tab completion, see the MATLAB documentation.

Data Tips Added

Data tips are now available for any RF plot. For more information on data tips, see Data Cursor — Displaying Data Values Interactively.

Demos Added and Updated

Visualizing Mixer Spurs shows how to use the toolbox to perform mixer spur analysis using data from an intermodulation table and then plot the output power spectrum of the desired signal and the undesired spurs.

Modeling High-Speed Backplane (Measured 4-Port S-Parameters to Rational Function Model) now uses the timeresp method to compute the time-domain response of a system characterized by measured data.

Modeling High-Speed Backplane (Rational Function Model to Simulink Model) now includes code that you can use to generate a Simulink® model for any rfmodel.rational object.

R2006b

Version: 2.0

New Features

S-Parameter Conversion Function Added

Use the `s2tf` function to convert 2-port scattering parameters into a transfer function that represents the normalized voltage gain of a 2-port network.

rfmodel Class Added

Use objects from the `rfmodel` class to represent components and networks with mathematical equations. The `rfmodel.rational` object stores a rational function model of a component or network.

rationalfit Function Added

Use the `rationalfit` function to fit a rational function to passive data that represents an RF component or network and then store the result in an `rfmodel.rational` object. This type of model is useful to signal integrity engineers, whose goal is to reliably connect high-speed semiconductor devices with, for example, multi-Gbit/s serial data streams across backplanes and printed circuit boards.

freqresp and impulse Functions Added

Use the `freqresp` method of the `rfmodel` class to compute the frequency response of an `rfmodel` object.

Use the `impulse` method of the `rfmodel` class to compute the impulse response of an `rfmodel` object.

Support for Exporting Verilog-A Models Added

Use the `writeva` method of the `rfmodel` class to export a description of an RF component or network for use in a time-domain circuit simulator.

Demos Added

"Modeling a High-Speed Backplane (Part 1: Measured 4-Port S-Parameters to a Rational Function Model)" shows how to use the toolbox to model a differential high-speed backplane using rational functions.

"Modeling a High-Speed Backplane (Part 2: Rational Function Model to a Verilog-A Module)" shows how to use toolbox functions to generate a Verilog-A module that models the high-level behavior of a high-speed backplane.

"Modeling a Differential High-Speed Backplane in Simulink" shows how to use Simulink to simulate a differential high-speed backplane.

R2006a

Version: 1.3

New Features

S-Parameter Conversion Functions Added

Use the `s2scc` function to convert 4-port, single-ended S-parameters to 2-port, common mode S-parameters.

Use the `s2scd` function to convert 4-port, single-ended S-parameters to 2-port, cross mode S-parameters.

Use the `s2sdc` function to convert 4-port, single-ended S-parameters to 2-port, cross mode S-parameters.

Use the `s2sdd` function to convert 4-port, single-ended S-parameters to 2-port, differential mode S-parameters.

R14SP3

Version: 1.2

New Features

extract Function Added

Use the `extract` function to extract specified network parameters from a circuit or data object and return the result in an array.

Circuit Object Added

Use `rfckt.rlcgline` to construct an RLCG transmission line object.

Transmission Line Object Improved

The new `Freq` property of the circuit object, `rfckt.txline`, is a vector of positive frequencies at which the parameter values are known.

The `Loss`, `PV`, and `Z0` properties of the circuit object, `rfckt.txline`, can now be vectors of line loss, phase velocity, and characteristic impedance values that correspond to the frequencies specified in the `Freq` property.

The new `IntpType` property of the circuit object, `rfckt.txline`, is the interpolation method used to calculate the parameter values between the known frequencies.

Touchstone Data File Support Improved

You can now read data from Touchstone data files that contain comments and spaces between sections of data.

Demos Improved

The demos have new documentation and can be accessed using an improved interface.

Command Window Help for Functions That Act on Circuit Objects Added

You can access help for functions that act on circuit objects by using the syntax `help` *`functionname`* at the MATLAB command prompt.

R14SP2

Version: 1.1

New Features

Plot Figures Integrated into the RF Tool GUI

In earlier versions, a plot figure would appear in a separate window after clicking the **Plot** button. In this version, plot figures are integrated into the GUI itself.

Five Objects Added

These objects can be used to store rfdata such as network parameters, noise figure, power, IP3, and spot noise.

Three Circuit Objects Added

Use `rfckt.delay` to model delay lines, `rfckt.hybridg` to model hybrid G connected networks, and `rfckt.passive` to model RF passive networks.

Methods Added

The new `write` method allows saving of RF network data into files for all `rfckt` objects.

The new methods, `read` and `restore`, read and restore data for `rfckt.datafile`, `rfckt.amplifier`, and `rfckt.mixer`.

Method Enhanced

The `analyze` method now takes three additional optional inputs for the load, source, and reference impedances.

Functions Added

The functions `stabilitymu` and `stabilityk` calculate the stability factors μ and k.

The functions `h2g` and `g2h` convert between hybrid G and hybrid H parameters.

General Enhancements

It is now possible to create the objects `rfckt.amplifier` and `rfckt.mixer` from a MATLAB variable.

The frequency-dependent NF and IP3 data types were added to the AMP format.