Agilent Technologies

Advanced Design System 2002

Tuning, Optimization and Statistical Design

February 2002

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Agilent Technologies
395 Page Mill Road
Palo Alto, CA 94304 U.S.A.

Contents

Contents

1 Tuning

Basic Procedure 1-2
Using the Tune Control Dialog Box.... 1-3
Using the Tune Control Details Dialog Box.... 1-5
Slider Control Settings.... 1-6
Component Button 1-7
Brief, Update, and Cancel Buttons 1-7
Using the Tune Parameter Selection Dialog Box.... 1-7
Setting Preferences for Tuning 1-9
Tuning Example—Analog/RF Systems.... 1-10
Tuning Example—Signal Processing.... 1-14
Tuning Hierarchical Networks 1-17
Tunable and Non-Tunable Parameters 1-17

2 Performing Nominal Optimization

Nominal Optimization Minimum Requirements 2-2
Specifying Component Parameters for Optimization.... 2-2
Placing an Appropriate Simulation Control Component 2-5
Setting Optimization Goals.... 2-6
Setting Optimization Job Parameters 2-8
Selecting an Optimizer and Goals 2-8
Setting Parameter Information.... 2-10
Displaying Analysis Data on the Schematic 2-13
Initiating an Optimization 2-14
Updating Optimized Parameters on the Design 2-15
Nominal Optimization Example 2-15
Discrete Optimization Example 2-21
Setting Up Inductor L2 for Discrete Optimization 2-22
Setting Up Capacitor C2v for Discrete Optimization.... 2-24
Setting Up the Nominal Optimization Component.... 2-25
Simulation Results.... 2-25
Performing Discrete Optimization Using a Data File 2-25
Understanding the Discrete Data File 2-29
Sensitivity Analysis.... 2-30
Normalized Sensitivities 2-32
Sensitivity Analysis Example.... 2-33
Swept Optimization.... 2-34
Swept Optimization Example.... 2-35

How ADS Names Optimization Variables 2-36

3 Using Statistical Design

Statistical Design Minimum Requirements 3-2
Performing Yield Analysis 3-2
Specifying Component Parameters for Yield Analysis 3-3
Placing an Appropriate Simulation Control Component for Yield Analysis 3-6
Setting Up a Yield Specification 3-6
Setting Job Parameters for Yield Analysis 3-9
Selecting a Yield Specification 3-9
Setting Parameter Information 3-11
Selecting a Shadow Model Type for Yield Analysis 3-12
Enabling Post Production Tuning 3-13
Displaying Analysis Data on the Schematic 3-14
Initiating Yield Analysis 3-15
Swept Yield Analysis 3-15
Yield Analysis Example 3-16
Creating a Measurement Histogram 3-17
Setup Requirements 3-18
Generating the Measurement Histogram 3-19
Creating a Sensitivity Histogram 3-21
Setup Requirements 3-22
Generating the Sensitivity Histogram 3-23
Performing Yield Optimization 3-25
Setting Job Parameters for Yield Optimization 3-25
Selecting a Specification for Yield Optimization 3-26
Setting Parameter Information 3-27
Selecting a Shadow Model Type for Yield Optimization 3-28
Enabling Post Production Tuning 3-29
Displaying Analysis Data on the Schematic 3-29
Swept Yield Optimization 3-30
Yield Optimization Example 3-30
Statistical Correlation 3-33

4 Using Design of Experiments (DOE)

DOE and Computer Simulation 4-1
Minimum DOE Requirements 4-2
Specifying Component Parameters for DOE 4-2
Placing an Appropriate Simulation Control Component for DOE 4-4
Setting DOE Goals 4-5
Setting Job Parameters for DOE 4-7
Selecting a DOE Specification 4-8
Setting Parameter Information 4-9

Displaying Analysis Data on the Schematic 4-11
Initiating Design of Experiments 4-11
Swept DOE 4-12
DOE Terminology 4-12
DOE Concepts 4-14
Design Matrix 4-19
Multilevel Designs 4-21
DOE Outputs 4-21
Response Values 4-23
DOE Basic Example 4-24
Setup the DOE Goal Components 4-25
Setup the DOE Component 4-28
Start the Experiment 4-29
Analyze the Experiment 4-29
Optimizing Using DOE Outputs 4-33
How Goals Are Affected 4-36
Performing the DOE Confirmation Experiment 4-38
Analyzing the DOE Confirmation Experiment 4-38
DOE References 4-40

5 Available Value Types

Value Types for Nominal Optimization 5-1
Value Types for Statistical Design 5-2
Value Types for DOE 5-3

6 Summary of Optimizers

Available Optimizers 6-1
Search Methods 6-2
Random Search 6-2
Gradient Search 6-3
Quasi-Newton Search 6-3
Gauss-Newton/Quasi-Newton (minimax) Search 6-3
Discrete Optimizer 6-4
Genetic Algorithm Search 6-4
Sensitivity Analysis 6-7
Error Function Formulation 6-8
Least-Squares 6-8
Minimax 6-10
Least Pth 6-11
Worst Case 6-12

7 Using Monte Carlo Yield Analysis

Monte Carlo Trials and Confidence Levels 7-1

Confidence Tables 7-5

Index

Chapter 1: Tuning

Advanced Design System's tuning capability allows you to change one or more design parameter value and quickly see its effect on the output without resimulating the entire design. Multiple plots generated from various tuning trials can be overlaid in the Data Display window. This can help you find the best results and the most sensitive components or parameters more easily.

In this chapter, you will find:

- An explanation of tuning
- Basic tuning procedure
- Detailed description of each tuning dialog box
- Tuning example for Analog/RF Systems
- Tuning example for Signal Processing
- Tunable and non-tunable parameters

When you analyze a network (*Simulate* > *Simulate*), a lot of information is compiled by the simulator prior to the actual network simulation. The simulator must set up your network topology, load all the values of the component parameters, and organize your measurement requests.

With the tuning feature, you can avoid the pre-processing. Tuning performs the pre-processing once and then assumes that you are now just trying to change some of the parameter values. A new simulation will take place, but using the same network topology and list of measurements. You can tune a large number of components, including those that are processed by a measurement equation component, such as VSWR.

Only the small changes regarding the new parameter values are needed. For each tune value(s) specified, the Data Display window overlays the new response curve beside the previous responses for comparison.

The benefits of this feature include:

- A reduction in total simulation time (by avoiding the pre-processing step).
- The ability to view the effects of changing parameter values. This can help you understand or locate the sensitive components or parameters in your design.
- A convenient means to quickly enter new parameter values.

Basic Procedure

Basic tuning consists of the following steps:

1. Build the design you want to tune.
2. Simulate your design.
3. Set up the Data Display window to display the results you want from your design.
4. Choose **Simulate** > **Tuning** or choose the **Tuning icon** (tuning fork) from the toolbar.
5. Wait for the initial analysis to complete. The Tune Control dialog box appears.
6. Select the parameter(s) you want to tune by clicking it on the schematic.
7. Select the tune analysis mode you want from the *Simulate* drop-down list. There are three choices:
    - After pressing the *Tune* button
    - After each change
    - While slider is moving
8. Change the tunable parameter by moving the slider, clicking the left/right arrows, or typing the value in the box and pressing *Enter* or *Tab*.
9. Depending on the tuning mode you chose in step 7, your results will be updated as soon as the new simulation is completed.

This basic procedure can be expanded for complete flexibility by the use of three additional dialog boxes:

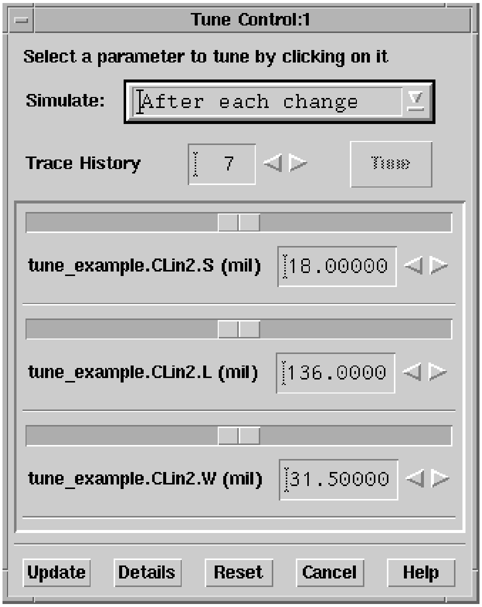
- **Tune Control Details** This dialog box contains the same controls as the brief Tune Control dialog box, but with fields for choosing each slider's minimum and maximum range, step size, and linear or logarithmic scaling.
- **Tune Parameter Selection** This dialog box allows you to select and edit all tunable parameters for a given component, including those *not* displayed on the schematic.
- **Preferences for Schematic, Tuning tab** This dialog box allows you to choose the defaults for tuning. These include the tuning analysis mode (tune after each

change, after one or more changes, or continuously while the slider is moving), number of traces to be displayed, the slider's minimum and maximum values, step size, and linear or logarithmic scaling.

More information about tuning and each of the tuning dialog boxes will be presented next.

Using the Tune Control Dialog Box

The Tune Control dialog box controls basic tuning functions. It can be expanded by choosing *Details* at the bottom and toggled back by choosing *Brief* from the detailed dialog box that appears.



Tune Control:1

Select a parameter to tune by clicking on it

Simulate: After each change

Trace History 7

tune_example.CLin2.S (mil) 18.00000

tune_example.CLin2.L (mil) 136.0000

tune_example.CLin2.W (mil) 31.50000

As stated in the Basic Procedure section, to start tuning:

1. Build the design you want to tune.
2. Simulate your design.
3. Set up the Data Display window to display the results you want from your design.

4. Choose **Simulate** > **Tuning** or choose the **Tuning icon** (tuning fork) from the toolbar.

5. Wait for the initial analysis to complete. The Tune Control dialog box appears.

6. Select the parameter(s) you want to tune by clicking it on the schematic. Remember to select the parameter itself, not the component symbol.

   For each parameter you select, a slider control will be added to the dialog box. In the example figure, three microstrip element parameters were selected.

   Selecting a parameter on the schematic works as a toggle; if you select it again, the slider is removed from the dialog box.

   An alternate way to select parameters is to bring up the Tune Parameter Selection dialog box. (Choose the *Details* button, then the *Component* button on the next dialog box, to display this dialog box.) This method displays all tunable parameters for a selected component (including those not displayed on the schematic) and lets you easily add or delete parameters for tuning. See "Using the Tune Parameter Selection Dialog Box" on page 1-7 for more details.

7. Select the tune analysis mode. This tells the program when you want tuning simulation to occur. Near the top of the dialog box is the label *Simulate* and a drop-down list. There are three choices. You can tune:

   - After pressing the *Tune* button (just below the list). This method is preferred if you are making multiple changes.

   - After each change

   - While slider is moving. This method creates continuous tuning.

8. Change the tunable parameter by any of the following methods:

   - Move the slider.

   - Click the left or right arrows.

   - Type the value in the box and press *Enter* or *Tab*.

   At this point, you may notice that the step size jump is too large or the range of tuning is too high. To set the fine control for the sliders, choose the **Details** button at the bottom of the dialog box. The Tune Control Details dialog box appears. To return to the current dialog box, choose *Brief*. See "Using the Tune Control Details Dialog Box" on page 1-5 for more information.

9. Depending on the tuning mode you chose in step 7, your results will be updated as soon as the new simulation is completed.

10. You can control the number of traces displayed on your plots with the *Trace History* field. The default is 7, or the number entered in the Schematic Preferences, Tune tab dialog box. Type in a number or use the left/right arrows. Keep in mind that once a history trace is overwritten, it is not recoverable.

11. You can choose the *Update* button at the bottom if you want the new parameter values you currently have entered in the Tune Control dialog box to be written into your schematic. If you do not want to change the values in your schematic, do *not* choose this button.

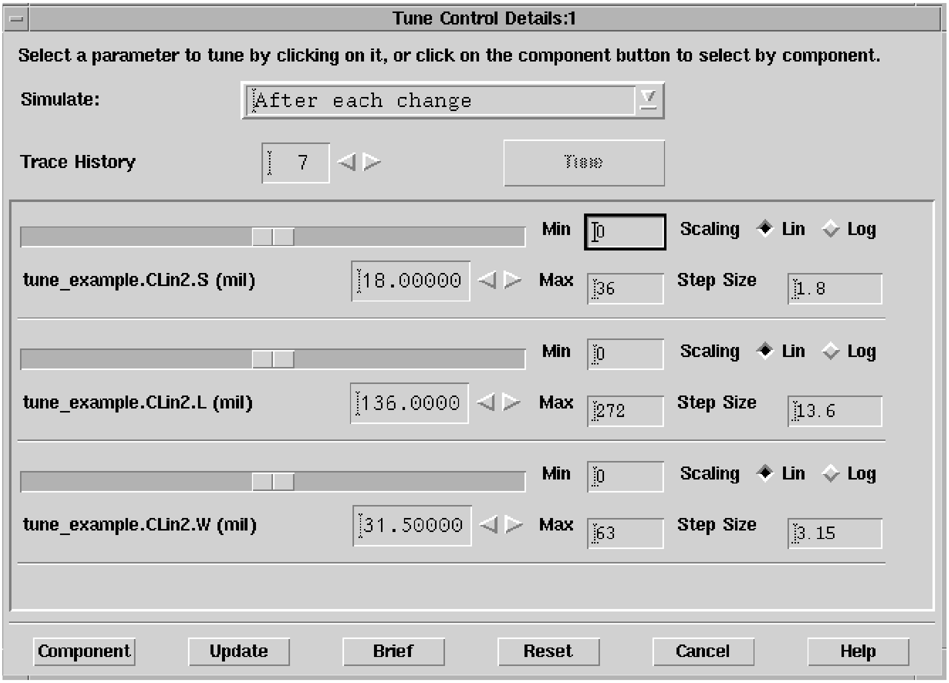
12. Choose the *Reset* button at the bottom to restore all slider values to the initial or last updated value.

13. Choose *Cancel* to cancel tuning. (If you choose Cancel, you will have to start tuning over from the beginning.)

14. All trace history is deleted when tune mode is exited.

Using the Tune Control Details Dialog Box

The Tune Control Details dialog box contains all the functions of the simpler Tune Control dialog box (see "Using the Tune Control Dialog Box" on page 1-3) plus fine control for the sliders and a Component button to access the Tune Parameter Selection dialog box.



Tune Control Details:1

Select a parameter to tune by clicking on it, or click on the component button to select by component.

Simulate: After each change

Trace History 7

tune_example.CLin2.S (mil) 18.00000 Min 0 Max 36 Scaling ● Lin ○ Log Step Size 1.8

tune_example.CLin2.L (mil) 136.0000 Min 0 Max 272 Scaling ● Lin ○ Log Step Size 13.6

tune_example.CLin2.W (mil) 31.50000 Min 0 Max 63 Scaling ● Lin ○ Log Step Size 3.15

Slider Control Settings

For each slider, you can control:

- **Min** Enter a minimum value for the tuning range.
- **Max** Enter a maximum value for the tuning range.
- **Step Size** Enter a value that represents the step size you want. The slider will jump from one value to the next based on this step size when the right or left arrows are selected.
- **Scale** Choose Linear or Logarithmic.

Component Button

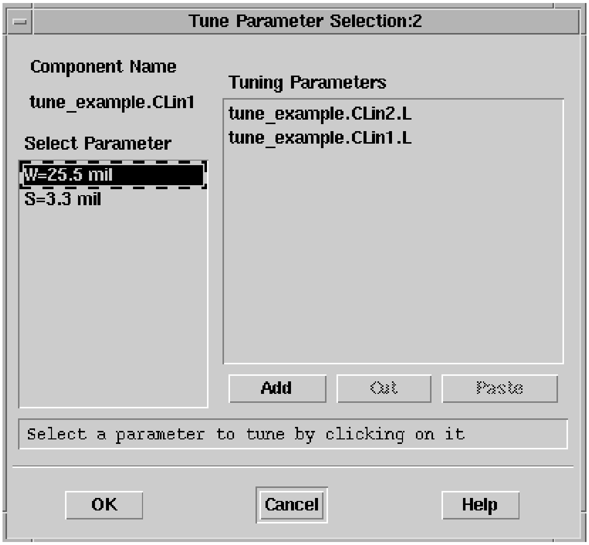
The Component button at the bottom left of the dialog box takes you to the Tune Parameter Selection dialog box, described later. This dialog displays all tunable parameters for a selected component or instance (including those not displayed on the schematic) and lets you easily add, delete, or reorder parameters for tuning.

Brief, Update, and Cancel Buttons

Choose *Brief* to return to the simpler Tune Control dialog box. Choose *Update* if you want the new parameter values you currently have entered in the Tune Control Details dialog box to be written into your schematic. Choose *Reset* to restore all slider values to the initial or last updated value. Choose *Cancel* to cancel the tuning operation. (If you choose Cancel, you will have to start tuning over from the beginning, and the trace history is deleted.)

Using the Tune Parameter Selection Dialog Box

To access the Tune Parameter Selection dialog box, choose the **Details** button at the bottom of the **Tune Control** dialog box and then the **Component** button at the bottom of the **Tune Control Details** dialog box. The purpose of this dialog box is to:



Tune Parameter Selection:2

Component Name

tune_example.CLin1

Select Parameter

W=25.5 mil
S=3.3 mil

Tuning Parameters

tune_example.CLin2.L
tune_example.CLin1.L

Select a parameter to tune by clicking on it

- Let you select tunable parameters that are not displayed on the schematic.
- Easily add, delete, and reorder parameters for tuning.

To use the Tune Parameter Selection dialog box:

1. Select the component or instance on your schematic that you want to tune. The component or instance name appears in the upper left under Component Name.
2. A list of tunable parameters associated with this component or instance appears in the Select Parameter box on the left. Click on the parameter or parameters you want to tune.
3. Choose the **Add** button to move the parameter you chose to the Tuning Parameters box on the right.
4. Repeat steps 2 and 3 for additional parameters from the same component.
5. You can edit the Tuning Parameters list by choosing the Add, Cut, or Paste buttons at the bottom.

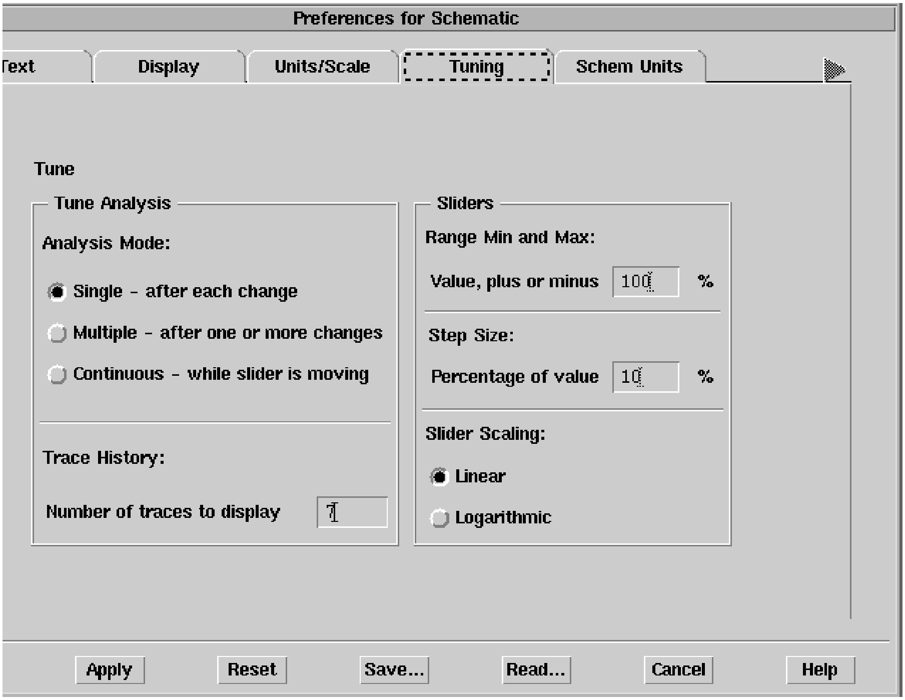
   Cut moves a selected parameter back to the list on the left (or into a buffer if that parameter's component is not currently selected as Component Name).

   Add moves a selected parameter to the list on the right.

   Paste moves the last parameter that was cut back to the box on the right.
6. Select additional components or instances.
7. Repeat steps 2 through 5.
8. When you are finished building/editing your list of parameters for the components you want to tune, choose the **OK** button. This action accepts your changes and returns you to the Tune Control dialog box. Choose **Cancel** to disregard your changes and return to the Tune Control Details dialog box.

Setting Preferences for Tuning

You can set the defaults for tuning by choosing *Options > Preferences > Tuning tab* from any Schematic window.



Preferences for Schematic

Tune

Tune Analysis

Analysis Mode:

● Single – after each change

○ Multiple – after one or more changes

○ Continuous – while slider is moving

Trace History:

Number of traces to display 1

Sliders

Range Min and Max:

Value, plus or minus 100 %

Step Size:

Percentage of value 10 %

Slider Scaling:

● Linear

○ Logarithmic

You can set the defaults for Tune Analysis, as follows:

- **Analysis Mode**

  Single Perform analysis after each change.

  Multiple Perform analysis only after the Tune button is pressed. This is designed for tuning after multiple changes, but can be used for single changes.

  Continuous Perform analysis while the slider is moving.

- **Trace History** Set the number of traces to be displayed on your plots.

You can set the defaults for the Sliders, as follows:

- **Range Min and Max** Set the minimum and maximum range to the initial parameter value plus or minus the percent you enter here.

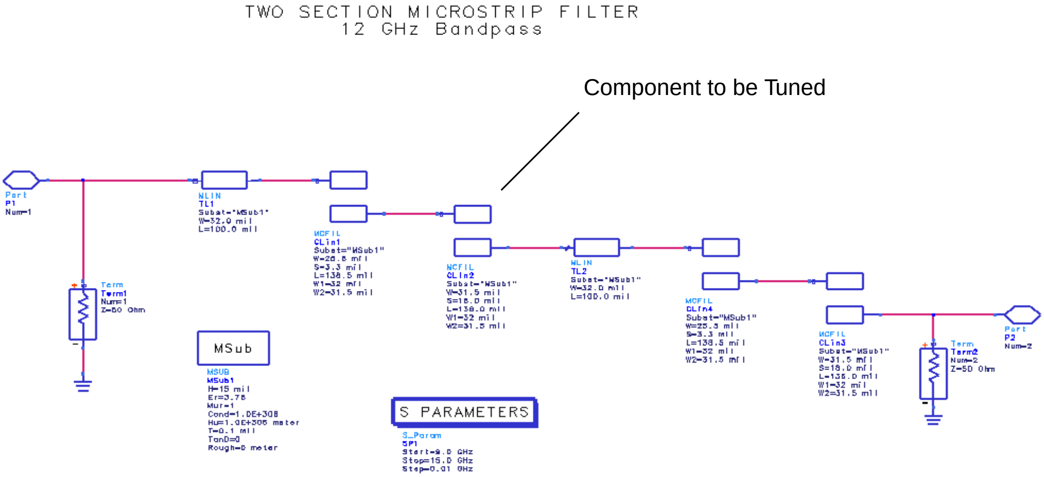
- **Step Size** Enter a percentage of the initial parameter value.
- **Slider Scaling** Choose Linear or Logarithmic.

These values can be changed later in the Tune Control and Tune Control Details dialog boxes.

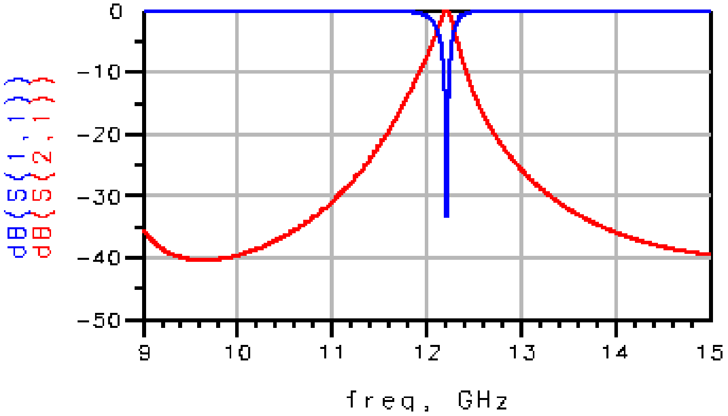
When you have made your changes, choose *Apply* or *OK*.

Tuning Example—Analog/RF Systems

This section shows a tuning example for Analog/RF Systems simulation. If you want to follow the same steps, but with a Signal Processing example, skip to "Tuning Example—Signal Processing" on page 1-14. The design is a two-section microstrip filter with a 12 GHz bandpass. We will vary the effect of the filter by tuning the microstrip coupled-line filter component and look at plots of S_{11} and S_{21}. We will tune component MCFIL instance 2's spacing parameter.



Before tuning, the simulation results are as follows:

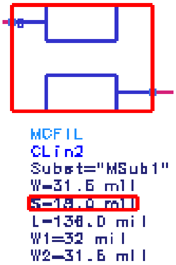


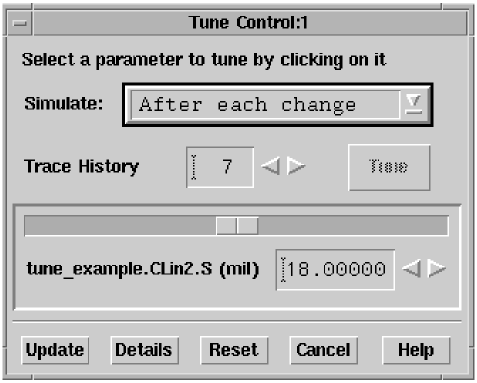
To tune this circuit, do the following:

1. Build the design or copy the example project from the Examples directory.

   /examples/Tutorial/Learn_Tune_prj

   to a directory that you have write permission.
2. Simulate the design.
3. Set up the Data Display window to display the S_{11} and S_{21} results in dB as shown above or choose *Window* > *Open Data Display* > *Before_Tune* from the example project.
4. Choose **Simulate** > **Tuning** or choose the **Tuning icon** (tuning fork) from the toolbar.
5. Wait for the initial analysis to complete. The Tune Control dialog box appears.
6. For the **MCFIL instance 2** component, select the **S** (spacing) parameter on the schematic, as shown below.





Tune Control:1

Select a parameter to tune by clicking on it

Simulate: After each change

Trace History 7

tune_example.CLin2.S (mil) 18.00000

7. Select the tune analysis mode from the Simulate drop-down list. This tells the program when you want tuning to occur. For this example, choose **After each change.** We will tune for three values of one parameter.

   When you finish with this example, try each tuning analysis method (after pressing the Tune button, after each change, while the slider is moving) to see which one works best for you.

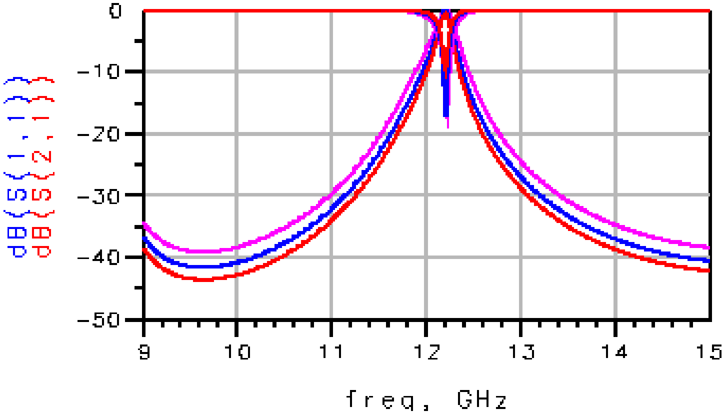
8. The results of our tuning will be displayed in the same Data Display window that the initial simulation was displayed in.

9. You can change the tunable parameter by any of the following methods:

   - Move the slider
   - Click the left or right arrows
   - Type the value in the box

We will move the slider for **tune_example.CLin2.S** (spacing) so that values of **15** mil (slightly below in the initial value of 18 mil) to **25** mil in **5** mil steps are simulated and observe the results each time we make a change.

10. The plot below shows the simulation results after three tuning trials.

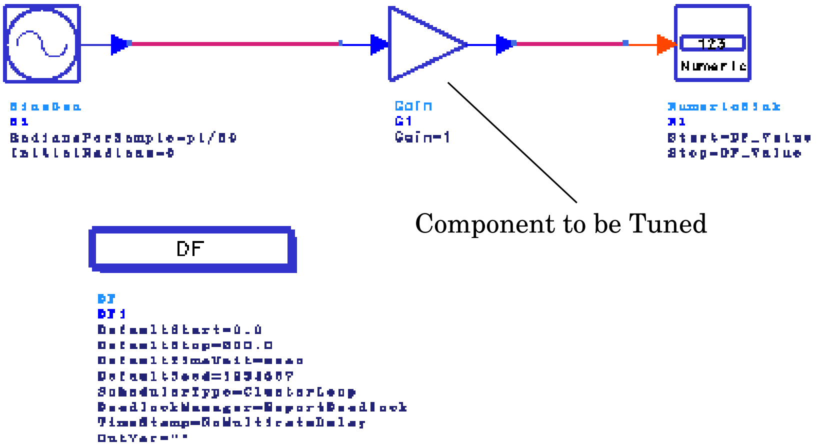


11. You can choose the *Update* button at the bottom if you want the value you currently have entered in the Tune Control dialog box to be written into your schematic. If you do not want to change the values in your schematic, do *not* choose this button.

12. Choose *OK* to accept your changes or *Cancel* to cancel tuning.

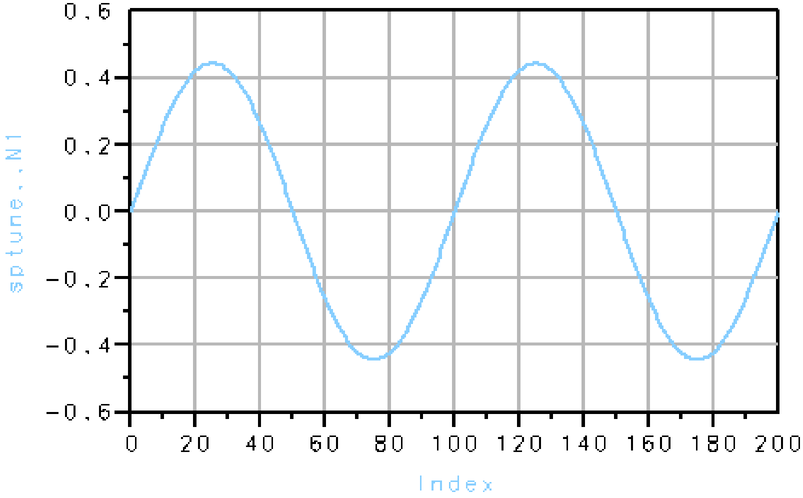
Tuning Example—Signal Processing

This section shows a tuning example for Signal Processing simulation. The steps are the same as in the previous example, but with a Signal Processing design. The design consists of a sine wave source, a gain component, and a numeric sink.



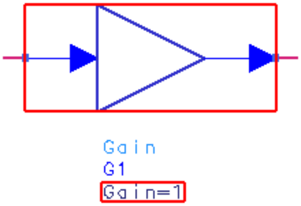
Component to be Tuned

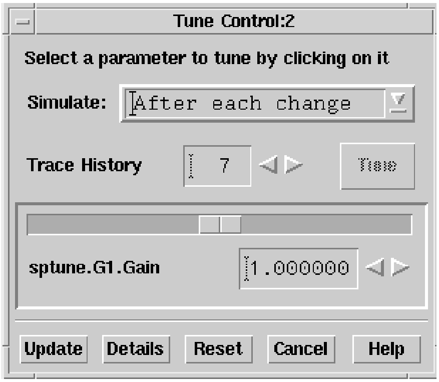
Before tuning, the simulation results are as follows:



To tune this circuit, do the following:

1. Build the design.
2. Simulate the design.
3. Set up the Data Display window to display N1 (from the numeric sink) as shown in the last figure.
4. Choose **Simulate** > **Tuning** or choose the **Tuning icon** (tuning fork) from the toolbar.
5. Wait for the initial analysis to complete. The Tune Control dialog box appears.
6. For the **Gain** component, select the **Gain** parameter on the schematic, as shown below.





Tune Control:2

Select a parameter to tune by clicking on it

Simulate: After each change

Trace History 7 Tile

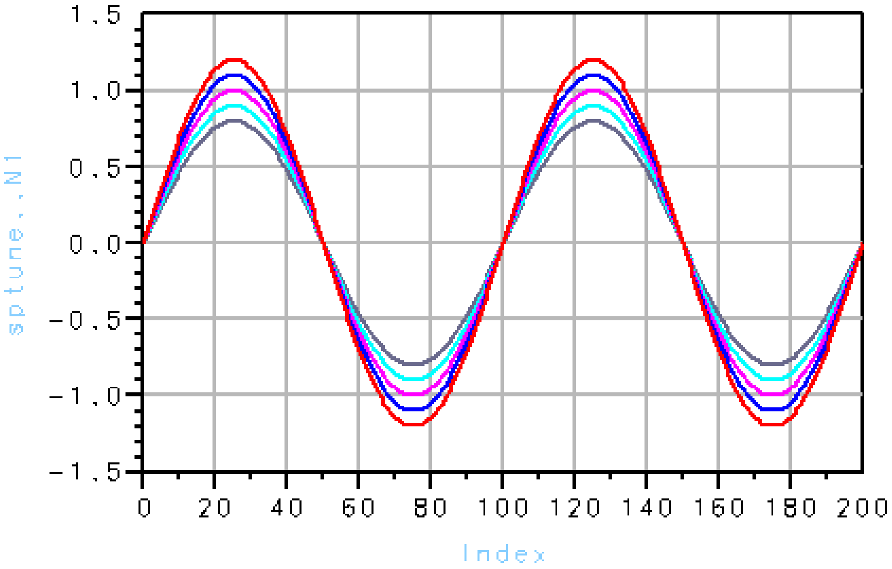
sptune.G1.Gain 1.000000

7. Select the tune analysis mode from the Simulate drop-down list. This tells the program when you want tuning to occur. For this example, choose **After each change.** We will tune for three values of one parameter.

When you finish with this example, try each tuning analysis method (after pressing the Tune button, after each change, while the slider is moving) to see which one works best for you.

8. The results of our tuning will be displayed in the same Data Display window that the initial simulation was displayed in.
9. You can change the tunable parameter by any of the following methods:
   - Move the slider
   - Click the left or right arrows
   - Type the value in the box and press *Enter* or *Tab*

   We will move the slider for **sptune.G1.Gain** so that values of **0.8** to **1.2** mil in **0.1** steps are simulated and observe the results each time we make a change.
10. The plot below shows the simulation results after five tuning trials.



11. You can choose the *Update* button at the bottom if you want the value you currently have entered in the Tune Control dialog box to be written into your schematic. If you do not want to change the values in your schematic, do *not* choose this button.

12. Choose *OK* to accept your changes or *Cancel* to cancel tuning.

Tuning Hierarchical Networks

If your schematic design is hierarchical, i.e., if it contains subnetworks, you can tune the components within those subnetworks without having to exit tuning.

While in the Tune mode, select *Push Into Hierarchy* from the View menu in the Schematic window or choose the *Push Into Hierarchy* icon. Click on the subnetwork of interest. The Schematic window now displays the subnetwork design and at this point, you can proceed to tune items inside the subnetwork.

Note Changes are made at the definition, not instance, level. This means that if you update a subnetwork using tune mode, *all* instances of that subnetwork will be changed, not just the one you are pushed into.

Tunable and Non-Tunable Parameters

Most Advanced Design System parameters can be tuned, but not all. The following rules govern whether or not a parameter can be tuned. In order for a parameter to be tunable, *all* six of the following statements must be true:

1. It is a VAR or a simulator component parameter.
2. It is contained in the hierarchy of the network.
3. It is optimizable, although specification of an optimization range is not required.
4. It is real- or integer-valued (can include a scale factor with or without units); it cannot be expression-valued.
5. It is a *specified* value, entered in the ADS design (parameters that default to simulator default values are not tunable).
6. Its value is less than 1,000,000 (scale factor not included).

Next are some examples of these rules.

- It is a VAR or a simulator component parameter.

  Valid: VAR1.xx

Valid: C1.C

Not Valid: SMT_Pad.W (SMT_Pad is not a simulator component)

- It is optimizable, although specification of an optimization range is not required.

Valid: MLIN1.L = 50 mil

Valid: MLIN1.L = 50 mil opt {25 mil to 100 mil}

Not Valid: MLIN1.Subst = “Msub1” (not optimizable)

- It is real- or integer-valued (can include a scale factor with or without units); it cannot be expression-valued.

Valid: R1.R = 50

Valid: R1.R = 50 K

Valid: R1.R = 50 kOhm

Not Valid: R1.R = Rnom (expression-valued)

Not Valid: R1.R = 50 * Rscale (expression-valued)

Note To use the Agilent Ptolemy (Signal Processing) Interactive Controls and Displays library components (such as TkPlot) with tune mode, you must dismiss the Interactive Controls and Displays component between each tune with its pop-up dialog box.

Chapter 2: Performing Nominal Optimization

Nominal Optimization, also known as Performance Optimization, is the process of modifying a set of parameter values to satisfy predetermined performance goals. Optimizers compare computed and desired responses and modify design parameter values to bring the computed response closer to that desired. Nominal Optimization is available in the Advanced Design System simulators as follows:

- For Analog/RF systems simulation using any analysis type (such as AC, DC, S-Parameter, Harmonic Balance, Circuit Envelope, and Transient simulation types)
- For Agilent Ptolemy signal processing simulation

Nominal optimization can be performed in conjunction with any frequency-domain or time-domain Analog/RF Systems simulation component as well as most Signal Processing components. For example:

- To optimize the response of a low-pass filter, you can perform an S-parameter simulation or an AC simulation to calculate the output amplitude of the filter over a frequency range, then change filter parameter values to refine filter response shape.
- To optimize the rise time of a pulse, you can perform a transient simulation to calculate the output voltage over a period of time, then change circuit parameter values to fine-tune the rise time of the pulse.
- You can optimize the gain of a carrier recovery loop to achieve a desired lock time and residual loop error.
- You can optimize a fixed-point bit-width parameter in a DSP design.

Note Some Agilent Ptolemy parameter types (Complex, Precision, Array, String, or Filename) require additional steps to complete optimization. These steps are described in *Optimizing Various Parameter Types* in Chapter 8 of the *Agilent Ptolemy Simulation* manual.

Examples of goals include characteristics of an output signal such as rise time, bandpass shape, or harmonic output. Minimum and/or maximum acceptable performance are used to define the limits of a goal. These limits can be a function of a fixed or a swept variable evaluated at the beginning of the optimization process.

The steps required to perform nominal optimization include:

1. Running a simulation.
2. Comparing results with the goal.
3. Changing the circuit parameters to obtain results that are likely to be closer to the goal.
4. Running a simulation again with the new parameter values.

Nominal Optimization Minimum Requirements

Prior to applying Nominal Optimization, you need:

- At least one component parameter in your design identified as an optimization variable. You specify details in the Component Parameter dialog box by choosing the Optimization/Statistics/DOE Setup button.
- At least one optimization goal component (*Goal*) placed in the Schematic window.
- One nominal optimization component (*Optim*) placed in the Schematic window.

  The *Goal* and *Optim* components are accessed as follows:

  - For Analog/RF Systems simulation, from the *Optim*/*Stat*/*Yield*/*DOE* palette or component library.
  - For Signal Processing simulation, from the *Controller*s palette or component library.
- One simulation control component (a Data Flow controller for Agilent Ptolemy simulating or an AC, DC, S-Parameter, Harmonic Balance, Circuit Envelope, and Transient simulation component for Analog/RF Systems simulation).

Specifying Component Parameters for Optimization

In this part of the procedure, you can enable or disable the optimization status of the parameter and specify the type and format for the parameter range over which optimization is to take place.

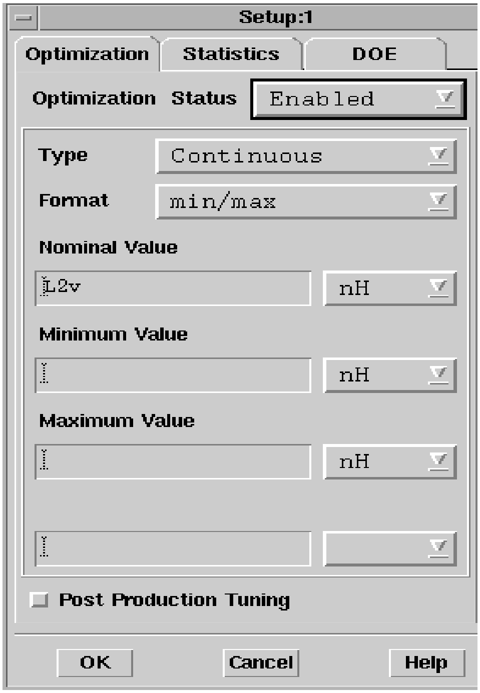
The procedure for specifying components for optimization is as follows:

1. Select and place an appropriate component from one of the component palettes or component libraries. For example, place a parallel resistor-inductor-capacitor (PRLC) from the Lumped Components palette in Analog/RF Systems or a Gain component in Signal Processing.

2. Double-click the component in the Schematic window to edit its parameters.

3. From the component dialog box, highlight the parameter that you want to optimize in the Select Parameters box (for example R for parallel resistance), then choose the **Optimization/Statistics/DOE Setup** button, which will only appear for optimizable parameters (and when the default *Standard* Parameter Entry Mode is selected). The *Setup* dialog box appears, with the **Optimization** tab active. (Neither the Statistics nor DOE tab is needed for optimization. These functions are described in Chapter 3, Using Statistical Design and Chapter 4, Using Design of Experiments (DOE).)

4. From the Optimization Status drop-down list, select **Enabled** so you can edit the appropriate fields. *Enabled* causes the parameter to be optimized when the simulation is run. *Disabled* temporarily deactivates this parameter from being optimized, and *Clear* removes the values you previously applied to the design after you select *Clear* in this box, followed by *Apply* in the component dialog box.



Optimization Status: Enabled

Type: Continuous

Format: min/max

Nominal Value: L2v nH

Minimum Value: nH

Maximum Value: nH

□ Post Production Tuning

5. From the Type drop-down list, select an appropriate optimization Value Type (Continuous or Discrete). For a description of Discrete optimization refer to "Discrete Optimization Example" on page 2-21. For descriptions of Value Types, refer to the section "Value Types for Nominal Optimization" on page 5-1 in Chapter 5, Available Value Types.

6. From the Format drop-down list, select an appropriate optimization format (*min/max, +/- Delta %, +/- Delta,* or *Unconstrained)*. Generally, you should pick as narrow a range as you believe will work. A large range or *Unconstrained* could use more simulation time. For descriptions of the available formats, refer to the section "Value Types for Nominal Optimization" on page 5-1 in Chapter 5, Available Value Types.

7. If you selected a *min/max* format, you can optionally enter values for nominal, minimum, and maximum in the appropriate boxes, and select an appropriate unit assignment for each from the drop-down list next to the boxes. If you selected an *Unconstrained* format, only a nominal value and associated unit

need to be specified. If you selected either of the *Delta* formats, the companion limit values must be specified.

8. From the Nominal Value field and the Units drop-down list, the value and units in your design for this component are displayed. You can change these to set your starting point for your optimization if you wish.

Note The Post Production Tuning button is used in Statistical Design, not in Nominal Optimization, and is covered in the next chapter.

Placing an Appropriate Simulation Control Component

An appropriate simulation control component must be placed in the design prior to initiating an optimization. Typically, this step will already be done from an earlier simulation or yield analysis.

For Analog/RF Systems simulation, all analysis types are supported, for example place one of the following components:

- **AC** from the AC Simulation palette or library
- **DC** from the DC Simulation palette or library
- **S-Param** from the S-Param Simulation palette or library
- **Harmonic Balance** from the HB Simulation palette or library
- **ENV** from the Envelope Simulation palette or library
- **Tran** from the Transient Simulation library

For details on specifying parameters for each of these control components, refer to the *Circuit Simulation* manual.

For Agilent Ptolemy simulation, place a

- **Data Flow** controller from the Controllers library or palette

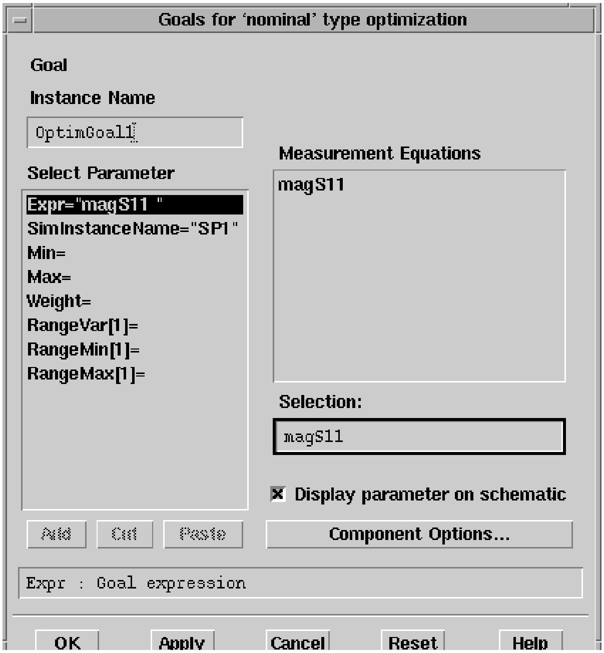
For details on specifying parameters for the Data Flow controller, refer to the *Agilent Ptolemy Simulation* manual.

Setting Optimization Goals

Optimization goals are specified by placing a *Goal* component and double-clicking it to display the *Goals for Nominal Type Optimization* dialog box. The Goal component can be found as follows:

- For Analog/RF Systems simulation, from the *Optim / Stat / Yield / DOE* palette or library
- For Signal Processing simulation, from the *Controller*s palette or library

You can specify and place more than one Goal if needed. The goals to be used are referenced by the Nominal Optimization component, as described in the later section, "Selecting an Optimizer and Goals" on page 2-8. By default, all goals placed apply to all Nominal Optimization components in a design.



Goals for 'nominal' type optimization

Goal

Instance Name: OptimGoal1

Select Parameter:
Expr="magS11 "
SimInstanceName="SP1"
Min=
Max=
Weight=
RangeVar[1]=
RangeMin[1]=
RangeMax[1]=

Measurement Equations: magS11

Selection: magS11

☒ Display parameter on schematic

Expr : Goal expression

To set appropriate goal specifications in this dialog box:

1. If desired, enter a name in the Instance Name field that is different from the assigned default name shown.

2. In the Select Parameter list box on the left, click on each parameter that you want to modify, then make other associated changes in the box on the right. When you select a parameter, such as Expr, all relevant items in your design will be displayed in the box. The style of this box varies depending on the parameter, as described in the table below.

Table 2-1. Parameter Goals for Nominal Optimization

Parameter	**Description**	**Use Model**
Expr	A valid AEL expression that operates on the simulation results, such as mag(S11), or the name of a MeasEqn. For more information on AEL expressions, refer to the AEL manual or the Expressions, Measurements, and Simulation Data Processing manual.	The list box label becomes Measurement Equations. All associated expressions are displayed in the box. Select the one you want to optimize and it will appear just below in the Selection box. For expressions not related to MeasEqns, you must type them in the Selection box.
SimInstanceName	Enter the instance name for the simulation control component that you placed in your design, which will generate the data used by the Expr field.	The list box label becomes Analysis Components. Select the analysis component (simulation controller), such as S-parameter, that you want to optimize and it will appear just below in the Selection box.
Min	Enter a number for a minimum acceptable response value.	Fields for Parameter Entry Mode and Equation editor are used as in any component parameter dialog box. Type a value in the box. Note: Both Min and Max do not have to be specified, but at least one does.
Max	Enter a number for a maximum acceptable response value.	Same as above.
Weight	Enter a weighting valued to be used in error function calculation. Default is 1.	Fields for Parameter Entry Mode and Equation editor are used as in any component parameter dialog box. Type a value in the box.

Table 2-1. Parameter Goals for Nominal Optimization

Parameter	Description	Use Model
RangeVar	Independent variable name.	Same as above, but note that this parameter is “indexable” and can be applied to more than one independent variable.
RangeMin	Minimum limit of range for independent variable during optimization.	Same as above.
RangeMax	Maximum limit of range for independent variable during optimization.	Same as above.

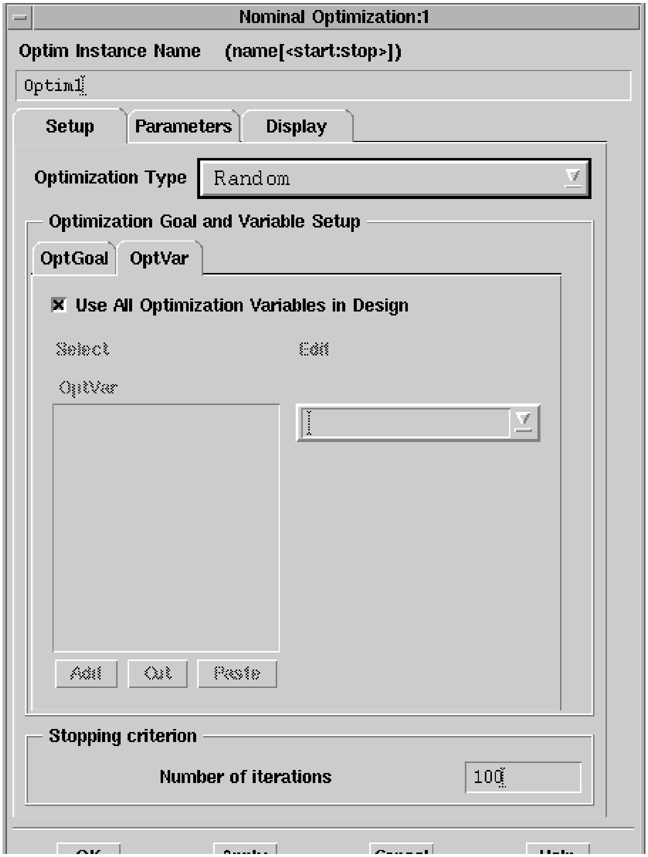
Setting Optimization Job Parameters

To set job parameters, you need to specify data in the Nominal Optimization dialog box, as follows:

1. Place an *Optim* component in the appropriate Schematic window. It is found as follows:
    - For Analog/RF Systems simulation, from the *Optim*/*Stat*/*Yield*/*DOE* palette or library
    - For Signal Processing simulation, from the *Controller*s palette or library
2. Double-click the component to bring up the Nominal Optimization dialog box, which has three tabs. It is displayed with the *Setup* tab active.
3. Make specifications in each tab of the dialog box, as described in the next three sections.

Selecting an Optimizer and Goals

To set up an optimization in the Setup tab of the Nominal Optimization dialog box:



Optim Instance Name (name[<start:stop>])

Optim1

Optimization Type Random

☒ Use All Optimization Variables in Design

Number of iterations 100

1. In the Optimization type field, select an appropriate type from the drop-down list, such as Random (the default), Gradient, Least Pth, Minmax, Genetic, etc. For details on selecting an appropriate optimizer, refer to Chapter 6, Summary of Optimizers. To perform a sensitivity analysis (which is not a type of optimization) select Sensitivity from the drop-down list. For more information on sensitivity analysis. Refer to “Sensitivity Analysis” on page 2-30.

2. In the Optimization Goal and Variable Setup box, first choose the OptGoal tab and accept the default *Use All goals in Design* checkbox. This is the best approach for most designs, and all goal components placed in a design will be implicitly associated with the optimization controller.

To associate a *subset* of all goals with a given optimization controller, deselect the *Use All goals in Design* checkbox. Select a goal from the *Edit* drop-down list, which will include all Goal components that are currently placed in the design, as described in the preceding section, "Setting Optimization Goals." Choose *Add* to place it in the OptGoal box, and repeat this procedure as necessary. Choose the *Cut* or *Paste* buttons, if necessary to make any changes in the OptGoal box.

3. Under Stopping criterion, specify the number of desired trial iterations to use during the optimization process.

   For the random optimizers (Random, Random Minimax, Random Max, and Genetic) this value represents the number of trials to attempt. Values in the range from 25-100 are recommended initially.

   For the iterative optimizers (Gradient, Gradient Minimax, Quasi-Newton, Least P^{th}, and Minimax) this value represents the number of iterations (improvements in the error function) to attempt. Less than 10 iterations are recommended initially. The default for all optimizers is 25 trials/iterations.

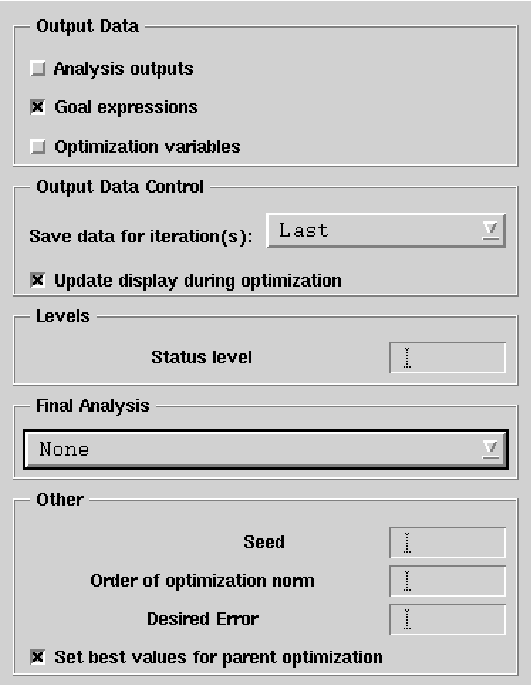
4. In the Optimization Goal and Variable Setup box, next choose the OptVar tab and accept the default *Use All Optimization Variables in Design* checkbox, unless you want to use only some optimization variables.

   You may have many optimization variables, specified by placing a one or more VAR (variables and equations) components, in your design and want to associate only one or two of the variables, for example, with a given optimization controller. To associate a *subset* of all optimization variables with a given optimization controller, deselect the *Use All Optimization Variables in Design* checkbox. Then enter the name of an optimization variable you want to use in the *Edit* field and choose the *Add* button. It is added to the OptVar box. Repeat this procedure as necessary. Choose the *Cut* or *Paste* buttons, if necessary to make any changes in the OptVar box.

5. Choose **Apply** to retain the specifications that you have made while you enter data into the Parameters tab, as described in the next section.

Setting Parameter Information

You set parameter information in the Parameters tab of the Nominal Optimization dialog box, such as what data to save and other information. To do this, follow these steps:



Output Data

□ Analysis outputs

☒ Goal expressions

□ Optimization variables

Output Data Control

Save data for iteration(s): Last

☒ Update display during optimization

Levels

Status level

Final Analysis

None

Other

Seed

Order of optimization norm

Desired Error

☒ Set best values for parent optimization

1. In the Output Data field, specify which data you want to retain in your dataset following optimization. Check the following choices that apply.

   - *Analysis outputs* sends all measurements (including measurement equations) to the dataset. This can create a substantial amount of data.

   - *Goals expressions* (default) sends the result of each active Goal's *Expr* field to the dataset.

   - *Optimization variables* sends the values of all active optimization variables to the dataset for each improvement found during the optimization.

2. In the Output Data Control field, specify whether you want to:

   - *Save data for iterations*. Choices are:

     Last -- Only the last iteration is saved to the dataset.

     Nominal & last (default) -- Only the nominal and last (best) iterations are saved to the dataset.

All -- Data for all iterations is saved. This can create a substantial amount of data.

- *Update display during optimization* (default). This updates the dataset on each optimization iteration so you can see the results in the Data Display window as they occur (instead of waiting to the end where all the traces are displayed at once).

3. In the Levels field, enter a number for the desired annotation level. Levels are 0-4, with increasing information displayed in the Status window. (4 is the default.)

4. In the Final Analysis field, specify whether you want to employ an analysis run after your optimization is complete. This analysis can be of any analysis controller component that does not introduce circularity in analysis execution. The drop-down list allows you to select *None* (for no Final Analysis, the default) or any analysis controller component currently in your design (such as an S-Parameter or Data Flow controller).

   Final analysis is useful when the analysis executed by the optimizer uses a different sweep grid than the one you want for your output. For example, if a course grid is required for optimization, but a finer grid, or a different range, is desired for output, then the analysis setup to generate this finer grid may be run after the optimization is completed, using the Final Analysis feature.

   Multiple analyses are run by grouping them together in a parameter sweeper (without specifying a sweep variable), and choosing this sweeper in the Final Analysis drop-down list box. Note that for swept or nested optimization, the Final Analysis parameter should not refer to any controller that is already executing. Data output for Final Analysis follows the information (flags) set in the companion *Optim* controller component.

5. In the Other field, specify a *Seed* value, *Order of optimization norm*, and *Desired Error* for use during optimization.

   - *Seed* is a value for the random number generator used to initiate an optimization. If *Seed* is not specified, the simulator chooses its own seed, which will be different each time an optimization type requiring a seed is used (Random, Random Max, Random Min, Genetic, and Discrete).

   - Possible values for *Order of optimization norm* are 2, 4, 8, or 16. (2 is the default.) For more information, refer to the section, "Error Function Formulation" on page 6-8 in Chapter 5, Summary of Optimizers.

- The *Desired Error* field represents the value of the error function that is acceptable to terminate the analysis. If you want all goals to be met, accept the default of 0.
- If you want to *Set the best values for the parent optimization*, leave the checkbox in its default setting (checked). With this box checked, the optimal values are saved internally so that they can be either user-updated or utilized by a subsequent analysis. To disable this setting, click the checkbox.

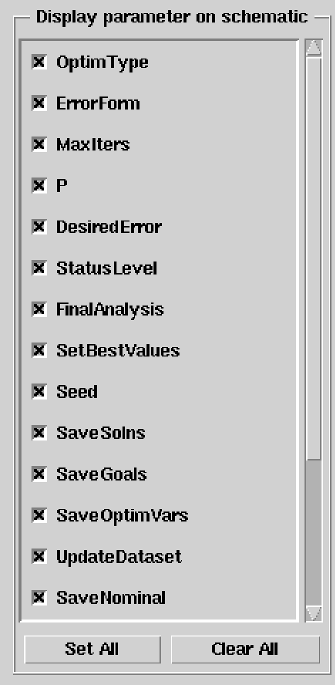
6. Choose **Apply** to retain the specifications that you have made while you enter data into the Display tab, as described in the next section.

Displaying Analysis Data on the Schematic

The Display tab of the Nominal Optimization dialog box is used to select the parameters that will be displayed on your schematic related to nominal optimization. The same Display tab and procedure is used for yield analysis and yield optimization described in Chapter 3, Using Statistical Design. Generally, it is not necessary to modify the visible parameters. Check the appropriate boxes in the Display tab dialog box (the default is all boxes are checked).

Make specifications as follows:

- To select or unselect any parameter, click in the appropriate checkbox.
- Choose **Set All** to check all parameters. Choose **Clear All** to uncheck all parameters.



Display parameter on schematic

☒ OptimType
☒ ErrorForm
☒ MaxIters
☒ P
☒ DesiredError
☒ StatusLevel
☒ FinalAnalysis
☒ SetBestValues
☒ Seed
☒ SaveSolns
☒ SaveGoals
☒ SaveOptimVars
☒ UpdateDataset
☒ SaveNominal

- After all selections have been correctly made, click **Apply** if you want to continue entering data in the other tabs of this dialog box. If you are finished entering data in this dialog box, click **OK**. Above is a nominal optimization dialog box example.

Initiating an Optimization

To initiate an optimization, select **Simulate** or click on the **Simulate** button on the toolbar. The optimization status is displayed in the Status window, including the initial and current error function (EF) values, and the current trial/iteration.

If the error function reaches the terminating error value, the simulator ceases optimization and indicates success.Choose **Cancel** to discontinue the optimization process.

Note If the optimization process becomes exceedingly long, you can use the *Stop Simulation* command on the Simulation/Synthesis menu in the status window to interrupt the process.

Updating Optimized Parameters on the Design

Select **Simulate** > **Update Optimization Values** if you want to update your Schematic window with the new parameter values resulting from a successful optimization. If you want save the design at this point, select **File** > **Save** or **File** > **Save As** and assign an appropriate name.

Nominal Optimization Example

The example used in this chapter is from Analog/RF Systems simulation. However, the nominal optimization procedure is the same for either Signal Processing or Analog/RF Systems type of simulation. For a bit-width example used in Agilent Ptolemy simulation, refer to Using Nominal Optimization in the *Agilent Ptolemy Simulation* manual.

This example is called *optex1_prj*, and it is located in the directory <*installation_dir*>/*examples*/*Tutorial* (where <*installation_dir*> represents your Advanced Design System installation directory). To access this example project and enable simulation, first copy it to a work directory, by choosing *File* > *Copy Project* in the Main window.

The circuit topology for the following simple example represents a 2-to-1 impedance matching transformer with a *passband* of one octave. The example circuit is intended to match the 100-ohm load resistance of R1 to 50 ohms over the range of 200 to 400 MHz. The circuit is swept from 100 to 500 MHz to view the out-of-band response as well as the passband response.

Figure 2-1 shows the beginning design, used for this example. The initial component values are shown in Figure 2-1. Figure 2-2 demonstrates that the initial response of the circuit is far from optimum.

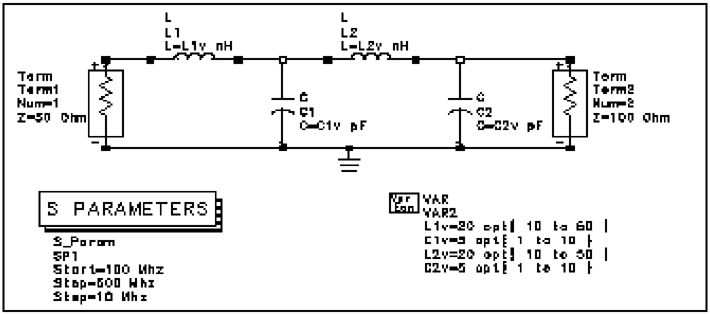


Figure 2-1. Design Used for Optimization Example

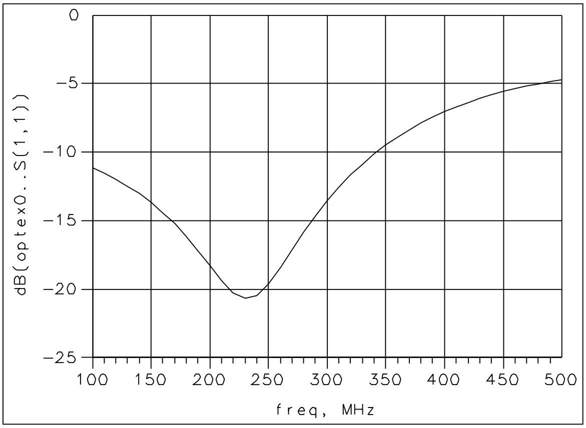
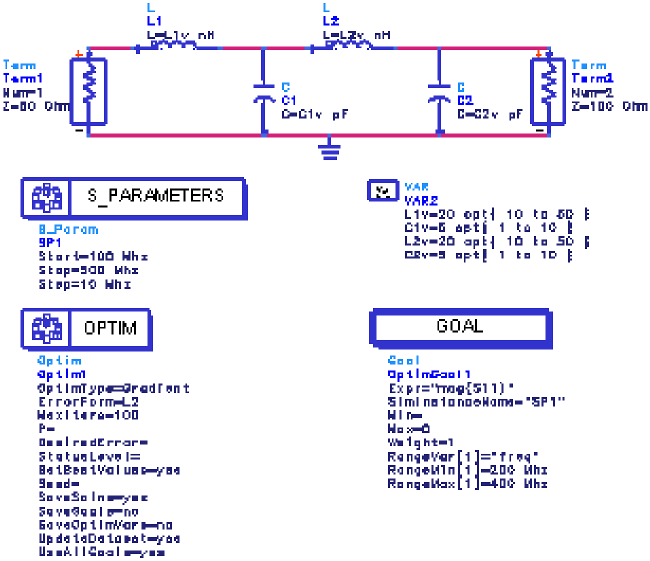


Figure 2-2. Initial Response of Example Circuit

To optimize the example circuit:

1. Since the intent of this design is to match the 100-ohm load resistance to 50 ohms, and we are measuring the input reflection coefficient S11, the goal is to make S11 as small as possible. We place a Nominal Optimization component and a Goal component from the *Optim/Stat/Yield/DOE* palette into the design, as shown in Figure 2-3 and make appropriate specifications.



S_Param
SP1
Start=100 MHz
Stop=500 MHz
Step=10 MHz

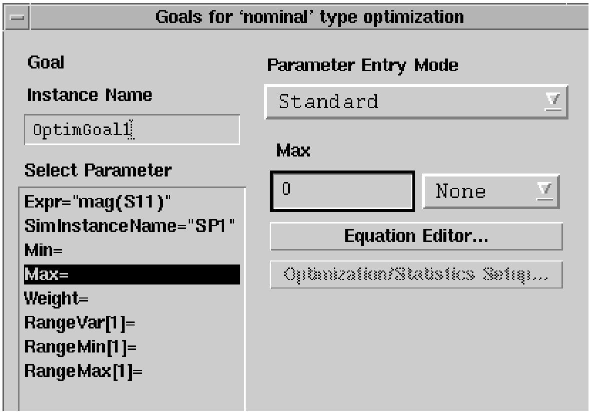
VAR
VAR2
L1v=20 opt{ 10 to 50 }
C1v=5 opt{ 1 to 10 }
L2v=20 opt{ 10 to 50 }
C2v=5 opt{ 1 to 10 }

Optim
Optim1
OptimType=Gradient
ErrorForm=L2
MaxIters=100
P=
DesiredError=
StatusLevel=
SetBestValues=yes
Seed=
SaveSolns=yes
SaveGoals=no
SaveOptimVars=no
UpdateDataset=yes
UseAllGoals=yes

Goal
OptimGoal1
Expr="mag(S11)"
SimInstanceName="SP1"
Min=
Max=0
Weight=1
RangeVar[1]="freq"
RangeMin[1]=200 MHz
RangeMax[1]=400 MHz

Figure 2-3. Design with Nominal Optimization and Goal component Added

Notice that in the Goal component, we have set the goal magnitude of S11 to zero in the *Goals for 'nominal' type optimization* dialog box, as shown in Figure 2-4. This box is accessed by double-clicking on the placed Goal component as shown Figure 2-3in Figure 2-4.



Goals for 'nominal' type optimization

Goal

Instance Name

OptimGoal1

Parameter Entry Mode: Standard

Select Parameter

Expr="mag(S11)"
SimInstanceName="SP1"
Min=
Max=
Weight=
RangeVar[1]=
RangeMin[1]=
RangeMax[1]=

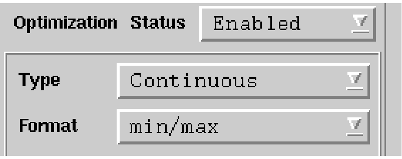
Max

0 None

Figure 2-4. Specifying Optimization Goals

2. The next step is to specify which circuit parameters are to be adjusted by the optimizer. For this example, all four parameters are optimized. These include the inductors L1 and L2, and the capacitors C1 and C2. Suppose the smallest inductance we can achieve in practice is 10 nH and the largest is 50 nH. Suppose the smallest capacitance we can achieve is 1 pF, and the largest is 10 pF.

   In the Setup dialog box for each of these four components, we specify a constrained optimization Value Type (specifically *min/max*), as shown in Figure 2-5. Refer to the section "Specifying Component Parameters for Optimization" on page 2-2 for more details.



Optimization Status: Enabled

Type: Continuous

Format: min/max

Figure 2-5. Enabling a Min/Max Constrained Optimization

This allows the circuit parameter to be adjusted within the constraints shown in Figure 2-6.

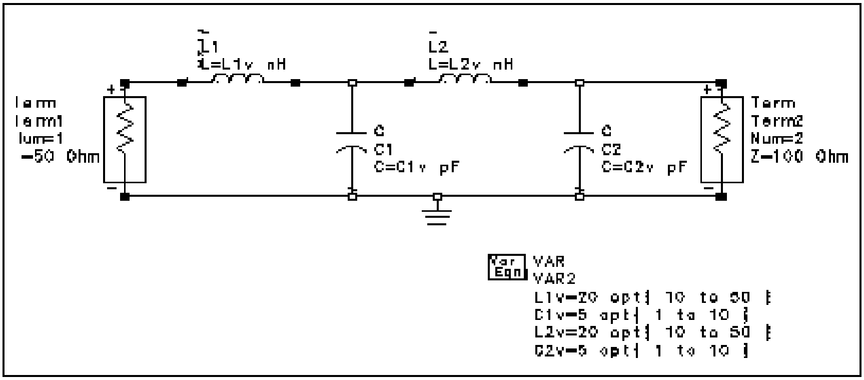


Figure 2-6. Specifying Circuit Parameters to be Adjusted by Optimizer

3. Now the circuit can be optimized. It turns out that the *least-squares* error function (explained in the section, "Error Function Formulation" on page 6-8 in Chapter 5, Summary of Optimizers) is relatively smooth, so the Gradient optimizer is a good initial choice.

4. To simulate this setup, choose *Simulate* > *Simulate*, or the Simulate icon on the toolbar.

5. Figure 2-7 shows the results of selecting 100 iterations of the Gradient optimizer on this circuit. The optimizer terminated due to a zero gradient, or minimum, in the error function after 18 iterations.

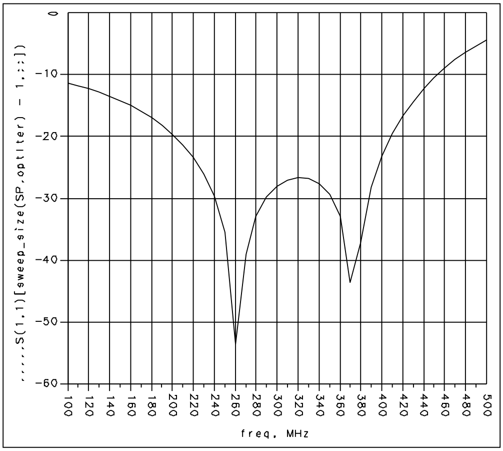


Figure 2-7. Results After 18 Iterations Using Gradient Optimizer

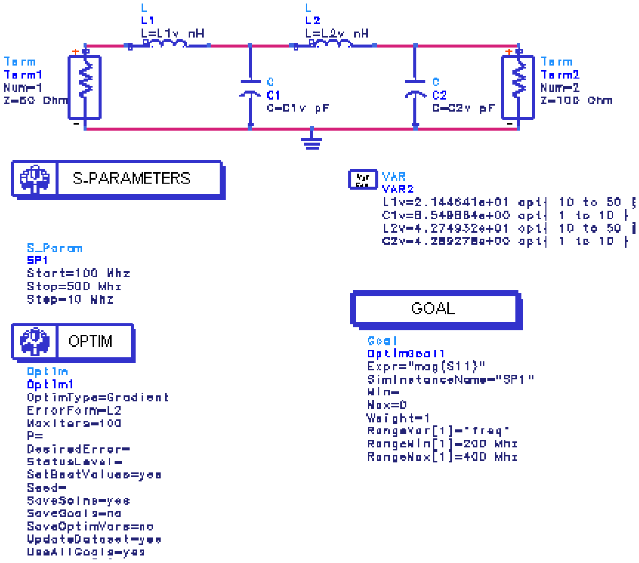
Notice that the response, although much improved over the initial response, is still not quite optimal. The return loss (s11 in dB) is less than 20 dB at 200 MHz (the low-frequency end of the passband), about 23 dB at 400 MHz (the high-frequency end of the passband), and about 26 dB near the band center. An optimal design would have equal values of return loss at these three frequencies.

Note ADS names optimization variables in a specific way to avoid confusion in the dataset or Status window, as follows:

<enclosing_definition_name>.<instance_name>.variable_name.

For more information on this naming convention, refer to "How ADS Names Optimization Variables" on page 2-36.

6. The next step is to select *Simulate > Update Optimization Values* to see the values that the Gradient optimizer has arrived at, as shown in Figure 2-8.



```
VAR2
L1v=2.144641e+01 opt{ 10 to 50 }
C1v=8.540884e+00 opt{ 1 to 10 }
L2v=4.274932e+01 opt{ 10 to 50 }
C2v=4.289278e+00 opt{ 1 to 10 }

S_Param
SP1
Start=100 Mhz
Stop=500 Mhz
Step=10 Mhz

Goal
OptimGoal1
Expr="mag(S11)"
SimInstanceName="SP1"
Min=
Max=0
Weight=1
RangeVar[1]="freq"
RangeMin[1]=200 Mhz
RangeMax[1]=400 Mhz

Optim
Optim1
OptimType=Gradient
ErrorForm=L2
MaxIters=100
P=
DesiredError=
StatusLevel=
SetBestValues=yes
Seed=
SaveSolns=yes
SaveGoals=no
SaveOptimVars=no
UpdateDataset=yes
UseAllGoals=yes
```

Figure 2-8. Schematic Showing Optimized Parameter Values

7. Although the design shown in Figure 2-8 is not optimal, it does provide a good starting point for further optimization. An optimal response would have equal values of return loss at the band edges and at the band center (an *equal-ripple* response). The *Minimax* optimizer, using a method described in the section “Error Function Formulation” on page 6-8 in Chapter 5, Summary of Optimizers is designed to achieve such a response.

Figure 2-9 shows the results of selecting 100 iterations of the Minimax optimizer on this circuit. The optimizer terminated after 8 iterations.

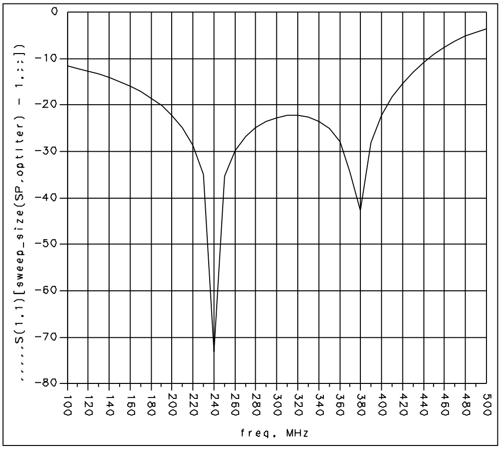
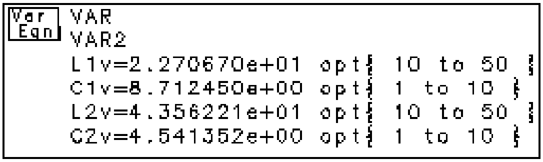


Figure 2-9. Results Using Minimax Optimizer

8. Notice in Figure 2-9 that the values of the return loss at the band edges and at the band center are now equal: about 22.5 dB. This is the *equal-ripple* response, which is optimal for the design. Selecting *Simulate* > *Update Optimization Values* causes the design to be updated with the optimal design values in the Var/Eqn component, as shown in Figure 2-10.



```
VAR
VAR2
L1v=2.270670e+01 opt{ 10 to 50 }
C1v=8.712450e+00 opt{ 1 to 10 }
L2v=4.356221e+01 opt{ 10 to 50 }
C2v=4.541352e+00 opt{ 1 to 10 }
```

Figure 2-10. Var/Eqn Component Showing Optimal Design Values

Discrete Optimization Example

The previous nominal optimization example was based on continuous-value optimization. Advanced Design System also allows you to perform nominal optimization in which the results are limited to real-world, discrete-value parts. This example uses the same project, *optex1_prj*, as the previous example, but we will modify it for discrete optimization. Most of the discrete optimization procedure is the same as other types of nominal optimization, so we will only describe the differences here.

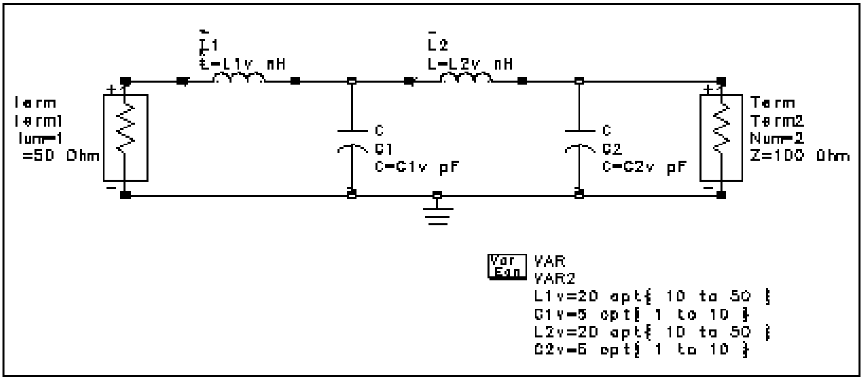


Figure 2-11. Impedance Matching Transformer Before Discrete Optimization

There are several methods to set up discrete optimization. We will change the *optex1_prj* example to learn how to do two of them by:

- Replacing inductor L2 with an SMT inductor from the RF Passive SMT Library and setting it up for discrete optimization. This step illustrates the use of Vendor Component Libraries with discrete optimization. Here values are based on the manufacturer's standard values.
- Changing capacitor C2v in the VAR component to be used for discrete optimization. This step illustrates the use of a component referenced in a VAR with discrete optimization. Here you specify the range and step size.
- Changing the Nominal Optimization component from *Gradient* to *Random*. Discrete optimization is compatible *only* with the Random, Random Minimax, Random Max, Discrete, and Genetic optimization types.

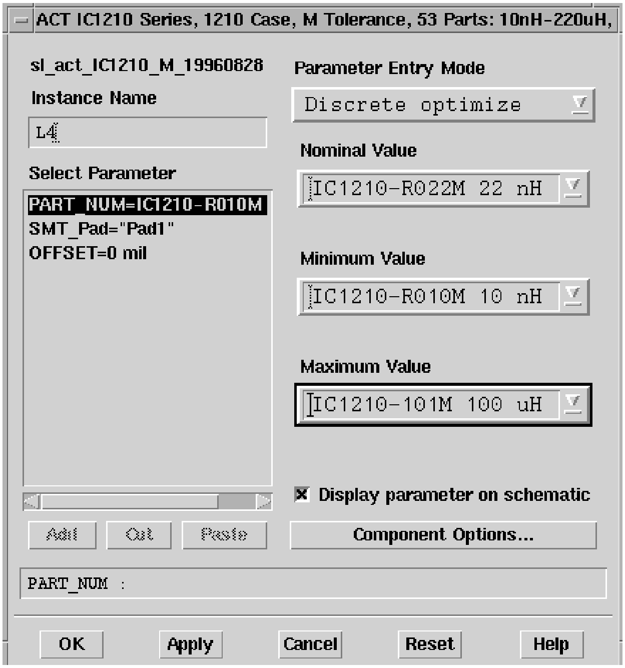
A third method uses a user-defined data file to set up discrete optimization and is described at the end of this section. Refer to "Performing Discrete Optimization Using a Data File" on page 2-25 for details.

Setting Up Inductor L2 for Discrete Optimization

To set up an SMT inductor for discrete optimization:

1. Delete inductor L2.
2. Choose **Insert** > **Component** > **Component Library**. The Component Library window is displayed.

3. From the Libraries list, select the **RF Passive SMT Library**. Then select **SMT Inductor** (a sub-type under RF Passive SMT Library).
4. From the Components list on the right, select part **sl_act_IC 1210_M_19960828**.
5. Place the new part and wire it in to the design.
6. Double-click the inductor to bring up the Component Parameter dialog box.
7. Choose the **Parameter Entry Mode** drop-down list button and select **Discrete optimize** from the list.
8. Three fields appear under Discrete optimize for Nominal, Minimum, and Maximum values. Each field has a drop-down list with the inductor series associated with the inductor family we placed. Select:
   - The **22 nH** part for the Nominal Value
   - The **10 nH** part for the Minimum Value
   - The **100 nH** part for the Maximum Value
9. After selecting these values, the dialog box appears as shown below. Click **OK**.



ACT IC1210 Series, 1210 Case, M Tolerance, 53 Parts: 10nH-220uH,

sl_act_IC1210_M_19960828

Instance Name: L4

Select Parameter:
PART_NUM=IC1210-R010M
SMT_Pad="Pad1"
OFFSET=0 mil

Parameter Entry Mode: Discrete optimize

Nominal Value: IC1210-R022M 22 nH

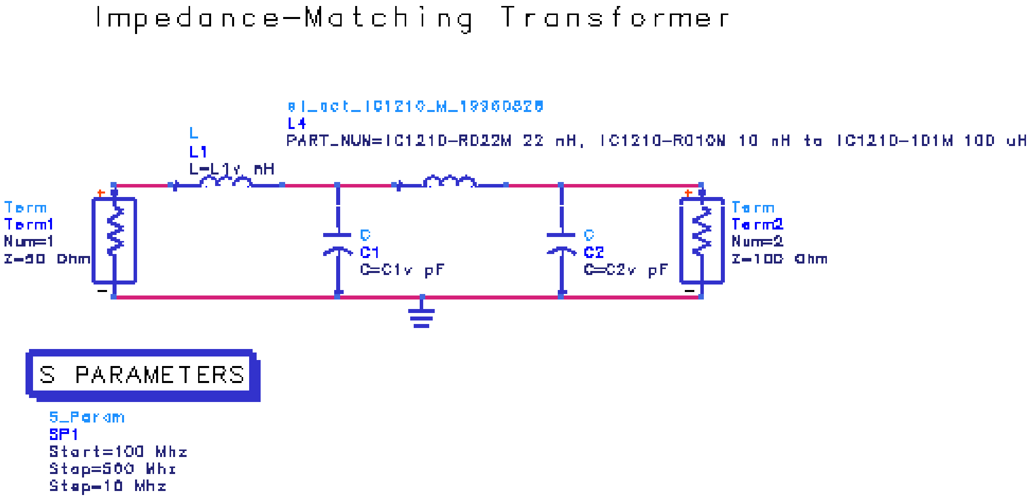
Minimum Value: IC1210-R010M 10 nH

Maximum Value: IC1210-101M 100 uH

☒ Display parameter on schematic

PART_NUM :

At this point, the design appears as below. Note that the annotation has changed for the inductor, to reflect the parts to be used for discrete optimization.

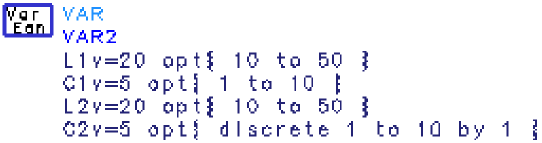


Setting Up Capacitor C2v for Discrete Optimization

To set up capacitor C2v in the VAR component for discrete optimization:

1. Double-click the **VAR** component to bring up the Component Parameter dialog box.
2. Select **C2v** from the Select Parameter list.
3. Choose the Optimization/Statistics/DOE Setup button. A dialog box appears.
4. In the Type field, change **Continuous** to **Discrete**.
5. You then enter the Nominal, Minimum, Maximum, and Step values you want for discrete optimization. For this example, leave the defaults for the first three and enter **1** in the Step Value field.
6. Click **OK** to return to the previous dialog box. Click **OK** again.

The VAR component is updated to show the discrete optimization parameters, as shown below.



```
VAR
VAR2
L1v=20 opt{ 10 to 50 }
C1v=5 opt{ 1 to 10 }
L2v=20 opt{ 10 to 50 }
C2v=5 opt{ discrete 1 to 10 by 1 }
```

Setting Up the Nominal Optimization Component

To set up the Nominal Optimization component for discrete optimization:

1. Double-click the **Nominal Optimization** component to bring up the Component Parameter dialog box.
2. In the Optimization type drop-down list, change **Gradient** to **Random**.
3. Click OK.

This example is now ready for discrete optimization.

Simulation Results

After the design is simulated, the inductor will be updated for a value to 39 nH, based on the goal criteria we specified. The schematic will be updated only if you select *Simulate* > *Update Optimization Values* from the Schematic window.

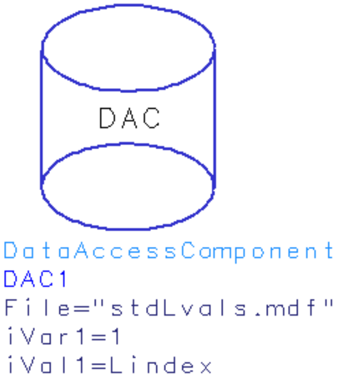
Performing Discrete Optimization Using a Data File

A third method to set up discrete optimization employs a user-defined data file that contains a list of the discrete values. This file is referenced by placing a Data Access Component (DAC) on your schematic. This method is desirable when you have a long list of values for a part that is not found in the Advanced Design System Vendor Component Libraries.

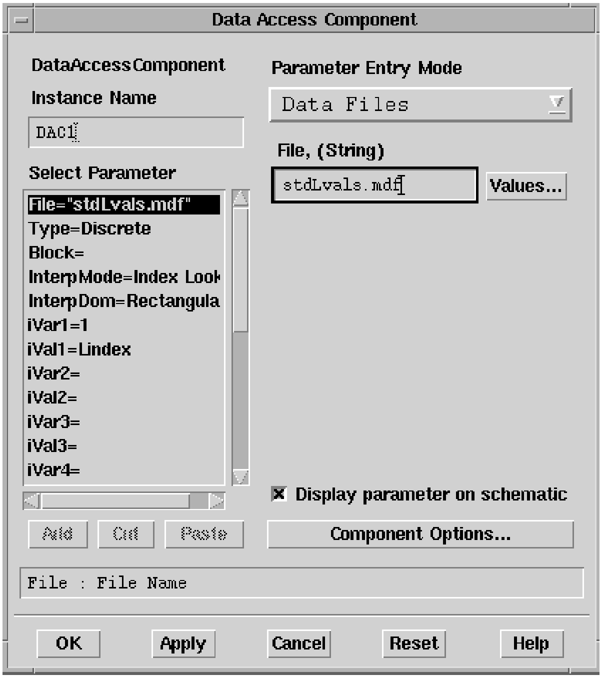
Note The Data Access Component is only available from an Analog/RF Systems schematic. It is not available for use with Signal Processing in the present release.

To specify parameters for discrete optimization using a data file:

1. Create your data file with an editor or program suited for this task.
2. Place your data file in the *data* subdirectory of the desired project. (Other directories are allowed but their paths must be specified.) The format of this file is described in the next section, "Understanding the Discrete Data File" on page 2-29.
3. Place a DAC component in the Schematic window by choosing **Insert** > **Component** > **Component Library** > **Data Items** > **DataAccessComponent** (or choose a DAC from the Data Items palette) and placing it in your design. An example is shown below.



4. Double-click the component to bring up the Component Parameter dialog box.
5. In the Select Parameter list on the left, set the **File** parameter to refer to your data file. In our example, the file is called *stdLvals.mdf*, as shown below.



DataAccessComponent

Instance Name: DAC1

Parameter Entry Mode: Data Files

File, (String): stdLvals.mdf

Select Parameter

File="stdLvals.mdf"
Type=Discrete
Block=
InterpMode=Index Look
InterpDom=Rectangula
iVar1=1
iVal1=Lindex
iVar2=
iVal2=
iVar3=
iVal3=
iVar4=

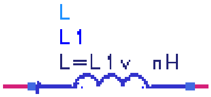
☒ Display parameter on schematic

File : File Name

There are a number of other parameters in the Select Parameter list. Only a few are needed for our example.

6. Accept the default for the Type parameter, which is *Discrete*. Also accept the defaults for InterpMode (Index Lookup) and InterpDom.
7. Below these are pairs of parameters, called iVar1 and iVal1, iVar2 and iVal2, iVar3 and iVal3, etc. These mean independent variable 1 and independent value 1, independent variable 2 and independent value 2, etc. Each pair tells the software to refer to a row number in your data file for each variable. In our example, the only variable is inductance, so we only need to specify iVar1 and iVal1.
8. Set iVar1 to **1** (for the first independent variable).
9. Set iVal1 to **Lindex** (for inductance index; this is a user-defined label).
10. Choose **OK** to dismiss the dialog box.

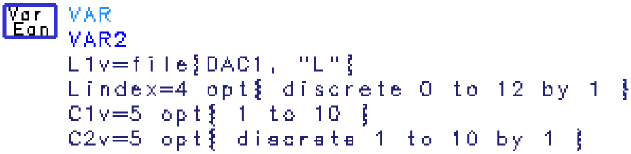
11. In the Schematic window, place or edit your discrete-valued component. For this example, we will use inductor L1 from the previous example. Also as in the previous example, we will reference the VAR component to specify the parameter, as shown below.



12. Double-click the VAR component to edit its parameters.
13. For parameter L1v:
    - Choose **File Based** in the Variable or Equation Entry Mode list box.
    - Enter **L1v** in the Name field.
    - Accept **DAC1** for the Data Access Component Id.
    - Enter **"L"** in the Dependent Parameter Name field.
14. For parameter Lindex:
    - This is the parameter (inductance) that we will set up for discrete optimization.
    - Choose **Standard** in the Variable or Equation Entry Mode list box.
    - Enter **Lindex** in the Name field.
    - The Variable Value field will be used for the nominal value set in the next step. (It can be ignored for now.)
15. Now choose the **Optimization/Statistics/DOE Setup** button. The Setup dialog box appears.
16. This dialog box is set up like any other for discrete optimization.
    - The Optimization Status should be **Enabled**.
    - The Type field should be **Discrete**.
    - For this example, we will set the Nominal Value to **4**, the Minimum Value to **0**, the Maximum Value to **12**, and the Step Value to **1**.

- What this means for a file-based discrete optimization is that inductor values from row 0 to row 12 will be used and the step will be one row. (If the Step Value were set to 2, the simulator would skip every other row.) These rows correspond to inductor values of 22, 10, and 100 in our example data file described in the next section.
- Choose OK to dismiss the Setup dialog box.

17. Choose OK again to complete the setup of the Variables and equations parameter dialog box. At this point the Schematic window will appear as shown below for the VAR component. Note how the annotation appears for parameters L1v and Lindex.



```
VAR
VAR2
L1v=file{DAC1, "L"}
Lindex=4 opt{ discrete 0 to 12 by 1 }
C1v=5 opt{ 1 to 10 }
C2v=5 opt{ discrete 1 to 10 by 1 }
```

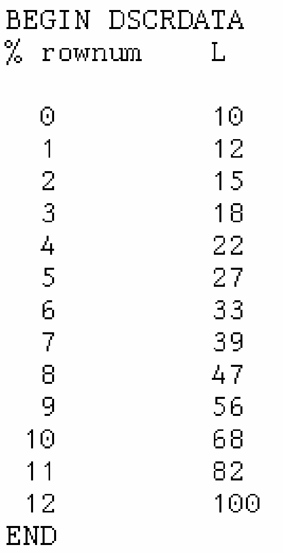
Understanding the Discrete Data File

It’s important for you to understand the DAC data file format when building discrete data files.

The DAC data file consists of a matrix of data arranged in rows and columns. Each row represents a different possible component value or part number. Each column represents a different parameter or characteristic of your component. At the top of each column is a name used to identify that column in the file. These column names may be any alphanumeric string defined by the user. Refer to the following example.

DAC Inductor Data File Example

Below is a simple file-based list of discrete-value parts for an inductor.



```
BEGIN DSCRDATA
% rownum    L

  0         10
  1         12
  2         15
  3         18
  4         22
  5         27
  6         33
  7         39
  8         47
  9         56
 10         68
 11         82
 12         100
END
```

Sensitivity Analysis

While sensitivity analysis is not strictly a type of optimization, it is a fundamental element of gradient optimization methods. This feature is documented here as it is selected from the Nominal Optimization dialog box, as are the other optimizers.

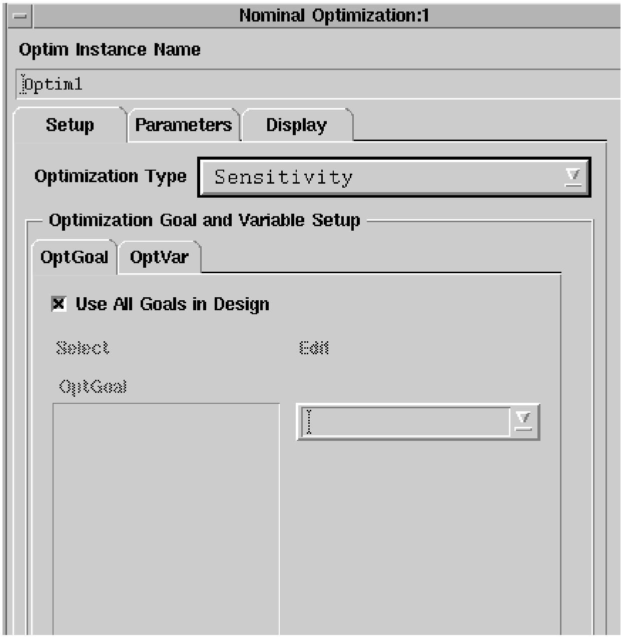
The prerequisites for sensitivity analysis are the same as for any optimization. You must first have:

- At least one component parameter in your design identified as an optimization variable. You specify details in the Component Parameter dialog box by choosing the Optimization/Statistics/DOE Setup button. As with any optimization, choose *Enabled*. Choose *Continuous* for Type. The Format field (such as min/max) has no affect with sensitivity analysis. For more information, about specifying component parameters, refer to “Specifying Component Parameters for Optimization” on page 2-2.

- At least one optimization goal component (*Goal*) placed in the Schematic window. For more information, refer to "Setting Optimization Goals" on page 2-6.

To perform a sensitivity analysis:

1. Place a Nominal Optimization (Optim) component in a Schematic window.
2. Double click the component to edit its parameters. the Setup tab is active.:



Optim Instance Name: Optim1

Optimization Type: Sensitivity

☒ Use All Goals in Design

3. In the Optimization Type field, select **Sensitivity** from the drop-down list.

Sensitivity analysis comprises a single-point or infinitesimal sensitivity analysis of a design variable. For circuit design, it involves taking partial derivatives of the response with respect to a design variable of interest. It is thought that these numbers can help pinpoint variables that contribute disproportionately to performance variance.

ADS contains other techniques for robust design, including statistical design using both yield analysis and yield optimization. These features are described in Chapter 3, Using Statistical Design.

The method used to compute sensitivities is based on finite difference approximation requiring N+1 circuit simulations, where N is the number of optimization variables.

By choosing Nominal Optimization dialog box > Setup tab > OptVar tab (in the Optimization Goal and Variable Setup box), you can specify a subset of all optimization variables within the design project.

Results are sent to the Status window for immediate feedback without the need to open the Data Display window, by choosing Nominal Optimization dialog box > Parameters tab > Levels box > Status level field and entering a proper annotation level. Sensitivity analysis results are also unconditionally sent to the dataset, and this data can be examined in the Data Display window. The remainder of the Parameters tab fields are inactive with sensitivity analysis.

Sensitivities are approximated as follows:

$$S_{P_i} = \frac{\partial R(P_i)}{\partial P_i} \approx \frac{R\left(P_i^0\right) - R\left(P_i^+\right)}{P_i^0 - P_i^+}$$

where R(P^0) is the response evaluated at the nominal point and R(P_i^+) is the response due to a pertubation in the ith parameter.

Note that in ADS, the response R is actually the expression found in the goal(s) given in the Optim (Nominal Optimization) component performing the sensitivity analysis. Additionally, if a goal contains several simulation points, only the point with the highest sensitivity is reported.

Normalized Sensitivities

The ADS sensitivity analysis feature outputs normalized sensitivities. Normalized sensitivities use the approximate gradient (single-point sensitivity) to predict the percentage change in the response due to a 1% change in the design variable. Normalized sensitivity is defined as:

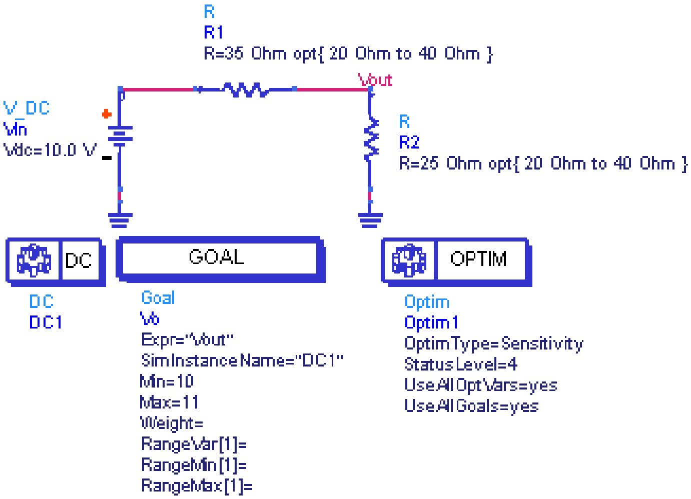
$$S_N(f(x)) = S(f(x))\frac{|x|}{|f(x)|}, \text{ where}$$

$$S(f(x)) = \frac{\partial f(x)}{\partial x}$$

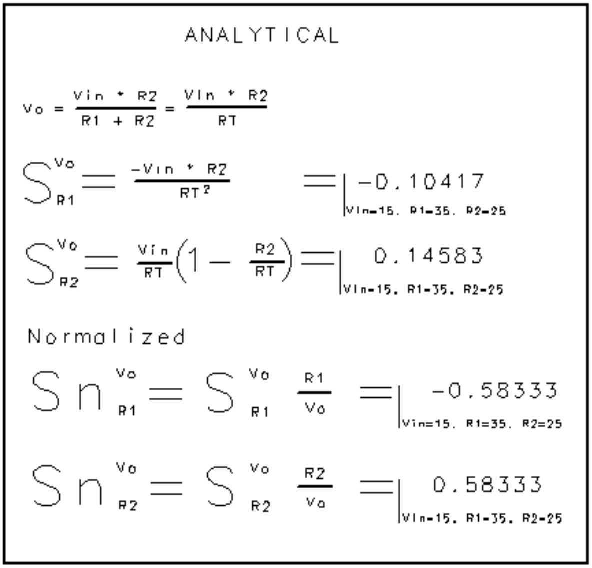
The sensitivity information is saved in the dataset using the Goal instance name. Normalized sensitivities are save with the name: *norm_<goal_instance_name>*.

Sensitivity Analysis Example

The following example will help illustrate the use of sensitivity analysis. This example can be found in ../examples/Tutorial/sensitivity_ex1_prj.



The results of this example are shown below.



ANALYTICAL

$$Vo = \frac{Vin * R2}{R1 + R2} = \frac{Vin * R2}{RT}$$

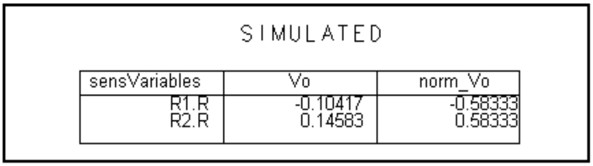
$$S_{R1}^{Vo} = \frac{-Vin * R2}{RT^2} = \Big|_{Vin=15,\ R1=35,\ R2=25} -0.10417$$

$$S_{R2}^{Vo} = \frac{Vin}{RT}\left(1 - \frac{R2}{RT}\right) = \Big|_{Vin=15,\ R1=35,\ R2=25} 0.14583$$

Normalized

$$Sn_{R1}^{Vo} = S_{R1}^{Vo}\ \frac{R1}{Vo} = \Big|_{Vin=15,\ R1=35,\ R2=25} -0.58333$$

$$Sn_{R2}^{Vo} = S_{R2}^{Vo}\ \frac{R2}{Vo} = \Big|_{Vin=15,\ R1=35,\ R2=25} 0.58333$$



SIMULATED

sensVariables	Vo	norm_Vo
R1.R	-0.10417	-0.58333
R2.R	0.14583	0.58333

Swept Optimization

Nominal optimization, yield analysis, yield optimization, and design of experiments (DOE) can all be swept as any other ADS analysis. When these analysis controllers are referenced by a parameter sweep controller, the nominal optimization, yield analysis, yield optimization, or DOE is performed for each value of the sweep variable and the results are output as a function of the sweep variable. Any level of sweep nesting can be used and multiple optimization, yield, yield optimization, or DOE controllers may be referenced at any level.

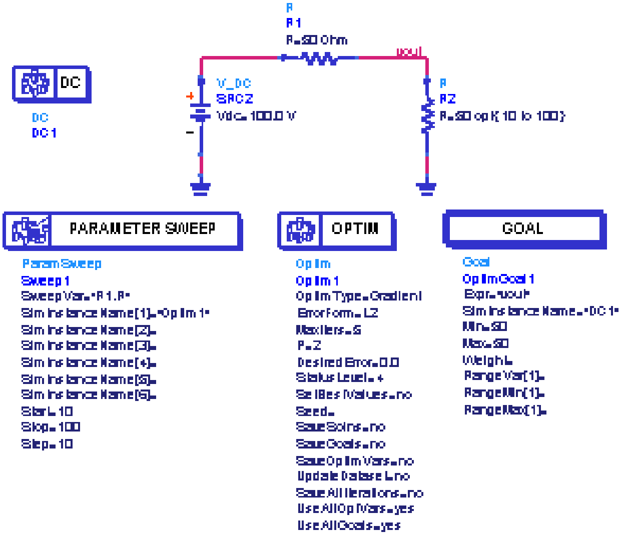
Note that a ParamSweep component can be used to sequence optimization jobs with other analyses. For example, an optimization can reference an S-parameter analysis with fewer frequency points. This makes the optimization more efficient. Then, a follow-on “final analysis” can be specified that uses additional frequency points to display the results of the optimization.

Another application of swept optimization combined with the ParamSweep component is to interleave optimization methods. For example, you might interleave random optimization with gradient optimization in an effort to attain a more robust solution.

The notion of sequential optimization is further enhanced by the ability designate a particular subset of both optimization variables and goals, using the *OptimVar* and *OptimGoal* parameters of the optimization component (Optim). By associating which variables and goals go with a given optimization analysis, you can literally program the optimization approach. For example, to simulate a full-blown tuning/test procedure.

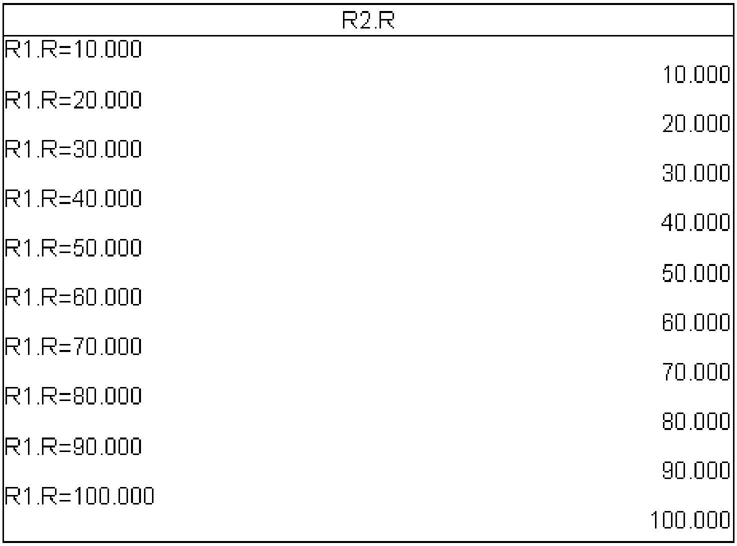
Swept Optimization Example

The following simple schematic and output data will help illustrate this feature. The goal is to optimize R2.R to form a perfect voltage divider for each value of the swept variable R1.R.



In this simulation, note that the *SetBestValues* parameter of the Optim component is set to *No*. With this setup, the optimization at each sweep point resets the value of the optimization variables (R2.R) to the initial nominal value.

After simulation, the results appear as follows:



	R2.R
R1.R=10.000	10.000
R1.R=20.000	20.000
R1.R=30.000	30.000
R1.R=40.000	40.000
R1.R=50.000	50.000
R1.R=60.000	60.000
R1.R=70.000	70.000
R1.R=80.000	80.000
R1.R=90.000	90.000
R1.R=100.000	100.000

Notice how the optimization variable R2.R tracks the swept variable R1.R to form a voltage divider for each sweep point.

How ADS Names Optimization Variables

ADS names optimization (as well as yield and DOE) variables in a specific way to avoid confusion in the dataset or Status window. Ignoring that part of the name introduced by the optimization controller, optimization variable names have the following format:

<enclosing_definition_name>.<instance_name>.variable_name.

For example, you might have a name *Def1.Inst1.X*, where a variable X is defined in the definition Def1.

The instantiation path has the enclosing definition name pre-pended. Optimization input variables may be derived from variables, or derived from component parameters. The former are always associated with the definition and so have no instantiation path. The latter are associated with an instantiation and therefore have an instantiation path, with the exception of parameters where the default value is an optimization input variable. In this case, if the user does not provide a value, the parameter is associated with the definition.

Chapter 3: Using Statistical Design

Statistical design is the process of:

- Accounting for the random (statistical) variations in the parameters of a design.
- Measuring the effects of these variations.
- Modifying the design to minimize these effects.

Yield analysis is the process of varying a set of parameter values, using specified probability distributions, to determine how many possible combinations result in satisfying predetermined performance specifications.

Yield is the unit of measure for statistical design. It is defined as the ratio of the number of designs that pass the performance specifications to the total number of designs that are produced. It may also be thought of as the probability that a given design sample will pass the specifications.

Because the total number of designs produced may be large or unknown, yield is usually measured over a finite number of design samples or *trials* in the process known as *yield estimation*. As the number of trials becomes large, the yield estimate approaches the true design yield. Parameter values that have statistical variations are referred to as *yield variables*.

Two statistical design options are available:

- **Yield analysis** This process involves simulating the design over a given number of trials in which the yield variables have values that vary randomly about their nominal values with specified probability distribution functions. The numbers of passing and failing trials are recorded and these numbers are used to compute an estimate of the yield.
- **Yield optimization** Also known as *design centering*, this process involves multiple yield analyses with the goal of adjusting the yield variable nominal values to maximize the yield estimate. During yield optimization, each yield improvement is referred to as a design iteration.

Yield analysis and yield optimization are supported as follows:

- **Analog/RF Systems** Any analysis type (such as AC, DC, S-Parameter, Harmonic Balance, Circuit Envelope, and Transient simulation types).
- **Signal Processing** For Agilent Ptolemy simulation.

Statistical Design Minimum Requirements

Prior to performing a statistical design, you need:

- At least one component parameter in your design identified as a yield variable. You specify details in the Component Parameter dialog box by choosing the Optimization/Statistics/DOE Setup button.
- At least one yield specification (*YieldSpec*) component specified, then placed in the design window. A yield specification defines a single (or double) sided range of acceptable performance for a given response. For each trial during yield analysis, the yield spec is compared to the simulated response to determine the pass/fail status of the current trial.
- One yield analysis (*Yield*) or yield optimization (*YldOpt*) component placed in the design window to specify all *YieldSpec* components to be included, data to be saved, Shadow model type (if used), enabling of post production tuning, and parameters to be displayed
- One simulation control component (for example, an AC, DC, S-Parameter, Harmonic Balance, Circuit Envelope or Transient component for Analog/RF Systems or a Data Flow Controller for Signal Processing)

The design components needed for yield analysis are located in the *Optim*/*Stat*/*Yield*/*DOE* library or palette for Analog/RF Systems and the *Controllers* library or palette for Signal Processing. They include the following:

- Yield analysis (*Yield*)
- Specification for yield analysis (*YieldSpec*)
- Yield Optimization (*YldOpt*)

Performing Yield Analysis

Yield analysis determines the percentage of acceptable and unacceptable units based on the *YieldSpec* component definitions. Yield analysis randomly varies network parameter values according to statistical distributions while comparing network measurements to the user-specified performance criteria found in the *YieldSpec*.

Yield analysis is based on the Monte Carlo method. A series of trials is run in which random values are assigned to all of your design's statistical variables, a simulation is performed, and the yield specifications are checked against the simulated measurement values. The number of passing and failing simulations is accumulated over the set of trials and used to compute the yield estimate.

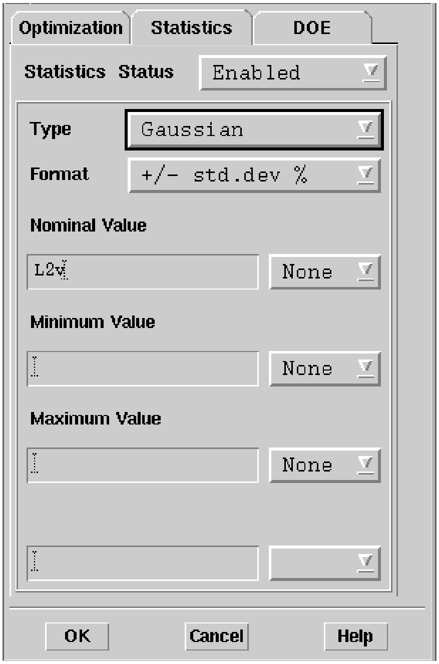
Other capabilities of yield analysis include the following:

- Accumulated sets of selected network responses can be viewed or plotted.
- Performance histograms can be viewed or plotted to display the distribution of measured circuit responses. Refer to "Creating a Measurement Histogram" on page 3-17 for full details. Refer to *Traces* in the *Data Display* manual for details on viewing these histograms.
- Overall performance variation can be assessed as a result of random variations in component parameter values.

Specifying Component Parameters for Yield Analysis

The procedure for specifying components for yield analysis is as follows:

1. Select and place an appropriate component from one of the component palettes or component libraries. For example, place a parallel resistor-inductor-capacitor (PRLC) from the Lumped Components palette.
2. Double-click on the component in the design window to access its associated dialog box.
3. From the dialog box, highlight the parameter that you want to vary in the Select Parameters box (for example R for parallel resistance), then choose the **Optimization/Statistics/DOE Setup** button, which will only appear for valid statistical parameters. The Setup dialog box appears, with the Optimization tab active. Click the **Statistics** tab.



Statistics Status Enabled

Type Gaussian

Format +/- std.dev %

Nominal Value

L2v None

Minimum Value

None

Maximum Value

None

4. From the Statistics Status drop-down list, select **Enabled** so that you can set specification of the appropriate fields. *Enabled* causes the parameter to be varied when the simulation is run. *Disabled* allows you to temporarily suspend any parameter variation previously assigned, and *Clear* removes the values you previously applied to the design.

5. From the Type drop-down list, select an appropriate statistical Value Type from:

   - Gaussian
   - Uniform
   - Discrete
   - LogNormal

   For descriptions of Value Types, refer to the section “Value Types for Statistical Design” on page 5-2 in Chapter 5, Available Value Types.

6. From the Format drop-down list, select an appropriate statistical value format:

- For Gaussian, choose +/- *std.dev. % or +/- std.dev.*
- for Uniform, choose *min/max*, *+/- Delta % or +/- Delta*.
- for Discrete, only *min/max/step* is available.
- for LogNormal, choose +/- *std.dev. %* or +/- *std.dev.*

For complete descriptions of the available format, refer to the section "Value Types for Statistical Design" on page 5-2 in Chapter 5, Available Value Types.

7. If you selected *std.dev* or +/- *std.dev % or +/-Delta* or +/-*Delta %* formats, specify the deviation value. For these formats, the units can also be specified in the drop-down list next to each input field.

Note For a *Gaussian* or *LogNormal* distribution, std.dev refers to the standard deviation (or sigma), either as a percentage of the nominal value or an absolute value. For a *Uniform* distribution, Delta refers to the distribution limit values, either as a percentage of the nominal value or an absolute value.

8. If you selected a *min/max* format, you can optionally enter values for nominal, minimum, and maximum in the appropriate boxes, and select an appropriate unit assignment for each from the drop-down list next to the boxes.

Note Unit specification via the Setup dialog box is not possible for variables defined in the *Var/Eqn* component.

9. From the Nominal Value field and the Units drop-down list, the value and units in your design for this component are displayed. You can change these if you wish.
10. If you intend to include any of the parameters of this component for post production tuning, click the **Optimization** tab and click the Post Production Tuning checkbox. For more details, refer to the section, "Enabling Post Production Tuning" on page 3-13.
11. Choose **OK**.

Placing an Appropriate Simulation Control Component for Yield Analysis

An appropriate simulation control component must be placed in the design prior to initiating a yield analysis.

For Analog/RF Systems simulation, all analysis types are supported, for example place one of the following components:

- **AC** from the AC Simulation palette or library
- **DC** from the DC Simulation palette or library
- **S-Param** from the S-Param Simulation palette or library
- **Harmonic Balance** from the HB Simulation palette or library
- **ENV** from the Envelope Simulation palette or library
- **Tran** from the Transient Simulation library

For Signal Processing, place a:

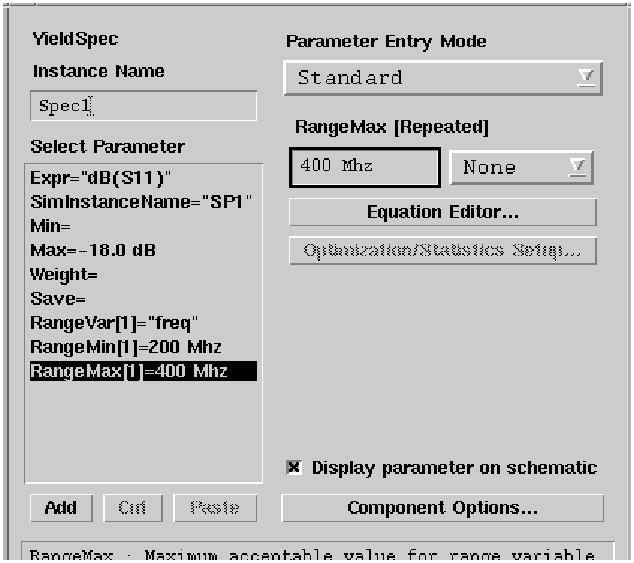
- **Data Flow** controller

For details on specifying parameters for each of these control components, refer to the Advanced Design System *Circuit Simulation* or *Agilent Ptolemy Simulation* manual.

Setting Up a Yield Specification

Yield specifications are defined by placing a *YieldSpec* component (which is accessed from the Optim/Stat/Yield/DOE palette or library for Analog/RF Systems or from the Controllers palette or library for Signal Processing). Once placed, you can double-click it to display the *Specification for Yield Analysis* dialog box.

You can place more than one *YieldSpec* component if needed. The *YieldSpec*s to be used are referenced in the Yield Simulation dialog box, as described in the section, "Setting Job Parameters for Yield Analysis" on page 3-9.



YieldSpec

Instance Name: Spec1

Select Parameter:

```
Expr="dB(S11)"
SimInstanceName="SP1"
Min=
Max=-18.0 dB
Weight=
Save=
RangeVar[1]="freq"
RangeMin[1]=200 Mhz
RangeMax[1]=400 Mhz
```

Parameter Entry Mode: Standard

RangeMax [Repeated]: 400 Mhz, None

☒ Display parameter on schematic

RangeMax : Maximum acceptable value for range variable

To set appropriate yield specifications in this dialog box:

1. If desired, enter a name in the Instance Name field that is different from the assigned default name shown.

2. In the Select Parameter list box on the left, click on each parameter that you want to set up, then make other associated specifications in the box on the right. When you select a parameter, such as Expr, all relevant items in your design will be displayed in the box. The style of this box varies depending on the parameter, as described in the table below.

Table 3-1. Parameters for Yield Spec Components

Parameter	Description	Use Model
Expr	A valid AEL expression that operates on the simulation results, such as mag(S11). For more information on AEL expressions, refer to the AEL manual or the Expressions, Measurements, and Simulation Data Processing manual.	The list box label becomes Measurement Equations. All associated expressions are displayed in the box. Select the one you want to use and it will appear just below in the Selection box. For expressions unrelated to MeasEqns, you must type them in the Selection box.
SimInstanceName	Enter the instance name for the simulation control component that you placed in your design, which will generate the data used by the Expr field.	The list box label becomes Analysis Components. Select the analysis component (simulation controller), such as S-parameter, that you want to use and it will appear just below in the Selection box.
Min	Enter a number for a minimum acceptable response value.	Fields for Parameter Entry Mode and Equation editor are used as in any component parameter dialog box. Type a value in the box. Note: Both Min and Max do not have to be specified, but at least one does.
Max	Enter a number for a maximum acceptable response value.	Same as above.
Weight	Enter a weighting valued to be used in error function calculation. Default is 1.	Fields for Parameter Entry Mode and Equation editor are used as in any component parameter dialog box. Type a value in the box.
RangeVar	Independent variable name.	Same as above, but note that this parameter is “indexable” and can be applied to more than one independent variable.

Table 3-1. Parameters for Yield Spec Components

Parameter	Description	Use Model
RangeMin	Minimum limit of range for independent variable during optimization.	Same as above.
RangeMax	Maximum limit of range for independent variable during optimization.	Same as above.

Setting Job Parameters for Yield Analysis

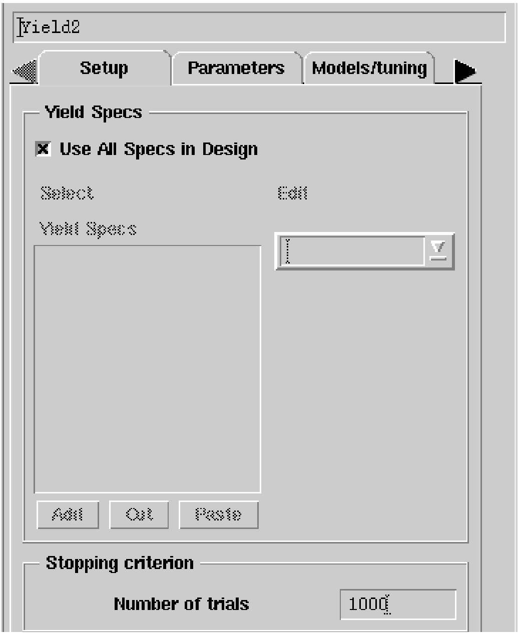
To set job parameters, you need to specify appropriate data in the Yield Simulation dialog box.

This four-tabbed dialog appears when you place a Yield Analysis component (labeled *Yield*). Do the following:

1. Place the *Yield Analysis* component in the appropriate design window.
2. Double-click the component to being up the dialog box. The Setup tab is active.
3. Make specifications in each tab (Setup, Parameters, Models/tuning, and Display) of the dialog box, as described in the next sections.

Selecting a Yield Specification

First select the Setup tab of the Yield Simulation dialog box to set up a yield analysis.



Yield2

☒ Use All Specs in Design

Number of trials 1000

1. In the Yield Specs box, accept the default *Use All Specs in Design* checkbox. This is the best approach for most designs, and all Yield Spec components placed in a design will be implicitly associated with the Yield controller.

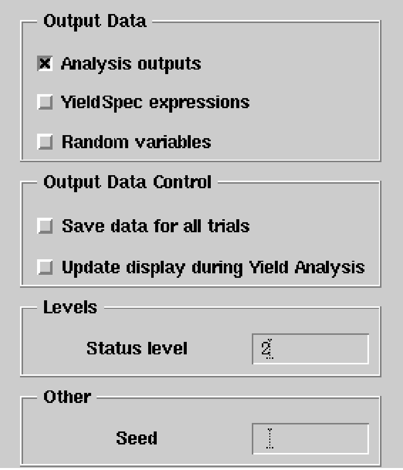
   To associate a *subset* of all Yield Specs with a given Yield controller, deselect the *Use All Specs in Design* checkbox. Select a Yield Spec from the Edit drop-down list, which will include all yield specification components that are currently placed in the design, as described in the section, “Setting Up a Yield Specification” on page 3-6. Choose *Add* to place in the *Yield Specs* box, and repeat this procedure if necessary. Choose the *Cut* or *Paste* buttons, if necessary to make any changes in the *YieldSpecs* box.

2. Under *Stopping criterion*, specify the number of desired trials to use during the yield analysis process.

3. Choose **Apply** to retain the specifications that you have made while you enter data into the Parameters tab, as described in the next section.

Setting Parameter Information

You set parameter information in the Parameters tab of the Yield Simulation dialog box, such as what data to save and parameter attributes. To do this, follow these steps:



Output Data

☒ Analysis outputs

□ YieldSpec expressions

□ Random variables

Output Data Control

□ Save data for all trials

□ Update display during Yield Analysis

Levels

Status level 2

Other

Seed

1. In the Output Data field, specify which data you want to retain in your dataset following yield analysis. Check the following choices that apply.
   - *Analysis outputs* sends all measurements (including measurement equations) to the dataset. This can create a substantial amount of data.
   - *YieldSpec expressions* (default) sends the result of each active YieldSpec's *Expr* field to the dataset.
   - *Random variables* sends the values of all random variables to the dataset for each Monte Carlo trial.
2. In the Output Data Control field, specify whether you want to:
   - *Save data for all trials*. Data for all trials is saved. This can create a substantial amount of data.

Note For yield analysis, enabling this feature can slow the analysis time considerably when many trials are being run. The default is off, where only the first and last trials are saved to the dataset.

- *Update display during Yield Analysis* (default). This updates the dataset on each yield analysis trial so you can see the results in the Data Display window as they occur (instead of waiting to the end where all the traces are displayed at once).

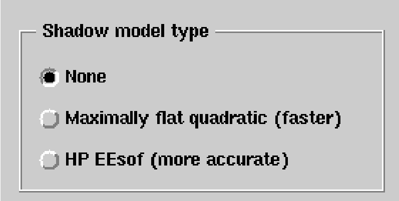
3. In the *Levels* field, enter a number for the desired annotation level. Levels are 0-4, with increasing information displayed in the Status window. (2 is the default.)
4. In the *Other* field, specify a seed value. *Seed* is a value for the random number generator used by the simulator to initiate yield analysis. If *Seed* is not specified, the simulator chooses its own seed, which will be different each time a yield analysis is performed.
5. Choose **Apply** to retain the specifications that you have made while you enter data into the Models/tuning tab, as described in the next section.

Selecting a Shadow Model Type for Yield Analysis

You use the Models/tuning tab of the Yield Simulation dialog box to select the Shadow Model, an optional method of yield analysis.

The Shadow Model works as follows: A series of trials is run in which the random variations in your design's statistical parameters are used in a mathematical model of the design's performance to compute the yield. This allows a greater number of trials and therefore greater accuracy in the yield estimate without a significant increase in the computation time required.

To enable one of two available methods of Shadow Model analysis:



Shadow model type

● None

○ Maximally flat quadratic (faster)

○ HP EEsof (more accurate)

Click the selection of your choice, using either of two methods:

- Maximally flat quadratic Shadow Model (which is usually faster)
- HP EEsof Shadow Model (which is usually more accurate)

If **None** is selected, the Monte Carlo method will be applied to the simulator, not to the mathematical Shadow Model.

Enabling Post Production Tuning

The Models/tuning tab of the Yield Simulation dialog box is also used to enable post production tuning. This section includes a description of the post production tuning feature, followed by a section on setting up post production tuning.

Purpose of Post Production Tuning

During yield analysis, the component parameters that are specified as yield variables are allowed to vary statistically about their nominal values using given probability distribution functions. The yield estimate that results is derived from a given number of trials with these parameter variations.

If, for example, your circuit contained several fixed valued resistors and capacitors with manufactured values known to vary about their nominal values, this yield estimate would take this variation into account.

Suppose that in addition to these fixed valued components, your circuit also contains a trimmer-capacitor and potentiometer that you can tune at the end of production in one last effort to meet specifications. The standard yield estimate would not take this tunability into account.

For such a situation, the simulator can employ *post production tuning*. This feature, combined with yield analysis, allows certain parameter values to receive additional tuning if they first resulted in a failure to meet the yield specifications. The parameters that receive this special treatment are referred to as *post production tunables*.

Setting Up Post Production Tuning

Post production tunables are allowed to take on values over a range as specified in the Optimization tab of the parameter component Setup dialog box. To set up post production tuning:

- First use the Optimization tab of the Component Parameter Setup dialog box (which you access by selecting the Optimization/Statistics/DOE Setup button) to specify the optimization range for the Post Production Tuning parameter, and be sure to check the **Post Production Tuning** checkbox to enable the variable for tuning. Refer to the section “Specifying Component Parameters for

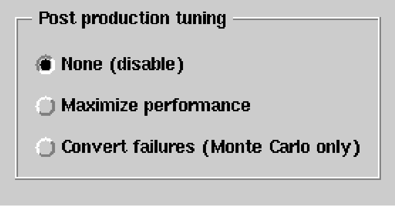
Optimization" on page 2-2 in Chapter 2, Performing Nominal Optimization for more information.

- Remember that a parameter designated for Post Production Tuning must have a yield distribution assigned to it.

Post production tuning is supported for both yield analysis and yield optimization. Note that some trials take longer because of the tuning that must take place in the background.

To set specifications for post production tuning, select one of the following from the Models/tuning tab of the Yield Simulation dialog box:

- **None** (to disable post production tuning)
- **Maximize performance** (unconditionally attempts to find the best performance for each trial)
- **Convert failures** (attempts to convert any failed trial into a passing trial, but does not attempt to tune for maximum performance)



Post production tuning

● None (disable)

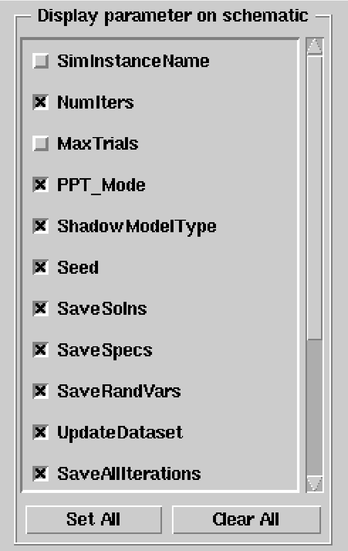
○ Maximize performance

○ Convert failures (Monte Carlo only)

- Choose **Apply** to retain the specifications that you have made while you enter data into the Display tab of the Yield Simulation dialog box, as described in the next section.

Displaying Analysis Data on the Schematic

Selecting the yield analysis parameters that will be displayed on your schematic is done the same way as in nominal optimization. Refer to "Displaying Analysis Data on the Schematic" on page 2-13 for details. Below is a yield analysis example.



Display parameter on schematic

- □ SimInstanceName
- ☒ NumIters
- □ MaxTrials
- ☒ PPT_Mode
- ☒ ShadowModelType
- ☒ Seed
- ☒ SaveSolns
- ☒ SaveSpecs
- ☒ SaveRandVars
- ☒ UpdateDataset
- ☒ SaveAllIterations

When you have finished setting up all the tabs in the Yield Simulation dialog box, click **OK**.

Initiating Yield Analysis

To initiate a yield analysis, select **Simulate** or click the **Simulate** button on the toolbar. The analysis status is displayed in the Status window. Upon completion of the analysis, the simulator ceases analysis and indicates success.

Note If the yield analysis process becomes exceedingly long, you can use the *Stop Simulation* command on the Simulation/Synthesis menu in the status window to interrupt the process.

Swept Yield Analysis

Yield analysis can be swept as any other ADS analysis. When the Yield analysis controller is referenced by a parameter sweep controller, the yield analysis is performed for each value of the sweep variable and the results output as a function of the sweep variable. For details, see "Swept Optimization" on page 2-34 in Chapter 2, Performing Nominal Optimization.

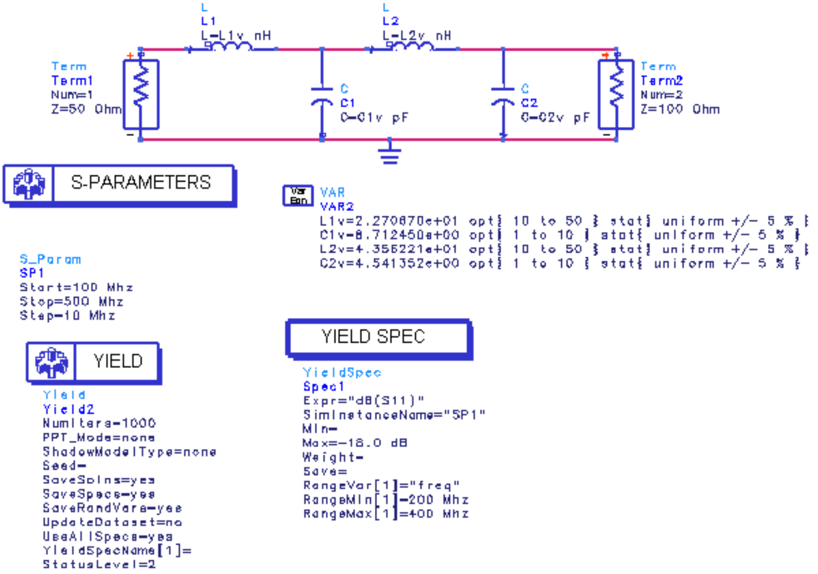
Yield Analysis Example

Note The example used in this chapter is from Analog/RF Systems simulation. However, the yield analysis procedure is the same for Agilent Ptolemy Signal Processing simulation.

For the following yield analysis example, suppose that we start with the octave-band, 2-to-1 matching transformer that we optimized to have an optimal equal-ripple response in the previous example. Now suppose there is a specification on the design that it must have an input return loss of at least 18 dB from 200 MHz to 400 MHz.

This example is called *yldex1_prj*, and it is located in the directory <*installation_dir*>/*examples*/*Tutorial* (where <*installation_dir*> represents your Advanced Design System installation directory). To access this example project and enable simulation, first copy it to a work directory, by choosing *File* > *Copy Project* in the Main window.

1. A *YieldSpec* component is added to the design to define that specification, as shown in Figure 3-1



```
VAR
VAR2
L1v=2.270870e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C1v=8.712450e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
L2v=4.356221e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C2v=4.541352e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
```

```
S_Param
SP1
Start=100 Mhz
Stop=500 Mhz
Step=10 Mhz
```

```
Yield
Yield2
NumIters=1000
PPT_Mode=none
ShadowModelType=none
Seed=
SaveSolns=yes
SaveSpecs=yes
SaveRandVars=yes
UpdateDataset=no
UseAllSpecs=yes
YieldSpecName[1]=
StatusLevel=2
```

```
YieldSpec
Spec1
Expr="dB(S11)"
SimInstanceName="SP1"
Min=
Max=-18.0 dB
Weight=
Save=
RangeVar[1]="freq"
RangeMin[1]=200 Mhz
RangeMax[1]=400 Mhz
```

Figure 3-1. Design Including YieldSpec Component

2. Assume the following:

    - You can obtain inductances of any nominal value between 10 and 50 nH.
    - You can obtain capacitors having any nominal value between 1 and 10 pF.
    - The tolerances on the inductance and capacitance values is +/- 5%.

    This information can be specified in the schematic by changing the Value Types of the appropriate parameters to *Uniform*, and the value type format to +/- *Delta%*.

    The schematic represents the octave-band, equal-ripple, 2-to-1 matching transformer, modified to specify the tolerances on the inductance and capacitance values.

3. A yield analysis of the design indicates about 77% yield in the status window.

Creating a Measurement Histogram

A measurement histogram displays the frequency of occurrence of a selected single scalar measurement function. A histogram is a bar graph in which the height of each bar represents the number or the percent of the recorded measurement values that occurred within each measurement value bin.

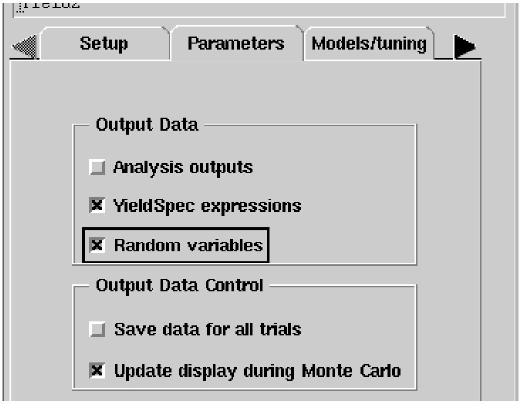
- The scalar measurement value appears on the X-axis.
- The X-axis is divided into bins.
- The Y-axis displays the number or percentage of simulated measurement points that fall into the range of each measurement bin.

You can set the minimum and maximum display range on the X-axis, as well as the number of bins using the histogram AEL expression. Refer to *histogram* in the *Expressions, Measurements, and Simulation Data Processing* manual for more information.

Setup Requirements

Histograms are created after your yield analysis is complete. The stored data is post-processed into histogram form via a set of equations and expressions in the Data Display window. The prerequisites for creating a measurement histogram are:

1. Set up and perform a yield analysis. For the general procedure refer to "Performing Yield Analysis" on page 3-2. However, there are a few specific settings that must be selected to enable histogram generation, as described next.
2. Edit the Yield component to select the Parameters tab of the Yield Simulation dialog box.
3. In the Output Data field, select the **YieldSpec expressions** and **Random variables** check boxes as shown below. This will save the data needed for post-processing.



Output Data

☐ Analysis outputs

☒ YieldSpec expressions

☒ Random variables

Output Data Control

☐ Save data for all trials

☒ Update display during Monte Carlo

4. Click **OK**.
5. Initiate your yield analysis by selecting **Simulate** or click the **Simulate** button on the toolbar.

Generating the Measurement Histogram

To learn how to generate a histogram we will use the same filter design shown in "Design Including YieldSpec Component" on page 3-17 and, in addition, we will use the post-processing capability of the Data Display window. Do the following:

1. After the yield analysis simulation is complete, choose **Window** > **New Data Display**. The Data Display window appears.
2. Choose **Insert** > **Equation** or click the **Eqn** button on the left side of the window.
3. Position the pointer on the display and click. The *Enter Equations* dialog box appears.

Note For more information on entering equations, refer to Chapter 5, Equations in the *Data Display* manual.

4. Type in the following series of equations and expressions, one at a time and click **Apply**. When the last one has been entered, click **OK**. Or, for the finished example, access the data display file *../yldex1_prj/measurement_hist.dds*.

   freq=indep(Spec1)

```
Spec1_freq_low_limit=200 Mhz
Spec1_freq_high_limit=400 Mhz
Spec1_low_index=find_index(freq[0,::], Spec1_freq_low_limit)
Spec1_high_index=find_index(freq[0,::], Spec1_freq_high_limit)
Spec1_subrange=Spec1[::,Spec1_low_index::Spec1_high_index]
Spec1_subrange_freq_collapsed=collapse(Spec1_subrange)
Spec1_perfHist=histogram(Spec1_subrange_freq_collapsed)
```

When these equations and expressions are entered using our example, the following histogram is generated.

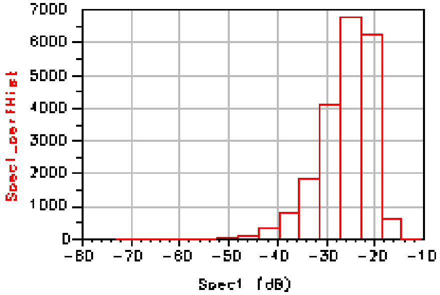


Figure 3-2. Histogram With the Count on the Y-Axis

The design-specific parameters shown above, such as a low-frequency limit of 200 MHz and a high-frequency limit of 400 MHz, only apply to this example. Your design will vary. The idea here is to understand the methodology of entering a series of equations and expressions to generate a histogram from your data.

5. If you want to generate a histogram with percent on the Y-axis instead of the number of outcomes, repeat Step 4, and in addition, type in the following two expressions, one at a time and click **Apply**. When the last one has been entered, click **OK.**

```
num_samples = sweep_size(Spec1_subrange_freq_collapsed)
Spec1_perfHist_normalized = 100 * Spec1_perfHist/num_samples
```

When these equations and expressions are entered using our example, the following histogram is generated.

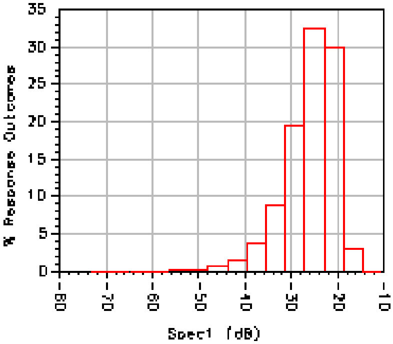


Figure 3-3. Histogram With Percent on the Y-Axis

The following ADS expressions are used in the preceding example:

- Histogram
- Find_index
- yield_sens
- collapse

For more information, see the *Expressions, Measurements, and Simulation Data Processing* manual.

Note ADS names optimization/yield/DOE variables in a specific way to avoid confusion in the dataset or Status window, as follows:

<enclosing_definition_name>.<instantiation_path>.variable_name.

For more information on this naming convention, refer to “How ADS Names Optimization Variables” on page 2-36 in Chapter 2.

Creating a Sensitivity Histogram

A sensitivity histogram displays the sensitivity of a measurement statistical response to a selected statistical variable. See Figure 3-4 for a sample sensitivity histogram.

A sensitivity histogram is a bar graph that shows the effect on a specified statistical response versus the value of a selected statistical variable. [1]

- The X-axis shows the range of value of the selected statistical variable divided into “bins.”
- The Y-axis shows, for each bin, a point estimate of the performance response. This point estimate is the statistical response.

  Possible examples for statistical response are yield, average performance, performance variance, as well as others. Figure 3-4 shows a sensitivity histogram with yield as the statistical response.

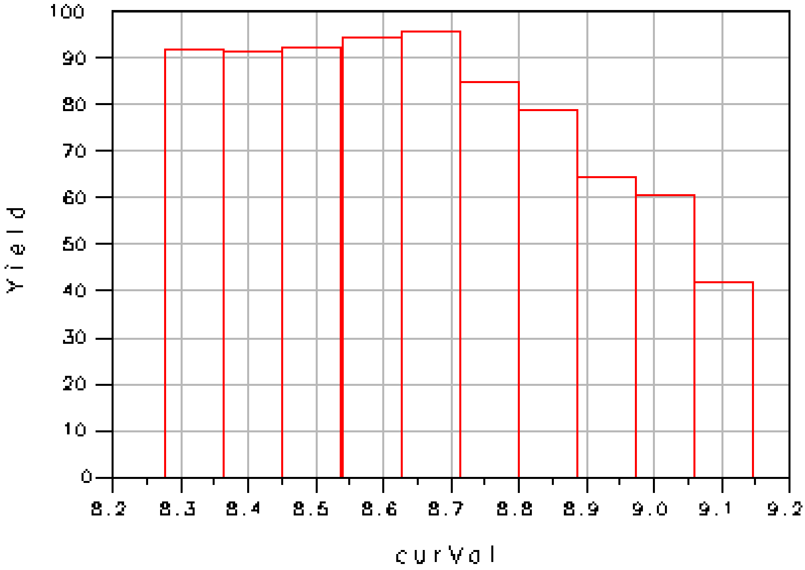


Figure 3-4. Parts of a Sensitivity Histogram Display

[1] Michael D. Meehan and John Purviance, *Yield and Reliability in Microwave Circuit and System Design*, Artech House Inc., 1993.

Setup Requirements

Sensitivity histograms are created the same way as measurement histograms, as explained in the last section. The only difference is the additional equations and expressions entered after your yield analysis is complete. Refer to "Setup Requirements" on page 3-18 for the setup procedure.

Generating the Sensitivity Histogram

Note Generating a sensitivity histogram involves the same steps as generating a measurement histogram, described in the last section, except that additional equations and expressions are required.

To learn how to generate a sensitivity histogram, we will use the same filter design shown in "Design Including YieldSpec Component" on page 3-17 and we will use the post-processing capability of the Data Display window. Do the following:

1. After the yield analysis simulation is complete, choose **Window** > **New Data Display**. The Data Display window appears.
2. Choose **Insert** > **Equation** or click the **Eqn** button on the left side of the window.
3. Position the pointer on the display and click. The *Enter Equations* dialog box appears.

Note For more information on entering equations, refer to Chapter 5, Equations in the *Data Display* manual.

4. Type in the following series of equations and expressions, one at a time and click **Apply**. When the last one has been entered, click **OK**. Or, for the finished example, access the data display file *../yldex1_prj/yield_sens.dds*.

   freq=indep(Spec1)

   Spec1_freq_low_limit=200 Mhz

   Spec1_freq_high_limit=400 Mhz

   Spec1_low_index=find_index(freq[0,::], Spec1_freq_low_limit)

   Spec1_high_index=find_index(freq[0,::], Spec1_freq_high_limit)

   Spec1_subrange=Spec1[::,Spec1_low_index::Spec1_high_index]

   maxS11=max(Spec1_subrange)

```
maxS11_vs_curVar=vs(maxS11, curVar)

fail=0.0

pass=1.0

pf_maxS11=if(maxS11_vs_curVar > curSpec1_value) then fail else pass

yldsens= yield_sens( pf_maxS11)

curVar=C1v

curSpec1_value=-18.0
```

When these equations and expressions are entered using our example, the following sensitivity histogram is generated.

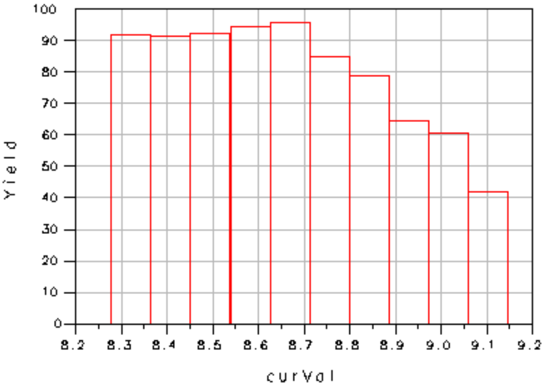


Figure 3-5.
Figure 3-6. Yield Sensitivity Histogram

Notice that the yield is higher for smaller values of C1v. This indicates that yield can be improved by reducing the nominal value of C1v. Also notice that this sensitivity plot includes the effects of all other statistical variables in the design (C2v, L1v, and L2v). In addition, sensitivity plots provide valuable insight into component tolerances and can be used to identify “problem” components.

The design-specific parameters shown above, such as a low-frequency limit of 200 MHz and a high-frequency limit of 400 MHz, only apply to this example. Your design will vary. The idea here is to understand the methodology of entering a series of equations and expressions to generate a histogram from your data.

Performing Yield Optimization

Yield optimization adjusts nominal values of selected element parameters to maximize yield. Also referred to as design centering, yield optimization is the process in which the nominal values of yield variables are adjusted to maximize the yield estimate.

When you activate yield optimization, you are required to enter the number of *design iterations*. This is the number of yield improvements you wish the simulator to obtain. Each design iteration may require several yield analyses (yield estimations).

You are not required to enter the number of trials to be used for each yield analysis. The number of trials is a dynamic variable computed during yield optimization, varying with changing yield estimates and confidence levels. Therefore, the yield estimate derived from yield optimization often differs from that for a single yield analysis with a user-specified number of trials.

To have control over the confidence level and hence the accuracy of the yield estimate, it is recommended that you perform a yield analysis after the yield optimization is completed, using the nominal parameter values obtained from the yield optimization. Choose an appropriate number of trials based upon the following formula, where *N* is the number of trials.

For a 95.4% confidence level $(C_\sigma = 2)$, an Error = ± 2% and a yield of 80%

$$N = \left(\frac{2}{0.02}\right)^2 \bullet 0.8 \bullet (1 - 0.8)$$

N = 1600 trials

For more information, refer to "Monte Carlo Trials and Confidence Levels" on page 7-1 in Chapter 7, Using Monte Carlo Yield Analysis.

Setting Job Parameters for Yield Optimization

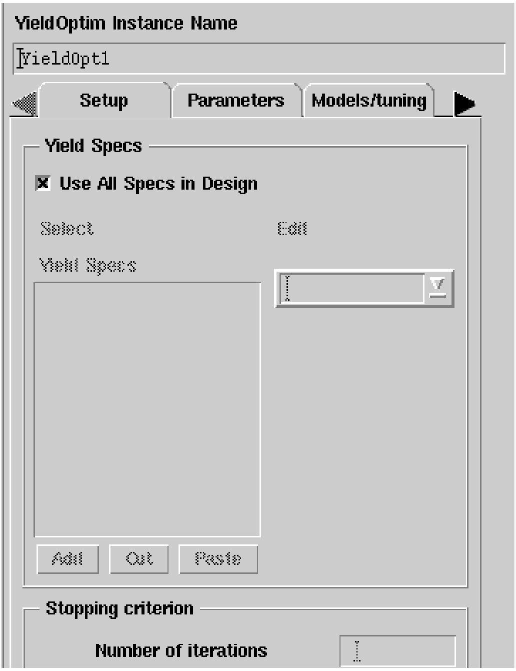
To set job parameters, you need to specify appropriate data in the Yield Optimization dialog box.

This four-tabbed dialog box appears when you place and double-click a Yield Optimization component (labeled *YldOpt*) located in the *Optim/Stat/Yield/DOE* library or palette for Analog/RF Systems and the *Controllers* library or palette for Signal Processing). Do the following:

1. Place the *Yield Optimization* (*YldOpt)* component in the appropriate design window.
2. Double-click the component to bring up the dialog box. The Setup tab is active.
3. Make specifications in each tab (Setup, Parameters, Models/tuning, and Display) of the dialog box, as described in the next sections.

Selecting a Specification for Yield Optimization

To set up a yield optimization in the Setup tab of the Yield Optimization dialog box:



YieldOptim Instance Name: YieldOpt1

☒ Use All Specs in Design

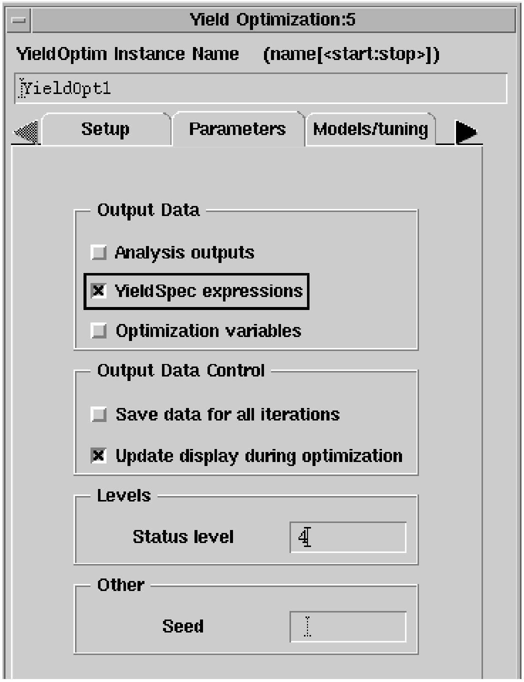
1. In the Yield Specs box, accept the default *Use All Specs in Design* checkbox. This is the best approach for most designs, and all Yield Spec components placed in a design will be implicitly associated with the Yield Optimization controller.

To associate a *subset* of all Specs with a given Yield Optimization controller, deselect the *Use All Specs in Design* checkbox. Select a Yield Spec from the Edit drop-down list, which will include all yield specification components that are currently placed in the design, as described in the section, “Setting Up a Yield Specification” on page 3-6. Choose *Add* to place in the *Yield Specs* box, and repeat this procedure if necessary. Choose the *Cut* or *Paste* buttons, if necessary to make any changes in the *YieldSpecs* box.

2. Under *Stopping criterion*, specify the number of desired iterations to use during the yield optimization process.
3. Choose **Apply** to retain the specifications that you have made while you enter data into the Parameters tab, as described in the next section.

Setting Parameter Information

To set parameter information, such as what data to save and parameter attributes, in the Parameters tab of the Yield Optimization dialog box.



YieldOptim Instance Name (name[<start:stop>])

YieldOpt1

Output Data

☐ Analysis outputs

☒ YieldSpec expressions

☐ Optimization variables

Output Data Control

☐ Save data for all iterations

☒ Update display during optimization

Levels

Status level 4

Other

Seed

1. In the Output Data field, specify which data you want to retain in your dataset following yield optimization. Check the following choices that apply.
   - *Analysis outputs* sends all measurements (including measurement equations) to the dataset for each yield optimization iteration. This can create a substantial amount of data.
   - *YieldSpec expressions* (default) sends the result of each active Yield Spec's *Expr* field to the dataset.
   - *Optimization variables* sends the values of all optimization variables to the dataset for each improvement found during the yield optimization.
2. In the Output Data Control field, specify whether you want to:
   - *Save data for all iterations*. Data for all iterations is saved. This can create a substantial amount of data.

   Note For yield optimization, enabling this feature can slow the simulation time considerably when many iterations are being run. The default is off, where only the first and last iterations are saved to the dataset.

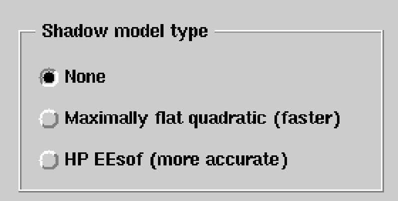
   - *Update display during optimization* (default). This updates the dataset on each optimization iteration so you can see the results in the Data Display window as they occur (instead of waiting to the end where all the traces are displayed at once).
3. In the *Levels* field, enter a number for the desired annotation level. Levels are 0-4 (default is 4), with increasing information displayed in the Status window.
4. In the *Other* field, specify a seed value for use with the random optimizer. *Seed* is a value for the random number generator used to initiate an optimization. If *Seed* is not specified, the simulator chooses its own seed, which will be different each time a yield analysis or yield optimization is performed.
5. Choose **Apply** to retain the specifications that you have made while you enter data into the Models/tuning tab, as described in the next section.

Selecting a Shadow Model Type for Yield Optimization

You use the Models/tuning tab of the Yield Optimization dialog box to select the Shadow Model, an optional method of yield optimization.

The Shadow Model works as follows: A series of trials is run in which the random variations in your design's statistical parameters are used in a mathematical model of the design's performance to compute the yield. This allows a greater number of trials and therefore greater accuracy in the yield estimate without a significant increase in the computation time required.

To enable one of two available methods of Shadow Model analysis:



Shadow model type

● None

○ Maximally flat quadratic (faster)

○ HP EEsof (more accurate)

Click the selection of your choice, using either of two methods:

- Maximally flat quadratic Shadow Model (which is usually faster)
- HP EEsof Shadow Model (which is usually more accurate)

If **None** is selected, the Monte Carlo method will be applied to the simulator, not to the mathematical Shadow Model.

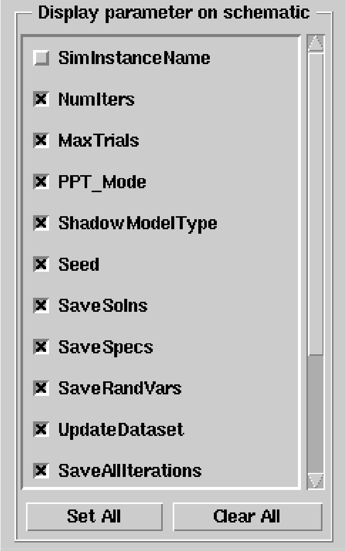
Enabling Post Production Tuning

Post Production Tuning can be used with yield optimization just as in yield analysis. The feature is enabled in the lower part of the Models/tuning tab of the Yield Optimization dialog box. Refer to "Enabling Post Production Tuning" on page 3-13 for a description of this feature.

Choose **Apply** to retain the specifications that you have made while you enter data into the Display tab of the Yield Optimization dialog box, as described in the next section.

Displaying Analysis Data on the Schematic

Selecting the yield optimization parameters that will be displayed on your schematic is done the same way as in nominal (performance) optimization by choosing the Display tab. Refer to "Displaying Analysis Data on the Schematic" on page 2-13 for details. Below is a yield optimization dialog box example.



□ SimInstanceName
☒ NumIters
☒ MaxTrials
☒ PPT_Mode
☒ ShadowModelType
☒ Seed
☒ SaveSolns
☒ SaveSpecs
☒ SaveRandVars
☒ UpdateDataset
☒ SaveAllIterations

When you have finished setting up all the tabs in the Yield Optimization dialog box, click **OK**.

Swept Yield Optimization

Yield Optimization can be swept as any other ADS analysis. When the Yield Optimization controller is referenced by a parameter sweep controller, the yield optimization is performed for each value of the sweep variable and the results output as a function of the sweep variable. For details, see "Swept Optimization" on page 2-34 in Chapter 2, Performing Nominal Optimization.

Yield Optimization Example

This example demonstrates how to optimize the yield of the same octave-based, 2-to-1 matching transformer that was used in the example at the beginning of this chapter. Refer to the section "Yield Analysis Example" on page 3-16.

This example is called *yldoptex1_prj*, and it is located in the directory *<installation_dir>/examples/Tutorial* (where *<installation_dir>* represents your Advanced Design System installation directory). To access this example project and enable simulation, first copy it to a work directory, by choosing *File* > *Copy Project* in the Main window.

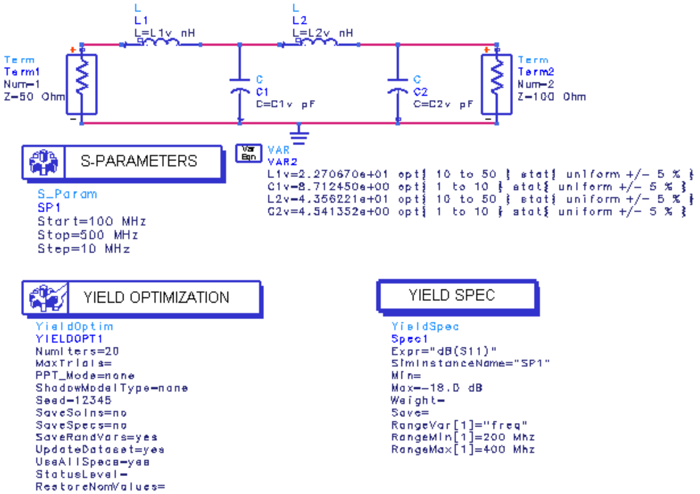
In the Yield Analysis example, the network was optimized to have an equal-ripple response. The yield of the equal-ripple design, assuming ± 5% tolerance on the inductances and capacitances and a minimum return loss of 18 dB, was approximately 77%. This figure is based on using 1000 trials and therefore has a margin or error between ± 2 and ± 3% at a confidence level of 95%.

To optimize the yield of the transformer:

1. From the Schematic window, select **File** > **Open** to open the example *yldoptex0.dsn*, which uses the schematic design from the *yldex0.dsn* example (shown in the section "Yield Analysis Example" on page 3-16), except that the *YldOpt* control component is used instead of the *Yield* component.

   The optimization range for the nominal parameter is identical to that used in the example in the section, "Nominal Optimization Example" on page 2-15 in Chapter 2, Performing Nominal Optimization. Twenty iterations are specified on the *YldOpt* control component.

   The initial design is shown in Figure 3-7. The initial yield analysis, as noted in the section "Yield Analysis Example" on page 3-16, indicated a yield of about 77%.



```
S_Param
SP1
Start=100 MHz
Stop=500 MHz
Step=10 MHz
```

```
VAR
VAR2
L1v=2.270670e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C1v=8.712450e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
L2v=4.356221e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C2v=4.541352e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
```

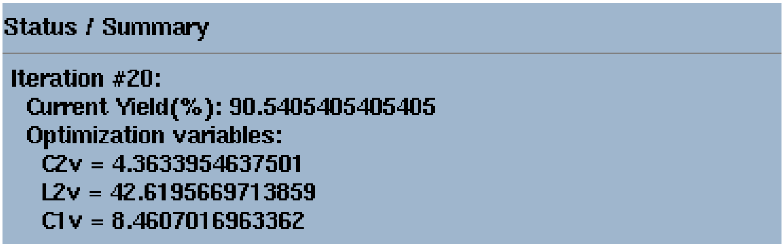
```
YieldOptim
YIELDOPT1
NumIters=20
MaxTrials=
PPT_Mode=none
ShadowModelType=none
Seed=12345
SaveSolns=no
SaveSpecs=no
SaveRandVars=yes
UpdateDataset=yes
UseAllSpecs=yes
StatusLevel=
RestoreNomValues=
```

```
YieldSpec
Spec1
Expr="dB(S11)"
SimInstanceName="SP1"
Min=
Max=-18.0 dB
Weight=
Save=
RangeVar[1]="freq"
RangeMin[1]=200 Mhz
RangeMax[1]=400 Mhz
```

Figure 3-7. .Initial Design Prior to Yield Optimization

2. Select **Simulate** > **Simulate** or click the tool bar **Simulate** button.

   The analysis status in the status window indicates progress by displaying the current iteration number and yield estimate, along with the current optimal nominal component values, as shown in Figure 3-8. Use the scroll bar to view more iterations and variables.

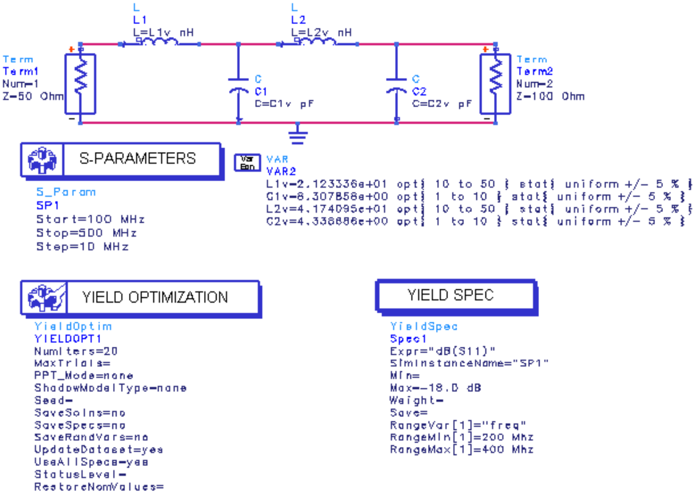


```
Status / Summary

Iteration #20:
  Current Yield(%): 90.5405405405405
  Optimization variables:
    C2v = 4.3633954637501
    L2v = 42.6195669713859
    C1v = 8.4607016963362
```

Figure 3-8. Progress of Yield Optimization

   Upon completion of the simulation, the optimized yield should be more than 90%, indicating less than half as many failures as the equal ripple design.

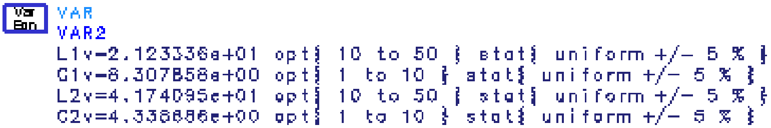
3. Select **Simulate** > **Update Optimization Values** to change the nominal values on the schematic to the optimized ones. The results are shown in Figure 3-9.



```
S_Param
SP1
Start=100 MHz
Stop=500 MHz
Step=10 MHz

VAR
VAR2
L1v=2.123336e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C1v=8.307858e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
L2v=4.174095e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C2v=4.338686e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }

YieldOptim
YIELDOPT1
NumIters=20
MaxTrials=
PPT_Mode=none
ShadowModelType=none
Seed=
SaveSolns=no
SaveSpecs=no
SaveRandVars=no
UpdateDataset=yes
UseAllSpecs=yes
StatusLevel=
RestoreNomValues=

YieldSpec
Spec1
Expr="dB(S11)"
SimInstanceName="SP1"
Min=
Max=-18.0 dB
Weight=
Save=
RangeVar[1]="freq"
RangeMin[1]=200 Mhz
RangeMax[1]=400 Mhz
```

Figure 3-9. Schematic Design with Optimized Values

If we look at the VarEqn component, we see that the optimized values have been updated on the schematic.



```
VAR
VAR2
L1v=2.123336e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C1v=8.307858e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
L2v=4.174095e+01 opt{ 10 to 50 } stat{ uniform +/- 5 % }
C2v=4.338856e+00 opt{ 1 to 10 } stat{ uniform +/- 5 % }
```

Note The values that you obtain if you run through the example may differ slightly from those shown here due to the random nature of the yield optimization algorithm.

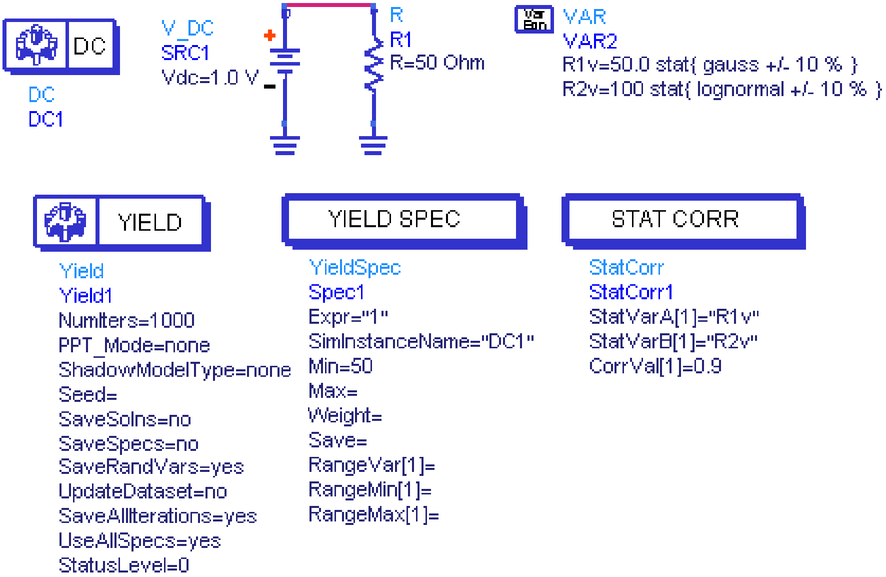
The example shows that an equal-ripple design, having equal distance between the specification and the actual response at both ends and the middle of the frequency band is not optimal when component variations are taken into account. The yield of such a design can be improved dramatically merely by shifting the nominal value by a small amount using yield optimization.

Statistical Correlation

The Statistical Correlation component (Stat Corr) is used in yield analysis and yield optimization. The following schematic illustrates the use of this feature.

Any statistical variable can be correlated to another statistical variable regardless of the distribution of the variables. The correlation coefficient parameter *CorrVal* should be in the range of (-1, 1). If for a system of correlated variables, the correlation matrix is not *positive definite*, a warning message will be displayed and the correlation values altered so that the analysis can proceed.

In this example, a “dummy” yield analysis is used to generate the random outcomes for R1v and R2v. The Statistical Correlation component is used to correlate R1v and R2v at a level of 0.9, which is a strong positive relationship.



DC
DC1

V_DC
SRC1
Vdc=1.0 V

R
R1
R=50 Ohm

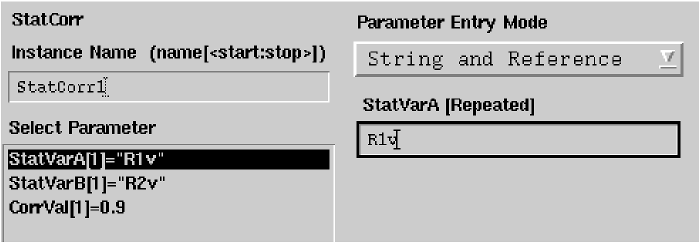
VAR
VAR2
R1v=50.0 stat{ gauss +/- 10 % }
R2v=100 stat{ lognormal +/- 10 % }

Yield
Yield1
NumIters=1000
PPT_Mode=none
ShadowModelType=none
Seed=
SaveSolns=no
SaveSpecs=no
SaveRandVars=yes
UpdateDataset=no
SaveAllIterations=yes
UseAllSpecs=yes
StatusLevel=0

YieldSpec
Spec1
Expr="1"
SimInstanceName="DC1"
Min=50
Max=
Weight=
Save=
RangeVar[1]=
RangeMin[1]=
RangeMax[1]=

StatCorr
StatCorr1
StatVarA[1]="R1v"
StatVarB[1]="R2v"
CorrVal[1]=0.9

The Stat Corr component has only a few parameters as shown in the Statistical Correlation for Yield Analysis dialog box, below:



StatCorr

Instance Name (name[<start:stop>])

StatCorr1

Select Parameter

StatVarA[1]="R1v"
StatVarB[1]="R2v"
CorrVal[1]=0.9

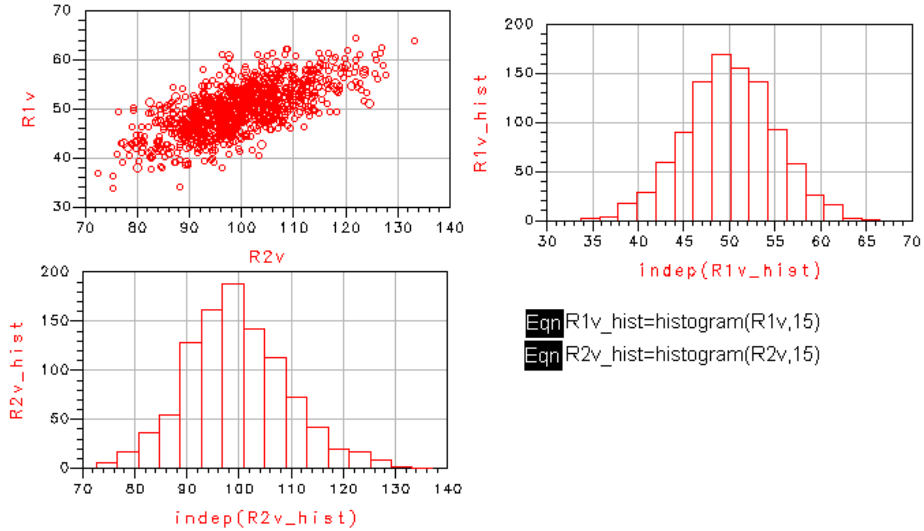
Parameter Entry Mode

String and Reference

StatVarA [Repeated]

R1v

The next figure shows a scatter plot and histograms of both random variables R1v and R2v.



```
R1v_hist=histogram(R1v,15)
R2v_hist=histogram(R2v,15)
```

Note The Stat Corr component is not available on a Signal Processing schematic for the ADS 2001 release.

Chapter 4: Using Design of Experiments (DOE)

Design of experiments (commonly referred to as DOE) is a data-driven technique for robust design. In the early 1900's, DOE was used by agricultural engineers to improve crop yields. Today circuit and system designers are applying the method as a means to the same end—yield improvement.

A typical DOE experiment includes three primary steps:

1. Plan the experiment:
    - Assess the experimental resource budget.
    - Identify the input and response variables.
    - Assign levels (values) to input variables.
2. Perform the experiment and collect response data.
3. Analyze the data using statistical methods.

Sequential application of this methodology can be used to improve the statistical performance of a given circuit or system. Because of an inherent compromise between statistical performance prediction accuracy and the number of input variables, a *screening* experiment is used to identify variables that contribute significantly to performance variation. Next a *refining* experiment can be used to *hone in* on the target statistical response.

DOE and Computer Simulation

In a general application, DOE methods are designed to accommodate errors of the type found in any experiment. But because circuit and system simulators provide identical results for any analysis having the same input values, complexity in setting up, performing, and analyzing experiments is reduced.

Since the computer is being used to perform the experiment, a more complete characterization of input/output relationships can be realized. Finally, since the computer handles the tedious tasks of bookkeeping during the experiment, there is a further reduction in the possibility of human error.

The primary purpose of DOE is to characterize an unknown process. In circuit or system simulation, the unknown process is predicting the response of the design under test (DUT). A simple technique for characterizing a DUT is to perturb each input variable (*factor*) in turn, and to record the resulting output response. However this approach breaks down if the response due to a change in one factor depends on the value of a different factor.

Minimum DOE Requirements

Prior to performing a Design of Experiments, you need:

- At least one component parameter in your design identified as a DOE variable. You specify details in the Component Parameter dialog box by choosing the Optimization/Statistics/DOE Setup button.
- At least one *DOE Goal* component specified, then placed in the design window.
- At least one Design of Experiments (*DOE*) Simulation component specified, then placed in the design window.
- One simulation analysis control component (for example, an AC, DC, S-Parameter, Harmonic Balance, Circuit Envelope or Transient component for Analog/RF Systems).

Note DOE is not available on a Signal Processing schematic for the ADS 2001 release.

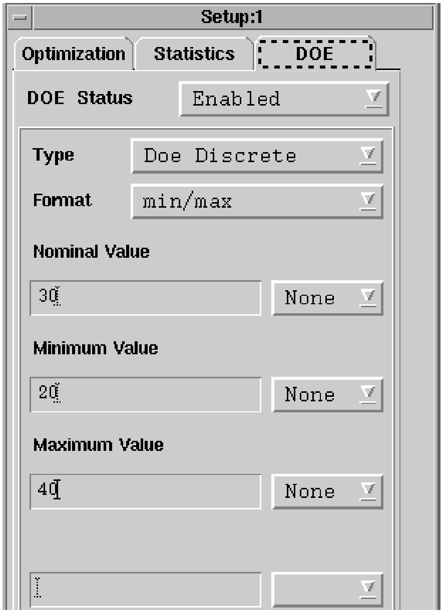
The design components needed for DOE are located in the *Optim/Stat/Yield/DOE* library or palette.

Specifying Component Parameters for DOE

The procedure for specifying components for DOE is as follows:

1. Select and place an appropriate component from one of the component palettes or component libraries. For example, place a parallel resistor-inductor-capacitor (PRLC) from the Lumped Components palette.
2. Double-click on the component in the design window to access its associated dialog box.
3. From the dialog box, highlight the parameter that you want to vary in the Select Parameters box (for example R for parallel resistance), then choose the

Optimization/Statistics/DOE Setup button, which will only appear for valid DOE parameters. The Setup dialog box appears, with the Optimization tab active. Click the **DOE** tab.



DOE Status: Enabled

Type: Doe Discrete

Format: min/max

Nominal Value: 30 None

Minimum Value: 20 None

Maximum Value: 40 None

4. From the DOE Status drop-down list, select **Enabled** so that you can set specification of the appropriate fields. *Enabled* causes the parameter to be varied when the simulation is run. *Disabled* allows you to temporarily suspend any parameter variation previously assigned, and *Clear* removes the values you previously applied to the design.

5. In the Type drop-down list, accept the default DOE Value Type of **DOE Discrete**.

6. From the Format drop-down list, select an appropriate statistical value format:

   - min/max
   - +/-Delta %
   - +/-Delta

   For complete descriptions of the available format, refer to the section “Value Types for DOE” on page 5-3 in Chapter 5, Available Value Types.

7. If you selected *+/-Delta* or *+/-Delta %* formats, specify the deviation value. For these formats, the units can also be specified in the drop-down list next to each input field.
8. If you selected a *min/max* format, you can optionally enter values for nominal, minimum, and maximum in the appropriate boxes, and select an appropriate unit assignment for each from the drop-down list next to the boxes.

Note Unit specification via the Setup dialog box is not possible for variables defined in the *Var/Eqn* component.

9. From the Nominal Value field and the Units drop-down list, the value and units in your design for this component are displayed. You can change these if you wish.
10. Choose **OK**.

Placing an Appropriate Simulation Control Component for DOE

An appropriate simulation control component must be placed in the design prior to initiating a DOE analysis.

For Analog/RF Systems simulation, all analysis types are supported, for example place one of the following components:

- **AC** from the AC Simulation palette or library
- **DC** from the DC Simulation palette or library
- **S-Param** from the S-Param Simulation palette or library
- **Harmonic Balance** from the HB Simulation palette or library
- **ENV** from the Envelope Simulation palette or library
- **Tran** from the Transient Simulation library

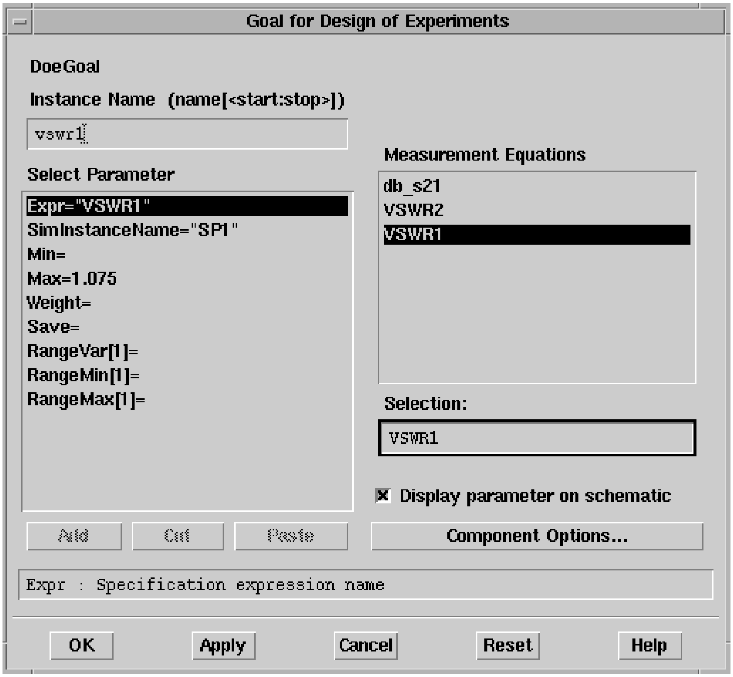
For details on specifying parameters for each of these control components, refer to the Advanced Design System *Circuit Simulation* manual.

Setting DOE Goals

DOE goals are specified by placing a *DOE Goal* component and double-clicking it to display the *Goals for DOE* dialog box. The Goal component can be found as follows:

- For Analog/RF Systems simulation, from the *Optim/Stat/Yield/DOE* palette or library

You can specify and place more than one Goal if needed. The goals to be used are referenced by the DOE component, as described in the later section, "Setting Job Parameters for DOE" on page 4-7. By default, all goals placed apply to all DOE components in a design.



Goal for Design of Experiments

DoeGoal

Instance Name (name[<start:stop>])

vswr1

Select Parameter

Expr="VSWR1"
SimInstanceName="SP1"
Min=
Max=1.075
Weight=
Save=
RangeVar[1]=
RangeMin[1]=
RangeMax[1]=

Measurement Equations

db_s21
VSWR2
VSWR1

Selection:

VSWR1

☒ Display parameter on schematic

Expr : Specification expression name

To set appropriate goal specifications in this dialog box:

1. If desired, enter a name in the Instance Name field that is different from the assigned default name shown.
2. In the Select Parameter list box on the left, click on each parameter that you want to modify, then make other associated changes in the box on the right. When you select a parameter, such as Expr, all relevant items in your design

will be displayed in the box. The style of this box varies depending on the parameter, as described in the table below.

Table 4-1. Parameter Goals for Nominal Optimization

Parameter	Description	Use Model
Expr	A valid AEL expression that operates on the simulation results, such as mag(S11), or the name of a MeasEqn. For more information on AEL expressions, refer to the AEL manual or the Expressions, Measurements, and Simulation Data Processing manual.	The list box label becomes Measurement Equations. All associated expressions are displayed in the box. Select the one you want to analyze and it will appear just below in the Selection box. For expressions not related to MeasEqns, you must type them in the Selection box.
SimInstanceName	Enter the instance name for the simulation control component that you placed in your design, which will generate the data used by the Expr field.	The list box label becomes Analysis Components. Select the analysis component (simulation controller), such as S-parameter, that you want to analyze and it will appear just below in the Selection box.
Min	Enter a number for a minimum acceptable response value.	Fields for Parameter Entry Mode and Equation editor are used as in any component parameter dialog box. Type a value in the box. Note: Both Min and Max do not have to be specified, but at least one does.
Max	Enter a number for a maximum acceptable response value.	Same as above.
Weight	Enter a weighting valued to be used in error function calculation. Default is 1.	Fields for Parameter Entry Mode and Equation editor are used as in any component parameter dialog box. Type a value in the box.
RangeVar	Independent variable name.	Same as above, but note that this parameter is “indexable” and can be applied to more than one independent variable.

Table 4-1. Parameter Goals for Nominal Optimization

Parameter	Description	Use Model
RangeMin	Minimum limit of range for independent variable during optimization.	Same as above.
RangeMax	Maximum limit of range for independent variable during optimization.	Same as above.

Setting Job Parameters for DOE

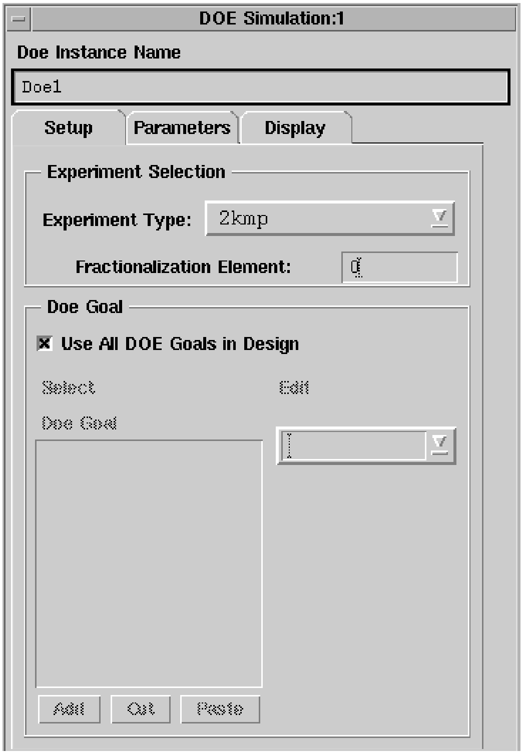
To set job parameters, you need to specify appropriate data in the DOE Simulation dialog box.

This three-tabbed dialog appears when you place a DOE Simulation component (labeled *DOE*). Do the following:

1. Place the *DOE* component in the appropriate Schematic window.
2. Double-click the component to being up the dialog box. The Setup tab is active.
3. Make specifications in each tab (Setup, Parameters, and Display) of the dialog box, as described in the next sections.

Selecting a DOE Specification

First select the Setup tab of the DOE dialog box to set up a DOE analysis.



Doe Instance Name: Doe1

Experiment Type: 2kmp

Fractionalization Element: 0

☒ Use All DOE Goals in Design

1. In the Experiment Selection box, select the desired Experiment Type from the drop-down list. Refer to "DOE Concepts" on page 4-14 for more information on DOE theory and experiment types. The available types are as follows:

Table 4-2. Available DOE Experiment Types

Experiment Type	Description
2kmp	2 raised to the power of k minus p, where k is the number of factors and p is the *fractionalization* element. When p = 0, a full factorial experiment is identified.
Plackett-Burman	Allows the study of k=N-1 variables in N runs, where N is a multiple of 4.

Table 4-2. Available DOE Experiment Types

Experiment Type	Description
CCD	Combines a 2-level experiment with the center point and *star* points along the coordinate axis. Star points lie outside the 2-level experiment and their distance from the center point is a function of the number of factors, i.e., d=$2^{(k/4)}$
Box-Behnken	Consists of the zero point (nominal) and a 2-level, 2-factor factorial design for all combinations of factors, while holding other factors at their nominal value.
3k	A 3-level full factorial experiment.

2. If you select the 2kmp method, enter a *Fractionalization* element in the Fractionalization Element field.

3. In the DOE Goal box, accept the default *Use All DOE Goals in Design* checkbox. This is the best approach for most designs, and all DOE components placed in a design will be implicitly associated with the DOE Goal component.

   To associate a *subset* of all DOE Goals with a given DOE analysis controller, deselect the *Use All DOE Goals in Design* checkbox. Select a DOE spec from the Edit drop-down list, which will include all DOE components that are currently placed in the design. This step is similar to the same procedure for Yield, as described in the section, "Setting Up a Yield Specification" on page 3-6. Choose *Add* to place in the *DOE Goal* box, and repeat this procedure if necessary. Choose the *Cut* or *Paste* buttons, if necessary to make any changes in the *DOE Goal* box.

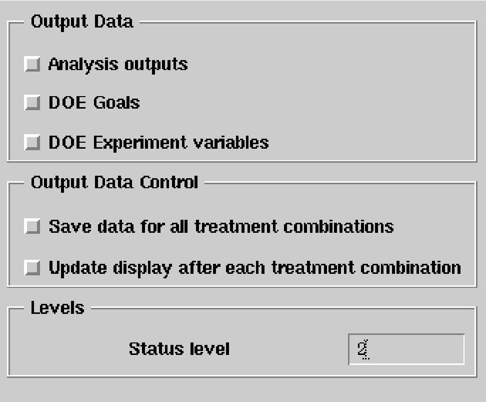
4. Choose **Apply** to retain the specifications that you have made while you enter data into the Parameters tab, as described in the next section.

Setting Parameter Information

You set parameter information in the Parameters tab of the DOE Simulation dialog box, such as what data to save and when the data is output.

During DOE analysis, a complete set of DOE outputs (Pareto, Effects, and Interactions diagrams) are implicitly generated for each DOE Goal component. In addition to the implicitly generated outputs, an ASCII file of the experiment results is created for each DOE analysis component. This file is stored in the */data* subdirectory. You can use this file to input your DOE results into third-party spreadsheet or statistical analysis programs.

To do this, follow these steps:



Output Data

☐ Analysis outputs

☐ DOE Goals

☐ DOE Experiment variables

Output Data Control

☐ Save data for all treatment combinations

☐ Update display after each treatment combination

Levels

Status level 2

1. In the *Output Data* field, specify which data you want to retain in your dataset following DOE analysis.

   - *Analysis outputs* sends all measurements (including measurement equations) to the dataset for each trial. This can create a substantial amount of data.

   - *DOE Goals* sends the result of each Goal's *Expr* field to the dataset for each trial.

   - *DOE Experiment variables* sends the values of all DOE experiment variables to the dataset for each trial.

2. In the *Output Data Control* field, specify whether you want to:

   - *Save data for all treatment combinations.* Data for all treatment combinations is saved. This can create a substantial amount of data.

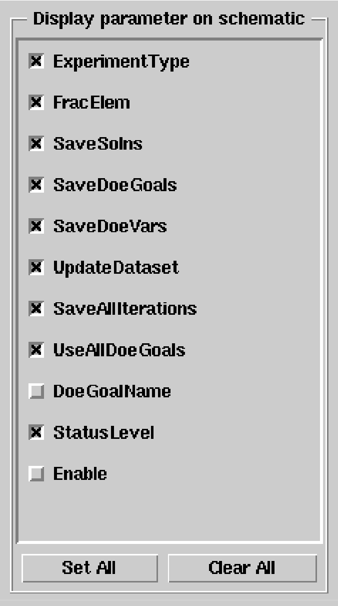
Note For DOE experiments, enabling this feature can slow the analysis time considerably when the experiment is large. The default is off where only the first and last treatment combinations are saved to the dataset.

   - *Update display after each treatment combination* updates the dataset on each DOE treatment combination so you can see the results in the Data Display window as they occur instead of waiting to the end where all the traces are displayed at once.

3. In the *Levels* box, enter a number for the desired annotation level in the Status level field. Levels are 0-4, with increasing information displayed in the Status window. (2 is the default.)
4. Choose **Apply** to retain the specifications that you have made while you enter data into the Display tab, as described in the next section.

Displaying Analysis Data on the Schematic

Selecting the DOE parameters that will be displayed on your schematic is done the same way as in nominal optimization. Refer to "Displaying Analysis Data on the Schematic" on page 2-13 for details. Below is a DOE example.



Display parameter on schematic

☒ ExperimentType
☒ FracElem
☒ SaveSolns
☒ SaveDoeGoals
☒ SaveDoeVars
☒ UpdateDataset
☒ SaveAllIterations
☒ UseAllDoeGoals
☐ DoeGoalName
☒ StatusLevel
☐ Enable

When you have finished setting up all the tabs in the DOE Simulation dialog box, click **OK**.

Initiating Design of Experiments

To initiate a DOE analysis:

1. Choose **Simulate** or click the **Simulate** button on the toolbar. The analysis status, including information about the current treatment combination number as well as result computation progress, is displayed in the Status window. Upon completion of the analysis, the simulator ceases analysis and indicates success.

> **Note** If the DOE analysis process becomes exceedingly long, you can use the *Stop and Release Simulator* command on the Simulate menu to interrupt the process.

2. When the simulation is complete, you are ready to view the DOE output, which is available for each specified DOE goal.

Swept DOE

DOE can be swept as any other ADS analysis. When the DOE controller is referenced by a parameter sweep controller, the DOE is performed for each value of the sweep variable and the results output as a function of the sweep variable. For details, see "Swept Optimization" on page 2-34 in Chapter 2, Performing Nominal Optimization.

DOE Terminology

Following are definitions of the most frequently used *design of experiments* (DOE) terms:

- **Design of experiments** (commonly referred to as DOE). A data-driven technique for robust product design. It is used to improve the statistical performance of a given circuit or system by predicting the response of the device-under-test (DUT).
- **Multilevel experiment.** An experiment with more than 2 levels.
- **Screening experiment.** A screening experiment is used to identify the *significant few* factors that contribute the most to response variation.
- **Refining experiment.** The refining experiment is used to more thoroughly investigate how factors affect the output response. One aim of a refining experiment might be to detect curvature in the factor/response relationship by using a multilevel experiment.
- **Factor.** An input variable.

- **Levels.** Levels represent the values that an input variable will take on during the course of an experiment. For example, for a two-level experiment, variable levels might be assigned to reflect the ±1 standard deviation of the variable value.
- **Response.** An output response due to a particular set of factor level combinations.
- **Design units**. Usually factor levels are encoded such that the maximum and minimum physical values correspond to +1 and -1 respectively. The +1 -1 notation indicates the factor values are in design units, and are obtained from physical values using the following equation for the two-level experiment:

  $$\frac{X - X_{mid}}{(X_{hi} - X_{lo})/2}$$

  where X is the minimum (maximum) physical value of the variable, and X_{lo}, X_{mid}, and X_{hi} are the minimum, middle, and maximum physical values. For example, a capacitor value might be 100pF ±10%, leading to low, mid, and high values of 90, 100, and 110pF, respectively.
- **Interaction**. Any time the factor/response relationship changes as a function of a different factor, there is said to be an interaction between the two factors.
- **Factorial or Full Factorial experiment.** In the factorial experiment, response results are collected for *all* combinations of factor levels.
- **Fractional Factorial.** If the designer can reasonably assume that effects due to high-order interaction terms is negligible, then the information on main and low order interactions can be obtained by running a subset (fraction) of the full factorial experiment. The result is a significant reduction in the amount of work required to obtain the desired information.
- **Main effect**. The main effect for a 2-level experiment is defined as the difference in average response at the two levels of a given factor.
- **Half-effect.** This metric is simply the main effect divided by two.
- **Design matrix**. The design matrix provides a compact representation of an experiment, showing factor level combinations and associated response values tabulated in row-column format.

- **Treatment combination.** An experiment will usually have several treatment combinations (tc) where each one represents a particular set of factor level combinations. A tc is simply a row in the design matrix.
- **Orthogonal design.** There are many ways an experiment can be structured in terms of factor level combinations. If the factor level combinations are such that each column in the design matrix is linearly independent, then the design is said to be orthogonal. In short, for an orthogonal design, the total variation in the response can be decomposed into components due to each factor and interaction. This decomposition makes it possible to rank the importance of factors with respect to their contribution to total performance variance.
- **Aliasing.** The purpose of the fractional factorial experiment is to reduce the overall work required to obtain the desired information about factor/response relationships. To facilitate this reduction in work, effects due to changes in factor levels are added together (aliased) with effects from interactions between factors. As such, a fixed amount of total response variance is attributable to more than one source. Aliasing is sometimes referred to as *confounding*.
- **Saturated experiment.** A saturated experiment is one which provides for the study of k=N-1 variables in N runs.
- **Fractionalization component.** The fractionalization component is representative of the fraction of a full factorial experiment to be used in a given Fractional Factorial experiment. The actual fraction of the full factorial experiment is obtained using the simple formula: $(1/2)^P$, where P is the fractionalization component.
- **Pareto diagrams**. Pareto diagrams are bar charts that show the percentage of the total response variance attributable to each factor and interaction.
- **Effects plots**. Effects plots depict average response values as a function of factor level.
- **Interaction diagrams**. Interaction diagrams indicate how the change in response due to one factor changes with respect to a second factor.

DOE Concepts

Figure 4-1 shows two factors, A and B, and the associated response at various values (*levels*) of the factors.

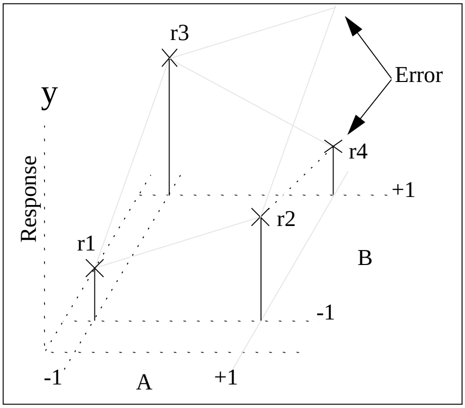


Figure 4-1. Comparison of Responses of Factors A and B

Notice that the factors have two levels: one low (–1) and one high (+1). The ±1 notation indicates the factor values are in *design units*, and are obtained from physical values using the following equation:

$$\frac{X - X_{mid}}{(X_{hi} - X_{lo})/2}$$

where X is the minimum (maximum) physical value of the variable, and X_{lo}, X_{hi}, and X_{mid} are the minimum, middle, and maximum physical values. For example, a capacitor value might be 100pF ± 10%, leading to low, mid, and high values of 90, 100, and 110pF respectively.

If we were to note the change in response due to a change in factor A (from low to high), we would be led to believe that increasing A causes an increase in the response—the same would be observed for factor B. A model from the three response points r1, r2, and r3 can be formulated as the plane surface which contains them:

$$y = [(r2-r1)/2]A + [(r3-r1)/2]B + BIAS$$

where the BIAS term is found by equating the response at a given factor level condition, for example, if A and B are -1, then

$$y = r1 = \frac{r1 - r2}{2} + \frac{r1 - r3}{2} + BIAS$$

This leads to:

$$BIAS = \frac{r2 + r3}{2}$$

However, if either factor A or B is held high, and the same experiment is performed, an inverse relationship exists between factor level and response. The plane surface model from the one-factor-at-a-time experiment would significantly overestimate response r4.

Whenever the factor-response relationship changes as a function of a different factor, there is said to be an *interaction* between the two factors.

To account for interactions, a modification to the simple *one-factor-at-a-time* experiment scheme is necessary. The *factorial* experiment is generally accepted as one of the most efficient methods for characterizing the effects of two or more factors.

In the factorial experiment, response results are collected for *all* combinations of factor levels. For 2-level factorial designs, 2^k data points must be collected for each response, where k is the number of factors. The factorial experiment not only accounts for interactions, but also is formulated using average response values as opposed to the raw ones used in the one-factor-at-a-time method. Figure 4-2a shows four response points r1-r4, as well as the orientation for the plane surface used to model the factor-response relationship.

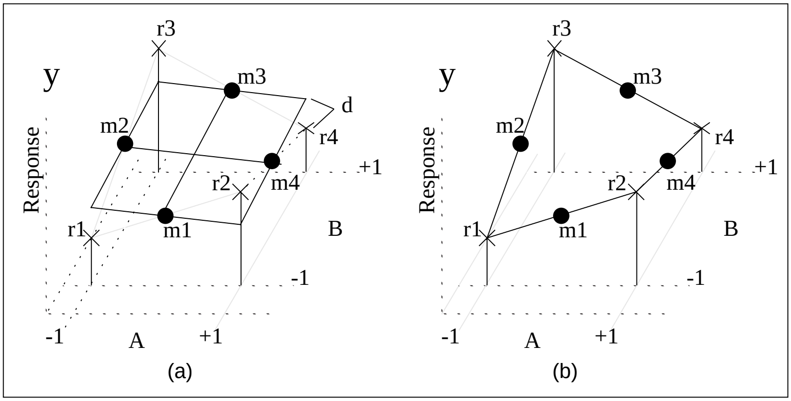


Figure 4-2. (a) Response Points and Plane Surface Orientation, and (b) Modified Surface

The orientation is found by evenly allocating any estimation error due to factor interactions—this error is labelled as *d* in the figure.

The four points on the plane surface m1-m4 represent the average response for the corresponding edge (i.e., m1 = [r1 + r2]/2). Without the interaction term, the two-level factorial model equation is of the form:

$y + S_A A + S_B B + \text{BIAS}$

where S_A and S_B represent the slope of the average response for the given variable, i.e., $S_A = [\text{m4-m2}]/2$. The BIAS term is simply the grand average of all raw response values so that

$$\text{BIAS} = \bar{y} = \frac{[r1 + r2 + r3 + r4]}{4}$$

It turns out that the two lines having endpoints m4, m2, and m3, m1 respectively, intersect in the middle of the plane surface. You can prove this by equating responses in the center of the plane surface, as follows:

$\text{m1} + S_A(1) = \text{m2} + S_B(1).$

To account for the interaction, an additional term must be added to the prediction equation:

$$y = \bar{y} + S_A A + S_B B + S_{AB} AB$$

To find S_{AB}. evaluate the response at one of the response points, for example, r1, and solve for S_{AB}:

$$y = r1 = \bar{y} + S_A(-1) + S_B(-1) + S_{AB}(-1)(-1)$$

so that

$$S_{AB} = S_A + S_B - \bar{y}$$

The solution is:

S_{AB} = [r1 + r4 -r2 -r3]/4

The response depicted in Figure 4-2b shows the modified surface.

Notice that along either factor coordinate axis, the response is linear. However, the slope of the linear model changes as a function of the other factor. For example, with B = –1 the response as a function of A, y(A) indicates a positive slope. But as B increases, the slope of y(A) decreases. Also note that the *off-axis* response contour is quadratic. Figure 4-3 shows the wire mesh plot for the new surface.

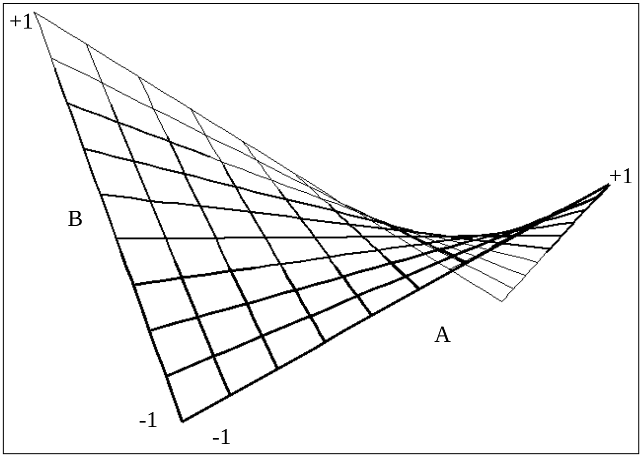


Figure 4-3. Wire Mesh Plot of the Modified Surface of Figure 4-2

The new prediction equation can be restated in terms related more closely to DOE:

$$y = \bar{y} + [ME_A/2]A + [ME_B/2]B + [I_{AB}/2]AB$$

where ME_A and ME_B are, in DOE parlance, the *main effect* of factor A and B respectively. I_{AB} is referred to as the *interaction* between A and B. The main effect for a 2-level experiment is defined as the difference in average response at the two levels of a factor. Referring to Figure 4-2, ME_A is simply m4–m2. The interaction term for the same experiment is defined as half the difference between the main effects of one factor at the two levels of a second factor.

Again using Figure 4-2, I_{AB} can be taken as half the difference in the main effect of factor A when B is high— $ME_A(B+)$ and the main effect of factor A when B is low— $ME_A(B–)$, i.e., $I_{AB} = [(r4–r3) – (r2–r1)]/2$. (For the 2-factor case, $ME_A(B+)$ and $ME_A(B–)$ are simply differences in response values. Usually, these terms will be differences of averages.) When there are more than two factors, it is possible to define *higher order* interaction terms, such as I_{ABC}—half the difference between the two factor interaction effects at the two levels of a third factor, and so on.

There are some additional DOE terms and concepts that are easy to discuss in the context of the 2-level factorial design. First, recall that factor coefficients in the prediction equation were all divided by 1/2. Appropriately there is a DOE term called the *half-effect*, which is simply one half of the main effect. The prediction equation then becomes:

$$y = HE_AA + HE_BB + HE_{AB}AB.$$

Design Matrix

There are many types of experiments that can be applied to any given situation. Differences in these designs are readily seen by examining the *design matrix*. Table 4-3 shows a design matrix for the 2-factor factorial experiment.

Table 4-3. Design matrix for 2-factor factorial experiment

	Factors		**Interactions**	
tc	**A**	**B**	**AB**	**Response**
1	-	-	+	r1
a	+	-	-	r2
b	-	+	-	r3
ab	+	+	+	r4

Notice that columns are delineated into four main groups—tc or *treatment combination*, factors, interactions, and response. Under the tc column, a shorthand is used to indicate the unique conditions of each experiment run (trial). Lower case letters are used to indicate the factor(s) having +1 levels for the trial. (A 1 is used to indicate the run where all factors are held low.) Factor levels are designated using another shorthand where unity is implied in the symbols + and –. The levels of the interaction columns are found by taking the product of the factors involved. The response column is simply a log of the computed response.

Notice that each pair of factor/interaction columns is orthogonal, e.g., linearly independent. This construct allows independent analysis of each factor as well as interactions between factors. In short, for an *orthogonal design*, the total variation in the response can be divided into components due to each factor and interaction, thereby making it possible to rank the importance of factors. (For additional details, refer to the section "DOE References" on page 4-40.)

As a final observation concerning this 2-level, 2-factor design, consider the case where little or no interaction exists between factors A and B. In this situation, the interaction term would be superfluous. But suppose there is a third factor, C, that would be desirable to study. By letting C = AB, a three factor experiment could be conducted in half as many treatment combinations as a factorial experiment. This experimental scheme is referred to as a *fractional factorial* experiment. (Often the factorial experiment is referred to as a *full factorial*.) Obviously significant savings can be afforded by the fractional factorial, but the downside of this approach is that a priori knowledge of the strength of interactions is usually unavailable.

Since the change in the response due to factor C is *aliased* (or *confounded*) with interaction AB, fractional-factorial designs are usually used as screening experiments to identify a small subset of variables that contribute significantly to performance variation. One such screening experiment, the Plackett-Burman design, allows the study of k=N-1 variables in N runs, where N is a multiple of 4. (Any design where k=N-1 is referred to as *saturated*.)

A convenient way to designate all 2-level designs is with the nomenclature 2kmp—which stands for "2 raised to the power of k minus p," where k is the number of factors and p is the so-called *fractionalization component*. With p equal to zero, a full factorial experiment is identified. For the example above, p equals one, which denotes a $(1/2)^P$ or 1/2 fractional factorial, indicating 1/2 the number of tc's of the full factorial are required.

Multilevel Designs

Multilevel designs are characterized as those having more than two levels. These designs are useful in detecting and modeling curvature in the response as a function of the factors. After a screening experiment is performed, and the *vital few* factors are identified, multilevel designs such as those in Figure 4-4 can be used to more accurately predict the response.

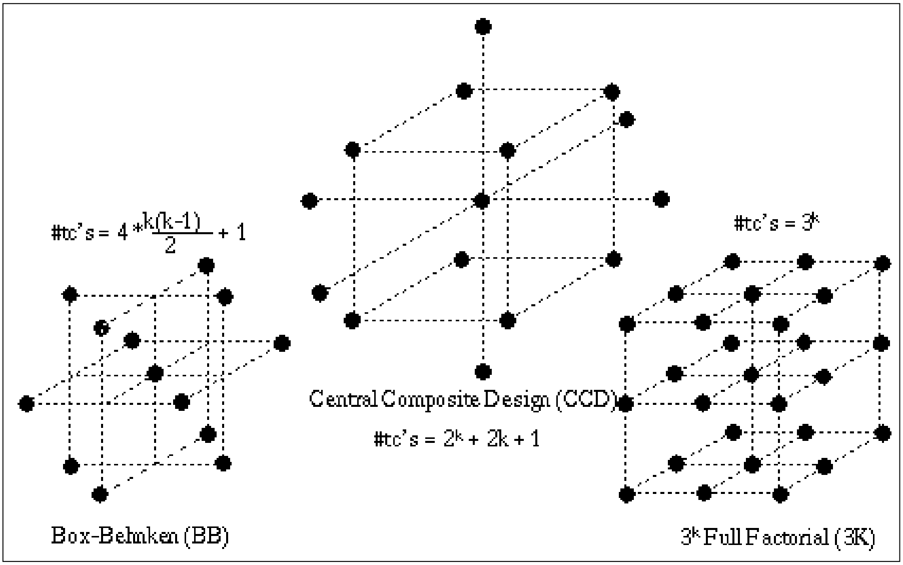


Figure 4-4. Common Multilevel Designs in Three Factors

The Central Composite Design (CCD) combines a 2-level experiment with the center point and so-called star points along the coordinate axis. The star points lie outside of the 2-level experiment and their distance from the center point is a function of the number of factors, i.e., d = $2^{(k/4)}$. The Box-Behnken design consists of the zero point and a 2-level, 2-factor factorial design for all combinations of factors, while holding other factors at their middle value.

DOE Outputs

Prediction equations are not the only output from DOE. In fact, prediction equation(s) are usually obtained by examining *Effects plots*. *Pareto diagrams* are used to rank factors in order of their contribution to the total variance in the response, and *interaction diagrams* allow detection of interactions between factors.

Effects plots are produced by simply computing the overall average response with the factor at each of its levels. For example, in the response shown in Figure 4-5a, the average response for all tc's having A low is 3, and 5 when A is high. The coefficient of the prediction equation (2-level case) comes from the half effect (slope) of the main effect.

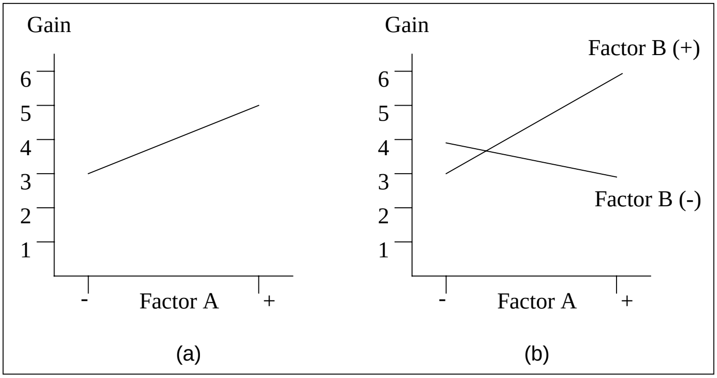


Figure 4-5. a) Effects Plot of Factor A vs. Gain, and b) Interaction Diagram for Factors A and B vs. Gain.

Interaction diagrams are iso-plots of two factors versus the response (see response shown in Figure 4-5b). Once again, average responses are computed over all tc's but this time with one additional constraint due to the companion factor. For example, the average response over all tc's having both A and B at low levels is 4, while it is 6 for the case when both A and B are high. Notice that the slopes of the lines change as a function of B. This indicates an interaction between the factors.

Pareto diagrams are bar charts that show the percentage of the total response variance attributable to each factor and interaction. See Figure 4-6.

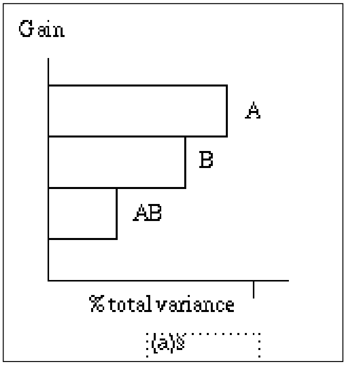


Figure 4-6. Pareto diagram of factors with respect to gain.

Response Values

In a typical industrial application of DOE, it is usually no problem to identify the response variables. In an injection molding application, the response variables might be the number of cracks produced in the casting, or the hardness of the product.

There are relatively few response variables. However, in computer-aided circuit or system design, where a continuous response over frequency, power, or other swept variable is approximated by discrete samples, the number of DOE response variables can number in the hundreds. The amount of data could be overwhelming to the designer.

Currently there are two schools of thought on accommodating response complexity. In the first approach, key points in a frequency/power comb are considered as individual DOE response variables to be analyzed *in parallel*. For example, low, mid, and high band edge samples of gain and noise figure would require six responses to be considered simultaneously.

An alternate approach involves the so-called Taguchi [4] loss-function. The overall DOE response is computed by combining each measurement's loss-function. The loss-function formulation depends on the relational operator used in defining each specification statement.

The ">" and "<" operations are interpreted as *bigger is better* and *smaller is better*, respectively. The equality constraint suggests that the response average be put "on target" with as little variance about the target as possible. The following loss-function formulations are used in the implementation of DOE:

Smaller is better (<): $-10 \text{ Log}_{10}(\text{Sum}(x^2) / n)$

Larger is better (>): $-10 \text{ Log}_{10}(\text{Sum}(1/x^2) / n)$

Target is best (=): $10 \text{ Log}_{10}(M^2/s^2)$, where $s^2 = \text{Sum}(x-M)^2 / n-1$

DOE Basic Example

The following example provides a basic introduction to the use of the software's DOE feature.

Set up the Schematic window as shown in Figure 4-7, or copy it from the ADS Examples: ../examples/Tutorial/doe2_prj.

Note Since this example has already been set up, the steps shown in this section can be followed to learn the general approach to using the DOE feature.

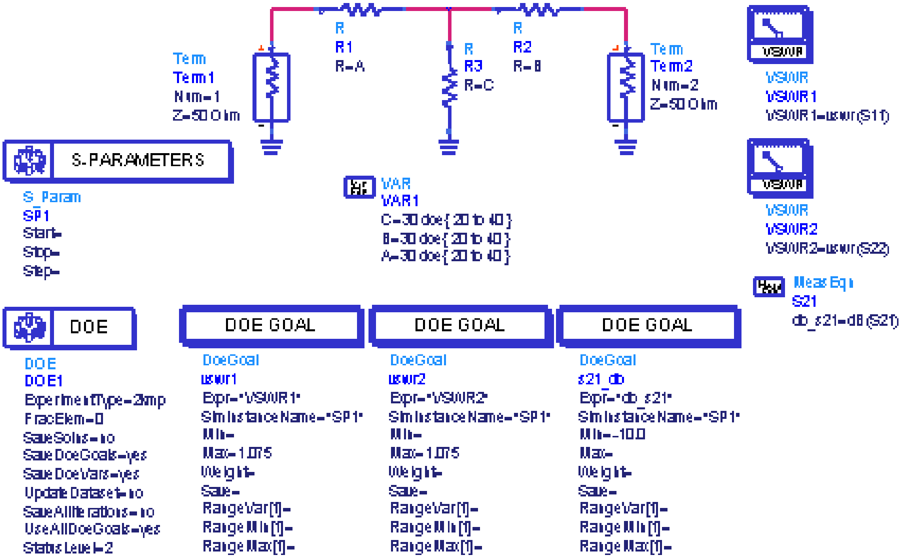
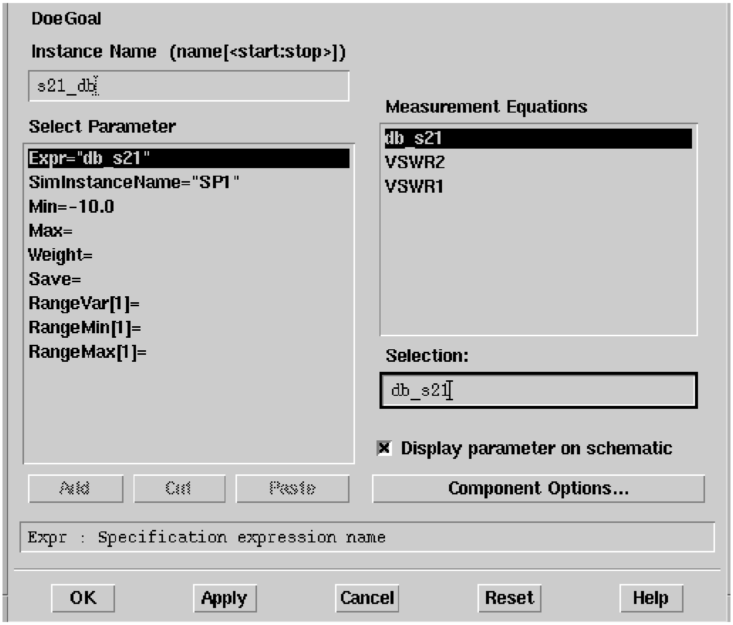


Figure 4-7. Schematic Used as Starting Point for DOE Example

Setup the DOE Goal Components

1. Each DOE Goal will use a prespecified measurement. In this case, VSWR1, VSWR2, and dB_s21. To setup each DOE Goal Component, double click the component. The Goal for Design of Experiment dialog box appears. The Goal for dB_s21 is shown below.



DoeGoal

Instance Name (name[<start:stop>])

s21_db

Select Parameter

Expr="db_s21"
SimInstanceName="SP1"
Min=-10.0
Max=
Weight=
Save=
RangeVar[1]=
RangeMin[1]=
RangeMax[1]=

Measurement Equations

db_s21
VSWR2
VSWR1

Selection:

db_s21

☒ Display parameter on schematic

Expr : Specification expression name

2. In our example, the DOE goal for the S21 measurement is already setup. The steps needed to setup a DOE goal were described in “Setting DOE Goals” on page 4-5, and are similar to setting up optimization goals. In this example, note the following fields:

- The desired measurement is **Expr="db_S21"**.
- The Min and Max fields control the target DOE value. The Min field is set to **-10** and the Max field is left blank, which means that the target value is ≥ -10.
- Leave the Weight field at its default setting (**1**).
- Leave the other fields blank in this example.
- Click **OK** when done.

3. The procedure described in step 2, above, is repeated for the other two DOE goals for the VSWR measurements, by setting the goals as follows:

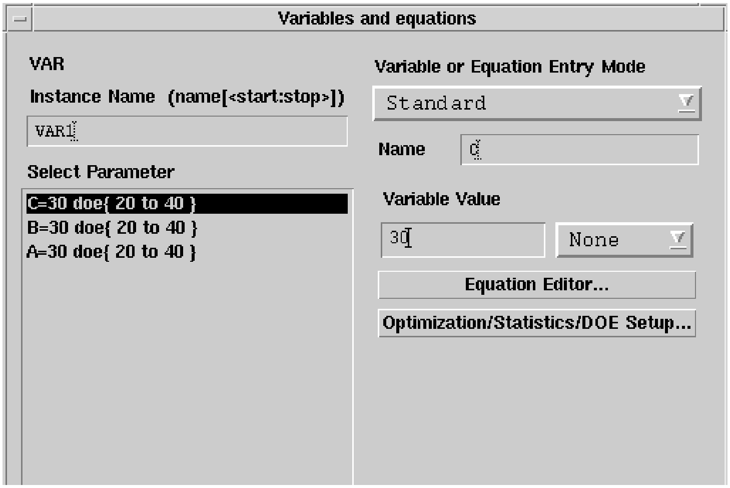
- VSWR1 max = 1.075
- VSWR2 max = 1.075

Note During DOE analysis, a complete set of DOE outputs (Pareto, Effects, and Interaction diagrams) are generated for each DOE goal component. For the case when there are several measurements (perhaps over frequency and/or power), the DOE response is computed by combining each measurement's loss-function. The loss-function, as defined by Taguchi [4] depends on the Min and/or Max fields used in defining each DOE goal. (Refer to the subsection "Response values" under "DOE Outputs" on page 4-21).

If there is only a single measurement and only one sweep point, the DOE response is computed from the actual measurement minus the number in the Value field of the DOE specification. If the actual response value is desired, leave the Value field fixed and the DOE specification blank (or zero).

The variable values for resistors is set using the VAR component as follows:

1. Double-click the **Var/Eqn** component to bring up the Variables and equations dialog box, shown below.



Variables and equations

VAR

Instance Name (name[<start:stop>]) VAR1

Select Parameter

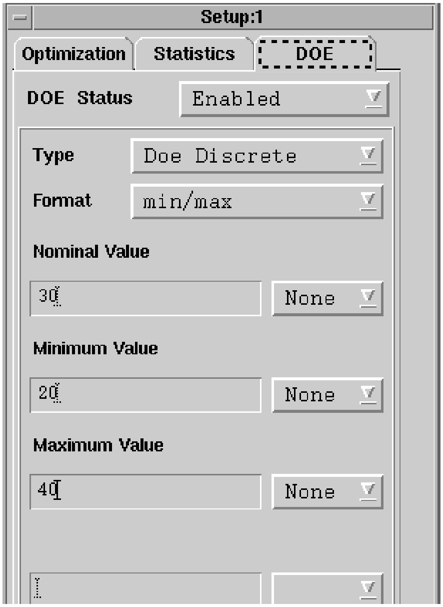
C=30 doe{ 20 to 40 }
B=30 doe{ 20 to 40 }
A=30 doe{ 20 to 40 }

Variable or Equation Entry Mode: Standard

Name: C

Variable Value: 30 None

2. In the Name field, enter **C**.
3. Click the Optimization, Statistics/DOE button.
4. In the dialog box that appears, choose the **DOE** tab.
5. In the DOE status field, select **Enabled** and Format to **min/max**.
6. Set the Nominal Value fields as follows and as shown in the figure below.
   - Nominal Value: **30**
   - Minimum Value: **20**
   - Maximum Value: **40**



Setup:1

DOE Status: Enabled

Type: Doe Discrete

Format: min/max

Nominal Value: 30 None

Minimum Value: 20 None

Maximum Value: 40 None

7. Choose **OK**. The following appears in the Select Parameter list box:

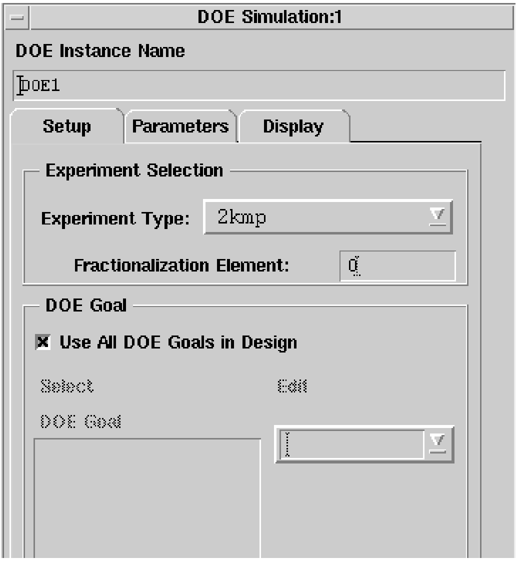
   C= 30 doe {20 to 40}

8. Repeat step 7, changing *C* to *B*, then choose **Add**.
9. Repeat step 7, changing *B* to *A*, then choose **Add**.
10. Choose **OK** to dismiss the dialog box.

Setup the DOE Component

Next, we will setup the DOE Component.

1. Double click the DOE Component. The DOE Simulation dialog box appears with the Setup tab active, as shown below.



DOE Simulation:1

DOE Instance Name: DOE1

Experiment Type: 2kmp

Fractionalization Element: 0

☒ Use All DOE Goals in Design

In DOE, the type of experiment to run is both problem-dependent and subjective. Because there are only three factors in the current example, the 2kmp is initially used (refer to the section“DOE Concepts” on page 4-14).

2. Select **2kmp** in the Experiment Type drop-down list and leave the *Fractionalization Element* at its default setting, **0**.
3. Click **Apply**.
4. Select the Parameters tab.
5. The Parameters tab controls output data. Accept the defaults and click **Apply**.
6. Select the Display tab.
7. The Display tab controls which parameters are displayed on the schematic. Accept the defaults and click **OK.**

Start the Experiment

1. In the Schematic window, select **Simulate** > **Simulate**.

You see messages in the Status window showing the current treatment combination number. For the 2kmp with k = 3 and p = 0, there are 8 treatment combinations necessary for the experiment. Once the number of simulations is complete, a message appears, reporting the progress of DOE data computation and display.

Analyze the Experiment

Once the treatment combinations have been simulated and the DOE data computation and display task is complete, you can access the three main DOE reporting tools (refer to the section, "DOE Outputs" on page 4-21 for explanations of these plots):

- Pareto diagrams
- Effects plots
- Interactions diagrams

Pareto Diagrams

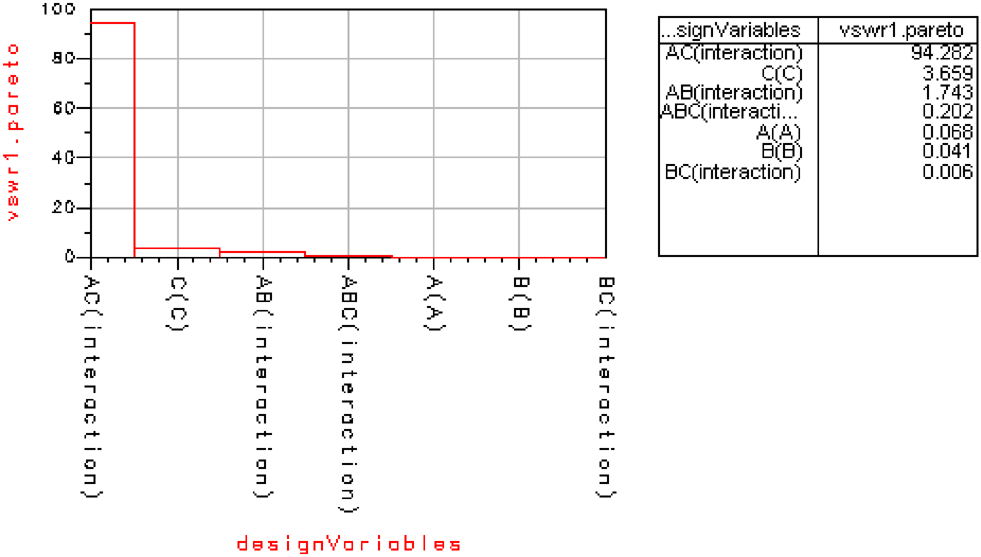
To examine the Pareto diagrams:

1. Select **Window** > **New Data Display**.
2. Select a Rectangular Plot and place it in the middle of the window. The Plot Traces & Attributes dialog box appears.
3. From the Datasets and Equations drop-down list, select **s21_db.pareto**.
4. Choose the **Add** button.
5. Choose **OK**.
6. A diagram showing s21_db versus design variables is shown.

Note that the Datasets and Equations drop-down list contains a series of plots and diagrams for each goal. Using the Plot Traces & Attributes dialog box, you can choose the diagram you are interested in, and add or delete traces in the plots.

Let's select a Pareto diagram for a different DOE goal:

7. Double-click the Rectangular Plot and select **vswr1.pareto** from the Datasets and Equations drop-down list that appears. The plot is shown in Figure 4-8 below.



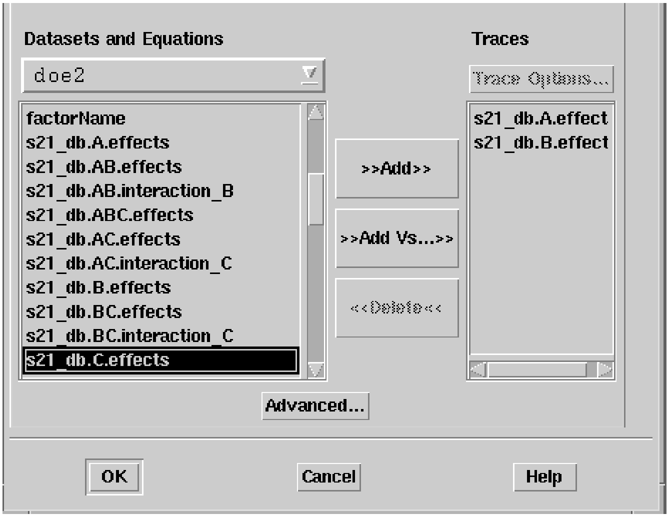
...signVariables	vswr1.pareto
AC(interaction)	94.282
C(C)	3.659
AB(interaction)	1.743
ABC(interacti...	0.202
A(A)	0.068
B(B)	0.041
BC(interaction)	0.006

Figure 4-8. Pareto Diagram for VSWR1 Measurement

Effects Plots

Next let's examine the Effects plots for the S21 DOE goal:

1. From the opened Data Display window, Choose **File** > **New**.
2. Place a Rectangular Plot in the window. The Plot Traces & Attributes dialog box appears.
3. From the Datasets and Equations drop-down list, select **s21_db.A.effects** and click the **Add** button.
4. Next, select **s21_db.B.effects** and click **Add**.
5. Lastly, select **s21_db.C.effects.** The dialog box should appear as shown below:



Datasets and Equations

doe2

factorName
s21_db.A.effects
s21_db.AB.effects
s21_db.AB.interaction_B
s21_db.ABC.effects
s21_db.AC.effects
s21_db.AC.interaction_C
s21_db.B.effects
s21_db.BC.effects
s21_db.BC.interaction_C
s21_db.C.effects

Traces

s21_db.A.effect
s21_db.B.effect

6. Click **Add** one more time and click **OK**. The Effect plot appears as shown in Figure 4-9.

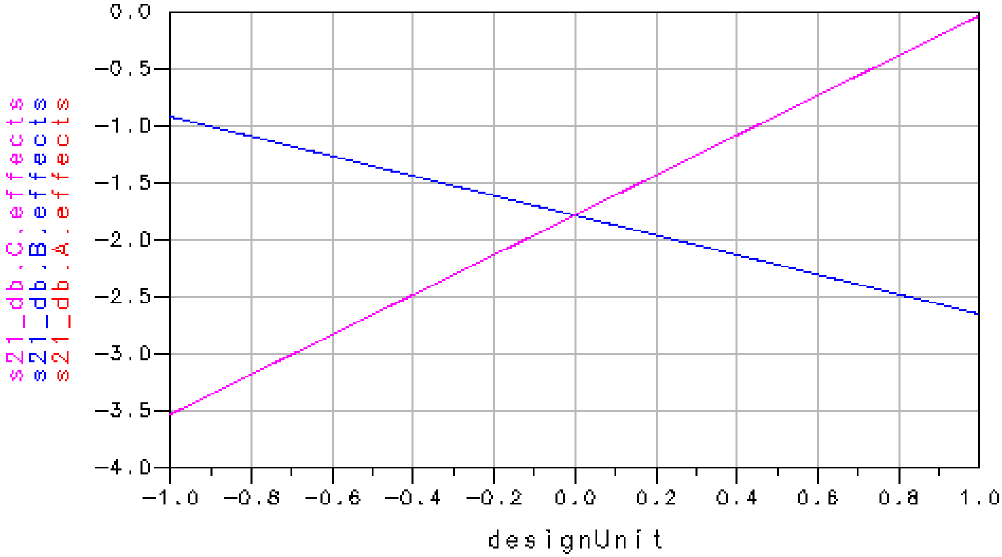


Figure 4-9. Effects Plot for DOE Goal S21

Notice that factor C has the largest slope (Effect) of the three factors. Notice also that factors A and B have identical effects. Finally, note that the response has been offset by the specified goal appearing in the DOE goal components. For example, the goal for S21 is –10 dB. Because the goal is subtracted from each response, the target response on the Effects plot is zero.

Interaction Diagrams

Interaction diagrams are used to examine the effect of one factor on a different factor.

To obtain the Interaction diagram for factors AB on the S21 DOE goal:

1. Select **Window** > **New Data Display**.
2. Select a Rectangular Plot and place it in the middle of the window. The Plot Traces & Attributes dialog box appears.
3. From the Datasets and Equations drop-down list, select **s21_db.AB.interaction_B**.
4. Choose the **Add** button.
5. Choose **OK**. The AB interaction diagram for DOE goal S21 appears, as shown in Figure 4-10 below.

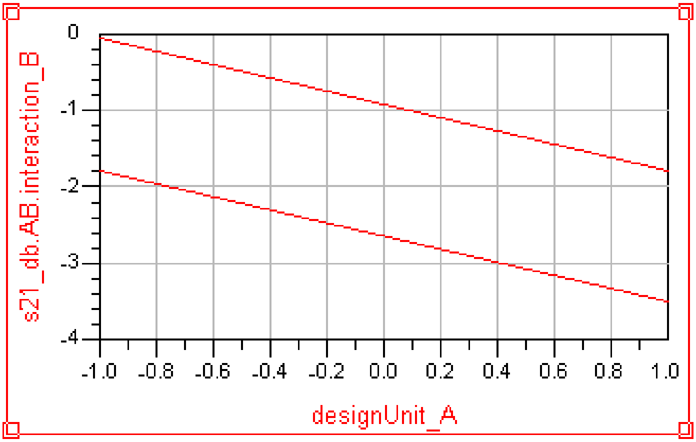


Figure 4-10. AB Interaction Diagram for S21 Measurement

Notice that the traces are parallel, indicating that there is little or no interaction between factors A and B—the change in average response as a function of factor A does not change as a function of factor B. However, there is an offset indicating that to achieve the (adjusted) target goal of zero, both A and B should be set to the low levels.

Optimizing Using DOE Outputs

In the DOE implementation, there are two methods to accomplish design improvement. The first and easiest to apply involves examining Effects plots and making approximate changes to the design factors in an effort to put the response on target. The second method involves solution of the set of model equations. In this section, the first method is examined, with the same example used previously in this chapter.

By examining the Effects plot for the DOE goal S21, it is clear that an increase in C and a decrease in the value of A and B will work toward putting the nominal response on the (adjusted) target value of zero. However, while this observation is true for the S21 goal, it may not be accurate for the VSWR goals.

Let’s continue following along with our example (../examples/Tutorial/doe2_prj). To view the optimized DOE output, select the design *doe2b.dsn* (from the Schematic window, choose *File* and select *doe2b* from the file history list at the bottom of the menu).

To examine the Effects plot for Vswr1:

1. Select **Window** > **New Data Display**.
2. Select a Rectangular Plot and place it in the middle of the window. The Plot Traces & Attributes dialog box appears.
3. From the Datasets and Equations drop-down list, select **vswr1.A.effects**.
4. Choose the **Add** button.
5. Using the scroll bar of the list, locate, select and **Add** the first three components associated with Effects of Vswr1:

   vswr1.B.effects

   vswr1.C.effects
6. Click **OK** to review the Vswr1 Effects plot.

If you do not have a display for the S21 and Vswr2 effects, follow the above instructions to obtain any missing plot. Figure 4-11 through Figure 4-13 show the Effects plots for Vswr1, Vswr2, and S21, respectively.

Once Effects plots for S21, Vswr1, and Vswr2 are available, arrange them so that they can be viewed simultaneously. Notice that the traces of the Vswr1 and Vswr2 plots are the same. The difference is that factor A in one plot is replaced by factor B in the other. This makes sense due to the symmetry in the network—the series resistors (factors A and B) take on the same values.

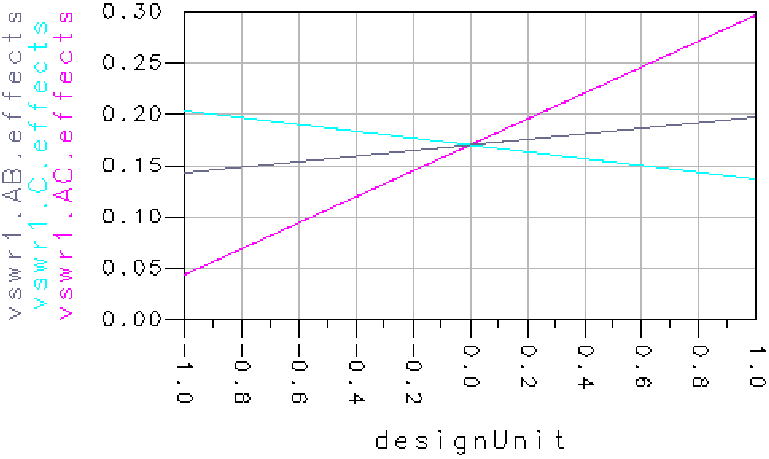


Figure 4-11. Effects Plot for Vswr1

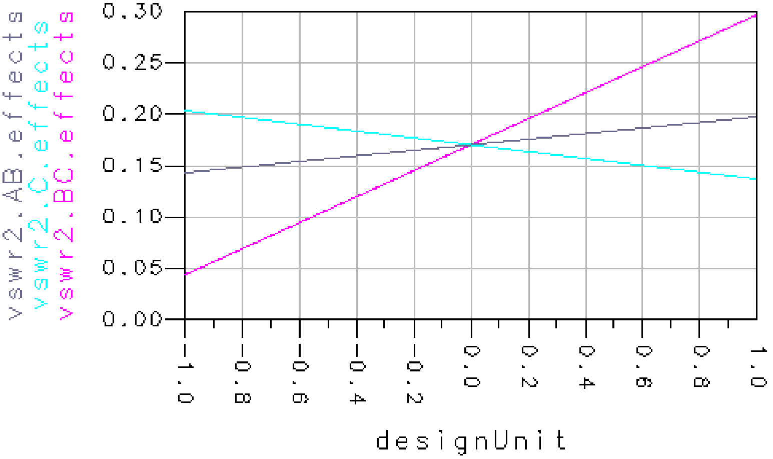


Figure 4-12. Effects Plot for Vswr2

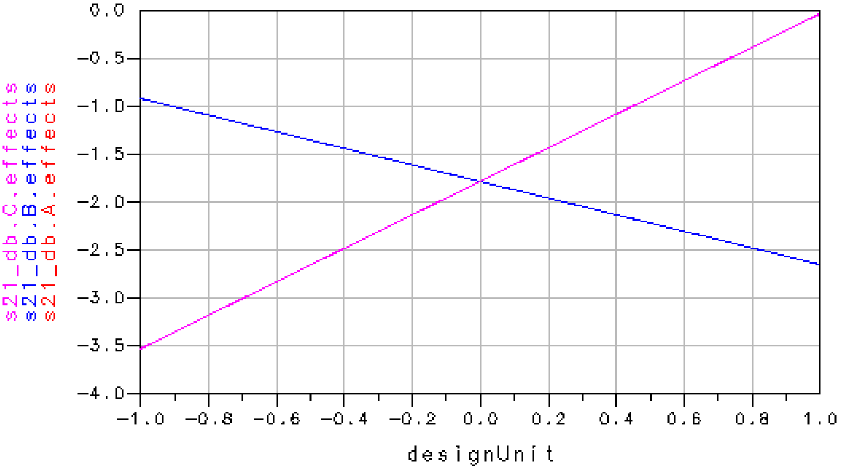


Figure 4-13. Effects Plot for S21

How Goals Are Affected

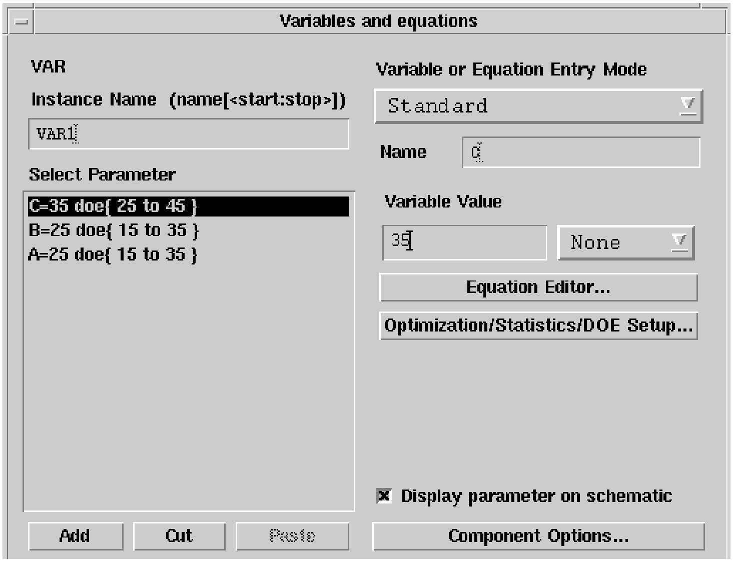
The motivation behind viewing Vswr1, Vswr2, and S21 effects concurrently is to ensure that any factor modifications are commensurate with the overall performance goals. As mentioned previously, it appears that an increase in C and a decrease in the value of A and B will work toward satisfying the S21 goal. We must also consider how this affects the *Vswr* goals. Noting the significant positive slope of the AC (Vswr1) and BC (Vswr2) interaction effects, it appears that an increase in C and a decrease in A and B would be favorable to our overall goals.

The only question that remains now is how much to change the factor level nominal values. Let's try an increase in C by 1/2 unit and a decrease in A and B by the same amount.

Noting that a 1/2 unit change in design units equates to a 5-ohm change in resistance values, the nominal values for series resistors drop from 30 to 25 ohms, and the shunt resistor should be changed from 30 to 35 ohms.

To modify the nominal, minimum, and maximum values for the factors:

1. Double-click the **Var/Eqn** component to bring up the Variables and equations dialog box, shown below.



Instance Name (name[<start:stop>])

VAR1

Select Parameter:

C=35 doe{ 25 to 45 }
B=25 doe{ 15 to 35 }
A=25 doe{ 15 to 35 }

Variable or Equation Entry Mode: Standard

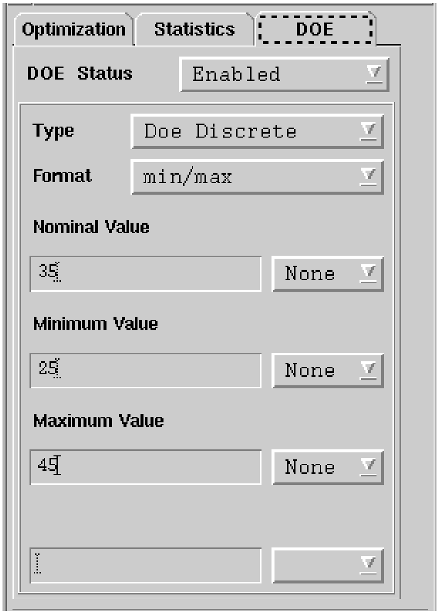
Name: C

Variable Value: 35 None

☒ Display parameter on schematic

2. In the Name field, enter **C**.

3. Click the Optimization, Statistics/DOE button.
4. In the dialog box that appears, choose the **DOE** tab.
5. In the DOE status field, select **Enabled** and Format to **min/max**.
6. Set the Nominal Value fields as follows and as shown in the figure below.
    - Nominal Value: **35**
    - Minimum Value: **25**
    - Maximum Value: **45**



DOE Status: Enabled

Type: Doe Discrete

Format: min/max

Nominal Value: 35 None

Minimum Value: 25 None

Maximum Value: 45 None

7. Choose **OK**. The following appears in the Select Parameter list box:

    C= 35 doe {25 to 45}

8. Repeat step 7, changing *C* to *B*, then choose **Add**.
9. Repeat step 7, changing *B* to *A*, then choose **Add**.
10. Click **OK**. The changes in the *Var/Eqn* component are reflected in the schematic. The new schematic should look similar to the one shown in Figure 4-14.

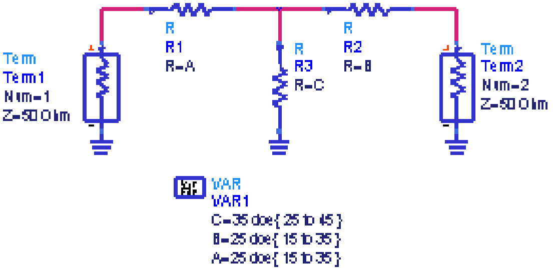


Figure 4-14. Schematic with Modified Var/Eqn Component

Performing the DOE Confirmation Experiment

To start the experiment using the new factor nominal values:

1. In the Schematic window, select **Simulate** > **Simulate**.

You see messages in the Status window showing the current treatment combination number. For the 2kmp with k = 3 and p = 0, there are 8 treatment combinations necessary for the experiment. Once the number of simulations is complete, a message appears, reporting the progress of DOE data computation and display.

Analyzing the DOE Confirmation Experiment

Once the treatment combinations have been simulated and the DOE data computation and display task is complete, you can re-examine the three Effects plots created earlier for Vswr and S21 DOE goals.

To examine the Effects plot for Vswr1:

1. Repeat the steps described in the section “Effects Plots” on page 4-30, except now we are using the results from the *doe2b* dataset.
2. Repeat steps 1-5 to obtain similar plots for the first three effects of S21 and Vswr2. The Effects plots for Vswr1, Vswr2, and S21 are shown in Figure 4-15, Figure 4-16, and Figure 4-17, respectively.

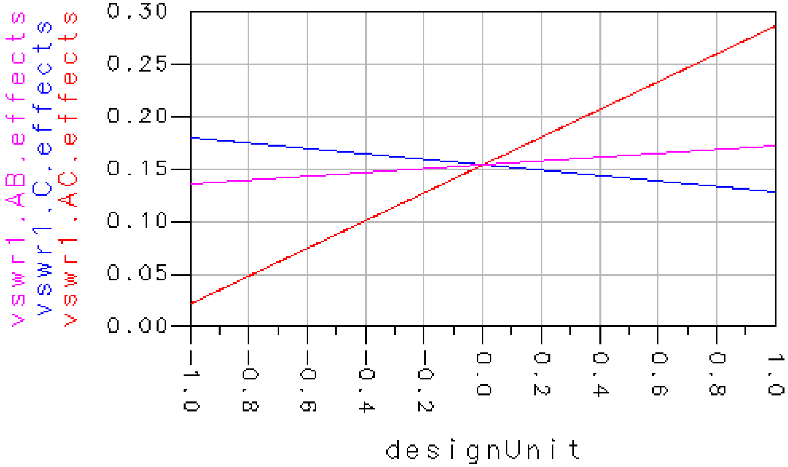


Figure 4-15. Effects Plot for Vswr1

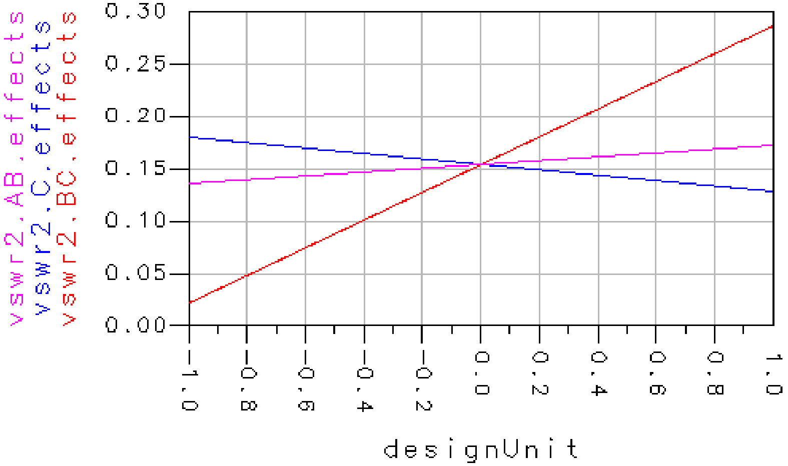


Figure 4-16. Effects Plot for Vswr2

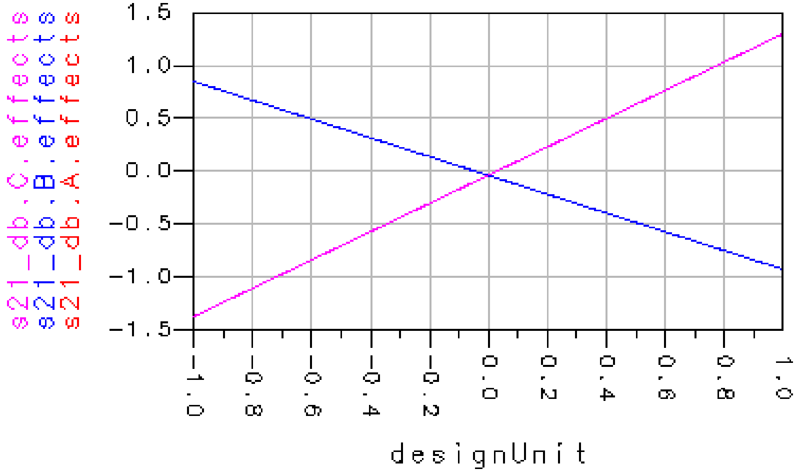


Figure 4-17. Main Effects Plot for S21

Once Effects plots for S21, Vswr1, and Vswr2 are available, arrange them so that they can be viewed simultaneously. Notice that our S21 target is very nearly satisfied while the VSWR goals are off target by about 0.17. The movement in the average response for VSWR is small—from about 0.16 to 0.17. If this performance were deemed unacceptable, another iteration of the procedure could be applied.

DOE References

[1] Robert L. Mason, Richard F. Gunst, and James L. Hess, *Statistical Design and Analysis of Experiments (with Applications to Engineering and Science*, New York, NY: John Wiley & Sons, 1989.

[2] Douglas C. Montgomery, *Design and Analysis of Experiments, 2nd Ed.*, New York, NY: John Wiley & Sons, 1984.

[3] Thomas B. Barker, *Quality by Experimental Design*, New York, NY; Marcel Dekker, 1985.

[4] Thomas B. Barker, *Engineering Quality by Design—Interpreting the Taguchi Approach*, New York, NY; Marcel Dekker, 1990.

[5] Stephen R. Schmidt and Robert G. Launsby, *Understanding Industrial Designed Experiments*, 3rd Ed., Colorado Springs, Co.: Air Academy Press, 1992.

[6] Keki R. Bhote, *World Class Quality (Understanding Design of Experiments to Make it Happen)*, New York, NY: AMACOM.

Chapter 5: Available Value Types

This appendix provides descriptions of the available parameter value types for Nominal Optimization, Statistical Design, and Design of Experiments (DOE). For the procedures in which these value types are implemented, refer to the following chapters:

- Chapter 2, Performing Nominal Optimization
- Chapter 3, Using Statistical Design
- Chapter 4, Using Design of Experiments (DOE)

Value Types for Nominal Optimization

As described in the section, "Specifying Component Parameters for Optimization" on page 2-2 in Chapter 2, Performing Nominal Optimization, the Optimization tab of the Setup dialog box is used to enable or disable the optimization status of a parameter and to specify the type and format for the parameter range over which optimization is to take place.

In the Optimization tab, the *Type* drop-down list includes the following options:

Discrete Denotes a variable to be used only for discrete optimization. The range of discrete values is directly specified when you enter nominal value, minimum value, maximum value, and a step value. Notice that for this option, the *Format* drop-down list only includes *min/max/step*.

Note The discrete variable type is compatible only with the Random, Random Minimax, Random Max, Discrete, and Genetic optimization types. (Refer to the section, "Available Optimizers" on page 6-1. This variable type is ignored for all other nominal optimization methods, such as Gradient.

Continuous Denotes a variable that can be one of four types, which are selected from the Format drop down list, as follows:

- **min/max** Allows you to specify a nominal value, minimum value, and maximum value and to specify appropriate units for each.
- **+/- Delta %** Allows you to specify a nominal value and an appropriate unit for it, as well as Delta%.

- **+/- Delta** Allows you to specify a nominal value and an appropriate unit for it, as well as Delta.
- **Unconstrained** Allows you to specify only a nominal value and an appropriate unit for it. The range of values is unlimited, except in the case of random optimization. In this case the range is constrained between zero and twice the initial nominal value. Caution must be used when assigning this format to an item parameter. Many component parameters have a limited range for valid numeric assignment. Refer to your Advanced Design System *Circuit Component* manual for component parameter value limits.

Value Types for Statistical Design

As described in the section, "Specifying Component Parameters for Yield Analysis" on page 3-3, the Statistics tab of the Setup dialog box is used to enable or disable the yield analysis status of a parameter and to specify the type and format for the parameter range over which yield analysis is to take place.

In the Statistics tab, the *Type* drop-down list includes the following options:

Gaussian Denotes a Gaussian distributed statistical variable that can be one of two types, which are selected from the Format drop-down list, as follows:

- **+/- Std.Dev.%** Specifies the +/- *1 sigma* deviation range as a percentage of the nominal value.
- **+/-Std.Dev.** Specifies the +/- *1 sigma* deviation value as an absolute value.

Uniform Denotes a variable that can be one of three types, which are selected from the Format drop down list, as follows:

- **min/max** Allows you to specify a nominal value, minimum value, and maximum value and to specify appropriate units for each.
- **+/- Delta %** Specifies the deviation range as a percentage of the nominal value.
- **+/- Delta** Specifies the deviation value as an absolute value.

Discrete Denotes a discrete uniform statistical variable. The set of discrete values is directly specified when you enter nominal value, minimum value, maximum value, and a step value. Notice that for this option, the *Format* drop-down list only includes *min/max/step*.

LogNormal Denotes a log-normal distributed statistical variable. A log-normal distribution is a probability distribution in which the logarithm of the parameter has a normal distribution. It is the minimum-information distribution for positive quantities with a given geometric mean and standard deviation. It is also the multiplicative analog of the bell curve. This variable can be one of two types, which are selected from the Format drop-down list, as follows:

- **+/- Std.Dev.%** Specifies the +/- *1 sigma* deviation range as a percentage of the nominal value.
- **+/-Std.Dev.** Specifies the +/- *1 sigma* deviation value as an absolute value.

Value Types for DOE

As described in the section, "Specifying Component Parameters for DOE" on page 4-2, the DOE tab of the Setup dialog box is used to enable or disable the DOE status of a parameter and to specify the type and format for the parameter range over which DOE is to take place.

In the DOE tab, the *Type* drop-down list includes the following options:

DOE Discrete Denotes a variable that can be one of three types, which are selected from the Format drop down list, as follows:

- **min/max** Allows you to specify a nominal value, minimum value, and maximum value and to specify appropriate units for each.
- **+/- Delta %** Specifies the deviation range as a percentage of the nominal value.
- **+/- Delta** Specifies the deviation value as an absolute value.

Chapter 6: Summary of Optimizers

This chapter provides details on the various optimizers and search methods available for performing nominal optimization, as well as guidelines on error function formulation.

The optimizers are differentiated by their search methods and error function formulations. The *search method* determines how the optimizer arrives at new parameter values, while the *error function* measures the difference between computed and desired responses. The smaller the value of the error function, the more closely the responses coincide. When optimizers execute their search method, they substitute new parameter values to effect a reduction in the error function value.

For details on setting up a nominal optimization, refer to Chapter 2, Performing Nominal Optimization.

Available Optimizers

Table 6-1 includes a description of each optimizer.

Table 6-1. Available Optimizers

Optimizer	Description
Random	Random search method with least-squares error function
Random Minimax	Random search method with minimax error function
Gradient	Gradient search method with least-squares error function
Gradient Minimax	Gradient search method with minimax error function
Quasi-Newton	Quasi-Newton search method with least-squares error function
Least P^{th}	Quasi-Newton search method with least Pth error function
Minimax	Two-stage, Guass-Newton/Quasi-Newton method with minimax error function
Random Max	Random search method with procedure to internally negate the error functions to get error function maximization (worst case analysis)
Discrete	Discrete optimization, provided there is at least one discrete-valued optimization parameter in the design.

Table 6-1. Available Optimizers

Optimizer	Description
Genetic	Direct search method using evolving parameter sets
Sensitivity	Single-point or infinitesimal sensitivity analysis of a design variable.

Search Methods

There are six search methods used to arrive at new parameter values: Random, Gradient, Quasi-Newton, combined Gauss-Newton/Quasi-Newton, Direct, and Genetic Algorithms (GA).

Random optimization (Random search) is typically used initially. Gradient optimization (Gradient search) is generally used in later stages of optimization. Discrete optimization only affects discrete-valued variables. The genetic algorithm search is well suited to the discrete and mixed (continuous and discrete) problems.

Random Search

The random optimizers arrive at new parameter values by using a random-number generator, that is, by picking a number at random within a range, which is sometimes a slower process compared to the gradient optimizers.

Random optimization is a trial and error process. Starting from an initial set of parameter values for which the error function is known, a new set of values is obtained by perturbing each of the initial values, and the error function is re-evaluated.

With random optimization, a *trial* consists of two error function evaluations. A trial performed by random optimization is completed by reversing the algebraic sign of each parameter value perturbation and re-evaluating the error function. These two values, corresponding to positive and negative perturbations, are compared to the value at the initial point.

If either value is less than the initial value, then the set of parameter values for which the error function has its least value becomes the initial point for the next trial. If neither value is less than the initial value, then the initial point remains the same for the next trial.

Gradient Search

The gradient optimizers find the gradient of the network's error function. These optimizers usually progress more quickly to a point where the error function is minimized, though it is possible for them to terminate in a local minimum.

The optimizers find the gradient of the error function (i.e., the direction to move a set of parameter values in order to reduce the error function). Once the direction is determined, the set of parameter values is moved in that direction until the error function is minimized. Then the gradient is re-evaluated. This cycle is equal to one iteration of the gradient optimizers.

A design that is optimized by a gradient optimizer has the least sensitivity (more stable) to slight variations in its parameter values.

A single iteration usually includes many function evaluations; therefore, an iteration in gradient optimization takes much longer than a trial in random optimization.

Quasi-Newton Search

The Quasi-Newton optimizers use second-order derivatives of the error function and the gradient to find a direction for the search.

The quasi-Newton optimization routine estimates the second-order derivatives using the DFP (Davidson-Fletcher-Powell) formula or its complement. Appropriately combined with the gradient, this information is used to find a direction and an inexact line-search is conducted. The optimization terminates when the gradient vanishes or the change in the variables is less than 1.0e-5. It also stops when the number of trials (iterations), that you have specified, is reached. The bounds imposed on the optimization variables are handled using a transformation of variables.

Like the gradient optimizers, an iteration in the Quasi-Newton optimizers consists of many function evaluations, and takes longer than a trial in the random optimizers.

Gauss-Newton/Quasi-Newton (minimax) Search

The minimax optimizer consists of two stages. In the first stage of the algorithm, the optimizer solves a minimax problem using a linear programming technique. In doing so, the status and potential of each individual error function component are analyzed. Its contribution to the minimax problem is mathematically assessed and taken into account during optimization.

In the second stage, the optimizer works with a Quasi-Newton method using approximate second-order derivatives. Such extra effort becomes necessary for an accurate and efficient solution to certain ill-conditioned problems (i.e., singular problems).

The minimax optimizer terminates when responses become optimally equal-ripple or the relative change in the variables is less than 0.05 percent. It also stops when the number of trials (iterations) is reached.

Note that the bounds imposed on the variables are formulated and treated directly as linear constraints without having to resort to variable transformation; therefore, a source of nonlinearity is eliminated.

Discrete Optimizer

The exhaustive search method is used exclusively by the Discrete optimizer. This optimizer only affects parameter values specified as discrete-valued optimization (Discrete Opt) variables.

The discrete search method involves a comprehensive search for the combination of discrete values that results in the best design performance. Starting from an initial set of parameter values for which the error function is known, an update in the parameter values occurs upon an improvement in the error function.

Because the parameter values that may change are not continuous variables, this search method is more a series of trials than iterations. Moreover, the number of trials required to attempt all combinations may often be prohibitive.

To reduce optimization time, keep the number of discrete variables to a minimum and reduce the number of values the discrete variables may take on.

Genetic Algorithm Search

Genetic algorithms (GA's) provide another direct search optimization method. The basis of the procedure is a set of trial parameter sets, sometimes called chromosomes, which are allowed to evolve towards a set that gives progressively better performance. The key to the genetic optimization is the strategy of change, sometimes likened to survival of the fittest. The idea is that with each change in the parameter population, i.e., each generation of parameters, the performance given by the parameter population improves. This whole process is achieved using a five-step process called (1) representation, (2) evaluation, (3) reproduction, (4) breeding and crossover, and (5) mutation.

1. *Representation*: Genetic algorithms require the input parameter set to be represented as a string of digits. It is straightforward to map each parameter onto the interval 0 to 1, for instance, and then have each of the n parameters occupy a position in the string of *n* bounded numbers. The algorithm then manipulates and optimizes this string of numbers as a whole. An individual string of parameters is called an element within the population of parameter strings.

2. *Evaluation:* Each generation of parameters begins with a performance evaluation of each string in the population. Usually this involves determining the performance G(P) for each representation of P in the population. Each element is then graded as to how well it performed, often using an error function, known as the *fitness* function.

3. *Reproductio*n: Some of the members of the population for this generation are copied, i.e. reproduced, and added to the next generation population. The number of copies depends on the performance evaluation. The elements that perform well are copied several times, and those with poor performance are not copied at all. The copies, or *offspring*, then make up the next generation. Elements that are not copied are not represented in the next generation.

   Note that the number of elements in each generation is constant. There are several suggested methods of ranking and reproduction, including *ratioing*, where the number of copies is directly related to the element's performance, and *ranking* where the performances are ranked, with the top performers being copied more times than the lower ranked performers.

4. *Breeding and crossover*: The previous step, reproduction, produced a population of strings where each evaluated well. Breeding then combines parts of two strings to form two different and new strings. In this way good representations are mixed with poorer representations, with the result eventually being evaluated in the next generation of the algorithm.

There are many methods for breeding; the most common is c*rossover.* Crossover typically takes two elements, splits them at a random location in the string, and swaps the two parts to create two new strings (see Figure 6-1). This provides a controlled statistical exploration of the performance space.

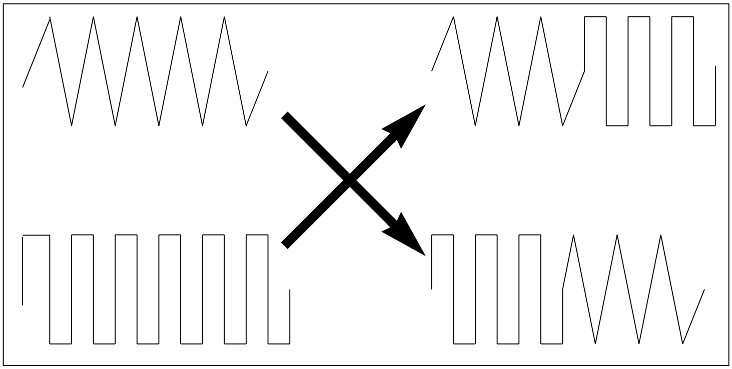


Figure 6-1. Breeding and Crossover in the Simple Genetic Algorithm

5. *Mutation:* The last step in creating a new generation of elements is the random changing of parameters in some of the surviving strings. This comprises a completely random search of the performance space, and can be viewed as the injection of information into the surviving population.

Figure 6-2 presents a completed flow diagram of the genetic algorithm. The application of these techniques requires many tuning parameters that are not available with yield optimization. Genetic optimization techniques will prove useful for many complex optimization problems, including discrete value and tolerance optimization.

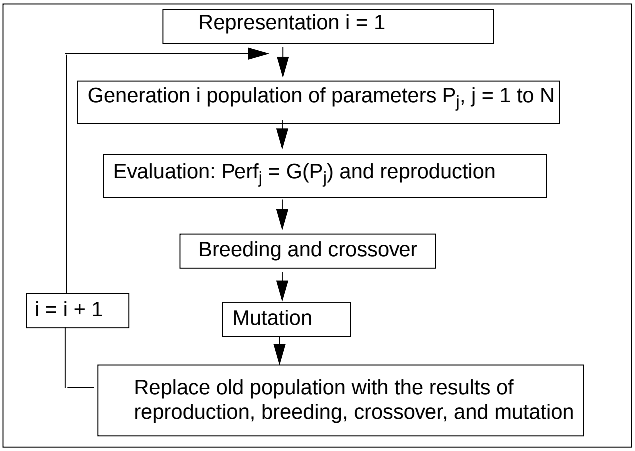

Representation i = 1
Generation i population of parameters P_j, j = 1 to N
Evaluation: $Perf_j = G(P_j)$ and reproduction
Breeding and crossover
i = i + 1
Mutation
Replace old population with the results of reproduction, breeding, crossover, and mutation

Figure 6-2. Random-Search Optimization Using a Genetic Algorithm

Sensitivity Analysis

While sensitivity analysis is not a type of optimization, the feature is documented here as it is selected from the *Optimization Type* drop-down list, from the Nominal Optimization dialog box, as are the other optimizers.

This feature comprises a single-point or infinitesimal sensitivity analysis of a design variable. Sensitivity analysis for circuit design involves taking partial derivatives of the response with respect to a design variable of interest. It is thought that these numbers can help pinpoint variables that contribute disproportionately to performance variance.

The method used to compute sensitivities is based on finite difference approximation requiring N+1 full circuit simulations, where N is the number of design variables.

By choosing Nominal Optimization dialog box > Setup tab > OptVar tab (in the Optimization Goal and Variable Setup box), you can specify a subset of all optimization variables within the design project.

Results are sent the Status window for immediate feedback without the need to open the Data Display window, by choosing Nominal Optimization dialog box > Parameters tab > Levels box > Status level field and entering a proper annotation level. Sensitivity analysis results are also unconditionally sent to the dataset, and this data can be examined in the Data Display window.

Sensitivities are approximated as follows:

$$S_{P_1} = \frac{\partial R(P)}{\partial P} \approx \frac{R(P^0) - R\left(P_i^+\right)}{P^0 - P_i^+}$$

> where R(P^0) is the response evaluated at the nominal point and R(P_i^+) is the response due to a pertubation in the ith parameter.

Note that in ADS, the response R is actually the expression found in the goal(s) given in the Optim (Nominal Optimization) component performing the sensitivity analysis.

Error Function Formulation

The optimizers also have different methods for error function formulation. The types of error function formulations are shown in Table 6-2

Table 6-2. Use of Error Function Formulation

Error Function Formulation	Optimizers
Least-squares	Gradient, Quasi-Newton, Random, Discrete, Genetic
Minimax	Gradient Minimax, Minimax, Random Minimax
Least P^{th}	Least P^{th}
Worst-case	Random Maximizer

Least-Squares

The least-squares error function is calculated by evaluating the error for each specified goal at each frequency/power point individually, then squaring the magnitudes of those errors. Then those squared magnitudes are averaged over frequency and/or power.

To help you understand the error function calculation in more generality for a measurement as a function of frequency, consider the following variable definitions.

F_j the set of frequency values specified by the “jth” frequency range

$R_{ij}(f)$ the “ith” frequency dependent response that is being optimized over the “jth” frequency range

g_{ij} the “ith” goal value within the “jth” frequency range that is the optimization criterion corresponding to the R_{ij} response

W_{ij} the “ith” goal weighting factor, within the “jth” frequency range, associated with the R_{ij} response and g_{ij} goal

$e_{ij}(f)$ the frequency dependent error contribution due to differences between R_{ij} and g_{ij}, evaluated at frequency “f.”

When a goal specification is satisfied, the error contribution is zero, i.e., e_{ij} (f)=0. The condition for satisfying a goal involves a specified relational operator (RelationOp). The following table summarizes these conditions.

Relational operator	**Goal satisfaction condition**
Equal to (=)	$R_{ij}(f) = g_{ij}$
Less than (<)	$R_{ij}(f) \le g_{ij}$
Greater than (>)	$R_{ij}(f) \ge g_{ij}$

When a goal is not satisfied, $e_{ij}(f)$ has a nonzero value given by

$$e_{ij}(f) = \left| R_{ij}(f) - g_{ij} \right|^2$$

The contribution to the total error function from response $R_{ij}(f)$ over the set of frequencies in the j^{th} range is then given by

$$E_{ij} = W_{ij} \bullet \sum_{f \in F_j} e_{ij}(f$$

The next step is to sum the contributions from all responses within the frequency range and divide by the number of frequencies. One way to express this mathematically is with the equation

$$E_j = \frac{\left(\sum_i E_{ij}\right)}{N_j}$$

in which N_j is the number of frequencies in frequency range F_j.

The final step in the error function calculation is to sum over all frequency ranges

$$E = \sum_j E_j$$

The weighting factors in different frequency ranges can be used to emphasize one of the E_j's with respect to others, just as weights within a frequency range can be used to attach greater or lesser importance to a given response relative to other responses within that frequency range.

In summary, the error function calculation can be represented by the following triple summation

$$E = \sum_j \frac{\left[\sum_f \left(\sum_i W_{ij} \bullet \left|R_{ij}(f) - g_{ij}\right|^2\right)\right]}{N_j}$$

where

The inner summation index i runs over all responses R_{ij} in frequency range F_j.

The second or middle summation is over all frequencies in frequency range F_j.

The outer summation index j runs over all frequency ranges.

For optimization with an additional 2nd swept variable (VAR) parameter, there is an additional summation over the parameter range. This error function formulation is represented in the following equation:

Ranges
Meas (1st independent measurement parameter)
Var (2nd independent parameter)
Response

$$E = \sum_j \left(\sum_f \frac{\left[\sum_P \left(\sum_i W_{ij} \bullet \ \left|R_{ij}(f, P) - g_{ij}\right|^2\right)\right]}{N_j \bullet (P_i)} \right)$$

Where the additional summation index p is over the swept variable levels in the swept variable range for the *ith* response, P_i represents the total number of swept variable levels in the *ith* responses power range. Thus, each response can have a unique swept variable range associated with it.

Remember that a response is any individual measurement on any network. And, that the weights W_{ij} have the value 1 unless some other value is given with the g_{ij} goal specification.

Minimax

The minimax optimizers calculate the difference between the desired response and the actual response over the entire measurement parameter range of optimization. Then the optimizer tries to minimize the point that constitutes the greatest difference between actual response and desired response.

Basically, minimax means *minimizing the maximum (of a set of functions generally denoted as errors)*. The objective function is defined as the maximum among all error functions, regardless of their signs or, expressed mathematically:

$U = \max_i (E_i)$

The minimax objective function always represents the worst case, where the specifications are either most severely violated (in which case U > 0) or, are satisfied with the worst error (in which case U < 0). The minimax optimizer will spend all its effort trying to minimize these. A minimax solution is one such that the goal specifications are met in an optimal, typically equal-ripple manner.

Least Pth

The Least P^{th} optimizer uses an error function formulation similar in makeup to the least squares method found in the random, gradient, and the quasi-Newton optimizers. But, instead of squaring the magnitudes of the individual errors at each frequency, it raises them to the P^{th} power, where p = 2, 4, 8, or 16. The optimizer automatically increases p in that sequence. This emphasizes the errors that have high values much more strongly than those that have small values. As p increases, the Least P^{th} error function approaches the minimax error function.

Least P^{th} allows the error function to become negative. That happens when you specify a performance window and the response moves inside that window.

For example, there may be a minimum and maximum gain specification on an amplifier and the Least P^{th} optimizer can go beyond the specification and place the gain halfway between the two limits.

The Least P^{th} optimization routine is the exponential sum of the error function, where the exponent p is not necessarily equal to 2. It can be a positive number, usually an integer.

First of all, the maximum error is found as:

$E_{max} = \max_i(E_i)$

According to the sign of E_{max}, the Least P^{th} objective is defined as follows:

$$U = \left(\sum_{i,\, E_i > 0} (E_i)^P \right)^{\frac{1}{P}} \quad \text{if } E_{max} > 0$$

$$U = 0 \text{if } \mathrm{E}_{\mathrm{max}} = 0$$

$$U = \left(\sum_i (-E_i)^{-p}\right)^{\frac{-1}{p}} \quad \text{if } \mathrm{E}_{\mathrm{max}} < 0$$

The cases $E_{max} > 0$ and $E_{max} < 0$ are clearly distinguished. The first case, where $E_{max} > 0$, only positive errors enter the objective function. For the second case, where $E_{max} < 0$, all errors become part of the objective function. In this case, all errors are negative, i.e., all the specifications are satisfied.

The Least P^{th} objective function is well-defined whether or not there are positive errors. It has the same sign as the maximum error Em and becomes negative when all the specifications are met. Therefore, the optimizer continues to improve the performance of the design, even after there are no more violations of the specifications.

The Least P^{th} formulation is used as an indirect method to achieve a minimax design. Minimax error functions can contain *edges* or discontinuities in their derivatives. These occur at points where the error contributions due to different goals intersect in the parameter space. The Least P^{th} error functions avoid this problem.

For a large value of p, the errors having the maximum value ($E_i = E_{max}$) are more strongly emphasized over the other errors, i.e., they are given higher priority in optimization. As p increases to infinity, the Least P^{th} formulation leads to a minimax objective. The problem is solved though a sequence of Least P^{th} optimizations with p being gradually increased. The sequential Least P^{th} optimization used in the program uses $p = 2 \rightarrow 4 \rightarrow 8 \rightarrow 16$. This strategy often provides a smooth path towards a minimax solution.

Worst Case

The optimizer that provides worst-case analysis is called the Random Maximizer. The random maximizer internally negates the least-squares error function so that the effect is a maximization of the error function.

Note The effect of this error function is to drive the values to the extreme of their ranges. To prevent destroying the originally desired response, you may want to save a copy of your optimized design, then change your optimization constraints to be equal to the tolerances on these variables.

For more information on the least-squares error function, refer back to the beginning of this section, keeping in mind the effect of the negation.

Chapter 7: Using Monte Carlo Yield Analysis

Monte Carlo yield analysis methods have traditionally been widely used and accepted as a means to estimate yield. The method simply consists of performing a series of trials. Each trial results from randomly generating yield variable values according to statistical-distribution specifications, performing a simulation and evaluating the result against stated performance specifications.

The power of the Monte Carlo method is that the accuracy of the estimate rendered is independent of the number of statistical variables and requires no simplifying assumptions about the probability distribution of either component parameter values or performance responses.

The weakness of this method is that a full network simulation is required for each trial and that a large number of trials is required to obtain high confidence and an accurate estimate of yield. Fortunately, the simulator uses state-of-the-art techniques to significantly boost the efficiency of the Monte Carlo method [1, 2, 3] while retaining its generality.

Consult the following references for details concerning state-of-the-art Monte Carlo techniques.

- M. D. Meehan and J. Purviance. *Yield and Reliability Design for Microwave Circuits and Systems,* Norwood, MA: Artech House, 1993.
- R. Spence and R. S. Soin. *Tolerance Design of Electronic Circuits*, Addison-Wesley, 1988.
- D. C. Hocevar, M. R. Lightner, and T. N. Trick. “A study of variance reduction techniques for estimating circuit yields,” *IEEE Trans. CAD*, vol. CAD-2, pp. 180-192, July 1983.

Monte Carlo Trials and Confidence Levels

The following discusses how to calculate the number of trials necessary for a given confidence and estimate error.

Confidence level is the area under a normal (gaussian) curve over a given number of standard deviations. Common values for confidence level are shown in the following.

Standard Deviations	Confidence Level
1	68.3%
2	95.4%
3	99.7%

Error is the absolute difference between the actual yield, Y, and the yield estimate, $\overline{Y}$, given by:

$$\varepsilon = \left| Y - \overline{Y} \right|$$

where ε is the percent error. The low value limit of $\overline{Y}$ is given by:

$$\overline{Y} = Y - \varepsilon$$

The sample or trial size, N, is then calculated from:

$$N = \left(\frac{C_\sigma}{\varepsilon}\right)^2 \bullet Y(1 - Y)$$

where C_σ is the confidence expressed as a number of standard deviations.

Example

For a 95.4% confidence level $(C_\sigma = 2)$, an Error = ±2% and a yield of 80%

$$N = \left(\frac{2}{0.02}\right)^2 \bullet 0.8 \bullet (1 - 0.8)$$

N = 1600 trials

Refer to the section, “Confidence Tables” on page 7-5 for help in determining the number of trials suitable for yield analysis.

The graphs shown in Figure 7-1 through Figure 7-3 may also be helpful in determining the accuracy of a yield analysis that you’ve already performed. These graphs plot error bounds of actual yield versus estimated yield for various values of *N* (number of trials).

The graph shown in Figure 7-1 plots error bounds with a confidence interval of 1, or 68.3%.

The graph shown in Figure 7-2 plots error bounds with a confidence interval of 2, or 95.4%.

The graph shown in Figure 7-3 plots error bounds with a confidence interval of 3, or 99.7%.

Suppose you ran a yield analysis on your design using 100 trials and the estimated yield was 50%. Referring to the graph in Figure 7-1, the lower bound on the actual yield is 45% and the upper bound is 55%.

From the graph shown in Figure 7-2, for 100 trials and an estimated yield of 50%, the lower bound on the actual yield is 40% and the upper bound is 60%.

Finally, from the graph shown in Figure 7-3, for 100 trials and an estimated yield of 50%, the lower bound on the actual yield is about 35% and the upper bound is about 65%.

Thus if you performed a yield analysis (either Monte Carlo or shadow model) using 100 trials, and the estimated yield was 50%, you have a 68.3% probability (confidence) that the actual yield is between 45% and 55%. You have a 95.4% probability that the actual yield is between 40% and 60%, and the probability is 99.7% that the actual yield is between 35% and 65%.

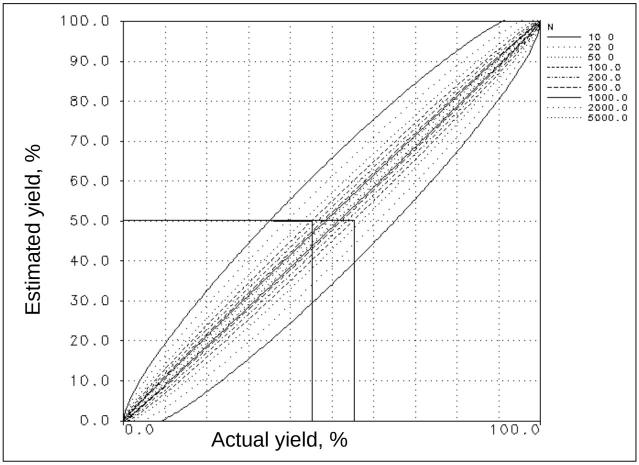


Figure 7-1. Yield for C = 1 (68.3% confidence)

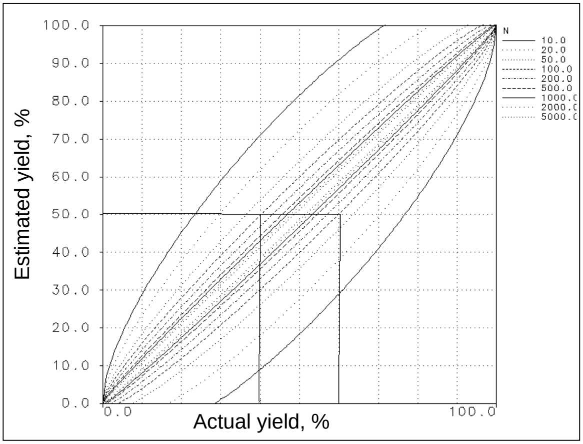


Figure 7-2. Yield for C = 2 (95.4% confidence)

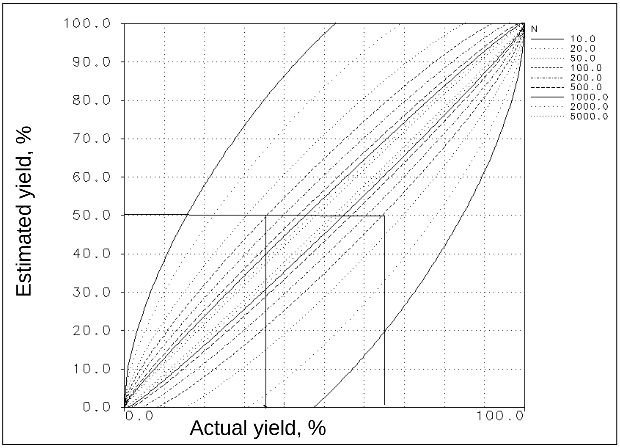


Figure 7-3. Yield for C = 3 (99.7% confidence)

Confidence Tables

The confidence table that follow can be used to determine the number of trials suitable for yield analysis.

Confidence = 68.3% Actual Yield = 90%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	89.00	91.00	900
2.0	88.00	92.00	225
3.0	87.00	93.00	100
4.0	86.00	94.00	56
5.0	85.00	95.00	36
6.0	84.00	96.00	25
7.0	83.00	97.00	18
8.0	82.00	98.00	14
9.0	81.00	99.00	11
10.0	80.00	100.00	9

Confidence = 95% Actual Yield = 90%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	89.00	91.00	3457
2.0	88.00	92.00	864
3.0	87.00	93.00	384
4.0	86.00	94.00	216
5.0	85.00	95.00	138
6.0	84.00	96.00	96
7.0	83.00	97.00	70
8.0	82.00	98.00	54
9.0	81.00	99.00	42
10.0	80.00	100.00	34

Confidence = 99% Actual Yield = 90%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	89.00	91.00	5967
2.0	88.00	92.00	1491
3.0	87.00	93.00	663
4.0	86.00	94.00	372
5.0	85.00	95.00	238
6.0	84.00	96.00	165
7.0	83.00	97.00	121
8.0	82.00	98.00	93
9.0	81.00	99.00	73
10.0	80.00	100.00	59

Confidence = 68.3% Actual Yield = 80%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	79.00	81.00	1600
2.0	78.00	82.00	400
3.0	77.00	83.00	177
4.0	76.00	84.00	100
5.0	75.00	85.00	64
6.0	74.00	86.00	44
7.0	73.00	87.00	32
8.0	72.00	88.00	25
9.0	71.00	89.00	19
10.0	70.00	90.00	16

Confidence = 95%   Actual Yield = 80%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	79.00	81.00	6146
2.0	78.00	82.00	1536
3.0	77.00	83.00	682
4.0	76.00	84.00	384
5.0	75.00	85.00	245
6.0	74.00	86.00	170
7.0	73.00	87.00	125
8.0	72.00	88.00	96
9.0	71.00	89.00	75
10.0	70.00	90.00	61

Confidence = 99%   Actual Yield = 80%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	79.00	81.00	10609
2.0	78.00	82.00	2652
3.0	77.00	83.00	1178
4.0	76.00	84.00	663
5.0	75.00	85.00	424
6.0	74.00	86.00	294
7.0	73.00	87.00	216
8.0	72.00	88.00	165
9.0	71.00	89.00	130
10.0	70.00	90.00	106

Confidence = 68.3% Actual Yield = 70%

Error +/- %	**Estimated % Yield**		**Number of Trials**
	Low	**High**	
1.0	69.00	71.00	2100
2.0	68.00	72.00	525
3.0	67.00	73.00	233
4.0	66.00	74.00	131
5.0	65.00	75.00	84
6.0	64.00	76.00	58
7.0	63.00	77.00	42
8.0	62.00	78.00	32
9.0	61.00	79.00	25
10.0	60.00	80.00	21

Confidence = 95% Actual Yield = 70%

Error +/- %	**Estimated % Yield**		**Number of Trials**
	Low	**High**	
1.0	69.00	71.00	8067
2.0	68.00	72.00	2016
3.0	67.00	73.00	896
4.0	66.00	74.00	504
5.0	65.00	75.00	322
6.0	64.00	76.00	224
7.0	63.00	77.00	164
8.0	62.00	78.00	126
9.0	61.00	79.00	99
10.0	60.00	80.00	80

Confidence = 99% Actual Yield = 70%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	69.00	71.00	13924
2.0	68.00	72.00	3481
3.0	67.00	73.00	1547
4.0	66.00	74.00	870
5.0	65.00	75.00	556
6.0	64.00	76.00	386
7.0	63.00	77.00	284
8.0	62.00	78.00	217
9.0	61.00	79.00	171
10.0	60.00	80.00	139

Confidence = 68.3% Actual Yield = 60%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	59.00	61.00	2400
2.0	58.00	62.00	600
3.0	57.00	63.00	266
4.0	56.00	64.00	150
5.0	55.00	65.00	96
6.0	54.00	66.00	66
7.0	53.00	67.00	48
8.0	52.00	68.00	37
9.0	51.00	69.00	29
10.0	50.00	70.00	24

Confidence = 95% Actual Yield = 60%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	59.00	61.00	9219
2.0	58.00	62.00	2304
3.0	57.00	63.00	1024
4.0	56.00	64.00	576
5.0	55.00	65.00	368
6.0	54.00	66.00	256
7.0	53.00	67.00	188
8.0	52.00	68.00	144
9.0	51.00	69.00	113
10.0	50.00	70.00	92

Confidence = 99% Actual Yield = 60%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	59.00	61.00	15913
2.0	58.00	62.00	3978
3.0	57.00	63.00	1768
4.0	56.00	64.00	994
5.0	55.00	65.00	636
6.0	54.00	66.00	442
7.0	53.00	67.00	324
8.0	52.00	68.00	248
9.0	51.00	69.00	196
10.0	50.00	70.00	159

Confidence = 68.3%　　　　　Actual Yield = 50%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	49.00	51.00	2500
2.0	48.00	52.00	625
3.0	47.00	53.00	277
4.0	46.00	54.00	156
5.0	45.00	55.00	100
6.0	44.00	56.00	69
7.0	43.00	57.00	51
8.0	42.00	58.00	39
9.0	41.00	59.00	30
10.0	40.00	60.00	25

Confidence = 95%　　　　　Actual Yield = 50%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	49.00	51.00	9604
2.0	48.00	52.00	2401
3.0	47.00	53.00	1067
4.0	46.00	54.00	600
5.0	45.00	55.00	384
6.0	44.00	56.00	266
7.0	43.00	57.00	196
8.0	42.00	58.00	150
9.0	41.00	59.00	118
10.0	40.00	60.00	96

Confidence = 99%   Actual Yield = 50%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	49.00	51.00	16576
2.0	48.00	52.00	4144
3.0	47.00	53.00	1841
4.0	46.00	54.00	1036
5.0	45.00	55.00	663
6.0	44.00	56.00	460
7.0	43.00	57.00	338
8.0	42.00	58.00	259
9.0	41.00	59.00	204
10.0	40.00	60.00	165

Confidence = 68.3%   Actual Yield = 40%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	39.00	41.00	2400
2.0	38.00	42.00	600
3.0	37.00	43.00	266
4.0	36.00	44.00	150
5.0	35.00	45.00	96
6.0	34.00	46.00	66
7.0	33.00	47.00	48
8.0	32.00	48.00	37
9.0	31.00	49.00	29
10.0	30.00	50.00	24

Confidence = 95% Actual Yield = 40%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	39.00	41.00	9219
2.0	38.00	42.00	2304
3.0	37.00	43.00	1024
4.0	36.00	44.00	576
5.0	35.00	45.00	368
6.0	34.00	46.00	256
7.0	33.00	47.00	188
8.0	32.00	48.00	144
9.0	31.00	49.00	113
10.0	30.00	50.00	92

Confidence = 99% Actual Yield = 40%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	39.00	41.00	15913
2.0	38.00	42.00	3978
3.0	37.00	43.00	1768
4.0	36.00	44.00	994
5.0	35.00	45.00	636
6.0	34.00	46.00	442
7.0	33.00	47.00	324
8.0	32.00	48.00	248
9.0	31.00	49.00	196
10.0	30.00	50.00	159

Confidence = 68.3% Actual Yield = 30%

Error	**Estimated % Yield**		**Number**
+/- %	**Low**	**High**	**of Trials**
1.0	29.00	31.00	2100
2.0	28.00	32.00	525
3.0	27.00	33.00	233
4.0	26.00	34.00	131
5.0	25.00	35.00	84
6.0	24.00	36.00	58
7.0	23.00	37.00	42
8.0	22.00	38.00	32
9.0	21.00	39.00	25
10.0	20.00	40.00	21

Confidence = 95% Actual Yield = 30%

Error	**Estimated % Yield**		**Number**
+/- %	**Low**	**High**	**of Trials**
1.0	29.00	31.00	8067
2.0	28.00	32.00	2016
3.0	27.00	33.00	896
4.0	26.00	34.00	504
5.0	25.00	35.00	322
6.0	24.00	36.00	224
7.0	23.00	37.00	164
8.0	22.00	38.00	126
9.0	21.00	39.00	99
10.0	20.00	40.00	80

Confidence = 99% Actual Yield = 30%

Error +/- %	**Estimated % Yield**		**Number of Trials**
	Low	**High**	
1.0	29.00	31.00	13924
2.0	28.00	32.00	3481
3.0	27.00	33.00	1547
4.0	26.00	34.00	870
5.0	25.00	35.00	556
6.0	24.00	36.00	386
7.0	23.00	37.00	284
8.0	22.00	38.00	217
9.0	21.00	39.00	171
10.0	20.00	40.00	139

Confidence = 68.3% Actual Yield = 20%

Error +/- %	**Estimated % Yield**		**Number of Trials**
	Low	**High**	
1.0	19.00	21.00	1600
2.0	18.00	22.00	400
3.0	17.00	23.00	177
4.0	16.00	24.00	100
5.0	15.00	25.00	64
6.0	14.00	26.00	44
7.0	13.00	27.00	32
8.0	12.00	28.00	25
9.0	11.00	29.00	19
10.0	10.00	30.00	16

Confidence = 95% Actual Yield = 20%

Error +/- %	**Estimated % Yield**		**Number of Trials**
	Low	**High**	
1.0	19.00	21.00	6146
2.0	18.00	22.00	1536
3.0	17.00	23.00	682
4.0	16.00	24.00	384
5.0	15.00	25.00	245
6.0	14.00	26.00	170
7.0	13.00	27.00	125
8.0	12.00	28.00	96
9.0	11.00	29.00	75
10.0	10.00	30.00	61

Confidence = 99% Actual Yield = 20%

Error +/- %	**Estimated % Yield**		**Number of Trials**
	Low	**High**	
1.0	19.00	21.00	10609
2.0	18.00	22.00	2652
3.0	17.00	23.00	1178
4.0	16.00	24.00	663
5.0	15.00	25.00	424
6.0	14.00	26.00	294
7.0	13.00	27.00	216
8.0	12.00	28.00	165
9.0	11.00	29.00	130
10.0	10.00	30.00	106

Confidence = 68.3% Actual Yield = 10%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	19.00	21.00	899
2.0	18.00	22.00	224
3.0	17.00	23.00	100
4.0	16.00	24.00	56
5.0	15.00	25.00	36
6.0	14.00	26.00	25
7.0	13.00	27.00	18
8.0	12.00	28.00	14
9.0	11.00	29.00	11
10.0	10.00	30.00	9

Confidence = 95% Actual Yield = 10%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	19.00	21.00	3457
2.0	18.00	22.00	864
3.0	17.00	23.00	384
4.0	16.00	24.00	216
5.0	15.00	25.00	138
6.0	14.00	26.00	96
7.0	13.00	27.00	70
8.0	12.00	28.00	54
9.0	11.00	29.00	42
10.0	10.00	30.00	34

Confidence = 99% Actual Yield = 10%

Error +/- %	Estimated % Yield		Number of Trials
	Low	High	
1.0	19.00	21.00	5967
2.0	18.00	22.00	1491
3.0	17.00	23.00	663
4.0	16.00	24.00	372
5.0	15.00	25.00	238
6.0	14.00	26.00	165
7.0	13.00	27.00	121
8.0	12.00	28.00	93
9.0	11.00	29.00	73
10.0	10.00	30.00	59

Index

D

design of experiments (DOE), 4-1
- concepts, 4-14
- confirmation experiment, 4-38
- design matrix, 4-19
- Effects plots, setting up, 4-30
- example, 4-24
- optimizing using outputs, 4-33
- outputs, 4-21
- Pareto diagrams, setting up, 4-29
- selecting specification, 4-8
- setting job parameters, 4-7
- setting parameter information, 4-9
- terminology, 4-12
- value types, 5-3

discrete
- optimizer, 6-4
- optimizer search method, 6-4

discrete optimization
- data file format, 2-29
- example, 2-21
- using data file, 2-25

E

Effects plots, 4-21
- setting up, 4-30

error function, 6-12
- formulation, 6-8
- worst-case analysis, 6-12

examples
- design of experiments, 4-24
- discrete optimization, 2-21
- nominal optimization, 2-15
- yield analysis, 3-16
- yield optimization, 3-30

G

genetic
- algorithms search method, 6-4

gradient
- optimizer, 6-3
- optimizer search method, 6-3

H

histograms
- sensitivity, 3-21

I

interaction diagrams, 4-21

L

Least Pth
- error function, 6-11

least-squares error function, 6-8

M

maximizer, random, 6-12

minimax
- error function, 6-10
- optimizer, 6-3, 6-10
- optimizer search method, 6-3

Monte Carlo yield analysis
- confidence tables, 7-1
- trials and confidence levels, 7-1
- using, 7-1

N

naming optimization variables, 2-36

O

optimization variables
- naming, 2-36

optimization, discrete
- data file format, 2-29
- example, 2-21
- using data file, 2-25

optimization, nominal
- displaying data on schematic, 2-13
- example, 2-15
- initiating, 2-14
- minimum requirements, 2-2
- optimizers, list and descriptions, 6-1
- performing, 2-1
- placing control items, 2-5
- selecting goals, 2-8
- selecting optimizer, 2-8
- setting goals, 2-6, 4-5
- setting job parameters, 2-8
- setting parameter information, 2-10
- updating parameters, 2-14
- value types, 5-1

P

Pareto diagrams, 4-21
- setting up, 4-29

post production tuning, 3-13

Q

quasi-Newton
- optimizer, 6-3
- optimizer search method, 6-3

R

random
- maximizer, 6-12
- optimizer, 6-2
- optimizer search method, 6-2

S

sensitivity analysis, 2-30
sensitivity histograms, 3-21
shadow model, 3-12, 3-28
statistical correlation, 3-33
statistical design
- methods, 3-1
- minimum requirements, 3-2
- using, 3-1
- value types, 5-2

swept optimization, 2-34

T

tuning
- analog/RF example, 1-10
- basic procedure, 1-2
- general description, 1-1
- setting preferences, 1-9
- signal processing example, 1-14
- tunable and non-tunable parameters, 1-17
- Tune Control Details dialog box, 1-5
- Tune Control dialog box, 1-3
- Tune Parameter Selection dialog box, 1-7

U

Update Optimization Values command, 2-15

V

value types
- design of experiments (DOE), 5-3
- nominal optimization, 5-1
- statistical design, 5-2

Y

yield analysis
- displaying data on schematic, 3-14
- example, 3-16
- histograms, 3-21
- initiating, 3-15
- minimum requirements, 3-2
- post production tuning, 3-13
- selecting shadow model, 3-12
- selecting specification, 3-9
- setting job parameters, 3-9
- setting parameter information, 3-11
- setting specification, 3-6
- specifying parameters, 3-3
- using Monte Carlo, 7-1

yield optimization
- displaying data on schematic, 3-29
- example, 3-30
- minimum requirements, 3-2
- performing, 3-25
- post production tuning, 3-13
- selecting shadow model, 3-28
- selecting specification, 3-26
- setting job parameters, 3-25
- setting parameter information, 3-27