
Affirma RF Simulator User Guide

Product Version 4.4.6

April 2001

© 1994-2001 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 1-800-862-4522.

All other trademarks are the property of their respective holders.

Restricted Print Permission: This publication is protected by copyright and any unauthorized use of this publication may violate copyright, trademark, and other laws. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This statement grants you permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used solely for personal, informational, and noncommercial purposes;
2. The publication may not be modified in any way;
3. Any copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement; and
4. Cadence reserves the right to revoke this authorization at any time, and any such use shall be discontinued immediately upon written notice from Cadence.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

Preface	19
Related Documents	19
1	
SpectreRF Analyses	20
Periodic Large-Signal Analysis	21
<u>Periodicity Assumption</u>	22
<u>Linearity Assumption</u>	22
Periodic Moderate-Signal Analysis (Pdisto)	23
<u>Large and Moderate Signals in Periodic Distortion Analysis</u>	23
<u>Periodicity Not Assumed</u>	23
<u>Linearity Assumption</u>	23
Periodic Small-Signal Analysis	23
<u>Fundamental Assumptions for the Periodic Small-Signal Analyses</u>	25
Description of SpectreRF Analyses	25
<u>Periodic Steady-State Analysis</u>	26
<u>Periodic Distortion Analysis (Pdisto)</u>	29
<u>Periodic AC Analysis</u>	32
<u>Intermodulation Distortion Computation</u>	34
<u>Periodic S-Parameter Analysis</u>	36
<u>Periodic Transfer Function Analysis</u>	40
<u>Periodic Noise Analysis</u>	43
<u>Noise Figure</u>	46
<u>Flicker Noise</u>	47
<u>Quasi-Periodic Noise Analysis</u>	49
<u>Envelope Following Analysis</u>	52

Affirma RF Simulator (SpectreRF) User Guide

2

SpectreRF Simulation Forms	55
Choosing Analyses Form	55
For Periodic Steady-State (PSS) Analysis (Top)	56
For Periodic Steady-State (PSS) Analysis (Bottom)	57
Modification for Periodic Steady State (PSS) Analysis Sweep (Bottom)	58
For Periodic Distortion (Pdisto) Analysis	59
For Periodic AC (PAC) Analysis with PSS	60
Modification for Periodic AC Analysis (PAC) with PSS Sweep	61
For Periodic Noise Analysis (Pnoise) with PSS	62
Modification for Periodic Noise Analysis (Pnoise) with PSS Sweep	63
For Periodic Transfer Function (PXF) Analysis with PSS	64
Modification for Periodic Transfer Function (PXF) Analysis with PSS Sweep	65
For Periodic S-Parameter (PSP) Analysis with PSS	66
Modification for Periodic S-Parameter (PSP) Analysis with PSS Sweep	67
For Quasi-Periodic Noise (QPnoise) Analysis with Pdisto	68
For Envelope Following Analysis	69
Options Forms	70
PSS Analysis Options Form (Top)	70
PSS Analysis Options Form (Bottom)	71
Pdisto Analysis Options Form (Top)	72
Pdisto Analysis Options Form (Bottom)	73
PAC Analysis Options Form	74
PXF Analysis Options Form	75
Pnoise Analysis Options Form	76
PSP Analysis Options Form	77
QPnoise Analysis Options Form	78
Envelope Following Analysis Options Form (Top)	79
Envelope Following Analysis Options Form (bottom)	80
Results Forms	81
Spectral Plots and Time Waveforms	81
PSS Results Form	83
Pdisto Results Form	87
PAC Results Form (with Swept PSS)	90
Pnoise Results Form (with PSS)	92

Affirma RF Simulator (SpectreRF) User Guide

<u>PXF Results Form (with PSS)</u>	93
<u>PSP Results Form (with PSS)</u>	95
<u>Envelope Following Results Form</u>	96
<u>QPnoise Results Form (with Pdisto)</u>	98
<u>Form Field Descriptions</u>	99
<u>Choosing Analysis Form</u>	99
<u>Options Forms</u>	123
<u>Results Forms</u>	130

3

<u>Setting Up for the Examples</u>	140
<u>Setting Up the Software</u>	140
<u>Copying the SpectreRF Simulator Examples</u>	140
<u>Setting Up the Cadence Libraries</u>	140

4

<u>Simulating Mixers</u>	143
<u>The ne600p Mixer Circuit</u>	144
<u>Simulating the ne600p Mixer</u>	146
<u>Opening the ne600p Mixer Circuit in the Schematic Window.</u>	146
<u>Choosing Simulator Options</u>	147
<u>Setting Up Model Libraries</u>	149
<u>Setting Design Variables</u>	150
<u>Harmonic Distortion Measurement with PSS</u>	151
<u>Setting Up the Simulation</u>	151
<u>Editing the Schematic</u>	152
<u>Setting Up the PSS Analysis</u>	153
<u>Running the Simulation</u>	155
<u>Calculating Harmonic Distortion</u>	156
<u>Noise Figure Measurement with PSS and Pnoise</u>	160
<u>Setting Up the Simulation</u>	160
<u>Editing The Schematic</u>	160
<u>Setting up the PSS and Pnoise Analyses</u>	161
<u>Running the Simulation</u>	163
<u>Plotting the Noise Figure</u>	164

Affirma RF Simulator (SpectreRF) User Guide

<u>Noise Figure Measurement and Periodic S-Parameter Plots with PSS and PSP</u>	165
<u>Setting Up the Simulation</u>	166
<u>Editing the Schematic</u>	166
<u>Setting up the PSS and PSP Analyses</u>	168
<u>Running the Simulation</u>	174
<u>Plotting the Noise Figure</u>	174
<u>Plotting Periodic S-Parameters</u>	176
<u>Conversion Gain Measurement with PSS and PXF</u>	180
<u>Setting Up the Simulation</u>	180
<u>Editing the Schematic</u>	180
<u>Setting Up the PSS and PXF Analyses</u>	181
<u>Running the Simulation</u>	184
<u>Plotting the Conversion Gain</u>	185
<u>Plotting the Power Supply Rejection</u>	188
<u>Calculating the 1 dB Compression Point with Swept PSS</u>	188
<u>Setting Up the Simulation</u>	188
<u>Editing the Schematic</u>	189
<u>Setting Up the Swept PSS Analysis</u>	189
<u>Running the Simulation</u>	191
<u>Plotting the 1 dB Compression Point</u>	192
<u>Third-Order Intercept Measurement with Swept PSS and PAC</u>	194
<u>Setting Up the Simulation</u>	195
<u>Editing the Schematic</u>	195
<u>Setting Up the PSS and PAC Analyses</u>	196
<u>Running the Simulation</u>	199
<u>Plotting the IP3 Curve</u>	200
<u>Intermodulation Distortion Measurement with Pdisto</u>	202
<u>Setting Up the Simulation</u>	203
<u>Editing the Schematic</u>	203
<u>Setting Up the Pdisto Analysis</u>	204
<u>Selecting Simulation Outputs</u>	206
<u>Running the Simulation</u>	207
<u>Plotting the Voltage and Power</u>	208
<u>Noise Figure with Pdisto and QPnoise</u>	211
<u>Setting Up the Simulation</u>	211
<u>Editing the Schematic</u>	211

Affirma RF Simulator (SpectreRF) User Guide

<u>Setting Up the Pdisto and QPnoise Analyses</u>	213
<u>Selecting Simulation Outputs</u>	219
<u>Running the Simulation</u>	220
<u>Plotting the Noise Figure</u>	220
<u>Plotting the Output Noise</u>	222
5	
<u>Simulating Oscillators</u>	225
<u>Autonomous PSS Analysis</u>	225
<u>Phases of Autonomous PSS Analysis</u>	225
<u>Phase Noise and Oscillators</u>	226
<u>Starting the Oscillator</u>	226
<u>Stabilizing the Oscillator</u>	226
<u>About the <i>tline3oscRF</i> Example</u>	227
<u>Simulating <i>tline3oscRF</i></u>	227
<u>Analysis Setup</u>	232
<u>Running the Simulation</u>	237
<u>Plotting the Fundamental Frequency</u>	237
<u>Plotting the Steady State Solution</u>	238
<u>Plotting the Phase Noise</u>	240
<u>A Balanced, Tunable Differential Oscillator: <i>oscDiff</i></u>	241
<u>Simulating <i>oscDiff</i></u>	242
<u>Analysis Setup</u>	246
<u>Editing Design Variables</u>	250
<u>Running the Simulation</u>	251
<u>Plotting the Fundamental Frequency</u>	251
<u>Troubleshooting for Oscillators</u>	254
6	
<u>Simulating Low-Noise Amplifiers</u>	256
<u>Types of Analysis Shown in this Chapter</u>	256
<u>Procedures for Simulating the LNA Example</u>	256
<u>Choosing Simulator Options</u>	259
<u>Setting Up Model Libraries</u>	261

Affirma RF Simulator (SpectreRF) User Guide

<u>Voltage Gain Calculation and Output Voltage Distribution</u>	261
<u>Editing Schematic</u>	261
<u>Analysis Setup</u>	262
<u>Running Simulation</u>	263
<u>Plotting Voltage Gain</u>	264
<u>Output Voltage Distribution</u>	266
<u>Editing Schematic</u>	266
<u>Analysis Setup</u>	267
<u>Running the Simulation</u>	269
<u>Plotting Output Voltage Distribution</u>	270
<u>S-Parameter Analysis for Low Noise Amplifiers</u>	272
<u>Editing the Schematic</u>	272
<u>Analysis Setup</u>	273
<u>Running Simulation</u>	274
<u>Plotting S-Parameters</u>	274
<u>Plotting the Voltage Standing Wave Ratio</u>	278
<u>Linear Two-Port Noise Analysis</u>	279
<u>Editing the Schematic</u>	279
<u>Analysis Setup</u>	280
<u>Running the Simulation</u>	281
<u>Plotting the Noise Figure and Minimum Noise Figure</u>	282
<u>Plotting the Equivalent Noise Resistance</u>	285
<u>Plotting Load and Source Stability Circles</u>	286
<u>Plotting the Noise Circles</u>	290
<u>Noise Calculations Using Pnoise</u>	292
<u>Editing the Schematic</u>	293
<u>Analysis Setup</u>	293
<u>Running the Simulation</u>	296
<u>Plotting the Noise Calculations</u>	297
<u>Plotting the 1dB Compression Point</u>	301
<u>Editing the Schematic</u>	302
<u>Analysis Setup</u>	302
<u>Running the Simulation</u>	304
<u>Plotting the 1dB Compression Point</u>	305
<u>Plotting the Third-Order Intercept Point</u>	307
<u>Editing the Schematic</u>	307

Affirma RF Simulator (SpectreRF) User Guide

<u>Analysis Setup</u>	308
<u>Running the Simulation</u>	309
<u>Plotting the Third-Order Intercept Point</u>	310
<u>Plotting Conversion Gain and Power Supply Rejection</u>	312
<u>Editing the Schematic</u>	312
<u>Analysis Setup</u>	313
<u>Running Simulation</u>	316
<u>Plotting Conversion Gain and Power Supply Rejection</u>	316

7

<u>Simulating Transmission Lines</u>	320
<u>Use Model One: Using the LMG Visual Interface</u>	320
<u>Use Model Two: Modeling Transmission Lines Without the Visual Interface</u>	321
<u>The Transmission Line Model Generator Form</u>	322
<u>Menus</u>	322
<u>Display Section</u>	325
<u>Data Entry Section</u>	325
<u>Function Buttons</u>	326
<u>Default Values</u>	326
<u>Example of Modeling a Transmission Line Using LMG</u>	327
<u>Setup</u>	327
<u>Creating the Transmission Line Model</u>	327
<u>Using Macromodels Created Externally by LMG</u>	334
<u>Example of Automatically Creating a Transmission Line</u>	338
<u>Accuracy of the LMG Macromodels</u>	340

8

<u>Modeling RF IC Packages</u>	345
<u>The Four PKG Building Blocks</u>	347
<u>Package Physical Geometry Modeling</u>	347
<u>EM Solvers and Mesh Making</u>	349
<u>Macromodel Generation</u>	349
<u>PKG Setup Requirements</u>	350
<u>License Checking</u>	350
<u>Executables</u>	350

Affirma RF Simulator (SpectreRF) User Guide

<u>PKG GUI Files</u>	350
<u>Initialization Files</u>	350
<u>Running pkg</u>	351
<u>The PKG RF IC Package Modeler Window</u>	351
<u>Menu Buttons</u>	352
<u>Function Buttons</u>	354
<u>Package Dimension Setup</u>	355
<u>QFP Packages</u>	355
<u>Data Entries for QFP Packages</u>	356
<u>SOIC Packages</u>	358
<u>Data Entries for SOIC Packages</u>	358
<u>DIP Packages</u>	359
<u>Data Entries for DIP Packages</u>	360
<u>Die Attach Setup</u>	361
<u>Data Entries for SOIC Packages</u>	362
<u>Internal Lead Frame Setup</u>	362
<u>Menu Buttons</u>	362
<u>Function Buttons</u>	363
<u>Data Entries for the Internal Lead Frame Setup</u>	364
<u>More about Using Symmetry</u>	365
<u>Lead Frame File Format</u>	365
<u>Bond Wire Setup</u>	367
<u>Menu Buttons</u>	367
<u>Function Buttons</u>	367
<u>Data Entries for the Bond Wire Setup</u>	369
<u>Bond Wire Configurations</u>	370
<u>Bond Wire File Format</u>	371
<u>Save the Current Setting</u>	372
<u>Parameter Extraction</u>	372
<u>Capacitance Calculation</u>	373
<u>Inductance Calculation</u>	375
<u>Function Buttons for Parameter Extraction</u>	377
<u>Macromodel Generation</u>	377
<u>Data Entries and Buttons for Macromodel Generation</u>	377
<u>Die Attach Terminal</u>	378
<u>Macromodel Schematic</u>	379

Affirma RF Simulator (SpectreRF) User Guide

<u>High Frequency Resonance Damping for the Spectre Simulator</u>	380
<u>Accuracy Analysis</u>	381
<u>Simulation of Package Models in the Affirma Analog Circuit Design Environment</u>	381
<u>Example of Package Modeling With a Mixer Using PKG</u>	385
9	
<u>Extracting K-Models</u>	409
<u>Procedures for Simulating k mod extraction example</u>	410
<u>Analysis Setup</u>	417
<u>Running the Simulation</u>	424
<u>Checking the Simulation Results</u>	424
<u>Creating the K-Model</u>	427
<u>More About the K-Model</u>	430
<u>K-model data files</u>	436
10	
<u>Extracting J-Models</u>	439
<u>Procedures for Simulating j mod extraction example</u>	439
<u>Analysis Setup</u>	446
<u>Running the Simulation</u>	451
<u>Creating the J-Model</u>	452
<u>Using the J-model in a circuit</u>	453
<u>More About the J-model</u>	466
11	
<u>Modeling Transmitters</u>	478
<u>Envelope Following</u>	478
<u>Taking a Closer Look at the Envelope Following Procedure</u>	482
<u>Following the Change in the Information (Baseband Signal) Through the Circuit</u>	486
<u>Plotting the Complete Baseband Signal</u>	491
<u>Plotting the Baseband Trajectory</u>	492
<u>Plotting the Transmitted Power Spectral Density for ACPR Calculations</u>	495
<u>Measuring Load Pull Contours and Reflection Coefficients</u>	504
<u>Setting Up and Running the Simulation</u>	504

Affirma RF Simulator (SpectreRF) User Guide

<u>Displaying Load Contours</u>	514
<u>Adding the Reflection Contours to the Plot</u>	517
<u>Moving to Differential Mode</u>	526
<u>Using S-Parameter Input Files</u>	527
<u>Setting Up the Schematic for the First Simulation</u>	527
<u>Setting Up and Running the First Simulation</u>	532
<u>The S-Parameter File</u>	533
<u>Setting Up and Running the Second Simulation</u>	533
<u>Using an S-Parameter Input File with a SpectreRF Analysis</u>	538

12

<u>Modeling Spiral Inductors and Bondpads</u>	544
<u>Modeling Spiral Inductors</u>	544
<u>Process File Preparation</u>	545
<u>Spiral Inductor Simulation in the Schematic Flow</u>	549
<u>Spiral Inductor Modeling under the RF Sub-Menu</u>	554
<u>Equivalent Circuit Models</u>	556
<u>Example</u>	559
<u>Modeling Bondpads</u>	569
<u>Input to the Modeler</u>	570
<u>Parameter Sets</u>	570
<u>Modeling Issues</u>	572
<u>LNA Example</u>	573
<u>References for the Bondpad Section</u>	581

13

<u>Behavioral Models and Methods for Top Down Design</u>	582
<u>Top Down Design of RF Systems</u>	583
<u>Baseband Modeling</u>	586
<u>Plotting Baseband Equivalent Output Signals for Baseband and Passband Circuits</u>	589
<u>About Baseband Modeling</u>	590

Affirma RF Simulator (SpectreRF) User Guide

<u>Library Overview</u>	591
<u>Warnings You Can Ignore</u>	595
<u>Use Model and Design Example</u>	596
<u>Relationship Between Baseband and Passband Noise</u>	620
<u>Baseband Library</u>	630
<u>Assumptions</u>	630
<u>Compatibility</u>	631
<u>Library Categories</u>	631
<u>Inputs and Outputs on Baseband Models</u>	631
<u>Some Parameters Common to Several Models</u>	632
<u>Library Description</u>	651
<u>Notes on models involving frequency translation:</u>	652
<u>Power Amplifier Model</u>	653
<u>Low Noise Amplifier Baseband Model</u>	655
<u>IQ Modulator Models</u>	657
<u>IQ Demodulator</u>	664
<u>RF-to-IF and IF-to-RF Mixers</u>	672
<u>Passive Devices</u>	679
<u>Linear Time Invariant Filters</u>	687
<u>Comparison of baseband and passband models</u>	695
<u>BB Loss</u>	697
<u>Phase Shifter</u>	698
<u>Phase Shifter</u>	702
<u>Ideal Transformer</u>	709
<u>Rectangular-to-Polar Transformation</u>	711
<u>Polar-to-Rectangular Transformation</u>	712
<u>Instrumentation and Terminating Blocks</u>	713
<u>Baseband Drive Signals</u>	716
<u>References</u>	723

A

<u>Oscillator Noise Analysis</u>	724
<u>Phase Noise Primer</u>	725
<u>Models for Phase Noise</u>	728
<u>Linear Time-Invariant (LTI) Models</u>	728

Affirma RF Simulator (SpectreRF) User Guide

<u>Linear Time-Varying (LTV) Models</u>	729
<u>Amplitude Noise and Phase Noise in the “Linear” Model</u>	733
<u>Details of the SpectreRF Calculation</u>	734
<u>Calculating Phase Noise</u>	737
<u>Setting Simulator Options</u>	737
<u>Tips about Getting PSS to Converge</u>	738
<u>How to Tell If the Answer Is Correct</u>	739
<u>Troubleshooting Phase Noise Calculations</u>	739
<u>Known Limitations of the Simulator</u>	740
<u>What Can Go Wrong</u>	740
<u>Phase Noise Error Messages</u>	742
<u>The tstab Parameter</u>	743
<u>Frequently Asked Questions</u>	744
<u>Further Reading</u>	749
<u>References</u>	749

B

<u>Running PSS Analysis Effectively</u>	752
<u>General Convergence Aids</u>	752
<u>The steadyratio and tstab Parameters</u>	752
<u>Additional Convergence Aids</u>	753
<u>Convergence Aids for Oscillators</u>	754
<u>Running PSS Analysis Hierarchically</u>	755

C

<u>Using the psin Component</u>	757
<u>Independent Resistive Source (psin)</u>	757
<u>Parameter Types for the psin Component</u>	760
<u>Name Parameters</u>	763
<u>psin Instance Parameter</u>	763
<u>General Waveform Parameters</u>	763
<u>Sinusoidal Waveform Parameters</u>	764
<u>Amplitude Modulation Parameters</u>	765
<u>FM Modulation Parameters</u>	767

Affirma RF Simulator (SpectreRF) User Guide

<u>Noise Parameters</u>	770
<u>Port Parameters</u>	771
<u>Temperature Effect Parameters</u>	771
<u>Small-Signal Parameters</u>	772
<u>Additional Notes</u>	774

D

<u>The RF Library</u>	775
<u>Top-Down Design Elements</u>	775
<u>Filter</u>	777
<u>Balun</u>	780
<u>Low Noise Amplifier</u>	780
<u>Mixer</u>	783
<u>Power Amplifier</u>	787
<u>Oscillator</u>	789
<u>Quadrature Signal Generator</u>	791
<u>Phase Shifter</u>	792
<u>Bottom-Up Design Elements</u>	793
<u>Testbench Elements</u>	793
<u>CDMA Signal Source (IS-95 rev xmit) and Eye-Diagram Generator</u>	794
<u>GSM Signal Source (GSM-xmtr)</u>	796
<u>P/4-DQPSK Signal Source (π over4 dqpsk)</u>	798
<u>Creating Variations on the Library Baseband Signal Generators Using the Modelwriter</u> 801	
<u>Uncategorized Elements</u>	805

E

<u>Plotting Spectre S-Parameter Simulation Data</u>	807
<u>The Equations for the S-Parameter Waveform Calculator in the Affirma Analog Circuit Design Environment:</u>	807

Affirma RF Simulator (SpectreRF) User Guide

F

<u>Using Pdisto Analysis Effectively</u>	818
<u>When to Use Pdisto Analysis</u>	818
<u>Essentials of the MFT Method</u>	820
<u>Pdisto and PSS Analyses Compared</u>	823
<u>Pdisto and PAC Analyses Compared</u>	824
<u>Pdisto Analysis Parameters</u>	825
<u>Application Examples</u>	825
<u>Switched Capacitor Filter Example</u>	826
<u>High-Performance Receiver Example</u>	827
<u>Running a Pdisto Analysis</u>	829
<u>Picking the Large Fundamental</u>	829
<u>Setting Up Sources</u>	830
<u>Sweeping a Pdisto Analysis</u>	831
<u>Convergence Aids</u>	832
<u>Memory Management</u>	834
<u>Dealing with Sub-Harmonics</u>	835
<u>Understanding the Narration from the Pdisto Analysis</u>	835
<u>References</u>	836

G

<u>An Introduction to the PLL library</u>	838
<u>The Models in the PLL library</u>	840
<u>Introduction</u>	840
<u>Phase-Domain Model of a Simple PLL</u>	841
<u>Assembling a Model of a PFD-based PLL</u>	854
<u>How the PFD Model Works</u>	864
<u>References</u>	870

H

<u>Using <i>port</i> in SpectreRF Simulations</u>	872
<u><i>port</i> Parameter Types</u>	873
<u>Port Parameters</u>	876

Affirma RF Simulator (SpectreRF) User Guide

<u>General Waveform Parameters</u>	876
<u>DC Waveform Parameters</u>	878
<u>Pulse Waveform Parameters</u>	879
<u>PWL Waveform Parameters</u>	881
<u>Sinusoidal Waveform Parameters</u>	885
<u>Exponential Waveform Parameters</u>	895
<u>Noise Parameters</u>	898
<u>Small-Signal Parameters</u>	899
<u>Temperature Effect Parameters</u>	902
<u>Additional Notes</u>	903

I

<u>Analyzing Time-Varying Noise Properties</u>	904
<u>Time-Domain Noise Characterization</u>	905
<u>Calculating Noise Correlation Coefficients</u>	910

J

<u>Using Tabulated S-parameters</u>	913
<u>Using the nport component</u>	914
<u>Controlling Model Accuracy</u>	914
<u>Using relerr and abserr</u>	915
<u>Using the ratorder parameter</u>	917
<u>Troubleshooting</u>	917
<u>Assessing the quality of the rational interpolation</u>	917
<u>Model Reuse</u>	918
<u>References</u>	919

K

<u>Using the Calculator IPN Functions for RF Simulations</u>	920
<u>How to Use the ipn and ipnVRI Functions</u>	920
<u>Setting Up</u>	920
<u>Editing the Schematic</u>	921
<u>Setting Up the Analysis</u>	922

Affirma RF Simulator (SpectreRF) User Guide

<u>Running the Simulation</u>	926
<u>Plotting the IP3 Curve Using Direct Plot</u>	926
<u>Plot IP3 Using the ipn and ipnVRI Buttons in the Waveform Calculator</u>	930
<u>Using the IPN Function with Scalar Inputs</u>	934
<u>Setting Up</u>	934
<u>Setting Up the Analysis</u>	934
<u>Running Simulation</u>	938
<u>Plotting the Single Point IP3 Using the Calculator</u>	939

L

<u>Using PSP and Pnoise Analyses</u>	942
<u>Overview</u>	942
<u>Periodic S-parameters</u>	943
<u>Linear Time-Invariant S-Parameters</u>	943
<u>Frequency Translating S-Parameters</u>	944
<u>Upper and Lower Sidebands</u>	946
<u>PSP Analysis Example</u>	946
<u>Noise and Noise Parameters</u>	950
<u>Calculating Noise in Linear Time-Invariant (DC Bias) Circuits</u>	950
<u>Calculating Noise in Time-Varying (Periodic Bias) Circuits</u>	950
<u>The maxsideband Parameter</u>	951
<u>Noise Correlation Matrices and Equivalent Noise Sources</u>	951
<u>Noise Figure</u>	954
<u>Performing Noise Figure Computations</u>	954
<u>Noise Figure From Noise and SP Analyses</u>	955
<u>Pnoise (SSB) Noise Figure</u>	955
<u>DSB Noise Figure</u>	957
<u>IEEE Noise Figure</u>	958
<u>Noise Computation Example</u>	961
<u>Input Referred Noise</u>	962
<u>Using Input Referred Noise</u>	963
<u>How IRN is Calculated</u>	964
<u>Relation to Gain</u>	966
<u>Referring Noise to Ports</u>	966

Affirma RF Simulator (SpectreRF) User Guide

<u>Gain Calculations</u>	967
<u>Definitions of Gain</u>	967
<u>Gain Calculations in Pnoise</u>	971
<u>Phase Noise</u>	973
<u>Frequently Asked Questions</u>	974
<u>Known Problems and Limitations</u>	977
<u>Dubious AC-Noise Analysis Features</u>	977
<u>Gain in Pnoise and PSP Analyses Inconsistent</u>	978
<u>Harmonics and Sidebands in PSP, PAC, PXF, and Pnoise Analyses</u>	978
<u>Index</u>	982

Preface

This manual assumes that you are familiar with RF circuit design and that you have some familiarity with SPICE simulation. It contains information about the Affirma™ RF circuit simulator as it is used within the Affirma analog circuit design environment. The Affirma RF circuit simulator is also known as SpectreRF in the software and in this manual.

SpectreRF analyses support the efficient calculation of the operating point, transfer function, noise, and distortion of common RF and communication circuits, such as mixers, oscillators, sample and holds, and switched capacitor filters.

Related Documents

The following documents can give you more information about SpectreRF and related products.

- To learn more about the Affirma analog circuit design environment, consult the [Affirma Analog Circuit Design Environment User Guide](#).
- To learn more about SpectreRF theoretical concepts, see [SpectreRF Theory](#).

SpectreRF Analyses

The SpectreRF analyses add new capabilities to Spectre simulation, such as direct, efficient computation of steady-state solutions and simulation of circuits that translate frequency. You use the SpectreRF simulator analyses in combination with the Fourier analysis capability of Spectre simulation and with the SpectreHDL behavioral modeling language.

The SpectreRF analyses add several kinds of functionality to Spectre simulation.

- Periodic Steady-State (PSS) analysis
- Periodic Small-Signal analyses
 - Periodic AC (PAC) analysis
 - Periodic S-Parameter (PSP) analysis
 - Periodic Transfer Function (PXF) analysis
 - Periodic Noise (Pnoise) analysis
- Periodic Distortion (Pdisto) analysis
- Quasi-Periodic Noise (QPnoise) analysis
- Envelope Following analysis

Periodic Steady-State (PSS) analysis is a large-signal analysis that directly computes the periodic steady-state response of a circuit. With PSS, simulation times are independent of the time constants of the circuit, so PSS can quickly compute the steady-state response of circuits with long time constants, such as high-Q filters and oscillators. You can perform sweeps using PSS; you can sweep frequency, a time period, or a variable.

After completing a PSS analysis, the SpectreRF simulator can model frequency conversion effects by performing one or more Periodic Small-Signal analyses (PAC, PSP, PXF, and Pnoise). The periodic small-signal analyses, Periodic AC analysis (PAC), Periodic S-Parameter analysis (PSP), Periodic Transfer Function analysis (PXF) and Periodic Noise analysis (Pnoise), are similar to the Spectre AC, SP, XF, and Noise analyses, but you can apply them to periodically driven circuits that exhibit frequency conversion. Examples of

important frequency conversion effects include conversion gain in mixers, noise in oscillators, and filtering using switched-capacitors.

Therefore, with Periodic Small-Signal analyses you apply a small signal at a frequency that may not be harmonically related (noncommensurate) to the periodic response of the undriven system, the clock. This small signal is assumed to be small enough so that it is not distorted by the circuit.

Periodic Distortion (Pdisto) analysis, a large-signal analysis, is used for circuits with multiple large tones. With Pdisto, you can model periodic distortion and include harmonic effects. (Periodic small-signal analyses assume the small signal you specify generates no harmonics). Pdisto computes both a large signal, the periodic steady-state response of the circuit, and also the distortion effects of a specified number of moderate signals, including the distortion effects of the number of harmonics that you choose.

With Pdisto, you can apply one or two additional signals at frequencies not harmonically related to the large signal, and these signals can be large enough to create distortion. This analysis is also called Quasi-Periodic Steady-State analysis.

Quasi-Periodic Noise (QPnoise) analysis is similar to the Pnoise analysis, except that it includes frequency conversion and intermodulation effects. QPnoise analysis is useful for predicting the noise behavior of mixers, switched-capacitor filters and other periodically or quasi-periodically driven circuits. QPnoise analysis linearizes the circuit about the quasi-periodic operating point computed in the prerequisite Pdisto analysis. It is the quasi-periodically time-varying nature of the linearized circuit that accounts for the frequency conversion and intermodulation.

Envelope Following analysis allows RF circuit designers to efficiently and accurately predict the envelope transient response of the RF circuits used in communication systems.

Periodic Large-Signal Analysis

There are two fundamental assumptions that apply to PSS analysis

- Periodicity
- Linearity

Understanding these two assumptions and their consequences allows you to anticipate whether you can apply PSS analysis successfully.

Periodicity Assumption

PSS analysis requires that all stimuli be periodic during the shooting interval and that the circuit support a T -periodic response, where T is the PSS analysis period. If the circuit is driven by multiple periodic stimuli, all the stimulus frequencies must be commensurate or coperiodic, and T must be the common period or some integer multiple of it. Simulation time increases when the T is long compared to the periods of the stimuli.

Some circuits, such as frequency dividers, generate subharmonics. PSS can simulate such circuits if you specify the period to be that of the subharmonic. For other circuits, such as delta-sigma modulators, the periodically driven circuits respond chaotically, and you must use transient analysis instead of PSS.

If the period T used by PSS is not an integral multiple of each period you specify for each selected time-varying independent source in the circuit, the simulator omits the PSS analysis and sends a warning. If you use the SpectreHDL behavioral modeling language to describe independent sources, then *you* must be sure that sources are coperiodic with the analysis period T . Failure to use coperiodic sources rarely causes PSS to generate incorrect results, but it usually prevents PSS from converging.

If a circuit is driven by T -periodic stimuli but lacks a T -periodic solution, you must use transient analysis instead of PSS.

PSS analysis might fail to converge if the circuit does not have a T -periodic solution, such as when the circuit contains an oscillator or is not driven by T -periodic stimuli. In such cases, you must find and fix the problem.

Linearity Assumption

The relationship between the initial and final points over the shooting interval must be close to linear. The more nonlinear the relationship between initial and final points, the longer the simulation time. If the relationship is sufficiently nonlinear, PSS analysis might not converge.

However, if PSS converges, there is no accuracy degradation or other negative consequence. Occasionally, delaying the starting time of the shooting interval can improve convergence. Arrange for the shooting interval to start when signals are quiescent or changing slowly.

Periodic Moderate-Signal Analysis (Pdisto)

Large and Moderate Signals in Periodic Distortion Analysis

Pdisto lets you model the distortion effects of multiple moderate signals on a large fundamental frequency with a single analysis. With moderate signals, you can model the effects of a few harmonics of a signal. This capability differs from small-signal analysis, in which the harmonics of the small signal are not considered in the analysis. In Pdisto, you select one signal to be the large fundamental frequency.

Choose Pdisto when you need to compute the steady-state responses of a circuit driven by two or more signals at unrelated frequencies. If only one periodic signal is large enough to create distortion, choose PSS followed by PAC or PXF.

Periodicity Not Assumed

Unlike for PSS analysis, Pdisto does not require that multiple periodic stimuli be commensurate or coperiodic.

Linearity Assumption

Pdisto starts with a PSS analysis that computes the response to the large signal. When computing the response to the large signal, Pdisto makes the same assumptions regarding linearity as any PSS analysis. The less linear the clock signal or fundamental frequency, the longer the simulation time. If the signal is sufficiently nonlinear, the PSS analysis might not converge.

The signals you apply in addition to the clock are assumed to be moderate-sized sinusoids.

Periodic Small-Signal Analysis

The SpectreRF simulation provides the periodic small-signal analyses PAC, PSP, PXF, and Pnoise. These analyses start by linearizing the circuit about the periodically time-varying operating point computed by a preceding PSS analysis. The periodic small-signal analyses can accurately model the frequency translation effects of a periodically time-varying circuit. Instead of using conventional small-signal analyses for amplifiers and filters, you can simulate such periodic circuits that exhibit frequency translation using periodic small-signal analyses.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

Circuits designed to translate from one frequency to another include mixers, detectors, samplers, frequency multipliers, phase-locked loops, and parametric oscillators. Such circuits are commonly found in wireless communication systems.

Other circuits that translate energy between frequencies as a side effect include oscillators, switched-capacitor and switched-current filters, chopper-stabilized and parametric amplifiers, and sample-and-hold circuits. These circuits are found in both analog and RF circuits.

Applying a periodic small-signal analysis is a two-step process.

- Initially, you ignore the small input or noise signals. You perform PSS analysis to compute the periodic steady-state response to the remaining large-signals (such as the clock or the LO).

During the initial PSS analysis, the circuit is linearized about the periodic large-signal operating point.

- Subsequent periodic small-signal analyses use this periodic operating point to predict the circuit response to a small sinusoid at an arbitrary frequency. You can perform any number of periodic small-signal analyses after calculating the periodic large-signal operating point.

The input signals for periodic small-signal analyses must be small enough so that the circuit does not respond to them in a significantly nonlinear fashion. Use input signals that are at least 10 dB smaller than the 1 dB compression point. This restriction does not apply to the signals you apply in the large-signal analysis. The only restriction for those signals is that they be periodic.

This two-step process is widely applicable because most circuits that translate frequency react in a strongly nonlinear manner to one stimulus (the LO or the clock) but react in a nearly linear manner to other stimuli (the inputs). A mixer is a typical example. Its noise and conversion characteristics improve if it is discontinuously switched between two states by the LO, yet it must respond linearly to the input signal over a wide dynamic range.

With the periodic small-signal analyses, unlike conventional small-signal analyses, there are many transfer functions between any single input and output. There are, in fact, as many transfer functions as there are harmonics in the periodic operating point (zero, one, or an infinite number). Usually, however, only one or two harmonics provide useful information. For example, when you analyze the down-conversion mixers found in receivers, you want to know about the transfer function that maps the input signal at the RF to the output signal at the IF, which is usually the LO minus the RF.

With SpectreRF simulation, you can request as few or as many harmonics as you want without reducing accuracy. The only exception to this rule is the Periodic Noise analysis. Periodic Noise analysis models the noise folding found in periodically varying circuits. If you request fewer harmonics, the accumulated total contains fewer folds.

Unlike harmonic balance methods, the combination of PSS and periodic small-signal analyses is efficient with circuits that respond in a strongly nonlinear manner to the LO or the clock. Consequently, you can use the SpectreRF simulation with strongly nonlinear circuits such as switched-capacitor filters, switching mixers, chopper-stabilized amplifiers, PLL-based frequency multipliers, sample-and-holds, and samplers.

Fundamental Assumptions for the Periodic Small-Signal Analyses

Periodic small-signal analyses make two fundamental assumptions that determine whether you can use them for your simulation.

Linearity

The periodic small-signal analyses all assume that the circuit responds linearly to the sinusoidal (PAC or PXF) or noise (Pnoise) stimulus. There is no such assumption concerning the periodic signals (such as the LO or the clock) applied in the PSS analysis.

Analysis Frequency

You can use the *maxacfreq* parameter of the PSS analysis to specify the highest frequency that the SpectreRF simulation uses in subsequent small-signal analyses. If the analysis frequency of a small-signal analysis is too high, simulation accuracy degrades. In such cases, the simulator does not perform the small-signal analysis and sends you a diagnostic message telling you to adjust *maxacfreq*. When you specify an appropriate value for *maxacfreq*, the PSS analysis chooses the PSS time-step value to ensure that subsequent small-signal analyses are accurate. For most simulations, adjusting *maxacfreq* is unnecessary because *maxacfreq* is never allowed to be smaller than 40 times the PSS fundamental. Specifying a very large *maxacfreq* causes both the PSS and small-signal analyses to run slowly and use a lot of memory.

With periodic small-signal analyses, the frequency of the stimulus and the response can be different. This is an important difference between periodic small-signal analyses and conventional small-signal analyses. For PAC and PXF, you use the *freqaxis* parameter to specify whether the results are shown versus the input frequency (*in*), the output frequency (*out*), or the absolute value of the output frequency (*absout*).

Description of SpectreRF Analyses

This section describes the individual SpectreRF analyses. SpectreRF simulation provides unique analyses that are useful on RF circuits. These analyses directly compute the steady-

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

state response of circuits and the small-signal behavior of circuits that exhibit frequency translation. The analyses are

- PSS, periodic steady-state analysis
- PSP, periodic S-parameter analysis
- PAC, periodic AC analysis
- PXF, periodic transfer function analysis
- Pnoise, periodic noise analysis
- Pdisto, periodic distortion analysis (also QPSS, quasi-periodic steady-state analysis)
- QPnoise, quasi-periodic noise analysis
- Envp, envelope following analysis

If a parameter or feature described in the *Spectre Reference* manual is not mentioned here, it is not accessible through the Affirma analog circuit design environment.

Periodic Steady-State Analysis

PSS analysis directly computes the periodic steady-state response of a circuit. SpectreRF simulation uses a technique called the shooting method to implement PSS analysis. This method is an iterative, time-domain method that finds an initial condition that directly results in steady-state. It starts with a guess of the initial condition.

The shooting method requires few iterations if the final state of the circuit after one period is a near-linear function of the initial state. This is usually true even for circuits that have strongly nonlinear reactions to large stimuli (such as the clock or the local oscillator). Typically, shooting methods need about five iterations on most circuits, and they easily simulate the nonlinear circuit behavior within the shooting interval. This is the strength of shooting methods over other steady-state methods such as harmonic balance.

A new approach to Fourier analysis makes PSS more accurate than previous methods with strongly nonlinear circuits. Cadence's new Fourier integral method approaches the accuracy of harmonic balance simulators for near-linear circuits, and far exceeds it for strongly nonlinear circuits.

PSS Synopsis

This analysis computes the periodic steady-state response of a circuit at a specified fundamental frequency. Also, it calculates the circuit's periodic operating point that is required to run PAC, PSP, PXF, and Pnoise analyses.

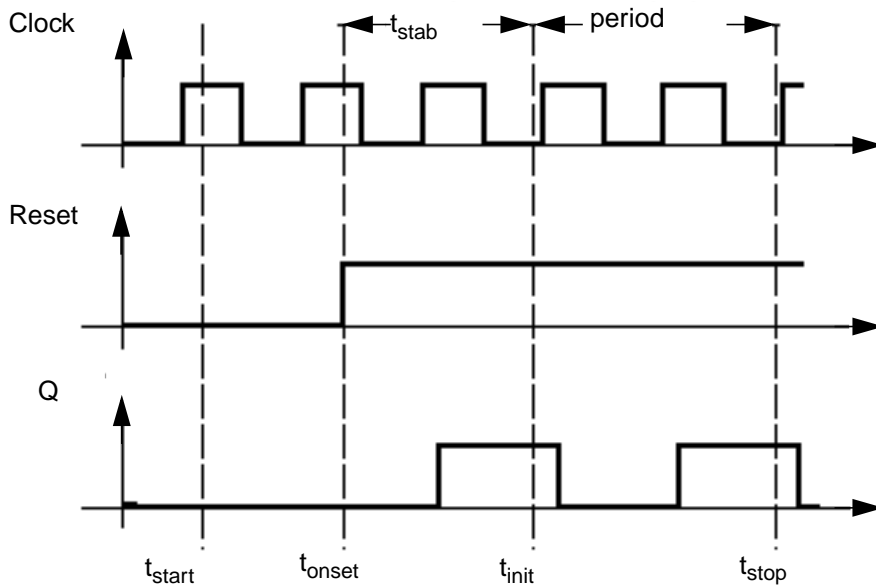
As shown in [Figure 1-1](#) on page 28, PSS starts by performing a standard transient analysis for the interval from t_{start} to $t_{onset} + t_{stab} + period$. Within the Affirma analog circuit design environment, you define t_{start} and t_{stab} in the *Simulation Interval Parameters* section of the [PSS Options form](#). PSS determines the *period* value from the fundamental frequency you specify. The onset of periodicity, t_{onset} , is the earliest time for which all sources are periodic. PSS automatically computes the t_{onset} value after it inspects all of the sources. As in transient analysis, the DC solution is the initial condition unless you specify otherwise.

PSS analysis then estimates the initial condition for another transient analysis with an interval *period*. For an accurate estimate for this initial condition, the final state of the circuit must closely match its initial state. PSS then performs a transient analysis, prints the maximum mismatch, and generates an improved estimate of the necessary initial condition if the convergence criteria are not satisfied. This procedure repeats until the simulation converges.

Typically, the simulation requires three to five such iterations to reach the steady-state circuit response. After completion, if you request it, PSS computes the frequency-domain response.

The initial transient analysis can direct the circuit to a particular steady-state solution and avoid unwanted solutions. The initial transient simulation also helps convergence by eliminating large, but fast decaying, modes in the circuit. In [Figure 1-1](#) on page 28, the initial transient analysis is executed from the starting time to t_{stop} . If you want to see the initial transient results, you can preserve them by setting *saveinit* to *yes* on the [PSS Options form](#). The steady-state results are always computed for the period from t_{init} to t_{stop} . By default, t_{start} and t_{stab} are set to zero, while t_{init} , t_{onset} , and t_{stop} are always automatically computed.

Figure 1-1 Initial Transient Analysis and Timing Relationships for PSS.



t_{start} Start time for the transient analysis that you specify. This value can be negative or positive and defaults to 0.

t_{onset} Time when the last stimulus waveform becomes periodic. This value is automatically determined by the simulator for built-in independent sources.

t_{stab} Additional time that you specify to let the circuit settle. This parameter is useful if you want a particular solution for a circuit that has multiple periodic solutions. A long t_{stab} can also improve convergence.

t_{init} Time when the PSS analysis begins. $t_{init} = t_{stab} + \max(t_{start}, t_{onset})$

period Analysis interval for the PSS analysis determined from the fundamental frequency that you specify.

t_{stop} Time when the PSS analysis ends. $t_{stop} = t_{init} + period$

In some circuits, the linearity of the relationship between the initial and final state depends on when the shooting interval begins. Theoretically, the starting time of the shooting interval does not matter, as long as it begins after the stimuli become periodic. Practically, it is better to start the shooting interval when signals are quiescent or changing slowly and to avoid

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

starting times when the circuit displays strongly nonlinear behavior. Choosing a poor starting time slows the analysis.

The accuracy of results depends only on the accuracy parameters you set, not on the number of harmonics you request. Except for a few new parameters, like *steadyratio* and *maxacfreq*, all PSS parameters work exactly as they work in transient analysis.

The *steadyratio* parameter specifies the maximum allowed mismatch between node voltages or current branches from the beginning of the steady-state period to its end. The *steadyratio* value is multiplied by the *Iteratio* and *reltol* parameter values to determine the convergence criterion. You can follow the progress of the steady-state iterations because the relative convergence norm is printed along with the actual mismatch value at the end of each iteration. The iterations continue until the convergence norm value is one.

The *maxacfreq* parameter automatically adjusts *maxstep* to reduce errors in subsequent periodic small-signal analyses. By default, *maxacfreq* is four times the frequency of the largest harmonic you request, but it is never less than 40 times the fundamental.

Periodic Distortion Analysis (Pdisto)

Pdisto models intermodulation distortion resulting from multiple moderately sized signals. A moderately sized signal is one that has a few harmonics that contribute significantly to the output. You might use Pdisto, for example, to measure how well an amplifier avoids magnification of unwanted frequencies.

If you treat one of the input signals as a small signal, a signal whose harmonics do not contribute significantly to the output, you can also model intermodulation distortion efficiently with PAC, but Pdisto has the following advantages that make it the analysis of choice for some situations:

- With Pdisto, you can measure the harmonic distortion and frequency translation effects created by multiple moderate-signal inputs, including all third order products. These measurements are impossible using PAC because PAC does not model the effects of small-signal harmonics.
- With Pdisto, you can measure the effects of multiple moderate signals. With PAC, you are restricted to modeling the effects of a single small signal on a fundamental you compute using PSS. Pdisto lets you model all moderate-signal inputs, including sums of sinusoids that are not periodic.

To see the difference between the information available with Pdisto and PAC, compare how the two approaches determine the output signals produced by input signals at 900MHz and 905MHz. The discussion below assumes you are interested in data only for fundamentals and their first harmonics.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

If you perform a PSS analysis for the 900MHz signal followed by a PAC analysis that applies 905 MHz as a small-signal, you get output data at each fundamental (900 MHz and 905 MHz) and at the first harmonic of the 900 MHz signal (1800 MHz). You also get output data about the following output signals created by frequency translation:

Frequency Translation Output Signals Computed by PAC

Frequency Pair	Frequency Translation from Difference	Frequency Translation from Sum
900 MHz, 905 MHz	5 MHz	1805 MHz
900 MHz, 1800 MHz	900 MHz	2700 MHz
905 MHz, 1800 MHz	895 MHz	2705 MHz

If you perform Pdisto with input signals at 900 MHz and 905 MHz, you get data about the first harmonic of the 905 MHz signal (1810 MHz) that PAC does not supply. You also get information about the following additional frequency translation signals created by interactions with the 1810 MHz harmonic.

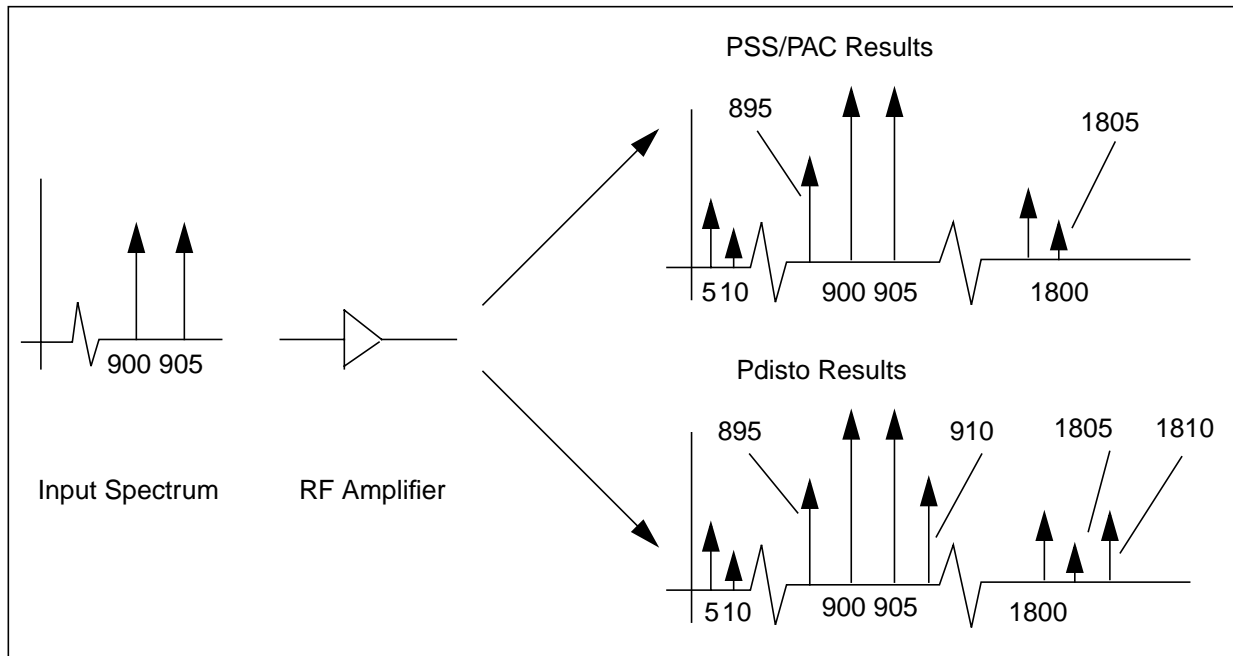
Additional Frequency Translation Output Signals Computed by Pdisto

Frequency Pair	Frequency Translation from Difference	Frequency Translation from Sum
900 MHz, 1810 MHz	910 MHz	2710 MHz
905 MHz, 1810 MHz	905 MHz	2715 MHz
1800 MHz, 1810 MHz	10 MHz	3610 MHz

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

The figure below shows that more information is available from Pdisto than from a PSS analysis followed by a PAC analysis.



With Pdisto, you select one of the input signals as the large signal. Usually, the signal you choose for the large signal has one or more of the following characteristics:

- It is not a sinusoidal.
- It causes the most nonlinearity.
- It causes the largest response.

The number of integrations required per iteration, and therefore the simulation time, increases rapidly as you specify more harmonics for the moderate signals. For example, if you model four moderate signals with k_1 , k_2 , k_3 , and k_4 numbers of harmonics, respectively, the number of integrations for the simulation is

$$(2k_2 + 1)(2k_3 + 1)(2k_4 + 1)$$

Intermodulation distortion that considers harmonics of the signals can also be performed using PSS. The following guidelines are helpful in deciding whether PSS or Pdisto might run faster and use less memory. If the signals are coperiodic and well separated in frequency, PSS is the better choice. If the signals are closely spaced, use Pdisto.

Periodic AC Analysis

The Periodic AC (PAC) analysis computes transfer functions for circuits that exhibit frequency translation. Such circuits include mixers, switched-capacitor filters, samplers, phase-locked loops, sample-and-holds, and similar circuits. A PAC analysis cannot be used alone. It must follow a PSS analysis. However, any number of periodic small-signal analyses can follow a PSS analysis.

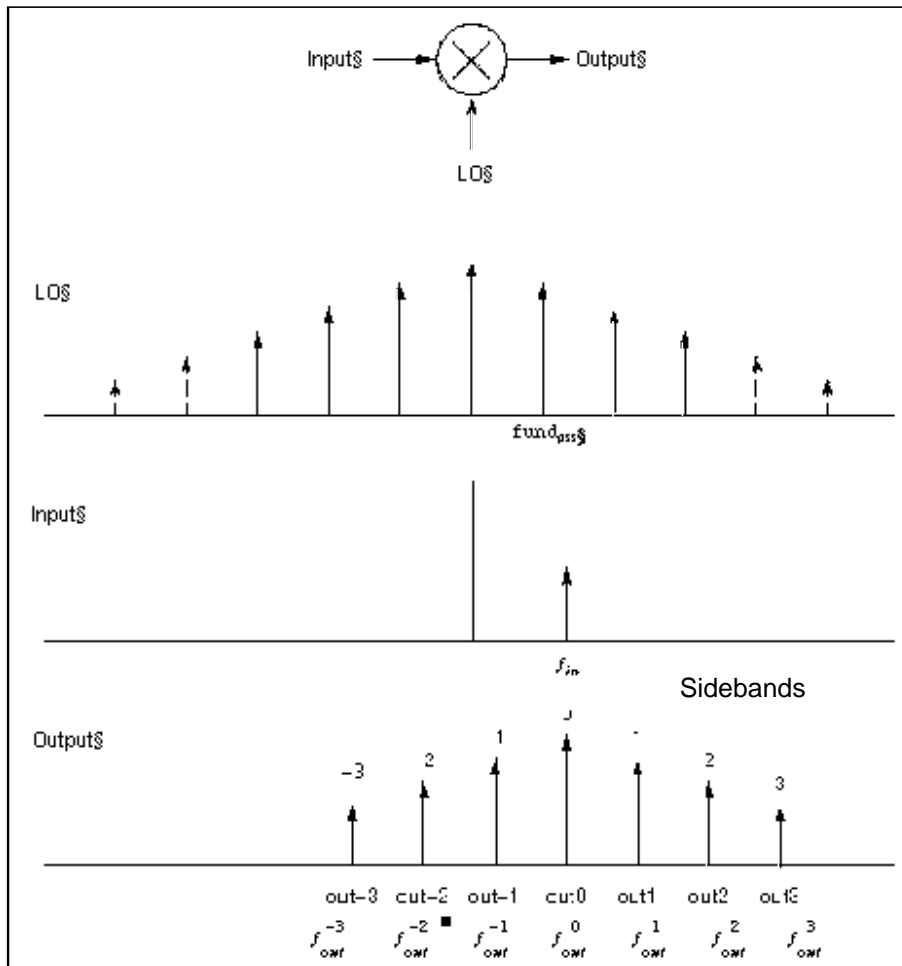
When you apply a small sinusoid to a linear time-invariant circuit, the steady-state response is a sinusoid at the same frequency. However, when you apply a small sinusoid to a linear circuit that is periodically time-varying, the circuit responds with sinusoids at many frequencies, as is shown in [Figure 1-2](#) on page 33.

Because PAC is a small-signal analysis, the magnitude and phase of each tone computed by PAC is linearly related to the magnitude and phase of the input signal. PAC computes a series of transfer functions, one for each frequency. These transfer functions are unique because the input and output frequencies are offset by the harmonics of the LO. The SpectreRF simulation labels the transfer functions with the offsets from the input signal in multiples of the LO fundamental frequency. These same labels identify the corresponding sidebands of the output signals. The labels are used as follows:

- For circuits in which the input and output are at the same frequency, such as switched-capacitor filters, the transfer function or sideband is labeled 0.
- For down-conversion mixers, the transfer function or sideband is labeled -1 because the output frequency is offset from the input frequency by -1 times the LO frequency.
- For up-conversion mixers, the transfer function or sideband is labeled +1.
- For samplers, sidebands far from zero might be used.

In [Figure 1-2](#) on page 33, all transfer functions from -3 to +3 are computed. As shown in the figure, the input signal is replicated and translated by each harmonic of the LO. In down-conversion mixers, the -1 sideband usually represents the IF output.

Figure 1-2 The small-signal response of a mixer as computed by PAC analysis.



PAC performance is not reduced if the input and LO frequencies are close or equal.

PAC Synopsis

You select the periodic small-signal output frequencies you want by specifying either the maximum sideband or an array of sidebands.

For a set of n integer numbers representing the sidebands k_1, k_2, \dots, k_n , the output frequency at each sideband is computed as $f_{out}^{k_{out}} = f_{in} + k \times fund_{pss}$

where

- f_{in} is the (possibly swept) input frequency.

- $fund_{pss}$ is the fundamental frequency used in the corresponding PSS analysis.

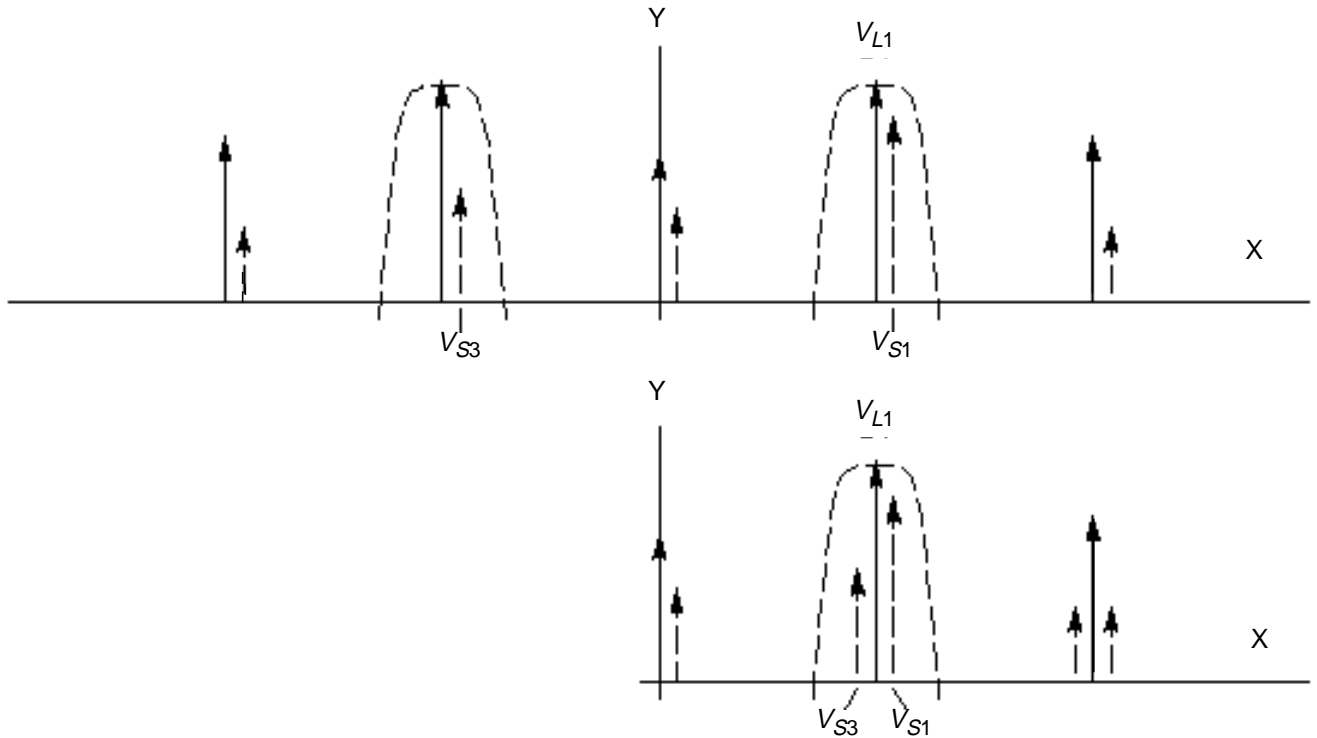
If you specify the maximum sideband value as k_{max} , all $2 \times k_{max} + 1$ sidebands from $-k_{max}$ to $+k_{max}$ are generated.

Intermodulation Distortion Computation

You can measure the intermodulation distortion of amplifiers and mixers with PAC. You can also measure intermodulation distortion with Pdisto by applying two large, closely spaced tones to the input and then measuring the third-order intermodulation products. The PSS/PAC approach is slightly different. You apply only one large tone in the PSS analysis. PSS is therefore faster than Pdisto. Once PSS computes the circuit response to one large tone, then PAC applies the second tone close to the first. If you consider the small input signal to be one sideband of the large input signal, then the response at the other sideband is the third-order intermodulation distortion, as shown in [Figure 1-3](#) on page 35.

In this figure, V_{L1} is the fundamental of the response due to the large input tone. V_{S1} is the fundamental of the response due to the small input tone and is the upper sideband of V_{L1} . V_{S3} is the lower sideband of V_{L1} (in this case, it is the -2 sideband of the response due to the small tone). V_{S3} represents the intermodulation distortion. In the lower figure, all of the signals are mapped into positive frequencies, which is the most common way of viewing such results.

Figure 1-3 Intermodulation Distortion Measured with PAC Analysis



Intermodulation distortion is efficiently measured by applying one large tone (L1), performing a PSS analysis, and then applying the second small tone (S1) with a PAC analysis. In this case, the first tone drives the circuit hard enough to cause distortion and the second tone is used to measure only the intermodulation distortion. Once V_{L1} , V_{S1} , and V_{S3} are measured in dB at the output, the output third-order intercept point is computed using the following equation.

$$IP_3 = V_{L1} - \frac{V_{S3} + V_{S1}}{2}$$

In the equation, V_{L1} , V_{S1} , and V_{S3} must be given in some form of decibels. Currently, in the Affirma analog circuit design environment, dBV is used. In this example, V_{L1} , V_{S1} , and V_{S3} are given in dBV. Consequently, the intercept point is also computed in dBV. If V_{L1} , V_{S1} , and V_{S3} are given in dBm, the resulting intercept point is computed in dBm.

The intermodulation distortion of a mixer is measured in a similar manner, except that the PSS analysis must include both the LO and one large tone. For an example of measuring the intermodulation distortion of a mixer, see Chapter 4, “[Simulating Mixers with SpectreRF simulation.](#)”

If you must measure intermodulation distortion using two large tones to compare against bench measurements, you can run PSS analysis with two large tones. However, do so only when absolutely necessary. At other times, use the quicker PSS/PAC approach.

Periodic S-Parameter Analysis

The Periodic S-Parameter (PSP) analysis is used to compute scattering and noise parameters for n-port circuits that exhibit frequency translation. Such circuits include mixers, switched-capacitor filters, samplers, phase-locked loops, and other similar circuits.

PSP is a small-signal analysis similar to SP analysis, except, as for PAC and PXF analyses, the circuit is first linearized about a periodically varying operating point as opposed to a simple DC operating point. Linearizing about a periodically time-varying operating point allows the computation of S-parameters between circuit ports that convert signals from one frequency band to another.

PSP analysis also calculates noise parameters in frequency-converting circuits. PSP computes noise figure (both single-sideband and double-sideband), input referred noise, equivalent noise parameters, and noise correlation matrices. The noise features of PSP analysis, as for Pnoise analysis but unlike SP analysis, include noise folding effects due to the periodically time-varying nature of the circuit.

Computing the n-port S-parameters and noise parameters of a periodically varying circuit is a two step process.

- First, the small stimulus is ignored and the periodic steady-state response of the circuit to possibly large periodic stimulus is computed using PSS analysis.

As a normal part of the PSS analysis, the periodically time-varying representation of the circuit is computed and saved for later use.

- Then, using the PSP analysis, small-signal excitations are applied to compute the n-port S-parameters and noise parameters.

A PSP analysis cannot be used alone, it must follow a PSS analysis. However, any number of periodic small-signal analyses such as PAC, PSP, PXF, and Pnoise, can follow a single PSS analysis.

Unlike other analyses in Spectre, the PSP analysis can sweep only frequency.

PSP Synopsis

For a PSP analysis, you need to specify the port and port harmonic relations. Select the ports of interest by setting the *port* parameter. Set the periodic small-signal output frequencies of interest by setting the *portharmsvec* or the *harmsvec* parameters.

For a given set of n integer numbers representing the harmonics K_1, K_2, \dots, K_n , the scattering parameters at each port are computed at the frequencies $f(scattered) = f(rel) + K_i * fund(pss)$, where $f(rel)$ represents the relative frequency of a signal incident on a port, $f(scattered)$ represents the frequency to which the relevant scattering parameter represents the conversion, and $fund(pss)$ represents the fundamental frequency used in the corresponding PSS analysis.

Thus, when analyzing a down-converting mixer, with signal in the upper sideband, and sweeping the RF input frequency, the most relevant harmonic for RF input is $K_i = 1$ and for IF output $K_i = 0$. Hence we can associate $K_2 = 0$ with the IF port and $K_1 = 1$ with the RF port. S_{21} will represent the transmission of signal from the RF to IF, and S_{11} the reflection of signal back to the RF port. If the signal was in the lower sideband, then a choice of $K_1 = -1$ would be more appropriate.

You can use either the *portharmsvec* or the *harmsvec* parameters to specify the harmonics of interest. If you give *portharmsvec*, the harmonics must be in one-to-one correspondence with the ports, with each harmonic associated with a single port. If you specify harmonics with the optional *harmsvec* parameter, then all possible frequency-translating scattering parameters associated with the specified harmonics are computed.

For PSP analysis, the frequencies of the input and of the response are usually different (this is an important way in which PSP differs from SP). Because the PSP computation involves inputs and outputs at frequencies that are relative to multiple harmonics, the *freqaxis* and *sweepstype* parameters behave somewhat differently in PSP than they do in PAC and PXF.

The *sweepstype* parameter controls the way the frequencies are swept in PSP analysis. Specifying *relative* sweep, sweeps relative to the analysis harmonics (not the PSS fundamental). Specifying *absolute* sweep, sweeps the absolute input source frequency. For example, with a PSS fundamental of 100MHz, the *portharmsvec* set to [9 1] to examine a down-converting mixer, *sweepstype=relative*, and a sweep *range* of $f(rel) = 0 \rightarrow 50\text{MHz}$, then S_{21} would represent the strength of signal transmitted from the input port in the range 900- > 950MHz to the output port at frequencies 100- > 150MHz.

Using *sweepstype=absolute* and sweeping the frequency from 900- > 950MHz would calculate the same quantities, since $f(abs) = 900 \rightarrow 950\text{MHz}$, and $f(rel) = f(abs) - K_1 * fund(pss) = 0 \rightarrow 50\text{MHz}$, because $K_1 = 9$ and $fund(pss) = 100\text{MHz}$.

In a PSP analysis, it is not usually necessary to sweep frequency over more than one fundamental PSS period.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

The *freqaxis* parameter is used to specify whether the results should be output versus the scattered frequency at the input *port(in)*, the scattered frequency at the output *port(out)*, or the absolute value of the frequency swept at the input *port(absin)*.

Unlike in PAC, PXF, and Pnoise analyses, increasing the number of requested ports and harmonics *increases* the simulation time substantially.

To insure accurate results in PSP analysis, you should set the *maxacfreq* parameter for the corresponding PSS analysis to guarantee that $|max\{f(scattered)\}|$ is less than the *maxacfreq* parameter value, otherwise the computed solution might be contaminated by aliasing effects.

PSP analysis computes noise figures, equivalent noise sources, and noise parameters. The noise computation, which is skipped only when the *donoise* parameter is set to no, requires additional simulation time.

If

N_o = total output noise at frequency f

N_s = noise at the output due to the input probe (the source)

N_{sj} = noise at the output due to the image harmonic at the source

N_{so} = noise at the output due to harmonics other than input at the source

N_l = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor (single side band)

NF = noise figure (single side band)

F_{dsb} = double sideband noise factor

NF_{dsb} = double sideband noise figure

F_{ieee} = IEEE single sideband noise factor

NF_{ieee} = IEEE single sideband noise figure

then,

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

$$IRN = \sqrt{\frac{N_o^2}{G^2}}$$

$$F = \frac{(N_o^2 - N_i^2)}{N_s^2}$$

$$NF = 10\log F$$

$$F_{dsb} = \frac{(N_o^2 - n_i^2)}{(N_s^2 + N_{si}^2)}$$

$$NF_{dsb} = 10\log F_{dsb}$$

$$F_{ieee} = \frac{(N_o^2 - N_i^2 - N_{so})}{N_s^2}$$

$$NF_{ieee} = 10\log(F_{ieee})$$

When the results are output, IRN is named in , G is named $gain$, F , NF , F_{dsb} , NF_{dsb} , F_{ieee} , and NF_{ieee} are named F , NF , F_{dsb} , NF_{dsb} , F_{ieee} , and NF_{ieee} respectively.

Note that the gain computed by PSP is the voltage gain from the actual circuit input to the circuit output, not the gain from the internal port voltage source to the output.

To insure accurate noise calculations, You need to set the *maxsideband* or *sidebands* parameters to include the relevant noise folding effects. The *maxsideband* parameter is only relevant to the noise computation features of PSP.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep.

Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can specify a step size parameter (*step*, *lin*, *log*, *dec*) to determine whether

the sweep is linear or logarithmic. If you do not give a step size parameter, the sweep is linear when the ratio of stop to start values is less than 10:1, and logarithmic when this ratio is equal to or greater than 10:1.

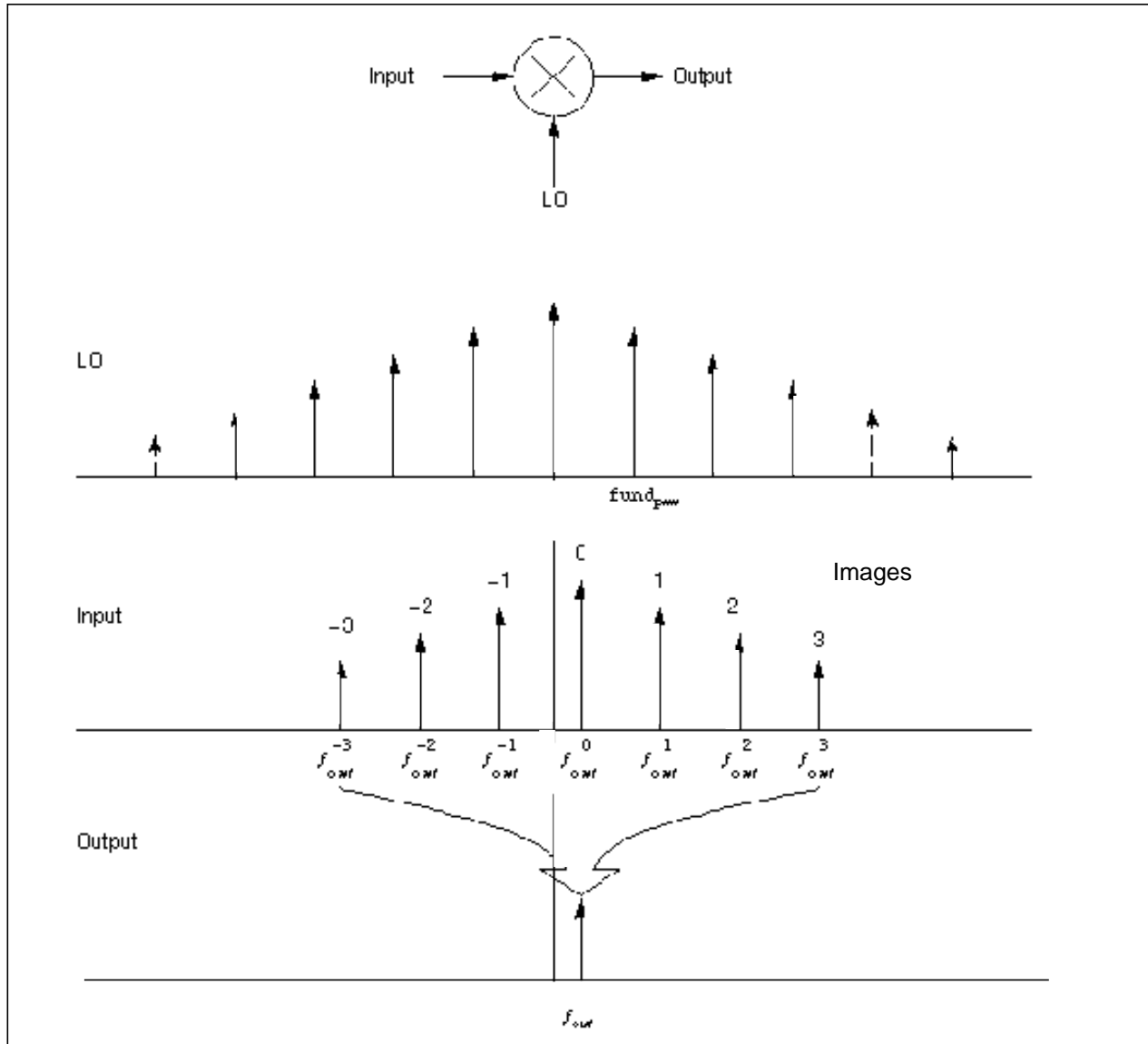
Alternatively, you may specify particular values for the *sweep* parameter using the *values* parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Periodic Transfer Function Analysis

Whereas a conventional AC analysis computes the response from a single stimulus to every node in the circuit, a conventional transfer function (XF) analysis computes the transfer function from every source in the circuit to a single output. The difference between PAC and PXF analysis is similar. The PXF analysis computes the transfer functions from any source at any frequency to a single output at a single frequency. Like PAC analysis, PXF analysis models frequency conversion effects. This is illustrated in [Figure 1-4](#) on page 41. PXF computes such useful quantities as conversion efficiency, image and sideband rejection, and LO feed through and power supply rejection.

The output is sensitive to signals at many frequencies at the input of the mixer. The input signals are replicated and translated by each harmonic of the LO. The signals shown in the figure are those that end up at the output frequency.

Figure 1-4 Mixer Output signals Shown by PXF Analysis



PXF analysis can measure conversion gains, especially those from the input source to the output. PXF computes the conversion gain of the specified sideband as well as various unwanted images, including the baseband feed through. It also computes the coupling from other inputs such as the LO and the power supplies. These computations model frequency translation, and you can determine the sensitivity of the output to either up-converted or down-converted noise from the power supplies or the LO.

A PXF analysis, like the PAC analysis, must follow a PSS analysis.

PXF Synopsis

The output variable you measure can be voltage or current, and the variable frequency is not limited by the period of the large-periodic solution. When you sweep a selected output frequency, you can select the periodic small-signal input frequencies by specifying either the maximum sideband or an array of sidebands.

For a set of n integer numbers representing the sidebands k_1, k_2, \dots, k_n , the input signal frequency at each sideband is computed as $f^{k_{in}} = f_{out} + k \times fund_{pss}$

where

- f_{out} represents the (possibly swept) output signal frequency
- $fund_{pss}$ represents the fundamental frequency used in the corresponding PSS analysis

When you analyze a down-converting mixer and sweep the IF output frequency, $k = +1$ for the RF input represents the first upper sideband, and $k = -1$ for the RF input represents the first lower sideband. If you specify k_{max} as the maximum sideband value, you select all $2 \times k_{max} + 1$ sidebands from $-k_{max}$ to $+k_{max}$. The number of sidebands you request does not affect accuracy, so request only sidebands that you want to see.

You can specify the output with a pair of nodes or a probe component. Any component with two or more terminals can be a voltage probe. Any component that naturally computes current as an internal variable, such as a voltage source, can be a current probe. In the Affirma analog circuit design environment, you set probes on the Choosing Analysis form when you specify input or output sources, and also when you specify positive or negative output nodes.

The *stimuli* parameter on the [PXF Options form](#) specifies the inputs for the transfer functions. You select from the following choices:

- *stimuli = sources* specifies that the sources present in the circuit are used. The *xmag* component parameter provided by the sources can adjust the computed gain to compensate for gains or losses in a test fixture.
- *stimuli = nodes_and_terminals* specifies that all possible transfer functions are computed. Use this option when you cannot anticipate which transfer functions you might need to examine.

Transfer functions for nodes are computed assuming that a unit magnitude flow (current) source is connected from the node to ground. Transfer functions for terminals are computed assuming that a unit magnitude value (voltage) source is connected in series with the terminal. By default, PXF computes the transfer functions from a small set of terminals.

Periodic Noise Analysis

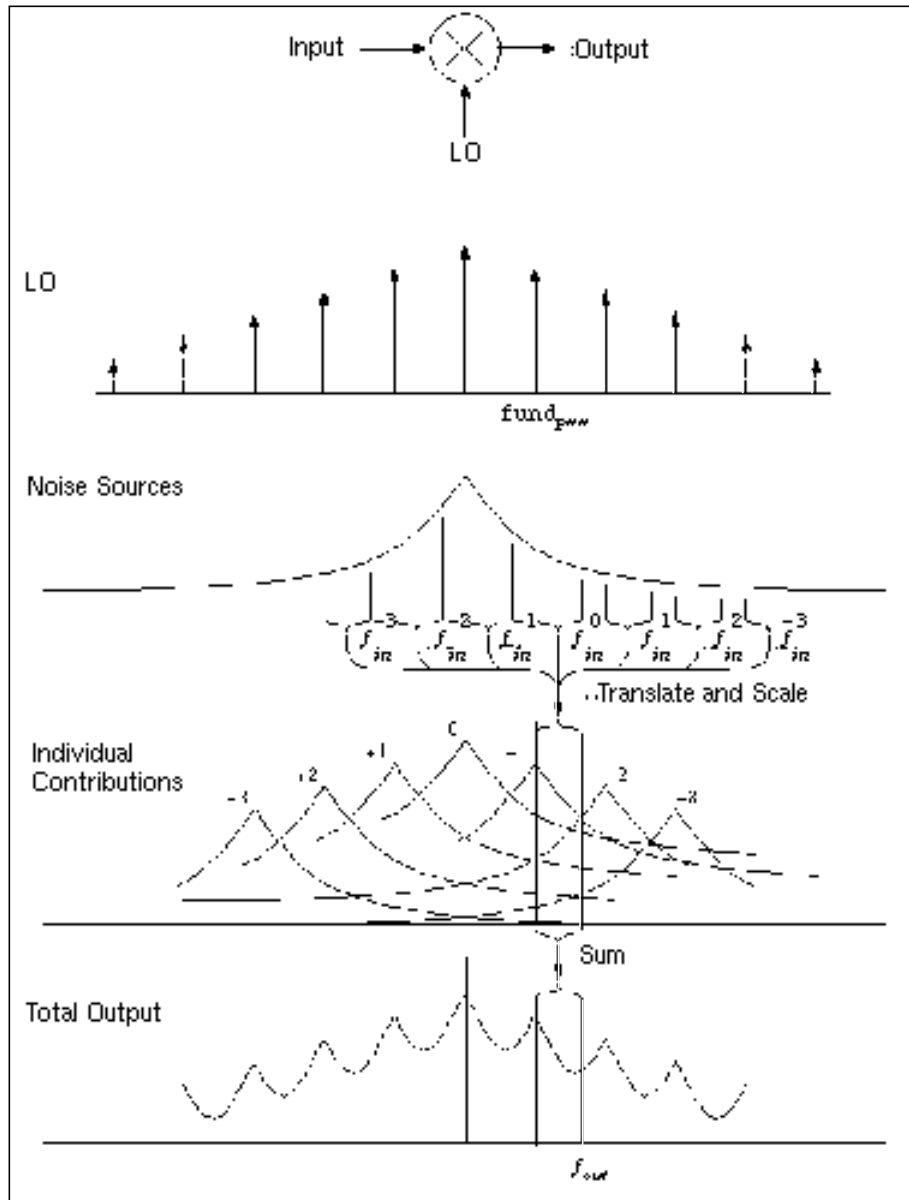
The Periodic Noise analysis (Pnoise) is similar to conventional noise analysis except that it models frequency conversion effects. It can compute the phase noise of oscillators as well as the noise behavior of mixers, switched-capacitor filters, chopper-stabilized amplifiers, and other similar circuits.

PSS analysis linearizes the circuit about the periodic operating point. The periodically time-varying qualities of the linearized circuit create the frequency conversion. Then the effect of a periodically time-varying bias point on component-generated noise is simulated by Pnoise.

Initially, PSS computes the response to a large periodic signal such as a clock or a LO. These results are labeled LO and shown in [Figure 1-5](#) on page 44. The subsequent Pnoise analysis computes the resulting noise performance.

In periodic systems, there are two effects that act to translate noise in frequency. First, for noise sources that are bias dependent, such as shot noise sources, the time-varying operating point modulates the noise sources. Second, the transfer function from the noise source to the output is also periodically time-varying and modulates the noise source contribution to the output.

Figure 1-5 How noise is moved around by a mixer



The time-average of the noise at the output is computed as a spectral density versus frequency. You identify the output by specifying a probe component or a pair of nodes. To specify the output with a probe, the preferred approach, use the *oprobe* parameter. If the output is voltage (or potential), choose a *resistor* or a *port* component for the output probe. If the output is current (or flow), choose a *vsource* or *iprobe* component for the output probe.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

The reference sideband (*refsideband*) specifies which conversion gain is used to compute the input-referred noise, the noise factor, and the noise figure. The reference sideband satisfies the following equation.

$$|f(input)| = |f(output) + refsideband \times fund(pss)|$$

Use *refsideband*=0 when the input and output are at the same frequency, such as with amplifiers and filters. When *refsideband* does not equal 0, the single side band noise figure is computed.

The noise analysis computes the total noise at the output, which includes contributions from the input source and the output load. The output noise that is attributable to each noise source is specified in the results. If the input source is identified, the input-referred noise is also computed, and this includes the noise from the input source itself. Finally, if the input source is identified and is also noisy, the noise factor and noise figure are computed.

Thus, if

N_o = Total output noise

N_s = Noise at the output due to the input source

N_l = Noise at the output due to the load

IRN = Input referred noise

G = Voltage gain of input node voltage to output node voltage

F = Noise factor

NF = Noise figure

then

$$IRN = \sqrt{N_o^2 \div G^2}$$

$$NF = 10\log(F)$$

$$F = \sqrt{\frac{N_o^2 - N_l^2}{N_s^2}}$$

Pnoise Synopsis

Noise can mix with each harmonic of the periodic drive signal from the PSS analysis and appear at the output frequency. However, Pnoise analysis models only noise that mixes with a set of harmonics that you normally specify with the *maxsideband* parameter, but which you might specify with the *sidebands* parameter in special circumstances. If K_i represents sideband i , then

$$f(\text{NoiseSource}) = f(\text{out}) + K_i \text{fund}(\text{pss})$$

The *maxsideband* parameter specifies the maximum $|K_i|$ included in the *Pnoise* calculation. Therefore, Pnoise ignores noise at frequencies less than $f(\text{out}) - \text{maxsideband fund}(\text{pss})$ and greater than $f(\text{out}) + \text{maxsideband fund}(\text{pss})$. If you specify sidebands with the *sidebands* parameter, then Pnoise includes only the specified sidebands in the calculation. When you specify *sidebands* parameter values, be careful not to omit any sidebands that might contribute significant output noise.

In the Affirma analog circuit design environment, you specify the *maxsideband* parameter with the *Maximum sideband* specification in the Choosing Analyses form. You specify the *sidebands* parameter in the *Array of Sidebands* type-in field.

Increasing the number of harmonics computed by Pnoise improves simulation accuracy because it increases the number of harmonics used to compute the effects of noise folding. Noise folding is noise mixing with the harmonics into the frequency range of interest.

Noise Figure

When you use Pnoise analysis to compute the noise factor or noise figure of a circuit, and the load generates noise, specify the output with the *oprobe* parameter rather than using a pair of nodes. Using the *oprobe* parameter explicitly specifies the load as the output probe. This is preferable because it excludes the noise of the load from the calculation of the noise figure.

As an alternative, you can specify the output with a pair of nodes and make the load component a noiseless resistor. Results with this approach are similar to those computed if you specify a resistor or a port as the output probe (load). The only difference is that the noiseless resistor is considered noiseless for other noise calculations, such as total output noise and input-referred noise, and the resistor is noiseless at all frequencies. When you specify a conventional resistor or port for the load, its noise is subtracted from only the noise factor and noise figure calculations, and only at the output frequency. Consequently, noise from load frequencies other than the output frequency can appear at the output frequency if the circuit has a nonlinear output impedance.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

Pnoise computes the single-sideband noise figure. To match the IEEE definition of noise figure, you must use a *port* as the input probe and a *resistor* or a *port* as the output probe. In addition, the input port noise temperature must be 290 K (*noisetemp* = 16.85) and have no excess noise. (You must not specify *noisevec* and *noisefile* on the input port.) The 290K temperature is the average noise temperature of an antenna used for terrestrial communication. However, you can specify the input port noise temperature to be any appropriate value. For example, the noise temperature for antennas pointed at satellites is usually much lower.

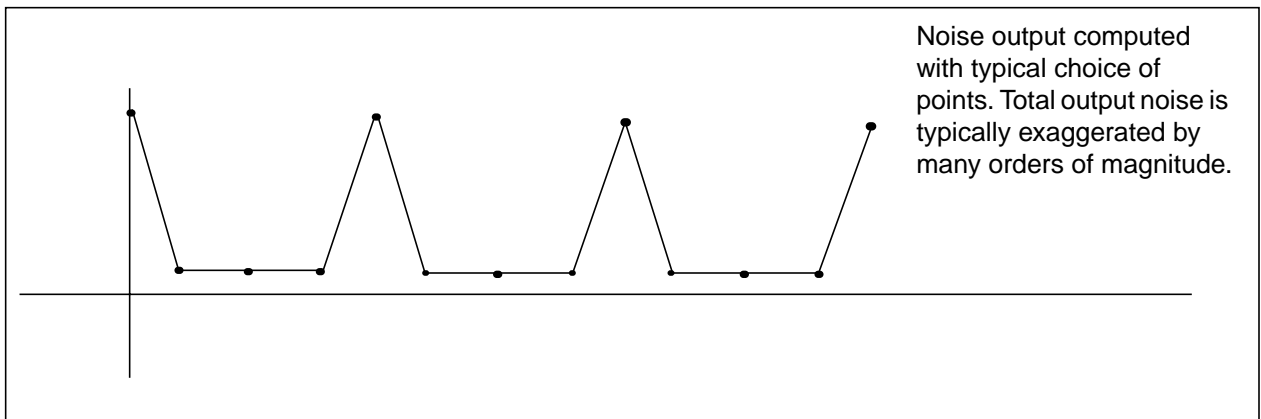
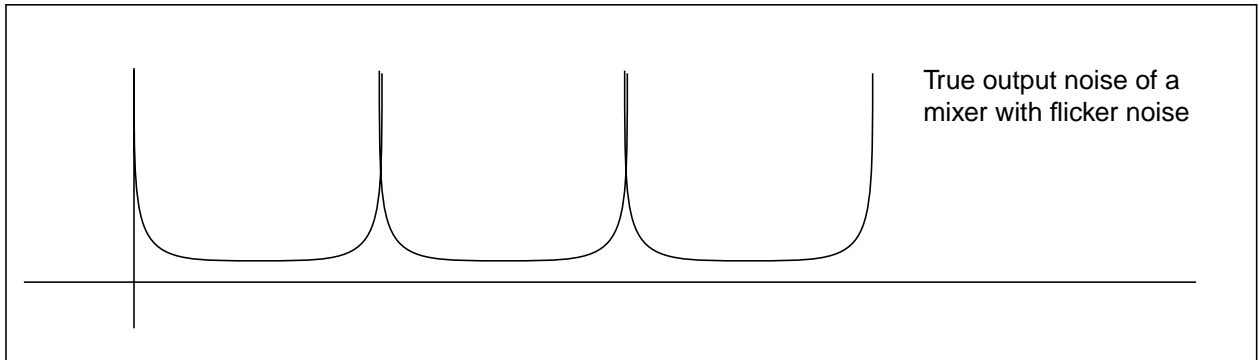
Flicker Noise

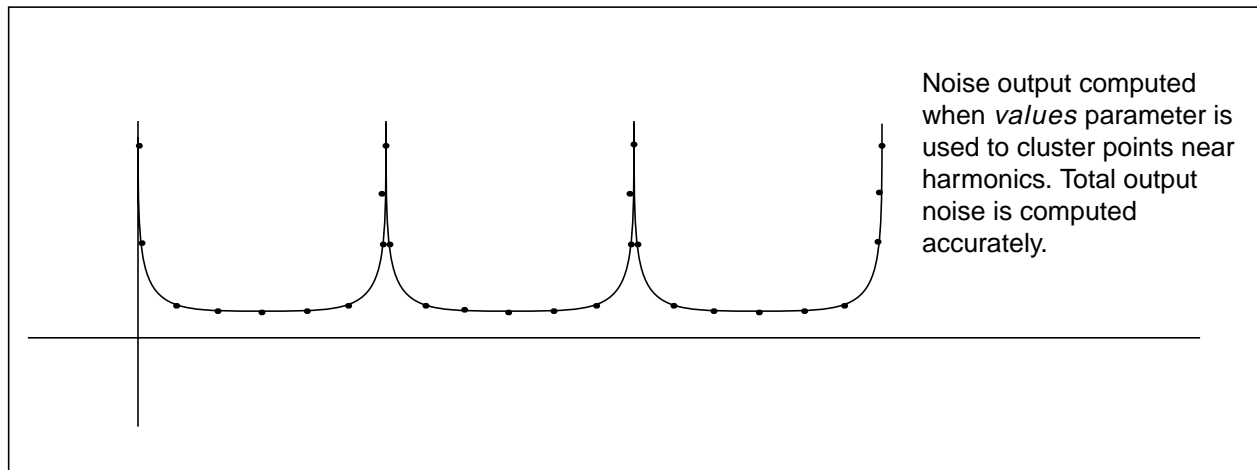
To avoid inaccurate results with Pnoise analysis on a circuit that mixes $1/f$ noise up to the carrier or its harmonics, place a cluster of frequencies near each harmonic to resolve the noise peaks accurately, but avoid putting frequency points precisely on the harmonics. In addition, choose Pnoise start and stop frequencies to avoid placing points precisely on the harmonics of the periodic drive signal. Then use the *values* parameter to specify a vector of additional frequency points near the harmonics. In the Affirma analog circuit design environment, this parameter is set in the Add Specific Points field in the Choosing Analyses

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

form. The effect of specifying appropriate additional frequency points is shown in the following three diagrams.





Quasi-Periodic Noise Analysis

The Quasi-Periodic Noise (QPnoise) analysis is similar to the conventional noise analysis, except that it includes frequency conversion and intermodulation effects. Hence is it useful for predicting the noise behavior of mixers, switched-capacitor filters, and other periodically or quasi-periodically driven circuits.

QPnoise analysis first linearizes the circuit about the quasi-periodic operating point computed in the prerequisite Pdisto analysis. It is the quasi-periodically time-varying nature of the linearized circuit that accounts for the frequency conversion and intermodulation. In addition, the affect of a quasi-periodically time-varying bias point on the noise generated by the various components in the circuit is also included.

The time-average of the noise at the output of the circuit is computed in the form of a spectral density versus frequency. The output of the circuit is specified with either a pair of nodes or a probe component. To specify the output of a circuit with a probe, specify it using the *oprobe* parameter. If the output is voltage (or potential), choose a *resistor* or a *port* as the output probe. If the output is current (or flow), choose a *vsource* or *iprobe* as the output probe.

If the input-referred noise is desired, specify the input source using the *iprobe* parameter. Currently, you can use only a *vsource*, an *isource*, or a *port* as an input probe. If the input source is noisy, as is a *port*, the noise analysis will compute the noise factor (F) and noise figure (NF). To match the IEEE definition of noise figure, the input probe must be a port with no excess noise and you need to set its *noisetemp* to 16.85C (290K). In addition, the output load must be a *resistor* or *port* and must be identified as the *oprobe*.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

The reference sideband (*refsideband*) specifies which conversion gain is used when computing input-referred noise, noise factor, and noise figure. The reference sideband satisfies:

$$|f(input)| = |f(out) + refsideband\ frequency\ shift|$$

The reference sideband option (*refsidebandoption*) specifies whether to consider the input at the frequency or the input at the individual quasi-periodic sideband specified. Note that different sidebands might lead to the same frequency.

Sidebands are vectors in QPnoise analysis.

Assume you have one large tone and one moderate tone in a Pdisto analysis. A sideband might be $[K_1\ K_2]$. It gives the frequency as follows

$$K_1 * fund(large\ tone\ of\ Pdisto) + K_2 * fund(moderate\ tone\ of\ Pdisto)$$

Use *refsideband*=[0 0 ...] when the input and output of the circuit are at the same frequency (such as with amplifiers and filters). When *refsideband* differs from the 0 vector, the single side-band noise figure is computed.

The noise analysis always computes the total noise at the output, which includes contributions from the input source and the output load. The amount of the output noise that is attributable to each noise source in the circuit is also computed and output individually. If the input source is identified, the input-referred noise is computed, which includes the noise from the input source itself. Finally, if the input source is identified and is noisy, the noise factor and noise figure are computed.

Thus if

N_o = total output noise

N_s = noise at the output due to the input probe (the source)

N_l = noise at the output due to the output probe (the load)

IRN = input referred noise

G = gain of the circuit

F = noise factor

NF = noise figure

then,

$$IRN = \sqrt{\frac{N_o^2}{G^2}}$$

$$F = \frac{(N_o^2 - N_i^2)}{N_s^2}$$

$$NF = 10\log F$$

When the results are output, N_o is named *out*, IRN is named *in*, G is named *gain*, F is named F , and NF is named NF . The computation of *gain* and IRN in QPnoise analysis assumes that the circuit under test is impedance-matched to the input source. This can introduce inaccuracy into the *gain* and IRN computation.

Unlike other analyses in Spectre, the QPnoise analysis can sweep only frequency.

QPnoise Synopsis

The optional terminals (p and n) specify the output of the circuit. If you do not give the terminals, then you must specify the output with a probe component.

In practice, noise can mix with each of the harmonics of the quasi-periodic drive signal applied in the Pdisto analysis and end up at the output frequency. However, the QPnoise analysis only includes the noise that mixes with a finite set of harmonics that you typically specify using the *maxsideband* parameter; but, in special circumstances, you might specify them with the *sidebands* parameter. If K_i represents sideband i , then

$$f(\text{noise_source}) = f(\text{out}) + K_i * \text{fund}(\text{clock of pdisto}) + \text{moderate tones}$$

The *maxsideband* parameter specifies the maximum $|K_i|$ included in the QPnoise calculation. If you specify selected sidebands using the *sidevec* parameter, then only those sidebands are included in the calculation.

You should take care when specifying the *sidevec* parameter because the results will be in error if you do not include a sideband that contributes significant noise to the output.

The number of requested sidebands does not change substantially the simulation time.

You can specify sweep limits by giving the end points or by providing the center value and the span of the sweep.

Steps can be linear or logarithmic, and you can specify the number of steps or the size of each step. You can specify a step size parameter (*step*, *lin*, *log*, *dec*) to determine whether the sweep is linear or logarithmic. If you do not give a step size parameter, the sweep is linear when the ratio of stop to start values is less than 10:1, and logarithmic when this ratio is equal to or greater than 10:1.

Alternatively, you may specify the particular values for the *sweep* parameter using the *values* parameter. If you give both a specific set of values and a set specified using a sweep range, the two sets are merged and collated before being used. All frequencies are in Hertz.

Envelope Following Analysis

RF circuit designers who simulate the modulation schemes of communications systems want efficient and accurate prediction of the envelope transient response of RF circuits. Typically, these modulation schemes are difficult to simulate because they are clocked at a frequency whose period is orders of magnitude smaller than the time interval of interest. Using classical transient simulation for these circuits is time-consuming, and new RF analyses, such as PSS and PDISTO, might not work directly because the modulation signal is sometimes stochastic and might be neither periodic nor quasi-periodic.

Envelope Following Analysis permits efficient simulation of these modulation schemes and reduces the simulation time without compromising accuracy. This analysis depends on the property that circuit behavior in a given high-frequency clock cycle is similar, but not identical, to the behavior of the preceding and following clock cycles. In particular, the “envelope” of the high-frequency clock can be followed by accurately computing the circuit behavior over occasional cycles that accurately model the fast transient behavior. The slow-varying modulation is accurately followed by a piecewise polynomial. As a result, the spectrum of the circuit response can be obtained by combining the piecewise polynomial and the integration of the occasional clock cycles.

To compute the envelope response of a circuit, you specify the analysis clock name. The simulator automatically determines the clock period by examining all the sources with the specified name. The envelope response is computed over the interval from Start to Stop. If the interval you specify is not a multiple of the clock period, it is rounded off to the nearest multiple before the Stop time. The initial condition is assumed to be the DC steady-state solution unless you specify otherwise.

Envelope Following Analysis is most efficient for circuits whose modulation bandwidth is orders of magnitude lower than the clock frequency. This is typically the case, for example, in circuits whose clock is the only fast varying signal, and whose other input signals have a spectrum with a frequency range orders of magnitude lower than the clock frequency.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

In general, Envelope Following Analysis is not intended for circuits working with multiple fundamentals. However, like Periodic Steady-State (PSS) Analysis, you can use it for specific classes of circuits that operate with multiple fundamentals.

You can, for example, use Envelope Following Analysis when the multiple fundamentals are commensurate. In this situation, you use the greatest common denominator of all fundamental frequencies as the clock frequency. This is similar to computing the beat frequency for PSS analysis. The analysis efficiency is also analogous to PSS analysis. At each integration of the clock period, many fast cycles might be required. The analysis efficiency depends on how many fast cycles are required for the beat period. Greater numbers of fast cycles decrease the efficiency of the analysis.

Envelope Following analysis differs from PSS in that you can give the fundamentals the same name by using the *clockname* parameter, and the simulator then computes the beat frequency automatically. With PSS analysis, in similar circumstances, you must either specify the beat frequency or have it computed from fundamental tones that you specify in the UI.

There is a second situation when Envelope Following Analysis is useful for circuits with multiple fundamentals. This is when the down conversion of two closely placed frequencies can also generate a slow-varying modulation envelope whose frequency is orders of magnitude lower than the input frequencies. In this case, you can use Envelope Following Analysis to trace out the modulation envelope by choosing either of the fast-varying signals as the clock. However, you usually choose the signal that causes the most nonlinearity.

Envelope Following Analysis generates two types of output files: a voltage versus time (*td*) file, and an amplitude/phase versus time (*fd*) file for each of specified harmonic of the clock fundamental.

The *td* file contains time-domain real waveforms. They are similar to the waveforms generated by transient analysis. The difference between the analyses is that the integration of a clock cycle occurs much less frequently with Envelope Following Analysis. Consequently, you usually see big gaps between integrated clock cycles.

The *fd* file contains time varying Fourier coefficients (complex) of circuit response at clock harmonics. The most interesting harmonic is probably the first one. The real and imaginary parts of this harmonic are called the in-phase and quadrature components of the baseband signal. Assume, for example, that you simulate a transmitter that is driven by a complex baseband signal. For a phase-modulation scheme, you plot the phase of the time-varying fundamental versus time to see the transmitted phase. To see the trajectory of the transmitted baseband signal, plot the real versus the imaginary parts of the time-varying fundamental. The trajectory lets you visually check the error vector magnitude (EVM). To compute the transmitted power spectral density about the carrier, compute the power spectral density of the time-varying fundamental. From the power spectral density, you can estimate Adjacent Channel Power Ratio (ACPR).

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Analyses

The spectrum of each harmonic response is calculated from the Affirma analog circuit design environment. This is useful for applications such as ACPR calculation.

You can also use the fd result for applications such as constellation diagrams, by plotting amplitude versus phase.

SpectreRF Simulation Forms

You can use the SpectreRF simulation forms documentation in two ways. The first way is online viewing through the Cadence® online documentation library. You can look at a picture of a form in the documentation and then look up the section, later in this chapter, that describes the area of the form named in the callout.

Accuracy Defaults (errpreset) <input type="checkbox"/> conservative <input type="checkbox"/> moderate <input type="checkbox"/> liberal Additional Time for Stabilization (tstab) <input type="text"/> Save Initial Transient Results (saveinit) <input type="checkbox"/> no <input type="checkbox"/> yes
Oscillator <input type="checkbox"/>
Sweep <input type="checkbox"/>

Accuracy Defaults (errpreset)

Oscillator

Sweep

Look up these sections for detailed descriptions of these areas on the form.

If the documentation takes up an inconvenient amount of screen space, you can print out the text descriptions. The text description topics are presented alphabetically, according to the top-level headings on the forms, at the end of this chapter. The top-level headings are usually found at the leftmost margin of the form.

Choosing Analyses Form

When you select an Analysis at the top of the Choosing Analyses form, the content of the form changes to reflect choices for the selected analysis. The following sections describe this form.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

For Periodic Steady-State (PSS) Analysis (Top)

Choosing Analyses -- Affirma Analog Circuit Design Enviro

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId

Moderate

Clear/Add Delete Update From Schematic

Beat Frequency Auto Calculate

Beat Period

Output harmonics

Number of harmonics

Accuracy Defaults (errpreset)

conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep

Analysis

Fundamental Tones List Box

Beat Frequency

Auto Calculate

Beat Period

Output harmonics (includes information on menu choices)

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

For Periodic Steady-State (PSS) Analysis (Bottom)

Accuracy Defaults (errpreset)
 conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep

Enabled

Accuracy Defaults

tstab

Oscillator

Sweep

Options

Enabled

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Modification for Periodic Steady State (PSS) Analysis Sweep (Bottom)

Oscillator

Sweep Frequency Variable? no yes

Variable Variable Name

Select Design Variable

Sweep Range

Start-Stop Start Stop

Center-Span

Sweep Type

Linear Logarithmic

Step Size

Number of Steps

Add Specific Points

Enabled Options...

Oscillator

Sweep (includes information on menu choices)

Frequency Variable

Variable Name

Select Design Variable

Sweep Range

Sweep Type

Add Specific Points

Options

Enabled

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

For Periodic Distortion (Pdisto) Analysis

The dialog box 'Choosing Analyses -- Affirma Analog Circuit Design Env' contains the following elements:

- Buttons:** OK, Cancel, Defaults, Apply, Help
- Analysis Options:**
 - Analysis: tran dc ac noise xf
 - sens sp pdisto pss pac
 - pnoise pxf envlp
- Periodic Distortion Analysis Section:**
 - Fundamental Tones Table:**

#	Name	Expr	Value	Signal	SrcId	Harms
2	flo	flo	1G	Moderate	lo	0
1	frf	frf	900M	Moderate	rf	0
 - Input fields for Name, Expr, Value, Signal, SrcId, and Harms.
 - Buttons: Clear/Add, Delete, Update From Schematic
- Accuracy Defaults (errpreset):**
 - conservative moderate liberal
 - Additional Time for Stabilization (tstab): [input field]
 - Save Initial Transient Results (saveinit): no yes
- Sweep:**
- Enabled:** Options...

Analysis

Fundamental Tones List Box

Accuracy Defaults

tstab

saveinit

Sweep

Options

Enabled

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

For Periodic AC (PAC) Analysis with PSS

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic AC Analysis

PSS Beat Frequency (Hz)

Sweeptype

Frequency Sweep Range (Hz)

Start-Stop Start Stop

Sweep Type

Automatic

Add Specific Points

Sidebands

Maximum sideband

Enabled Options...

Analysis

Sweeptype

Frequency Sweep Range (Hz)

Sidebands (includes information on menu choices)

Options

Enabled

Modification for Periodic AC Analysis (PAC) with PSS Sweep

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic AC Analysis

PSS Beat Frequency (Hz)

Sweeptype

Frequency Sweep Range (Hz)

Single-Point [] Freq

Because the sweep section of the PSS analysis is active,
only a single point for this analysis is currently supported.

Sidebands

Maximum sideband

Enabled Options...

Frequency (Hz)

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

For Periodic Noise Analysis (Pnoise) with PSS

Sweeptype

Frequency Sweep Range (Hz)

Sweep Type

Sidebands (includes information on menu choices)

Output

Input Source (includes information on menu choices)

Reference Side-Band

Noise Type

Options

Enabled

Modification for Periodic Noise Analysis (Pnoise) with PSS Sweep

The screenshot shows a dialog box titled "Choosing Analyses -- Affirma Analog Circuit Design Envir". The dialog has buttons for "OK", "Cancel", "Defaults", "Apply", and "Help". The main section is titled "Periodic Noise Analysis".

Fields and options in the "Periodic Noise Analysis" section include:

- PSS Beat Frequency (Hz): []
- Sweep type: []
- Frequency Sweep Range (Hz):
 - Single-Point [] Freq []
- Message: "Because the sweep section of the PSS analysis is active, only a single point for this analysis is currently supported."
- Sidebands:
 - Maximum sideband []
- Output:
 - probe [] Output Probe Instance [] Select
- Input Source:
 - port [] Input Port Source [] Select
- Reference Side-Band []
- Message: "Because the sweep section of the PSS analysis is active, the Time Domain Analysis option is not currently supported."
- Enabled []
- Options... []

Freq

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

For Periodic Transfer Function (PXF) Analysis with PSS

The screenshot shows a dialog box titled "Choosing Analyses -- Affirma Analog Circuit Design Env". It contains several sections:

- Analysis:** A grid of radio buttons for selecting analysis types: tran, dc, ac, noise, xf, sens, sp, pdisto, pss, pac, pnoise, pxf (selected), and envlp.
- Periodic XF Analysis:**
 - PSS Beat Frequency (Hz): []
 - Sweeptype: []
 - Frequency Sweep Range (Hz): Start-Stop [] Start [] Stop []
 - Sweep Type: Automatic []
 - Add Specific Points: []
- Sidebands:** Maximum sideband [] []
- Output:**
 - voltage (selected) Positive Output Node [] [Select]
 - probe Negative Output Node [] [Select]
- Enabled:** []
- Options...** []

Analysis

Sweeptype

Frequency Sweep Range (Hz)

Sweep Type

Add Specific Points

Sidebands (includes information on menu choices)

Output

Options

Enabled

Modification for Periodic Transfer Function (PXF) Analysis with PSS Sweep

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic XF Analysis

PSS Beat Frequency (Hz)

Sweeptype

Frequency Sweep Range (Hz)

Single-Point Freq

Because the sweep section of the PSS analysis is active, only a single point for this analysis is currently supported.

Sidebands

Maximum sideband

Output

voltage Positive Output Node Select
 probe Negative Output Node Select

Enabled Options...

Analysis

Sweeptype

Freq

Sidebands

Output

Positive Output Node

Negative Output Node

For Periodic S-Parameter (PSP) Analysis with PSS

☺
Choosing Analyses — Affirma Analog Circuit Design

OK
Cancel
Defaults
Apply
Help

Analysis

<input type="checkbox"/> tran	<input type="checkbox"/> dc	<input type="checkbox"/> ac	<input type="checkbox"/> noise
<input type="checkbox"/> xrf	<input type="checkbox"/> sers	<input type="checkbox"/> sp	<input type="checkbox"/> envtp
<input type="checkbox"/> pss	<input type="checkbox"/> pac	<input type="checkbox"/> pnoise	<input type="checkbox"/> port
<input checked="" type="checkbox"/> psp	<input type="checkbox"/> pdlsto	<input type="checkbox"/> qnoise	

Periodic S-Parameter Analysis

SweepType

Frequency Sweep Range(Hz)

Start Stop

Sweep Type

Add Specific Points

Select Ports

Input Port(#1) <input type="text" value="/r1"/>	<input type="button" value="Select"/>	Harmonic <input type="text" value="1"/>
Output Port(#2) <input type="text" value="/r2"/>	<input type="button" value="Select"/>	Harmonic <input type="text" value="1"/>
Aux. Port(#3) <input type="text" value=""/>	<input type="button" value="Select"/>	Harmonic <input type="text" value=""/>

Additional Harmonics

Do Noise

yes

no

Enabled

Analysis

SweepType

Frequency Sweep Range

Sweep Type

Select Ports

Do Noise

Modification for Periodic S-Parameter (PSP) Analysis with PSS Sweep

Choosing Analyses — Affirma Analog Circuit Design

OK Cancel Defaults Apply Help

Analysis tran ac ac noise
 nt sars sp mvtp
 pss psc pnoise pzd
 psp pdista qnoise

Periodic S-Parameter Analysis

Sweep type

Frequency Sweep Range(Hz)

Freq

Because the sweep section of the PSS analysis is active, only a single point for this analysis is currently supported.

Select Parts

Input Port(#1) Select Harmonic

Output Port(#2) Select Harmonic

Aux. Port(#3) Select Harmonic

Additional Harmonics

Do Noise

yes

no Maximum Strideband

Enabled Options...

Frequency

For Quasi-Periodic Noise (QPnoise) Analysis with Pdisto

Choosing Analyses — Affirma Analog Circuit Design E

OK Cancel Defaults Apply Help

Analysis

tran dc ac noise
 xf soss sp envlp
 pss pac pnoise pxf
 psp pdisto qpnoise

Quasi Periodic Noise Analysis

Frequency Sweep Range (Hz)

Start-Stop Start Stop

Sweep Type

Automatic

Add Specific Points

Sidebands

Maximum clock order

Output

probe Output Probe Instance Select

Input Source

part Input Port Source Select

Reference Side-Band refsidebandoption
 freq individual

Enabled Options...

Analysis

Frequency Sweep Range

Sweep Type

Sidebands

Output

Input Source

Reference Side Band

For Envelope Following Analysis

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Envelope Following Analysis

Clock Name

Stop Time

Output Harmonics
 Number of harmonics

Accuracy Defaults (errpreset)
 conservative moderate liberal

Enabled

Clock Name

Select Clock Name

Stop Time

Output harmonics (includes information on menu choices)

Accuracy Defaults

Options

Enabled

Options Forms

PSS Analysis Options Form (Top)

Periodic Steady State Options	
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>
<input type="button" value="Defaults"/>	<input type="button" value="Apply"/>
<input type="button" value="Help"/>	
TIME STEP PARAMETERS	
step	<input style="width: 100%;" type="text"/>
maxstep	<input style="width: 100%;" type="text"/>
INITIAL CONDITION PARAMETERS	
ic	<input type="checkbox"/> dc <input type="checkbox"/> node <input type="checkbox"/> dev <input type="checkbox"/> all
skipdc	<input type="checkbox"/> yes <input type="checkbox"/> no <input type="checkbox"/> waveless <input type="checkbox"/> rampup <input type="checkbox"/> autodc
readic	<input style="width: 100%;" type="text"/>
CONVERGENCE PARAMETERS	
readns	<input style="width: 100%;" type="text"/>
cmin	<input style="width: 100%;" type="text"/>
STATE FILE PARAMETERS	
write	<input style="width: 100%;" type="text"/>
writefinal	<input style="width: 100%;" type="text"/>
swapfile	<input style="width: 100%;" type="text"/>
writepss	<input style="width: 100%;" type="text"/>
readpss	<input style="width: 100%;" type="text"/>
INTEGRATION METHOD PARAMETERS	
method	<input type="checkbox"/> euler <input type="checkbox"/> trap <input type="checkbox"/> traponly
	<input type="checkbox"/> gear2 <input type="checkbox"/> gear2only

TIME STEP PARAMETERS

INITIAL CONDITION
PARAMETERS

CONVERGENCE PARAMETERS

STATE FILE PARAMETERS

INTEGRATION METHOD
PARAMETERS

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

PSS Analysis Options Form (Bottom)

OK	Cancel	Defaults	Apply	Help
ACCURACY PARAMETERS				
relref	<input type="checkbox"/> pointlocal	<input type="checkbox"/> alllocal	<input type="checkbox"/> sigglobal	<input type="checkbox"/> allglobal
iteratio	<input type="text"/>			
steadyratio	<input type="text"/>			
maxacfreq	<input type="text"/>			
maxperiods	<input type="text"/>			
finitediff	<input type="checkbox"/> yes <input type="checkbox"/> samegrid <input type="checkbox"/> no			
ANNOTATION PARAMETERS				
stats	<input type="checkbox"/> yes <input type="checkbox"/> no			
annotate	<input type="checkbox"/> no <input type="checkbox"/> title <input type="checkbox"/> sweep <input checked="" type="checkbox"/> status <input type="checkbox"/> steps			
OUTPUT PARAMETERS				
save	<input type="checkbox"/> selected <input type="checkbox"/> lvipub <input type="checkbox"/> lvi <input type="checkbox"/> allpub <input type="checkbox"/> all			
reslvi	<input type="text"/>			
oppoint	<input type="checkbox"/> rawfile <input type="checkbox"/> screen <input type="checkbox"/> logfile <input type="checkbox"/> no			
skipstart	<input type="text"/>			
skipstop	<input type="text"/>			
skipcount	<input type="text"/>			
strobeperiod	<input type="text"/>			
strobedelay	<input type="text"/>			
compression	<input type="checkbox"/> no <input type="checkbox"/> yes			
outputtype	<input type="checkbox"/> freq <input type="checkbox"/> time <input type="checkbox"/> all			
NEWTON PARAMETERS				
maxiters	<input type="text"/>			
restart	<input type="checkbox"/> yes <input type="checkbox"/> no			
SIMULATION INTERVAL PARAMETERS				
tstart	<input type="text"/>			

ACCURACY PARAMETERS

ANNOTATION PARAMETERS

OUTPUT PARAMETERS

NEWTON PARAMETERS

SIMULATION INTERVAL
PARAMETERS

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Pdisto Analysis Options Form (Top)

Periodic Distortion Options	
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>
<input type="button" value="Defaults"/>	<input type="button" value="Apply"/>
<input type="button" value="Help"/>	
TIME STEP PARAMETERS	
step	<input type="text"/>
maxstep	<input type="text"/>
INITIAL CONDITION PARAMETERS	
ic	<input type="checkbox"/> dc <input type="checkbox"/> node <input type="checkbox"/> dev <input type="checkbox"/> all
skipdc	<input type="checkbox"/> yes <input type="checkbox"/> no <input type="checkbox"/> waveless <input type="checkbox"/> rampup <input type="checkbox"/> autodc
readic	<input type="text"/>
CONVERGENCE PARAMETERS	
readns	<input type="text"/>
cmin	<input type="text"/>
STATE FILE PARAMETERS	
write	<input type="text"/>
writefinal	<input type="text"/>
swapfile	<input type="text"/>
writeqps	<input type="text"/>
readqps	<input type="text"/>
INTEGRATION METHOD PARAMETERS	
method	<input type="checkbox"/> euler <input type="checkbox"/> trap <input type="checkbox"/> traonly <input type="checkbox"/> gear2 <input type="checkbox"/> gear2only
ACCURACY PARAMETERS	
relref	<input type="checkbox"/> pointlocal <input type="checkbox"/> alllocal <input type="checkbox"/> sigglobal <input type="checkbox"/> allglobal
iteratio	<input type="text"/>
steadyratio	<input type="text"/>
maxperiods	<input type="text"/>

TIME STEP PARAMETERS

INITIAL CONDITION
PARAMETERS

CONVERGENCE
PARAMETERS

STATE FILE PARAMETERS

INTEGRATION METHOD
PARAMETERS

ACCURACY PARAMETERS

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Pdisto Analysis Options Form (Bottom)

ANNOTATION PARAMETERS	
stats	<input type="checkbox"/> yes <input type="checkbox"/> no
annotate	<input type="checkbox"/> no <input type="checkbox"/> title <input type="checkbox"/> sweep <input checked="" type="checkbox"/> status <input type="checkbox"/> steps
<hr/>	
OUTPUT PARAMETERS	
save	<input type="checkbox"/> selected <input type="checkbox"/> lv/pub <input type="checkbox"/> lvi <input type="checkbox"/> all/pub <input type="checkbox"/> all
results	<input type="text"/>
oppoint	<input type="checkbox"/> rawfile <input type="checkbox"/> screen <input type="checkbox"/> logfile <input type="checkbox"/> no
skipstart	<input type="text"/>
skipstop	<input type="text"/>
skipcount	<input type="text"/>
strobeperiod	<input type="text"/>
strobedelay	<input type="text"/>
compression	<input type="checkbox"/> no <input type="checkbox"/> yes
<hr/>	
NEWTON PARAMETERS	
maxiters	<input type="text"/>
restart	<input type="checkbox"/> yes <input type="checkbox"/> no
<hr/>	
SIMULATION INTERVAL PARAMETERS	
tstart	<input type="text"/>

ANNOTATION
PARAMETERS

OUTPUT PARAMETERS

NEWTON PARAMETERS

SIMULATION INTERVAL
PARAMETERS

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

PAC Analysis Options Form

Periodic AC Options

OK Cancel Defaults Apply Help

CONVERGENCE PARAMETERS

tolerance

gear_order 1 2 3 4 5 6

solver std exp turbo

oscsolver std turbo

ANNOTATION PARAMETERS

stats yes no

annotate no title sweep status steps

OUTPUT PARAMETERS

freqaxis absout in out

save selected lvlpub lvl allpub all

nestlvl

outputperiod

CONVERGENCE
PARAMETERS

ANNOTATION
PARAMETERS

OUTPUT PARAMETERS

PXF Analysis Options Form

Periodic XF Options

OK Cancel Defaults Apply Help

CONVERGENCE PARAMETERS

tolerance

gear_order 1 2 3 4 5 6

solver std exp turbo

oscsolver std turbo

ANNOTATION PARAMETERS

stats yes no

annotate no title sweep status steps

OUTPUT PARAMETERS

stimuli sources nodes_and_terminals

freqaxis absin in out

save selected |v|pub |v| allpub all

msbvl

CONVERGENCE PARAMETERS

ANNOTATION PARAMETERS

OUTPUT PARAMETERS

Pnoise Analysis Options Form

Periodic Noise Options

OK Cancel Defaults Apply Help

CONVERGENCE PARAMETERS

tolerance

gear_order 1 2 3 4 5 6

solver std exp turbo

oscsolver std turbo

ANNOTATION PARAMETERS

stats yes no

annotate no title sweep status steps

OUTPUT PARAMETERS

save selected lvlpub lvl allpub all

reslvl

saveallsidebands yes no

CONVERGENCE
PARAMETERS

ANNOTATION
PARAMETERS

OUTPUT
PARAMETERS

PSP Analysis Options Form

Periodic S-Parameter Options

OK Cancel Defaults Apply Help

CONVERGENCE PARAMETERS

tolerance

gear_order 1 2 3 4 5 6

solver std exp turbo

oscsolver std turbo

ANNOTATION PARAMETERS

annotate no title sweep status steps

stats yes no

OUTPUT PARAMETERS

freqs absout in out

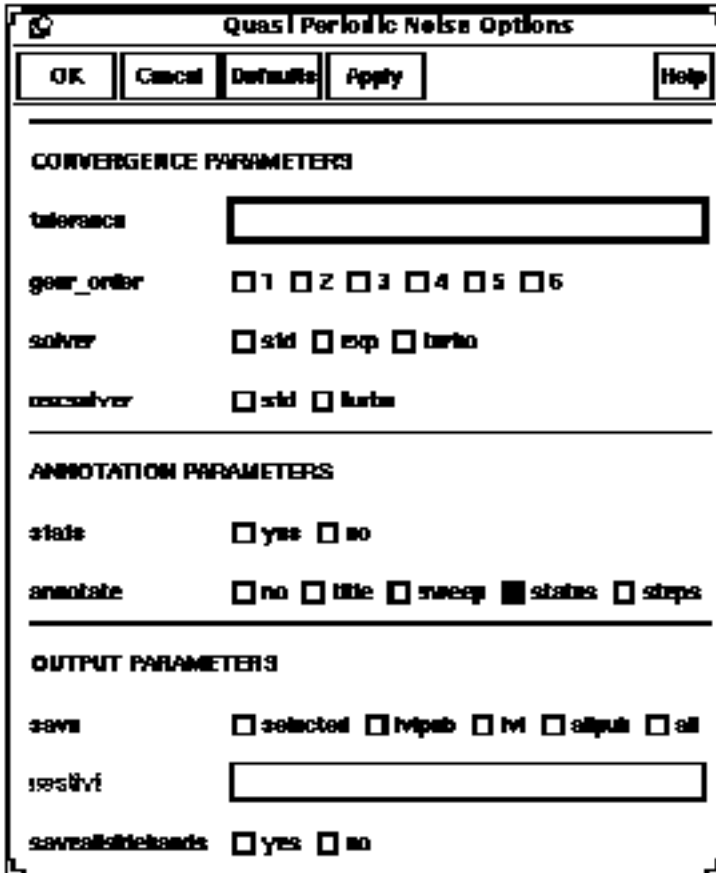
Accuracy
Parameters

Annotation
Parameters

Output
Parameters

Newton
Parameters

QPnoise Analysis Options Form



Quasi Periodic Noise Options

OK Cancel Defaults Apply Help

CONVERGENCE PARAMETERS

tolerance

gear_order 1 2 3 4 5 6

solver std exp turbo

nosolver std turbo

ANNOTATION PARAMETERS

state yes no

annotate no title sweep states steps

OUTPUT PARAMETERS

save selected ivpab tv allput all

reslvt

saveallbitbands yes no

Convergence
Parameters

Annotation
Parameters

Output
Parameters

Envelope Following Analysis Options Form (Top)

Envelope Following Options

OK Cancel Defaults Apply Help

SIMULATION INTERVAL PARAMETERS

start []

substart []

total []

SIMULATION BANDWIDTH PARAMETERS

modulationbw []

TIME STEP PARAMETERS

maxstep []

envelopestep []

INITIAL CONDITION PARAMETERS

ic etc none div all

skipic yes no waveless rampup autetic

readic []

CONVERGENCE PARAMETERS

readms []

cmis []

Simulation Interval
Parameters

Simulation Bandwidth
Parameters

Time Step
Parameters

Initial Condition
Parameters

Convergence
Parameters

Envelope Following Analysis Options Form (bottom)

STATE FILE PARAMETERS	
write	<input type="text"/>
writefinal	<input type="text"/>
wrapfile	<input type="text"/>
INTEGRATION METHOD PARAMETERS	
method	<input type="checkbox"/> euler <input type="checkbox"/> trap <input type="checkbox"/> trapony <input type="checkbox"/> gear2 <input type="checkbox"/> gear2only
ACCURACY PARAMETERS	
relref	<input type="checkbox"/> pointlocal <input type="checkbox"/> alllocal <input type="checkbox"/> sigglobal <input type="checkbox"/> allglobal
iteratio	<input type="text"/>
erriteratio	<input type="text"/>
ANNOTATION PARAMETERS	
stats	<input type="checkbox"/> yes <input type="checkbox"/> no
annotate	<input type="checkbox"/> no <input type="checkbox"/> title <input type="checkbox"/> samp <input checked="" type="checkbox"/> status <input type="checkbox"/> steps
OUTPUT PARAMETERS	
save	<input type="checkbox"/> selected <input type="checkbox"/> inputs <input type="checkbox"/> ivt <input type="checkbox"/> outputs <input type="checkbox"/> all
isobits	<input type="text"/>
compression	<input type="checkbox"/> no <input type="checkbox"/> yes
outputtype	<input type="checkbox"/> both <input type="checkbox"/> envelope <input type="checkbox"/> spectrum
strobeperiod	<input type="text"/>
NEWTON PARAMETERS	
modifiers	<input type="text"/>
restart	<input type="checkbox"/> yes <input type="checkbox"/> no
newmodifiers	<input type="text"/>

State File
Parameters

Integration Method
Parameters

Accuracy
Parameters

Annotation
Parameters

Output
Parameters

Newton
Parameters

Results Forms

You can use Affirma analog circuit design environment direct plot commands to plot most RF simulation results. There are two Simulation window commands specifically designed for plotting periodic steady state simulation results:

- *Results – Direct Plot – PSS*
- *Results – Direct Plot – Pdisto.*

Each command opens up a Results form. You define the plots by filling out the PSS or Pdisto Results forms and selecting nets or terminals on the schematic.

The plots appear, by default, in the current Affirma analog circuit design environment Waveform window. If no Waveform window or Schematic window is open, selecting a direct plot function automatically opens them both.

For sweeps only, the last section in the form includes a scrolling window that lists different values of the design variable you selected when setting up the analysis. Click on the design variable values that you want to plot. The name and type of the variable is displayed at the top of the window. For second and third order intercept functions, the Results form displays a second list of variable values for the reference variable.

You click on sidebands in a list box to select them. You can select adjacent sidebands by clicking and dragging with the mouse over the sidebands you want to select. To select sidebands that are not adjacent, hold the Control key down while you click on the individual sidebands. Clicking on a selected sideband while you hold the Control key down deselects the sideband.

Spectral Plots and Time Waveforms

Spectral plots show the function level at each frequency component of a single analysis point (one step in a sweep of frequency, input power, or design variable).

To perform a spectral plot:

1. Open the *Waveform window*, the design schematic, and the PSS Results form by choosing *Results – Direct Plot – PSS*.
2. Click on a *Plot Mode* button to specify if the curves you plot are appended to or replace existing curves in the *Waveform window*.
3. Click on a *Function* button to select the function you want to plot.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

If you need a function that is not included on the PSS Results form, use the RF version of the Affirma analog circuit design environment [calculator](#).

4. For PSS sweeps, set *Sweep* to *spectrum*.
5. Select a plot modifier.
6. To plot a curve in the [Waveform window](#), follow the instructions at the bottom of the Results form.
7. To add the last plot to the *Outputs* list in the [Simulation window](#), click on the *Add to Outputs* button and then click on *Replot*.

When you select a curve to plot from the *Outputs* list, the expression plotted is always the magnitude of the signal, not the original modifier. If you want a scale like dB20, you must modify the expression using *Curve – Edit – Scale* in the [Waveform window](#) or paste the expression from the *Outputs Setup* form into the [calculator](#) buffer and edit it.

8. To change the noise floor of a spectral plot, use the *Axes – Y axis* command in the [Waveform window](#), set *Range* to *Min-Max*, and specify a *Min* value.

To use the PSS Results form to plot voltage and current time waveforms, repeat the process for spectral plots. Set *Sweep* to *time*.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

PSS Results Form

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves
 Power Contours Beat Frequency
 Reflection Contours

Sweep spectrum time

Modifier

Magnitude Phase dB20
 Real Imaginary

Add To Outputs

> Select net on schematic...

Plot Mode

Analysis Type

Function

Sweep

Modifier

Add to Outputs


Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Additional PSS Results Form Fields That Appear for Some Functions

Input Harmonic


0	0
1	40M
2	80M
3	120M
4	160M



Input Harmonic

Output Harmonic

0	0
1	40M
2	80M
3	120M
4	160M



Output Harmonic

Variable Value (prf)

10m
30m
50m

Variable Value

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Input Power Value (dBm)

Input Referred IP3 **Order**

3rd Order Harmonic

0	0
1	40M
2	80M
3	120M
4	160M

1st Order Harmonic

0	0
1	40M
2	80M
3	120M
4	160M

Add To Outputs

Extrapolation Point

Input Power Value (dBm)

Input(Output) Referred IP3

Order

Nth Order Harmonic

1st Order Harmonic

Add to Outputs

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Gain Compression (dB) <input style="width: 50px;" type="text" value="1"/>	Gain Compression										
"prf" ranges from 10m to 50m											
Extrapolation Point (dBm) <input style="width: 100px;" type="text"/> (Defaults to 10m)	Extrapolation Point										
Input Referred 1dB Compression <input type="checkbox"/>	Input (Output) Referred 1dB Compression										
1st Order Harmonic	1st Order Harmonic										
<table border="1" style="border-collapse: collapse;"><tr><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">0</td></tr><tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;">40M</td></tr><tr><td style="padding: 2px 10px;">2</td><td style="padding: 2px 10px;">80M</td></tr><tr><td style="padding: 2px 10px;">3</td><td style="padding: 2px 10px;">120M</td></tr><tr><td style="padding: 2px 10px;">4</td><td style="padding: 2px 10px;">160M</td></tr></table> <div style="display: inline-block; vertical-align: middle; text-align: center;"><div style="border: 1px solid black; width: 15px; height: 15px; margin: 0 auto; line-height: 15px;">▲</div><div style="border: 1px solid black; width: 15px; height: 15px; margin: 0 auto; line-height: 15px;">▼</div></div>	0	0	1	40M	2	80M	3	120M	4	160M	
0	0										
1	40M										
2	80M										
3	120M										
4	160M										
Add To Outputs <input type="checkbox"/>	Add to Outputs										

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Pdisto Results Form

PDISTO Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pdisto

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves

Currently, only spectrum data is available

Modifier

Magnitude Phase dB20
 Real Imaginary

Add To Outputs

> Select net on schematic...

Plot Mode

Function

Modifier

Add To Outputs

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Additional Pdisto Results Form Fields That Appear for Some Functions

	Freq. (Hz)	flo	frf	
Input Harmonic	0	0	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	80M	1	-1	
	840M	-1	2	
	920M	0	1	
	1G	1	0	
Add To Outputs <input type="checkbox"/>				Add To Outputs

	Freq. (Hz)	flo	frf	
Output Harmonic	0	0	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
	80M	1	-1	
	840M	-1	2	
	920M	0	1	
	1G	1	0	

Variable Value (prf)	<div style="border: 1px solid black; padding: 5px; min-height: 100px;"> <p>10m</p> <p>30m</p> <p>50m</p> </div>	Variable Value
-----------------------------	---	----------------

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Circuit Input Power Single Point
 Variable Sweep ("prf")

Input Power Value (dBm)

Plot Curves Points

Input Referred IP3 Order

	Freq. (Hz)	flo	frf	
3rd Order Harmonic	0	0	0	▲
	80M	1	-1	■
	840M	-1	2	■
	920M	0	1	■
	1G	1	0	▼
1st Order Harmonic	0	0	0	▲
	80M	1	-1	■
	840M	-1	2	■
	920M	0	1	■
	1G	1	0	▼

Circuit Input Power

Input Power Value (dBm)

Plot

Input(Output) Referred IP3

Order

Nth Order Harmonic

1st Order Harmonic

Gain Compression (dB)

"prf" ranges from 10m to 50m

Extrapolation Point (dBm)

(Defaults to 10m)

Input Referred 1dB Compression

	Freq. (Hz)	flo	frf	
1st Order Harmonic	0	0	0	▲
	80M	1	-1	■
	840M	-1	2	■
	920M	0	1	■
	1G	1	0	▼

Gain Compression (dB)

Extrapolation Point

Input (Output) Referred 1dB
Compression

1st Order Harmonic

PAC Results Form (with Swept PSS)

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss pxf pnoise pac

Function

Voltage Current
 IPN Curves Pss Beat Frequency

Currently, only spectrum data is available

Modifier

Magnitude Phase dB20
 Real Imaginary

Add To Outputs

> Select net on schematic...

- ← Plot Mode
- ← Analysis Type
- ← Function
- ← Modifier
- ← Add To Outputs

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Additional Periodic AC Results Form Fields That Appear for Some Functions

Circuit Input Power Single Point
 Variable Sweep ("prf")

Input Power Value (dBm)

Plot Curves Points

Input Referred IP3 **Order**

3rd Order sideband		1st Order sideband	
-10	400M	-10	400M
-9	360M	-9	360M
-8	320M	-8	320M
-7	280M	-7	280M
-6	240M	-6	240M

Circuit Input Power

Input Power Value (dBm)

Plot

Order

Nth Order Harmonic

1st Order Harmonic

Output Sideband

-10	400M
-9	360M
-8	320M
-7	280M
-6	240M

Output Sideband

Variable Value (prf)

10m
30m
50m

Variable Value

Pnoise Results Form (with PSS)

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss pxf pnoise pac

Function

Output Noise Input Noise
 Noise Figure Noise Factor
 Transfer Function Phase Noise
 Pss Beat Frequency

Currently, only variable data is available

Modifier

Magnitude dB20

Add To Outputs Plot

- ← Plot Mode
- ← Analysis Type
- ← Function
- ← Modifier
- ← Add To Outputs
- ← Plot

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

PXF Results Form (with PSS)

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss pxf pnoise pac

Function

Voltage Gain Transimpedance

Pss Beat Frequency

Currently, only spectrum data is available

Modifier

Magnitude Phase dB20

Real Imaginary

Add To Outputs

> Select port or voltage source on schematic ...

Plot Mode

Analysis Type

Function

Modifier

Add to Outputs

Additional PXF Results Form Fields That Appear for Some Functions

Output Sideband	
-10	400M
-9	360M
-8	320M
-7	280M
-6	240M

Output Sideband

PSP Results Form (with PSS)

PSS Results
OK Cancel Help

Plot Mode Append Replace

Analysis

pss psp

Function

SP HF F HFreal
 Fieee HFieee Ftsb HFtsb
 RM IRN NNR GAIN

Description: S-Parameter

Plot Type

Auto Rectangular Z-Smith
 Y-Smith Polar

Modifier

Magnitude Phase dB20
 Real Imaginary

S11	S12
S21	S22

Part 1 active harmonic is -9
Part 2 active harmonic is 1

Add To Outputs

> To plot, press **Sig**-button on this form...

Plot Mode

Analysis

Function

Plot Type

Modifier

Plot Buttons

Add to Outputs

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Envelope Following Results Form

Envelope Following Results
OK Cancel Help

Plot Mode Append Replace

Function

Voltage Current Power
 Description: Harmonic Voltage Spectrum

Sweep spectrum harmonic time
 time

Modifier

Magnitude dB10 dBm

Harmonic Number

0	▲
1	□
2	□
3	▼

Power Spectral Density Parameters

Time Interval

From To

Nyquist half-bandwidth

Frequency bin width

Max. plotting frequency

Min. plotting frequency

Windowing Hamming

Detrending Linear

Add To Outputs

> Select "Reference Harmonic" on this form...

Plot Mode

Function

Sweep

Modifier

Harmonic Number

Power Spectral Density
Parameters

Add to Outputs

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Additional Envelope Following Results Form Fields that Appear for Some Functions

Get Reference Reflection Coefficients by Selecting a Separate Terminal

Maximum Power

Minimum Power

Reference Resistance

Number of Contours

Close Contours

Get Reference...
Maximum Power
Minimum Power
Reference Resistance
Close Contours

Get Reference Reflection Coefficients by Selecting a Separate Terminal

Max Reflection Mag

Min Reflection Mag

Reference Resistance

Number of Contours

Close Contours

Max Reflection Mag
Min Reflection Mag

QPnoise Results Form (with Pdisto)

PDISTO Results

OK **Cancel** **Help**

Plot Mode **Append** **Replace**

Analysis

qpnoise **qpss**

Function

Output Noise **Input Noise**
 Noise Figure **Noise Factor**
 Transfer Function

Currently, only frequency data is available

Modifier

Magnitude **dB20**

Add To Outputs **Plot**

Form Field Descriptions

The remaining sections of this chapter presents text descriptions for the SpectreRF forms.

- [Choosing Analysis Form](#)
- The [Options Forms](#) for each analysis
- The [Results Forms](#) for each analysis

The text description topics are presented alphabetically, according to the top-level headings on the forms. The top-level headings are usually found at the leftmost margin of the form.

Choosing Analysis Form

Accuracy Defaults (*errpreset*)

Lets you adjust the simulator parameters to fit your needs quickly. You can set *errpreset* to *conservative* if the circuit is very sensitive, or you can set it to *liberal* for a fast. but possibly inaccurate, simulation. The setting *errpreset=moderate* suits most needs.

The effect of *errpreset* on other parameters is shown in the following table. In this table, $T = stop - start$.

errpreset	reltol	relref	method	maxstep	Iteratio
liberal	×10.0	allglobal	gear2	≤ $T/10$	3.5
moderate	×1.0	sigglobal	traonly	≤ $T/50$	3.5
conservative	×0.1	alllocal	gear2only	≤ $T/100$	10.0

The value of *reltol* is increased or decreased from its value in the options statement, but it is not allowed to be larger than 0.01. Spectre simulation sets the value of *maxstep* so that it is no larger than the value given in the table. Except for *reltol* and *maxstep*, *errpreset* does not change the value of any parameters you have explicitly set. The actual values used for the transient analysis are given in the log file.

Additional Time for Stabilization (*tstab*)

The amount of additional time that you specify to allow the circuit to settle. Use *tstab* if the circuit exhibits more than one periodic solution and you want only one. A long *tstab* can also improve convergence.

Add Specific Points

See [Sweep Type](#).

Analysis

Selects the SpectreRF analysis type.

The RF analyses are

- The periodic large-signal *pss* analysis
- The periodic medium-signal *pdisto* (*qpss*) analysis
- The periodic small-signal analyses: *pac*, *psp*, *pnoise*, and *pxf*
- The quasi-periodic small-signal *qpnoise* analysis

You can run the small-signal analyses after a large-signal analysis.

Auto Calculate Button

When selected, the Choosing Analysis form will automatically calculate the Beat Frequency and Beat Period field value based on the tones present in the Fundamental Tones list box. The value of the frequency will be the greatest common multiple of all the tone frequencies that are not small signal. The period will be set to the inverse of the frequency.

Beat Frequency

When you open the Choosing Analysis form, this type-in field is empty. Entering a value into this field can be accomplished in two ways. You can type in a value in which all the tone frequencies must be an integer multiple of the value, or you can click the *Auto Calculate* button so the *Choosing Analysis* form automatically calculates and enters a value into the field.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Beat Period

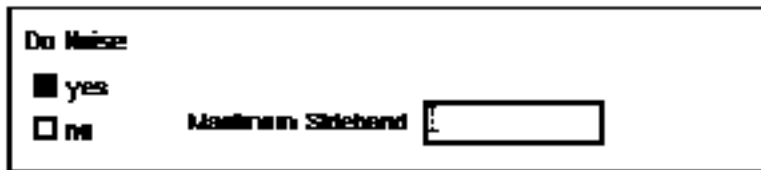
When you open the Choosing Analysis form and set the associated button, this type-in field is empty. Entering a value into this field can be accomplished in two ways. You can type in a value in which all the tone frequencies must be an integer multiple of the inverse of the value, or you can click the *Auto Calculate* button so the Choosing Analysis form automatically calculates and enters a value into the field.

Clock Name

A field that lets you type the name of the clock for an Envelope Following Analysis.

Do Noise (PSP only)

Lets you compute noise figures, equivalent noise sources, and noise parameters for a PSP analysis. Specify a maximum sideband value in the *MaximumSideband* field to include relevant noise folding effects. A sideband array of the form [-max. sideband . . . 0 . . . + max. sideband] is automatically generated.



The image shows a rectangular control panel with a black border. Inside, the text "Do Noise" is at the top left. Below it are two radio buttons: the first is checked and labeled "yes", and the second is unchecked and labeled "no". To the right of these is the text "Maximum Sideband" followed by a small rectangular input field.

Enabled

Click on the appropriate *Enabled* button to include an analysis in the next simulation.

Frequency (Hz)

This field specifies the specific frequency for a small-signal analysis that follows each analysis of a PSS sweep. This field is shown when the *Frequency Sweep Range (Hz)* field is set to *Single-Point*.

Frequency Sweep Range (Hz)

Defines the analysis range for a small-signal analysis that follows PSS.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

When you click on a *Frequency Sweep Range (Hz)* selection, the form fields change to let you specify appropriate data.

- **Start - Stop** defines the beginning and ending points for the sweep.

Frequency Sweep Range (Hz)

Start-Stop
 Start
 Stop

- **Center - Span** defines the center of the sweep and its span.

Frequency Sweep Range (Hz)

Center-Span
 Center
 Span

- **Single - Point** defines the frequency range as a single point and prompts you for the point value.

Frequency Sweep Range (Hz)

Single-Point
 Freq

Fundamental Tones

Fundamental Tones					
#	Name	Expr	Value	Signal	SrcId
1	flo	flo	1G	Moderate	lo
2	frf	frf	920M	Moderate	rf

Moderate

list box terms

data entry fields

data entry buttons

Fundamental Tones list box – This section appears in the Choosing Analysis form for the PSS and Pdisto analyses. The *Fundamental Tones* list box displays information about every top-level tone in the circuit that has both a non-zero frequency or period value, and a non-zero amplitude value (absolute). For each tone entry, you can edit either the *Signal* level designation or the *Harms* value (Pdisto only). Edits are performed by selecting a tone row and then editing the data entry fields.

For tones that are not at the top level of the schematic, you must manually insert a tone entry into the list box by entering the pertinent information into the data entry fields.

All tones in the list box are arranged alphabetically by name.

Each name and correlated frequency value of non-small-signal tones are used in the *Select from range* and *Array of coefficients* fields in the harmonics and sidebands sections of the PSS, PAC, PNOISE and PXF analysis forms.

■ List Box Terms and Corresponding Data Entry Fields

- ❑ **Name** – Displays the name assigned to each tone. This name must be entered into the pertinent Component Description Format (CDF) fields of each source in the schematic with a tone. The current CDF name field prompts are “First frequency name”, “Second frequency name”, “Frequency name”, and “Frequency name for 1/ period”.
- ❑ **Expr** – Displays the value or expression representing the frequency of a particular tone. The expression can also be a user variable or contain user variables. If the frequency for the tone is specified in a variable, this field displays the name of the variable. Otherwise, the field displays the numerical value of the frequency.
- ❑ **Value** – Displays the evaluated value of the *Expr* field using the current values of the user variables.
- ❑ **Signal** – This field displays one of three values: *Large*, *Moderate*, or *Small*. You must select a single *Large* tone for *Pdisto*, and this is the only time you use the *Large* specification. Specify all other tones for *Pdisto* simulation as *Moderate*. For *PSS*, specify *Moderate* for all tones you want to include in the simulation. Tones that you specify as *Small* are ignored by *PSS* and *Pdisto*.
- ❑ **SrcId** – Displays the instance name of the source in the schematic where the tone is declared.

To specify a new tone, make sure there are no selected tones in the list box. If there is a selected tone, click the *Clear/Add* button. Next, enter a value into any of the data entry fields (typically starting with the name field). As you advance to a second data entry field, a new tone line will automatically be added to the list box and become selected. Continue editing all the data entry fields until all the pertinent

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

information is complete. The last value in the data entry fields will be recorded in the list box when a preceding operation is performed on the analysis form. Such operations typically include moving the cursor off the analysis form, clicking either the *OK* or *Apply* button, clicking the *Clear/Add* button, or changing a value in a non-related field.

- ❑ **Harms** – Used only with *Pdisto*. This field displays the maximum number of harmonics for the tone to be used for the simulation. If the value for this field is not 1 or higher (the default is 0), the tone is ignored by *Pdisto*.

- **Data Entry Fields** – You use these fields to edit the specifications in the *Fundamental Tones* list box or to add new tones.

If you want to edit an existing tone, select the tone in the list box. The tone then appears in the data entry fields where you can edit it. Values you cannot edit in the data entry fields, such as values specified in the schematic, are grayed out. Values originally specified in the schematic must be edited in the schematic.

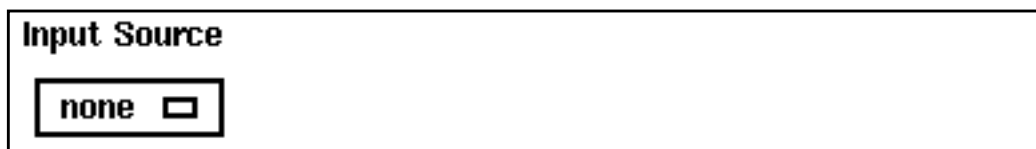
To specify a new tone, you type values directly into the list box and select values from the *Signal* menu below the list box.

- **Data Entry Buttons**

- ❑ **Clear/Add** – Clears the data entry fields for the purpose of manually adding a new tone to the list box. This button also resets the list box so that no line is currently selected.
- ❑ **Delete** – Deletes a tone selected in the *Fundamental Tones* list box. You cannot use this button to delete tones that are specified in the schematic. Such tones are deleted by setting the CDF frequency or period field value to zero or blank and CDF amplitude value(s) to zero (absolute) or blank.
- ❑ **Update From Schematic** – Updates the values in the *Fundamental Tones* list box from the schematic.

Note: Before you can update from the schematic, you must perform a *Check and Save* on the design.

Input Source (Pnoise and QPnoise)



The screenshot shows a rectangular window titled "Input Source". Inside the window, there is a smaller rectangular box containing the text "none" followed by a small square icon, likely representing a dropdown menu or a selection button.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **voltage** – Type in the name of a noise voltage generator, or click on *Select* and then click on the generator in the schematic.

Input Source			
<input type="text" value="voltage"/>	Input Voltage Source	<input type="text" value="/rff"/>	Select

- **current** – Type in the name of a noise current generator, or click on *Select* and then click on the generator in the schematic.

Input Source			
<input type="text" value="current"/>	Input Current Source	<input type="text"/>	Select

- **port** – The analysis computes the noise voltage across the port, but the contribution of this port will be subtracted in noise figure calculation.

Input Source			
<input type="text" value="port"/>	Input Port Source	<input type="text"/>	Select

Noise Type

- **Noise Type – sources** computes the total time-average noise at an output over a given frequency range. Each noise source's contribution to the total noise is computed at each frequency.

Noise Type	
<input type="text" value="sources"/>	

- **Noise Type – timedomain** computes the time-varying instantaneous noise power in a circuit with periodically driven components.

Noise Type	
<input type="text" value="timedomain"/>	Noise Skip Count <input type="text"/>

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **Noise Type – correlations** computes correlations between the noise at different ports of a multiport circuit. For example, you can compute the correlation between the noise at different outputs or the correlation between the noise at the input and output of a circuit that exhibits frequency conversion. You can extract equivalent noise sources from these calculations.

Noise Type	
<input checked="" type="checkbox"/> correlations	Maximum Cycles <input type="text"/>

Options

Opens the appropriate *Options* form for the associated analysis.

Oscillator

Specifies that the circuit is an oscillator. The form changes to let you enter the *Oscillator node* and the *Reference node*. You can either type the appropriate node names or click on *Select* and then click on the nodes in the schematic. If you leave the *Reference node* field empty, it defaults to `gnd`.

Oscillator <input checked="" type="checkbox"/>	Oscillator node <input type="text"/>	<input type="button" value="Select"/>
	Reference node <input type="text"/>	<input type="button" value="Select"/>

Output (PXF, Pnoise, and QPnoise)

Lets you specify the output for the PXF, Pnoise and QPnoise analyses.

Output for PXF Analysis

- **voltage** – The form changes to let you specify a *Positive Output Node* and a *Negative Output Node*. You can specify this value either by typing it into the type-

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

in field or by clicking on the adjacent *Select* button and then clicking on the node in the schematic. If you leave the *Negative Output Node* field empty, it defaults to `gnd`.

Output			
<input checked="" type="checkbox"/> voltage	Positive Output Node	<input type="text" value="/Pif"/>	<input type="button" value="Select"/>
<input type="checkbox"/> probe	Negative Output Node	<input type="text"/>	<input type="button" value="Select"/>

In PXF, the voltage across the two nodes that you specify is the output for each transfer function.

- **current** – The form changes to let you specify an output voltage source. You can specify this value either by typing it into the *Output Voltage Source* field or by clicking the adjacent *Select* button and then clicking on the node in the schematic.

Output			
<input type="checkbox"/> voltage	Output Probe Instance	<input type="text"/>	<input type="button" value="Select"/>
<input checked="" type="checkbox"/> probe			

In PXF, the current through the point that you select is the output of each transfer function.

Output for Pnoise Analysis

- **voltage** – The analysis computes the noise voltage across the two nodes. If you leave the *Negative Output Node* field empty, it defaults to `gnd`. You can specify this value either by typing it into the type-in field or by clicking on the adjacent *Select* button and then clicking on the node in the schematic.

Output			
<input type="checkbox"/> voltage	Positive Output Node	<input type="text" value="/Pif"/>	<input type="button" value="Select"/>
	Negative Output Node	<input type="text"/>	<input type="button" value="Select"/>

- **probe** – The analysis computes the noise voltage across the port, but the contribution of this port will be subtracted in noise figure calculation. You can specify this value either

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

by typing it into the type-in field or by clicking on the adjacent *Select* button and then clicking on the node in the schematic.

Output

probe Output Probe Instance

Output for QPnoise Analysis

- probe** – The analysis computes the noise voltage across the port, but the contribution of this port will be subtracted in noise figure calculation. You can specify this value either by typing it into the type-in field or by clicking on the adjacent *Select* button and then clicking on the node in the schematic.

Output

probe Output Probe Instance

- voltage** – The analysis computes the noise voltage across the two nodes. If you leave the *Negative Output Node* field empty, it defaults to `gnd`.

Output

voltage Positive Output Node

voltage Negative Output Node

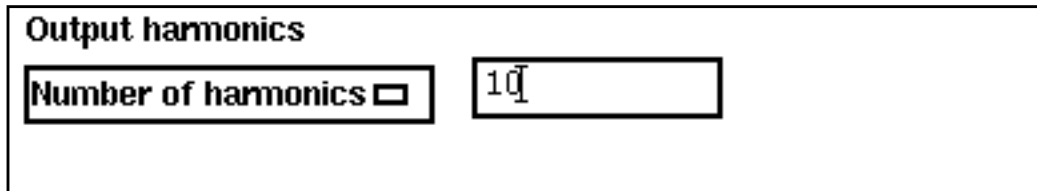
Output harmonics

Lets you select output harmonics. When you select from the cyclic field, the form changes to let you specify appropriate data. You can use the choices in the *Output harmonics* cyclic field in combination. For example, if you add a harmonic using *Array of Coefficients*, the value you added appears in the *Select from Range* list box and as a currently active index field for the *Array of Indices*.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **Number of harmonics** lets you specify the number of output harmonics. You type a single integer into the field to specify the number of harmonics. Harmonics in this field are referred to the value of the fundamental. To specify no harmonics, type 0 in the field.



Output harmonics

Number of harmonics

- **Select from range** lets you enter a frequency range and then select harmonics within this range. When you click on this choice, two fields and a cyclic field open. You type the range limits in the *From (Hz)* and *To (Hz)* fields and select the maximum order of harmonics that contribute to the output harmonics from the *Max. Order* cyclic field. Harmonics within these specifications then appear in the list box, and you highlight harmonics in the list box to select them. If, for example, you select 5 as the *Max. Order* value, the sum of the *absolute values* of the tone coefficients contributing to the output harmonics in the list box must be less than or equal to five. In the list box, negative integers represent the tone coefficients of harmonics below the fundamental and positive integers represent the tone coefficients of harmonics above the fundamental.

You click on harmonics in the list box to select them. You can select adjacent harmonics by clicking and dragging with the mouse over the harmonics you want to select. To select harmonics that are not adjacent, hold the `Control` key down while you click on the individual harmonics. Clicking on a selected harmonic while you hold the `Control` key down deselects the harmonic.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

In the list box, the first column is the index of a harmonic, the second column specifies its frequency, and the remaining columns specify tone coefficients for each fundamental tone that contributed to the listed harmonic.

Output harmonics		From (Hz)	<input type="text" value="1G"/>	Max. Order
<input type="checkbox"/> Select from range		To (Hz)	<input type="text" value="2G"/>	<input type="text" value="2"/>
Index	Frequency	flo	frf	
25	1G	1	0	
46	1.84G	0	2	
48	1.92G	1	1	
50	2G	2	0	

- Array of coefficients** lets you specify an output harmonic by typing its tone coefficients, separated by a space, in the *Tone Coefficients* field and then clicking on the *Clear/Add* button. Values that appear in the list box are absolute values or relative to the fundamental, depending on the selection in the *Sweeptype* cyclic field. To delete a harmonic, select it in the list box and click on *Delete*.

Output harmonics			
<input type="checkbox"/> Array of coefficients			
Index	Frequency	flo	frf
50	2G	2	0
Tone Coefficients <input type="text" value="3"/> <input type="button" value="Clear/Add"/> <input type="button" value="Delete"/>			

- Array of Indices** lets you specify an array of harmonics by typing their indices in the *Additional Indices* type-in field. When you select this choice, currently selected

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

indices appear in the *Currently active indices* field. You type the additional harmonic indices you want in any order and separate these indices with spaces.

Output harmonics	
<input type="checkbox"/> Array of indices	
Currently active indices	<input type="text" value="50"/>
Additional indices	<input type="text" value="3 -2 5"/>

Reference Side-band (Pnoise and QPnoise)

This field appears when you select the Input Source for a Pnoise analysis. It specifies which conversion gain to use when the SpectreRF simulation computes the input-referred noise, noise factor, and noise figure.

- For Pnoise, type an integer value into the field.
 - 0 for amplifiers and filters
 - $-K$ for down converters
 - $+K$ for up converters

where K is the mixing harmonic.

For Pnoise analysis, the reference sideband, as well as the sideband zero, are included in the pool of sidebands used in noise calculations. For example, for the *ne600* test schematic, if `refsideband=-1` and `sidebands=[-2]`, then Pnoise computes contributions from sidebands -2, -1, and 0. If `refsideband=-2` and `sidebands=[-2]`, then Spectre computes contributions from sidebands -2 and 0. The output total noise is different for the two simulation setups. The input-referred noise, noise factor, and noise figure are also different. Pnoise analysis internally includes the `refsideband` as a contribution to the total noise. This inclusion is not reflected in the netlist.

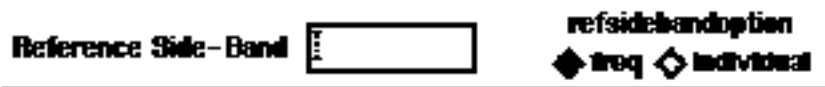
Reference Side-Band	<input type="text" value="0"/>
----------------------------	--------------------------------

- For QPnoise, type a vector value into the field.
 - When the input and output are at the same frequency, use the zero vector `[0 0 ...]`

Affirma RF Simulator (SpectreRF) User Guide

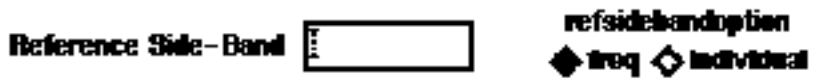
SpectreRF Simulation Forms

- When you do not use the zero vector, the single sideband noise figure is computed.



refsidebandoption (QPnoise)

Lets you specify whether to consider the input at the frequency or at the specific quasi-periodic sideband specified in the Reference Side-Band field.



Relative to fundamental

See [Frequency Sweep Range \(Hz\)](#)

Save Initial Transient Results

Directs the simulator to save the initial transient waveforms. Default is *no*.

Select Ports (PSP only)

Lets you specify the list of active ports for a PSP analysis. For purposes of noise figure computation, the input port is port1, the output port is port2, and an optional auxiliary port is port3. Click the *Select Ports* button to begin specifying ports. Click a *Select* button and then click on a port in the schematic editor. The name of the port appears in the *port(#n)* field.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Enter an integer harmonic value for the port in the *Harmonic* field. Enter additional harmonic values in the *Additional Harmonics* field.

The dialog box titled "Select Parts" contains the following elements:

- Select Parts** (checkbox)
- Input Part(#1)**: A text input field, a **Select** button, and a **Harmonic** field.
- Output Part(#2)**: A text input field, a **Select** button, and a **Harmonic** field.
- Aux. Port(#3)**: A text input field, a **Select** button, and a **Harmonic** field.
- Additional Harmonics**: A text input field.

Select Clock Name

Clicking on this button brings up a form that lets you highlight and choose the clock name for an Envelope Following Analysis.

The dialog box titled "Select Clock Name" contains the following elements:

- Select Clock Name** (title bar)
- OK**, **Cancel**, and **Help** (buttons)
- List of clock names: flo, frf

Select

Clicking on this button lets you specify an output node or voltage source by clicking on a net in the schematic editor. The bottom of the schematic editor window displays *Select Net*.

Sidebands

Lets you select the set of periodic small-signal output frequencies of interest. You can use the choices in the *Sidebands* cyclic field in combination. For example, if you add a sideband using *Array of Coefficients*, the value you added appears in the *Select from Range* list box and as a currently active index for the *Array of Indices*.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **Maximum sideband** prompts you for a maximum sideband value and automatically generates a sideband array of the form [-max. sideband . . . 0 . . . + max. sideband].

Sidebands	
Maximum sideband <input type="checkbox"/>	<input type="text" value="10"/>

- **Select from range** lets you select sidebands in a list box from a range of sideband values that you specify. Values in the list box and the values you specify are absolute values or relative to the fundamental, depending on the selection in the *Sweep type* cyclic field. If you specify a range in the *From (Hz)* and *To (Hz)* type-in fields, all the sideband frequencies displayed in the list box are within that range. You also select the maximum order of harmonics that contribute to the sidebands from the *Max. Order* cyclic field. If, for example, you select 3 as the *Max. Order* value, the sum of the *absolute values* of the tone coefficients contributing to the sidebands in the list box must be less than or equal to three.

You click on sidebands in the list box to select them. You can select adjacent sidebands by clicking and dragging with the mouse over the sidebands you want to select. To select sidebands that are not adjacent, hold the Control key down while you click on the individual sidebands. Clicking on a selected sideband while you hold the Control key down deselects the sideband.

In the list box, the first column is the index of a sideband, the second column and third columns specify the frequency range of the sideband, and the remaining columns specify tone coefficients for each fundamental tone that contributed to the listed sideband.

Affirma RF Simulator (SpectreRF) User Guide
SpectreRF Simulation Forms

Sidebands		From (Hz) <input style="width: 80px;" type="text" value="1G"/>	Max. Order
<input type="checkbox"/> Select from range	To (Hz) <input style="width: 80px;" type="text" value="2G"/>	<input style="width: 40px;" type="text" value="2"/>	
Index	Frequencies	flo	frf
25	1G	1G	0
-46	1.84G	1.84G	2
46	1.84G	1.84G	2
-48	1.92G	1.92G	1
48	1.92G	1.92G	1
-50	2G	2G	0

- **Array of Coefficients** lets you specify a sideband by typing its tone coefficients, separated by a space, in the *Tone Coefficients* type-in field and then clicking on the *Clear/Add* button. Values that appear in the list box are absolute values or relative to the fundamental, depending on the selection in the *Sweeptype* cyclic field. You can place the specified sideband above or below the fundamental using the *upper* cyclic field. To delete a sideband, select it in the list box and click on *Delete*.

Sidebands			
<input style="width: 100%;" type="text" value="Array of coefficients"/>			
Index	Frequencies	flo	frf
73	2.92G	2.92G	1
146	5.84G	5.84G	2
<input type="checkbox"/> upper	Tone Coefficients <input style="width: 150px;" type="text"/>	<input type="button" value="Clear/Add"/>	<input type="button" value="Delete"/>

- **Array of Indices** lets you specify an array of sidebands by typing their indices in the *Additional Indices* field. When you select this choice, currently selected indices

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

appear in the *Currently active indices* field. You type the additional sideband indices you want in any order and separate these indices with spaces.

Sidebands	
Array of indices <input type="checkbox"/>	
Currently active indices	<input type="text" value="73 146"/>
Additional indices	<input type="text"/>

Stop Time

The end point in an Envelope Following Analysis. Make the time interval long enough to let slow signals complete at least one cycle.

Sweep

Lets you specify how a sweep is performed. You have the following choices:

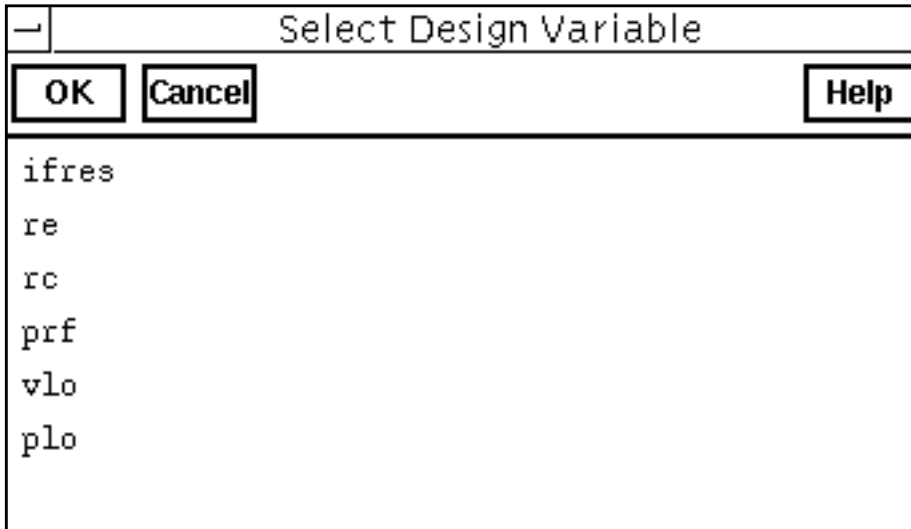
Sweep <input checked="" type="checkbox"/>	Frequency Variable? <input checked="" type="checkbox"/> no <input type="checkbox"/> yes
Variable <input type="checkbox"/>	Variable Name <input type="text"/>
Select Design Variable	

- **Variable** – With this choice, you type the variable name in the *Variable Name* field or click on the *Select Design Variable* button to the right of this field. If you click on the *Select Design Variable* button, the Select Design Variable form comes up with

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

a list of available choices. You select a variable in this form and then click *OK*. If the variable you want is a frequency variable, you click *On yes* for *Frequency Variable*.



– Select Design Variable

OK Cancel Help

ifres
re
rc
prf
vlo
plo

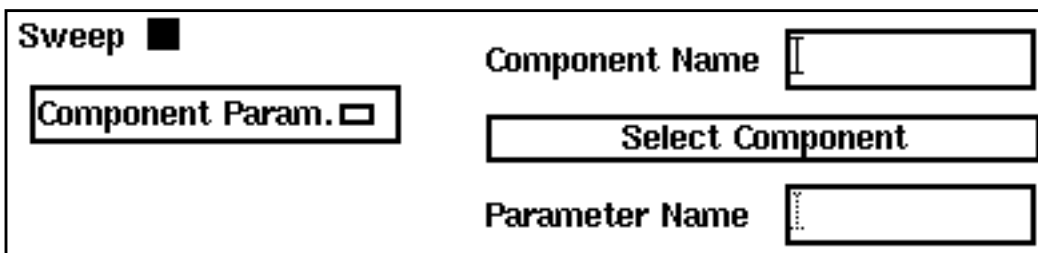
- **Sweep – Temperature** lets you obtain temperature data in your sweep.



Sweep

Temperature

- **Sweep – Component Param** modifies the form to let you sweep a parameter of a component. You can type in the component and parameter names in their respective fields, or you can click on *Select Component* and then click on the component in the schematic. After you click on the component, the *Select Component Parameter* form appears. This form is very similar to the *Select Design Variable* form. You select a parameter in this form and then click *OK*.



Sweep

Component Param.

Component Name

Select Component

Parameter Name

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **Sweep – Model Param** lets you sweep a model parameter. You type the model name and the parameter name in their respective type-in fields.

Sweep

Model Param.

Model Name

Parameter Name

Sweep Range (Hz)

Defines the analysis range.

- **Start – Stop** defines the limits of the sweep. Type the start and stop sweep values for frequencies, periods, or design variables that correspond to your sweep choice.

Sweep Range

Start-Stop **Start** **Stop**

Center-Span

- **Center – Span** lets you specify the limits of the sweep as a *Center* point and a *Span* value.

Sweep Range

Start-Stop **Center** **Span**

Center-Span

Sweeptype (PAC, Pnoise, and PXF)

Controls the inclusion of the *sweeptype* parameter in the Spectre netlist. You have three choices for *Sweeptype*: *absolute*, *relative*, and *blank*, with *blank* selecting the appropriate Spectre default. Results vary depending on whether you are simulating an autonomous circuit (an oscillator) or a driven circuit (a mixer) as determined by the *Oscillator* button selection on the PSS Choosing Analysis form.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

In general, If you are simulating an autonomous circuit (*Oscillator* section of the PSS or SPSS Choosing Analysis form is active), you can select *Sweeptype=relative*. If you leave it blank, it will default to relative. If you are simulating a driven circuit (*Oscillator* section of the PSS or SPSS Choosing Analysis form is not active), you can either select *Sweeptype=relative* or *Sweeptype=absolute*. If you leave it blank, it will default to absolute.

- **Sweeptype - Absolute** puts *sweeptype=absolute* in the Spectre netlist.

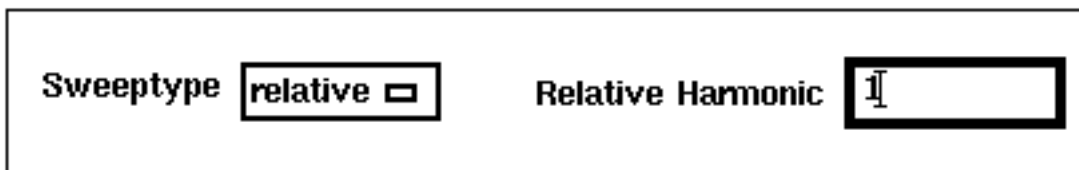
In the output, the x-axis corresponds to the *Start* and *Stop* values in the Pnoise sweep. There is no indication on the plot that you selected *sweeptype=absolute*.



A screenshot of a software interface showing a dropdown menu for 'Sweeptype'. The word 'Sweeptype' is on the left, and a box contains the word 'absolute' with a small square icon to its right.

- **Sweeptype - Relative** puts *sweeptype=relative* in the Spectre netlist. When you select *Sweeptype=relative*, you must also enter a value for *Relative Harmonic*. If you enter a 1, it appears in the netlist as *relharmnum=1*.

In the output, the x-axis corresponds to the *Start* and *Stop* values in the Pnoise sweep. The plot label indicates the selected relative harmonic (for example, *relharm=1*). In prior versions of the software, there is no indication.



A screenshot of a software interface showing two dropdown menus. The first is labeled 'Sweeptype' and has 'relative' selected. The second is labeled 'Relative Harmonic' and has '1' selected.

- **Sweeptype - Blank** does not put the *sweeptype* parameter in the Spectre netlist.
 - For autonomous circuits, Spectre automatically sets the *Sweeptype=relative* and *relharmnum=1*.
 - For driven circuits, Spectre automatically sets the *Sweeptype=absolute* and displays the message *Sweep is Currently Absolute*.

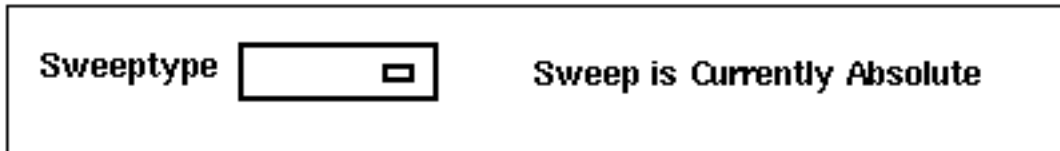
In the output, the x-axis corresponds to the *Start* and *Stop* values in the Pnoise sweep.

- For autonomous circuits, the plot label indicates the default relative harmonic (*relharm=1*). In prior versions of the software, there is no indication.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- ❑ For driven circuits, Spectre automatically sets the *Sweeptype=absolute*. There is no indication you selected *sweeptype=blank*.



Sweeptype (PSP)

Controls the inclusion of the *sweeptype* parameter in the Spectre netlist. You have three choices for *Sweeptype*: *absolute*, *relative*, and *blank*, with *blank* selecting the appropriate Spectre default.

With PSP analysis, the frequency of the input and the frequency of the response are usually different. Since PSP computations involve inputs and outputs at frequencies that are relative to multiple harmonics, *Sweeptype* behaves differently in PSP analysis than it does in PAC, Pnoise, and PXF analyses.

- Specifying *Sweeptype=relative*, indicates to sweep relative to the analysis harmonics (rather than the PSS fundamental).
- Specifying *Sweeptype=absolute*, indicates to sweep the absolute input source frequency.

Sweep Type (For small-signal analyses)

- **Sweep Type- Automatic** lets the simulator decide whether the sweep type is linear or logarithmic. The sweep is linear if the ratio of start to stop values is less than 10 and logarithmic if this ratio is 10 or higher.



Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **Sweep Type- Linear** specifies a linear sweep. The form changes to let you type either the step size or the total number of points (steps).

Sweep Type		
<input checked="" type="checkbox"/> Linear	<input type="checkbox"/> Step Size	<input type="text" value="2"/>
	<input checked="" type="checkbox"/> Number of Steps	

- **Sweep Type- Logarithmic** specifies a logarithmic sweep. The form changes to let you type either the number of points per decade or the total number of points (steps).

Sweep Type		
<input checked="" type="checkbox"/> Logarithmic	<input checked="" type="checkbox"/> Points Per Decade	<input type="text"/>
	<input type="checkbox"/> Number of Steps	

Sweep Type (For PSS, PSP, and Pdisto)

- **Linear** specifies a linear sweep. The form changes to let you type either the step size or the total number of points (steps).

Sweep Type		
<input checked="" type="checkbox"/> Linear	<input checked="" type="checkbox"/> Step Size	<input type="text" value="2"/>
<input type="checkbox"/> Logarithmic	<input type="checkbox"/> Number of Steps	

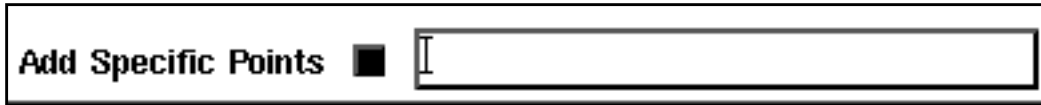
- **Logarithmic** specifies a logarithmic sweep. The form changes to let you type either the number of points per decade or the total number of points (steps).

Sweep Type		
<input type="checkbox"/> Linear	<input checked="" type="checkbox"/> Points Per Decade	<input type="text"/>
<input checked="" type="checkbox"/> Logarithmic	<input type="checkbox"/> Number of Steps	

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- **Add Specific Points** lets you specify additional sweep points for the analysis. When you click on this button, a field opens to the right of the button. You type the additional sweep point values into the field and separate them with spaces.



Options Forms

Clicking on an *Options* button on the Choosing Analysis form opens an Options form that lets you specify simulation options for an analysis.

- The *Options* button that appears when the Choosing Analyses form first opens lets you specify options for the analysis you originally selected.
- You see a second *Options* button in the expanded form if you specify data for an additional small-signal analysis. This second *Options* button brings up an Options form for the small-signal analysis.

Options that are not relevant for a particular analysis do not appear on its Options form.

The parameters you can specify on an options form are described in the following sections.

ACCURACY PARAMETERS

relref is the reference used for relative convergence criteria. Your accuracy default choice sets the default value.

pointlocal compares the relative errors in quantities at each node to that node alone.

alllocal compares the relative error at each node to the largest values ever found for that node.

sigglobal compares relative errors in each signal to the maximum value for all signals.

allglobal is the same as sigglobal except that it also compares the residues for each node to the historical maximum.

iteratio is the ratio the simulator uses to compute LTE tolerances from the Newton tolerance. The default value is based on the accuracy default.

steadyratio is the ratio the simulator uses to compute steady-state tolerances from the LTE tolerance. This parameter adjusts the maximum allowed mismatch in node voltages and current branches during the steady-state period. The default is based on the accuracy default.

maxacfreq is the maximum frequency used in a subsequent periodic small-signal analysis. This parameter automatically adjusts *maxstep* to reduce errors due to aliasing in frequency-domain results. The default is based on *maxstep* and *harms* values. Read "[SpectreRF Theory](#)" in the online documentation for more information about specifying this parameter.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

maxperiods is the maximum number of simulated periods allowed for the simulation to reach steady-state. Default is 20.

Here are the default values controlled by the accuracy parameter settings:

accuracy	relref	method	steadyratio	Iteratio
liberal	allglobal	gear2	0.1	3.5
moderate	sigglobal	traonly	0.001	3.5
conservative	alllocal	gear2only	0.00001	10.0

finitediff lets you choose options for the finite difference method. Possible choices are *yes*, *samegrid*, or *no*.

For *yes*, PSS analysis improves the grid of time-steps and applies the finite difference method.

For *samegrid*, PSS analysis applies the finite difference method but keeps the grid of time-steps used by the shooting method.

For *no*, PSS analysis does not use the finite difference method. If you reuse a previously saved PSS analysis result, *fdd* is turned on automatically.

ANNOTATION PARAMETERS

stats tells the simulator to generate analysis statistics. Default is *no*.

annotate lets you specify what information is printed at the beginning of the output to identify the results. Default is *status*.

CONVERGENCE PARAMETERS

readns lets you specify the name of a file that contains an estimate of the initial transient solution. Enter the complete path to the file. No default.

cmin is the minimum capacitance from each node to ground. Default is 0.

tolerance is the relative tolerance for the linear solver when solving for convergence. Default is 10^{-9} .

gearorder is the order used for Gear-type interpolations. Default is 2 (second order).

solver

std specifies that Krylov subspaces not be used. Simulations with this setting are slower but more robust than simulations using the turbo setting. Use this setting if simulation with *turbo* is unsuccessful.

exp currently has the same effect as *std*.

turbo specifies the use of Krylov subspaces and makes small-signal analyses run more quickly. However, it is less robust than *std*. This setting is the default.

oscsolver

std specifies that Krylov subspaces not be used. The setting is more robust than *turbo* for oscillator small-signal analyses, although the simulation is slower. Use this setting if simulation with *turbo* is unsuccessful.

turbo specifies the use of Krylov subspaces and makes oscillator small-signal analyses run more quickly. However, it is less robust than *std*. This setting is the default.

INITIAL CONDITION PARAMETERS

ic specifies the methods used to set the initial condition. For an explanation of each of these methods, consult the "[Spectre Reference](#)" manual. Default is *all*.

skipdc

yes omits the DC analysis from the transient analysis.

no includes a DC analysis with the transient analysis. This is the default.

waveless omits the initial DC analysis and fixes the independent source values to their initial values rather than their DC values.

rampup starts the independent source values at 0 and ramps them up to the initial values in the first 10% of the analysis interval.

autodc automatically selects either *waveless* or *rampup*, depending on whether you specify a nonzero initial condition.

readic lets you specify the name of the file that contains the initial conditions.

INTEGRATION METHOD PARAMETERS

method specifies the integration method. Your accuracy default choice sets the default value. The possible settings are

euler is backward Euler.

trap is the backward Euler and trapezoidal methods.

traponly is the trapezoidal rule only.

gear2 is the backward Euler and second-order Gear methods.

gear2only is Gear's second-order backward difference method only.

The trapezoidal rule is best when you want high accuracy, but it can exhibit point-to-point ringing, which you can control with tighter error tolerances. Euler and Gear work better with looser tolerances for quick simulation, but they can make systems appear more stable than they actually are.

Use the *gear_order* option on the small-signal analyses Options form to set the order of a Gear-type interpolation.

NEWTON PARAMETERS

maxiters is the maximum number of iterations per time step.

restart tells the simulator not to use the previous DC solution as an initial guess. Default is *yes* (means do not use).

envmaxiters lets you specify the maximum number of Newton iterations per envelope step. The default is 3.

OUTPUT PARAMETERS

save tells the simulator what signal to save. You have the following choices:

selected saves only signals that you specifically request from the *Outputs* menu in the Simulation window. It is the default setting.

lvlpub saves all signals that are normally useful up to *nestlvl* deep in the subcircuit hierarchy. When you click on this button, the *nestlvl* type-in field below the *lvlpub* button activates to let you specify *nestlvl*.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

lvl saves all signals that are up to *nestlvl* deep in the subcircuit hierarchy. When you click on this button, the *nestlvl* field below the *lvlpub* button activates to let you specify *nestlvl*.

allpub saves all signals that are normally useful.

all saves all signals.

nestlvl is used with *lvl* and *lvlpub*. You type an integer that specifies the number of levels in the hierarchy that you want to save signals.

oppoint specifies whether the simulator outputs the operating point information. You can send the information to a rawfile, the logfile, or the screen. Default is *no*.

skipstart specifies when the simulator starts skipping output data. Default is *starttime s* (seconds).

skipstop specifies when the simulator stops skipping output data. Default is *stoptime s* (seconds).

skipcount specifies how many points to skip before saving a point. No default.

strobeperiod is the output strobe interval in seconds.

strobedelay is the delay (phase shift) between the skipstart time and the first strobe point. Default is 0.

compression directs the simulator to perform data compression on the output. Default is *no*.

stimuli specifies what PXF uses for inputs to the transfer functions.

sources specifies that the sources present in the circuit are used as the inputs to the transfer functions.

nodes_and_terminals specifies that all possible transfer functions are computed.

freqaxis specifies what version of the frequency to plot the output against in spectral plots.

- For the PAC and PXF analyses

absin is the absolute value of the input frequency.

in is the input frequency.

out is the output frequency.

- For the PSP analysis

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

- ❑ *absin* is the absolute value of the frequency swept at the input
- ❑ *in* is the scattered frequency at the input
- ❑ *out* is the scattered frequency at the output

compression performs data compression on the output.

outputperiod lets you specify the time-domain output period. The time-domain small-signal response is computed for the period specified, rounded to the nearest integer multiple of the PSS analysis period.

outputtype lets you specify the output type for Envelope Following Analysis. Possible values are *both*, *envelope*, or *spectrum*. The default is *both*.

saveallsidebands lets you save all sidebands.

SIMULATION INTERVAL PARAMETERS

tstart is the start time you specify for transient analysis. It can be negative or positive. Default is 0.

start lets you specify the start time for the Envelope Following Analysis.

outputstart lets you specify the timepoint when the simulator starts to save output.

SIMULATION BANDWIDTH PARAMETERS

modulationbw lets you specify the modulation bandwidth.

STATE FILE PARAMETERS

write lets you specify the name of the file to which the SpectreRF simulation writes the initial transient solution.

writefinal lets you specify the name of the file to which the SpectreRF simulation writes the final transient solution.

swapfile is a temporary file that holds steady-state information. If you enter a filename, the Spectre circuit simulator stores the operating point in that file rather than in virtual memory. Use this option if you receive a warning about not having enough memory to complete the analysis. Enter the complete path to the file.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

writepss specifies the file to which the steady-state solution is written. Small-signal analyses can read the steady-state solution from this file so rerunning the PSS analysis is unnecessary.

readpss specifies the file from which the steady-state solution is read. Small-signal analyses can read the steady-state solution from this file so rerunning the PSS analysis is unnecessary.

TIME STEP PARAMETERS

step is the smallest simulator time step used to improve the look of the results. Default is 0.001 x fundamental period seconds.

maxstep is the largest allowable time step. The default is set by the accuracy default.

liberal = 0.1/max AC frequency

moderate = 2 x liberal

conservative = 4 x liberal

maxenvstep specifies the maximum outer envelope size. The default is derived from the *errpreset* setting.

Results Forms

1st Order Harmonic

See First-Order Harmonic

2nd-7th Order Harmonic

See [Nth Order Harmonic](#)

Add To Outputs

Adds the expression plotted in the Direct Plot window to the Outputs list in the [Simulation window](#).

Analysis Type

Selects which analysis you can plot the results from. If you include a small-signal analysis, it has its own button. Clicking on a different analysis button changes the appearance of the Results form.

Circuit Input Power lets you choose between *Single Point* and an appropriate sweep. If you choose *Single Point*, the *Input Power Value (dBm)* field appears, and you must type a value into it. If you choose a sweep, the *Extrapolation Point* field appears, and you can type a value. Placing a value in the *Extrapolation Point* field is optional.

Close Contours

Is used in plotting power and reflection contours. When you select this option, the plot appears as a closed figure. The default is to leave the two most distant points in the plot unconnected.

Extrapolation Point

The optional *Extrapolation Point* field lets you specify the value of the design variable from which the straight line approximations of first- and third-order harmonics are produced. If you do not specify a value, the default value is the smallest X-axis sweep value.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

First-Order Harmonic

This list box appears when you select the IPN curves function. It shows all available harmonics for plotting by their number and associated frequency values. You select one item from the list box and then select the appropriate net on the schematic.

1st Order Harmonic	
0	0
1	40M
2	80M
3	120M
4	160M

1st Order Harmonic (For Pdisto)

This list box gives you additional information that is relevant for Pdisto as well as the information normally available in the *1st Order Harmonic* list box.

In the list box, the first column is the frequency of a harmonic, and the second and third columns specify the tone coefficients for each fundamental tone that contributed to the listed harmonic.

	Freq. (Hz)	flo	frf
1st Order Harmonic	0	0	0
	80M	1	-1
	840M	-1	2
	920M	0	1
	1G	1	0

Function

Each *Function* button specifies a different quantity that you can plot. The buttons are different depending on the *Analysis Type* you select. For some functions, you must select one or

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

two objects on the schematic after selecting the *Function* button. The chart below tells you what object to select for each of these functions.

Function	Select Object(s)
<i>Voltage</i>	Net
<i>Current</i>	Terminal
<i>Power</i>	Terminal
<i>Voltage Gain</i>	Signal & Reference Nets
<i>Current Gain</i>	Signal & Reference Terminals
Power Gain	Signal & Reference Ports
<i>Transconductance</i>	Signal Terminal & Reference Net
<i>Transimpedance</i>	Signal Net & Reference Terminal
<i>1dB Compression</i>	Net
<i>IPN curves</i>	Net

Gain Compression

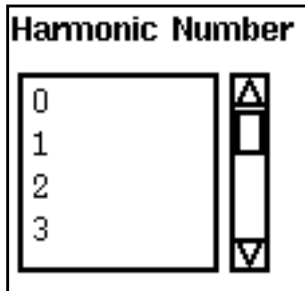
Lets you specify the Gain Compression when you plot the compression point.

Get Reference Reflection Coefficients by Selecting a Separate Terminal

When you highlight this button, you can select a second point on the schematic as a reference point for the reflection coefficients.

Harmonic Number

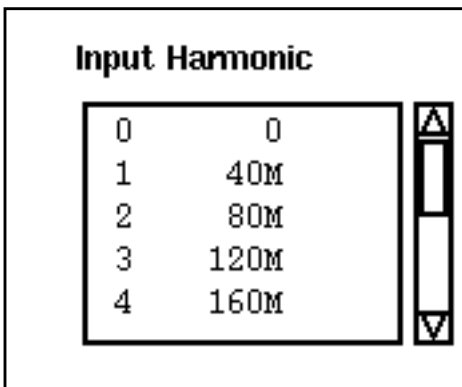
This list box lets you choose the harmonic number in the Envelope Following Analysis.



The screenshot shows a window titled "Harmonic Number" with a list box containing the values 0, 1, 2, and 3. To the right of the list box is a vertical scrollbar with up and down arrow buttons.

Input Harmonic

This list box lets you select the value of an input harmonic for plotting. It appears on the PSS Results form when you select the functions *Voltage Gain*, *Current Gain*, *Power Gain*, *Transconductance*, or *Transimpedance*. The values in the list box are those you requested in the Output Harmonics specification in the Choosing Analyses form.



The screenshot shows a window titled "Input Harmonic" with a list box containing the following pairs: 0 (0), 1 (40M), 2 (80M), 3 (120M), and 4 (160M). To the right of the list box is a vertical scrollbar with up and down arrow buttons.

Input Harmonic (For Pdisto)

This list box gives you additional information that is relevant for Pdisto as well as the information normally available in the *Input Harmonic* list box.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

In the list box, the first column is the frequency of a harmonic, and the second and third columns specify the tone coefficients for each fundamental tone that contributed to the listed harmonic.

	Freq. (Hz)	flo	frf
Input	0	0	0
Harmonic	80M	1	-1
	840M	-1	2
	920M	0	1
	1G	1	0

Add To Outputs

Input Power Value (dBm)

You must specify a value for this field if you run a single point analysis as opposed to a power sweep or if you want information about a single point after running a power sweep.

Input or Output Referred 1dB Compression – Lets you choose input or output referred compression. Output referred compression is referred to the Y axis and input referred compression is referred to the X axis.

Input or Output Referred IP3 – Lets you choose input or output referred IP3. Output referred IP3 is referred to the Y axis and input referred IP3 is referred to the X axis.

Maximum Power

Lets you specify a maximum power level for your plot when you plot power contours.

Max Reflection Mag

Lets you specify a maximum reflection magnitude when you plot reflection contours.

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Minimum Power

Lets you specify a minimum power level for your plot when you plot power contours.

Min Reflection Mag

Lets you specify a minimum reflection magnitude when you plot reflection contours.

Modifier

This function sets the units of the y axis.

Magnitude is the raw value, in volts, amps, or no units at all.

Phase sets the Y axis to degrees.

dB20 sets the Y axis to decibels with tick marks every 20 dB.

dBm sets the Y axis to dB 10 plus 30.

dB10 sets the Y axis to decibels with tick marks every 10 dB.

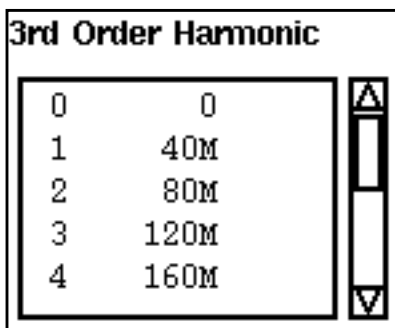
Real and **Imaginary** restrict plots to only the real or imaginary range of the curve.

Number of Contours

Lets you specify the number of power or reflection contours that you want to plot.

Nth Order Harmonic

This list box appears when you select the IPN curves function. It shows all the harmonics available for plotting by their number and associated frequency values. You select one item from the list box and then select the appropriate net on the schematic.



Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

Nth Order Harmonic (For Pdisto) – This list box gives you additional information that is relevant for Pdisto as well as the information normally available in the *Nth Order Harmonic* list box.

In the list box, the first column is the frequency of a harmonic, and the second and third columns specify the tone coefficients for each fundamental tone that contributed to the listed harmonic.

	Freq. (Hz)	flo	frf
3rd Order Harmonic	0	0	0
	80M	1	-1
	840M	-1	2
	920M	0	1
	1G	1	0

Order

Lets you select the order of the harmonics you wish to plot.

Order	<input type="text" value="3rd"/>
-------	----------------------------------

Output Harmonic

You click on sidebands in the list box to select them. You can select adjacent sidebands by clicking and dragging with the mouse over the sidebands you want to select. To select sidebands that are not adjacent, hold the `Control` key down while you click on the individual

Affirma RF Simulator (SpectreRF) User Guide

SpectreRF Simulation Forms

sidebands. Clicking on a selected sideband while you hold the `Control` key down deselects the sideband.

0	0
1	40M
2	80M
3	120M
4	160M

Output Harmonic (For Pdisto) – This list box gives you additional information that is relevant for Pdisto as well as the information normally available in the *Output Harmonic* list box. In the list box, the first column is the frequency of a harmonic, and the second and third columns specify the tone coefficients for each fundamental tone that contributed to the listed harmonic.

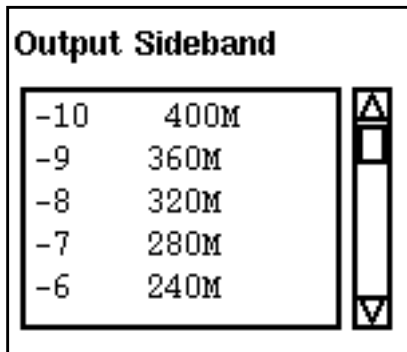
	Freq. (Hz)	flo	frf
Output Harmonic	0	0	0
	80M	1	-1
	840M	-1	2
	920M	0	1
	1G	1	0

Output Sideband

This list box appears when you choose *variable* for *Sweep* on a small-signal Results form. It lists all the sidebands you requested on the small-signal Choosing Analyses form.

You click on sidebands in the list box to select them. You can select adjacent sidebands by clicking and dragging with the mouse over the sidebands you want to select. To select sidebands that are not adjacent, hold the `Control` key down while you click on the individual

sidebands. Clicking on a selected sideband while you hold the `Control` key down deselects the sideband.



Plot

Displays a waveform in the *Waveform* window.

Plot Mode

Use *append* to combine the next plot you perform with other curves already plotted in the *Waveform window*, or use *replace* to clear the *Waveform* window just prior to the next plot.

Power Spectral Density Parameters

- **Time Interval** – The starting and ending times for the spectral analysis interval. They are usually the start and stop times for the simulator.
- **Nyquist half-bandwidth** – The maximum frequency at which there are signals of interest. This is usually three to five times the maximum band frequency.
- **Frequency bin width** – The frequency resolution, such as the width of the frequency bins.
- **Max. plotting frequency** – Sets the maximum X axis value for the waveform you want to plot.
- **Min. plotting frequency** – Sets the minimum X axis value for the waveform you want to plot.
- **Windowing** – A preset list of available windowing functions used during the spectrum calculation.
- **Detrending** – Lets you remove trends from the data before the spectral analysis.

Reference Resistance

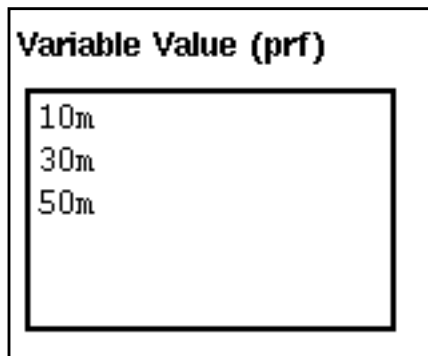
Lets you specify the resistance of the port adapter when you plot power or reflection contours.

Sweep

- *spectrum*, *sideband*, and *frequency* set the X axis of the plot to frequency
- *time* sets the X axis of the plot to seconds
- *variable* sets the X axis to the value of a design variable.

Variable Value

The PSS and Pdisto Results forms display this list of sweep values that you can specify for a specific variable, temperature, component parameter, or model parameter. The range of values is determined by your Sweep Range specification and the number of values by your *Sweep Type* specification in the Choosing Analyses form. In the sample figure, which shows the PSS version, the values are listed for the design variable *prf*. The Pdisto version of the form is formatted slightly differently but gives the same information.



The image shows a screenshot of a software form titled "Variable Value (prf)". The form contains a list of three values: "10m", "30m", and "50m". The list is enclosed in a rectangular box within the form's border.

Setting Up for the Examples

This chapter tells you how to set up your software so you can follow the instructions and run the examples in the chapters about simulating specific components.

Setting Up the Software

Before you perform the SpectreRF simulator analyses, you must set up the component files and start the Cadence® software.

Copying the SpectreRF Simulator Examples

Copy the *rfExamples* library into your account so that you can edit the sample design.

The library is located at

```
<CDSHOME>/tools/dfII/samples/artist/rfExamples
```

where *CDSHOME* is the installation directory for your Cadence software.

Setting Up the Cadence Libraries

The Cadence Libraries are defined in the UNIX text file *cds.lib*. You can edit this file in a UNIX shell window or by using the Library Path Editor while running the Cadence software.

Using a UNIX Shell Window

To set up the libraries in a UNIX shell window, use the following procedure:

1. In a UNIX shell window, open the *cds.lib* file for editing using *vi*, *emacs* or a similar text editor.

The *cds.lib* file is in your installation directory.

Affirma RF Simulator (SpectreRF) User Guide

Setting Up for the Examples

2. In the *cds.lib* file, specify definitions for the *basic*, *analogLib*, *sample*, and *spectreSModels* libraries. In addition, define a user library where the sample circuits can be tested. The definitions in the *cds.lib* file look like the following:

```
DEFINE rf_test my_dir/rf_test
DEFINE basic <CDSHOME>/tools/dfII/etc/cdslib/basic
DEFINE analogLib <CDSHOME>/tools/dfII/etc/cdslib/artist/analogLib
DEFINE sample <CDSHOME>/tools/dfII/samples/cdslib/sample
DEFINE spectreSModels <CDSHOME>/tools/dfII/samples/artist/
    spectreSModels
```

You can label your test library any name you choose. This example assumes that you have called the library *rf_test*. The name *my_dir* represents the directory into which you copied the design library.

You must use the names *basic*, *sample*, *analogLib*, and *spectreSModels*. You cannot rename those four libraries.

Using the Library Path Editor

To access the Library Path Editor, use the following procedure.

1. In a UNIX window, type *icms* to start the Cadence software.
The Command Interpreter Window (CIW) appears.
2. In the CIW, choose *Tools – Library Path Editor*.
The Library Path Editor form appears.

Affirma RF Simulator (SpectreRF) User Guide

Setting Up for the Examples

3. In the Library Path Editor, follow the instructions at the bottom of the form.

Library Path Editor: /oldusr/mnt2/frank/cds.lib [

Library	Path
ahdlLib	/mnt3/sruba/ahdlLib/ahdlLib
US_8ths	\$CDS_INST_DIR/tools/dfII/etc/cdslib/sheets/US_8ths
rfExamples	/mnt2/frank/rfExamples/rfExamples
test	/mnt3/sruba/test/goran/libs/biCmDesign
ckts	/mnt3/sruba/ckts/uwTest
uwTest	/mnt3/sruba/uwTest/uwTest
test441	/mnt3/sruba/test441/uwTest
rfExamples1	/mnt3/sruba/rfExamples1/rfExamples
rfExamples2	\$CDS_INST_DIR/tools/dfII/samples/artist/rfExamples
tmodels	/mnt3/sruba/tmodels/models
spectreSMode	/mnt3/sruba/spectreSModels/spectreSModels

To add a new library definition, type the name in the Library column and the path to the library in the Path column. Press RETURN to continue adding libraries. When done, select the menu 'File -> Save ^s' to save your edits.

4. Type in each required library and its associated path. See the previous section for a list of the required libraries.

Library Conversion of SpectreS Libraries

If you have libraries that you have been using with SpectreS simulation, you need to convert them before you can use them with Spectre direct simulation. To learn more about the necessary conversion, see the [information about library conversion in the *Compatibility Guide*](#).

Simulating Mixers

To use this chapter, you must be familiar with the SpectreRF simulator analyses as well as know about mixer design. For more information about the SpectreRF simulator analyses, see [Chapter 1](#).

The SpectreRF simulator can simulate circuits, such as mixers, that show frequency conversion effects. This chapter uses a commercially available integrated circuit mixer, the [ne600p](#), configured as a down converter to illustrate how the SpectreRF simulator can determine the characteristics of your design. The ne600p circuit examples illustrate both the capabilities and the requirements of the SpectreRF simulator. The examples show you how to perform a simulation with an RF to IF ratio of almost 12 to 1. The examples also show you how to be sure that all time-varying independent signal sources have a common integral multiple (40 MHz in this case).

In the mixer examples, you plot the following nonlinear characteristics of the ne600p mixer:

Measurements	Analyses
Harmonic distortion	PSS
Noise figure	PSS and Pnoise
Noise figure	PSS and PSP
Periodic S-parameters	PSS and PSP
Conversion gain	PSS and PXF
Power supply rejection	PSS and PXF
1 dB compression point	Swept PSS
Intermodulation distortion	Swept PSS
Third-order intercept point	Swept PSS and PAC
Intermodulation distortion	Pdisto
Noise figure	Pdisto and QPnoise

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

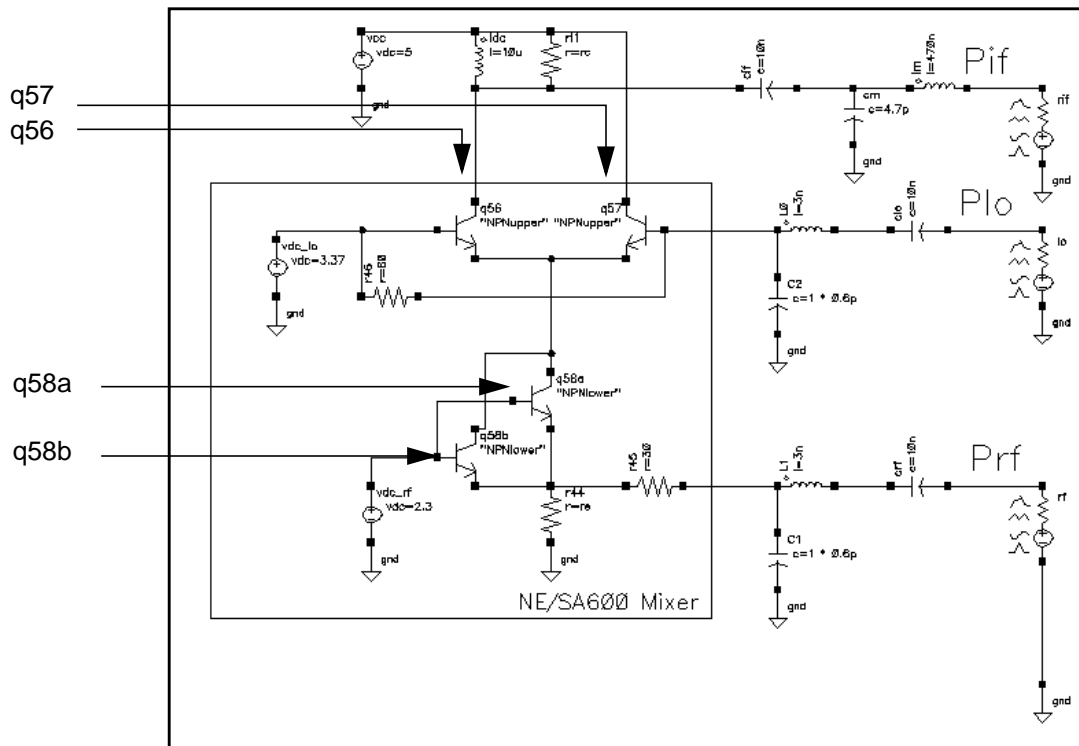
In order to make these measurements, you run one of the following combination of analyses.

- A Periodic Steady-State analysis (PSS), and the following periodic small-signal analyses: Periodic AC (PAC) analysis, Periodic Transfer Function (PXF) analysis, Periodic Noise (Pnoise) analysis, and Periodic S-Parameter (PSP) analysis.
- A Periodic Distortion (Pdisto) Analysis, and the quasi-periodic small-signal analysis: Quasi-Periodic Noise (QPnoise) analysis. The Pdisto analysis is also referred to as the Quasi-Periodic Steady-State (QPSS) Analysis

The ne600p Mixer Circuit

The ne600p integrated circuit is commercially available and is designed for low power communication systems from 800-1200 MHz. The receiver is a single balanced mixer containing four bipolar transistors. the schematic for the ne600p mixer circuit is shown in Figure 4-1.

Figure 4-1 Schematic for the ne600p Mixer Circuit



Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The high-level local oscillator (LO) signal (~0 dBm at 50 Ohms) enters through the emitter follower q57. The transistor q56 is in common base configuration. The q56 emitter follows the q57 emitter. When the LO signal goes high, both emitters also go high and the B-E voltage on q56 decreases. As a result, the two BJTs form a balanced pair.

The collector currents of q56 and q57 are 180 degrees out of phase. The q58 pair feed the low-level (-30 to -20 dBm) RF signal to the balanced pair. The IF signal is drawn from the open collector output of q56 through a bandpass passive network.

The following tables list some measured values for different aspects of the ne600p mixer.

Measurement	Measured
LO frequency (Hz)	1 GHz
RF frequency (Hz)	920 MHz
IF frequency (Hz)	80 MHz
LO power	0 dBm
RF power	-30 dBm
Conversion gain	-2.6 dB
Noise figure	16 dB
Input 1dB compression point	-4 dBm
Input IP3 (from swept power)	6 dBm
Input IP3 (from PAC analysis)	6 dBm

Design Variable	Default Value
prf (RF power)	-30 dBm
vlo (LO magnitude)	316.2 m
frf (RF frequency)	920 MHz
flo (LO frequency)	1 GHz

Before you use the ne600p circuit from the sample library, be sure that the *frf* design variable is set to the appropriate value, either 920 MHz or 900 MHz, depending on the simulation. If you adjust the RF frequency to the center of the IF tuned circuit, approximately 100 MHz, the conversion gain becomes -2.57dB and the other values change slightly

Simulating the ne600p Mixer

Before you start, perform the setup procedures described in [Chapter 3](#).

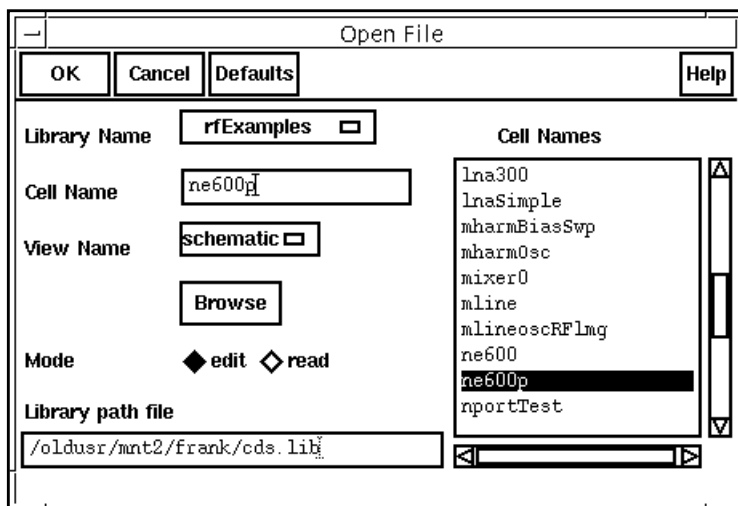
Opening the ne600p Mixer Circuit in the Schematic Window.

1. In the CIW, choose *File – Open*.

The Open File form appears.

2. In the Open File form, choose *rfExamples* in the *Library Name* menu and choose *ne600p* in the *Cell Names* list box.

The completed Open File form appears like the one below.



3. Click *OK*.

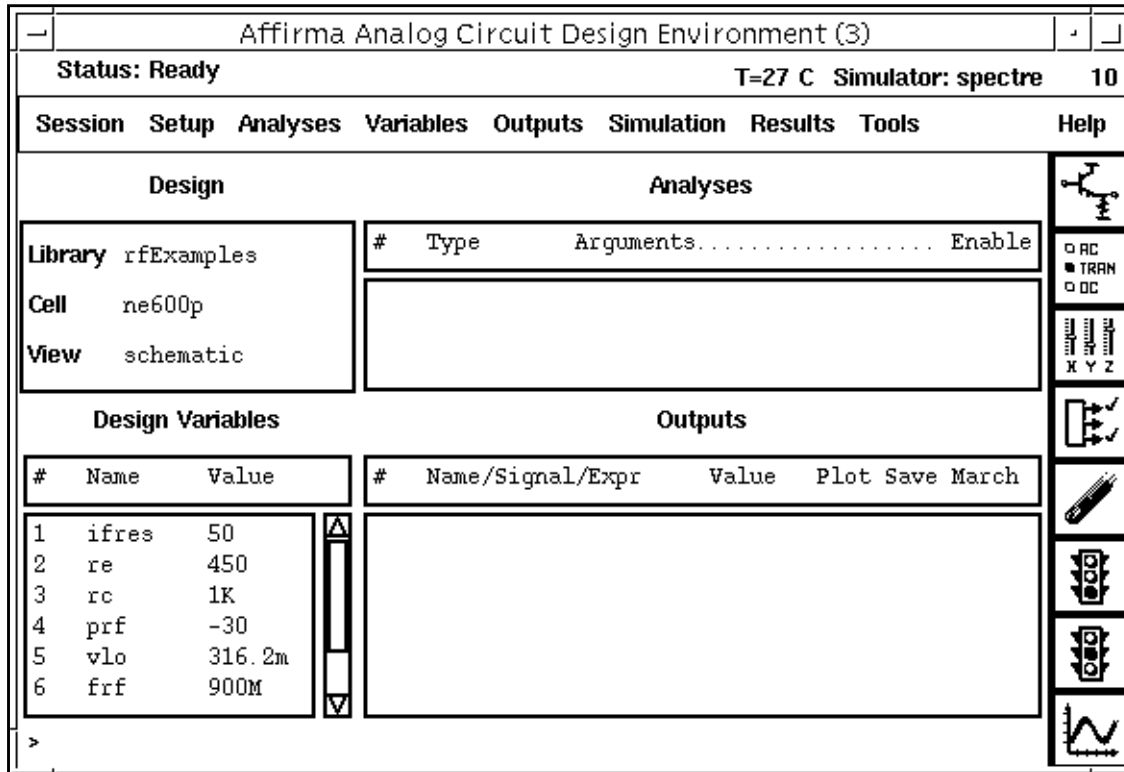
The Schematic window for the *ne600p* mixer appears.

4. In the Schematic window, choose *Tools – Analog Environment*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The Simulation window opens.



You can also use *Tools – Analog Environment – Simulation* in the CIW to open the Simulation window without opening the design. You can open the design later by choosing *Setup – Design* in the Simulation window and choosing the *ne600p* in the Choosing Design form.

Choosing Simulator Options

1. Choose *Setup – Simulator/Directory/Host* in the Simulation window.

The Simulator/Directory/Host form appears.

2. In the Simulator/Directory/Host form, specify the following:
 - a. Choose *spectre* for the *Simulator*.
 - b. Type the name of the project directory, if necessary.
 - c. Highlight the *local* or the *remote* button to specify the *Host Mode*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

For remote simulation, type the name of the host machine and the remote directory in the appropriate fields.

The completed form appears like the one below.

The screenshot shows a dialog box titled "Choosing Simulator/Directory/Host -- Affirma Analog Circuit Desi". The dialog has a standard Windows-style title bar and a button bar at the top with "OK", "Cancel", "Defaults", and "Help" buttons. The main area contains the following fields and options:

- Simulator:** A text box containing "spectre" with a small square icon to its right.
- Project Directory:** A text box containing "~/simulation".
- Host Mode:** Three radio buttons labeled "local", "remote", and "distributed". The "local" radio button is selected.
- Host:** An empty text box.
- Remote Directory:** An empty text box.

3. In the Simulator/Directory/Host form, click *OK*.
4. In the Simulation window, choose *Outputs – Save All*.

The Save Options form appears.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

5. In the *Select signals to output section*, be sure *allpub* is highlighted.

Save Options

OK Cancel Defaults Apply Help

Select signals to output (save) none selected lvlpub lvl allpub all

Select power signals to output (pwr) none total devices subckts all

Set level of subcircuit to output (nestlvl)

Select device currents (currents) selected nonlinear all

Set subcircuit probe level (subcktprobelvl)

Select AC terminal currents (useprobes) yes no

Select AHDL variables (saveahdlvars) selected all

Save model parameters info

Save elements info

Save output parameters info

Setting Up Model Libraries

1. In the Simulation window, choose *Setup – Model Libraries*.

The Model Library Setup form appears.

2. In the *Model Library File* field, type the full path to the model file including the file name, `rfModels.scs`.

3. In the Model Library Setup form, click on *Add*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed form appears like the one below.

The screenshot shows a dialog box titled "spectre0: Model Library Setup". At the top, there are buttons for "OK", "Cancel", "Defaults", "Apply", and "Help". The main area is divided into two sections. The first section is labeled "Model Library File" and "Section". The "Model Library File" field contains the text "...5/pink/tools/dfII/samples/artist/models/spectre/rfModels.scs". The "Section" field is empty. The second section is labeled "Model Library File" and "Section (opt.)". Both fields in this section are empty. Below the second section are buttons for "Add", "Delete", "Change", "Edit File", and "Browse...". An arrow points to the empty "Model Library File" field in the second section with the text "Type here."

4. In the Model Library Setup form, click *OK*.
5. Disable any analyses you ran previously.

Setting Design Variables

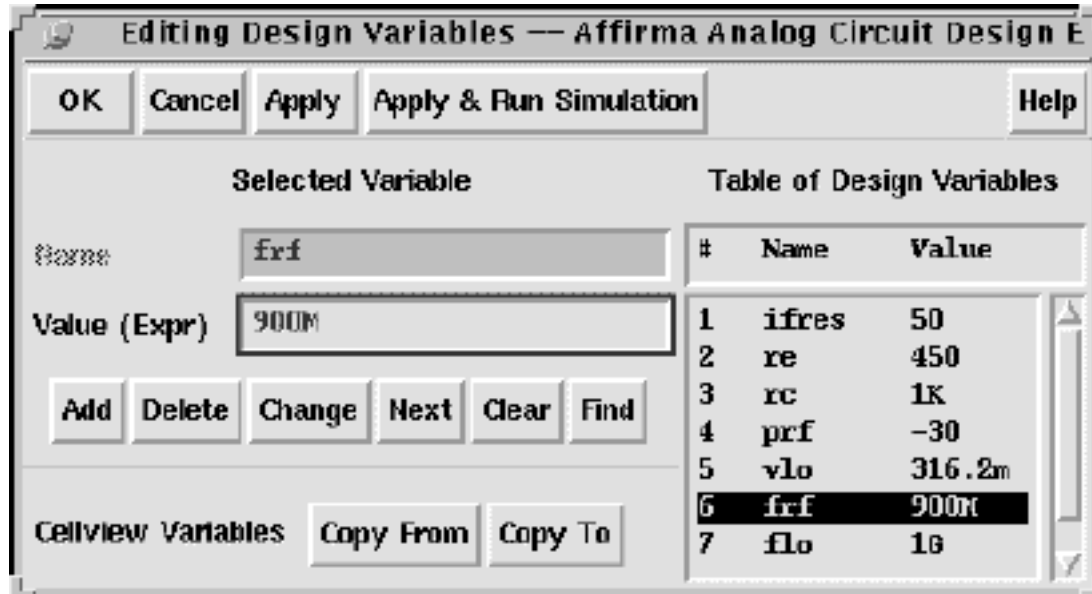
In the Simulation window, use the following procedure to set up the design variables `frf` and `flo` to the values required for each simulation. (See the description of each simulation for the required value.)

1. Highlight the `frf` in the *Design Variable* list box.
2. Choose *Variables – Edit*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The Editing Design Variables form appears.



3. In the *Value (Expr)* field, type 900M (or 920M) for the value of `frf` and click *Change*.
4. Highlight `flo` in the *Table of Design Variables* list box.
5. In the *Value (Expr)* field, type 1G for the value of `flo` and then click *Change*.
6. In the Editing Design Variables form, click *OK*.

Harmonic Distortion Measurement with PSS

In this first example, a single PSS analysis determines the harmonic distortion of the mixer.

Setting Up the Simulation

1. If necessary, open the ne600p mixer circuit.
2. If necessary, specify the full path to the model file in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, set the design variables `frf` to 920M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

A critical part of this analysis is the correct use of the programmable voltage sources in the `ne600p` circuit. The RF voltage source is based on the `port` sample component. You must change the behavior of this component for each analysis.

1. Choose *Edit – Properties – Objects* in the Schematic window.

The Edit Object Properties form appears. You use this form to change the list of CDF (component description format) properties for the *rf* voltage source and modify the schematic for this simulation.

2. In the Schematic window, click on the *rf* voltage source.

The Edit Object Properties form changes to display information for the voltage source.

3. In the Edit Object Properties form, be sure that *sine* is specified as the *Source type*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed Edit Object Properties form appears like the one below:

Check this entry. →

Edit Object Properties			
<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Apply"/> <input type="button" value="Defaults"/> <input type="button" value="Previous"/> <input type="button" value="Next"/>			
Apply To		<input type="checkbox"/> only current <input type="checkbox"/> instance	
Show		<input type="checkbox"/> system <input checked="" type="checkbox"/> user <input type="checkbox"/> CDF	
<input type="button" value="Browse"/> <input type="button" value="Reset Instance Labels Display"/>			
Property	Value	Display	
Library Name	analogLib	off <input type="checkbox"/>	
Cell Name	port	off <input type="checkbox"/>	
View Name	symbol	off <input type="checkbox"/>	
Instance Name	rf	off <input type="checkbox"/>	
		<input type="button" value="Add"/> <input type="button" value="Delete"/> <input type="button" value="Modify"/>	
User Property	Master Value	Local Value	Display
lvignore	TRUE		off <input type="checkbox"/>
pwr			off <input type="checkbox"/>
CDF Parameter	Value	Display	
Resistance	50 Ohms	off <input type="checkbox"/>	
Port number	1	off <input type="checkbox"/>	
DC voltage		off <input type="checkbox"/>	
Source type	sine <input type="checkbox"/>	off <input type="checkbox"/>	
Frequency name 1	frf	off <input type="checkbox"/>	
Frequency 1	frf Hz	off <input type="checkbox"/>	
Amplitude 1 (Vpk)		off <input type="checkbox"/>	
Amplitude 1 (dBm)	prf	off <input type="checkbox"/>	
Phase for Sinusoid 1		off <input type="checkbox"/>	
Sine DC level		off <input type="checkbox"/>	
Delay time		off <input type="checkbox"/>	
Display second sinusoid	<input type="checkbox"/>	off <input type="checkbox"/>	
Display modulation params	<input type="checkbox"/>	off <input type="checkbox"/>	
Display small signal params	<input type="checkbox"/>	off <input type="checkbox"/>	
Display temperature params	<input type="checkbox"/>	off <input type="checkbox"/>	
Display noise parameters	<input type="checkbox"/>	off <input type="checkbox"/>	

Setting Up the PSS Analysis

1. Choose *Analyses* – Choose in the Simulation window.

The Choosing Analyses form appears.

2. In the Choosing Analyses form, click on *pss*.
3. In the *Fundamental Tones* list box, be sure the *Auto Calculate* button is highlighted.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The *Beat Frequency* is now displayed. The 40 M value is the minimum period for which both RF at 920 MHz and the LO at 1 GHz are periodic (or for which RF and LO are integer multiples of the fundamental frequency).

The *Beat Frequency* button is highlighted by default.

In the Choosing Analyses form, perform the following steps:

1. In the *Output harmonics* cyclic field, choose *Number of harmonics* for the *Output harmonics* choice and type 30 for the *number of harmonics* in the field.

This entry expands the plotted frequency range to include the areas of interest around both 40 MHz and 1 GHz. (30 x 40 MHz equals 1.2 GHz, or 0 to 1.2 GHz, and includes all important frequencies.)

2. Highlight *conservative* for the *Accuracy Defaults (errpreset)* setting.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The correctly filled out form appears below.

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
2	flo	flo	10	Moderate	lo
1	frf	frf	920M	Moderate	rf

Moderate

Clear/Add Delete Update From Schematic

Beat Frequency 40M Auto Calculate

Beat Period

Output harmonics

Number of harmonics 30

Accuracy Defaults (errpreset)
 conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep

3. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. In the Schematic window, choose *Design – Check and Save*.
2. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

3. Look in the CIW for a message that says the simulation completed successfully.

Calculating Harmonic Distortion

- Choose *Results – Direct Plot – PSS* in the Simulation window.

The PSS Results form appears.

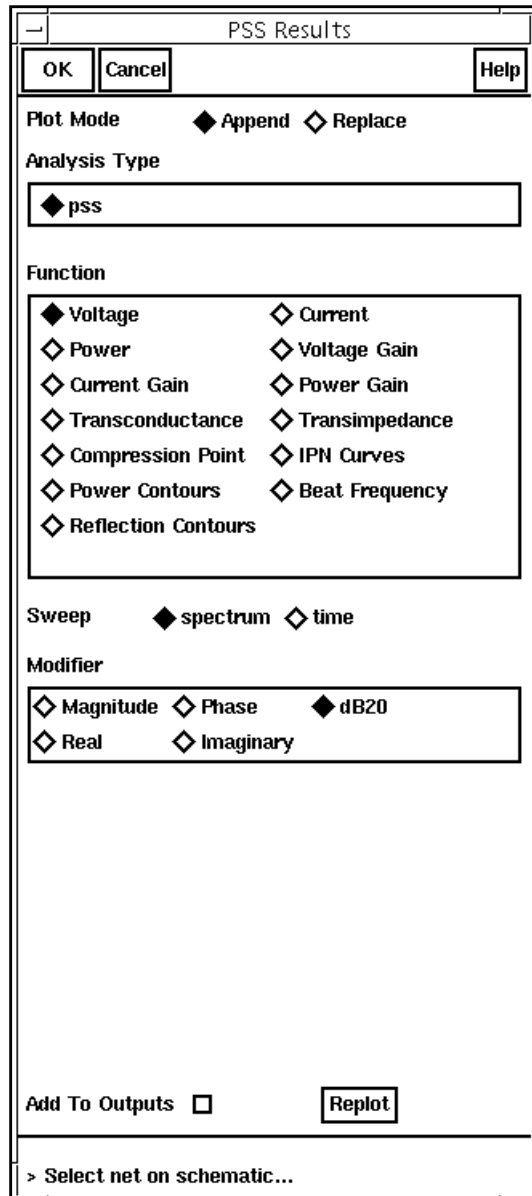
In the PSS Results form, do the following:

1. Highlight *Append* for *Plot Mode*.
2. Highlight *pss* for *Analysis Type*.
3. Highlight *Voltage* for *Function*.
4. Highlight *spectrum* for *Sweep*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

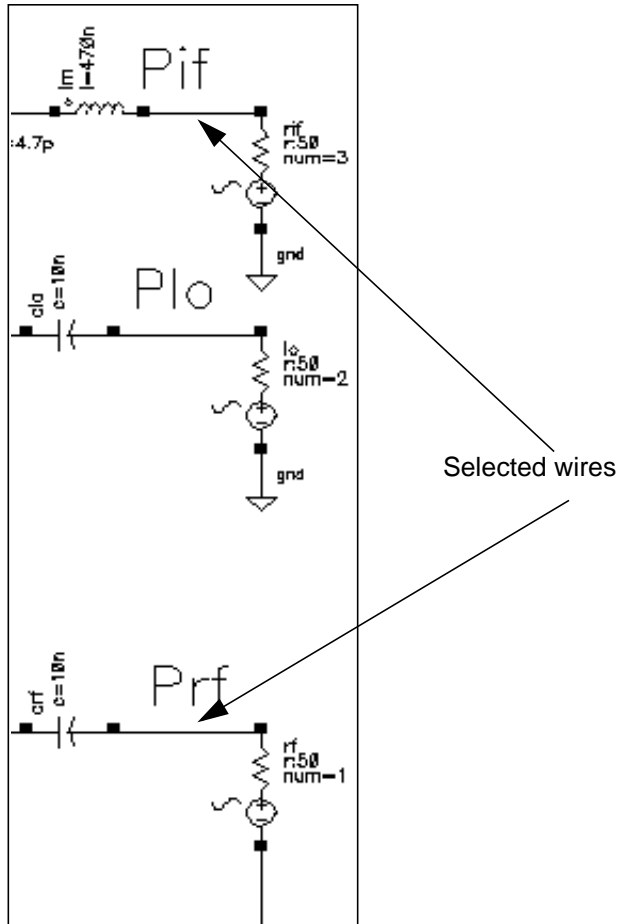
5. Highlight *db20* for *Modifier*.



Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

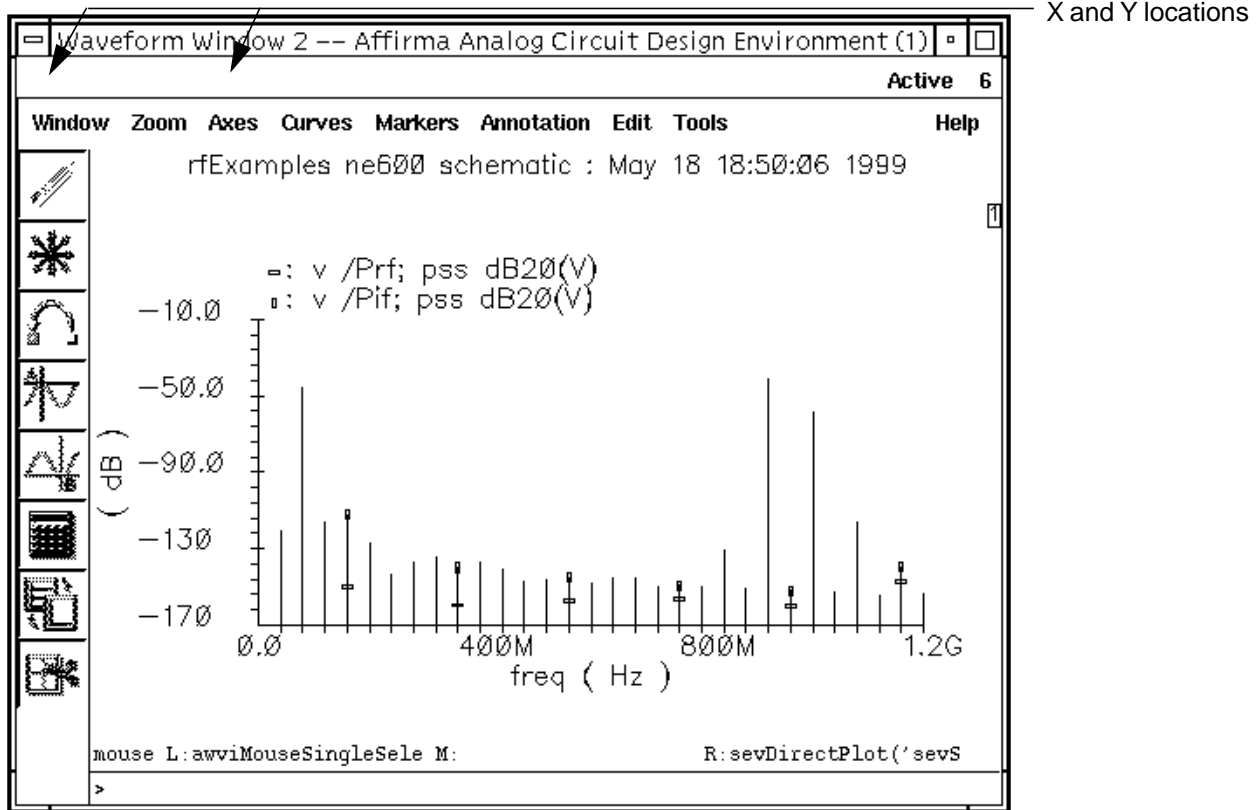
- To plot the voltage against frequency, click on the *Pif* and *Prf* wires in the Schematic Window.



Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The Waveform Window display appears like the one below.



7. Measure V_{if} at 80, 160, and 240 MHz by placing the cursor at the top of the appropriate lines in the plot. The X and Y values appear at the top left corner of the Waveform window.
8. Calculate the total harmonic distortion (THD) with the following formula:

$$THD = dB10((h4)^2 + (h6)^2) - dB20(h2)$$

where

$h2$ is 80 MHz

$h4$ is 160 MHz

$h6$ is 240 MHz.

The value of THD is -61.0 dB.

Noise Figure Measurement with PSS and Pnoise

You can combine a PSS analysis with a small-signal Pnoise analysis to determine the noise figure for the [ne600p circuit](#).

Pnoise analysis calculates the total noise at the output of the circuit. The equation for the Noise Figure is given in the [SpectreRF Theory](#) document. The SpectreRF Pnoise analysis computes the single sideband noise figure (-1 in this case). The total noise can vary with the number of harmonics you choose because each harmonic contributes a noise component.

Setting Up the Simulation

1. If necessary, [open the ne600p mixer circuit](#).
2. If necessary, [specify the full path to the model file](#) in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, [set the design variables](#) `frf` to 920M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing The Schematic

Modify the schematic to be sure the PSS analysis is the response of the ne600p mixer to *only* the LO signal.

1. Choose *Edit – Properties – Objects* in the Schematic window.
The Edit Object Properties form appears. You use this form to change the list of CDF properties and modify the schematic for this simulation.
2. In the Schematic Window, click on the *rf* voltage source.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

3. In the Edit Object Properties form, choose *dc* as the *Source type*, if necessary, and click *OK*.

Change this entry. →

CDF Parameter	Value
Resistance	50 Ohms
Port number	1
DC voltage	
Source type	dc <input type="checkbox"/>
Display small signal params	<input type="checkbox"/>
Display temperature params	<input type="checkbox"/>
Display noise parameters	<input type="checkbox"/>
Multiplier	

4. In the Schematic window, choose *Design – Check and Save*.

Setting up the PSS and Pnoise Analyses

- In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

Setting Up the PSS Analysis

1. In the Choosing Analyses form, click on *pss* for the *Analysis*.
2. At the lower right corner of the Fundamental Tones section, be sure the *Auto Calculate* button is highlighted.

The *Beat Frequency* is now displayed as 1G. The *Beat Frequency* button is highlighted by default.

3. In the *Output harmonics* cyclic field, choose *Number of harmonics* and type 0 in the field.
Pnoise requires that you set this value to 0 to determine the circuit's response to LO only.
4. Highlight *conservative* for the *Accuracy Defaults (errpreset)* setting.

Setting Up the Pnoise Analysis

1. At the top of the Choosing Analyses form, highlight *pnoise*.

The Choosing Analyses form changes to let you specify data for a *Pnoise* analysis.

2. In the *Frequency Sweep Range (Hz)* cyclic field, choose *Start-Stop*.
3. Type 1K and 2G as the *Start* and *Stop* values, respectively in the fields.

This frequency range covers the frequencies of interest, but avoids the value 0. You cannot include the value 0 in a logarithmic sweep.

4. In the *Sweep Type* cyclic field, choose *Logarithmic* for the sweep type and highlight the *Points Per Decade* button. Type 10 for the *Points Per Decade* value in the field.
5. In the *Sidebands* cyclic field, choose *Maximum sideband* and type 30 for the number of sidebands.

With this setting, you specify that 30 sidebands contribute noise to the output. The value is taken from the first example (*PSS analysis only*) in this chapter.

6. In the *Output Voltage option* cyclic field, choose *voltage* for the *Output* value.
7. Highlight the *Positive Output Node Select* button. Then click on the appropriate wire in the Schematic window to choose *Pif*.

/Pif appears in the *Positive Output Node* field.

8. Leave the *Negative Output Node* field empty.

This field defaults to */gnd!* You can set the *Negative Output Node* to a different value by clicking on the *Negative Output Node Select* button and then choosing the output node in the schematic.

9. In the *Input Source* cyclic field, choose *voltage*.
10. Click on the *Input Voltage Source Select* button. Then click on the appropriate component in the Schematic window to choose *rf*.

/rf appears in the *Input Voltage Source* field.

11. Type *-1* in the *Reference Sideband* field.

In this field, you specify the difference between the input and output frequencies in the whole *frf*. The *Reference Sideband* must be *-1* because this is a down converter. Other possible choices are 0 and *+1*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed *Periodic Noise Analysis* section of the Choosing Analyses form appears like the one below.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Noise Analysis

PSS Beat Frequency (Hz)

Sweeptype

Frequency Sweep Range (Hz)

Start-Stop Start Stop

Sweep Type

Logarithmic Points Per Decade
 Number of Steps

Add Specific Points

Sidebands

Maximum sideband

Output

voltage Positive Output Node
 Negative Output Node

Input Source

voltage Input Voltage Source

Reference Side-Band

Enabled

12. In the Choosing Analyses form, click *OK* or *Apply*.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.
The output log file appears and displays information about the simulation as it runs.
2. Look in the CIW for a message that says the simulation completed successfully.

Plotting the Noise Figure

- To open the PSS Results form, choose *Results – Direct Plot – PSS* in the Simulation window.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pnoise* for *Analysis Type*.
3. Highlight *Noise Figure* for *Function*.

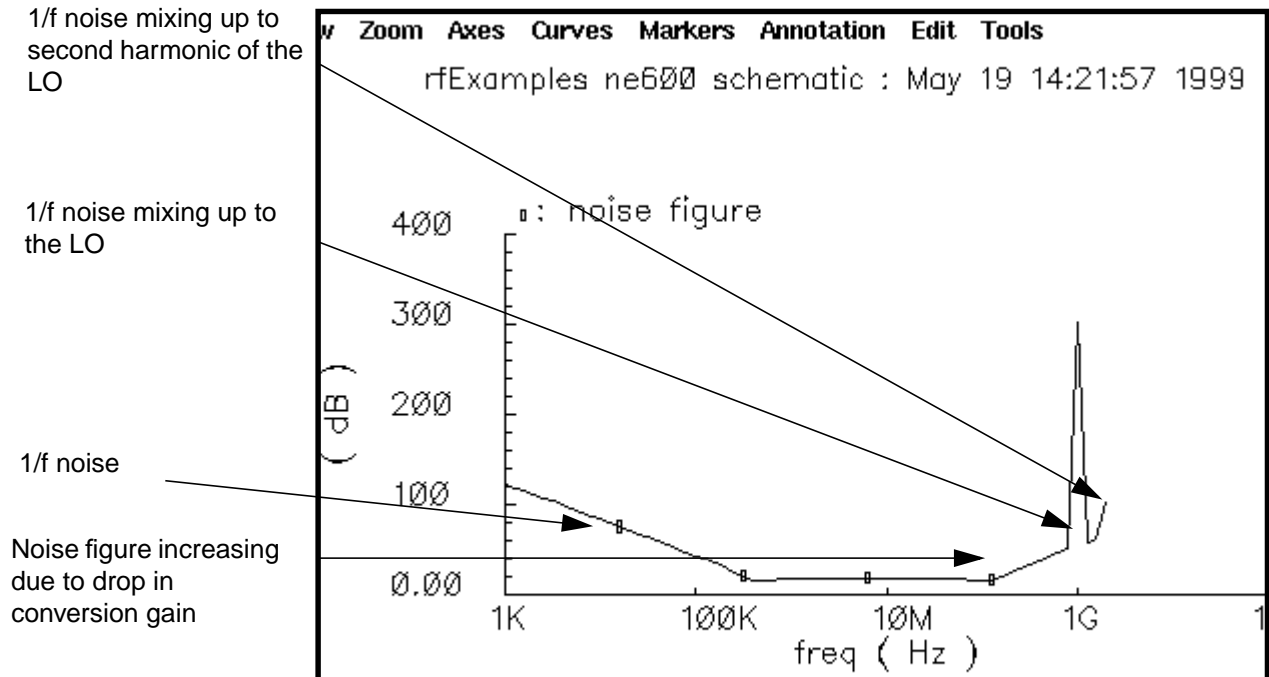
The completed PSS Results form appears like the one below.

The screenshot shows the 'PSS Results' dialog box. At the top, there are buttons for 'OK', 'Cancel', and 'Help'. Below that, the 'Plot Mode' section has radio buttons for 'Append' and 'Replace', with 'Replace' selected. The 'Analysis Type' section has radio buttons for 'pss' and 'pnoise', with 'pnoise' selected. The 'Function' section has a list of options: 'Output Noise', 'Input Noise', 'Noise Figure', 'Noise Factor', 'Transfer Function', 'Phase Noise', and 'Pss Beat Frequency'. 'Noise Figure' is selected. At the bottom left, there is a checkbox for 'Add To Outputs' which is unchecked. At the bottom right, there is a 'Plot' button.

4. Click on *Plot* in the PSS Results form.

The plot appears in the graph.

The Waveform window displays the noise figure:



- To determine the noise figure at different frequencies, move the cursor along the noise figure curve in the Waveform window. In the above plot, the noise figure is 16.26 dB at 80 MHz.

Noise Figure Measurement and Periodic S-Parameter Plots with PSS and PSP

The periodic SP (PSP) analysis computes scattering and noise parameters for n-port circuits that exhibit frequency translation. Such circuits include mixers, switched-capacitor filters, samplers, phase-locked loops, and similar circuits.

In this example you follow a large-signal PSS analysis with a small-signal PSP analysis to analyze noise folding terms induced by the RF input for the [ne600p mixer circuit](#).

This PSP analysis computes

- Periodic S-parameters
- Periodic noise correlation matrices

- Noise figure
- Equivalent noise parameters

The PSP analysis is a small-signal analysis like the SP analysis, except that, as is true for PAC and PXF analysis, the circuit is first linearized about a periodically varying operating point as opposed to a simple DC operating point. Linearizing the circuit about a periodically time-varying operating point allows for the computation of S-parameters between circuit ports that convert signals from one frequency band to another.

The PSP analysis can also calculate noise parameters in frequency-converting circuits. PSP computes noise figure (both single-sideband and double-sideband), input referred noise, equivalent noise parameters, and noise correlation matrices. The noise features of the PSP analysis include noise folding effects due to the periodically time-varying nature of the circuit. This is also true for the Pnoise analysis, but not for the SP analysis.

Setting Up the Simulation

1. If necessary, open the ne600p mixer circuit.
2. If necessary, specify the full path to the model file in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, set the design variables `frf` to 900M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

The rf source used in the ne600p mixer circuit has a *Source type* property that you can set to either *sine* or *dc* depending on your application. Suppose that your RF input signal is at 900 MHz, the LO is at 1 GHz, and the IF is at 100 MHz. Before you perform a PSP analysis, you must first perform a PSS analysis.

- In many cases for small-signal analysis, it is sufficient to treat the RF input as a small-signal. For example, most gain measurements are performed under small-signal conditions in which case you set the *Source type* to *dc*.
- When it is important to analyze additional noise folding terms induced by the rf input, you treat the rf source as a large signal in which case you set the *Source type* to *sine*.

See [Appendix L, “Using PSP and Pnoise Analyses”](#) for more information.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

In this example, you assume the rf source is a large signal so you edit the rf port on the schematic and set the *Source type* parameter to *sine*.

1. Choose *Edit – Properties – Objects* in the Schematic window.

The Edit Object Properties form appears. You use this form to change the list of CDF properties and modify the schematic for this simulation.

2. In the Schematic window, click on the *rf* voltage source.

The Edit Object Properties form changes to display information for the voltage source.

3. In the Edit Object Properties form, choose *sine* as the *Source type*, if necessary, and click *OK*.

CDF Parameter	Value
Resistance	50 Ohms
Port number	1
DC voltage	
Source type	sine
Frequency name 1	frf
Frequency 1	frf Hz
Amplitude 1 (Vpk)	
Amplitude 1 (dBm)	prf
Phase for Sinusoid 1	
Sine DC level	
Delay time	
Display second sinusoid	<input type="checkbox"/>

4. In the Schematic window, choose *Design – Check and Save*.

Setting up the PSS and PSP Analyses

Use the Choosing Analyses form, to set up two analyses, first PSS then PSP.

- ▶ Choose *Analyses – Choose* in the Simulation window.

The Choosing Analyses form appears.

Setting Up the PSS Analysis

1. At the top of the Choosing Analyses form, highlight *pss*.
2. In the *Fundamental Tones* list box, be sure the *Auto Calculate* button is highlighted.

The *Beat Frequency* is now displayed. (The *Beat Frequency* button is highlighted by default.) The 100 M value for beat frequency is the minimum period for which both RF at 900 MHz and LO at 1 GHz are periodic (or for which RF and LO are integer multiples of the fundamental frequency).

3. In the *Output harmonics* cyclic field, choose *Number of harmonics*. Type 30 in the *Number of harmonics* field.

This entry expands the plotted frequency range to include the areas of interest around both 100 MHz and 1 GHz. (30×100 MHz equals 3GHz, or 0 to 3GHz, and includes all important frequencies.)

4. Highlight *moderate* for the *Accuracy Defaults (errpreset)* setting.
5. Be sure the *Enabled* button is highlighted.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The top of the correctly filled out form appears below.

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1	flo	flo	16	Moderate	lo
2	frf	frf	900M	Moderate	rf

◆ Beat Frequency

 Auto Calculate

◇ Beat Period

Output harmonics

Accuracy Defaults (empreset)

conservative
 moderate
 liberal

Setting Up the PSP Analysis

1. In the *Analysis* section at the top of the Choosing Analyses form, highlight *psp*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The Choosing Analyses form changes to let you specify data for the *PSP* analysis.

2. In the *Sweeptype* cyclic field, choose *relative*.

The *Sweeptype* parameter controls the way that frequencies are swept in the PSP analysis. When you specify a *relative* sweep, the sweep is relative to the analysis harmonics (not the PSS fundamental frequency).

For PSP analysis, the frequencies of the input and response are usually different. This is an important difference between the PSP and SP analyses. Because the PSP computation involves inputs and outputs at frequencies that are relative to multiple harmonics, the *freqaxis* and *sweeptype* parameters behave somewhat differently in PSP analysis than they behave in PAC and PXF analyses.

3. In the *Frequency Sweep Range (Hz)* cyclic field, choose *Start-Stop* and type $-20M$ and $30M$, respectively, in the *Start* and *Stop* fields.

The PSP analysis is performed from 20 MHz below the RF center frequency to 30 MHz above it. This frequency range accounts for inputs on the rf port in the range of -920 MHz to -870 MHz. Noise parameters (such as noise figure) are computed in a 50 MHz band around the frequency specified by the output harmonic.

4. Leave the *Sweep Type* cyclic field set to *Automatic*.
5. Highlight *Select Ports* to select the input and output ports on the schematic.

The input port is the rf port and the output port is the rif port.

6. Click the *Input Port (#1) Select* button. Then, in the Schematic window, click on the appropriate port to choose */rf*.

/rf appears in the *Input Port (#1)* field.

7. Click the *Output Port (#2) Select* button. Then, in the Schematic window, click on the appropriate port to choose */rif*.

/rif appears in the *Output Port (#2)* field.

8. Leave the *Aux Port (#3)* field empty.

This field is used to include in the analysis an additional port/harmonic pair.

9. Type -9 in the *Input Port (#1) Harmonic* field. Type 1 in the *Output Port (#2) Harmonic* field.

With the 1 GHz LO, small-signal inputs around 900 MHz (harmonics ± 9 of the PSS fundamental) appear at around 100 MHz (or harmonics ∓ 1 of the fundamental). You select harmonic 1 as the output harmonic, and harmonic -9 as the input harmonic

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

because a single complex-exponential input ($e^{i\omega_s t}$) on the lower side of 900 MHz (harmonic -9) appears as the upper sideband of harmonic 1 at around 100 MHz.

Harmonics -9 and 1 are separated by 10 fundamental periods, which corresponds to the LO frequency of 1 GHz.

10. Below *Do Noise* near the bottom of the form, highlight *Yes* to calculate noise parameters as part of the PSP analysis.

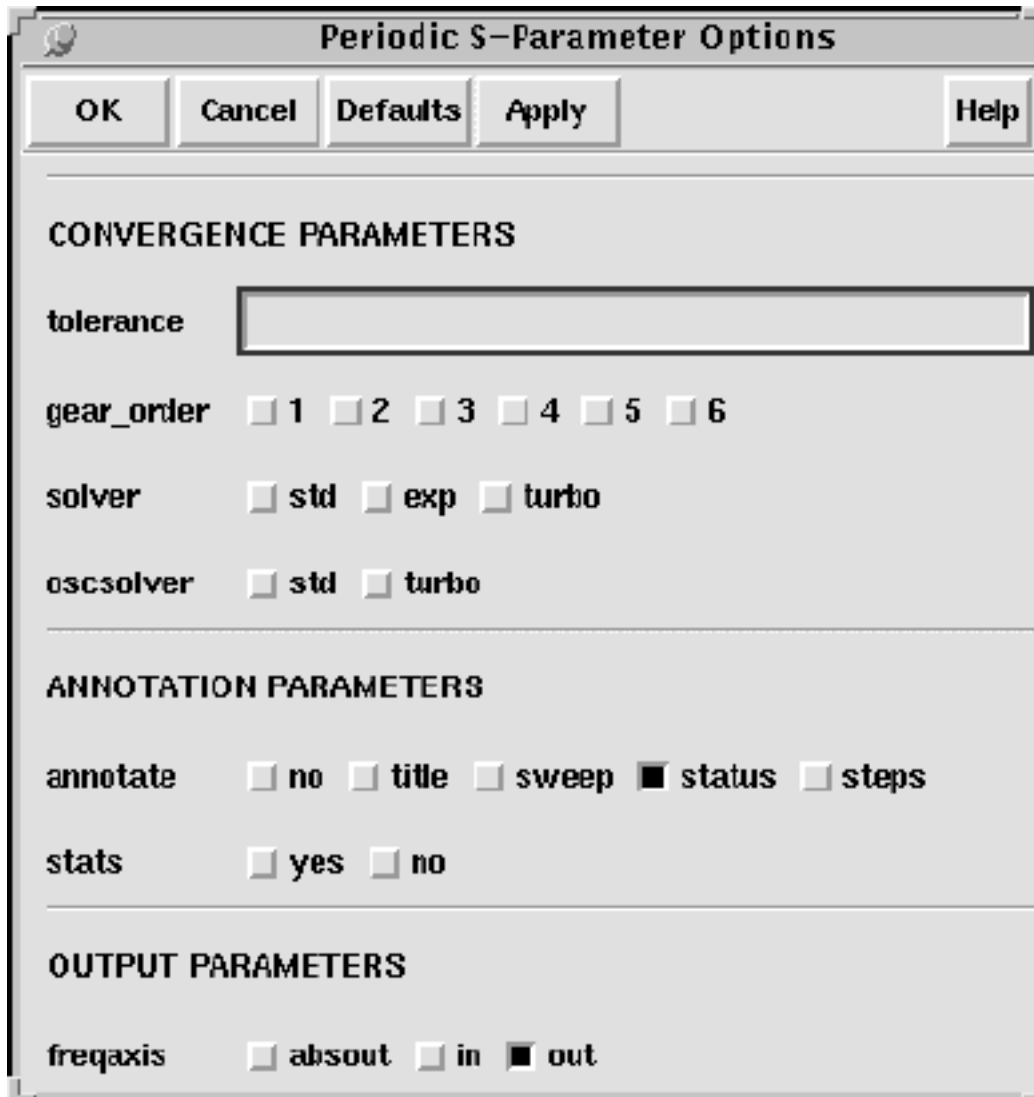
11. Type 50 in the *Maxsideband* field.

Setting *Maxsideband* to 50 accounts for noise folding from up to 5 GHz in frequency.

All noise computations in PSP analysis involve noise folding effects. The *maxsideband* parameter specifies the maximum sideband to include for summing noise contributions either up-converted or down-converted to the output at the frequency of interest.

12. Click the *Options* button to display the PSP Options form.

13. Set the *freqaxis* parameter to *out*.



Periodic S-Parameter Options

OK Cancel Defaults Apply Help

CONVERGENCE PARAMETERS

tolerance

gear_order 1 2 3 4 5 6

solver std exp turbo

oscsolver std turbo

ANNOTATION PARAMETERS

annotate no title sweep status steps

stats yes no

OUTPUT PARAMETERS

freqaxis absout in out

Selecting *out* changes the scale used for the output axis so that it runs from 80 MHz to 130 MHz.

The *freqaxis* parameter specifies whether the results should be output versus the absolute value of the input frequency (*absout*), the input frequency (*in*), or the output frequency (*out*).

14. Click *OK* in the Options form.
15. Be sure the *Enabled* button is highlighted in the Choosing Analysis form.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed Periodic S-Parameters Analysis section of the Choosing Analyses form appears like the one below.

Periodic S-Parameter Analysis

Sweeptype

Frequency Sweep Range(Hz)

Start-Stop

Sweep Type

Add Specific Points

Select Ports

Input Port(#1)	<input type="text" value="/rif"/>	<input type="button" value="Select"/>	Harmonic	<input type="text" value="-9"/>
Output Port(#2)	<input type="text" value="/rif"/>	<input type="button" value="Select"/>	Harmonic	<input type="text" value="1"/>
Aux. Port(#3)	<input type="text"/>	<input type="button" value="Select"/>	Harmonic	<input type="text"/>

Additional Harmonics

Do Noise

yes

no

Maximum Sideband

Enabled

16. In the Choosing Analyses form, click *OK*.

Running the Simulation

You can now run the simulation. In addition to computing the periodic S-parameters, the periodic noise correlation matrixes are also computed, as are noise figure and the equivalent noise parameters.

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.
The output log file appears and displays information about the simulation as it runs.
2. Look in the CIW for a message that says the simulation completed successfully.

Plotting the Noise Figure

- To open the PSS Results form, choose *Results – Direct Plot – PSS* in the Simulation window.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *psp* for *Analysis*.
3. Highlight *NF* for *Function* to display the Noise Figure.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

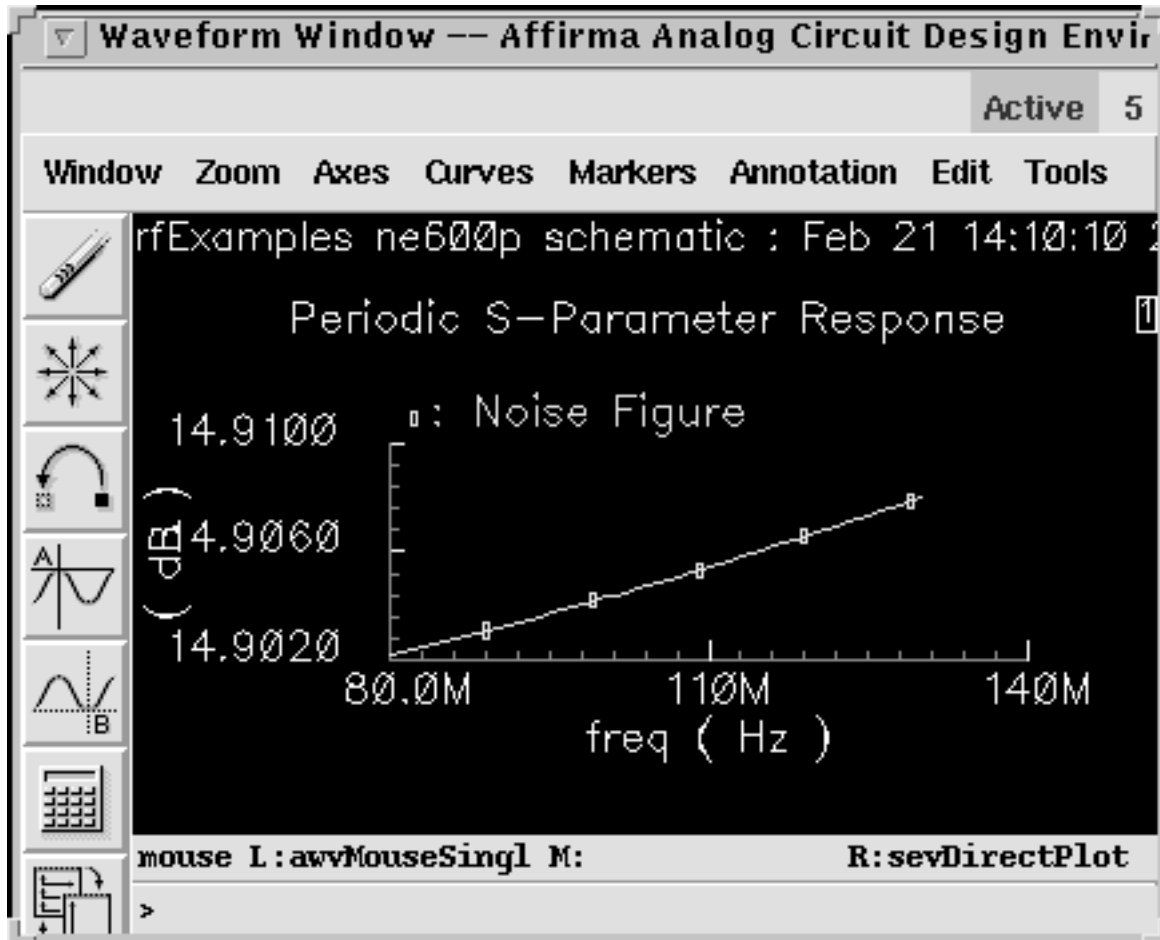
The completed PSP Results form appears like the one below.

The screenshot shows a dialog box titled "PSS Results". At the top, there are three buttons: "OK", "Cancel", and "Help". Below the buttons, the "Plot Mode" is set to "Append" (indicated by a diamond) and "Replace" (indicated by a filled diamond). The "Analysis" section has a dropdown menu showing "pss" (diamond) and "psp" (filled diamond). The "Function" section contains a grid of radio buttons for various metrics: SP (diamond), NF (filled diamond), F (diamond), NFmin (diamond), Fieee (diamond), NFieee (diamond), Fdsb (diamond), NFdsb (diamond), RN (diamond), IRN (diamond), NNR (diamond), and GAIN (diamond). The "Description: Noise Figure" is displayed. The "Plot Type" section has a dropdown menu showing "Rectangular" (filled diamond). At the bottom, there is an "Add To Outputs" checkbox (unchecked) and a "Plot" button.

4. Click *Plot* to display the Noise Figure.

The plot appears in the graph.

The Waveform window display appears like the one below:



5. To determine the noise figure at different frequencies, move the cursor along the noise figure curve.

Plotting Periodic S-Parameters

- If necessary, open the PSS Results form, choose *Results – Direct Plot – PSS* in the Simulation window.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *psp* for *Analysis*.
3. Highlight *SP* for *Function* to display periodic S-parameters.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

4. Highlight *Z-Smith* for *Plot Type* to display periodic S-parameters.
5. Click *S22* to plot output match

Recall from basic S-parameter theory that for a 2-port circuit *S22* represents the match at port 2. In this example *S22* represents the match at the if port.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed PSP Results form appears like the one below.

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis

pss psp

Function

SP NF F NFmin
 Fieeb NFieeb Fdsb NFdsb
 RN IRN NNR GAIN

Description: S-Parameter

Plot Type

Auto Rectangular Z-Smith
 Y-Smith Polar

S11 S12
S21 S22

Port 1 active harmonic is -9
Port 2 active harmonic is 1

Add To Outputs

> To plot, press Sij-button on this form...

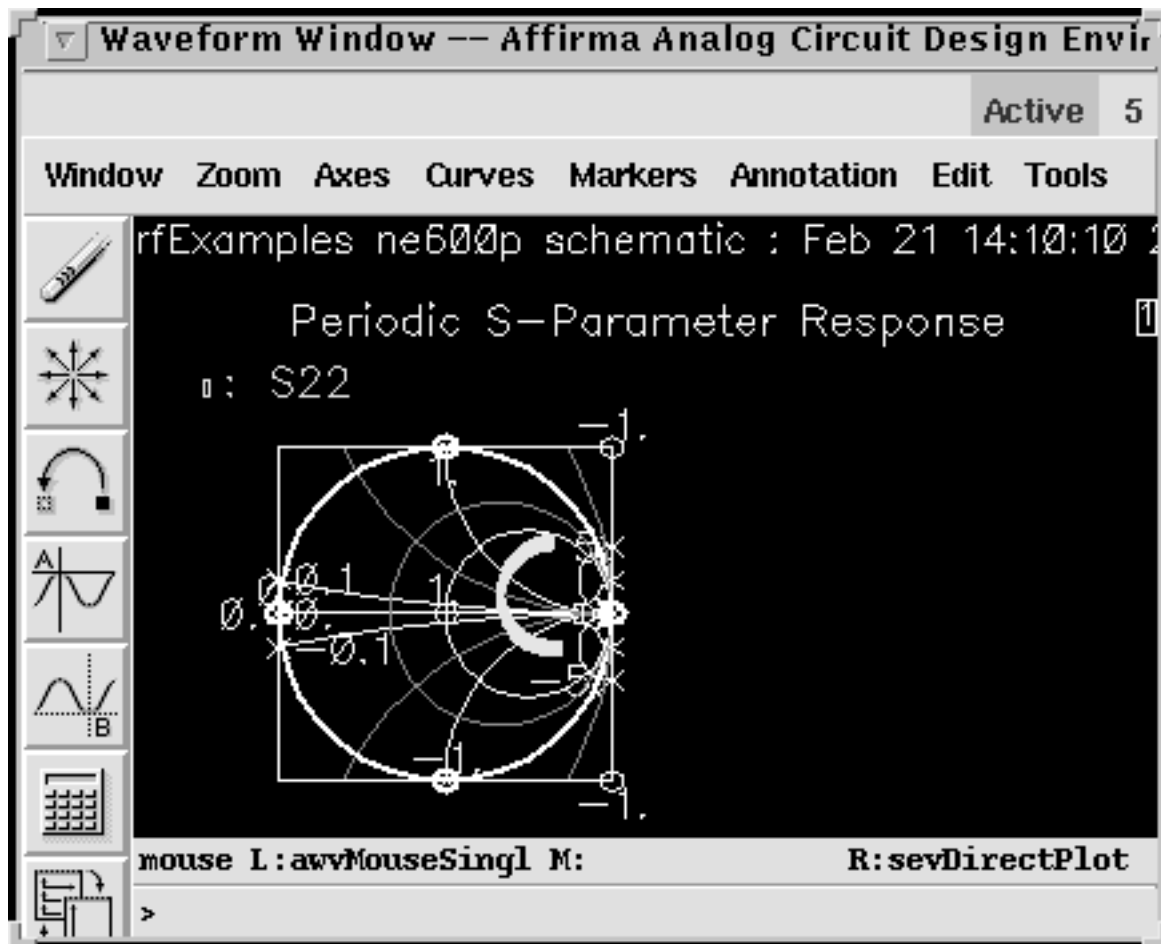
Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

- Click *Plot* to display the S-parameters.

The plot appears in the graph.

The Waveform window display appears like the one below:



At the center of the Smith chart (at 1), the match is perfect. The match is better as the curve gets closer to the center of the Smith chart. Notice that frequency changes with match. The displayed frequency and position on the Smith chart change as you move the cursor along the curve.

The frequency where the output match is the best is approximately 105.4 MHz, and the reflection coefficient at that frequency is 341.2 m at a phase of 9.017 degrees

Conversion Gain Measurement with PSS and PXF

You can combine a PSS analysis with a periodic small-signal transfer function PXF analysis to determine the conversion gain of the down converter. With changes to the plotting form, you can also calculate the power supply rejection and local oscillator feedthrough.

Setting Up the Simulation

1. If necessary, open the ne600p mixer circuit.
2. If necessary, specify the full path to the model file in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, set the design variables `frf` to 920M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

You must modify the schematic to set the RF source to a DC source to be sure the PSS analysis is the response of the `ne600p` mixer to only the LO signal.

1. Choose *Edit – Properties – Objects* in the Schematic window.

The Edit Object Properties form appears. You use this form to change the list of CDF properties and modify the schematic for this simulation.

2. In the Schematic window, click on the *rf* voltage source.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

3. In the Edit Object Properties form, choose *dc* as the *Source type*, if necessary, and click *OK*.

Change this entry. →

CDF Parameter	Value
Resistance	50 Ohms
Port number	1
DC voltage	
Source type	dc <input checked="" type="checkbox"/>
Display small signal params	<input type="checkbox"/>
Display temperature params	<input type="checkbox"/>
Display noise parameters	<input type="checkbox"/>
Multiplier	

4. In the Schematic window, choose *Design – Check and Save*.

Setting Up the PSS and PXF Analyses

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

Setting Up the PSS Analysis

1. In the Choosing Analyses form, highlight *pss* for the *Analysis*.
2. In the *Fundamental Tones* section, be sure the *Auto Calculate* button is highlighted.

The *Beat Frequency* is now displayed as 1G. LO is the only time-varying signal in the circuit, so it becomes the fundamental frequency. The *Beat Frequency* button is highlighted by default.

3. In the *Output Harmonics* cyclic field, choose *Number of Harmonics* and type 0 in the *Number of harmonics* field.

You must turn off harmonic generation because it is required to set up the subsequent PXF analysis.

4. Highlight *conservative* for the *Accuracy Defaults (errpreset)* setting.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed PSS analysis form looks like this.

The screenshot shows the 'Choosing Analyses' dialog box with the following configuration:

- Analysis types: sens, sp, pdisto, pss, pac, pnoise, pxf, envlp
- Periodic Steady State Analysis section:
 - Fundamental Tones table:

#	Name	Expr	Value	Signal	SrcId
2	flo	flo	1G	Moderate	lo
 - Buttons: Clear/Add, Delete, Update From Schematic
 - Beat Frequency: Beat Frequency, Beat Period, value: 1G, Auto Calculate
- Output harmonics: Number of harmonics 0
- Accuracy Defaults (errpreset): conservative, moderate, liberal
- Additional Time for Stabilization (tstab):
- Save Initial Transient Results (saveinit): no, yes
- Oscillator:
- Sweep:
- Enabled: Options...

Setting Up the PXF Analysis

1. At the top of the Choosing Analyses form, click on *pxf*.

The Choosing Analyses form changes to let you specify data for the PXF analysis.

2. In the *Frequency Sweep Range (Hz)* cyclic field, choose *Start – Stop*, and type 1M as the *Start* value and 300M as the *Stop* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

This range specification leaves space between the plots, but still shows the dramatic gain changes in the first 100 Mhz of each plot.

3. In the *Sweep Type* cyclic field, choose *Linear* and highlight *Number of Steps*. Type 50 for the *Number of Steps*.

Larger numbers of total points increase the resolution of the plot but also require a longer simulation time.

4. In the *Sidebands* cyclic field, choose *Maximum sideband* and type 3 as the value.
5. In the *Output* section, highlight *voltage*.
6. Click on the *Positive Output Node Select* button. Then click on the appropriate wire in the Schematic window to choose *Pif*.

`/Pif` appears in the *Positive Output Node* field.

7. Leave the *Negative Output Node* field empty.

This field defaults to `/gnd!`. You can set the *Negative Output Node* to a different value by clicking on the *Negative Output Node Select* button and then choosing the output node in the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The Choosing Analyses form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic XF Analysis

PSS Beat Frequency (Hz) 1G

Sweep Type

Frequency Sweep Range (Hz)

Start-Stop Start 1M Stop 300M

Sweep Type

Linear Step Size 50
 Number of Steps

Add Specific Points

Sidebands

Maximum sideband 3

Output

voltage Positive Output Node /Pif Select
 probe Negative Output Node Select

Enabled Options...

8. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation while it runs.

2. Look in the CIW for a message that says the simulation completed successfully.

Plotting the Conversion Gain

- Choose *Results – Direct Plot – PSS* in the Simulation window.

The PSS Results form appears.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pxf* for *Analysis Type*.
3. Highlight *Voltage Gain* for *Function*.
4. Highlight *sideband* for *Sweep*.
5. Highlight *dB20* for *Modifier*.
6. Highlight *-1* and *1* for *Output Sideband*.

You must hold the `Control` key down while you highlight the second sideband.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed PSS Results form looks like this.

The screenshot shows the 'PSS Results' dialog box with the following settings:

- Plot Mode:** Append Replace
- Analysis Type:** pss pxf
- Function:** Voltage Gain Transimpedance Pss Beat Frequency
- Sweep:** spectrum sideband
- Modifier:** Magnitude Phase dB20 Real Imaginary
- Output Sideband:** A list with four entries:

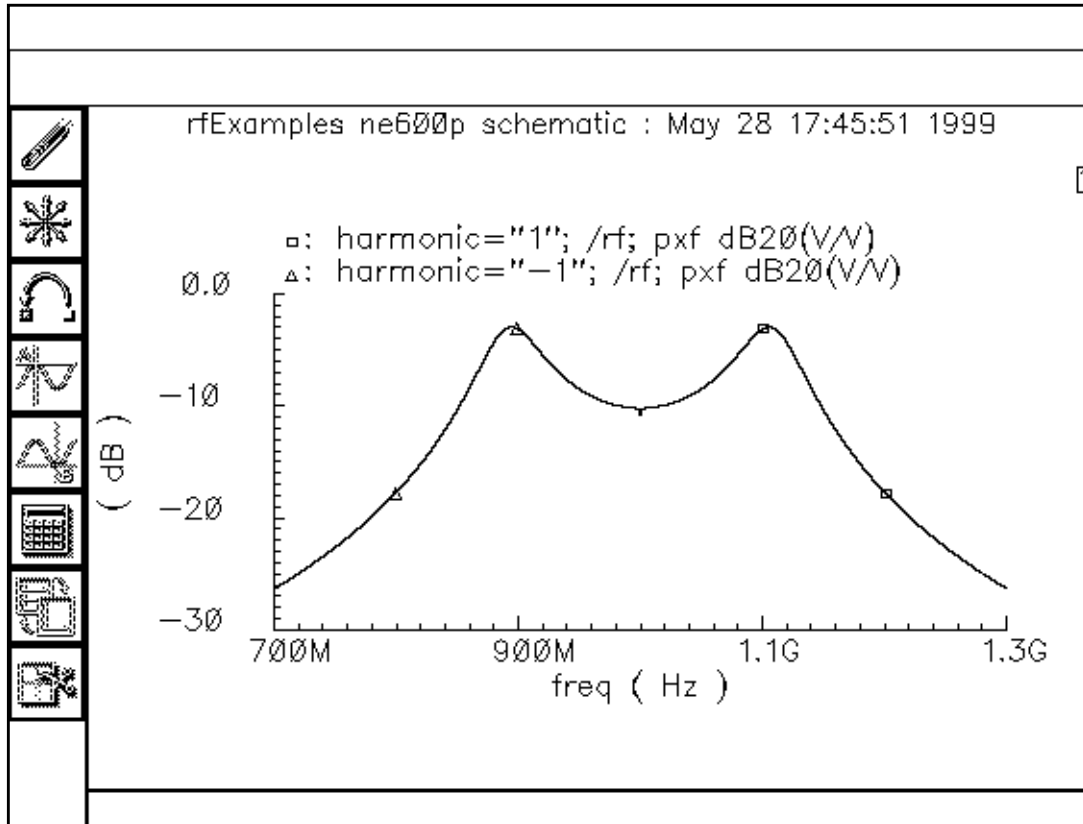
-1	700M	-	99%
0	1M	-	30%
1	1.001G	-	1%
2	2.001G	-	2.3%
- Add To Outputs:**
- Buttons:** OK, Cancel, Help, Replot
- Footer:** > Select port or voltage source on schematic ...

7. Click on the *Prf* source on the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The plot appears in the Waveform window.



- To determine the gain at different frequencies, move the cursor along the curve and use the readout at the top of the Waveform window.

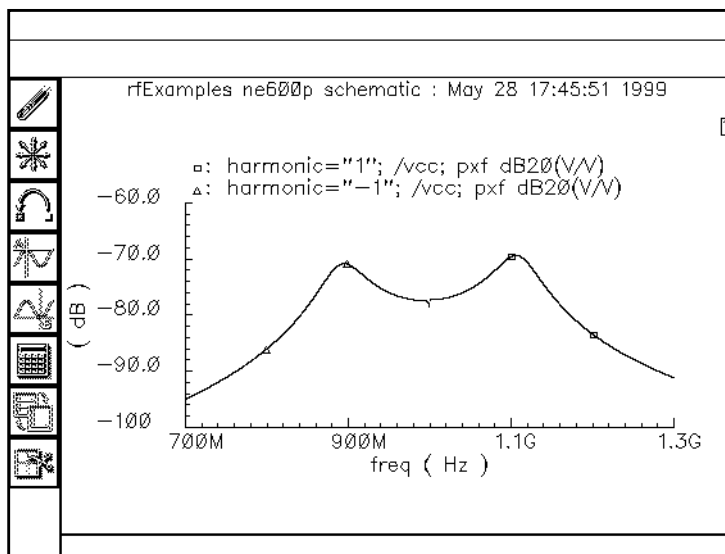
For example, the gain is -5.47 dB at 920 MHz.

Plotting the Power Supply Rejection

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pxf* for *Analysis Type*.
3. In the Schematic window, click on the *vcc* power supply.

The Waveform window appears as follows.



4. To determine the rejection at different frequencies, move the cursor along the curve and use the readout at the top of the Waveform window.

Calculating the 1 dB Compression Point with Swept PSS

In this example, a swept PSS analysis determines the 1dB compression point for the *ne600p* mixer configured as a down converter.

Setting Up the Simulation

1. If necessary, open the ne600p mixer circuit.
2. If necessary, specify the full path to the model file in the Model File Set-up form.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, set the design variables `frf` to 920M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

1. In the Schematic window, click on the *rf* voltage source.
2. Choose *Edit – Properties – Objects* in the Schematic window.
The Edit Object Properties form appears. You use this form to change the list of CDF properties and modify the schematic for this simulation.
3. In the Edit Object Properties form, specify the following and click *OK*.
Choose *sine* for *Source type*, and type `prf` in the *Amplitude (dBm)* field.
4. In the Schematic window, choose *Design – Check and Save*.

Setting Up the Swept PSS Analysis

1. In the Simulation window, choose *Analyses – Choose*.
The Choosing Analyses form appears.
2. In the Choosing Analyses form, choose *pss* for the *Analysis*.
3. In the *Fundamental Tones* list box, be sure the *Auto Calculate* button is highlighted.
The *Beat Frequency* is now displayed as 40M. Now that the RF signal is active again, the fundamental frequency of the circuit goes back to 40 MHz. The *Beat Frequency* button is highlighted by default.
4. In the *Output harmonics* cyclic field, choose *Number of harmonics* and type 2 in the *Number of harmonics* field.
Only two harmonics are required to determine the 1 dB compression point.
5. Highlight *conservative* for the *Accuracy Defaults (errpreset)* setting.
6. Highlight the *Sweep* button.
The form changes to let you specify data for the sweep.
7. In the *Sweep* cyclic field, choose *Variable*.

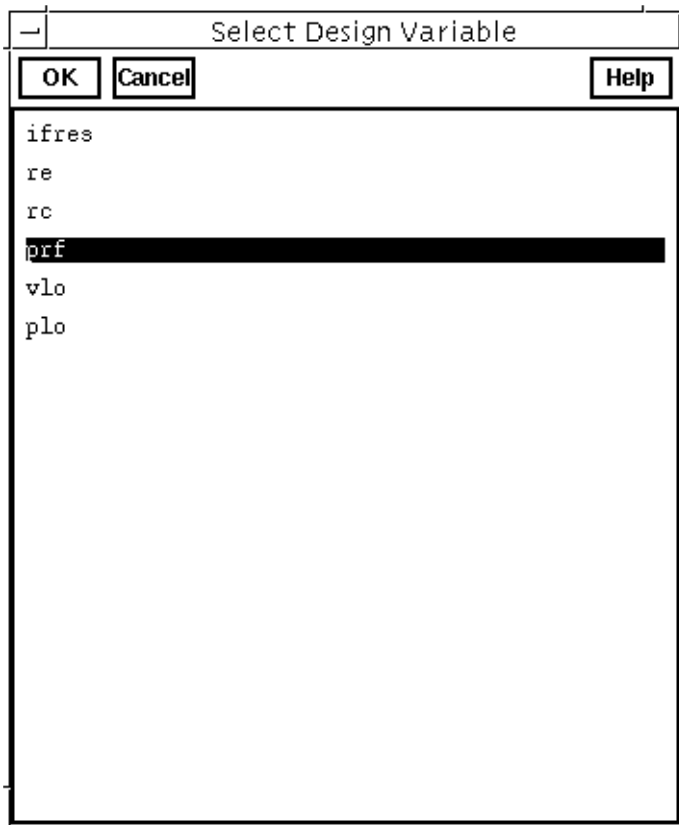
Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

8. Click on the *Select Design Variable* button.

The Select Design Variable form appears.

9. In the Select Design Variable form, highlight *prf* and click *OK*.



The *Variable Name* `prf` appears in the Choosing Analyses form. The RF input is swept.

10. Choose *Start-Stop* for the *Sweep Range*, and then type `-30` and `10` for the *Start* and *Stop* values, respectively, in the fields.

Both the signal source and the sweep are done in dBm. You learn where to set the sweep limits and how many points to include in the sweep with experience.

11. Choose *Linear* for the *Sweep Type*, and specify `10` for the *Number of Steps*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed Choosing Analyses form looks like this. (You need to use the scroll bar to see the entire form.)

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

◆ Beat Frequency Auto Calculate

◇ Beat Period

Output harmonics

Number of harmonics

Accuracy Defaults (errpreset)

conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep Frequency Variable? ◆ no ◇ yes

Variable

Variable Name

Select Design Variable

Sweep Range

◆ Start-Stop Start Stop

◇ Center-Span

Sweep Type

◆ Linear ◇ Step Size

◇ Logarithmic ◆ Number of Steps

Add Specific Points

Enabled Options...

12. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation completed successfully.

Plotting the 1 dB Compression Point

The first sideband of the IF signal is the second harmonic of the 40 MHz fundamental. To plot the 1 dB compression point, perform the following:

- Choose *Results – Direct Plot – PSS* in the Simulation window.

The PSS Results form appears.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pss* for *Analysis Type*.
3. Highlight *Compression Point* for *Function*.
4. Type 1 for *Gain Compression (dB)*.
5. Type -25 for *Extrapolation Point (dBm)*.

This value specifies the point where the ideal amplification curve intersects the output curve. You must estimate where this point is located. If you do not specify a value, the plot defaults to the minimum variable value.

6. Choose *Input Referred 1dB Compression*.
7. In the *Harmonic* list box, highlight harmonic 2 (80 MHz).

This is the second harmonic of the 40 MHz fundamental frequency, which is the IF frequency.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed PSS Results form looks like this.

PSS Results

OK Cancel Plot

Plot Mixer Analyze Replot

Analysis Type

pss

Functions

<input checked="" type="checkbox"/> Voltage	<input checked="" type="checkbox"/> Current
<input checked="" type="checkbox"/> Power	<input checked="" type="checkbox"/> Voltage Gain
<input checked="" type="checkbox"/> Current Gain	<input checked="" type="checkbox"/> Power Gain
<input checked="" type="checkbox"/> Transconductance	<input checked="" type="checkbox"/> Transimpedance
<input checked="" type="checkbox"/> Compression Point	<input checked="" type="checkbox"/> IPM Current
<input checked="" type="checkbox"/> Power Dynamic	<input checked="" type="checkbox"/> 3rd Frequency
<input checked="" type="checkbox"/> Radiation Currents	

Gain Compression (dB)

Type range: from -30 to 100

Envelope Filter (dB)

Replot, Refresh Table Compression

Test Order Summary:

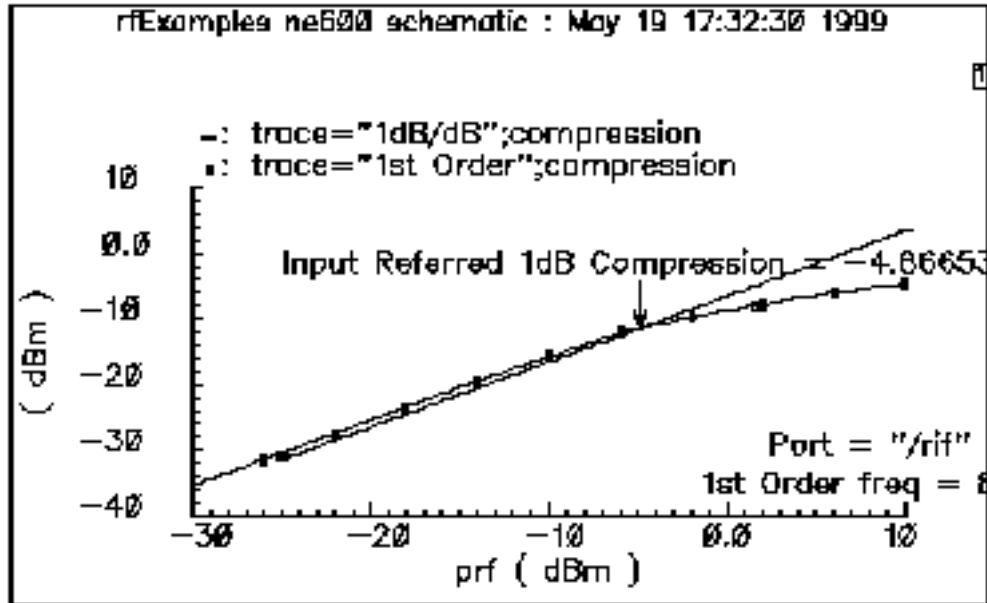
0	0
1	-60

Add To Outputs Replot

Select port as reference ...

8. In the Schematic window, click on the *vif* node.
9. In the PSS Results form, click on *Replot*.

The Waveform window display looks like this.



The SpectreRF simulation plots the signal curve and an ideal linear slope and automatically calculates a value of -4.91427 dBm for the 1 dB compression point.

If you have trouble reading the label attached to the 1 dB point, you can use the cursor to drag it to another location.

Third-Order Intercept Measurement with Swept PSS and PAC

In this example, you combine a swept PSS analysis with a Periodic AC (PAC) analysis to produce data for an IP3 plot.

This IP3 Calculation example obtains the same information as the 1 dB compression point example, but this simulation runs more quickly for two reasons

- It processes the second tone only during PAC analysis
- It considers only two of the second tone sidebands.

The swept PSS analysis for the 1 dB compression point example considered all sidebands for all signals.

Setting Up the Simulation

1. If necessary, open the ne600p mixer circuit.
2. If necessary, specify the full path to the model file in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, set the design variables `frf` to 920M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

1. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears. You use this form to modify the schematic by changing the list of CDF properties.

2. To modify the schematic for this simulation, click on the rf voltage source in the Schematic window.

In the Edit Object Properties form, do the following:

1. Highlight *Display second sinusoid* and remove any values that are set from previous analyses.
2. Choose *sine* for *Source type*.
3. Type `frf` for *Frequency*.
4. Type `prf` in the *Amplitude (dBm)* field.
5. Highlight *Display small signal params*.

The form changes to let you specify small signal parameters.

6. Type `prf` for the *PAC Magnitude (dBm)* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed Edit Object Properties form appears like the one below.

CDF Parameter	Value	Display
Resistance	<input type="text" value="50 Ohms"/>	<input type="checkbox"/> off
Port number	<input type="text" value="1"/>	<input type="checkbox"/> off
DC voltage	<input type="text"/>	<input type="checkbox"/> off
Source type	<input type="text" value="sine"/>	<input type="checkbox"/> off
Frequency name 1	<input type="text" value="frf"/>	<input type="checkbox"/> off
Frequency 1	<input type="text" value="frf Hz"/>	<input type="checkbox"/> off
Amplitude 1 (Vpk)	<input type="text"/>	<input type="checkbox"/> off
Amplitude 1 (dBm)	<input type="text" value="prf"/>	<input type="checkbox"/> off
Phase for Sinusoid 1	<input type="text"/>	<input type="checkbox"/> off
Sine DC level	<input type="text"/>	<input type="checkbox"/> off
Delay time	<input type="text"/>	<input type="checkbox"/> off
Display second sinusoid	<input type="checkbox"/>	<input type="checkbox"/> off
Display modulation params	<input type="checkbox"/>	<input type="checkbox"/> off
Display small signal params	<input checked="" type="checkbox"/>	<input type="checkbox"/> off
PAC Magnitude	<input type="text"/>	<input type="checkbox"/> off
PAC Magnitude (dBm)	<input type="text" value="prf"/>	<input type="checkbox"/> off
PAC phase	<input type="text"/>	<input type="checkbox"/> off
AC Magnitude	<input type="text"/>	<input type="checkbox"/> off
AC phase	<input type="text"/>	<input type="checkbox"/> off
XF Magnitude	<input type="text"/>	<input type="checkbox"/> off
Display temperature params	<input type="checkbox"/>	<input type="checkbox"/> off
Display noise parameters	<input type="checkbox"/>	<input type="checkbox"/> off
Multiplier	<input type="text"/>	<input type="checkbox"/> off

7. In the Edit Object Properties form, click *OK*.

This simulation uses dBm values rather than magnitude.

8. In the Schematic window, choose *Design – Check and Save*.

Setting Up the PSS and PAC Analyses

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

Setting Up the PSS Analysis

1. In the Choosing Analyses form, choose *pss* for the *Analysis*.
2. In the *Fundamental Tones* section, be sure the *Auto Calculate* button is highlighted.

The value 40M is specified as the *Beat Frequency*. The PAC analysis is responsible for the two tones, and the PSS analysis is now a single signal analysis. Consequently, the fundamental frequency is set to the original 40 MHz. The *Beat Frequency* button is highlighted by default.

3. In the *Output harmonics* cyclic field, choose *Number of harmonics* and type 2 in the field.
4. Highlight *conservative* for the *Accuracy Defaults (errpreset)* setting.
5. Highlight the *Sweep* button.

The form changes to let you specify data for the sweep.

6. In the *Sweep* cyclic field, choose *Variable*.
7. Click on the *Select Design Variable* button.

The Select Design Variable form appears.

8. Highlight *prf* in this form and click *OK*.
9. Choose *Start-Stop* for the *Sweep Range* value, and then type -25 and 5 for the *Start* and *Stop* values, respectively.

The sweep is skewed downward for a down converter.

10. Choose *Linear* for the *Sweep Type*, and specify 5 for the *Step Size*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed swept PSS section of the Choosing Analyses form looks like this. (You must scroll to see the whole form.)

The screenshot shows a dialog box with the following sections:

- Buttons:** OK, Cancel, Defaults, Apply, Help.
- Clear/Add, Delete, Update From Schematic** buttons.
- Beat Frequency:** Beat Frequency, Beat Period. Value: 40M. Auto Calculate.
- Output harmonics:** Number of harmonics: .
- Accuracy Defaults (errpreset):** conservative, moderate, liberal. Additional Time for Stabilization (tstab): . Save Initial Transient Results (saveinit): no, yes.
- Oscillator:** Oscillator.
- Sweep:** Sweep. Frequency Variable? no, yes. Variable: . Variable Name: . .
- Sweep Range:** Start-Stop, Center-Span. Start: . Stop: .
- Sweep Type:** Linear, Logarithmic. Step Size, Number of Steps. Step Size: .
- Add Specific Points:** Add Specific Points.

Setting Up the PAC Analysis

1. At the top of the Choosing Analyses form, highlight *pac*.

The Choosing Analyses form changes to let you specify data for a *pac* analysis.

2. Highlight the *Enabled* button.
3. Type 921M for the *Single-Point [] Frequency (Hz)* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

4. In the *Sidebands* cyclic field, choose *Array of indices* and type -21 and -25 for the values.

Given a fundamental tone of 40 MHz, the LO at 1 GHz, and two RF tones at 920 MHz and 921 MHz, the sidebands of -25 and -21 represent respectively the first-order harmonic of the IF output at 79 MHz ($921 - 25 \cdot 40 = 79$) and the third-order harmonic at 81 MHz ($921 - 21 \cdot 40 = 81$).

The completed Periodic AC Analysis section of the Choosing Analyses form looks like this.

The screenshot shows the 'Choosing Analyses' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. Below these are several analysis options, each with a diamond-shaped selection icon: 'tran', 'dc', 'ac', 'noise', 'xf', 'sens', 'sp', 'pdisto', 'pss', 'pac' (which is selected with a filled diamond), 'pnoise', 'pxf', and 'envlp'. The 'Periodic AC Analysis' section contains a text box for 'PSS Beat Frequency (Hz)' with the value '40M'. Below this is a 'Sweeptype' dropdown menu set to 'Single-Point []' and a 'Frequency Sweep Range (Hz)' section with a 'Freq' text box containing '921M'. A note states: 'Because the sweep section of the PSS analysis is active, only a single point for this analysis is currently supported.' The 'Sidebands' section has a dropdown menu set to 'Array of indices', a 'Currently active indices' text box, and an 'Additional indices' text box containing '-21 -25'. At the bottom left, there is an 'Enabled' checkbox which is checked, and at the bottom right, there is an 'Options...' button.

5. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

Note: This example compares two signals only 1MHz apart. An analysis of two signals so close together would have taken much longer with the previous analysis.

The output log file appears and displays information about the simulation as it runs.

2. Check the output log file to be sure the simulation is completed successfully.

Plotting the IP3 Curve

- In the Simulation window, choose *Results – Direct Plot – PSS*.

The PSS Results form appears.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pac* for *Analysis Type*.
3. Highlight *IPN Curves* for *Function*.
4. Choose *Variable Sweep (“prf”)* for *Circuit Input Power*.
5. Type *-15* for *Extrapolation Point (dBm)*.

This value is the intercept point for the ideal amplification extrapolation. If you do not specify a value, the plot defaults to the minimum variable value.

6. Choose *Input Referred IP3*.
7. Highlight *-21 81M* in the *3rd Order Harmonic* list box.
8. Highlight *-25 79M* in the *1st Order Harmonic* list box.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed form looks like this.

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss pac

Function

Voltage Current

IPN Curves Pss Beat Frequency

Circuit Input Power Single Point Variable Sweep ("prf")

"prf" ranges from -25 to 5

Extrapolation Point (dBm)

Input Referred IP3 Order

3rd Order sideband

-25	79M
-21	81M
0	921M

1st Order sideband

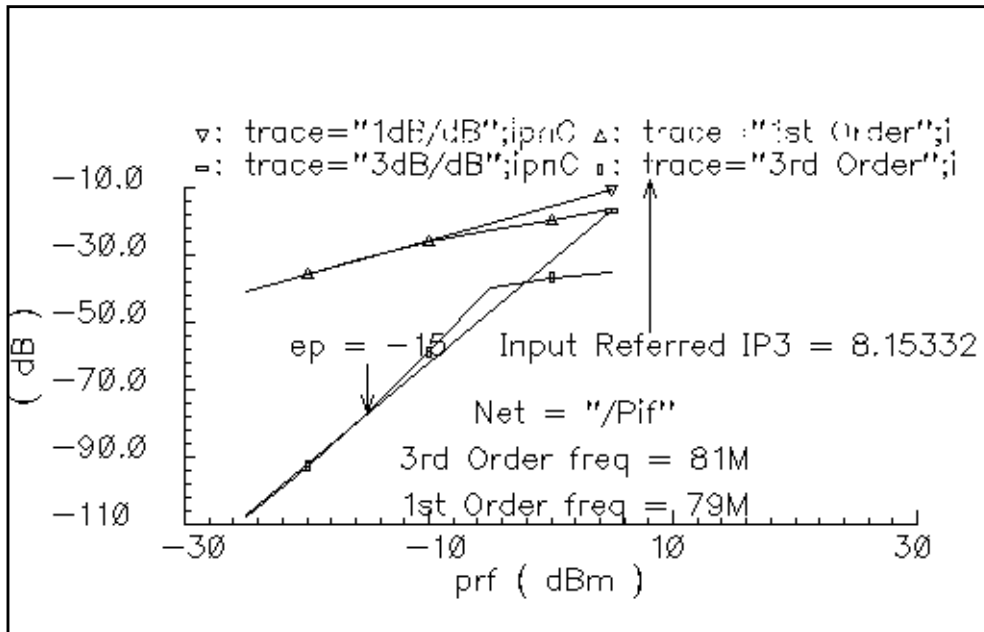
-25	79M
-21	81M
0	921M

Add To Outputs

> Select port or net on schematic ...

9. In the Schematic window, click on the *Pif* signal net.

The Waveform window display appears as shown below. (You can click and drag to move the two labels with arrows, *ep* and *Input Referred IP3*, to improve visibility.)



The general equation to use for computing the third-order intercept point is

$$IP3 = V_{rf} + \frac{(V_{if(-25)} - V_{if(-21)})}{2}$$

where V_{rf} is the value in the *Start* field on the Results form.

Intermodulation Distortion Measurement with Pdisto

The Pdisto analysis lets you consider the effects of a few harmonics in intermodulation distortion calculations. This example describes how to run a Pdisto analysis with the [ne600p mixer circuit](#).

With the Pdisto analysis, you can compute the distortion of moderately sized signals, as opposed to the small signals you investigated with the [swept PSS analysis with PAC](#) approach where you assumed there were no significant distortion effects from the small signal harmonics. With Pdisto, you specify a large, or clock, signal for the fundamental and one or more moderately sized signals whose distortion effects you want to measure.

Setting Up the Simulation

1. If necessary, open the ne600p mixer circuit.
2. If necessary, specify the full path to the model file in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, set the design variables `frf` to 900M and `flo` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

- In the Schematic window, choose the *rf* input port and then choose *Edit – Properties – Objects*.

The Edit Object Properties form for the *rf* port appears.

In the Edit Object Properties form, do the following:

1. Choose *Display small signal params* and remove any remaining values from the previous analysis.
2. Choose *sine* for *Source type*.
3. Choose *Display second sinusoid* and specify the following when the forms changes to display new fields:
 - a. Type `fund2`, or any name you choose, in the *Frequency name 2* field.
 - b. Type `prf - 10` for the *Amplitude 2 (dBm)* value.
 - c. Type `920M` for the *Frequency 2* value.
4. In the Edit Object Properties form, click *OK*.
5. In the Schematic window, choose *Design – Check and Save*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed form looks like this.

Edit Object Properties

OK Cancel Apply Defaults Previous Next Help

Apply To only current instance

Show system user CDF

Property	Value	Display
Library Name	analogLib	off <input type="checkbox"/>
Cell Name	port	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	rf	off <input type="checkbox"/>

User Property	Master Value	Local Value	Display
IvIgnore	TRUE		off <input type="checkbox"/>

CDF Parameter	Value	Display
Resistance	50 Ohms	off <input type="checkbox"/>
Port number	1	off <input type="checkbox"/>
DC voltage		off <input type="checkbox"/>
Source type	sine <input type="checkbox"/>	off <input type="checkbox"/>
Frequency name 1	frf	off <input type="checkbox"/>
Frequency 1	frf Hz	off <input type="checkbox"/>
Amplitude 1 (Vpk)		off <input type="checkbox"/>
Amplitude 1 (dBm)	prf	off <input type="checkbox"/>
Phase for Sinusoid 1		off <input type="checkbox"/>
Sine DC level		off <input type="checkbox"/>
Delay time		off <input type="checkbox"/>
Display second sinusoid	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
Frequency name 2	fund	off <input type="checkbox"/>
Frequency 2	920M Hz	off <input type="checkbox"/>
Amplitude 2 (Vpk)		off <input type="checkbox"/>
Amplitude 2 (dBm)	prf - 10	off <input type="checkbox"/>
Phase for Sinusoid 2		off <input type="checkbox"/>

Setting Up the Pdisto Analysis

1. In the Simulation window, choose *Analyses – Choose*.
The Choosing Analyses form appears.
2. At the top of the Choosing Analyses form, highlight *pdisto*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The form changes to let you specify information for the Pdisto analysis. There are three tones in the *Fundamental Tones* list box, two that were present in the original specifications and one that you added.

To update the fundamental tones, perform the following steps:

1. In the *Fundamental Tones* list box, click on the tone named *f_{rf}*. The tone and its associated values appear in the fields below the list box.
2. In the *Harms* data entry field, type 2 to specify the range of harmonics.
3. Repeat steps 1 and 2 for the fundamental tone named *fund2* (or whatever name you gave it).
4. In the *Fundamental Tones* list box, click on the tone named *f_{lo}*. Choose *Large* from the *Signal* cyclic field placed among the fields. In addition, type 1 for the *Harms* field value.

By choosing *Large* as the *Signal* value, you specify *f_{lo}* to be the large or clock signal. Each Pdisto analysis must have one tone set to be the large signal. When you choose a signal for the large signal, choose a tone that is larger, less sinusoidal, or more nonlinear than the other tones.

Note: You must choose at least one harmonic for each signal that you want to include in a Pdisto analysis. A signal with no harmonics chosen is ignored by the simulation.

In the Choosing Analyses form, do the following:

1. Highlight *moderate* for the *Accuracy Defaults (errpreset)* value.
2. Be sure the *Enabled* button is highlighted.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed form appears like the one below.

The screenshot shows the 'Choosing Analyses' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. Below these are several analysis options, each with a diamond-shaped selection icon: tran, dc, ac, noise, xf, sens, sp, pdisto (selected), pss, pac, pnoise, pxf, and envlp.

The 'Periodic Distortion Analysis' section contains a table titled 'Fundamental Tones' with columns: #, Name, Expr, Value, Signal, SrcId, and Harms. The table has three rows:

#	Name	Expr	Value	Signal	SrcId	Harms
1	flo	flo	1G	Large	lo	1
2	frf	frf	920M	Moderate	rf	2
3	fund2	920M	920M	Moderate	rf	2

Below the table, there are input fields for 'fund2', '920M', '920M', 'Moderate', 'rf', and '2'. There are also buttons for 'Clear/Add', 'Delete', and 'Update From Schematic'.

The 'Accuracy Defaults (emreset)' section has radio buttons for 'conservative', 'moderate' (selected), and 'liberal'. There is a text field for 'Additional Time for Stabilization (tstab)' and a checkbox for 'Save Initial Transient Results (saveinit)' with 'no' selected.

At the bottom, there is a 'Sweep' checkbox and an 'Options...' button.

3. In the Choosing Analyses form, click *OK*.

Selecting Simulation Outputs

1. In the Simulation window, choose *Outputs – To Be Saved – Select on Schematic*.

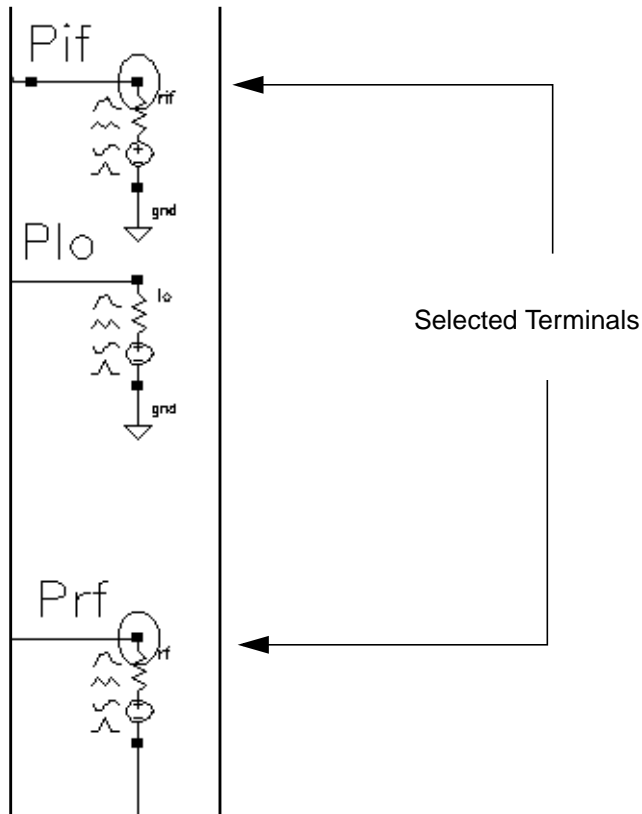
If you want to plot current or power at the end of the simulation, you must explicitly save the currents necessary for the calculations. The most economical way to do this, in terms of simulation time, is to choose specific currents on the schematic.

2. In the Schematic window, click on the appropriate terminals to choose *rf* and *rif*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The terminals are circled in the Schematic window after you choose them.



The selected terminals also appear in the Simulation window.

#	Name/Signal/Expr	Value	Plot	Save	March
1	rif/PLUS		no	yes	no
2	rf/PLUS		no	yes	no

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.
The output log file appears and displays information about the simulation as it runs.
2. Check the CIW for a message that says the simulation completed successfully.

Note: P_{disto} sends less information to the output log than most other SpectreRF

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

analyses. While Pdisto is running for this analysis, there might be periods of several minutes during which there are no messages. This does not mean that the simulation is not progressing.

Plotting the Voltage and Power

- In the Simulation window, choose *Results – Direct Plot – PDISTO*.

The Pdisto Results form and the Waveform window appear.

In the Pdisto Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pdisto* for *Analysis Type*.
3. Highlight *Voltage* for *Function*.
4. Highlight *dB20* for *Modifier*.

The completed form looks like this.

PDISTO Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pdisto

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves

Currently, only spectrum data is available

Modifier

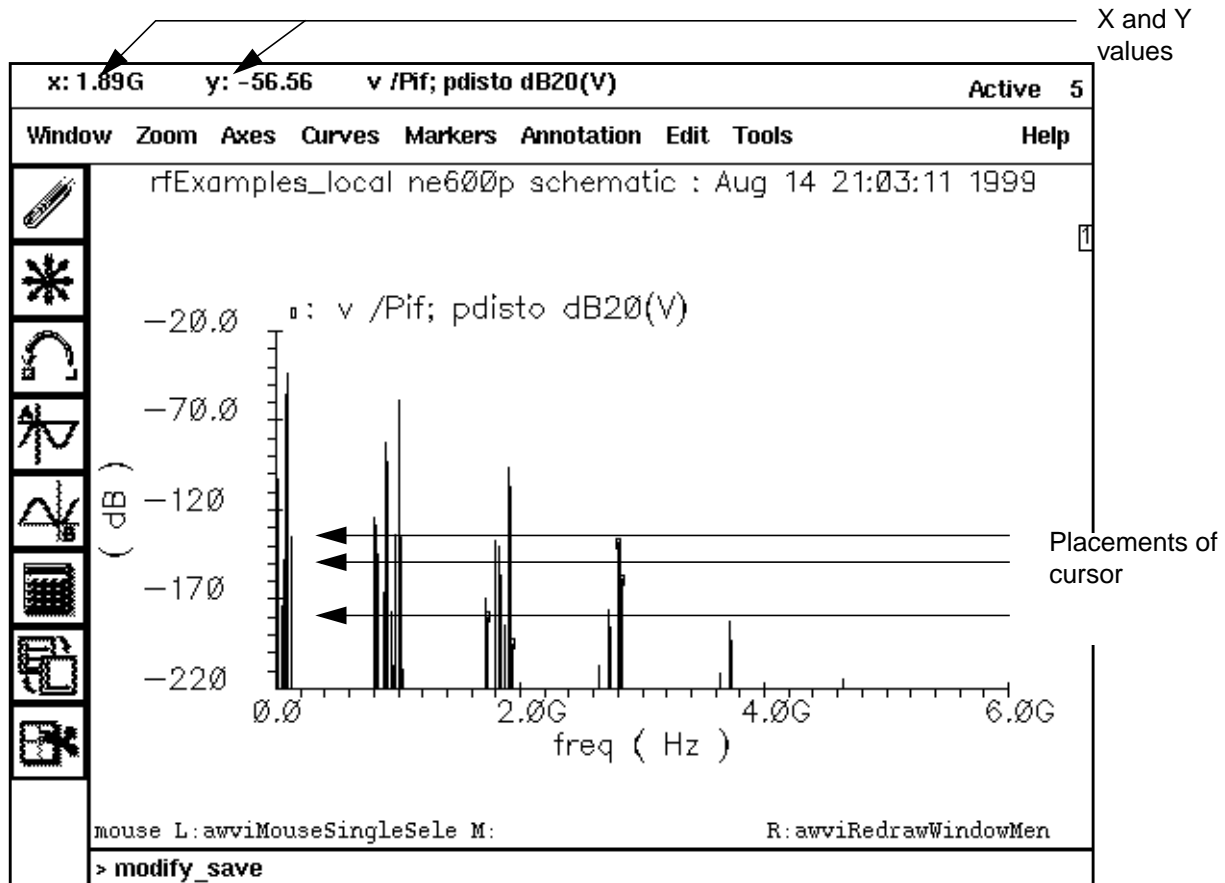
Magnitude Phase dB20
 Real Imaginary

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

- In the Schematic window, click on the *rf* terminal that you previously chose to save the current.

The voltage plot for the *rf* terminal appears.



Because this is a down-converting mixer, the cluster of frequencies from 20 M to 120 M near the Y axis of the plot are of interest. To find a frequency and its associated voltage, place the cursor at the tip of a line. The X and Y values that represent the frequency and the voltage, respectively, appear at the top left corner of the Waveform window. If you place the cursor at the points shown above, you display the following voltage and frequency values:

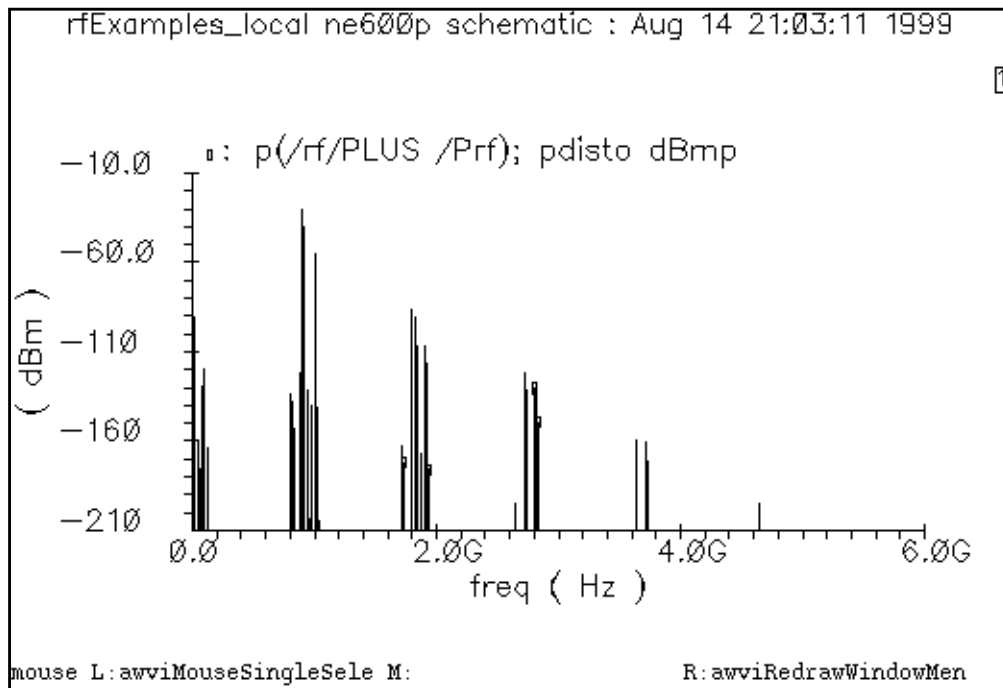
Difference Translation	Frequency	Voltage
920M - 900M	20M	-100.2V
1G - 920M	80M	-129V
1G - 900M	100M	-119V

Plotting Power

To plot the power, do the following in the PDISTO Results form:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *Power* for *Function*.
3. Highlight *dBm* for *Modifier*.
4. In the Schematic window, click on the *rf* terminal.

The plot for the power appears in the Waveform window.



5. If you want to see the voltage and power for the *rif* terminal, repeat the steps to create the plots for the *rif* terminal, but choose the *rif* terminal in the schematic.

Note: If you want to run simulations in other chapters, reset the design variable *frf* to 920 M.

Noise Figure with Pdisto and QPnoise

You can combine a Pdisto analysis with a small-signal QPnoise analysis to determine the degradation of the noise figure for the [ne600p mixer circuit](#) caused by the two interfering signals present at the input of the mixer. The noise figure for the ne600p mixer without the interferer signals is determined in “[Noise Figure Measurement with PSS and Pnoise](#)” on page 160. The Pdisto and QPnoise analyses described in this example include the interferer signals which allow more noise foldings to occur into the output IF band.

The Quasi-Periodic Noise, or QPnoise analysis, is similar to the conventional noise analysis, except that it includes frequency conversion and intermodulation effects. Hence is it useful for predicting the noise behavior of mixers, switched-capacitor filters, and other periodically or quasi-periodically driven circuits.

Setting Up the Simulation

1. If necessary, [open the ne600p mixer circuit](#).
2. If necessary, [specify the full path to the model file](#) in the Model File Set-up form.
3. In the Simulation window, use *Analysis - Disable* to disable any analyses you ran previously. (Check the Simulation window to verify whether or not an analysis is enabled.)
4. If necessary, [set the design variables](#) `frf` to 900M and `fl0` to 1G. (Check the Simulation window to verify the current design variable values.)

Editing the Schematic

1. In the Schematic window, click on the *rf* input port and then choose *Edit – Properties – Objects*.

The Edit Object Properties form for the *rf* input port appears. You use this form to change the list of CDF properties and modify the schematic for this simulation.

In the Edit Object Properties form, do the following:

1. Choose *Display small signal params* and remove any remaining values from the previous analysis.
2. Since the *rf* signal is moderate signal, choose *sine* for *Source type*.
3. Choose *Display second sinusoid* and specify the following when the forms changes to display new fields:

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

- a. Type `fund2`, or any name you choose, in the *Frequency name 2* field.
- b. Type `920M` for the *Frequency 2* value.
- c. Type `prf - 10` for the *Amplitude 2 (dBm)* value.

The completed form looks like this.

CDF Parameter	Value
Resistance	50 Ohms
Port number	1
DC voltage	
Source type	sine <input type="checkbox"/>
Frequency name 1	frf
Frequency 1	frf Hz
Amplitude 1 (Vpk)	
Amplitude 1 (dBm)	prf
Phase for Sinusoid 1	
Sine DC level	
Delay time	
Display second sinusoid	<input checked="" type="checkbox"/>
Frequency name 2	fund2
Frequency 2	920M Hz
Amplitude 2 (Vpk)	
Amplitude 2 (dBm)	prf - 10
Phase for Sinusoid 2	

4. In the Edit Object Properties form, click *OK*.
5. In the Schematic window, choose *Design – Check and Save*.

Setting Up the Pdisto and QPnoise Analyses

In the Choosing Analyses form, set up two analyses, first pdisto (qpss) then qpnoise.

- ▶ In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

Setting Up the Pdisto Analysis

1. At the top of the Choosing Analyses form, highlight *pdisto*.

The form changes to let you specify information for the Pdisto analysis.

2. In the *Fundamental Tones* list box, highlight the tone named *frf*.

The *frf* tone and its associated values appear in the fields below the list box.

3. If necessary, choose *Moderate* from the *Signal* cyclic field.

4. In the *Harms* data entry field, highlight the current value and then type 2 to specify the range of harmonics for the *frf* tone. Any increase in the harmonic range for a moderate tone directly increases the simulation run time.

5. In the list box, highlight the tone named *fund2*.

The new values for *frf* display in the list box. The *fund2* tone and its associated values appear in the fields below the list box.

6. If necessary, choose *Moderate* from the *Signal* cyclic field.

7. In the *Harms* data entry field, highlight the current value and then type 2 to specify the range of harmonics for the *fund2* tone.

8. In the list box, highlight the tone named *f1o*.

The new values for *fund2* display in the list box. The *f1o* tone and its associated values appear in the fields below the list box.

9. Choose *Large* from the *Signal* cyclic field. In addition, highlight the current value in the *Harms* field and type 5 to specify the range of harmonics for the *f1o* tone. Click *Clear/Add*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The new values for `fl0` display in the list box. The harmonic range of 5 gives 11 harmonics (-5, ... ,0, ... +5) for the large tone. Entering a large *Harms* value for the large tone does not affect the simulation run time.

By choosing *Large* as the *Signal* value, you specify `fl0` to be the large or clock signal. Each `pdisto` analysis must have one tone set to be the large signal. When you choose a large signal, select a tone that is larger, less sinusoidal, or more nonlinear than the other tones.

You must choose at least one harmonic for each signal that you want to include in a `pdisto` analysis. A signal with a harmonic range value of 0 is ignored by the simulation. In general,

- ❑ The harmonic range (*Harms*) value for the *Large* tone should be at least 5, guaranteeing 11 harmonics.
- ❑ The *Harms* value for *Moderate* tones should be approximately 2 or 3. For *Moderate* tones, increasing the *Harms* value increases the simulation run time.

When, as in this example, you specify *Harms* values of 5 for the *Large* signal and 2 for the *Moderate* signals, you get `maxharms = [5,2]` which gives you 11 harmonics for the Large tone and 5 harmonics (-2 -1 0 1 2) for the Moderate tone. As a result, you get noise from $11 \times 5 = 55$ frequency sidebands. SpectreRF considers all these combinations when performing QPnoise calculations.

10. Highlight *moderate* for the *Accuracy Defaults (errpreset)* value.
11. Be sure the *Enabled* button is highlighted.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed form appears like the one below.

Periodic Distortion Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId	Harms
1	fln	fln	1G	Large	ln	5
2	frf	frf	900M	Moderate	rf	2
3	fund2	920M	920M	Moderate	rf	2

Moderate

View Harmonics

Accuracy Defaults (empreset)
 conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Sweep

Enabled

Setting Up the QPnoise Analysis

1. In the Analysis section at the top of the Choosing Analyses form, highlight *qpnoise*.

The Choosing Analyses form changes to let you specify data for a Quasi-Periodic Noise Analysis.

2. In the *Frequency Sweep Range (Hz)* cyclic field, choose *Start-Stop*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

3. Type 50M and 150M as the *Start* and *Stop* values, respectively in the fields.

This frequency range covers the frequencies of interest.

4. In the *Sweep Type* cyclic field, choose *Linear* for the sweep type and highlight *Number of Steps*. Type 20 for the *Number of Steps* value in the field.

5. In the *Sidebands* cyclic field, choose *Maximum clock order* and type 5.

Maximum clock order is related to large signal harmonics range. A *Maximum clock order* of 5 insures that sufficient large-signal harmonics are processed. You can increase this value.

6. In the *Output* cyclic field, choose *probe* for the *Output* value.

7. Click on the *Output Probe Instance Select* button. Then click on the appropriate component in the Schematic window to choose *Pif*.

`/rif` appears in the *Output Probe Instance* field.

8. In the *Input Source* cyclic field, choose *port*.

9. Click on the *Input Port Source Select* button. Then click on the appropriate component in the Schematic window to choose *rf*.

`/rf` appears in the *Input Port Source* field.

10. Type 1 0 0 in the *Reference Sideband* field. One entry for each tone.

In the *Reference Sideband* field, enter a vector whose indices have a one-to-one correspondence with the fundamental tones. For this example, the correspondence is illustrated in the following table.

Ref Sideband Vector	1	0	0
Fundamental Tones	LO	FUND2	RF
Frequencies	1 GHz	920 MHz	900 MHz

The reference sideband vector 1 0 0 mixes with the LO tone only to get the output.

11. Highlight *freq* for the *refsidebandoption*.

This selects the frequency as the reference and the reference sideband frequency shift is computed relative to all combinations of fundamental tones.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

When the frequency shift between input and output is calculated, the fundamental tones are multiplied by the *Reference Sideband* vector entries and summed. Since different combinations might lead to the same frequency shift, *refsidebandoption* specifies how to perform this calculation.

- As in this example, you select *freq* to calculate frequency shift based on all possible combinations.
- Select *individual* when you want to calculate frequency shift based on the one specified combination only.

12. Be sure the *Enabled* button is highlighted.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed *Quasi Periodic Noise Analysis* section of the Choosing Analyses form appears like the one below.

The screenshot shows the 'Quasi Periodic Noise Analysis' configuration window. It is divided into several sections:

- Frequency Sweep Range (Hz):** Includes a 'Start-Stop' dropdown menu, a 'Start' text box with '50M', and a 'Stop' text box with '150M'.
- Sweep Type:** Includes a 'Sweep Type' dropdown menu set to 'Linear', a 'Step Size' text box with '20', and a 'Number of Steps' text box with '20'.
- Add Specific Points:** A checkbox that is currently unchecked.
- Sidebands:** Includes a 'Maximum clock order' dropdown menu and a text box with '5'.
- Output:** Includes an 'Output Probe Instance' dropdown menu set to 'probe', a text box with '/rf1', and a 'Select' button.
- Input Source:** Includes an 'Input Port Source' dropdown menu set to 'port', a text box with '/rf1', and a 'Select' button.
- Reference Side-Band:** A text box with '1 0 0'.
- refsidebandoption:** Radio buttons for 'freq' (selected) and 'individual'.
- Enabled:** A checkbox that is checked.
- Options...:** A button at the bottom right.

13. In the Choosing Analyses form, click *OK*.

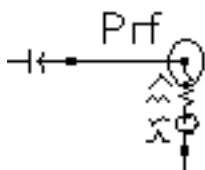
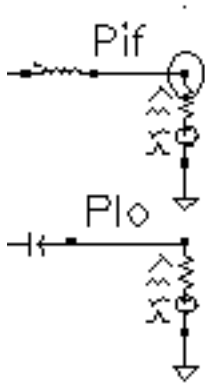
Selecting Simulation Outputs

1. In the Simulation window, choose *Outputs – To Be Saved – Select on Schematic*.

If you want to plot current or power at the end of the simulation, you must explicitly save the currents necessary for the calculations. The most economical way to do this, in terms of simulation time, is to select the specific currents to save on the schematic.

2. In the Schematic window, click on the appropriate terminals to choose *rf* and *rif*.

The terminals are circled in the Schematic window after you choose them.



The selected terminals also appear in the Simulation window.

Outputs					
#	Name/Signal/Expr	Value	Plot	Save	March
1	rif/PLUS		no	yes	no
2	rf/PLUS		no	yes	no

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.
The output log file appears and displays information about the simulation as it runs.
2. Check the CIW for a message that says the simulation completed successfully.

Note: Pdisto analysis sends less information to the output log than most other SpectreRF simulator analyses. While Pdisto is running for this analysis, there might be periods of several minutes during which there are no messages. This does not mean that the simulation is not progressing.

Plotting the Noise Figure

- In the Simulation window, choose *Results – Direct Plot – PDISTO*.
The Pdisto Results form and the Waveform window appear.

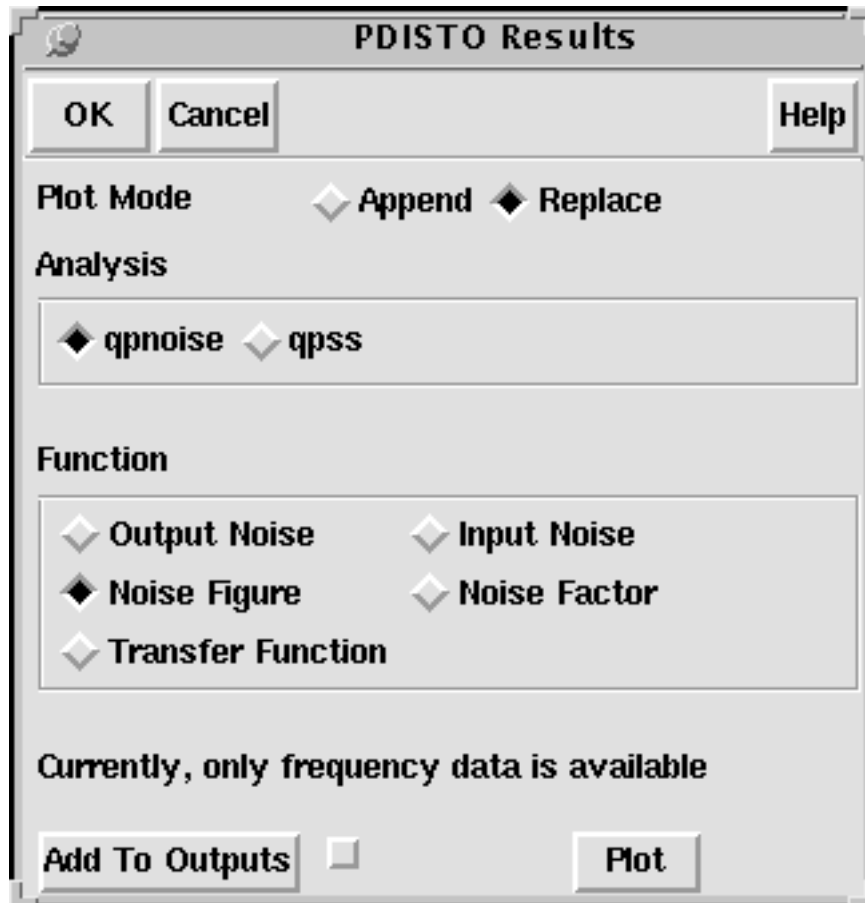
In the Pdisto Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *qnoise* for *Analysis Type*.
3. Highlight *Noise Figure* for *Function*.

Affirma RF Simulator (SpectreRF) User Guide

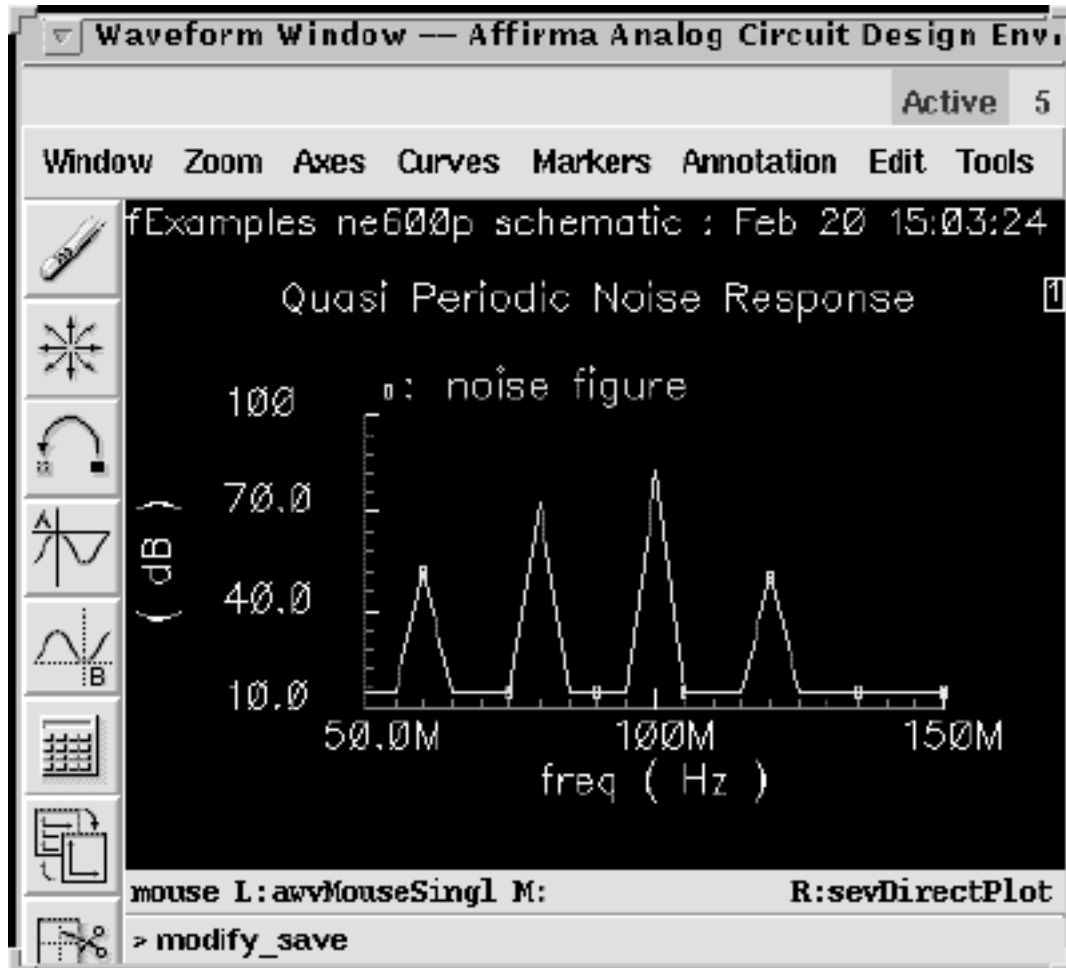
Simulating Mixers

The completed Pdisto Results form for Qnoise appears like the one below.



4. Click on *Plot* in the Pdisto Results form.
The plot appears in the Waveform window.

The Waveform window display appears like the one below:



5. To determine the noise figure at different frequencies, move the cursor along the noise figure curve in the Waveform window. In the above plot, the noise figure is about 73.24dB at 80 MHz.

Plotting the Output Noise

In the Pdisto Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *qnoise* for *Analysis Type*.
3. Highlight *Output Noise* for *Function*.
4. Highlight *dB20* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The completed Pdisto Results form for Qnoise appears like the one below.

The screenshot shows a dialog box titled "PDISTO Results". At the top, there are three buttons: "OK", "Cancel", and "Help". Below the buttons, the "Plot Mode" is set to "Append" (indicated by a diamond icon) and "Replace" (indicated by a filled diamond icon). The "Analysis" section contains two radio buttons: "qpnoise" (filled diamond) and "qpss" (diamond). The "Function" section contains five radio buttons: "Output Noise" (filled diamond), "Input Noise" (diamond), "Noise Figure" (diamond), "Noise Factor" (diamond), and "Transfer Function" (diamond). Below the function section, the text "Currently, only frequency data is available" is displayed. The "Modifier" section contains two radio buttons: "Magnitude" (diamond) and "dB20" (filled diamond). At the bottom, there is a checkbox labeled "Add To Outputs" which is unchecked, and a "Plot" button.

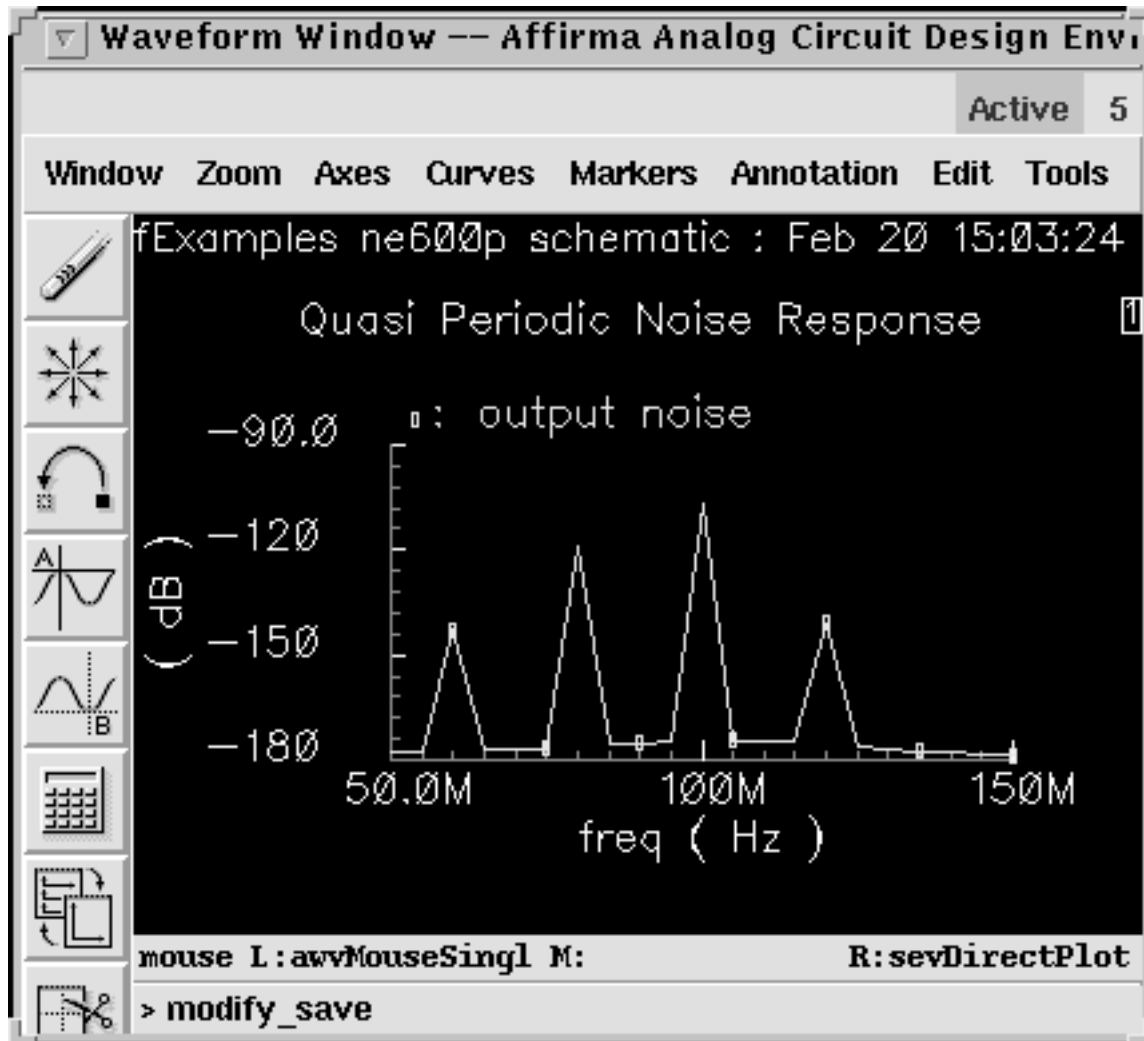
5. Click on *Plot* in the Pdisto Results form.

The plot appears in the Waveform window.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Mixers

The Waveform window display appears like the one below:



- To determine the output noise at different frequencies, move the cursor along the output noise curve in the Waveform window. In the above plot, the output noise is about -119 dB at 80 MHz.

Note: Before you run most other simulation examples in this chapter, be sure to reset the *Design Variable* `frf` to 920M.

Simulating Oscillators

Autonomous PSS Analysis

Periodic steady state (PSS) analysis lets you simulate both driven and autonomous circuits. Driven circuits, such as amplifiers, filters, or mixers, require a time-varying stimulus to create a time-varying response. Autonomous circuits, such as oscillators, however, have time-varying responses and generate non-constant waveforms even though the circuits themselves are time-invariant.

With driven circuits, you specify an analysis period that is an integer multiple of the period of the drive signals. You cannot specify the analysis period for autonomous circuits because you do not know the precise oscillation period in advance. Instead, you specify an estimate of the oscillation period. The PSS analysis uses this estimate to compute the precise period and the periodic solution waveforms.

To enable PSS analysis to compute the precise period, you specify a pair of nodes. PSS analysis monitors the potential difference between these nodes and uses this information to refine its estimate of the period.

Phases of Autonomous PSS Analysis

A PSS Analysis has two phases, an initial transient analysis phase to initialize the circuit and a shooting phase to compute the periodic steady state solution.

The transient analysis phase is divided into the following three intervals:

- A beginning interval that starts at *tstart*, which is normally 0, and continues through the onset of periodicity for the independent sources
- A second, optional stabilization interval of length *tstab*
- A final interval

During this interval, which is four times the estimate of the oscillation period that you specify in the PSS Analysis form, PSS analysis monitors the waveforms in the circuit and improves the estimate of the oscillation period.

In the shooting phase that follows the transient analysis phase, the circuit is simulated repeatedly over one period. The length of the period and the initial conditions are modified to find the periodic steady state solution.

Phase Noise and Oscillators

Oscillators tend to amplify any noise present near the oscillation frequency. The closer the noise frequency to the oscillation frequency, the greater the amplification. Noise amplified by the oscillator in this manner is called *phase noise*. Phase noise is the most significant source of noise in oscillators, and because phase noise is centered about the oscillation frequency, filtering can never completely remove the it.

You can understand phase noise if you recognize that the phase of an oscillator is arbitrary because there is no drive signal to lock to. Any waveform that is a solution to an oscillator can be shifted in time and still be a solution. If a perturbation disturbs the phase, nothing restores the phase, so it drifts without bound. If the perturbation is random noise, the drift is a random walk. Furthermore, the closer the perturbation frequency is to the oscillation frequency, the better it couples to the phase and the greater the drift. The perturbation need not come from random noise. Noise might also couple into the oscillator from other sources, such as the power supplies.

The two examples in this chapter each examine phase noise.

Starting the Oscillator

When you simulate oscillators using PSS analysis, you must start the oscillator. You can start an oscillator with either of the following techniques:

- By supplying a brief impulsive stimulus
 - The stimulus should couple strongly into the oscillatory mode of the circuit and poorly into other long-lasting modes such as bias circuitry.
- By supplying initial conditions for the components of the oscillator's resonator

Stabilizing the Oscillator

With either technique, let the oscillator run long enough to stabilize before starting the shooting phase to compute the steady state solution. Adjust the *tstab* parameter to supply the additional stabilization period.

About the *tline3oscRF* Example

This example computes the periodic steady state solution and the phase noise for the *tline3oscRF* circuit. Phase noise computation is important because phase noise is the most significant source of noise in oscillators. Because phase noise is centered about the oscillation frequency, the noise can never be completely removed by filtering.

The circuit supplies the brief impulsive stimulus needed to start the oscillator, so you do not need to set initial conditions. You supply the data necessary for PSS analysis to estimate the period when you specify the fundamental frequency.

Simulating *tline3oscRF*

Follow the steps below to simulate the *tline3oscRF* circuit:

- In the Command Interpreter Window (CIW), choose *File – Open*.

The Open File form appears. Filling in the Open File form lets you open the schematic.

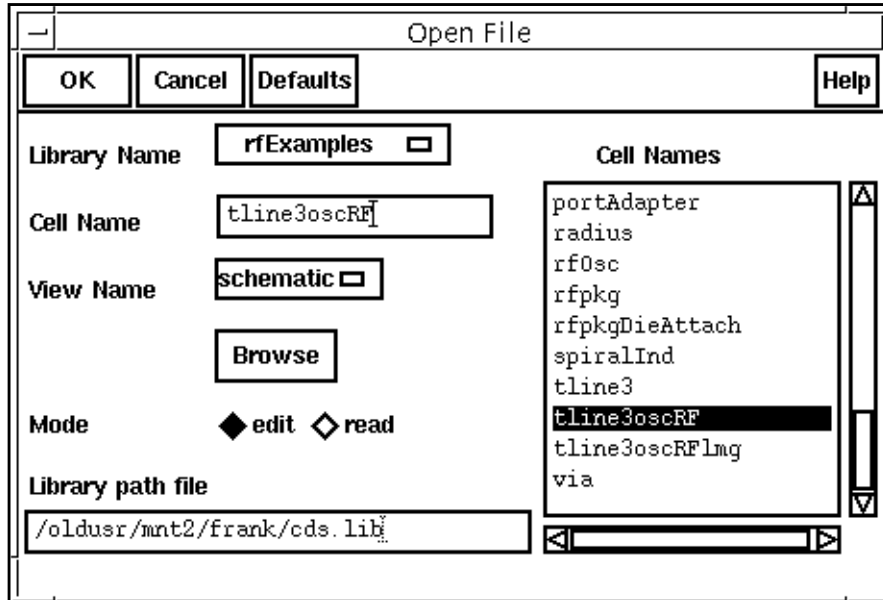
In the Open File form, do the following:

1. Choose *rfExamples* for *Library Name*.
2. Choose *schematic* for *View Name*.
3. In the *Cell Names* list box, highlight *tline3oscRF*. *tline3oscRF* appears in the *Cell Name* field.

Affirma RF Simulator (SpectreRF) User Guide

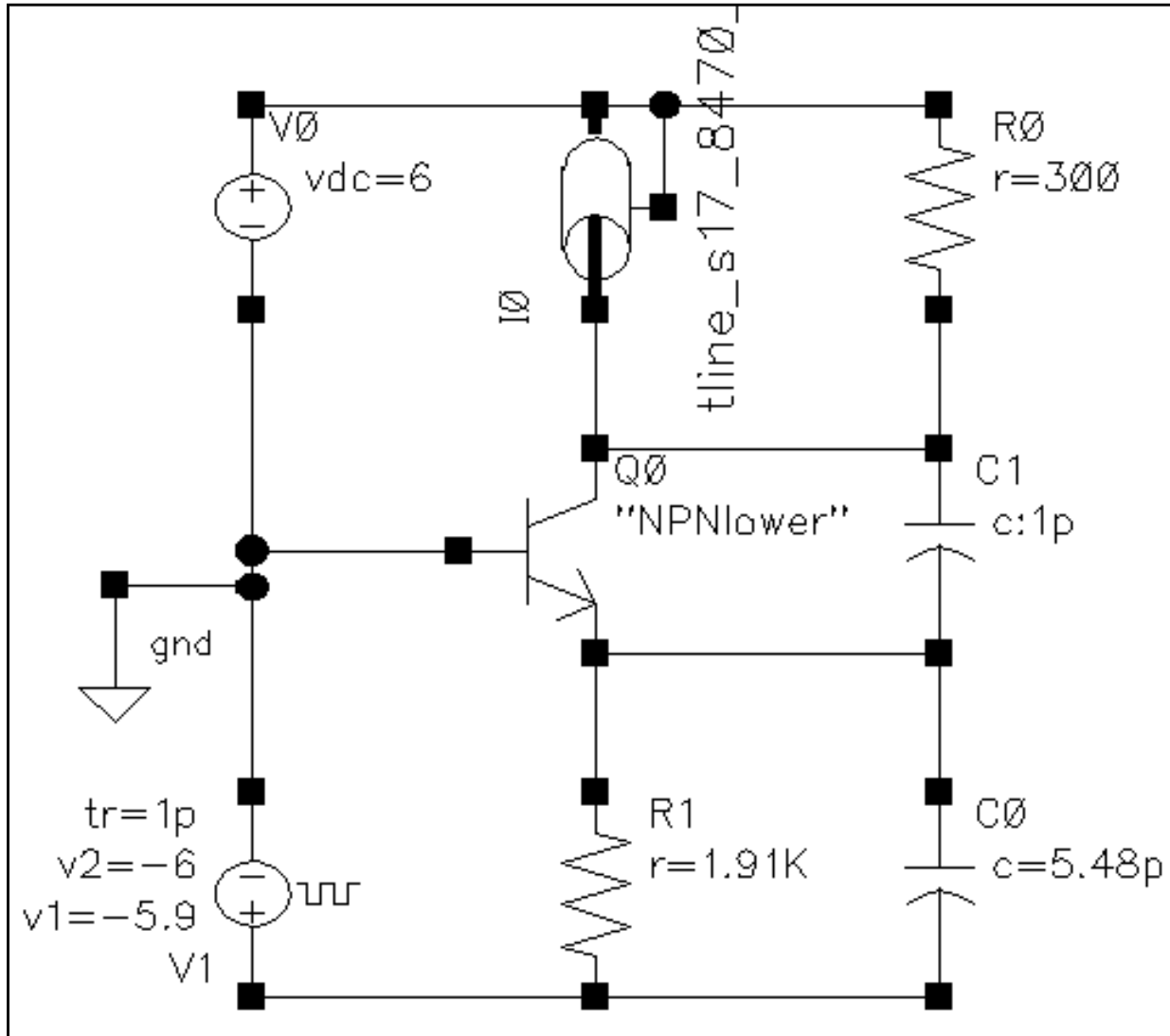
Simulating Oscillators

4. Highlight *edit* to choose the *Mode*.



5. Click *OK*.

The Schematic window opens.

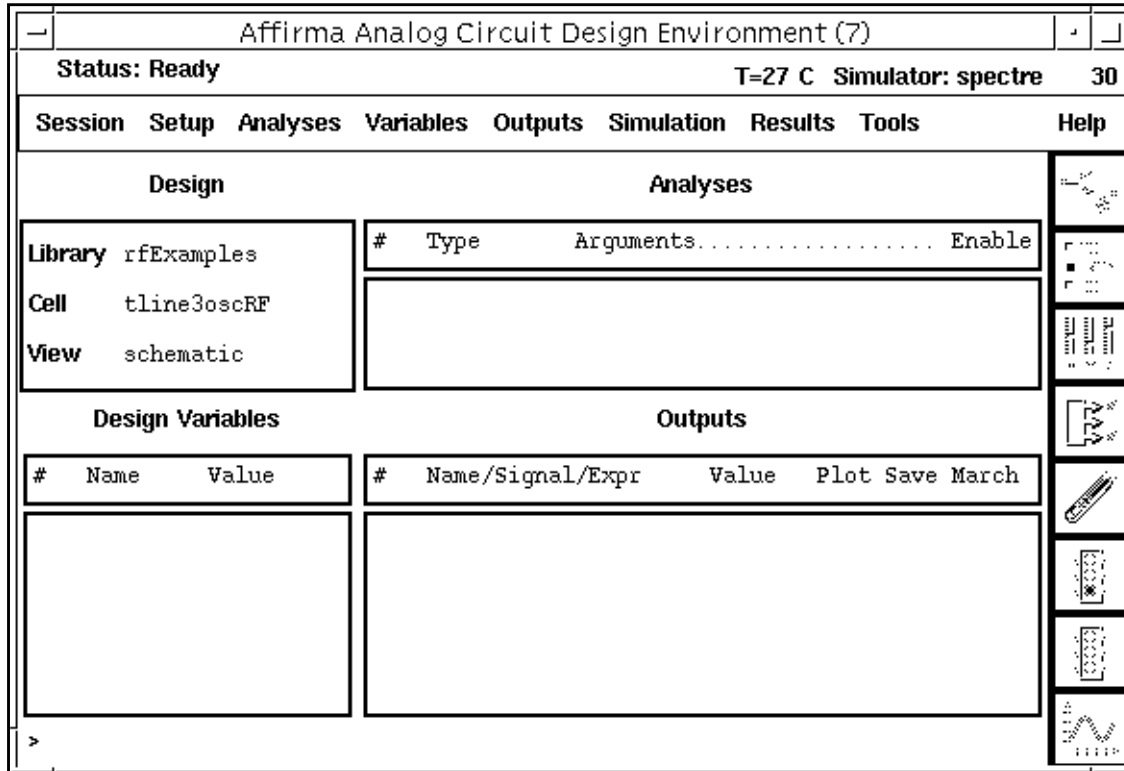


6. In the Schematic window, choose *Tools – Analog Environment*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Simulation window appears.



In the Choosing Simulator/Directory/Host form, do the following:

1. Choose *spectre* for *Simulator*.
2. Type in the name of the project directory, if necessary, in the *Project Directory* field.
3. Highlight the *local* or the *remote* button to specify the *Host Mode*.

For remote simulation, type the name of the host machine and the remote directory in the appropriate fields.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The completed form looks like this.

The dialog box is titled "Choosing Simulator/Directory/Host -- Affirma Analog Circuit Desi". It contains the following fields and controls:

- Buttons: OK, Cancel, Defaults, Help
- Simulator: spectre
- Project Directory: ~/simulation
- Host Mode: local (selected), remote, distributed
- Host: (empty text field)
- Remote Directory: (empty text field)

4. In the *Choosing Simulator/Directory/Host* form, click *OK*.

In the Simulation window, choose *Setup – Model Libraries*.

The Model Library Setup form appears.

1. In the *Model Library File* field, type the full path to the model file including the file name, `rfModels.scs`.
2. In the *Model Library Setup* form, click on *Add*.
3. In the *Model Library File* field, type the full path to the model file including the file name, `tline_s17_8470_40G.scs` and click on *Add*.

The completed form looks like this.

The dialog box is titled "spectre6: Model Library Setup". It contains the following fields and controls:

- Buttons: OK, Cancel, Defaults, Apply, Help
- Model Library File Section:
 - Table with 2 columns: Model Library File, Section
 - Row 1: ... 4.5.3/tools/dfII/samples/artist/models/spectre/rfModels.scs
 - Row 2: ... ls/dfII/samples/artist/models/spectre/tline_s17_8470_40G.scs
- Model Library File Section (opt.):
 - Model Library File: (empty text field)
 - Section (opt.): (empty text field)
- Buttons: Add, Delete, Change, Edit File, Browse...

Type here. →

4. In the *Model Library Setup* form, click *OK*.

5. In the Simulation window, choose *Setup – Simulator/Directory/Host*.

The Choosing Simulator/Directory/Host form appears.

6. Disable any analyses you have run previously.

Analysis Setup

You perform PSS analysis first because the periodic steady state solution must be determined before you can perform a small-signal analysis to determine the phase noise.

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

In the Choosing Analyses form, do the following:

1. Click on *pss* for the analysis.

The form changes to display options needed for PSS analysis.

Note: If an oscillator circuit does not contain a stimulus to start the oscillator, you must run a transient analysis before PSS analysis. You save the node voltages from the transient analysis, and use them as initial conditions in PSS analysis to start the oscillator.

2. In the field next to the *Beat Frequency* and *Beat Period* buttons, type your best estimate of the oscillation frequency.

The *Beat Frequency* button is highlighted by default.

Be sure the *Auto Calculate* button is not highlighted.

Type 1.4G in this example.

3. In the *Output harmonics* cyclic field, choose *Number of harmonics* and type in a reasonable number, such as 10.

4. Highlight *conservative* for the *Accuracy Default (errpreset)*.

5. Highlight *Oscillator*.

The form changes to let you specify the two nodes needed for oscillator simulation.

6. Click on the *Oscillator node Select* button. Then click on the appropriate wire in the Schematic window to choose the oscillator node. In this example, choose *net5*.

`/net5` appears in the *Oscillator node* field.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

7. Leave the *Reference node* field empty.

This field defaults to */gnd!*. You can set *Reference node* to a different value by clicking on the *Reference node Select* button and then choosing the appropriate node in the schematic.

The Choosing Analyses form looks like this.

The screenshot shows the 'Choosing Analyses' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. Below these are several analysis options, each with a diamond icon: 'sens', 'sp', 'pdisto', 'pss' (which is selected with a filled diamond), 'pac', 'pnoise', 'pxf', and 'envlp'. The 'Periodic Steady State Analysis' section contains a table for 'Fundamental Tones' with columns for '#', 'Name', 'Expr', 'Value', 'Signal', and 'SrcId'. Below the table are input fields for 'Fundamental Tones' and a 'Moderate' checkbox. There are also 'Clear/Add', 'Delete', and 'Update From Schematic' buttons. The 'Beat Frequency' is set to '1.4G' and 'Auto Calculate' is unchecked. The 'Output harmonics' section has a 'Number of harmonics' set to '10'. The 'Accuracy Defaults (errpreset)' section has 'conservative' selected, with 'moderate' and 'liberal' as options. There is an input field for 'Additional Time for Stabilization (tstab)' and a 'Save Initial Transient Results (saveinit)' checkbox with 'no' and 'yes' options. The 'Oscillator' section has 'Oscillator node' set to '/net5' and 'Reference node' set to an empty field, both with 'Select' buttons. There is a 'Sweep' checkbox and an 'Enabled' checkbox at the bottom left, and an 'Options...' button at the bottom right.

8. At the top of the Choosing Analyses form, click on *pnoise*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The form changes to let you specify data for the periodic noise analysis.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Noise Analysis

PSS Beat Frequency (Hz) 1.46

Sweeptype

Frequency Sweep Range (Hz)

Start-Stop Start Stop

Sweep Type

Automatic

Add Specific Points

Sidebands

Maximum sideband

Output

probe Output Probe Instance Select

Input Source

port Input Port Source Select

Reference Side-Band

Enabled Options...

In the *pnoise* section of the Choosing Analyses form, do the following.

1. In the *Sweeptype* cyclic field, choose *Relative*.

This choice specifies that the *Frequency Sweep Range (Hz)* values you choose represent frequency values away from the fundamental frequency. For example, if you specify 2K, you choose a value 2K away from the fundamental frequency.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

2. Choose *Start-Stop* for the type of *Frequency Sweep Range* (Hz). Type in reasonable values, such as 1K and 100M as the *Start* and *Stop* values.
3. In the *Sweep Type* cyclic field, choose *Logarithmic*. Use a log sweep when you plot phase noise because of the size of the phase noise values.
4. Highlight *Number of Steps*, and type 201 in the field.

Note: Be sure to use a nonzero starting value for logarithmic sweeps.

5. In the *Sidebands* cyclic field, choose *Maximum sideband* and type 0 in the *Sidebands* field.

6. In the Output cyclic field, choose *voltage* for the *Output* choice.

7. Click on the *Positive Output Node Select* button. Then click on the appropriate wire in the Schematic window to choose *net 5*.

`/net5` appears in the *Positive Output Node* field.

8. Leave the *Negative Output Node* field empty.

This field defaults to `/gnd!`. You can set the *Negative Output Node* to a different value by clicking on the *Negative Output Node Select* button and then choosing the output node in the schematic.

9. In the Input Source cyclic field, choose *none*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Choosing Analyses form looks like this.

The screenshot shows the 'Choosing Analyses' dialog box with the following settings:

- Analysis:** tran, dc, ac, noise, xf, sens, sp, pdisto, pss, pac, pnoise, pxf, envlp
- Periodic Noise Analysis:**
 - PSS Beat Frequency (Hz): 1.46
 - Sweep type: relative, Relative Harmonic: 1
 - Frequency Sweep Range (Hz): Start-Stop, Start: 1K, Stop: 100M
 - Sweep Type: Logarithmic, Points Per Decade: 201, Number of Steps
 - Add Specific Points:
- Sidebands:**
 - Maximum sideband: 0
- Output:**
 - voltage, Positive Output Node: /net5, Select
 - Negative Output Node: [empty], Select
- Input Source:** none
- Enabled: Options...

10. In the Choosing Analyses form, click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The *pss* and *pnoise* options you choose appear in the *Analyses* list box in the Simulation window.

Session	Setup	Analyses	Variables	Outputs	Simulation	Results	Tools
Design		Analyses					
Library	rfExamples	#	Type	Arguments.....			Enable
Cell	tline3oscRF	1	pnoise	0	1K	100M	201 .. yes
View	schematic	2	pss	1.4G	10	/net5	yes

Running the Simulation

1. In the Simulation window, choose *Simulation – Netlist and Run* to run the simulation.
The output log file appears and displays information about the simulation as it runs.
2. Look in the CIW for a message that says the simulation completed successfully.

Plotting the Fundamental Frequency

- In the Simulation Window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear.

To plot the fundamental frequency, do the following in the PSS Results form:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pnoise* for *Analysis Type*.
The form changes to list functions specific to pnoise analyses.
3. Highlight *Pss Beat Frequency* for *Function*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The PSS beat frequency appears on the *PSS Results* form.

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss pnoise

Function

Output Noise Input Noise

Noise Figure Noise Factor

Transfer Function Phase Noise

Pss Beat Frequency

PSS Beat Frequency = 2.022G

Add To Outputs Plot

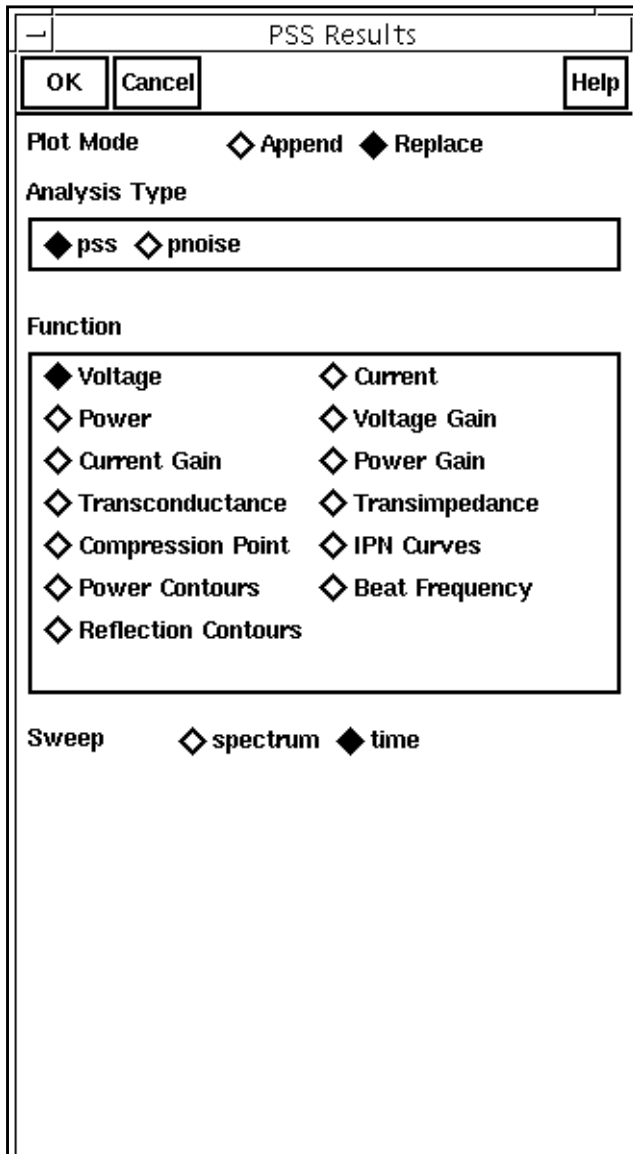
The PSS beat frequency appears here.

Plotting the Steady State Solution

To plot the periodic steady state solution waveform for distortion, do the following in the PSS Results form:

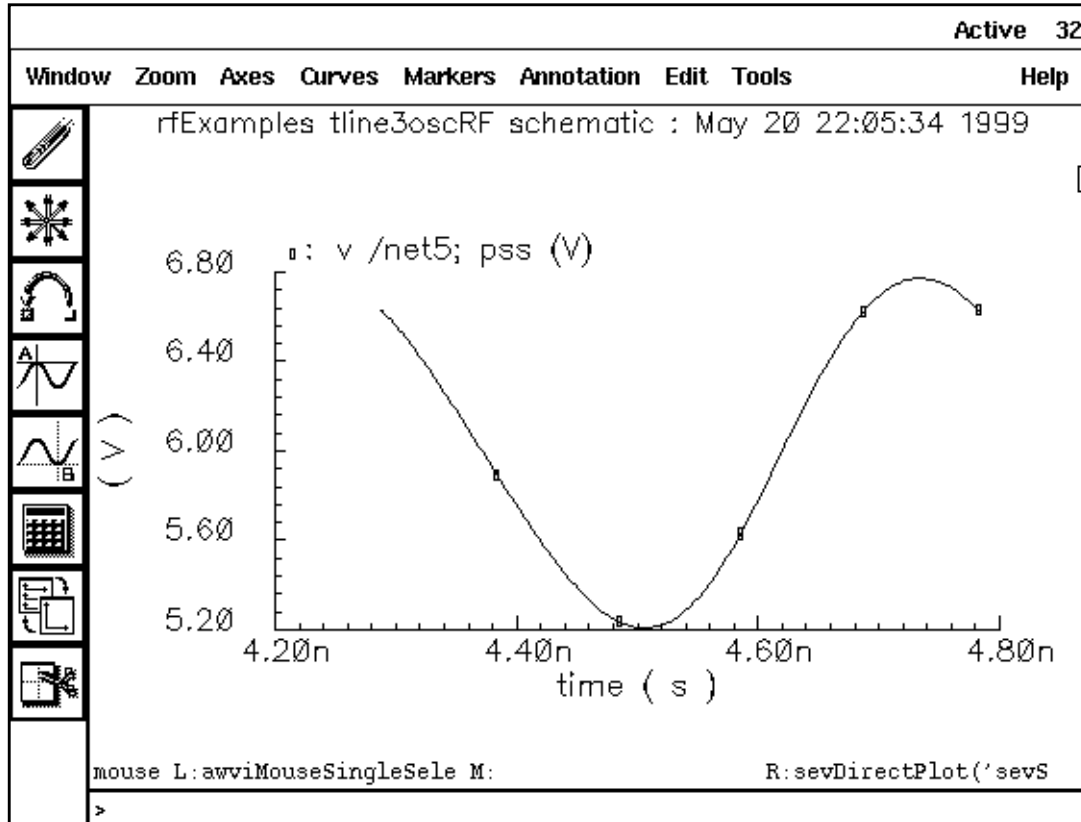
1. Click on *Voltage* for *Function*.

2. Highlight *time* for the *Sweep* value.



3. In the Schematic window, click on the *net5* wire.

The Waveform window displays the sweep of voltage versus time.



Plotting the Phase Noise

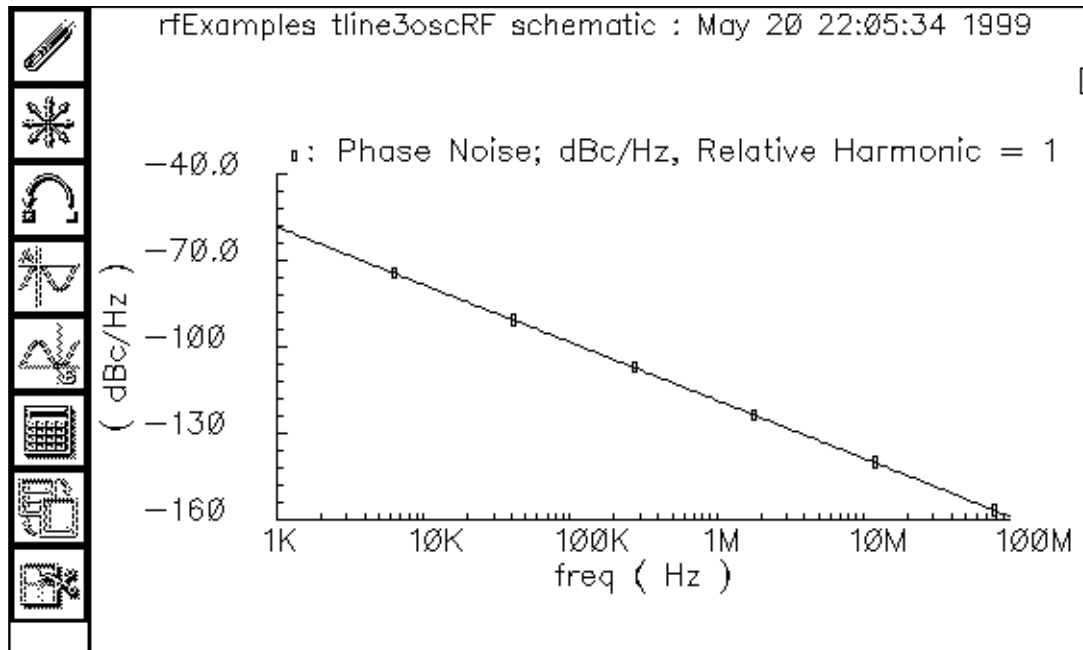
To plot the phase noise, do the following in the PSS Results form:

1. Highlight *Replace* as *Plot Mode*.
2. Highlight *pnoise* as *Analysis Type*.
3. Highlight *Phase Noise* for *Function*.
4. Click on *Plot*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Waveform window displays the phase noise versus frequency with the frequency in logarithmic scale. When you plot phase noise, you get more useful information if you use the log scale.



A Balanced, Tunable Differential Oscillator: oscDiff

This second example, *oscDiff*, is a balanced, tunable differential oscillator. It generates sinusoids in antiphase for differential nodes *VoL* and *VoR*. The oscillation frequency is primarily governed by inductor *L0* and capacitor *C3*. The *C3* value is voltage-dependent and obeys the standard Schottky barrier capacitance equation shown below:

$$C_{tune} = \frac{C_{j0}}{\left(1 + \frac{V_{ctrl}}{\phi}\right)^{\gamma}}$$

where

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

C_{tune} = $C3$ capacitance value

C_{j0} = Zero-bias junction capacitance

V_{cntrl} = Applied varactor voltage

ϕ = Barrier potential

γ = Junction grading coefficient

Simulating *oscDiff*

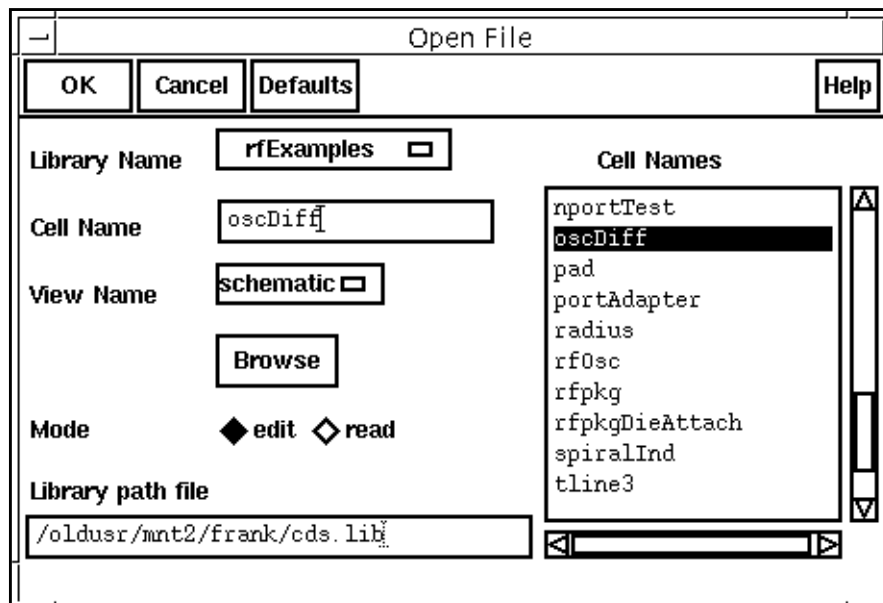
Follow the steps below to simulate the *oscDiff* circuit:

- In the Command Interpreter Window (CIW), choose *File – Open*.

The Open File form appears. Filling in the Open File form lets you open the schematic.

In the Open File form, do the following:

1. Choose *rfExamples* for *Library Name*.
2. Choose *schematic* for *View Name*.
3. In the *CellNames* list box, highlight *oscDiff*.
`oscDiff` appears in the *Cell Name* field.
4. Highlight *edit* for *Mode*.

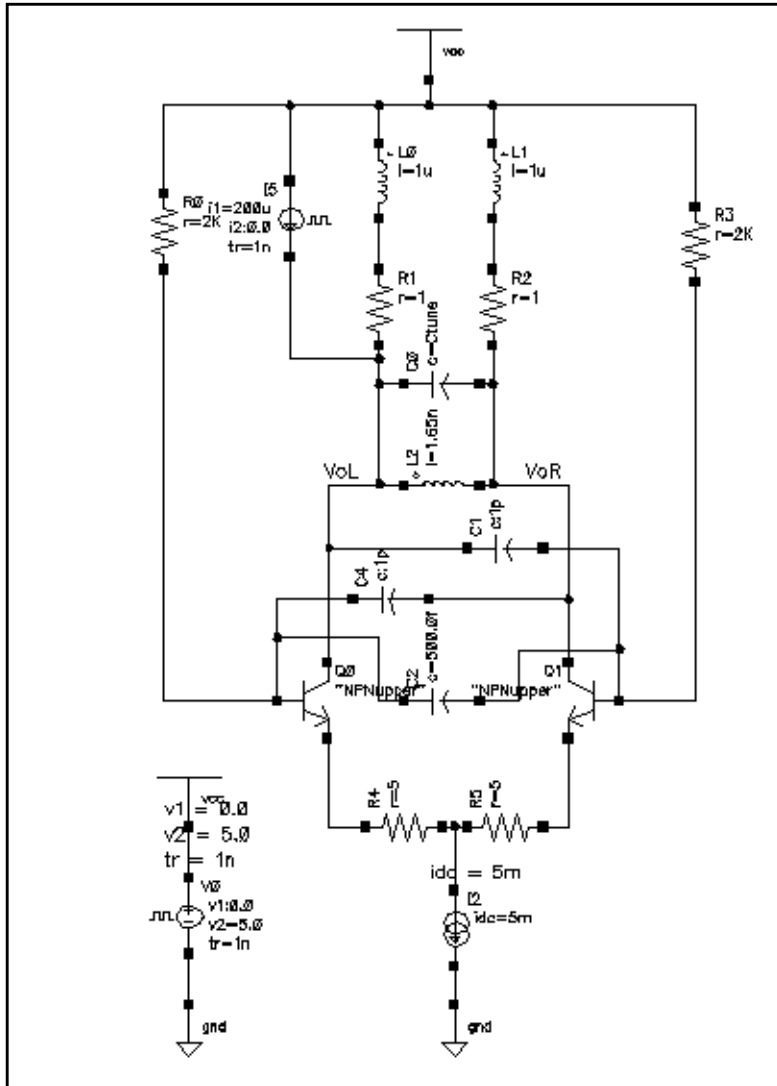


Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

5. Click *OK*.

The *oscDiff* schematic opens.

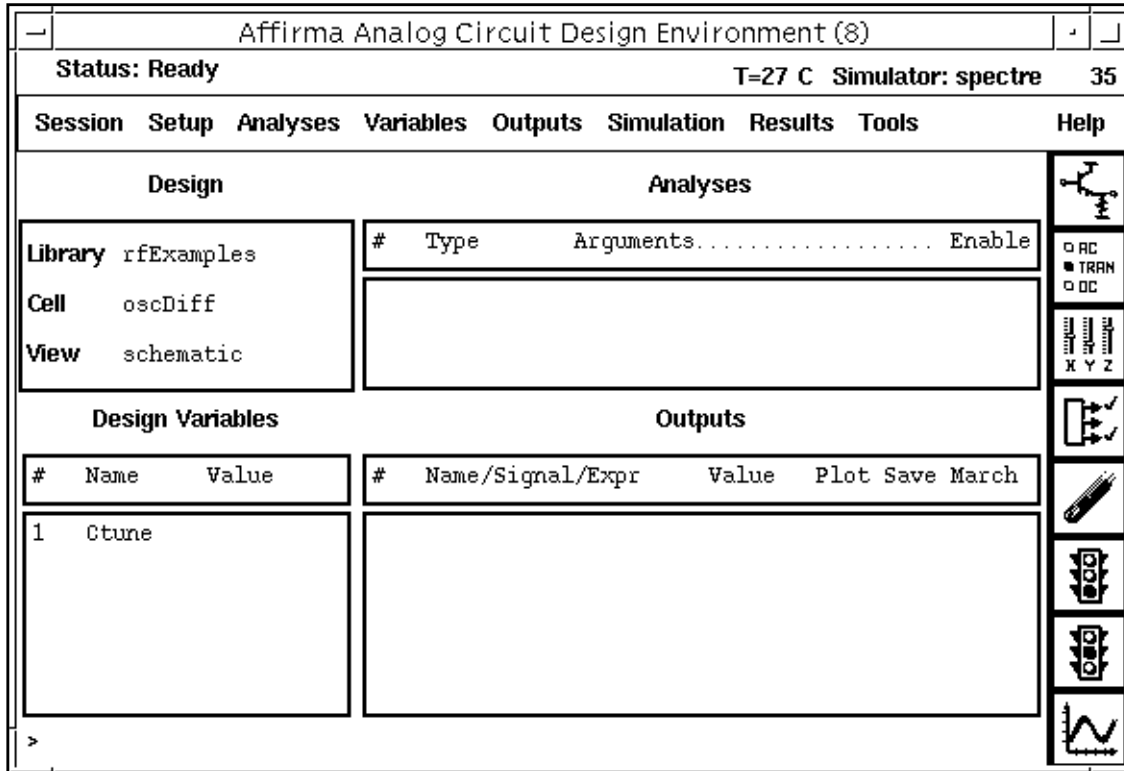


1. In the Schematic window, choose *Tools – Analog Environment*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Simulation window appears.



In the Choosing Simulator/Directory/Host form, do the following:

1. Choose *spectre* for *Simulator*.
2. Type in the name of the project directory, if necessary.
3. Highlight the *local* or the *remote* button to specify the *Host Mode*.
For remote simulation, type in the name of the host machine and the remote directory in the appropriate fields.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The completed form appears like the one below.

The dialog box is titled "Choosing Simulator/Directory/Host -- Affirma Analog Circuit Desi". It has buttons for "OK", "Cancel", "Defaults", and "Help". The "Simulator" field is set to "spectre". The "Project Directory" field contains "~/.simulation". The "Host Mode" field has radio buttons for "local" (selected), "remote", and "distributed". The "Host" and "Remote Directory" fields are empty.

4. In the *Choosing Simulator/Directory/Host* form, click *OK*.

5. In the Simulation window, choose *Setup – Model Libraries*.

The Model Library Setup form appears.

6. In the *Model Library File* field, type the full path to the model file including the file name, `rfModels.scs`.

7. Click *Add*.

The completed form looks like this.

The dialog box is titled "spectre7: Model Library Setup". It has buttons for "OK", "Cancel", "Defaults", "Apply", and "Help". The "Model Library File" field contains the path "... 4.5.3/tools/dfII/samples/artist/models/spectre/rfModels.scs". The "Section" field is empty. Below this, there is another "Model Library File" field and a "Section (opt.)" field, both empty. At the bottom, there are buttons for "Add", "Delete", "Change", "Edit File", and "Browse...". An arrow points to the empty "Model Library File" field with the text "Type here."

8. Click *OK*.

9. In the Simulation window, choose *Setup – Simulator/Directory/Host*.

The Choosing Simulator/Directory/Host form appears.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Transient Analysis

Stop Time

Accuracy Defaults (errpreset)
 conservative moderate liberal

Enabled Options...

2. In the Choosing Analyses form, choose *pss* for the analysis.

The form changes to display options needed for PSS.

You perform PSS first because the periodic steady state solution must be determined before you can perform a small-signal analysis to determine the phase noise.

Note: If an oscillator circuit does not contain a stimulus to start the oscillator, you must run a transient analysis before the PSS analysis. You save the node voltages from the transient analysis, and use them as initial conditions in the PSS analysis to start the oscillator.

In the Choosing Analyses form, do the following:

1. Type 1.9G in the *Beat Frequency* field.

Be sure the *Auto Calculate* button is not highlighted.

The *Beat Frequency* button is highlighted by default.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

2. In the *Output harmonics* cyclic field, choose *Number of harmonics* and type 5 in the field.
3. Highlight *moderate* for the *Accuracy Default (errpreset)*.
4. Highlight *Oscillator*.

The form changes to let you specify the two nodes needed for oscillator simulation.

5. Click on the *Oscillator node Select* button. Then click on the appropriate wire in the Schematic window to choose the oscillator node. In this example, choose the node labelled V_{OR} .

$/V_{OR}$ appears in the *Oscillator node* field.

6. Click on the *Reference node Select* button. Then click on the appropriate wire in the Schematic window to choose the oscillator node. In this example, choose the node labeled V_{OL} .

$/V_{OL}$ appears in the *Reference node* field. Because this is a differential oscillator, the reference node is not *gnd* as it was in the last example.

7. Type $100.5n$ in the *Additional Time for Stabilization (tstab)* field.

You can use a slightly different value as long as it is above 100.

8. If you want to save the results of the initial transient analysis, set *Save Initial Transient Results (saveinit)* to *yes*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Choosing Analyses form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId

Moderate

Clear/Add Delete Update From Schematic

Beat Frequency Auto Calculate

Beat Period

Output harmonics

Number of harmonics

Accuracy Defaults (errpreset)

conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator Oscillator node Select
Reference node Select

Sweep

Periodic Noise Analysis Set up

- At the top of the Choosing Analyses form, click on *pnoise*.

The form changes to let you set up the pnoise analysis.

In the pnoise section of the Choosing Analyses form, do the following:

1. In the *Sweeptype* cyclic field, choose *Relative*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

This choice specifies that the *Frequency Sweep Range (Hz)* values you choose represent frequency values away from the fundamental frequency. For example, if you specify 2K, you choose a value 2K away from the fundamental frequency.

2. In the *Frequency Sweep Range(Hz)* cyclic field, choose *Start- Stop* and type 1 and 100M as the *Start* and *Stop* values, respectively.

3. In the *Sweep Type* cyclic field, choose *Logarithmic*.

Use a log sweep when you plot phase noise because of the size of the phase noise values.

4. Highlight *Points Per Decade* and type 5 in the field.

Note: Be sure to use a nonzero starting value for logarithmic sweeps.

5. In the *Sidebands* cyclic field, choose *Maximum sideband* and type 0 in the field.

6. In the Output cyclic field, choose *voltage* for the *Output* choice.

7. Click on the *Positive Output Node Select* button. Then click on the appropriate wire in the Schematic window to choose *VoR*.

/VoR appears in the *Positive Output Node* field.

8. Click on the *Negative Output Node Select* button. Then click on the appropriate wire in the Schematic window to choose *VoL*.

/VoL appears in the *Negative Output Node* field. Because this is a differential oscillator, the negative output node is not *gnd* as it was for the *tline3oscRF* oscillator.

9. In the *Input Source* cyclic field, choose *none*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Choosing Analyses form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Noise Analysis

PSS Beat Frequency (Hz) 1.9G

Sweeptype relative Relative Harmonic 1

Frequency Sweep Range (Hz)

Start-Stop Start 1 Stop 100M

Sweep Type

Logarithmic Points Per Decade 5
 Number of Steps

Add Specific Points

Sidebands

Maximum sideband 0

Output

voltage Positive Output Node /VoR
 Negative Output Node /VoL

Input Source

none

Enabled

10. In the Choosing Analyses form, click *OK*.

The *pss* and *pnoise* analyses you set up appear in the *Analyses* list box in the Simulation window.

Editing Design Variables

► In the Simulation window, choose *Variables – Edit*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

The Editing Design Variables form appears. For this example, you use this form to set the value of the variable *Ctune*.

In the Editing Design Variables form, do the following:

1. In the Table of Design Variables list box, highlight *Ctune*.
2. Type 3.5p in the *Value (Expr)* field.
3. Click on *Change*.

The completed form looks like this.

Selected Variable		Table of Design Variables	
Name	Ctune	#	Name Value
Value (Expr)	3.5p	1	Ctune 3.5p

4. Click *OK*.

Ctune appears in the *Design Variables* section of the Simulation window with its modified value

Running the Simulation

1. In the Simulation window, choose *Simulation – Netlist and Run* to run the simulation.
The output log file appears and displays information about the simulation as it runs.
2. Look in the CIW for a message that says the simulation completed successfully.

Plotting the Fundamental Frequency

- In the Simulation Window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear,

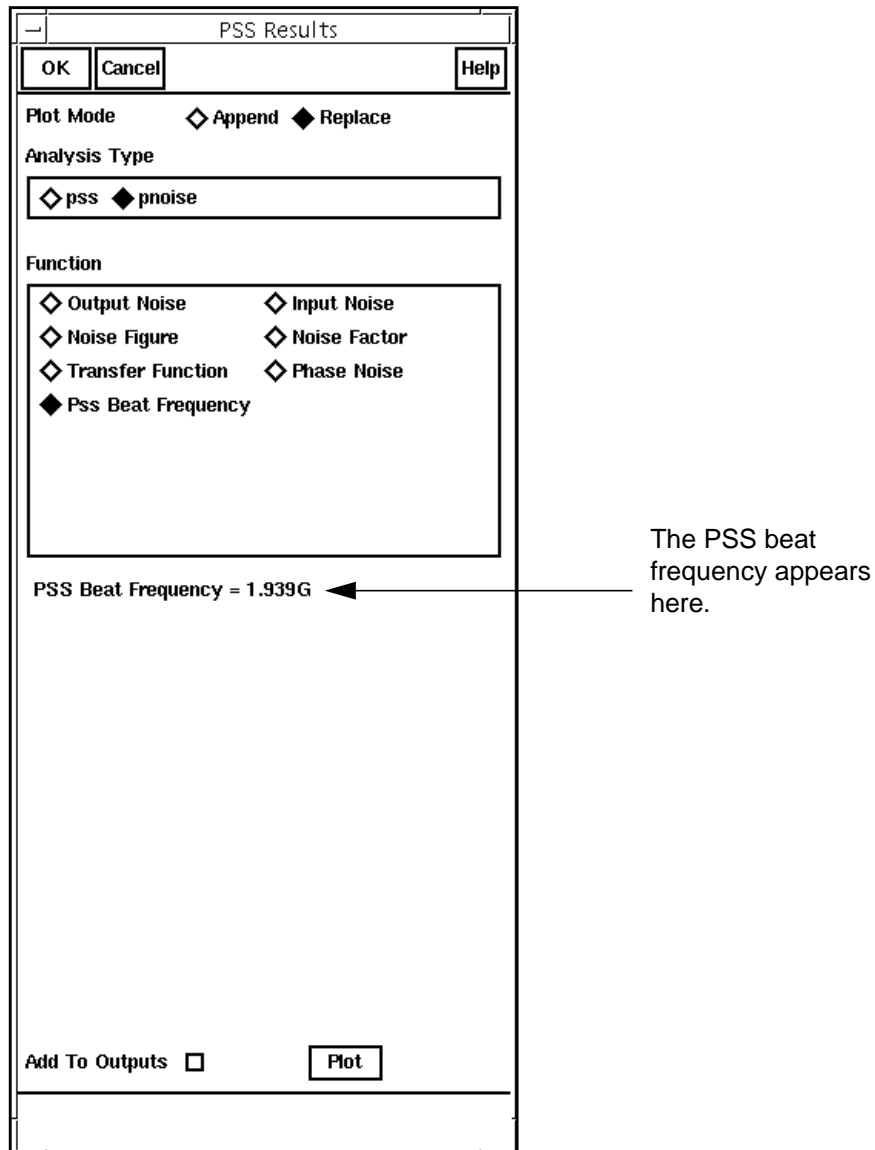
To look at the fundamental frequency, do the following in the PSS Results form:

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pnoise* for *Analysis Type*.
3. Highlight *Pss Beat Frequency* for *Function*.

The PSS beat frequency appears on the *PSS Results* form



The PSS beat frequency appears here.

Plotting the Output and Phase Noise

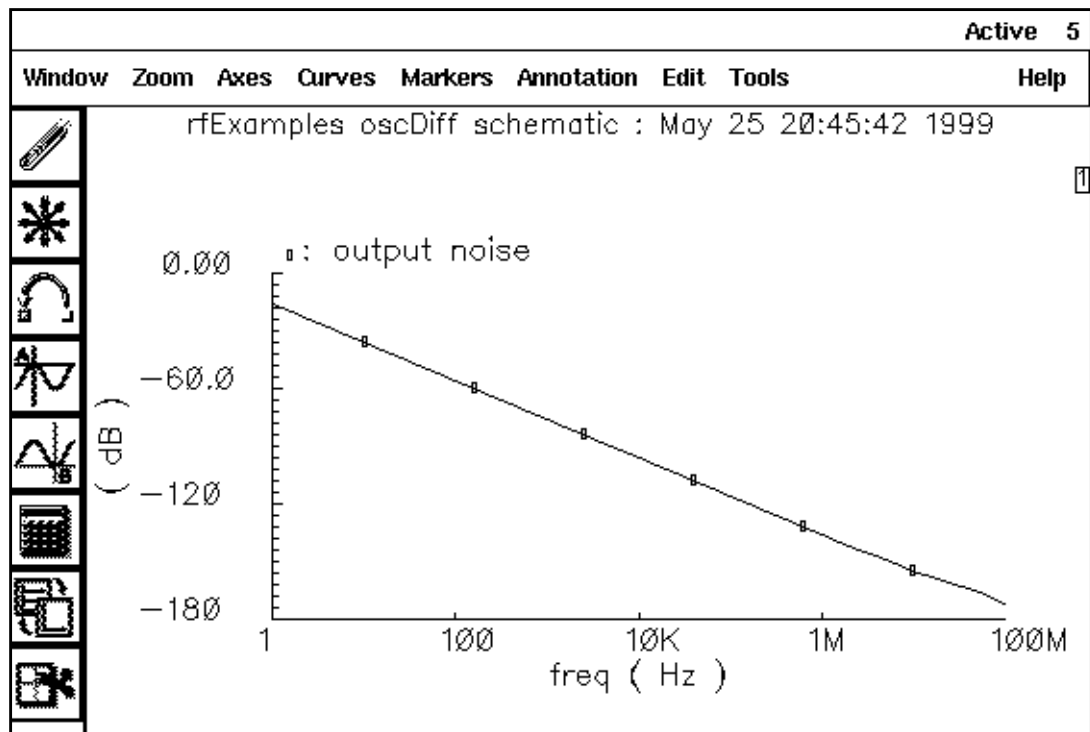
To plot the output noise, do the following in the PSS Results form:

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *Output Noise* for *Function*.
3. Highlight *dB20* for *Modifier*.
4. Highlight *Plot*.

The output noise is plotted in the Waveform window.

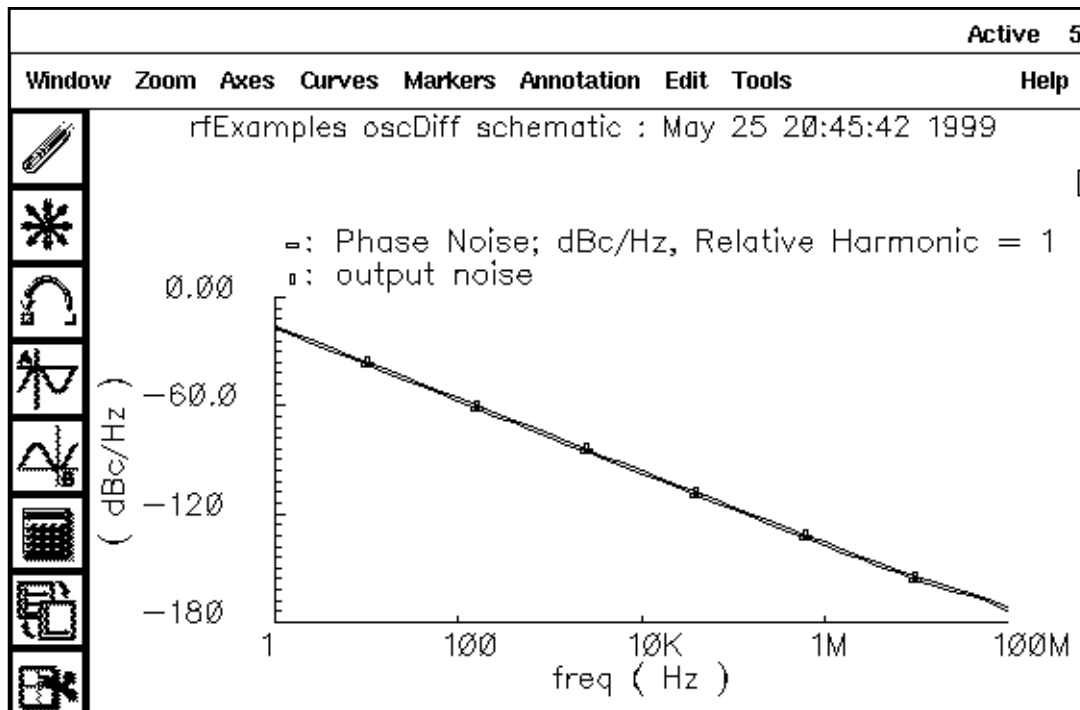


If you plot the phase noise on the same graph, you can see the linear relationship between the output noise and the phase noise. The output noise is scaled by the carrier amplitude to produce the phase noise value.

In the PSS Results form, do the following:

1. Highlight *Append* for *Plot Mode*.
2. Highlight *Phase Noise* for *Function*.
3. Click on *Plot*.

The output noise and phase noise plots are closely parallel.



Troubleshooting for Oscillators

If your oscillator simulation fails to converge, try the following techniques to help the SpectreRF simulator find the solution. You can find more information about each of these techniques in the *SpectreRF Theory* document available in the OpenBook[®] online documentation library.

- Increase the value of the *tstab* parameter.
- Be sure that you successfully started the oscillator.

PSS fails to converge if the circuit cannot sustain an autonomous oscillation. Start the oscillator so that it responds with a signal level between 25 percent and 100 percent of the expected final level. Also, do not “kick” the oscillator so hard that its response is unnecessarily nonlinear. If possible, avoid exciting response modes that are unrelated to the oscillation, especially those associated with longtime constants.

- Improve your estimate of the period.

Be careful that the period you specify is not too short. Overestimating the period is better than underestimating the period.

- When possible, use *gear2only* as the value of the *method* parameter.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Oscillators

- Increase the value of the *maxperiods parameter* to increase the maximum number of iterations for the shooting methods.
- If the shooting iteration approaches convergence and then fails, increase the value of the *steadyratio parameter*.

The *steadyratio* parameter controls how much the final solution can deviate from being periodic.

- Change the value of the *tolerance* parameter.

Changing the value of the *tolerance* parameter does not change the accuracy of the final solution, but it increases the chance that the simulation converges.

- Tighten the normal simulation tolerances by changing the values of the *maxstep*, *reltol*, *lteratio*, and *errpreset* parameters. Avoid setting *errpreset* to *liberal*.

Simulating Low-Noise Amplifiers

Readers of this chapter must have basic familiarity with the SpectreRF simulator analyses as well as knowledge about low-noise amplifier design. For more information about the SpectreRF simulator analyses, see Chapter 1, “[SpectreRF Analyses](#).”

Types of Analysis Shown in this Chapter

This chapter shows you how to simulate and plot the following:

- [Voltage Gain](#)
- [Output Voltage Distribution](#)
- [S-Parameter Analysis](#), including the [Voltage Standing Wave Ratio](#)
- [S-Parameter Noise Analysis](#), including
 - [Noise Figure and Minimum Noise Figure](#)
 - [Equivalent Noise Resistance](#)
 - [Load and Source Stability Circles](#)
 - [Noise Circles](#)
- [Noise Figure Calculations with the Pnoise Analysis](#)
- [The 1dB Compression Point](#)
- [The Third-Order Intercept Point](#)
- [Conversion Gain and Power Supply Rejection with the Periodic Transfer Function Analysis](#)

Procedures for Simulating the LNA Example

1. In the Command Interpreter Window ([CIW](#)), choose *File – Open*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The Open File form appears.

2. In the Open File form, do the following:
 - a. Choose *rfExamples* for *Library Name*.
 - b. Choose *InaSimple* in the *Cell Names* list box.
 - c. Choose *schematic* for *View Name*.

The filled-in form appears like the one below.

Open File

OK Cancel Defaults Help

Library Name

Cell Name

View Name

Mode edit read

Library path file

Cell Names

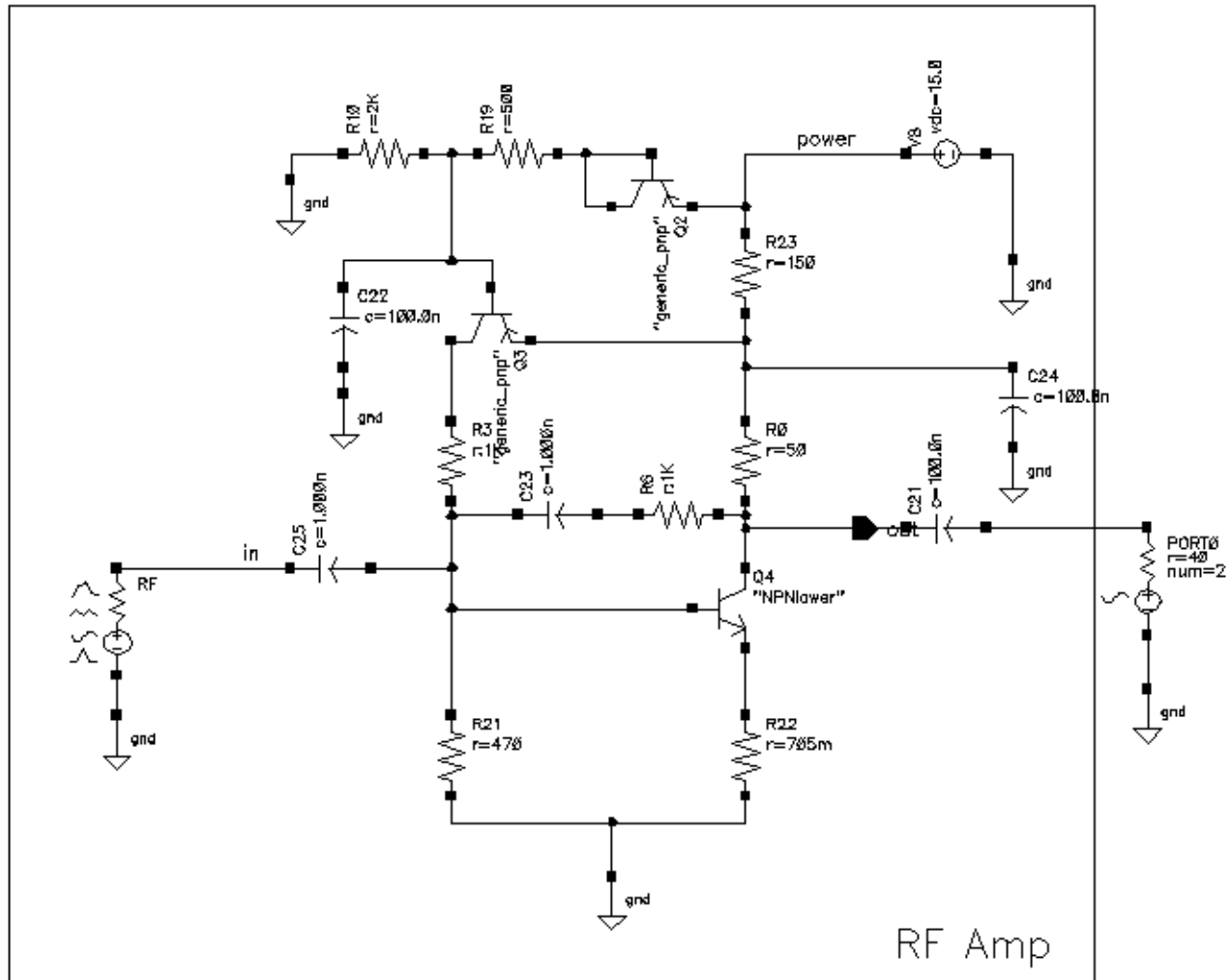
- libraOsc
- lna300
- lnaSimple**
- mharmBiasSwp
- mharmOsc
- mixer0
- mline
- mlineoscRFmg
- ne600
- nportTest

3. In the Open File form, click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The [Schematic window](#) now shows the *InaSimple* schematic.

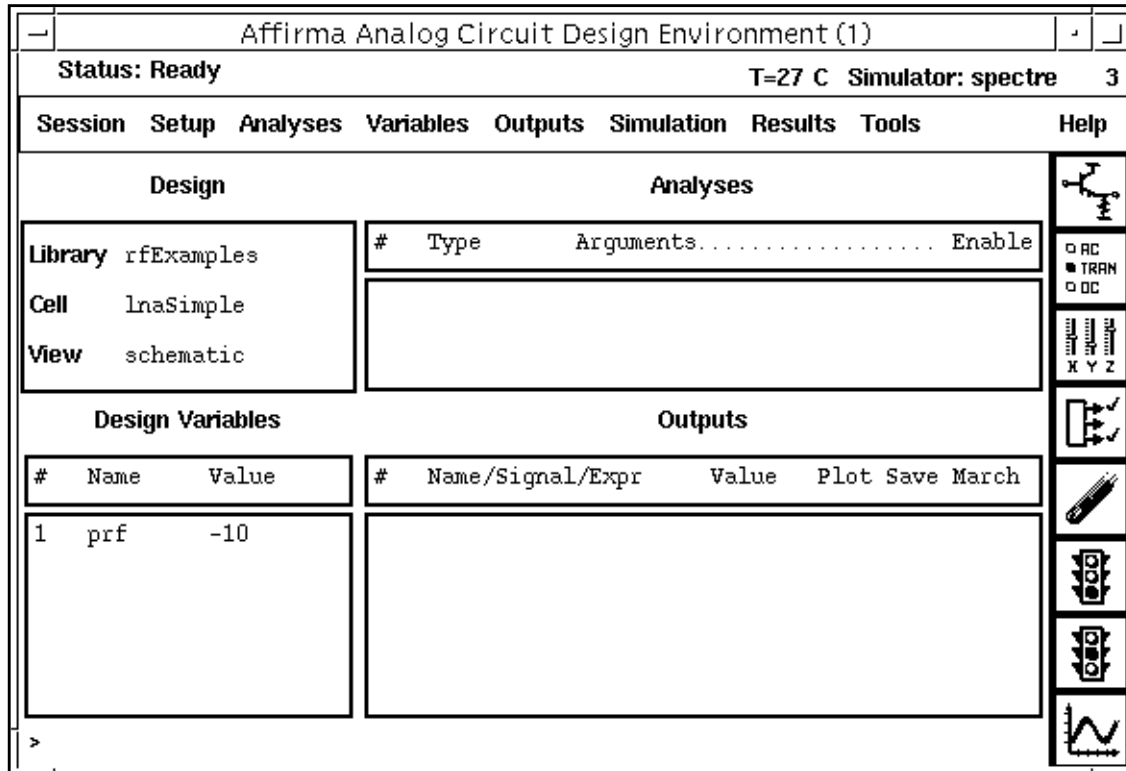


4. In the [Schematic window](#), choose *Tools – Analog Environment*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The Simulation window appears.



Choosing Simulator Options

1. In the Simulation window, choose *Setup – Simulator/Directory/Host*.

The Choosing Simulator/Directory/Host form appears.

2. In the Choosing Simulator/Directory/Host form, do the following:
 - a. Choose *spectre* for *Simulator*.
 - b. Type the name of the project directory, if necessary.
 - c. Highlight the *local* or the *remote* button to specify the *Host Mode*.

For remote simulation, type in the name of the host machine and the remote directory in the appropriate fields.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The completed form looks like this.

The screenshot shows a dialog box titled "Choosing Simulator/Directory/Host -- Affirma Analog Circuit Desi". It has buttons for "OK", "Cancel", "Defaults", and "Help". The "Simulator" field is set to "spectre" with a dropdown arrow. The "Project Directory" field contains "~/simulation". The "Host Mode" field has three radio buttons: "local" (selected), "remote", and "distributed". The "Host" and "Remote Directory" fields are empty text boxes.

3. In the Choosing Simulator/Directory/Host form, click *OK*.

4. In the Simulation window, choose *Outputs – Save All*.

The Save Options form appears.

5. In the *Select signals to output section*, be sure *allpub* is highlighted.

The screenshot shows a dialog box titled "Save Options" with buttons for "OK", "Cancel", "Defaults", "Apply", and "Help". It contains several sections of options:

- Select signals to output (save)**: none selected lvlpub lvl allpub all
- Select power signals to output (pwr)**: none total devices subckts all
- Set level of subcircuit to output (nestlvl)**: []
- Select device currents (currents)**: selected nonlinear all
- Set subcircuit probe level (subcktprobelvl)**: []
- Select AC terminal currents (useprobes)**: yes no

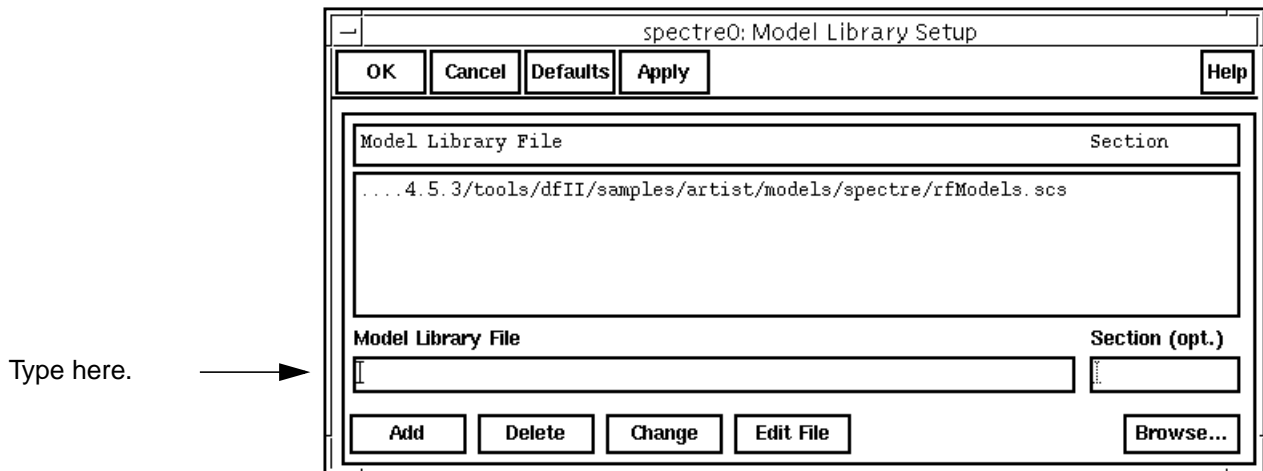
Setting Up Model Libraries

1. In the Simulation window, choose *Setup – Model Libraries*.

The Model Library Setup form appears.

1. In the *Model Library File* field, type the full path to the model file including the file name, `rfModels.scs`.
2. In the *Model Library Setup* form, click on *Add*.

The completed form looks like this.



3. In the *Model Library Setup* form, click *OK*.

Voltage Gain Calculation and Output Voltage Distribution

The most important characteristic of an amplifier is its gain. In the following example, a periodic steady state (PSS) analysis determines the voltage gain of the low noise amplifier.

Editing Schematic

1. In the Schematic window, choose the RF voltage source.
2. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears.

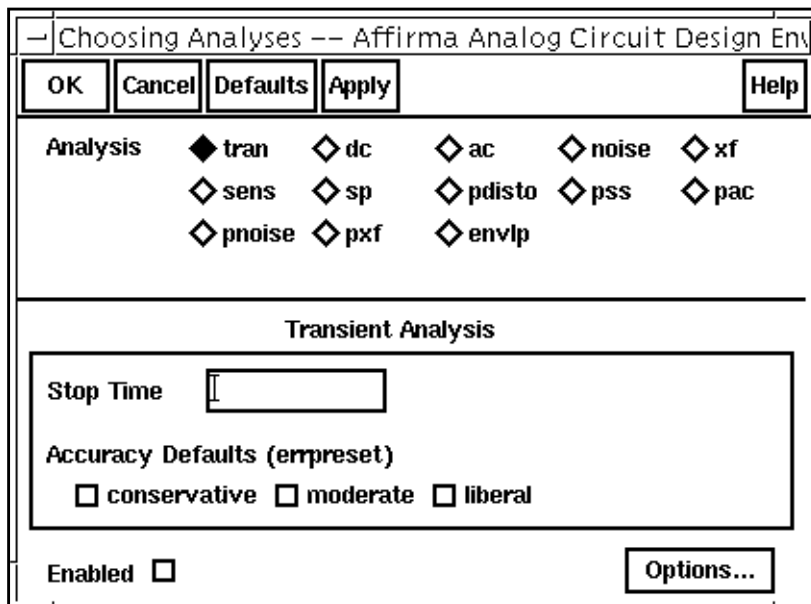
3. In the Edit Object Properties form, type `sine` as the *Source type*, if necessary, and click *OK*.

4. In the Schematic window, choose *Design – Check and Save*.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.



2. In the Choosing Analyses form, choose *pss*.

The form changes to let you specify data for the PSS analysis.

3. In the *Fundamental Tones* list box, be sure the *Auto Calculate* button is highlighted.

The fundamental frequency for this analysis, which is 900M, is now specified in the *Beat Frequency* field.

The *Beat Frequency* button is highlighted by default.

4. In the Choosing Analyses form, do the following:
 - a. In the *Output harmonics* cyclic field, choose *Number of harmonics* for the *Output harmonics* choice and type a reasonable value, such as 10, in the field. (You can choose a different number.)
 - b. Highlight *moderate* for the *Accuracy Defaults (errpreset)* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

c. Highlight *Enabled*, if necessary.

The filled-in form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1	frf	900M	900M	Moderate	RF

Moderate

Clear/Add Delete Update From Schematic

Beat Frequency 900M Auto Calculate

Beat Period

Output harmonics

Number of harmonics 10

Accuracy Defaults (errpreset)

conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep

Enabled Options...

5. In the Choosing Analyses form, click *OK*.

Running Simulation

1. To run the pss analysis, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the output log file to be sure the simulation is completed successfully.

Plotting Voltage Gain

- In the Simulation window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear.

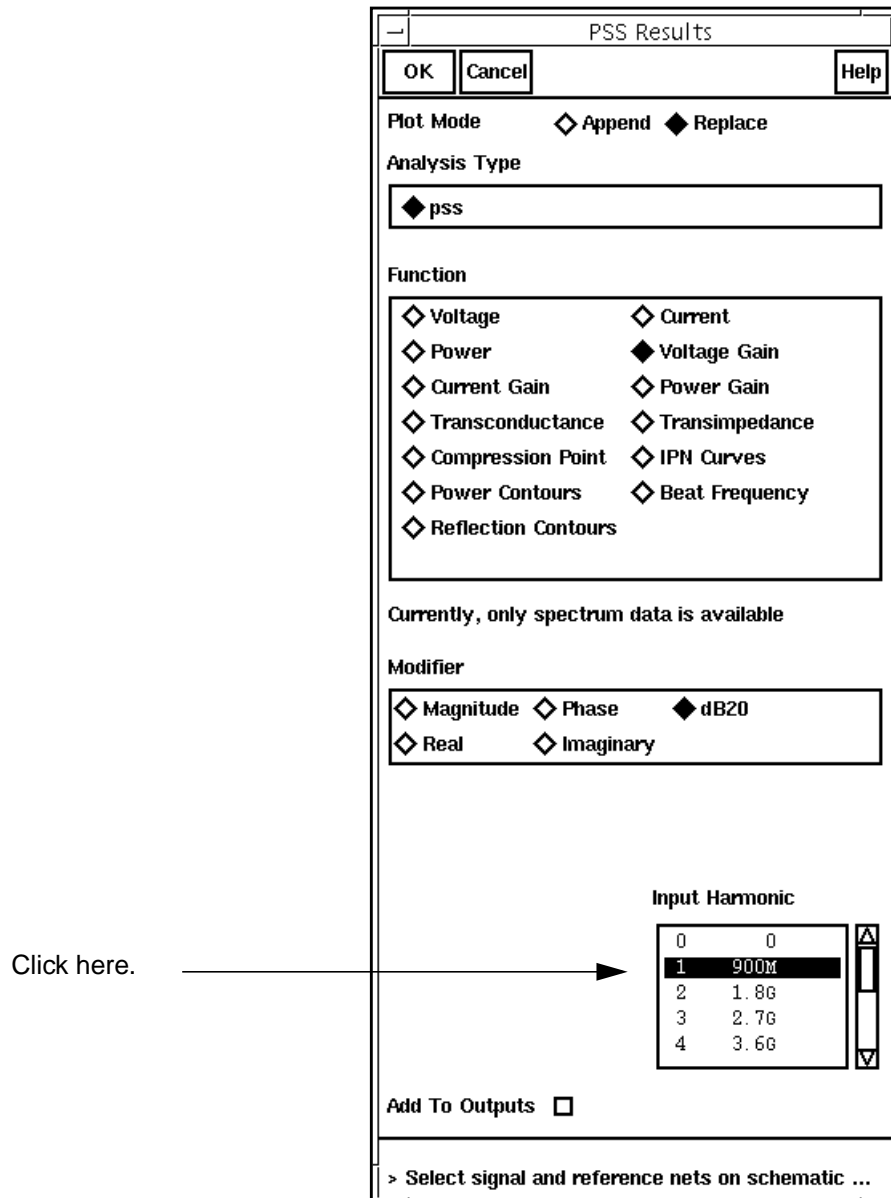
In the PSS Results form, do the following:

1. *Highlight Replace for Plot Mode.*
2. *Highlight pss for Analysis Type.*
3. *Highlight Voltage Gain for Function.*
4. *Highlight dB20 for Modifier.*
5. *Highlight 900M in the Reference Harmonic window.*

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.



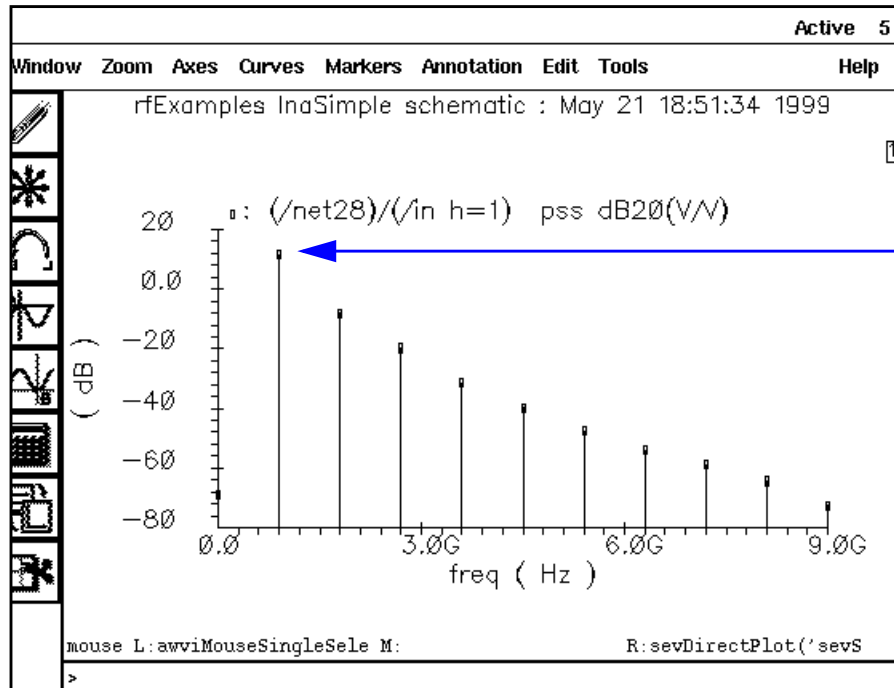
6. In the Schematic window, calculate the voltage gain as V_{out}/V_{in} by clicking first on the output and then on the input.

In the plot, place your cursor at the top of the line representing the 900M fundamental. The x and y axis values, which are displayed in the top left corner of the Waveform window, show that the voltage gain at 900MHz is 11.44.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

x, y values



Put cursor here.

Output Voltage Distribution

The output voltage distribution of an amplifier can be important in analyzing intermodulation distortion when the two RF input signals are close together. To model this situation, you modify a parameter to make the voltage source component generate two separate frequencies.

Editing Schematic

1. In the Schematic window, select the RF voltage source.
2. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears

3. In the Edit Object Properties form, do the following:
 - a. Highlight *Display second sinusoid*.

The form changes to let you specify data for a second sinusoid.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

- b. Type `fund2`, or any name that you choose, in the *Second Frequency name* field.
- c. Type `920M` as the *Frequency 2* value.
The amplitude of the second tone is set to a value equal to that of the first tone.
- d. Type `prf` in the *Amplitude 2 (dBm)* field.
- e. Click *OK*.

4. In the Schematic window, choose *Design – Check and Save*.

The filled-in form looks like this.

CDF Parameter	Value	Display
Resistance	40 Ohms	off <input type="checkbox"/>
Port number	1	off <input type="checkbox"/>
DC voltage		off <input type="checkbox"/>
Source type	sine <input type="checkbox"/>	off <input type="checkbox"/>
Frequency name 1	frf	off <input type="checkbox"/>
Frequency 1	900M Hz	off <input type="checkbox"/>
Amplitude 1 (Vpk)	V	off <input type="checkbox"/>
Amplitude 1 (dBm)	prf	off <input type="checkbox"/>
Phase for Sinusoid 1	0	off <input type="checkbox"/>
Sine DC level		off <input type="checkbox"/>
Delay time		off <input type="checkbox"/>
Display second sinusoid	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
Frequency name 2	fund2	off <input type="checkbox"/>
Frequency 2	920M Hz	off <input type="checkbox"/>
Amplitude 2 (Vpk)		off <input type="checkbox"/>
Amplitude 2 (dBm)	prf	off <input type="checkbox"/>
Phase for Sinusoid 2		off <input type="checkbox"/>

5. Be sure the full path to the model file is specified in the Model File Set-up form.
6. Disable any analyses you have run previously.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The Choosing Analyses form appears.

2. Click on *pss* for the *Analysis* choice.
3. In the *Fundamental Tones* list box, be sure the *Auto Calculate button is highlighted*.

The *Beat Frequency* field shows 20M because there are two RF tones at 900M and 920M.

The *Beat Frequency* button is highlighted by default.

In the Choosing Analyses form, do the following:

1. In the *Output harmonics* cyclic field, choose *Number of Harmonics* and type 60 in the *Number of Harmonics* field.

The largest common multiple of 900Mhz and 920Mhz is 20Mhz. Sixty harmonics extends the results out to 1.2 GHz, but you can choose another number if you want.

2. Highlight *moderate* as the *Accuracy Defaults* value

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1	frf	900M	900M	Moderate	RF
2	fund2	920M	920M	Moderate	RF

Moderate

Clear/Add Delete Update From Schematic

Beat Frequency Auto Calculate
 Beat Period

Output harmonics

Number of harmonics

Accuracy Defaults (errpreset)
 conservative moderate liberal
Additional Time for Stabilization (tstab)
Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep

Enabled Options...

3. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. To run the PSS analysis, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation completed successfully.

Plotting Output Voltage Distribution

- In the Simulation window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *Voltage* for *Function*.
3. Highlight *spectrum* for *Sweep* value.
4. Highlight *dB20* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

The screenshot shows the 'PSS Results' dialog box. At the top, there are three buttons: 'OK', 'Cancel', and 'Help'. Below these are several sections of options:

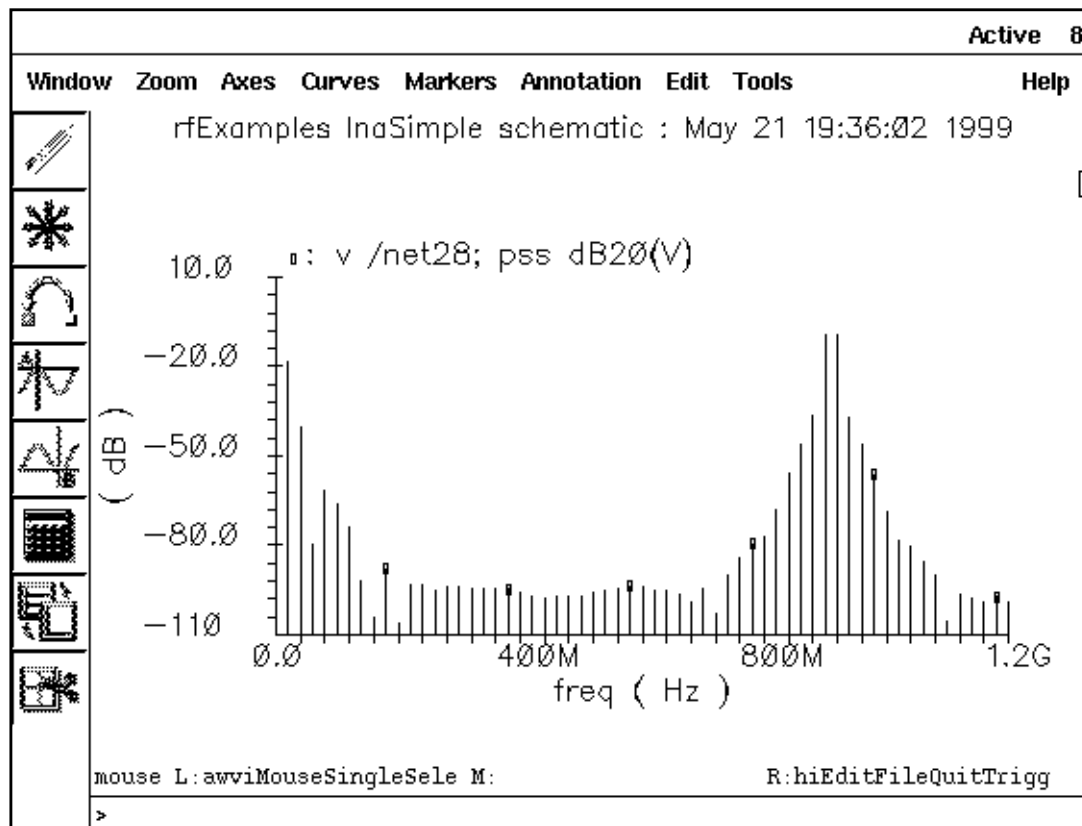
- Plot Mode:** Append Replace
- Analysis Type:** pss
- Function:** A list of functions with checkboxes:
 - Voltage
 - Current
 - Power
 - Voltage Gain
 - Current Gain
 - Power Gain
 - Transconductance
 - Transimpedance
 - Compression Point
 - IPN Curves
 - Power Contours
 - Beat Frequency
 - Reflection Contours
- Sweep:** spectrum time
- Modifier:** Magnitude Phase dB20
 Real Imaginary
- Buttons:** 'Add To Outputs' with an unchecked checkbox and a 'Replot' button.
- Footer:** '> Select net on schematic...'

5. In the Schematic window, click on the amplifier output.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

You can see spikes at 900MHz, 920MHz (-9.3 dB) and a third spike at 20MHz. The next strongest harmonics are lower than -36 dB.



S-Parameter Analysis for Low Noise Amplifiers

Before you can perform S-parameter analyses of the low noise amplifier, you need to disable the setups from the previous example.

1. Be sure the full path to the model file is specified in the [Model File Set-up](#) form.
2. Disable any analyses you have run previously.

Editing the Schematic

1. In the [Schematic window](#), select the RF voltage source.
2. In the [Schematic window](#), choose *Edit – Properties – Objects*.

The [Edit Object Properties](#) form appears.

3. In the Edit Object Properties form, be sure the *Source type* is set to `sine`, delete the *Frequency 2* value, if there is one, and click *OK*.

The input voltage source now has only one frequency at 900MHz. *Frequency2* is zero.

4. In the Schematic window, choose *Design – Check and Save*.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

2. In the Choosing Analyses form, click on *sp* for the *Analysis* choice.

The form changes to let you specify data for an S-parameter analysis.

3. In the Choosing Analyses form, do the following:

- a. Highlight *Frequency* for *Sweep Variable*.
- b. Highlight *Start-Stop* for *Sweep Range and type* `100M` and `1.2G` as the *Start* and *Stop* values, respectively.
- c. In the *Sweep Type* cyclic field, choose *Linear* and click on the *Number of Steps* button.
- d. Type `100` for the *Number of Steps* in the field.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

S-Parameter Analysis

Sweep Variable

Frequency
 Design Variable
 Temperature
 Component Parameter
 Model Parameter

Sweep Range

Start-Stop Start Stop
 Center-Span

Sweep Type

Linear Step Size
 Number of Steps

Add Specific Points

Do Noise

yes
 no


Enabled Options...

4. In the Choosing Analyses form, click *OK*.

Running Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.
The output log file appears and displays information about the simulation as it runs.
2. Look in the CIW for a message that says the simulation completed successfully.


Plotting S-Parameters

1. In the Simulation window, choose *Results – Direct Plot – S-Parameter*.
The Waveform window and the S-Parameter Results form appear.
2. In the Waveform window, click on the add subwindow icon 

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

A subwindow is added to the Waveform window,

3. To choose a subwindow for a plot, click on the small number in a rectangle at the top right corner of each subwindow. 

Note: You can also divide the Waveform window into horizontal strips by highlighting *Axes – To Strip* in the Waveform window.

To plot the S11 and S22 values, do the following in the S-Parameter Results form:

1. Highlight *SP* for *Function*.
2. Highlight *Rectangular* for *Plot Type*.
3. Highlight *db20* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

S-Parameter Results

Plot Mode Append Replace

Function

SP ZP YP HP
 GD VSWR NFmin Gmin
 Rn m NF Kf
 B1f GT GA GP
 Gmax Gmsg Gumx ZM
 NC GAC GPC LSB
 SSB

Description: S-Parameter

Plot Type Auto Rectangular
 Z-Smith Y-Smith
 Polar

Modifier

Magnitude Phase dB20
 Real Imaginary

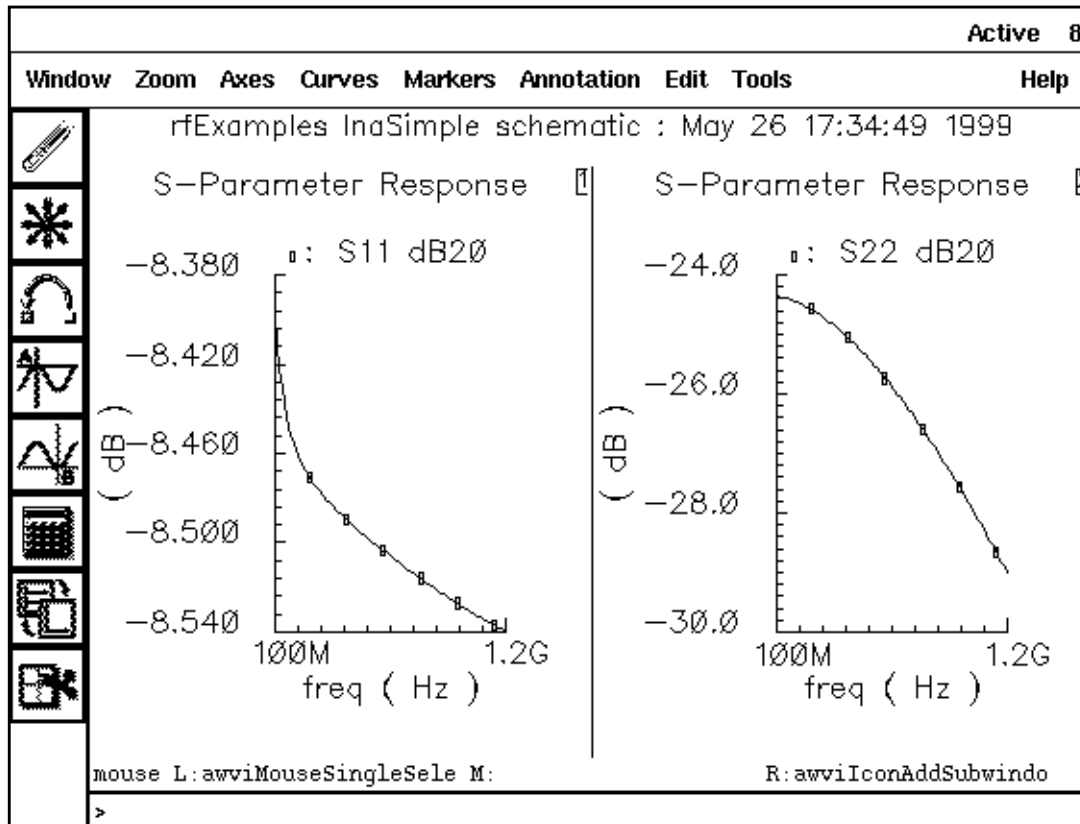
S11	S12
S21	S22

4. In the lower left corner of the S-Parameter Results form, click on the appropriate buttons.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The S11 and S22 parameters are plotted

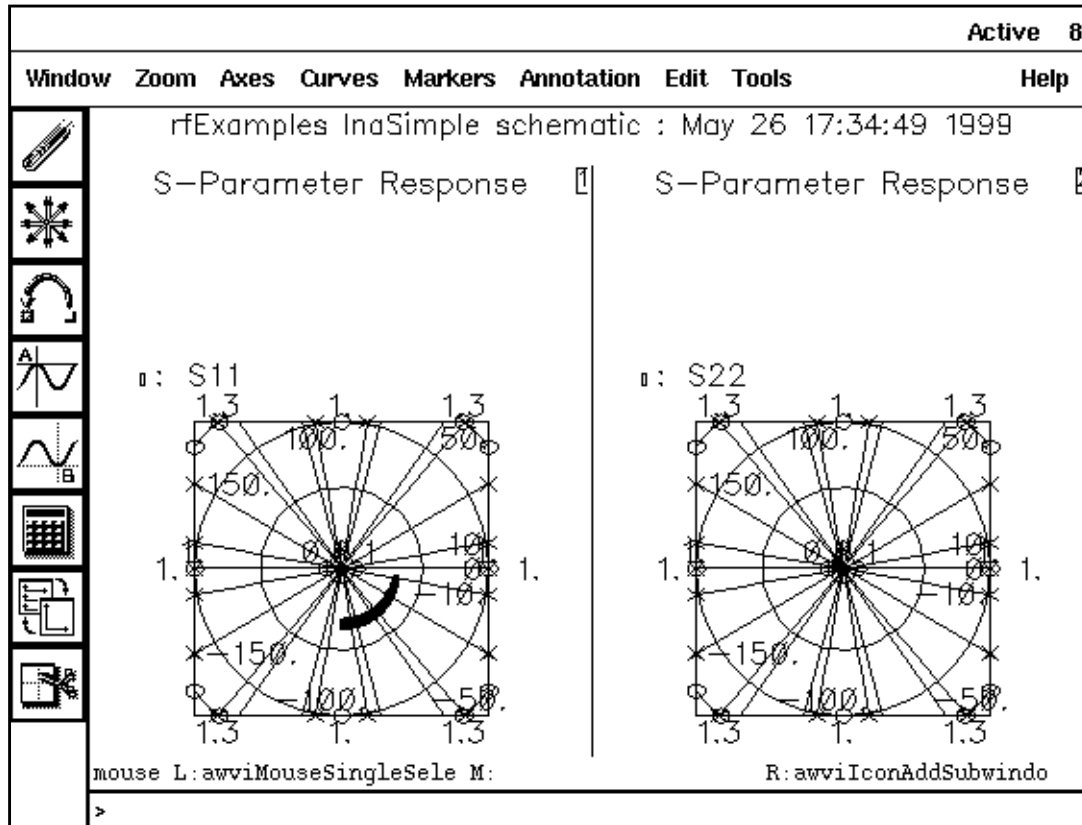


The values of S11 and S22 measure whether input and output are matched.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

If you want a different plot type, you can choose among several in the [S-Parameter Results form](#). The following plots show S11 and S22 with the *Plot Type* set to *Polar*.



You can also choose S12 and S21 for plotting in the [S-Parameter Results form](#). S12 is the reverse gain and a measure of isolation.

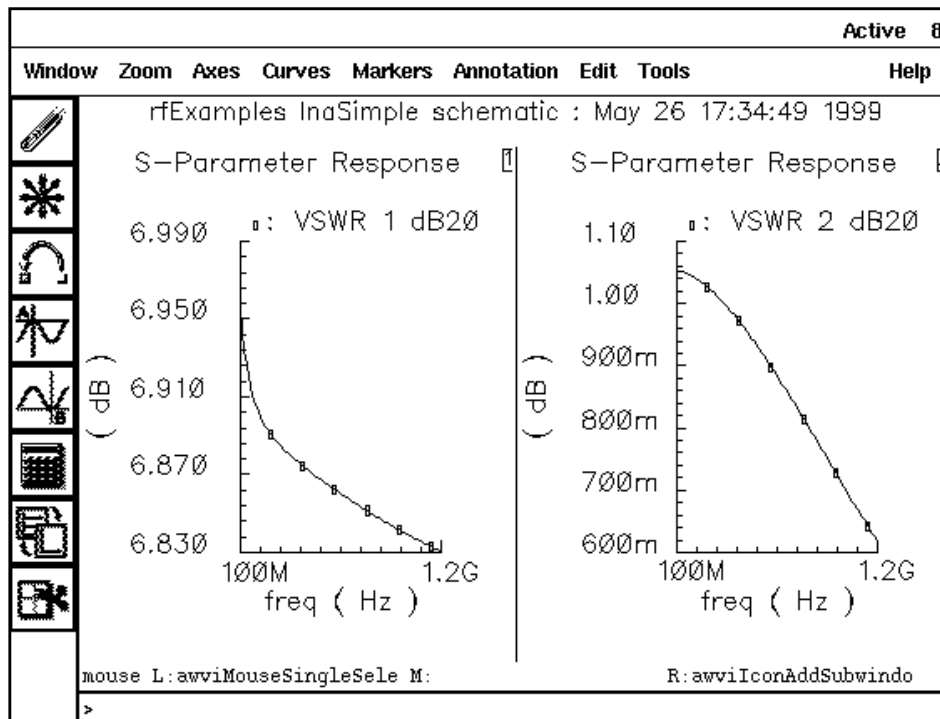
Plotting the Voltage Standing Wave Ratio

In the [S-Parameter Results form](#), do the following:

1. Highlight on *Replace* for *Plot Mode*.
2. Highlight *VSWR* for *Function*.
3. Highlight *Rectangular* for *Plot Type*.

4. Highlight *dB20* for *Modifier*.
5. Click on the *VSWR1* or *VSWR2* button.

The plot appears like the one below.



Linear Two-Port Noise Analysis

1. Be sure the full path to the model file is specified in the Model File Set-up form.
2. Disable any analyses you have run previously.

Editing the Schematic

1. In the Schematic window, select the RF source.
2. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears.

3. In the Edit Object Properties form, choose *dc* for the *Source type* and click *OK*.

Note: *isnoisy* is a parameter you can set on the Edit Object Properties form that you might want to be aware of. *isnoisy* specifies whether a resistor generates noise. If you set the

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

parameter to *YES* or *NO*, the parameter setting appears in the netlist. If you leave the parameter unspecified the blank value in the cyclic field), the default setting is *YES* but the parameter does not appear in the netlist.

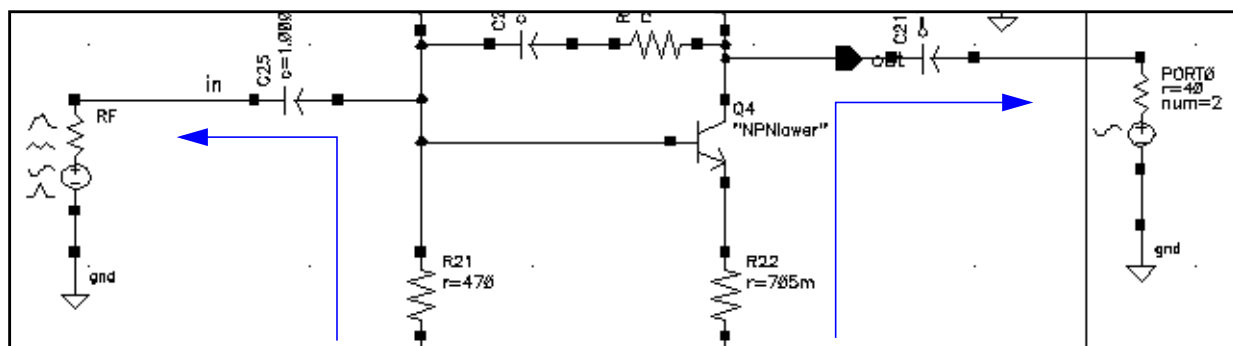
4. In the Schematic window, choose *Design – Check and Save*.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.
2. In the Choosing Analyses form, click on *sp* for the *Analysis* choice.
3. Highlight *Frequency* for the *Sweep Variable*.
4. Highlight *Start-Stop* for the *Sweep Range* type. Type 800M and 1.1G as the *Start* and *Stop* values, respectively.
5. In the *Sweep Type* cyclic field, choose *Linear for the sweep type* and click on the *Number of Steps* button. Type 30 for the *Number of Steps* in the field.
6. Highlight *yes* for *Do Noise*.

The form changes to let you specify values for the *Output port* and *Input port* fields.

7. For both the *Output port field* and the *Input port fields*, click on the *Select* button for each field and then click on each port in the Schematic window.



Click on input port.

Click on output port.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The values for the fields are entered automatically.

Do Noise	Output port	<input type="text" value="/PORT0"/>	<input type="button" value="Select"/>
<input checked="" type="checkbox"/> yes	Input port	<input type="text" value="/RF"/>	<input type="button" value="Select"/>
<input type="checkbox"/> no			

The filled-in form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

S-Parameter Analysis

Sweep Variable

Frequency
 Design Variable
 Temperature
 Component Parameter
 Model Parameter

Sweep Range

Start-Stop Start Stop
 Center-Span

Sweep Type

Linear Step Size
 Number of Steps

Add Specific Points

Do Noise

yes no

Output port
Input port

Enabled

8. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation is completed successfully.

Plotting the Noise Figure and Minimum Noise Figure

- In the Simulation window, choose *Results – Direct Plot – S-Parameter*.

The Waveform window and the S-Parameter Results form appear.

In the S-Parameter Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *NF* for *Function*.
3. Highlight *Rectangular* for *Plot Type*.
4. Highlight *dB10* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The completed form looks like this.

S-Parameter Results

OK Cancel Help

Plot Mode Append Replace

Function

<input type="radio"/> SP	<input type="radio"/> ZP	<input type="radio"/> YP	<input type="radio"/> HP
<input type="radio"/> GD	<input type="radio"/> VSWR	<input type="radio"/> NFmin	<input type="radio"/> Gmin
<input type="radio"/> Rn	<input type="radio"/> m	<input checked="" type="radio"/> NF	<input type="radio"/> Kf
<input type="radio"/> B1f	<input type="radio"/> GT	<input type="radio"/> GA	<input type="radio"/> GP
<input type="radio"/> Gmax	<input type="radio"/> Gmsg	<input type="radio"/> Gumx	<input type="radio"/> ZM
<input type="radio"/> NC	<input type="radio"/> GAC	<input type="radio"/> GPC	<input type="radio"/> LSB
<input type="radio"/> SSB			

Description: Noise Factor

Plot Type Rectangular

Modifier

<input type="radio"/> Magnitude	<input checked="" type="radio"/> dB10
---------------------------------	---------------------------------------

Plot

Add To Outputs

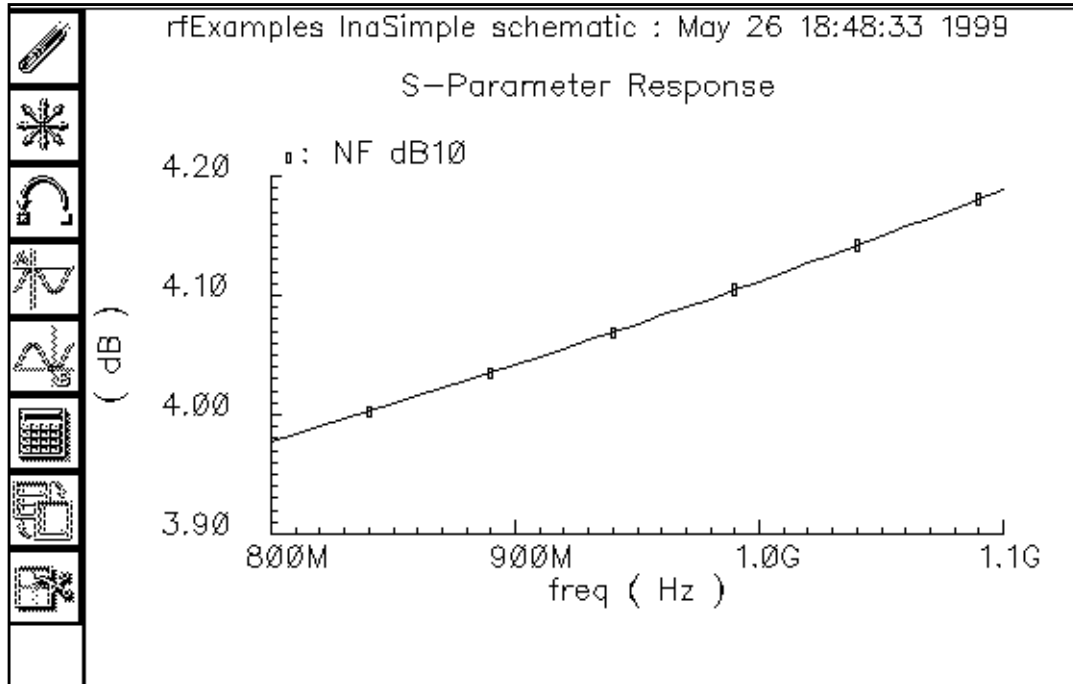
> Press "Plot" button to plot...

1. In the S-Parameter Results form, click on *Plot*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

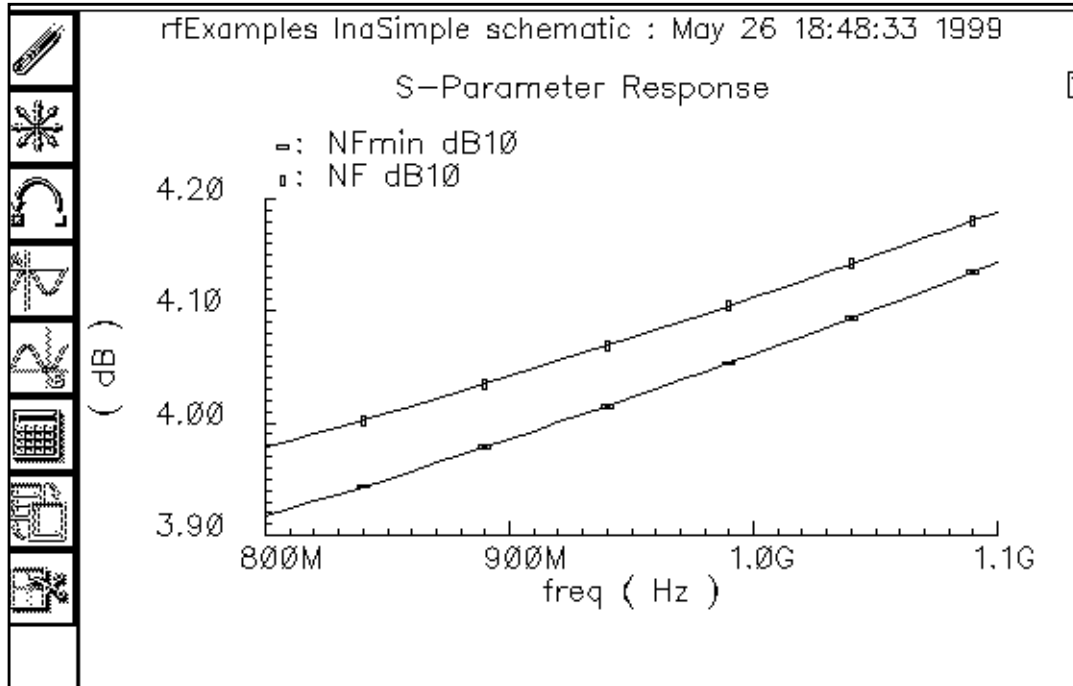
The plot for the noise figure appears in the Waveform window.



In the S-Parameter Results form, do the following:

1. Highlight *Append* for *Plot Mode*.
2. Highlight *NFmin* for *Function*.
3. Click on *Plot*.

The plot for the minimum noise figure is added to the noise figure plot.



Plotting the Equivalent Noise Resistance

In the S-Parameter Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *Rn* for *Function*.

Affirma RF Simulator (SpectreRF) User Guide

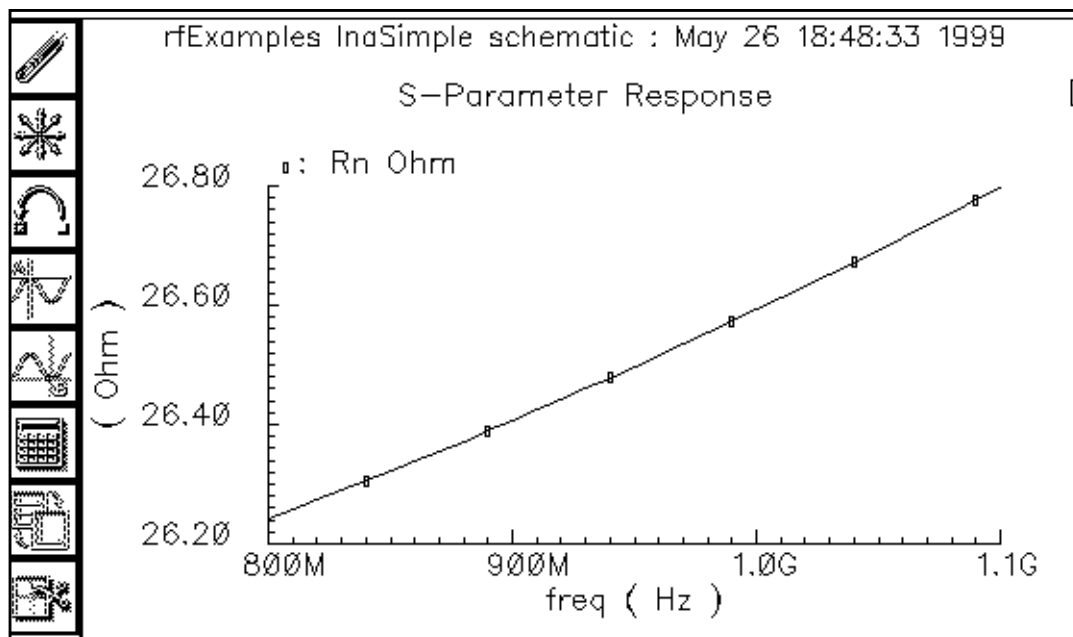
Simulating Low-Noise Amplifiers

The completed form looks like this.

OK	Cancel	Help
Plot Mode <input type="checkbox"/> Append <input checked="" type="checkbox"/> Replace		
Function		
<input type="checkbox"/> SP	<input type="checkbox"/> ZP	<input type="checkbox"/> YP
<input type="checkbox"/> GD	<input type="checkbox"/> VSWR	<input type="checkbox"/> NFmin
<input checked="" type="checkbox"/> Rn	<input type="checkbox"/> m	<input type="checkbox"/> NF
<input type="checkbox"/> B1f	<input type="checkbox"/> GT	<input type="checkbox"/> GA
<input type="checkbox"/> Gmax	<input type="checkbox"/> Gmsg	<input type="checkbox"/> Gumx
<input type="checkbox"/> NC	<input type="checkbox"/> GAC	<input type="checkbox"/> GPC
<input type="checkbox"/> SSB	<input type="checkbox"/> HP	<input type="checkbox"/> Gmin
<input type="checkbox"/> Kf	<input type="checkbox"/> GP	<input type="checkbox"/> ZM
<input type="checkbox"/> LSB		
Description: Equivalent Noise Resistance		

3. In the S-Parameter Results form, click on *Plot*.

The equivalent noise resistance is plotted in the Waveform window.



Plotting Load and Source Stability Circles

Load and source stability circles show the boundaries between values of load and source impedance that cause instability and values that do not. Either the inside or the outside of the

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

circles might represent the stable region, depending on other values. If, as in the example below, the values of S-parameters S_{11} and S_{22} are less than one for a 50 Ohm impedance system, the center of the normalized Smith chart falls within the stable region.

To Plot the Load Stability Circles

In the S-Parameter Results form, do the following:

1. Highlight *Append* for *Plot Mode*.
2. Highlight *LSB* for *Function*.
3. Highlight *Z-Smith* for *Plot Type*.
4. For the *Frequency Range (Hz)* values, type 800M for the *Start* value, 1G for the *Stop* value, and 100M for the *Step* value.

The completed form looks like this.

The screenshot shows the 'S-Parameter Results' dialog box with the following settings:

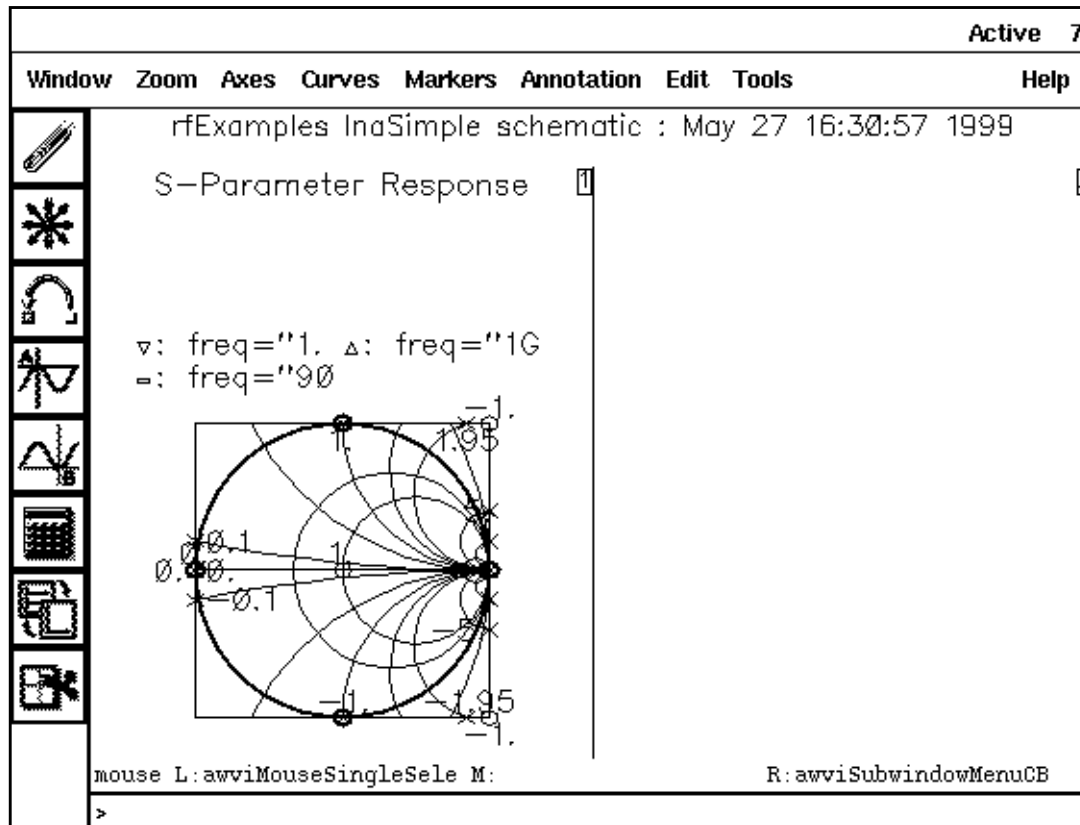
- Plot Mode:** Append Replace
- Function:** A list of functions with LSB selected. Other functions include SP, ZP, YP, HP, GD, VSWR, NFmin, Gmin, Rn, m, NF, Kf, B1f, GT, GA, GP, Gmax, Gmsg, Gumx, ZM, NC, GAC, GPC, SSB.
- Description:** Load Stability Circles
- Plot Type:** Z-Smith Y-Smith
- Frequency Range (Hz):**
 - Start: 800M
 - Stop: 1G
 - Step: 100M
- Plot:** A button to execute the plot.
- Add To Outputs:**
- Footer:** > Press "Plot" button to plot...

5. In the S-Parameter Results form, click on *Plot*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The load stability circles are plotted, but they are outside the range of the graph. If you want to see them at this point, choose *Zoom – Fast Zoom Out* several times in the Waveform window.



To Plot the Source Stability Circles

In the S-Parameter Results form, do the following:

1. Highlight *Append* for *Plot Mode*.
2. Highlight *SSB* for *Function*.
3. Highlight *Z-Smith* for *Plot Type*.
4. For the *Frequency Range (Hz)* values, type 800M for the *Start* value, 1G for the *Stop* value, and 100M for the *Step* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The completed form looks like this.

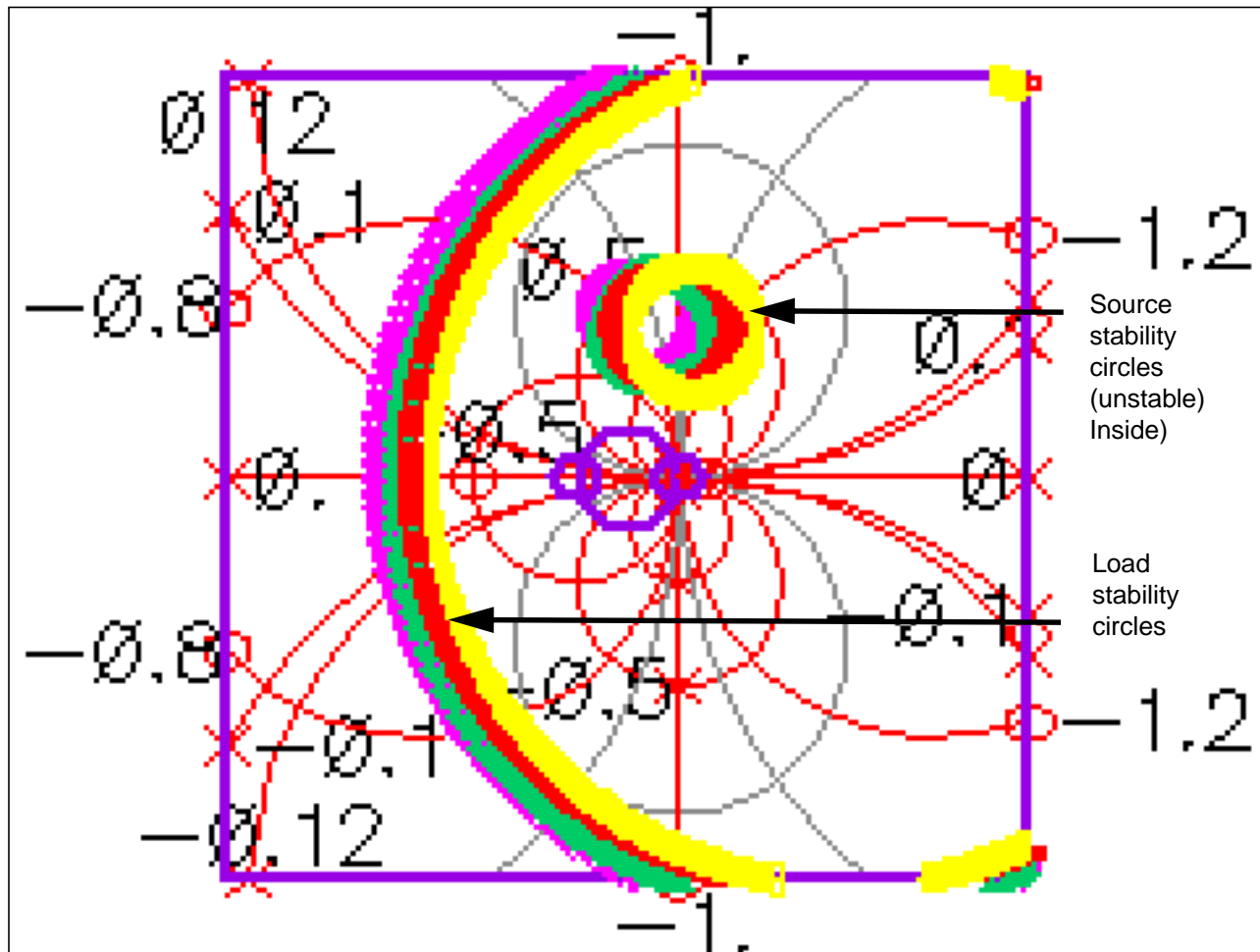
The screenshot shows the 'S-Parameter Results' dialog box. It has a title bar with 'S-Parameter Results' and buttons for 'OK', 'Cancel', and 'Help'. The 'Plot Mode' section has 'Append' selected with a diamond icon and 'Replace' with an empty diamond icon. The 'Function' section contains a grid of options: SP, ZP, YP, HP, GD, VSWR, NFmin, Gmin, Rn, m, NF, Kf, B1f, GT, GA, GP, Gmax, Gmsg, Gumx, ZM, NC, GAC, GPC, LSB, and SSB (selected with a diamond icon). Below the grid is the text 'Description: Source Stability Circles'. The 'Plot Type' section has 'Z-Smith' selected with a diamond icon and 'Y-Smith' with an empty diamond icon. The 'Frequency Range (Hz)' section has input fields for 'Start' (800M), 'Stop' (1.1G), and 'Step' (100M). A 'Plot' button is located below these fields. At the bottom, there is an 'Add To Outputs' checkbox (unchecked) and a note: '> Press "Plot" button to plot...'

5. In the S-Parameter Results form, click on *Plot*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The source stability circles are plotted in the Waveform window along with the load stability circles. To see the plot as it appears below, you must choose *Zoom – Fast Zoom Out* three times consecutively in the Waveform window.



In the center of the Smith Chart, the values of S_{11} and S_{22} are greater than 1, so the center of the Smith chart is part of the unstable region. Therefore, all points inside the load stability circle and all points outside the source stability circle belong to the unstable region.

Plotting the Noise Circles

In the S-Parameter Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *NC* for *Function*.

3. Highlight *Z-Smith* for *Plot Type*.
4. Highlight *Noise Level (dB)* for the *Sweep* value.
5. Type 900M for the *Frequency (Hz)* value.
6. For the *Level Range (dB)* values, type -30 for *Start*, 30 for *Stop*, and 5 for *Step*.

The completed form looks like this.

S-Parameter Results

OK Cancel Help

Plot Mode Append Replace

Function

<input type="radio"/> SP	<input type="radio"/> ZP	<input type="radio"/> YP	<input type="radio"/> HP
<input type="radio"/> GD	<input type="radio"/> VSWR	<input type="radio"/> NFmin	<input type="radio"/> Gmin
<input type="radio"/> Rn	<input type="radio"/> m	<input type="radio"/> NF	<input type="radio"/> Kf
<input type="radio"/> B1f	<input type="radio"/> GT	<input type="radio"/> GA	<input type="radio"/> GP
<input type="radio"/> Gmax	<input type="radio"/> Gmsg	<input type="radio"/> Gumx	<input type="radio"/> ZM
<input checked="" type="radio"/> NC	<input type="radio"/> GAC	<input type="radio"/> GPC	<input type="radio"/> LSB
<input type="radio"/> SSB			

Description: Noise Circles

Plot Type Z-Smith Y-Smith

Sweep frequency Noise Level (dB)

Frequency (Hz)

Level Range (dB)

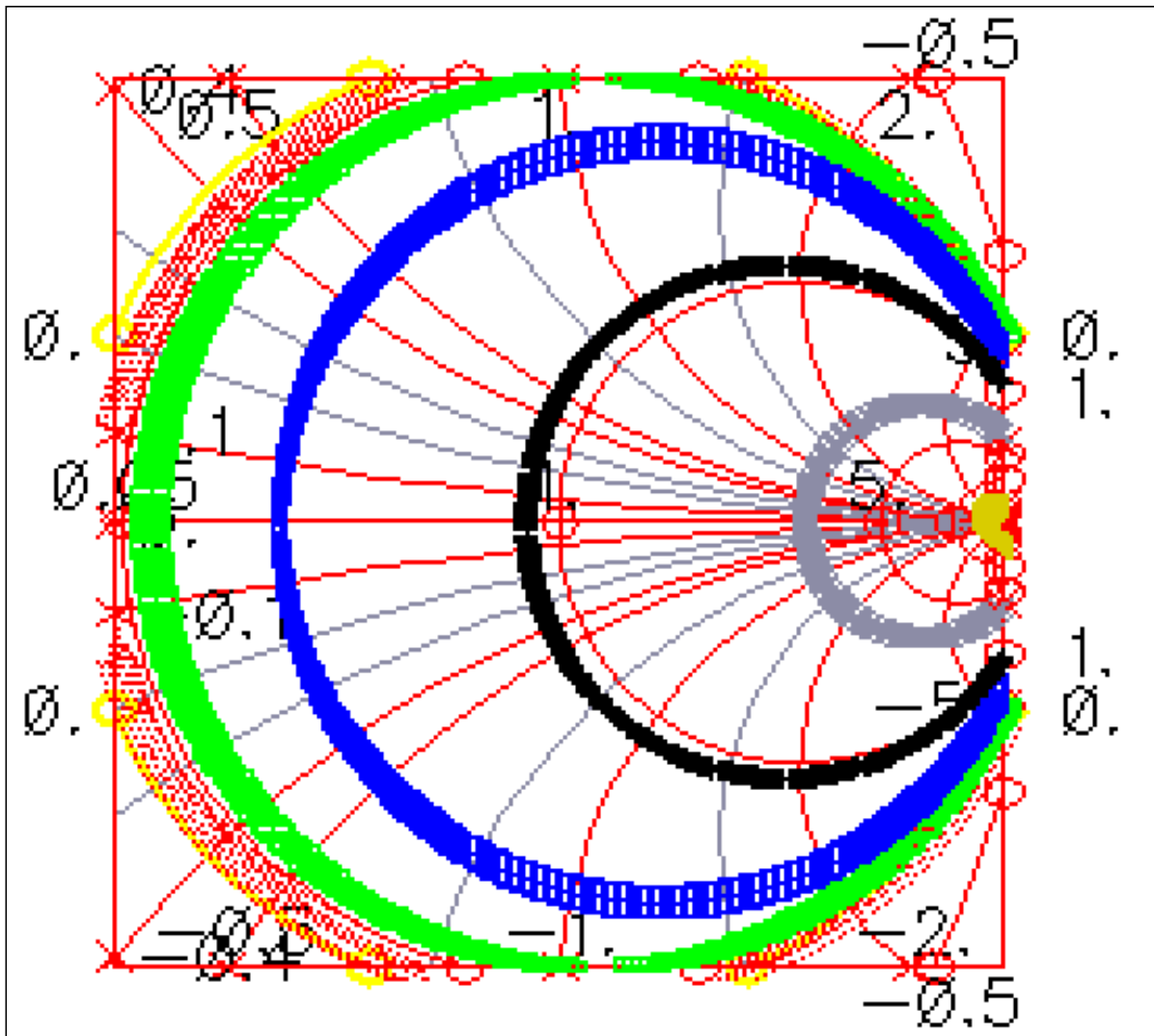
Start	<input type="text" value="-30"/>	Stop	<input type="text" value="30"/>
Step	<input type="text" value="5"/>		

Add To Outputs

> Press "Plot" button to plot...

7. In the *S-Parameter Results* form, click on *Plot*.

The noise circles are plotted in the Waveform window. By choosing the *Zoom – Fast Zoom Out* feature of the Waveform window, you can see the plot as it appears below.



Noise Calculations Using Pnoise

1. Specify the full path to the model file in the Model File Set-up form.
2. Disable any analyses you have run previously.

Editing the Schematic

1. In the Schematic window, choose *Edit – Properties – Objects*.
2. In the Schematic window, click on the *RFvoltage* source.
The Edit Object Properties form appears.
3. In the Edit Object Properties form, choose *dc* for the *Source Type* and then click *OK*.
4. In the Schematic window, choose *Design – Check and Save*.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.
The Choosing Analyses form appears.
2. In the Choosing Analyses form, click on *pss*.
3. Type *900M* for the *Beat Frequency*.
With the *dc source type*, you must enter the fundamental frequency manually.
Be sure the *Auto Calculate* button is not highlighted.
4. The *Beat Frequency* button is highlighted by default.
5. In the *Output harmonics* cyclic field, choose *Number of harmonics* for the *Output harmonics* choice, and type in a reasonable value, such as 20, for the number of harmonics.
6. Highlight *conservative* for the *Accuracy Defaults (errpreset)* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The completed form looks like this. (You must scroll to see the whole form.)

The screenshot shows the 'Choosing Analyses' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. Below these are several analysis options, each with a diamond-shaped selection icon. The 'pnoise' option is selected. The 'Periodic Steady State Analysis' section is expanded, showing a table for 'Fundamental Tones' with columns for '#', 'Name', 'Expr', 'Value', 'Signal', and 'SrcId'. Below the table are several input fields and buttons, including 'Moderate', 'Clear/Add', 'Delete', and 'Update From Schematic'. The 'Beat Frequency' is set to 900M and 'Auto Calculate' is unchecked. The 'Number of harmonics' is set to 20. At the bottom, there are sections for 'Accuracy Defaults (empreset)' with radio buttons for 'conservative', 'moderate', and 'liberal', and a section for 'Oscillator' and 'Sweep'.

7. At the top of the Choosing Analyses form, click on *pnoise*.

The form changes to let you specify data for the periodic noise analysis.

In the periodic noise analysis section of the Choosing Analyses form, do the following:

1. In the *Frequency Sweep Range (Hz)* cyclic field, choose *Start – Stop* for the *Frequency Sweep Range (Hz)* value. Type 800M and 1.2G as the *Start* and *Stop* values, respectively.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

2. In the *Sweep Type* cyclic field, choose *Linear for the sweep type* and click on the *Number of Steps* button. Type 201 as the number of steps.
3. In the *Sidebands* cyclic field choose *Maximum sideband* and type 20 for the *Maximum sideband* value in the field.
4. Choose *voltage* for the *Output*.

The value is appropriate because the reference sideband is an LNA.

5. Click on the *Positive Output Node Select* button, and then click in the schematic to choose the amplifier output.
6. Leave the *Negative Output Node* field empty. It defaults to /gnd!

You can set the *Negative Output Node* to a different value by clicking on the *Negative Output Node Select* button and then clicking on the output node in the schematic.

7. In the *Input Voltage Source* cyclic field, choose *voltage*.
8. Click on the *Input Source Select* button, and then click in the schematic on the RF voltage source.
9. Type 0 for the *Reference Side-Band* value.

The reference sideband is specified as 0 because the circuit is an LNA, and there is no frequency conversion from the input to the output.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The completed noise section of the form looks like this.

The screenshot shows the 'Choosing Analyses' dialog box for 'Periodic Noise Analysis'. The dialog has a title bar that reads 'Choosing Analyses -- Affirma Analog Circuit Design Envir'. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. The main content area is divided into several sections:

- Periodic Noise Analysis**:
 - PSS Beat Frequency (Hz): 900M
 - Sweeptype:
 - Frequency Sweep Range (Hz):
 - Start-Stop: Start: 800M Stop: 1.2G
 - Sweep Type:
 - Linear:
 - Step Size: 20M
 - Number of Steps:
 - Add Specific Points:
- Sidebands**:
 - Maximum sideband: 20M
- Output**:
 - Output: voltage
 - Positive Output Node: /net28M Select
 - Negative Output Node: Select
- Input Source**:
 - Input Source: voltage
 - Input Voltage Source: /RF1M Select
- Reference Side-Band:
- Noise Type**:
 - Noise Type: sources

At the bottom, there is an 'Enabled' checkbox which is checked, and an 'Options...' button.

10. In the Choosing Analyses form, click OK.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation is completed successfully.

Plotting the Noise Calculations

- In the Simulation window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pnoise* for *Analysis Type*.
3. Highlight *Noise Figure* for *Function*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

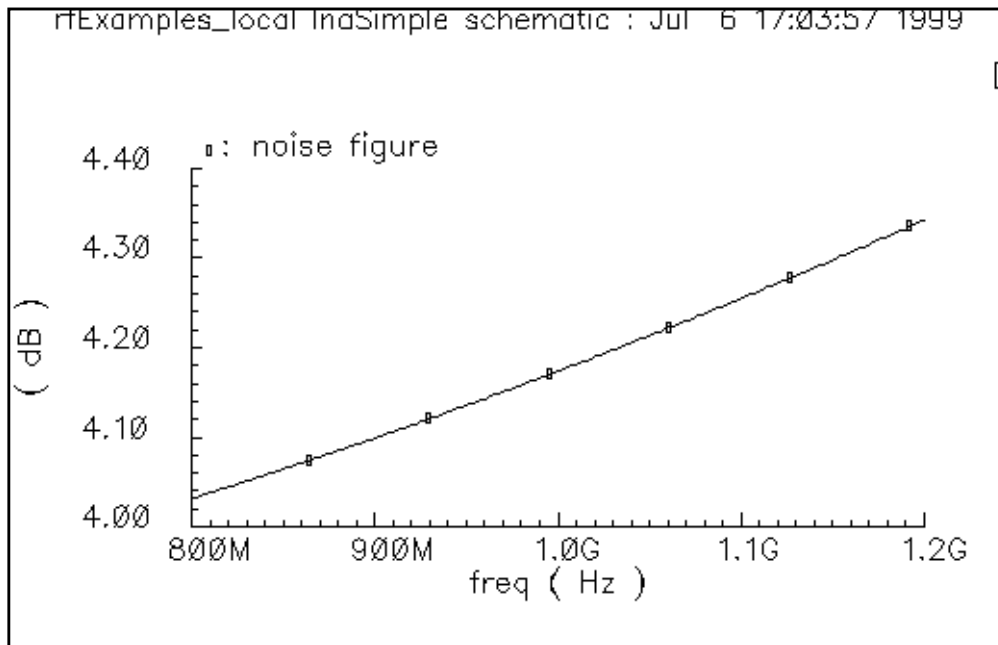
The screenshot shows a dialog box titled "PSS Results". At the top, there are three buttons: "OK", "Cancel", and "Help". Below the buttons, the "Plot Mode" is set to "Append" (indicated by a diamond symbol) and "Replace" (indicated by a filled diamond symbol). The "Analysis Type" is set to "pss" (diamond) and "pnoise" (filled diamond). The "Function" section contains a list of options: "Output Noise" (diamond), "Input Noise" (diamond), "Noise Figure" (filled diamond), "Noise Factor" (diamond), "Transfer Function" (diamond), "Phase Noise" (diamond), and "Pss Beat Frequency" (diamond). Below the function list, a message states "Currently, only frequency data is available". At the bottom left, there is a checkbox labeled "Add To Outputs" which is currently unchecked. At the bottom right, there is a "Plot" button.

4. in the PSS Results form, click on *Plot*

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The plot appears in the Waveform window. The Noise Figure at 900MHz is about 4.10 dB.



5. Close the PSS Results form and the Waveform window.
6. In the Simulation window, choose *Results – Print – PSS Noise Summary*.
The Noise Summary form appears.

In the Noise Summary form, do the following:

1. Highlight *spot noise* for *Type*.
2. Type 900M for *Frequency Spot (Hz)*.
3. Click on *All Types* for the *include* value of the *FILTER* list box.


The *include* value for the *FILTER* list box is *All Types* by default, however, you must click on the button to activate the choice.

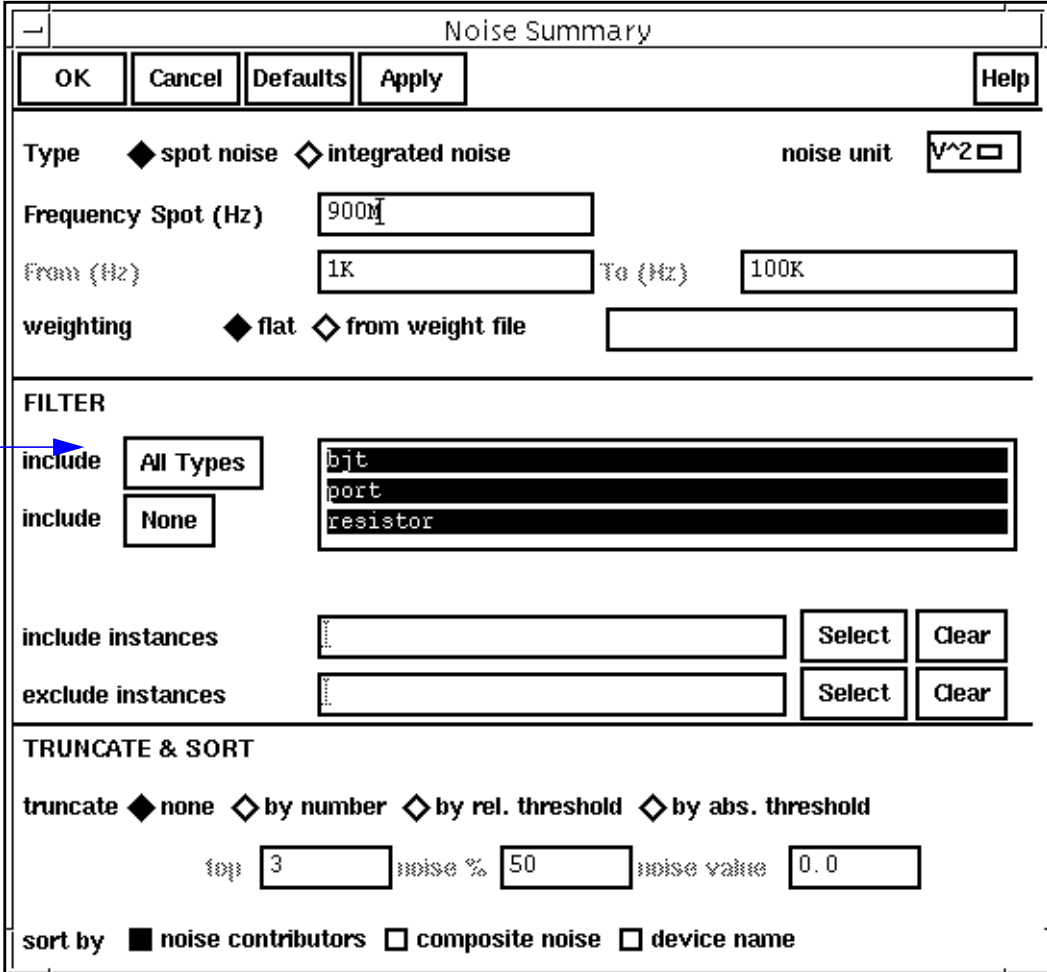
4. Highlight *none* for the truncate value in the *TRUNCATE & SORT* section.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

Click here. 



Noise Summary

OK Cancel Defaults Apply Help

Type spot noise integrated noise noise unit

Frequency Spot (Hz)

From (Hz) To (Hz)

weighting flat from weight file

FILTER

include

include instances

exclude instances

TRUNCATE & SORT

truncate none by number by rel. threshold by abs. threshold

top noise % noise value

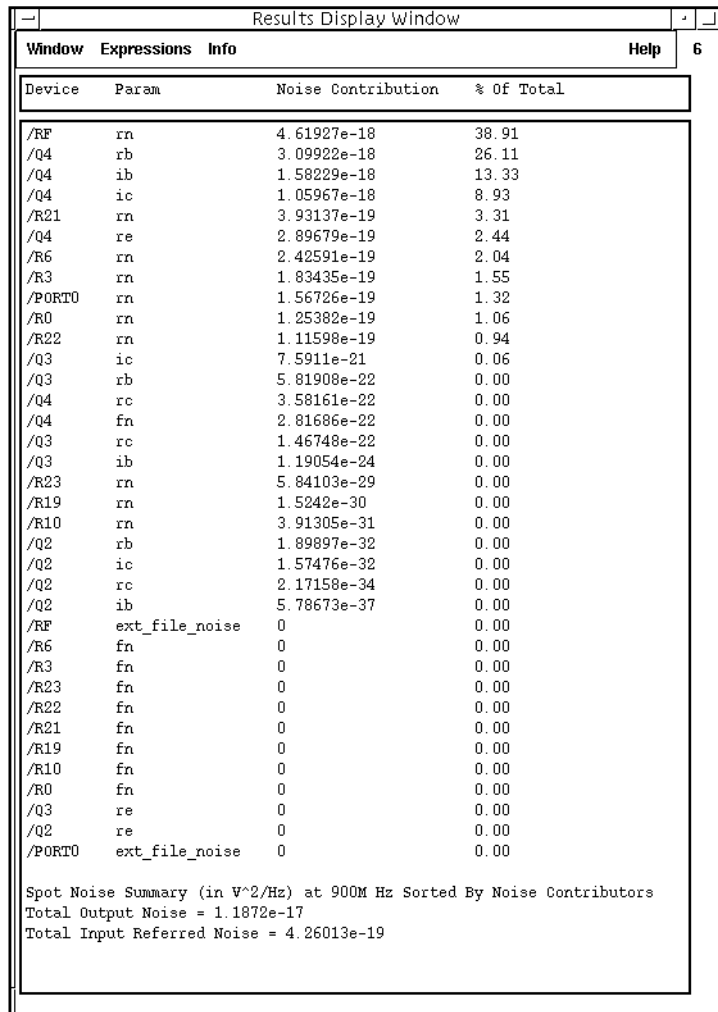
sort by noise contributors composite noise device name

5. In the Noise Summary form, click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The noise contributors now appear in a new window.



The screenshot shows a window titled "Results Display Window" with a menu bar containing "Window", "Expressions", "Info", "Help", and a page number "6". The main content is a table with four columns: "Device", "Param", "Noise Contribution", and "% Of Total". The table lists various components and their noise contributions, sorted by the percentage of total noise. At the bottom of the window, there is a "Spot Noise Summary" section.

Device	Param	Noise Contribution	% Of Total
/RF	rn	4.61927e-18	38.91
/Q4	rb	3.09922e-18	26.11
/Q4	ib	1.58229e-18	13.33
/Q4	ic	1.05967e-18	8.93
/R21	rn	3.93137e-19	3.31
/Q4	re	2.89679e-19	2.44
/R6	rn	2.42591e-19	2.04
/R3	rn	1.83435e-19	1.55
/PORT0	rn	1.56726e-19	1.32
/R0	rn	1.25382e-19	1.06
/R22	rn	1.11598e-19	0.94
/Q3	ic	7.5911e-21	0.06
/Q3	rb	5.81908e-22	0.00
/Q4	rc	3.58161e-22	0.00
/Q4	fn	2.81686e-22	0.00
/Q3	rc	1.46748e-22	0.00
/Q3	ib	1.19054e-24	0.00
/R23	rn	5.84103e-29	0.00
/R19	rn	1.5242e-30	0.00
/R10	rn	3.91305e-31	0.00
/Q2	rb	1.89897e-32	0.00
/Q2	ic	1.57476e-32	0.00
/Q2	rc	2.17158e-34	0.00
/Q2	ib	5.78673e-37	0.00
/RF	ext_file_noise	0	0.00
/R6	fn	0	0.00
/R3	fn	0	0.00
/R23	fn	0	0.00
/R22	fn	0	0.00
/R21	fn	0	0.00
/R19	fn	0	0.00
/R10	fn	0	0.00
/R0	fn	0	0.00
/Q3	re	0	0.00
/Q2	re	0	0.00
/PORT0	ext_file_noise	0	0.00

Spot Noise Summary (in V²/Hz) at 900M Hz Sorted By Noise Contributors
Total Output Noise = 1.1872e-17
Total Input Referred Noise = 4.26013e-19

This summary helps you determine the percentage of noise contributed by the different devices in the circuit.

Plotting the 1dB Compression Point

1. Specify the full path to the model file in the Model File Set-up form.
2. Disable any analyses you have run previously.

Editing the Schematic

- Click on the input voltage source (instance RF) and then choose *Edit – Properties – Objects* in the Schematic window.

The Edit Object Properties form appears.

In the Edit Object Properties form, do the following:

1. Choose *sine* for the *Source Type*.
2. Type 900M for the *Frequency* value.
3. Type prf for the *Amplitude (dBm)* value.
4. Be sure the *Amplitude* field is empty.
5. In the Edit Objects Properties form, click *OK*.
6. In the Schematic window, choose *Design – Check and Save*.

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

2. In the Choosing Analyses form, click on *pss*.

In the Choosing Analyses form, do the following:

1. Type 100M for the *Fundamental Frequency*.

The *Beat Frequency* button is highlighted by default.

Be sure the *Auto Calculate* button is not highlighted.

2. In the *Output harmonics* cyclic field, choose *Number of harmonics* for the *Output harmonics* choice and type in a reasonable value, such as 10, in the number of harmonics field.
3. Highlight *moderate* for the *Accuracy Default (errpreset)*.
4. Highlight the *Sweep* button.
The form changes to let you specify data for the sweep.
5. In the *Sweep* cyclic field, choose *Variable*.
6. Click on the *Select Design Variable* button.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The Select Design Variable form appears.

7. In the Select Design variable form, highlight *prf* and click *OK*.

prf is the amplitude, in dBm, of the RF input source. If you are not sure of this, use the Edit Object Properties form to examine the *Amplitude (dBm)* field for the RF voltage source.

8. Choose *Start-Stop for the Sweep range and* type -30 and 10 as the *Start* and *Stop* values, respectively, in the fields.
9. Choose *Linear* for the *Sweep Type* and click on the *Number of Steps* button. Type 10 as the number of steps in the field.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form appears like the one below. (You must use the scroll bar to see the whole form.)

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
1	frf	900M	900M	Moderate	RF

Moderate

Clear/Add Delete Update From Schematic

Beat Frequency 100M Auto Calculate

Beat Period

Output harmonics

Number of harmonics 10

Accuracy Defaults (errpreset)
 conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep Frequency Variable? no yes

10. In the Choosing Analyses form, click *OK*.

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation is completed successfully.

Plotting the 1dB Compression Point

- In the Simulation window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear.

In the PSS Results form, do the following:

1. Highlight *pss* for *Analysis Type*.
2. Highlight *Compression Point* for *Function*.
3. Type *-25* for the *Extrapolation Point*.

You learn what value to use for the extrapolation point through experience. If you do not specify a value, the plot defaults to the minimum variable value.

4. Choose *Input Referred 1dB Compression*.
5. Click on *900M* in the *1st Order Harmonic* list box.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves
 Power Contours Beat Frequency
 Reflection Contours

Gain Compression (dB)

"prf" ranges from -30 to 10

Extrapolation Point (dBm)

Input Referred 1dB Compression

1st Order Harmonic

6	600M
7	700M
8	800M
9	900M
10	1G

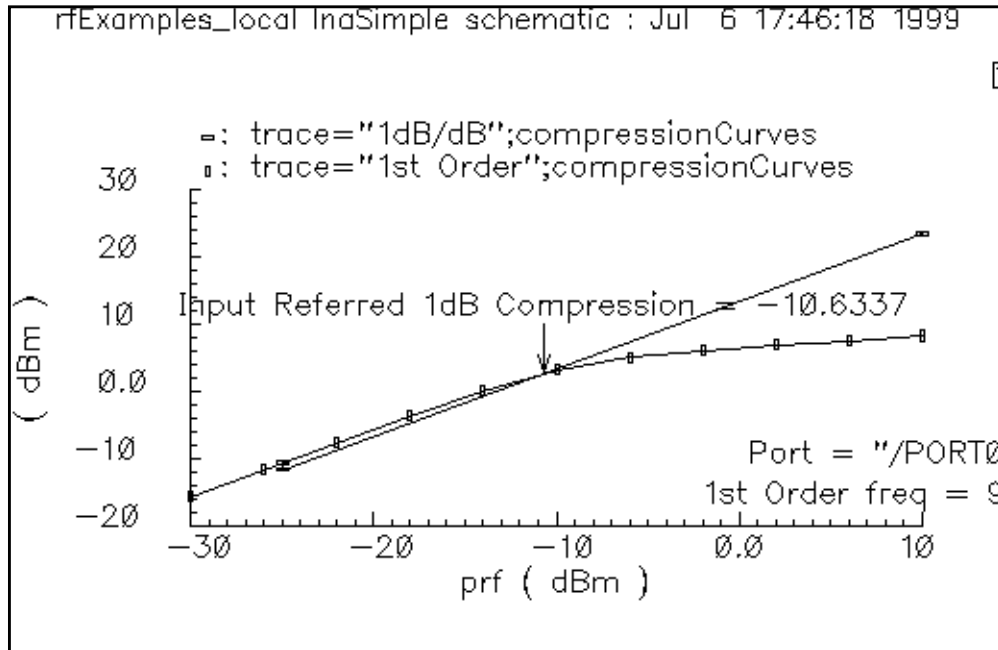
Add To Outputs

> Select port on schematic ...

Click here.

6. In the Schematic Window, click on the PORT 0 output port.

The plot appears in the Waveform window.



The SpectreRF simulator calculates the 1dBCompressionPoint as -10.6337.

Plotting the Third-Order Intercept Point

The method described here uses only swept PSS analysis. An [alternative method](#), which runs more quickly if the input signals are very close together, uses both swept PSS and PAC analyses and is described in [Chapter 4](#).

1. Disable any analyses you have run previously.
2. Be sure the full path to the model file is specified in the [Model File Set-up](#) form.

Editing the Schematic

1. In the [Schematic window](#), click on the input voltage source and then choose *Edit – Properties – Objects*.

The [Edit Object Properties](#) form appears.

2. In the Edit Object Properties form, do the following:
 - a. Choose *sine* for the *Source Type*.

- b. Highlight *Display second sinusoid*.
 - c. Type `fund2`, or any name you choose, in the *Second frequency name* field.
 - d. Type `prf` for the *Amplitude (dBm)* and *Amplitude 2 (dBm)* values.
 - e. Type `900M` for the *Frequency* value.
 - f. Type `920M` for the *Frequency 2* value.
 - g. Be sure the *Amplitude* and *Amplitude2* fields are empty.
3. In the Edit Object Properties form, click *OK*.
4. In the Schematic window, choose *Design – Check and Save*.

Analysis Setup

- In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

In the Choosing Analyses form, do the following and then click *OK*:

1. In the Choosing Analyses form, click on *pss*.
2. In the *Fundamental Tones* list box, be sure the *Auto calculate* button is highlighted.

The fundamental frequency, 20M, is displayed.

The *Beat Frequency* button is highlighted by default.
3. In the *Output harmonics* cyclic field, choose *Number of harmonics* for the *Output harmonics* choice and type in a reasonable value, such as 60, in the field.
4. Highlight *moderate* for the *Accuracy Defaults (errpreset)* value.
5. Highlight the *Sweep* button.

The form changes to let you specify data for the sweep.
6. In the *Sweep* cyclic field, choose *Variable*.
7. Click on the *Select Design Variable* button.

The Select Design Variable form appears.
8. In the Select Design Variable form, highlight *prf* and click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

9. Choose *Start-Stop* for the *Sweep Range*, and type -30 and 10 as the *Start* and *Stop* values, respectively, in the fields.
10. Choose *Linear* for the *Sweep Type* and click on the *Step Size* button. Type 5 as the step size.

The filled-in form looks like this. (You need to use the scroll bar to see the whole form.)

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

Clear/Add Delete Update From Schematic

◆ Beat Frequency
◇ Beat Period 20M Auto Calculate ■

Output harmonics
Number of harmonics 60

Accuracy Defaults (empreset)
 conservative moderate liberal
Additional Time for Stabilization (tstab)
Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep Frequency Variable? ◆ no ◇ yes
Variable Variable Name prf
Select Design Variable

Sweep Range
◆ Start-Stop Start -30 Stop 10
◇ Center-Span

Sweep Type
◆ Linear ◆ Step Size 5
◇ Logarithmic ◇ Number of Steps 1

Add Specific Points

Running the Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation is completed successfully.

Plotting the Third-Order Intercept Point

- In the Simulation window, choose *Results – Direct Plot – PSS*.

The Waveform window and the PSS Results form appear.

In the PSS Results form, do the following:

1. Highlight *Replace* for *Plot Mode*.
2. Highlight *pss* for *Analysis Type*.
3. Highlight *IPN Curves* for *Function*.
4. Highlight *Variable Sweep (“prf”)* for *Circuit Input Power*.
5. Choose *Input Referred IP3*.
6. Type *-25* for the *Extrapolation Point*.

You learn what value to use for the extrapolation point through experience. If you do not specify a value, the plot defaults to the minimum variable value.

7. In the *order* cyclic field, choose *order 3rd*.
8. In the *3rd Order Harmonic* list box, click on *940M*.
9. In the *1st Order Harmonic* list box, click on *900M*.

Because the two input frequencies are 900MHz and 920MHz, the two-tone, third-order harmonics are 880MHz and 940MHz. These correspond to the 45th and 47th harmonics.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The filled-in form looks like this.

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss

Function

Voltage Current

Power Voltage Gain

Current Gain Power Gain

Transconductance Transimpedance

Compression Point IPN Curves

Power Contours Beat Frequency

Reflection Contours

Circuit Input Power Single Point

Variable Sweep ("prf")

"prf" ranges from -30 to 10

Extrapolation Point (dBm)

(Defaults to -30)

Input Referred IP3 Order

3rd Order Harmonic

45	900M
46	920M
47	940M
48	960M
49	980M

1st Order Harmonic

42	840M
43	860M
44	880M
45	900M
46	920M

Add To Outputs

> Select port or net on schematic ...

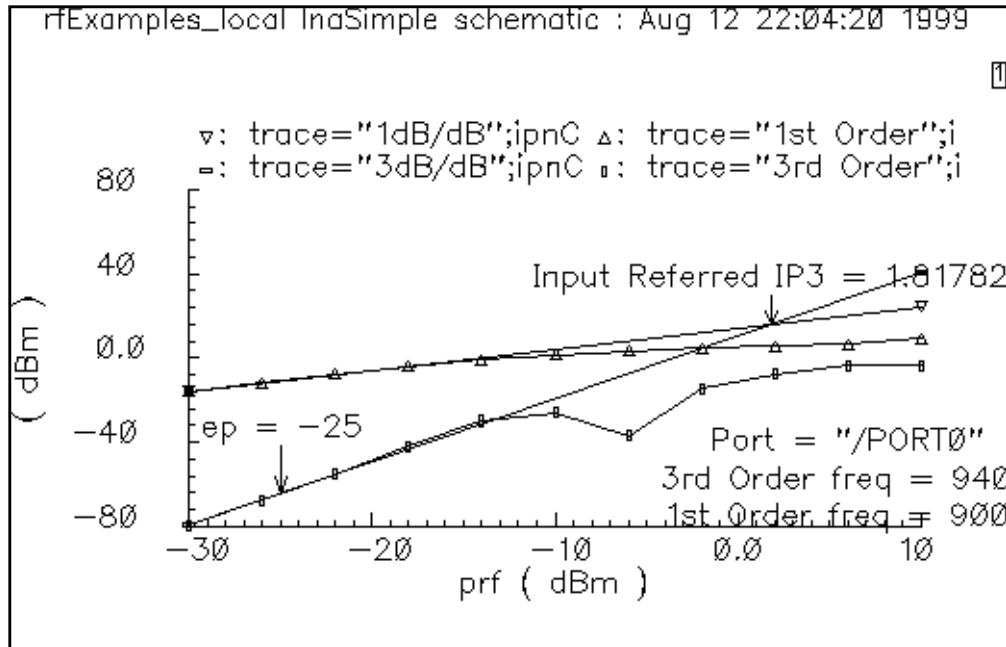
Click here.

10. In the Schematic window, click on the PORT 0 output port.

The IPN point is plotted in the Waveform window.

11. Click on *Replot*.

The SpectreRF simulator plots the first-order and third-order curves and identifies the intersection of their slopes.



The third-order interpolation point for this example is -1.81782dB.

Plotting Conversion Gain and Power Supply Rejection

1. Be sure the full path to the model file is specified in the Model File Set-up form.
2. Disable any analyses you have run previously.

Editing the Schematic

1. In the Schematic window, choose *Edit – Properties – Objects*.


The Edit Object Properties form appears.

2. In the Schematic window, click on the RFinput voltage source.

The Edit Object Properties form changes to let you specify data for the input voltage source.

In the Edit Object Properties form, do the following, if necessary:

1. Type `dc` for the *Source Type*.

2. In the Edit Object Properties form, click *OK*.
3. To check and save the design, click on the check mark icon in the top left corner of the Schematic window. 

Analysis Setup

1. In the Simulation window, choose *Analyses – Choose*.

The Choosing Analyses form appears.

2. In the Choosing Analyses form, choose *pss*.
3. Type 900M for the *Beat Frequency* value.

Because the source type is dc, you must manually enter the fundamental frequency.

The *Beat Frequency* button is highlighted by default.

Be sure the *Auto Calculate* button is not highlighted.

4. In the *Output harmonics* cyclic field, choose *Number of harmonics* and type in a reasonable value, such as 4, in the field.
5. Highlight *conservative* for the *Accuracy Defaults (errpreset)* value.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The Choosing Analyses form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

◇ sens ◇ sp ◇ pdisto **◆ pss** ◇ pac
◇ pnoise ◇ pxf ◇ envlp

Periodic Steady State Analysis

Fundamental Tones

#	Name	Expr	Value	Signal	SrcId
---	------	------	-------	--------	-------

1 2 3 Moderate

Clear/Add Delete Update From Schematic

◆ Beat Frequency 900M Auto Calculate
◇ Beat Period

Output harmonics

Number of harmonics

Accuracy Defaults (errpreset)

conservative moderate liberal

Additional Time for Stabilization (tstab)

Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep

Enabled Options...

6. At the top of the Choosing Analyses form, click on *pxf*.

The form changes to let you specify data for the PXF analysis.

In the Periodic XFA section of the Choosing Analyses form, do the following:

1. In the *Frequency Sweep Range(Hz)* cyclic field, choose *Start-Stop*, and type 10M and 1.2G for the *Start* and *Stop* values, respectively in the fields.
2. In the *Sweep Type* cyclic field, choose *Linear* and highlight the *Number of Steps* button.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

3. Type 101 for *Number of Steps* in the field.
4. In the *Sidebands cyclic field*, choose *Maximum sideband* for the *Sidebands* choice. Type 0 in the field.
5. Choose *voltage* for the *Output* choice.
6. Click on the *Select* button for the *Positive Output Node*, and then click on the output signal in the Schematic window.
7. Highlight the *Enabled* button for the small-signal analysis, if necessary.
8. Leave the *Negative Output Node* field empty.

This field defaults to */gnd!* You can set the *Negative Output Node* to a different value by clicking on the *Negative Output Node Select* button and then clicking on the output node in the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The pxf section of the form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Periodic XF Analysis

PSS Beat Frequency (Hz) 900M

SweepType

Frequency Sweep Range (Hz)

Start-Stop Start 10M Stop 1.2G

Sweep Type

Linear Step Size 10M
 Number of Steps

Add Specific Points

Sidebands

Maximum sideband 1

Output

voltage Positive Output Node /net28 Select
 probe Negative Output Node Select

Enabled Options...

9. In the Choosing Analyses form, click *OK*.

Running Simulation

1. To run the simulation, choose *Simulation – Netlist and Run* in the Simulation window.

The output log file appears and displays information about the simulation as it runs.

2. Look in the CIW for a message that says the simulation is completed successfully.

Plotting Conversion Gain and Power Supply Rejection

► In the Simulation window, choose *Results – Direct Plot – PSS*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The Waveform window and the PSS Results form appear.

In the PSS Results form, do the following:

1. *Highlight Replace* for *Plot mode*.
2. *Highlight pxf* for *Analysis Type*.
3. *Highlight Voltage Gain* for *Function*.
4. *Highlight dB20* for *Modifier*.

The filled-in form looks like this.

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis Type

pss pxf

Function

Voltage Gain Transimpedance
 Pss Beat Frequency

Currently, only spectrum data is available

Modifier

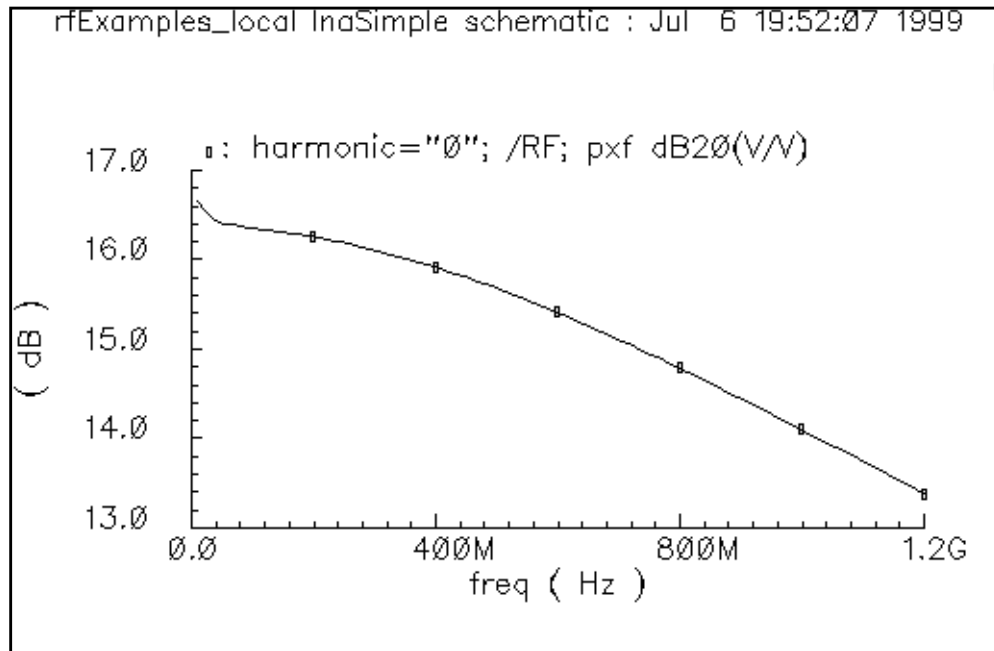
Magnitude Phase dB20
 Real Imaginary

5. In the Schematic window, click on the RF source component.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

The transfer function for the LNA is plotted in the WAVEform window.



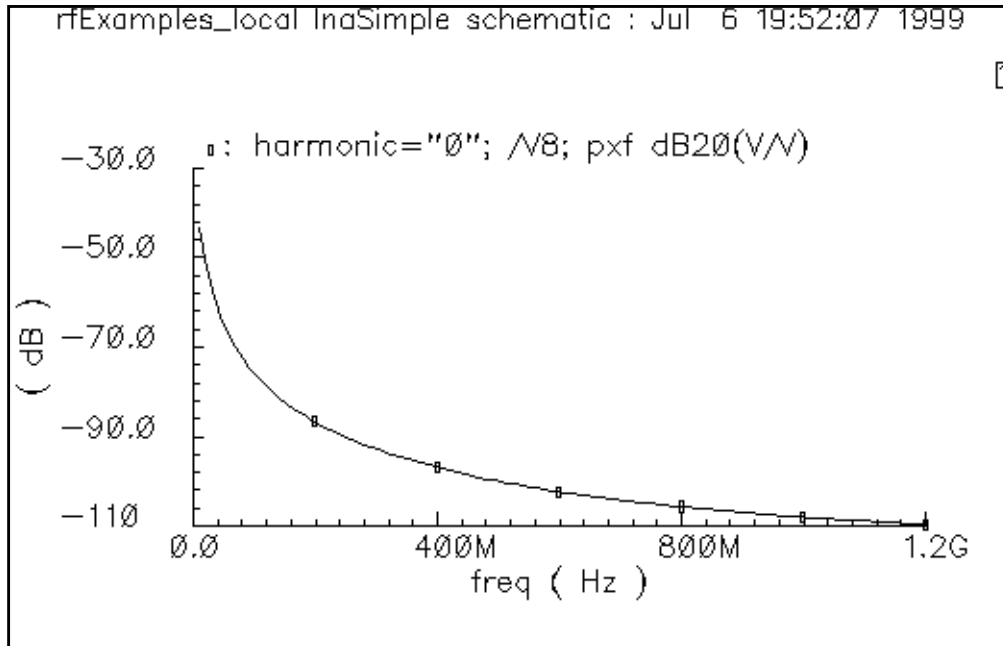
The conversion gain at 900M is approximately 14.4 dBV.

6. In the PSS Results form, choose *Replace* for *Plot Mode*.
7. To see the power supply rejection, click on the vdb DC voltage source in the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Low-Noise Amplifiers

Be sure to click on the components, not on the wires.



The power supply rejection at 900M is approximately -106 dBV.

Simulating Transmission Lines

The Affirma RF simulator can model transmission lines using a quadrature model approach. You can choose between two use models to simulate transmission lines.

- Using the LMG Visual Interface independent of the Affirma analog circuit design environment
- Modeling transmission lines from within the Affirma analog circuit design environment

This chapter shows you the procedures for each use model.

In the first use model, you manually run the Transmission Line Model Generator (LMG) for each unique transmission line configuration. You create a separate model for each change in line type, geometry, parameters, and operating frequency. After you create the models manually, you instruct the Affirma™ analog circuit design environment to include the resulting external model files.

You might want to run LMG manually if you want to see a display of the transmission line cross section. This cross section display is not currently available as part of the Affirma analog circuit design environment, but you can call the manual generation procedure from within the design environment. You also might want to run LMG manually if you want to share the model or if you have a circuit with a number of identical transmission line instances. If you run LMG manually, you create a single model file that can be applied to many instances. If you run LMG completely within the Affirma analog circuit design environment, the software creates each instance separately.

In the second use model, you run LMG completely within the Affirma analog circuit design environment.

Use Model One: Using the LMG Visual Interface

This use model lets you view a cross section of the transmission line you specify. If you modify your specifications, the two-dimensional cross-section updates to reflect your changes.

You start by using LMG to create a lumped-circuit macromodel for each transmission line in the circuit. For each model, you specify line-parameter information, the length of the line, and the operating frequencies in the LMG form.

You also specify a unique name for the resulting file that contains the Spectre netlist that describes the subcircuit. It is helpful to use a name that describes the configuration of the particular transmission line. For example, you might name a stripline of length 1000 with a maximum frequency of 20GHz `tline_s1000_20G`. (A maximum frequency of 20GHz means that the model has diminished accuracy above this value.)

Using your input specifications, LMG automatically generates a quadrature model subcircuit for each line. The quadrature model subcircuits are only accurate for the length and frequencies that you specify. If you modify a transmission line, you must rerun LMG for the new configuration and update the appropriate instances in the schematic to reflect the new LMG output.

When you simulate the circuit, each model file is incorporated into the design by specifying the filename as a property on the corresponding `tline3` instance(s).

Note: LMG can also create four-terminal transmission line models, but SpectreRF simulation does not use these models. For more information about the four-terminal models, see the application note “Including Transmission Line Models in Spectre/SpectreRF Simulations in the 4.4.1 Release.”

Use Model Two: Modeling Transmission Lines Without the Visual Interface

The second use model is easier to use than the first use model. It includes the following steps:

1. Open the schematic.
2. Choose or place a `tline3` instance in the schematic.
3. Open the *Edit – Properties – Objects* form for the transmission line.
4. Describe the transmission line by typing in parameter values. (Units are meters and Hertz only.)
5. Repeat steps 2 through 4 for each `tline3` instance in the design.

The Transmission Line Model Generator Form

This section describes the Transmission Line Model Generator form. This form is divided into four sections and shown below.

The screenshot shows the 'Transmission Line Model Generator' window. At the top, there are 'File' and 'Options' menus. Below the menu bar is a diagram of a transmission line structure. The diagram shows a top layer labeled 'air, er=1', a middle layer labeled 'er = 10' with a thickness 'd = 400 um', and a bottom layer labeled 'Ground Plane'. A small rectangular conductor is shown on the middle layer. Below the diagram is a form with the following parameters and controls:

Transmission Line Type	<input type="radio"/> Microstrip	<input type="radio"/> Stripline
Number of Conductors	<input type="text" value="1"/>	
Dielectric Constant (er)	<input type="text" value="10"/>	
Dielectric Thickness (d)	<input type="text" value="400"/>	um
Conductor Width	<input type="text" value="300"/>	um
Conductor Thickness	<input type="text" value="100"/>	um
Conductor Length	<input type="text" value="5000"/>	um
Conductivity	<input type="text" value="5.6e7"/>	S/m
Fmax	<input type="text" value="10"/>	GHz

At the bottom of the form are two buttons: 'Calculate Parameters' and 'Create Macromodel'.

Menus

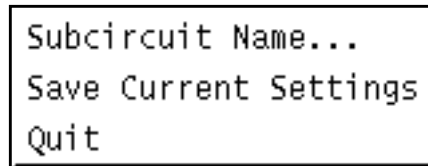
There are two pull-down menus at the top of the form, *File* and *Options*, in which you choose various options.

Affirma RF Simulator (SpectreRF) User Guide

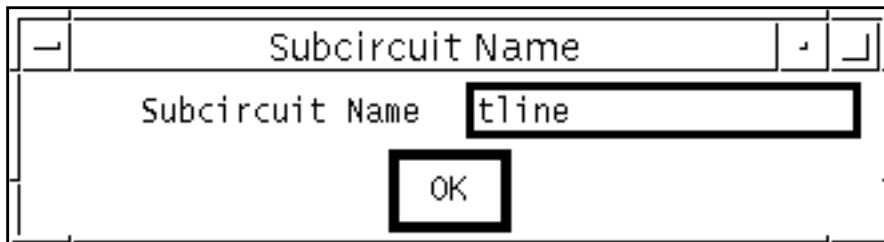
Simulating Transmission Lines

File Menu

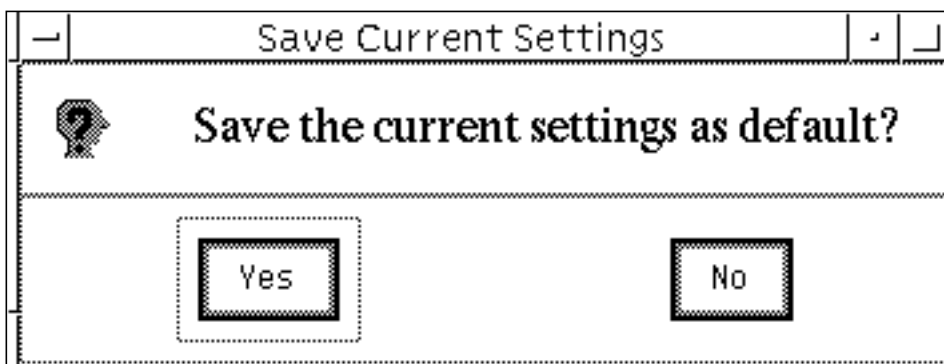
The *File* menu has the following three options:



Subcircuit Name: This choice brings up a form in which you type the name of the output subcircuit. The default subcircuit name is `tline`. The output subcircuit file name is created from the name you choose by appending `.scs`



Save Current Settings: This choice lets you save the current transmission line information as a default to file `./.initlmg` in the current directory.



Quit: This button exits `lmg`. Before `lmg` quits, it compares the current transmission line information to the information stored in file `./.initlmg`. If they are different, or if the `./.initlmg` file does not exist, the current setting is saved to file `./.initlmg`.

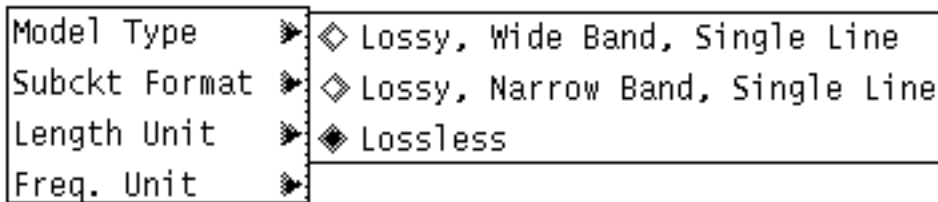
Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

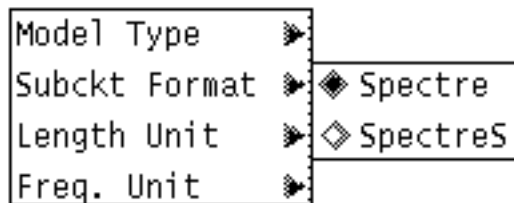
Options Menu

The *Options* menu has three main options. Each option has a submenu in which you choose a value. After you choose a value, the geometry of the diagram is updated automatically to implement your choice.

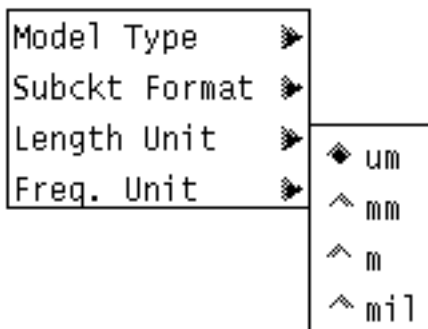
- **Model Type:** The three options you can choose are *Lossy, Wide Band, Single Line*; *Lossy, Narrow Band, Single Line*; and *Lossless*.



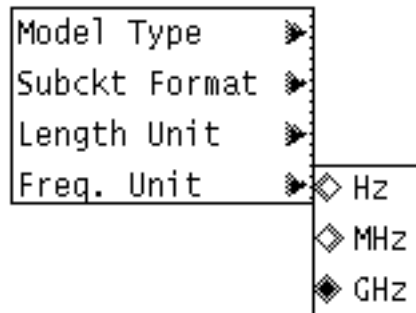
- **Subckt Format:** You can choose *Spectre* or *SpectreS*.



- **Length Unit:** You can choose *um*, *mm*, *m*, or *mil* as valid units. The currently selected units are displayed to the right of the data entry fields in the form. All the dimension quantities use the same unit for consistency.



- **Freq. Unit:** You can choose *Hz*, *MHz*, and *GHz* as valid units. The currently selected units are displayed to the right of the data entry fields in the form.



Display Section

Below the menu buttons is a display of the 2D cross section of the transmission line. Inside the cross section, the dielectric thickness, the dielectric constant, and the conductor height are specified. The display is to scale and applies the same scale for both the horizontal and vertical dimensions. The display changes immediately when you specify new parameter information to reflect your changes.

Data Entry Section

In the data entry section, you type information about the actual transmission line. If a field has a label, you can leave the default value unmodified, but you cannot leave the field blank. Do not put values in the fields that are grayed out. Grayed out fields represent quantities that are not needed for the model you requested. The meanings of the data entries are explained below.

Transmission Line Type: The buttons let you choose *Microstrip* or *Stripline*.

Number of Conductors: In this field, you type the number of conductors for the transmission line.

Dielectric Constant (ϵ_r): For *Microstrip*, ϵ_r is the relative dielectric permittivity of the substrate with air above the substrate. For *Stripline*, ϵ_r is the relative permittivity of the dielectric between the upper and lower ground planes.

Dielectric Thickness (d): For *Microstrip*, d is the thickness of the substrate. For *Stripline*, d is the distance between the upper and lower ground planes.

Conductor Width: This field specifies the width of the conductor wire.

Conductor Thickness: This field specifies the thickness of the conductor wire.

Conductor Height (h): This entry is for *Stripline* only. It specifies the distance between the lower face of the conductor wire and the upper face of the bottom ground plane.

Conductor Distances: If there are multiple conductors, this field specifies the distance between them.

Conductor Length: This field specifies the actual length of the transmission line in the unit shown to the right of the data entry section.

Conductivity: This field specifies the conductivity of the conductor wire. This parameter is used to calculate the DC resistance per meter.

Fmax: The maximum frequency at which the macromodel must work properly. You usually set *Fmax* to the frequency of the highest harmonic that has significant energy in the transmission line simulations. LMG generates a macromodel of the transmission line that is good from DC to the value chosen for *Fmax*.

Function Buttons

The function buttons let you update the figure display, calculate the transmission line parameters, and create the macromodels. There are two function buttons at the bottom of the Transmission Line Modeler form.

Calculate Parameters: After you specify the 2D transmission line geometry, click on this button to calculate the transmission line parameters. The display shows the characteristic impedance, propagation velocity, and the DC resistance per meter of the transmission line in the upper right corner of the form.

Create Macromodel: After you click on the *Calculate Parameters* button, this button is enabled. Click on this button to create the transmission *tline* model. A pop-up window tells you the name of the transmission line model file.

Default Values

When you start lmg, the `./ .initlmg` file is checked for the defaults. If file `./ .initlmg` does not exist, then the following set of default values is used:

```
lmgLineType = Microstrip
lmgLengthUnit = um
lmgFreqUnit = GHz
```

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

```
lmgNumberof Conductors = 1
lmgDielectricPermittivity = 10
lmgDielectricThickness = 400
lmgConductorWidth = 300
lmgConductorDistances = For multiconductors, the width of the first
                          conductor
lmgConductorThickness = 100
lmgConductorHeight = For multiconductors, the height of the first
                      conductor
lmgConductorLength = 5000
lmgConductivity = 5.6e7
lmgMaxFreq = 10
lmgSubCircuitName = tline
```

Example of Modeling a Transmission Line Using LMG

This section presents an example of modeling a transmission line using the Transmission Line Model Generator form following the first use model. Because the microstrip and stripline procedures are very similar, only a stripline example is shown here.

Setup

Before you run lmg, you must set up the executables correctly. You must install the Spectre or SpectreRF circuit simulator, and also Coeffgen, with the correct path set.

Creating the Transmission Line Model

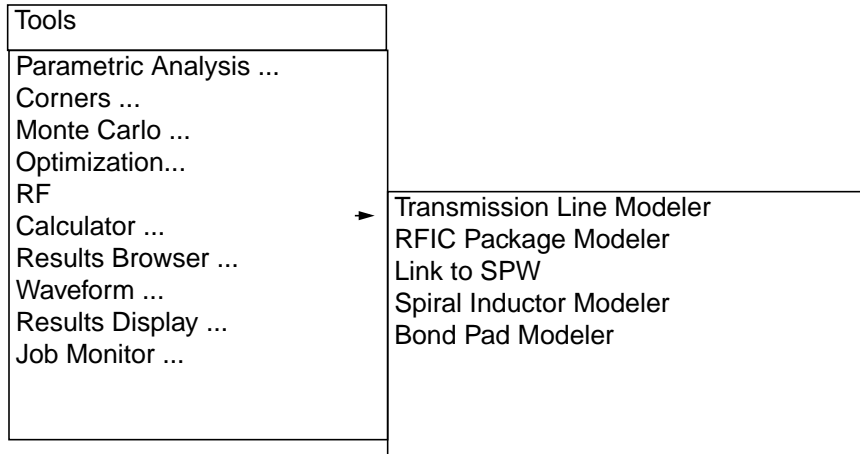
To model the transmission line, perform the following steps:

1. Choose *Tools – RF – Transmission Line Modeler* in the Simulation Window or type lmg from the UNIX command line.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

The Transmission Line Model Generator form appears.



The example assumes that this is the first run of LMG and that file `./initlmg` does not yet exist. Consequently, LMG uses internal default values.

2. In the Options menu, choose *Subckt Format – Spectre*.

This choice assumes you run the simulation in the Affirma analog circuit design environment.

3. In the Options menu, choose *Model Type – Lossless*.

This example assumes a lossless transmission line.

4. To specify a stripline, highlight *Stripline* in the Transmission Line Model Generator form.

5. To specify *um* as the length unit, choose *Length Unit – um* in the Options menu.

6. To specify *GHZ* as the frequency unit, choose *Freq. Unit – GHz* in the Options menu.

The display section of the form now shows a stripline, and the length and frequency values in the data entry section reflect your input.

The form appears like the one below. (The values in the data entry fields are default values. If you have other values stored in an `./initlmg` file, these stored values appear in the data entry fields.)

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

Figure 7-1 Transmission Line Model Generator Form

Transmission Line Model Generator

File Options

Ground Plane

d=400 um
er = 10

h=150 um

Ground Plane

Transmission Line Type Microstrip Stripline

Number of Conductors

Dielectric Constant (er)

Dielectric Thickness (d) um

Conductor Width um

Conductor Thickness um

Conductor Height (h) um

Conductor Length um

Conductivity S/m

Fmax MHz

7. Specify the following values in the data entry fields:

Number of Conductors: 1

Dielectric Constant (er): 5

Dielectric Thickness (d): 500 um

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

Conductor Width: 200 μm

Conductor Thickness: 50 μm

Conductor Height (h): 225 μm

Conductor Length: 1000 μm

Conductivity: 5.6e7

Fmax: 20

The macromodel is created using the Gaussian quadrature algorithm[1]. This model is usable only below a certain frequency.

Generally, the higher the order of the quadrature, the wider the useful band width. The order of the quadrature is chosen so that the magnitude of the transmission coefficient of the equivalent transmission line is no less than 97.5 percent at the highest nonnegligible frequency. One way to determine this frequency is to use harmonics. Given the fundamental frequency of the signal and the number of harmonics, *Fmax* equals the fundamental frequency multiplied by the highest harmonic number.

For this example, the value of *Fmax* assumes a fundamental frequency of 2 GHz with 10 harmonics considered. The *Fmax* value is (2x10) GHz.

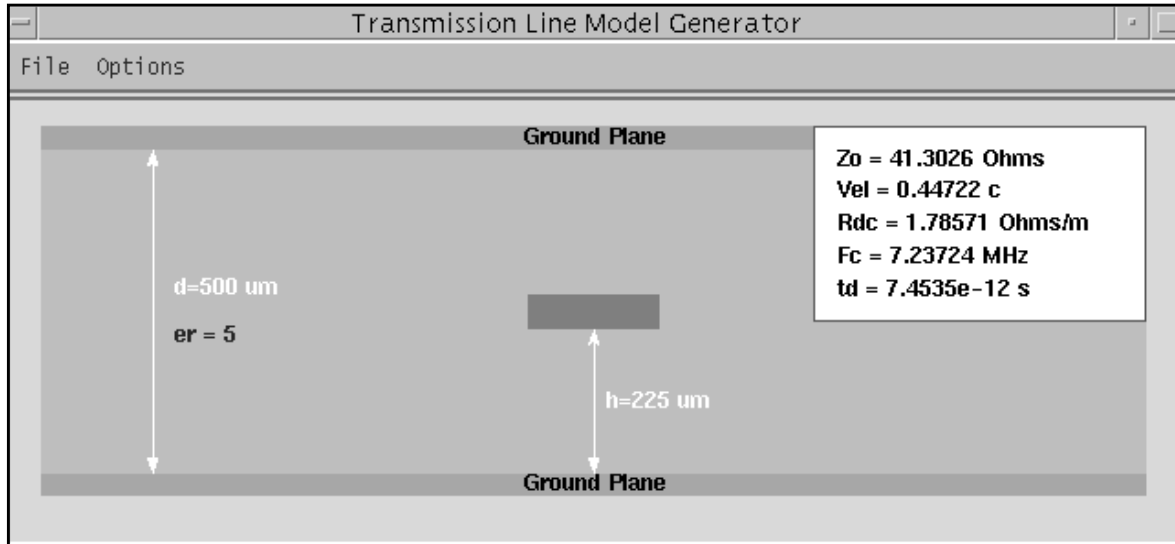
8. Click on the *Calculate Parameters* function button.

This step calculates the characteristic impedance, the propagation velocity, the DC resistance per meter, the corner frequency of the conductor, and the time delay of the transmission line. The display section changes to reflect the values you specified, and the calculated parameters appear in the top right corner of the display section. At this point, the *Create Macromodel* button is enabled.

Affirma RF Simulator (SpectreRF) User Guide

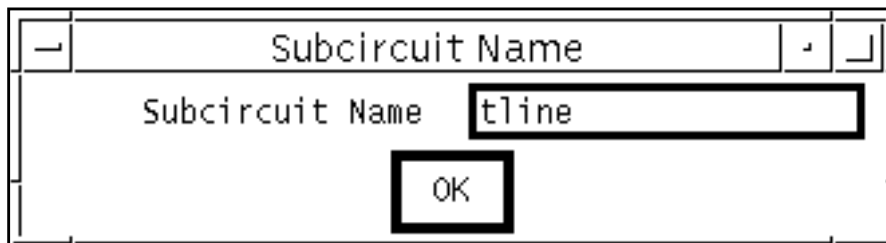
Simulating Transmission Lines

The display section of the Transmission Line Model Generator form is shown below.



9. In the File menu, choose *Subcircuit Name*.

The Subcircuit Name form appears.

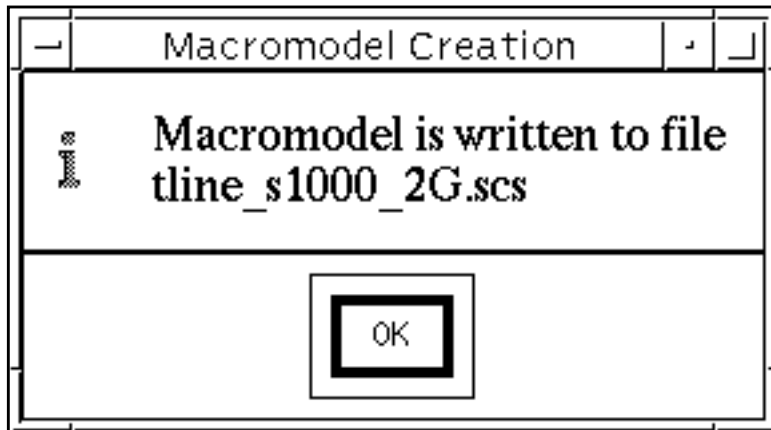


10. Type `tline_s1000_20G` as the subcircuit name.

The name gives information about the type, length, and maximum frequency of the subcircuit.

11. To create the macromodel, click on the *Create Macromodel* function button.

The Macromodel Creation dialog box shows the name of the file containing the macromodel netlist.



The final results are written to a file of the of the name you specified with a `.scs` suffix. This file name is used later by the Affirma analog circuit design environment as a property of a `tline3` symbol.

12. In the Macromodel Creation dialog box, click *OK*.

The contents of the `tline_s1000_20G.scs` file are shown below.

tline_s1000_20G.scs

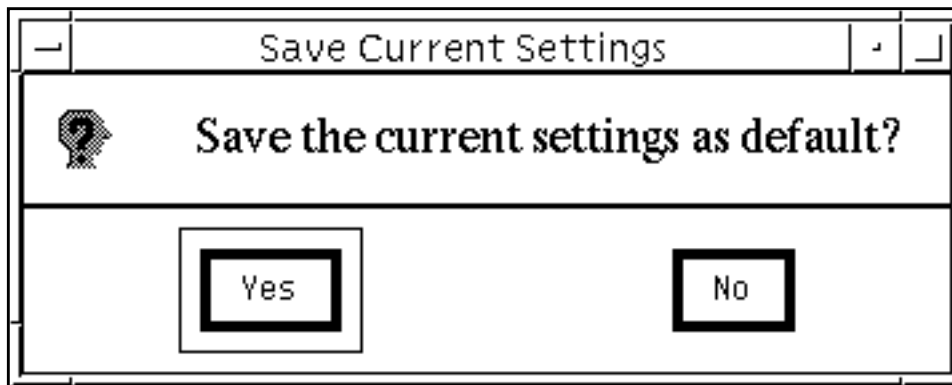
```
// Lossless single conductor tranmission line  model
// Zo = 3.829700e+01
// td = 7.453560e-12
subckt lmg in out ref
  l0 (in  n10) inductor l=1.33904e-11
  l1 (n10 n11) inductor l=5.24812e-11
  l2 (n11 n12) inductor l=7.68529e-11
  l3 (n12 n13) inductor l=7.68529e-11
  l4 (n13 n14) inductor l=5.24812e-11
  l5 (n14 out) inductor l=1.33904e-11
  c0 (n10 ref) capacitor c=2.30559e-14
  c1 (n11 ref) capacitor c=4.65765e-14
  c2 (n12 ref) capacitor c=5.536e-14
  c3 (n13 ref) capacitor c=4.65765e-14
```

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

```
c4 (nl4 ref) capacitor c=2.30559e-14
ends lmg
```

13. To save the current settings as default values for subsequent LMG runs, choose *Save Current Settings* in the File menu.



The current setting is saved in file `./.initlmg`. LMG reads this file in subsequent startups. The `./.initlmg` file contains all the quantities you can change. As long as you preserve the format, you can edit this file directly to choose the startup defaults of LMG. The contents of the `./.initlmg` file are shown below.

.initlmg

```
lmgLineType = Stripline
lmgLengthUnit = um
lmgFreqUnit = GHz
lmgDielectricPermittivity = 5
lmgDielectricThickness = 500
lmgConductorWidth = 200
lmgConductorThickness = 50
lmgConductorHeight = 225
lmgConductorLength = 1000
lmgConductivity = 5.6e7
lmgMaxFreq = 20
lmgSubCircuitName = tline_s1000_20G
```

14. In the File menu, click on *Quit* to exit LMG at the end of the run.

Using Macromodels Created Externally by LMG

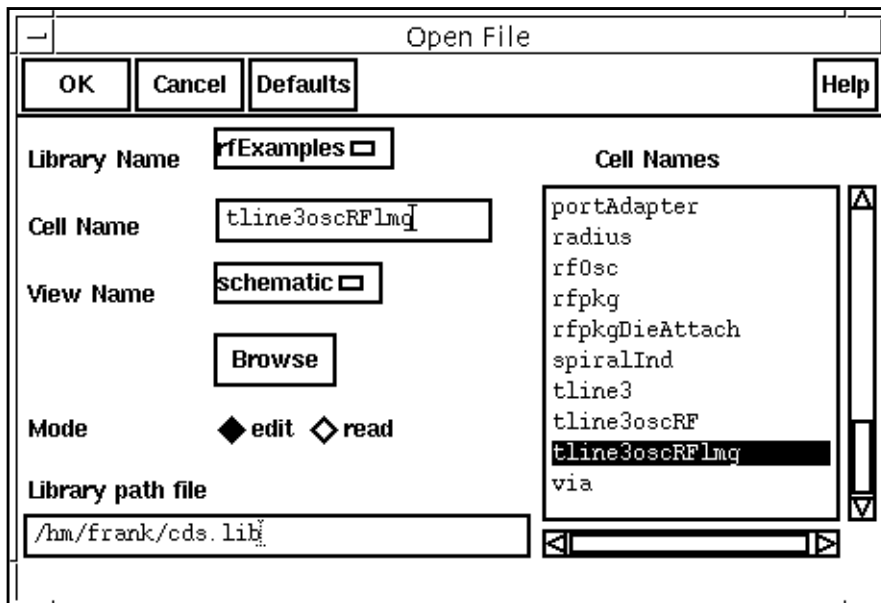
This example uses the circuit `tline3oscRF1mg`, which you can find in your `rfExamples` directory along with all the other sample circuits used in this manual. If you need assistance accessing the `rfExamples` directory, refer to Chapter 3, “[Setting Up the Examples](#).” This example illustrates how to call externally-created LMG macromodels from within the analog circuit design environment.

► Choose *File – Open* in the Command Interpreter Window (CIW).

The Open File form appears. Filling in the Open File form lets you open the schematic.

1. In the Open File form, perform the following:

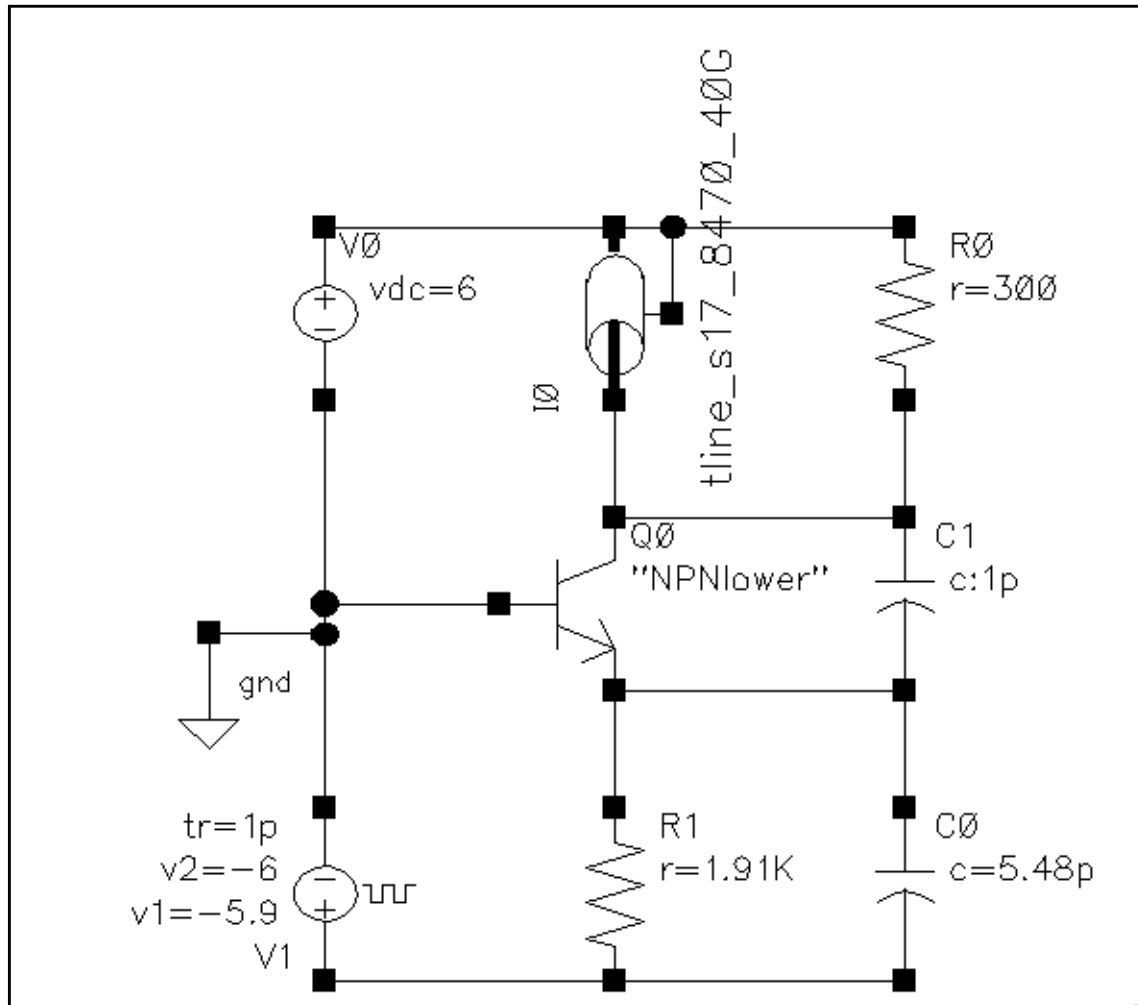
- ❑ Choose *rfExamples* for the *Library Name*.
- ❑ Choose *schematic* for the *View Name*.
- ❑ In the *Cell Names* list box, highlight *tline3oscRF1mg*. *tline3oscRF1mg* appears as the Cell Name.
- ❑ Highlight *edit* to choose *Mode*.



2. Click *OK* in the Open File form.

The Schematic window opens and displays the schematic,

The tline3oscRF Schematic



3. In the Schematic window, click on the transmission line *tline_s17_8470_40G*, the current transmission line in the circuit.
The transmission line is now selected.
4. In the Schematic window, choose *Edit – Properties – Objects*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

The Edit Object Properties form for *tline_s17_8470_40G* appears.

Property	Value	Display
Library Name	rfExamples	off <input type="checkbox"/>
Cell Name	tline3	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	I0	off <input type="checkbox"/>

CDF Parameter	Value	Display
use external model file	<input type="checkbox"/>	off <input type="checkbox"/>
line type	◆ microstrip ◇ stripline	off <input type="checkbox"/>
er: permittivity	12.9	off <input type="checkbox"/>
d: dielectric thickness (m)	100e-6	off <input type="checkbox"/>
w: conductor width (m)	71e-6	off <input type="checkbox"/>
t: conductor thickness (m)	5e-6	off <input type="checkbox"/>
len: conductor length (m)	8.47e-3	off <input type="checkbox"/>
model type	◇ lossless ◆ lossy	off <input type="checkbox"/>
lossy type	◆ narrow band ◇ wide band	off <input type="checkbox"/>
sigma: conductivity	5.6e7	off <input type="checkbox"/>
freq: max frequency (Hz)	40e9	off <input type="checkbox"/>

Click to select.

- In the Edit Object Properties form, highlight the *use external model file* button.

You ignore the fields that specify transmission line parameter values.

After you click the *use external model file* button, the form changes to let you specify the model name.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

6. In the Edit Object Properties form, type `tline_s1000_20G`, the name of the macromodel you created using LMG.

Edit Object Properties

OK Cancel Apply Defaults Previous Next Help

Apply To only current instance

Show system user CDF

Browse Reset Instance Labels Display

Property	Value	Display
Library Name	rfExamples	off <input type="checkbox"/>
Cell Name	tline	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	I0	off <input type="checkbox"/>

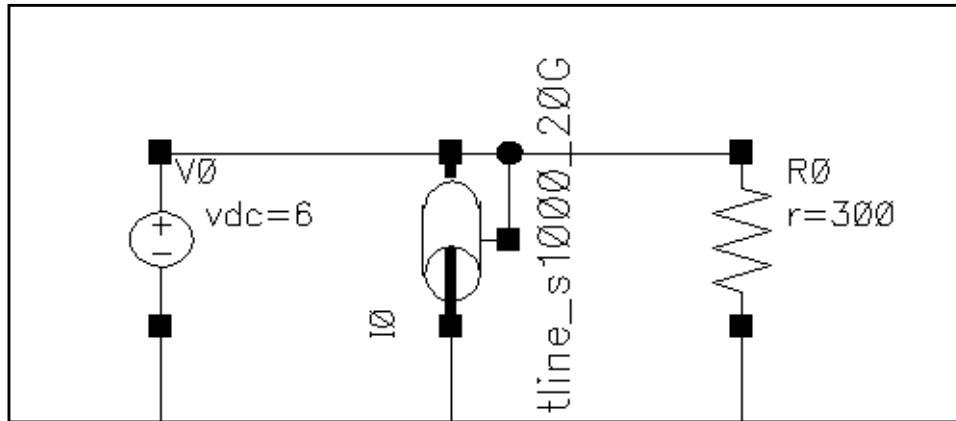
Add Delete Modify

CDF Parameter	Value	Display
use external model file	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
Model Name	tline_s17_8470_40G	value <input type="checkbox"/>

Type model name here.

7. In the Edit Object Properties form, click *OK*.

The component name in the Schematic window changes to reflect the modification.



You can now set up and run a simulation using the Affirma analog circuit design environment.

Example of Automatically Creating a Transmission Line

This procedure shows you how to create a transmission line for SpectreRF simulation using only the Affirma analog circuit design environment. This example follows the second use model. You can use this procedure if you do not need to see the two-dimensional cross-section display that LMG provides. This example creates a transmission line that is identical to the one created in the previous section by using LMG.

1. Open the *tline3oscRFImg schematic*.

The procedures to open the schematic are documented in the previous section of this chapter, “Example of Modeling a Transmission Line Using LMG Externally.”

2. In the Schematic window, click on the transmission line *tline_s17_8470_40G*, the current transmission line in the circuit.

The transmission line is now selected.

3. In the Schematic window, choose *Edit – Properties – Objects* to bring up the Edit Object Properties form for *tline_s17_8470_40G*.
4. In the Edit Object Properties form, perform the following:
 - Highlight *stripline* for the line type.
 - Type 5 for *er: permittivity*.
 - Type $500e-6$ for *d: dielectric thickness (m)*.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

Note: Notice that you enter values for this form in meters and Hertz.

- Type $200e-6$ for *w*: *conductor width*.
- Type $5e-6$ for *t*: *conductor thickness (m)*.
- Type $5e-6$ for *h*: *conductor height (m)*.
- Type $1e-3$ for *len*: *conductor length (m)*
- Highlight *lossless* for the *model type*.

Note: If you choose *lossy* for the model type, two additional fields open to let you specify the lossy type and conductivity.

- Type $40e9$ for *freq*: *max frequency (Hz)*.

For an [explanation](#) of this value see page 16 in the procedure for modeling transmission lines using LMG.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

The completed form appears like the one below.

Property	Value	Display
Library Name	rfExamples	off <input type="checkbox"/>
Cell Name	tline3	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	I0	off <input type="checkbox"/>

CDF Parameter	Value	Display
use external model file	<input type="checkbox"/>	off <input type="checkbox"/>
line type	<input checked="" type="radio"/> microstrip <input checked="" type="radio"/> stripline	off <input type="checkbox"/>
er: permittivity	9	off <input type="checkbox"/>
d: dielectric thickness (m)	500e-6	off <input type="checkbox"/>
w: conductor width (m)	200e-6	off <input type="checkbox"/>
t: conductor thickness (m)	5e-6	off <input type="checkbox"/>
h: conductor height (m)	5e-6	off <input type="checkbox"/>
len: conductor length (m)	1e-3	off <input type="checkbox"/>
model type	<input checked="" type="radio"/> lossless <input type="radio"/> lossy	off <input type="checkbox"/>
freq: max frequency (Hz)	40e9	off <input type="checkbox"/>

5. In the Edit Object Properties form, click *OK*.

You can now set up and run a simulation using the Affirma analog circuit design environment.

Accuracy of the LMG Macromodels

This section discusses the accuracy of the LMG macromodels that are used with instances of the *tline3* Affirma analog circuit design environment symbol. For guidelines about using the transmission line parameters with the Affirma analog circuit design environment symbol *tline*, see the Spectre online help or [Spectre Reference](#) manual.

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

The line parameters of a real transmission line are functions of frequency. [Figure 7-2](#) shows how the resistance and inductance per unit length of a hypothetical transmission line vary with frequency. [Figure 7-3](#) illustrates the current distributions in the stripline conductor for various frequencies. Below the corner frequency f_c , the current distribution is uniform throughout the conductor cross-section. At f_c , the current begins to crowd towards the outside of the conductor, increasing the resistance from its DC value and decreasing the inductance by taking away flux linked current. In the range $10f_c - 100f_c$, all the current is close enough to the surface so that the inductance saturates at a high-frequency limit value, and the resistance increases with increased frequency. For microstrip lines, although more current flows to the bottom surface of the conductor wire, this analysis of inductive effects is still true.

Figure 7-2

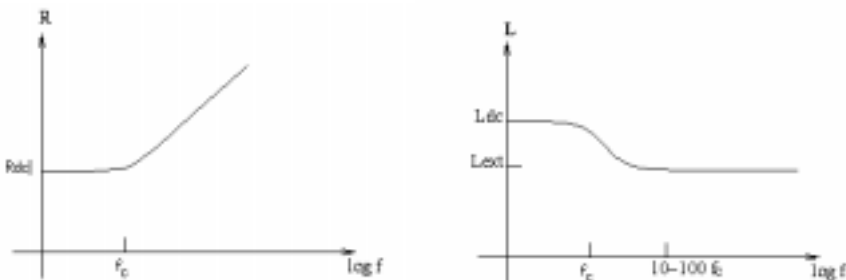
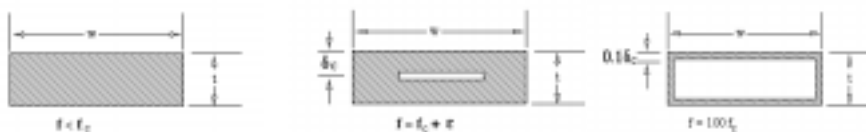


Figure 7-3 Cross-sectional current distribution at different frequencies



To model the skin-depth effect of the transmission line, LMG provides a *Wide Band Lossy* model which accurately approximates the skin-depth effect from DC to the maximum frequency specified. This model employs a pole-residual approach. Most models require five to 10 poles to approximate frequency dependent line parameters, so the subcircuits generated by such models are usually about seven to 10 times the size of the lossless model.

Besides the *Wide Band Lossy* model, LMG also provides a *Narrow Band Lossy* model for applications that require only a small bandwidth. In this model, only a single pole is used, so the generated subcircuit is much smaller than the wide band lossy model and only 1.5 times

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

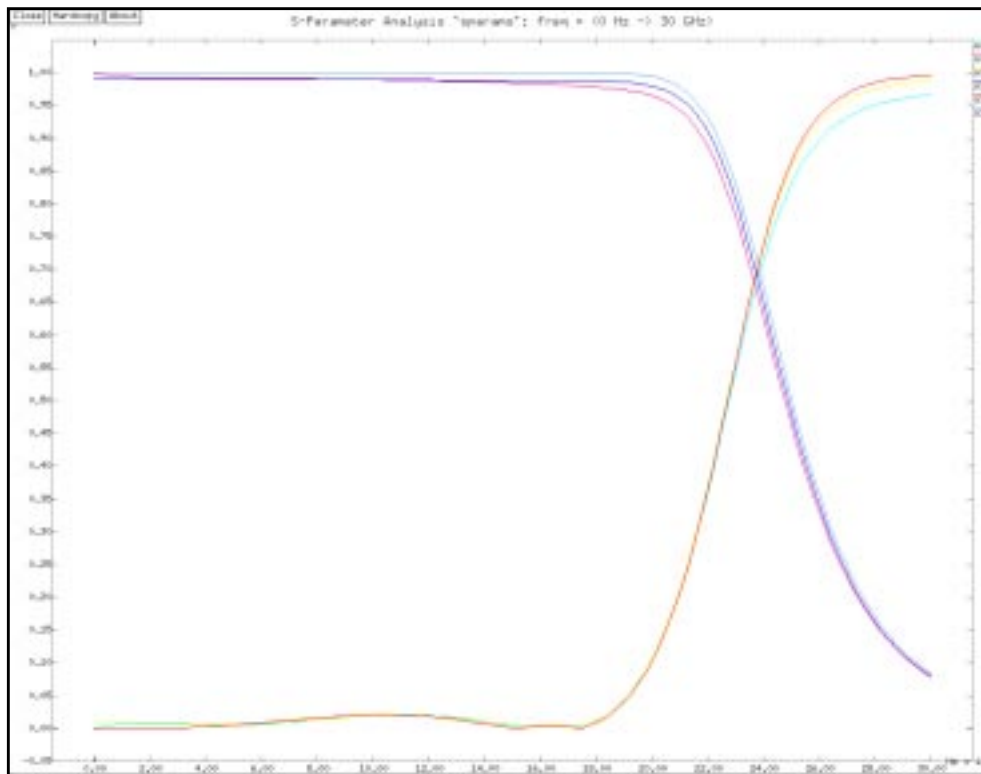
the size of the lossless model. The narrow band lossy model matches the wide band lossy model in the vicinity of the given working frequency.

For very low loss applications, LMG provides a *Lossless* model. In this model, the series resistance of the transmission line is ignored, and only the external inductance is considered.

Figure 7-4 compares the S-parameter results for each of the three models for a microstrip line with the following parameters: substrate permittivity 12.9, substrate thickness 200 μm , conductor width 70 μm , conductor thickness 3 μm , and conductor length 5000 μm . The working frequency is set at 10 GHz. Port resistance is 65.5 Ohms.

As shown in the figure, the lossless model calculates s_{12} as 1 in the frequency range, as expected. The narrow band lossy model has a constant lossy across the frequency range. Only the wide band lossy model correctly calculates the skin-depth effect over the whole frequency range.

Figure 7-4 Comparison of S-parameters for Lossless, Lossy Narrow Band, and Lossy Wide Band Models.



To calculate the per unit length capacitance of the transmission line, LMG employs a boundary-element method (BEM) type extraction algorithm. The high frequency inductance

Affirma RF Simulator (SpectreRF) User Guide

Simulating Transmission Lines

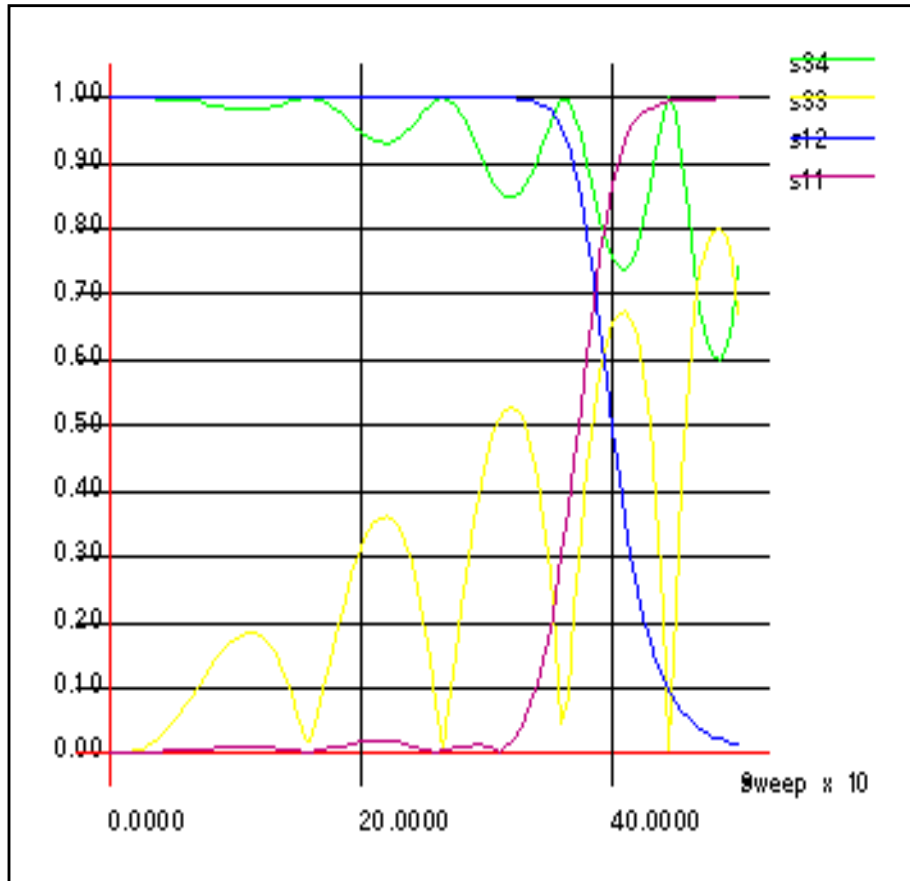
per unit length is calculated by assuming the current flows on the surface of the conductor, an appropriate assumption for RF simulations as shown above. The extraction algorithm has an error of less than 1 percent.

For the macromodel, LMG uses a nonuniform Gaussian quadrature algorithm.

[Figure 7-5](#) shows the frequency response of nonuniform and uniform lumping for a stripline with the conductor length set to 6000 μm . Both models use eight LC lumps to approximate the transmission line. [Figure 7-5](#) shows that within the bandwidth, the reflection coefficient of the nonuniform lumping is below 1 percent, but the uniform lumping causes much more reflection. For the same number of lumps, the nonuniform case has a much wider bandwidth. For the same accuracy, nonuniform lumping requires fewer lumps than uniform lumping.

By requiring the transmission coefficient to be above 0.975, the Gaussian nonuniform lumping requires an average of five lumps per wavelength, while the macromodel requires 8-10 lumps per wavelength. For the current version of LMG, the highest order allowed is 30, equivalent to six wavelengths at the maximum frequency. For longer lines, divide the line into sections shorter than six wavelengths. Then apply LMG to generate macromodels for each section.

Figure 7-5 S-Parameters of Nonuniform (S11 and S12) and Uniform (S33 and S34) Macromodels of a Stripline



The procedures shown here let you generate line models directly from line cross-sectional geometry and operating frequency information. These models permit transmission lines to be simulated in all Spectre analyses, including SpectreRF analyses. The nonuniform macromodels used in SpectreRF simulation are usually more accurate, and result in shorter simulation run times, than uniform macromodels of the same size.

Modeling RF IC Packages

As the working frequency of RF devices in PCS and wireless communications increases, the effect of the RF IC package parasitics, like reflection caused by impedance mismatch, high frequency resonances, and crosstalk among pins and bond wires, becomes important. Accurate simulation of the systems requires accurate modeling of these parasitics.

There are many IC package types, but more than 95 percent of commercial products use non-routable packages such as the small outline IC (SOIC), quad flat pack (QFP), and dual in line package (DIP) families. PKG, an RF IC package modeler, has been developed based on this information.

The PKG core algorithm contains the following four building blocks: the package physical geometry modeling block, mesh making for EM solvers, 3D EM solvers for capacitance and inductance calculations, and a post processing block for creating the macromodel and die attach terminal.

To set up the physical geometry, you must specify the following:

- Package dimensions, including the dimensions of the package body
- Die cavity size
- External part of the lead frame
- Internal lead frame setup
- Die attach setup
- Bond wire setup

PKG supports the following package families:

- SOIC families including shrunken small outline IC (SSOP), thin shrunken small outline IC (TSSOP), very small outline IC (VSO) and others
- QFP families, including TQFP and LQFP
- DIP families

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

PKG supports both the gull-wing pin and J-pin types.

The mesh-making function is based on the physical modeling block. The regularity of the actual package geometry permits the creation of high-quality mesh for higher efficiency and faster convergence. You can control the mesh to satisfy your accuracy requirements.

In the EM solver, Fastcap and Fasthenry are the capacitance and inductance solvers, respectively. Because the mesh-making module is relatively independent of the solvers, new solvers can be added to the current ones.

The postprocessing block takes the capacitance and inductance results from the EM solvers as input and generates macromodels. The supported macromodel types are Spectre, SpectreS and Spice.

The graphical user interface (GUI) consists of the main window and five additional windows for geometry modeling and setup of solution parameters. The GUI also lets you import the lead frame and bond wire data that you can extract from the layout data or library.

For information about specific topics, consult the subsections below:

- [“The Four PKG Building Blocks” section on page 347](#) explains the four basic building blocks of PKG.
- The [“PKG Setup Requirements” section on page 350](#) explains the license and file requirements to set up and run PKG.
- [“The PKG RF IC Package Modeler Window” section on page 351](#) explains the main window of the PKG GUI.
- The [“Package Dimension Setup” section on page 355](#) explains how to set up the general package parameters such as the package body and the external lead frame dimensions.
- The [“Die Attach Setup” section on page 361](#) explains the die attach setup.
- The [“Internal Lead Frame Setup” section on page 362](#) explains the details of the setup of the internal pin leads and the export and import of lead frame files.
- The [“Bond Wire Setup” section on page 367](#) explains the details of how to set up the bond wires and how to export and import bond wire files.
- The [“Parameter Extraction” section on page 372](#) tells you about issues concerning the calculation of the capacitance, resistance, and inductance parameters.
- The [“Macromodel Generation” section on page 377](#) explains the schematic of the macromodel and the generation of macromodels for SpectreRF simulations including macromodel name and type, and it should include an optional die attach terminal.

- The [“Accuracy Analysis” section on page 381](#) explains accuracy limits and issues that affect the PKG accuracy.
- The [“Simulation of Package Models in the Affirma Analog Circuit Design Environment” section on page 381](#) explains the Affirma™ analog circuit design environment symbol for the RF package model and how to simulate RF IC packages in the design environment.

The Four PKG Building Blocks

The four large building blocks in PKG are

- Setup for package physical geometry modeling
- Mesh making for EM solvers
- The EM solvers
- The postprocessor.

Most of the physical geometry modeling is done with the PKG GUI, the rest is done with the PKG core algorithms.

Package Physical Geometry Modeling

PKG models the following three families of non-routable IC packages:

- The small outline IC (SOIC) family, including varieties like the shrunken small outline IC (SSOP), the thin shrunken small outline (TSSOP), and the very small out line IC (VSO)
- The quad flat pack (QFP) family, including LQFP, TQFP and others
- The dual inline package (DIP) family

The SOIC and QFT packages can have gull-wing or J-pin types. Each of these packages contain the following four components:

- The package body, which is often plastic or ceramic
- The lead frame, which includes the pins and die attach
- The bond wires
- The die

The die is modeled separately from the package model because it contains the actual circuits. The whole package geometry setup is divided into three steps: the package general

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

dimension setup, in which you specify the general dimension parameters; the internal lead frame setup, which includes both the pins and the die attach; and the bond wire setup.

Package General Dimension Setup

The package general dimension parameters include the dimensions of the package body, the die cavity and the external lead frame. The package body dimensions are length, width, and thickness. Because the die cavity is assumed to be located at the center of the package, you specify only its length, its width, and the bonding pad height, which is used to determine the height of the bonding pads. Because the external lead frame is usually standard, you set up all the pins using dimensions such as pin pitch, pin width, and pin thickness. However, for irregular packages, you can override these parameters for individual pins in the internal lead frame setup stage.

Internal Lead Frame Setup

The internal lead frame includes the die attach and the pins. The die attach is approximated with a rectangular plate placed at the center of the package. You must specify its length, width, thickness, and elevation.

The internal portion of the pins are approximated using a three-segment model. The first segment connects to the external part of the pin, and the third segment is where the bond wires are attached. The second segment, connecting the first and the third segment, is generated automatically from the first and the third segment data. You specify the first and third segments by their location, length, and width. The location of the first segment takes the standard values as specified by the pin pitch and pin width. You can override these data for special packages. You can also choose *Using Symmetry* to speed the setup.

Bond Wire Setup

You specify each bond wire by its length, the coordinates of its two end points, and its shape function. You can choose between two shape functions. One is quadratic, and the other is an exponential-distribution-like function. Each function is determined by three parameters, and these parameters are determined by the length and the two end points of the bond wire.

There are three types of bond wire connection. They are pin to die connection, pin to da down bonding, and die to da down bonding.

You can specify double and multiple bonding with bond wires of different lengths and thicknesses.

EM Solvers and Mesh Making

Capacitance Calculation

Because the RF IC package is usually electrically small at RF frequencies, the quasi-static approach creates a relatively accurate model, in which the capacitance, the inductance, and the resistance are the parameters of interest. Fastcap is used to calculate the package. Fastcap is a multipole, accelerated boundary element solver that solves the capacitance of the lead frame by calculating the electric charge distribution on the conductor surfaces and on the dielectric interfaces that are discretized into quadrilaterals. The edge effects of the electric charge distribution are modeled using a nonuniform mesh. The ground plane is approximated with a finite-sized rectangular plate when simulating the ground plane effect. Because the size of the bond wires are very small compared to the lead frame, they are ignored in the capacitance calculation. PKG provides three accuracy levels that use meshes of different density.

Inductance Calculation

Fasthenry is used to calculate the inductance and resistance of the pins and bond wires. Fasthenry is a PEEC-like (partial element equivalent circuit) solver with multipole acceleration. Because of the skin-depth effect, the inductance and resistance are frequency dependent and must be evaluated at each frequency point.

For packages with a large pin count number, evaluating the skin-depth effect at a relatively high frequency is very difficult. Because the skin-depth effect mainly affects the self-resistance and self-inductance, a two-step approach can reduce the computational difficulty. Because the mutual resistance is negligible and the mutual inductance does not change much with frequency, the mutual inductance is evaluated at the corner frequency, which is defined as the frequency at which the skin depth equals the pin thickness. The self-inductance and resistance are corrected at the working frequency by calculating the inductance and resistance of each pin separately. The results of the two steps are then combined together to create the impedance matrix at the working frequency.

Macromodel Generation

The fourth component of the package modeler is the post processor. The post processor creates the macromodel using data extracted from the capacitance and inductance calculations and creates the die attach terminal.

Because SpectreRF simulations produce time-domain solutions, the macromodel must be physical. The macromodel also must be adapted to SpectreRF simulations to avoid slowdown by high-frequency resonances.

For details about the macromodels, refer to the [“Macromodel Generation”](#) section on [page 377](#).

PKG Setup Requirements

Before running PKG, you must have the license, the executable, proper installation of the required files, and appropriate settings of the bin path.

License Checking

PKG checks for its own license.

Executables

The required executables are `pkg`, `fastcap`, and `fasthenry`. You must install them at `$CDSHOME/tools/dfII/bin`.

PKG GUI Files

The PKG GUI files are the following encrypted *tcl* files:

<code>pkgGUI.pkg</code>	<code>pkgDimensionSetup.pkg</code>
<code>pkgLeadframeSetup.pkg</code>	<code>pkgBondwireSetup.pkg</code>
<code>pkgFileIO.pkg</code>	<code>pkgSolve.pkg</code>

Initialization Files

The two required initialization files are

- `init.pkg`
- `init.coeffgen`

You must install them at `$CDSHOME/tools/dfII/etc/pkg`

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

There are two more optional initialization files.

- `~/ .pkginit`, which is installed in the user's home directory.
- `./ .pkginit`, which is installed at the current working directory.

When you start PKG, the initialization files, if they exist, are sourced in the following order:

1. `$CDSHOME/tools/dfII/etc/pkg/init.pkg`
2. `~/ .initpkg`
3. `./ .initpkg`

If a variable is set more than once, it takes the last set value.

Running pkg

PKG requires a working directory for the project. Usually the project name is the appropriate subdirectory name. If you do not specify a project name, the default subdirectory called `pkg.proj` is the working directory.

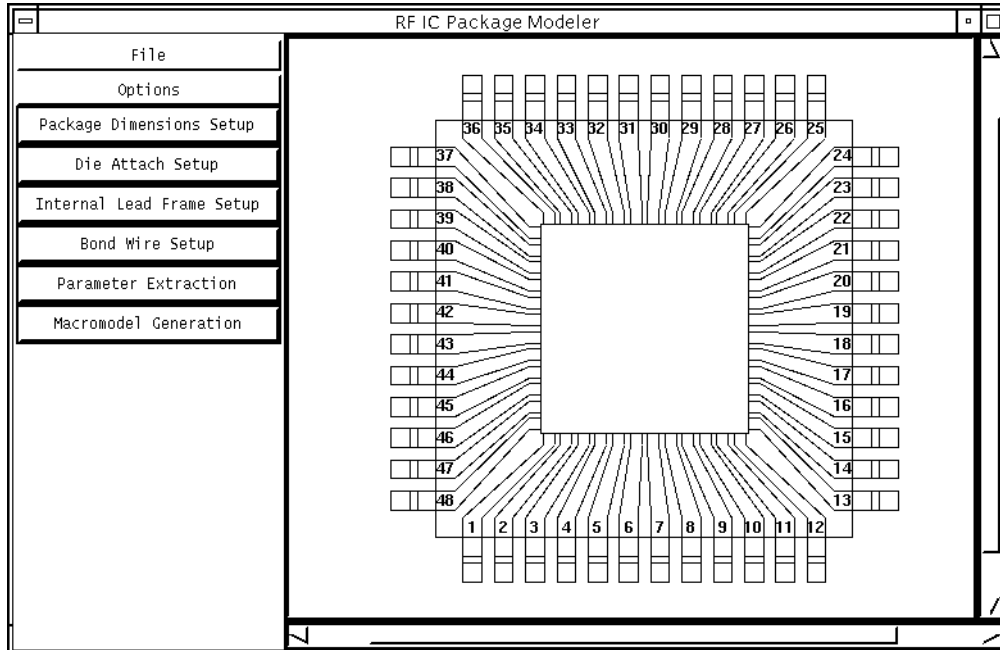
You can start PKG by performing the following command:

- In the Simulation Window, choose *Tools – RF – IC Package Modeler*.
- Or
- Type `pkg <projectName>` in a terminal window.

The PKG RF IC Package Modeler Window

As shown in [Figure 8-1](#) below, the RF IC Package Modeler form of the PKG GUI contains the two sections described in the bulleted list below:

Figure 8-1 The PKG RF IC Package Modeler Form



- Two menu buttons and six function buttons that control the option choices, the physical setup, EM calculation, and model creation
- The 2D display section on the right of the RF IC PackageModeler form where the current status of the package is displayed after each step

Menu Buttons

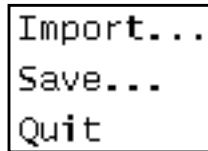
There are two pull-down menu buttons on the top left corner of the GUI

- *File*
- *Options.*

They are described below.

File

The *File* menu has the following three options:

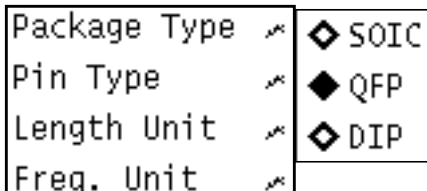


- **Import** lets you import an external package file. You specify the file name and path.
- **Save** lets you save the current package setup and calculation information to a file. The default file name is `pkg.dat` in the current project directory.
- **Quit** lets you quit PKG and return to the UNIX command line.

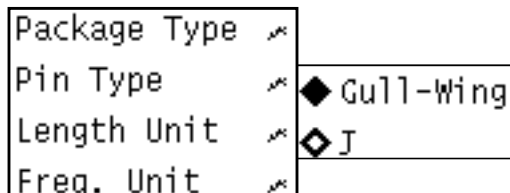
Options

The *Options* menu has four more menu buttons as follows.

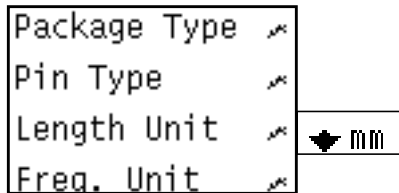
- **Package Type** lets you choose *SOIC*, *QFP* or *DIP*.



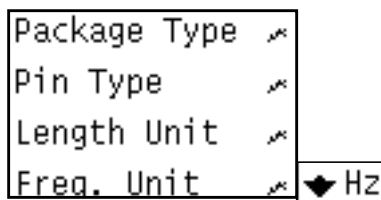
- **Pin Type** lets you choose *J* or *Gull-Wing* for the pin type for the SOIC and QFP packages.



- **Length Unit**, in this version, lets you use only *mm*. All the dimension quantities use the same unit for consistency.



- **Freq. Unit**, in this version, lets you use only *Hz*.



Function Buttons

There are six function buttons below the *Options* button.

- **Package Dimension Setup** launches the *Package Dimension Setup* form in which you specify the package, die cavity, and the external lead frame parameters.
- **Die Attach Setup** launches the *Die Attach Setup* form in which you specify the dimensions and elevation of the die attach.
- **Internal Lead Frame Setup** launches the *Lead Frame Setup* form in which you specify the internal portions of the pin lead. You make this specification on a pin-by-pin basis.
- **Bond Wire Setup** launches the *Bond Wire Setup* form. Each bond wire is assumed to be connected to a pin, and you specify the setup on a pin-by-pin basis.
- **Parameter Extraction** launches the *Parameter Extraction* form. External EM solvers are called to perform the capacitance calculation or calculate the resistance and inductance matrices.
- **Macromodel Generation** launches the *Macromodel Generation* form that lets you generate the macromodels in different formats. You can also create a die attach terminal at this time.

Package Dimension Setup

QFP Packages

The layout of the *Package Dimension Setup* form for QFP packages is shown in [Figure 8-2](#), in which a 48 pin rectangular QFP is displayed. The left portion of the display contains the data entries; the upper right portion of the display is the top view of the package; and the lower right portion is the side view of the package. Most data entries are illustrated in the display, which changes interactively with the entry data.

Figure 8-2 Package Dimension Setup Form, QFP Packages

PACKAGE

Length (L) mm

Width (W) mm

Thickness (T) mm

Elavation (e) mm

DIE CAVITY

Length (Ld) mm

Width (Wd) mm

Bonding Pad Height mm

PIN

Pins Per Side

Pitch (p) mm

Width (w) mm

Tilted Angle deg

Thickness (t) mm

Length (Lp1) mm

Length (Lp2) mm

Length (Lp3) mm

Data Entries for QFP Packages

Package Dimensions, Die Cavity Size and Bonding Pad Height

The first seven data entries on the *Package Dimension Setup* form are for the package dimensions, the die cavity size and the bonding pad height. They are explained below:

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

- **Length (L):** Length of the package body.
- **Width (W):** Width of the package body.
- **Thickness (T):** Thickness of the package body.
- **Elevation (e):** Distance from the bottom face of the package body to the lower horizontal pin segment.
- **Length (Ld):** Length of the die cavity formed by the leadframe. The cavity is assumed to be located in the center of the package body.
- **Width (Wd):** Width of the die cavity formed by the leadframe.
- **Bonding Pad Height:** Height of the upper face of the die, where the bonding pads are located. The reference height 0 is the bottom face of the package body. This height can be different from the height of the bond pad site on the pin, or on the die attach.

Lead Frame Dimensions

There are eight data entries on the *Package Dimension Setup* form for the lead frame dimensions.

- **Pins per Side:** For this entry, you specify one or two numbers. When you specify one number, all four sides have the same number of pins. If you specify two numbers separated by space, the first number represents the number of pins on sides 1 and 3 (bottom and top). The second number is the number of pins on sides 2 and 4 (left and right).
- **Pitch (P):** Distance between the left (or right) edges of two adjacent pins on the same side. If you specify one number, the four sides have the same pin pitch. If you specify two numbers, the first number is the pin pitch for the bottom and top sides, and the second number is the pin pitch for the left and right sides.
- **Width (w):** Accepts one or two numbers in the same manner as *Pitch (P)*. You can override this number by specifying values for individual pins.
- **Tilted Angle:** Angle of pins measured from the vertical direction. Positive angles represent outwardly tilted pins. Negative angles represent inwardly tilted pins.
- **Thickness (t):** Thickness of the pins. All the pins are assumed to have the same thickness.
- **Length (Lp1):** Length of the upper horizontal segment of the pins outside of the package body.
- **Length (Lp2):** Length of the vertical segment of the pins.

- **Length (Lp3):** Length of the lower horizontal segment of the pins.

SOIC Packages

The layout of a 14-pin SOIC package is shown in [Figure 8-3](#).

Figure 8-3 Package Dimension Setup Form, SOIC Packages

PACKAGE

Length (L) mm

Width (W) mm

Thickness (T) mm

Elevation (e) mm

DIE CAVITY

Length (Ld) mm

Width (Wd) mm

Bonding Pad Height mm

PIN

Pins Per Side

Pitch (p) mm

Width (w) mm

Tilted Angle deg

Thickness (t) mm

Length (Lp1) mm

Length (Lp2) mm

Length (Lp3) mm

Data Entries for SOIC Packages

All the data entries are the same as that of the QFP packages except that the following Lead Frame Dimension entries take only a single number as input:

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

- **Pins per Side:** For this entry, you specify one number since both sides have the same number of pins.
- **Pitch (P):** Distance between the left (or right) edges of two adjacent pins on the same side.
- **Width (w):** Accepts one number in the same manner as *Pitch (P)*.

DIP Packages

The layout of the DIP package dimension setup is shown in [Figure 8-4](#). The data entry part and the side view part are different from those of the SOIC packages.

Figure 8-4 Package Dimension Setup Form, DIP Packages

PACKAGE

Length (L) mm

Width (W) mm

Thickness (T) mm

Elevation (e) mm

DIE CAVITY

Length (Ld) mm

Width (Wd) mm

Bonding Pad Height mm

PIN

Pins Per Side

Pitch (p) mm

Width (w) mm

Tilted Angle deg

Thickness (t) mm

Length (Lp1) mm

Length (Lp2) mm

Length (Lp3) mm

Length (Lp4) mm

Width (w') mm

OK Cancel

Data Entries for DIP Packages

The data entry setup for DIP packages is almost the same as that of the SOIC packages except for the last five Lead Frame Dimension entries. Only the entries that are different from the SOIC packages are listed

- **Length (Lp1):** Length of the upper horizontal segment outside of the package.

- **Length (Lp2):** Length of the first vertical segment, which has the same width as the upper horizontal portion.
- **Length (Lp3):** Length of the tapered segment of the pin.
- **Length (Lp4):** Length of the lower vertical segment.
- **Width (w')**: Width of the lower vertical segment of the pin.

Die Attach Setup

The die attach is modeled with a rectangular plate placed horizontally in the center of the package. [Figure 8-5](#) shows the *Die Attach Setup* form for a 14-pin SOIC package.

Figure 8-5 Die Attach Setup Form

The screenshot shows a dialog box titled "Die Attach Setup". Inside the dialog, there is a diagram of a rectangular plate. The diagram shows a larger outer rectangle and a smaller inner rectangle. The inner rectangle is divided into two parts: a horizontal segment on the left and a vertical segment on the right. The horizontal segment is labeled "Lda" with a double-headed arrow indicating its length. The vertical segment is labeled "Wda" with a double-headed arrow indicating its width. Below the diagram, there are four input fields with labels and units:

Die Attach Length (Lda)	4	mm
Die Attach Width (Wda)	2.0	mm
Die Attach Thickness	0.144	mm
Die Attach Height	0.2	mm

At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Data Entries for SOIC Packages

- **Die Attach Length (Lda):** Length of the die attach
- **Die Attach Width (Wda):** Width of the die attach
- **Die Attach Thickness:** Thickness of the die attach
- **Die Attach Height:** Distance from the bottom face of the die attach to the bottom face of the package body.

You can add an optional die attach terminal to the package model during Macromodel Generation.

Internal Lead Frame Setup

You set up the internal lead frame on a pin-by-pin basis. Each pin is modeled with a three-segment approximation. You specify the two end segments with data entries, and the middle segment is generated automatically. For each individual pin, you can override the pin pitch and pin width information that you specify in the *Package Dimension Setup* form.

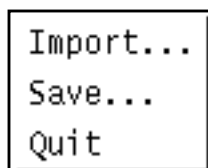
For different packages, you can turn symmetry on and off. For square QFP packages, when symmetry is used, only the first one-eighth of the pins must be specified. You can specify the rest using symmetry. You can override the symmetry specification by specifying new settings on the pins. For rectangular QFP packages, you must specify the first half of the pins on side 1 and side 2. Similarly, for SOIC and DIP packages, you must specify only one quarter of the pins.

When this form is first launched, the first pin is displayed and the data entries take the values of the first pin.

The *Lead Frame Setup* form for a 14-pin SOIC package is shown in Figure 8-6.

Menu Buttons

- **File:** This menu lets you save the current lead frame setup to an external file or import a previously setup lead frame file by specifying the file name and path.

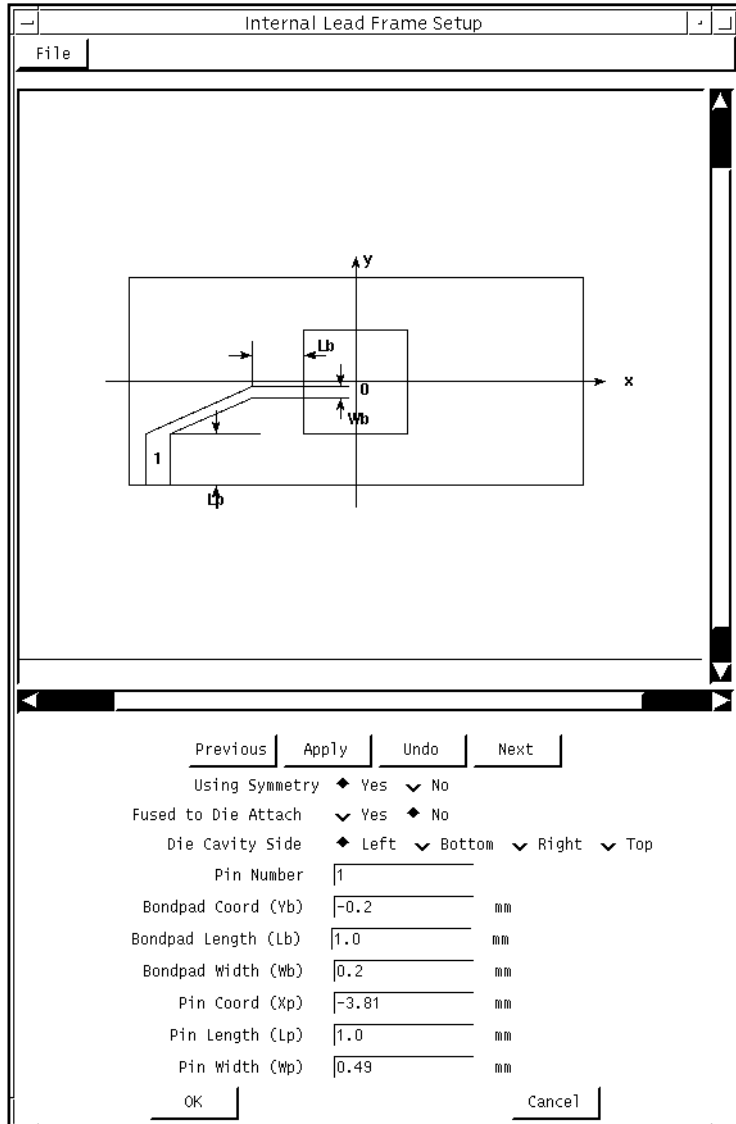


Function Buttons

There are four function buttons just below the lead frame display that control the pin-by-pin setup.

- **Previous:** Switches to the previous pin (the current pin number minus one. If the current pin is 1, then it switches to the last pin).
- **Accept:** Accepts data entry values for the current pin. Without pushing this, the entry values do not take effect.
- **Undo:** Cancel the specification and return to the initial values.
- **Next:** Switch to the next pin (the current pin number plus 1). If the current pin is the last pin, then switch to the first pin.

Figure 8-6 Lead Frame Setup Form



Data Entries for the Internal Lead Frame Setup

- **Using Symmetry:** Your choice turns symmetry on or off for the lead frame setup.
- **Fused to Die Attach:** There are two possible values, *yes* and *no*. The default value is *no*.
- **Die Cavity Side:** Specifies the side of the die cavity where the bondpad is located. There are four possible values: *left*, *bottom*, *right*, and *top*.

- **Pin Number:** The pin number of the current pin. This number increases or decreases when you push the *Next* or *Previous* buttons. You can specify the pin number directly to change to that specific pin.
- **Bondpad Coord (Yb):** If the bondpad is on the bottom or top side of the die cavity, then *Bondpad Coord* is the x-coordinate of the center of the bondpad. If the bondpad is on the left or right side of the die cavity, then *Bondpad Coord* is the y-coordinate of the center of the bondpad. The origin of the coordinate system is the center of the die cavity.
- **Bondpad Length (Lb):** Length of the end segment of the pin with bond wires connecting to it.
- **Bondpad Width (Wb):** Width of the end segment of the pin with bond wires connecting to it.
- **Pin Coord (Xp):** If the pin is on the bottom or top side of the package body, then *Pin Coord* is the x-coordinate of the center of the pin. If the pin is located at the left or right side of the package, then *Pin Coord* is the y-coordinate of the center of the pin. By changing this number, you move the location of the pin.
- **Length (Lp):** Length of the first segment of the pin inside the package.
- **Pin Width (Wp):** Width of the current pin. Changing this number changes the width of the current pin only.
- **OK:** Accepts all the changes you have made and exits from the *Lead Frame Setup* form.
- **Cancel:** Cancels all the changes you have made and returns to the original values.

More about Using Symmetry

When you choose symmetry and the internal lead frame is symmetric, you need specify only a quarter of the pins for SOIC, DIP, or rectangular QFP packages. For square QFP packages, you need specify only one-eighth of the pins.

Lead Frame File Format

You can write the internal lead frame into a file, or you can import a lead frame file that contains a lead frame you created externally. This feature allows you to manually edit the lead frame file without using the GUI or to get the lead frame from a layout library using a translator. The default location of the lead frame file is the current project directory. Because the definition of each pin depends on the package body and the die cavity dimensions, you must set those parameters correctly before you import an external lead frame file.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

In the lead frame file, each pin is described with two array elements, `pinPad(i)` and `bondPad(i)`. Each element contains the length, width, location of the two end segments of the pin. The `pinPad(i)` array element has the following format:

```
pinPad(i) = package_side Xp(Yp) pin_width  
pin_segment_length pin_number pin_fuse_to_da
```

```
bondPad(i) = bondpad_side Xb(Yb) bondpad_width  
bondpad_length
```

The leadframe file of a 14-pin SOIC package is as follows:

```
bondPad(1) = left -0.2 0.2 1.0 0  
pinPad(1) = bottom -3.81 0.49 1.0 1 0  
bondPad(2) = left -0.6 0.2 0.49  
pinPad(2) = bottom -2.54 0.49 1.0 2 0  
bondPad(3) = bottom -0.6 0.2 0.2  
pinPad(3) = bottom -1.27 0.49 0.2 3 0  
bondPad(4) = bottom -0.0 0.2 0.2  
pinPad(4) = bottom 0.0 0.49 0.2 4 0  
bondPad(5) = bottom 0.6 0.2 0.2  
pinPad(5) = bottom 1.27 0.49 0.2 5 0  
bondPad(6) = right -0.6 0.2 0.49  
pinPad(6) = bottom 2.54 0.49 1.0 6 0  
bondPad(7) = right -0.2 0.2 1.0  
pinPad(7) = bottom 3.81 0.49 1.0 7 0  
bondPad(8) = right 0.2 0.2 1.0  
pinPad(8) = top 3.81 0.49 1.0 8 0  
bondPad(9) = right 0.6 0.2 0.49  
pinPad(9) = top 2.54 0.49 1.0 9  
bondPad(10) = top 0.6 0.2 0.2  
pinPad(10) = top 1.27 0.49 0.2 10 0  
bondPad(11) = top 0.0 0.2 0.2  
pinPad(11) = top 0.0 0.49 0.2 11 0  
bondPad(12) = top -0.6 0.2 0.2  
pinPad(12) = top -1.27 0.49 0.2 12 0  
bondPad(13) = left 0.6 0.2 0.49  
pinPad(13) = top -2.54 0.49 1.0 13 0  
bondPad(14) = left 0.2 0.2 1.0  
pinPad(14) = top -3.81 0.49 1.0 14 0
```

Bond Wire Setup

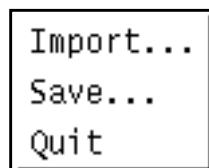
Like the lead frame, you set up the bond wires individually. Each bond wire is determined by its two end points, its length, its shape configuration, and its diameter. For the current version, only round bond wires are permitted. There are two different bond wire shape configurations, the quadratic and the exponential-distribution-like shape. There are three bondwire connections. They are pin-to-die, pin-to-da, and die-to-da connection.

The layout of the *Bond Wire Setup* form is shown in [Figure 8-7](#).

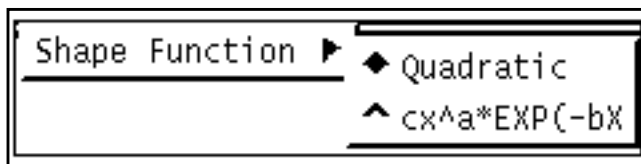
Menu Buttons

There are two menu buttons on the top of the form.

- **File:** This button lets you import and export bond wire data. Unless specified otherwise, the default location of the bond wire file is under the project directory. There is a difference between importing a bond wire file and a lead frame file. The new bond wire values do not take effect unless you push the *OK* button.



- **Options:** In the current version, there is only one variable under this button, *Shape Function*, which is a radio button with two values, *Quadratic* and $cx^a*EXP(-bx)$, which describe the shape of the bond wires.



Function Buttons

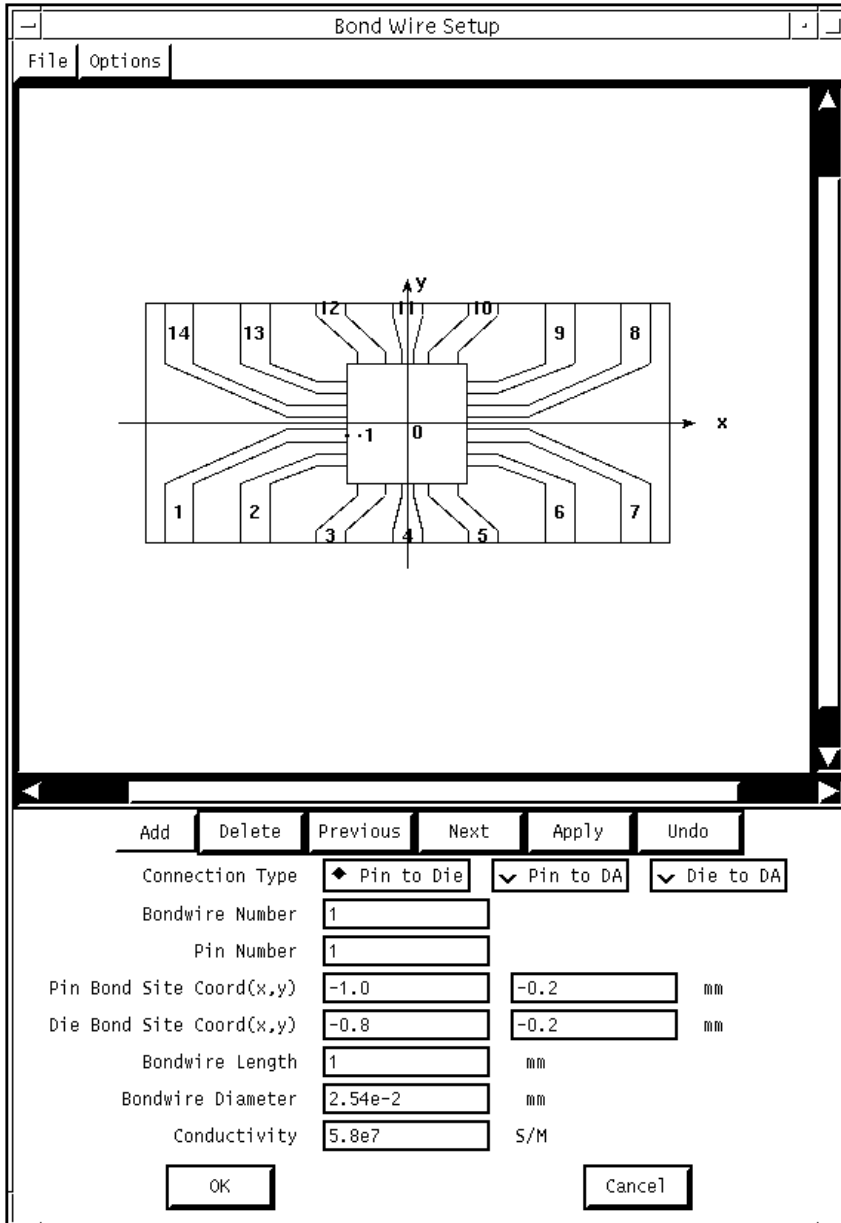
Under the display portion, there are five function buttons that are similar to the buttons in the *Lead Frame Setup* form. These buttons control setup of individual bond wires. There are two function buttons at the bottom of the form that let you accept the setup and exit or cancel the changes and exit.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

- **Add:** You must use the *Add* button if you add type-in values for a pin. After you initially add the information, you can edit the fields. This menu button can have two values: *single* or *multiple*. When you choose *single*, a new bondwire is created whose initial value is taken from that of the previous bondwire or the pin it connects to. When you choose *multiple*, a bondwire is added to the current bondwire so as to form multiple bonding. If a current bondwire does not exist, *multiple* has the same effect as *single*.
- **Delete:** Deletes the current bondwire.
- **Previous:** Switch to the setup of the bond wires connected to the previous pin. If the current pin is the first pin, then switch to the last pin.
- **Next:** Switch to the setup of the bond wires connected to the next pin. If the current pin is the last, then switch to the first pin.
- **Apply:** Accept the values of the bond wires connected to the current pin.
- **Undo:** Undoes any changes to the current pin.
- **OK:** Accept the new values for the bond wires and exit.
- **Cancel:** Cancel the changes and exit. All bond wires keep their original values.

Figure 8-7 Bond Wire Setup Form



Data Entries for the Bond Wire Setup

In the first line of the data entry area, there are three radio buttons for the bond wire connections. They are pin-to-die connection, pin-to-die attach, and die-to-die attach (down bonding).

- **Bondwire Number:** ID number for the current bondwire.

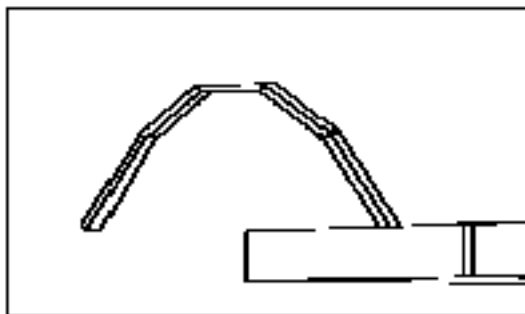
- **Pin Number:** Current pin number to which the bond wires are attached.
- **Pin Bond Site Coord (x, y):** Coordinates of the bond pad on the pin.
- **Die Bond Site Coord (x, y):** Coordinates of the bond pads on the die. The reference origin of the local coordinate system is the center of the die cavity.
- **Bondwire Length:** Length of the bond wire(s). Given the coordinates of the bond pads, the length must be big enough; otherwise, an error condition occurs.
- **Bondwire Diameter:** Diameter of the bond wires.
- **Conductivity:** Conductivity of the bond wires. The default metal for conductivity is gold, which is The unit is S/m.

Bond Wire Configurations

PKG supports two configurations of bond wires. One configuration is defined by the quadratic function, the other is defined by an exponential-distribution-like function. For the two shape functions, the three unknown parameters a , b , and c are determined from the two end points and the length of the bond wire

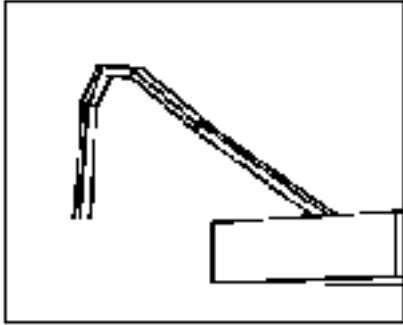
Quadratic function ($y=ax^2 +bx+c$)

Figure 8-8 Quadratic Shape Function



Exponential distribution $cx^a \cdot \text{Exp}(-bx)$

Figure 8-9 Exponential-Distribution-Like Shape Function



Bond Wire File Format

PKG lets you save the current bond wire setup to an external file and also import the data from an external source, such as a file from a database, or a file edited by hand. The file must conform to the proper the format. The format and a sample bond wire data file are shown below.

```
bw(i) = pin_number connection_type x_coord y_coord z_coord
        pin_bondpad_displacement bondwire_length
        bondwire_conductivity
```

There are three possible values for the connection type: `pin2die`, `pin2da`, and `die2die`.

When the connection type is `pin2die`, x_1 and y_1 are the coordinates of the bonding site on the end of the pin, x_2 and y_2 are the coordinates of the bonding site on the die. The die pad number is the index number of the pad the bondwire connects to.

When the connection type is `pin2da`, x_1 and y_1 are the coordinates of the bonding site on the end of the pin, and x_2 and y_2 are the coordinates of the bonding site on the die attach because the bonding wire does not connect to the die. The die pad number is -1.

When the connection type is `die2die`, x_1 and y_1 are the coordinates of the bonding site on the die, x_2 and y_2 are the bonding site coordinates on the die attach. The die pad number is the index number of the pin on the die. Because the bondwire does not connect to a pin, the die pad number is -1.

```
numBondwires = 14
bw(1) = 1 pin2die -1.0 -0.2 -0.8 -0.2 1 2.54e-2 5.8e7 1
bw(2) = 2 pin2die -1.0 -0.6 -0.8 -0.6 1 2.54e-2 5.8e7 2
bw(3) = 3 pin2die -0.6 -1.0 -0.6 -0.8 1 2.54e-2 5.8e7 3
```

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

```
bw(4) = 4 pin2die -0.0 -1.0 -0.0 -0.8 1 2.54e-2 5.8e7 4
bw(5) = 5 pin2die 0.6 -1.0 0.6 -0.8 1 2.54e-2 5.8e7 5
bw(6) = 6 pin2die 1.0 -0.6 0.8 -0.6 1 2.54e-2 5.8e7 6
bw(7) = 7 pin2die 1.0 -0.2 0.8 -0.2 1 2.54e-2 5.8e7 7
bw(8) = 8 pin2die 1.0 0.2 0.8 0.2 1 2.54e-2 5.8e7 8
bw(9) = 9 pin2die 1.0 0.6 0.8 0.6 1 2.54e-2 5.8e7 9
bw(10) = 10 pin2die 0.6 1.0 0.6 0.8 1 2.54e-2 5.8e7 10
bw(11) = 11 pin2die 0.0 1.0 0.0 0.8 1 2.54e-2 5.8e7 11
bw(12) = 12 pin2die -0.6 1.0 -0.6 0.8 1 2.54e-2 5.8e7 12
bw(13) = 13 pin2die -1.0 0.6 -0.8 0.6 1 2.54e-2 5.8e7 13
bw(14) = 14 pin2die -1.0 0.2 -0.8 0.2 1 2.54e-2 5.8e7 14
```

Save the Current Setting



Once you set up the bond wire, go back to the File menu in the RF IC Package Modeler form before starting the parameter solution process. This step prevents loss of data if the solution process fails.

Parameter Extraction

Once you set up the physical geometry, you can calculate capacitance, inductance, or both. You specify the calculations you want on the *Parameter Extraction* form, which is shown below.

Figure 8-10 Parameter Extraction Form

Parameter Extraction

CAPACITANCE

Calculate Capacitance yes no

Ground Plane Effect? yes no

Dielectric Constant

Capacitance Accuracy liberal moderate conservative

INDUCTANCE

Calculate Inductance yes no

Inductance Accuracy liberal moderate conservative

Leadframe Conductivity S/M

Working Frequency Hz

Capacitance Calculation

Once you set up the physical geometry, you can calculate the capacitance matrix of the lead frame. Because the capacitance of the bond wires is small compared to that of the lead frame, bond wire capacitance is ignored in the capacitance calculation.

Data Entries and Buttons for Capacitance Calculation

- **Calculate Capacitance:** Click on *yes* to calculate the capacitance.
- **Ground Plane Effect:** Includes the ground plane effect in the capacitance calculation. If you choose *yes*, you must specify the distance between the ground plane and the package. This distance must be large enough so that the pins do not touch the ground plane. Use this choice to include PCB effects.

- **Dielectric Constant:** Dielectric permittivity of the package body.
- **Gnd/pkg Distance:** This data entry is used only after you click yes on *Ground Plane Effect*. It is the distance between the ground plane and the bottom face of the package body.
- **Capacitance Accuracy:** Accuracy level that controls the mesh density. *liberal* uses a smaller mesh and therefore requires less memory and runs faster. *conservative* has the highest mesh density, uses more memory, and runs slower. *moderate* lies midway between the two and is the recommended default value.

Accuracy Versus Mesh Size

The accuracy level determines the size of the mesh. Level *Liberal* creates a mesh as shown in [Figure 8-11](#), in which the mesh of a 4-pin SOIC package is plotted as an illustration. Each pin width is divided into three panels. Fewer panels are used for thinner segments. [Figure 8-12](#) shows the mesh at level *moderate*, in which each pin width is divided into five panels, and the ratio of the adjacent panel width is 2. The grid size on the die attach is 1, and the grid size is 1/2 on the pin. The mesh for accuracy level *conservative* is not shown here because of the amount of computational work needed to plot the mesh.

Figure 8-11 Mesh for 4-Pin SOIC Package at Accuracy Level *liberal*

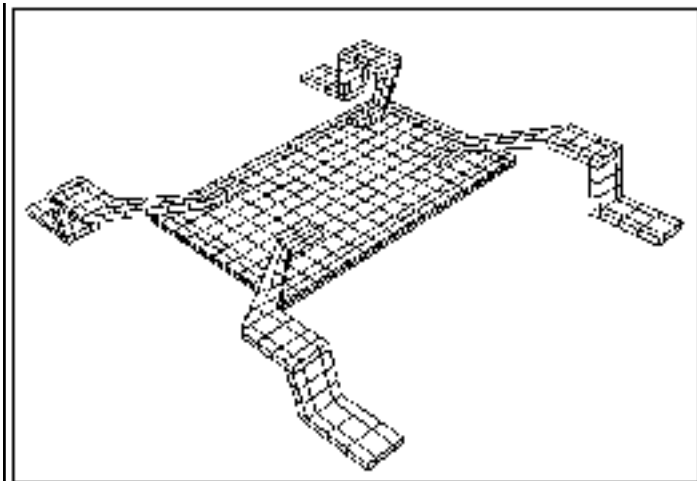
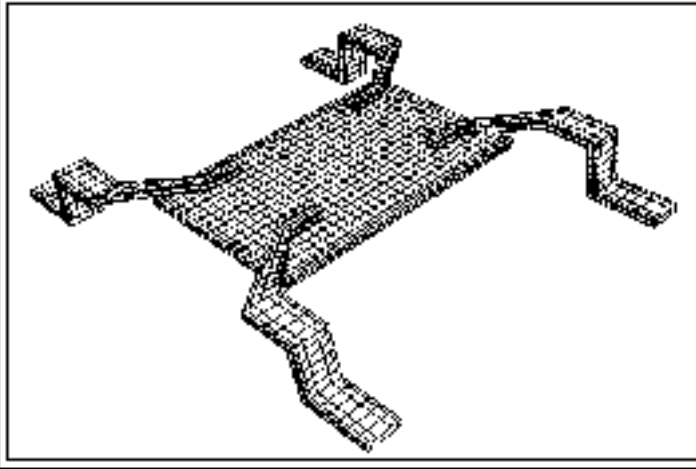


Figure 8-12 Mesh for 4-Pin SOIC Package at Accuracy Level *moderate*



You can stop the solve process by clicking on *Cancel*.

Inductance Calculation

You must specify the following parameters before starting the calculation.

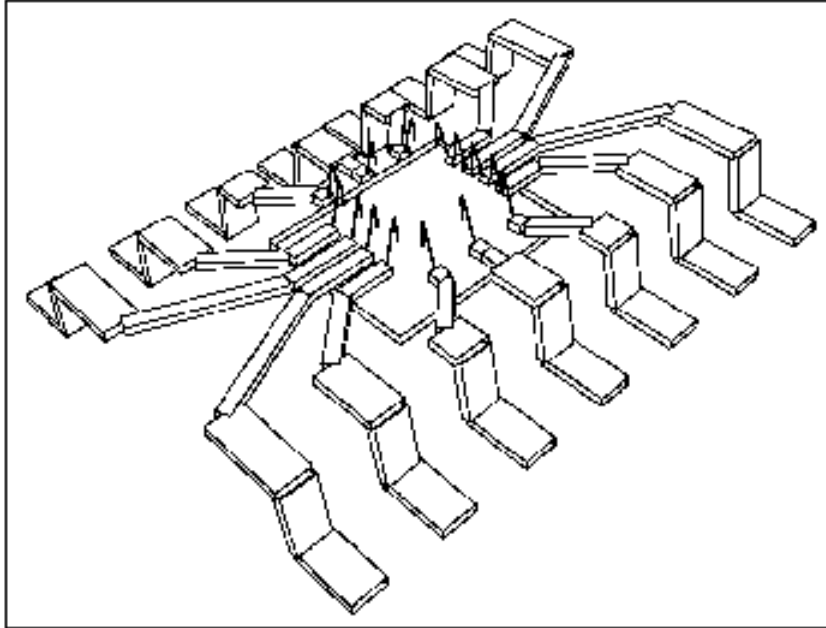
Data Entries and Buttons for Inductance Calculation

- **Calculate Inductance:** Click on *yes* to calculate the inductance.
- **Inductance Accuracy:** Mesh refinement control. There are three accuracy levels, *liberal*, *moderate*, and *conservative*. Each accuracy level corresponds to a different mesh size.
- **Leadframe Conductivity:** The conductivity of the lead frame, including the pins and the die attach. The unit is *S/m*.
- **Working Frequency:** Actual working frequency of the macromodel. The unit is *Hz*.

Mesh Size Versus Accuracy

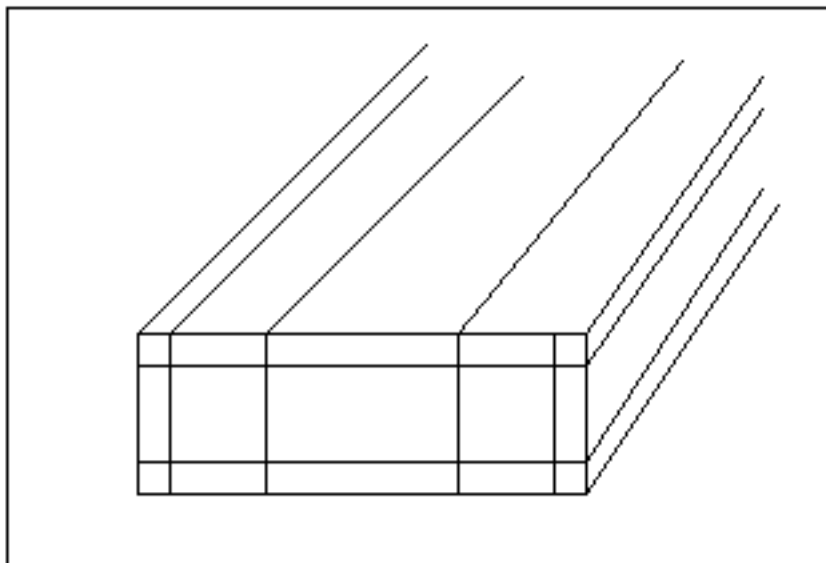
The mesh for inductance calculation is shown in [Figure 8-13](#), in which each pin is approximated with uniform rectangular solids. Each rectangular solid is further divided into smaller segments.

Figure 8-13 Mesh for Inductance Calculation



The discretization of a segment for accuracy level *liberal* is shown in [Figure 8-15](#). For *liberal*, the width of the segment is divided into five segments. Accuracy levels *moderate* and *conservative* correspond to seven and nine segments per width, respectively. However, you can reset these values in the initialization files.

Figure 8-14 Segment Discretization for Accuracy Level *Liberal*



The Inductance Solving Process

Because of the skin-depth effect, the electric current crowds out to the surface as the frequency increases. Consequently, you must generate a very fine mesh to catch the fast variation of current distribution in the vicinity of the surface. However, this large mesh makes the solving process very slow, or in some cases, impossible.

PKG employs a two-step solving process. The first step is to calculate the whole package at the corner frequency. Because the electric current is nearly uniform, a small mesh of a couple of segments per width is sufficient. This makes the solving process very fast. Because the mutual inductances between pins and vias do not change much when the frequency increases, the mutual inductance calculated at the low frequency can be used for the much higher working frequency. In the second step, to account for the skin depth effect, each pin (plus the bond wire, if present) is solved independently with a finer mesh at the working frequency. The impedance matrix at the working frequency can be obtained by combining the two sets of results.

Function Buttons for Parameter Extraction

There are three buttons at the bottom of the Parameter Extraction form. They are *Solve*, *Cancel*, and *Close*.

- **Solve** starts the parameter extraction calculations. Capacitance is calculated first, then inductance. If the capacitance or inductance has already been calculated, a warning message appears.
- **Cancel** stops the parameter extraction calculations.
- **Close** closes the parameter extraction form.

Macromodel Generation

After the parameter extraction has completed, you can generate the macromodel by pushing the *Macromodel Generation* button on the *RF IC Package Modeler* form. Pushing the button brings up the *Macromodel Generation* form shown in [Figure 8-15](#). Before you create the macromodel, you must specify the items described below:

Data Entries and Buttons for Macromodel Generation

- **Coupled Pins Per Side:** Number of the most adjacent pins on each side of the current pin. If you choose 0, the coupling between pins is ignored. If the number you specify is

bigger than half the number of pins, the coupling among all the pins is included in the macromodel.

- **Macromodel Name:** Name of the file containing the actual macromodel netlist. A warning message flashes if a file with the same name already exists.
- **Macromodel Type:** Currently you can choose among the following three different model types: *Spectre*, *SpectreS* and *Spice*.
- **Create DA Terminal:** Creates a model with a die attach node if you choose *yes*.
- **Create:** Creates a macromodel. Issues a warning if a macromodel with the same name already exists.
- **Close:** Closes the current form.

Die Attach Terminal

When you click *Yes* to create a DA terminal, the macromodel and symbol are created with a die attach node. Two subcircuits created for the same component one with and one without die attach terminals are not interchangeable because a subcircuit created for a component with a die attach terminal contains an extra node.

Figure 8-15 Macromodel Generation Form

The image shows a dialog box titled "Macromodel Generation". It contains the following elements:

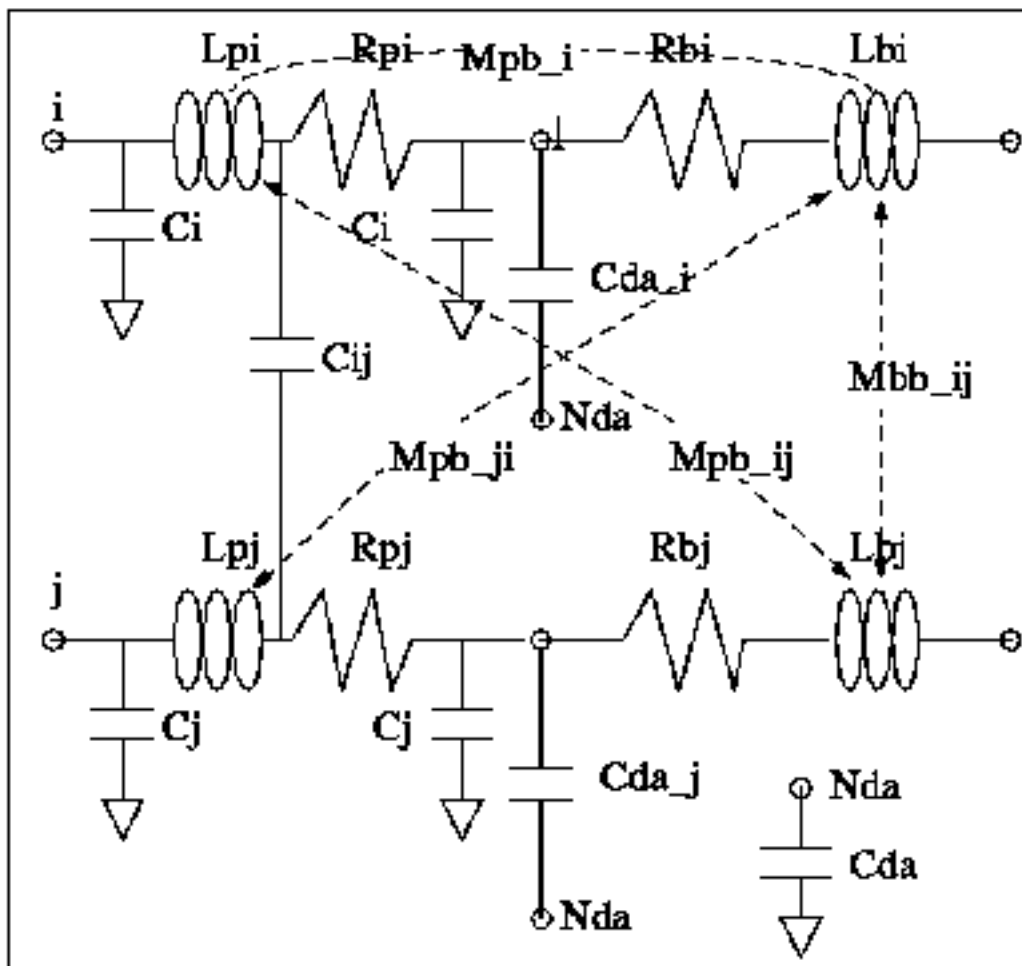
- Coupled Pins Per Side:** A text input field containing the value "0".
- Macromodel Name:** A text input field containing the value "rficpkg".
- Macromodel Type:** A radio button group with three options: "Spectre", "SpectreS", and "Spice".
- Create DA terminal?:** A radio button group with two options: "yes" and "no".
- Buttons:** Two buttons at the bottom, "Create" and "Close".

Macromodel Schematic

The schematic of the macromodel is shown in Figure 8-16. For simplicity, only three ports are shown, where ports i , j , and k correspond to pin i , pin j , and pin k . Pin i and pin k have bond wires attached to them, whereas pin j is left open inside the package. Each pin is approximated with a PI circuit, with C_i representing the self-capacitance of the pin, and L_{pi} and R_{pi} representing the series inductance and resistance of the pin, respectively. If a bond wire is connected to the pin, a series RL is attached to the internal end of the pin. For pin j , which does not have a bond wire attachment, a large resistor is added for the current path from the pin to ground.

The coupling between pins and bond wires are reflected by the coupling capacitors and mutual inductors. The full coupling for the number of coupled pins at each side is contained in the macromodel subcircuit.

Figure 8-16 Schematic of the Macromodel



Note: This illustration is included for instructional purposes. PKG creates only a netlist, not a schematic of the macromodel.

High Frequency Resonance Damping for the Spectre Simulator

As shown in [Figure 8-17](#), the *PI* equivalent circuit might form a resonator whose resonance frequency is usually much higher than the working frequency. This resonance can be modeled in Spectre simulation, and this results in a very small time-step that slows the simulation significantly. You can damp out the resonance by lowering the *Q* factor of the pin as an inductor. Although the series resistance *R_{pi}* can be combined into *L_{pi}* to form a lossy inductor, its *Q* value is usually too high for the resistance to damp out quickly. To solve this problem, a parallel resistor of relatively large value is attached to the inductor, as shown in [Figure 8-17](#) [Figure 8-18](#) [Figure 8-18](#). The value of the resistor is determined by the specified *Q* value at the resonance frequency, which is computed from *L_{pi}* and *C_{pi}*. In this version, the *Q* value is set at 10 in the initialization file.

Figure 8-17 PI Equivalent Circuit of the Pin

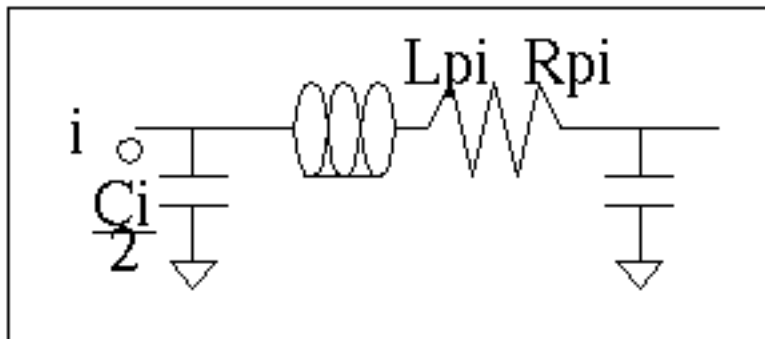
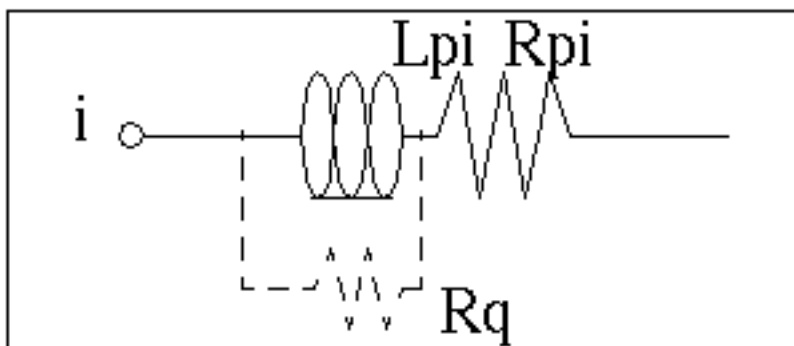


Figure 8-18 Reducing the Q Factor by Attaching a Parallel Resistor



Accuracy Analysis

One error source for the macromodel is the quasi-static assumption, in which the size of the package is assumed to be sufficiently small electrically so that the field retardation can be ignored. This assumption determines the upper frequency limit. Therefore, the upper frequency limit really depends upon which part of the package is of interest. If the long range coupling is of interest, then the upper frequency limit is lower because of the longer distance between the pins. If only the coupling between the neighboring pins is of interest, the upper frequency limit can be higher due to the smaller distance between the pins.

The error with physical geometry mainly results from the three-segment approximation of the pins, the rectangular plate approximation of the die attach, and the ignoring of the bonding pad on the die, and so on. The error is mostly with the dimensions of the conductors, such as length, thickness, and so on.

Another source of error is electrical modeling. For example, in inductance calculation, only the axial current is considered, and the transverse current is ignored. (Transverse current is small at low frequencies.)

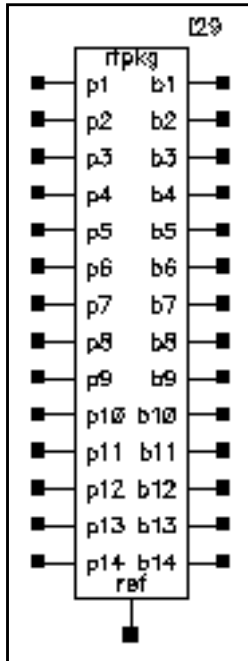
The error with capacitance calculation is small, usually under 1 percent.

The error with inductance calculation depends on the working frequency. Usually, error with inductance is very small. The error is relatively larger for resistance of the pins due to the skin-depth effect. However, the error has little effect on the accuracy of the result because of the attachment of a resistor that forces a decrease in the inductor Q factor.

Simulation of Package Models in the Affirma Analog Circuit Design Environment

The symbol *rfpkg* and the [Edit Object Properties](#) form for the cell view are shown in [Figure 8-19](#) and [Figure 8-20](#). The left side of the symbol represents the terminals of the pin leads outside of the package. The ports on the right side of the symbol represent the bonding pads on the die.

Figure 8-19 Affirma analog circuit design environment symbol for Rf Package Models



To instantiate the package model, perform the following steps:

- Decide which package parameters to use in the circuit
These parameters include the package dimensions, lead frame, bond wires dimension, working frequency, number of coupling pins, die attach terminal and so on.
- Run PKG.
Each time you run PKG, set the model file name to a unique name.
- For each package model in your schematic, place an *rfpkg* symbol cell view and connect the pins and internal ports to the rest of the schematic.
- For each *rfpkg* instance, set the *File Name* parameter to the corresponding package model.

If you modify any package parameters, the working frequency, the number of coupling pins considered, or you add or remove the die attach terminal, you must re-create the macromodel and update the appropriate instances, depending on the PKG output.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

Figure 8-20 The *rfpkg* Edit Object Properties Form

Edit Object Properties

OK
Cancel
Apply
Defaults
Previous
Next
Help

Apply To only current instance

Show system user CDF

Browse
Reset Instance Labels Display

Property	Value	Display
Library Name	<input type="text" value="rfExamples"/>	<input type="checkbox"/> off
Cell Name	<input type="text" value="rfpkg"/>	<input type="checkbox"/> off
View Name	<input type="text" value="spectre"/>	<input type="checkbox"/> off
Instance Name	<input type="text" value="I28"/>	<input type="checkbox"/> off

Add
Delete
Modify

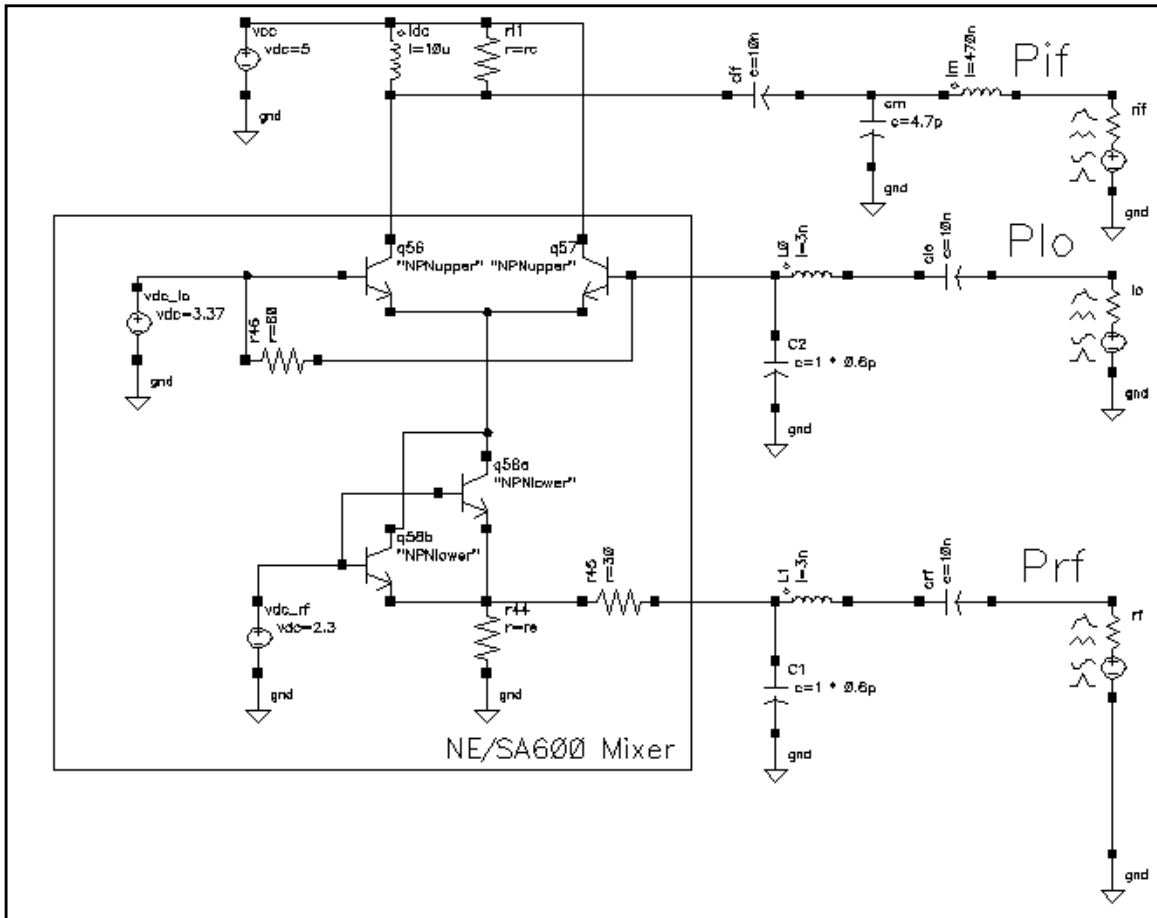
CDF Parameter	Value	Display
Die Attach Terminal	<input type="checkbox"/>	<input type="checkbox"/> off
Number of External Pin Ports	<input type="text" value="14"/>	<input type="checkbox"/> off
Number of Internal Bonding Pads	<input type="text" value="14"/>	<input type="checkbox"/> off
Model Name	<input type="text" value="rficpkg"/>	<input type="checkbox"/> off

As an example, [Figure 8-21](#) shows the schematic of the Philips ne600 mixer/LNA with an *rfpkg* cell view, which represents a 14-lead SOIC package.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

Figure 8-21 Schematic of NE600 Mixer



Example of Package Modeling With a Mixer Using PKG

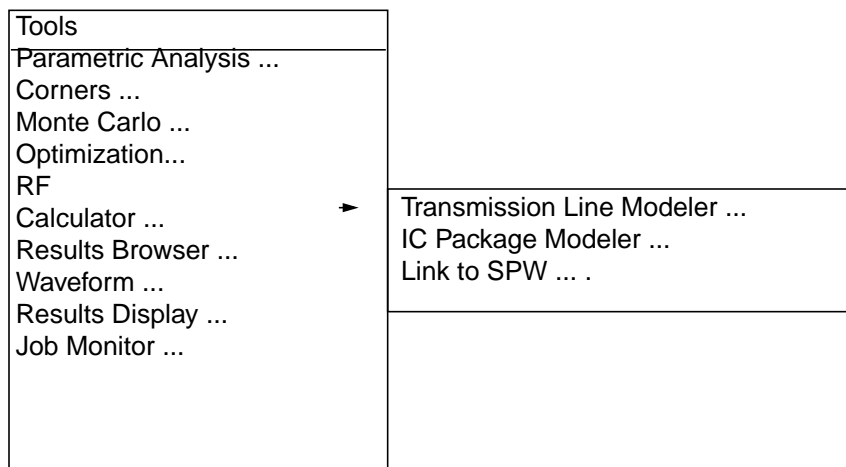
This section shows you how to use PKG to perform package modeling with the ne600 mixer. For more information about this circuit, see Chapter 4. The example assumes that you have sufficient knowledge about the Affirma analog circuit design environment to bring up the ne600 schematic and also the Simulation window. The example also assumes that you edited the value of the variable *frf* to equal 920M. If this is not the case, see Chapter 4 for more information.

Creating the Package Model With PKG

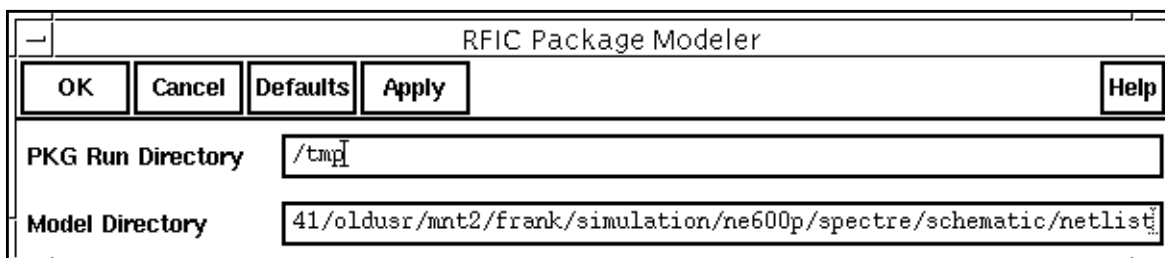
Starting PKG

- Choose *Tools – RF – IC Package Modeler* in the Simulation Window.

The *RF IC Package Modeler* form appears.



A form appears that lets you specify the PKG run directory and also the directory in which PKG places the model. Check the values in the fields and change them if necessary.

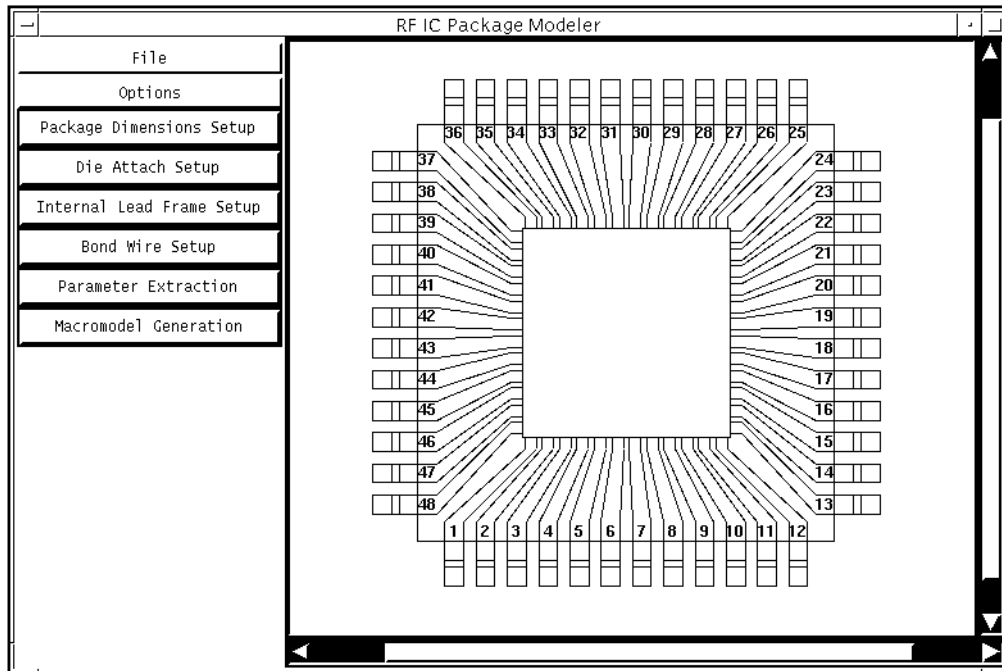


Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

- Click *OK* or *Apply* to approve the values in the *PKG Run Directory* and *Model Directory* fields.

The *RF IC Package Modeler* form appears.



Filling in the RF IC Package Modeler Form

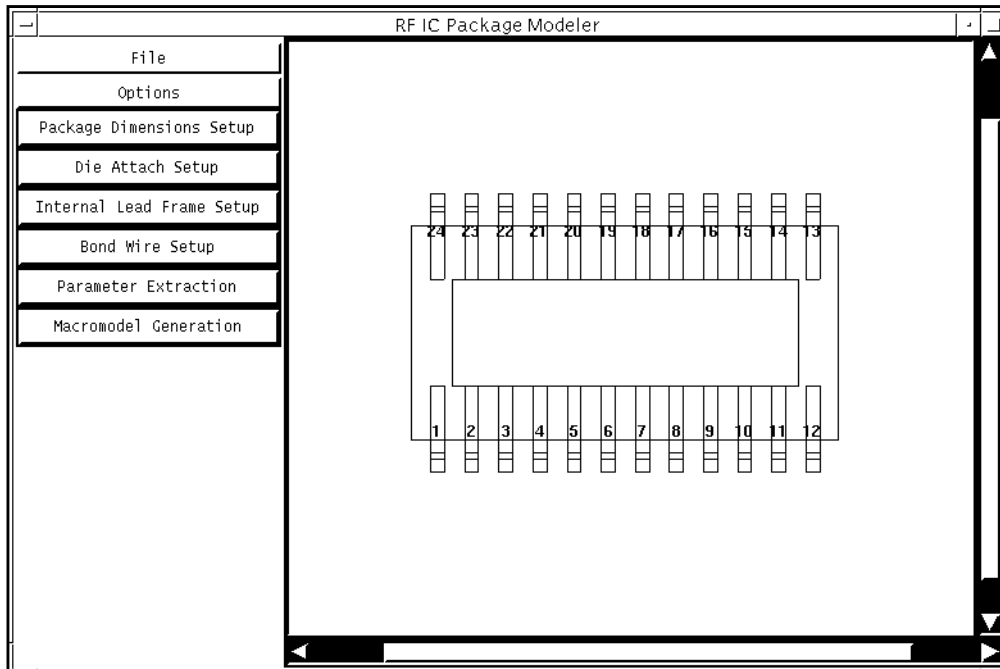
In the *RF IC Package Modeler* form, perform the following:

1. Choose *Options – Package Type – SOIC*.
2. Choose *Options – Pin Type – Gull-Wing*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The diagram on the *RF IC Package Modeler* form changes to reflect your modifications. When you edit subsequent forms, the diagram on the *RF IC Package Modeler* form is updated to reflect these changes also.



3. Click on *Package Dimensions Setup*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The *Package Dimensions Setup* form appears.

Package Dimensions Setup
✖

PACKAGE

Length (L) mm

Width (W) mm

Thickness (T) mm

Elevation (e) mm

DIE CAVITY

Length (Ld) mm

Width (Wd) mm

Bonding Pad Height mm

PIN

Pins Per Side

Pitch (p) mm

Width (w) mm

Tilted Angle deg

Thickness (t) mm

Length (Lp1) mm

Length (Lp2) mm

Length (Lp3) mm

Filling In the Package Dimensions Setup Form

1. In the *Package Dimensions Setup* form, type values as shown in the tables below.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

As you type values, the pictures on the right side of the form change to reflect the values you type.

Package Values Labels	Values You Type
------------------------------	------------------------

<i>Length (L)</i>	8.75
-------------------	------

<i>Width (W)</i>	4.0
------------------	-----

<i>Thickness (T)</i>	1.45
----------------------	------

<i>Elevation (e)</i>	0.2
----------------------	-----

Die Cavity Labels	Values You Type
--------------------------	------------------------

<i>Length (Ld)</i>	2.0
--------------------	-----

<i>Width (Wd)</i>	2.0
-------------------	-----

<i>Bonding Pad Height</i>	0.5
---------------------------	-----

Pin Labels	Values You Type
-------------------	------------------------

<i>Pins Per Side</i>	7
----------------------	---

<i>Pitch (P)</i>	1.27
------------------	------

<i>Width (W)</i>	0.49
------------------	------

<i>Tilted Angle</i>	8
---------------------	---

<i>Thickness (t)</i>	0.127
----------------------	-------

<i>Length (Lp1)</i>	0.2
---------------------	-----

<i>Length (Lp2)</i>	0.8
---------------------	-----

<i>Length (Lp3)</i>	0.875
---------------------	-------

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The completed form appears like the one below:

PACKAGE

Length (L) mm

Width (W) mm

Thickness (T) mm

Elavation (e) mm

DIE CAVITY

Length (Ld) mm

Width (Wd) mm

Bonding Pad Height mm

PIN

Pins Per Side

Pitch (p) mm

Width (w) mm

Tilted Angle deg

Thickness (t) mm

Length (Lp1) mm

Length (Lp2) mm

Length (Lp3) mm

OK Cancel

The schematic diagram shows a top view of a package with dimensions L, W, Ld, Wd, and w. It also shows a side view with dimensions T, e, Lp1, Lp2, Lp3, and t. The package has a central die cavity and pins on all four sides.

2. In the *Package Dimensions Setup* form, click *OK*.

The form closes.

3. In the *RF IC Package Modeler* form, click on *Die Attach Setup*

The *Die Attach Setup* form appears.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

Filling In the Die Attach Setup Form

1. In the *Die Attach Setup* form, type the values in the following table:

Die Attach Setup Labels	Values You Type
<i>Die Attach Length (Lda)</i>	2.0
<i>Die Attach Width (Wda)</i>	2.0
<i>Die Attach Thickness</i>	0.1024
<i>Die Attach Height</i>	0.2

The completed form appears like this.

The screenshot shows the 'Die Attach Setup' dialog box. At the top, the title bar reads 'Die Attach Setup'. The main area contains a diagram of a rectangular die with a smaller rectangle inside it. The inner rectangle's width is labeled 'Lda' with a double-headed arrow, and its height is labeled 'Wda' with a double-headed arrow. Below the diagram, there are four input fields with labels and units:

- Die Attach Length (Lda) mm
- Die Attach Width (Wda) mm
- Die Attach Thickness mm
- Die Attach Height mm

At the bottom of the dialog box, there are two buttons: 'OK' and 'Cancel'.

2. In the *Die Attach Setup* form, click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

3. In the *RF IC Package Modeler* form, click on *Internal Lead Frame Setup*.
The *Internal Lead Frame Setup* form appears.

Filling-In the Internal Lead Frame Setup Form

In the *Internal Lead Frame Setup* form

1. Highlight *yes* for *Using Symmetry*.
2. Highlight *no* for *Fused to Die Attach*.
3. Specify the values in the following table.

Internal Lead Frame Setup Labels	Values
Die Cavity Side	Left
<i>Pin Number</i>	1
<i>Bondpad Coord (Yb)</i>	-0.2
<i>Bondpad Length (Lb)</i>	1.0
<i>Bondpad Width (Wb)</i>	0.2
<i>Pin Coord (Xp)</i>	-3.81
<i>Pin Length (Lp)</i>	1.0
<i>Pin Width (Wp)</i>	0.49

4. Click on *Apply*.

Because you have chosen *Using Symmetry*, PKG places your settings on pins 7, 8, and 14, as well as pin 1. It duplicates your settings in all four quadrants each time you apply them.

5. Click on *Next*.

The form changes to let you specify values for pin 2.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

6. Specify the values in the following table:

Internal Lead Frame Setup Labels	Values
Die Cavity Side	Left
<i>Pin Number</i>	2
<i>Bondpad Coord (Yb)</i>	-0.6
<i>Bondpad Length (Lb)</i>	0.49
<i>Bondpad Width (Wb)</i>	0.2
<i>Pin Coord (Xp)</i>	-2.54
<i>Pin Length (Lp)</i>	1.0
<i>Pin Width (Wp)</i>	0.49

7. Click on *Apply*.

8. Click on *Next*.

The form changes to let you specify values for pin 3.

9. Specify the values in the following table:

Internal Lead Frame Setup Labels	Values
Die Cavity Side	Bottom
<i>Pin Number</i>	3
<i>Bondpad Coord (Yb)</i>	-0.6
<i>Bondpad Length (Lb)</i>	0.2
<i>Bondpad Width (Wb)</i>	0.49
<i>Pin Coord (Xp)</i>	-1.27
<i>Pin Length (Lp)</i>	0.2
<i>Pin Width (Wp)</i>	0.49

10. Click on *Apply*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

11. Click on *Next*.

The form changes to let you specify values for pin 4.

12. Specify the values in the following table:

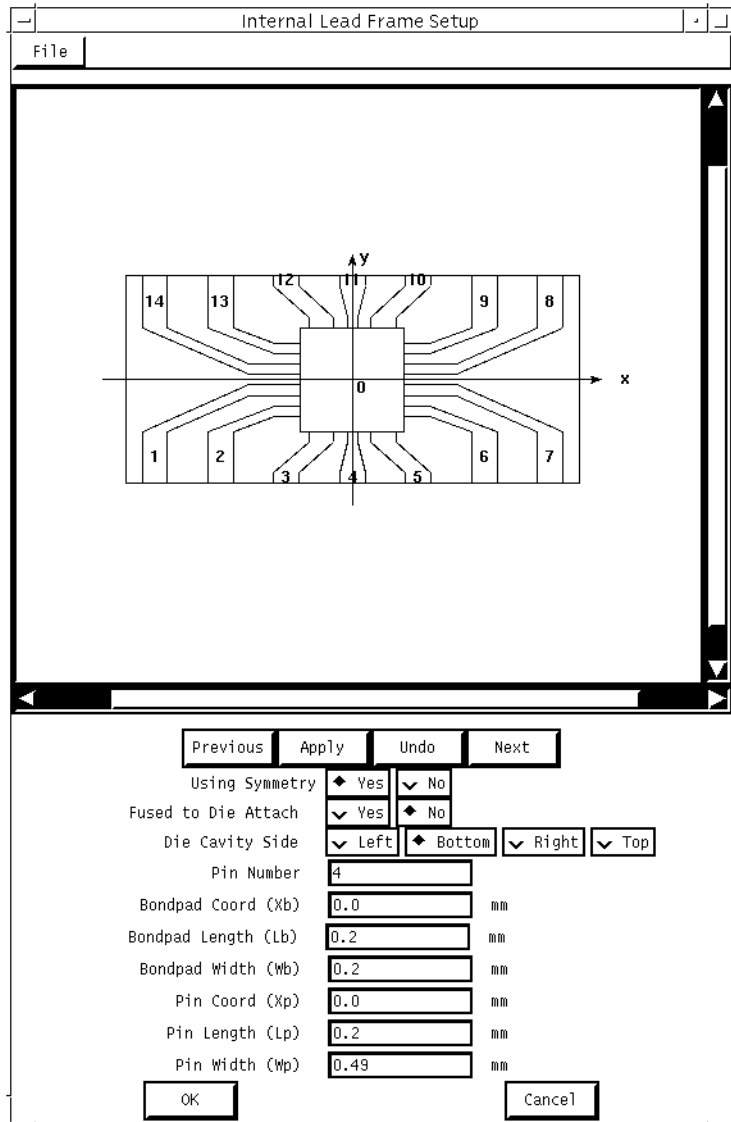
Internal Lead Frame Setup Labels	Values
Die Cavity Side	Bottom
<i>Pin Number</i>	4
<i>Bondpad Coord (Yb)</i>	0.0
<i>Bondpad Length (Lb)</i>	0.2
<i>Bondpad Width (Wb)</i>	0.2
<i>Pin Coord (Xp)</i>	0.0
<i>Pin Length (Lp)</i>	0.2
<i>Pin Width (Wp)</i>	0.49

13. Click on *Apply*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The completed form appears as follows:



14. In the *Internal Lead Frame Setup* form, click *OK*.

The form closes.

15. In the *RF IC Package Modeler* form, click on *Bond Wire Setup*.

The *Bond Wire Setup* form appears.

Filling In the Bond Wire Setup Form

In the *Bond Wire Setup* form, perform the following steps.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

1. Highlight *Pin to Die* for the *Connection Type*.
2. Click on *Add* and choose *Add – Single*.

You must click on Add to place a bondwire. You can add either a single bondwire or multiple bondwires. After you complete this step, you can specify values for the first bondwire.

3. Edit the values in the form to the values in the following table:

Bond Wire Setup Labels	Values You Type
<i>Bondwire Number</i>	1
<i>Pin Number</i>	1
<i>Pin Bond Site Coord(x, y)</i>	-1.0 and -0.2
<i>Die Bond Site Coord(x, y)</i>	-0.8 and -0.2
<i>Bondwire Length</i>	1
<i>Bondwire Diameter</i>	2.54e-2
<i>Conductivity</i>	5.8e7

4. Click on *Apply*.

You have completed specification for the first bondwire.

5. Repeat the sequence of steps “choose *Add – Single* followed by click on *Apply*” 13 times, once for each remaining bondwire.

This example does not require any editing of the bondwire specifications after the first bondwire, but you still must click on *Apply* for each bondwire to place each bondwire. The values you changed for the first bondwire remain as defaults for the following specifications. If you want to edit values for individual bondwires, you type the new values in the form before you click on *Apply*. After you perform the *Add – Single* operation, the number of the bondwire you are editing appears in the diagram in the *Bondwire Setup* form. After you click on *Apply*, the numbers of all the bondwires you have placed appear in the same diagram.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The completed *Bond Wire Setup* form appears like the one below:

The screenshot shows the **Bond Wire Setup** dialog box. At the top, there are menu tabs for **File** and **Options**. The main area displays a schematic of a die with 14 pins and 14 bondwires. The bondwire number **14** is selected. Below the schematic, there are several fields and buttons:

- Add**, **Delete**, **Previous**, **Next**, **Apply**, **Undo** buttons.
- Connection Type**: **Pin to Die** (selected), **Pin to DA**, **Die to DA**.
- Bondwire Number**: **14**
- Pin Number**: **14**
- Pin Bond Site Coord(x,y)**: **-1.0** (x), **0.2** (y) mm
- Die Bond Site Coord(x,y)**: **-0.8** (x), **0.2** (y) mm
- Bondwire Length**: **1** mm
- Bondwire Diameter**: **2.54e-2** mm
- Conductivity**: **5.8e7** S/M

At the bottom, there are **OK** and **Cancel** buttons.

6. In the *Bond Wire Setup* form, click **OK**.

The form closes.

7. In the *RF IC Package Modeler* form, click on *Parameter Extraction*.

The *Parameter Extraction* form appears. You can use this form to calculate capacitance, inductance, or both.

Extracting Parameters

In the *Parameter Extraction* form

1. Highlight *yes* for *Calculate Capacitance*.
2. Highlight *no* for *Ground Plane Effect*.
3. Type 4 . 3 for the *Dielectric Constant*.
4. Type 0 . 4 for the *Gnd/pkg Distance*.
5. Highlight *liberal* to specify the *Capacitance Accuracy*.
6. Highlight *yes* for *Calculate Inductance*.
7. Highlight *liberal* for the *Inductance Accuracy*.
8. Type 5 . 6e7 for the *Leadframe Conductivity*.
9. Type 1e9 for the *Working Frequency*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The completed form appears like the following:

The screenshot shows a dialog box titled "Parameter Extraction". It is divided into two main sections: "CAPACITANCE" and "INDUCTANCE".

CAPACITANCE

- Calculate Capacitance: yes no
- Ground Plane Effect?: yes no
- Dielectric Constant:
- Capacitance Accuracy: liberal moderate conservative

INDUCTANCE

- Calculate Inductance: yes no
- Inductance Accuracy: liberal moderate conservative
- Leadframe Conductivity: S/M
- Working Frequency: Hz

At the bottom of the dialog box are three buttons: "Solve", "Cancel", and "Close".

10. Click on *Solve*.

The capacitance and inductance are calculated. When the calculations are finished, the following dialog box appears:

The screenshot shows a dialog box titled "Capacitance Calculation". It contains a message box with the text "Parameter extraction is done" and an "OK" button.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

- To close the *Parameter Extraction* forms, click on *Close* and *OK*, respectively.
- In the *RF IC Package Modeler* form, click on *Macromodel Generation*.
The *Macromodel Generation* form appears.

Generating the Macromodel

In the *Macromodel Generation* form

1. Type 0 for the number of coupled pins per side.
2. Specify a *Macromodel Name*.
(For this example, *rficpkg* is used.)
3. Highlight *Spectre* for the *Macromodel Type*.
4. Highlight *no* for *Create DA Terminal?*

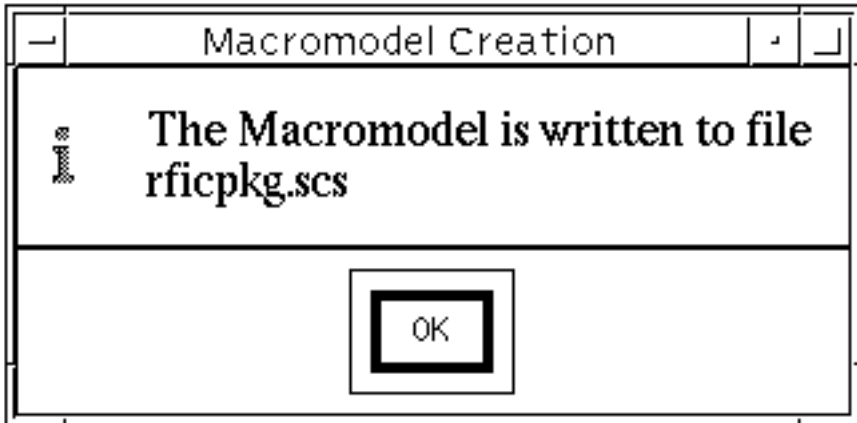
The completed form appears like the one below:

The screenshot shows a dialog box titled "Macromodel Generation". It contains the following fields and controls:

- Coupled Pins Per Side:** A text input field containing the value "0".
- Macromodel Name:** A text input field containing the value "rficpkg".
- Macromodel Type:** A radio button group with three options: "Spectre" (selected), "SpectreS", and "Spice".
- Create DA terminal?:** A radio button group with two options: "yes" and "no" (selected).
- Buttons:** Two buttons at the bottom, "Create" and "Close", both highlighted with a thick black border.

5. Click on *Create*.

The Macromodel is generated and the following dialog box is displayed.



The actual name of the file depends on the name you specify.

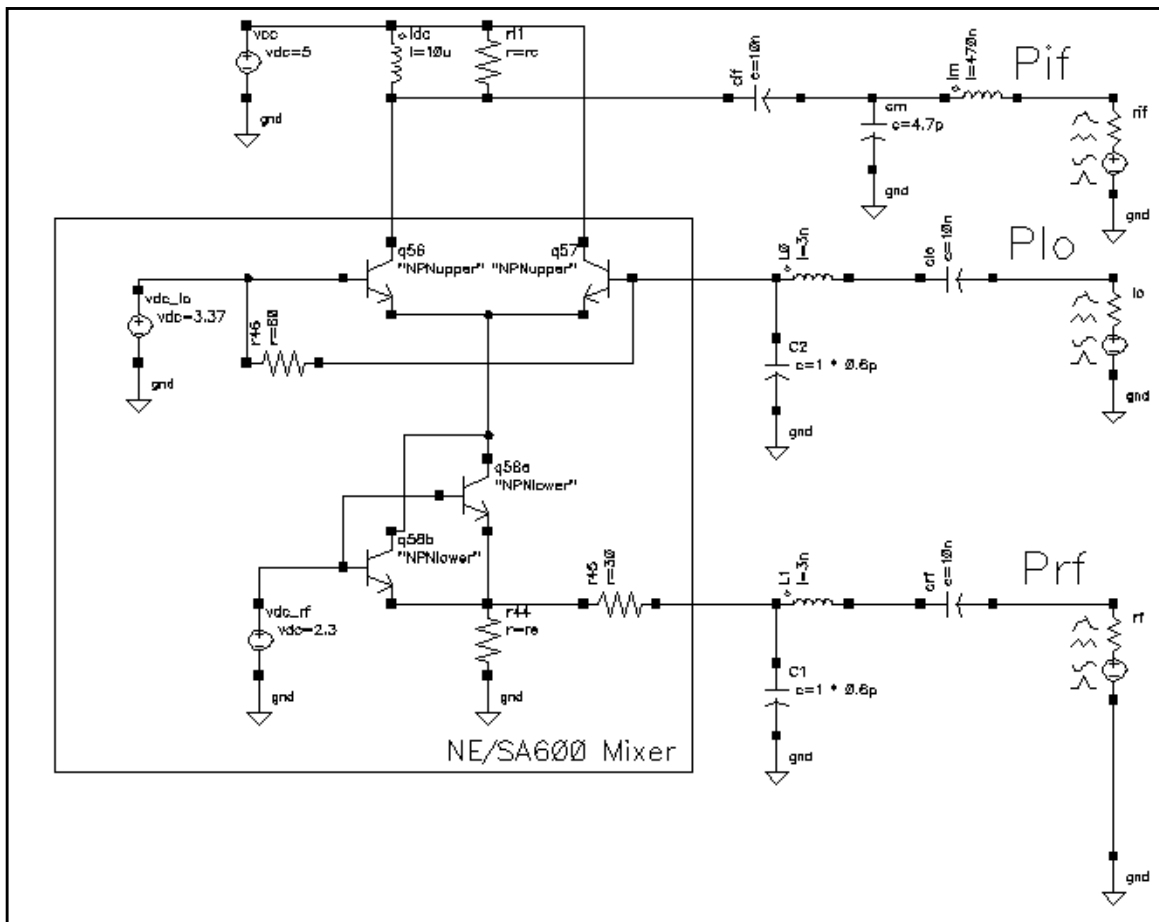
- To close the *Macromodel Generation* and *Macromodel Creation* forms, click on *Done* and *OK*, respectively.

Using the Macromodel in the Affirma Analog Circuit Design Environment to Simulate the ne600p Circuit

This section tells you how to place a PKG macromodel in a schematic and then simulate the circuit in the Affirma analog circuit design Environment. It assumes you have basic familiarity with simulation within this environment, including bringing up the CIW, opening a schematic, and bringing up the Simulation window. It also assumes that you have followed the instructions in Chapter 4 to edit the ne600p *frf* variable. If you are unable to perform any of these tasks, see the more detailed simulation setup instructions for the ne600p in Chapter 4. When you first bring up the ne600p schematic, it appears like the following picture.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages



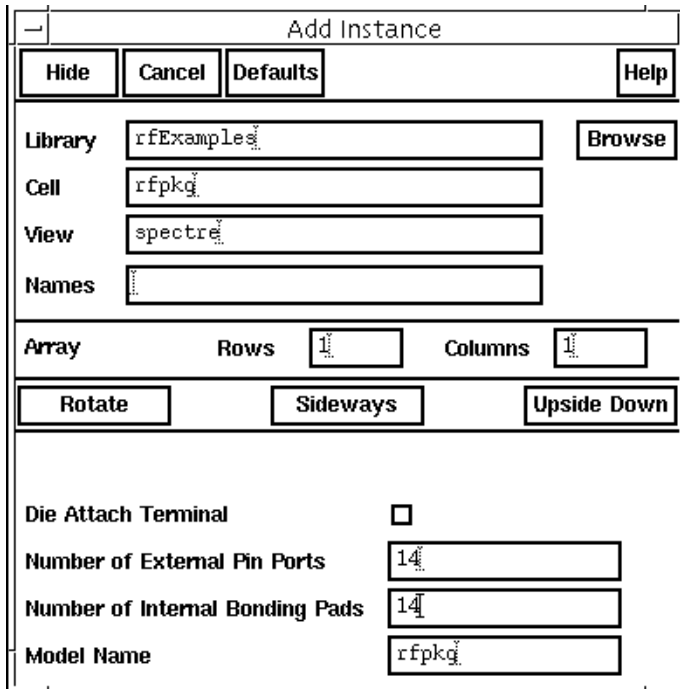
To simulate the ne600 circuit, perform the following steps:

1. In the Schematic window, choose *Add – Instance*.
The Add Instance form appears.
2. In the Add Instance form, perform the following:
 - a. Type `rfExamples` for the *Library*.
 - b. Type `rfpkg` for the *Cell*.
 - c. Type `spectre` for the *View*.
 - d. Type 14 for the *Number of External Pin Ports*.
 - e. Type 14 for the *Number of Internal Bonding Pads*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The completed form appears as follows:



Hide Cancel Defaults Help

Library rfExamples Browse

Cell rfpkg

View spectre

Names

Array Rows 1 Columns 1

Rotate Sideways Upside Down

Die Attach Terminal

Number of External Pin Ports 14

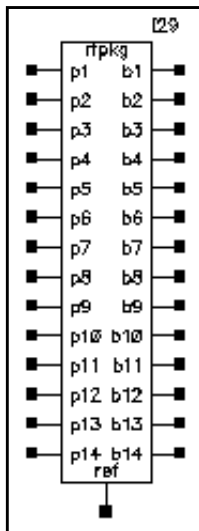
Number of Internal Bonding Pads 14

Model Name rfpkg

To place the package model in the schematic

1. Place the cursor in the Schematic window and click once.

The instance for the PKG macromodel appears in the Schematic window. It looks like the following figure. You create a new instance of the macromodel each time you click with the mouse.



Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

2. With your cursor in the Schematic window, press the `Escape` key.

The Add Instance form closes, and clicking with the mouse no longer creates new instances.

3. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears.

4. In the Schematic window, select the PKG macromodel that you placed in the window.

The Edit Object Properties form changes to accept specifications for the macromodel.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

5. In the Edit Object Properties form, type the name of the PKG model in the *Model Name* field and then click *OK*.

Apply To only current instance

Show system user CDF

Property	Value	Display
Library Name	rfExamples	off <input type="checkbox"/>
Cell Name	rfpkg	off <input type="checkbox"/>
View Name	spectre	off <input type="checkbox"/>
Instance Name	I28	off <input type="checkbox"/>

CDF Parameter	Value	Display
Die Attach Terminal	<input type="checkbox"/>	off <input type="checkbox"/>
Number of External Pin Ports	14	off <input type="checkbox"/>
Number of Internal Bonding Pads	14	off <input type="checkbox"/>
Model Name	rficpkg	off <input type="checkbox"/>

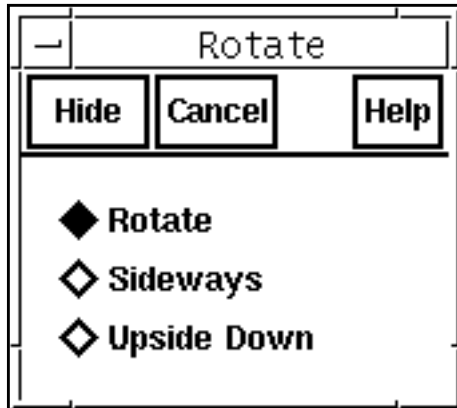
PKG model name

6. In the Schematic window, choose *Edit – Rotate* and simultaneously press the F3 key.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

The Rotate form appears. This form lets you rotate the macromodel instance 180 degrees on its vertical axis. After you connect wires to the macromodel, it is much more legible if you perform this rotation.

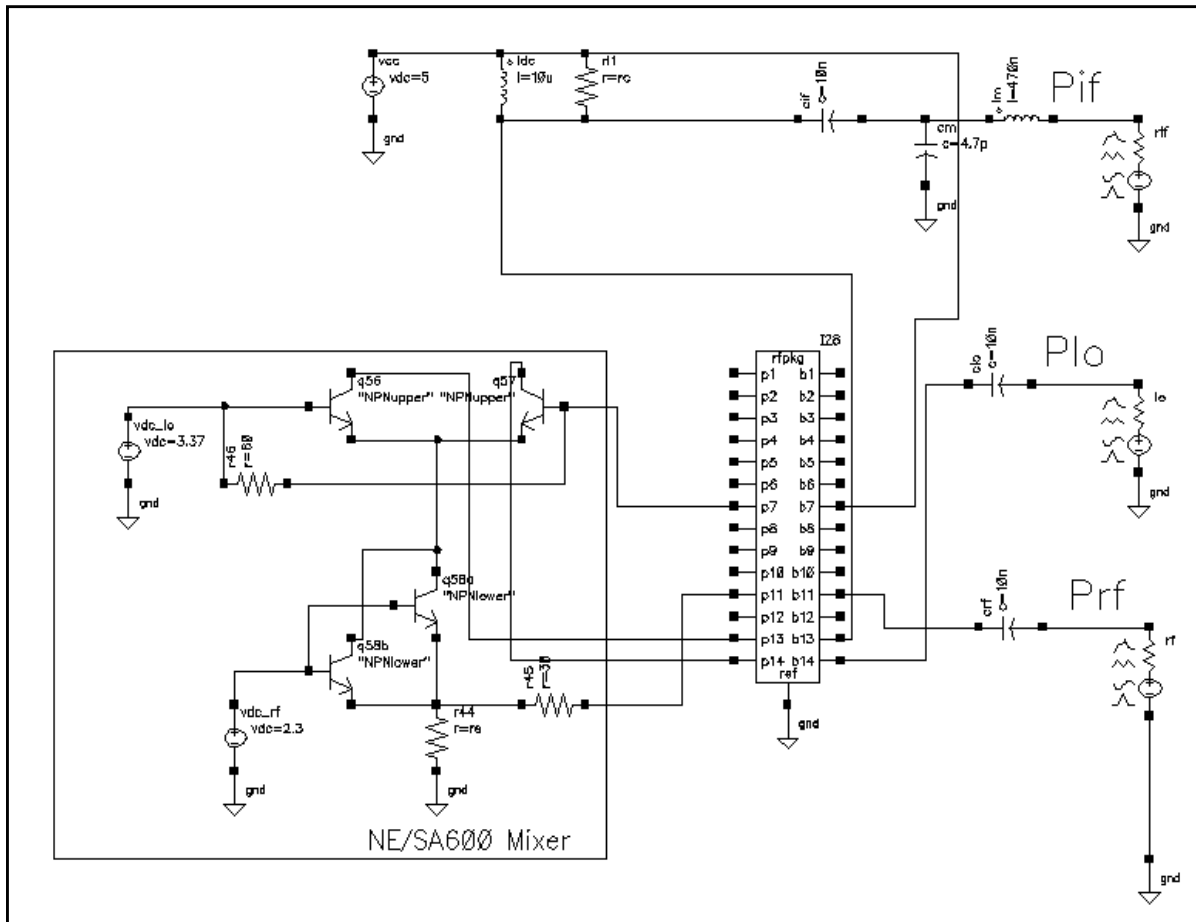


7. In the Rotate form, perform the following steps:
 - a. Highlight *Sideways*.
 - b. Click on *Hide*.
8. In the Schematic window, click on the PKG macromodel.
The macromodel rotates 180 degrees on its vertical axis.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

9. Connect the PKG macromodel to the schematic as shown in the following diagram.



Note: If you need assistance manipulating the wires and components on the schematic, consult the [Composer™: Design Entry User Guide](#).

10. Choose *Tools – Analog Environment* in the [Schematic window](#).

The [Simulation window](#) appears.

11. In the [Simulation window](#), choose *Setup – Model Libraries*.

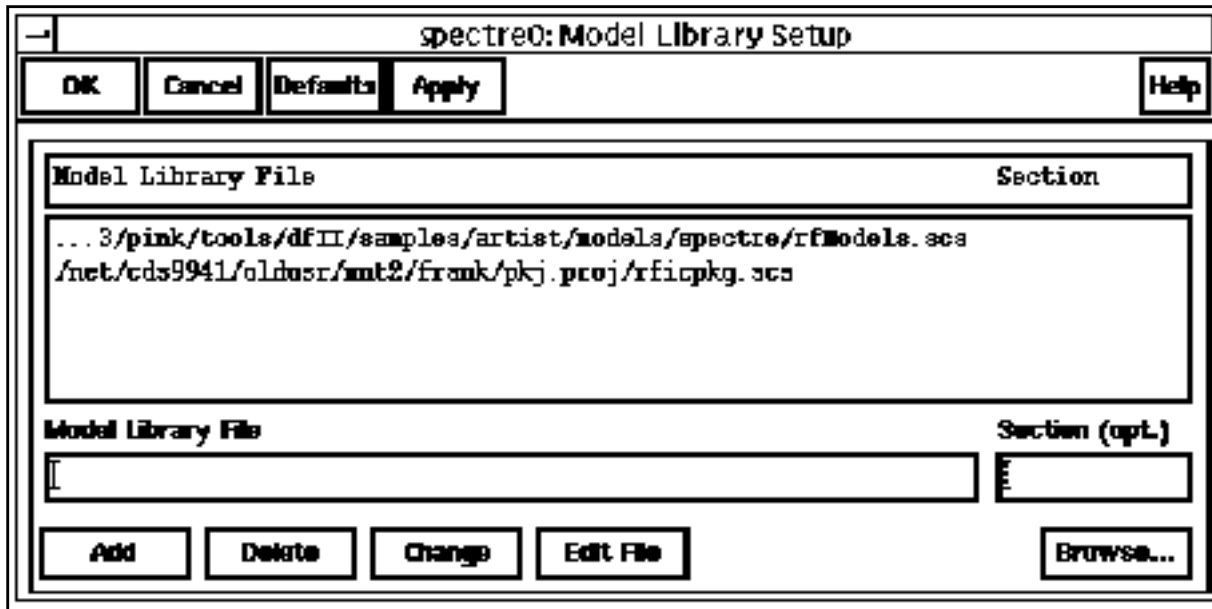
The [Model Library Setup](#) form appears.

12. In the [Model Library Setup](#) form, type the full path to the PKG output file in the [Model Library File](#) list box and then click *Add*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling RF IC Packages

Two model files now appear in the *Model Library File* list box. The `rfModels.scs` file was entered with your original setup to run circuits in the `rfExamples` directory. The second model file is the output file created by PKG.



You are now ready to choose an analysis in the Choosing Analyses form and run a simulation.

Extracting K-Models

The K-model links SpectreRF and Alta™ SPW as shown in the Figure below. The Ocean extraction script automatically runs a set of SpectreRF analyses and writes the results to a directory you specify. The results characterize the baseband performance of the RF circuit. There are three kinds of K-models.

- **Transmitter:** This is actually a J-model and is discussed in chapter 10.
- **Receiver (linear):** This is a linearized model of the receiver. It includes noise except for phase noise. Extraction is very similar to extraction of the non-linear K-model.
- **Receiver (non-linear):** This is a noiseless non-linear model of the receiver.

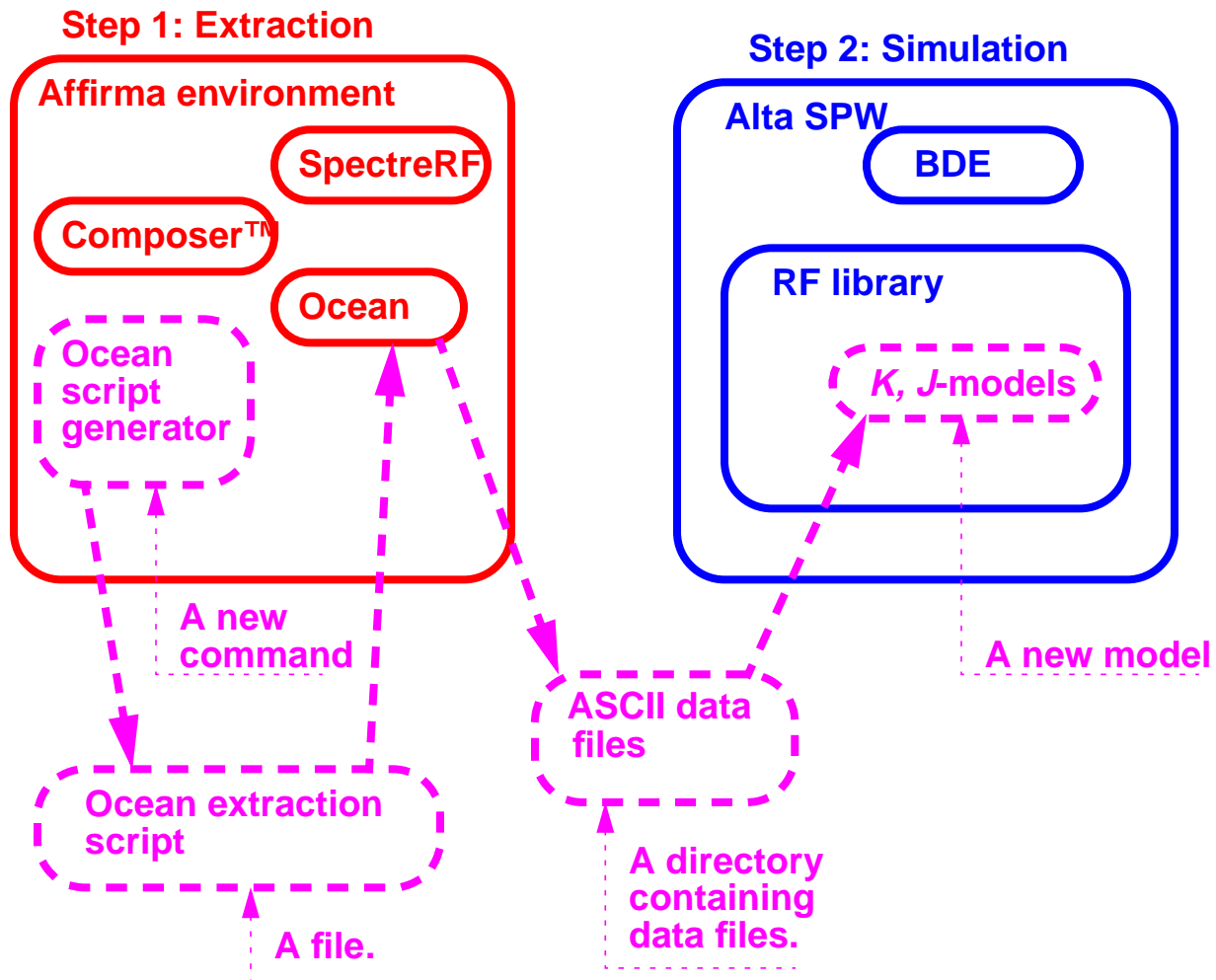
The receiver K-models only exist in the Cierto™ Alta SPW RF library but you extract the K-model data using SpectreRF in the Affirma analog circuit design Environment.

This chapter describes the following:

- The procedure for extracting a non-linear receiver K-model.
- The K-model's limitations
- The procedure for handling strong out-of-band blockers

The K-model data files and their format are described at the end of the chapter.

Figure 9-1 Task Flow of K-Model Extraction and Simulation



Procedures for Simulating `k_mod_extraction_example`

Follow the steps below to simulate the `k_mod_extraction_example`:

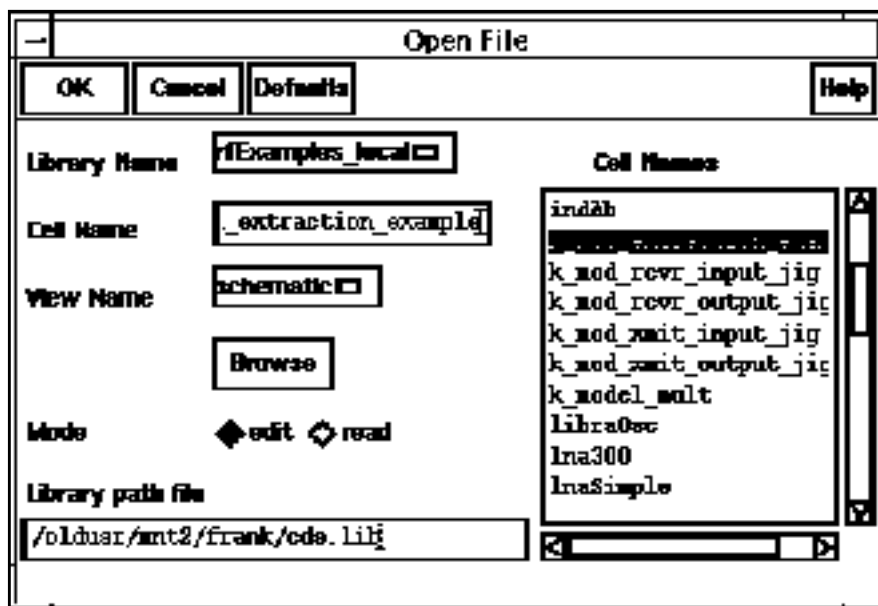
- Copy `k_mod_extraction_example` from the Cadence `rfExamples` directory into a local directory.
Be sure the path to the local directory is defined in your `cds.lib` file.
- Choose *File – Open* in the Command Interpreter Window (CIW).
The Open File form appears. Filling in the Open File form lets you open the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

In the Open File form, do the following:

1. Choose *<Path_to_local_copy>* for *Library Name*.
2. Choose *schematic* for *View Name*.
3. In the *Cell Names* list box, highlight *k_mod_extraction_example*.
k_mod_extraction_example appears in the *Cell Name* field.
4. Highlight *edit* to select the *Mode*.

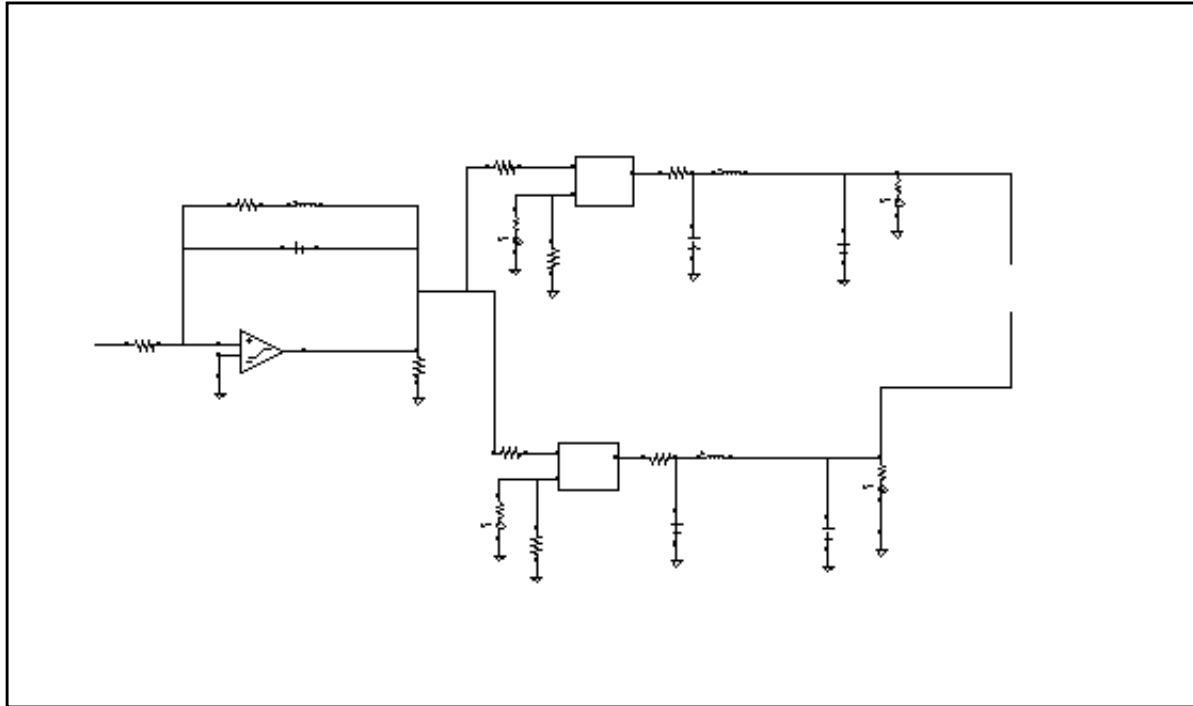


5. Click *OK* in the *Open File* form.

The Schematic window displays the schematic.

If you want to see the whole procedure for setting up the K-model, including the editing of the schematic, delete the input and output test jigs at the far left and right of the schematic. The edited schematic now looks like the one below.

The K_mod_extraction_example Schematic



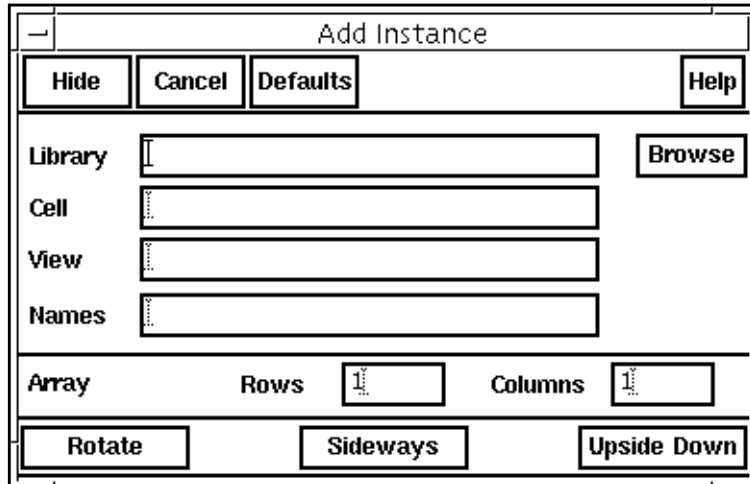
From the Schematic window, do the following:

1. Select *Add – Instance....*

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The Add Instance form appears.

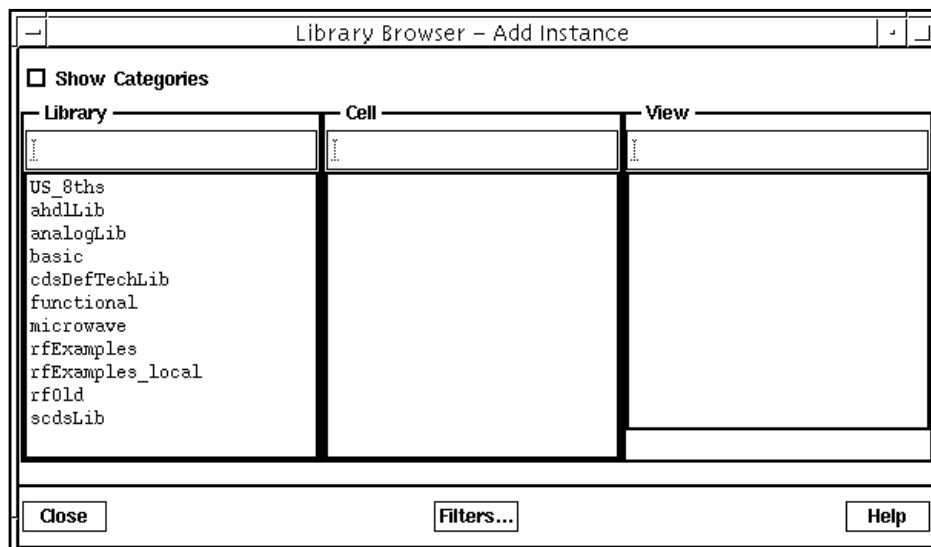


The 'Add Instance' dialog box contains the following elements:

- Buttons: Hide, Cancel, Defaults, Help
- Fields: Library (with a 'Browse' button), Cell, View, Names
- Array section: Rows (set to 1), Columns (set to 1)
- Orientation buttons: Rotate, Sideways, Upside Down

2. In the Add Instance form, click on *Browse*.

The Library Browser – Add Instance form appears.



The 'Library Browser – Add Instance' dialog box features a table with three columns: Library, Cell, and View. The 'Library' column contains a list of library names:

Library	Cell	View
US 8ths		
ahdLib		
analogLib		
basic		
cdsDefTechLib		
functional		
microwave		
rfExamples		
rfExamples_local		
rfOld		
scdsLib		

At the bottom of the dialog are buttons for Close, Filters..., and Help.

In the Library Browser – Add Instance form, do the following:

1. Select *rfExamples* for the *Library*.

The form changes to let you select a cell.

2. Select *k_mod_rcvr_input_jig* for the *Cell*.

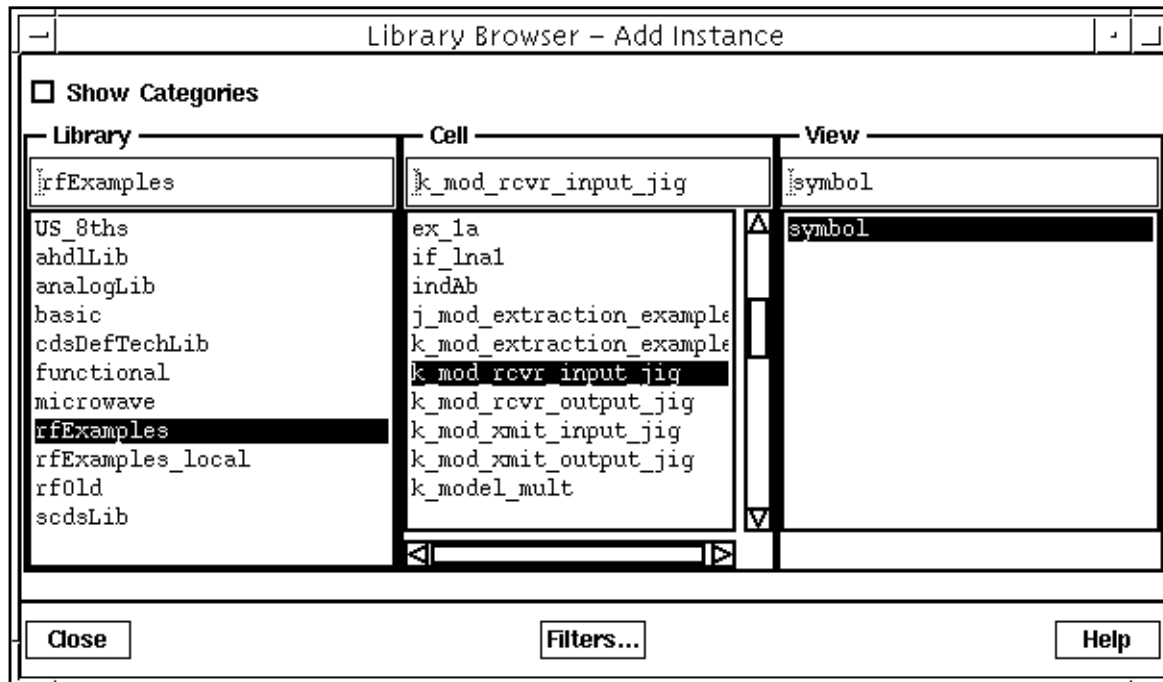
The form changes to let you select a *View*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

3. Select *symbol* for the *View*.

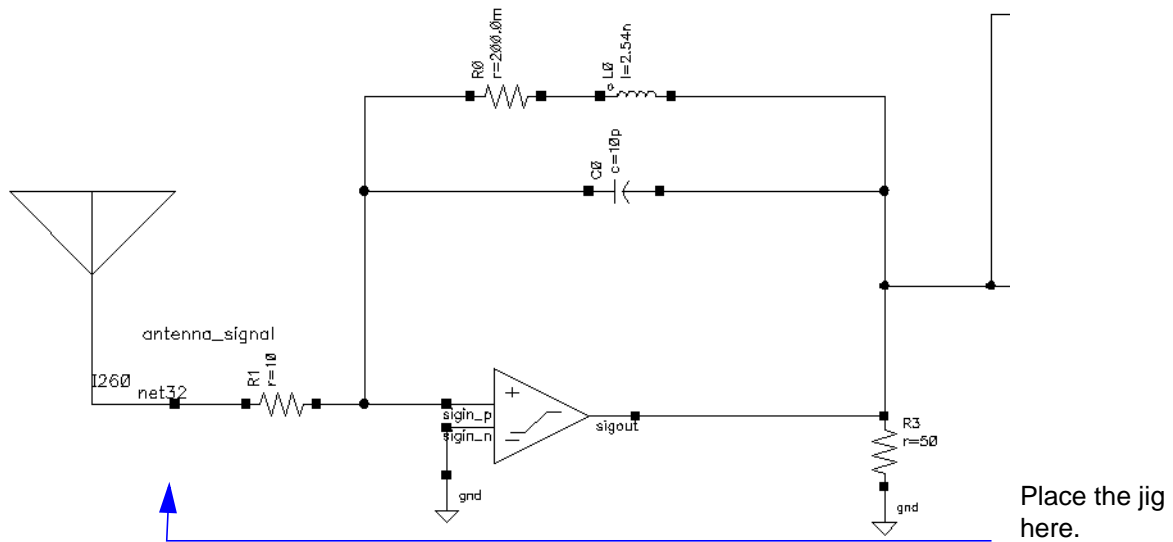
The completed form appears like the one below



Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

When you place the cursor back in the Schematic window, an instance of *k_mod_rcvr_input_jig* is attached to it. Place the input jig at the appropriate place in the schematic.



If you need assistance with the procedures for editing a schematic, see the [Virtuoso® Schematic Composer™ User Guide](#).

In the Schematic window, again select *Add – Instance....*

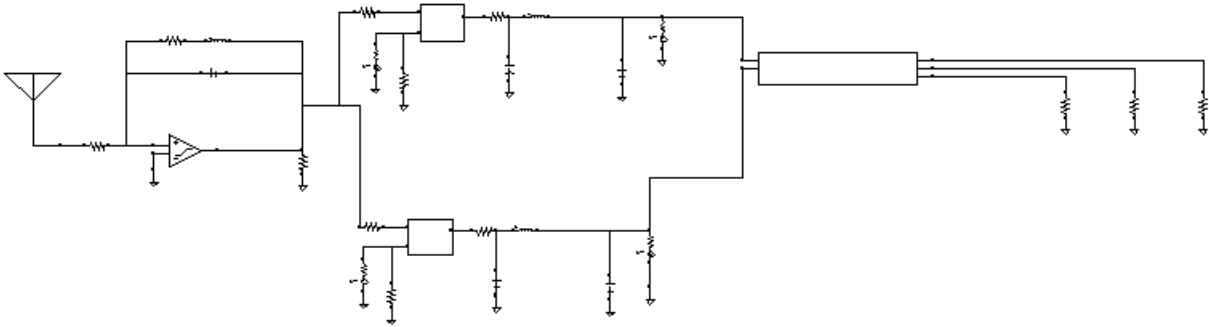
Now repeat the steps you used to add the input jig to the schematic and add the output jig *k_mod_rcvr_output_jig*. Also add a resistor connected to ground to each of the three output connections of the output jig. The table below tells you the libraries, cells, and views of the components you add.

Component	Library	Cell	View
output jig	rfExamples	k_mod_rcvr_output_jig	symbol
resistor	analogLib	res	symbol
ground	analogLib	gnd	symbol

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The edited schematic looks like this.



In the Schematic window, perform the following steps:

1. Click on the input jig to select it.
2. Select *Edit – Properties – Objects* in the Schematic window menu choices.

The Edit Object Properties form appears.

- a. In the Edit Object Properties form, type 1G for the *carrier frequency* value.
- b. In the Edit Object Properties form, click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The completed form appears like the one below:

Property	Value	Display
Library Name	rfExamples	off
Cell Name	k_mod_rcvr_input_jig	off
View Name	symbol	off
Instance Name	I27	off

User Property	Master Value	Local Value	Display
interfaceLastC..	7 14:57:15 1998		off

CDF Parameter	Value	Display
carrier frequency	1G Hz	off
Kmodel: signal bias	sig_bias V	off
Kmodel: carrier phase	car_ph deg	off
Kmodel: signal bias end point	bias_end V	off

3. In the Schematic window, select *Design – Check and Save*.

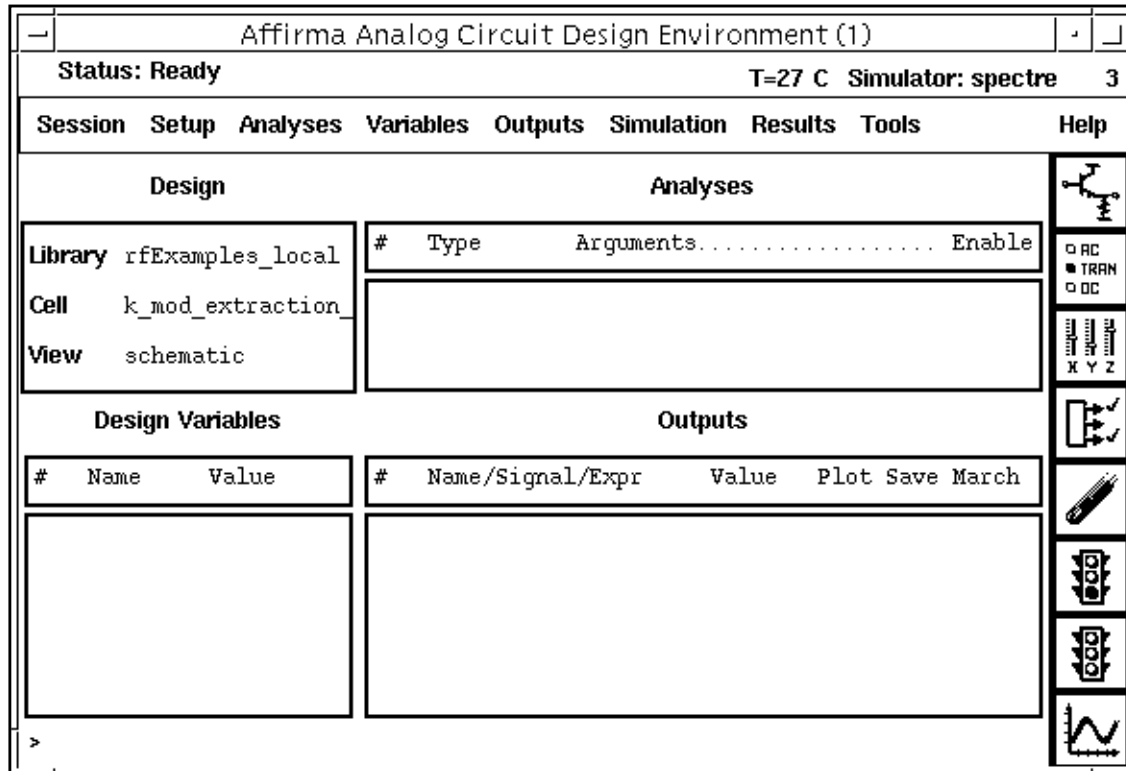
Analysis Setup

1. Choose *Tools – Analog Environment* in the Schematic window.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The Simulation window appears.



2. Choose *Setup – Simulator/Directory/Host* in the Simulation window.

The Choosing Simulator/Directory/Host form appears.

In the Choosing Simulator/Directory/Host form, do the following:

1. Highlight *spectre* for *Simulator*.
2. Type in the name of the project directory, if necessary.
3. Highlight the *local* or the *remote* button to specify the *Host Mode*.
For remote simulation, type in the name of the host machine and the remote directory in the appropriate type-in fields.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The completed form appears like the one below.

Choosing Simulator/Directory/Host -- Affirma Analog Circuit Desi

OK Cancel Defaults Help

Simulator

Project Directory

Host Mode local remote distributed

Host

Remote Directory

4. In the Choosing Simulator/Directory/Host form, click *OK*.

In the Simulation window, do the following:

1. Choose *Variables – Copy From Cellview*.

The variables *sig_bias*, *car_ph*, and *bias_end* appear in the Design Variables list box of the Simulation window.

2. Choose *Variables – Edit....*

The Editing Design Variables form appears.

Editing Design Variables -- Affirma Analog Circuit Design Env

OK Cancel Apply Apply & Run Simulation Help

Selected Variable

Name

Value (Expr)

Add Delete Change Next Clear Find

Table of Design Variables

#	Name	Value
1	sig_bias	
2	car_ph	
3	bias_end	

Cellview Variables

Copy From Copy To

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

In the Editing Design Variables form, do the following:

1. Click on `sig_bias` in the Table of Design Variables list box.
2. Type `.01` in the *Value(Expr)* type-in field.
3. Click *Apply*.
4. Repeat steps one through three for the `car_ph` and `bias_end` variables. Type `0` for *Value(Expr)* for `car_ph` and `1.8` for the `bias_end`.

The completed form looks like the one below:

Selected Variable		Table of Design Variables	
Name	bias_end	#	Name Value
Value (Expr)	1.8	1	sig_bias 10m
		2	car_ph 0
		3	bias_end 1.8

5. Click *OK*.
6. Choose *Analyses – Choose* in the Simulation window.
The Choosing Analyses form appears.
7. In the Choosing Analyses form, click on *pSS* for the analysis.
The form changes to display options needed for PSS.

You perform PSS first because the periodic steady state solution must be determined before you can perform a small-signal analysis.

In the Choosing Analyses form, do the following:

1. Type `1G` for the *Beat Frequency*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

2. Choose *Number of Harmonics* in the *Output Harmonics* cyclical field, and type 0 for the number of harmonics.
3. Highlight *moderate* for the *Accuracy Defaults (errpreset)* value.
4. Click on *Options....*

The Periodic Steady State Options form appears.

- a. In the Periodic Steady State Options form, type 1.6G for the *maxacfrequency* value.

ACCURACY PARAMETERS	
relref	<input type="checkbox"/> pointlocal <input type="checkbox"/> allocal <input type="checkbox"/> sigglobal <input type="checkbox"/> allglobal
iteratio	<input type="text"/>
steadyratio	<input type="text"/>
maxacfreq	<input type="text" value="1.6G"/>
maxperiods	<input type="text"/>
finitediff	<input type="checkbox"/> yes <input type="checkbox"/> samegrid <input type="checkbox"/> no

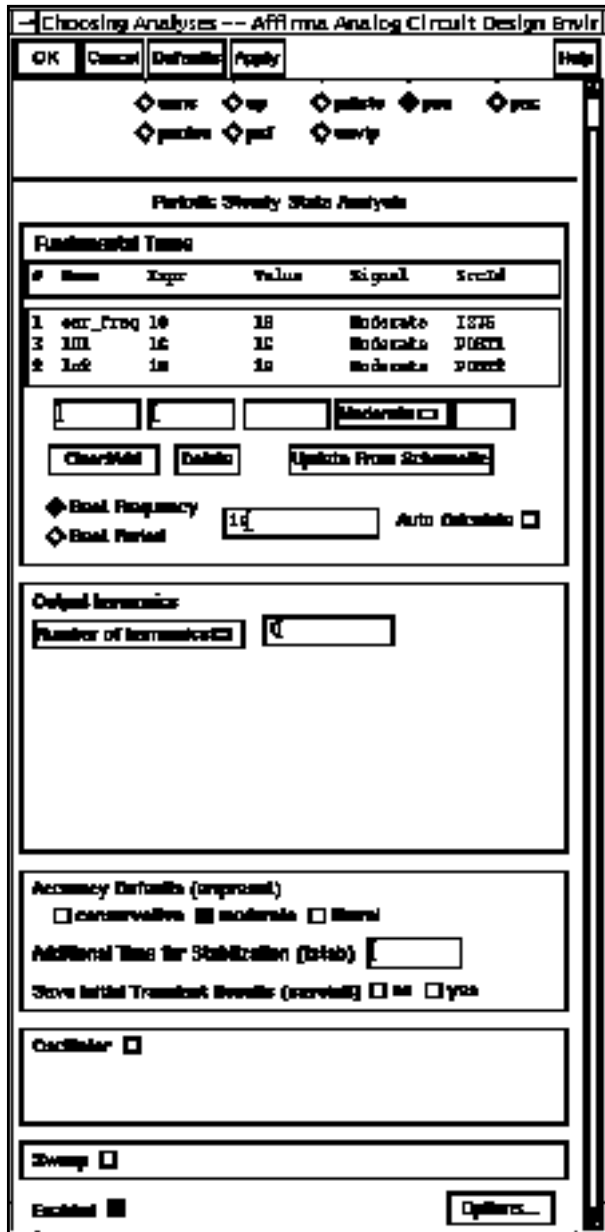
- b. In the Periodic Steady State Options form, click *OK*.

5. Be sure the *Enabled* button is highlighted.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The completed *pss* section looks like this.



6. Choose *pac* for the *Analysis*.

The form changes to let you specify data for the *pac* analysis that follows the *pss* analysis.

7. Choose *Start – Stop* in the *Frequency Sweep Range (Hz)* cyclical field and type .01 and 500M for the *Start* and *Stop* values, respectively.

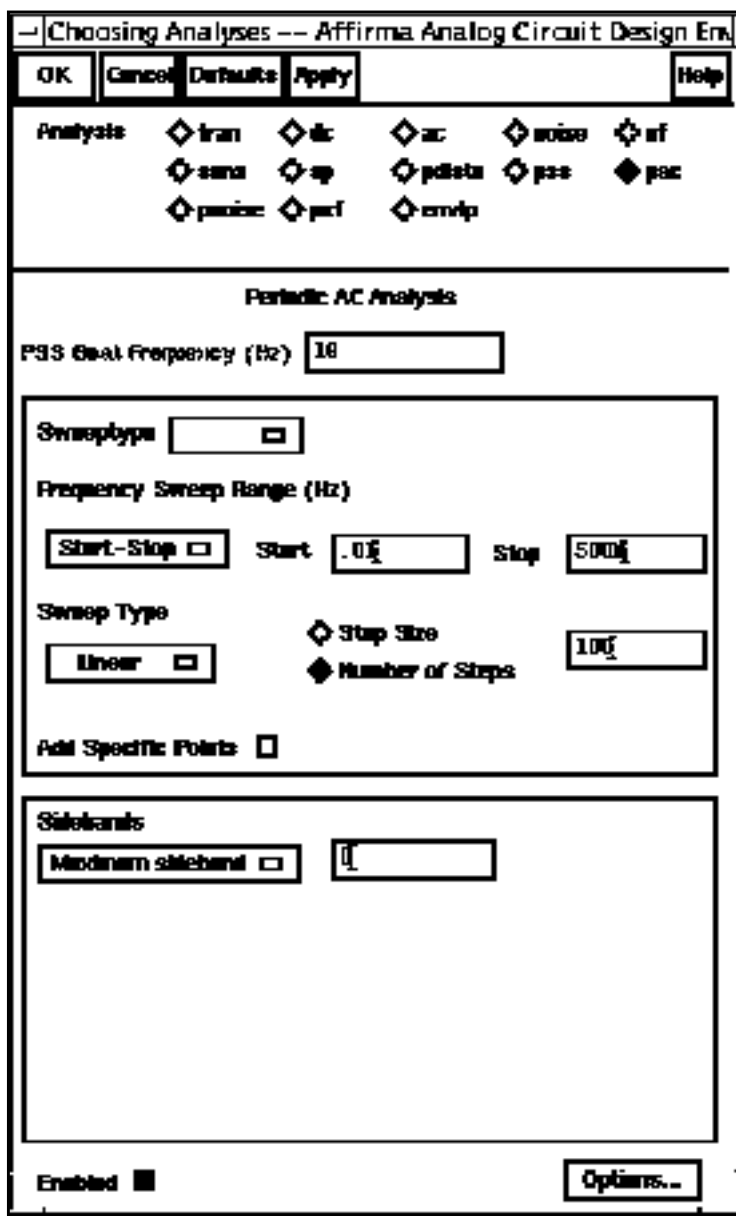
Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The starting frequency must be very low but not zero. If the DC values of the PAC results are not captured, the K-model can incur large errors.

8. In the *Sweep Type* cyclical field, choose *Linear*.
9. Highlight *Number of Steps*, and type 100 in the field to the right of the button.
10. In the *Sidebands* cyclical field, choose *Maximum Sidebands*, and type 0 for the *Maximum Sidebands* value.

The completed *pac* section looks like this.



11. Click *OK* in the Choosing Analyses form.

Running the Simulation

1. In the Simulation window, select *Simulation – Netlist and Run* to run the simulation.

The output log file appears and displays information about the simulation as it runs.
Check the CIW for a message saying the simulation completed successfully.

Checking the Simulation Results

➤ Choose *Results – Direct Plot – PSS* in the Simulation Window.

The Waveform window and the PSS Results form appear.

In the PSS Results form, do the following:

1. Choose *Replace* for *Plot Mode*.
2. Choose *pac* for *Analysis Type*.
3. Choose *Voltage* for *Function*.
4. Choose *Magnitude* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

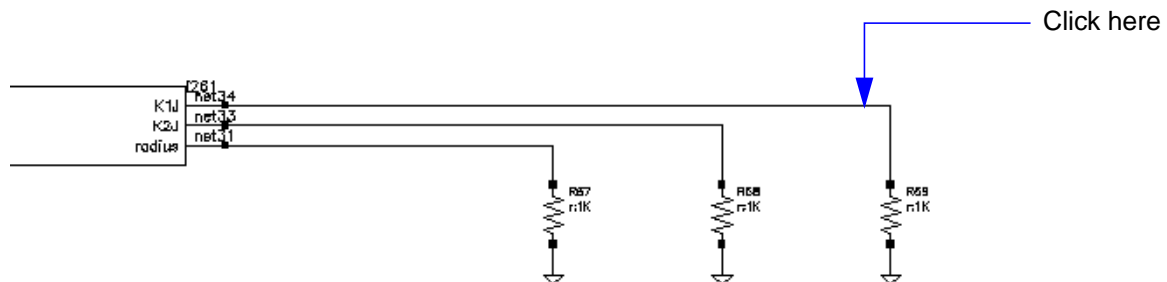
Extracting K-Models

The completed form looks like this.

The screenshot shows the 'PSS Results' dialog box with the following settings:

- Plot Mode:** Append (checked), Replace (unchecked)
- Analysis:** pss (checked), pac (unchecked)
- Function:** Voltage (checked), Current (unchecked), IPN Curves (checked)
- Select:** Net (dropdown menu)
- Sweep:** spectrum (checked), sideband (unchecked)
- Signal Level:** peak (checked), rms (unchecked)
- Modifier:** Magnitude (checked), Phase (unchecked), dB20 (unchecked), Real (checked), Imaginary (unchecked)
- Add To Outputs:** (checkbox)
- Footer:** > Select Net on schematic...

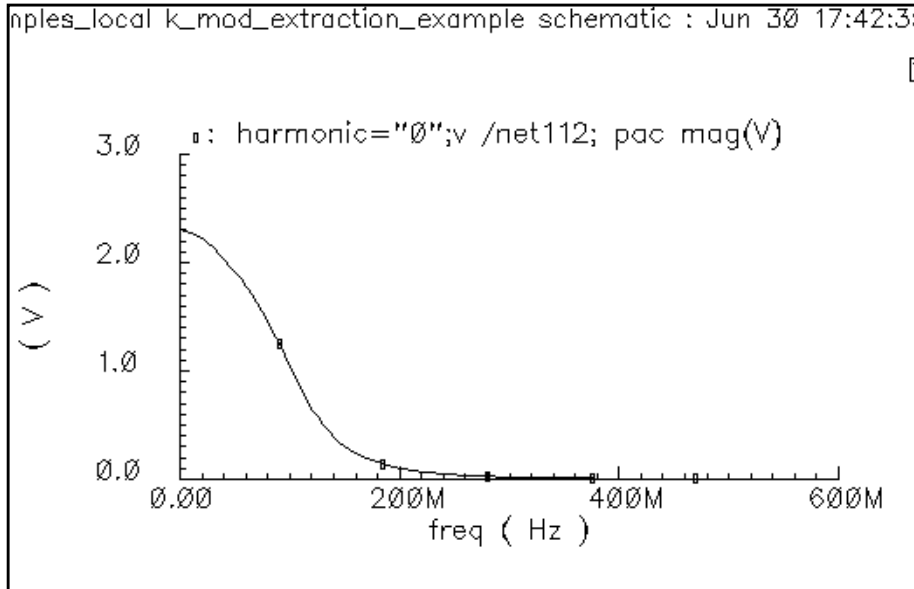
In the Schematic window, click on the appropriate wire.



Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

The plot in the Waveform window looks like the one below.



Creating the K-Model

- In the Simulation window, select *Tools – RF – Link to SPW....*

The Affirma RFIC Modeler for Cierto SPW form appears.

In the Affirma RFIC Modeler for Cierto SPW form, do the following:

1. Choose *Receiver(non-linear)* for the *Circuit Type*.
2. Edit the *Script File Name* and the *Extracted Kmodel Directory Name*, if necessary.

The file name is the name of the Ocean script to be run later, and the directory name is the output directory.

3. Click *OK*.

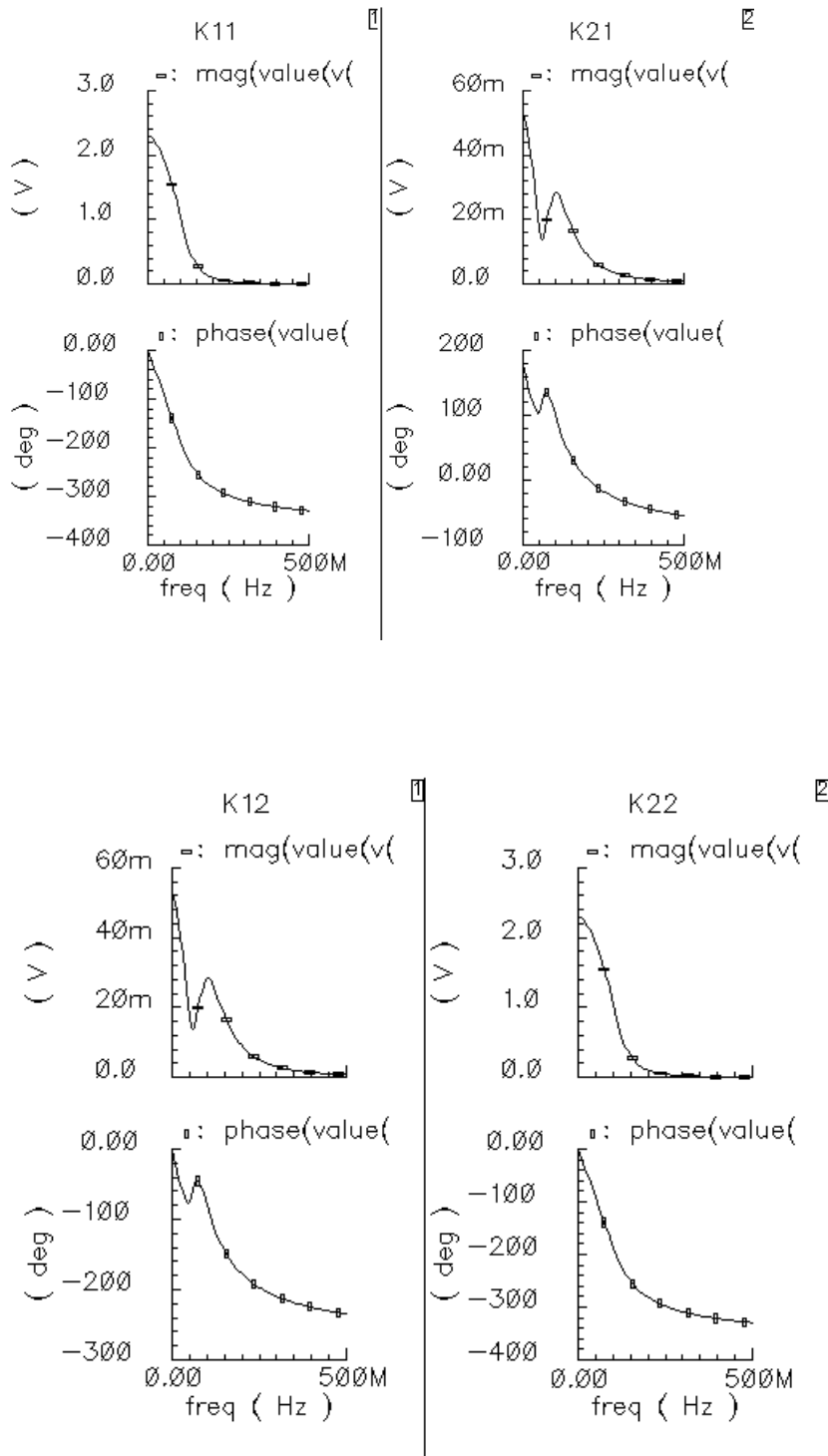
- In the CIW, type the following line and press Return.

```
load "<script_file_name>"
```

Affirma RF Simulator (SpectreRF) User Guide

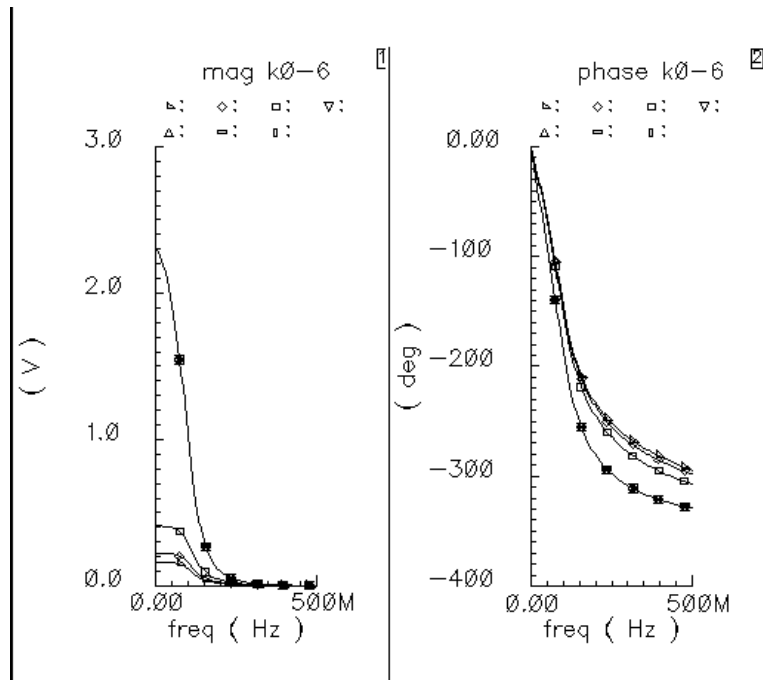
Extracting K-Models

The script runs and generates the k-model. When it finishes, it generates the following plots:



Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models



The K11, K12, K21, and K22 plots are the I-to-I, Q-to-I, I-to-Q, and Q-to-Q transfer functions in the linear K-model. The linear K-model is extracted at the *sig_bias* signal level. The overlaid curves are the K11 transfer functions. These transfer functions are extracted with I-input signal biases that range linearly from *sig_bias* to *bias_end*. You can reconstruct these plots by loading the Ocean script in the file located at `<k_model>/plotDir/oceanKmodQuickPlot`. Examine the curves for the following features:

- **Narrow-band dynamics:** The K-model does not handle narrow-band dynamics well because it uses FIR (Finite Impulse Response) filters. The most likely narrow-band dynamics result from AGC loops and large DC blocking capacitors. They can introduce ultra-low frequency dynamics that require many FIR filter taps. Ultra-low frequency dynamics appear in the above curves as sharp dips near DC.
- **Non-negligible response at the maximum frequency:** Be sure the data goes to a high enough frequency. If the Alta SPW sample rate, divided by two, exceeds the maximum frequency in the data, Alta SPW will linearly interpolate from the highest frequency data point down to zero at half the sample frequency. Because the receiver response probably decays exponentially rather than linearly, this interpolation can introduce errors.

Alta SPW ignores data beyond half the sample frequency. This does not cause a problem because an Alta SPW input signal probably would have no power beyond half the sampling frequency.

More About the K-Model

Like any behavioral model, the K-model has limitations and requirements. The assumptions on which the K-model is based cause the basic limitations. Other limitations are caused by practical considerations. There are also requirements that are specific to the Alta SPW environment. Finally, this section discusses some limitations of the K-model's ability to model out-of-band blockers.

Limitations and requirements resulting from linear K-model assumptions

The linear K-model makes the following two assumptions beyond the small signal assumption:

- Unless the SpectreRF model includes detailed A/D converter models, the extracted linear K-model assumes that noise at the base-band outputs does not change significantly over the A/D aperture time.

The first K-model assumption involves noise and A/D conversion. The linear model includes noise generated within the front end¹. "Aperture time" is the time the A./D converter requires just to measure a voltage, not to measure and hold it. The assumption is that the noise is constant over the aperture time. This implies the noise is band limited. However, as the following discussion explains, the noise bandwidth must also not be too small.

- The two equivalent output noise sources, one for the I output and one for the Q output, are uncorrelated.

Frequently, much of the noisy circuitry is common to both outputs. Despite the common circuitry, the K-model assumes the two output noise signals are independent. This is a good assumption if the noise is white over the frequencies of interest. In summary, the noise must be white with respect to filters preceding the A/D conversion but band-limited with respect to $1/(\text{aperture time})$.

Why isn't there just one K-model, a noisy non-linear K-model instead of a noisy linear K-model and a noise-less non-linear K-model? Putting small signal noise in the non-linear K-model would suggest that the model's noise varied with input power. It does not. Furthermore, noise analysis is usually of interest in bit error rate simulations. Such simulations are usually long and the desired signal is weak. The linear K-model suffices for weak desired signals and runs faster than the non-linear K-model.

1. The linear model does not account for phase noise. However, if one knows the power spectral density of the phase noise one can easily follow the K-model with a complex multiplier that shifts the phase of the K-model output. A filter driven by white Gaussian noise would generate the random phase shift.

Limitations and requirements resulting from non-linear K-model assumptions

The K-model requires two basic assumptions:

- Small-signal components of the base-band input can be wide-band, but large components must fall within the receiver's bandwidth.

The signal of interest is usually within the receiver's bandwidth, and interferers usually lie outside this bandwidth. However, if the receiver has a band selection filter immediately after the antenna, and that filter can be modeled apart from the K-model, this first assumption is not a severe restriction. This is true because the filter makes the interferer small by the time it reaches the K-model.

- Non-linear behavior depends only on the input radius¹. Non-linear PM/PM and PM/AM effects are negligible.

Requirements resulting from the Alta SPW environment

Both K-models are part of Alta SPW, the Cierto Signal Processing Worksystem. Alta SPW is a DSP design tool. This environment creates the following two consequences you must remember:

- Model extraction requires base-band input and output ports.

Because it is a base-band tool, Alta SPW operates on the information impressed on the RF carrier rather than the instantaneous value of the RF signal. Each of two carrier phases, sine and cosine, carry information. Because Alta SPW represents base-band signals as complex numbers, the modeled RF circuit must have base-band input and output ports.

The input test jig creates the input port. The input test jig mixes the base-band input signal up to the appropriate carrier frequency, thereby synthesizing an antenna signal from a base-band input. Changing the phase of the carrier by 90 degrees switches the base-band input between the "in-phase" and "quadrature" input ports. The Ocean extraction script changes carrier phase automatically when it extracts a K-model.

If the circuitry does not contain two base-band output ports, you must synthesize them. You can synthesize them easily using ideal behavioral elements. For example, to build a K-model of only the low noise amplifier, you add ideal downconverters to the SpectreRF model and extract a K-model of the combination.

1. If the input signal is $rf(t) = a(t) \cos(\omega_c t) - b(t) \sin(\omega_c t)$, the base-band signal is $a(t) + j b(t)$, a trajectory in the complex plane. This is the rectangular representation. The polar representation is in terms of radius and angle. The radius is $\text{Sqrt}[a(t)^2 + b(t)^2]$. The angle is $\text{ArcTan}[b(t)/a(t)]$.

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

You can partition a receiver into smaller K-models only if there is no non-linear interstage loading at the break points and if the carrier fundamental is the only important quantity.

The input and output test jigs also tell the extraction script where to excite the SpectreRF model and what quantities to measure.

- The SpectreRF noise analysis must extend far enough to capture all important features, including those beyond half the Alta SPW sampling frequency.

This is true because the power spectral density suffers aliasing in the DSP environment. The only data available for DSP is for discrete, evenly-spaced time-points. This restriction imposes a constraint on possible input signals that is usually met within the design. Specified input signals must not have significant power beyond half the sampling frequency.¹ Because the noise is not a signal you specify, it need not meet that constraint. Noise power that is well beyond half the sampling frequency can alias down to the band of interest. If the noise is characterized to a high-enough frequency, the K-model automatically aliases it.

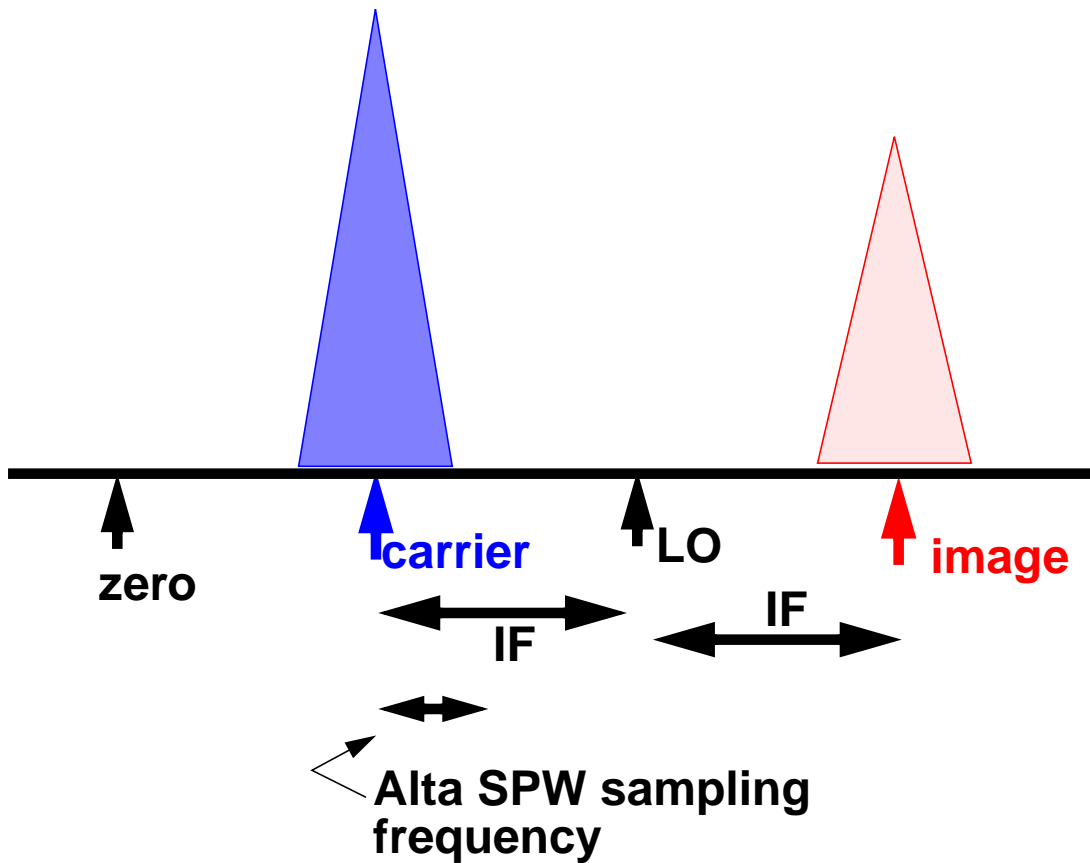
Practical limitations of the K-model

■ Image rejection

Although the K-model theoretically models image rejection, such modeling is not practical for the Alta SPW implementation [Figure 9-2](#) on page 433 shows how the carrier, local oscillator, and image frequencies can be related. The Alta SPW sampling frequency is probably approximately the bandwidth of the RF receiver, shown in blue. To simulate dynamics at the image frequency, the Alta SPW sampling frequency must span twice the difference between the carrier frequency and the image frequency. Such a high sampling frequency requires an impractical number of FIR filter taps to model the receiver dynamics. Simulation time is almost as lengthy as with an unsuppressed carrier simulation.

1. The channel and RF front end are cascaded physical entities with no sampler between them. But because they are modeled in Alta SPW as separate blocks, Alta SPW implicitly assumes an ideal sampler lies between the channel and front end. That assumption is only valid for base-band signals that have no significant spectral components near or beyond half the sampling frequency. If that assumption were not valid the overall design would have far more serious problems than K-model accuracy.

Figure 9-2 Carrier, Local Oscillator, and Image Frequency Relationships for the Alta SPW Implementation



■ Ultra-low frequency dynamics

Ultra-low frequency effects such as those caused by PLLs, AGC loops, and large DC blocking capacitors again require many FIR filter taps. However, behavioral phaselock loop models or AGC loop models can precede the K-model in Alta SPW to approximate these low frequency effects.

Limitations in modeling large out-of-band blockers

The Figure 9-2 illustrates a basic K-model limitation. The basic K-model does not accurately simulate the effects of large out-of-band blockers. However, if the blocker is narrow-band enough to be considered deterministic (ideally, a single tone), you can extract a model that

accurately simulates blocker-induced gain compression by extracting a K-model with the blocker present.

Figure 9-3

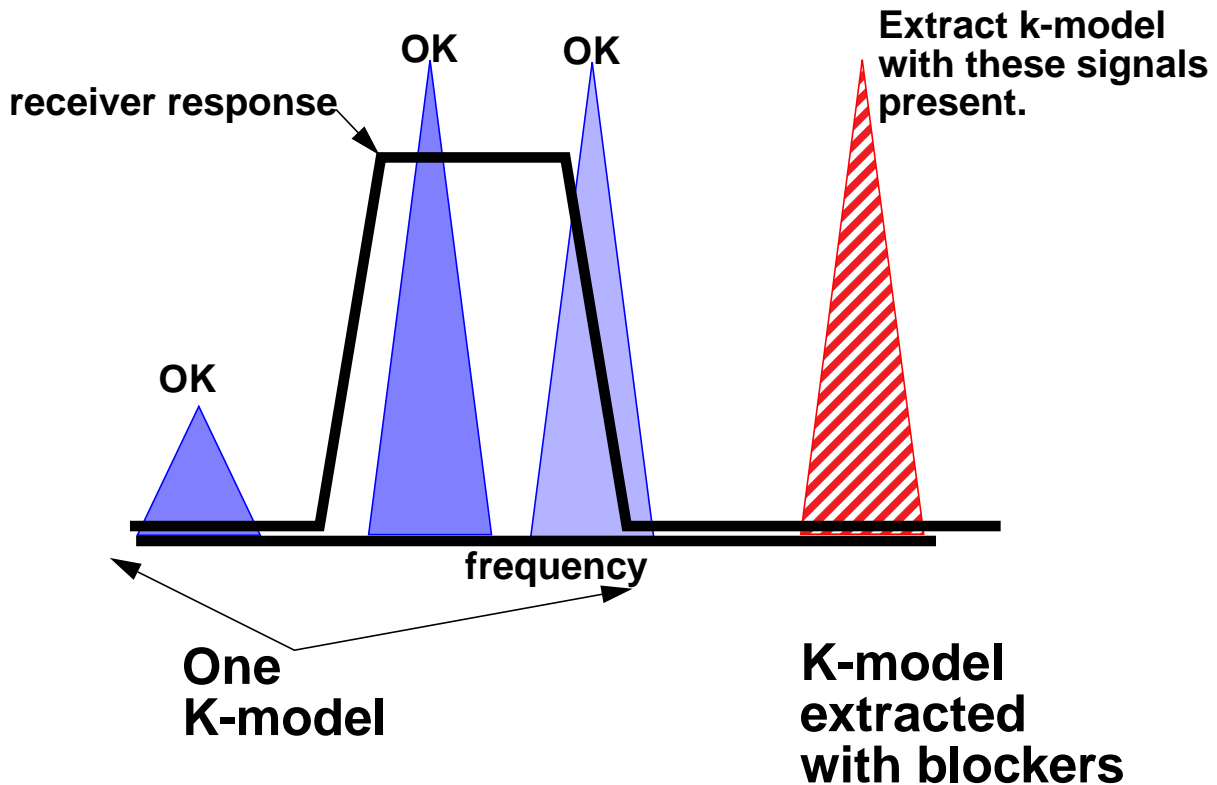


Figure 9-3 compares receiver outputs computed using a transistor-level model in SpectreRF with outputs that result from using K-models in Alta SPW. One K-model includes the blocker. The other does not.

The I-input signal is in-band and the Q-input signal is on the band edge. The blocker is more than two decades out of band. The blocker's amplitude is more than 10 times the in-band signal level that saturates the receiver.

The red dotted trajectory shows direct SpectreRF simulation results. The solid blue line shows results from the K-model simulation. The larger trajectories are without the blocker.

The K-model extracted without the blocker matches device-level simulation quite well. The K-model extracted with the blocker present also matches the corresponding device-level simulation well except for a fixed rotational error. Rotating the blue trajectory by 7 degrees

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

puts it nearly on top of the trajectory computed in SpectreRF. Such a fixed rotational error is usually irrelevant to the DSP section because the equalizer hides the error, and a fixed rotation does not affect the spectrum.

However, you can reduce the error. The K-model internally calibrates for rotational error. A blocker introduces an offset that creates calibration error. Alta SPW can modify the K-model to calibrate at a stronger input signal so that the offset has less effect.

Figure 9-4

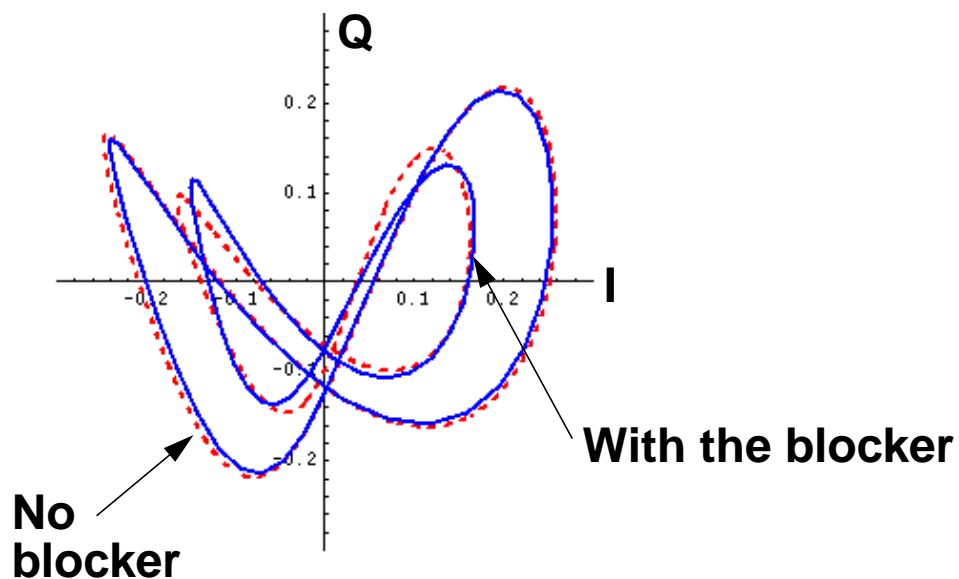
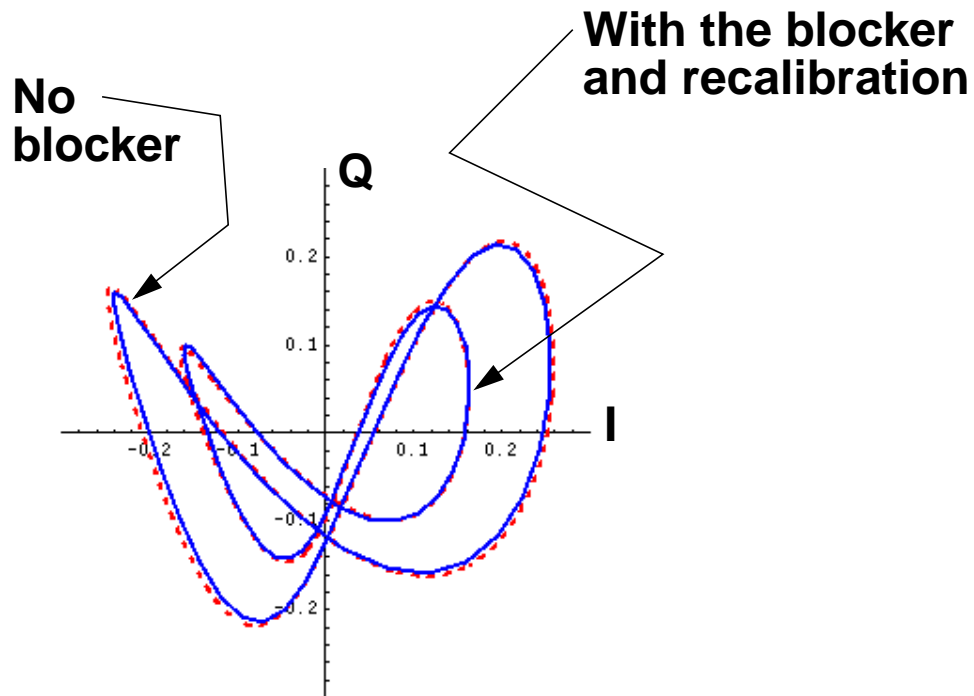


Figure 9-4 shows the results with the K-model calibrated at the stronger signal.

Figure 9-5



K-model data files

This section describes the K-model data files.

I_data.ascsig, Q_data.ascsig:

These files characterize the static behavior of the *I* and *Q* outputs, respectively, as the input bias increases along the input *I*-axis. The format is given below:

```
data
$
signal type = double
vector type = interlaced
vector length = 14
number of vectors = 2
number of signal points = 14
sampling frequency = 1
starting time = 0
$
```

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

```
0.001    0.002300
0.3008333  0.692013
0.6006667  1.381728
0.9005    2.071483
1.200333  2.568851
1.500167  2.655311
1.8      2.710871
```

dck

This file contains the lowest frequency gains of the *K11* and *K21* transfer functions. The format is shown below:

```
2.299950
-0.051965
```

sig_biases

This file lists the input signal biases. The format is shown below:

```
0.001
0.3008333
0.6006667
0.9005
1.200333
1.500167
1.8
```

psd_I.ascsig, psd_Q.ascsig

These files contain the power spectral densities of the noise observed at the baseband *I* and *Q* outputs. The first column is frequency, and the second column is power spectral density in volts/Sqrt(Hz). The format is given below:

```
# wavew4sli1() from 10e-3 to 25e6
#          wavew4sli1()

10e-3      1.334e-6
250e3      943.1e-9
500e3      943.1e-9
750e3      943e-9
1e6        942.7e-9
1.25e6     942.4e-9
```

Affirma RF Simulator (SpectreRF) User Guide

Extracting K-Models

1.5e6	941.8e-9
1.75e6	940.9e-9
2e6	939.6e-9
2.25e6	938e-9
2.5e6	935.9e-9

K11.ascsig, K21.ascsig, K12.ascsig, K22.ascsig, k11_1 through k11_6, and k21_1 through k21_6

The first four files contain the *I-I*, *I-Q*, *Q-I*, and *Q-Q* transfer functions at the smallest input bias. The format is identical to the *k21_j* and *k11_j* transfer functions.

The *k11_1* through *k11_6* files contain the transfer functions from the *I*-input to the *I*-output as the input bias sweeps from the *sig_bias* level to the *bias_end* level. The *k21_1* through *k21_6* files contain the transfer functions from the *I*-input to the *Q*-output for the same input bias levels. The format of all of these files is the same. The first column is the frequency, the second is the magnitude in volts/volt, and the third is the phase in degrees. You must keep exactly three lines of header. The format is shown below:

freq (Hz)	mag(harmonic(v	phase(harmonic
10e-3	2.300	-18.860e-9
1e6	2.300	-1.886
2e6	2.299	-3.771
3e6	2.298	-5.656
4e6	2.297	-7.540
5e6	2.295	-9.424
6e6	2.293	-11.310
7e6	2.290	-13.190

The rest of the files are obsolete binary files. They formerly supported the Alta SPW beta version of the K-model. The released version uses only the ascii files.

Extracting J-Models

The J-model, also known as a transmitter K-model, has two primary uses. It facilitates Adjacent Channel Power Ratio (ACPR) estimates, and it imports transmitter impairments into the Cierto™ Alta™ SPW environment. You can extract a J-model and then feed it directly into SPW by instantiating a J-model from the Alta SPW library and specifying the directory containing the extracted J-model data. An Alta SPW user can use the J-model to examine how a direct conversion transmitter affects the end-to-end bit error rate. Only the model extraction part of the SPW flow is described here. The Affirma™ analog circuit design environment *rfLib* includes SpectreHDL versions of the J-model that read the same data files. The *j_mod_extraction_example* simulation procedures in this chapter describe how to use the SpectreHDL version of the J-model to compute ACPR in the Affirma analog circuit design environment. The last part of this chapter briefly describes the structure of the J-model and files created by the extraction script.

The example in this chapter is highly ideal but is still a good example because it simulates quickly yet contains the main distortion mechanisms captured by the J-model.

Procedures for Simulating *j_mod_extraction_example*

Follow the steps below to simulate the *j_mod_extraction_example*:

- Copy *j_mod_extraction_example* from the Cadence *rfExamples* directory into a local directory.

Be sure the path to the local directory is defined in your *cds.lib* file.

- Choose *File – Open* in the Command Interpreter Window (CIW).

The Open File form appears. Filling in the Open File form lets you open the schematic.

In the Open File form, do the following:

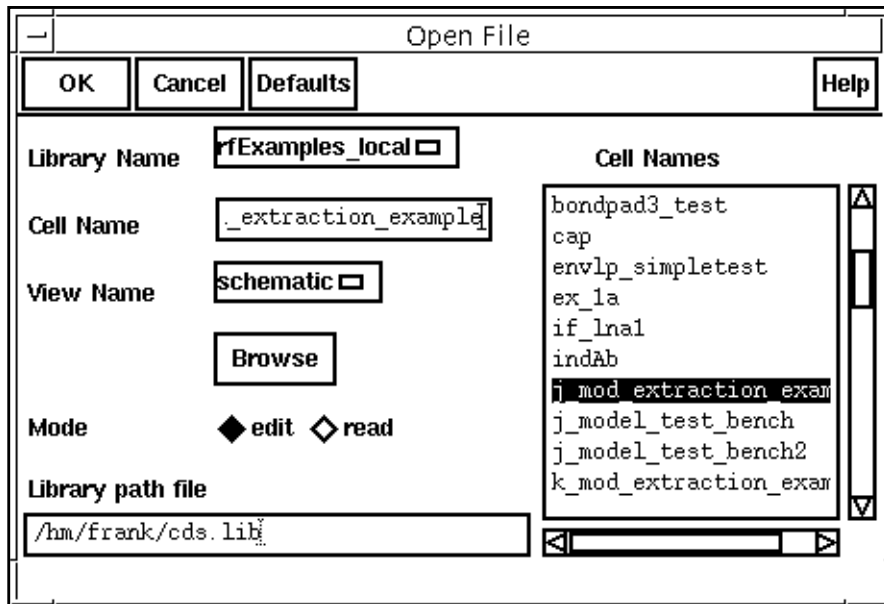
1. Choose *<Path_to_local_copy>* for *Library Name*.
2. Choose *schematic* for *View Name*.
3. In the *Cell Names* list box, highlight *j_mod_extraction_example*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

j_mod_extraction_example appears in the *Cell Name* field. (The j-model is another name for a transmitter k-model.)

4. Highlight *edit* to choose the *Mode*.



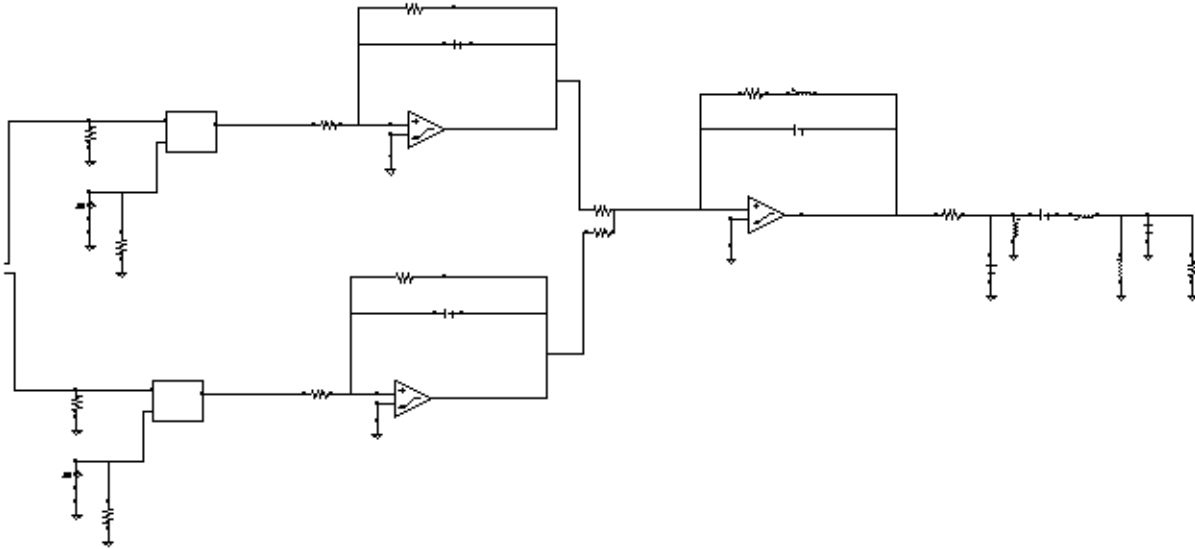
5. Click *OK* in the *Open File* form.

The Schematic window opens and displays the schematic,

If you want to see the whole procedure for setting up the J-model, including the editing of the schematic, delete the input and output test jigs at the far left and right of the schematic. The edited schematic now looks like the one below.

If you do not want to edit the schematic but only want to set up and run the simulation, skip to the Analysis Setup section.

The j_mod_extraction_example schematic



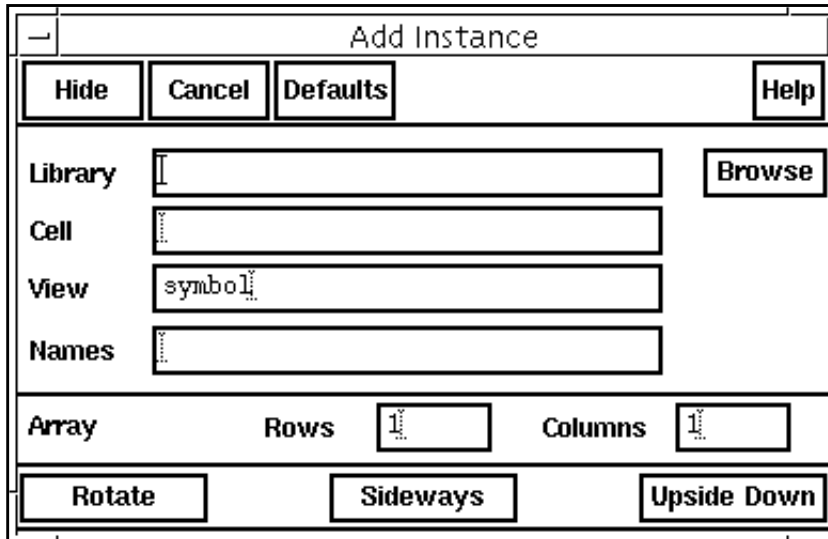
From the Schematic window, do the following:

1. Choose Add – *Instance*

Affirma RF Simulator (SpectreRF) User Guide

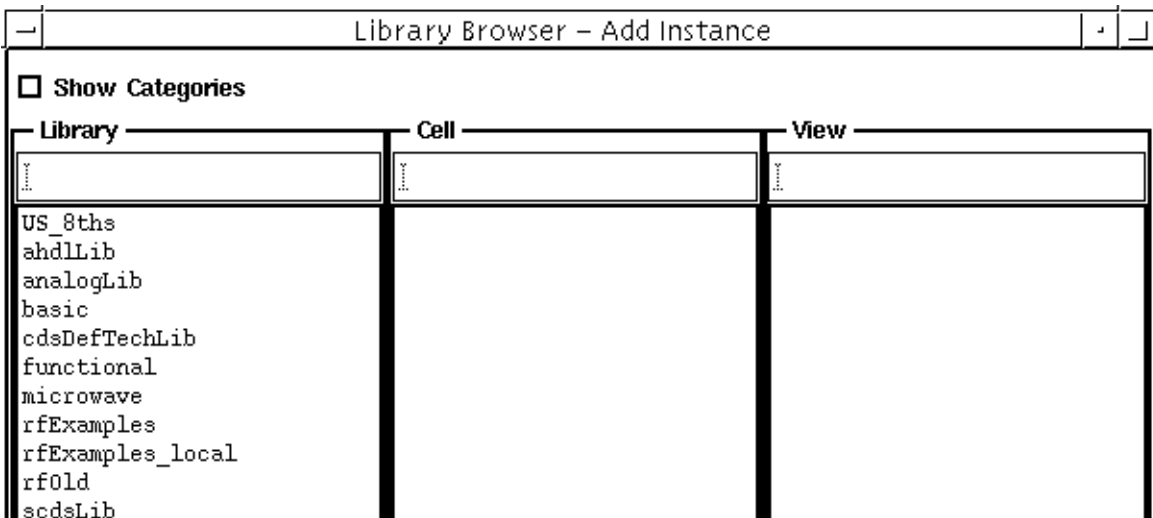
Extracting J-Models

The Add Instance form appears.



2. In the Add Instance form, click on *Browse*.

The Library Browser – Add Instance form appears.



In the Library Browser – Add Instance form, do the following:

1. Choose *rfExamples* for the *Library*.

The form changes to let you choose a cell.

2. Choose *k_mod_xmit_input_jig* for the *Cell*.

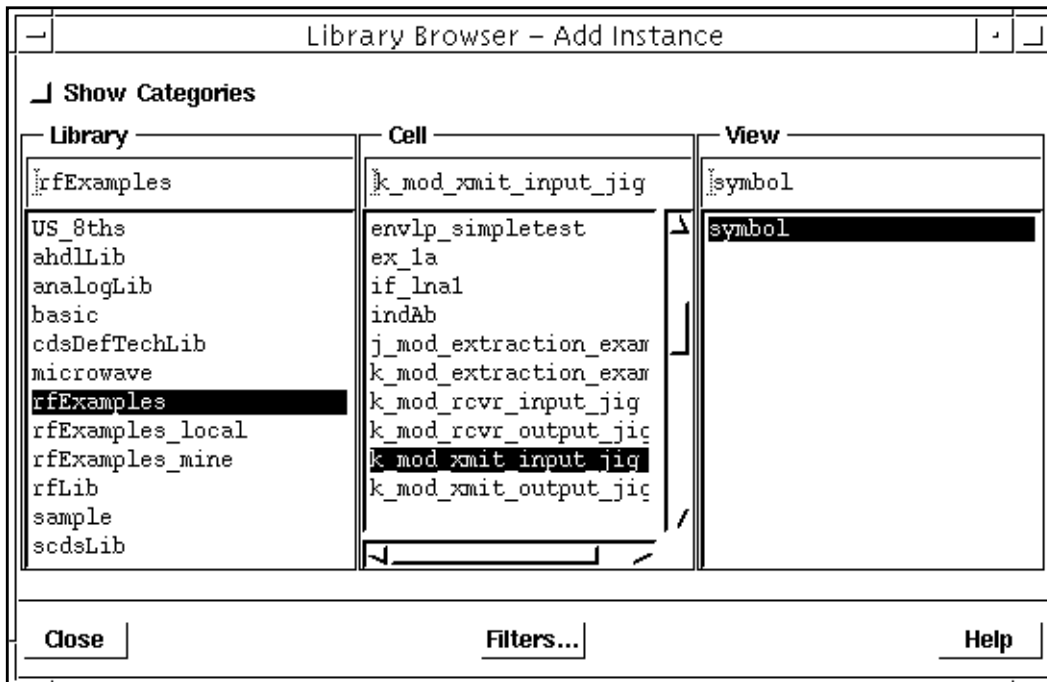
The form changes to let you choose a *View*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

3. Choose *symbol* for the *View*.

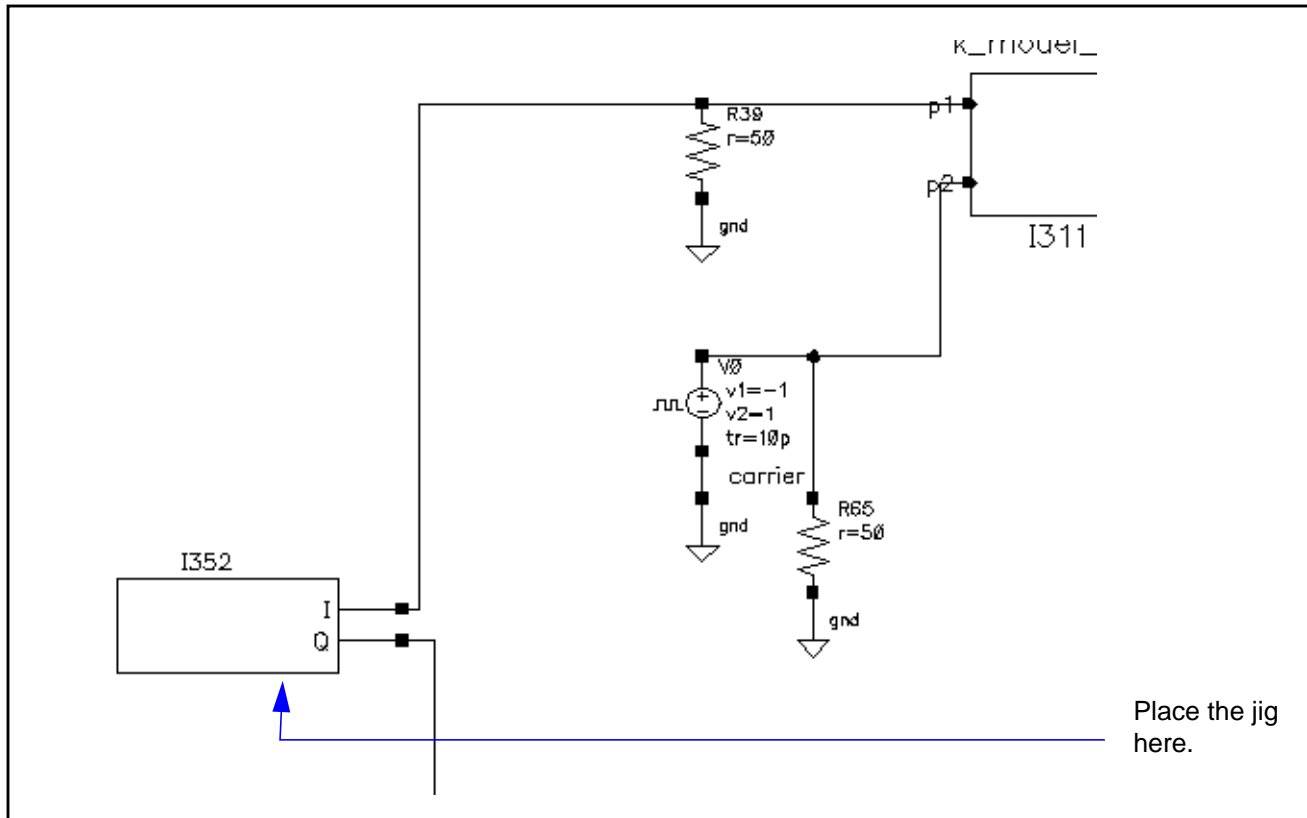
The completed form appears like the one below



Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

When you place the cursor back in the Schematic window, an instance of *k_mod_xmit_input_jig* is attached to it. Place the input jig at the appropriate place in the schematic.



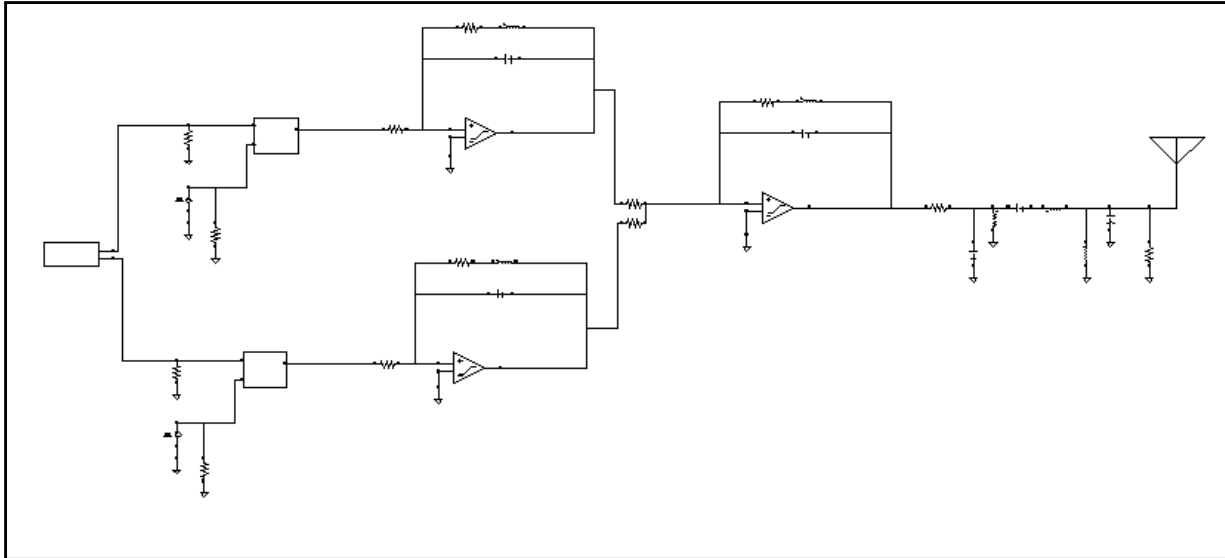
If you need assistance with the procedures for editing a schematic, see the Virtuoso® Schematic Composer™ User Guide.

In the Schematic window, again choose *Add – Instance....*

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

Now repeat the steps you used to add the input jig to the schematic and add the output jig `k_mod_xmit_output_jig`. The edited schematic looks like this.



In the Schematic window, perform the following steps:

1. Click on the input jig to select it.
2. Choose *Edit – Properties – Objects* in the Schematic window menu choices.

The Edit Object Properties form appears.

- a. In the Edit Object Properties form, type `1M` for the *Base band frequency* value.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The completed form appears like the one below:

Property	Value	Display
Library Name	rfExamples	off <input type="checkbox"/>
Cell Name	k_mod_xmit_input_jiq	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	I344	off <input type="checkbox"/>

User Property	Master Value	Local Value	Display
interfaceLastC..	09 12:39:42 1998		off <input type="checkbox"/>

CDF Parameter	Value	Display
Base band frequency	1M Hz	off <input type="checkbox"/>
Base band frequency name	bb_freq	off <input type="checkbox"/>
Wavelet: radius	radius V	off <input type="checkbox"/>

b. In the Edit Object Properties form, click *OK*.

3. In the Schematic window, choose *Design – Check and Save*.

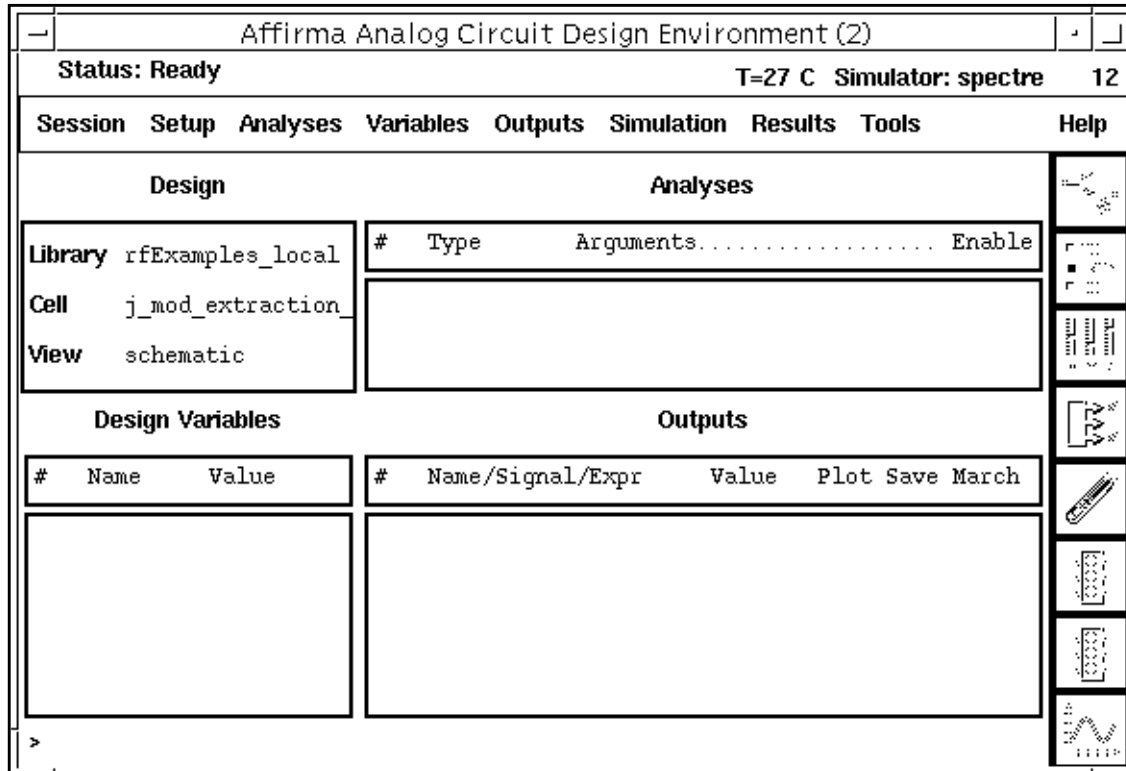
Analysis Setup

1. Choose *Tools – Analog Environment* in the Schematic window.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The Simulation window appears.



2. Choose *Setup – Simulator/Directory/Host* in the Simulation window.

The Choosing Simulator/Directory/Host form appears.

In the Choosing Simulator/Directory/Host form, do the following:

1. Choose *spectre* for *Simulator*.
2. Type in the name of the project directory, if necessary.
3. Highlight the *local* or the *remote* button to specify the *Host Mode*. For remote simulation, type in the name of the host machine and the remote directory in the appropriate fields.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The completed form appears like the one below.

The screenshot shows a dialog box titled "Choosing Simulator/Directory/Host -- Affirma Analog Circuit Desi". The dialog has four buttons at the top: "OK", "Cancel", "Defaults", and "Help". The main area contains the following fields and options:

- Simulator:** A text box containing "spectre" with a small square icon to its right.
- Project Directory:** A text box containing "~/simulation".
- Host Mode:** Three radio buttons labeled "local", "remote", and "distributed". The "local" radio button is selected.
- Host:** An empty text box.
- Remote Directory:** An empty text box.

4. In the Choosing Simulator/Directory/Host form, click *OK*.

In the Simulation window, do the following:

1. Choose *Variables – Copy From Cellview*.

The variable *radius* appears in the Design Variables list box of the Simulation window.

2. Choose *Variables – Edit....*

The Editing Design Variables form appears.

In the Editing Design Variables form, do the following:

1. In the Table of Design Variables list box, click on *radius*.
2. Type *1.5* in the *Value(Expr)* field.
3. Click *Apply*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The completed form looks like the one below:

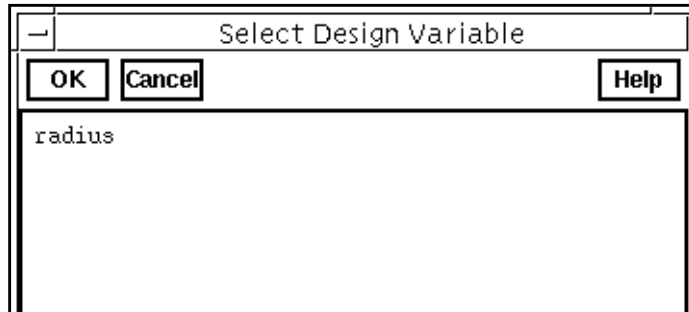
Selected Variable		Table of Design Variables	
Name	radius	#	Name Value
Value (Expr)	1.5	1	radius

4. Click *OK*.
5. Choose *Analyses – Choose* in the Simulation window.
The Choosing Analyses form appears.
6. In the Choosing Analyses form, click on *pdisto* for the analysis.
The form changes to display options needed for *Pdisto*.

In the Choosing Analyses form, do the following:

1. In the *Fundamental Tones* list box, choose one of the tones labeled *carrier*.
2. In the fields immediately below the *Fundamental Tones* list box, do the following:
 - a. Choose *Large* in the *Signal* cyclic field.
 - b. Type 1 for the *Harms* field value.
 - c. Choose *bb_freq*.
 - d. Type 9 for the *Harms* field value for *bb_freq*.
3. Highlight *moderate* for the *Accuracy Defaults (errpreset)* value.
4. Highlight *sweep*.
The form expands to let you specify sweep data.
5. Click on *Select Design Variable*.

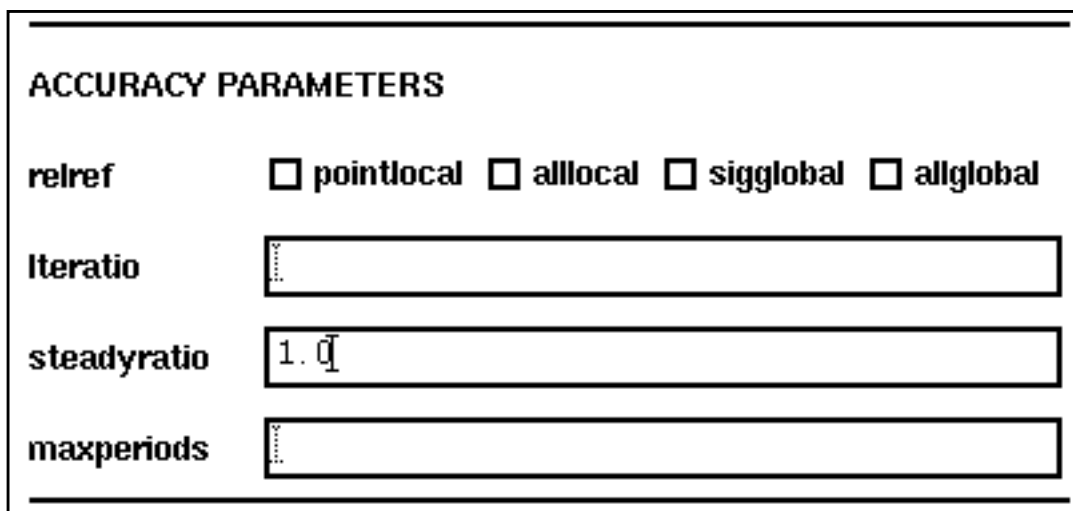
The Select Design Variable form appears.



6. In the Select Design Variable form, highlight *radius* and click *OK*.
7. Choose *Start – Stop* for *Sweep Range*.
8. Type .001 and 1.5 the *Start* and *Stop* values, respectively.
9. In the *Sweep Type* cyclic field, choose *Linear* and highlight *Number of Steps*. Type 20 in the field to the right of the button.
10. Choose *Options....*

The Periodic Distortion Options form appears.

- a. In the Periodic Distortion Options form, type 2.0 for the *steadyratio* value.¹



ACCURACY PARAMETERS

relref pointlocal allocal sigglobal allglobal

Iteratio

steadyratio

maxperiods

1. This circuit can cause convergence problems because the behavioral amplifiers have hard saturation curves. The *steadyratio* of 2 should work on all platforms but if you still experience convergence problems at any point in the Pdisto sweep, increase *steadyratio* and decrease *reltol* by the same factor. For example, if you double *steadyratio*, half *reltol*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

b. In the Periodic Distortion Options form, click *OK*.

11. Be sure the *Enabled* button is highlighted.

The completed Periodic Distortion Options form looks like this.

The screenshot shows the 'Choosing Analyses' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. Below these are several analysis options: 'tran', 'dc', 'ac', 'noise', 'xf', 'sens', 'sp', 'pdisto' (selected), 'pss', 'pac', 'pnoise', 'pxf', and 'envlp'. The 'Periodic Distortion Analysis' section contains a table of 'Fundamental Tones' and several input fields for 'bb_freq', 'Value', 'Moderate', 'I344', and 'g'. Below this is the 'Accuracy Defaults' section with radio buttons for 'conservative', 'moderate' (selected), and 'liberal', and a text field for 'Additional Time for Stabilization (tstab)'. The 'Sweep' section has a checked 'Sweep' checkbox, a 'Frequency Variable?' section with 'no' selected, and a 'Variable Name' field containing 'radius'. The 'Sweep Range' section has radio buttons for 'Start-Stop' (selected) and 'Center-Span', with 'Start' and 'Stop' fields containing '.001' and '1.5' respectively. The 'Sweep Type' section has radio buttons for 'Linear' (selected) and 'Logarithmic', and a 'Step Size' field containing '20'. At the bottom, there is an 'Add Specific Points' checkbox and an 'Enabled' checkbox which is checked, along with an 'Options...' button.

#	Name	Expr	Value	Signal	SrcId	Harms
1	bb_freq	1M	1M	Moderate	I344	9
3	carrier	1/(1n-0)	1G	Large	V1	1
2	carrier	1/(1n-0)	1G	Large	V0	1

Running the Simulation

1. In the Simulation window, choose *Simulation – Netlist and Run* to run the simulation.

The output log file appears and displays information about the simulation as it runs.

2. Check the CIW for a message saying the simulation completed successfully.

Creating the J-Model

- In the Simulation window, choose *Tools – Link to SPW....*

The Affirma RFIC Modeler for Cierto SPW form appears.

In the Affirma RFIC Modeler for Cierto SPW form, do the following:

1. Highlight *Transmitter* for the *Circuit Type*.
2. Edit the *Script File Name* and the *Extracted Kmodel Directory Name*, if necessary.

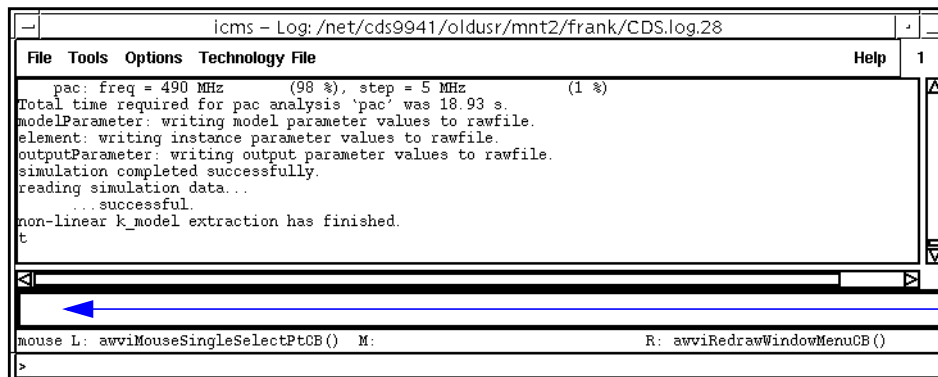
The file name is the name of the Ocean script to be run later, and the directory name is the output directory.

3. Click *OK*.

- In the CIW, type the following line and press Return.

```
load "<script_file_name>"
```

Note: Remember to put the script name in double quotes.



You can follow the progress of the script in the CIW. When it finishes, the j-model is created in the directory you specified.

Using the J-model in a circuit

In this section you create a circuit that contains the j-model you created and run an appropriate simulation.

Setting up the circuit for simulation

1. In the CIW, choose *File – New – Cellview*.

The Create New File form appears.

2. In the Create New File form, specify the following:
 - a. Choose a library name from the *Library Name* cyclic field.
 - b. Type a name for the *Cell Name*.
 - c. Type a name for the *View Name*.
 - d. Click *OK*.

The completed form looks like this.

The screenshot shows a dialog box titled "Create New File". At the top, there are four buttons: "OK", "Cancel", "Defaults", and "Help". Below these buttons are five input fields:

- Library Name:** A dropdown menu with "rfExamples_local" selected.
- Cell Name:** A text box containing "j_model_test_bench2".
- View Name:** A text box containing "schematic".
- Tool:** A dropdown menu with "Composer-Schematic" selected.
- Library path file:** A text box containing "/home/belinda/cds.lib".

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

An empty Schematic window appears.

3. In the new Schematic window, choose *Add – Instance*.

The Add Instance form appears.

The screenshot shows the 'Add Instance' dialog box. The title bar reads 'Add Instance'. Below the title bar are four buttons: 'Hide', 'Cancel', 'Defaults', and 'Help'. The main area contains four input fields: 'Library' (empty), 'Cell' (empty), 'View' (containing 'symbol'), and 'Names' (empty). To the right of the 'Library' field is a 'Browse' button. Below the input fields is an 'Array' section with 'Rows' and 'Columns' input fields, both containing '1'. At the bottom are three buttons: 'Rotate', 'Sideways', and 'Upside Down'.

4. In the Add Instance form, click on Browse.

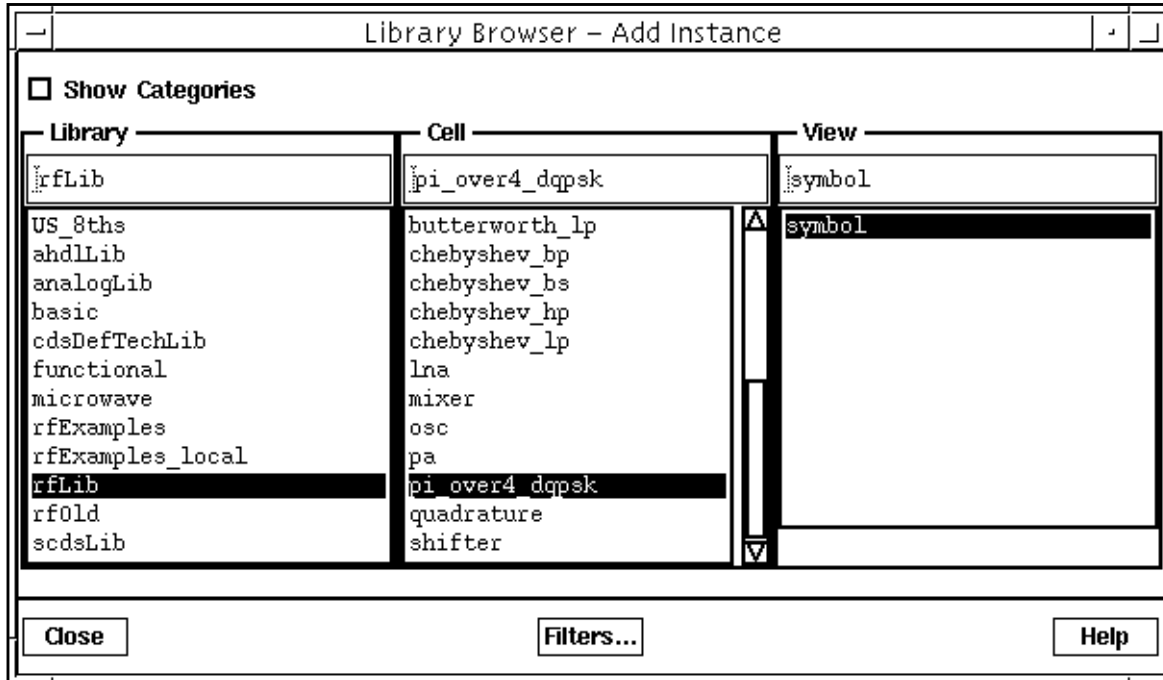
The Library Browser – Add Instance form appears.

5. In the *Library Browser – Add Instance* form, do the following:
 - a. Choose *rfLib* for *Library*.
 - b. Choose *pi_over4_dqpsk* for *Cell*.
 - c. Choose *symbol* for *View*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The completed form looks like this



- Place the cursor in the Schematic window and left click to place an instance in the schematic. Press Esc after placing the instance.

The appropriate instance is attached to the cursor when you place the cursor in the schematic. If you need assistance for working in the Schematic window, see [Virtuoso® Schematic Composer™ User Guide](#).

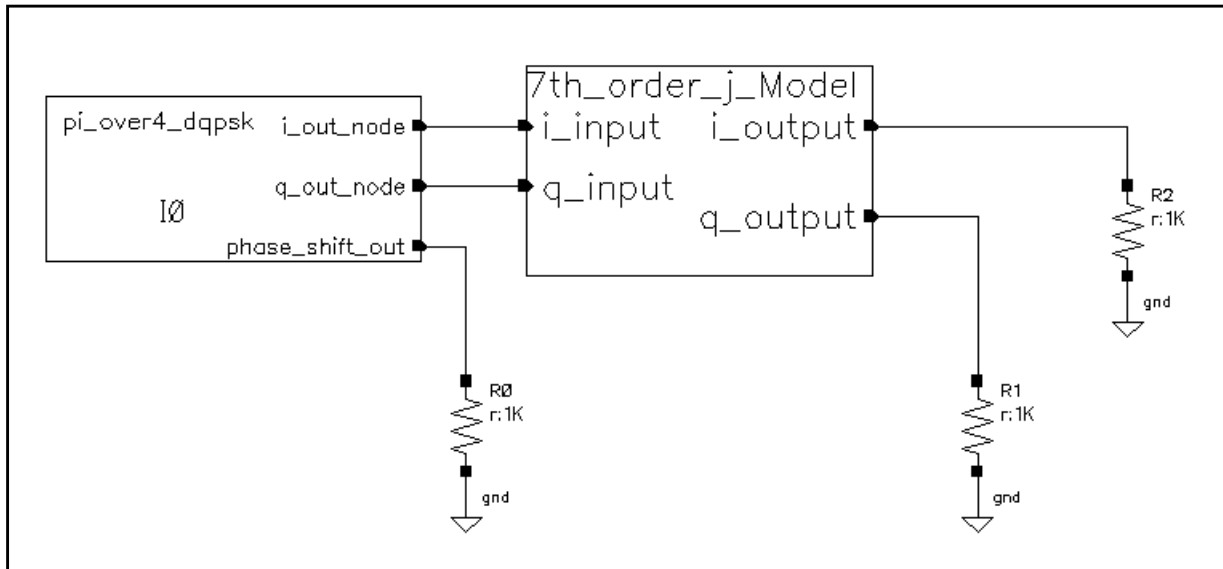
- Repeat the procedure you used to place the *pi_over4_dqpsk* instance in the schematic to add a *7th_order_j_model* instance from the *rfLib* along with three resistors and three grounds attached to the resistors. You create multiple copies of an instance with repeated left mouse clicks. The table below tells you the *Library*, *Cell*, and *View* values to choose.

Library	Cell	View
rfLib	7th_order_j_model	symbol
analogLib	res	symbol
analogLib	gnd	symbol

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

8. In the Schematic window, choose *Add – Wire (narrow)*. Click on beginning and ending locations in the schematic to add wires so the final schematic looks like the one below. Press *ESC* after you finish adding the wires.



9. In the Schematic window, highlight the `pi_over4_dqpsk` instance and choose *Edit – Properties – Objects....*

Specify the following in the Edit Object Properties form:

- a. Choose *verilog* for *CDF Parameter of view*.
- b. Type `.5` for *amplitude*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The completed form looks like this

Property	Value	Display
Library Name	rfLib	off <input type="checkbox"/>
Cell Name	pi_over4_dqpsk	value <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	I0	value <input type="checkbox"/>

c. Click *OK*.

10. In the *Schematic* window, highlight the *7th_order_j_model* instance, and repeat the previous steps to bring up the *Edit Object Properties* form. Edit the form as follows:

- a. Choose *ahdl* for *DCF Parameter of view*.
- b. Type the full path to the *j_model* for the *data_directory* value.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The completed form looks like this.

Property	Value	Display
Library Name	rfLib	off
Cell Name	7th_order_j_model	off
View Name	symbol	off
Instance Name	I1	off

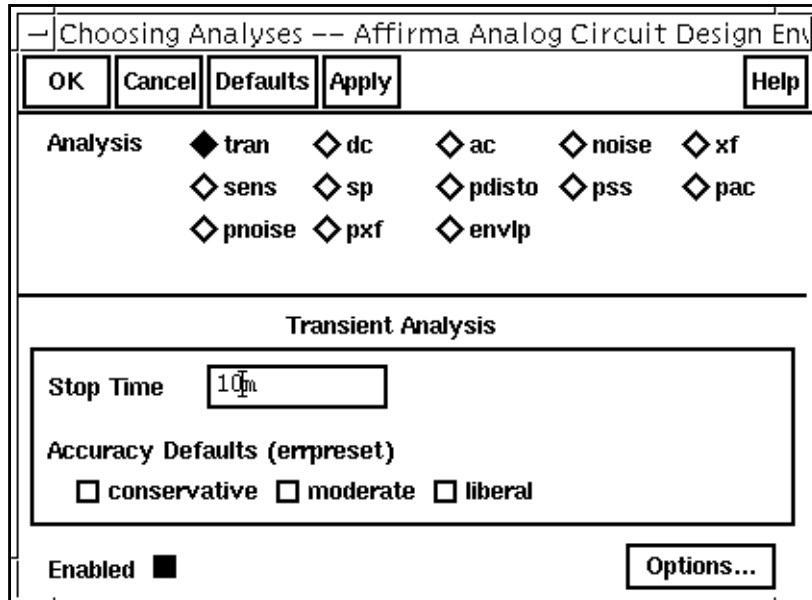
CDF Parameter of view	Value	Display
ahdl		
data_directory	jsr/mnt2/frank/j_model	off

c. Click *OK*.

11. In the Schematic window, choose *Design – Check and Save*.
12. In the Schematic window, choose *Tools – Analog Environment*.
The Simulation window appears.
13. In the Simulation window, choose *Analyses – Choose....*
The Choosing Analyses form appears.
14. Specify the following in the Choosing Analyses form:
 - a. Type 10m for the *Stop* time.
 - b. Highlight *moderate* for *Accuracy Defaults (errpreset)*.

You can also leave this value unselected because *moderate* is the default.

c. Be sure the *Enabled* button is highlighted.



d. Click *OK*.

15. In the Simulation window, choose *Simulation – Netlist and Run* to run the simulation.

Check the CIW for a message saying the simulation completed successfully.

Working with the simulation results

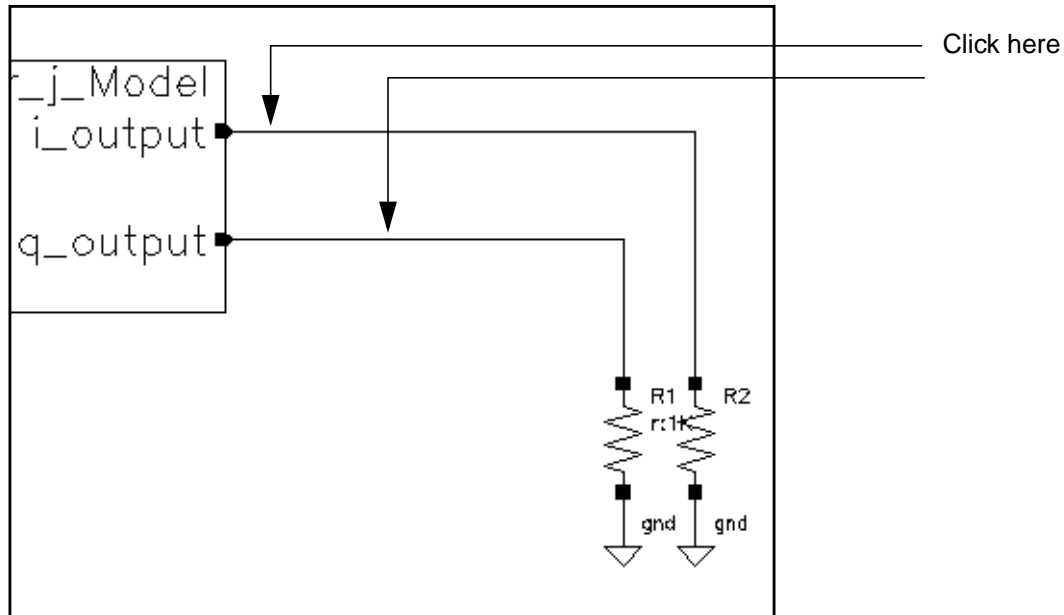
1. In the Simulation window, choose *Results – Direct Plot – Transient Signal*.

An empty Waveform window appears.

Affirma RF Simulator (SpectreRF) User Guide

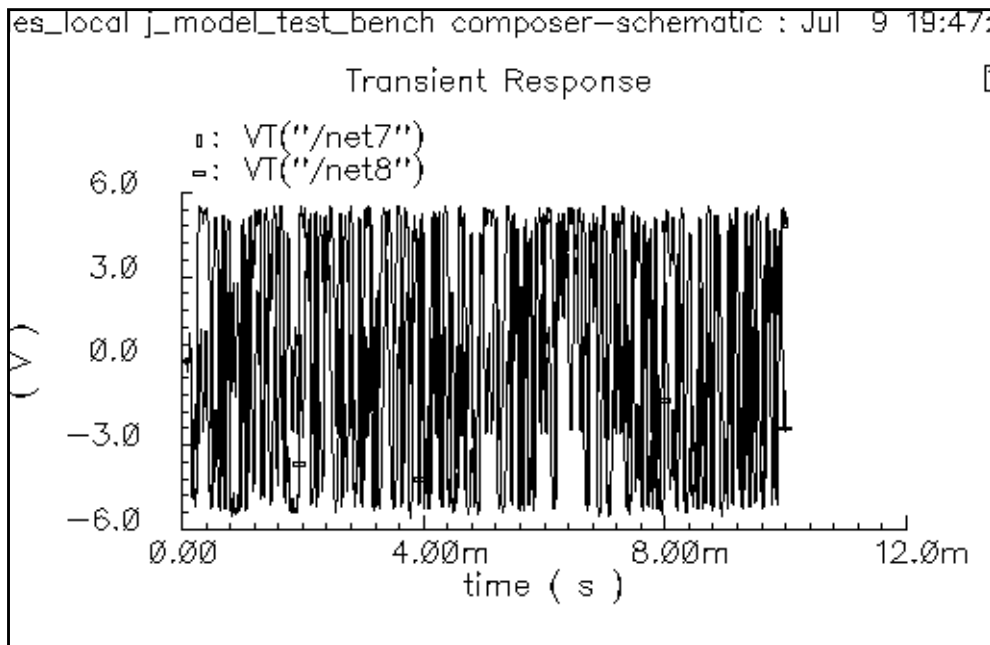
Extracting J-Models

2. In the Schematic window, click on the i_output net and then on the q_output net



3. Press the *ESC* key.

The plots for i_output and q_output appear in the Waveform window.



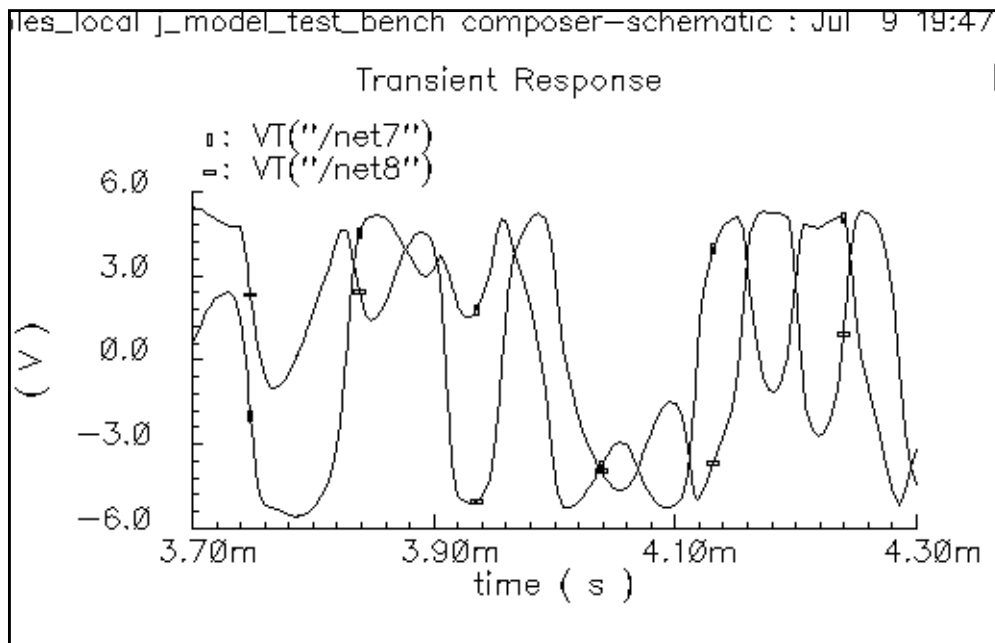
4. Highlight a small portion of the waveform by performing the following steps:

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

- a. In the Waveform window, choose *Zoom – Zoom In*.
- b. Left click and drag with the mouse to form a rectangle that includes the waveform over a small time value.
- c. Left click to complete the selection.

The waveform display changes to show the portion you select with the mouse.



5. In the Simulation window, choose *Tools – Calculator....*

The Waveform Calculator appears.

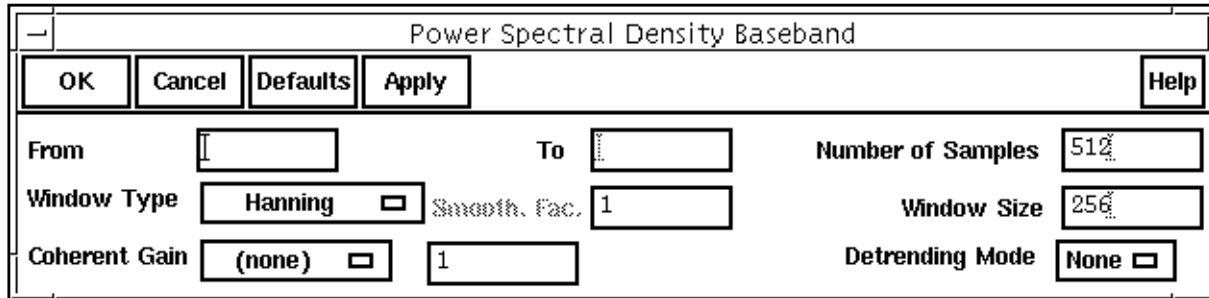
6. In the Waveform Calculator, click on the *wave* key.
7. In the Waveform window, click on each of the two waves that are displayed.

In the Waveform Calculator, choose *psdbb...* from the *Special Functions* cyclic field. The *psdbb* function is documented in the Waveform Calculator section.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The Power Spectral Density Baseband form appears.



The screenshot shows a dialog box titled "Power Spectral Density Baseband". At the top, there are buttons for "OK", "Cancel", "Defaults", "Apply", and "Help". Below the buttons, the dialog is organized into several sections:

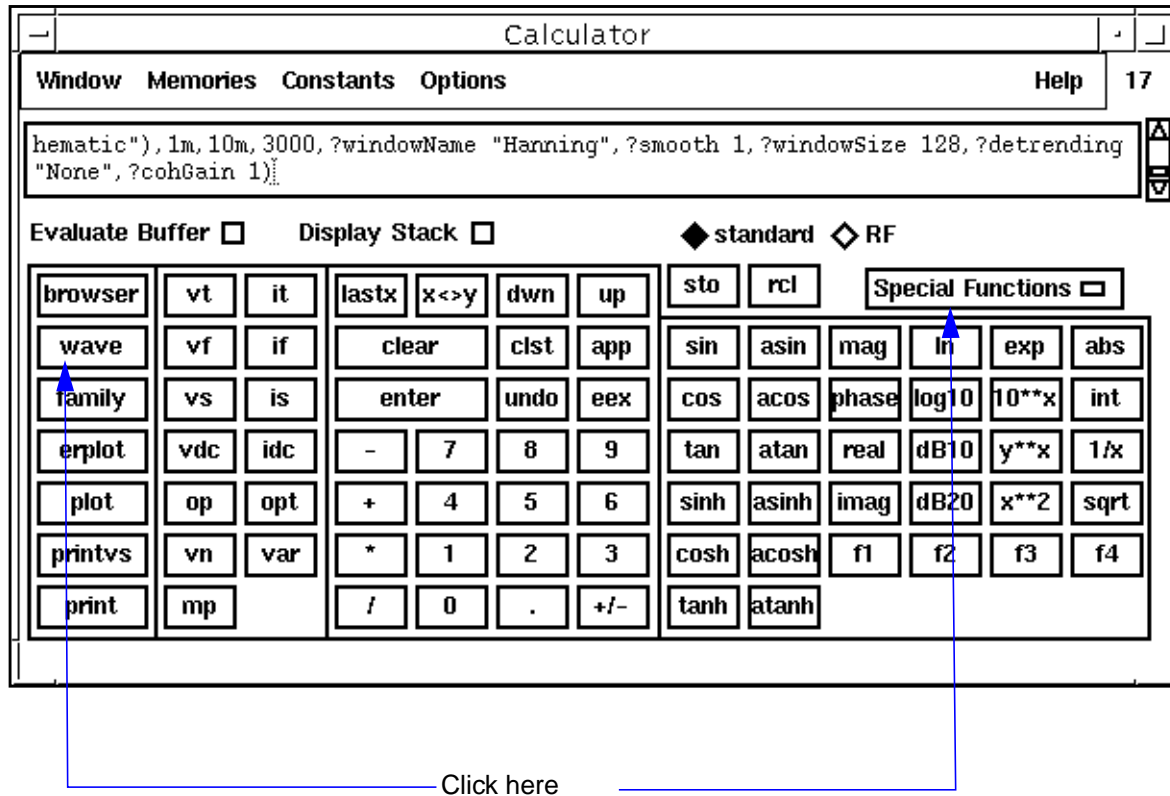
- From**: A text input field.
- To**: A text input field.
- Number of Samples**: A text input field containing "512".
- Window Type**: A dropdown menu showing "Hanning" with a small square icon to its right.
- Smooth. Fac.**: A text input field containing "1".
- Window Size**: A text input field containing "256".
- Coherent Gain**: A dropdown menu showing "(none)" with a small square icon to its right.
- Detrending Mode**: A dropdown menu showing "None" with a small square icon to its right.

8. In the Power Spectral Density Baseband form, do the following:
 - a. Type 1m in the *From* field.
 - b. Type 10m in the *To* field.
 - c. Type 3000 for the *Number of Samples*.
 - d. Type 128 for the *Window Size*.
 - e. Click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The Waveform Calculator now looks like this.

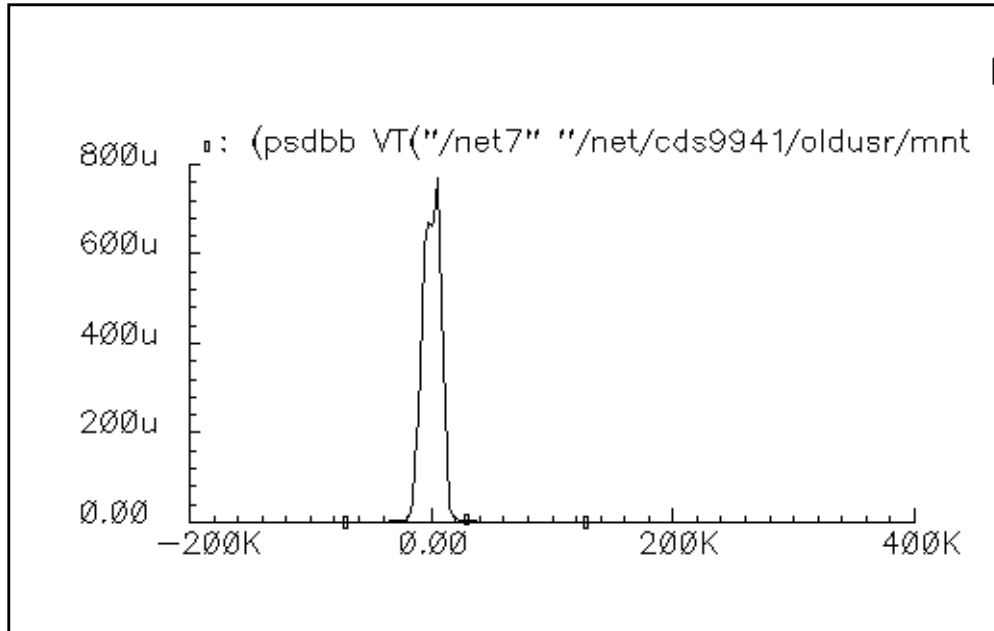


9. In the Waveform window, choose *Window – Reset*.
10. In the Waveform Calculator, click on *Plot*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The plot for *psdbb* appears in the *Waveform* window.

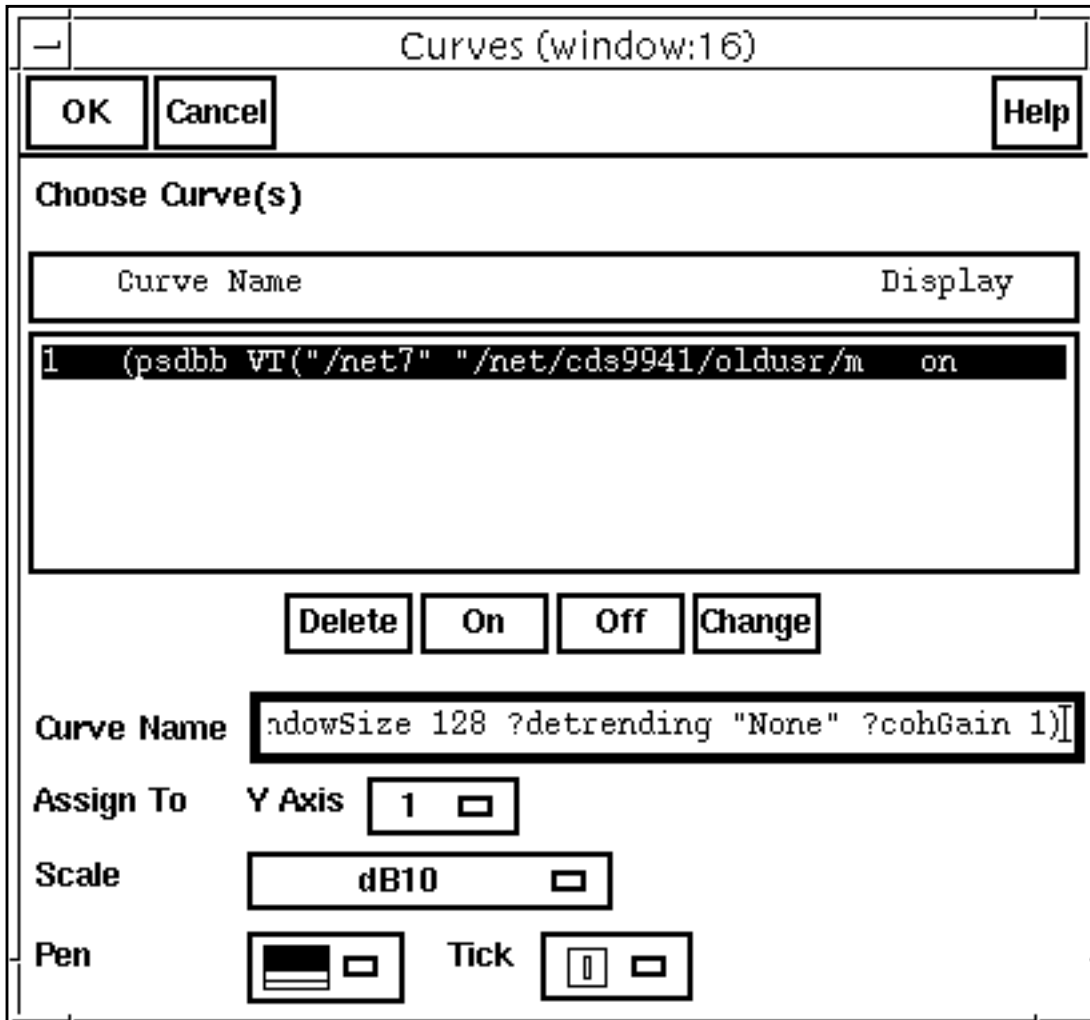


11. In the *Waveform* window, click on *Curves – Edit*.
The *Curves* window appears.
12. In the *Curves* window, do the following:
 - a. Click on the value in the *Choose Curve(s)* list box.
 - b. Choose *db10* for *Scale*.

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

The Choose Curve(s) list box looks like this.



Curves (window:16)

OK Cancel Help

Choose Curve(s)

Curve Name	Display
1	(psdbb VT("/net7" "/net/cds9941/oldusr/m on

Delete On Off Change

Curve Name ndowSize 128 ?detrrending "None" ?cohGain 1)

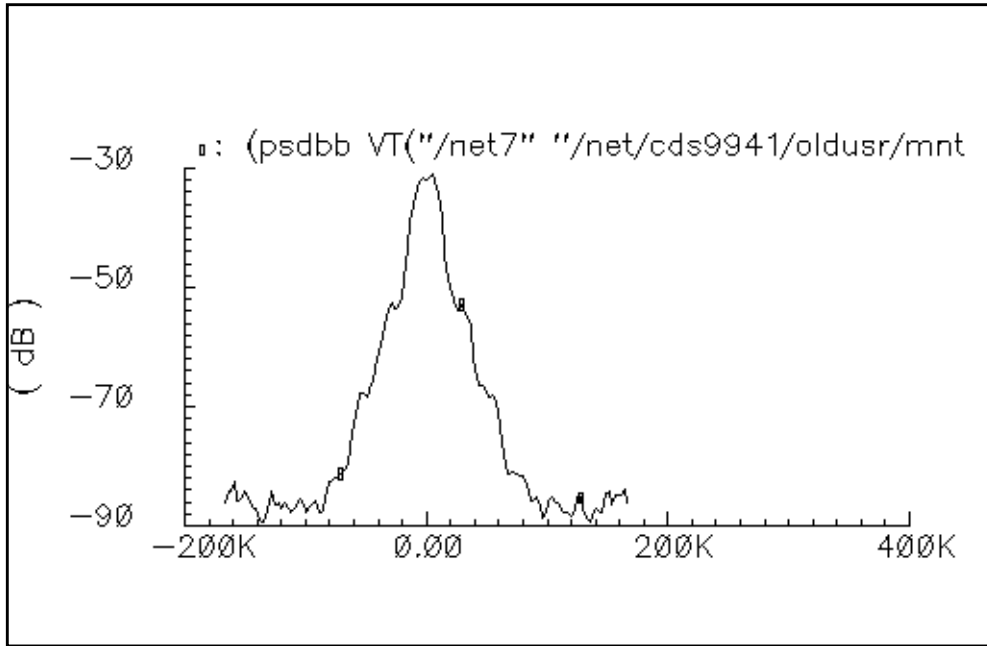
Assign To Y Axis 1

Scale dB10

Pen Tick

c. Click *OK*.

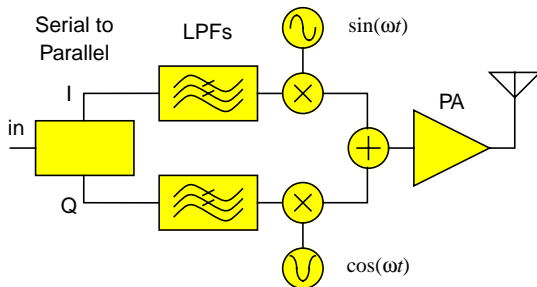
The plot changes to reflect your new specifications. ACPR is computed by taking the difference in db between the power spectral density measured at the center frequency and at the adjacent channel.



More About the J-model

Figure 1 shows a typical direct-conversion digital transmitter architecture.

Figure 10-1 Direct conversion transmitter



The low-pass filters can eliminate out-of-band signal components from the input baseband signal. However, these components might reappear because of intermodulation distortion

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

generated within the transmitter, such as in the downstream power amplifier. This return of unwanted signal components is called “spectral regrowth.”

Adjacent Channel Power Ratio (ACPR) is a popular measure of spectral regrowth in digital transmitters and is often a design specification. To get the frequency resolution required for accurate estimation of ACPR, the simulation often must process a random sequence of between 500 and 10k input symbols.

Envelope Following simulates the transmitter efficiently, but it must be driven by deterministic or unfiltered random signals, and a simulation of 10k signals is lengthy.

Most filters, like those in [Figure 1](#), are FIR filters. SpectreRF does not work with FIR filters (implemented in AHDL) because they contain hidden state. To drive an Envelope Following analysis with realistic random baseband signals, you must create the signals in a separate simulation, store them to a file, then read the data into the Envelope Following analysis with piece-wise linear sources. However, the Pdisto analysis can perform a fast, indirect ACPR estimation that can simulate 10k symbols within minutes after you extract the model. Extraction time is independent of the number of symbols you want to simulate.

Initially, Pdisto extracts a behavioral baseband-equivalent model of the transmitter. In transmitter circuits, the input baseband signal is usually well within the transmitter’s bandwidth, so a model without memory, such as the J-model, is often sufficient. Because the behavioral model eliminates the carrier and unnecessary circuit details, the subsequent ACPR calculation step is fast, even for large or complex circuits.

The most familiar spectral regrowth mechanism is the intermodulation distortion of amplitude modulation (AM) signal components that enter the power amplifier. Modulation schemes that carry information only on the carrier phase or frequency try to minimize spectral regrowth by eliminating this AM component. However, the digital baseband filters can still convert discontinuous phase changes into amplitude transients in the composite RF signal. Models based on AM/AM and AM/PM conversion [2] capture most such power amplifier related distortion mechanisms. However, imperfections in the I/Q modulators can convert input phase variations into output amplitude and phase variations that also contribute to the distortion. This is particularly true for GSM transmitters. For this reason, we also include PM/AM and PM/PM conversion effects in our model. Formally, the J-model is a narrowband approximation to a vector-valued, periodically time-varying, functional series expansion [3].

To understand the extraction procedure, write the inputs in polar coordinates, such as magnitude ρ and phase ϕ . Consider the mapping $f(\rho, \phi)$ from baseband input to the baseband representation of the RF output. That is, if

$$input = \rho \cos(\phi) + j\rho \sin(\phi),$$

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

$$output = Re[f(\rho, \phi)]\cos(\omega_c t) - Im[f(\rho, \phi)]\sin(\omega_c t) = Re[f(\rho, \phi)\exp(j\omega_c t)].$$

The baseband representation of the output is $f(\rho, \phi)$. The RF carrier frequency is ω_c . Because we assume the system is memoryless with respect to baseband signals, and that we have only one frequency translation, we can assume $f(\rho, \phi)$ is periodic in ϕ and can be expressed as a Fourier series,

$$f(\rho, \phi) = \sum_k A_k(\rho) e^{jk\phi} \quad ,$$

where the magnitude-dependent Fourier coefficients are

$$A_k(\rho) = \frac{1}{2\pi} \int_0^{2\pi} f(\rho, \phi) e^{-jk\phi} d\phi \quad .$$

To extract the Fourier coefficients, the *I* and *Q* inputs of the transmitter are driven with sinusoids in quadrature, at a frequency within the transmitter's bandwidth. For these circular input trajectories, $\phi = \omega_0 t$, and the Fourier coefficients are given by

$$A_k(\rho) = \frac{\omega_0}{2\pi} \int_0^{2\pi/\omega_0} f(\rho, \omega_0 t) e^{-jk\omega_0 t} dt \quad .$$

Therefore, for a given input magnitude, the Fourier coefficients are obtained directly from the output spectrum calculated by Pdisto. For example, if the input circle is large enough to alternately saturate the modulators, the -3 harmonic of the complex baseband tone dominates the distortion. The output spectrum has lines at $\omega_c + \omega_0$ and $\omega_c - 3\omega_0$. PDISTO computes the real and imaginary parts of the Fourier coefficients associated with those lines. The simulation is repeated for a range of input magnitudes to capture the magnitude dependence of the Fourier coefficients. At each input magnitude, the fundamental and its relevant harmonics are recorded for interpolation. The model implementation reads the Fourier coefficients and then processes any amount of input baseband data according to $f(\rho, \phi)$.

[Figure 2](#) shows the model extraction process for the fundamental and (-3) terms in the expansion $f(\rho, \phi)$. The extraction script records all harmonics up to 9th order, including even-numbered harmonics. [Figure 3](#) shows a block diagram of the associated model for just the fundamental and -3 harmonic.

Figure 10-2 Model Extraction

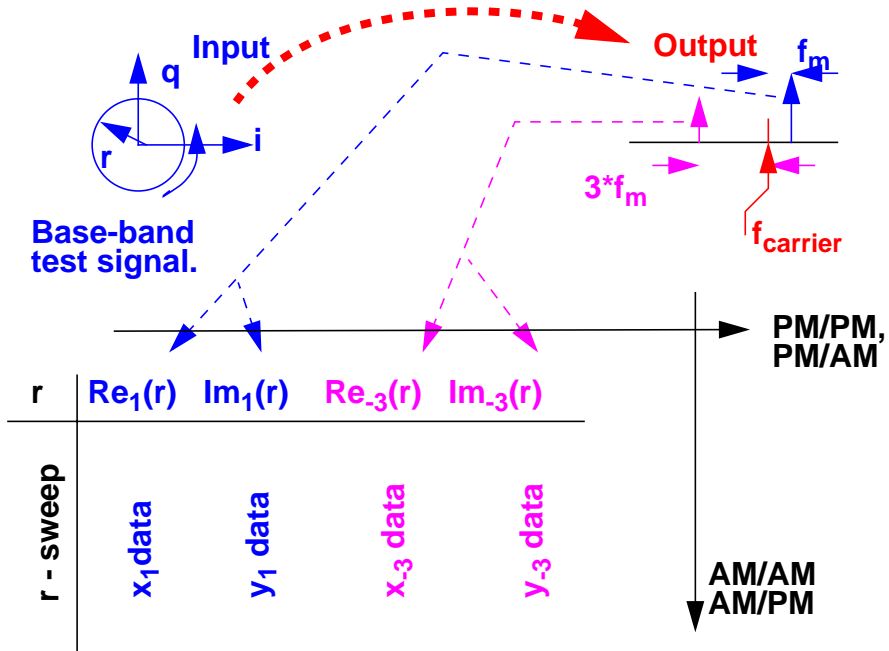
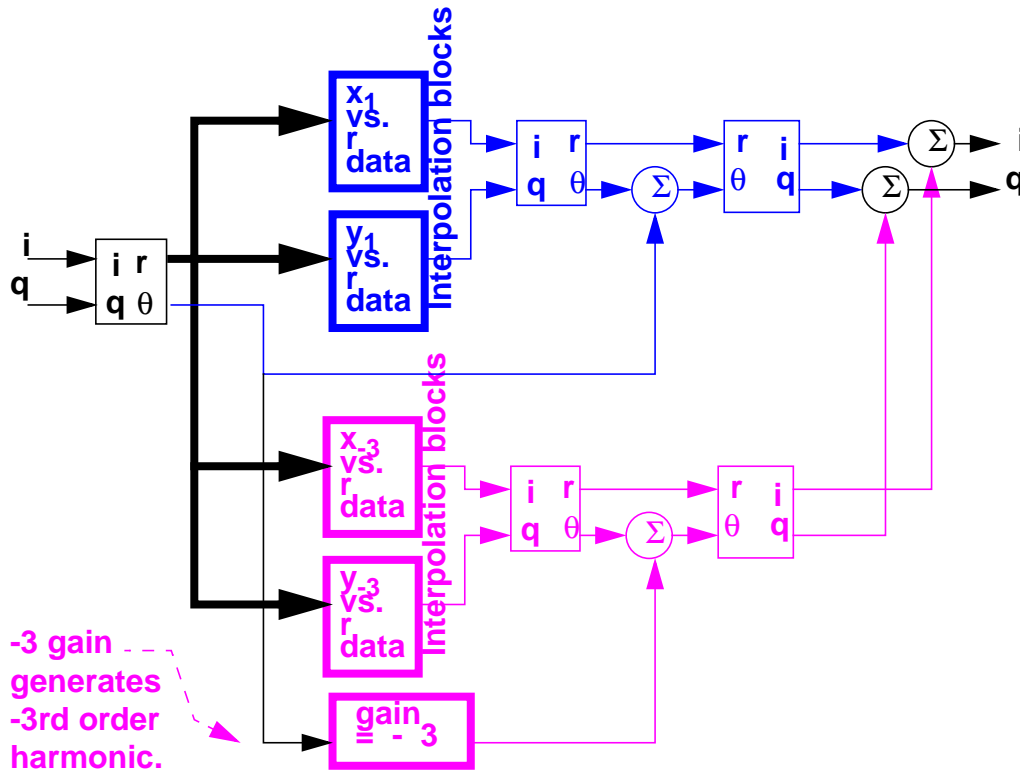


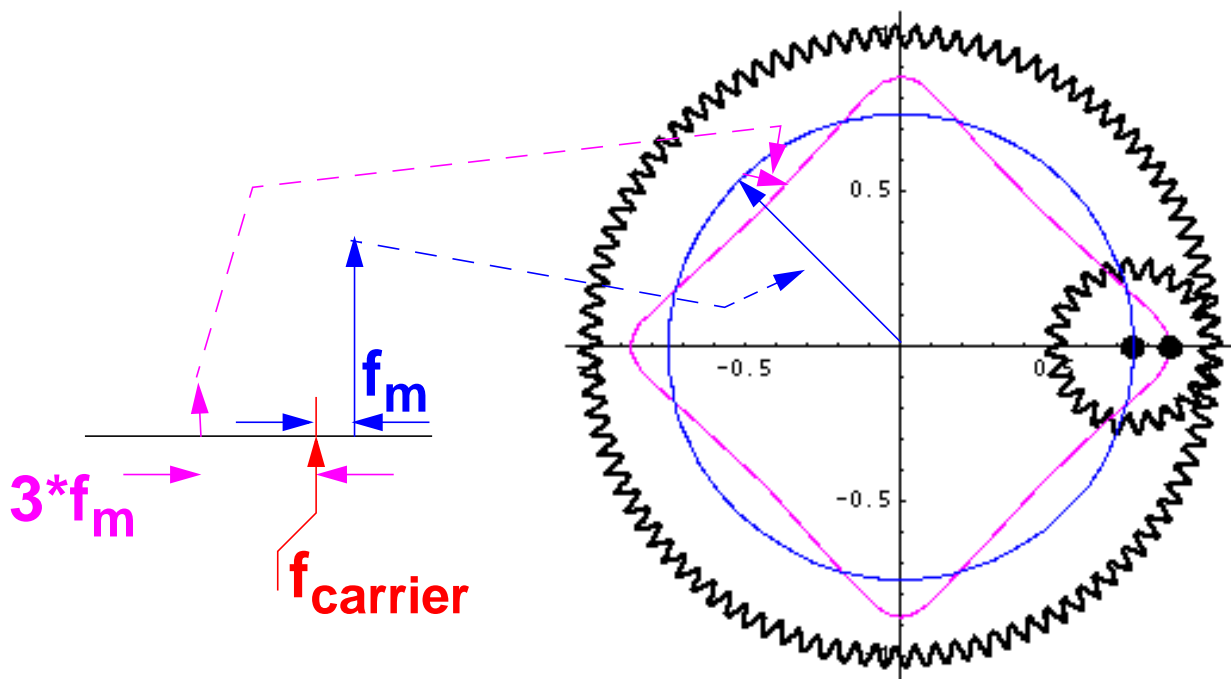
Figure 10-3 Block diagram



You can visualize PM/AM conversion, PM/PM conversion, and the new macromodel with the help of the Spirograph™ in [Figure 4](#). Consider just one distortion harmonic, the minus three harmonic, in the baseband output that is generated when the input trajectory is circular. There are no baseband harmonics generated by AM/AM or AM/PM conversion because the input radius is constant. The fundamental term, the ideal term, in the power amplifier output is represented by a vector from the center of the large inverted gear to the center of the small non-inverted gear. Since the input trajectory is circular, the fundamental vector rotates about the origin at the same frequency the input trajectory rotates about its origin. That is exactly the trajectory traced out by a pen placed in a hole at the center of the small gear as the small gear rolls around the inside of the inverted gear. If we place the pen in a hole **offset** from the center of the small gear, the vector from the center of the small gear to the hole represents the $-3\omega_0$ distortion term. The small gear rotates -3 times for each complete trip around the inverted gear. This is precisely what the model does for each recorded harmonic. The model determines the radii of both gears, as well as the phase offsets, from the recorded Pdisto results. (Although not intuitive, the -3 frequency ratio really does produce a four-cornered trace, demonstrating that the square pattern caused by independently saturating I/Q

modulators results from a minus third harmonic.) The gear analogy clearly illustrates PM/AM and PM/PM conversion. The offset hole distorts the otherwise circular trajectory with phase and amplitude perturbations that vary with the input phase.

Figure 10-4 Spirograph™ plot



To measure run times we contrived a benchmark transmitter that included the up-conversion mixers and power amplifier. The benchmark circuit contained 46 transistors and generated 328 equations in the circuit simulator. A 7th order model took 4 hours to extract and simulated 30ms of CDMA data (40k chips @ 1.25Mchips/sec) in less than 21 minutes using a Sun Ultra 1.

Driving 1st order and 7th order models with CDMA and GSM baseband signals shows when PM/AM effects become important. The first order model only accounts for AM/AM and AM/PM conversion. The 7th order model includes PM/PM and PM/AM conversion as well AM/AM and AM/PM conversion. Figures 5a and 5b show the output CDMA trajectories. Although the trajectory from the 7th order model is more square, the power spectral density (PSD) estimates are nearly identical, as Figure 6 shows. Figure 7 and Figure 8 show the output trajectories and power spectral densities when the models are driven by GSM signals of the same maximum amplitude. As in Figure 6, the 7th order trajectory is more square. However, unlike the CDMA spectral results, the GSM spectral results differ significantly. Because the

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

GSM input signal has no AM component, only the higher order models capture the true ACPR beyond about 150 KHz.

Figure 10-5

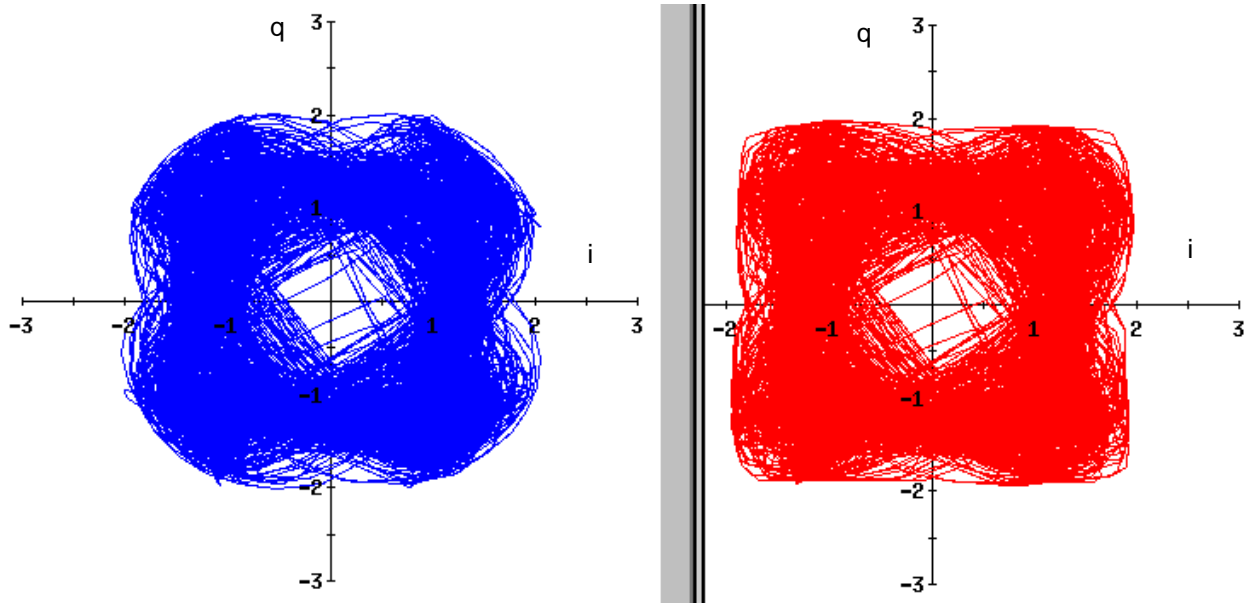


Figure 5a, 1st order output

Figure 5b, second order output

Figure 10-6

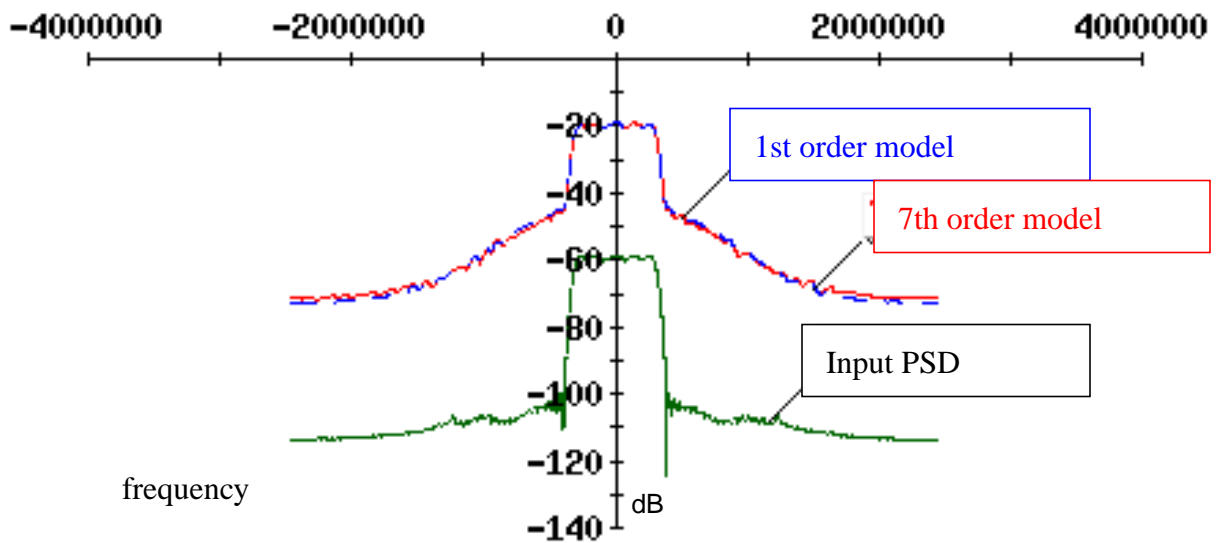


Figure 10-7 Trajectory for GSM

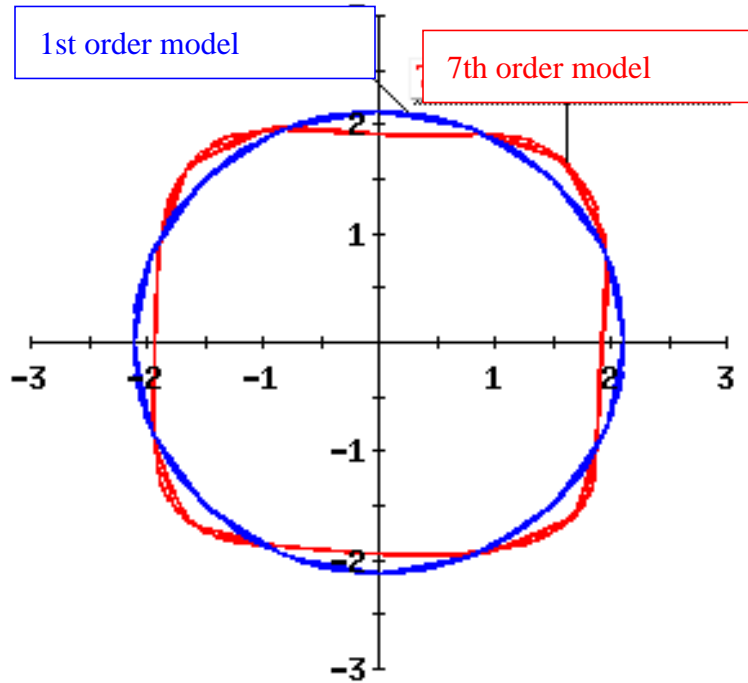
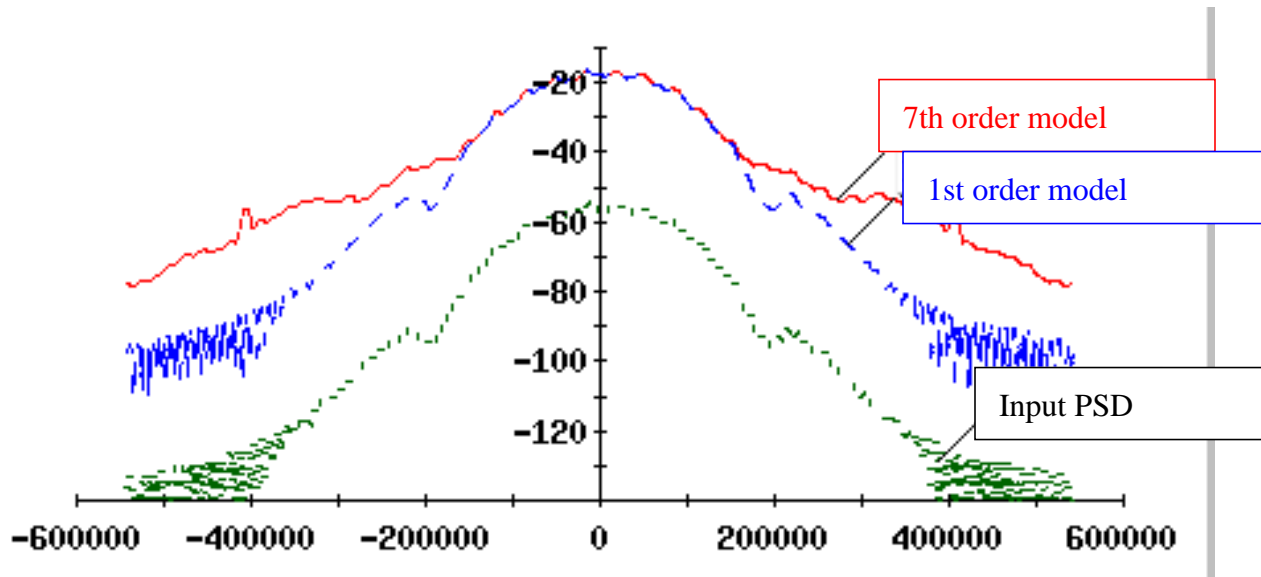


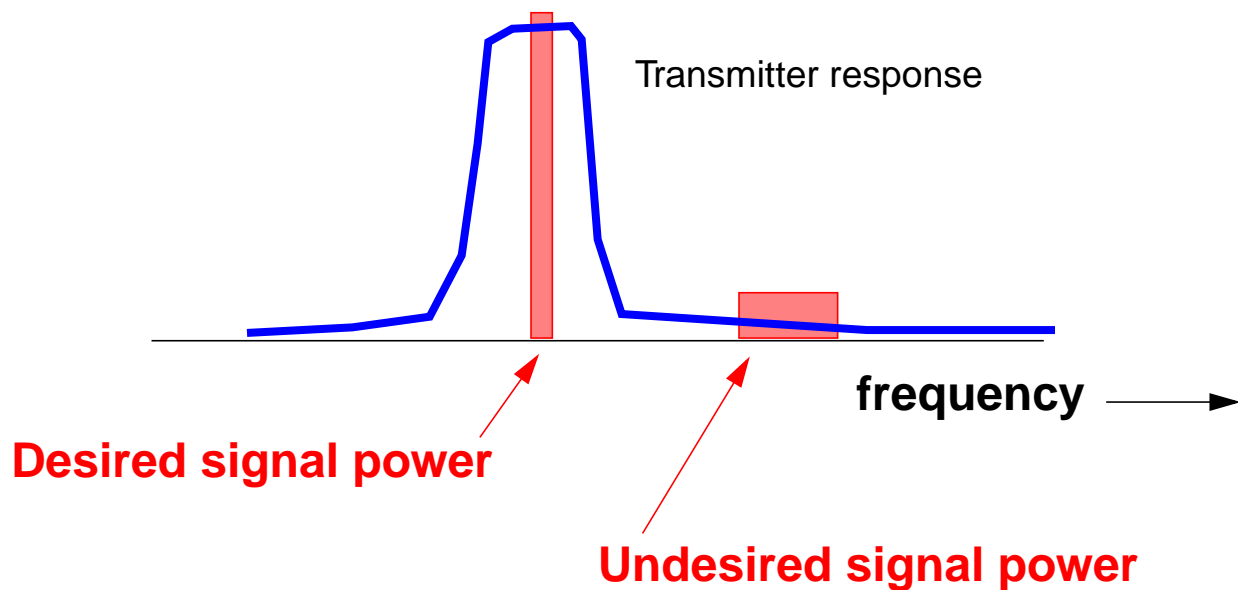
Figure 10-8 PSD for GSM



Limitations

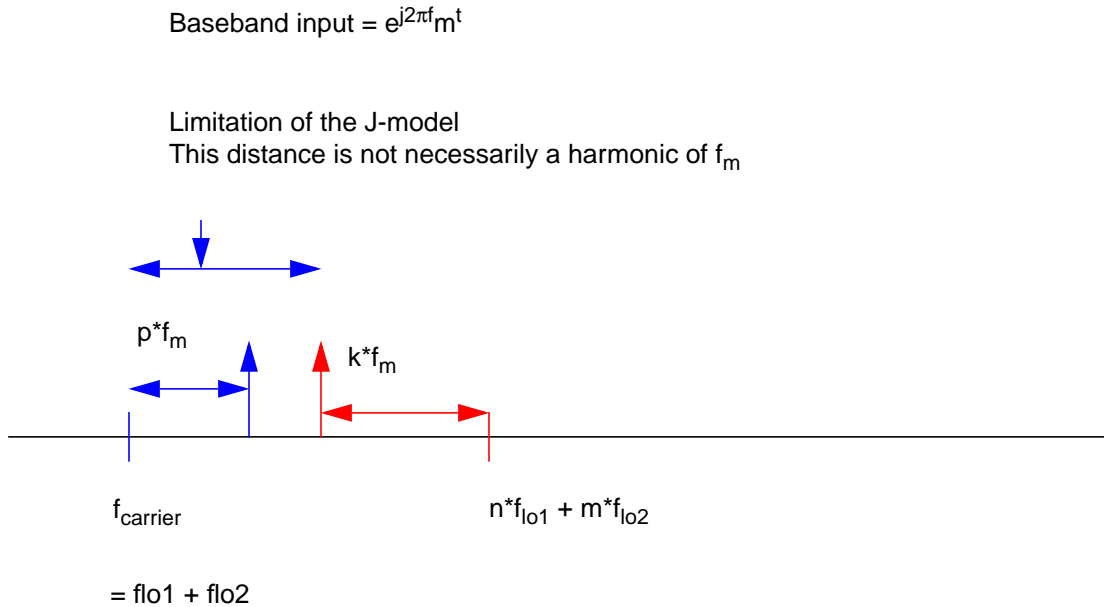
Like any behavioral model, the J-model has limitations. The J-model is a static model that has no memory. You can see this in the mechanical analogy. The pattern traced out by the pen in the small gear does not depend on how fast the pen and gear move. The Figure below illustrates the limitation in the frequency domain. Let the thick curve be the transmitter's frequency response to baseband input. The rectangles show where signal power exists. The smaller rectangle of signal power in the picture below is generated within the transmitter. It is not part of the input signal. The input signal must be contained in the larger rectangle. As long as the transmitter's frequency response is flat over bands containing any significant signal power, the J-model is valid.

Figure 10-9



Why is the J-model limited to direct conversion transmitters? Consider a double-conversion transmitter, one with an IF stage. With an IF stage, it might be possible for the output signal to contain power at some linear combination of the two local oscillator frequencies that falls near the output RF signal. Those frequencies are not displaced from the target RF carrier by some multiple of the input baseband frequencies. [Figure 10](#) below shows the problem. If you can ignore terms such as those in the figure below, you can use the J-model for a transmitter with more than one frequency translation.

Figure 10-10



J-model data files and data format

The J-model data files are listed below:

```
imag_minus1.ascsig  
imag_minus2.ascsig  
imag_minus3.ascsig  
imag_minus4.ascsig  
imag_minus5.ascsig  
imag_minus6.ascsig  
imag_minus7.ascsig  
imag_minus8.ascsig  
imag_minus9.ascsig  
imag_plus1.ascsig  
imag_plus2.ascsig  
imag_plus3.ascsig  
imag_plus4.ascsig  
imag_plus5.ascsig  
imag_plus6.ascsig
```

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

```
imag_plus7.ascsig
imag_plus8.ascsig
imag_plus9.ascsig
imag_zero.ascsig
real_minus1.ascsig
real_minus2.ascsig
real_minus3.ascsig
real_minus4.ascsig
real_minus5.ascsig
real_minus6.ascsig
real_minus7.ascsig
real_minus8.ascsig
real_minus9.ascsig
real_plus1.ascsig
real_plus2.ascsig
real_plus3.ascsig
real_plus4.ascsig
real_plus5.ascsig
real_plus6.ascsig
real_plus7.ascsig
real_plus8.ascsig
real_plus9.ascsig
real_zero.ascsig
```

The first word in the file name, up to the underscore, tells you whether the data file contains the real or imaginary part of the Fourier coefficient. The second part, following the underscore, specifies the signed harmonic number. The suffix identifies an ascii file for Alta SPW.

The format of all file is the same and is shown below. The format is ascii. Each file contains exactly three header lines and no blank lines at the end. The data is arranged in two columns. The first column is the input signal level in volts. The second column is the real or imaginary part of the baseband Fourier coefficient as described above.

sweep	imag(harmonic)
1.000e-03	-8.475e-07
2.450e-03	-1.366e-05
3.900e-03	-4.850e-05
5.350e-03	1.595e-05

Affirma RF Simulator (SpectreRF) User Guide

Extracting J-Models

6.800e-03	3.690e-05
8.250e-03	-7.153e-04
9.700e-03	-1.055e-04
1.115e-02	-4.324e-05
1.260e-02	-1.431e-04
1.405e-02	-1.143e-03
1.550e-02	-4.352e-03
1.695e-02	-5.443e-04
1.840e-02	1.107e-03
1.985e-02	1.913e-03
2.130e-02	3.774e-02
2.275e-02	8.256e-02
2.420e-02	1.277e-01
2.565e-02	1.680e-01
2.710e-02	2.114e-01
2.855e-02	2.474e-01
3.000e-02	2.841e-01

References

- [1] E. Ngoya and R. Larcheveque. Envelope transient analysis: a new method for the transient and steady state analysis of microwave communication circuits and systems. *IEEE MTT Symposium Digest*, pp. 1365-1368, June 1996.
- [2] M. Jeruchim, P. Balaban and K. Shanmugan, *Simulation of Communication Systems*, Plenum, 1992.
- [3] M. Schetzen, "*The Volterra & Wiener Theories of Non-Linear Systems*", Krieger Publishing Company, 1980.

Modeling Transmitters

This chapter tells you how to use several of the new SpectreRF features to design transmitters. It particularly emphasizes

- Running an Envelope Following analysis
- Measuring Load Pull Contours and Reflection Coefficients
- Using S-parameter Input Files

The Envelope Following example explains

- How to interpret the analysis results
- How to quickly detect distortion visually
- How to compute the transmitted power spectral density. ACPR (Adjacent Channel Power Ratio) is the number that characterizes transmitter power spectral densities.

The Load-Pull example explains how to select an optimal power amplifier load and determine whether the input matching network needs to be re-designed for the optimal load.

The S-parameter example explains how to generate tabulated S-parameters and include them as input in a SpectreRF analysis.

Envelope Following

This example tells you how to set up and run the Envelope Following analysis then explains why the *time* results look somewhat unusual.

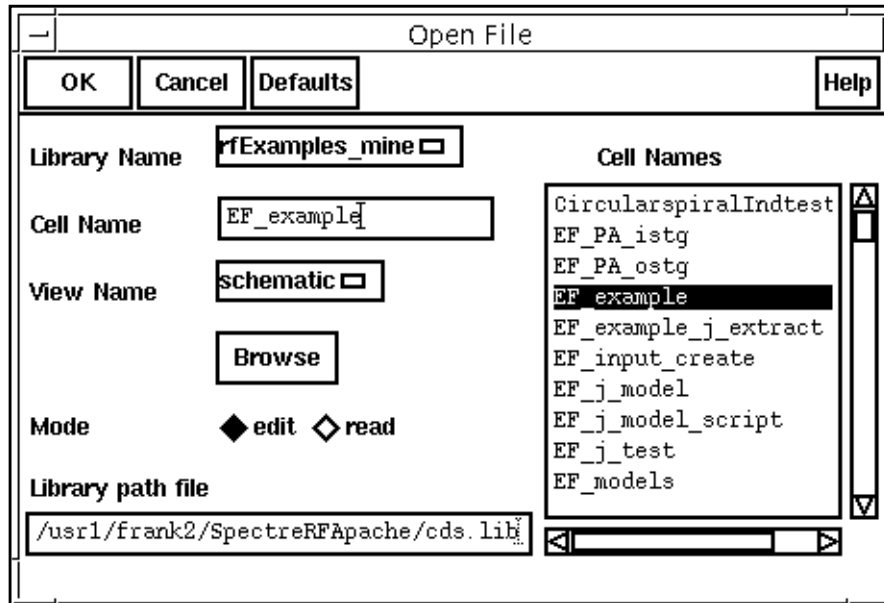
Before you start, perform the setup procedures described in [Chapter 3](#).

1. In the CIW, choose *File – Open....*

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The Open File form appears.



2. In the Open File form, do the following:

a. Choose *rfExamples* for *Library Name*.

Select the editable copy of the *rfExamples* library you created following the instructions in [Chapter 3](#).

b. In the *Cell Name* list box, highlight *EF_example*.

c. Choose *schematic* for *View Name*.

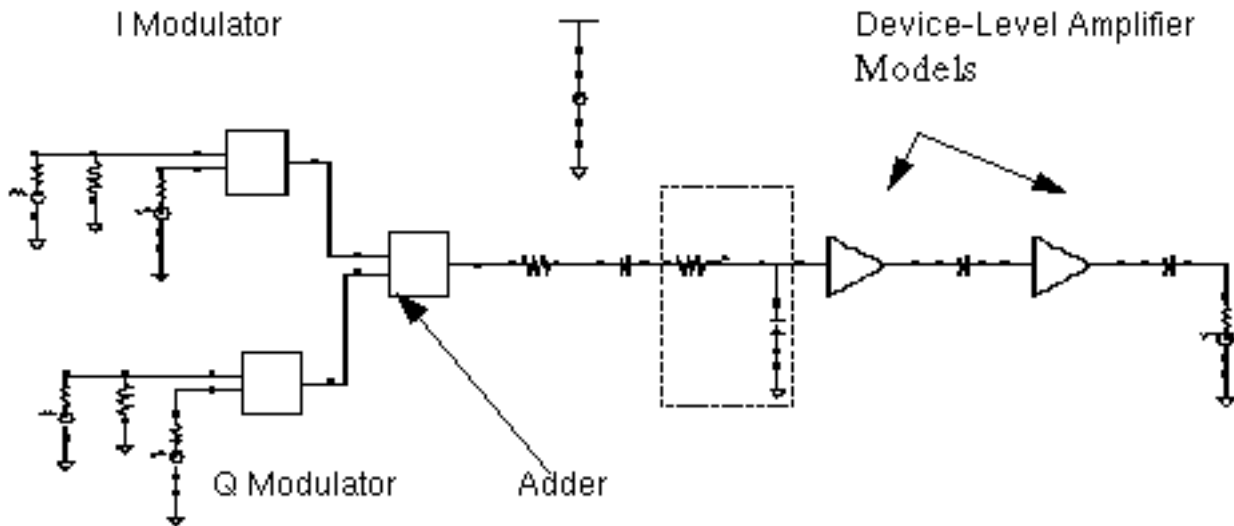
d. Highlight *edit* for *Mode*.

e. Click *OK*.

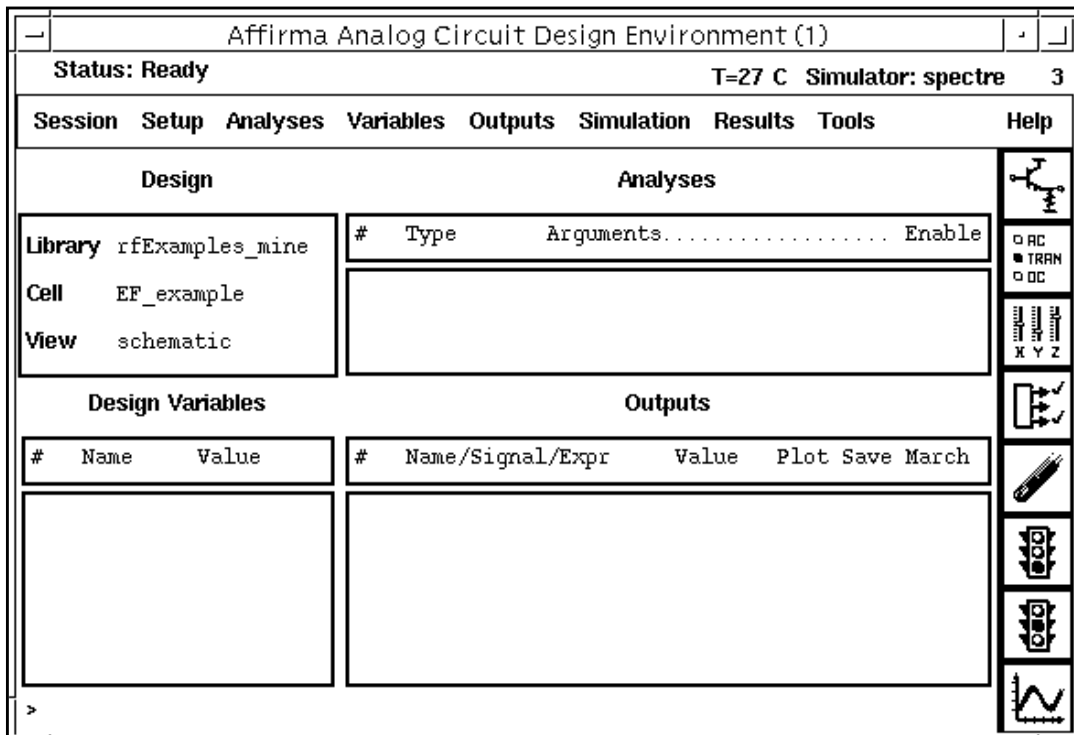
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The Schematic window appears with the *EF_example* schematic. This is a simple direct-conversion transmitter with ideal I/Q modulators.



- In the Schematic window, choose *Tools – Analog Environment*.
The Simulation window appears.



Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

4. In the Simulation window, choose *Setup - Model Libraries*.

The *Model Library Setup* form appears.

5. In the *Model Library File* field, type the following path, where *CDSHOME* is the installation directory for the Cadence software.

<CDSHOME>/tools/dfII/samples/artist/rfExamples/EF_models/npnStd.m

6. In the *Model Library Setup* form, first click on *Add*, then click on OK.

7. In the Schematic window, select PORT0. (The port in the top, left corner of the schematic.)

8. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears.

9. In the Edit Object Properties form, edit the *PWL filename* of PORT0. Type the following path, where *CDSHOME* is the installation directory for the Cadence software. Then click on *Apply*.

<CDSHOME>/tools/dfII/samples/artist/rfLib/cdma_2ms_idata.pwl

10. In the Schematic window, select PORT1. (The port in the bottom, left corner of the schematic.)

The Edit Object Properties form changes to display information for PORT1.

11. In the Edit Object Properties form, edit the *PWL filename* of PORT1. Type the following path, where *CDSHOME* is the installation directory for the Cadence software. Then click on *Apply*.

<CDSHOME>/tools/dfII/samples/artist/rfLib/cdma_2ms_qdata.pwl

12. In the Edit Object Properties form, click on OK.

13. In the Simulation window, choose *Analyses – Choose....*

The Choosing Analyses form appears.

14. In the Choosing Analyses form, highlight *envlp*.

The Choosing Analyses form changes to let you specify values for the Envelope Following analysis.

15. In the Choosing Analyses form, do the following and then click OK.

- a. Select `fff` for *Clock Name*.

- b. Type `300u` for *Stop Time*.

- c. Choose *Number of harmonics* for *Output Harmonics* and type 1 in the adjacent field.

The completed form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

Envelope Following Analysis

Clock Name

Stop Time

Output Harmonics

Accuracy Defaults (errpreset)
 conservative moderate liberal

Enabled

16. Check and save the schematic, then In the Simulation window, choose *Simulation – Netlist and Run*.

Look in the CIW for a message that says the simulation completes successfully.

Taking a Closer Look at the Envelope Following Procedure

1. In the Simulation window, choose *Results – Direct Plot – ENVLP....*
 The Envelope Following Results form appears.
2. In the Envelope Following Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.

b. Highlight *Voltage* for *Function*.

c. Highlight *time* for *Modifier*.

The completed form looks like this.

Envelope Following Results

OK Cancel Help

Plot Mode Append Replace

Function

Voltage Current Power

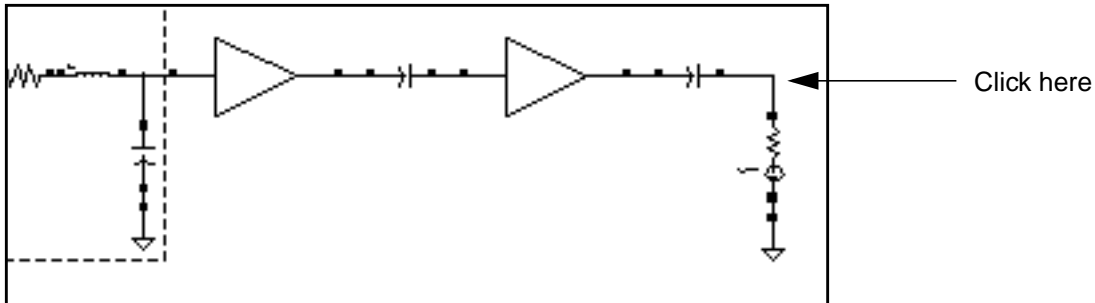
Description: Envelope Voltage vs Time

Sweep spectrum harmonic time
 time

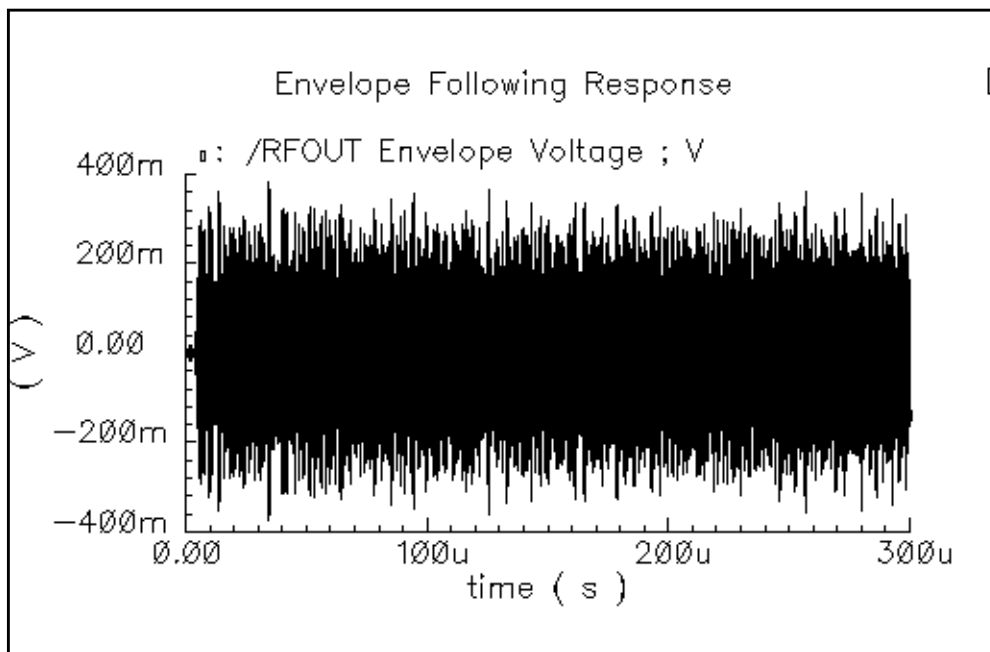
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

3. In the Schematic window, click on the net shown below.



The voltage waveform appears in the Waveform window.



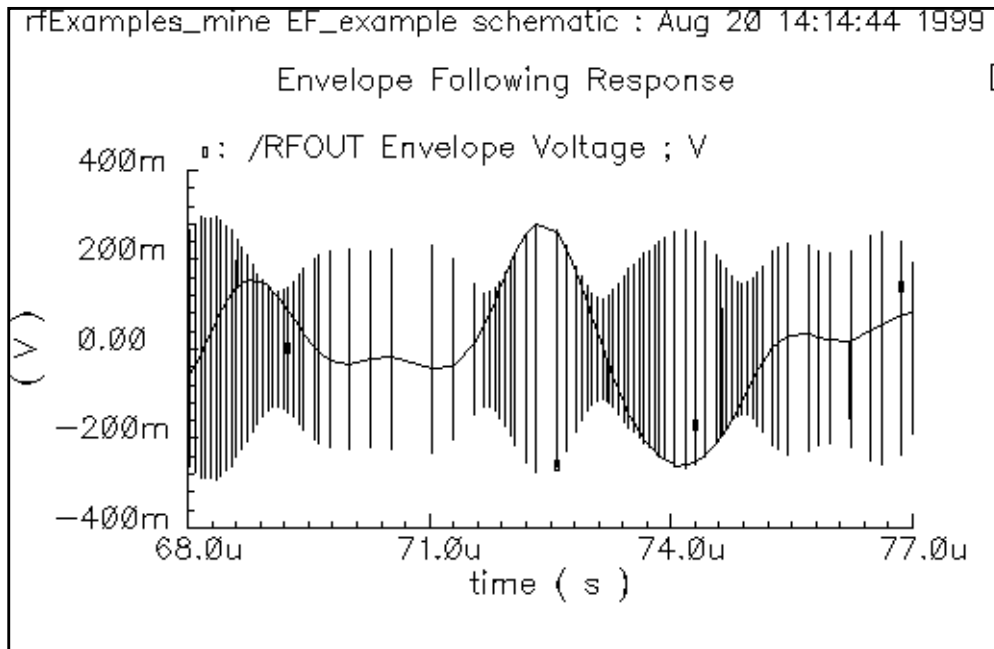
4. In the Waveform window, right click and drag to select a narrow section of the waveform. (The area you select covers a narrow vertical rectangle.)

The plot changes to display a number of vertical lines with a wavy line running through them. The vertical lines are the points at which detailed calculations are performed and the wavy line connects these points. The simulation runs much faster than a Spectre

Affirma RF Simulator (SpectreRF) User Guide

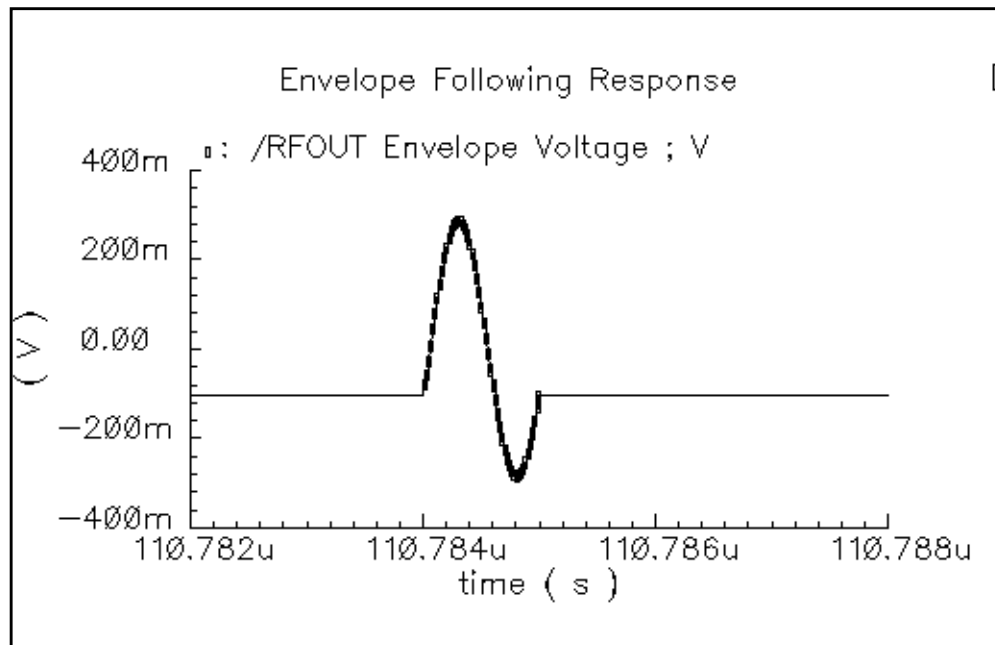
Modeling Transmitters

Transient simulation because Envelope Following skips carrier cycles when it can do so and still satisfy numerical tolerances.



5. To get a closer look, right click and drag to select one of the vertical lines. You will probably have to do this several times.

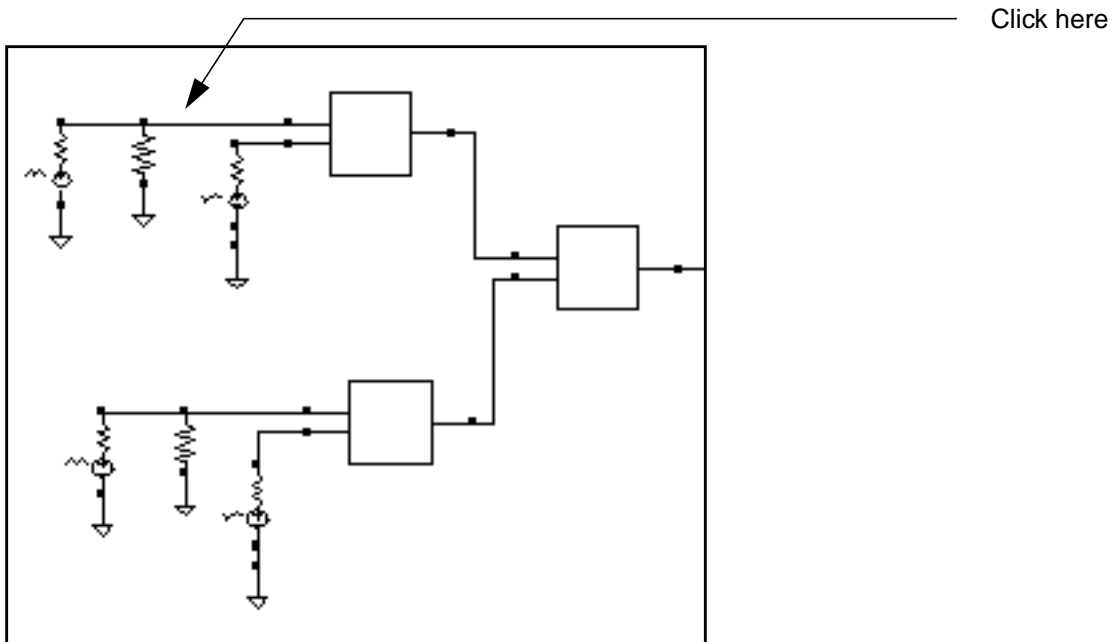
You can now see the detailed simulation for one complete cycle.



Following the Change in the Information (Baseband Signal) Through the Circuit

The modulation riding on the RF carrier is the baseband signal, the information to be transmitted. The baseband signal determines the amplitude and phase of the RF carrier. It is important to determine how the transmitter might alter the baseband signal. This section tells you how to extract the baseband signal at any point in the design.

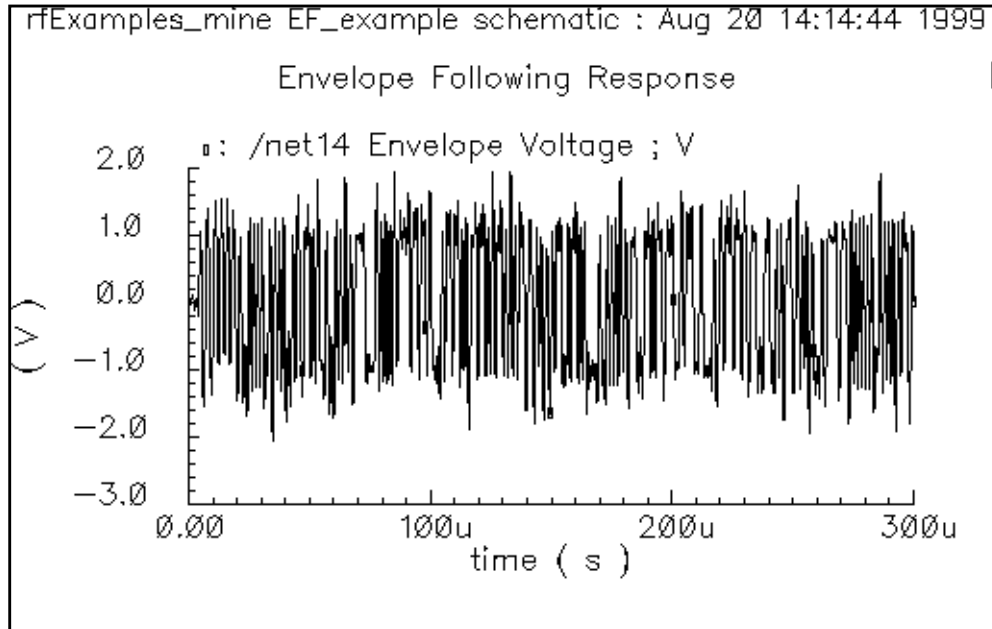
1. In the Envelope Following Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.
 - b. Highlight *Voltage* for *Function*.
 - c. Highlight *time* for *Sweep*.
2. In the Schematic window, click on the modulation source.



Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The plot for the modulation wave appears. You may have to click *Zoom – Fit*.

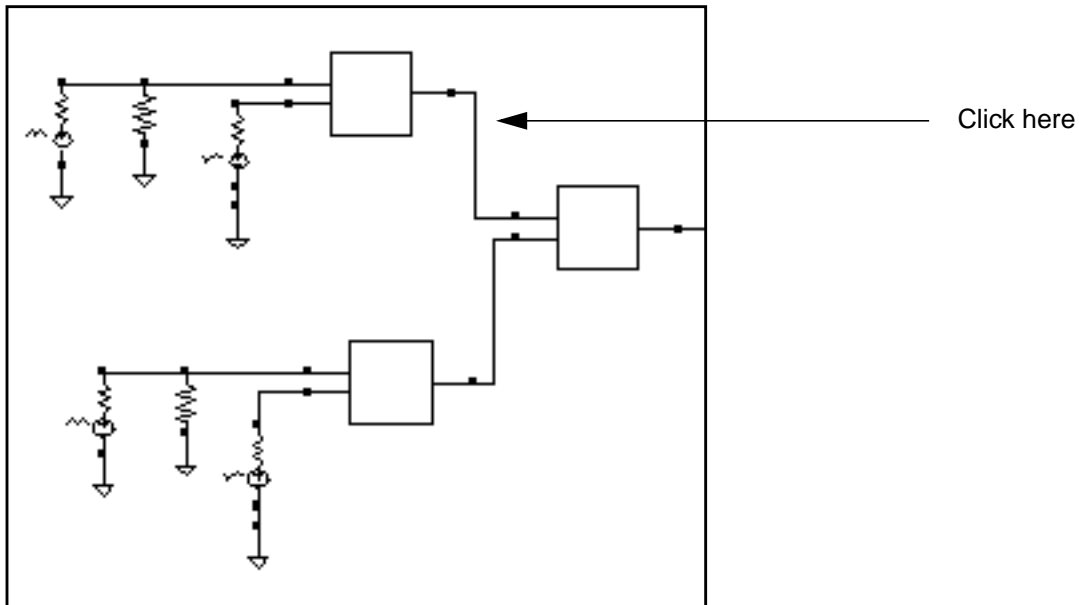


3. In the Envelope Following Results form, highlight *Append for Plot Mode*.
4. In the Waveform window, choose *Axes – To Strip*.
5. In the Schematic window, click on the I-modulator output. This is the “in-phase” carrier modulated by the I-component of the baseband signal. In this example, the in-phase

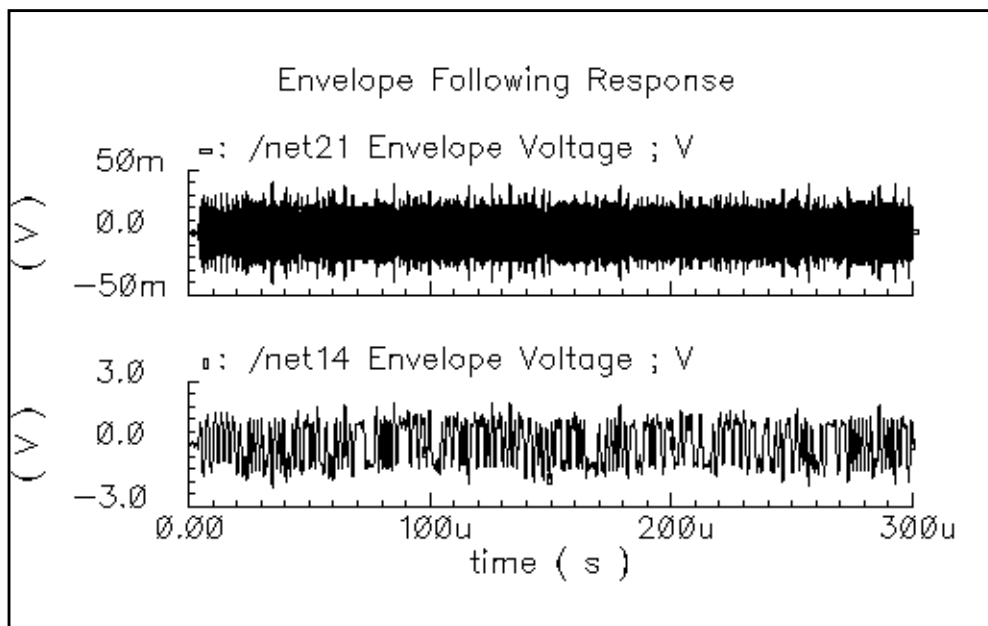
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

carrier is $\cos(\omega_c t)$, where ω_c is the carrier frequency in radians per second. The quadrature carrier is $\sin(\omega_c t)$.



The output wave is added to the waveform window.



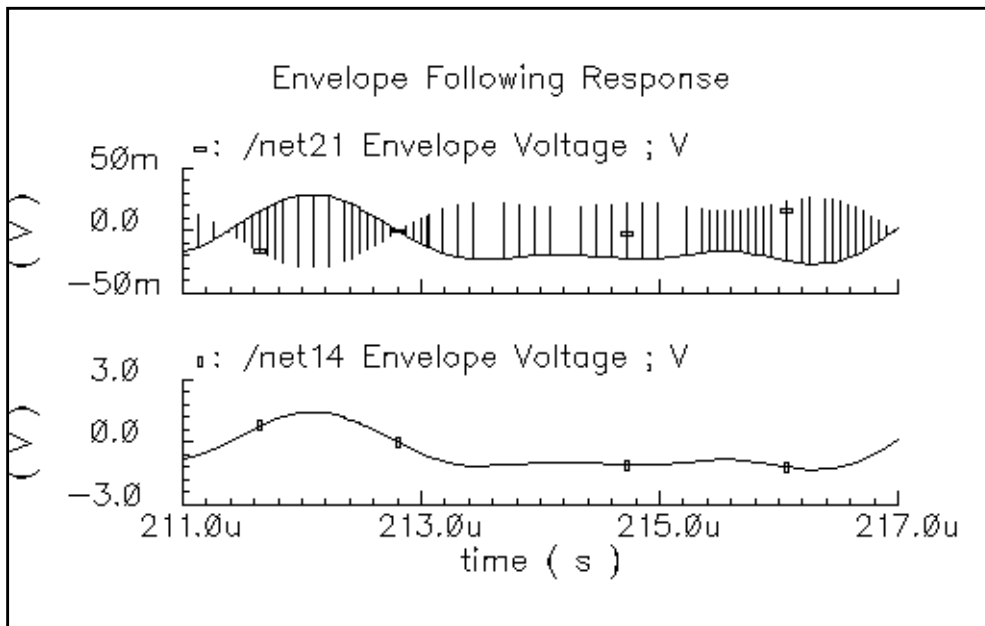
6. In the Waveform window, right click and drag over a narrow section of both waveforms.

You can see the modulation wave within the output wave. It is only coincidence that the bottom waveform and smooth curve in the top waveform look alike. If the individual cycles

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

in the top waveform were sampled at a different phase, the smooth curve in the top waveform might look different.



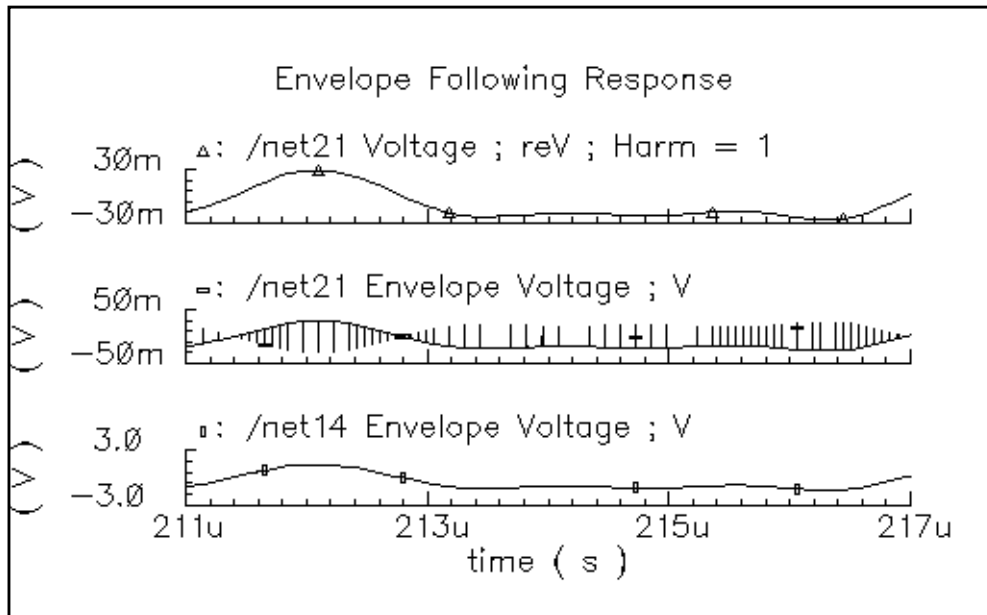
Note: The appearance of the graph for this plot and subsequent plots depends on which section of the Waveform window you perform the click and drag.

7. In the Envelope Following Results form, do the following:
 - a. Highlight *harmonic time* for *Sweep*.
 - b. Highlight *Real* for *Modifier*.
 - c. Highlight *1* for *Harmonic Number*.
8. In the Schematic window, click on the I-modulator output.

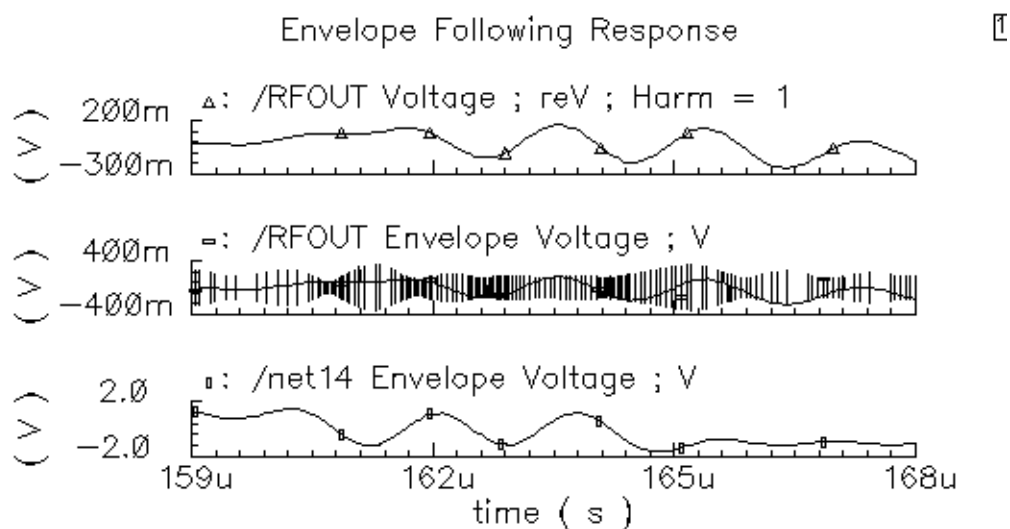
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The top plot in the Waveform window is the baseband waveform recovered from the modulated RF carrier. Aside from a linear scale factor, it looks exactly like the bottom waveform because the modulator is ideal.



To see the change in the signal as it passes through the length of the circuit, which is not ideal, reset the Waveform window and repeat steps 1-8 clicking on the transmitter output instead of the I-modulator output in step 5. The plot below shows a greater change in the signal.



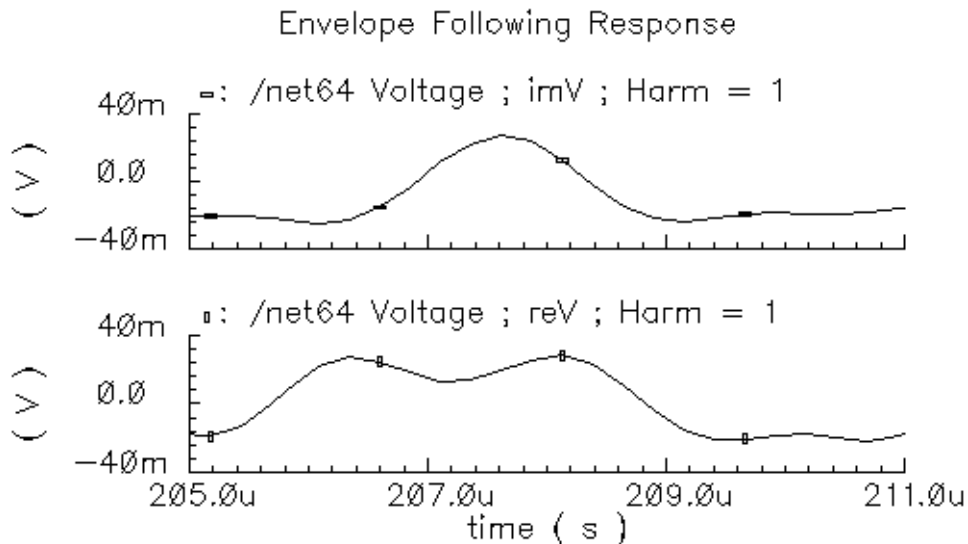
Plotting the Complete Baseband Signal

1. In the Envelope Following Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.
 - b. Highlight *Voltage* for *Function*.
 - c. Highlight *harmonic time* for *Sweep*.
 - d. Highlight *Real* for *Modifier*.
 - e. Highlight 1 for *Harmonic Number*.
2. In the Schematic window, click on the adder output.

A plot for the real portion appears in the Waveform window.
3. In the Envelope Following Results form, do the following:
 - a. Highlight *Append* for *Plot Mode*.
 - b. Highlight *Imaginary* for *Modifier*.
4. In the Schematic window, click on the adder output.

A plot for the imaginary portion is added to the Waveform window.
5. In the Waveform window, right click and drag to select a portion of the Waveform window over a narrow portion of the x-axis.

You can now see both the real and imaginary parts clearly.



Plotting the Baseband Trajectory

The baseband waveforms recovered from the modulated RF carrier, as displayed in the figures above, do not directly reveal much about how the transmitter affects them. The steps below tell you how to display the associated trajectory, the plot of one waveform against the other. The trajectory reveals much more about what kind of distortion the transmitter introduced. The steps below first display the input baseband trajectory and then the output baseband trajectory. A comparison of the two trajectories reveals whether the power amplifiers in this example introduce any phase shift.

1. In the Envelope Following Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.
 - b. Highlight *Voltage* for *Function*.
 - c. Highlight *harmonic time* for *Sweep*.
 - d. Highlight *Real* for *Modifier*.
 - e. Highlight 1 for *Harmonic Number*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed form looks like this.

Envelope Following Results

OK Cancel Help

Plot Mode Append Replace

Function

Voltage Current Power

Description: Harmonic Voltage vs Time

Sweep spectrum harmonic time
 time

Modifier

Magnitude Phase dB20
 Real Imaginary

Harmonic Number

0
1
2
3

Note:
Harmonic 0 is a real waveform; therefore,
the Real Modifier should be used.
All other harmonics are complex waveforms.

Replot

Add To Outputs

> Select net on schematic...

2. In the Schematic window, click on the adder output.
A plot for the real portion appears in the Waveform window.
3. In the Envelope Following Results form, do the following:
 - a. Highlight *Append* for *Plot Mode*.
 - b. Highlight *Imaginary* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

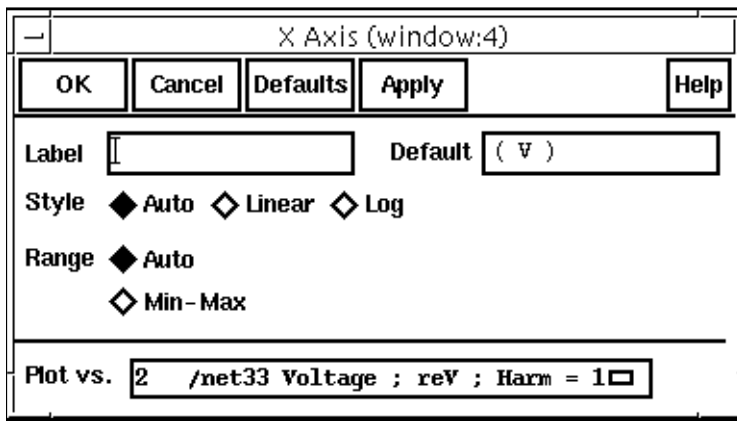
4. In the Schematic window, click on the adder output.

A plot for the imaginary portion is added to the Waveform window.

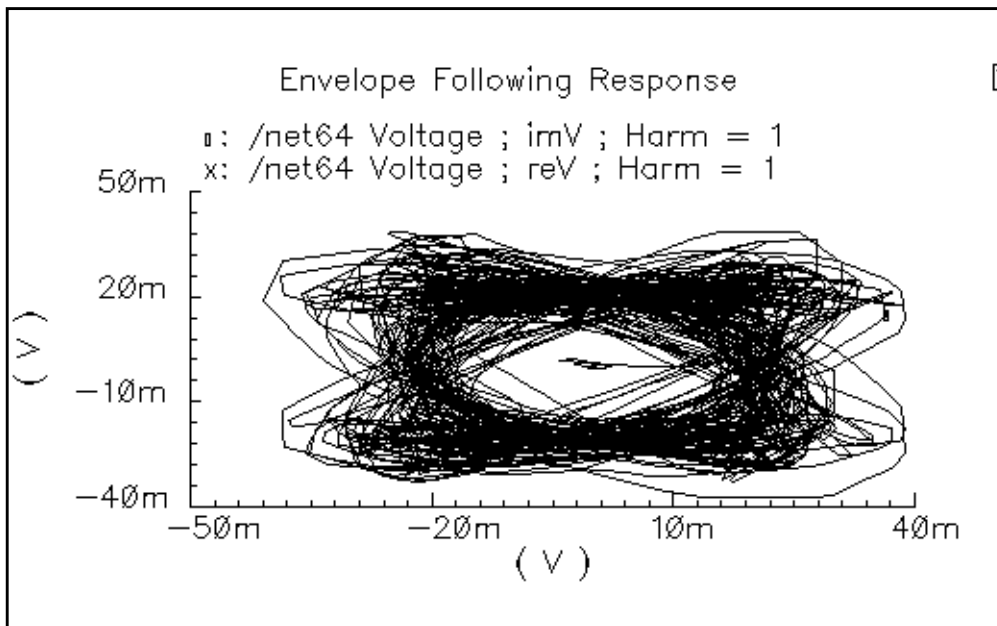
5. In the Waveform window, choose *Axes – X Axis....*

The X Axis form appears.

6. In the X Axis form, choose *Plot vs. /net64 Voltage ; reV ; Harm=1* and click *OK*. (If the net number is different, choose the menu selection that contains *reV*.)



The plot below appears in the Waveform window. This is the input baseband trajectory, undistorted by the power amplifiers.



7. In the Simulation window, choose *Tools – Waveform....*

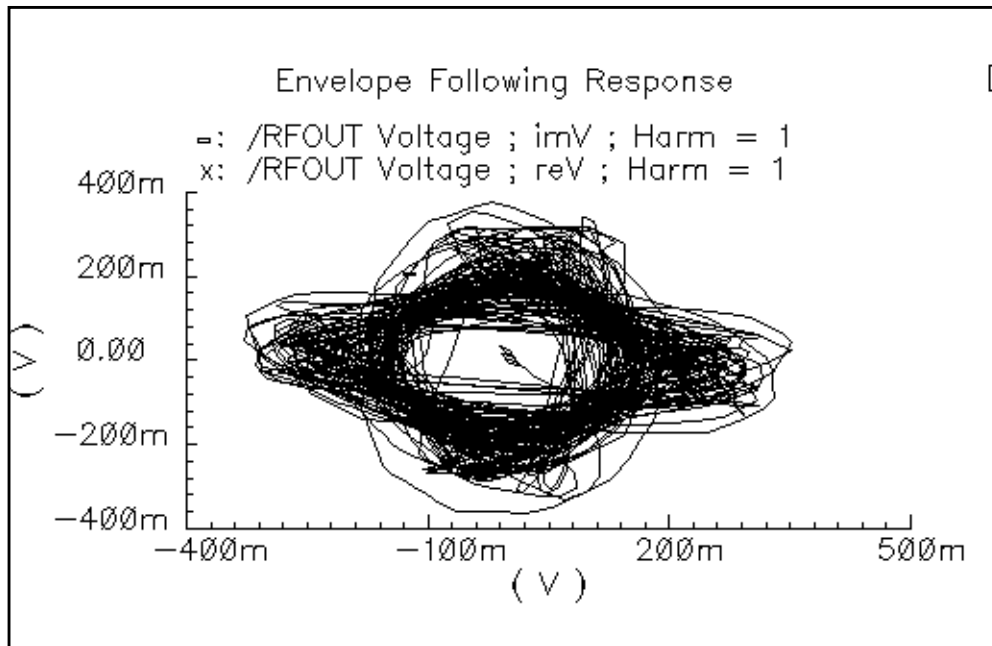
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

A second Waveform window appears.

- Repeat steps 1 through 6 that you used to display the plot for /net 64 but substitute the /RFOUT net for /net64.

The plot you create in the second Waveform window looks like this.



The entire trajectory is scaled linearly and rotated; The output baseband signal is the input baseband signal, multiplied by a complex constant. The input and output *waveforms* look different because of the rotation, not because of some non-linear distortion. A common non-linear distortion, such as saturation, would make the outer edges of the trajectory lie on a circle.

Plotting the Transmitted Power Spectral Density for ACPR Calculations

Adjacent Channel Power Ratio (ACPR) is a common measure of how much power a transmitter emits outside its allotted frequency band. It is the ratio of the power in an adjacent band divided by the power in the allotted band. Regardless of exactly how you chose the frequencies and bands for the ACPR measurement, it is always extracted from the power spectral density (PSD) of the transmitted signal.

Estimating the Power Spectral Density

This section describes how the Analog Artist estimates power spectral density (PSD).

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

PSDs are always estimated because the information riding on the carrier is a stochastic process and the Fourier transform of a stochastic process is ill defined. No matter how you chose to define the spectral nature of a stochastic process, it always involves an averaging process. Any empirically derived average is an estimate because you can never take an infinite number of samples.

The PSD is a frequency-by-frequency average of a set of DFTs (discrete Fourier transforms) of the baseband signal. Here, the baseband signal is the harmonic-time results of an Envelope Following analysis. The parameters that control the PSD estimate described below.

- The *Time Interval* specifies the time record you wish to analyze in the frequency domain. The time interval should be long enough to support the frequency resolution you want.
- The *Nyquist half bandwidth* is half the sampling frequency used in the DFTs. Make this big enough to prevent aliasing; the true spectrum of the baseband signal should be negligible at this frequency and beyond.
- The *Frequency bin width* is the frequency resolution you want. If you chose a small bin width, the resulting PSD will look “noisy”. The PSD will have a jagged appearance. If you make the bin width too large you may soften what should be sharp edges in the spectrum.

The waveform is first interpolated, if necessary, to generate evenly spaced data points in time. The spacing of the data points is the inverse of the DFT sampling frequency. The PSD is computed by first breaking the time interval up into overlapping segments. Each segment is multiplied, time point by time point, by the specified *Windowing* function. Windowing reduces errors caused by a finite time record. It is impossible to work with an infinite time record. Direct use of an unwindowed finite time record is equivalent to multiplying the infinite record by a rectangular pulse that lasts as long as the data record. Multiplication in the time domain corresponds to convolution in the frequency domain. The Fourier transform of a rectangular pulse is a sinc function. Considering the frequency domain convolution, the side lobes of the sinc function cause parts of the true spectrum to “leak” into the frequency of interest, the frequency of the main lobe. Ideally, the sinc function would be a Dirac delta function but that requires an infinite time record. Good window functions have smaller side lobes than a sinc function.

The DFT is performed on each windowed segment of the baseband waveform. At each frequency, the DFTs from all segments are averaged together. Fewer segments means fewer data points in the average at a particular frequency. The length of each segment is inversely proportional to the frequency bin width. That is why a small bin width produces a jagged PSD. A smaller bin width means a longer time segment. Fewer long segments fit into the given time interval so there are fewer DFTs to average together. In the extreme, there is only one segment and no averaging. Without averaging, the PSD is just the square of the magnitude of the DFT of a stochastic process. At the other extreme, large bin widths produce lots of points to average at each frequency but there are fewer frequencies at which to average.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

because fewer large bins fit into the Nyquist frequency. The PSD is smoother but it does not have as much resolution.

The Nyquist half bandwidth, frequency bin width, and time interval are used to calculate the parameters for a Waveform Calculator function, `psdbb`, that performs the actual PSD calculation. The calculated `psdbb` parameters are printed in the CIW window. The `psdbb` parameters are the *total number of samples*, *window size*, and *bin-width*. All three parameters are in terms of the number of DFT time samples. The following list displays the calculations that generate the `psdbb` parameters:

- $L = T_o - \text{From}$ (where T_o and From times are user inputs)
- f_{max} (Nyquist half-bandwidth) and *bin-width* are user inputs
- compute $\# \text{bins} = \text{floor}(L * \text{binwidth})$, with $\# \text{bins} \geq 1$. Here, “floor” means to *take the integer part of...*, i.e. truncate to the nearest integer
- compute the smallest m such that $2^m * (\# \text{bins}) > 2 * L * f_{\text{max}}$
- $\text{window size} = 2^m$
- $\text{number of samples} = \# \text{bins} * \text{window size}$

You may want to use the `psdbb` function directly when strobing time-harmonic results to eliminate interpolation error. Use of the `psdbb` function is described in the waveform calculator section.

The example below shows how to generate the power spectral density estimate. Continuing on with the Envelope Following example:

1. In the Envelope Following Results form, do the following:

- a. Highlight *Replace* for *Plot Mode*.
- b. Highlight *Voltage* for *Function*.
- c. Highlight *spectrum* for *Sweep*.
- d. Highlight *Magnitude* for *Modifier*.
- e. Highlight 1 for *Harmonic Number*.
- f. Type 5M for *Nyquist half-bandwidth*.
- g. Type .1M for *Frequency bin width*.
- h. Type 3M for *Max. plotting frequency*.
- i. Type -3M for *Min. plotting frequency*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

j. Highlight *dB10* for *Modifier*.

The completed form looks like the figure below.

Envelope Following Results

OK Cancel Help

Plot Mode Append Replace

Function

Voltage Current Power

Description: Harmonic Voltage Spectrum

Sweep spectrum harmonic time
 time

Modifier

Magnitude dB10 dBm

Harmonic Number

0
1
2
3

Power Spectral Density Parameters

Time Interval

From 0 To 0.0003

Nyquist half-bandwidth 5M

Frequency bin width 1M

Max. plotting frequency 3M

Min. plotting frequency -3M

Windowing Hanning

Detrending None

Replot

Add To Outputs

> Select net on schematic...

2. In the Schematic window, click on the RF output net.

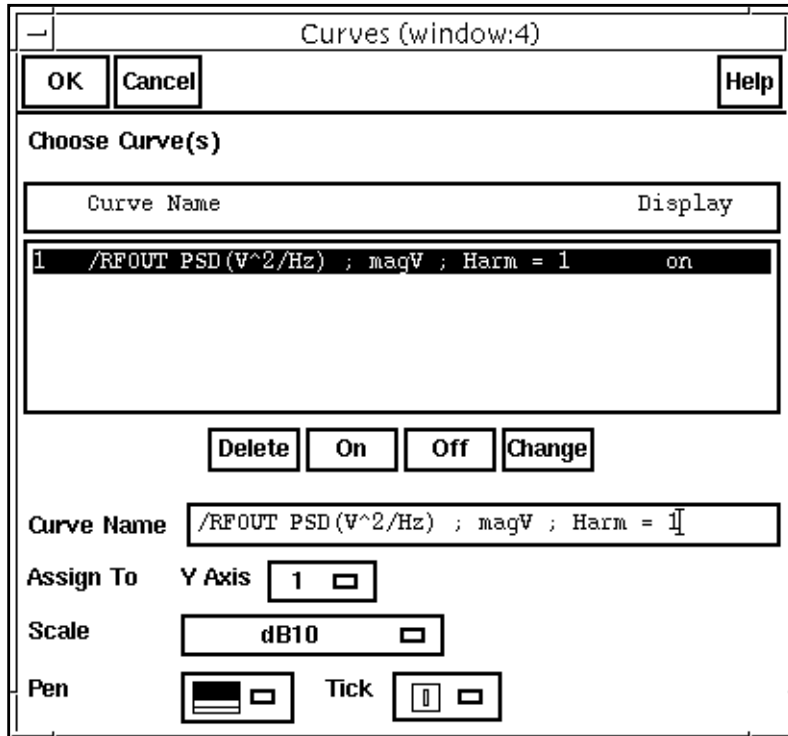
A plot appears in the Waveform window.

3. In the Waveform window, select *Curves – Edit....*

Affirma RF Simulator (SpectreRF) User Guide

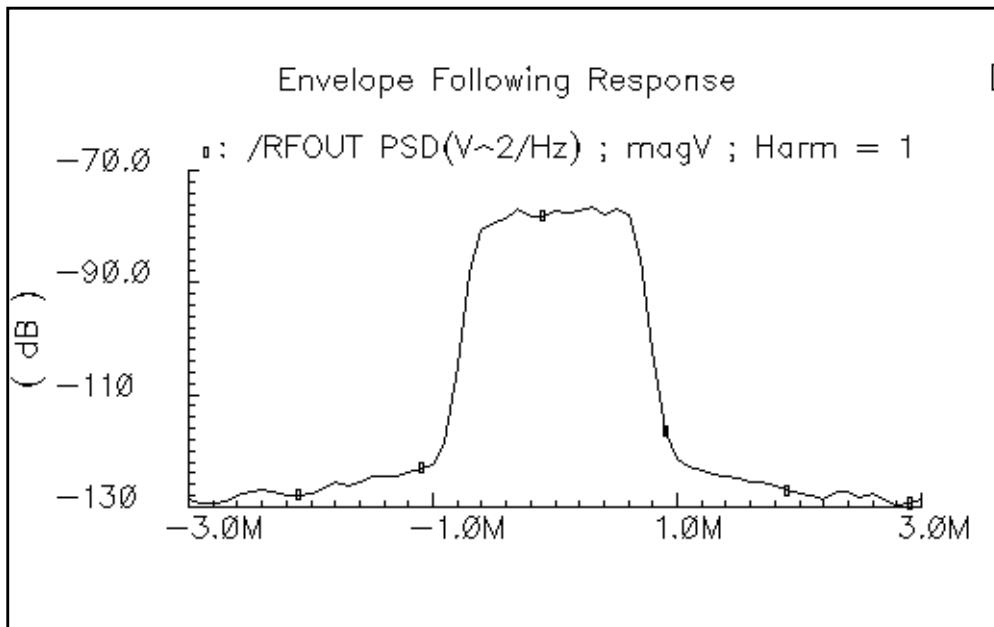
Modeling Transmitters

The Curves window appears.



4. In the Curves window, choose *dB10* for *Scale* and click *OK*.

The plot in the Waveform window changes to reflect the modification you specified.



Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

You can now calculate the ACPR for two x-axis values by subtracting their associated y-axis values. (The ACPR measured with respect to x_1 and x_2 is $y_1 - y_2$.)

For example, given the following spectral parameters, the adjacent channel is 2.5 MHz from the carrier and you choose to define ACPR in terms of power at just two frequencies, you can use the cursor to determine the RF output power at 2.5 MHz and zero frequency.

The power at 2.5 MHz equals -119.7 dB and the power near zero (at 18.56 KHz) equals -77.48 dB. (Remember that the horizontal scale is frequency offset from the carrier fundamental.) The difference between these measurements equals -42.22 dB. For this definition of ACPR, $ACPR = -42.22$ dB.

Other definitions of ACPR are possible. You can compute them by adjusting the spectral parameters and applying the waveform calculator to the spectral plot.

Calculation of Power Spectral Density

Power spectral density results are calculated in the following manner. For a constant baseband signal, and

Baseband signal = $i + j*q$, where $j = \text{sqrt}(-1)$

Passband signal = $i*\cos(w*t) - q*\sin(w*t)$

The power of the baseband signal is

Baseband signal power = $i*i + q*q$ volts*volts

The power of the passband signal is

Passband signal power = $i*i/2 + q*q/2 = (1/2)*(baseband\ power)$

Envelope Spectral analysis computes these values per Hz. They can also be expressed as

$(\text{rms passband volts}) * (\text{rms passband volts}) / \text{Hz}$

The Envelope Time-Harmonic analysis computes peak volts because they can be directly compared with the modulating signals.

For the example, you see a waveform displayed in $V^2/(Hz)$ versus frequency. You can think of this as $(\text{rms passband volts}) * (\text{rms passband volts}) / \text{Hz}$. The displayed spectrum is the estimated power spectral density of the complex envelope, divided by two. The division by two is there because the envelope is in units of peak carrier volts but power in the carrier equals the square of the peak, divided by two. It is convenient to express the envelope in peak units because it can then be directly compared against an input baseband signal.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

PSD and the Baseband Signal to be Transmitted

The baseband signal driving the transmitter dominates the transmitted PSD. In most cases, the baseband signals come from digital filters. The digital filters constrain the spectrum of the input baseband signal. Distortion in the transmitter causes the spectrum to grow where it should not, hence the need for an ACPR measurement. In measuring ACPR it is crucial to drive the transmitter with the proper baseband signals.

It is not practical to model digital filters in SpectreRF because SpectreRF cannot simulate state variables inside Verilog-A modules. Consequently, for now, you must pre-compute and store the baseband inputs. You then read them into the SpectreRF analysis through the *ppwlf* sources found in the *analogLib*, as shown in this example. The *ppwlf* sources also read SPW format, so you can also generate and record the input baseband waveforms using SPW.

The *rfLib* contains three sets of stored baseband waveforms, *cdma*, *dqpsk*, and *gsm*. These waveforms were created with the baseband signal generators in the *measurement* category of the *rfLib*.

If you want to measure ACPR with the noise floor much more than 40dB below the peak of the output power spectral density, you will have to create baseband drive signals with a noise floor at or below the required noise floor. If you use a DSP tool such as SPW to create the signals, the filters in the baseband signal generator will have to operate perhaps hundreds of times faster than those in the actual generator.

Note: Sometimes an ACPR specification exceeds the ACPR of the baseband drive signals. To see if the transmitter meets specifications in that event, it must be driven with an unrealistic baseband signal.

Otherwise the signals will not have enough resolution. The noise floor depends heavily on interpolation error.

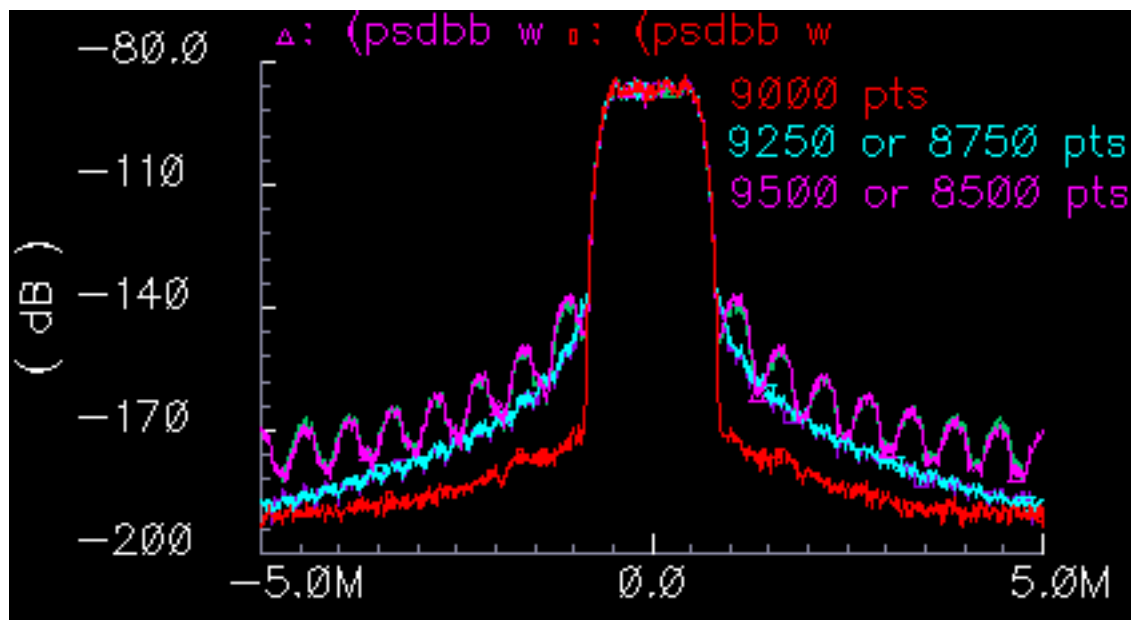
Note: The Fourier analysis used to compute the power spectral density uses evenly spaced time points. If data does not exist at one of the Fourier time points, the Fourier algorithm must interpolate to create one.

You can strobe the harmonic time results to eliminate interpolation of the output but you cannot eliminate interpolation of the baseband drive signals. The only way to reduce interpolation errors at the input is to use ultra-high-resolution drive signals so that no matter where the interpolation occurs, the error is small. For now, it is up to you to generate ultra-high-resolution drive signals.

The following results were obtained by generating high resolution baseband signals in SPW, storing them, then reading them into the Envelope Following simulation through the *pwl* sources. The FIR filters inside SPW operated at 300 times the chip rate.

Figure 11-1 on page 502 shows the spectrum at the output of the adder, which is where the RF signal is undistorted in this model, computed with three different “number of samples”. The simulation was run with `reltol` set to $1e-5$ and the Envelope Following analysis option called “`strobeperiod`” set to 100ns. **If `strobeperiod` does not equal an integer number of clock cycles, it is internally truncated to an integer number of clock cycles.** The time interval for the `psdbb` function was from 100us to 1ms. The window size¹ was 1024 samples and the window function was “Hanning”. With `strobeperiod` set to 100ns (100 clock cycles), the simulation produced 9000 evenly spaced samples inside the time interval. The spectra were calculated for 8500, 8750, 9000, 9250, and 9500 points for the “number of samples”. The figure clearly shows interpolation errors when the number of samples used in the `psdbb` function does not equal the actual number of samples.

Figure 11-1 Spectrum at the Output Adder Computed with Three Different Number of Samples



The figure below compares the input and output PSDs using the high resolution drive signals and the following set of `psdbb` parameters:

- `From=100u`
- `To=1m`

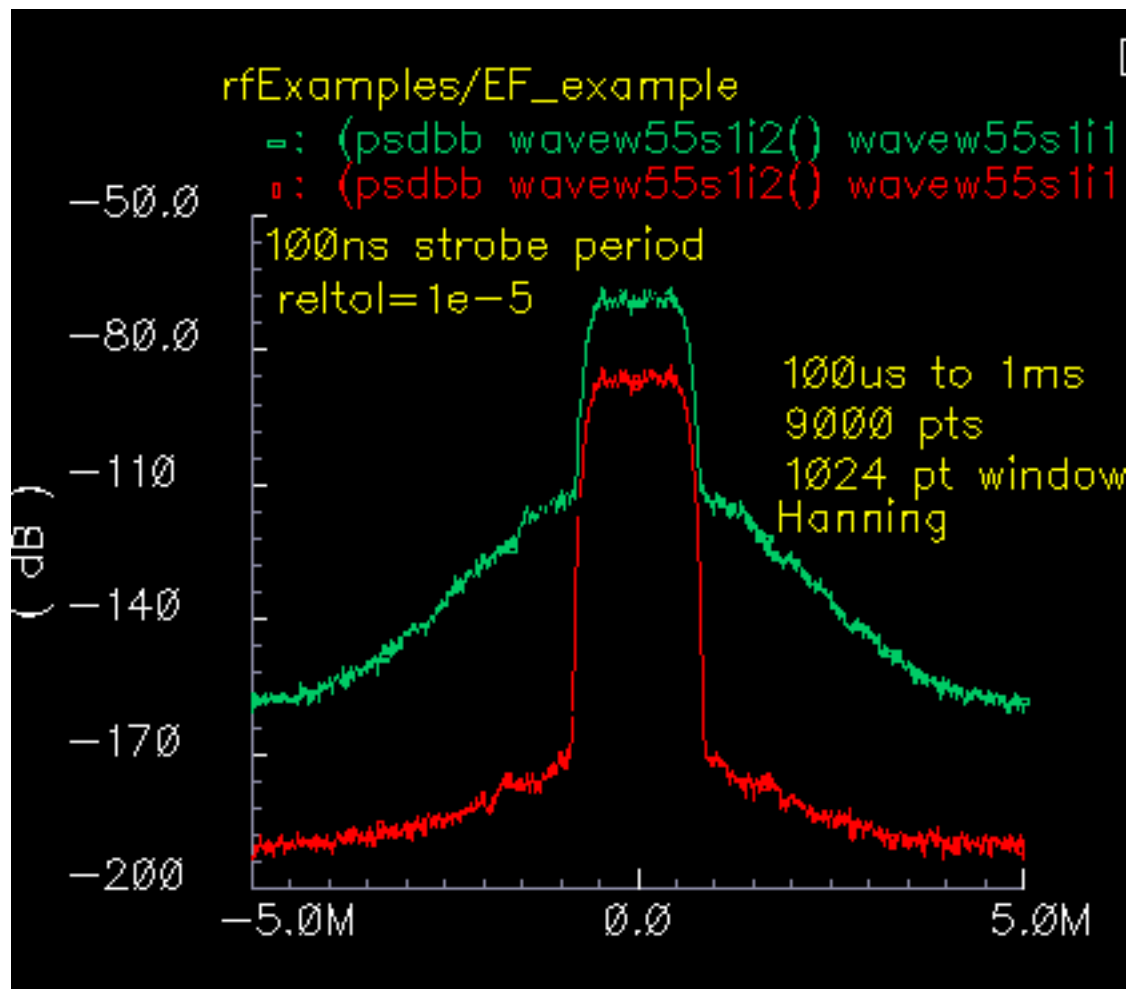
1. “Window size” must be a power of 2. If you enter something that is not a power of 2, the `psdbb` function will truncate to the nearest power of 2.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

- Number of Samples=9000
- Window size=1024
- Window Type=Hanning

The top PSD is from the power amplifier output and clearly shows spectral regrowth when compared to the input PSD.



Measuring Load Pull Contours and Reflection Coefficients

A load pull contour is a set of points on a Smith chart representing all the loads that dissipate a given amount of power. Just as a topographical map shows a hiker where the mountain peak lies and how steep the climb is, load pull contours show which load dissipates the most power and how sensitive that power is to small load perturbations. Load pull contours help you match a load to a power amplifier for maximum power transfer. This section describes how to generate load pull contours and determine whether the input matching network must be redesigned for the optimal load.

To properly use Load-Pull results, you must understand how the Load-Pull feature defines reflection coefficients. Reflection coefficients are computed from the SpectreRF PSS analysis. The load reflection coefficient is computed from the load impedance, which is computed as the ratio of the voltage across the load to the current flowing into the load at the RF carrier frequency. Suppose the RF carrier is 1GHz and the load waveforms are distorted. The impedance is not computed from small-signal perturbations about an operating point. Rather, it is computed as the ratio of the 1GHz components of the load voltage and current waveforms simulated by a PSS run. Because the load is passive and linear, the load reflection coefficient computed in this manner equals the small-signal, or incrementally computed, load impedance.

This is not necessarily true for the input reflection coefficient because the input circuitry can be non-linear and also contain input offset voltages and currents. However, because matching networks are usually designed only for the RF fundamental, defining the reflection coefficient as the ratio of fundamental Fourier components of the large signals is often justified.

Setting Up and Running the Simulation

This example tells you how to modify the *EF_example* schematic and then how to measure Load Pull Contours and Reflection Coefficients for the modified design.

Before you start, perform the setup procedures described in [Chapter 3](#).

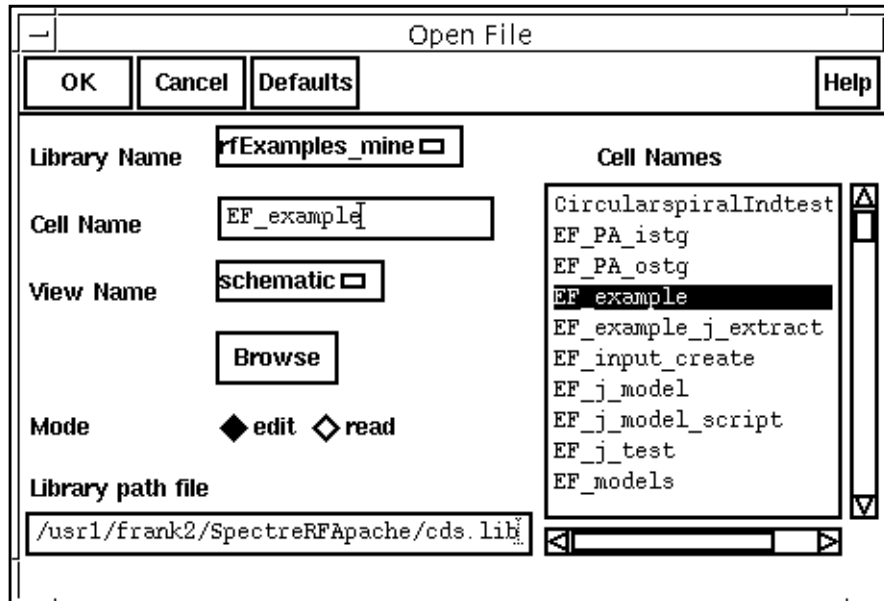
Opening the EF_examples Schematic

1. In the *CIW*, choose *File – Open....*

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The Open File form appears.



2. In the Open File form, do the following:

a. Choose *rfExamples* for *Library Name*.

Select the editable copy of the *rfExamples* library you created following the instructions in [Chapter 3](#).

b. In the *Cell Name* list box, highlight *EF_example*.

c. Choose *schematic* for *View Name*.

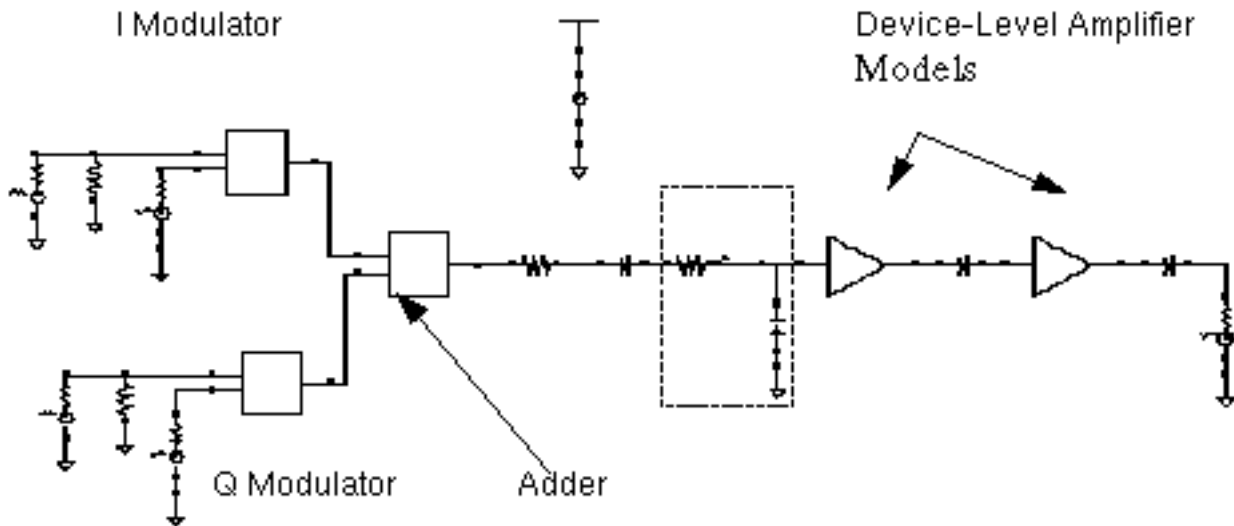
d. Highlight *edit* for *Mode*.

e. Click *OK*.

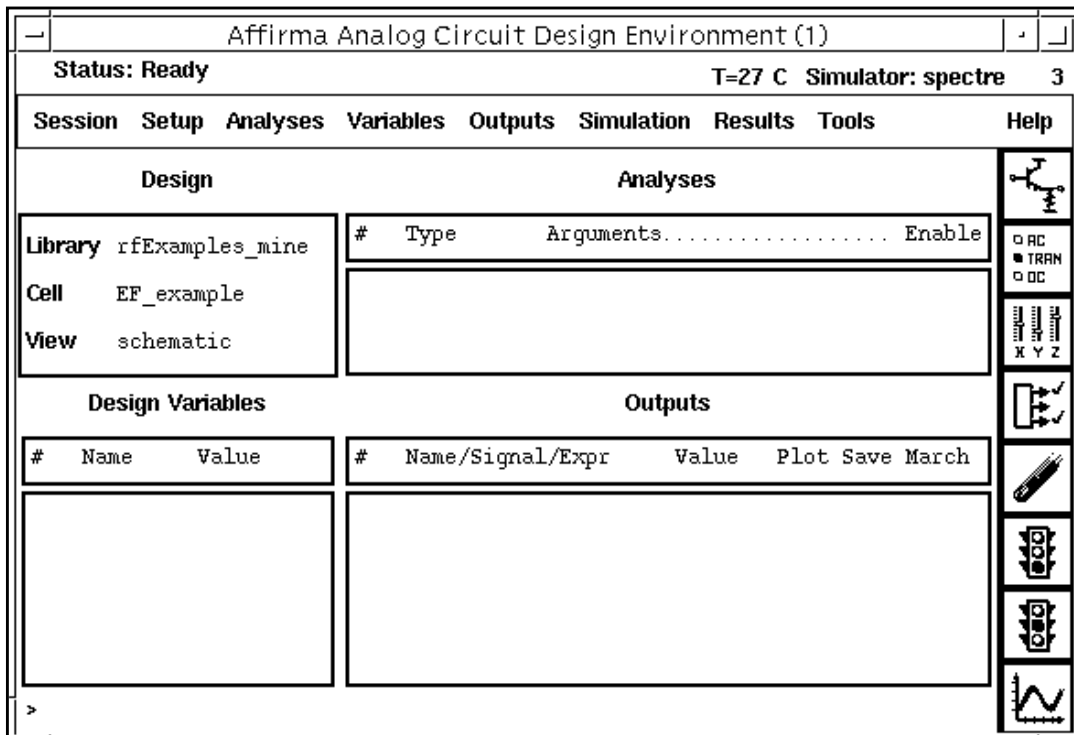
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The Schematic window appears with the *EF_example* schematic. This is a simple direct-conversion transmitter with ideal I/Q modulators.



- In the Schematic window, choose *Tools – Analog Environment*.
The Simulation window appears.



4. In the Simulation window, choose *Setup - Model Libraries*.

The *Model Library Setup* form appears.

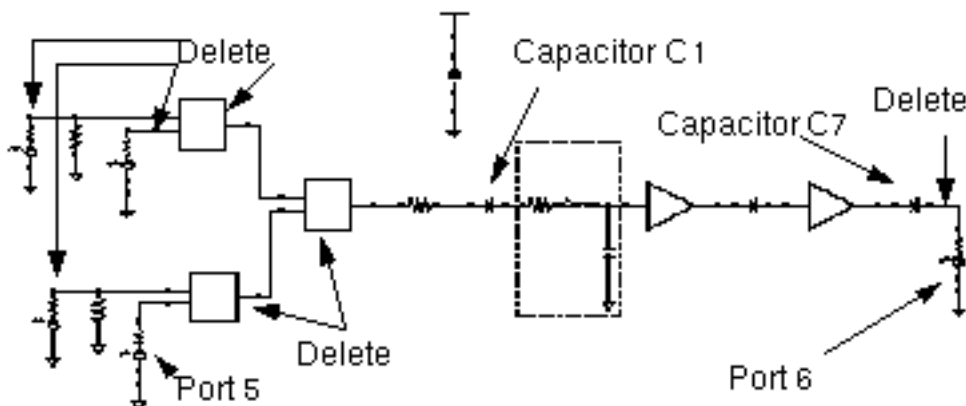
5. In the *Model Library File* field, type the following path, where *CDSHOME* is the installation directory for the Cadence software.

```
<CDSHOME>/tools/dfII/samples/artist/rfExamples/EF_models/  
nnpStd.m
```

6. In the *Model Library Setup* form, first click on *Add*, then click on OK.

Editing the Schematic

1. In the Schematic window, delete components and their associated connecting wires as shown in the schematic below.



Note: If you need assistance with methods for editing the schematic, see the [Virtuoso® Schematic Composer™ User Guide](#).

2. Connect Port 5 to Capacitor C1.
3. In the Schematic window, choose *Add - Instance*.
The Add Instance form appears.
4. In the Add Instance form, click on *Browse*.
The Library Browser - Add Instance form appears.
5. In the Library Browser - Add Instance form, do the following and then click *Close*:
 - a. Highlight *rfExamples* for Library.

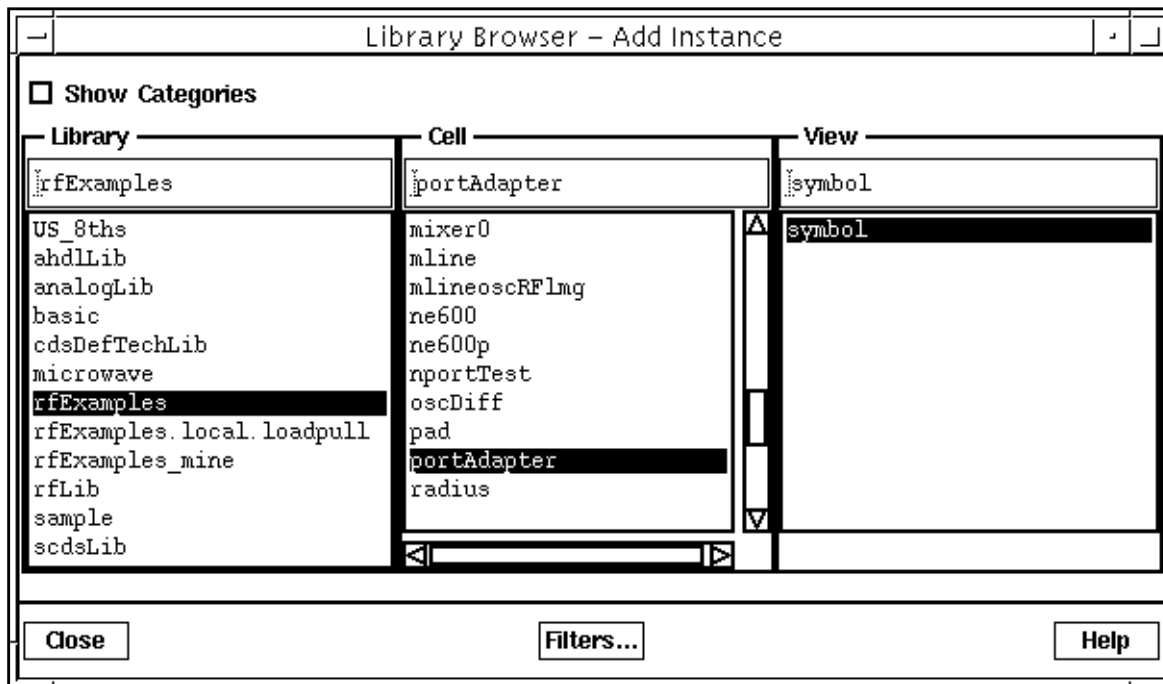
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

b. Highlight *portAdapter* for *Cell*.

c. Highlight *symbol* for *View*.

The completed form looks like this.



6. Place an instance of the port adapter in the schematic as shown in the following figure. Put your cursor in the Schematic window and then left click with the mouse.

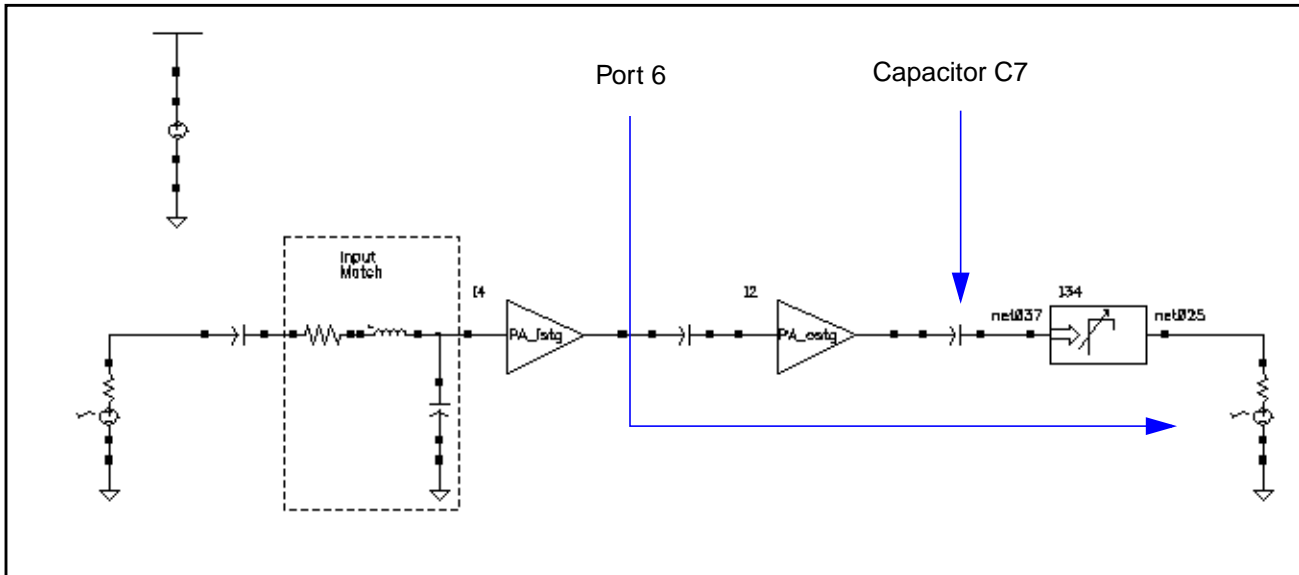
After you place the port adapter in the schematic, you can remove the port adapter symbol from your cursor by pressing the *ESC* key.

7. In the Schematic window, connect the port adapter between Capacitor C7 and Port 6.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The edited schematic looks like the one below.



The port adapter and its terminating port, Port6, must have the same reference resistance.

8. In the Schematic window, select the Port Adapter.
9. In the Schematic window, choose *Edit – Properties – Objects*.

The Edit Object Properties form appears with information for the Port Adapter displayed.

10. In the Edit Object Properties form, do the following and then click *Apply*.
 - a. Type `frf` for *Frequency*.
 - b. Type `theta` for *Phase of Gamma (degrees)*.
 - c. Type `mag` for *Mag of Gamma (linear scale)*.
 - d. Type `r0` for *Reference Resistance*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed form looks like this.

Property	Value	Display
Library Name	rfExamples	off
Cell Name	portAdapte	off
View Name	symbol	off
Instance Name	I34	off

User Property	Master Value	Local Value	Display
interfaceLastC..	26 16:05:39 1999		off

CDF Parameter	Value	Display
Frequency	frf	off
Phase of Gamma (degrees)	theta	off
Mag of Gamma (linear scale)	mag	off
Reference Resistance	r0	off
Gamma Phase Offset (deg)	0	off
Gamma Mag Offset (linear)	0	off
	atan(f*1)+3.14159/180	off

11. In the Schematic window, select Port 6.

The Edit Object Properties form changes to display data for Port 6.

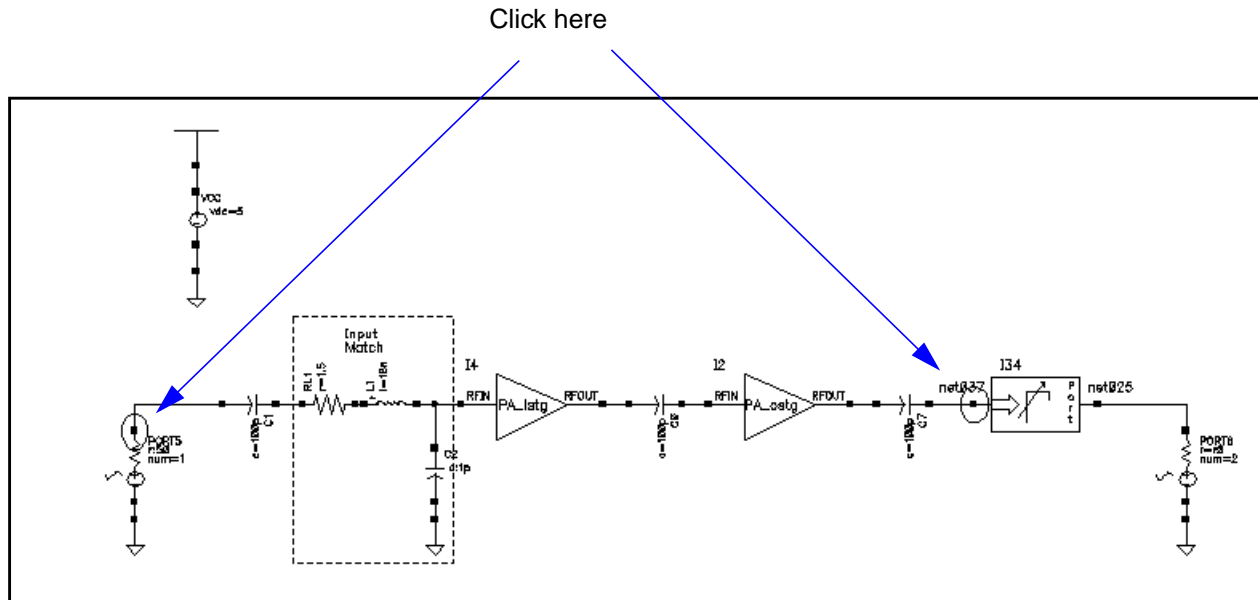
12. In the Edit Object Properties form, type $r0$ for *Resistance* and then click *OK*.

13. In the Simulation window, choose *Outputs – To Be Saved – Select on Schematic*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

14. In the Schematic window, click on the terminals that are circled in the diagram below.



After you click on a terminal, it is circled in the schematic.

15. In the Simulation window, choose *Variables – Copy from Cellview*.

The Design Variables list box in the Simulation window changes to reflect the copied variables from the schematic.

16. In the Simulation window, choose *Variables Edit....*

The Editing Design Variables form appears.

17. In the Editing Design Variables form, do the following for each variable in the *Table of Design Variables* and then click on *OK*.

- a. Highlight a variable in the *Table of Design Variables*.
- b. Type the appropriate value in the *Value (Expr)* field. (See below.)
- c. Click on *Apply*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The table shows you the appropriate value for each variable.

Variable	Value
theta	0
r0	50
mag	0
frf	1G

18. In the Simulation window, choose *Design – Check and Save*.

Performing the Simulation

1. In the Simulation window, choose *Analyses – Choose....*

The Choosing Analyses form appears.

2. In the Choosing Analyses form, highlight PSS.

The form changes to display options for PSS simulation.

3. In the Choosing Analyses form, do the following and then click *OK*:

- a. If information for f_{rf} is not displayed in the *Fundamental Tones* list box, type `frf` in the *Name* field below the list box and click *Clear/Add*.
- b. Click *Update From Schematic* to update the values in the *Fundamental Tones* list box from the schematic. Note that before you can update from the schematic, you must perform a *Check and Save* on the design.
- c. Highlight the *Auto Calculate* button to automatically calculate and enter a value in the *Beat Frequency* field.
- d. Choose *Number of Harmonics* for *Output harmonics* and type 9 in the adjacent field.
- e. Highlight the *Sweep* button and choose *Variable* in the associated cyclic field.
- f. Type `theta` for *Variable Name* (or click on *Select Design Variable* and highlight `theta` in the form that appears).
- g. Choose *Start – Stop* for *Sweep Range*.
- h. Type 0 for the *Start* value and 359 for the *Stop* value.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

- i. Choose *Linear* for *Sweep Type*.
- j. Highlight *Number of Steps* and type 20 in the adjacent field.

The completed form looks like this.

Choosing Analyses -- Affirma Analog Circuit Design Envir

OK Cancel Defaults Apply Help

◆ Beat Frequency
◇ Beat Period 16 Auto Calculate ■

Output harmonics
Number of harmonics 9

Accuracy Defaults (errpreset)
 conservative moderate liberal
Additional Time for Stabilization (tstab) :
Save Initial Transient Results (saveinit) no yes

Oscillator

Sweep ■ Frequency Variable? ◆ no ◇ yes
Variable Variable Name theta
Select Design Variable

Sweep Range
◆ Start-Stop Start 0 Stop 35
◇ Center-Span

Sweep Type
◆ Linear ◇ Step Size 20
◇ Logarithmic ◆ Number of Steps

Add Specific Points

Enabled ■ Options...

4. In the Simulation window, choose *Tools – Parametric Analysis....*

The Parametric Analysis form appears.

5. In the Parametric Analysis form do the following:

- a. Type `mag` for *Variable Name*.
- b. Choose *From/To* for *Range Type* and type 0 for *From* and .95 for *To* in the adjacent fields.
- c. Choose *Linear* for *Step Control* and type 10 in the adjacent *Total Steps* field.
- d. Choose *Analysis – Start*.

Look in the CIW for a message that says the simulation completes successfully.

Displaying Load Contours

1. In the Simulation window, choose *Results – Direct Plot – PSS....*

The PSS Results form appears.

2. In the PSS Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.
 - b. Highlight *Power Contours* for *Function*.
 - c. Highlight *Magnitude* for *Power Modifier*.
 - d. Type 9 for *Number of Contours*, if necessary.
 - e. Type 50.0 for *Reference Resistance*, if necessary.
 - f. Highlight 1 1G for *Output Harmonic*.

You select a different harmonic if the PSS fundamental frequency is smaller than 1GHz. For example, if another part of the circuit is driven at 1.5 GHz, the fundamental will be 500MHz. But because the part of the circuit where you want to do load-pull analysis operates at 1GHz, in order to plot the correct contours, the contours associated with 1GHz, you must specify the second harmonic of the fundamental, $2 \times 500\text{MHz} = 1\text{GHz}$.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed form looks like this.

Plot Mode Append Replace

Analysis

pss

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves
 Power Contours Reflection Contours
 Harmonic Frequency

Select

Power Modifier Magnitude dB10 dBm

Maximum Power

Minimum Power

Reference Resistance

Number of Contours

Close Contours

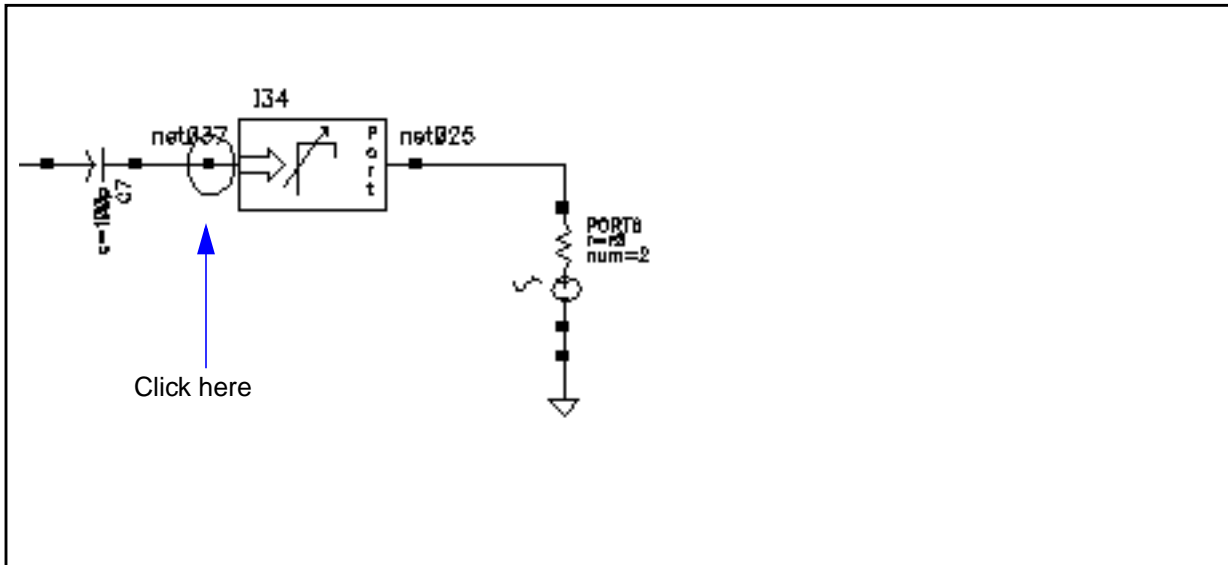
Output Harmonic

0	0
1	1G
2	2G
3	3G
4	4G

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

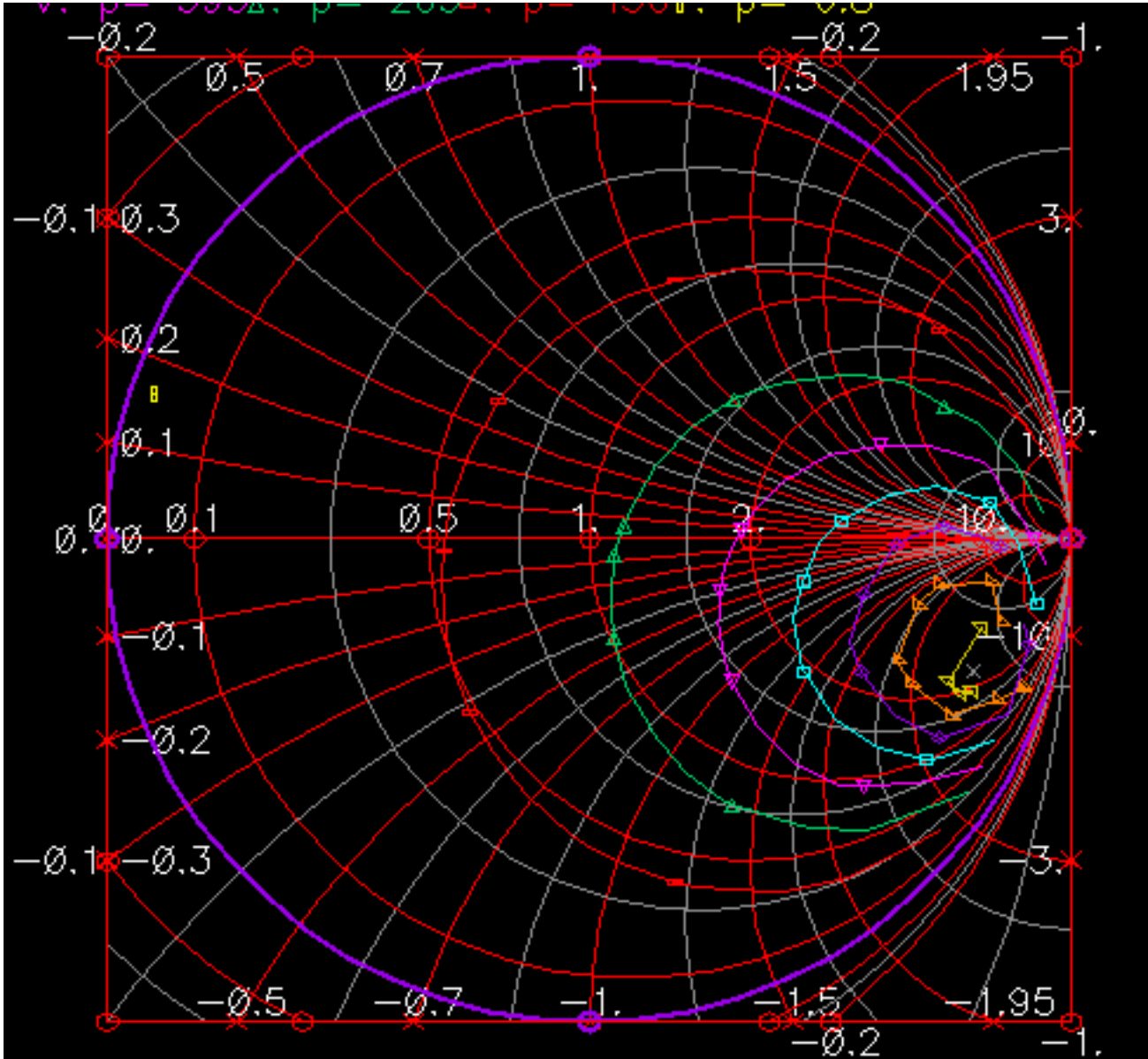
1. In the Schematic window, click on the terminal circled in the diagram below:



Subsequent instructions call this terminal “the terminal at net 037.”

2. The plot for the load contours appears in the Waveform window.

Figure 11-2 Load Contour Plot



From [Figure 11-2](#) on page 517, a normalized load impedance of about $2.48-j4.738$ dissipates the most power. A small “x” appears at the maximum power point, which in this case lies near the center of the smallest constant power contour.

Adding the Reflection Contours to the Plot

You might want to maximize load power subject to a constraint on the magnitude of the amplifier’s input reflection coefficient. Such a constraint can prevent unstable interactions with

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

the preceding stage. You can overlay load-pull contours with contours of constant input reflection coefficient magnitude. The optimal load corresponds to the reflection coefficient that lies on the “largest power” load-pull contour and also lies on a constant input reflection coefficient contour that is within the constraint. Here, “largest power” means the contour corresponding to the largest amount of power delivered to the load. To use the topographical map analogy again, this is like overlaying constant elevation contours and contours of constant temperature. The optimization objective would be like trying to find the highest point on the mountain such that the temperature is greater than 60 degrees.

1. In the PSS Results form, do the following:
 - a. Highlight *Append* for *Plot Mode*.
 - b. Highlight *Reflection Contours* for *Function*.
 - c. Select *Separate Refl and RefRefl Terminals*.
 - d. Type 9 for *Number of Contours*, if necessary.
 - e. Type 50.0 for *Reference Resistance*, if necessary.
 - f. Highlight 1 1G for *Output Harmonic*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed form looks like this.

Plot Mode Append Replace

Analysis

pss

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves
 Power Contours Reflection Contours
 Harmonic Frequency

Select

Max Reflection Mag

Min Reflection Mag

Reference Resistance

Number of Contours

Close Contours

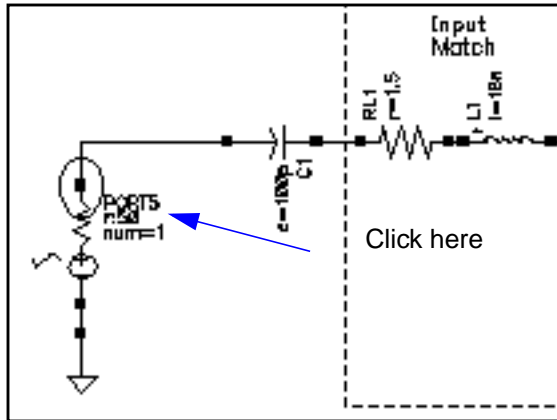
Output Harmonic

0	0
1	1G
2	2G
3	3G

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

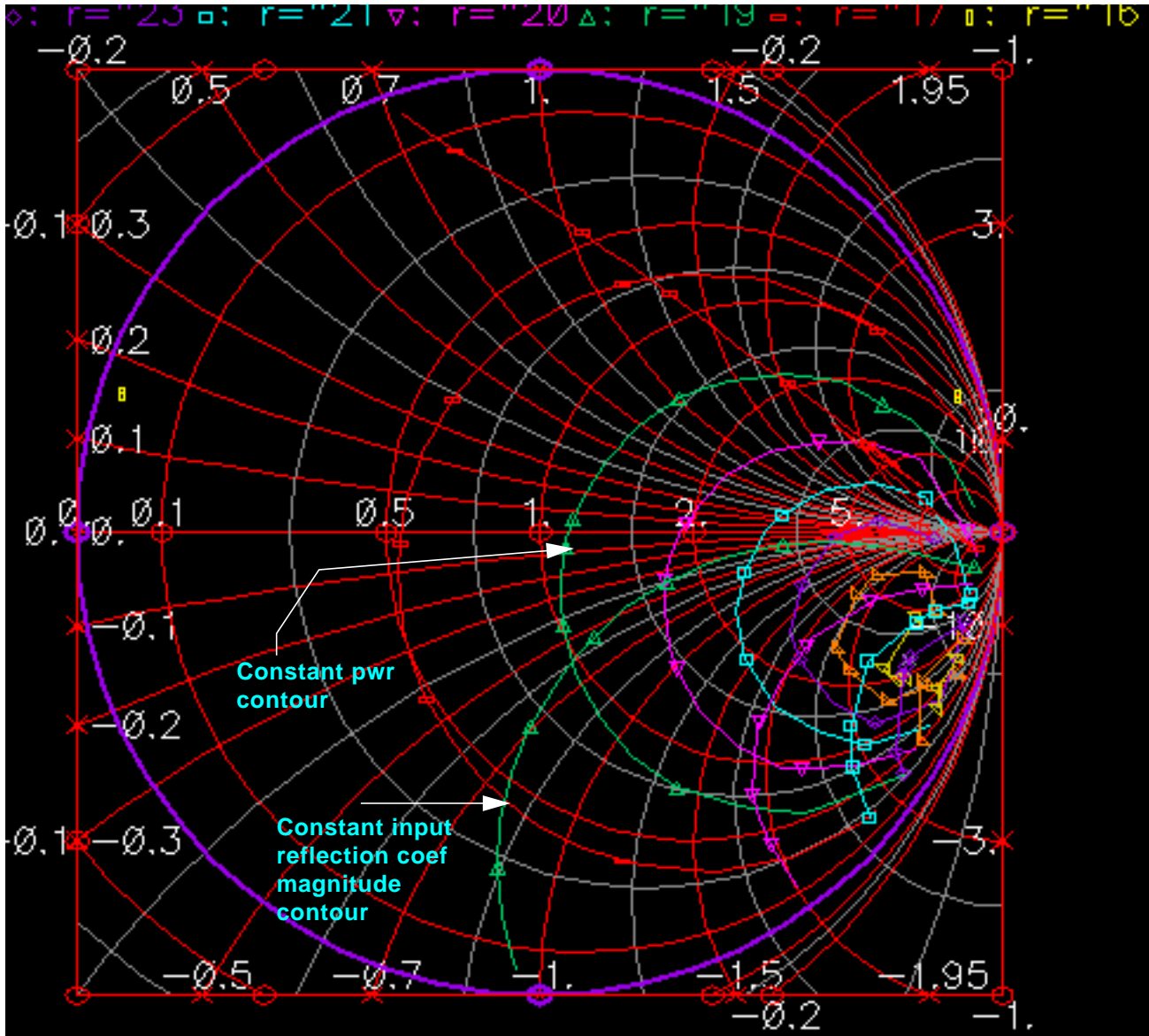
2. In the Schematic window, click on the terminal circled in the diagram below:



Subsequent instructions call this terminal “the terminal at Port 5.”

3. In the Schematic window, click on the terminal at net 037.

The plot for reflection contours is added to the Waveform window.



Let's assume that for stability, the input reflection coefficient should be less than or equal to 0.218. For the curves shown above, the optimal normalized load impedance is approximately $3.908 - j5.073$. Assuming you were free to draw smaller constant power contours, the true constrained optimal power point would occur when the constant power contour is just tangent to the constant input reflection coefficient contour of 0.218.

Are Constant Power Contours the Same as Constant Gain Contours?

In general, constant power contours are not the same as constant power gain contours. The simulator does not maintain any input impedance match while it sweeps the load to generate load-pull contours. The resulting contours are not constant power gain contours, but are only constant power contours. However, if the input reflection coefficient and input power do not change much over the load sweep range, the generated contours are good approximations of constant gain contours. By plotting constant power contours in the input reflection coefficient plane, you can see how both changed during the sweep of load reflection coefficient. If the input power does not vary with load, the constant power contours are also constant power gain contours. If it does vary, the contours tell you how much you have to change the input matching network during the load sweep to maintain a constant input power.

1. In the PSS Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.
 - b. Highlight *Power Contours* for *Function*.
 - c. Highlight *Select Power/Ref Terminal*.
 - d. Type 9 for *Number of Contours*, if necessary.
 - e. Type 50.0 for *Reference Resistance*, if necessary.
 - f. Highlight 1G for *Output Harmonic*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed form looks like this.

Plot Mode Append Replace

Analysis

pss

Function

Voltage Current
 Power Voltage Gain
 Current Gain Power Gain
 Transconductance Transimpedance
 Compression Point IPN Curves
 Power Contours Reflection Contours
 Harmonic Frequency

Select

Power Modifier Magnitude dB10 dBm

Maximum Power

Minimum Power

Reference Resistance

Number of Contours

Close Contours

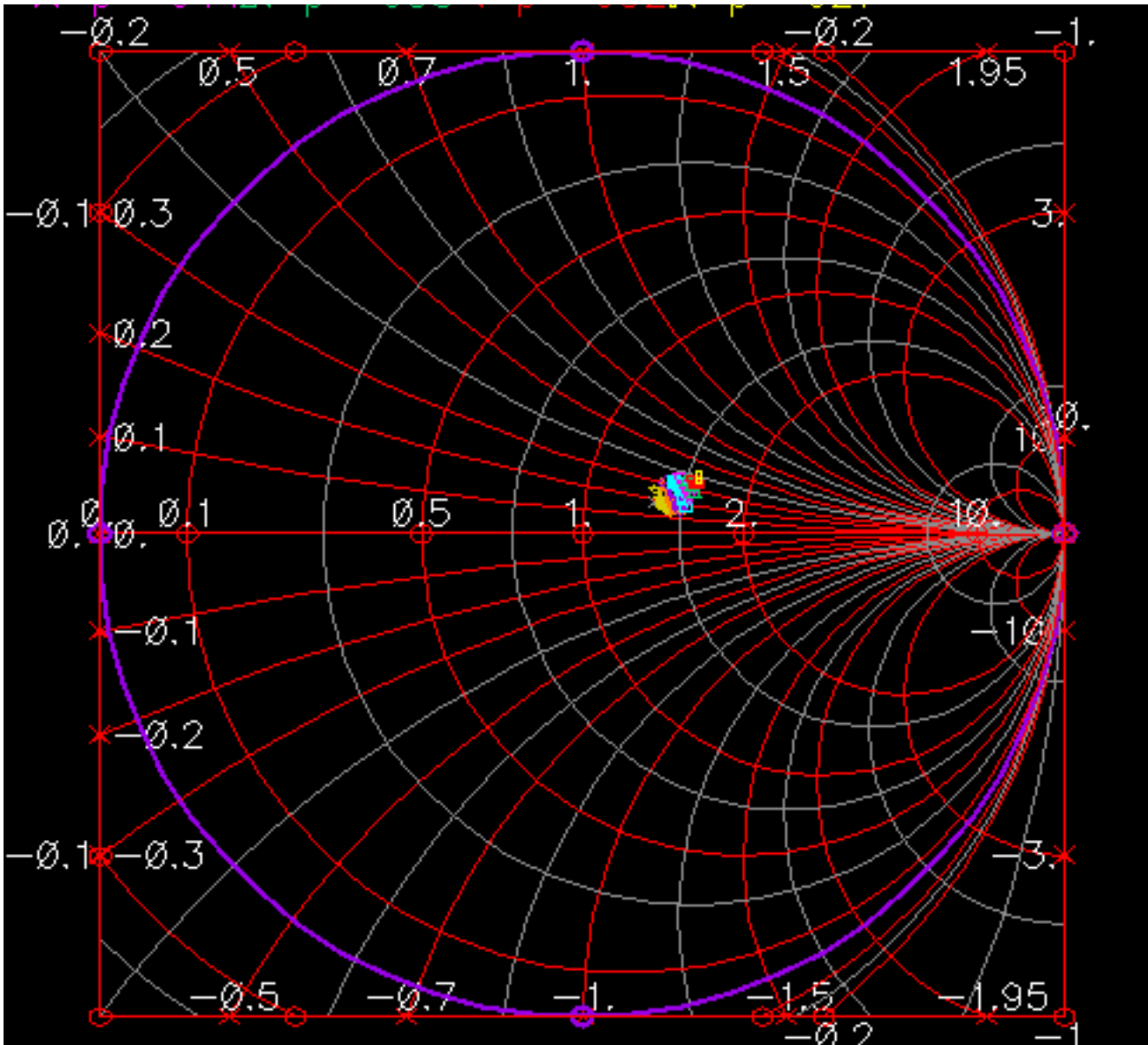
Output Harmonic

0	0
1	1G
2	2G
3	3G

2. In the Schematic window, click on the terminal at Port 5.

The plot of input reflection coefficients appears as in Figure 11-3.

Figure 11-3 Smith Chart of Input Reflection Coefficients Generated by Sweeping Load



3. In the Waveform window, choose *Zoom – Zoom In* and then click and drag with the mouse to form a rectangle that includes the area you want to see. The magnified view is shown in Figure 11-4 on page 525.

The real part of the input impedance varies from about 1.3 to 1.6 in normalized units and the imaginary part varies from 0.15 to 0.4 in normalized units. The input power varies

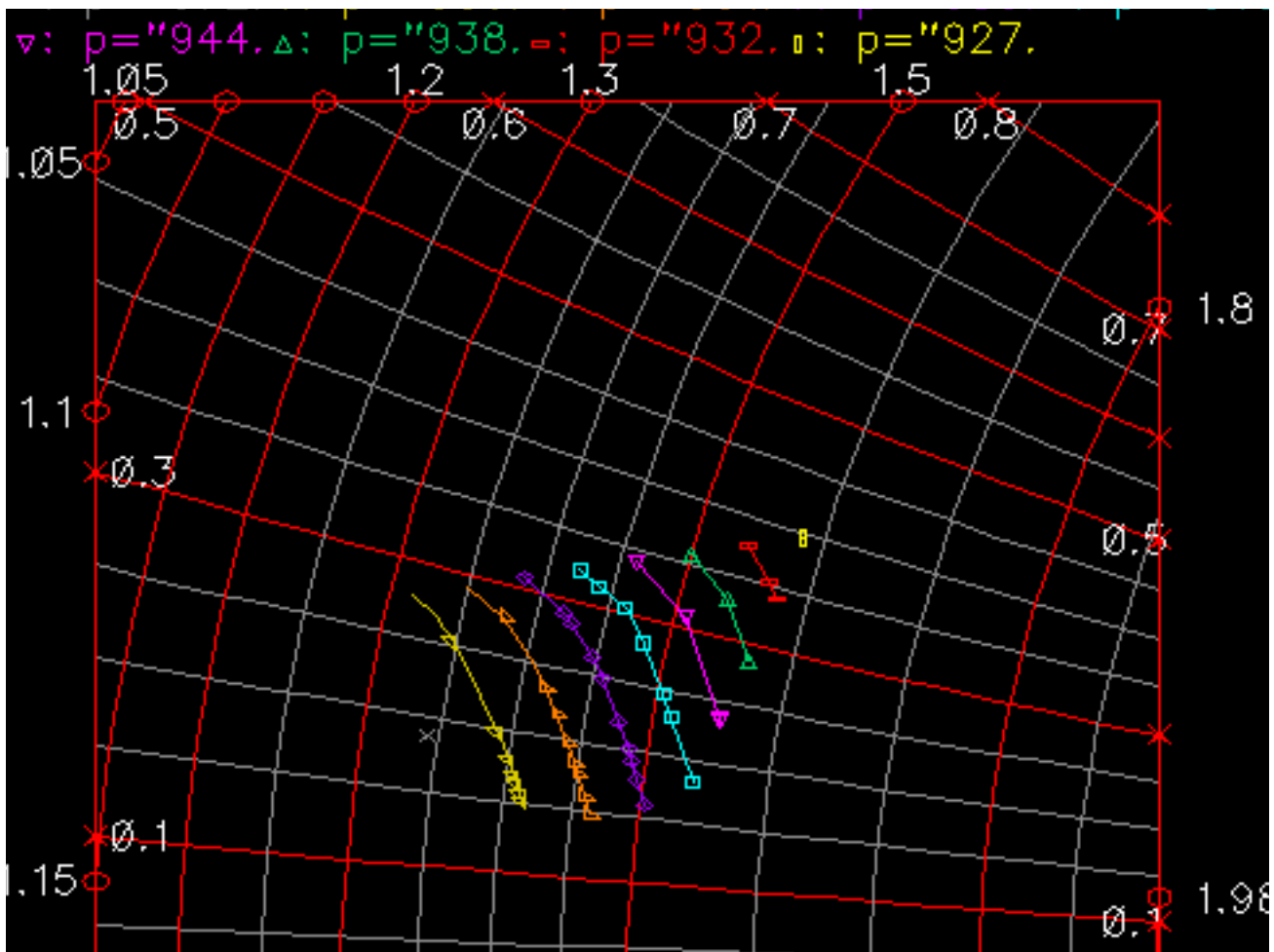
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

from 927nW to 967nW. You find these numbers by placing the cursor on the end points of the outside contours. You read the impedance and power/reflection for that cursor location at the top of the window.

The numbers varied slightly as the load was swept. The objective here is to determine how much the input reflection coefficient and the input power vary as the load varies. If the input quantities vary significantly, then the generated load pull contours do not coincide with the constant power *gain* contours because the input power did not remain constant. The amount of variation in the input reflection coefficient tells you by how much you would have to re-tune the input matching network to maintain constant input power as you sweep the load. You have to decide whether this much variation matters.

Figure 11-4 Magnified View of the Constant Power Contours



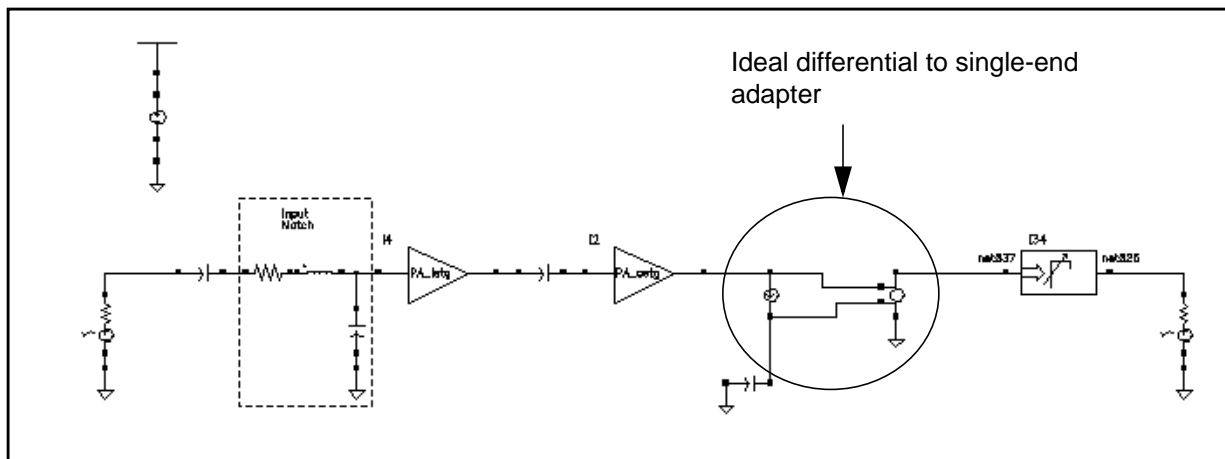
Moving to Differential Mode

Up until now, the example used in this chapter has been single-ended. Since many circuits are differential, [Figure 11-5](#) on page 526 shows a simple way to modify the example circuit to make it differential.

If your circuit has one or more differential ports, you can translate between single-ended and differential circuitry using linear-dependent controlled sources. The extra circuitry added to [Figure 11-5](#) on page 526 shows a modified version of the circuit where the output has been modified to make it differential. The transition from single-ended to differential output was made using the controlled sources F0 and E1 (circled in [Figure 11-5](#)).

The controlled sources are available in *analogLib* as *cccs* and *vcvs*. The gain of the *cccs* is -1 to be consistent with the polarity of the *vcvs*. The *vcvs* (E1) is the *Name of voltage source* parameter in the *cccs*.

Figure 11-5 Simple Differential Circuit



Using S-Parameter Input Files

This example shows you how to create files of S-parameter data and *how to specify a general n-port device in the frequency domain* for a new simulation. In this example, the created S-parameter data file is entered back into the original circuit. It replaces that portion of the circuit used to create the S-parameters.

To enter tabulated S-parameters from some other source, simply record the data in the format shown in this example and select the appropriate numerical options in the n-port.

To run a SpectreRF analysis that includes the new n-port, make sure you chose the rational fitting option.

Setting Up the Schematic for the First Simulation

This example requires that you

- Open the *EF_example* schematic.
- Create a new cellview.
- Copy part of the *EF_example* schematic and paste it in the new cellview.
- Simulate the new cellview. This simulation generates the S-parameters of a linear time-invariant part of the original circuit. These S-parameters will be stored and used later to import S-parameters into a simulation.

Opening the EF_example Schematic

1. Open and set up the *EF_example* schematic for simulation.

This is the same schematic that is used in the Envelope Following example of this chapter.

2. Leave the *EF_example* schematic open in the background.

Creating and Editing the New Cellview

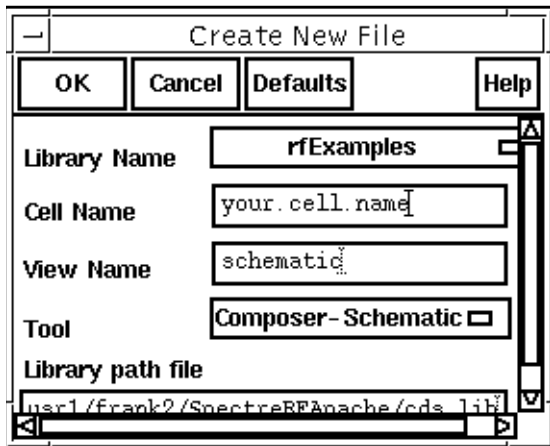
1. In the CIW, choose *File – New – Cellview....*

The Create New File form appears.

Affirma RF Simulator (SpectreRF) User Guide

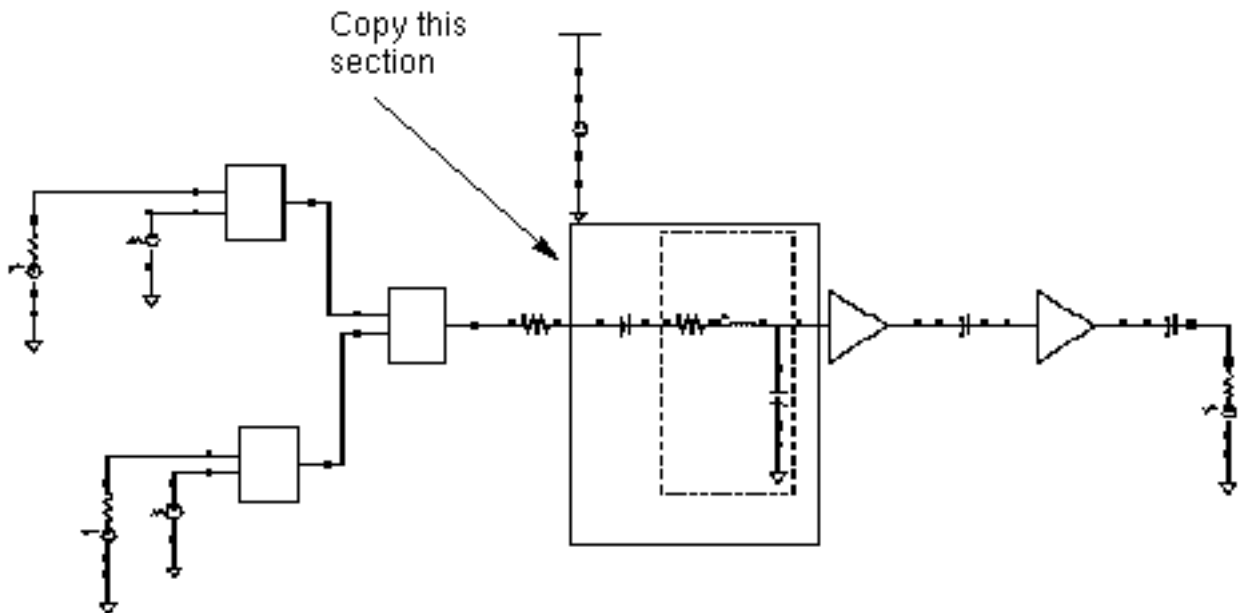
Modeling Transmitters

2. In the Create New File form, type a name for the new view in the *Cell Name* field and click *OK*.



After you click *OK*, a new empty Schematic window appears.

3. In the *EF_example* Schematic window, copy the part of the schematic shown in the picture below and paste it into the new Schematic window.



Note: If you need assistance with methods for editing the schematic, see the [Virtuoso® Schematic Composer™ User Guide](#).

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

4. In the new Schematic window, choose *Add – Instance....*

The Add Instance form appears.

The screenshot shows the 'Add Instance' dialog box. The title bar reads 'Add Instance'. Below the title bar are four buttons: 'Hide', 'Cancel', 'Defaults', and 'Help'. The main area contains four input fields: 'Library' (containing 'analogLib'), 'Cell', 'View', and 'Names'. A 'Browse' button is located to the right of the 'Library' field. Below these fields are 'Array', 'Rows' (with a value of 1), and 'Columns' (with a value of 1). At the bottom of the dialog are three buttons: 'Rotate', 'Sideways', and 'Upside Down'.

5. In the Add Instance form, click on *Browse*.

The Library Browser – Add Instance form appears.

6. Select the *analogLib* and highlight *Show Categories*.
7. In the Library Browser – Add Instance form, do the following:
 - a. Highlight *Sources* for *Category*.
 - b. In the *Category* list box, click on *Sources*.

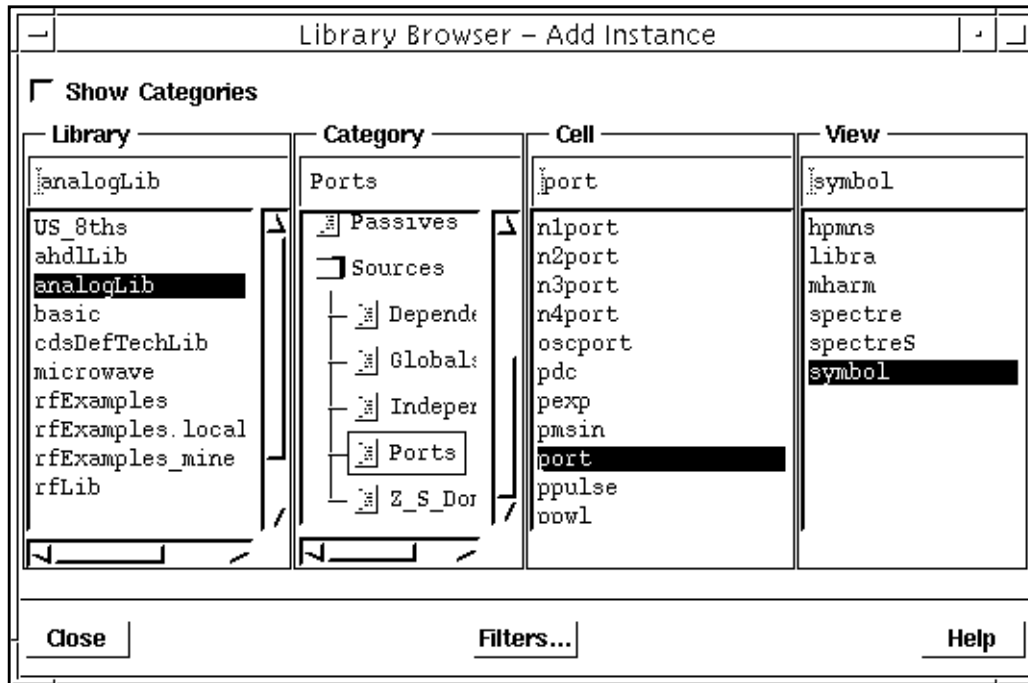
The *Category* list box expands to display several subcategories under *Sources*.

- c. Highlight the *Ports* subcategory under *Sources*.
- d. Highlight *port* for *Cell*.
- e. Highlight *symbol* for *View*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The form now looks like the one below:

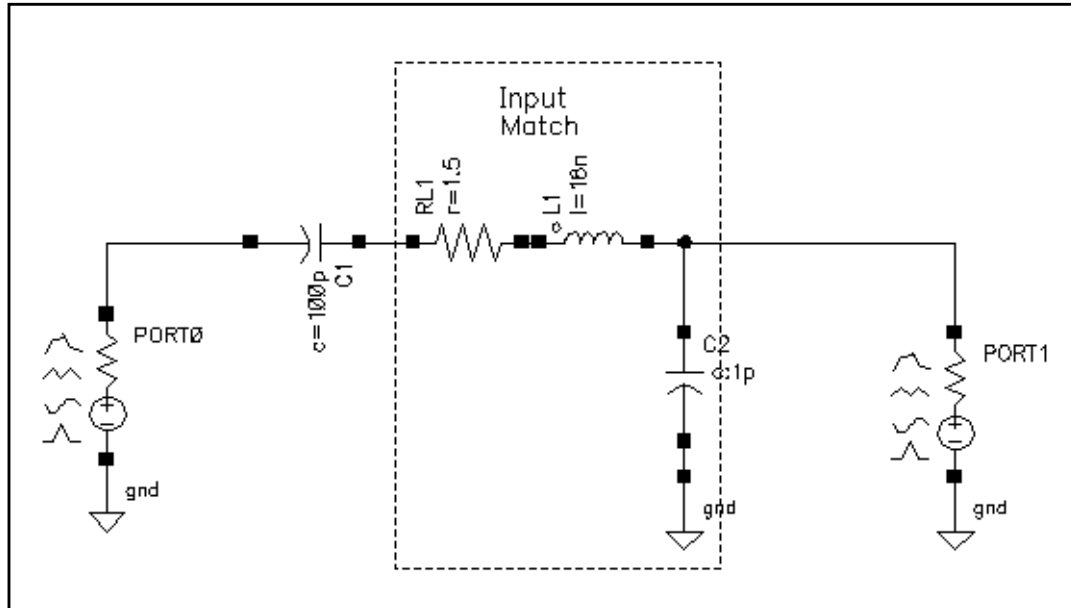


8. Place two *port* symbols in the new Schematic window.
9. In the Library Browser – Add Instance form, do the following:
 - a. Highlight *analogLib* for Library.
 - b. Highlight *Everything* for Category.
 - c. Highlight *gnd* for Cell.
 - d. Highlight *symbol* for View.
10. Place two gnd components in the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

11. Wire up the components in the new Schematic window to make them look like this.



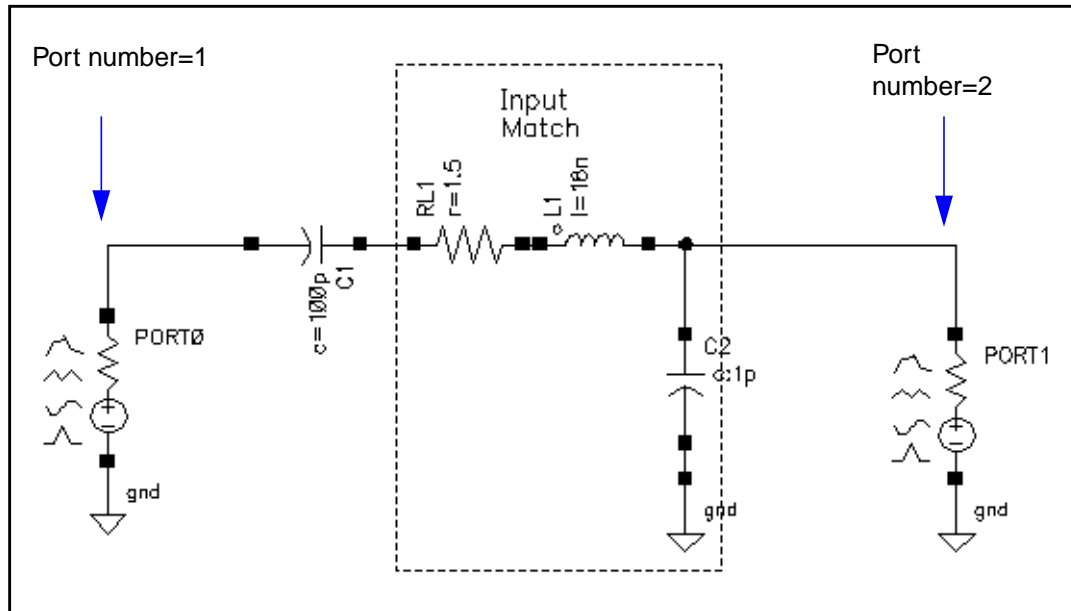
12. In the new Schematic window, choose *Edit – Properties – Objects....*

13. For each of the *port* components in the new schematic, do the following in the Edit Object Properties form:

- a. Select the port.
- b. Type the appropriate number in the *Port number* field.

The diagram below tells you the appropriate number to type. Leave the Port resistances at their default of 50 Ohms.

c. Click on *Apply*.



14. In the Edit Object Properties form, click *OK*.
15. In the new Schematic window, choose *Design – Check and Save*.

Setting Up and Running the First Simulation

1. In the new Schematic window, choose *Tools – Analog Environment*.
2. In the Simulation window, choose *Analyses – Choose....*

The Choosing Analyses form appears.

3. In the Choosing Analyses form, do the following:
 - a. Highlight *sp* for *Analysis*.
 - b. Highlight *Frequency* for *Sweep Variable*.
 - c. Highlight *Start – Stop* for *Sweep Range*.
 - d. Type *100K* for *Start* and *10G* for *Stop*.
 - e. Choose *Automatic* for *Sweep Type*.
 - f. Highlight *no* for *Do Noise*.
4. In the Choosing Analyses form, click on *Options....*

The S-parameter Options form appears.

5. In the S-parameter Options form, type the path to the output S-parameter file in the *file* field of the *OUTPUT PARAMETERS* section and click *OK*.

OUTPUT PARAMETERS

file

oppoint rawfile screen logfile no

6. In the Choosing Analyses form, click OK.
7. In the Simulation window, select *Simulation – Netlist and Run*.

The output S-parameter file from the first simulation is used as input for the second. Be sure to check in the CIW for a message that says the simulation completed successfully.

The S-Parameter File

This section shows you files for the previous simulation.

S-Parameter Formats

You can look at the S-parameter output file that the simulation created. You can use that format to import S-parameter files from other sources.

Setting Up and Running the Second Simulation

In this simulation you use the S-parameter output from the first simulation to model the first simulation components.

1. In the Schematic window you just created for the first simulation, add instances to the schematic as shown in the following table:

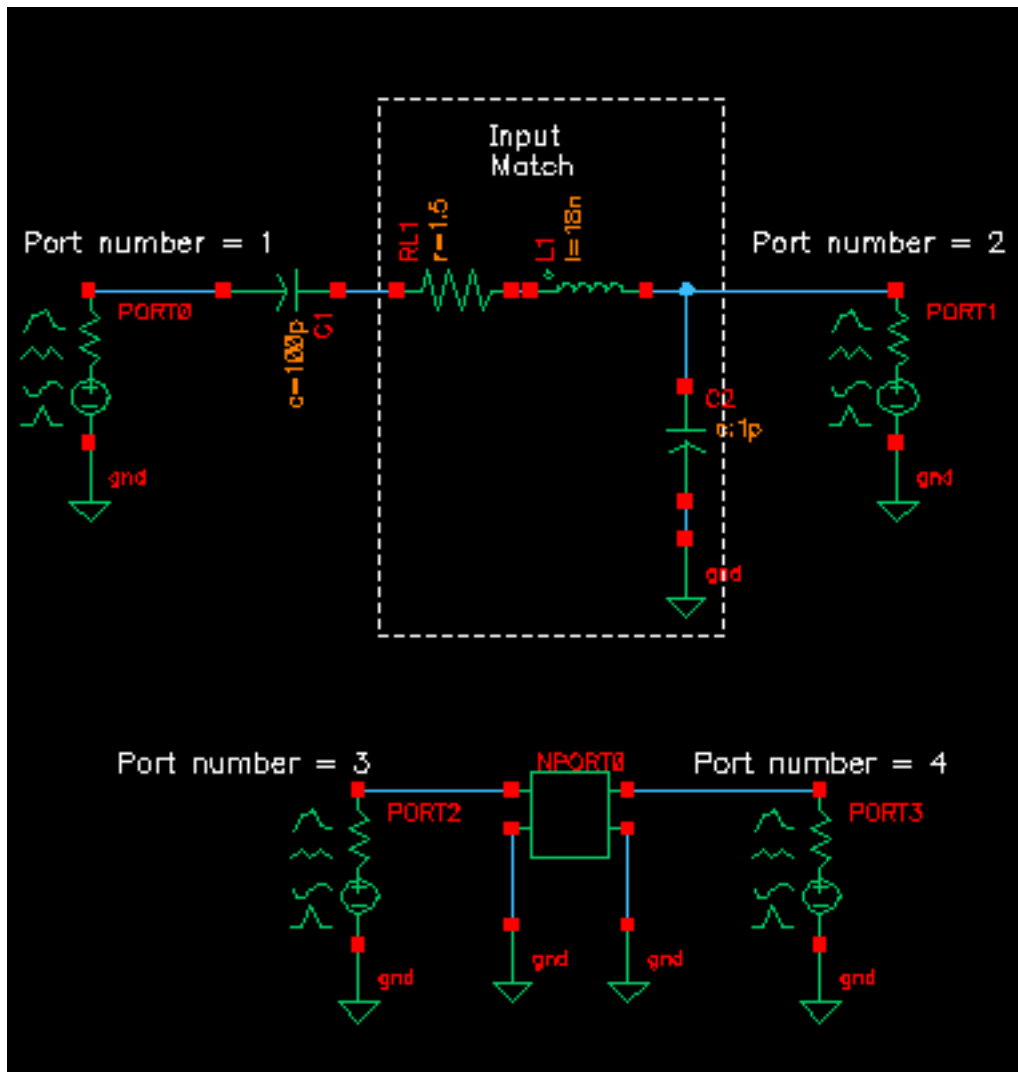
Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

Table 11-1

Library	Category	Subcategory	Cell	View	Number
analogLib	Sources	Ports	n2port	symbol	1
analogLib	Sources	Ports	port	symbol	2
analogLib	Everything	None	gnd	symbol	4

- Connect wires to the components to make the schematic look like the picture below and edit the properties on the new ports to assign port numbers as shown:



Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

3. In the Schematic window, select the *n2port*.
4. In the Edit Object Properties form, do the following and then click *OK*:
 - a. In the *S-parameter data file* field, type the absolute path to the S-parameter file you created in the last simulation.
 - b. Choose *rational* for *Interpolation method*.

The form changes to let you add additional information.
 - c. Type .001 for *Relative error*.
 - d. Type $1e-6$ for *Absolute error*.
 - e. Type 6 for *Rational order*.

This is mandatory if you plan to run SpectreRF.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed Edit Object Properties form looks like this.

Edit Object Properties

OK
Cancel
Apply
Defaults
Previous
Next
Help

Apply To only current instance

Show system user CDF

Browse
Reset Instance Labels Display

Property	Value	Display
Library Name	<input type="text" value="analogLib"/>	<input type="checkbox"/> off
Cell Name	<input type="text" value="n2port"/>	<input type="checkbox"/> off
View Name	<input type="text" value="symbol"/>	<input type="checkbox"/> off
Instance Name	<input type="text" value="NPORT"/>	<input type="checkbox"/> off

Add
Delete
Modify

CDF Parameter	Value	Display
S-parameter data file	<input type="text" value="2/frank/sparam.practice"/>	<input type="checkbox"/> off
Multiplier	<input type="text" value=""/>	<input type="checkbox"/> off
Scale factor	<input type="text" value=""/>	<input type="checkbox"/> off
Interpolation method	<input type="checkbox"/> rational	<input type="checkbox"/> off
Relative error	<input type="text" value=".001"/>	<input type="checkbox"/> off
Absolute error	<input type="text" value="1e-6"/>	<input type="checkbox"/> off
ROM data file	<input type="text" value=""/>	<input type="checkbox"/> off
Rational order	<input type="text" value="6"/>	<input type="checkbox"/> off

5. In the Schematic window, choose *Design – Check and Save*.
6. In the Simulation window, choose *Simulation – Netlist and Run*.

Plotting Results

1. In the Simulation window, choose *Results – Direct Plot – S-Parameter....*

The S-parameter Results form appears.

2. In the S-parameter Results form, do the following:

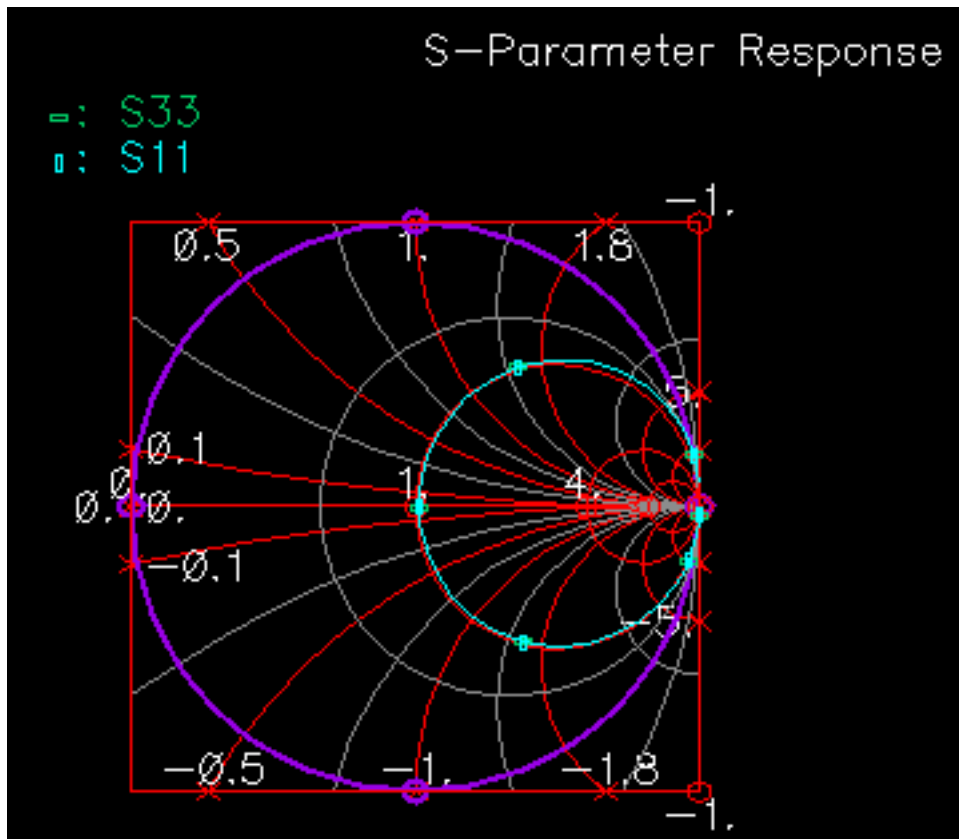
- a. Highlight *Replace* for *Plot Mode*.
- b. Highlight *SP* for *Function*.
- c. Highlight *Z-Smith* for *Plot Mode*.
- d. Click on *S11*.

The plot for S11 appears in the Waveform window.

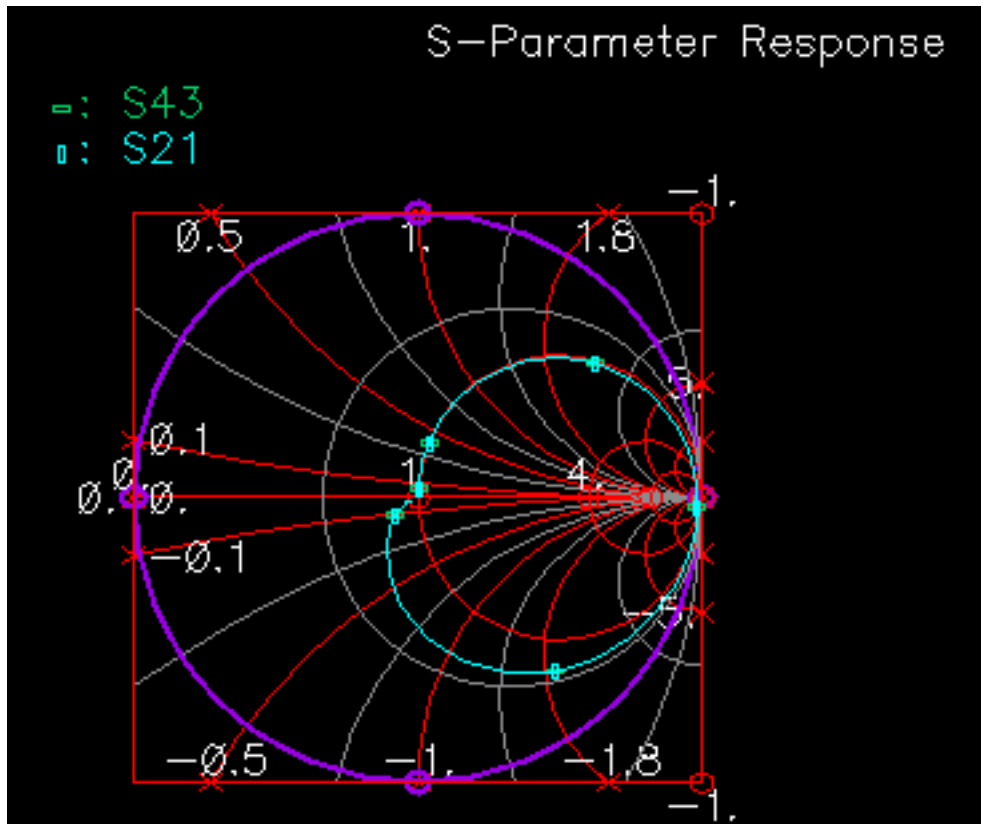
3. In the S-Parameter Results form, do the following:

- a. Highlight *Append* for *Plot Mode*.
- b. Click on *S33*.

The plot for S33 is appended to the S11 plot. The two plots are identical and show that results produced by the original components and the S-parameter input file are the same.



You can also check other plots for equivalency. Below is the plot produced by appending S21 to S43.



Using an S-Parameter Input File with a SpectreRF Analysis

This section uses the S-parameter file in an Envelope following Analysis and again compares results from using the file to results using the original components.

Setting Up the Schematic

1. In the *EF_example* schematic, choose *Design – Save As....*

The Save As form appears.

2. In the Save As form, do the following:

- a. In the *Library Name* field, type the name of a local library.

If you do not have a local library available, create one at the same directory level as the *rfExamples* directory.

Affirma RF Simulator (SpectreRF) User Guide

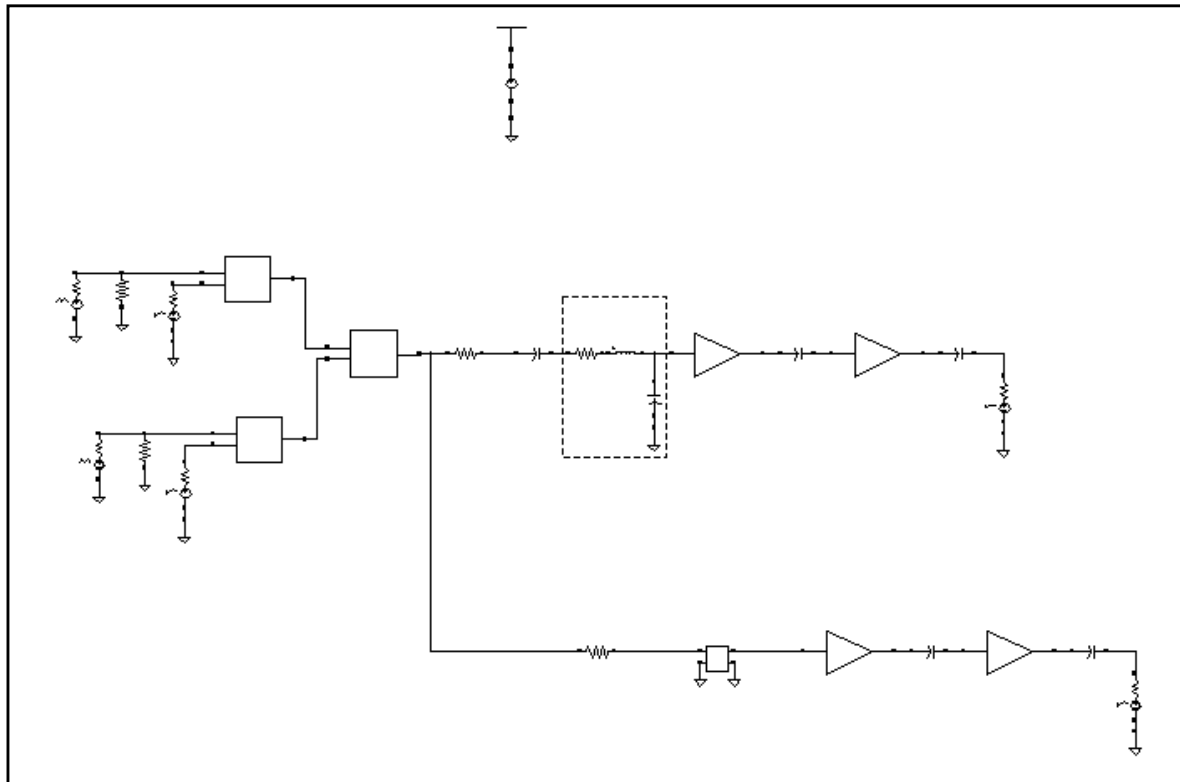
Modeling Transmitters

b. In the *Cell Name* field, type the name you want for a copy of the *EF_example* schematic.

c. Click *OK*.

You now have a copy of the *EF_example* schematic. Subsequent instructions refer to this copy as the *EF_example_copy* schematic.

3. Copy everything to the right of the adder and paste the copy below the original. Be sure to remove the “RFOUT” label from the duplicate branch or the two branches will be shorted together at their outputs.
4. Replace the circuitry you used to create the S-parameter file with the *n2port* from the last circuit. Wire the duplicate branch to the adder output. The circuit should look like this:



5. In the *EF_example_copy* schematic, choose *Design – Check and Save*.

Setting Up and Running the Simulation

1. Repeat steps 3 through 16 in the Envelope Following section to set up an Envelope Following analysis but this time only simulate 30 μ s.
2. In the Simulation window, choose *Simulation – Netlist and Run*.

Be sure to check the CIW for a message that says the simulation completed successfully.

Displaying the Results

1. In the Simulation window, choose *Results – Direct Plot – ENVLP....*

The Envelope Following Results form appears.

2. In the Envelope Following Results form, do the following:
 - a. Highlight *Replace* for *Plot Mode*.
 - b. Highlight *Voltage* for *Function*.
 - c. Highlight *Real* for *Modifier*.
 - d. Choose *1* for *Harmonic Number*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The completed form looks like this.

Envelope Following Results

OK Cancel Help

Plot Mode Append Replace

Function

Voltage Current Power

Description: Harmonic Voltage vs Time

Sweep spectrum harmonic time
 time

Modifier

Magnitude Phase dB20
 Real Imaginary

Harmonic Number

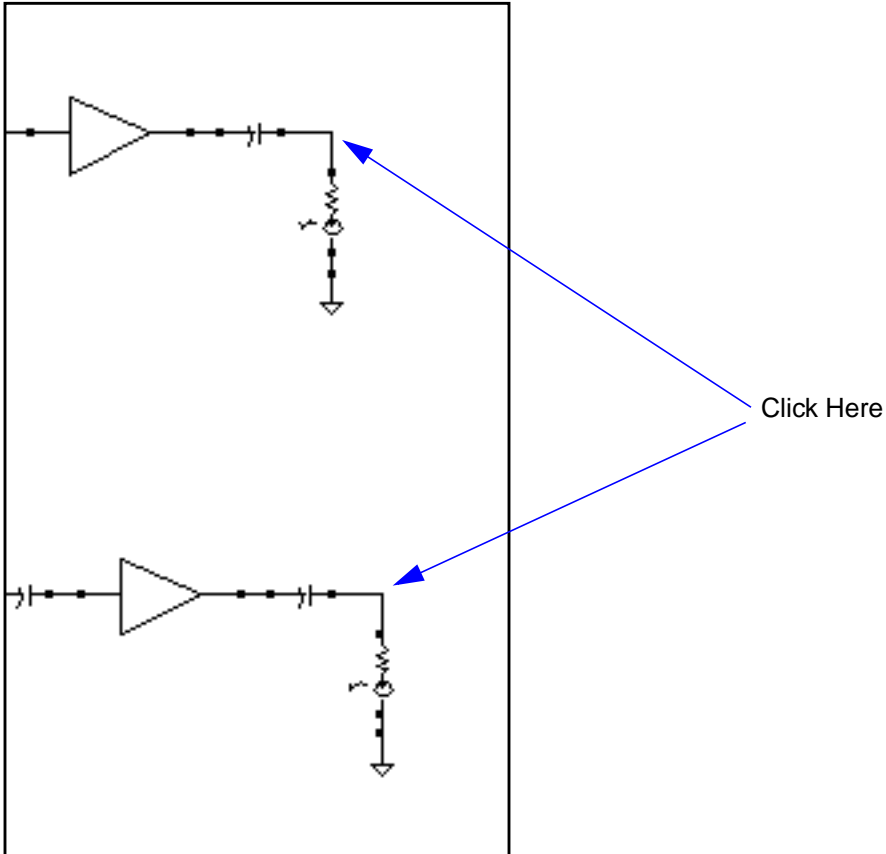
0
1
2
3

Note:
Harmonic 0 is a real waveform; therefore,
the Real Modifier should be used.
All other harmonics are complex waveforms.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

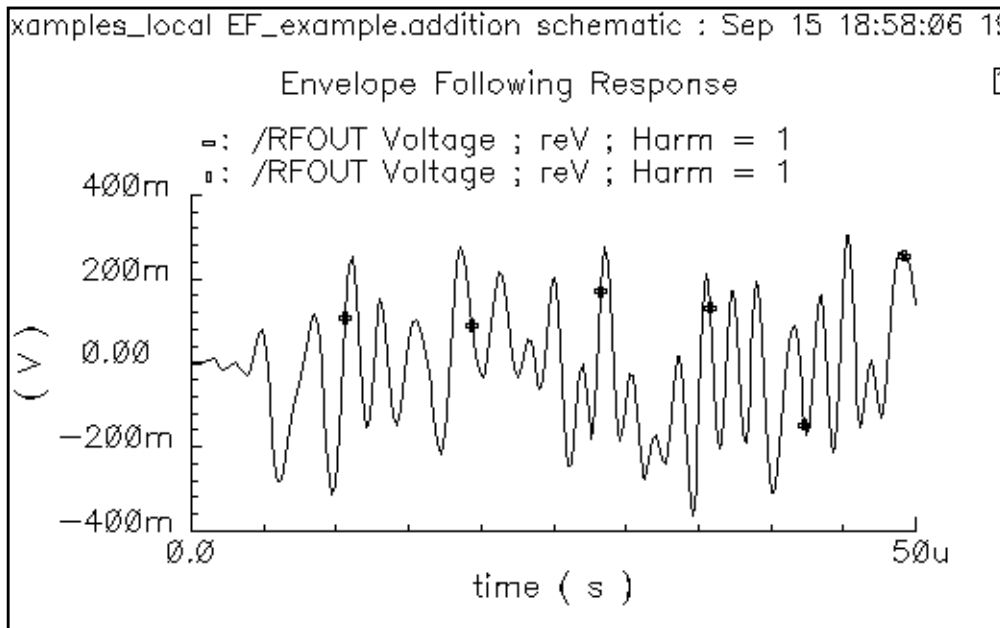
3. In the Schematic window, click on one of the output nodes, then change the *Plot Mode* to *Append* and click on the other output node. You can also confirm that the imaginary parts of the waveforms match.



Affirma RF Simulator (SpectreRF) User Guide

Modeling Transmitters

The coincident plots in the Waveform window show that the output is the same at both nodes. When the S-parameters are correct, the *n2port* device accurately simulates the circuitry represented by the S-parameters.



Modeling Spiral Inductors and Bondpads

The procedures for modeling spiral inductors and bondpads are similar in many ways.

Modeling Spiral Inductors

The on-chip passive component modeler simulates spiral inductors, planar transformers, and bonding pads for RF applications. The modeler has an extraction engine based on EM (Electromagnetics) techniques. The Affirma analog circuit design environment calls this engine using a command interface.

There are two use models for the modeler.

- One use model is using the modeler in the schematic flow. You specify the input parameters for the modeler in the CDF form, and model extraction is executed at netlisting time.
- The other use model is using the RF sub-menu option to access the modeler. You launch the modeler directly using the Affirma analog circuit design environment Tools menu. The inductor model is extracted and written to the specific model directory, and the electric parameters of the components are calculated and displayed on the UI form. You can include the extracted model in Spectre/SpectreRF simulations, with multiple instances if necessary, and therefore avoid a long extraction time at netlisting.

The solver for the on-chip passive modeler employs a PEEC (Partial Element Equivalent Circuit) algorithm to generate macromodels for the spiral components. Electro-static and magneto-static EM solvers are called separately to extract the capacitive and inductive parameters of the structure.

The capacitive parameter extraction solver uses the BEM (Boundary Element Method), which computes the lossy substrate using a complex Green's function, and the solver is SVD (Singular Value Decomposition) accelerated.

The inductance solver is based on Fasthenry. It considers skin-depth effects, and models the substrate effect by treating the lossy layers as lossy conductor planes. The meshes for inductive and capacitive extractions are inter-correlated. There is one more segment for

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

capacitive extraction than there is for the inductive mesh. There is also a half segment overlap between the two meshes. Because of the conductor and substrate loss, the extracted parameters are formulated as RLGC matrices.

With the extracted parameters, a multi-PI equivalent circuit is created. Depending on the bandwidth, the modeler generates either a narrow-bandwidth or wide-bandwidth model.

The narrow-bandwidth model uses the parasitic parameters extracted at the working frequency and builds a fully-coupled multi-PI equivalent circuit. Because the capacitive and inductive parameters are functions of frequency, this model is accurate only in the vicinity of the working frequency specified. The wide-bandwidth model extracts the capacitive and inductive parameters over a wide frequency range, and generates a model as an nport. In the nport, a ratio of rational polynomials represents the frequency dependent behavior. Unlike the narrow-bandwidth model, the wide-bandwidth model is accurate over the whole frequency range up to the specified maximum frequency of interest. However, the model extraction takes longer, and in some cases, the extracted model might not be passive and might cause simulation failure. If the simulation fails, the solver automatically switches to a narrow-bandwidth model.

The remainder of the spiral inductor section covers the following topics:

- The [Process File Preparation](#) section explains how to setup the process file and gives an example CMOS process file.
- The [Spiral Inductor Simulation in the Schematic Flow](#) section explains details about the spiral inductor symbol and CDF parameters.
- The [Spiral Inductor Modeling Using the RF Sub-menu](#) section tells you how to model a spiral inductor using the RF sub-menu and explains the electrical parameters.
- The [Equivalent Circuit Models](#) section explains the equivalent circuit models and discusses issues you need to be aware of.
- The [Example](#) section takes you step-by-step through a simulation that uses the spiral inductor modeling capability.

Process File Preparation

Before you use the passive modeler, you must create a process file that contains the layer information for both the metal and the substrate.

In the current release, you must hand edit the file to conform exactly to the required format.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

File Format

The process file is in the following format

```
processType type
numLayers # of layers
layerInfo 0
...
layerInfo # of layers-1
```

In the above format, the first line, *processType*, is currently used only for documentation. The second line specifies the number of layers, including both the dielectric and metal layers. The following lines are NI blocks of layer information. Each layer information block has a specific format you must follow.

In the process file, all separate structures are modeled so as to account for the effects of surrounding layers. For example, a two-metal layer structure immersed in the substrate is modeled as five different layers: the dielectric layer underneath the bottom metal layer, the bottom metal layer, the dielectric layer between the two metal layers, the top metal layer, and the layer above the top metal layer.

Layer Information Format

You can specify the following three types of layers:

- ground plane – The whole layer is a piece of metal.
- dielectric – The whole layer is dielectric of uniform dielectric constant,
- metal layer – Part of the structure is metal surrounded by dielectric.

The formats for these three layer types are shown below:

Format for the ground plane:

```
layer      0
  name      tab
  type      ground
  thickness  0.1e-6
  epsr      1
  rho       1.7857e-8
```

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

Format for the dielectric layer:

```
layer    1
  name      substrate
  type      dielectric
  thickness 300e-6
  epsr      11.9
  rho       0.1
```

Format for the metal layer:

```
layer    3
  name      metall
  type      metal
  thickness 1e-6
  epsr      4
  rho       1e10
  rhom      3e-8
```

For the metal layer, *rho* is the resistivity of the dielectric surrounding the metal structure, and *rhom* is the resistivity of the metals. All the units must be in MKS.

Sample CMOS Process File

The following example shows a process file for a CMOS process as shown in [Figure 12-1](#) on page 549.

```
*Sample process file. All layers are stacked from bottom up
processType    CMOS
numLayers      7
layer          0
  name          tab
  type          ground
  thickness     0.1e-6
  epsr          1
  rho           1.7857e-8
layer          1
  name          substrate
  type          dielectric
  thickness     300e-6
  epsr          11.9
  rho           0.1
layer          2
```

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

```
name          sio2
type          dielectric
thickness     1.5e-6
epsr          4.0
rho           1e10
layer        3
name          metall
type          metal
thickness     1e-6
epsr          4
rho           1e10
rhom          3e-8
layer        4
name          sio2
type          dielectric
thickness     1.5e-6
epsr          4
rho           1e10
layer        5
name          metal2
type          metal
thickness     1e-6
epsr          4
rho           1e10
rhom          3e-8
layer        6
name          sio2
type          dielectric
thickness     43.5e-6
epsr          4
rho           1e10
```

Figure 12-1 Sample CMOS process

sio2
metal2
sio2
metal1
sio2
Substrate
tab

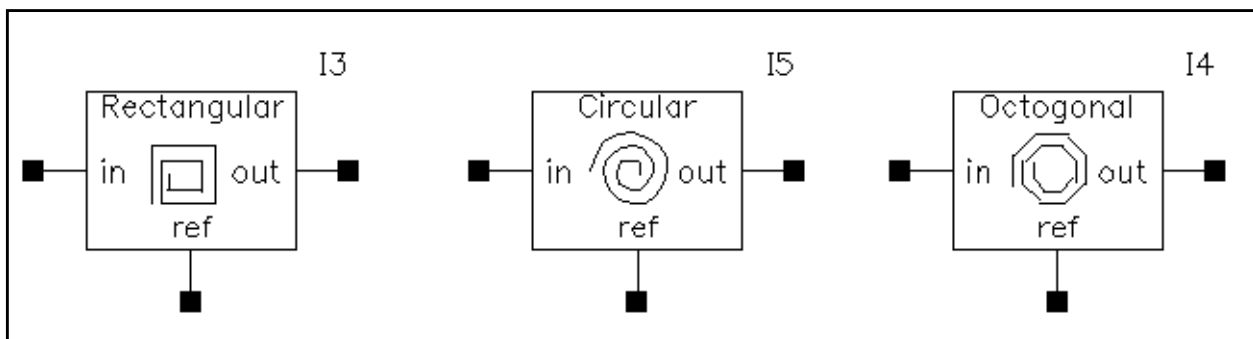
Spiral Inductor Simulation in the Schematic Flow

The spiral inductors, transformers, and bondpads are stored in the *passiveLib* library. The name of the spiral inductor component is *spirallnd*.

Symbols for Spiral Inductors

The spiral inductor symbol is a 3-port block with the actual spiral configuration displayed. [Figure 12-2](#) on page 549 shows the symbols for rectangular, circular and octagonal spiral inductors.

Figure 12-2 Symbols for spiral inductors: rectangular, circular and octagonal



Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

For the symbols shown in [Figure 12-2](#) on page 549, the input terminal represents the outer input terminal of the spiral inductor, the output terminal represents the return path, and the *ref* terminal represents the ground tab or the substrate bulk node.

Figure 12-3 CDF form for the single-spiral layer, rectangular spiral inductor

The screenshot shows the 'Add Instance' dialog box for a rectangular spiral inductor. The dialog is titled 'Add Instance' and has buttons for 'Hide', 'Cancel', 'Defaults', and 'Help'. The main configuration area includes:

- Library:** passiveLib
- Cell:** spiralInd
- View:** symbol
- Names:** (empty)
- Array:** Rows: 1, Columns: 1
- Orientation:** Rotate, Sideways, Upside Down (all unchecked)
- Use external model file:**
- Number of metal layers:** Single Double
- Spiral layer name:** metal2
- Return layer name:** metal1
- Inductor type:** Rectangular
- Process file name:** /tmp/cmosNew.data
- Outer area length(m):** 300
- Outer area width(m):** 300
- Conductor width(m):** 13
- Conductor spacing(m):** 7.1
- Number of turns:** 7
- Return path width(m):** 10
- Return path length(m):** 200
- Return path orientation:** 0
- Polarization:** Clockwise
- Accuracy level:** Liberal
- Model bandwidth:** Narrow Wide
- Working frequency:** 1e9

CDF Parameters

- **Use external model file:** This button lets you use externally-generated spiral-inductor model files directly in simulations. If you highlight this button, the CDF form looks like the

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

CDF form in [Figure 12-4](#) on page 552. The *Inductor Type* parameter only changes the appearance of the symbol in the schematic and does not affect the simulation results.

- **Number of Spiral Layers:** Possible choices are single spiral layer or double spiral layer inductors. For double layer inductors, the two spiral layers can be parallel or in series.
- **Spiral Layer Name:** The name of the layer the spiral sits in that you specify in the process file
- **Return layer name:** The name of the layer the return path sits in. For double-layer series inductors, this entry doesn't appear because the bottom spiral layer serves as return path.
- **Inductor Type:** The type of the spiral inductor. There are three inductor types: rectangular, circular and octagonal. Changing this option also changes the appearance of the CDF.
- **Process file name:** The name of the process file. You must specify the full path.
- **Outer area length:** The length of the outer dimension of the spiral
- **Outer area width:** The width of the outer area
- **Conductor width:** The width of the spiral
- **Conductor spacing:** The gap between two neighboring turns
- **Number of turns:** The number of spiral turns. The number can be fractional.
- **Return path width:** The width of the return path
- **Return path length:** The length of the return path
- **Return path orientation:** The angle between the return path and the innermost segment of the spiral, measured in degrees
- **Polarization:** Possible choices are clockwise or counterclockwise, looking from the top down
- **Accuracy level:** Determines the mesh density in capacitance and inductance extraction
- **Model bandwidth:** Possible choices are narrow bandwidth model and wide bandwidth model
- **Working frequency:** The actual working frequency of the spiral, used in the extraction of parasitics parameters
- **Maximum frequency:** The maximum frequency at which parasitic parameters are extracted for wide bandwidth models.

Figure 12-4 CDF form when Use external model file option is selected

The screenshot shows the 'Add Instance' dialog box. The title bar reads 'Add Instance'. Below the title bar are four buttons: 'Hide', 'Cancel', 'Defaults', and 'Help'. The main area contains several input fields: 'Library' (passiveLib), 'Cell' (spiralInd), 'View' (symbol), and 'Names' (I30). A 'Browse' button is located to the right of the 'Library' field. Below these fields are 'Array' settings with 'Rows' and 'Columns' input fields. At the bottom are three buttons: 'Rotate', 'Sideways', and 'Upside Down'. A section at the bottom has a checked 'Use external model file' checkbox, a 'Model name' field (indI30), and an 'Inductor type' dropdown menu set to 'Rectangular'.

CDF Form for the Circular Spiral Inductor

With the rectangular spiral, you specify the outer area length and width. With the circular spiral, you specify the outer radius of the spiral and the number of subsections in the polygon approximation of the spiral.

CDF Form for the Octagonal Spiral Inductor

An octagonal spiral inductor is a circular spiral inductor with eight subsections. [Figure 12-5](#) on page 553 shows the CDF file for an octagonal spiral inductor.

Figure 12-5 CDF form for an octagonal spiral inductor

The screenshot shows the 'Add Instance' dialog box for an octagonal spiral inductor. The dialog is titled 'Add Instance' and has buttons for 'Hide', 'Cancel', 'Defaults', and 'Help'. The main content area is divided into several sections:

- Library:** A text field containing 'passiveLib' and a 'Browse' button.
- Cell:** A text field containing 'spiralInd'.
- View:** A text field containing 'symbol'.
- Names:** A text field containing 'I3'.
- Array:** Two text fields for 'Rows' and 'Columns', both containing '1'.
- Orientation:** Three buttons: 'Rotate', 'Sideways', and 'Upside Down'.
- Use external model file:** A checkbox that is unchecked.
- Number of metal layers:** Radio buttons for 'Single' (selected) and 'Double'.
- Spiral layer name:** A text field containing 'metal2'.
- Return layer name:** A text field containing 'metal1'.
- Inductor type:** A dropdown menu showing 'Octogonal'.
- Process file name:** A text field containing '/tmp/cmosNew.data'.
- Conductor width(m):** A text field containing '13u'.
- Conductor spacing(m):** A text field containing '7.1u'.
- Number of turns:** A text field containing '7'.
- Outer radius(m):** A text field containing '1'.
- Return path width(m):** A text field containing '10u'.
- Return path length(m):** A text field containing '200u'.
- Return path orientation:** A text field containing '0'.
- Polarization:** A dropdown menu showing 'Clockwise'.
- Accuracy level:** A dropdown menu showing 'Liberal'.
- Model bandwidth:** Radio buttons for 'Narrow' (selected) and 'Wide'.
- Maximum frequency:** A text field containing '10G'.
- Working frequency:** A text field containing '1e9'.

Double Layered Spiral Inductors

When you set *number of spiral layers* to *double*, the CDF form changes to add a new entry, *Connection type*. In addition, the *Spiral layer name* field is replaced by two fields: *Spiral layer #1 name* and *Spiral layer #2 name*. If you choose *Series* for the *Connection type*, the *Return layer name* field is removed because one of the spiral layers functions as the return path.

Process File and Parameter Update

If you specify and apply new parameters, the model extraction is performed at netlisting time. If you do not change parameters, the previous model file is used in simulation and the model creation step omitted. If any parameters change, the inductor model is recreated. Because the process file name is the only reference, if you change the process file contents, the model is not updated unless you specify a new file name. Consequently, we recommend that you specify a new process file name for each new set of parameters.

Spiral Inductor Modeling under the RF Sub-Menu

You can launch the spiral inductor modeler without going into the schematic by choosing *Tools – RF – Spiral Inductor Modeler* in the Simulation window. The UI for the spiral inductor modeler using the RF sub-menu is similar to the CDF form for spiral inductors. The major difference is the addition of directory information and the design parameters at the bottom of the UI as shown in Figure 12-6 on page 554.

Figure 12-6 Design parameter display at the bottom half of the UI

The screenshot shows the 'Spiral Inductor Modeler' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. The dialog is divided into three main sections:

- DIRECTORY INFORMATION:** Contains two text input fields: 'Run Directory' and 'Model Directory'.
- MODEL GENERATION:** Contains a label 'Create Model and Display Design Params.' and a 'Run Modeler' button.
- DESIGN CHARACTERIZATION PARAMETERS (read-only):** Contains six text input fields: 'Series Resistance{Rs}', 'Series Inductance{Ls}', 'Quality Factor{Q}', 'Maximum Quality Factor{Qmax}', 'Frequency of Qmax{Fmax}', and 'Resonate Frequency{resFreq}'.

Setting Up and Running the Simulation

After you specify the physical geometries such as length, width, spacing, and frequency information, you must specify the run directory where the auxiliary files created by the modeler engine are located. You must also specify the model directory where the extracted model file is saved. You launch the modeler engine by clicking on *Run Modeler*. After the extraction, the design characterization parameters are displayed at the bottom of the form. The bottom half of the UI is shown in [Figure 12-6](#).

List of Design Parameters

In [Figure 12-6](#) on page 554, seven parameters are displayed for the spiral inductor. The working frequency mentioned below is specified in the UI. The seven parameters are

- **Series Resistance:** The AC resistance of the spiral at the working frequency with the skin-depth effect included.
- **Series inductance:** The total inductance of the spiral inductor with the skin depth effect included.
- **Quality factor:** The Q-factor at the working frequency
- **Maximum Quality factor:** The maximum quality factor of the inductor within the 10 MHz to 20 GHz frequency range.
- **Frequency of Qmax:** The frequency at which Q_{max} occurs
- **Resonant frequency:** The frequency where $Q=0$. The search is from 10 MHz to 20 GHz. If no resonant frequency is found, a value of -1 is displayed.

The calculations for these parameters are based on the equivalent circuit model. Because of the difference between the wide-bandwidth and narrow-bandwidth models, the slight difference in the design parameters between the two models is to be expected.

Including the Inductor Model Files in the Simulation

After model extraction, a netlist file that describes the equivalent circuit model of the inductor is written to the model directory you specify. You can use this file directly with the *Use external model file* option on the spiral inductor CDF form. To do this, perform the following steps:

1. In the [Simulation window](#) of the Affirma analog circuit design environment, choose *Setup – Simulation Files*.

The [Simulation Files Setup](#) form appears.

2. In the Simulation Files Setup form, specify the full path to the model definition file.

Equivalent Circuit Models

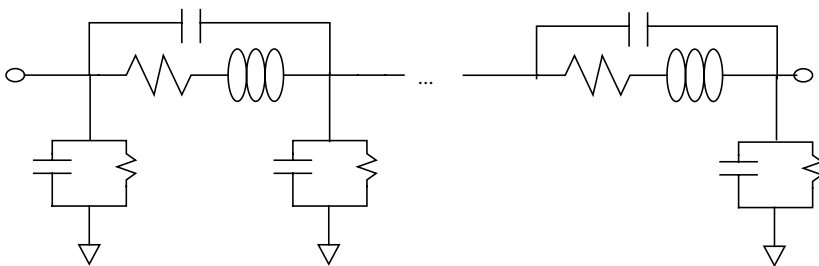
This section discusses the two types of equivalent circuit models:

- Narrow bandwidth model
- Wide bandwidth model

Narrow Bandwidth Model

This narrow bandwidth model, as shown in [Figure 12-7](#) on page 556, is constructed directly from the RLGC matrices obtained at the working frequency. All the elements are fully coupled through capacitors and inductors.

Figure 12-7 Narrow-bandwidth model for a spiral inductor



Because the RLGC parameters are functions of frequency, the above model is valid only in the vicinity of the working frequency. It is therefore a narrow-bandwidth model. Expect large simulation error for frequency components outside of the bandwidth. Because this model only requires the extraction of RLGC parameters at a single frequency, the extraction is faster. The resulting model is also guaranteed to be passive and stable.

Equivalent Circuit Order

The number of PIs in the equivalent circuit depends on the number of turns in the spiral inductor. For single-spiral layer inductors and the double-spiral layer with parallel connection inductors, the number of PIs equals the number of turns. For double spiral layer with series connection inductors, the number of PIs equals twice the number of turns.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

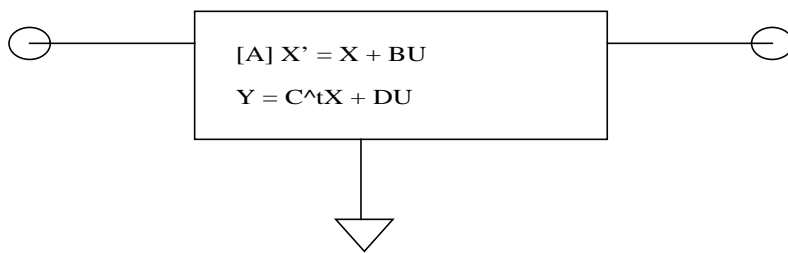
Working frequency selection and model accuracy

Because the narrow-bandwidth model is built using parameters extracted at the working frequency, the model is accurate only in the vicinity of the working frequency. The error increases as the frequency moves farther from the working frequency. Because the error is caused mostly by the large variation of the RLGC parameters with frequency, the higher the frequency, the narrower the bandwidth. If the skin depth or the substrate loss is not significant at the working frequency, the narrow-bandwidth model is rather accurate from DC up to the working frequency.

Wide Bandwidth Model

The wide-bandwidth model is designed to work over a wide frequency range. Unlike the narrow-bandwidth model, the RLCG parameters for the wide-bandwidth model are extracted over a frequency range. You specify the maximum frequency, and the minimum frequency is set to 10 MHz. Modeling the spiral inductor as a two-port system, permits the S-parameters of the two-port to be calculated at each frequency point. The resulting list of S-parameter matrices are then sent to a rational fitting module to create a wide-bandwidth nport component. [Figure 12-8](#) shows the mathematical representation of the wide bandwidth model for the inductor.

Figure 12-8 Spiral inductor wide-bandwidth model



Passivity of the Wide Bandwidth Model

The passivity of the wide-bandwidth models is tested over a frequency range by checking the energy conservation for each port at a set of sample frequencies. The default frequency range is from DC to 30GHz, which is normally wide enough for RF applications. Because the model is only guaranteed to work within this frequency range, it might not work for a simulation outside of the tested range. A mechanism is provided to change the test range.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

Set maximum Frequency for Passivity Check

You can reset the maximum frequency for the passivity checking by setting the environment variable *maxFreqForPassivityTest*.

```
setenv maxFreqForPassivityTest 50e9
```

Automatic Model Switch

To avoid simulation failure caused by non-passivity of the wide-bandwidth model, the modeler automatically switches to a narrow-bandwidth model if the passivity test fails. You can prevent this automatic switching by setting an environment variable in the window where you start *icms*.

```
setenv pcmModelAutoSwitch OFF
```

If you turn off the switch, the wide-bandwidth model is created and a warning message issued. Spectre uses the model in simulation even though it is not passive.

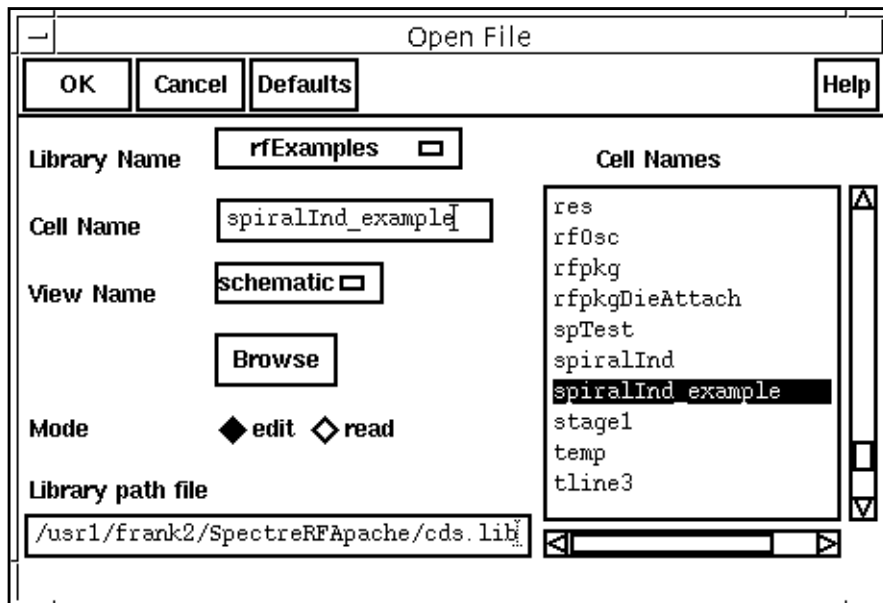
Accuracy Analysis

The *Accuracy level* entry on both the CDF and the RF sub-menu UI forms controls the mesh density for both the capacitance and inductance extraction. For capacitance, the number of subsections per width is three, five, or seven. These values correspond to the *Liberal*, *Moderate* and *Conservative* accuracy levels. Inductance calculation is similarly correlated with the accuracy level. Three, five, or seven filaments are used in the transverse direction to the current, corresponding to the three accuracy levels. The error is less than 3% at the *Liberal* level, and about 1% at the *Conservative* level.

Example

1. In the Command Interpreter Window (CIW), choose *File – Open....*

The Open File form appears.

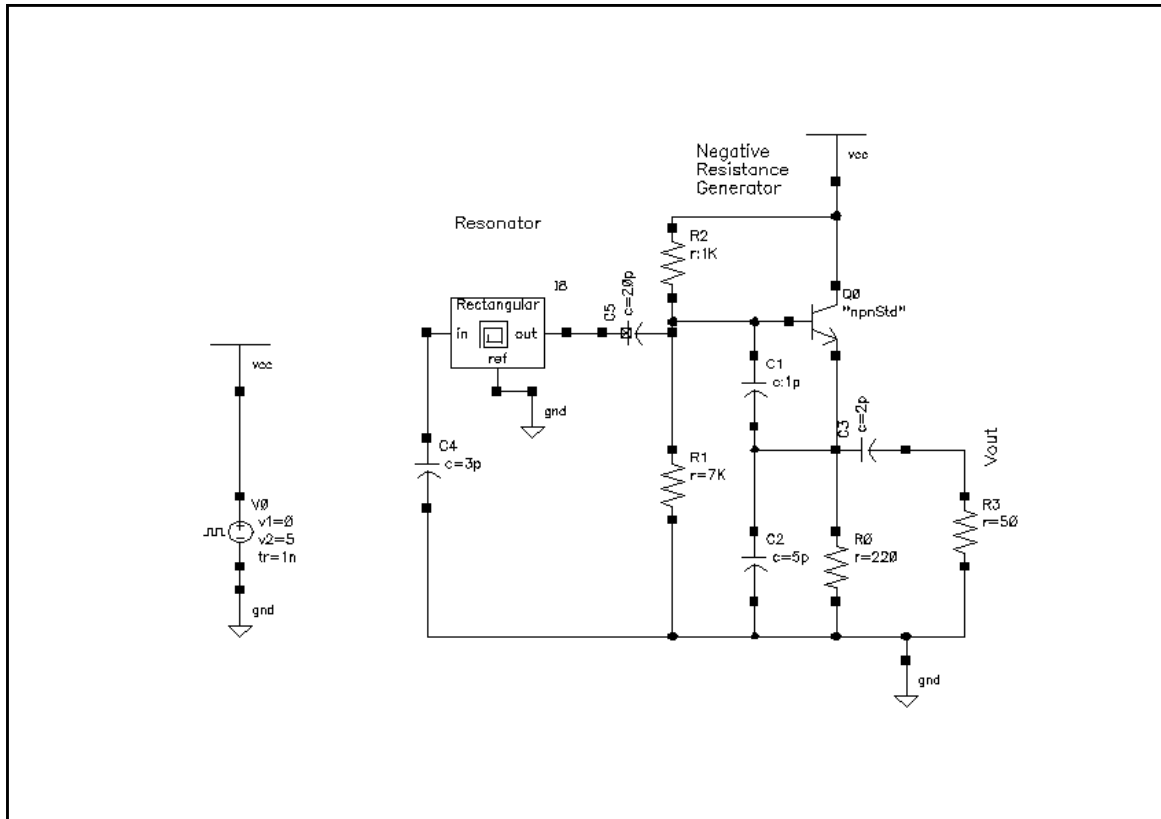


2. In the Open File form, do the following and then click *OK*:
 - a. Choose *rfExamples* for *Library Name*.
 - b. Highlight *spiralInd_example* for *Cell Name*.
 - c. Choose *schematic* for *View Name*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The *spiralInd_example* schematic appears.



3. In the Schematic window, select the Rectangular Spiral Inductor symbol and choose *Edit – Properties – Objects....*

The Edit Object Properties form appears.

4. In the Edit Object Properties form, highlight the *use external model file* button.

The form changes to let you specify the file name for the model you are about to extract.

Note: This example uses an external model file, but you could also run the example by specifying all the values in the Edit Object Properties form. To see this alternative method, look at the “LNA Example” on page 573. You can model both bondpads and spiral inductors using either method.

5. In the Edit Object Properties form, type *spIndRect* in the *Model name* field and then click *OK*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The completed form looks like this.

Edit Object Properties

OK
Cancel
Apply
Defaults
Previous
Next
Help

Apply To only current instance

Show system user CDF

Browse
Reset Instance Labels Display

Property	Value	Display
Library Name	<input type="text" value="passiveLib"/>	off <input type="checkbox"/>
Cell Name	<input type="text" value="spiralInd"/>	off <input type="checkbox"/>
View Name	<input type="text" value="symbol"/>	off <input type="checkbox"/>
Instance Name	<input type="text" value="I8"/>	off <input type="checkbox"/>

Add
Delete
Modify

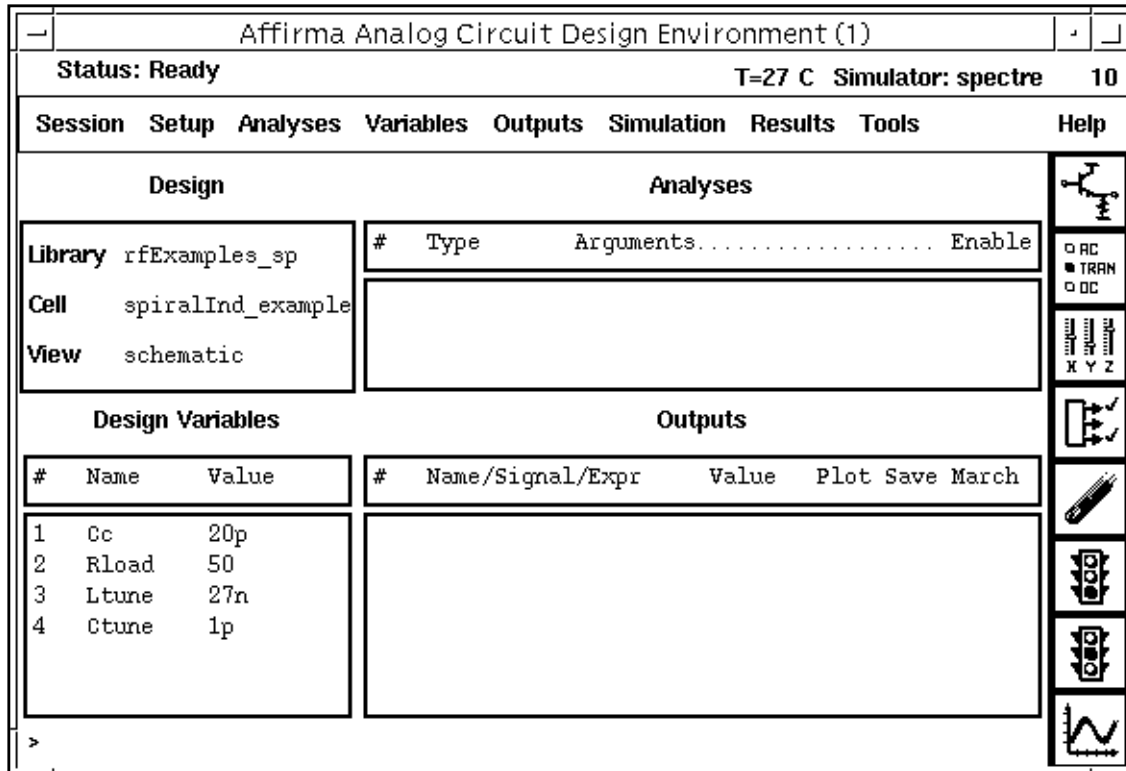
CDF Parameter	Value	Display
Use external model file	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
Model name	<input type="text" value="spIndRect"/>	off <input type="checkbox"/>
Inductor type	<input type="text" value="Rectangular"/>	off <input type="checkbox"/>

6. In the Schematic window, choose *Tools – Analog Environment....*

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The Simulation window appears.



7. In the Simulation window, choose *Setup – Model Libraries....*

The Model Library Setup form appears.

8. In the Model Library Setup form, do the following and then click *OK*.

- a. In the Model Library File field, type the path to the *npnStd.m* file and the *spIndRect.scs* file.

\$CDSHOME/tools/dfII/samples/artist/rfExamples/spiralInd_example/npnStd.m
/home/belinda/Spiral/InductorWork/spIndRect.scs

The *npnStd.m* file is in the *spiralInd_example* directory in the *rfExamples* library.

The *spIndRect.scs* file path points to the run directory where you want the extracted model file to be written.

- b. Click on *Add* after you specify each file.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The completed form looks like the one below.

Model Library File	Section
..ls/dfII/samples/artist/rfExamples/spiralInd_example/npnStd.m	
/home/belinda/SpiralInductorWork/spIndRect.scs	

Model Library File

Section (opt.)

Add Delete Change Edit File Browse...

9. In the Simulation window, choose *Tools – RF – Spiral Inductor Modeler...*

The Spiral Inductor Modeler form appears.

10. In the Spiral Inductor Modeler form, do the following:

a. Highlight *One* for *Number of metal layers*.

b. Type `metal1` for *Spiral layer #1 name*.

c. Type `metal2` for *Return layer name*.

d. Choose *Rectangular* for *Inductor type*.

e. In the *Process file name* field, type the path to the *cmos.data* file.

`$CDSHOME/tools/dfII/samples/artist/rfExamples/spiralInd_example/cmos.data`

The *cmos.data* file is in the *spiralInd_example* directory in the *rfExamples* library.

f. Type `300u` for *Outer area length(m)*.

g. Type `300u` for *Outer area width(m)*.

h. Type `10u` for *Conductor width(m)*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

- i. Type 7u for *Conductor spacing(m)*.
- j. Type 7 for *Number of turns*.
- k. Type 13u for *Return path width(m)*.
- l. Type 110u for *Return path length(m)*.
- m. Type -90 for *Return path orientation*.
- n. Choose *Clockwise* for *Polarization*.
- o. Choose *Liberal* for *Accuracy level*.
- p. In the *Model Name* field, type the name that you choose for the model, spIndRect.
- q. Highlight *Narrow* for *Model bandwidth*.
- r. Type 1.5G for *Working frequency*.
- s. In the *Run Directory* field, type the full path to the run directory.
/home/belinda/Spiral/InductorWork
- t. In the *Model Directory* field, type the full path to the model you are creating.
/home/belinda/Spiral/InductorWork

The completed form looks like the following two illustrations:

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

Top part of the Spiral Inductor Modeler Form

The image shows a screenshot of the "Spiral Inductor Modeler" dialog box. The title bar reads "Spiral Inductor Modeler". Below the title bar are buttons for "OK", "Cancel", "Defaults", "Apply", and "Help". The main area is titled "COMPONENT MODEL INFORMATION" and contains the following fields and options:

- Number of metal layers:** Three radio buttons are present: "One" (selected), "Two in series", and "Two in parallel".
- Spiral layer #1 name:** Text field containing "metal1".
- Spiral layer #2 name:** Text field (empty).
- Return layer name:** Text field containing "metal2".
- Inductor type:** A dropdown menu showing "Rectangular".
- Process file name:** Text field containing "/home/belinda/SpiralInductorWork/cmos.data".
- Outer area length(m):** Text field containing "1000j".
- Outer area width(m):** Text field containing "300j".
- Conductor width(m):** Text field containing "10j".
- Conductor spacing(m):** Text field containing "7j".
- Number of turns:** Text field containing "7".
- Number of subdivisions:** Text field (empty).
- Outer radius(m):** Text field (empty).

Affirma RF Simulator (SpectreRF) User Guide
Modeling Spiral Inductors and Bondpads

Bottom Part of the Spiral Inductor Model Form

Return path width(m)	13u
Return path length(m)	110u
Return path orientation	90
Polarization (from center)	Counterclockwise <input type="checkbox"/>
Accuracy level	Liberal <input type="checkbox"/>
Model Name	spIndRec1
Model bandwidth	<input checked="" type="radio"/> Narrow <input type="radio"/> Wide
Size/mm frequency	
Working frequency	1.5G
DIRECTORY INFORMATION	
Run Directory	/home/belinda/SpiralInductorWork
Model Directory	/home/belinda/SpiralInductorWork
MODEL GENERATION	
Create Model and Display Design Params.	Run Modeler

11. Click on the *Run Modeler* button adjacent to *Create Model and Display Design Params*.

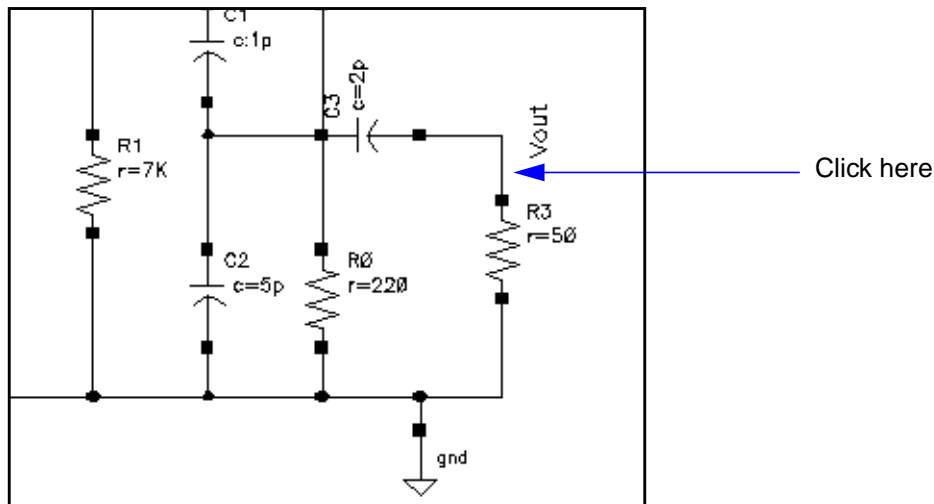
Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The model is created. When the process is completed, the calculated parameters are displayed in a read-only fields at the bottom of the Spiral Inductor Modeler form.

DESIGN CHARACTERIZATION PARAMETERS(read only)	
Series Resistance(R_s)	1.643710e+01
Series Inductance(L_s)	9.959747e-09
Quality Factor(Q)	2.543095e+00
Maximum Quality Factor(Q_{max})	2.836127e+00
Frequency of Q_{max} (f_{max})	1.130000e+09
Resonance Frequency(f_{res})	3.210000e+09

12. In the Simulation window, choose *Outputs – To Be Plotted – Select On Schematic*.
13. In the Schematic window, click on the *Vout* node.



14. In the Simulation window, choose *Analyses – Choose....*
The Choosing Analyses form appears.
15. In the Choosing Analyses form, do the following and then click *OK*:

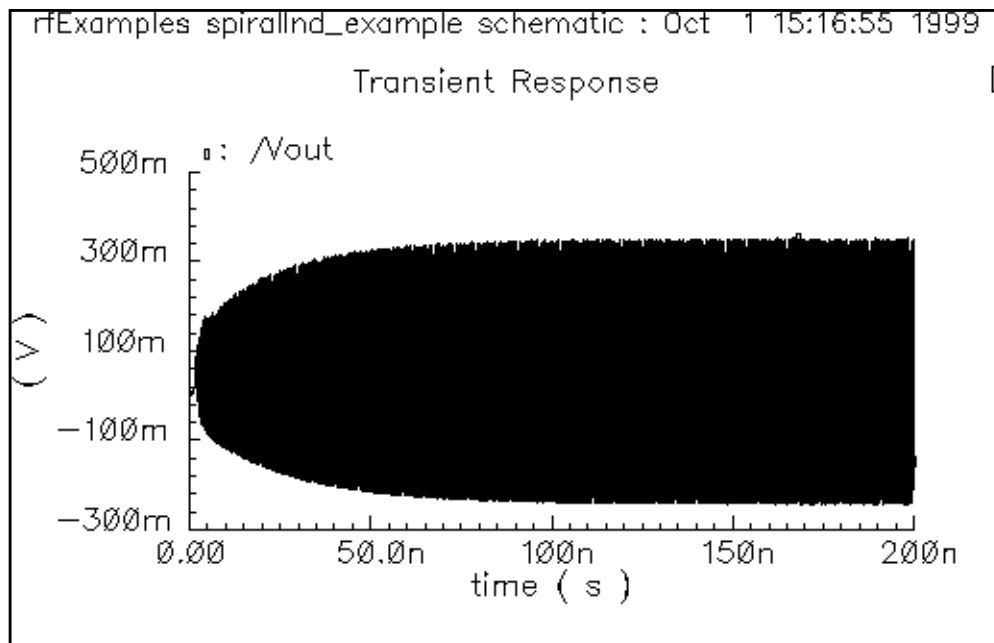
Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

- a. Highlight *tran* for *Analysis*.
- b. Type 200n for *Stop Time*.
- c. Highlight *Enabled*.

16. In the Simulation window, choose *Simulation – Netlist and Run*.

The simulation runs. When the simulation completes, the Waveform window appears with the plot for the example.



Modeling Bondpads

On-chip passive components, especially those on lossy substrate and operating at high frequencies, need careful modeling. Bonding pads are the interface between the chip and outside world, and their behavior significantly affects circuit performance. The core algorithms used in bonding pads are also used in the capacitance extraction for spiral inductor modeling.

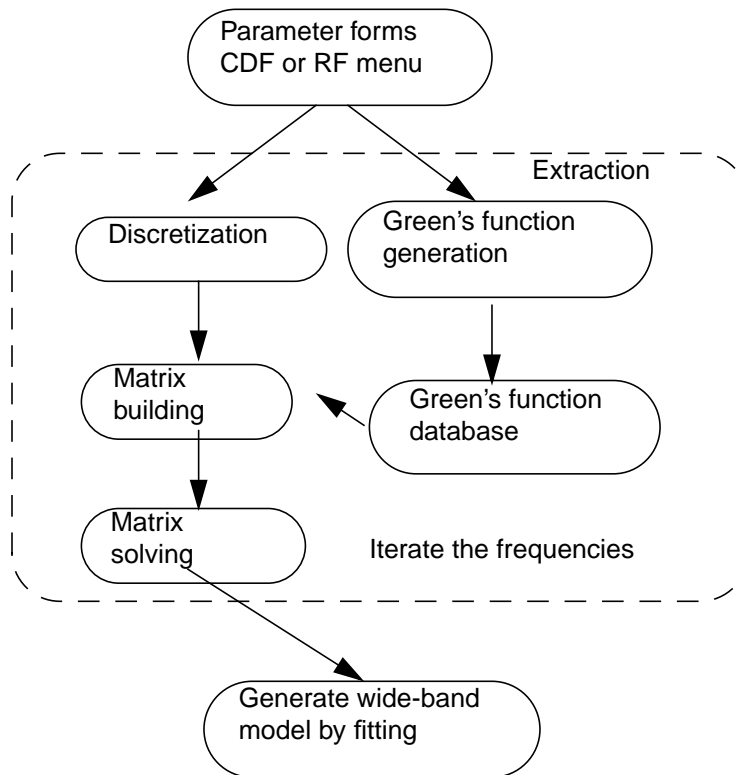
Capacitive coupling, appearing as capacitance and resistance caused by displacement current, normally is modeled in high-frequency or high-speed applications because its coupling characteristics. This displacement current is different from inductive current that are characterized by inductance. Accurate extraction of these coupling effects, however, requires solving Poisson's equation and becomes computation-intensive. For lossy substrate, when capacitive coupling and resistive coupling are of the same order of magnitude, the frequency-dependent effects require additional extraction efforts. This bonding pad model uses the complex multilayered Green's function [2] to capture capacitive couplings and loss information. It uses IES3[1] to accelerate its computation and further uses a novel technique to fit the frequency-dependent data. Because of this technique, you can perform PSS analysis as well as all the conventional circuit analyses.

You do not have to start the modeling tool manually. Instead, you place the symbol of the bonding pads into your schematic design. You can change the CDF parameters (the parameters that associate with the model) to reflect a design. If you netlist a circuit before the simulation, the extraction tool is automatically invoked and added or changed parameters are inserted into the netlist. You can also generate the model using the RF submenu in the Affirma analog circuit design environment. This approach only generates the internal model called ROM (reduced order model) that is stored in a file. You can use this file in CDF form to avoid unnecessarily generating the model during netlisting. The model is only generated during netlisting procedure when the model is new or the parameters are changed.

The current fitting algorithm still has not completely solved the passivity problem. Rarely, a non-passive model might cause simulation failure because of strong positive feedback within a circuit (such as an oscillator). When this happens, you can use a narrow-band model that is accurate at the specified working frequency.

The data flow for using and modeling the bonding pad is shown in [Figure 12-9](#) on page 570.

Figure 12-9 Data Flow for Bondpad Modeling



Input to the Modeler

Initially, you must create a process file to describe the vertical profile of the process. These parameters will be used in extraction. The file format is the same as that used for spiral inductors. For more information about creating the file, see the [Process File Preparation](#) section in this chapter.

Parameter Sets

[Figure 12-10](#) on page 571 shows the input form for the bonding pads.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

Figure 12-10 Component Description Format (CDF) for bonding pad model

Property	Value	Display
Library Name	passiveLib	off <input type="checkbox"/>
Cell Name	bondpad	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	I3	off <input type="checkbox"/>

CDF Parameter	Value	Display
Use external model file	<input type="checkbox"/>	off <input type="checkbox"/>
Pad shape	Octagonal <input type="checkbox"/>	off <input type="checkbox"/>
Process file name	./model/cmos.data	off <input type="checkbox"/>
Number of pads	4	off <input type="checkbox"/>
Pad spacing(m)	200u	off <input type="checkbox"/>
Pad diameter(m)	100u	off <input type="checkbox"/>
Pad layer name	metal1	off <input type="checkbox"/>
Maximum frequency	6Ghz	off <input type="checkbox"/>
Working frequency		off <input type="checkbox"/>

You can specify data in either the CDF form from the schematic or using the Tools – RF submenu in the Simulation window. For more information about the procedure using the RF submenu, see “Spiral Inductor Modeling under the RF Sub-Menu” on page 554. The bondpad procedures are similar to those for spiral inductors. The libraries that hold this model and the model for spiral inductors are under the Cadence dfll hierarchy in *tools/dfll/samples/artist/passiveLib*. The parameters you enter are described below:

- **Use external model file** lets you use externally-generated bondpad model files directly in simulations

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

- **Process file name** can be either an absolute file path or a relative path to the current directory.
- **Pad shape** can be square or octagonal. When this cyclic button is changed, the symbol shape changes accordingly.
- **Pad spacing** is the distance between the centers of two adjacent pads. Remember that this is not the spacing between the adjacent edges.
- **Pad diameter** is the diameter size of the single bonding pad. Remember that it is not the size of the edges.
- **Pad layer name** must have its corresponding layer name defined in the process file.
- **Number of pads** lets you specify the number of pads.
- **Maximum Frequency and Working frequency** – There are two types of models available: wide-band and narrow-band. When you specify the maximum frequency, a wide-band model accurate up to the maximum frequency is generated. The default maximum frequency is 10GHz.

When you supply the working frequency and also omit the maximum frequency, the modeler generates a narrow-band model that is accurate at the working frequency.

Modeling Issues

This section explains modeling issues related to the internal workings of the modeler and the generated models.

Wide-Band and Narrow-Band Models

Models are generated either as wide-band or narrow-band models. In most cases, you use the wide-band model because it is more accurate for time-domain simulation.

The wide-band model is generated after fitting the charge-voltage relationships over a frequency range from DC to the maximum frequency. The model is created as a ROM (reduced order model) file. The netlisting code generates an nport, whose *romdatfile* is set to the ROM file to represent bonding-pads.

The narrow-band model only extracts the model exactly at the working frequency. An RC network is derived from that extraction, and it is hoped that this model is accurate for a reasonable frequency range around the working frequency. This RC network is also represented internally as an nport.

Passivity Issues and Use of the Narrow-Band Model

The wide-band model is fitted to a sample of frequencies. For typical RF applications with operating frequency of 1-2GHz, a maximum frequency of 10GHz covers sufficient frequency content and the model is generally very satisfactory. However, the fitting algorithm might produce an unstable model at frequencies higher than the maximum frequency. This is true because the model is extrapolating at these frequencies. This possible nonpassivity has no impact on S-parameter small-signal analysis. It also does not harm simulation for most RF circuits in transient and PSS analyses. It might, however, cause simulation to diverge when it is within a feedback loop of an unstable circuit, such as an oscillator.

You can use the narrow-band model if the wide-band model does not converge. Mathematically, the narrow-band model is an RC network that is guaranteed to be stable and passive.

Noise Modeling

Currently, the bonding pad model does not model noise effects.

LNA Example

RF circuits operating at high frequencies are more susceptible to bonding pad effects. Using the process file in “[Process File Preparation](#)” on page 545, the series resistance of the bonding pad at 2GHz is about 450 Ohm and the series capacitance is 0.133p. These values are very similar to those in the literature[3].

This example shows the effects of the bonding pads in the following LNA circuit. The octagonal pad diameter is 100u and the spacing is 200u. It uses the process file described in “[Process File Preparation](#)” on page 545. A two-pad model is used, and the coupling effects are included.

The example runs an S-parameter analysis without the bondpad and plots the results. The bondpad is then added to the schematic, the simulation is rerun, and the results are compared.

Setting Up and Running the S-Parameter Simulation

1. In the [CIW](#), choose *File – Open...*

The [Open File](#) form appears.

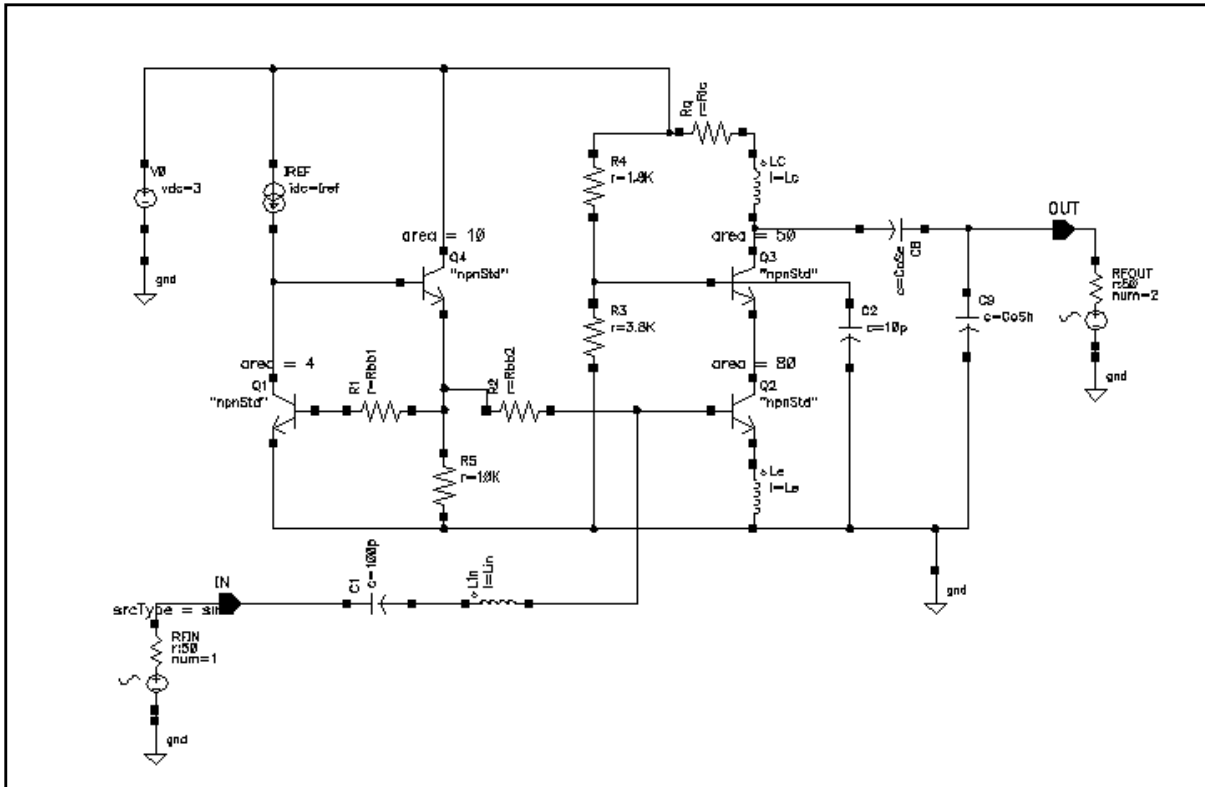
2. In the Open File form, do the following and click *OK*:

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

- a. Choose *rfExamples* for *Library Name*.
- b. Highlight *Ina-pad* for *Cell Name*.
- c. Choose *schematic* for *View Name*.

The schematic appears



3. In the Schematic window, choose *Tools – Analog Environment*.

The Simulation window appears.

4. In the Simulation window, choose *Setup – Model Libraries....*

The Model Library Setup form appears.

5. In the Model Library Setup form, do the following:
 - a. In the *Model Library File* field, type the complete path to the *npnStd.scs* file
 - b. Click on *Add*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The completed form looks like this;

spectre0: Model Library Setup

OK Cancel Defaults Apply Help

Model Library File Section

..\..\js\CADENCE/projects/pcm/appnote/testcircuit/model/npnStd.scs

Model Library File Section (opt.)

Add Delete Change Edit File Browse...

c. Click OK.

6. In the Simulation window, choose *Analyses – Choose....*

The Choosing Analyses form appears.

7. In the Choosing Analyses form, do the following and then click *OK*:

- a. Highlight *sp* for *Analysis*.
- b. Highlight *Frequency* for *Sweep Variable*.
- c. Highlight *Start – Stop* for *Sweep Range*.
- d. Type 100M for *Start* and 5G for *Stop*.
- e. Choose *Linear* for *Sweep Type*.
- f. Highlight *Number of Steps* and type 50 in the adjacent field.
- g. Highlight *no* for *Do Noise*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The completed form looks like the this.

Choosing Analyses -- Affirma Analog Circuit Design Env

OK Cancel Defaults Apply Help

Analysis tran dc ac noise xf
 sens sp pdisto pss pac
 pnoise pxf envlp

S-Parameter Analysis

Sweep Variable
 Frequency
 Design Variable
 Temperature
 Component Parameter
 Model Parameter

Sweep Range
 Start-Stop Start Stop
 Center-Span

Sweep Type

 Step Size
 Number of Steps

Add Specific Points

Do Noise
 yes
 no

Enabled Options...

8. In the Simulation window, choose *Simulation – Netlist and Run*.

Be sure to check the CIW for a message that the simulation completed successfully.

Plotting the Results

1. In the Simulation window, choose *Results – Direct Plot – S-Parameter....*

The S-Parameter Results form appears.

2. In the S-Parameter Results form, do the following:

- a. Highlight *Replace* for *Plot Mode*.
- b. Highlight *SP* for *Function*.
- c. Highlight *Rectangular* for *Plot Type*.
- d. Highlight *Magnitude* for *Modifier*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The completed form looks like this.

S-Parameter Results

OK Cancel Help

Plot Mode Append Replace

Function

SP ZP YP HP
 GD VSWR NFmin Gmin
 Rn m NF Kf
 B1f GT GA GP
 Gmax Gmsg Gumx ZM
 NC GAC GPC LSB
 SSB

Description: S-Parameter

Plot Type Auto Rectangular
 Z-Smith Y-Smith
 Polar

Modifier

Magnitude Phase dB20
 Real Imaginary

S11 S12
S21 S22

Add To Outputs

> Press Sij-button to plot...

- a. Click on **S21**.

The plot appears in the Waveform window. Leave all the windows up on your screen so you can resimulate and append the simulation results with the bondpad to the current plot.

Setting Up and Simulating with the Bondpad

1. In the Schematic window, choose *Add – Instance....*

The Add Instance form appears.

2. In the Add Instance form, click on *Browse*.

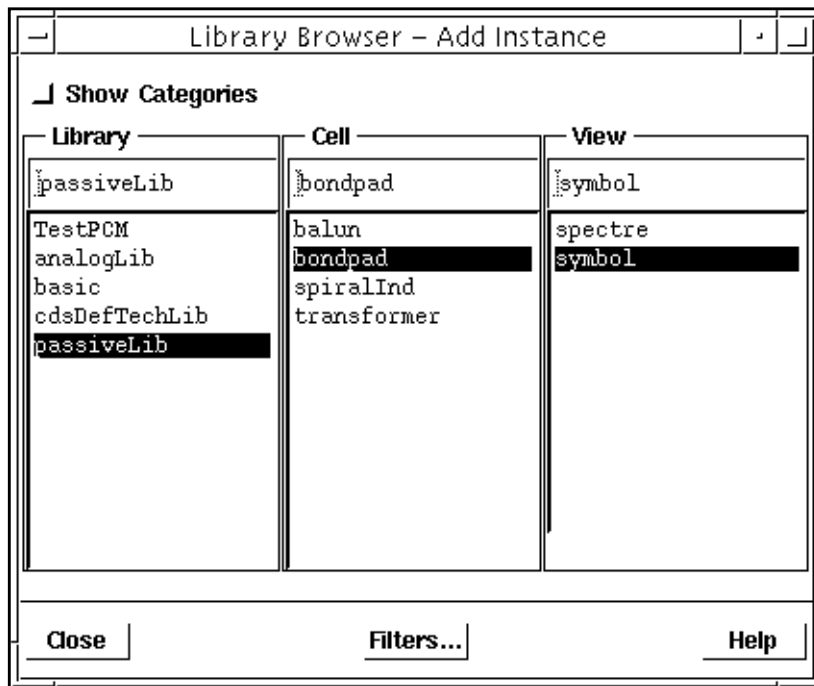
The Library Browser – Add Instance form appears.

3. In the Library Browser – Add Instance form, do the following:

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

- a. Highlight *passiveLib* for *Library*.
- b. Highlight *bondpad* for *Cell*.
- c. Highlight *symbol* for *View*



The Add Instance form changes to let you specify values for the bondpad.

4. In the Add Instance form, do the following:
 - a. Choose *Octagonal* for *Pad shape*.
 - b. Type the complete path to the process file in *Process file name*.
 - c. Type 2 for *Number of pads*.
 - d. Type 200u for *Pad spacing(m)*.
 - e. Type 100u for *Pad diameter(m)*.
 - f. Type `metal1` for *Pad layer name*.
 - g. Type 6G for *Maximum frequency*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The completed form looks like this.

Add Instance	
Hide	Cancel
Defaults	Help
Names	
Array	Rows: 1 Columns: 1
Rotate	Sideways
	Upside Down
Use external model file	<input type="checkbox"/>
Pad shape	Octagonal
Process file name	./model/cmos.data
Number of pads	2
Pad spacing(m)	200
Pad diameter(m)	100
Pad layer name	metal1
Maximum frequency	60
Working frequency	

An instance of the bondpad is now attached to the cursor if you place the cursor in the Schematic window.

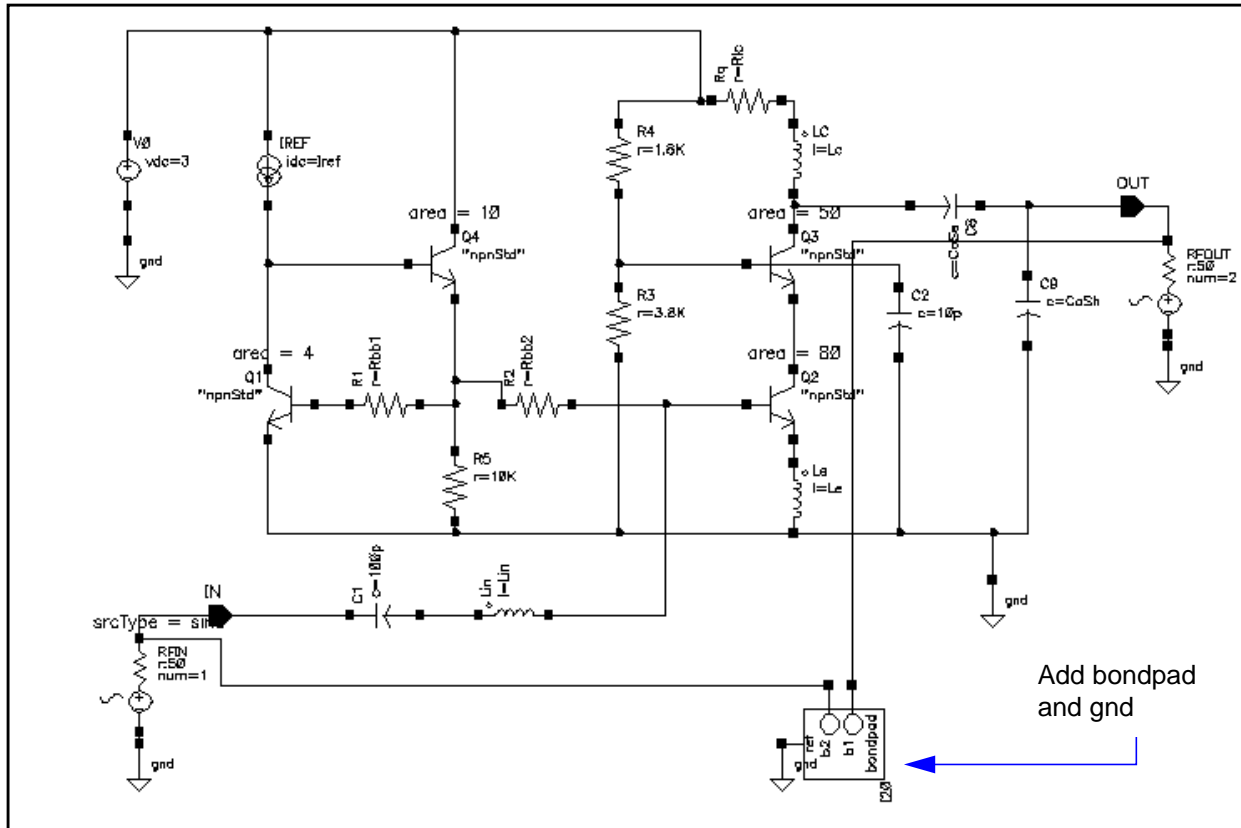
5. In the Schematic window, do the following:
 - a. Place an instance of the bondpad in the Schematic window.
 - b. Make a copy of one of the existing ground symbols in the schematic.

Note: If you need assistance with techniques for editing a schematic, see the Virtuoso® Schematic Composer User Guide.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

6. In the Schematic window, add wires to connect the components so the final schematic looks like this.

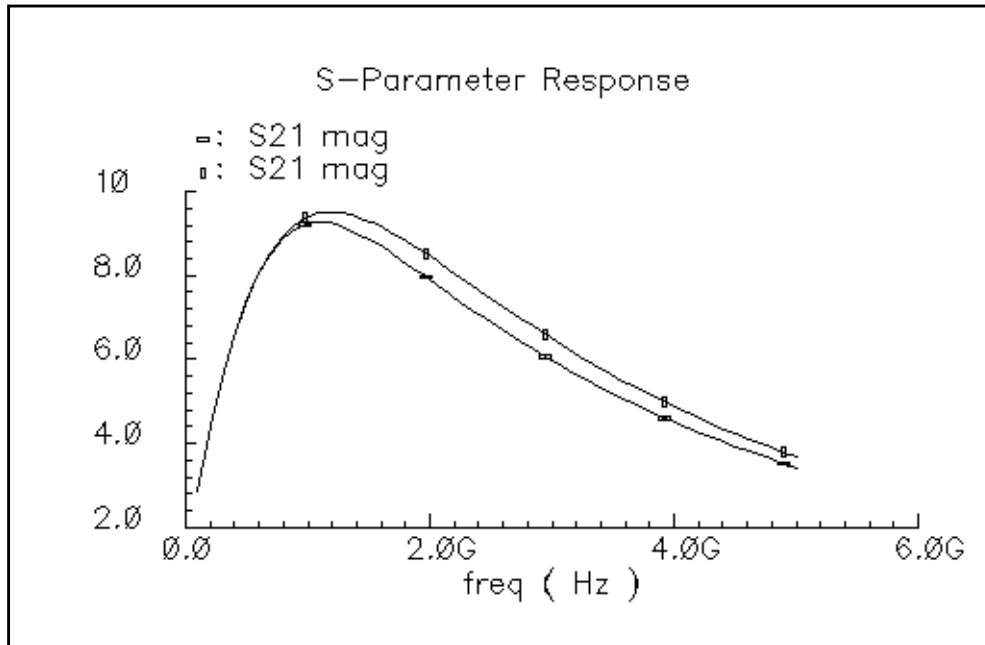


7. In the Schematic window, choose *Design – Check and Save*.
8. In the Simulation window, choose *Simulation – Netlist and Run*.
The simulation is rerun with the bondpad you added to the schematic.
9. In the S-Parameter Results form, do the following:
 - a. Highlight *Append* for *Plot Mode*.
 - b. Click on *S21*.

Affirma RF Simulator (SpectreRF) User Guide

Modeling Spiral Inductors and Bondpads

The new plot is appended to the Waveform window and the window now looks like this.



The waveform plot shows the comparative results. At 2GHz, the difference of gains is about 8% and the effects are more prominent when frequency is high. Also, the peak gain, tuned by the LC matching at the input, is shifted higher from 1GHz because of the capacitive effects of the input pad.

References for the Bondpad Section

[1] Jinsong Zhao, Wayne Dai, Sharad Kapur, David Long, "Efficient Three-Dimensional Extraction Based on Static and Full-Wave Layered Green's Functions", 35th Design Automation Conference.

[2] Sharad Kapur, David Long, "IES3: A Fast Integral Equation Solver for Efficient 3-Dimensional Extractions", 37th International Conference on Computer Aided Design.

[3] Bonding pad models for silicon VLSI technologies and their effects on the noise figure of RF NPNs, Natalio Camilleri etc, pp. 225-228, IEEE 1994 Microwave and Millimeter-wave Monolithic Circuits Symposium.

Behavioral Models and Methods for Top Down Design

The models and methods described in this document are for designing an analog RF subsystem that fits into a larger DSP system. In particular, this document describes a new library for Analog Artist. The library is a canonical set of top-down behavioral baseband models for exploring RF architectures in Analog Artist. The models have passband views for mapping DSP measurements into RF measurements.

This library addresses the needs of RF designers who require a faster way to map system specifications into detailed designs. The baseband models facilitate fast evaluations of candidate RF architectures with respect to DSP metrics. The passband views of the models provide a behavioral system testbench for checking detailed designs of individual system components.

Baseband models are behavioral models and all behavioral models sacrifice some accuracy for lots of speed. Such sacrifices are usually acceptable in architectural studies because many implementation-dependent details do not affect high level decisions. The modeling approach taken in top-down design is to simulate only those effects that drive the decisions at hand.

Baseband modeling in no way replaces passband modeling. Some effects missed by equivalent baseband models can affect high level decisions. However, the application of baseband models early and passband models later minimizes the number of slow simulations needed at the lower levels of design abstraction; Baseband models quickly weed out designs that would surely fail tests simulated with passband models.

The success of a modeling approach to top-down design hinges on knowing how the models fit into the design flow and knowing exactly what each modeling parameter means. This chapter has two goals:

- To describe the design flow from a modeling perspective
- To define, as clearly and concisely as possible, the parameters that specify the models

Top Down Design of RF Systems

Ideally, the digital signal processing, or DSP, team selects an RF subsystem that fits snugly into the DSP system. A *snug fit* means that the RF subsystem does exactly what it needs to do for the least general cost and it is described in full detail in a detailed design specification that describes how to build the RF subsystem.

Everything is known about a detailed design. It is the lowest level of abstraction. You can extract behavioral models of a detailed design from simulated measurements. The problem is that detailed designs usually do not exist until the project is complete. To jump directly to a detailed design implies that the design flow is bottom up. Bottom up flows are important in many projects but not in all.

In a top down design flow, the DSP team writes a functional specification describing an RF subsystem that has not yet been designed. The functional specification describes what the RF subsystem must do, as well as what it must not do. It does not describe how to build the RF subsystem.

The functional specification is the highest level of abstraction. Behavioral models created at this level of abstraction are specified, not measured. At this point, the functional specification might be incomplete or inconsistent. A good top down design flow detects such problems as early in the design cycle as possible because problems detected late in the design cycle can be extremely costly to resolve.

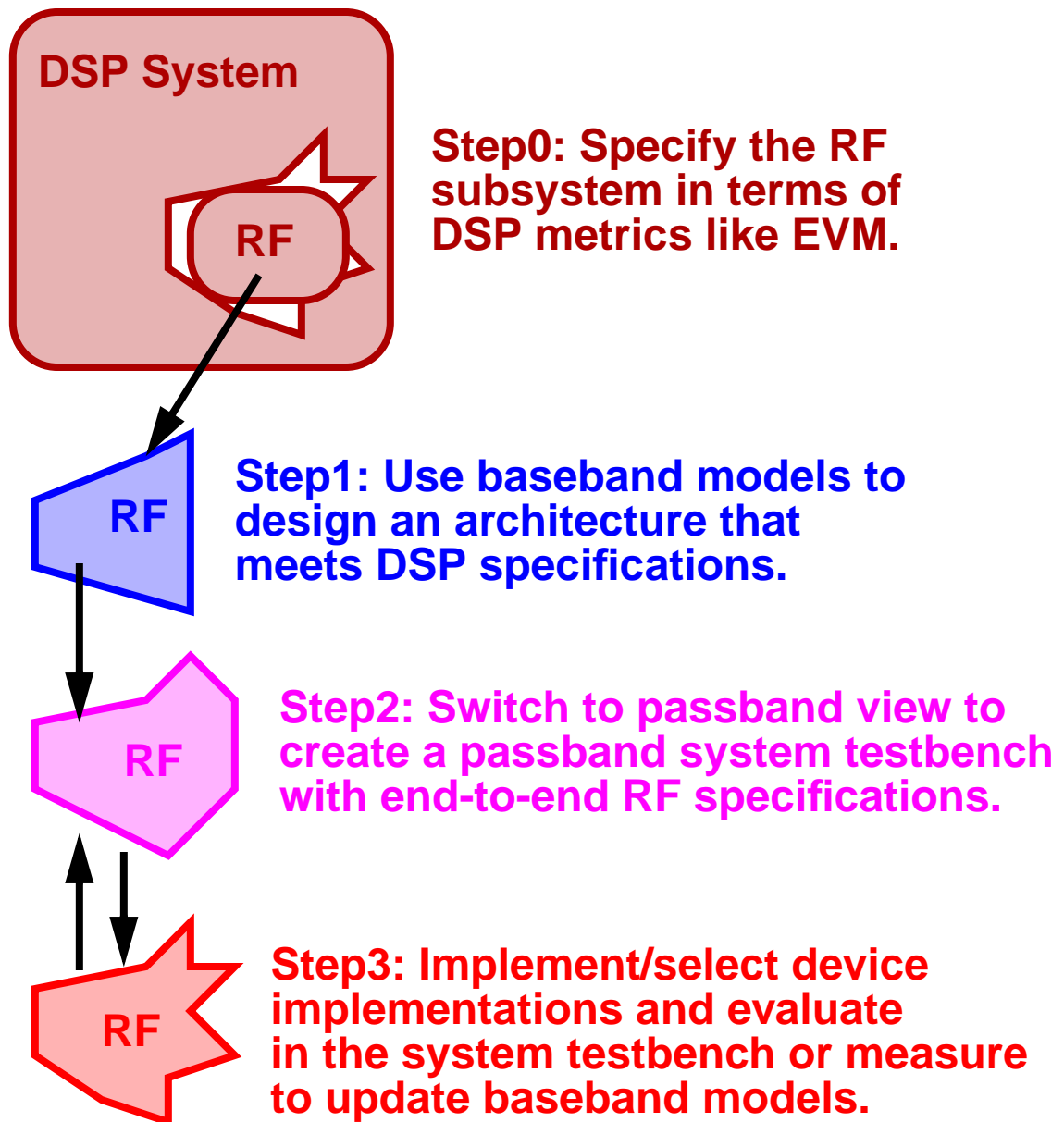
DSP and RF engineers sometimes have trouble communicating through specifications because the two groups deal with different metrics. For example, DSP engineers deal with *bit error rates* and *error vector magnitude statistics* whereas RF engineers deal with *intercept points* and *noise figures*.

The new models described here are designed to help RF system engineers in two ways.

- First, the models enable the RF system designers quickly explore the RF architectural space, as specified by the DSP metrics, while letting the RF engineers specify the RF system components with RF metrics. The circuit implementations of the RF system components are easier to design and test when the components are specified with standard RF metrics.
- Second, the models can be switched quickly to a different view where the RF system model can generate end-to-end RF metrics. With end-to-end metrics, the new view can quickly simulate how the detailed design of a particular RF system component affects end-to-end performance.

Figure 13-1 on page 584 summarizes the top down design flow.

Figure 13-1 The Proposed Top Down Design Flow and Use Model



The following steps outline the RF design process with focus on the early phases of the design.

Step 0: Specify the RF subsystem in terms of DSP metrics.

This is called step 0 because it comes before the RF subsystem design flow but it must be completed. This step distinguishes the end of the DSP design flow from the beginning of the RF design flow.

Step 1: Explore Candidate Architectures with Baseband Models

The first step in top down RF design is to select a candidate architecture. An architecture is a set of interconnected RF function blocks that, taken together, describe how a receiver or transmitter operates. The function blocks are specified in terms of standard RF metrics, for example, IP3, gain, bandwidth, and noise figure.

The models you use early in the design cycle as you explore RF architectures must run fast. Each simulation spans hundreds of symbols and each symbol can easily span thousands of RF carrier cycles. The space defined by all function block specifications in the architecture is far too vast to explore with slow (maximum precision) models. Models for architectural explorations must quickly reduce the design space down to a size that can be explored with more precision.

The most efficient models for architectural exploration suppress the RF (IF) carrier and are called *baseband* models. In contrast, the *passband* models introduced in Step 2 do not suppress carriers.

You can use the Circuit Optimizer during the exploration step to help balance the function block specifications for a candidate architecture. For example, the Circuit Optimizer can minimize RMS EVM while ensuring other measurements stay within acceptable limits.

After you determine the nominal specifications for each function block within the architecture you must put tolerances around them. In the analog world *specifications without tolerances are meaningless*. The tolerance space is usually explored with some mix of experience, feasibility, various analyses, and outright arbitrary decisions.

The baseband models can test candidate tolerances as well as help to determine some tolerances analytically. One way to test a candidate set of tolerances is to run a Monte Carlo analysis on the metric of interest, like RMS EVM or signal-to-noise ratio (SNR). Another approach is to use the Circuit Optimizer *in reverse*, as a de-optimizer, to determine worst case performance. Yet another approach is to compute each tolerance separately from a parametric plot. If all but 2 or 3 tolerances remain to be found, a multidimensional parametric analysis can map out the performance space for easy identification of the necessary tolerances.

Step 2: Switch to Passband Models and Create an RF System Testbench

The second step is to create a passband view of the system model. The passband view both confirms that the filters perform as expected and also creates an end-to-end testbench that you can use to design the individual function blocks. For computational efficiency, at any one time only one or two selected function blocks are modeled at the device level for system passband testing. All other blocks in the system are modeled behaviorally but with passband models.

It is not practical to use the passband view to assess DSP metrics because Spectre analysis would take too long. You also cannot use SpectreRF to reduce run times because the instrumentation blocks have internal state variables and SpectreRF does not allow state variables inside behavioral models.

However, you can apply SpectreRF to the passband behavioral view to derive end-to-end RF specifications if you do it before inserting any device-level models into the testbench. You derive the tolerances by performing the same Monte Carlo analysis or Circuit Optimizer analysis you used to test the function block tolerances in Step 1, but this time you replace the DSP metrics with end-to-end RF metrics. Once you know how far the end-to-end RF metrics can vary, you can insert a device-level model of a function block into the testbench to see how close it drives the system toward violating a derived end-to-end RF specification.

Step 3: Implement the Function Blocks With Active and Passive Devices

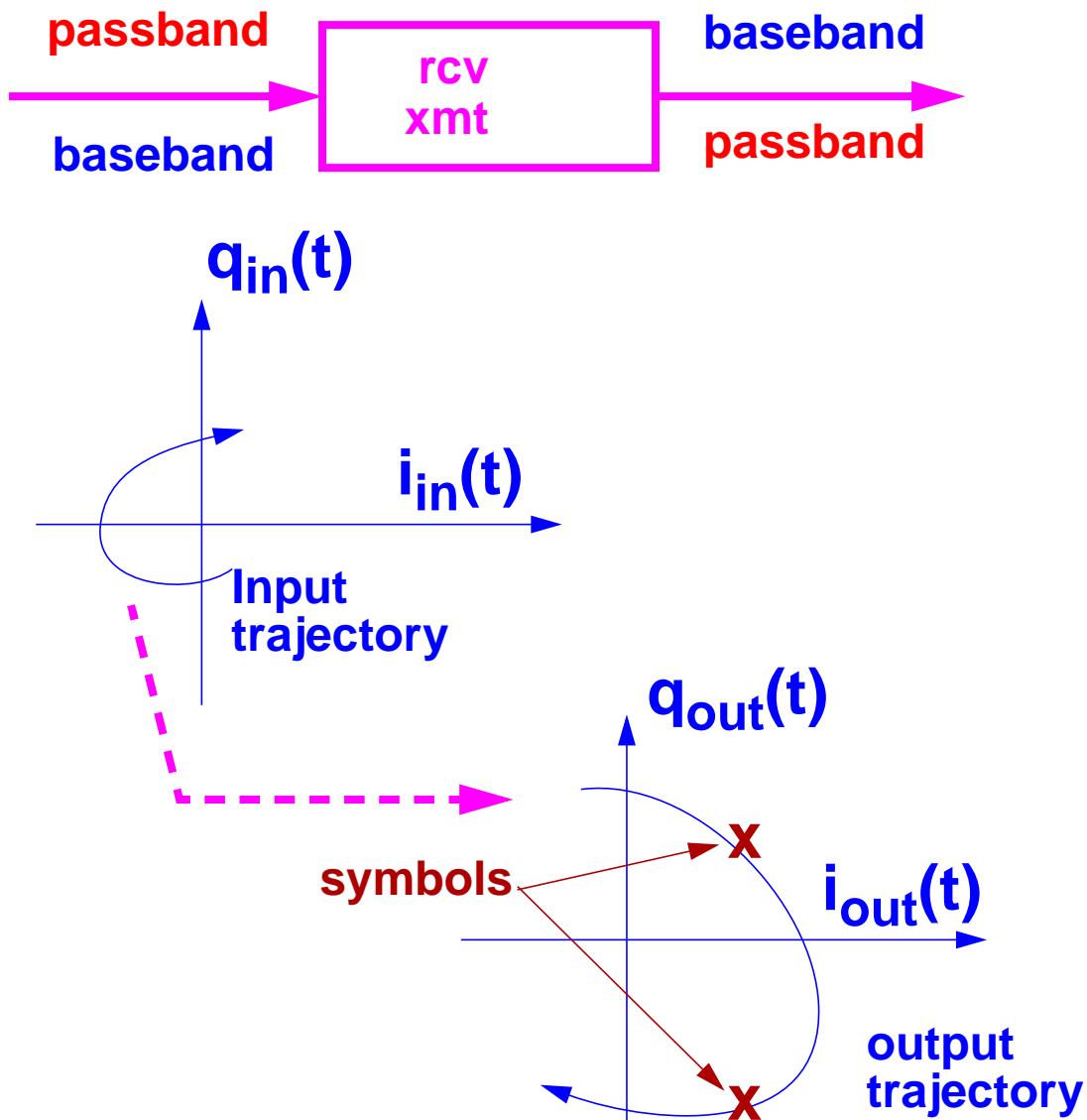
The last step in the top down design process is to implement the function blocks with device models. Since the function blocks are specified in terms of standard RF metrics, you can easily measure the modeling parameters to make sure they fall within the specified tolerances. You can also insert the measured parameters back into the baseband model of the system to check the DSP metrics, or insert the device-level model directly into the passband testbench to check the derived end-to-end RF specifications.

Baseband Modeling

Figure 13-2 on page 587 mathematically defines a baseband representation of a passband signal. The i and q signals are the real and imaginary parts of a complex signal that rides on the two phases of an RF carrier.

Figure 13-2 Baseband Representation and Models

passband signal = $i(t)\cos(\omega_{rf}t) - q(t)\sin(\omega_{rf}t)$ = real
baseband representation = $i(t) + j*q(t)$ = complex



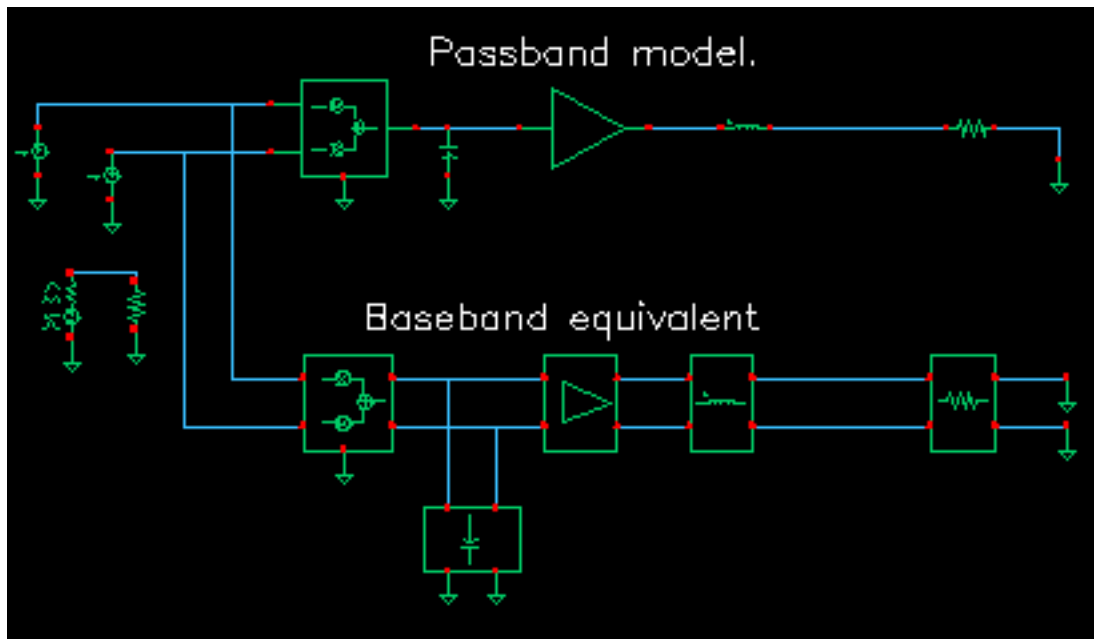
A baseband model for an RF function block simulates what happens to the baseband representation of a signal as it passes through the block. A baseband model maps input

baseband signal trajectories into output baseband signal trajectories. If a baseband signal is sampled periodically in time, and the samples are plotted in the complex plane, the resulting scatter plot shows the symbol constellation.

Baseband models simulate only what happens to the carrier fundamental. Consequently, they only account for non-linearities with odd symmetry. Non-linearities with even symmetry produce no output at the carrier fundamental; they affect the carrier fundamental only when cascaded. For example, a second order non-linearity in one block can create a DC offset at its output. Upon passing through a subsequent block with another second order non-linearity, the DC offset can mix with the carrier to affect the output carrier fundamental. Cascaded blocks producing unfiltered even harmonics should be modeled as a single baseband model rather than as separate baseband models cascaded together. The non-linearities that most often dominate performance have odd symmetry.

Figure 13-3 on page 588 shows a schematic that illustrates the difference between passband and baseband modeling. This circuit, which is called *BB_test_bench*, is located in the *rfExamples* library. (The Figure does not show the notes in the schematic. Those notes are outdated since there are no instrument states to recall.)

Figure 13-3 *BB_test_bench* Circuit



The *BB_test_bench* circuit shows a passband circuit and its baseband equivalent model. The same baseband signals drive both circuits but only in the passband circuit are the baseband signals first mixed up to RF. The power amplifier is not matched to either input or output impedances and both impedances are reactive.

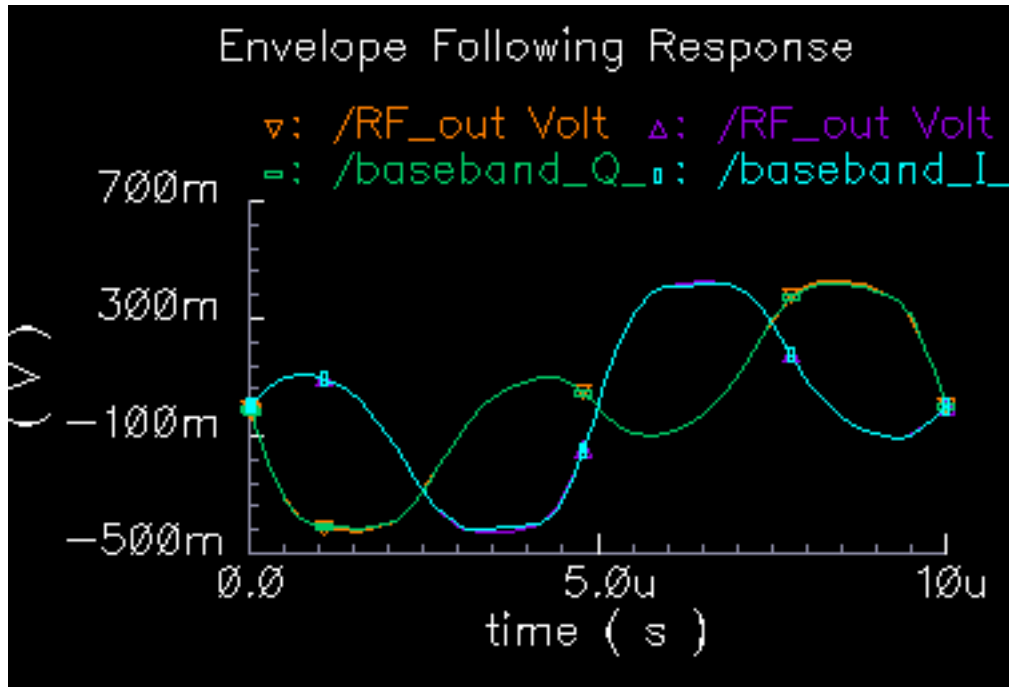
Plotting Baseband Equivalent Output Signals for Baseband and Passband Circuits

The following steps show the relationship between baseband and passband models by walking you through a single Envelope Analysis. Use the following procedure to plot the baseband equivalent output signals as computed by the baseband and passband circuits.

1. Recall the *rfExamples/BB_test_bench* schematic.
2. Start up the Analog Artist window from the schematic.
3. Select *Analysis-Choose* and press the *envlp* button.
4. Select *ff* as the clock name, set the *Stop time* to 10 μ and set the *number of harmonics* to 1.
5. Run the analysis.
6. Select *Results-Direct Plot-ENVLP*.
7. Press the *Replace*, *Voltage*, and *time* buttons.
8. In the schematic, click on the *baseband_I_out* node.
9. In the ENVLP window, press the *Append* button.
10. In the schematic window, select the *baseband_Q_out* node.
11. In the ENVLP window, press the *harmonic time* button, *real* button, and select 1 as the *harmonic number*.
12. In the schematic, select the *RF_out* node.
13. In the ENVLP window, press the *Imaginary* button.
14. In the schematic window, select the *RF_out* node again.

In the waveform display window you should now see two traces as shown in [Figure 13-4](#) on page 590. If you look more closely, you should see that each trace is actually two traces, one nearly on top of the other, making a total of four traces.

Figure 13-4 Baseband Representations and Models



About Baseband Modeling

The above example illustrated how baseband modeling corresponds to the time-varying fundamental Fourier component computed by Envelope Following analysis.

- Why use baseband models if Envelope Following analysis gives the same results?

If you de-activate the passband circuit by setting the *carrier_pb* variable to zero in the Analog Artist window, disable the Envelope Following analysis, and run a 10 μ s transient analysis, you will observe the same baseband results but the simulation will run over 100 times faster.

- Why not use baseband models all the time?

If you look closely at both waveforms you will notice that the baseband waveforms clip at a slightly lower level than the Envelope Following waveforms. This is because hard limiting of the carrier generates higher-than-third-order harmonics and the behavioral baseband model only simulates third order non-linearities.

Library Overview

The new library includes three kinds of models:

- Instrumentation Models
- Non-Linear Memoryless Models
- Linear Models With Memory

The instrumentation models provide stimuli, diagnostics, and performance metrics relevant to the DSP system.

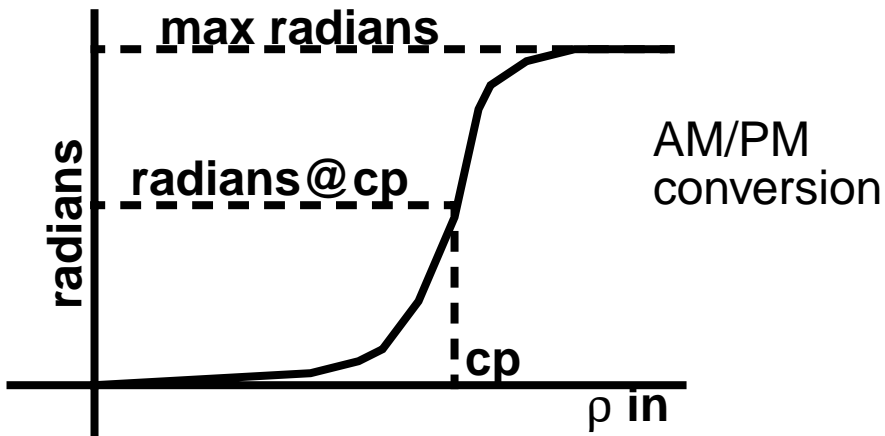
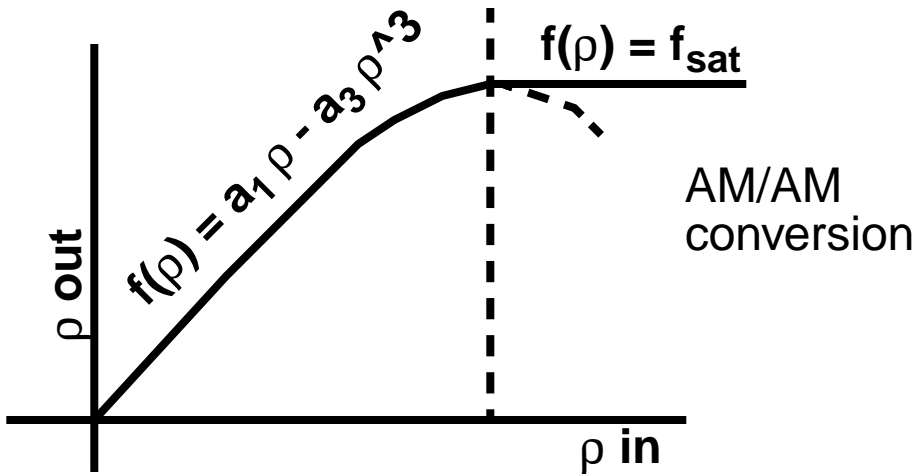
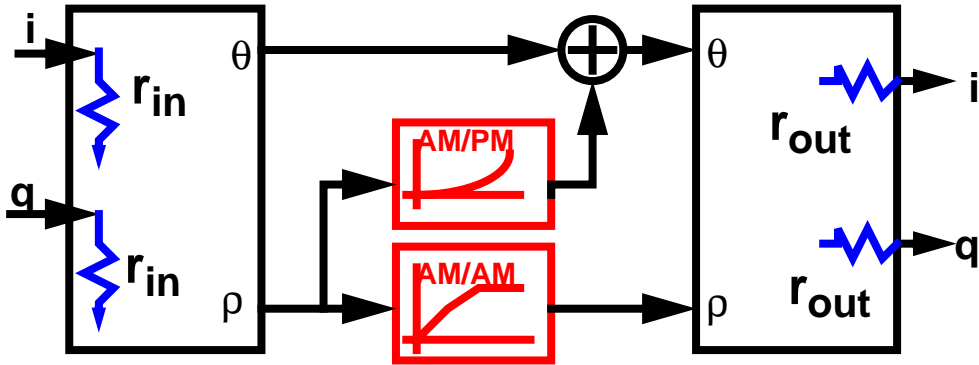
Both the non-linear and linear models simulate the function blocks within an RF architecture and are specified in terms of common RF metrics. The function block models all include input referenced white Gaussian noise as specified by noise figure. Models are provided for the following types of function blocks: amplifiers, mixers, filters¹, and phase shifters.

The non-linear models simulate AM/AM conversion [1] with a third-order polynomial that saturates at the peak of the transfer curve. The polynomial is specified by the gain and either the input-referred IP3 or output-referred 1 dB compression point. Only the baseband non-linear models simulate AM/PM conversion. AM/PM conversion [1] is an important effect that is hard, if not impossible, to simulate with passband behavioral models.

Figure 13-5 on page 592 shows the basic baseband non-linearity.

1. Here, *filters* includes single resistors, capacitors, and inductors.

Figure 13-5 Basic, Baseband Non-Linearity



The baseband models of reactive elements differentiate our approach from spreadsheet-based approaches to RF system design. The baseband capacitor and inductor models let you simulate reactive loading effects in the time domain, where non-linearities are more naturally modeled. In RF integrated circuits, loading effects are important because it is often hard to integrate impedance matching networks.

The baseband models of reactive elements also play a key role in modeling filters. Most digital communications text books [1,2] explain that you can model a passband transfer function at baseband by simply frequency-shifting the transfer function. What these books do not describe is how to implement the resulting transfer function in a general circuit simulator such as Spectre. The shifted transfer function usually lacks complex conjugate symmetry about zero frequency and therefore has a complex impulse response.

The first consequence of modeling RF function blocks at baseband is that all equivalent baseband models have four terminals instead of two:

- One set of terminals represents in-phase signals, $i(t)$ in [Figure 13-2](#) on page 587.
- The other set of terminals represents quadrature signals, $q(t)$ in [Figure 13-2](#) on page 587.

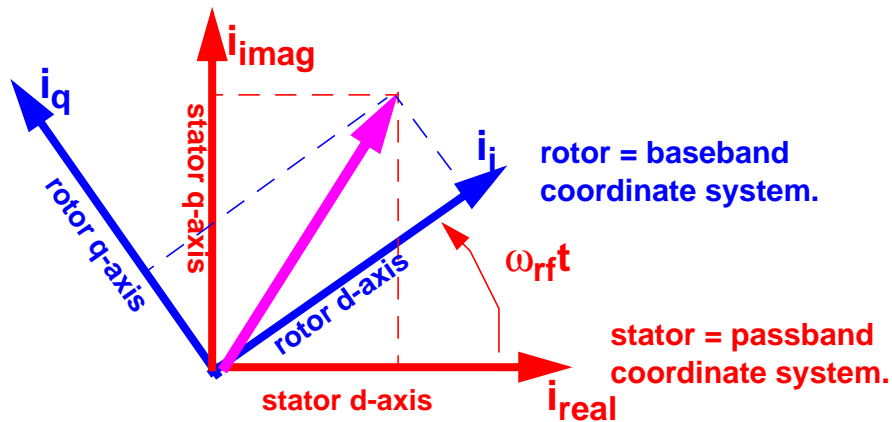
The mathematics below applies to capacitors as well as inductors¹. In [Figure 13-6](#) on page 594, the ideas behind a time-varying coordinate transformation used to model reactive elements at baseband are summarized.

1. The inductor baseband equivalent model derivation has a well-documented but little-known electro-mechanical analogy. The four inductor terminals resemble the stator windings of a two-phase rotating machine with shaft speed equal to the RF carrier frequency. Modulation is mathematically analogous to the flux linking a stator winding due to currents in orthogonal rotor windings. The flux depends on shaft angle just as a modulated signal depends on carrier phase. Transforming the vectorial equation for $v=Ldi/dt$ to the rotor reference frame suppresses the RF carrier and introduces a *speed voltage* [3,4,5,6,7,8,9], or back electro-motive force (back EMF), that couples the differential equations.

Figure 13-6 Modeling an Inductor at Baseband

$$i_{\text{real}} = i_i(t)\cos(\omega_{\text{rf}}t) - i_q(t)\sin(\omega_{\text{rf}}t) = \text{Real}[(i_i + j i_q) * e^{j\omega_{\text{rf}}t}]$$

$$i_{\text{imag}} = i_i(t)\sin(\omega_{\text{rf}}t) + i_q(t)\cos(\omega_{\text{rf}}t) = \text{Imag}[(i_i + j i_q) * e^{j\omega_{\text{rf}}t}]$$



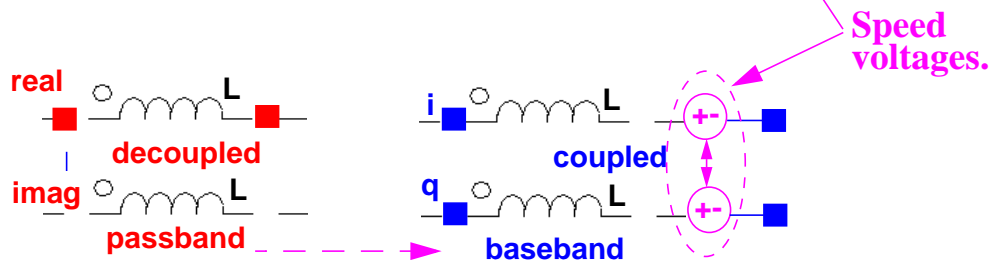
$$i = (i_i + j i_q) * e^{j\omega_{\text{rf}}t}$$

$$v = (v_i + j v_q) * e^{j\omega_{\text{rf}}t}$$

$$v = L di/dt$$

~~$$(v_i + j v_q) e^{j\omega_{\text{rf}}t} = L * [d(i_i + j i_q)/dt + j * \omega_{\text{rf}} * (i_i + j i_q)] e^{j\omega_{\text{rf}}t}$$~~

$$\begin{bmatrix} v_i \\ v_q \end{bmatrix} = L \frac{d}{dt} \begin{bmatrix} i_i \\ i_q \end{bmatrix} + L \omega_{\text{rf}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_i \\ i_q \end{bmatrix}$$



An expression for the real current (i.e. the passband current) appears at the top of the figure. The real current can be modeled as the projection of a two-dimensional rotating vector onto a stationary axis, the *real* axis. The vector rotates with an angular velocity equal to the RF carrier frequency.

The rotating vector also has a projection onto another stationary axis orthogonal to the real axis. In the baseband literature, this projection is the Hilbert transform of the real signal. The constitutive relationship of the inductor, $v=Ldi/dt$, can be expressed in terms of coordinates in a reference frame that rotates with the vector.

The middle part of the figure shows the constitutive inductor relationship between voltage and current in the rotating reference frame. Note that the trigonometric terms, the terms that slow simulation speed, are gone and the two projections are now coupled through *speed voltages*. The term speed voltage comes from the fact that the voltages depend on the angular speed of the rotating reference frame. In motor theory, that speed is the shaft speed. Speed voltage is similar to the back EMF in a motor. Because of speed voltages, baseband models of filters and reactive elements must have carrier frequency specified. The carrier frequency is the frequency for which the baseband signals are referenced. For example, the carrier frequency for an RF filter would be the RF frequency while the carrier frequency for an IF filter would be the IF frequency.

The baseband counterparts of the passband filter models are built up from inductors and capacitors modeled in the rotating reference frame.

In the complex expression for $v=Ldi/dt$, if we replace d/dt with $j\omega$, we find that the impedance of the inductor changes from $jL\omega$ to $jL(\omega+\omega_{rf})$. The same holds for capacitors, which means a filter transfer function, $H(\omega)$, has a baseband equivalent equal to $H(\omega+\omega_{rf})$. This is just the original passband transfer function shifted to the left by an amount equal to the carrier frequency. Our time domain baseband models are consistent with the text book frequency domain explanation of baseband modeling.

Warnings You Can Ignore

When you use the filters you will see warnings of *No DC path from ... to ground....* You can ignore those warnings.

When you use the modulators and mixers you will see warnings about *\$realtime*. They can be ignored too. *\$realtime* was not changed to *\$abstime* so that the library would work with version 4.4.5. The message is only a warning.

Use Model and Design Example

This section focuses on the baseband models and their use during the architectural design phase. The following steps walk you through

- The construction of a baseband model for a receiver
- The application of the Circuit Optimizer to help balance specifications among the function blocks
- The process of creating a passband testbench

The design goals were chosen arbitrarily. The example is meant simply to illustrate how to use the library. The example is a derivative of the design found in [10]. If in the steps below, some model parameters are not specified, leave them as default values.

The receiver is constructed from left to right, from input to output. Except for resistors and ports, all models come from the *rfLib*. Unless otherwise instructed, leave the port resistances at their default values of 50 Ohms.

1. Bring up a blank Composer schematic.
2. From the *measurement* category in the *rfLib*, instantiate a *CDMA_reverse_xmit* block. This model is a CDMA signal source. Connect a resistor between the *i_bin_node* and the *q_bin_node*. We will not need those outputs and do not want to be bothered with warnings about unused pins.
3. From the *measurement* category, instantiate a *BB_driver* to the right of the last block with its input pins aligned with the two top pins of the CDMA source. Wire the *i_out_node* pin of the source to the *I_in* pin of the *BB_driver*. Wire the *q_out_node* pin of the source to the *Q_in* pin of the *BB_driver*. Set the *dBm-out@1v peak* in parameter of the *BB_driver* to *-16*. The driver will convert 1 peak volt from the signal source to -16 dBm referenced to the output resistance of the driver.
4. To the right of the driver, instantiate a low noise amplifier, *LNA_BB* from the *top_dwnBB* category, align and wire the pins straight across, then set parameters as shown in [Figure 13-7](#) on page 597.

Figure 13-7 LNA_BB parameter settings

Available pwr gain[dB]	lna_gain
input resistance	50
output resistance	300
input referred IP3[dBm]	lna_ip3
noise figure [dB]	10
am/pm sharpness	2
cmp[dBm]	lna_ip3
[radians] @ cmp	.05
[radians] @ big input	.7
{1,0,-1} for {cw,none,ccw}	1

- To the right of the LNA, instantiate a *BB_butterworth_bp* filter from the *top_dwnBB* category, align the pins, wire the pins straight across, and set the parameters as shown in [Figure 13-8](#) on page 598.

The baseband equivalent model of the filter, and all reactive elements, must have the carrier frequency specified. The carrier frequency is used to compute the speed voltages shown in [Figure 13-6](#) on page 594. Since the filters are built up from inductors and capacitors, which have speed voltages, the filters need to have the carrier frequency specified. If a filter follows a RF-to-IF mixer, the carrier frequency is the IF frequency. The center frequency of the bandpass filter does not have to equal the carrier frequency. The center frequency is the frequency for which the filter was designed. The carrier frequency is the frequency to which the baseband signals are referenced.

Figure 13-8 BB_butterworth_bp settings

Filter Order	3
Input impedance	50
Output impedance	50
Center frequency(Hz)	frf
Relative bandwidth	rf_rbw
Insertion loss(dB)	3
carrier frequency	frf

6. To the right of the filter, instantiate a “dwn_cnvr” from the “top_dwnBB” category, align the pins, wire the pins straight across, and set the parameters as shown in [Figure 13-9](#) on page 599. Ground the “phase_err” pin. This block is an RF-to-IF mixer.

Figure 13-9 dwn_cnvrt parameter settings

available power gain[dB]	if_mx_gain
input resistance	50
output resistance	50
input referred IP3[dBm]	if_mx_ip
noise figure [dB]	10
RF frequency	frf
LO frequency	flo
AM/PM input point[dBm]	-30
phase shift at cmp[rad]	.7
phase shift at infinity	2
sharpness factor	2
{1,0,-1} = {cw,none,ccw}	0

- To the right of the mixer, instantiate another “BB_butterworth_bp” filter, align the pins, wire the pins straight across, and set the parameters as shown in [Figure 13-10](#) on page 600.

Figure 13-10 BB_butterworth_bp settings

Filter Order	50
Input impedance	50
Output impedance	50
Center frequency(Hz)	-frf+flo1
Relative bandwidth	if_rbw
Insertion loss(dB)	1
carrier frequency	-frf+flo1

8. To the right of the filter, instantiate an “IQ_demod_BB” block from the “top_dwnBB” category, align the pins, wire the pins straight across, and set the parameters as shown in [Figure 13-11](#) on page 601. Ground the “phase_err” pin.

Figure 13-11 IQ_demod_BB parameter settings.

available I-mixer gain[dB]	0
available Q-mixer gain[dB]	0
input resistance	50
output resistance	50
I- [dBm] input referred IP3	40
Q- [dBm] input referred IP3	40
noise figure [dB]	2
quadrature error	0
I-sharpness factor	2
Q-sharpness factor	2
I_cmp	-30
Q_cmp	-30
I-radians@I_cmp	.7
Q-radians@Q_cmp	.7
I-radians@big I-input	2
Q-radians@big Q-input	2
{1,0,-1} for {cw,none,ccw}	0
{1,0,-1} for {cw, none,ccw}	0

- To the right of the IQ demodulator, instantiate a “butterworth_lp” filter from the “top_dwnPB” category, align the input pin with the “I_out” pin of the demodulator, wire the pin straight across, and set the parameters as shown in [Figure 13-12](#) on page 602.

Figure 13-12 butterworth_lp settings

Filter Order	0
Input impedance	50
Output impedance	50
Corner frequency(Hz)	10M
Insertion loss(dB)	0

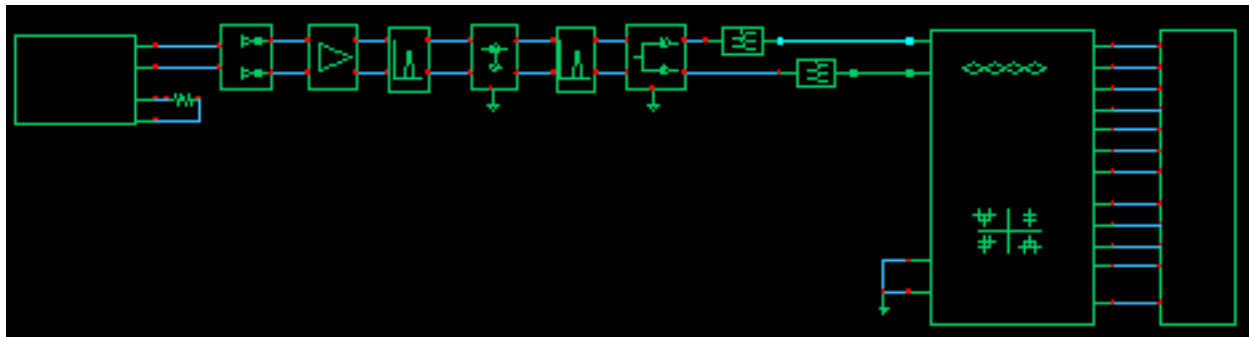
10. Copy the filter from step 9 and instantiate the copy to the right of the original but lower so that the input pin lines up with the “Q_out” pin of the IQ demodulator. Wire the pins straight across.
11. To the right of the low pass filters, instantiate an “offset_comms_instr” model from the “measurement” category and set the parameters according to [Figure 13-13](#) on page 603. This is one of the instrumentation models. Align the filter output pins with the “I_in” and “Q_in” pins on the instrumentation model. Wire the output of the top filter to the “I_in” pin of the instrumentation model. Wire the output of the lower filter to the “Q_in” pin of the instrumentation model. For now, ground the “I_ref” and “Q_ref” pins.

Figure 13-13 offset_comms_instr settings

symbols per second	1228800
I-sampling delay (secs)	134n
number of symbols	2
max eye-diag volts	1
min eye volts	-1
number of hstgm bins	100
I-noise (volts^2)	0
Q-noise (volts^2)	0
statistics start time	30u
input resistance	50

- To the right of the instrumentation model, instantiate an “instr_term” model. This model simply terminates the instrumentation model outputs so that the Composer does not inundate the user with warnings of unused pins. Align the pins and wire them straight across. This model has no parameters. By now the schematic should look like the one in [Figure 13-14](#) on page 603.

Figure 13-14 Completed receiver model



Affirma RF Simulator (SpectreRF) User Guide

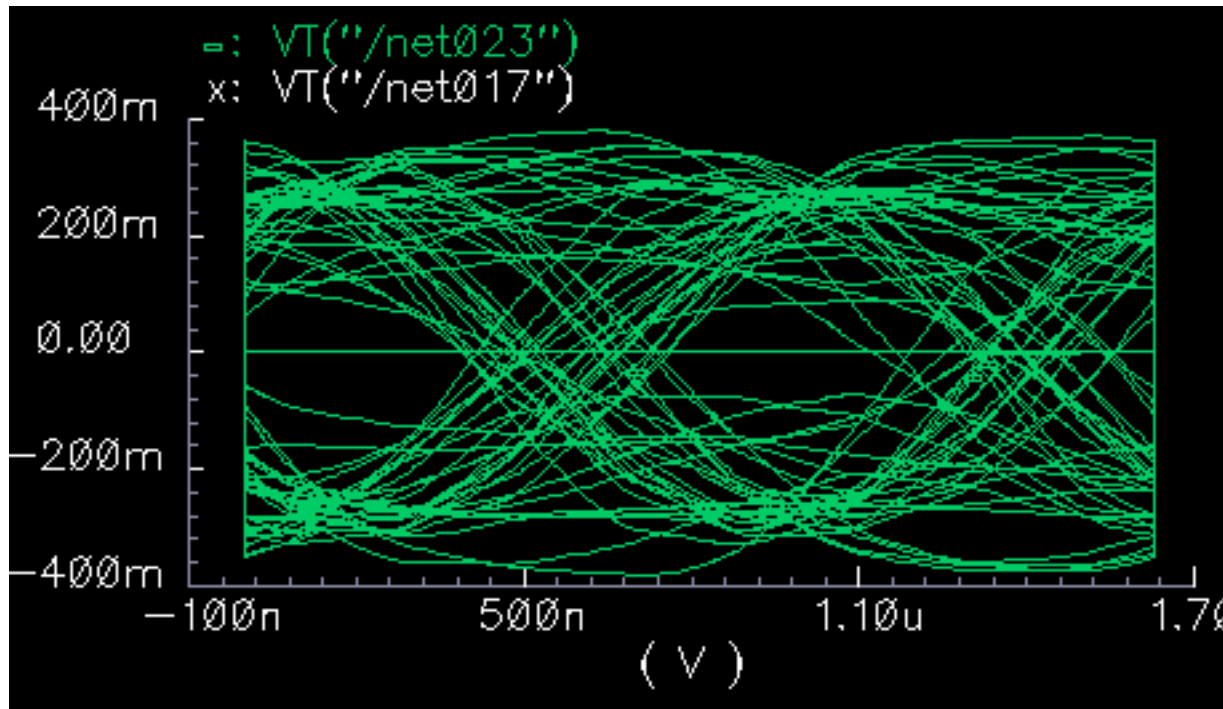
Behavioral Models and Methods for Top Down Design

13. Bring up an Analog Environment window, copy variables from the cell view, then set the variables according to table 1. You will have to add the variables rf_rbw and if_rbw to the list because variable filter parameters do not “copy from cell view”. (This is a known bug.)

Variable	Setting
lna_gain	15
lna_ip3	-5
if_mx_gain	10
if_mx_ip	35
frf	2.14G
flo1	2.354G
if_rbw	200m
rf_rbw	100m

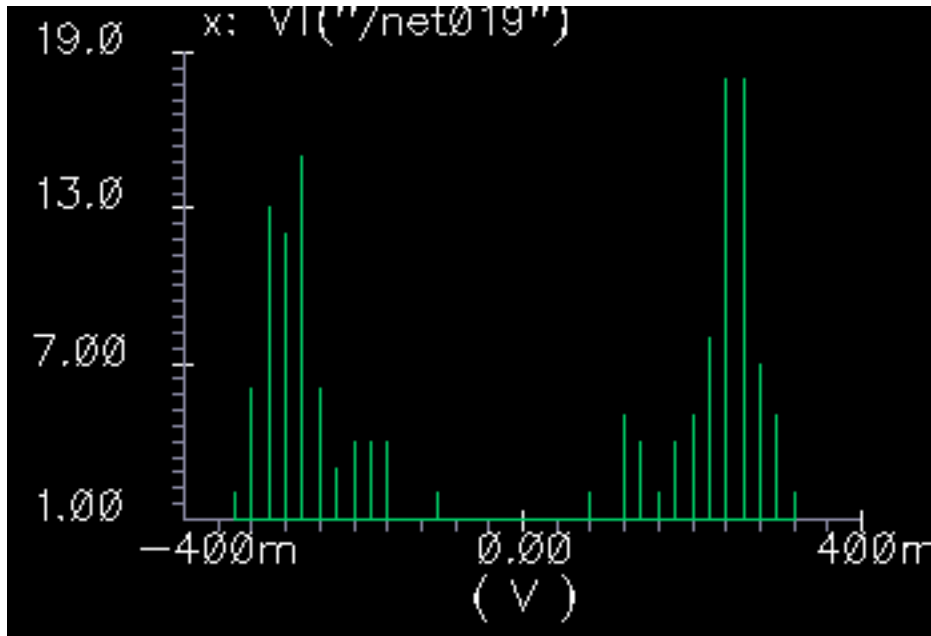
14. Set up and run a 130us Transient analysis with a 30us output start. By delaying the output start, you remove start-up transients from the eye-diagrams and scatter plots.
15. Now plot an eye diagram. You do this by first plotting the “sawtooth” and “I_eye” outputs of the instrumentation block. Select “sawtooth” as the x-axis. You should see the eye-diagram shown in [Figure 13-15](#) on page 605. The “I-sampling delay” parameter in the instrumentation block is chosen with respect to this eye diagram. The delay is the time when the eye opens the widest. The instrumentation block will sample the input waveforms with this delay to compute all statistics and to produce scatter plots.

Figure 13-15 Eye-diagram



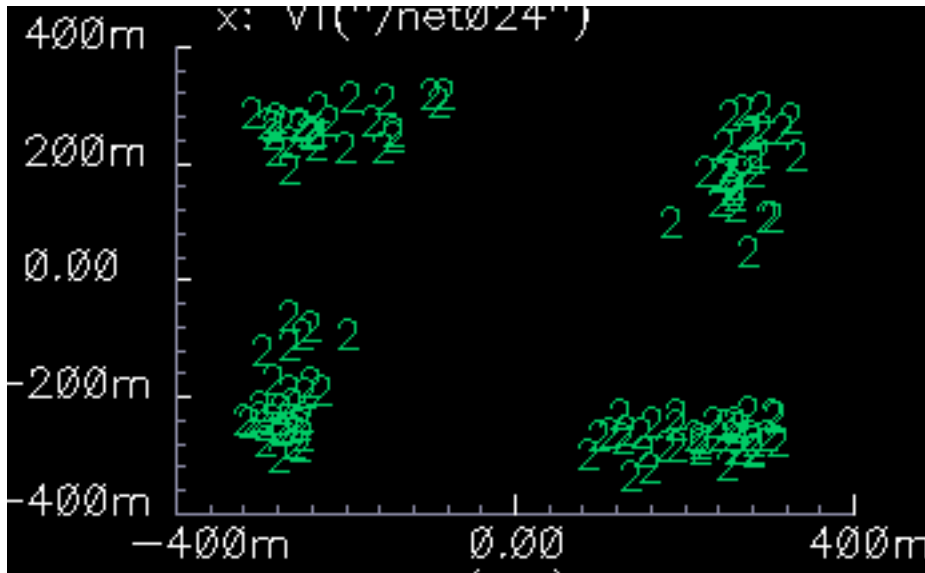
16. To generate a histogram of the I-voltage at the sampling times, plot the “eye_hist” and “eye_count_hist” outputs. Make “eye_hist” the x-axis. Under curves->options, change the plotting style to “bar”. Also change the number of ticks to zero. You should see the histogram in [Figure 13-16](#) on page 606. The “ave_eye” and “root_var_eye” give running estimates of the absolute value of the voltage at the sampling instants and the square root of the variance. These numbers can be used to crudely extrapolate bit error rate.

Figure 13-16 Histogram



17. To generate a scatter plot of the received symbols, plot the “I_scatter” and “Q_scatter” outputs and use “I_scatter” as the x-axis. Change the plotting style to “Data points only”. This should produce the plot in [Figure 13-17](#) on page 607. The CDMA source produced offset QPSK symbols. Offset QPSK modulation staggers the digital changes in the I and Q signals to avoid traversing the origin. Running the baseband trajectory through the origin increases spectral regrowth in the transmitters. The “offset_comms_instr” samples the I and Q signals at different times then plots the two staggered samples against each other. The resulting scatter plot shows the received symbols. Scatter plots of the unstaggered samples only reveals what’s happening to one dimension, I or Q. For non-offset QPSK and QAM modulation schemes, use the “comms_instr” block instead of the “offset_comms_instr” block.

Figure 13-17 Scatter plot



18. The instrumentation block also computes root-mean-squared (RMS) error vector magnitude (EVM). The error vector is the vectorial difference between the ideal received symbol and the actual received symbol. EVM is the magnitude of the error vector. The RMS EVM is the root-mean-squared value of a sequence of error vector magnitudes. RMS EVM is a measure of receiver's quality. RMS EVM can account for as much or as little distortion and noise as you like. The trick is in figuring out where the ideal received symbol lies. That is done with the "I_ref" and "Q_ref" inputs to the instrumentation block. You first duplicate the entire receiver chain. You then make the chain as ideal as you like by changing the specifications of the function blocks. For example, if you want to see what IP3 of the LNA alone does to rms EVM, make it absurdly large in the duplicate receiver chain. The "ideal" receiver chain is driven from the same input as the original receiver chain. The output of the ideal receiver drives the "I_ref" and "Q_ref" inputs of the instrumentation block. This step sets up and runs an RMS EVM analysis.
 - a. First, duplicate the entire receiver chain from the BB_driver to the butterworth_ip filters.
 - b. Remove the ground from the "I_ref" and "Q_ref" pins and connect the duplicate receiver to those pins.
 - c. Drive the duplicate receiver from the CDMA source. In this example you make every block in the ideal receiver ideal.
 - d. Starting with the LNA, set the input referred IP3 to 100.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

- e. Set the “{1,0,-1}...” parameter at the end of the properties form to 0. This eliminates AM/PM conversion.
 - f. Bring up the edit properties form for the RF bandpass filter. Change the cell name to “BB_loss”. The reference impedance should equal the output resistance of the filter and you should not have change it. The sole purpose of the BB_loss model is to replace a filter for an RMS EVM analysis. The BB_loss model retains the filter’s loss but eliminates the dynamics so you can see what, if any, affect the filter has on EVM through intersymbol interference. You can replace the filter with straight wires if you want to eliminate the loss too but this example does not do that.
 - g. Change the RF-to-IF mixer IP3 to 100.
 - h. Bring up the edit properties form for the IF filter and change the cell name to BB_loss.
 - i. Change the demodulator IP3 parameters to 100.
 - j. Replace the low pass filters with wires.
 - k. Check and save the circuit then re-run the analysis.
 - l. Plot the RMS_EVM output of the instrumentation block. The trace starts at 30us, which is the “statistics start time” parameter of the block. This parameter allows you to keep start-up transients out of the statistics. The trace settles out at 25.84 Volts. That means that after 130us of data, and ignoring the first 30us, the RMS EVM is 25.84%. The EVM measurement is normalized to the RMS magnitude of the ideal symbol then multiplied by 100 to express the measurement as a percentage.
- 19.** There is one more construction step before proceeding to the Circuit Optimizer application. The Circuit Optimizer will be set up to minimize rms noise subject to performance constraints. This step replicates the receiver model yet one more time to generate the noise measurement.
- a. Copy the original receiver chain from the BB_driver up to and including the low pass filters and place the copy above the original receiver model.
 - b. Ground the “Q_in” pin on the BB_driver and drive the “I_in” pin with a 50mV DC voltage source.
 - c. Load the low pass filters with a res_BB model from the top_dwnBB category. Use the default parameters and ground the output pins.
 - d. Check and save the schematic.
 - e. Set up a Noise analysis from 0 to 100MHz. In the noise form, set the Output Noise to “voltage”, select the “I_in” pin on the res_BB model as the Positive Output Node

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

and the “I_out” pin as the Negative Output Node. Select the input DC voltage source as the Input Noise.

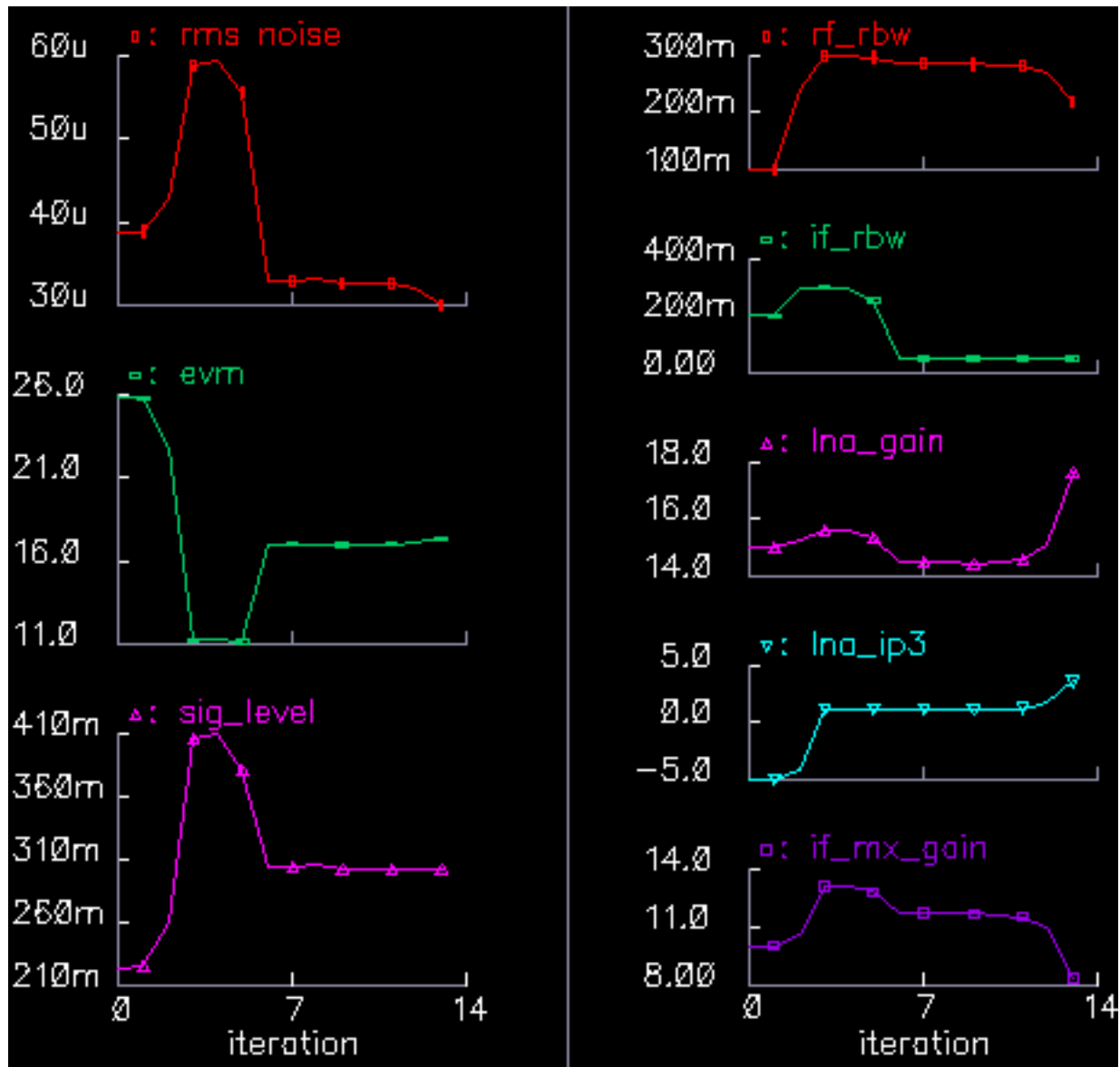
- f. Run both the Noise and the Transient analyses to make sure everything works.
- g. Bring up a Waveform Calculator and under Special Functions, click on “rmsNoise”. In the form that comes up enter From 0 To 100M.
- h. Bring up the Circuit Optimizer and click on “Goals” then “Add...”.
- i. Click on “Get Expression”, which is to the right of the Calculator label. Type in “rms noise” in the “Name” field. For Target, enter 0.1u. For Acceptable, enter 20u. Click OK. By default, the Circuit Optimizer will minimize a goal.
- j. In the Waveform Calculator, click on “vt” then click on the rms_EVM output of the instrumentation block in the schematic.
- k. In the Waveform Calculator, click on Special Functions then click on “Value...”. Interpolate at the last time point, 130us.
- l. In the Circuit Optimizer window, add another goal. Again, in the Calculator line, click “Get Expression”. Name this expression “evm”, set the Direction to \leq , set the target to 25, enter 10 for Acceptable then click the “% within Target” button. Click OK.
- m. In the Waveform Calculator, click on vt again then click on the I_in pin of the instrumentation block.
- n. In the Waveform Calculator, click on Special Functions then click on “rms”. We will try to keep the rms value of this signal level above 300mV. Note that all goals must be scalars.
- o. In the Circuit Optimizer window add another goal, name it sig_level, set the direction to \geq , the target to 300m, and set Acceptable to 10%.
- p. Add the variables such that the Optimizer window looks like [Figure 13-18](#) on page 610.
- q. Set the number of runs to 12 and start the Circuit Optimizer analysis. This will take a several hours to complete.

Figure 13-18 Circuit Optimizer setup

1	rms noise	minimize	100n				yes
2	evm	<=	20				yes
3	sig_level	>=	300m				yes
Variables							
#	Name	Min	Max	Initial	Prev	Current	Enable
1	rf_rbw	50m	300m	100m		50m	yes
2	if_rbw	50m	300m	200m		116.6m	yes
3	lna_gain	10	30	15		25.22	yes
4	lna_ip3	-9	10	-5		10	yes
5	if_mx_gain	1	50	10		26.41	yes

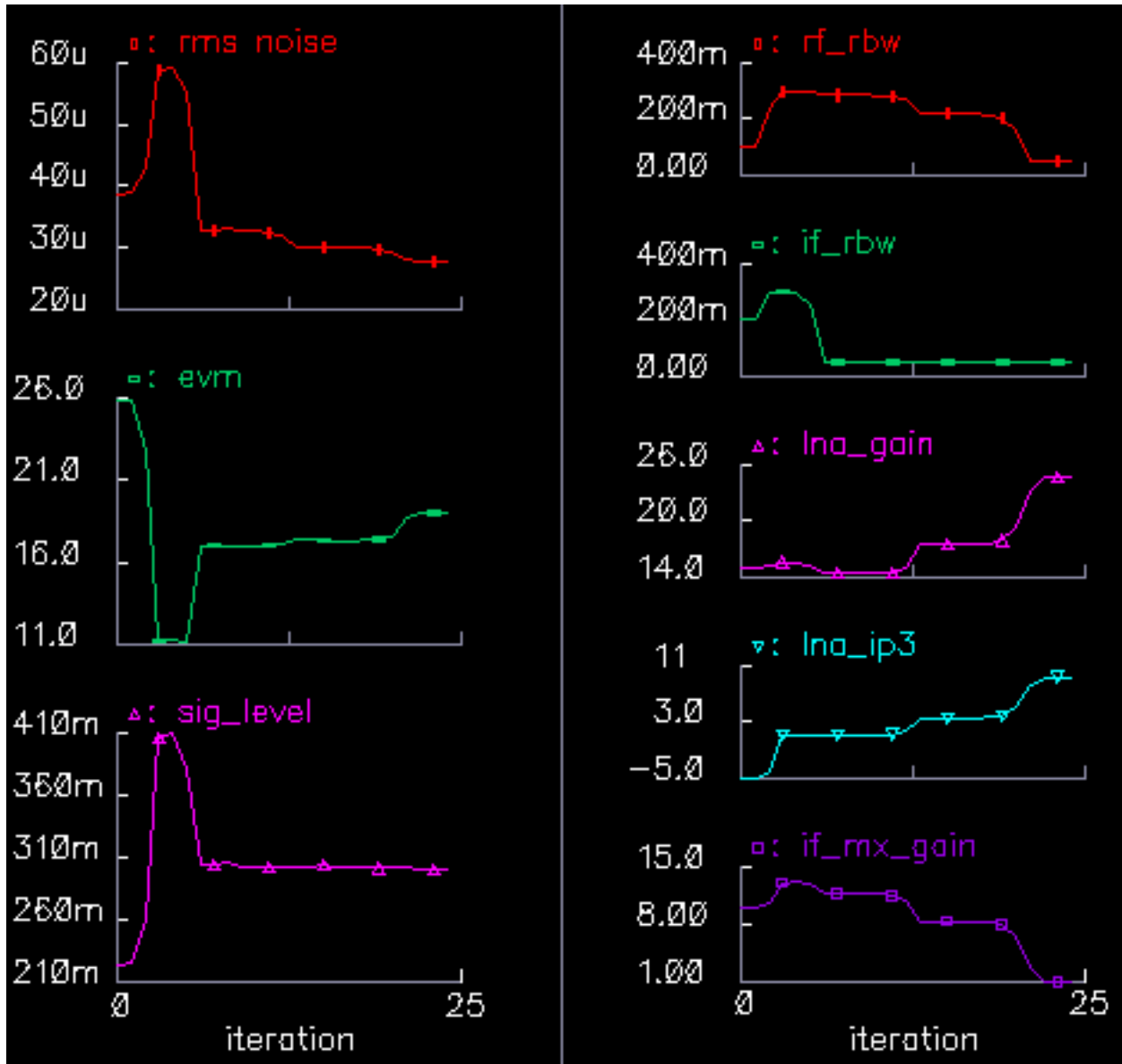
20. [Figure 13-19](#) on page 611 shows the Optimizer results. The traces on the left show the EVM, RMS output signal level, and RMS noise at each Optimizer iteration. The right-hand traces show the variable function block specifications at each Optimizer iteration. Although the example is contrived, it illustrates the use model. After the Circuit Optimizer met the constraints it tried to minimize RMS noise.
21. Save the initial state of the Analog Environment in case you want to start over. Then in the Optimizer window, click Results->Update Design. The last click updates the variables in the Analog Environment window with the last set of variables found by the Circuit Optimizer. You will use these states in the passband view.

Figure 13-19 Circuit Optimizer results.



22. If you have time to run the optimization for another 12 iterations you should see the results in [Figure 13-20](#) on page 612.

Figure 13-20 Circuit Optimizer results after 24 iterations



To summarize, the semi-automated design procedure consists of

- Setting up the measurements
- Placing tolerances on the block parameters
- Constraining the system performance
- Identifying a quantity to minimize (or maximize)

- Running the Optimizer
- Evaluating the results¹.

Each simulation covers 100s, or about 80 CDMA symbols. The suppressed carriers are a 2.14GHz RF carrier, an LO carrier at 2.354GHz, and an IF carrier at 214MHz. The symbol rate is 1.2288 Mega-symbols per second.

Each combined Transient and noise analysis runs in less than 3 minutes on an Ultra 1. Six optimizer iterations take between 1 and 2 hours depending the machine and the options. Based on run times from the BB_test_bench circuit, with Envelope Following analysis the same six optimizer iterations would have taken over 6 days. With Spectre Transient they would have taken over 2 months. There is no way either passband approach can optimize specifications and tolerances in any reasonable amount of time.

Once the architecture is designed you can quickly create a passband view of the architectural model². The passband view checks for problems that escape the baseband view. For example, although the baseband view quickly assesses what filters do to the baseband signal, baseband models do not indicate whether the filters are indeed removing undesired carrier harmonics. Baseband modeling is also not the best way to evaluate image rejection. Although the baseband model accurately simulates how the desired signal propagates through an image rejection receiver, it does not accurately simulate how much of the image signal propagates to the receiver output.

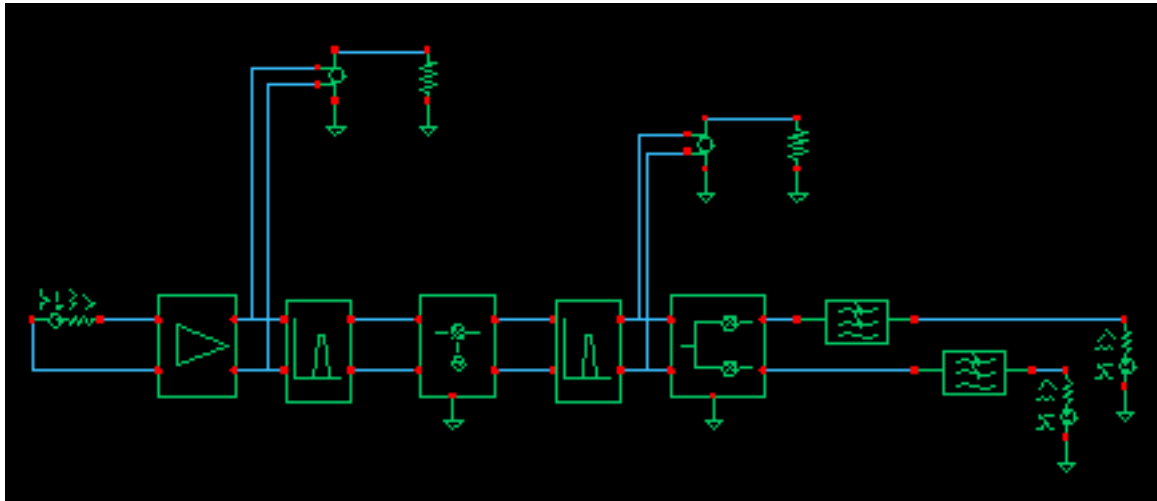
The passband view also creates a system testbench as mentioned in the section on Top Down Design of RF Systems. The steps below show how to switch views and make an end-to-end RF measurement, the one dB compression point. The one dB compression point is usually a transmitter specification but it is better for demonstrating this flow because it is easier to set up.

1. Copy the original receiver model from the Circuit Optimizer analysis to a new schematic window. Copy everything from the LNA to the low pass filters.
2. Edit the properties of the IQ_demodulator (IQ_demod_BB) to set the last parameter, flo, to “-frf+flo1”. The baseband view does not need the local oscillator frequency but the passband view does.

-
1. This is why the process is called **semi**-automated. Upon evaluating results for the first or second time you will probably have to refine tolerances, refine goals, add or delete constraints, add or delete variables.
 2. In this release, the passband behavioral models in the top_dwnPB category and in the passband view do not introduce any specifications that are not in the baseband models. Subsequent releases of the library may give the passband models the ability to introduce new specifications such as IIP2 and DC offsets.

3. Load the low pass filters with ports.
4. Connect a port across the LNA_BB inputs. Set the Frequency name to “fin”, the frequency to frf, and the amplitude to “power”. (Don’t abbreviate power to “pwr”. “pwr” is a reserved variable and you will not get any warning. Spectre may complain about a mysterious indexed undefined variable that increments from run to run.)
5. Add loaded voltage-controlled-voltage sources as shown in [Figure 13-21](#) on page 614 to observe intermediate differential voltages. Alternately, you could use the 4.4.6 engineering release Main Results form. It lets you display differential voltages without using the Waveform Calculator or inserting voltage-controlled-voltage sources.

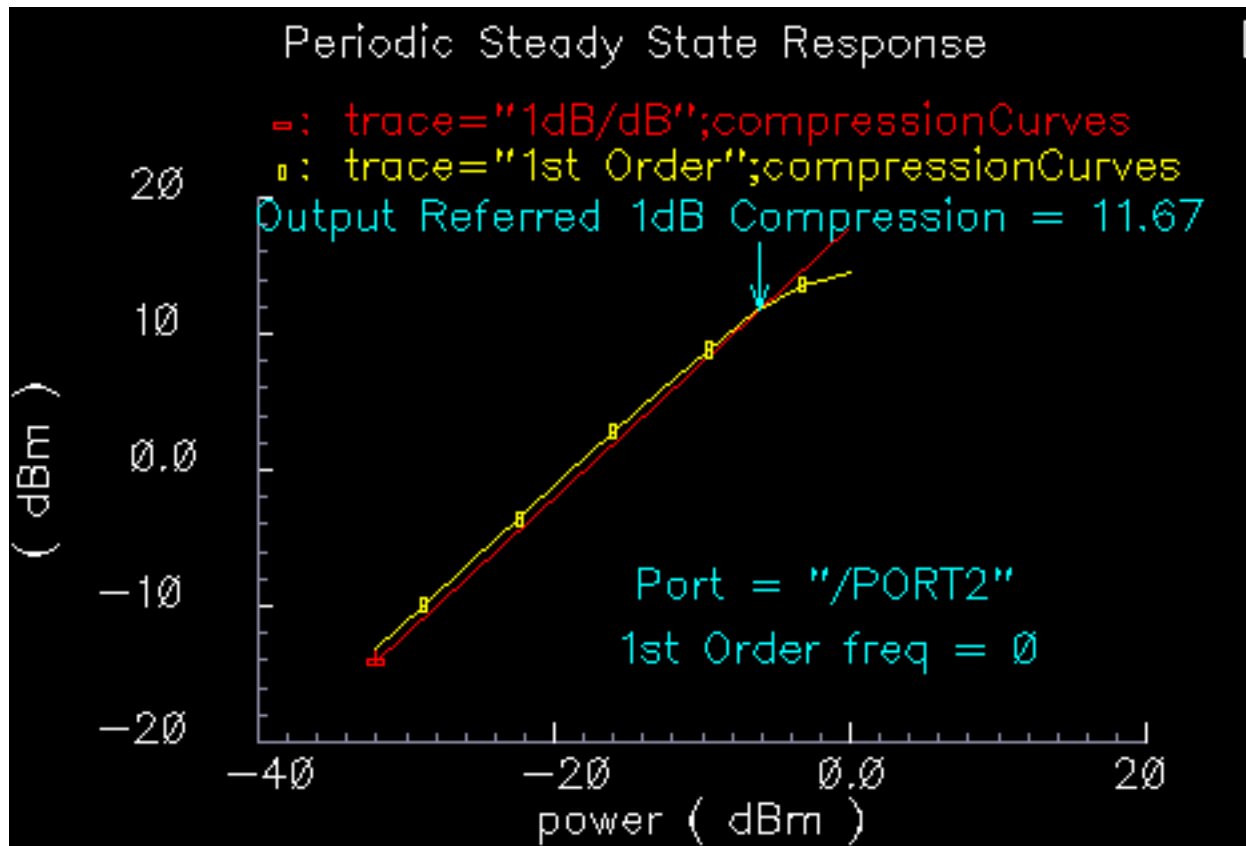
Figure 13-21 Passband view



6. Check and save the schematic
7. Close the schematic window.
8. Bring up a Library Manager and select but do not open the schematic you just created.
9. In the Library Manager window click File->New->Cell View.
10. In the window that comes up, select Hierarchy Editor for the Tool. Click OK.
11. Type in “schematic” for the View then click on “Use Template.”
12. Set the Name to Spectre
13. Enter “veriloga_PB” as the first item in the View List then click OK.
14. In the HE (Hierarchy Editor) window, click File->Save. This is important.

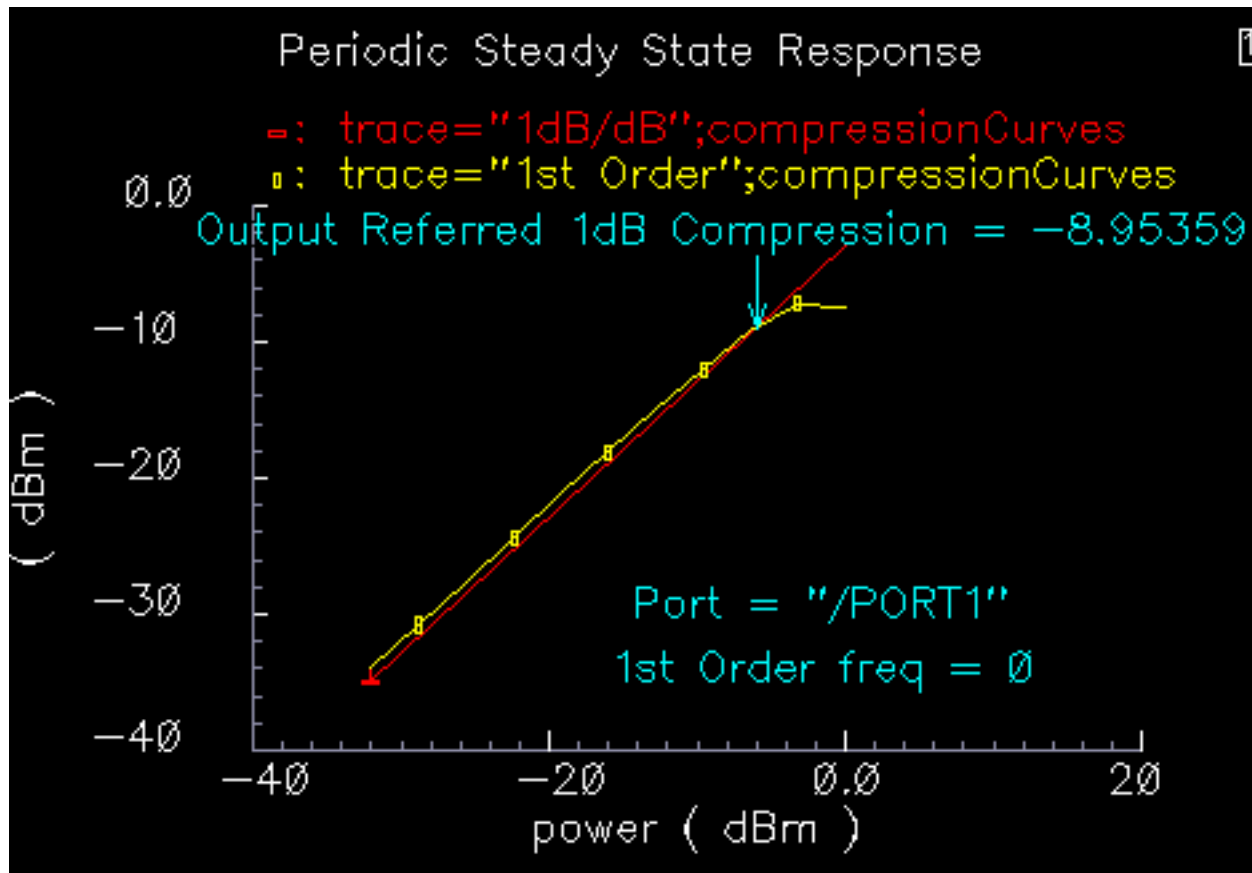
15. Click the Open button to bring up the schematic.
16. Bring up an Analog Environment tool and use the states from the last Circuit Optimizer iteration.
17. Recall the states you saved from the last Circuit Optimizer iteration. Add the “power” variable and set it to -16.
18. Delete the previous Analyses and set up a PSS analysis. Enter a new Fundamental Tone named “LO” and make it 2.354GHz. Auto Calculate the Beat Frequency and enter “1” for the Number of harmonics.
19. Run the analysis and observe the intermediate differential voltages. The model is indeed now a passband model. At the higher power levels the LNA output contains odd harmonics of the RF carrier. The filter reflects the odd harmonics back to the LNA and does not let them propagate forward. The baseband model does not simulate the odd harmonics but it does simulate the intermodulation term between the second harmonic and fundamental that falls at the fundamental. One reason to simulate the passband view is to check for peak voltage levels that might exceed voltage ratings. The baseband models only simulate peak voltage at the carrier fundamental, not the absolute peak.
20. Set up swept PSS analysis. Sweep “power” from -32 to 0 in 10 steps.
21. After the sweep finishes, click on Results->Direct Plot in the Environment window, select Compression point, 1dB, Output Referred. Select the 0 harmonic because the end-to-end system produces a baseband output. Then click on the port loading the top output low pass filter. You should see the compression point plot show in [Figure 13-22](#) on page 616.

Figure 13-22 End-to-end RF measurement, one dB compression point at the I-baseband output.



22. Repeat the last step but this time click on the lower output port. You should see the compression plot in [Figure 13-23](#) on page 617.

Figure 13-23 End-to-end RF measurement, one dB compression point at the Q-baseband output.



You may wish to compare baseband and passband models. The next set of steps show you how to set up a Transient analysis with the passband view and a Transient analysis with the baseband view for direct comparison of baseband and passband models. You will run one analysis of the baseband view and two of the passband view. The second passband analysis will be with tightened tolerances.

1. Save the passband schematic under a different name. You will use the new copy.
2. Repeat steps 9 through 17 from the last recipe for the new copy but do not enter the "veriloga_PB" view in the View List yet. You will do a baseband analysis first.
3. Delete the port driving the LNA.
4. You can also delete the loaded voltage-controlled-voltage sources.
5. You need to synthesize an antenna signal. Add an IQ_mod_BB from the top_dwnBB category. Set the I and Q gains to 0 dB. Set the 1dB compression points to 1000 so that

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

the modulator is ideal. Instantiate it in front of the LNA with the pins aligned then wire the pins straight across. Ground the “phase_err” pin.

6. Drive the “I_in” pin of the IQ modulator with a port. Set the port frequency to 2MHz and name the frequency BB1. Set the amplitude to -16dBm.
 7. Do the same for the “Q_in” modulator input.
 8. Load or duplicate the states from the 12-iteration Circuit Optimizer analysis but delete the Noise and Transient analyses.
 9. Remove AM/PM conversion from the LNA by setting the last parameter in the properties list to zero. It is not fair to compare passband and baseband views with AM/PM conversion because the passband view does not capture it.
 10. Set up a 1us Transient analysis with default options.
 11. Run the analysis and plot the filtered baseband outputs, the outputs of the low pass filters. Note how fast the simulation runs. Save the results so you can plot them again later. ¹
 12. Switch to the passband view by entering “veriloga_PB” in the View List in the Hierarchy Editor. Click the update button in the HE.
 13. After you switch to the veriloga_PB view, edit the IQ modulator properties to set “flo” to frf. Edit the demodulator properties and set its “flo” to flo1-frf.
 14. Click Results->Printing Plotting Options then click the “Overlay Plots” button.
 15. Overlay the passband results with the baseband results. You should see the waveforms in [Figure 13-24](#) on page 619. The comparison is not very good.
 16. Rerun the analysis with conservative options and set reltol to 1e-6. This run will take a while.
 17. Plot the results.
 18. Recall the saved baseband results and overlay them with those from the last simulation. You should see the waveforms in [Figure 13-25](#) on page 620. The passband results now lay right on top of the baseband results but took much longer to compute! It was not obvious without the baseband results that the first passband simulation did not run with tight enough numerical tolerances.
-
1. Note that the baseband outputs are out of phase with each other, even though the baseband inputs are in phase. In the baseband model, changing the RF-IF mixer LO from “flo1” to “-flo1” fixes the sign problem. In the passband model, the IQ_demodulator flo should be frf-flo1. To maintain the convention, in the baseband model the IF filter’s carrier frequency should be frf-flo1.

Figure 13-24 Passband and baseband results with default options in the passband analysis

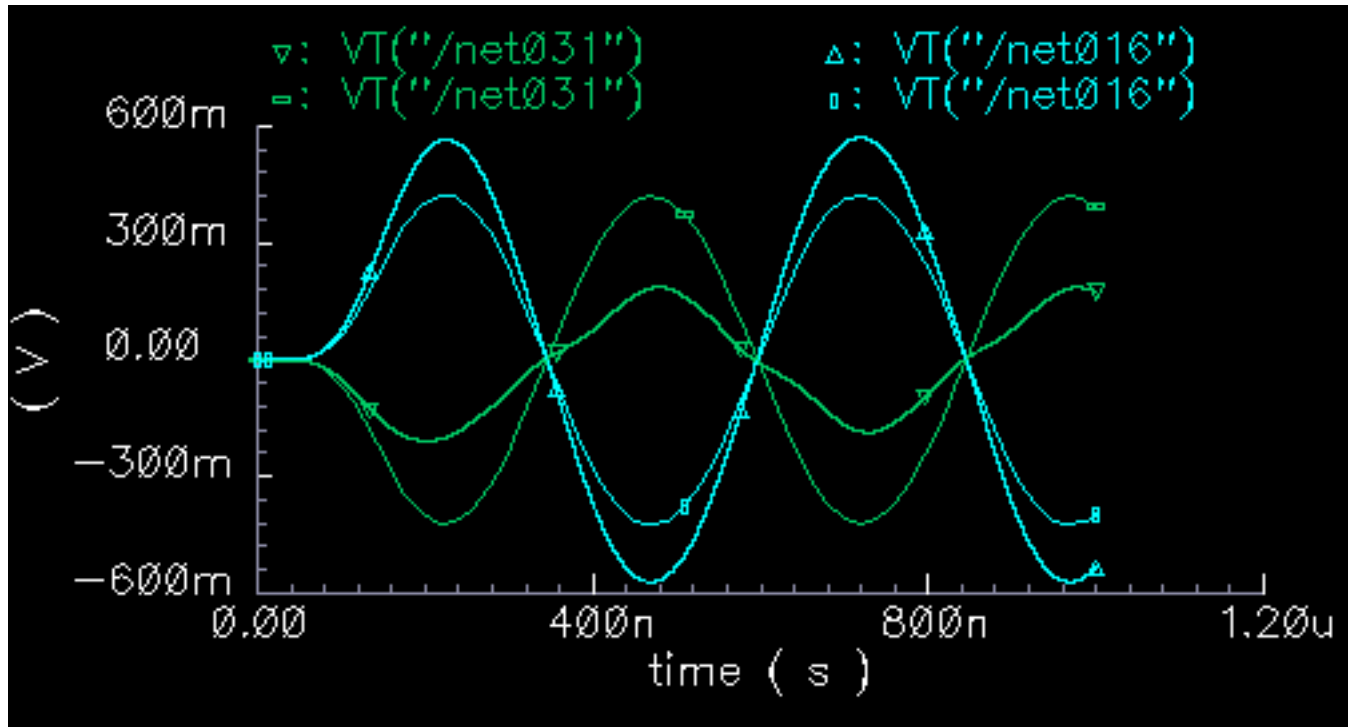
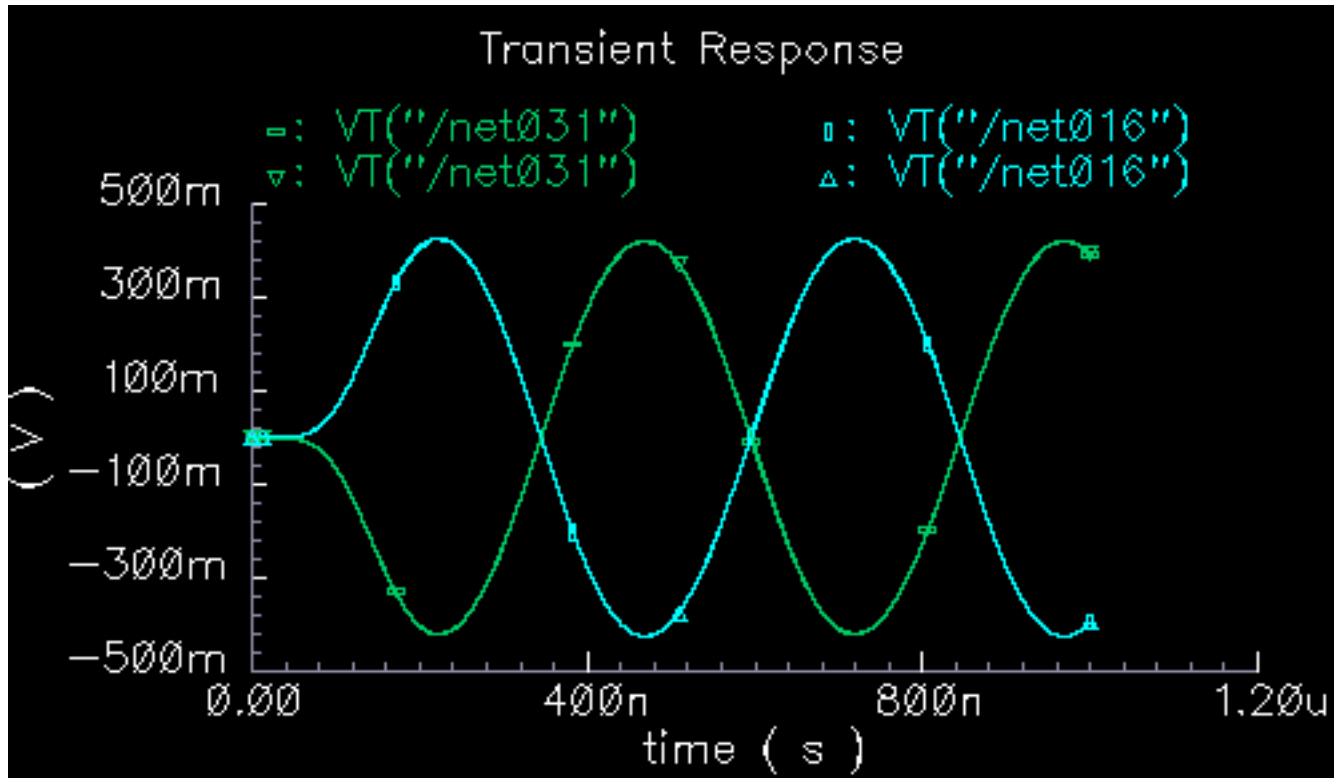


Figure 13-25 Passband and baseband results with tighter options in the passband analysis



Relationship Between Baseband and Passband Noise

Noise analysis at baseband can be confusing because factors of two appear in a number of places throughout the calculations. For example:

- Each passband node becomes two equivalent baseband nodes.

Does noise injected at a passband node split between the two equivalent baseband nodes?

If so, does the noise split evenly?

- As shown in the BB_test_bench example, baseband models simulate peak in-phase and peak quadrature components of the carrier.

When analyzing signal-to-noise ratios, does that mean you have to use half the square of the baseband signals?

- The Analog Artist displays single sided power spectral densities.

Since the baseband power spectral density is the two-sided passband density shifted down, is there another factor of two because we can only see the baseband density for positive frequencies?

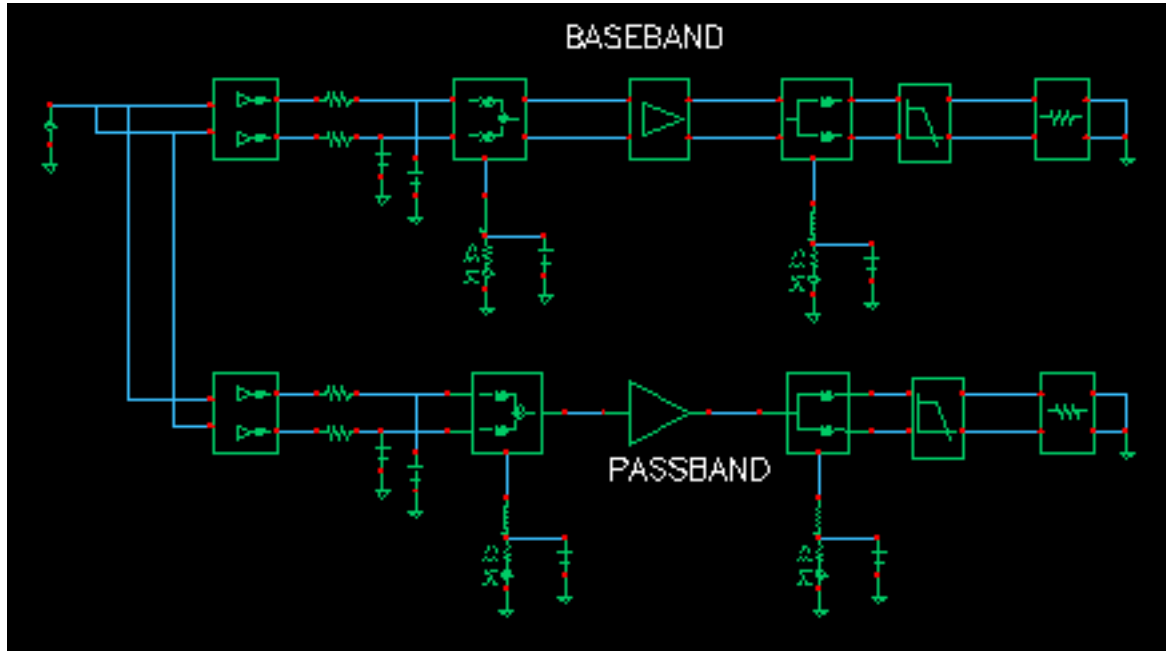
- White noise at the input of a mixer mixes up to the carrier from DC but there is also noise at twice the carrier that mixers down to the carrier.

Does the baseband model account for this?

Before sorting out the factors of two, please note that baseband noise analysis is valid only for small signals. If any element in the architecture operates in a non-linear fashion, the noise analysis might be inaccurate. This is due to the fact that a baseband noise analysis follows a DC operating point analysis, rather than a PSS analysis. Instantaneous incremental gain in a passband static non-linear model dithers at the carrier frequency. When the carrier swings through zero, the incremental gain is large and noise at the input is amplified. When the carrier reaches its peak and drives the circuit into saturation, the incremental gain is smaller. The average gain is greater than the minimum gain. The baseband model remains in the non-linear region because it only simulates peak voltages. Consequently, the incremental gain is always a minimum and the baseband model under-estimates the amount of noise propagating to the output. If the peak input signal drives the model into saturation be sure to scale the baseband noise results accordingly.

The circuit discussed here is called *noise_test_circuit* and you can find it in the *rfExamples* library. The circuit looks as shown in [Figure 13-26](#) on page 622.

Figure 13-26 The `noise_test_circuit` in the `rfExamples` Library



The `noise_test_circuit` shows the relationship between baseband and passband noise. One branch consists of passband models. The other branch is a baseband equivalent of the first.

You can assess noise at each of three observation points located in each branch of the circuit. At each observation point, you can examine both the noise and noise summaries.

The I and Q inputs are both driven by the same DC source so that you only have to view one baseband output, the other baseband output is identical by symmetry. Noise parameters in the passband and baseband models are identical. Aside from the behavioral blocks at the end of each branch, each behavioral block has noise injected at its input.

Setup a PSS analysis. Since the local oscillator is inside the passband mixer models you will have to manually enter the frequency (1GHz) into the PSS analysis form. Let the beat frequency be *autocalculated* and use 1 harmonic.

Setup a noise analysis with the start frequency equal to 0 Hz, the stop frequency equal to 100MHz, use a linear sweep with 100 steps. Set the Maximum sideband to 1. For the input source select *none* and for the output use *voltage*. For the positive output node, select the I-input of the `IQ_modulator` in the passband branch. Use ground for the negative node.

Setup a noise analysis that sweeps frequency from 0 to 100MHz linearly in 100 steps. Use voltage for the output noise and select the I-input of the `IQ_mod_BB` component in the

baseband branch for the positive node. Select ground for the negative node. Set the input noise port to any one of the ports in the circuit. Noise from that port will not affect either passband nor baseband branches.

Run the analysis.

When the analysis finishes, go to the Analog Artist simulation window and click on **results-print-pss noise summary**.

In the form that appears, include All types, set *Type* to *integrated noise*, look at noise from 0 to 100MHz, select "truncate by number", and view the top 2 noise contributors. There are only 2 noise contributors at this point, noise at the I/Q inputs and noise from the low pass resistors.

Now print the noise summary. (This is different from the pnoise summary.) Again, print integrated noise (from 0 to 100MHz). Select All types and print noise from the top 2 noise contributors.

Figure 13-27 on page 624 shows the noise summaries. The two summaries agree because at this point in the circuit, both nodes are really baseband nodes.

Figure 13-27 Noise Summaries for the First PSS PNoise and Noise Analyses

Device	Param	Noise Contribution	% Of Total
/I2	IQ_modulator_i	1.56676e-12	70.06
/R20	rn	6.69678e-13	29.94
Integrated Noise Summary (in V ²) Sorted By Noise Contributors			
Total Output Noise = 2.23644e-12			
No input referred noise available			
Device	Param	Noise Contribution	% Of Total
/I22	IQ_mod_BB_i	1.56676e-12	70.06
/R22	rn	6.69679e-13	29.94
Integrated Noise Summary (in V ²) Sorted By Noise Contributors			
Total Output Noise = 2.23644e-12			
Total Input Referred Noise = Inf			

Repeat the analyses but this time select the “out” pins on the low pass filters as outputs in the appropriate noise analyses. Again print the same noise summaries but this time look at the top 12 noise contributors in each summary.

Figure 13-28 on page 625 shows the noise summaries for the second analysis. The top 7 noise contributors in the baseband and passband branches agree. The remaining noise contributors are negligible, and should be negligible for the circuit since it has no AM/PM conversion.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

Figure 13-28 Noise Summaries for the Second PSS PNoise Analyses

Device	Param	Noise Contribution	% Of Total
/I77	PA_PB	7.49879e-10	82.29
/I2	IQ_modulator_i	9.59448e-11	10.64
/R20	rn	4.1437e-11	4.55
/I87	butterworth_lp_i_noise	1.52928e-11	1.68
/I94	IQ_demodulator	7.54549e-12	0.04
/PORT13	rn	3.77571e-14	0.00
/PORT14	rn	3.77571e-14	0.00
/I2	IQ_modulator_q	2.33139e-15	0.00
/R21	rn	9.96506e-16	0.00
/R22	rn	0	0.00
/R23	fn	0	0.00
/R23	rn	0	0.00

Integrated Noise Summary (in V²) Sorted By Noise Contributors
 Total Output Noise = 9.11279e-10
 No input referred noise available

Device	Param	Noise Contribution	% Of Total
/I76	PA_BB_i	7.49861e-10	82.29
/I22	IQ_mod_BB_i	9.59448e-11	10.64
/R22	rn	4.14371e-11	4.55
/I86	butterworth_lp_i_noise	1.52928e-11	1.68
/I94	IQ_demod_BB_i	7.54549e-12	0.04
/PORT15	rn	3.77571e-14	0.00
/PORT9	rn	3.77571e-14	0.00
/I76	PA_BB_q	1.80329e-14	0.00
/I22	IQ_mod_BB_q	2.33136e-15	0.00
/R23	rn	9.96493e-16	0.00
/I82	IQ_demod_BB_q	1.54184e-31	0.00
/R23	fn	0	0.00

Integrated Noise Summary (in V²) Sorted By Noise Contributors
 Total Output Noise = 9.11279e-10
 Total Input Referred Noise = 0.0151541

Now, repeat the analysis but this time change the Pnoise sweep to run from 900MHz to 1.1GHz in 200 linear steps and select the power amplifier output as the noise output node. For the noise analysis, leave the sweep at zero to 100MHz, use 100 steps as before, and change the output node to be the I-output of the power amplifier.

Affirma RF Simulator (SpectreRF) User Guide
Behavioral Models and Methods for Top Down Design

Look at the top 7 noise contributors in each analysis. This time, integrate noise from 900MHz to 1.1GHz for the pnoise run and integrate noise from 0 to 100MHz for the noise run. [Figure 13-29](#) on page 626 shows the new summaries. Although noise analyses agree at either ends of the branches, noise analyses seem to disagree at a point where the baseband node is only a baseband equivalent, not a true baseband node.

Figure 13-29 Noise Summaries for the Third PSS PNoise Analyses

Device	Param	Noise Contribution	% Of Total
/I77	PA_PB	2.3769e-09	83.70
/I2	IQ_modulator_i	1.53663e-10	5.41
/I2	IQ_modulator_q	1.53663e-10	5.41
/R20	rn	6.568e-11	2.31
/R21	rn	6.568e-11	2.31
/I84	IQ_demodulator	2.42376e-11	0.85
/PORT13	rn	1.19682e-13	0.00

Integrated Noise Summary (in V²) Sorted By Noise Contributors
Total Output Noise = 2.83995e-09
No input referred noise available

Device	Param	Noise Contribution	% Of Total
/I76	PA_BB_i	2.3769e-09	83.69
/I22	IQ_mod_BB_i	3.07294e-10	10.82
/R22	rn	1.31347e-10	4.62
/I82	IQ_demod_BB_i	2.42376e-11	0.85
/PORT9	rn	1.19682e-13	0.00
/I76	PA_BB_q	5.71606e-14	0.00
/I22	IQ_mod_BB_q	7.38993e-15	0.00

Integrated Noise Summary (in V²) Sorted By Noise Contributors
Total Output Noise = 2.83997e-09
Total Input Referred Noise = 0.0148587

The *apparent* disagreement shown in [Figure 13-29](#) on page 626 requires an explanation. Let us examine the noise contributors and try to answer some of the questions we posed earlier.

Noise at the power amplifier output due to noise injected at a passband node:

- Passband model contributors: *PA_PB* and *IQ_demodulator*, port 9.
- Baseband model contributors: *PA_BB_i*, *IQ_demod_BB_i*, port 13.

Passband and baseband counterparts contribute the same amount of noise. However, in the baseband model, from symmetry you see the same numbers if you look at the Q-node. This means the baseband model predicts twice as much total noise due to noise injected between the modulators.

This factor of two is intentionally introduced to maintain the correct signal-to-noise ratio. The baseband model simulates peak signals; the carrier is suppressed. Without the carrier, signal power equals the square of the peak rather than one half of the square. This factor of two is not as arbitrary as it seems. The baseband model predicts the correct noise after demodulation because the passband demodulator model includes an extra factor of two to offset the factor of two inherent in the demodulation process.

Let the modulated carrier be $i(t) \cdot \cos(\omega c \cdot t) - q(t) \cdot \sin(\omega c \cdot t) + \text{noise}(t)$.

Now consider the I-output. To generate the I-output, the demodulator multiplies the signal by $\cos(\omega c \cdot t)$. The only part that propagates through the subsequent filter is $(1/2)i(t) + \text{noise}(t) \cdot \cos(\omega c \cdot t)$. To recover $i(t)$, the passband demodulator model must scale this sum by two. (The baseband demodulator does not need to scale by two to extract the baseband signal because the carrier is suppressed.)

Thus, noise at passband demodulator model output equals $2 \cdot \text{noise}(t) \cdot \cos(\omega c \cdot t)$. The filtered noise power density is then $4 \cdot (\text{input noise density}) / 2$.¹ The filtered output noise density is twice the input noise density.

In the baseband model, doubling the noise injected at the passband nodes was not simply a matter of convenience.

So, to answer questions 1 and 2

- Yes, noise injected at a passband node splits evenly between the two equivalent baseband nodes
- but

1. The factor of 1/2 comes from the cosine.

- Since each split is doubled, the ratio of *peak* signal to total noise equals the true signal-to-noise ratio.

Question 3 is rendered moot by using the noise summary and integrating over the proper band. If the noise analysis only integrated from 1GHz to 1.1GHz, (instead of from 900MHz to 1.1GHz), there would be a mysterious factor of two error.

In the baseband model, phase noise entering on the phase error pin propagates to both the I and Q outputs. In the passband model, the same noise power appears on just the one output. Again, the total noise in the baseband model is twice that of the passband model to maintain the correct signal-to-noise ratio.

Noise injected at the modulator input (resistors and modulator noise):

Total noise in the baseband model due to modulator and input resistor noise is twice what it is from just one phase. Thus, the total noise due to sources on the input sides of the modulators differ by a factor of two. This occurs because the passband model is a real multiplier, which modulates the noise. If the peak signal voltage agrees with the baseband model, the passband modulator model attenuates input noise most of the time. The important thing is that the signal-to-noise ratios in the passband and baseband models agree anywhere in the system.

Now, copy the circuit so you can **remove the capacitors at the modulator inputs** and repeat the last set of noise analyses. You will see that in the passband model, the input resistors, R20 and R21, together contribute twice as much as the baseband counterpart, R22. With the capacitors, R20 and R21 together contribute just as much as R22. Without the capacitors, the input noise is truly white over the frequencies of interest. The same thing happens to the modulator noise itself. [Figure 13-30](#) on page 629 shows the results.

In particular, the modulator now also has noise at twice the carrier frequency and that noise mixes down to the carrier frequency. The baseband model is just that, a baseband model. The answer to question 4 is *no*. The baseband models do not account for noise, or signals, at carrier harmonics. The baseband equivalent noise analysis is valid only if noise injected into the modulators has no power beyond the local oscillator frequency. Phase noise injected at the phase noise pins should also be band limited.

Figure 13-30 Noise Summaries with the input capacitors removed.

Device	Param	Noise Contribution	% Of Total
/I2	IQ_modulator_q	4.22571e-09	29.21
/I2	IQ_modulator_i	4.22571e-09	29.21
/I77	PA_PB	2.3769e-09	16.43
/R21	rn	1.80619e-09	12.49
/R20	rn	1.80619e-09	12.49
/I84	IQ_demodulator	2.42376e-11	0.17

Integrated Noise Summary (in V²) Sorted By Noise Contributors
Total Output Noise = 1.44651e-08
No input referred noise available

Device	Param	Noise Contribution	% Of Total
/I22	IQ_mod_BB_i	4.22561e-09	50.11
/I76	PA_BB_i	2.3769e-09	28.19
/R22	rn	1.80615e-09	21.42
/I82	IQ_demod_BB_i	2.42376e-11	0.29
/PORT9	rn	1.19682e-13	0.00
/I22	IQ_mod_BB_q	1.01619e-13	0.00

Integrated Noise Summary (in V²) Sorted By Noise Contributors
Total Output Noise = 8.43322e-09
Total Input Referred Noise = 9.76997e-11

Baseband Library

This section explains a few assumptions and measurements common to several elements in the baseband library. Except for AM/PM conversion, all model specifications are defined by measurements on the passband models. Most baseband models have a differential passband view. Those that do not have a passband view are meant only for baseband use. The models in the top_dwnPB category are singled ended versions of the differential passband views. The single-ended passband models are used to describe the measurements but the measurements also apply to the differential versions.

Assumptions

All behavioral models require assumptions and the top-down models at hand are no exception. The assumptions are summarized below:

- The non-linear RF components are memoryless.
- Noise in the models is white additive Gaussian noise split equally between the two output pins. *I* and *Q* noise sources in any model are uncorrelated.
- Except for loading effects, signal flow is unilateral.
- Terminal impedances are purely resistive.
- The following is a single assumption stated three ways:
 - Only third order non-linearities matter; no even order non-linearity matters.
 - Harmonics of the carrier between any two baseband models is negligible
 - Baseband models need not simulate IP2¹ or DC offsets.
- All mixers, including those in modulators and demodulators, can be modeled as a static non-linearity followed by an ideal multiplier.
- All local oscillators are sinusoidal.

1. The basic problem is that "even" non-linearities do not produce output power at the carrier fundamental. Baseband models only model effects like IP3 because third order distortion affects the fundamental. Second order distortion only affects the fundamental at the output of a subsequent cascaded non-linear device. You can build one baseband model of the two cascaded non-linear devices and simulate how second order distortion in the first device affects the output of the second device at the fundamental. You can not cascade individual baseband models of the two devices and observe any second order effects.

Compatibility

- All baseband models except the baseband signal generators and instrumentation blocks are compatible with SpectreRF.
- All models are written in Verilog-A.

Library Categories

The rLib has six categories.

1. **Original_RFAHDL_lib:** This the original RF AHDL library originally released in 4.4.3. It is documented in Appendix D.
2. **bot_upBB:** This category contains only the J-models in this release.
3. **measurement:** This category contains models used to make measurements and includes the instrumentation blocks and baseband signal generators. They are not part of an RF architecture. These models only facilitate measurements and diagnostics.
4. **testbenches:** This category contains the test circuits mentioned in this document when defining model specifications. Where possible, the models are specified in terms of standard RF measurements. The most precise way to describe a measurement is with a test circuit, set up instructions, and sample measurements. The circuits in this category serve that purpose.
5. **top_dwnBB:** This category contains models of common architectural function blocks. The default view of the models in this category is the baseband view but all models in this category have a differential passband view called “veriloga_PB”. The only exceptions are the BB_loss and VGA_BB models. The BB_loss model is meant only for baseband analysis. The VGA_BB is an engineering release of a variable gain amplifier. It is not documented or supported in this release and is subject to change in future releases.
6. **top_dwnPB:** This category contains single-ended passband versions of the baseband models.

Inputs and Outputs on Baseband Models

Except where noted, all baseband models have inputs and outputs as shown in [Figure 13-31](#) on page 632. Do not confuse the baseband models with two-port S-parameter models.

Figure 13-31 Baseband Model, Inputs and Outputs



$$\rho = \text{Sqrt}[i*i+q*q]$$

$$\theta = \text{atan2}[q/i]$$

When a device loads or drives adjacent devices with finite resistances, the terminals on the symbol are wider apart than when they load/drive adjacent models with ideal resistances (zero or infinite resistance).

Some Parameters Common to Several Models

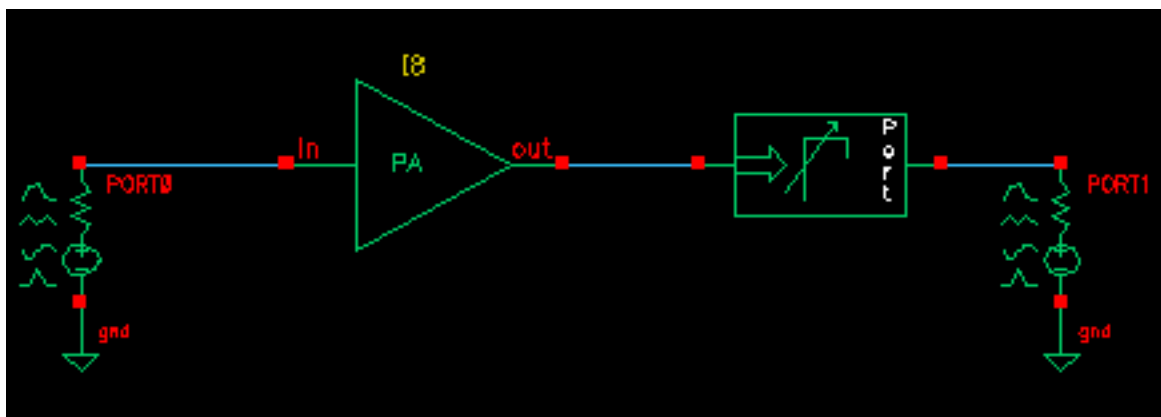
This section defines several parameters used for the baseband (*topdown_BB*) and passband (*topdown_PB*) behavioral models.

- Aside from AM/PM parameters, all parameters specify both passband and baseband models of a function block.
- The AM/PM conversion parameters specify the baseband models only.

Available Power Gain Parameter

Available power gain equals $10 * \text{Log}(\text{output power} / \text{input power})$ when the amplifier's load is equal to the amplifier's output resistance. The test circuit in [Figure 13-32](#) on page 633 is listed as *ava_pwr_gain* in the *testbenches* category in *rfLib*.

Figure 13-32 The *ava_pwr_gain* Circuit



The *ava_pwr_gain* test circuit is set up to compute constant power contours. As you would expect, maximum power transfer occurs when the load and output impedances are matched. The Port Adapter inserts reactive elements into the signal path to load the amplifier with the specified reflection coefficient.

Figure 13-33 on page 634 shows a Smith Chart that displays how the load power varies with the load reflection coefficient. The load pull contours were computed by sweeping the “pp” parameter in the PSS analysis and sweeping the “mm” parameter with the Parametric Tool. “pp” is the phase of the reflection coefficient. “mm” is the magnitude of the reflection coefficient. The load reflection coefficient is defined with reference to the amplifier output resistance, which is 300 Ohms in this case. The amplifier input resistance is 20 Ohms. The input source resistance is 50 Ohms. The amplifier 1dB compression point is set high enough to make the amplifier linear. The available power gain parameter is 20 dB. To generate the load pull contours you must save the current flowing into the Port Adapter. Also save the current flowing into Port0.

Placing the cursor on the smallest contour on the Smith Chart you can see that the amplifier delivers a maximum power of 81.63 mW to an optimum load of 300 Ohms (reflection coefficient = 0). If you plot the magnitude of the power coming from the input port against the sweep variable (phase of the reflection coefficient) you will find that input power equals 816.3 uW, independent of load, as shown in Figure 13-34 on page 635. The ratio of maximum output to input power equals 100, or 20 dB, as specified.

Figure 13-33 Smith Chart

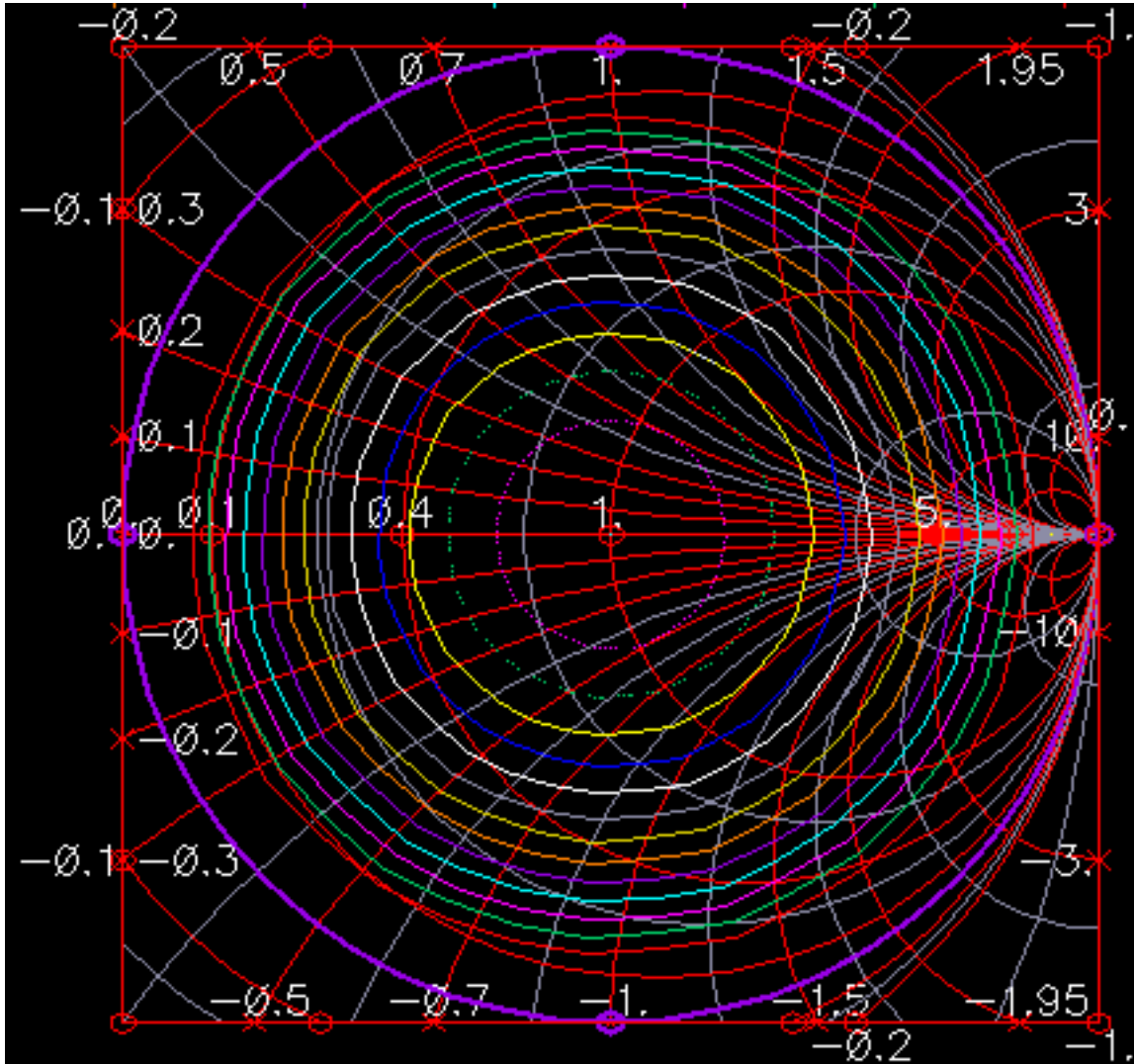
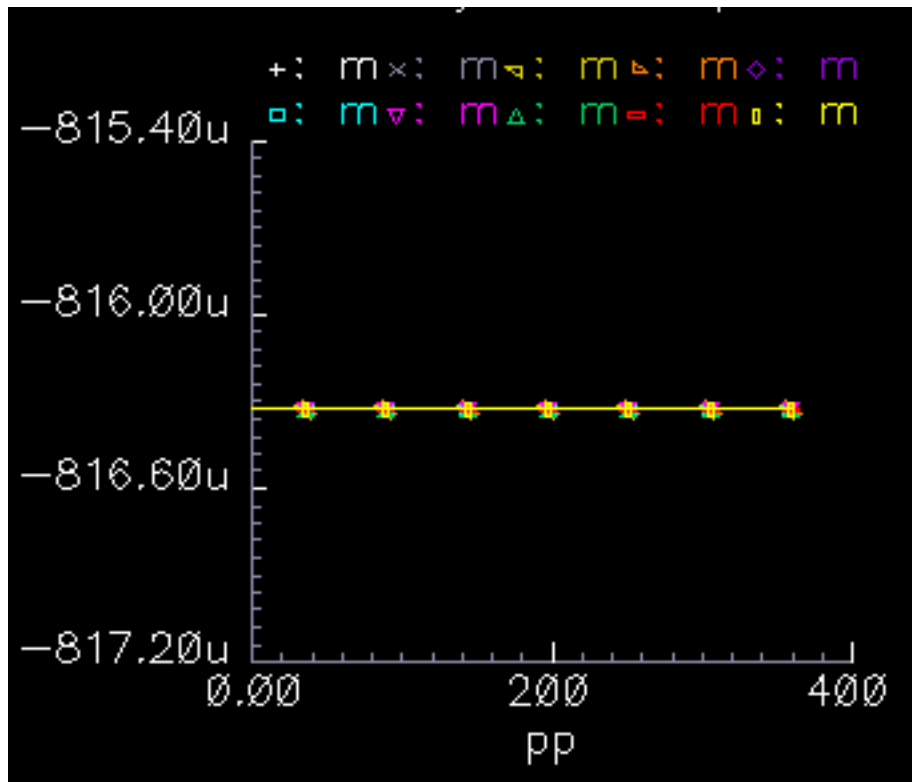


Figure 13-34 Input power



Note that the voltage gain in this test circuit does not equal 10 because the amplifier's input and output resistances are different. You can verify that the ratio of the output to input voltage is $10 \cdot \text{Sqrt}[R_{out}/R_{in}]$ where R_{out} is the amplifier output resistance and R_{in} is the amplifier input resistance. This assumes the amplifier is not driven into non-linear operation.

Input and Output Resistance Parameters

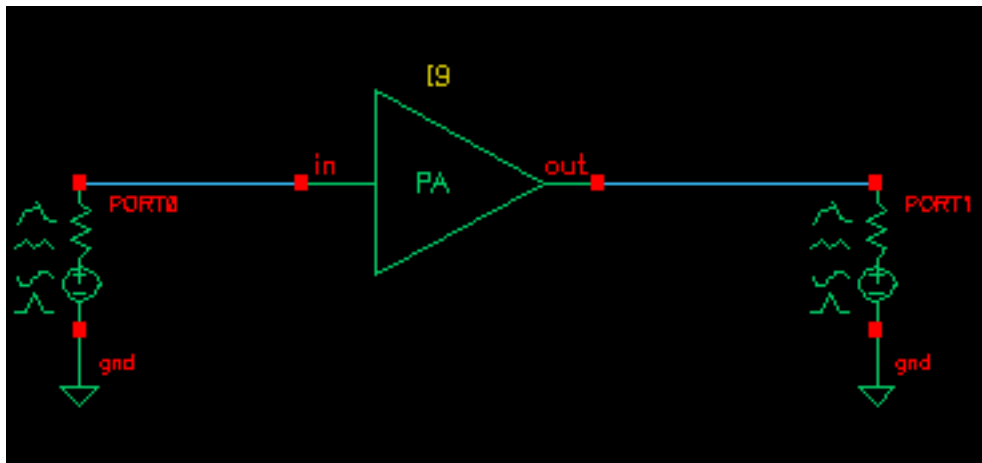
The input and output resistances specify the current drawn by the associated terminals as a linear function of terminal voltage. There is no test circuit for terminal resistances since the definition is so simple.

Output One dB Compression Point Parameter

The 1 dB compression point specifies a saturation non-linearity. It is the output power in dBm where the output power falls 1dB below the power extrapolated linearly from the amplifier's linear region of operation.

The test circuit in [Figure 13-35](#) on page 636 is listed as *one_db_cp* in the *testbenches* category in *rfLib*.

Figure 13-35 The *one_db_cp* Circuit

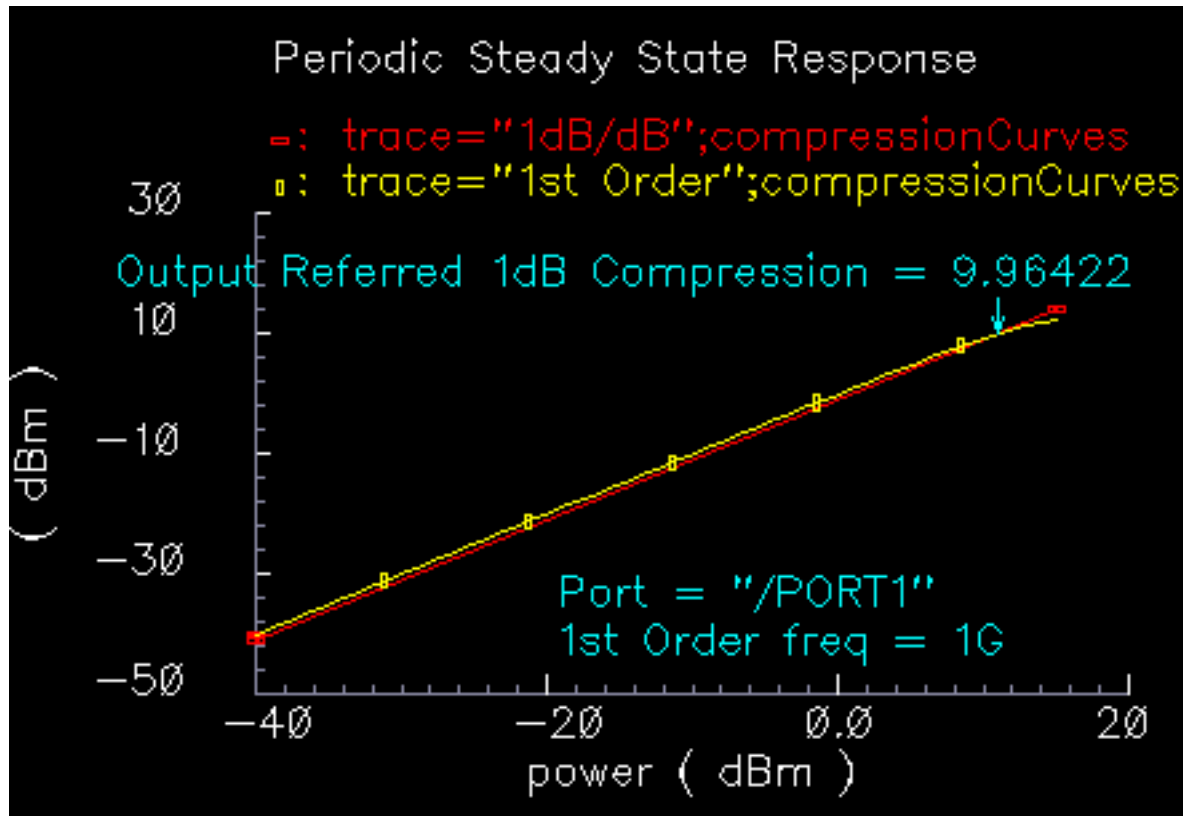


In the *one_db_cp* test circuit, *power* is the dBm of power delivered by the leftmost port. The available power gain is 0dB. The 1 dB compression point is 10dBm. The input and output resistances are 50 Ohms and so are the port resistances.

To measure the 1dB compression point, perform a swept PSS analysis. Sweep *power* from -40dBm to 15dBm in 50 linear steps. The output referred 1 dB compression point is computed for the 1st harmonic with an *Extrapolation Point (dBm)* of -40. Click on the rightmost Port device to display the output. [Figure 13-36](#) on page 637 shows the results.

The specified output referred compression point is 10dBm. The measured value is 9.964dBm, which is fairly close to the specified value. The measured 1db compression point is as specified only when the driving source resistance matches the amplifier input resistance and the load port resistance matches the amplifier's output resistance. In all compression point and IPN calculations input power is computed from the maximum power the input Port can deliver, not from an actual power measurement. If you mismatch either terminal you will not measure the specified compression point.

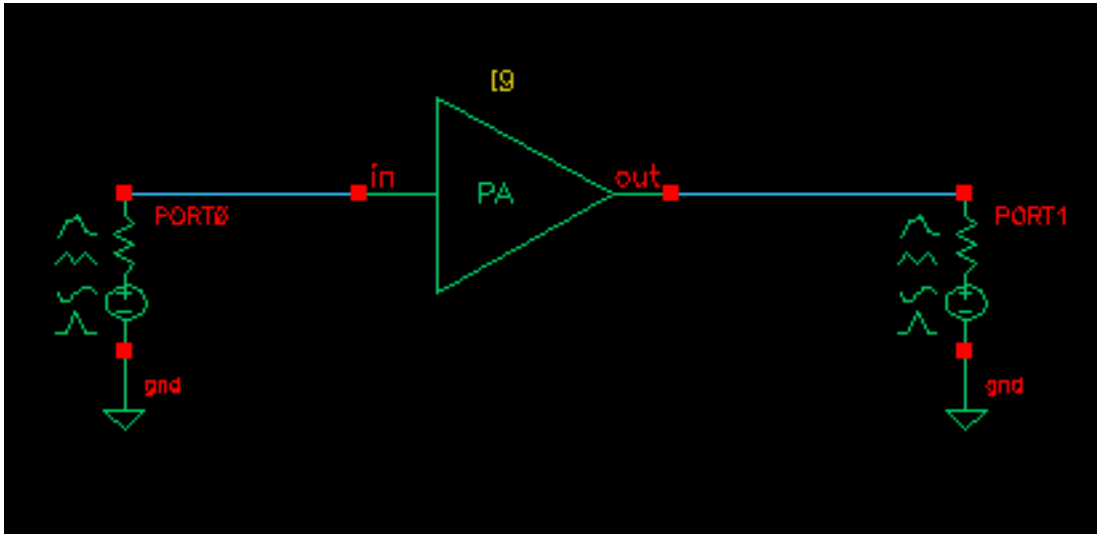
Figure 13-36 Results



Noise Figure Parameter

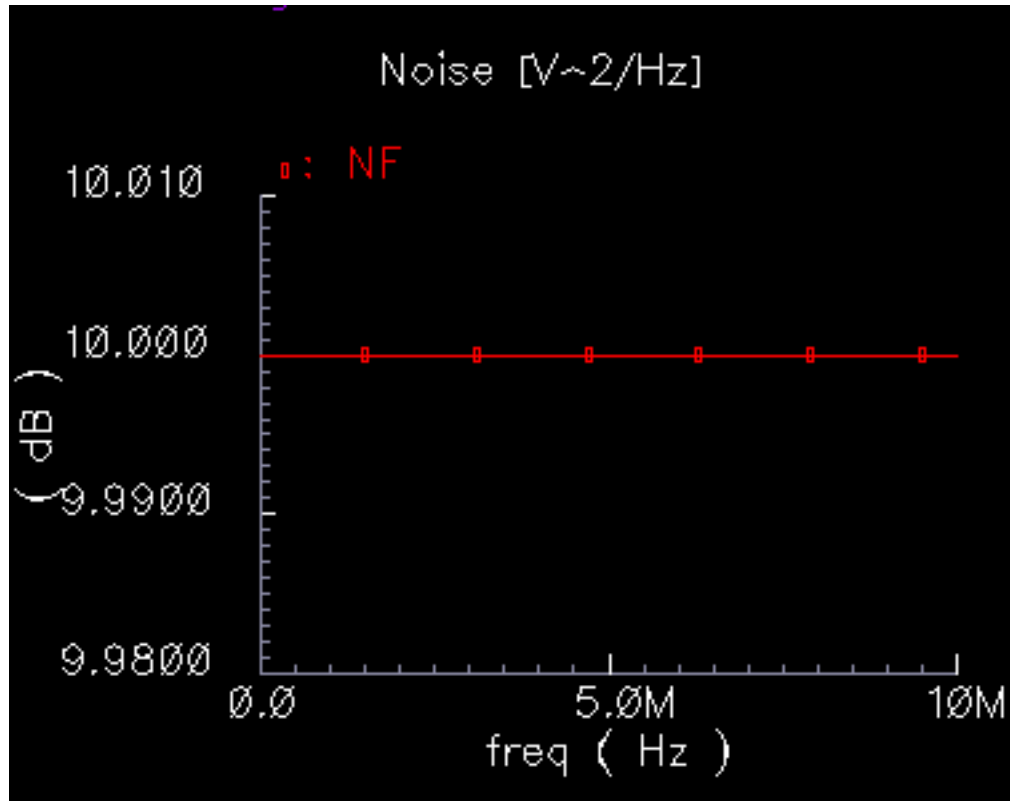
Noise figure is calculated as the input signal-to-noise ratio divided by the output signal-to-noise ratio. The test circuit for defining this parameter is similar to the *one_db_cp* test circuit. The circuit in [Figure 13-37](#) on page 638 is listed as *noise_figure* in the *testbenches* category of the *rfLib*.

Figure 13-37 The noise_figure Circuit



The specified noise figure is 10dB. A Spectre Noise analysis produces the noise figure shown in [Figure 13-38](#) on page 639. To measure the specified noise figure, the driving port resistance must match the amplifier's input resistance. The port at the output does not have to match the amplifier's output resistance but the Port impedance should be resistive. The input probe is the leftmost port, the output port is the rightmost port. Since the model is static, you can compute noise figure over any frequency interval.

Figure 13-38 Noise Figure Results

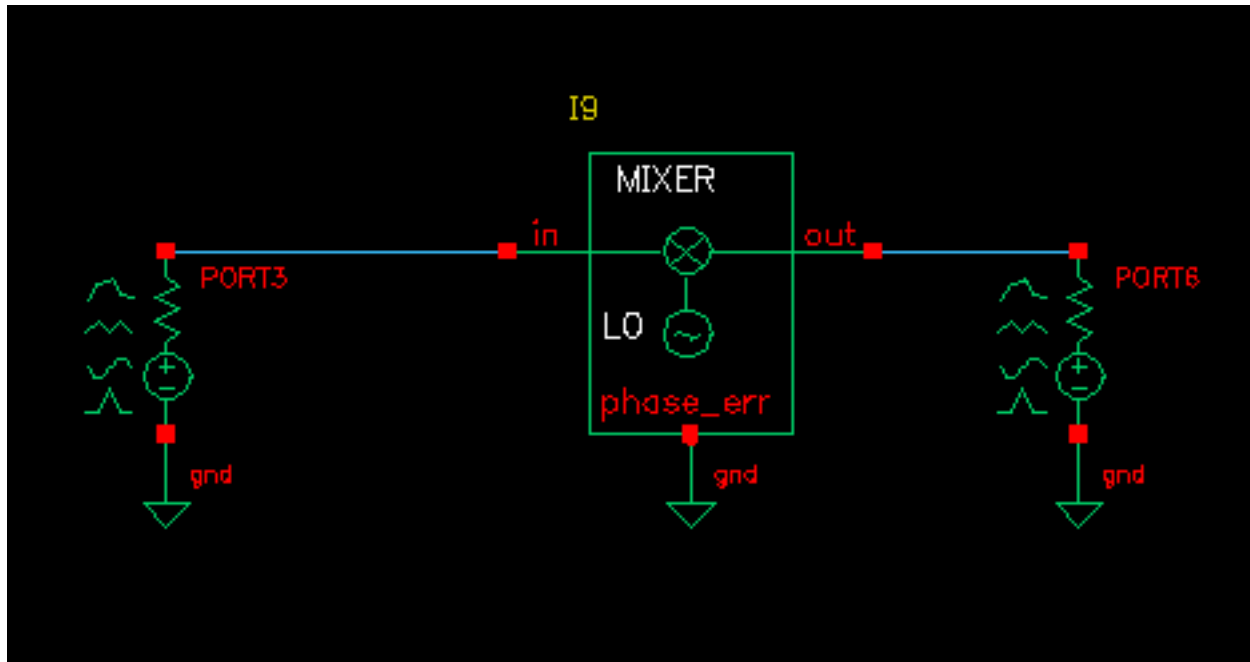


IP3 Parameter

IP3 is measured with a two-tone test. One tone is the fundamental PSS frequency while the other is the frequency in a single point PAC analysis. IP3 is defined as the input power level in dBm where the extrapolated power in one of the third order intermodulation terms equals the extrapolated power in the fundamental term. As with the 1 dB compression point measurement, input and output terminals must be matched to the source and load respectively.

The IP3 specification is demonstrated step by step on the mixer model because the mixer IP3 measurement can be confusing. [Figure 13-39](#) on page 640 shows the test circuit. The circuit is listed as `mixer_ip3` in the *testbenches* category of the *rfLib*.

Figure 13-39 The IP3 Test Circuit



1. Recall the circuit and bring up an Analog Environment window.
2. Set up a PSS analysis. Add a Fundamental Tone called “eee” (the name is arbitrary) at 1GHz. The 920MHz tone should already appear in the form. Autocalculate the Beat Frequency. It should be 40MHz. Use 2 Output harmonics. Sweep the PSS analysis using the “power” variable. Sweep power from -100 to 0 in 10 linear steps.
3. Set up a PAC analysis at 921MHz. Select the sidebands from an Array of indices. Add indices -21 and -25. **Why select the -21 and -25 sidebands?** Recall from the assumptions, the non-linearity occurs before the frequency translation. The input tones to the non-linearity are the large 920Mhz tone and the small signal 921MHz tone. In an IP3 measurement only one tone must be large, the other can be small. PAC analysis performs small signal perturbations on the PSS solution. One perturbation term exiting the non-linearity appears at 921MHz, right where is started. One of the third order intermodulation perturbation terms exiting the non-linearity appears at $2*920-921 = 919$ MHz. The ideal mixer, driven by a pure 1GHz local oscillator, translates the 921MHz tone to $921-1000=-79$ MHz while translating the 919MHz tone to $1000-919=81$ Mhz. A single point 921MHz PAC analysis produces tones displaced from harmonics of the fundamental by 921MHz. The PAC sidebands specify which harmonics to use. You save the 79MHz tone by saving the -25th sideband because the fundamental frequency is 40Mhz and $921 - 40*25 = -79$ MHz. You save the 81MHz tone by saving the -21 sideband because $921-40*21 = 81$ MHz. [Figure 13-40](#) on page 641 shows the PAC setup.

4. Run the analysis.
5. Do a Direct Plot of the PAC results. To do this, set up the PSS results form as shown in [Figure 13-41](#) on page 642.
6. In the Composer window, click on the output Port. You should see the results as shown in [Figure 13-42](#) on page 643.

Figure 13-40 PAC Setup

Sweeptype **Sweep is Currently Absolute**

Frequency Sweep Range (Hz)

Freq

Because the sweep section of the PSS analysis is active, only a single point for this analysis is currently supported.

Sidebands

Array of indices

Currently active indices

Additional indices

Figure 13-41 PSS Results Form

pss pac

Function

Voltage Current
 IPN Curves

Select

Circuit Input Power Single Point
 Variable Sweep ("power"

"power" ranges from -100 to 0

Extrapolation Point (dBm)
 (Defaults to -100)

 Order

3rd Order Sideband

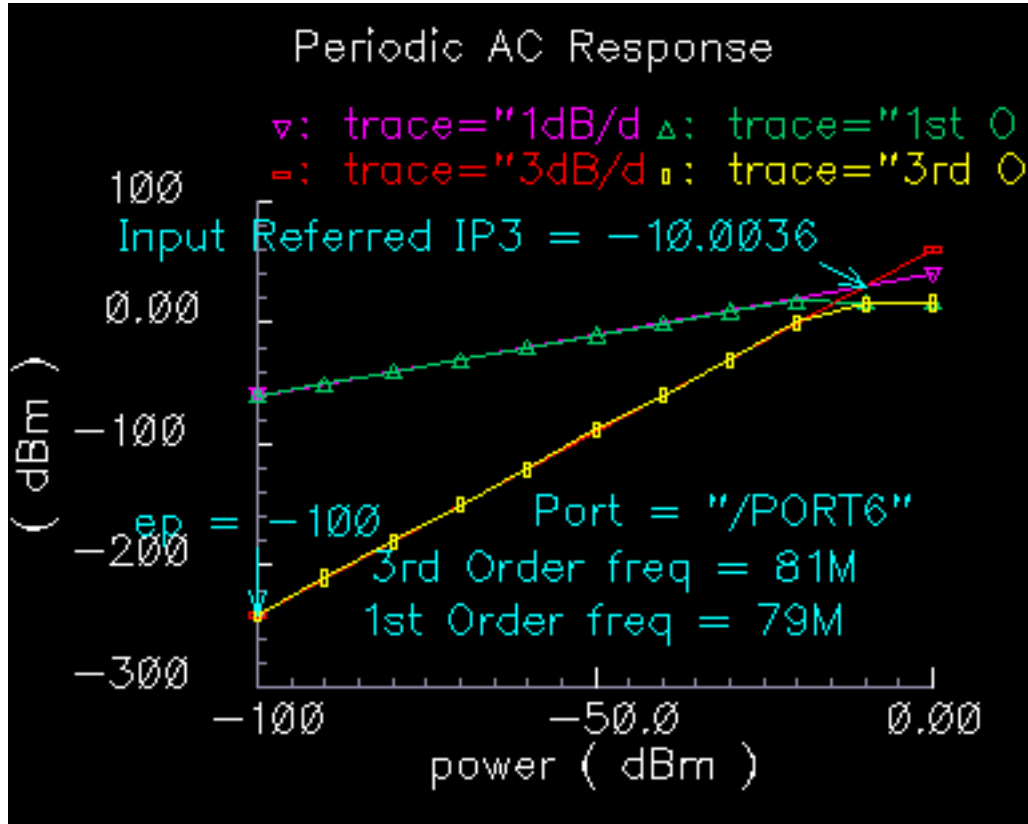
-25	79M
-21	81M
0	921M

1st Order Sideband

-25	79M
-21	81M
0	921M

> Select Port on schematic...

Figure 13-42 IP3 Results



The measured IP3 is, -10dBm, as specified. The measured IP3 is as specified only if the input port resistance matches the input resistance of the device-under-test. Other input resistances will produce a measured IP3 different than the one specified.

You can measure IP3 of an LNA by replacing the mixer with an LNA and ensuring the input terminal remains matched. In this example, remove the 1GHz Fundamental Tone from the PSS analysis. The Beat Frequency should now be 920MHz. In the PAC set up, change the additional indices from -21 and -25 to -1 and -2. After the analysis completes, set up the PSS Results form as shown in [Figure 13-43](#) on page 644. As before, the input referred IP3 is 10dBm, as specified. [Figure 13-44](#) on page 645 shows the LNA IP3 results.

Figure 13-43 PSS Results Form for the LNA

PSS Results

OK Cancel Help

Plot Mode Append Replace

Analysis

pss pac

Function

Voltage Current
 IPN Curves

Select

Circuit Input Power Single Point
 Variable Sweep ("power")

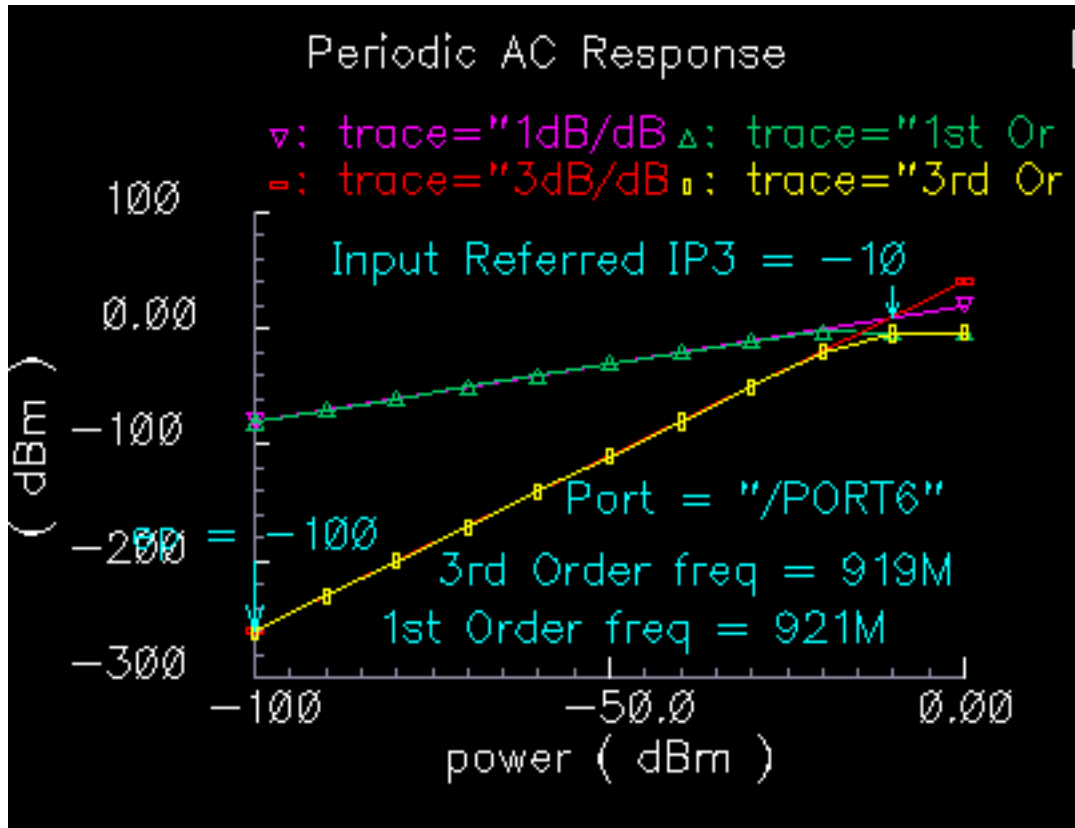
"power" ranges from -100 to 0
 Extrapolation Point (dBm)
 (Defaults to -100)

Order

3rd Order Sideband		1st Order Sideband	
-2	919M	-2	919M
-1	1M	-1	1M
0	921M	0	921M

> Select Port on schematic...

Figure 13-44 Results for the LNA



AM/PM Conversion Parameters

Only the baseband models include the 4 parameters for AM/PM conversion.

am/pm sharpness: Defines how steep the output phase shift changes are with respect to input power.

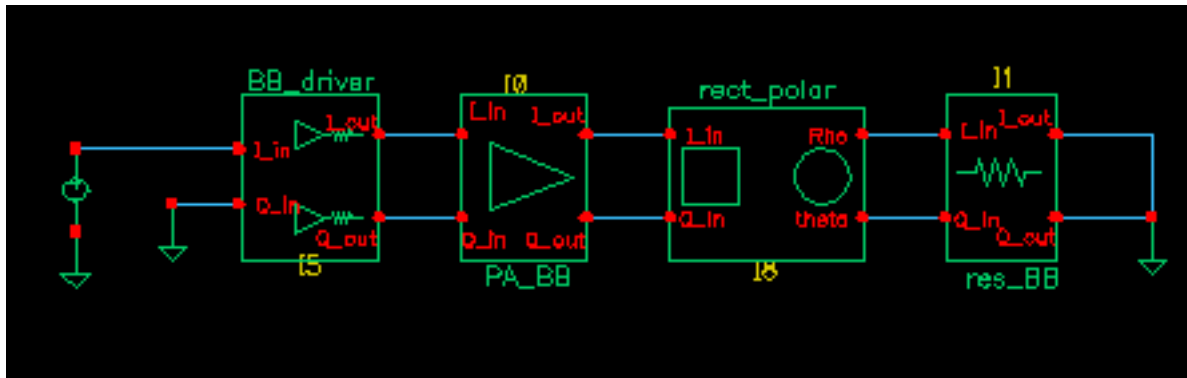
|radians|@1db cp: Defines the absolute value of the output phase shift at the 1dB compression point for power amplifiers. This is the phase shift at an arbitrary output power level for some models.

|radians|@big input: Defines the absolute value of the output phase shift as input power goes to infinity (if it could go to infinity).

{1,0,-1} for {cw,none,ccw}: Defines the direction of the phase shift. 1 for clockwise, 0 for no phase shift, -1 for counter clockwise.

The test circuit in [Figure 13-45](#) on page 646 is listed as *am_pm_test_ckt* in the *testbenches* category in *rfLib*.

Figure 13-45 The *am_pm_test_ckt* Circuit

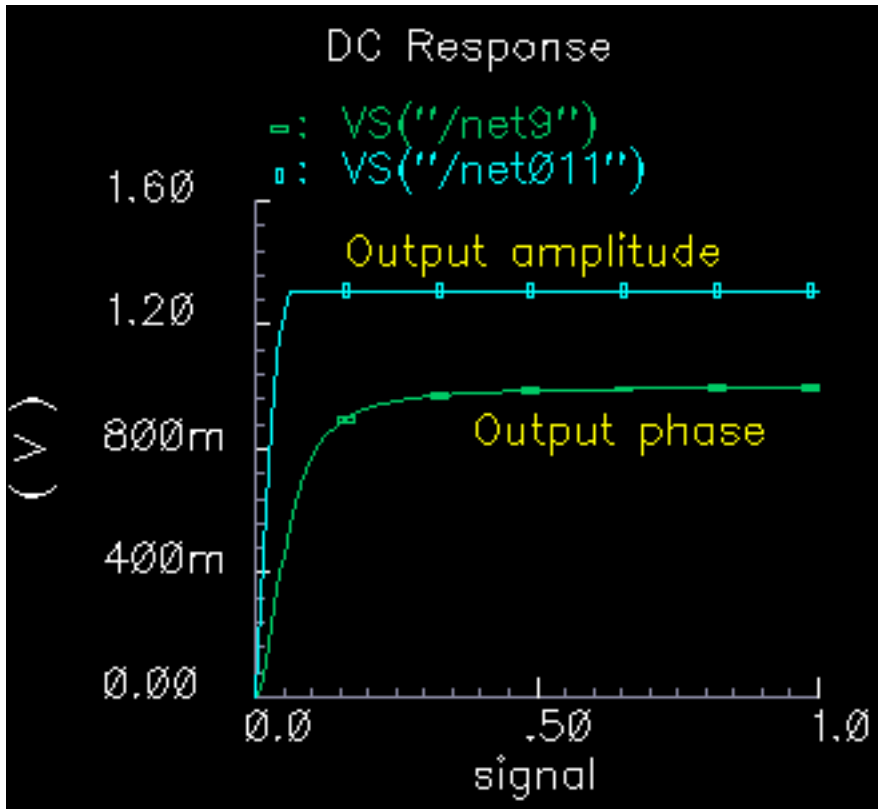


In the *am_pm_test_ckt* test circuit

- The first block (*BB_driver*) scales the control voltage generated by the leftmost element so that the output equals the specified dBm when the control voltage equals 1 volt. This is done so you can specify maximum dBm but still sweep linearly from zero signal.
- The *PA_BB* block is a power amplifier.
- The *rect_polar* block transforms the rectangular description of the baseband signal into polar coordinates so you can observe the phase shift and output signal level directly.

[Figure 13-46](#) on page 647 shows the output amplitude and phase as functions of the input signal level. Generate the plots in [Figure 13-46](#) on page 647 with a swept DC analysis. Sweep the *signal* variable from 0 to 1 in 200 linear steps and display the *rect_polar* outputs.

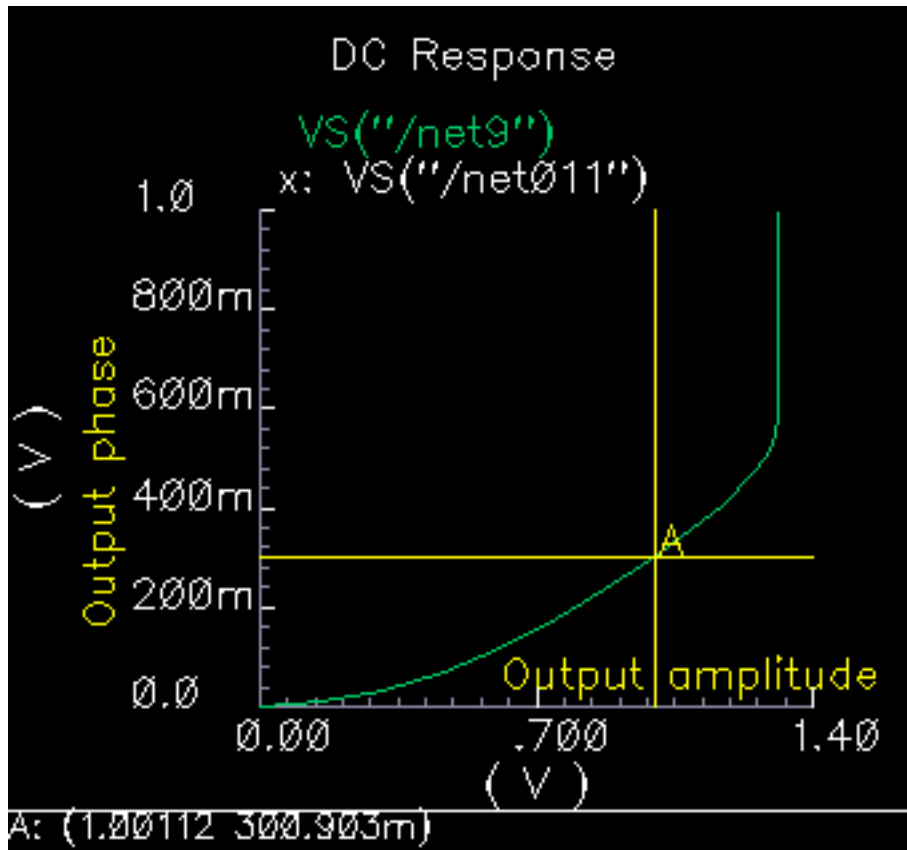
Figure 13-46 Output Amplitude and Phase



By changing the x-axis to be the output amplitude trace, you can confirm that the phase shift at the output referred 1 dB compression point of 10dBm (or 1 volt peak across a 50 ohm load) equals 0.3 radians, as specified. [Figure 13-47](#) on page 648 shows the plot.

Note that the measured power across the load is as specified only when the load matches the amplifier output resistance. If you mismatch the load you will not measure the specified phase shift at the specified output power level.

Figure 13-47 Output phase versus output amplitude



In the following three figures, output phase is plotted against input signal level. Each plot shows the effect of one of the AM/PM conversion parameters. You can generate the plots by applying the Parametric Tool to the existing analysis.

- [Figure 13-48](#) on page 649 shows the effect of the *|radians|@1 db cp* parameter. "rad_cp" was swept from 10m to 100m in 5 linear steps.
- [Figure 13-49](#) on page 649 shows the effect of the *am/pm sharpness* parameter. "sharpness" was swept from 1 to 6 in 5 linear steps.
- [Figure 13-50](#) on page 650 shows the effect of the *rad_inf* parameter. "rad_inf" was swept from 0.5 to 3 in 5 linear steps.

Figure 13-48 Output Modified by the |radians|@1 db cp Parameter

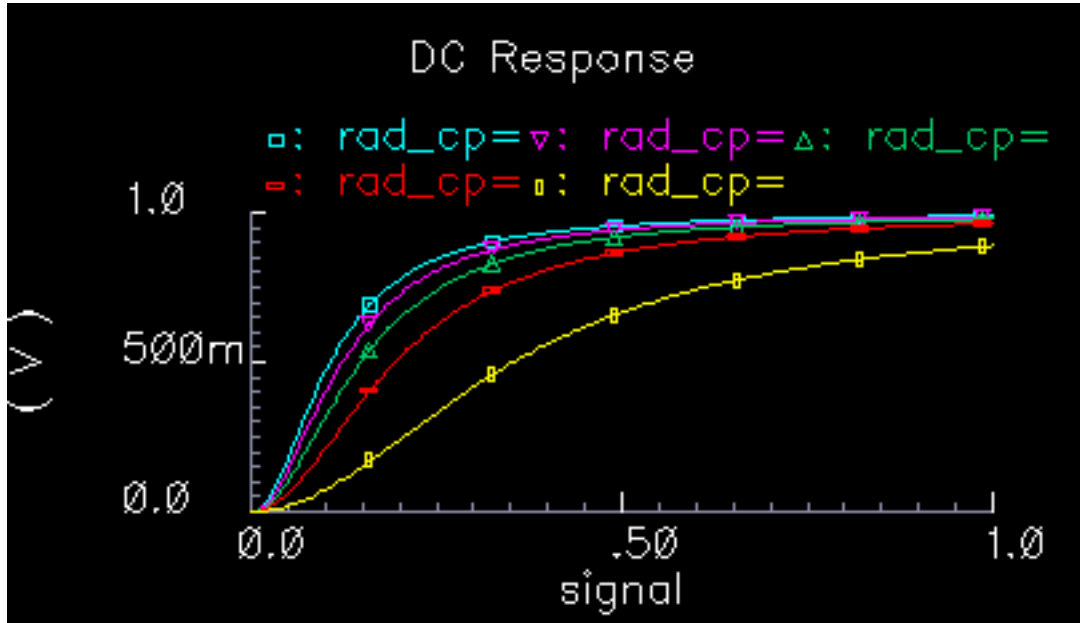


Figure 13-49 Output Modified by the Sharpness Parameter

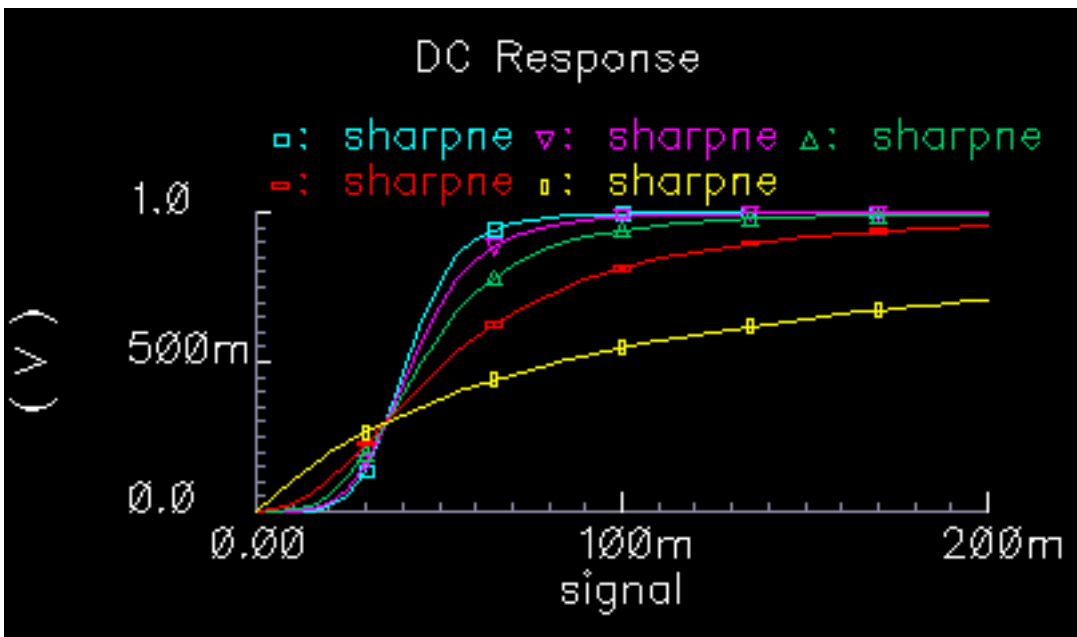
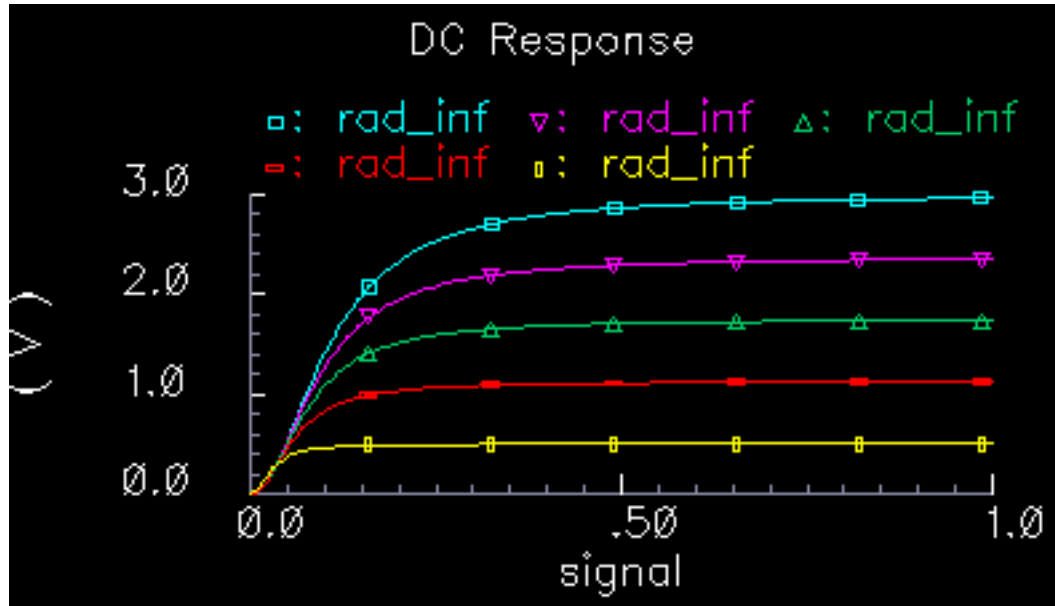


Figure 13-50 Output Modified by the rad_inf Parameter



Library Description

The new models are listed in the table and described in the following sections.

Top-Down Model	Model Name in <i>rfLib</i>
Power Amplifier	<i>PA_BB</i>
Low Noise Amplifier	<i>LNA_BB</i>
I/Q Modulator	<i>IQ_mod_BB</i>
I/Q Demodulator	<i>IQ_demod_BB</i>
Upconverting Mixer	<i>up_cnvrt</i>
Downconverting Mixer	<i>dwn_cnvrt</i>
Inductor	<i>ind_BB</i>
Capacitor	<i>cap_BB</i>
Resistor	<i>res_BB</i>
Linear Time-Invariant Filters	
Butterworth Bandpass Filter	<i>BB_butterworth_bp</i>
Butterworth Bandstop Filter	<i>BB_butterworth_bs</i>
Butterworth Highpass Filter	<i>BB_butterworth_hp</i>
Butterworth Lowpass Filter	<i>BB_butterworth_lp</i>
Chebyshev Bandpass Filter	<i>BB_chebyshev_bp</i>
Chebyshev Bandstop Filter	<i>BB_chebyshev_bs</i>
Chebyshev Highpass Filter	<i>BB_chebyshev_hp</i>
Chebyshev Lowpass Filter	<i>BB_chebyshev_lp</i>
Baseband Shifter Combiner	<i>BB_shifter_combiner</i>
Baseband Shifter Splitter	<i>BB_shifter_splitter</i>
Rectangular-to-Polar Coordinate Transformation	<i>rect_polar</i>
Polar-to-Rectangular Coordinate Transformation	<i>polar_rect</i>
Ideal Transformer	<i>BB_xfmr</i>

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

Top-Down Model	Model Name in <i>rfLib</i>
Loss	<i>BB_loss</i>
Instrumentation Block	<i>comms_instr</i>
Offset Instrumentation Block	<i>offset_comms_instr</i>
Instrumentation Terminator	<i>instr_term</i>
Baseband Driver	<i>BB_driver</i>
Phase Generator	<i>phase_generator</i>

Notes on models involving frequency translation:

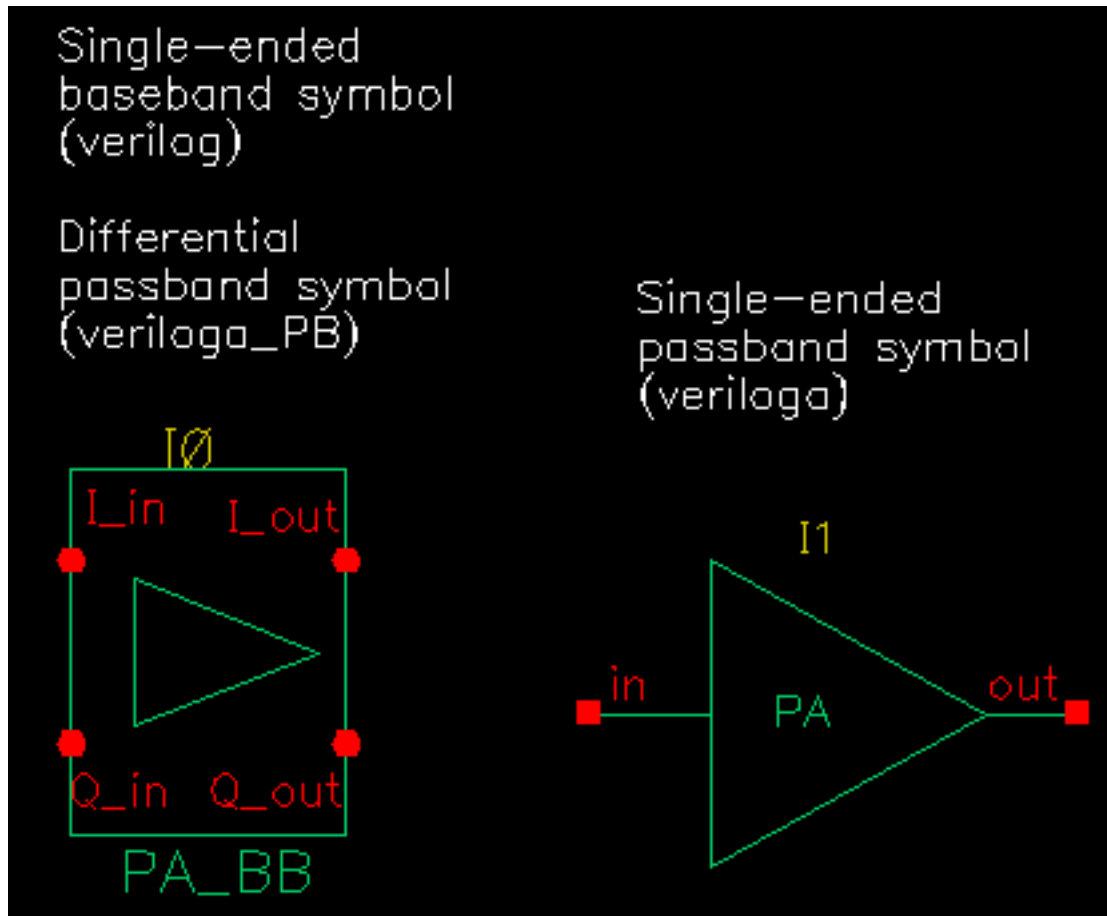
Passband models involving frequency translation have internal oscillators. When running a PSS or Transient analyses, be sure to make Maxstep small enough to prevent aliasing. Future releases of the models will include Boundstep lines to control maxstep from within the models. Envelope Following and PDISTO analyses do not require a non-default maxstep because those analyses require external sources at the same frequencies that exist inside the VerilogA modules. That is because there is currently no way to name sources inside VerilogA modules.

The local oscillators were absorbed into the mixer/modulator/demodulator models to maintain a close relationship between the baseband and passband models. With the ability to use an external source, it would be possible to drive a mixer with non-sinusoidal oscillator signals. Such signals introduce harmonics that the baseband models can not simulate. The first passband models encountered in the top-down flow, called level 1 passband models, are meant to introduce as few new effects as possible. The idea is simplify diagnostics by introducing new effects sequentially. The next level of detail in the behavioral passband models, i.e. level 2 models, are left to the user for now. Level 3 behavioral models are considered unnecessary because they do not offer a significant advantage over device-level models.

Power Amplifier Model

(baseband = PA_BB, passband = PA_BB)

Figure 13-51 Baseband and Passband Power Amplifier Models



The following parameters specify the power amplifier model.

Both Passband and Baseband models.

- available power gain
- input resistance
- output resistance
- output 1db cp in dBm

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

- noise figure

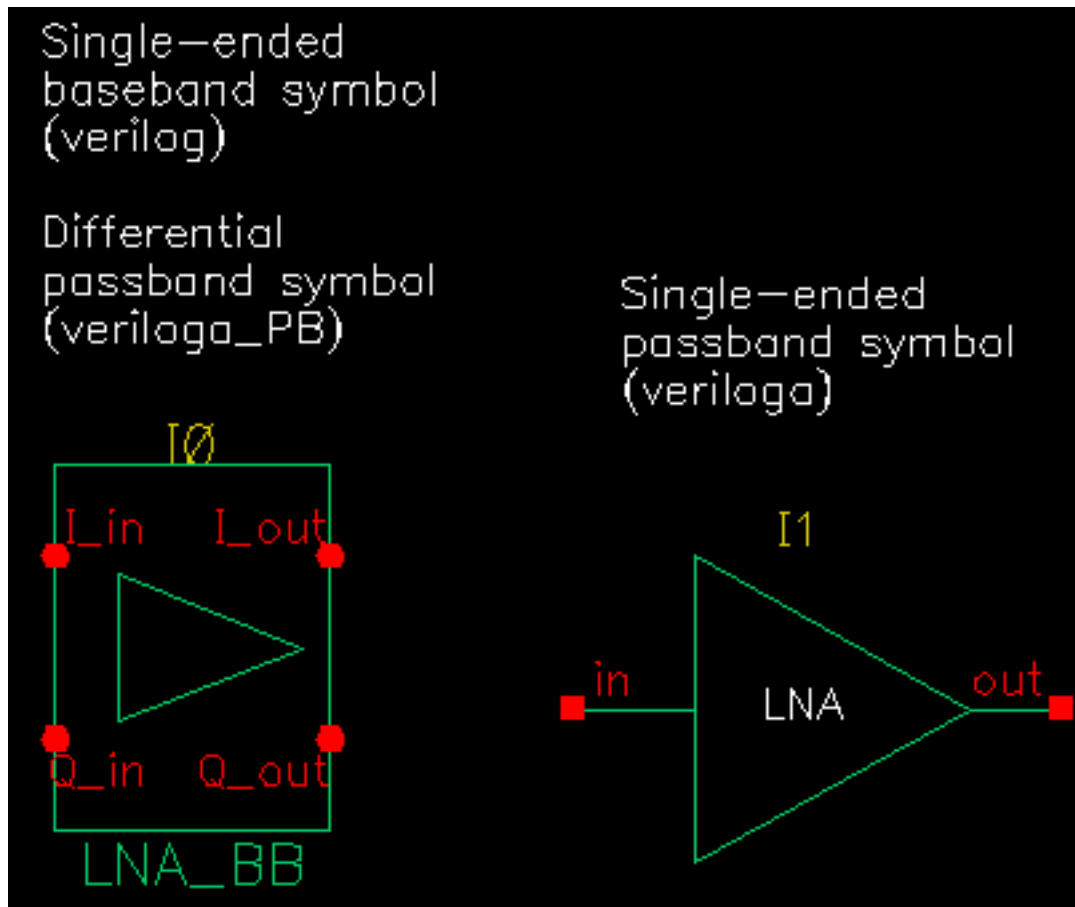
Baseband model only.

- am/pm sharpness
- |radians|@1db cp
- |radians|@ big input
- {1,0,-1} for {cw,none,ccw}

Low Noise Amplifier Baseband Model

(baseband = LNA_BB, passband = LNA_PB)

Figure 13-52 Baseband and Passband Power Amplifier Models



The following parameters specify the low noise amplifier model.

Both Passband and Baseband models.

- available power gain
- input resistance
- output resistance
- input referred IP3[dBm]

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

- noise figure

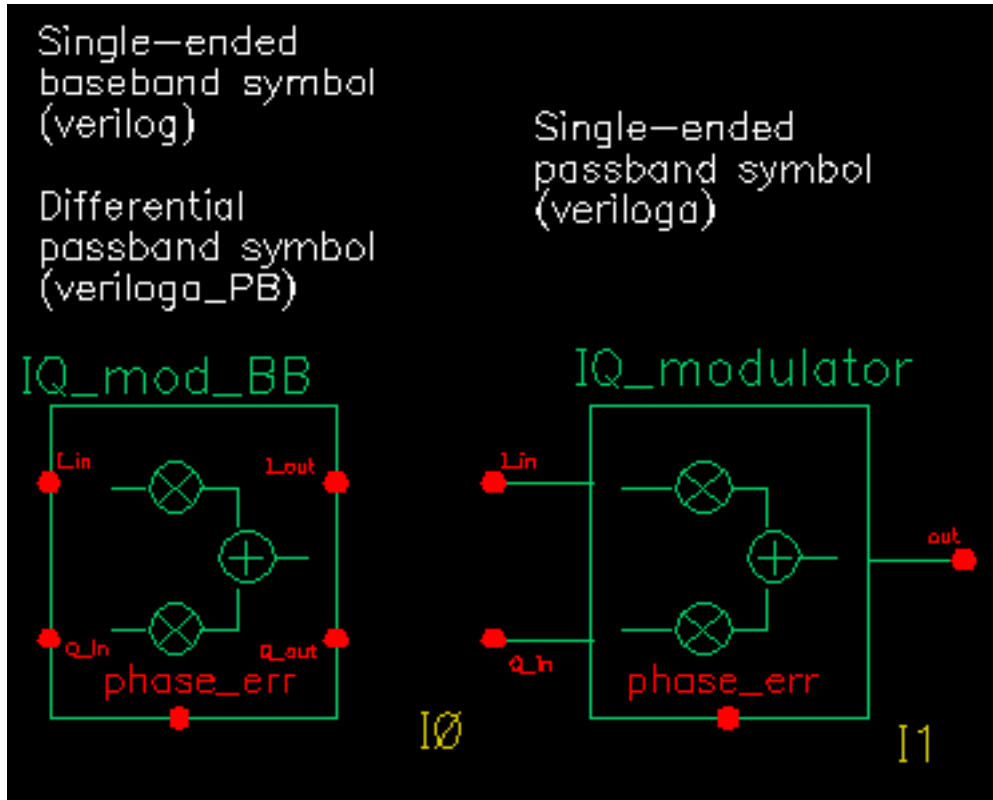
Baseband model only.

- am/pm sharpness
- cmp[dBm] = output power level where the next parameter is defined
- |radians|@cmp = output phase shift at output power of cmp.
- |radians|@ big input
- {1,0,-1} for {cw,none,ccw}

IQ Modulator Models

(baseband = IQ_mod_BB, passband = IQ_modulator)

Figure 13-53 Baseband and Passband IQ Modulator Models



The IQ_modulator converts baseband signals to RF or IF. [Figure 13-54](#) on page 659 summarizes exactly what the passband IQ modulator model does. The only difference between the baseband and passband models is carrier suppression. The non-linear functions, g_i and g_q , are specified by their available power gain and 1dB compression points just as in the power amplifier. The functions γ_i and γ_q characterize AM/PM effects in each mixer and are specified by the same parameters that specify power amplifier AM/PM conversion. Since noise is always added at the input, and the input is at baseband in this case, the noise sources are not doubled as they are in the power amplifier or LNA models. Noise figure is defined with reference to one input. Noise is injected at both inputs but the noise injected at just one input alone produces the specified noise figure. Thus, the noise figure parameter should be interpreted as noise figure per input. This model also includes a parameter called "quadrature error" which specifies how far away the two local oscillator signals are from being exactly in quadrature.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

"Phase error" is the voltage on the phase error pin. The phase error pin has a fixed noiseless resistive input impedance of 50 ohms. The phase error pin can be used to introduce a dynamic phase error or phase noise. Phase noise can be fed into the phase error pin from a phase-domain PLL model or from a Port. Noise in Port models can be specified either by the internal resistance or by a data file that tabulates a power spectral density. The phase error pin can also be driven by a ramp or circular integrator output to model a frequency error between the incoming carrier and local oscillator.

The following parameters specify the IQ modulator. The available power gain and one dB compression point are explained first. The effects of the phase_error pin and the quadrature error parameter are discussed at the end of this section.

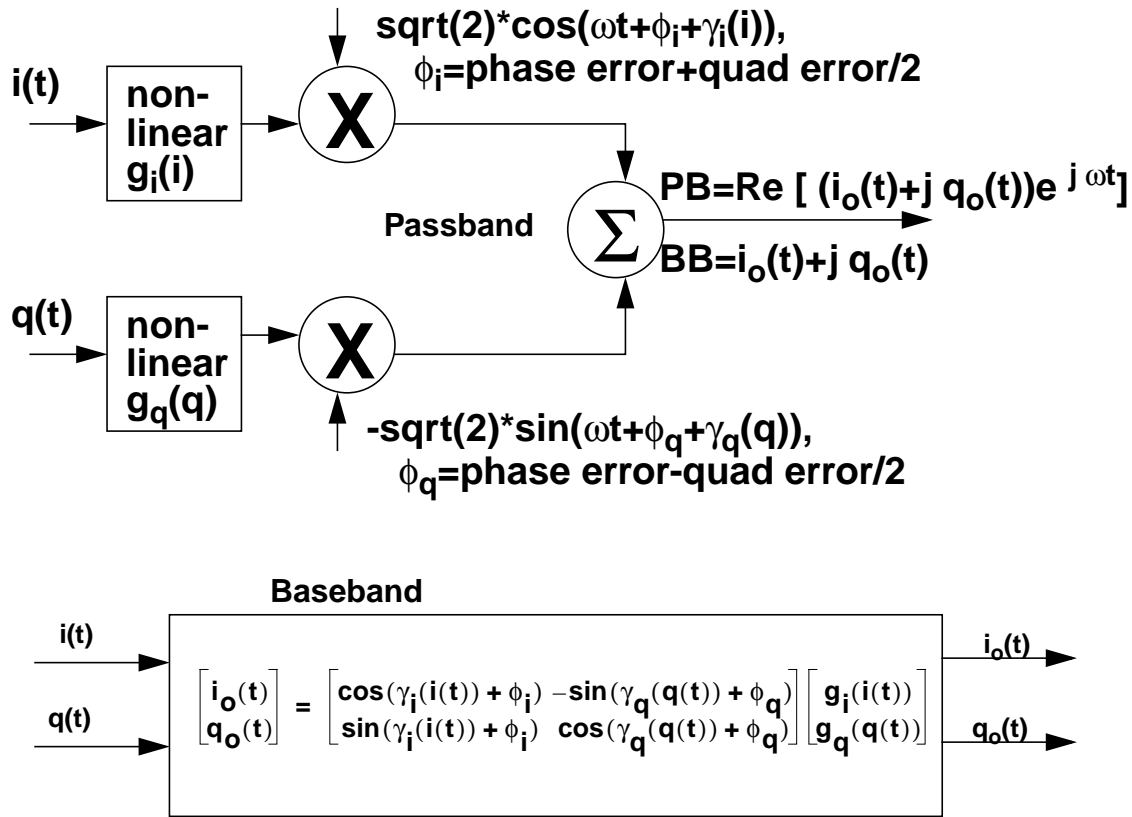
Both Passband and Baseband models.

- available I-mixer gain[dB]
- available Q-mixer gain[dB]
- input resistance
- output resistance
- I-output 1dB CP[dBm]
- Q-output 1dB CP[dBm]
- noise figure [dB]
- quadrature error

Baseband models only.

- I-sharpness factor
- Q-sharpness factor
- Q-radians@Q_cmp
- I-radians@I_cmp
- I-radians@big I-input
- Q-radians@big-input
- {1,0,-1} for {cw,none,ccw}
- {1,0,-1} for {cw,none,ccw}

Figure 13-54 IQ Modulator Calculations



Available power gain and 1dB compression point:

Available power gain of the IQ-modulator is best explained with an example. Recall the circuit called "mod_1dbcp" listed in the testbenches category of the rLib. The schematic contains two disjoint circuits. One shows how not to measure gain and compression point, the other shows the proper measurement.

1. Set up a pss analysis. Both test circuits will run in the same simulation. The beat frequency will be 100MHz. Save the first and 11th harmonics. In the options, set maxstep to 50ps. Sweep the variable, "power", linearly in 50 steps from -40 to 15.
2. After the analysis completes, plot the output referred 1dB compression point of the top circuit using -40 dBm as the Extrapolation point. First select the 11th harmonic (1.1GHz) and click on the output port in the top test circuit, the bad test circuit. Note that the linear

gain is 3dB lower than specified, as is the output referred 1dB compression point. The gain was specified as zero dB and the 1dB compression point was 10 dBm. The error arises from the fact that the input signal power splits between upper (1.1GHz) and lower (900MHz) sidebands but the Analog Artist measurement only looks at one output sideband. The bottom test circuit resolves the ambiguity by defining the gain of the IQ-modulator as the gain from the baseband input to an ideally-demodulated baseband output. The bottom test circuit follows the IQ-modulator with an ideal IQ-demodulator. The gain of the demodulator is zero dB and the 1dB compression point is high enough to render the demodulator distortionless.

3. Repeat the steps for plotting the 1dB compression point but this time chose the first harmonic and select the output port that loads the bottom circuit. Select the first (100MHz) harmonic and plot the 1dB compression point again. Now you should see a 1dB compression point plot that reflects the specified parameters of the IQ-modulator. The gain is now also correct, which can be computed from the ratio of the output to input power well below the compression point. [Figure 13-56](#) on page 661 shows such a plot.

Figure 13-55 1db Compression Point Test Circuit

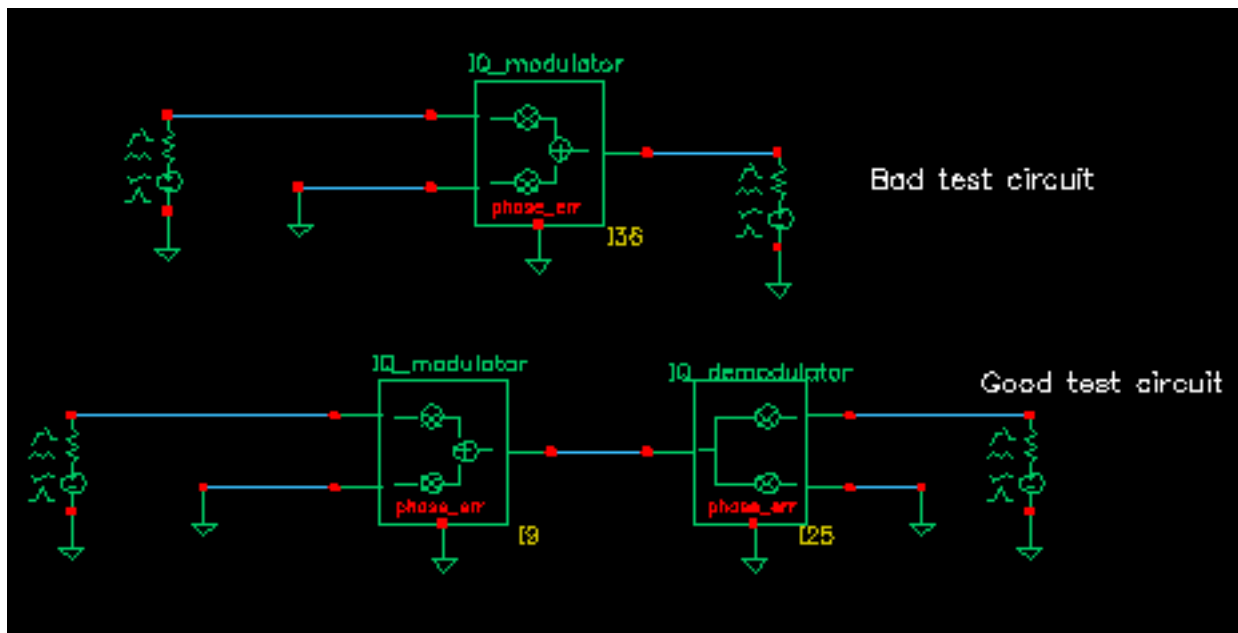
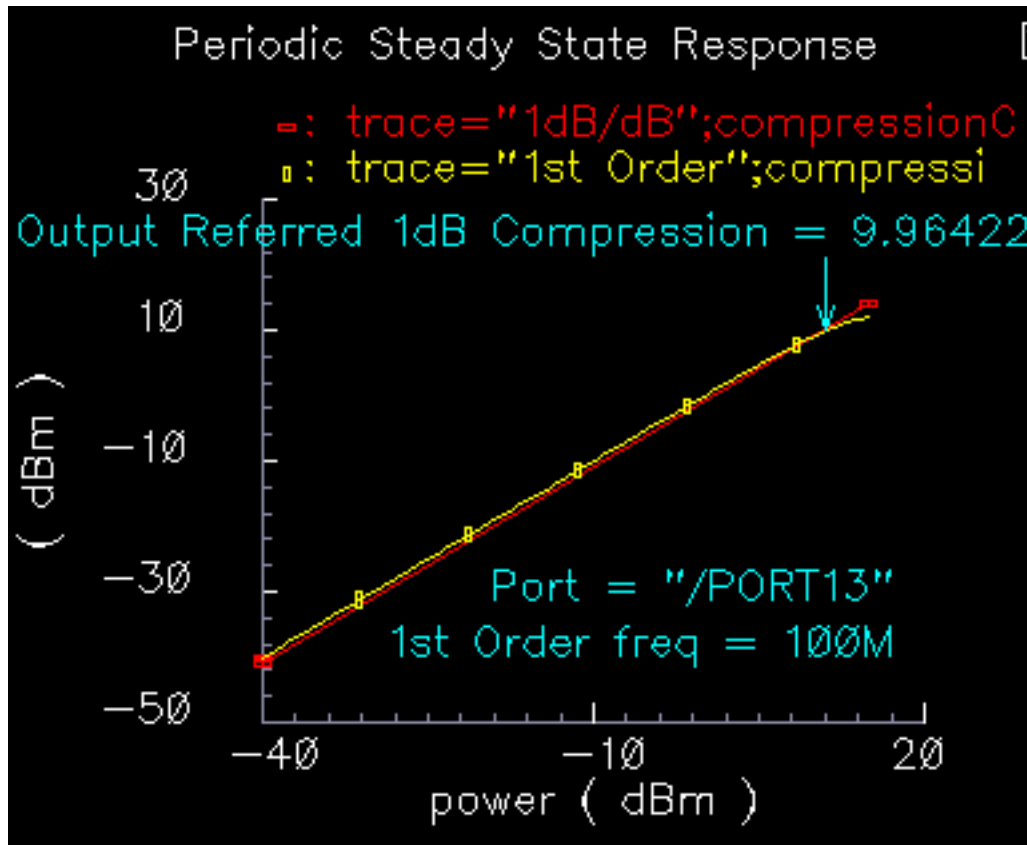


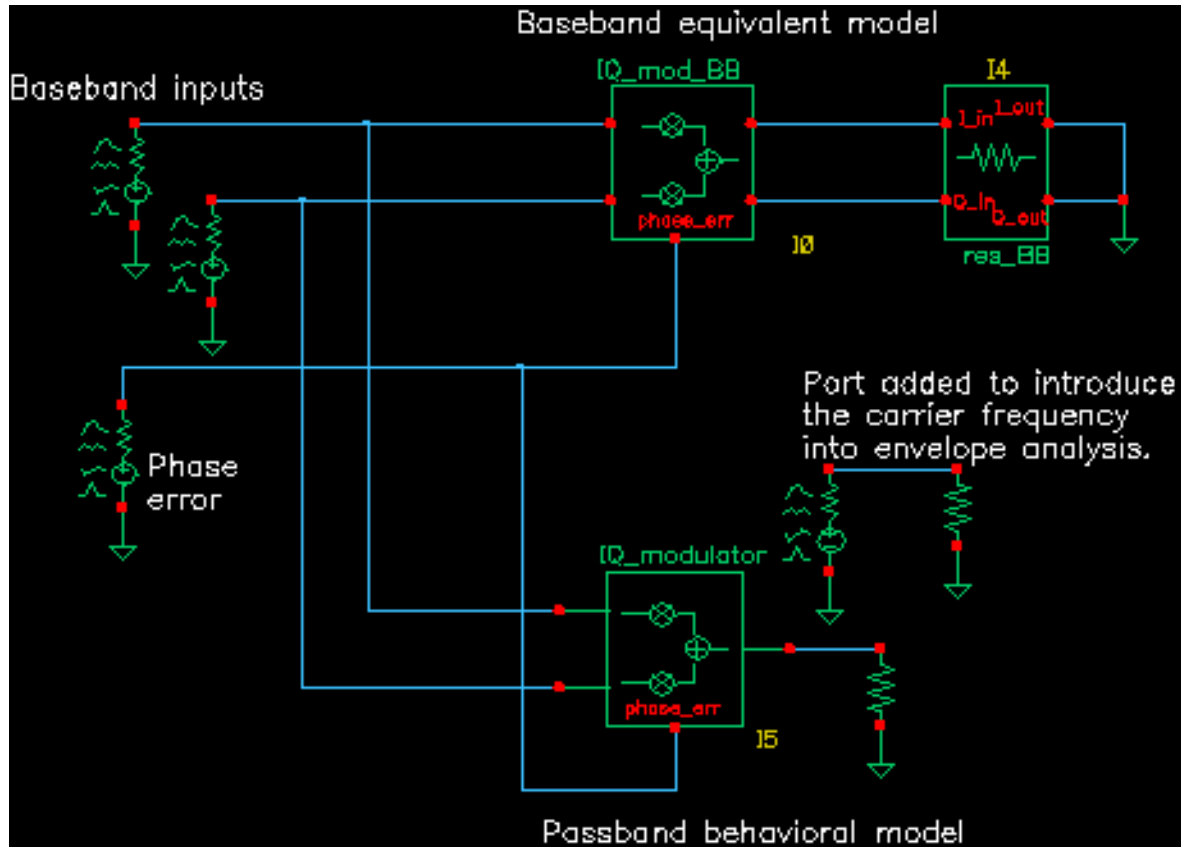
Figure 13-56 1db Compression Point Plot



Quadrature error and phase error:

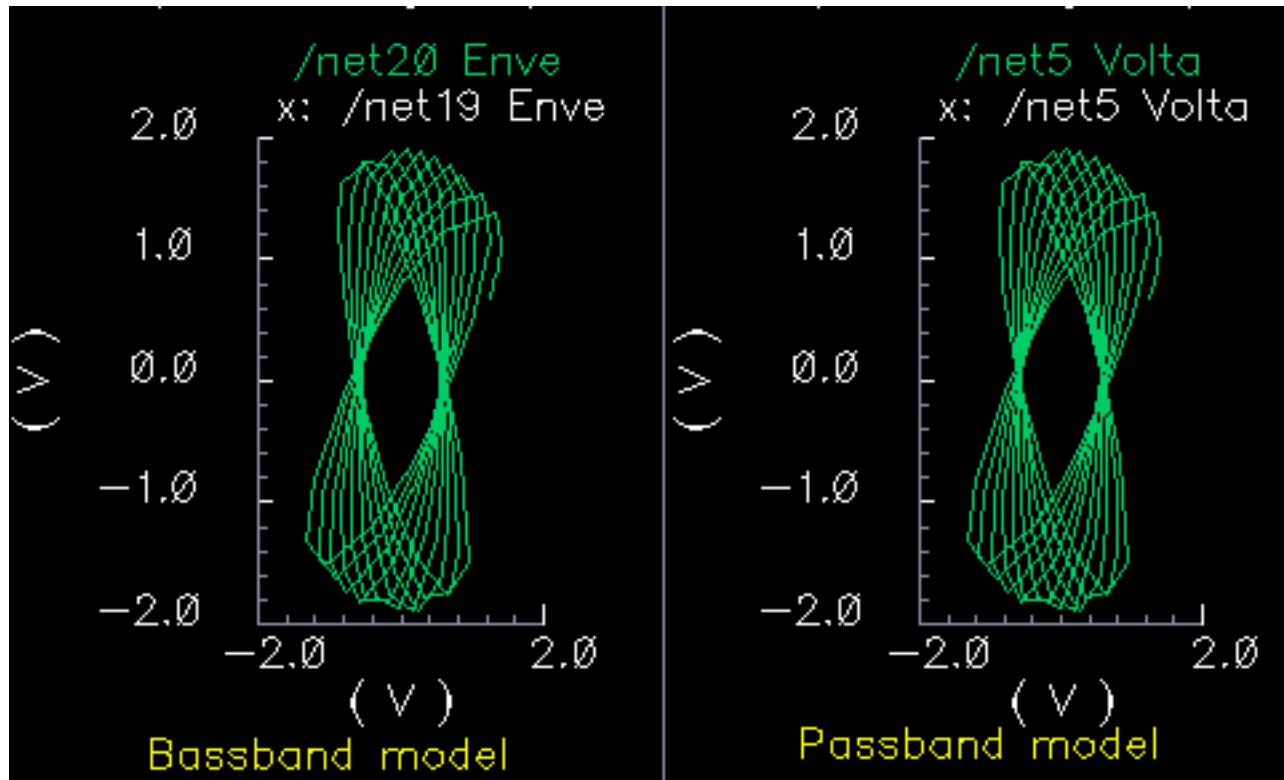
Quadrature error describes how far away from 90 degrees the two local oscillators are from each other. Ideally, they are exactly 90 degrees, or $\pi/2$ radians, apart in phase. In practice, parasitics and asymmetric delays can drive the phase shift away from $\pi/2$. Figure 13-57 on page 662 show a baseband test circuit and its passband equivalent. The schematic is listed in the rfLib testbenches category under the name "quad_and_phase_error_demo". Both circuits are driven from a common set of baseband sources. The test circuit serves two purposes, it shows the correspondence between baseband and passband models and it demonstrates how quadrature error and phase error affect the baseband trajectory. The baseband input signal is a complex tone, which makes a circular input baseband trajectory. If there were no quadrature error, the baseband representation of the modulator output would also be a circle. With quadrature error, the output trajectory is an ellipse. If the phase_err pins are driven by a ramp, the ellipse precesses. The ramp represents a small but fixed difference between carrier and local oscillator frequencies. To see these effects:

Figure 13-57 quad_and_phase_error_demo Circuit



1. Set up an Envelope analysis with "carrier" as the Clock Name. Simulate 10 us of action and save the first harmonic.
2. When the analysis completes, open the Envelope Following Results window and set the sweep to "time". Plot the two outputs of the IQ_mod_BB block.
3. Change the x-axis to be the I-output. You should see the left trajectory in [Figure 13-58](#) on page 663.
4. Add a subwindow for the passband equivalent result.
5. In the Envelope Results window, change the sweep to "harmonic time" and plot the real and imaginary parts of the first harmonic of the IQ_modulator output voltage.
6. Change the x-axis to be the real part of the first harmonic. You should now two pictures that match those in [Figure 13-58](#) on page 663.

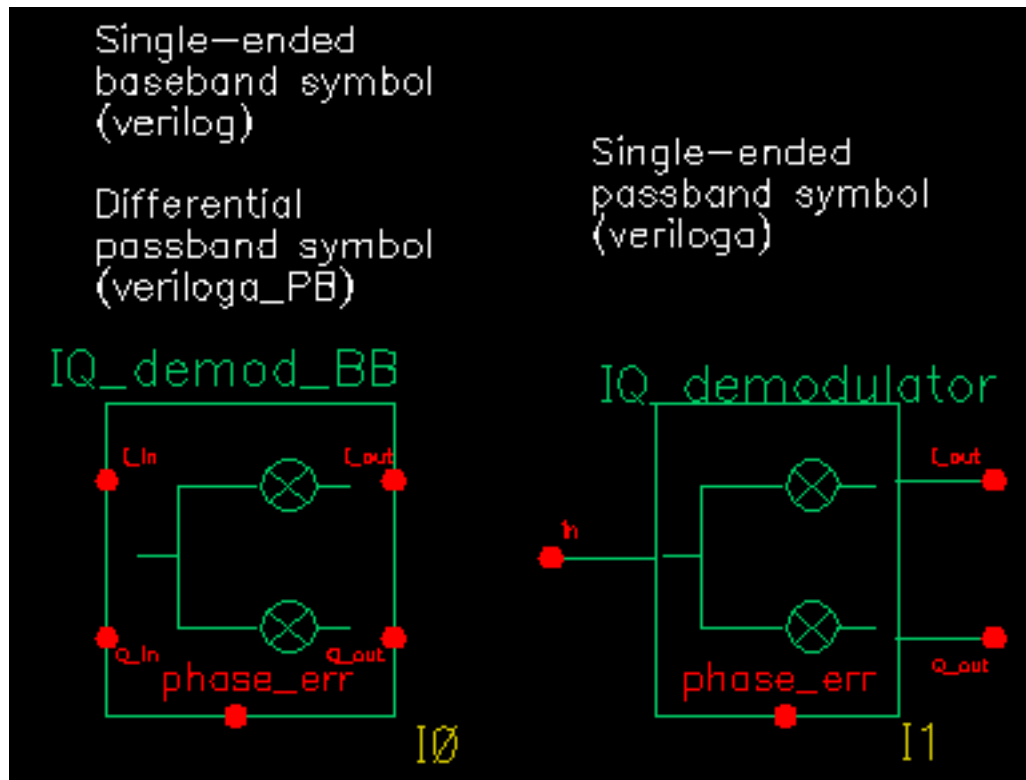
Figure 13-58 Output Trajectories



IQ Demodulator

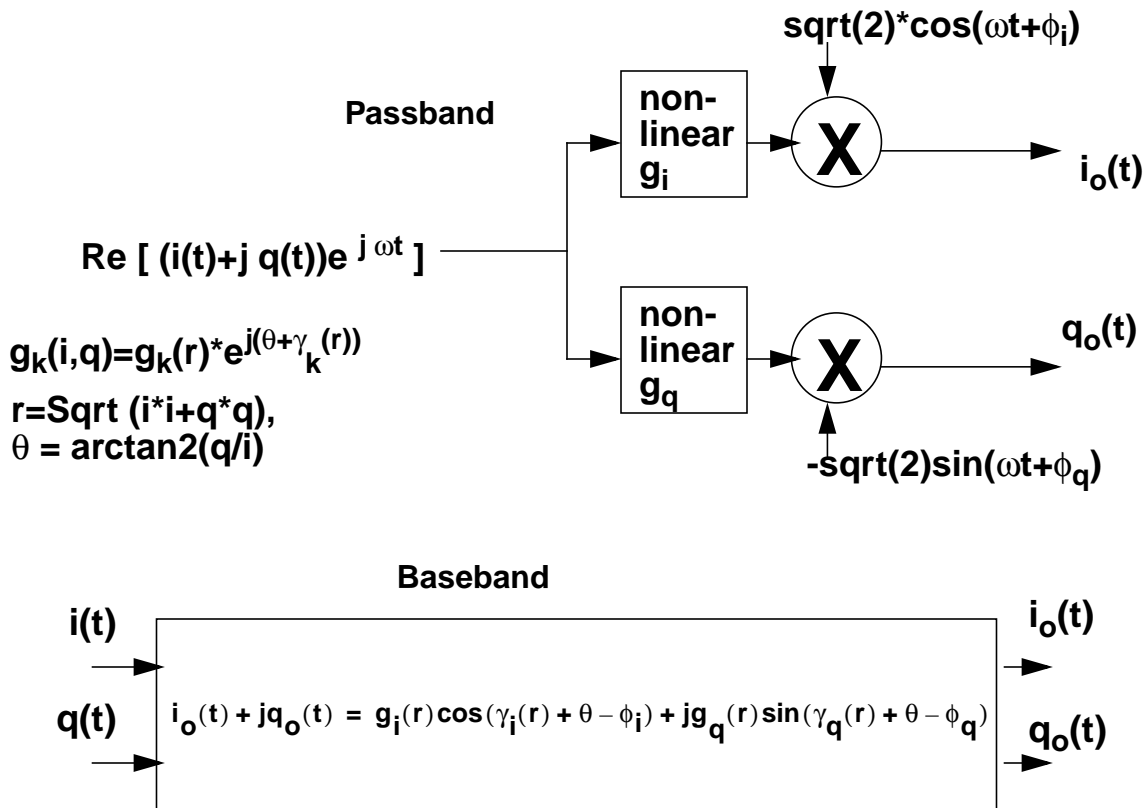
(baseband = IQ_demod_BB, passband = IQ_demodulator)

Figure 13-59 Baseband and Passband IQ Demodulator Models



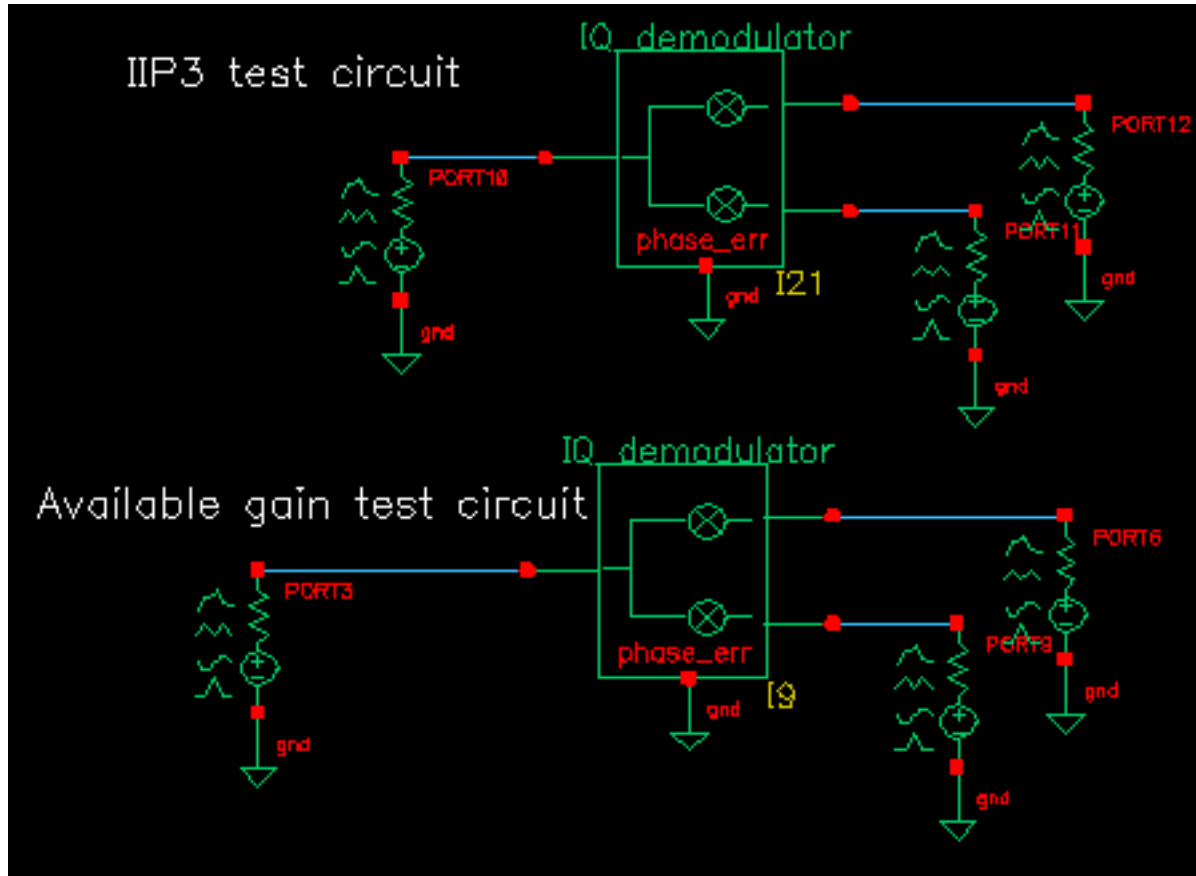
The IQ_demodulator converts RF (or IF) to baseband. [Figure 13-60](#) on page 665 shows exactly what the passband demodulator model does. The parameters are like those in the modulator blocks except saturation is specified by input referred IP3 instead of the 1 dB compression point. IP3 was chosen over the one dB compression point for specifying saturation because the demodulator usually lies in the receive path and receiver blocks are usually specified with IP3.

Figure 13-60 IQ Demodulator Calculations



The circuit called demod_ip3 in the testbenches category of the rLib shows how the gain and IP3 parameters are defined. [Figure 13-61](#) on page 666 shows the schematic. Both the input and the output resistances are matched.

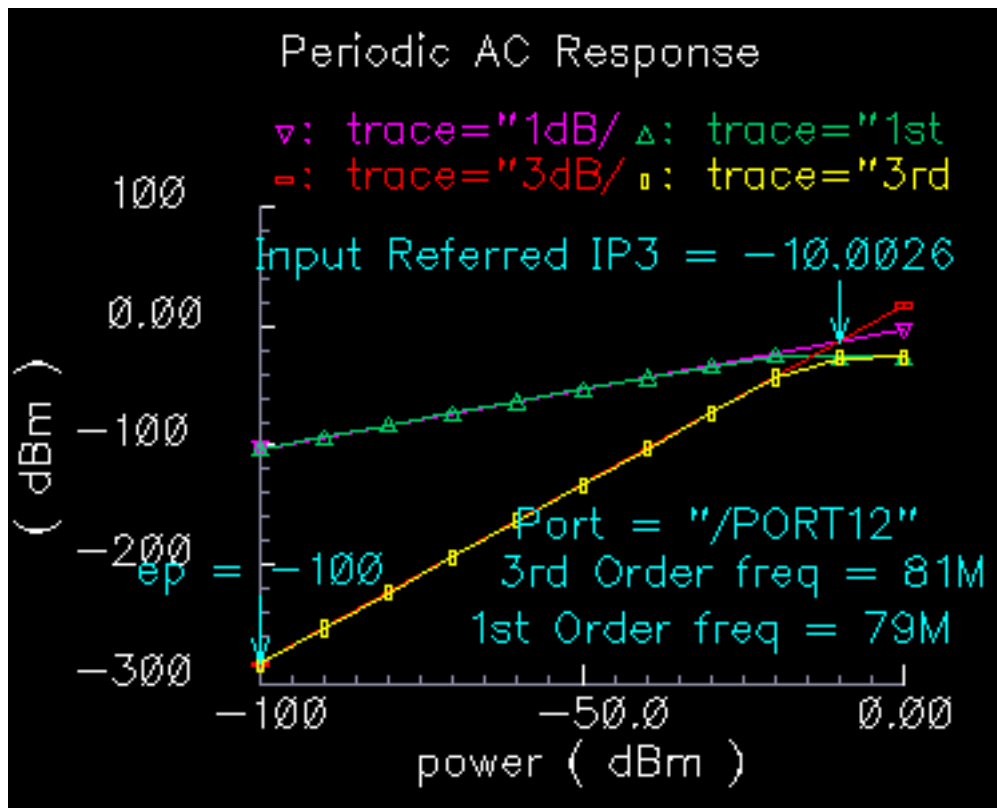
Figure 13-61 The demod_IP3 Schematic



1. Recall the circuit and set up a swept PSS analysis. Let the Beat Frequency be Auto Calculated. Keep 2 harmonics. Sweep the "power" parameter from -100 to 0 in 10 linear steps.
2. Set up a single point PAC analysis at 921MHz and keep the -25 and -21 sidebands.
3. After running the analysis, from the PAC output window plot the input-referred IP3 curves with 81MHz as the 3rd order sideband and 79MHz as the 1st order sideband. The procedure is similar to mixer IP3 example covered in the last section. Use "Variable Sweep" for the Circuit Input Power and -100 for the Extrapolation point. Make sure to plot Input Referred IP3. Click on the I-output port in the top circuit. You should see -10dBm as the IP3, just as specified. Figure 13-62 on page 667 shows the IP3 plot. Note that 1st order line indicates the gain is 3dB below the specified gain of 0dB. That is because not all of the power lies at 1000MHz-921MHz = 79MHz; Some of the power lies at 1000MHz + 921MHz = 1921MHz. Use the bottom test circuit to measure available power gain. The bottom circuit drives the demodulator at the same frequency as the demodulator's

internal local oscillator, which runs at 1GHz. Now the output power is not split, it lies in the zero harmonic of the I-output.

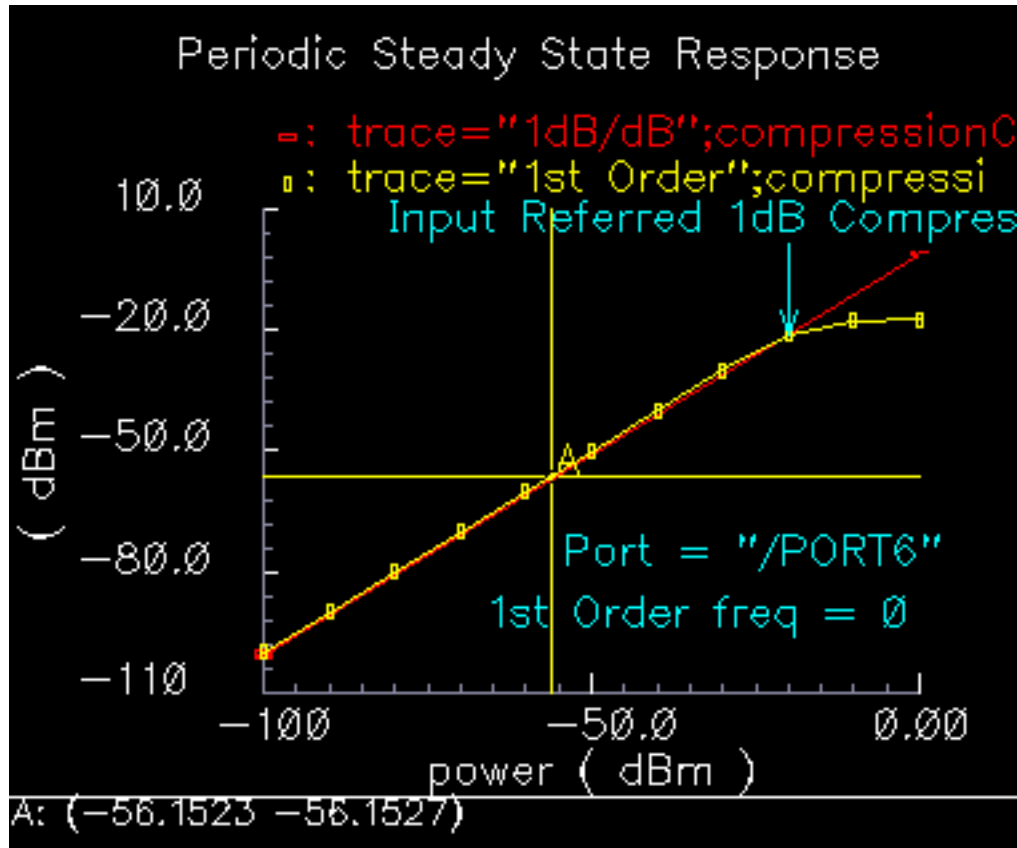
Figure 13-62 Demodulator IP3



4. Plot the 1dB compression point at the port loading the I-output of the bottom circuit. Use the zeroth harmonic. The ratio of output to input power should be unity in the linear region. [Figure 13-63](#) on page 668 shows the compression point plot. The measured 1dB compression point is of no use in this test. We want the gain. At low power levels where the gain is constant, the gain is as specified. ¹

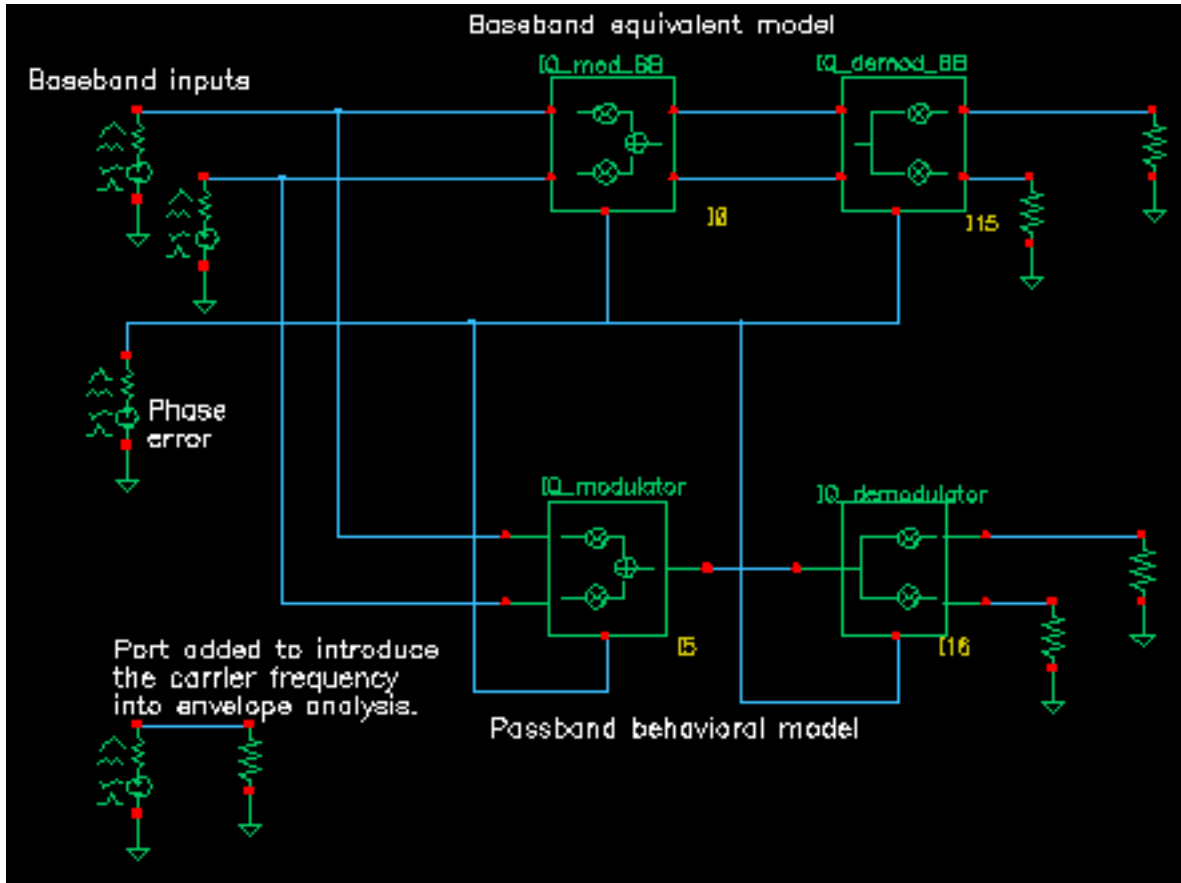
1. Remember, in this test circuit the load resistance and output resistance are equal so that the output power is maximal. Also, the input resistance equals the source resistance so that the horizontal axis truly equals input power.

Figure 13-63 Demodulator available power gain



Phase errors behave like their counterparts in the modulator models except for a change of sign. Quadrature error behaves exactly as it does in the modulator models. Figure 13-64 on page 669 shows a test circuit for illustrating the relationships between phase error and quadrature error in the modulators and demodulators. The test circuit is called "mod_demod_test" and is listed in the testbenches category. The test circuit also shows that the passband and baseband models give comparable results, as they should, as long as the passband carrier is not severely clipped. The baseband input trajectory is a complex 1MHz tone, which produces a circular input trajectory. The demodulator outputs are not matched and are not symmetric with respect to I and Q paths. The modulators and demodulators are not perfectly linear and the non-linearities are asymmetric with respect to I and Q. The modulators and demodulators are driven by the same phase error and the quadrature error parameters are a common variable set to 0.785 radians.

Figure 13-64 mod_demod_test Circuit



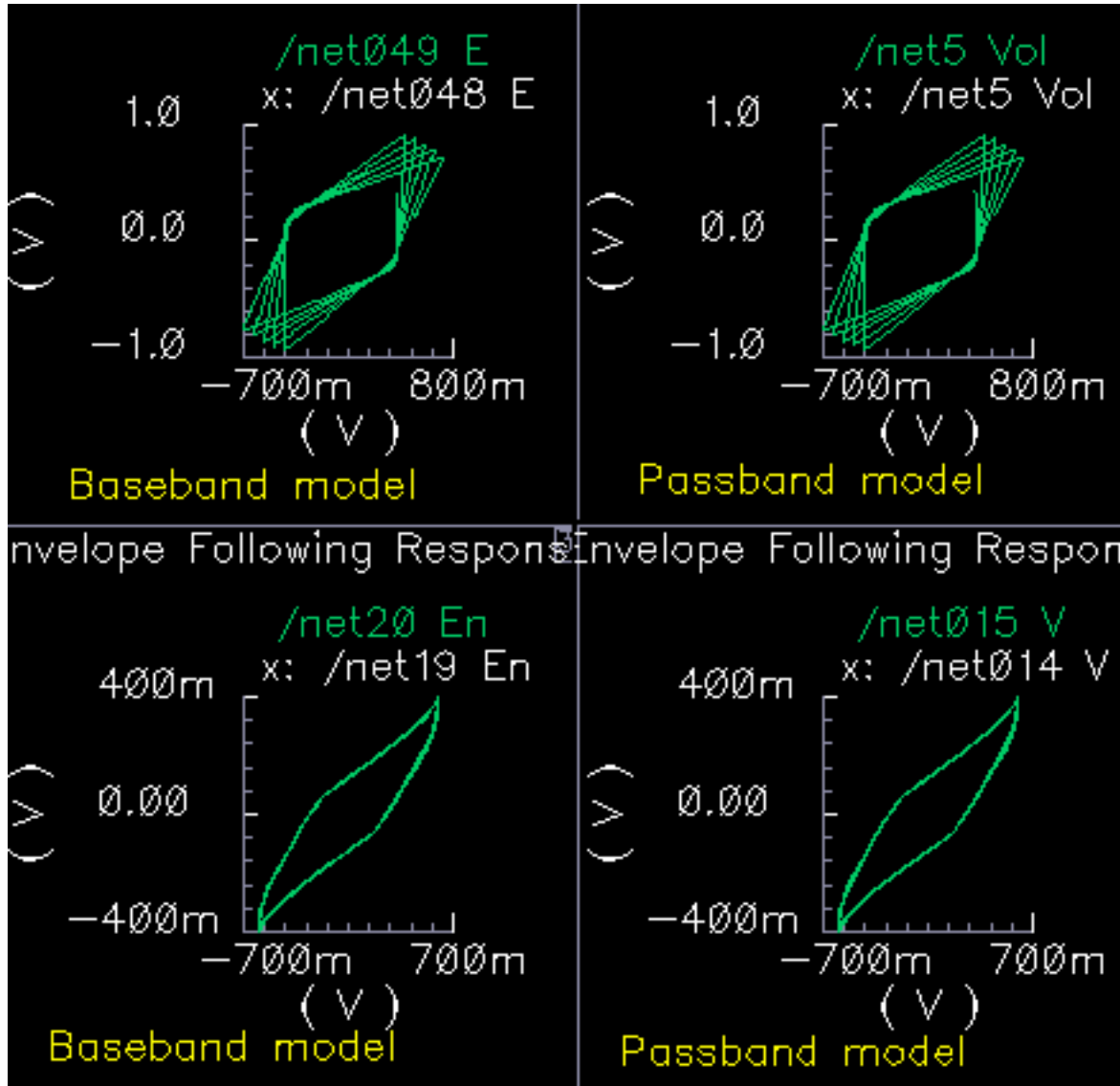
To use the circuit:

1. Recall the circuit and set up a 5us Envelope Following analysis with "carrier" as the Clock Name.
2. After the analysis completes, plot the IQ_mod_BB outputs and make the I_out signal the x-axis.
3. Open a subwindow and in it, plot the harmonic time waveforms of the IQ_modulator output. Use the first harmonic and plot the real and imaginary waveforms. Make the real waveform the x-axis.
4. Open a third subwindow and stretch the Waveform Display window so that the third subwindow appears below the first window.
5. Plot the time waveforms at the IQ_demod_BB outputs and make the I_out waveform the x-axis.

6. Open a fourth subwindow and plot the harmonic time results at the IQ_demodulator outputs but this time use the zeroth harmonic and only plot the real parts. Make the I_out waveform the x-axis. [Figure 13-65](#) on page 671 shows what you should now see.

The leftmost pictures are from the baseband models and the rightmost are from the passband models. Passband and baseband models agree quite well. The top pictures are the voltages at nodes that lie between the modulator and demodulator. Quadrature error squashes the baseband trajectory at that node. The trajectory precesses because phase error ramps up linearly with time just like in the last test. The non-linearities produce the sharp corners. The bottom trajectories do not precess because the same phase error rotates the demodulator output in the reverse direction; driven by the same phase error ramp, the demodulator undoes the precession introduced in the modulator. The demodulator outputs are nearly in phase because the quadrature errors of $\pi/4$ in the modulators and demodulators add to give a total quadrature error of $\pi/2$, which in this case puts the baseband I and Q outputs nearly in phase with each other.

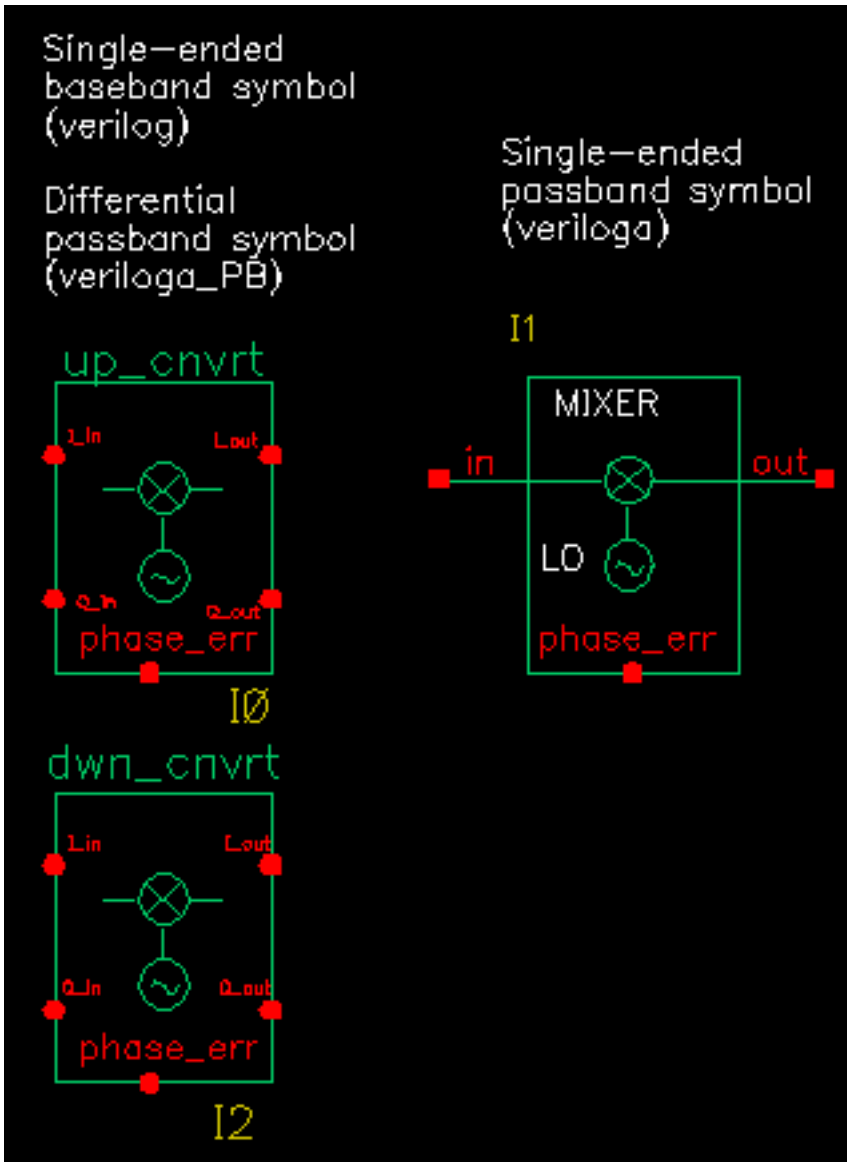
Figure 13-65 mod_demod results



RF-to-IF and IF-to-RF Mixers

(passband = MIXER_PB, baseband = dwn_cnvr and up_cnvr)

Figure 13-66 Baseband and Passband Mixer Models



MIXER_PB is a passband model that converts RF to IF and IF to RF. dwn_cnvr model is a baseband equivalent model of a mixer used to convert from RF to IF. up_cnvr model is a baseband equivalent model of a mixer used to convert from IF to RF. There are some minor

differences in the baseband models that depend on whether conversion is up or down. [Figure 13-67](#) on page 673 and [Figure 13-68](#) on page 674 show what the models do.

Figure 13-67 Calculations for up_cnrt Mixer

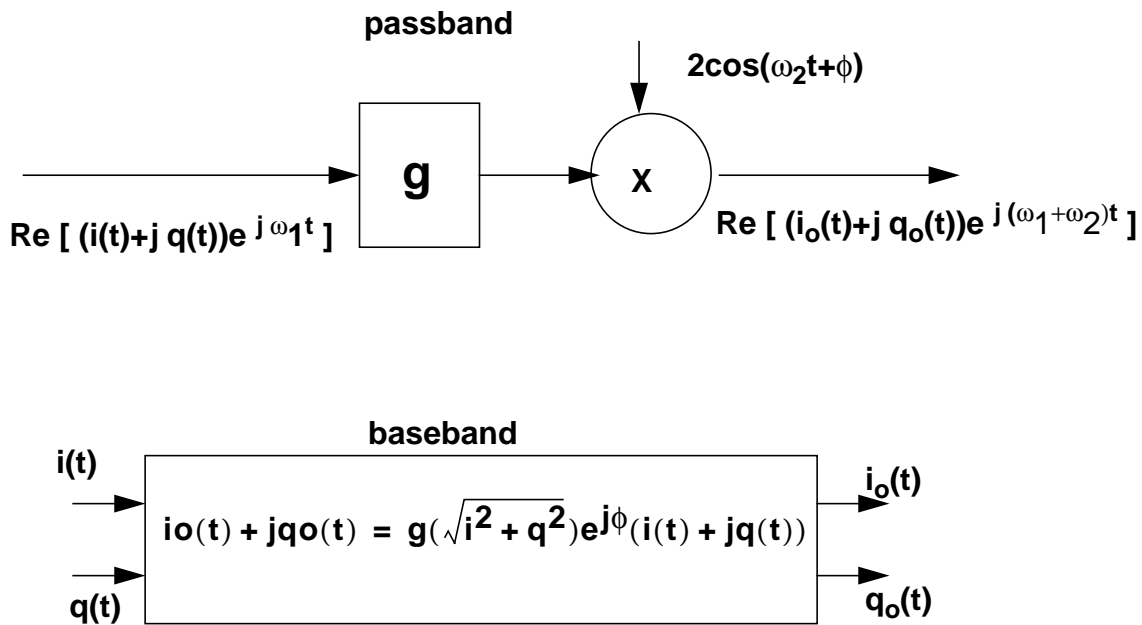
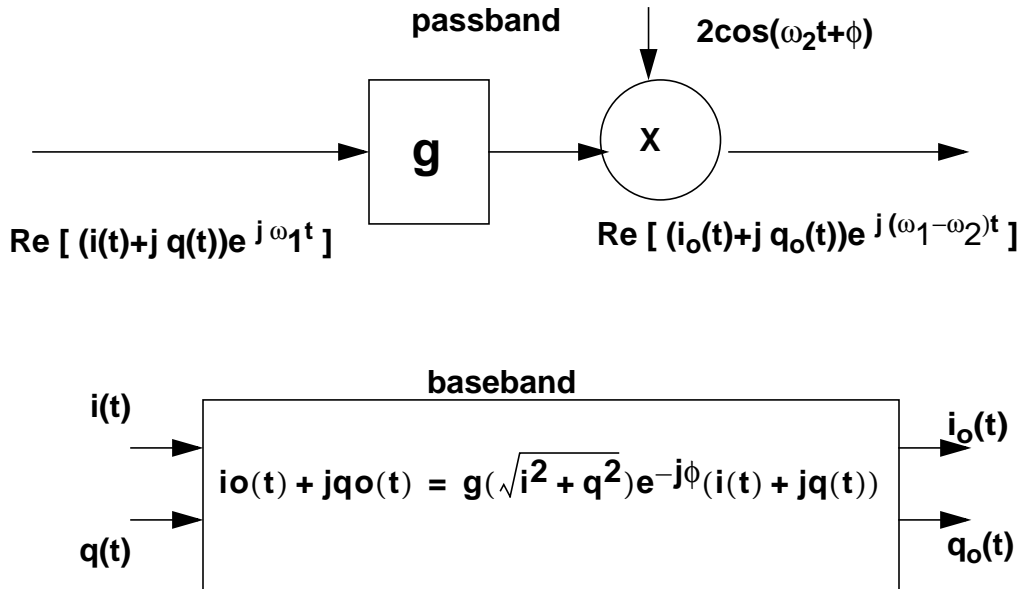


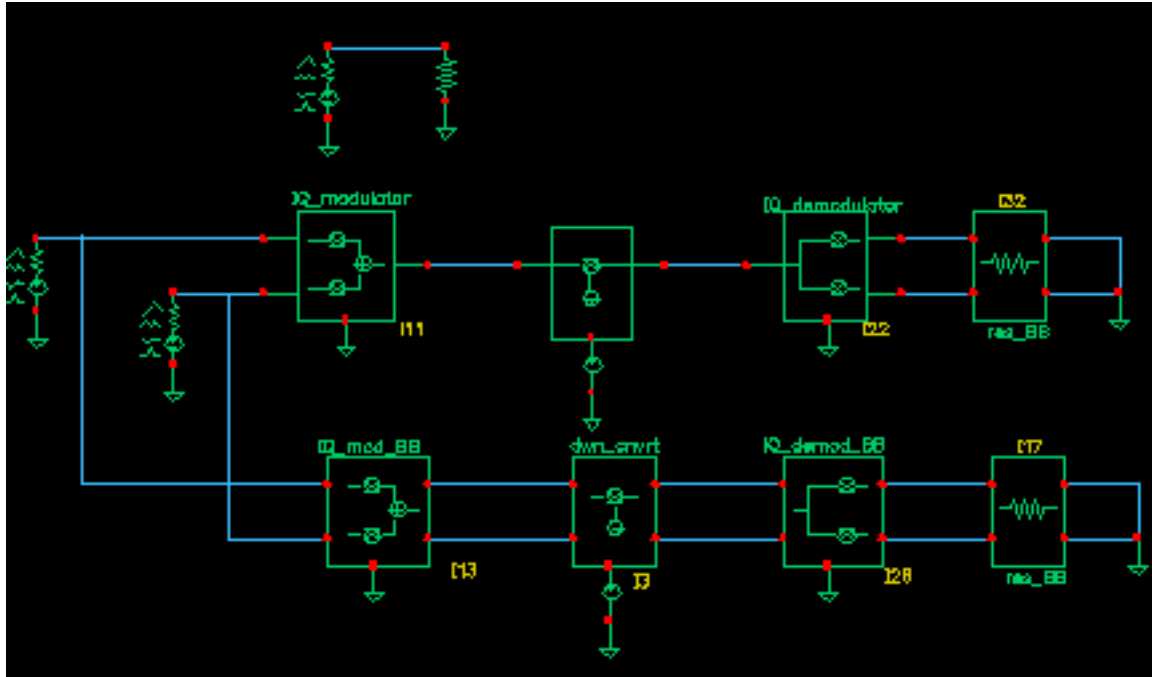
Figure 13-68 Calculations for dwn_cnrt Mixer



The noise figure and IP3 parameters are defined in [“Some Parameters Common to Several Models”](#) on page 632. Unlike the IQ_demodulator, the IP3 test circuit can be used to define the available power gain because the gain is defined from the input frequency to just one sideband.

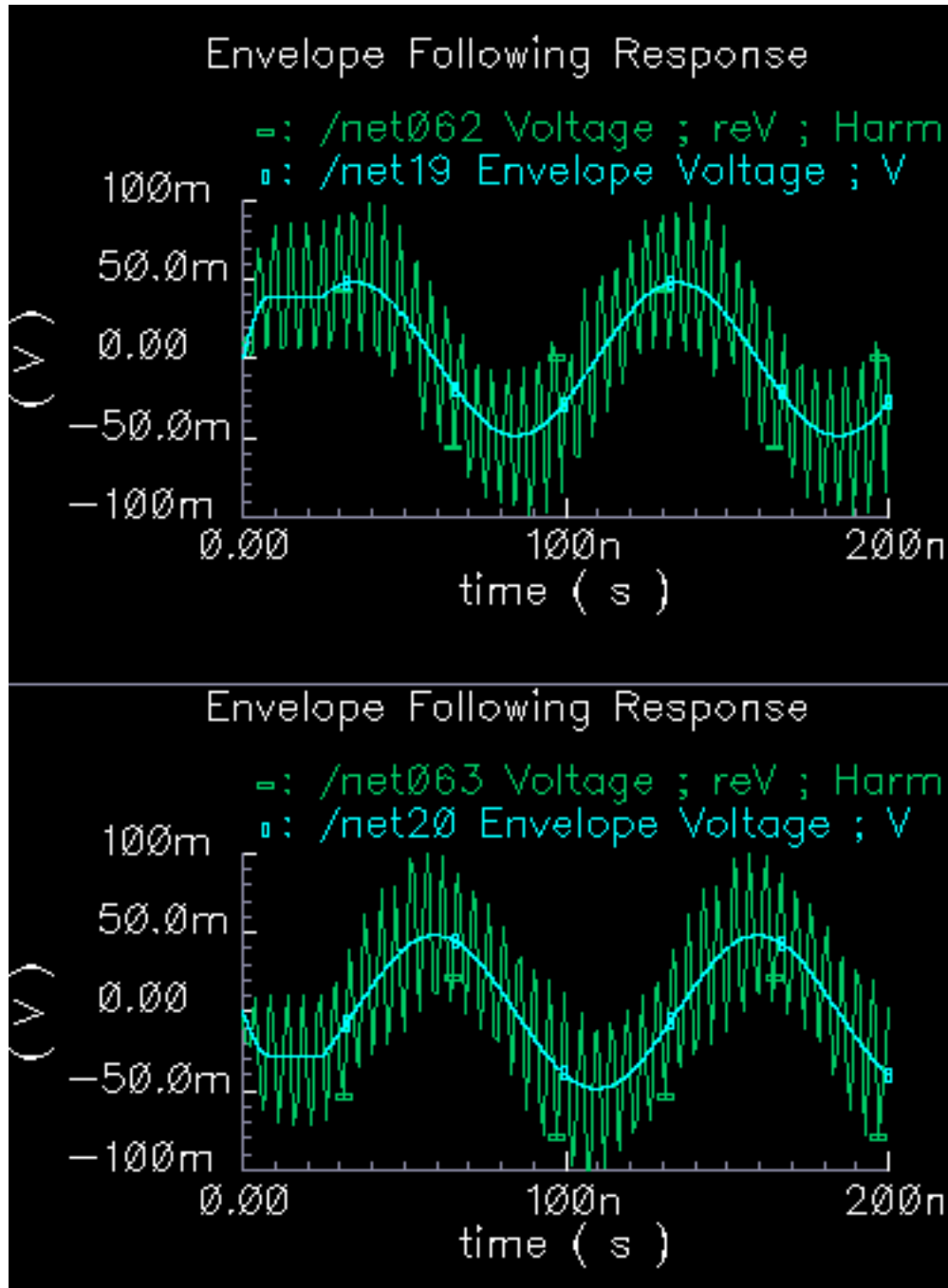
Typically the mixer would be used to create an IF stage. In that case, it is difficult to obtain a simple (i.e. filterless) envelope following analysis that overlays waveforms to show how well baseband and passband models agree. The test circuit shown in [Figure 13-69](#) on page 675, which is listed as "dwn_cnvt_test" in the testbenches category of the rflib, shows the relationship between baseband and passband models. The top branch of the circuit consists of passband models. The bottom branch consists of baseband models. To see the relationship:

Figure 13-69 dwn_cnrt_test Circuit



1. Recall the circuit and set up a 200ns envelope following analysis with "fclk" as the Clock Name. Keep 1 harmonics1.
2. After the analysis completes, plot the "time" waveform at the I_out pin of the IQ_demod_BB model. Append to the plot, the harmonic-time, real part of the zero harmonic of the I_out pin on the IQ_demodulator model.
3. Open a subwindow and do the same for the Q outputs. You should now see a picture like the one in [Figure 13-70](#) on page 676.

Figure 13-70 Output from an Envelope Following Analysis

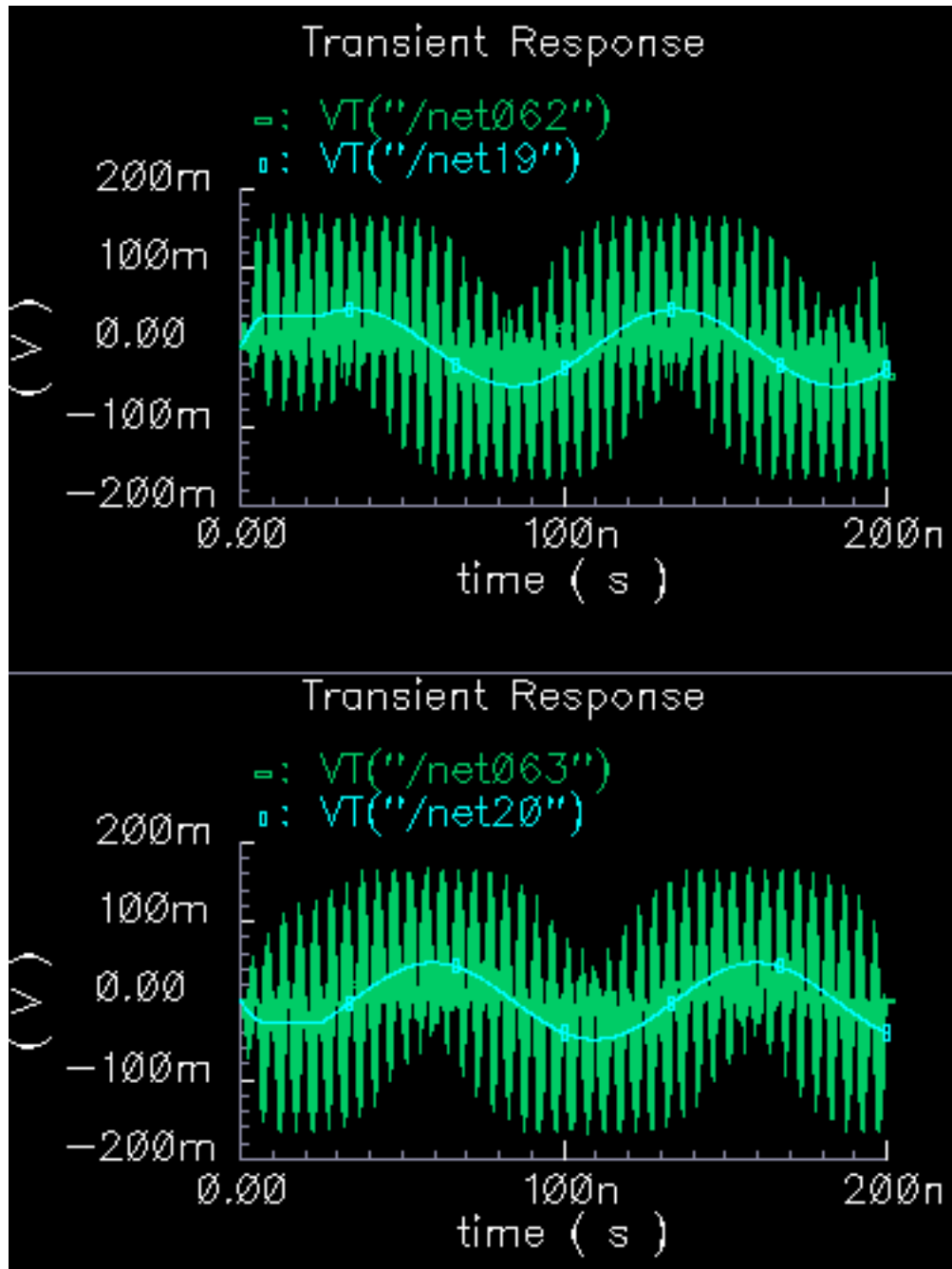


Let's trace the input signal through the passband branch to understand these results. A complex baseband 10MHz tone drives both branches. The modulator's local oscillator is 1GHz so that the IQ_modulator output is at 1.01GHz. There is no 990MHz sideband because

the input baseband trajectory is a circle ($= \sin + j\cos$), which represents a complex tone. The mixer local oscillator is 900MHz, which when mixed with 1.01GHz, produces 110MHz and 1.91GHz. The IQ_demodulator local oscillator is 100MHz, which produces 10MHz, 210MHz, 2.01GHz, and 1.81GHz. The 10MHz and 210MHz terms dominate the zero harmonic at the demodulator outputs. The higher frequencies average out to nearly zero. The baseband output is the 10MHz term and that is what the baseband branch generates, as shown in [Figure 13-70](#) on page 676. A Transient analysis actually runs about 13 times faster than envelope on this circuit. [Figure 13-71](#) on page 678 compares the same outputs using a Transient analysis. The Transient analysis shows that the zero harmonic of the envelope analysis averaged out all frequencies above the envelope clock frequency (1GHz).

There is a test circuit for the up_cnvrt model similar to the test circuit containing the dwn_cnvrt model. The up_cnvrt model is called up_cnvrt_test and is also in the testbenches category of rfLib. The steps parallel those for the dwn_cnvrt model.

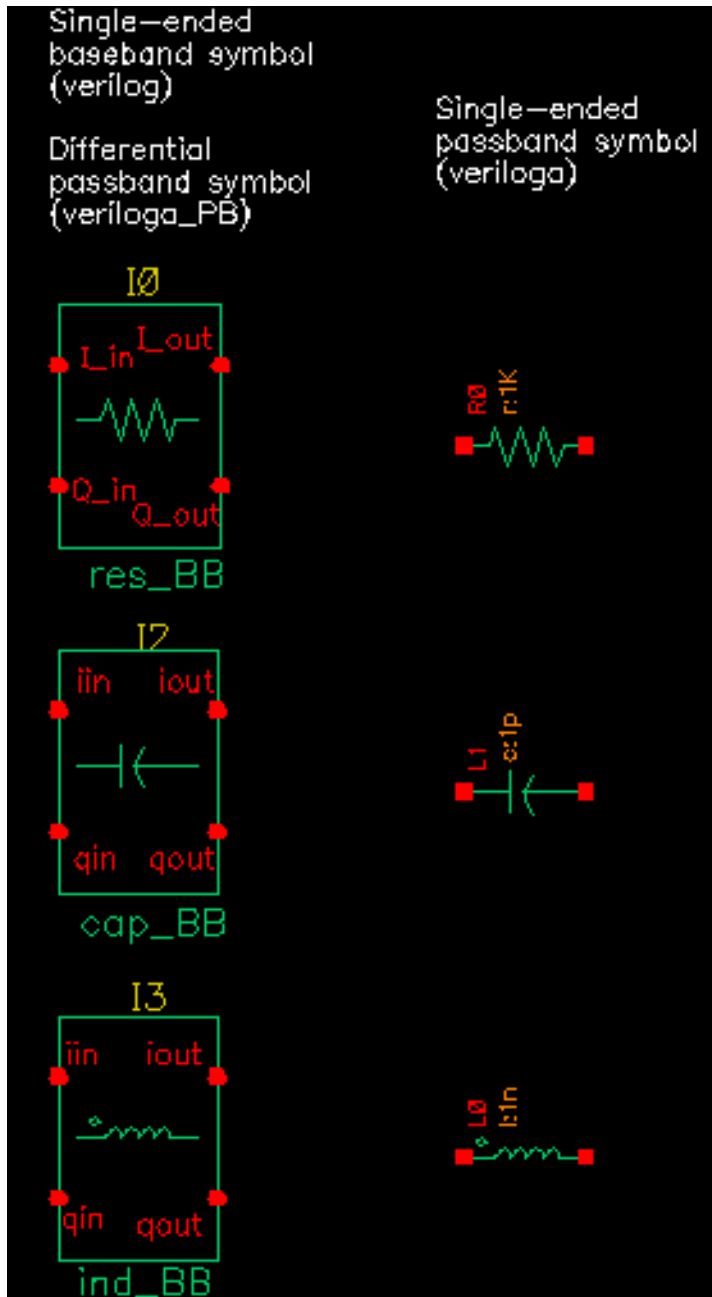
Figure 13-71 Results from a Transient Analysis



Passive Devices

(res_BB, cap_BB, ind_BB)

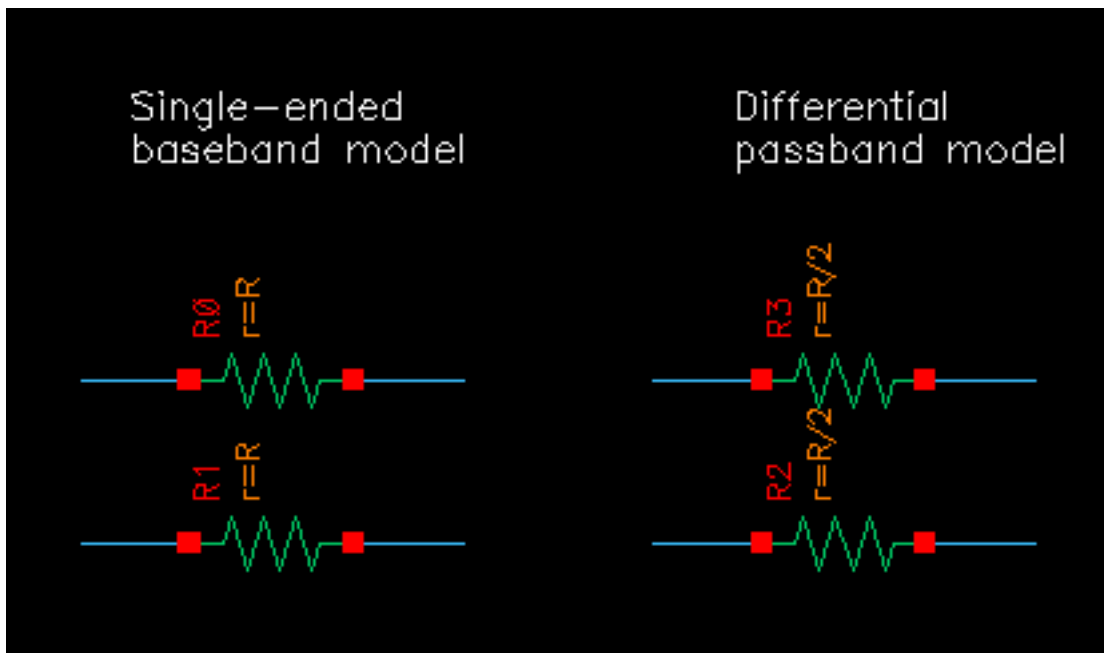
Figure 13-72 Circuit



Resistors

Besides the resistance, the baseband resistor model has a parameter for turning its thermal noise on or off. The baseband resistor is intended for use at a passband node because its noise is doubled. (This was discussed in the section entitled “Relationship between baseband and passband noise”). [Figure 13-73](#) on page 680 shows the symbol, baseband, and passband models. The total noise in the differential passband resistor model equals the noise in one resistor of R Ohms.

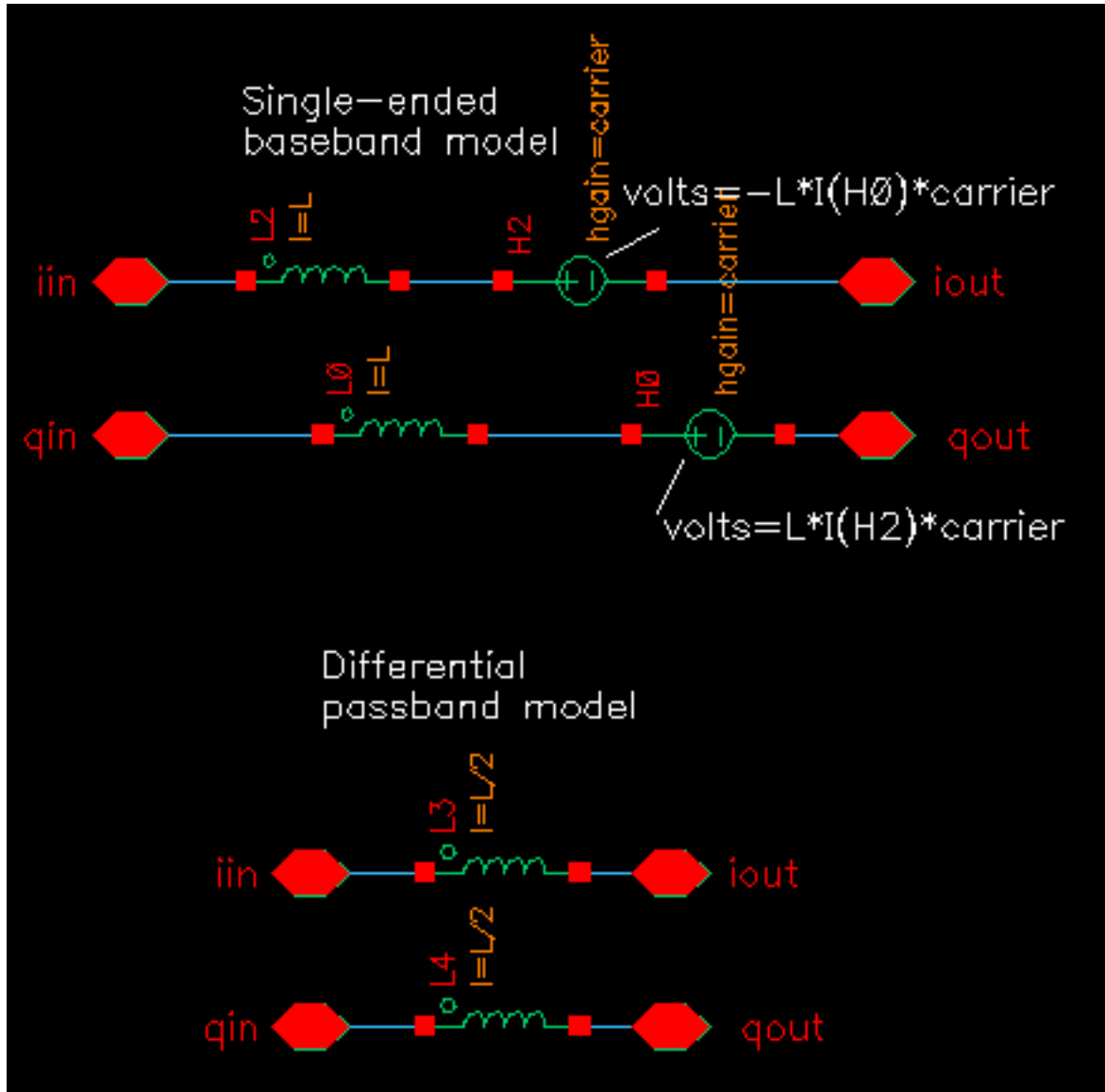
Figure 13-73 Resistor model



Inductors

The library overview described the basic idea behind baseband modeling of inductors. The baseband inductor model requires one additional parameter besides the inductance, the carrier frequency. [Figure 13-74](#) on page 681 shows equivalent schematics of the baseband and differential passband inductor models. The inductor models are noiseless.

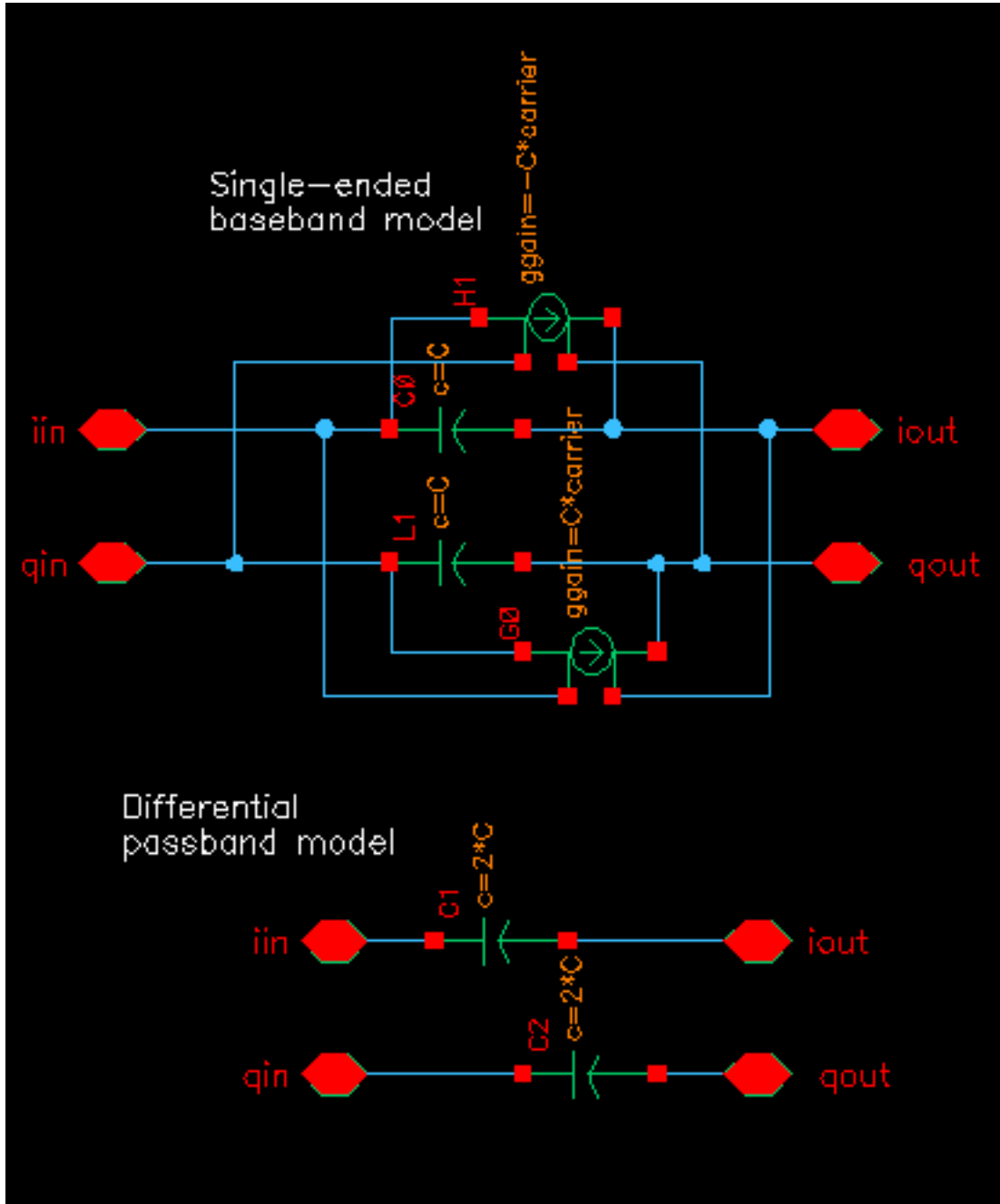
Figure 13-74 Inductor model



Capacitors

The capacitor is the mathematical dual of the inductor. [Figure 13-75](#) on page 682 shows the baseband and differential passband capacitor models.

Figure 13-75 Capacitor model



RLC test circuits

The two circuits discussed below demonstrate how passband and baseband reactive elements are related. The circuit in [Figure 13-76](#) on page 683 shows a simple passband RLC circuit driven by a modulated carrier. The circuit in [Figure 13-77](#) on page 684 shows the associated baseband equivalent circuit model. The circuits are "PB_ind_cap_test" and "BB_ind_cap_test". Both circuits reside in the rfLib under the testbenches category.

Figure 13-76 Simple Passband RLC Circuit

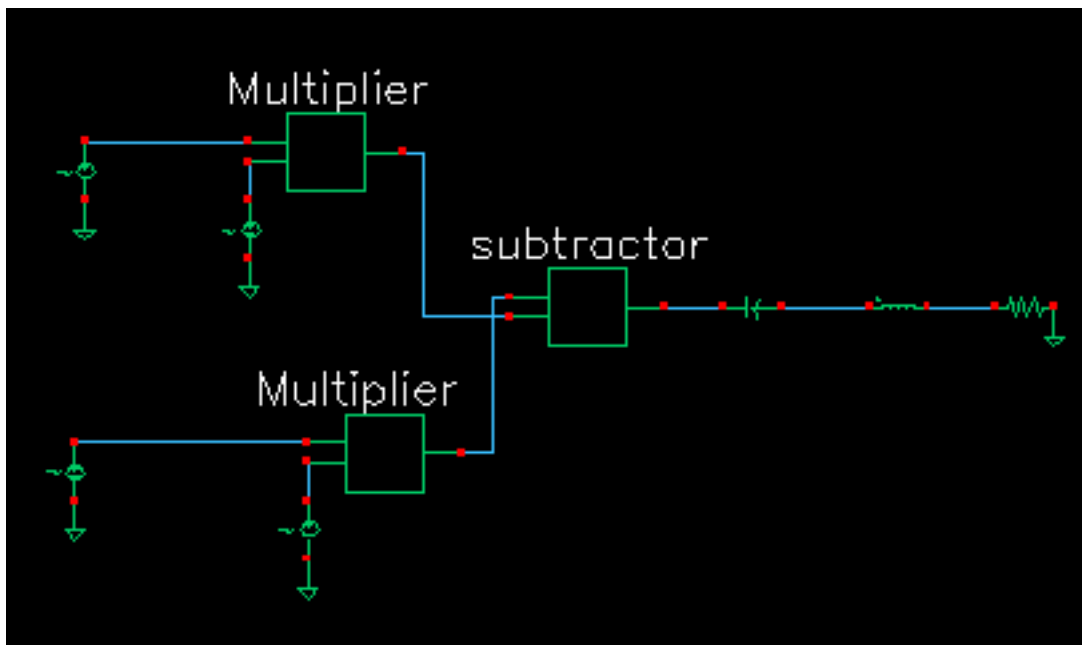
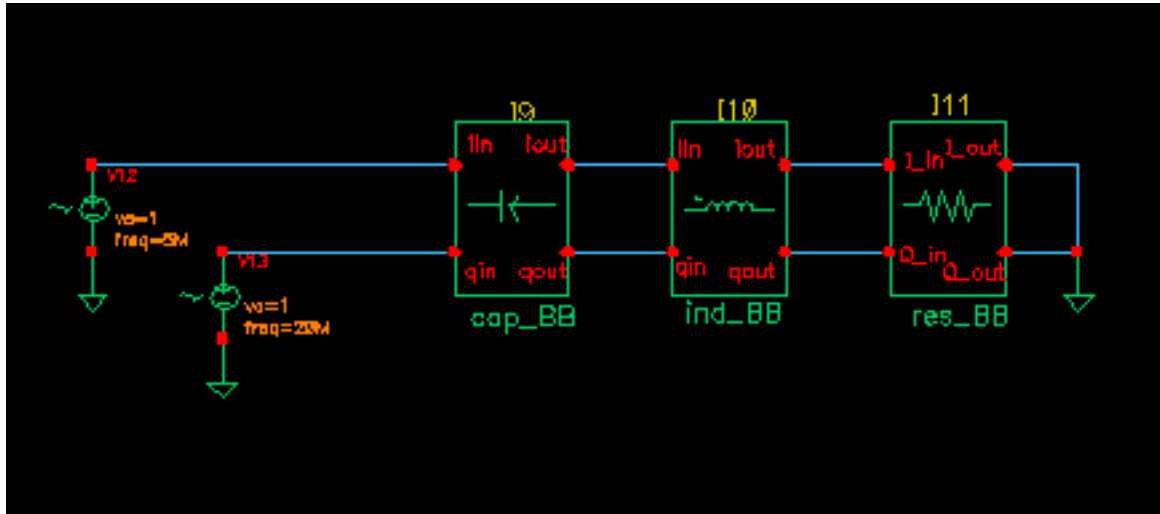


Figure 13-77 Baseband Equivalent To Figure 1-63



The following steps explain how to simulate each circuit and overlay the results.

1. Recall the PB_ind_cap_test circuit and bring up an Analog Circuit Design Environment window. Set up a 200ns Envelope analysis. Select "carrier" as the Clock Name. Set the Output Harmonics to 1.
2. Run the analysis and plot the real and imaginary parts of the harmonic-time voltage across the resistor. Use "1" for the harmonic number.
3. Recall the BB_ind_cap_test circuit and run a 200ns Transient analysis. Note the faster run time. That is the whole point to suppressing the carrier but it is only useful if the results match. Plot the "I_in" and "Q_in" voltages of the resistor model.
4. To overlay the results, bring up a waveform calculator.
5. Click the "wave" button on the calculator then click on one of the Envelope Following waveforms. If the waveform turns yellow you may have to hit the escape button a few times and click "clear" and "clst" a couple of times in the calculator then try again.
6. Make active the waveform display tool with the Transient results then click "Plot" in the calculator.
7. Repeat the last two steps for the other Envelope waveform. You should see the waveforms in [Figure 13-78](#) on page 685. The two models agree very well. The resonant frequency of the series RLC branch is just over 500MHz. Only by riding on a carrier can the 5MHz and 20Mhz baseband signals propagate to the resistor at their original voltage levels. The baseband model accurately predicts the effects of the RLC circuit on the

baseband signal. There are two effects, one due to phase shift at the carrier frequency and one due to filtering of the baseband signal itself.

8. In the waveform display tool that overlays the results, change the x-axis to be one of the I-signals. You should get the picture shown in [Figure 13-79](#) on page 686. The tilt in the resulting Lissajous plot indicates phase shift at the carrier frequency but not at the baseband frequencies. The aspect ratio of the Lissajous figure indicates the 20MHz component is attenuated more than the 5MHz component. The baseband model captures both effects well.

Figure 13-78 Waveforms

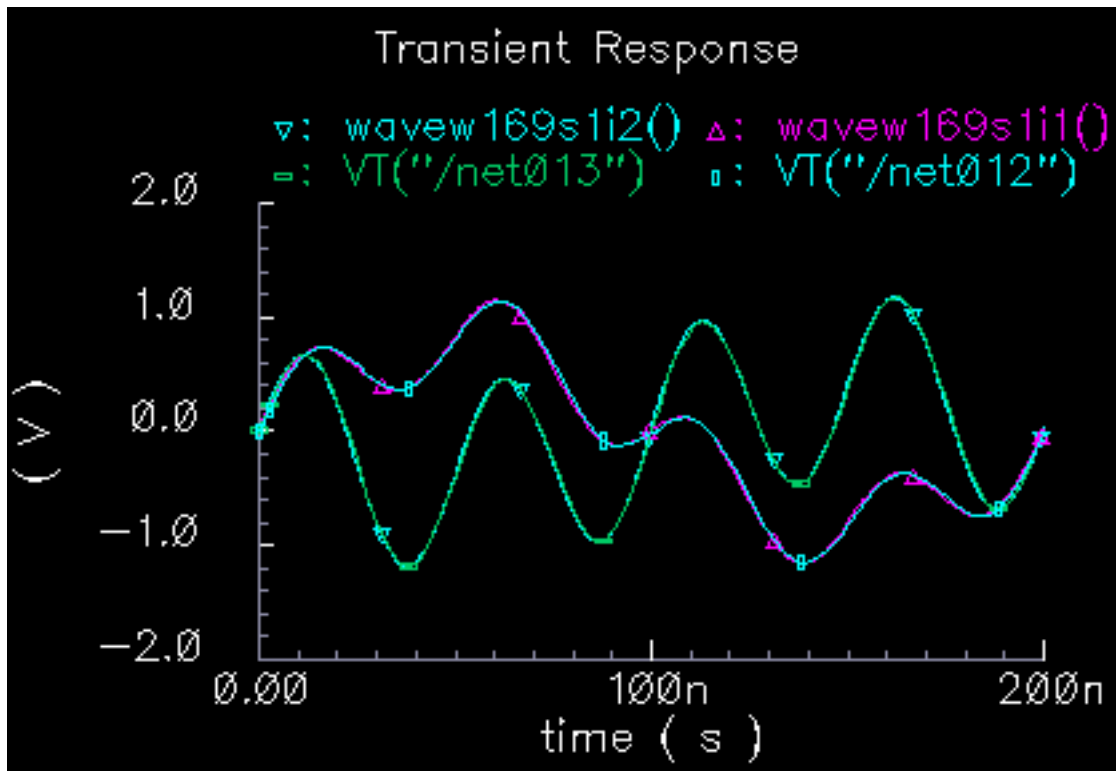
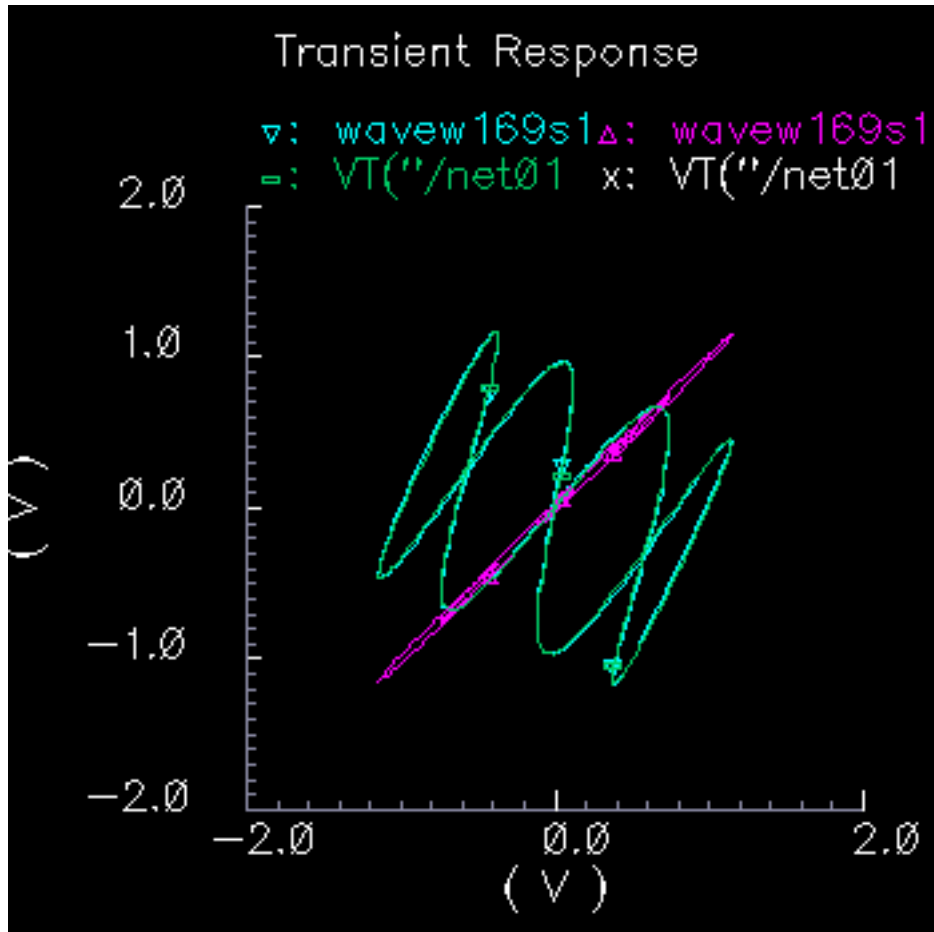


Figure 13-79 Lissajous plot



Linear Time Invariant Filters

Figure 13-80 Butterworth Filters

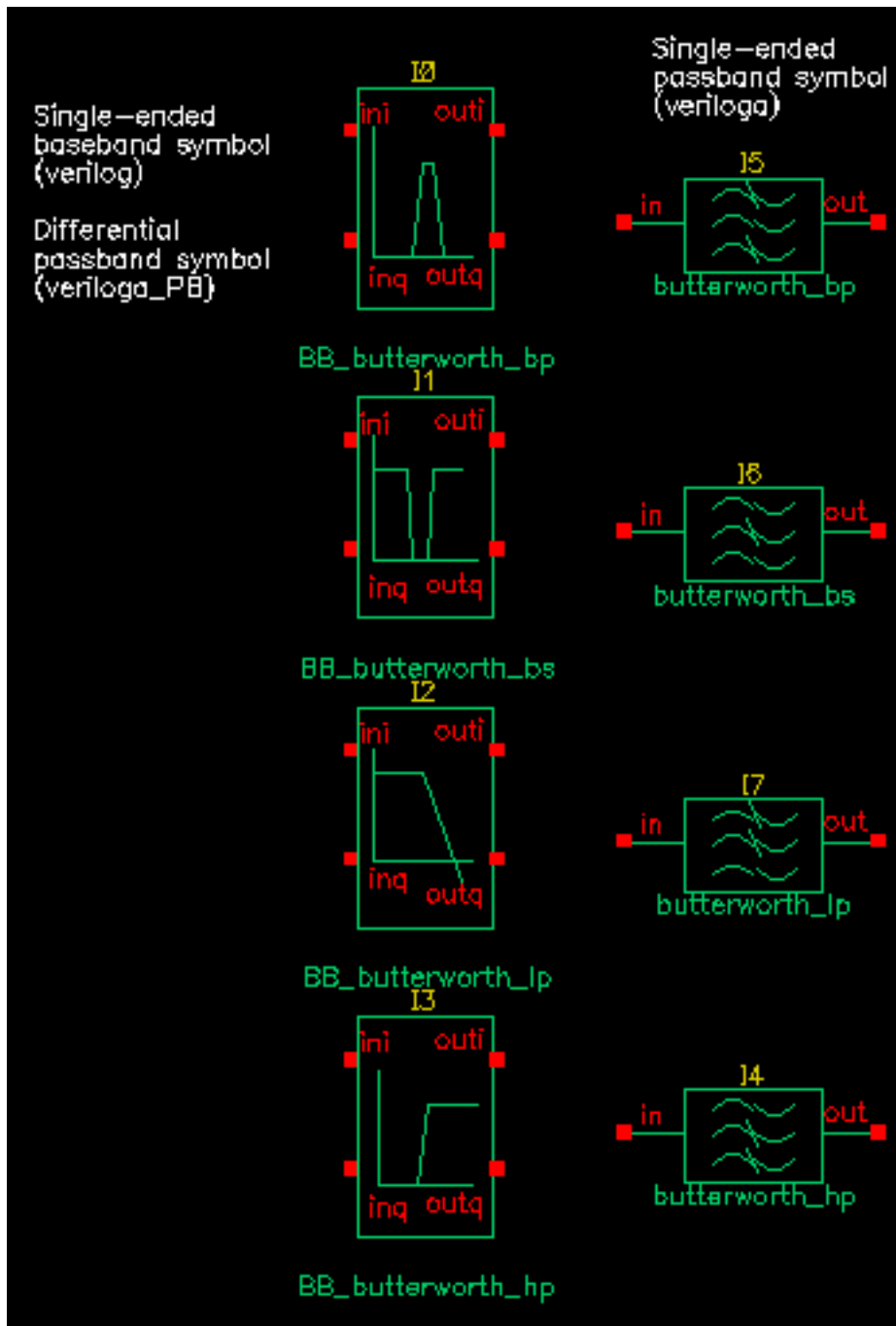
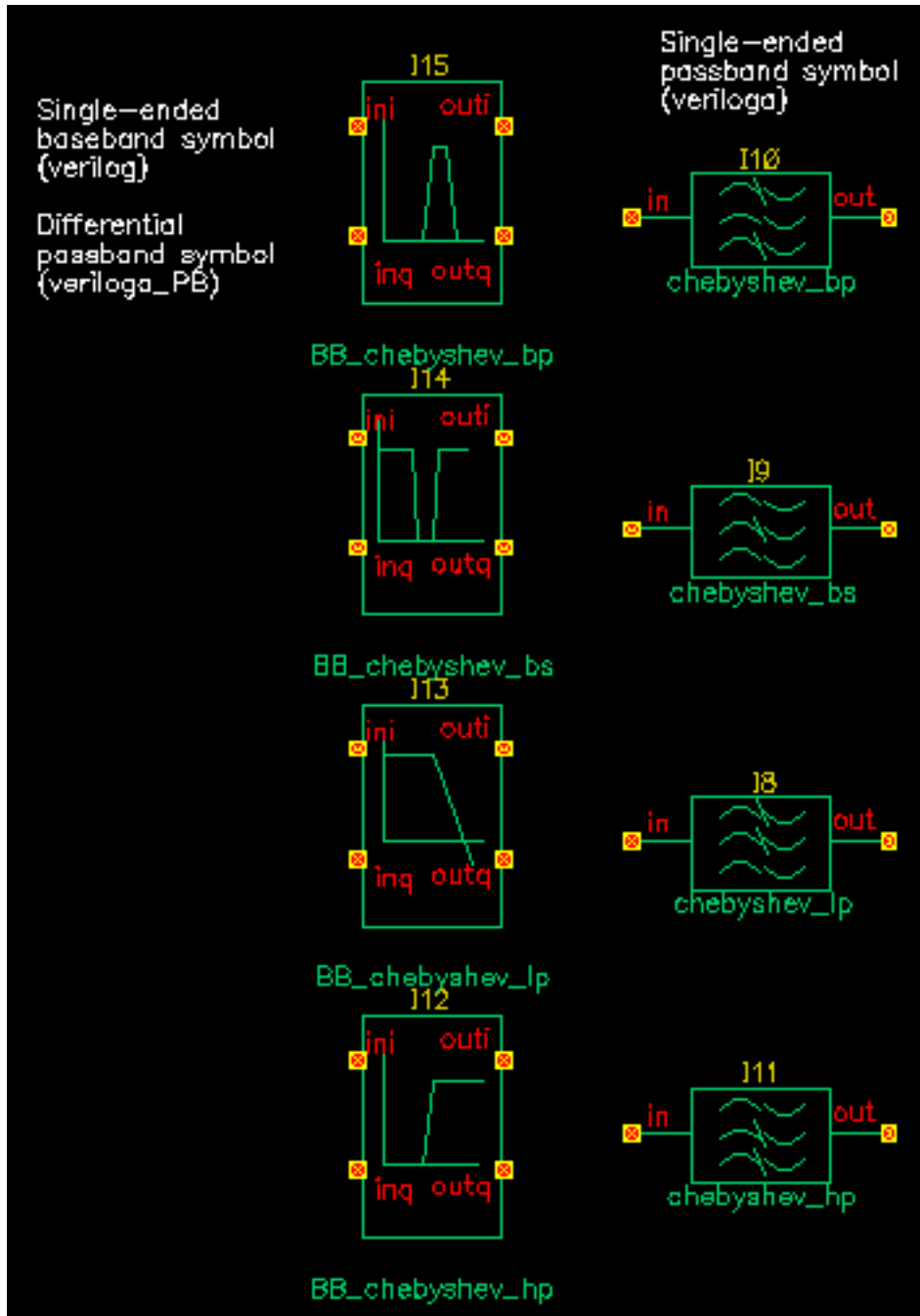


Figure 13-81 Chebyshev Filters



The single-ended passband filters are the filters in the original RF AHDL library described in Appendix D. The single-ended baseband filters are baseband versions of the filters described in Appendix D. The passband views of the baseband equivalent filter models are differential versions of the filters described in Appendix D.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

All filter models are LC ladder networks built up during the netlisting step. The only difference between the single-ended passband and baseband models is that the baseband models use baseband equivalent inductor and capacitor models, which means the baseband filter models have twice as many nodes as their passband counterparts. The only difference in the parameter list is that the baseband models require the carrier frequency. The carrier frequency is independent of all other parameters. It is the frequency at which the baseband signal is reference. If the input signal rides on a carrier that is offset from the reference carrier, the input baseband signal should be spun at the offset frequency before entering the filter model. This can be done at the preceding mixer stage by ramping the phase_err pin at a rate equal to the frequency offset.

Parameters (All parameters are defined in terms of passband models.)

Note: The input impedance must be less than or equal to the output impedance to produce the desired filter response.

Input and Output impedance:

The input and output impedances are resistive reference impedances used to compute the inductors and capacitors that make up the filter. **The input and output impedances are not terminal impedances.** If you load one side of a filter from the library with a resistor of R Ohms and drive the other side with a frequency-swept current source of 1 Amp, you will see a voltage of R Volts across the current source at the passband frequency. (If the filter is a bandstop filter the passband is on either side of the stop band.) The following steps demonstrate this point with the four Butterworth filters.

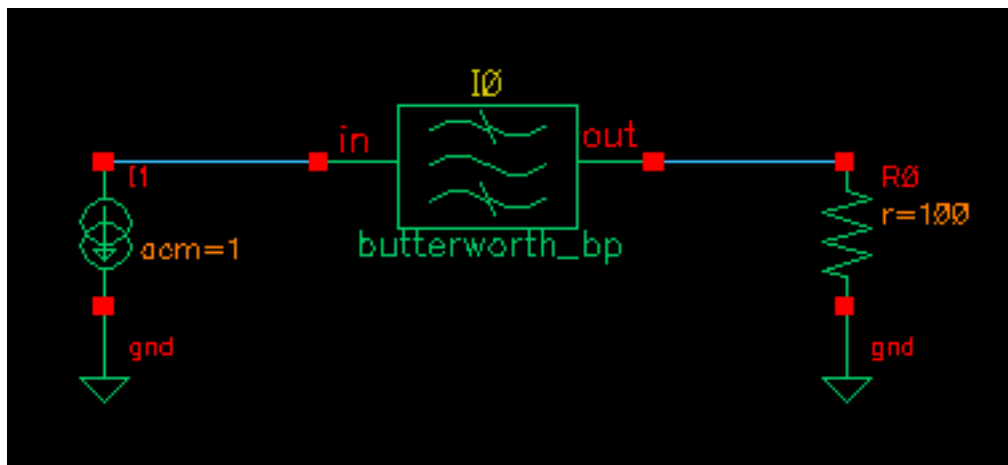
1. Bring up a new schematic in any library to which you can write.
2. Instantiate a butterworth_bp (bandpass) filter with the parameters as shown in

Figure 13-82 Filter parameters

Filter Order	3
Input impedance	200
Output impedance	200
Center frequency (Hz)	1e9
Relative bandwidth	0.1
Insertion loss (dB)	0

3. Drive the left terminal (the “input”) of the filter with a DC current source and set the source’s AC magnitude to 1 Amp.
4. Load the right terminal (the “output”) of the filter with a 100 Ohm resistor. The schematic should look like [Figure 13-83](#) on page 690.

Figure 13-83 Filter impedance test circuit



5. Set up an AC analysis. Sweep frequency from 950MHz to 1.05MHz in 100 linear steps.
6. Run the analysis.

7. Plot the voltage across the current source. You should see the voltage across the current source dip to 100 Volts right at the center frequency of 1GHz. At the center frequency, the impedance looking into a bandpass filter equals the impedance loading the other side, regardless of the filter's input and output impedance parameters.
8. Change the filter to `butterworth_bs` (bandstop filter).
9. Check and save the schematic.
10. In the AC analysis setup, change the AC analysis to sweep frequency from 1Hz to 100GHz in 1 linear step. Do not add any specific points.
11. Run the analysis.
12. Plot the voltage across the current source. You should see 100 Volts at either end point of the sweep. Far away from the center frequency of a bandstop filter, the impedance looking into the filter equals the impedance loading the other side.
13. Change the filter to `butterworth_lp` (low pass filter). Set the corner frequency to 10MHz.
14. Check and save the schematic.
15. Change the AC analysis to sweep from 1Hz to 1GHz logarithmically with 10 points per decade.
16. Run the analysis.
17. Plot the voltage across the current source. Near DC, the impedance looking into a low pass filter equals the impedance loading the other side.
18. Change the filter to `butterworth_hp` (highpass filter).
19. Change the AC analysis to sweep from 1GHz to 100GHz.
20. Run the Analysis.
21. Plot the voltage across the current source. As frequency goes to infinity, the impedance looking into a highpass filter equals the impedance loading the other side.

How are the input and output impedances used if they don't determine terminal impedances? Let the filter be specified with input and output resistances of `rin` and `rout` respectively. If you drive the filter with a Port of `rin` Ohms and load it with a Port of `rout` Ohms, an S-parameter analysis will produce an `S21` with the shape associated with the filter type and associated specifications. The following steps demonstrate this point.

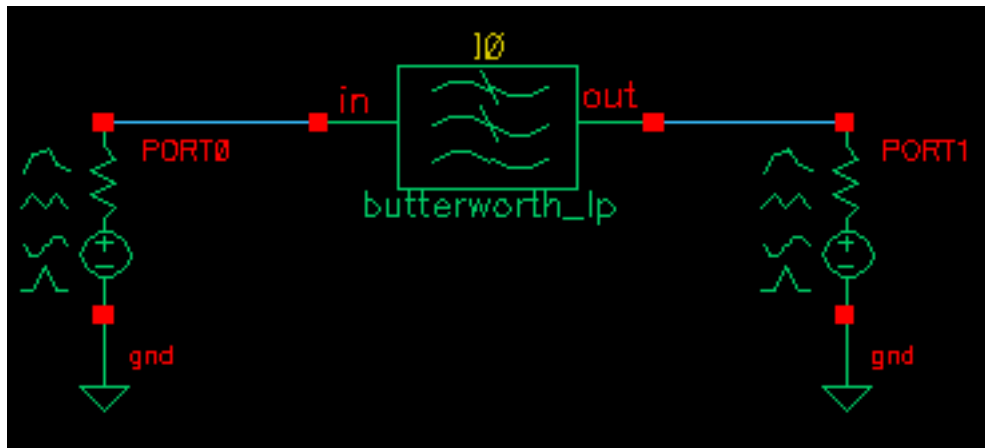
1. Bring up a new schematic window in a library you can write to.
2. Instantiate a `butterworth_lp` (low pass filter) and set the parameters as shown in [Figure 13-84](#) on page 692.

Figure 13-84 Filter parameters

Filter Order	100
Input impedance	rin
Output impedance	rout
Corner frequency(Hz)	10M
Insertion loss(dB)	0

3. Drive the filter a Port. Set the Port Resistance to rin Ohms and the Port number to 1.
4. Load the filter with another Port. Set the Port Resistance to rout Ohms and the Port number to 2. The schematic should look like [Figure 13-85](#) on page 692.

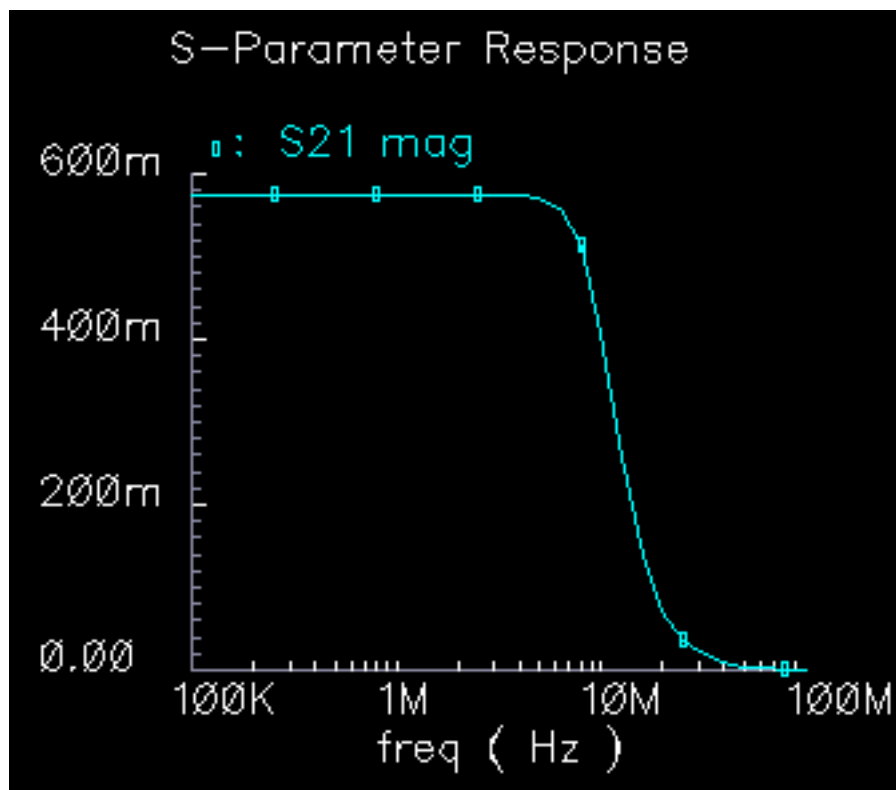
Figure 13-85 S21 test circuit



5. Bring up an Analog Environment window. Copy variables from the cell view. Set rout to 200 Ohms and rin to 20 Ohms.
6. Set up an sp (s-parameter) analysis. Sweep frequency from 100KHz to 100MHz logarithmically with 10 points per decade.
7. Run the analysis.

- Plot S21 and you should see the plot in [Figure 13-86](#) on page 693. The corner frequency is where S21 drops to 70.71% of its maximum value. If you check, that point is right where it was specified, at 10MHz. Note that the maximum value of S21 is not 1. That is because the input and output reference resistances are not equal. In the passband, the test circuit becomes a voltage divider. Using the basic definition of S21, for this voltage divider one can show that S21 equals the geometric mean of the input and output reference resistances divided by their arithmetic mean; the maximum value of S21 should equal $2*\text{Sqrt}(200+20)/(200+20) = 0.575$, just as the plot shows.

Figure 13-86 S21 plot



Insertion Loss:

This is the voltage drop in dB, at the passband or center frequency, the filter would suffer if it were driven and loaded with the input and output resistances respectively.

Filter noise is computed from the insertion loss according to:

$$\text{noise_current} = 2*\text{sqrt}((\text{'db10_real(loss)-1})*1.380620\text{e-}23*\text{\$temperature}/r1);$$

“r1” is the input impedance and “db10_real()” is a function that maps loss in dB into linear attenuation. The definition is as follows:

```
'define db10_real(x) (pow(10, (x)/10))
```

Filter Order:

Low pass-This is total number of reactive elements in the filter. If the order is odd, there is one more capacitor than inductors. If the order is even, there are equal numbers of capacitors and inductors.

High pass-This is the total number of reactive elements in the filter. If the order is odd, there is one more inductor than capacitors. If the order if even there are equal numbers of capacitors and inductors.

Band pass and Band stop- The filter order is the number of inductor/capacitor pairs.

Corner frequency:

This is the frequency where S21 is 3dB below the peak response when driven and loaded with the input and output resistances respectively.

Center frequency:

For bandpass filters, the center frequency is where S21 is maximum. For bandstop filters, the center frequency is where S21 is a minimum.

Relative bandwidth:

This applies to band pass and band stop filters. It is the normalized frequency difference between points straddling the center frequency that correspond a 3dB drop (or rise) from the peak (dip) of S21. The normalization factor is the center frequency. In other words, the relative bandwidth is specified in terms of a fraction of the center frequency. A relative bandwidth of 0.1 and a carrier frequency of 1GHz means that in the case of a bandpass filter, the pass band goes from 950MHz to 1.05GHz with the band edges defined by those points lying 3dB below the peak response.

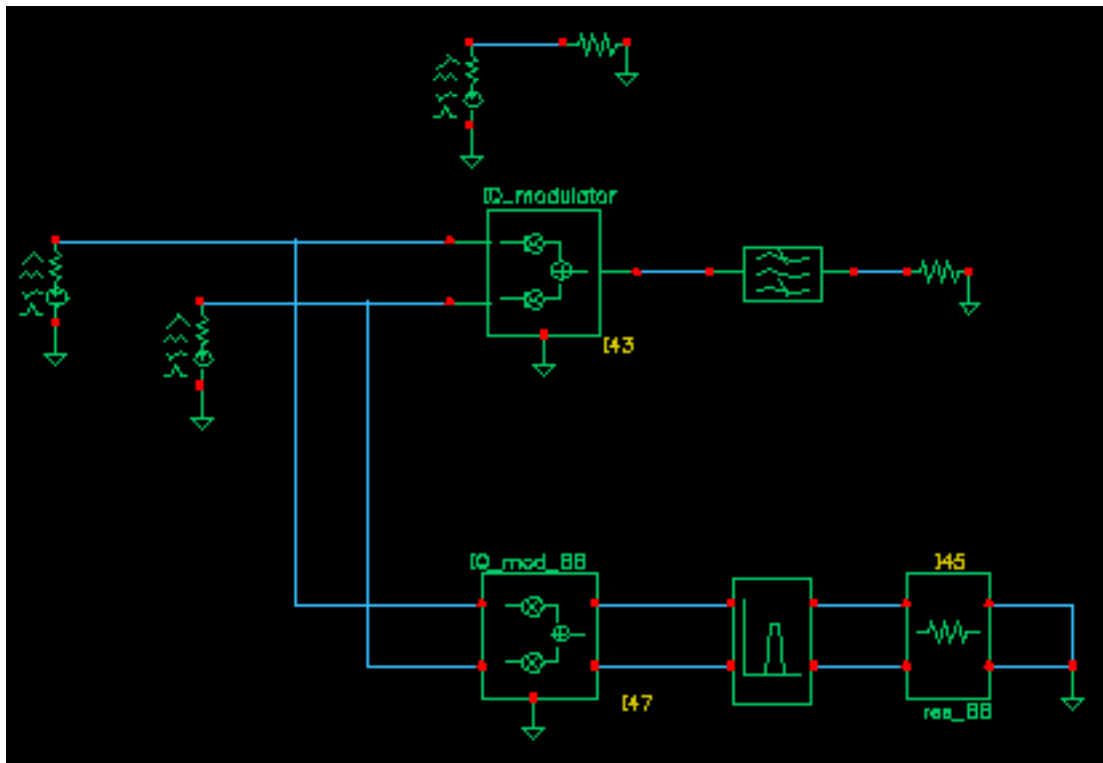
Passband ripple:

This applies only to Chebyshev filters. It is the peak-to-peak variation of the transfer function in the passband when the filter is driven and loaded with the input and output resistances.

Comparison of baseband and passband models

The circuit in [Figure 13-87](#) on page 695 shows how well the baseband and passband filters agree. The I-input is a 5MHz 1 volt peak sinusoid and the Q-input signal is a 20MHz 1 volt peak signal. The filter has a center frequency of 1.1GHz and a relative bandwidth of 0.1. The modulator LO is 1GHz. To make the analysis more interesting the carrier is not exactly aligned with the filter center frequency and the terminals are not matched. The circuit is listed as "PB_BB_filter_comparison" in the testbenches category of the rfLib.

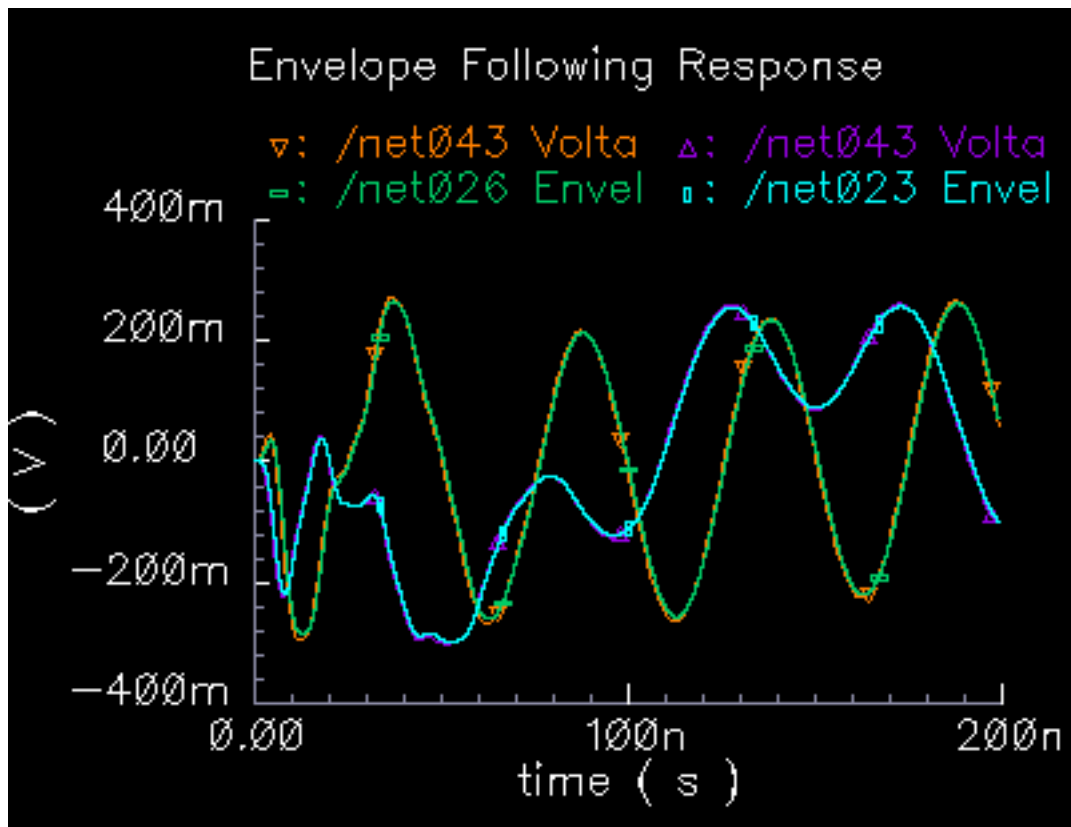
Figure 13-87 PB_BB_filter_comparison Circuit



1. Bring up the test circuit and an Analog Environment window.
2. Set up an Envelope Following analysis with "carrier" as the Clock Name. Set reltol in the analog options to 1e-5. You can use the default reltol of 1e-3 but you will not get the waveforms close to the baseband results.
3. Plot the "time" waveforms of the BB_butterworth_bp outputs. These waveforms are the response of the baseband equivalent model.
4. Plot the "harmonic time", 1 harmonic, real and imaginary waveforms at the butterworth_lp output. These waveforms are the baseband waveforms extracted from a

passband model. [Figure 13-88](#) on page 696 overlays the baseband and passband results. The baseband and passband filter models produce identical equivalent baseband waveforms. The slight offset in time is due to the ambiguity associated with deciding whether to plot a time-varying Fourier coefficient at the beginning or at the end of a clock cycle.

Figure 13-88 I and Q Baseband Equivalent Outputs



BB_Loss

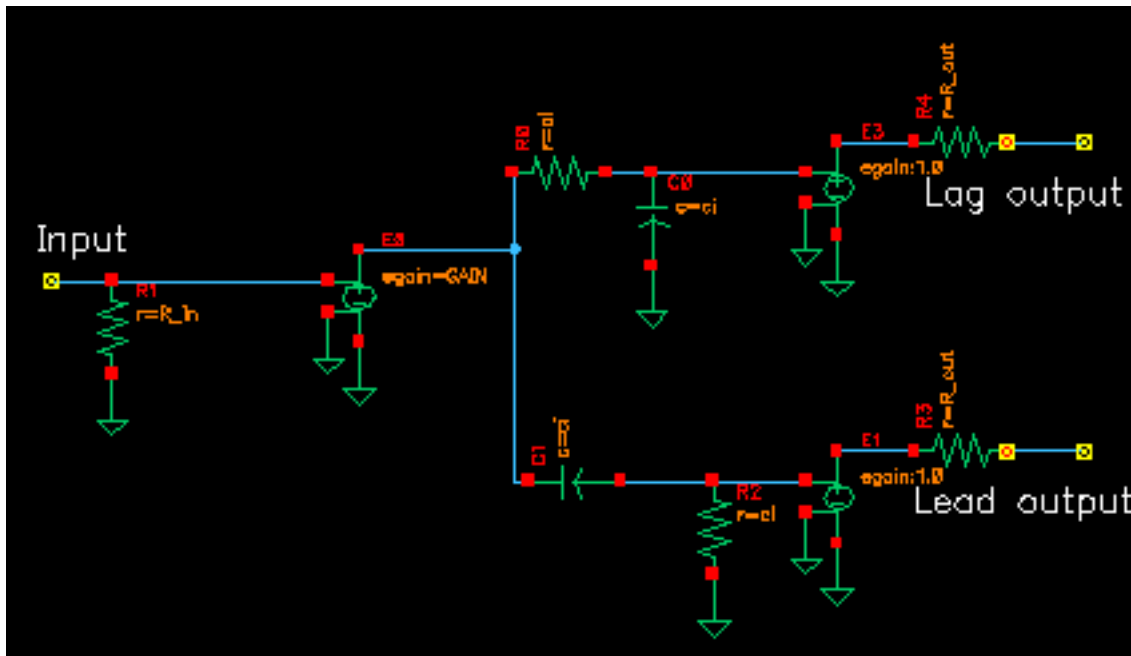
The BB_loss element is designed to be used with error vector magnitude (EVM) calculations. EVM is defined in terms of an ideal receiver or transmitter. If you want to remove a filter's response from the ideal receiver model while leaving only the passband attenuation, replace the filter with a BB_loss element and give it the same insertion loss as the filter. There is no passband view or counterpart for this model because a passband EVM analysis is not practical in this release.

Phase Shifter

(shifter-splitter)

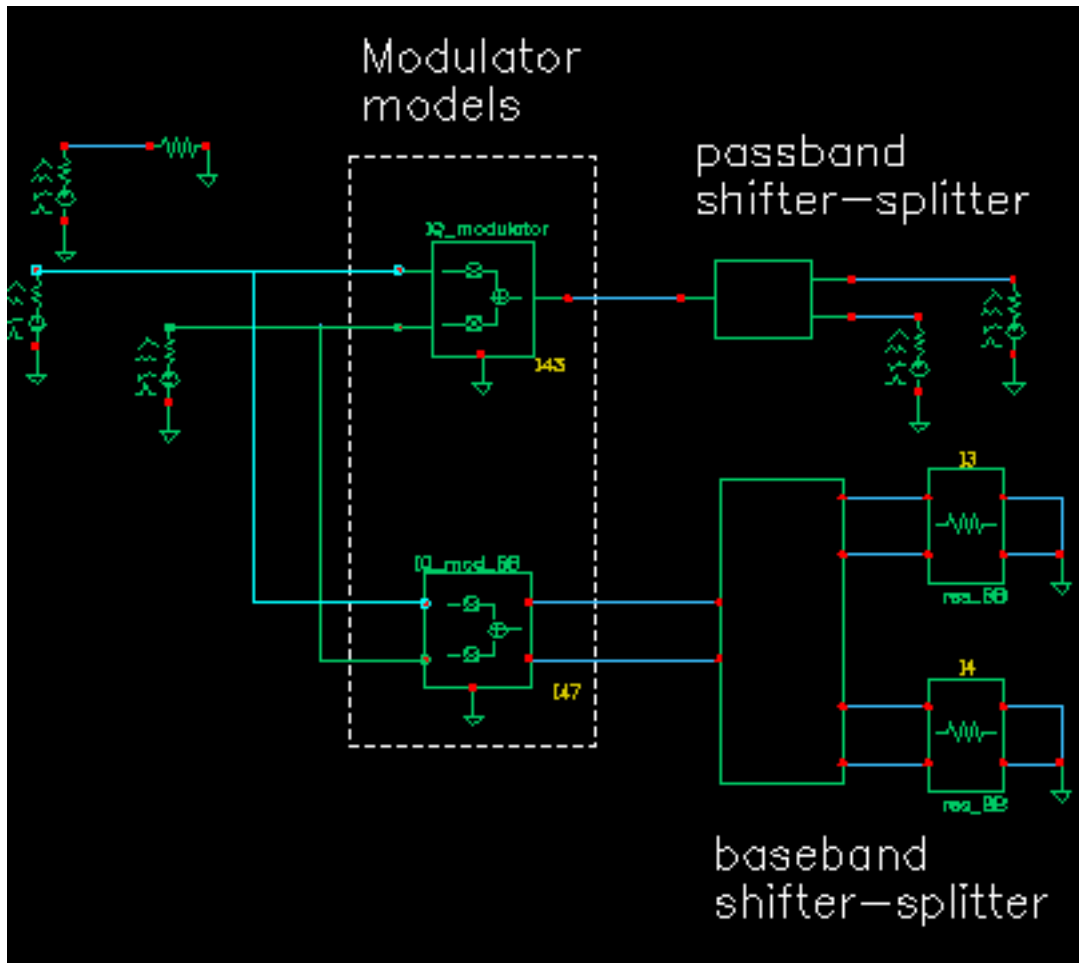
The phase shifter-splitter has one input and two outputs. The outputs are phase shifted versions of the input at the specified frequency. The phase difference between the two outputs is 90 degrees. The phase shifts are accomplished with VerilogA code that does the same thing as the circuit shown in [Figure 13-89](#) on page 698. The right-most voltage-controlled-voltage-sources (vcvs) are unity gain buffers. The left-most vcvs is also a buffer but the gain is a user-defined parameter. The input resistance, output resistance, intended operating frequency, and internal resistance are also user-defined. The internal resistance and operating frequency are used to calculate the capacitance necessary to provide ± 45 degrees of phase shifts at the operating frequency. The baseband view requires the carrier frequency.

Figure 13-89 Phase Shift Circuit



The test circuit in [Figure 13-90](#) on page 699 is for comparing the baseband responses of the passband and baseband equivalent models of the shifter-splitter. The circuit can be found in rfLib under the testbenches category. It is listed as "shifter_splitter_test".

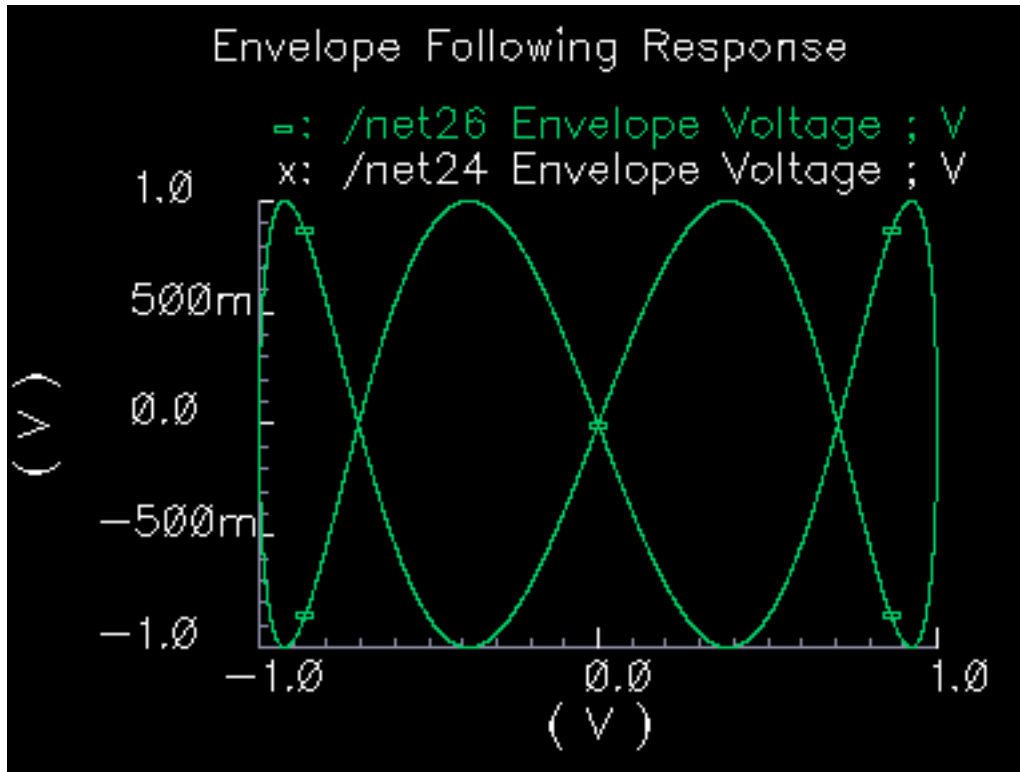
Figure 13-90 shifter_splitter_test Circuit



The following steps produce a set of Lissajous plots that show what the shifter-splitter does. We will observe phase shift in the carrier by observing the tilt of the output Lissajous figures generated by the equivalent baseband signals.

1. Bring up the test circuit in [Figure 13-90](#) on page 699 and an Analog Environment window.
2. Set up a 200ns Envelope Following analysis with “carrier” as the Clock Name. Set the Number of harmonics to 1. Set the Envelope Following analysis option called “modulationBW” equal to 100MHz.
3. Run the analysis.
4. Plot the “time” waveforms of the two input baseband signals. Change the x-axis to be the I-signal. You should see the Lissajous plot in [Figure 13-91](#) on page 700.

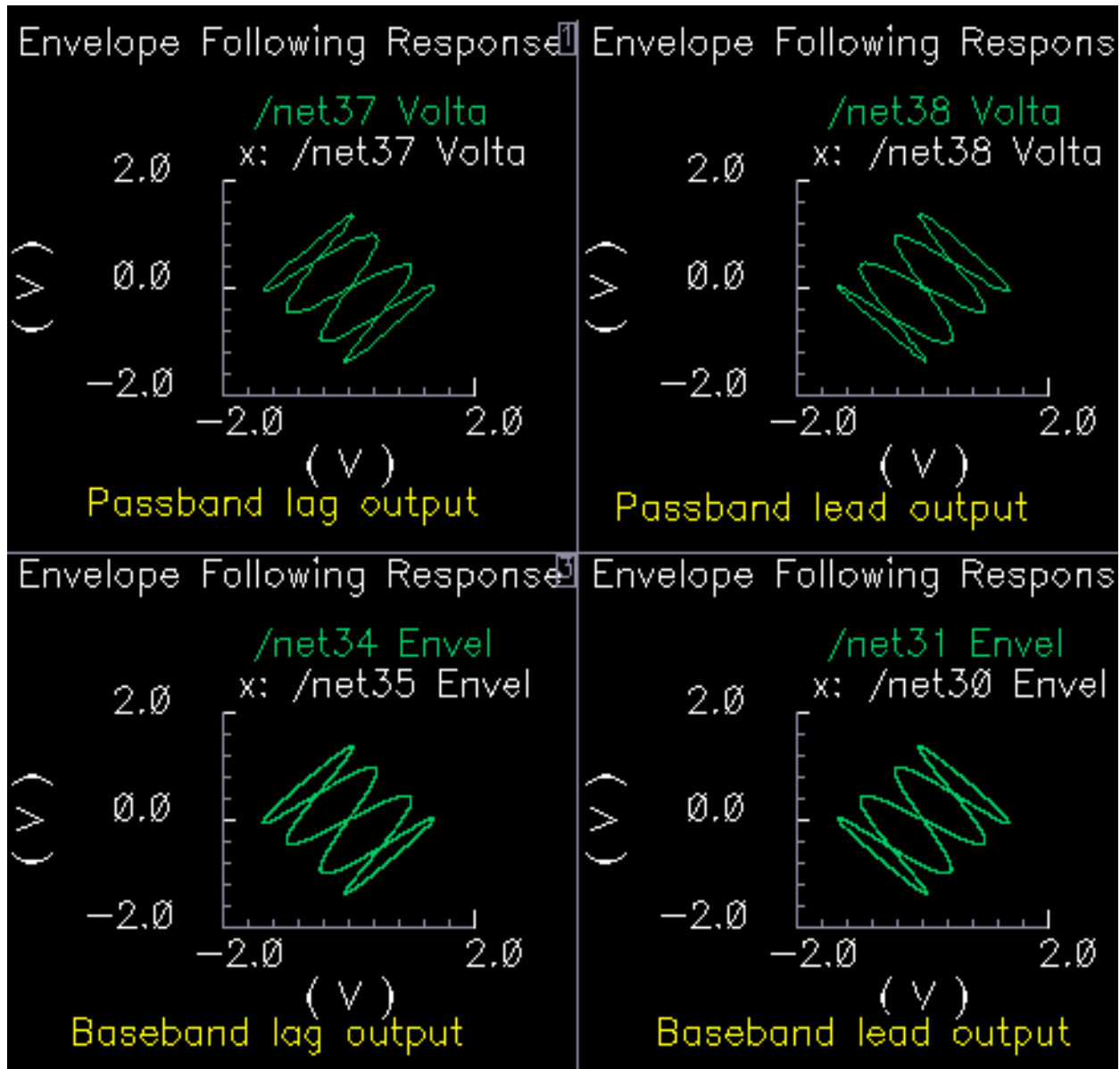
Figure 13-91 Lissajous Plot for Baseband Input Signals



5. Reset the Waveform Display window and plot the harmonic time, 1 harmonic, real and imaginary parts of the voltage across Port2. Set the x-axis to be the real part. Note that the Lissajous plot is tilted -45 degrees from the one in [Figure 13-91](#) on page 700.
6. Add a subwindow.
7. Repeat step 5 for the voltage across Port1. Notice that the Lissajous plot is tilted +45 degrees with respect to the Lissajous plot in [Figure 13-91](#) on page 700.
8. Add another subwindow.
9. Plot the "time" waveforms at the lag_I and lag_Q outputs of the BB_shifter_splitter model. Set the x-axis to be the lag_I waveform. The Lissajous plot should match the one produced in step 5.
10. Add another subwindow.
11. Repeat step 9 for the "lead" outputs of the BB_shifter_splitter model. The Lissajous plot should match the one produced in step 7. Aside from the labels, your Waveform Display tool should look like [Figure 13-92](#) on page 701. The time-results of the baseband model

faithfully duplicates the passband results but without simulating the carrier. The baseband model can be run with Spectre Transient analysis.

Figure 13-92 Comparison of Lag and Lead times for Passband and Baseband Models

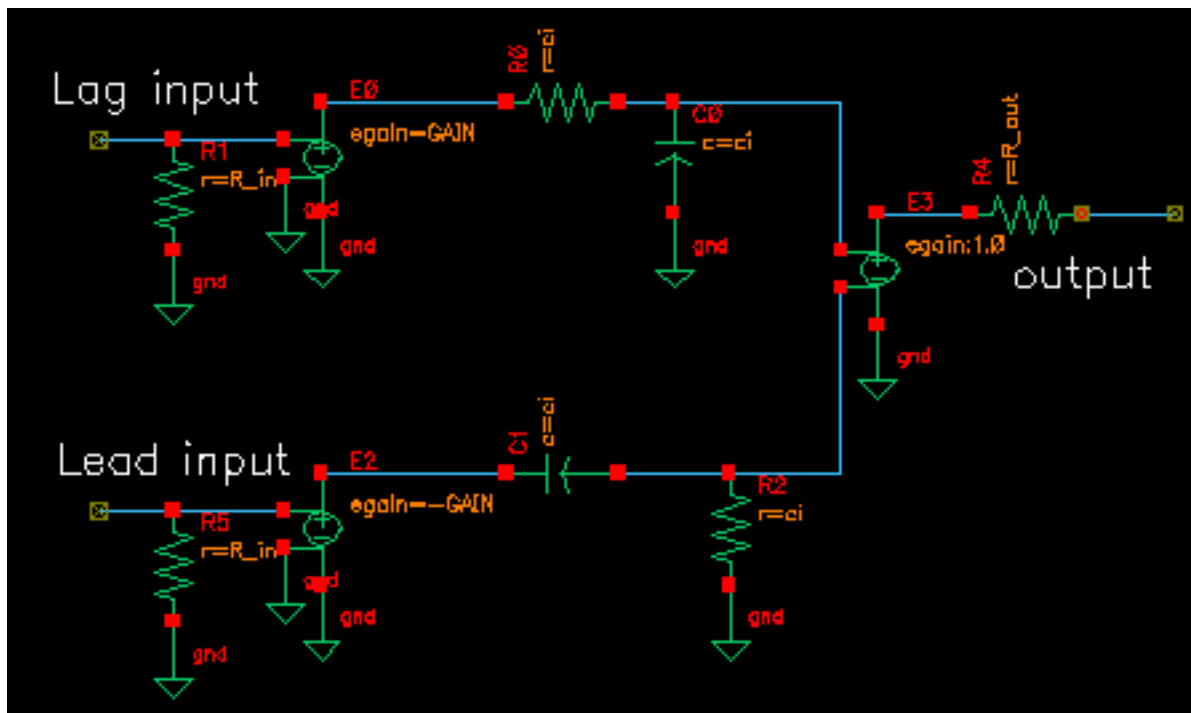


Phase Shifter

(shifter-combiner)

The phase shifter-combiner has two inputs and one output. The inputs are phase shifted by +/- 45 degrees then added together to form the output. All terminals are buffered and have the specified terminal resistances. The phase shifts are accomplished with VerilogA code that does the same thing as the circuit shown in [Figure 13-93](#) on page 702. The gains of the left-most voltage-controlled-voltage sources are user-defined. The input resistance, output resistance, intended operating frequency, and internal resistance are also user-defined. The internal resistance and operating frequency are used to calculate the capacitance necessary to provide 45 degrees of phase shifts at the operating frequency. The baseband view requires that the carrier frequency be specified.

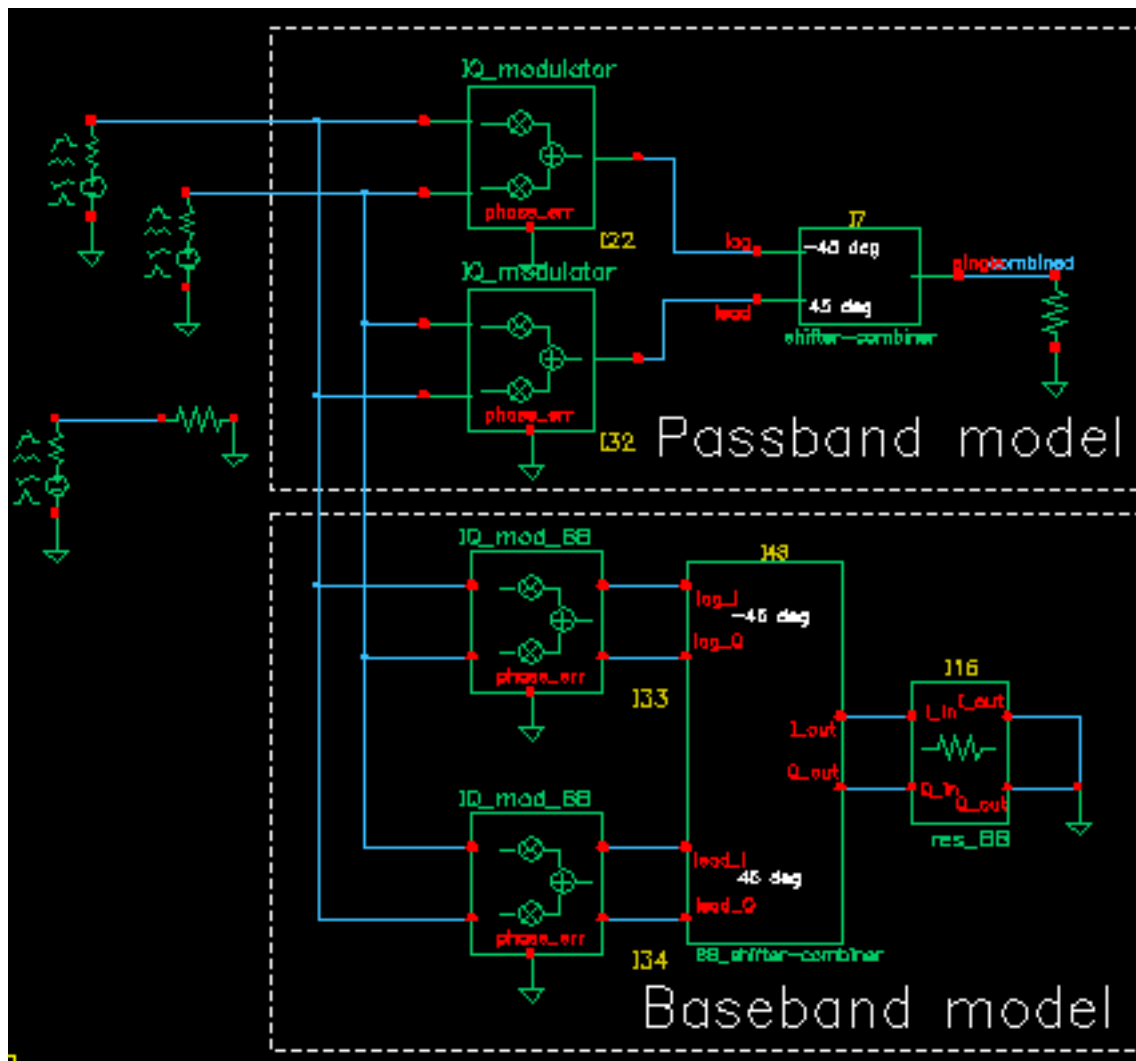
Figure 13-93 Phase Shift Circuit



The shifter-combiner can be used to eliminate one phase of the carrier. The test circuit in [Figure 13-94](#) on page 703 shows a simple test to demonstrate the idea. The circuit is in the rfLib under the testbenches category and listed as "shifter_combiner_test". The top circuit is a passband model and the bottom circuit is the baseband equivalent. Baseband input signals are mixed up to 1 GHz then passed into the shifter-combiner. The baseband signal contains

10MHz and 20MHz components. The modulators and shifter-combiner are arranged to produce only a 20MHz signal riding on the carrier.

Figure 13-94 shifter_combiner_test Circuit

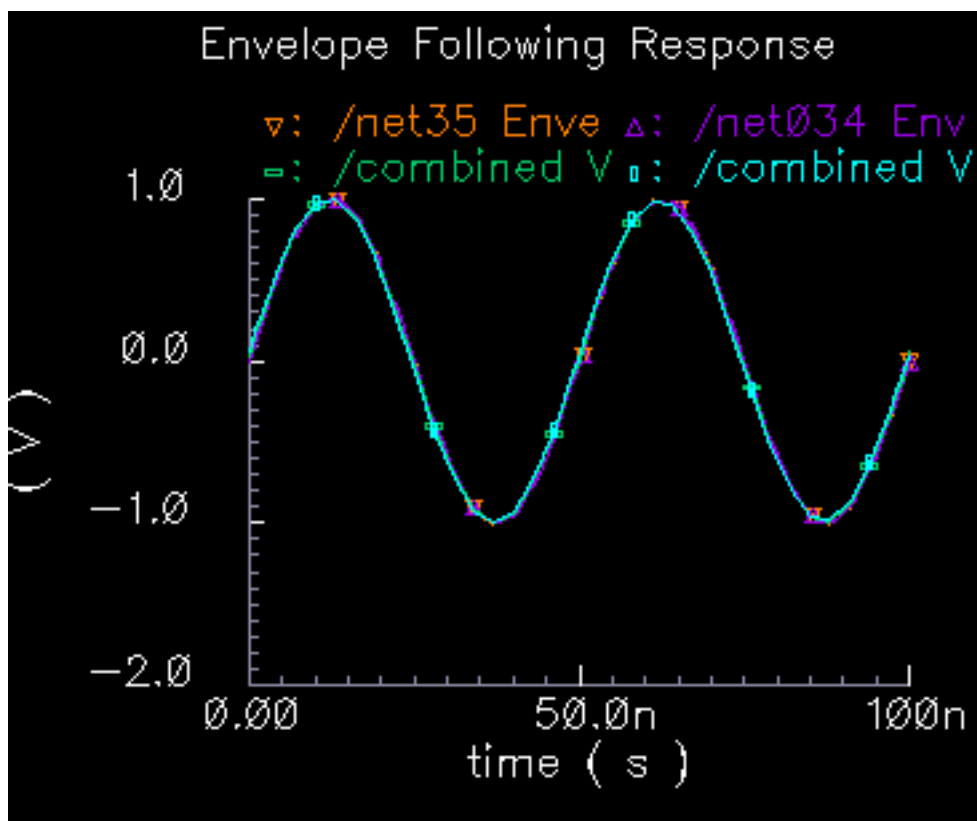


To check this assertion:

1. Bring up the test circuit and an Analog Environment window.
2. Set up a 100ns Envelope Following analysis on the circuit with the Clock Name set to "carrier" and the modulationbw option set to 40MHz. Set the Harmonic number to 1.
3. Run the analysis.

4. Plot the harmonic time, 1 Harmonic, real and imaginary parts of the passband shifter combiner output.
5. Append to the plot, the “time” waveforms at I_out and Q_out pins of the BB_shifter_combiner model. [Figure 13-95](#) on page 704 shows what you should now see in the Waveform Display window. All waveforms are the same and they contain only the 20Mhz baseband signal. The 10Mhz baseband input signal does not propagate to the output.

Figure 13-95 shifter_combiner_test results



One application of the shifter-combiner is an image rejection receiver. [Figure 13-96](#) on page 705 shows a very simple example of an image rejection receiver. [Figure 13-97](#) on page 705 shows the baseband equivalent model of the receiver. Both examples are in the rfExamples directory and are listed as "image_reject_rcvr_PB" and "image_reject_rcvr_BB". The local oscillator runs at 1GHz and the RF carrier is 1.1GHz, which places the image at 900Mhz. This example shows one of the limitations of the baseband equivalent models.

Figure 13-96 A Simple Image Rejection Receiver

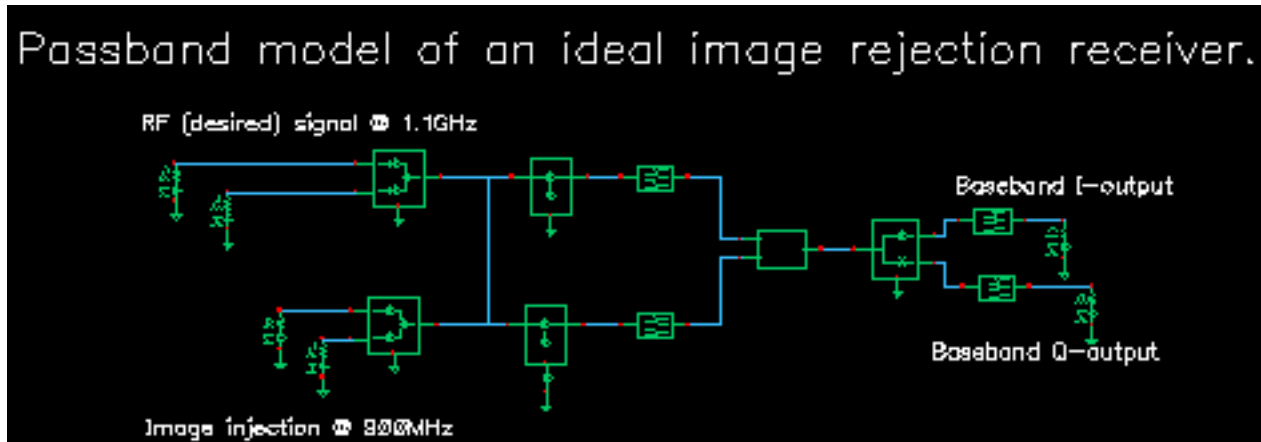
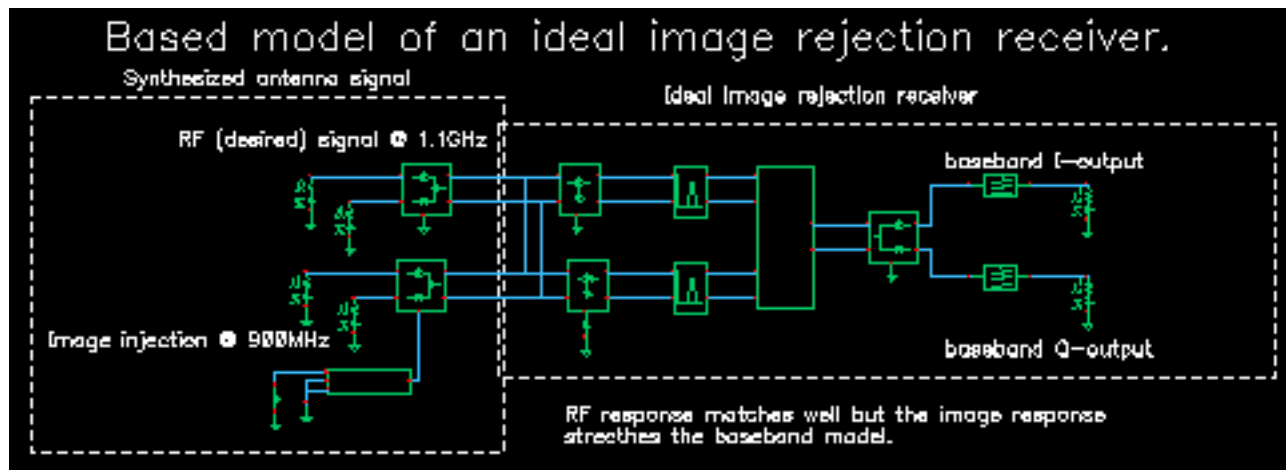


Figure 13-97 Baseband Equivalent Model of the Image Rejection Receiver



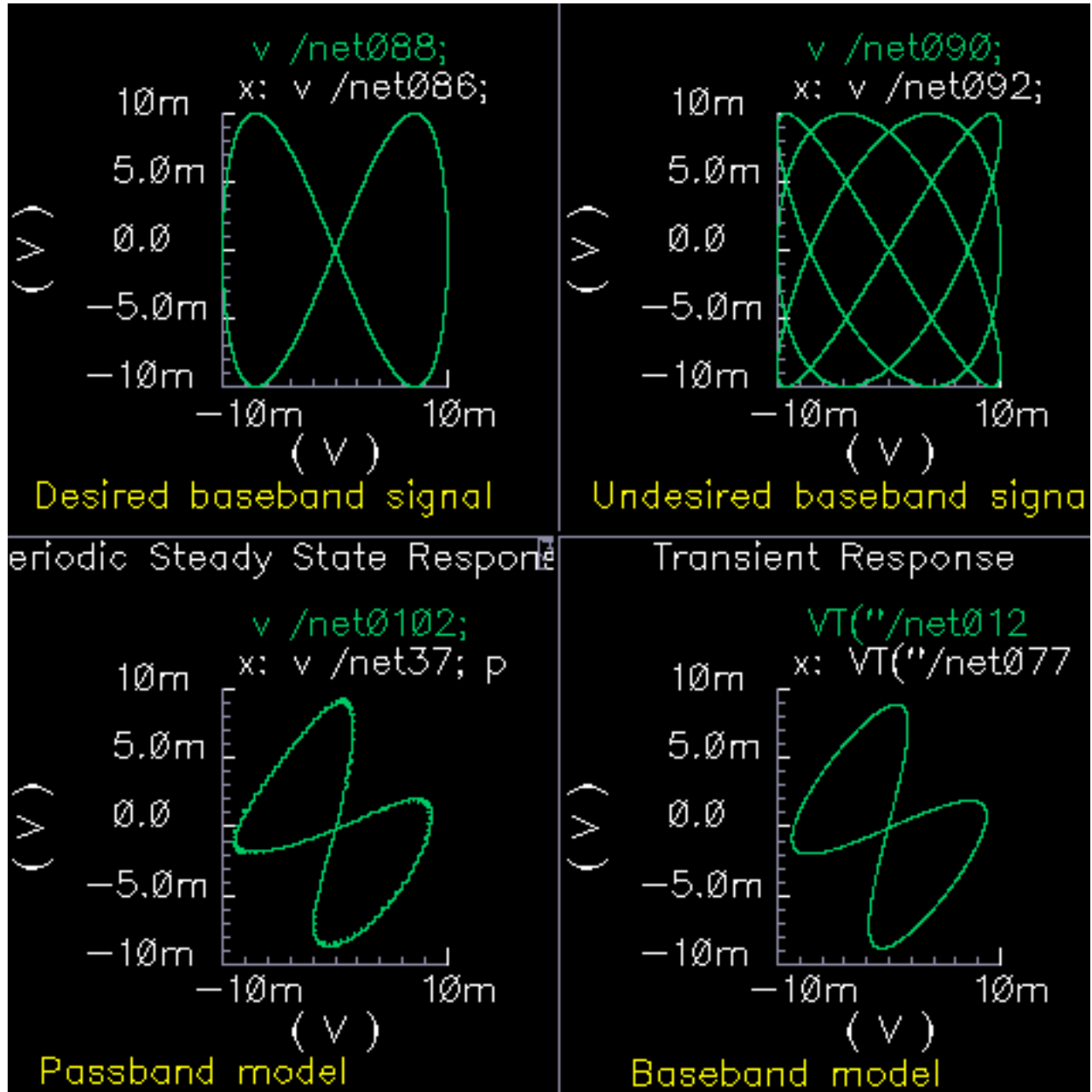
1. Bring up the passband test circuit and an Analog Environment tool.
2. Set up a PSS analysis. You will need to add the 1.1GHz, 1GHz, and 900MHz fundamental tones. Give them arbitrary but distinct names. AutoCalculate the Beat Frequency, which should be 2MHz. You need not save more than the 1st harmonic. Set the PSS maxstep option to 20ps so that it accurately simulates the oscillators hidden inside the Verilog-A modules. (In a future release, the modules will have Boundstep lines to take care of this problem.)
3. Run the analysis.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

4. Plot the voltages across Ports 5 and 6. Set the x-axis to be the voltage across Port 6. This is a Lissajous plot of the desired baseband signal, the one riding on the 1.1GHz carrier.
5. Add a subwindow.
6. Plot the voltages across Ports 8 and 7. Set the x-axis to be the voltage across Port 8. This is a Lissajous plot of the undesired baseband signal, the signal riding on the image of the carrier at 900MHz.
7. Add another subwindow.
8. Plot the I and Q- baseband outputs. Set the x-axis to be the I-output. The Lissajous plot is a tilted version of the desired baseband signal, indicating that most of the image was successfully rejected.
9. Bring up the baseband equivalent receiver model and another Analog Environment tool.
10. Run a 10us Transient analysis with 9.5us as the output start in the analysis options and maxstep set to 250ps. The phase_err pin on the image signal generator is being driven to spin the output at 200MHz, the frequency difference between the desired frequency and image frequency.
11. Add another subwindow to the Waveform Display tool showing the passband results and make sure it is active.
12. Plot the I and Q baseband outputs from the baseband equivalent receiver model. Set the x-axis to be the I-output. You may need to make the scales on the last two plots are the same. Aside from the labels, the Waveform Display tool should look like [Figure 13-98](#) on page 707.

Figure 13-98 Lissajous Plots for Baseband Signals



The baseband equivalent receiver model indeed rejects the image but the rejection is over-estimated. If you look closely, the baseband output of the passband model contains more ripple from the image. The over-attenuated ripple in the baseband model is explained as follows.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

Recall the rotating reference frame analogy for baseband modeling. With respect to the rotating 1.1GHz reference frame, the image signals rotate counter-clockwise at twice the IF, 200MHz in this case. The lower left block in the baseband receiver model spins the modulator output at -200MHz by ramping the phase error pin. The -200MHz signal propagates through the IF bandpass filters, as it should, because the response of the baseband model of the filter peaks at DC and at minus 200MHz. The trouble occurs in the final downconversion to baseband. In the baseband model, the final low pass filters severely attenuate the -200MHz image signal. However, in the passband model, image power at minus 100MHz contributes to the baseband signal through the low pass filters with less attenuation.

This example highlights one of the limitations of baseband equivalent models: at any point in the system, the signal should not have a bandwidth larger than any carrier (RF or IF) of the system. For this example, the baseband model is only valid for input RF signals between 1GHz and 1.2GHz.

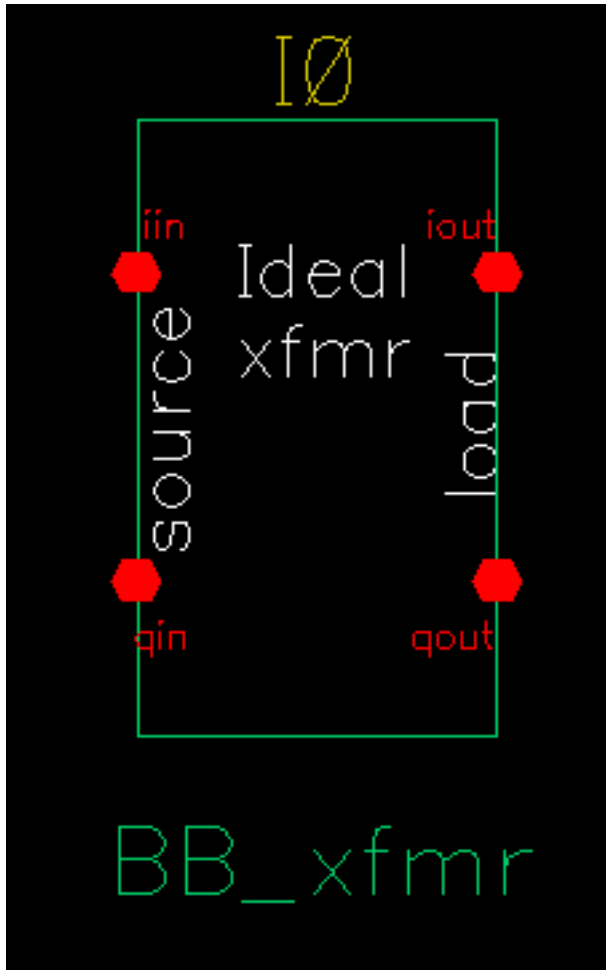
The limitation is somewhat moot because the idea behind a baseband equivalent model is to suppress all carriers. To simulate the image response with the baseband model we had to include a 200MHz source! We would have been better off simply not suppressing the 100MHz IF carrier, i.e. using baseband models for the RF stages but passband models for the IF stages.

In summary, an all-baseband equivalent model of an image rejection receiver is only good for simulating the response to the desired RF signal, not the image response.

Ideal Transformer

(BB_xfmr)

Figure 13-99 Lissajous Plots for Baseband Signals



The ideal transformer is in the "measurement" category of the rfLib. It's purpose is to help designers transform between different resistances.

- Parameters: source resistance R_s and load resistance R_l .
- Inputs = i and q input voltages, i and q output currents.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

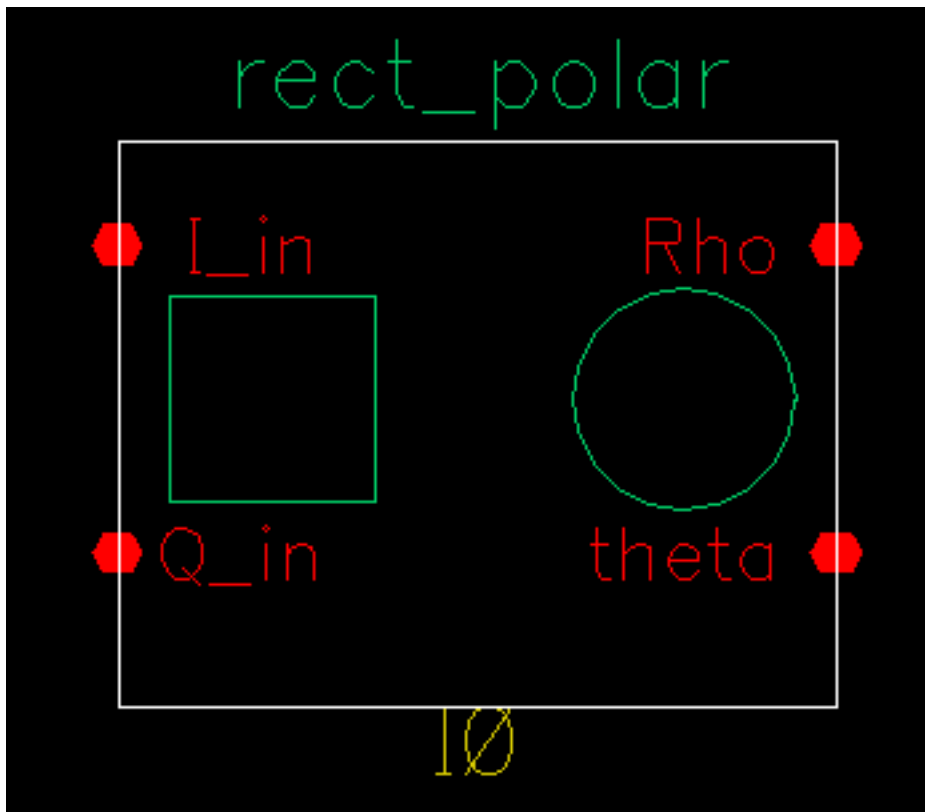
- Outputs = i and q output voltages, i and q input currents defined as follows:

$$\begin{bmatrix} i_{iin} \\ i_{qin} \end{bmatrix} = \sqrt{\frac{RI}{Rs}} \begin{bmatrix} i_{iout} \\ i_{qout} \end{bmatrix}$$

$$\begin{bmatrix} v_{iout} \\ v_{qout} \end{bmatrix} = \sqrt{\frac{RI}{Rs}} \begin{bmatrix} v_{iin} \\ v_{qin} \end{bmatrix}$$

Rectangular-to-Polar Transformation

Figure 13-100 Rectangular-to-Polar Block Symbol



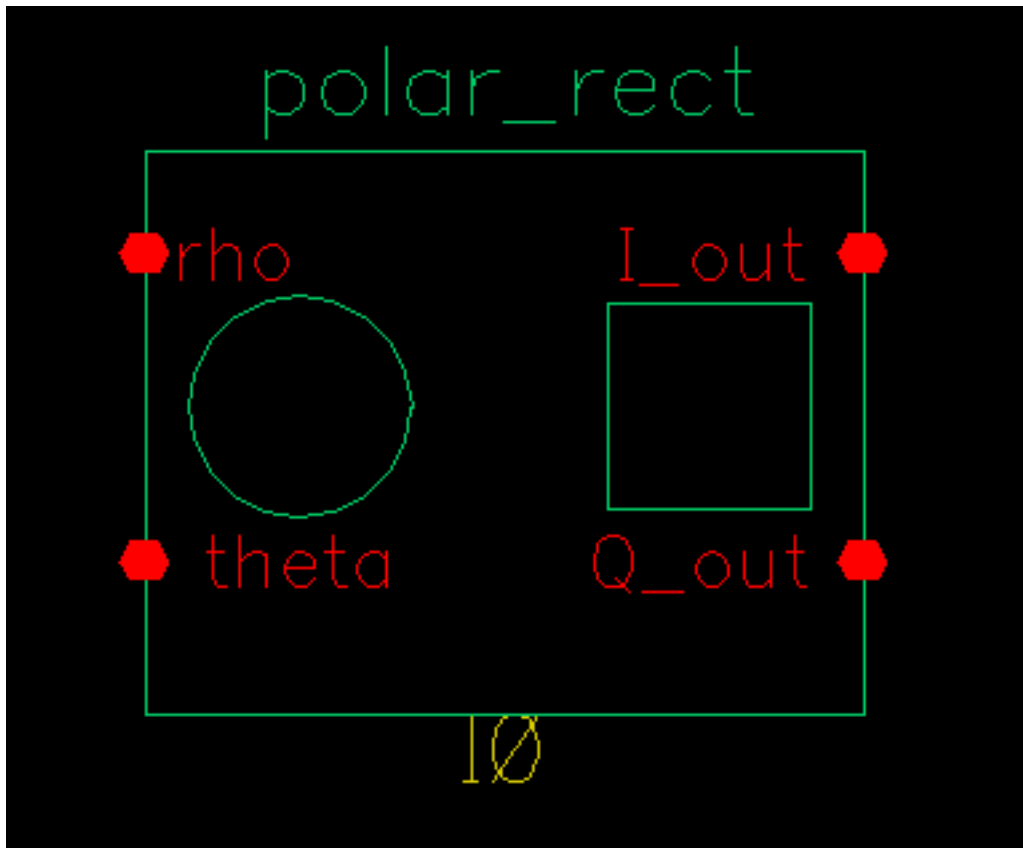
The rectangular-to-polar block is in the “measurement” category. The only parameters are input and output resistances. The inputs are the baseband signal in Cartesian coordinates, the outputs are the baseband signal in polar coordinates.

Parameters: Input and output resistances.

- Inputs = i and q volts.
- Outputs = ρ and θ volts such that
- $\rho = \text{Sqrt}[i*i+q*q]$ and
- $\theta = \text{arctan}[q/i] + \pi*(1+\text{sgn}[i])/2$ radians, with appropriate checks for the $i=0$ case.

Polar-to-Rectangular Transformation

Figure 13-101 Polar-to-Rectangular Block Symbol



The polar-to-rectangular block is in the “measurement” category. The only parameters are input and output resistances. The inputs are the baseband signal in polar coordinates, the outputs are the baseband signal in rectangular coordinates.

- Parameters: Input and output resistances.
- Inputs = ρ and θ volts.
- Outputs = i and q volts such that
- $i = \rho \cdot \cos(\theta)$ and
- $q = \rho \cdot \sin(\theta)$.

Instrumentation and Terminating Blocks

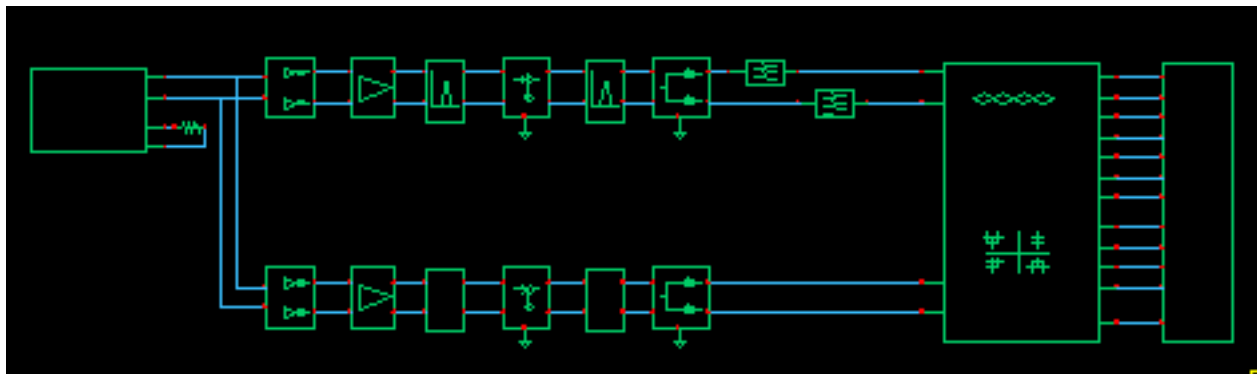
The instrumentation block is in the testbench category and generates waveforms that can be used to create eye-diagrams, eye-diagram statistics, scatter plots, and rms error-vector-magnitude. An example was given in [“Use Model and Design Example”](#) on page 596. This section summarizes the parameters and outputs.

There are two kinds of instrumentation blocks, one called "comms_instr" and one called "offset_comms_instr". The offset block is identical to the first one except the sampling time for scatter plots and eye-diagram statistics are delayed by half a symbol period. The delay makes it possible to plot symbols in an offset QPSK modulation scheme.

There is one other block in the library associated with the two instrumentation blocks, the "instr_term" model. The instr_term block simply loads all instrumentation output pins with 50 Ohms. The instr_block simply keeps the schematic editor from complaining about unconnected pins, nothing more.

[Figure 13-102](#) on page 713 shows how the offset_comms_instr and instr_term blocks should be used. The comms_instr block is used similarly. The circuit consists of two branches driven from a single baseband signal generator. The top branch is the non-ideal receiver model, the bottom branch is an ideal version of the top branch. The ideal version is ideal as you like. The ideal branch computes ideal symbol locations in the complex plane. The instrumentation block compares ideal and non-ideal symbols to compute error-vector-magnitude.

Figure 13-102 EVM setup



Parameters:

symbols per second. This parameter is necessary for generating the sawtooth which will be used as the x-axis to generate eye-diagrams. It also determines the rate at which the input waveforms will be sampled.

I-sampling delay. This parameter sets the phase of the symbol sampler. It is referenced to the eye-diagram of the I-output. Estimate the optimal delay by doing one simulation just to get the eye-diagram. The optimal delay is the time from the leftmost part of the eye-diagram, which should be zero, to the time at which the eye is widest.

number of symbols. This is the number of symbols to sweep in the eye-diagram. Sweeping two symbols ensures that you will see at least one continuous eye, if it is open.

max, min eye-diagram volts, and number of hstgm bins. These parameters are used to compute the bins which define the eye-diagram histogram. The bin width equals $(\text{max voltage} - \text{min voltage}) / (\text{number of bins})$. The histogram shows the distribution of the I_in voltage at the sampling instant.

I-noise and Q-noise(volts²). These parameters set the variance of Gaussian random variables which can be added to the received symbols before anything is computed or plotted.

statistics start time. This parameter delays the start of any statistical computations. The purpose is to exclude start-up transients from the statistics.

input resistance. This parameter is the input resistance of the input terminals of the instrumentation block.

Eye-diagrams. To generate an eye-diagram of the I-input signal, plot the sawtooth and I-eye outputs in one waveform display tool. Change the x-axis to be the sawtooth. This is done through the x-axis menu in the waveform display tool. The procedure for generating an eye-diagram of the Q-output is the same except you use the Q-eye output.

Histograms. You can only generate a histogram of the I-input signal. The histogram shows the distribution of the I-input voltage at the sampling instant. To generate a histogram, plot the eye_hist and eye_count_hist outputs in the same waveform display window. Change the x-axis to be the eye-hist signal then change the plot to "bar".

Eye-diagram statistics. The ave_eye output is the average absolute value of the I-input signal at the sampling instant. The root_var_eye output is the square root of the variance of the absolute value of the I-input voltage at the sampling instant. The voltages at these output pins represent running estimates of the associated statistics.

Scatter plots. A scatter plot is the I-input and Q-input samples plotted against each other. The scatter plot shows the locations of the received symbols. To generate a scatter plot, plot I_scatter and Q_scatter in the same waveform display window then change the x-axis to be the I_scatter signal. Finally, change the plot to plot data points only. A scatter plot of the reference model can be generated similarly by replacing I_scatter and Q_scatter with Iref_scatter and Qref_scatter.

rms Error Vector Magnitude. The rms EVM is defined as the square root of the sum of the squares of the vectorial differences between the ideal and non-ideal received

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

symbols, normalized to the rms value of the magnitude of the ideal received symbols. The output voltage at this pin is represents a running calculation of the rms EVM. The normalized EVM is in percent.

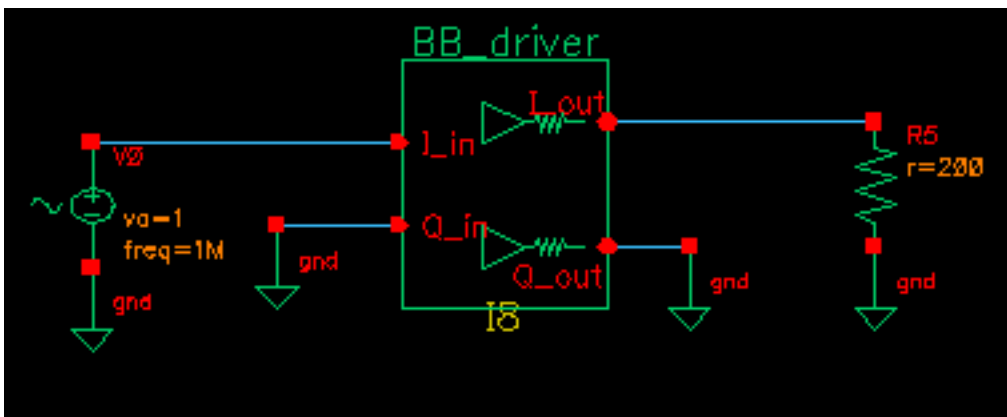
Baseband Drive Signals

Baseband signals can be imported into a Spectre Transient simulation through the Ports by setting the Source type to pwl and specifying the path to a file containing the data. You can also use the ppwlf sources. The ppwlf sources recognize SPW format.

You can also use one of the three baseband signal generators in the “measurement” category, which are described below. The baseband signal generators can drive J-models directly but should be used with a BB_driver model when driving top-down models. The BB_driver model is described first.

BB_driver

Figure 13-103 BB_driver



The BB_driver converts peak volts into dBm. It is intended for use with the baseband signal generators. The BB_driver is nothing more than an identical pair of voltage-controlled current sources with identical output resistances. The I_out pin is independent of the Q_in pin and the Q_out pin is independent of the I_in pin. The following example describes what this model does. [Figure 13-103](#) on page 716 shows the set up.

1. Build the schematic. Set the Output resistance parameter to 200 Ohms and the “dBm-out@1v peak in” parameter to 10.
2. Load one output with 200 Ohms and drive the associated input with a sinusoidal voltage that has a peak voltage of 1 volt.
3. Simulate a few cycles with a Transient analysis. The power delivered to the 200 Ohm load will be 10 dBm, which corresponds to a peak voltage of 2 Volts across the 200 Ohm load.

CDMA_reverse_xmit

Figure 13-104 on page 717 shows the symbol for the CDMA_reverse_xmit generator with the BB_driver. Figure 13-105 on page 718 shows the signal trajectory for the CDMA_reverse_xmit generator.

Figure 13-104 CDMA_reverse_xmit generator

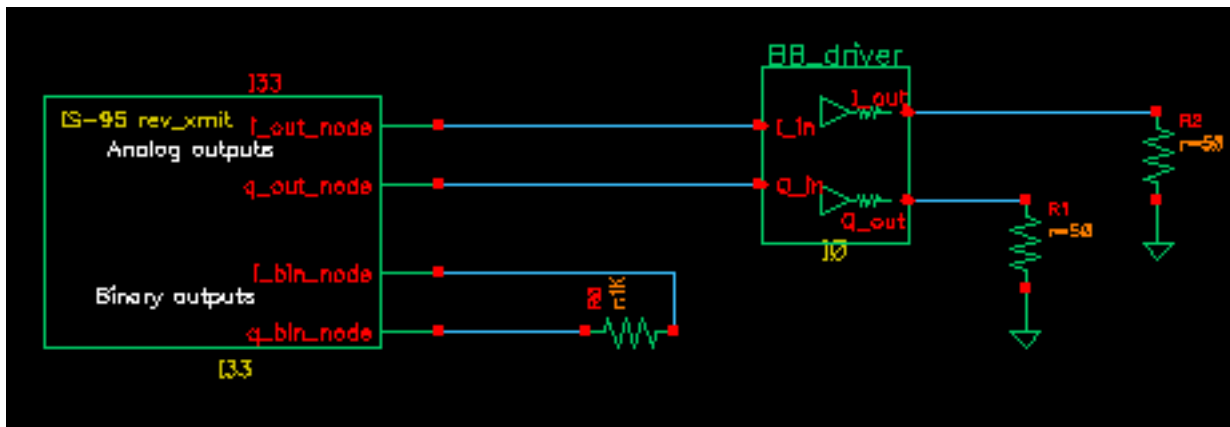
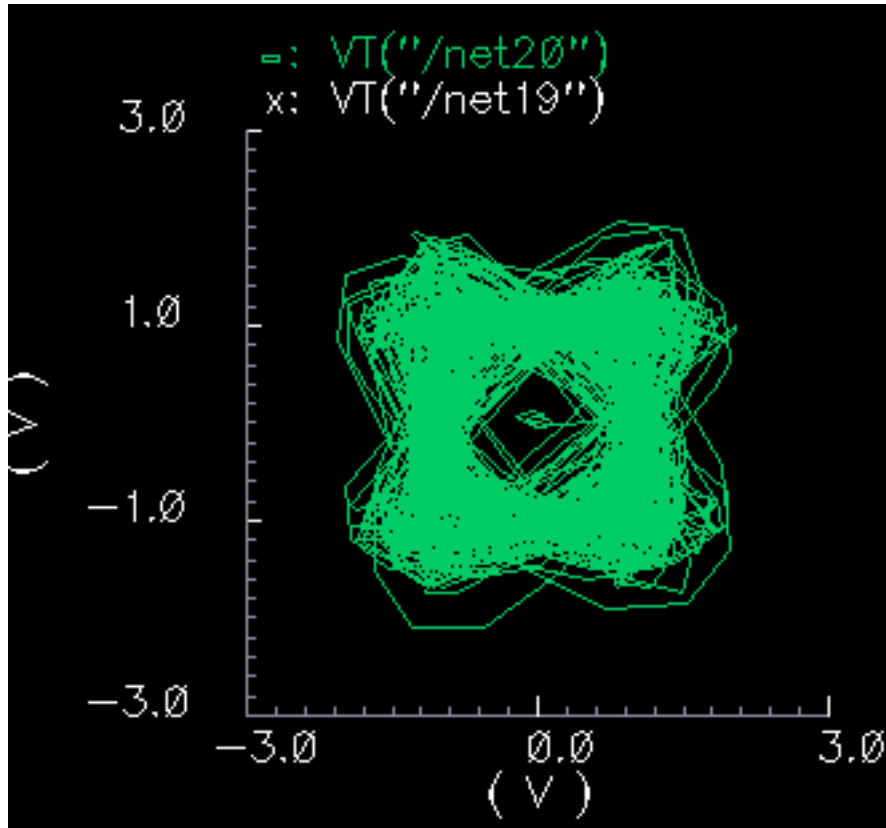


Figure 13-105 Signal Trajectory for CDMA_reverse_xmit



This model generates an IS-95 reverse link transmitter signal. The modulation is offset QPSK. A single stream of random bits is spread with two different spreading sequences to generate the I and Q outputs. The spreading sequences are generated with pseudo noise generators that have 15 states. The resulting chip¹ rate is 1.2288 Megachips per second on each output. Each chip is over sampled at a 4:1 rate. Each stream of samples drives its own 48-tap FIR filter. The Q-output is delayed by 2 samples, half a chip, to convert the QPSK signal to offset QPSK.

The `i_out_node` and `q_out_node` pins produce voltages equal to the digitally filtered I and Q outputs.

The `i_bin_node` and `q_bin_nodes` are the binary outputs. The binary outputs are the raw unfiltered chips.

The model has three parameters:

1. A chip is a piece of a bit. It is the product of a bit in the spreading sequence and the original bit that is being spread.

seed: This is the seed for the random number generator that generates the bits.

amplitude: This sets the peak amplitude of the i_out_node and q_out_node pins.

t-rise_fall: The digitally filtered outputs jump from one value to the next in a time specified by this parameter. To generate outputs consisting of data points connected by straight lines, set t-rise_fall to the chip (or symbol) period. For more of a staircase waveform, make this parameter a small fraction of the symbol period.

The spreading sequences and FIR filters can be redefined through the Modelwriter.

GSM_xmtr

Figure 13-106 on page 719 shows the symbol for the GSM_xmtr generator with the BB_driver. Figure 13-107 on page 720 shows the signal trajectory for the GSM_xmtr generator.

Figure 13-106 GSM_xmtr generator

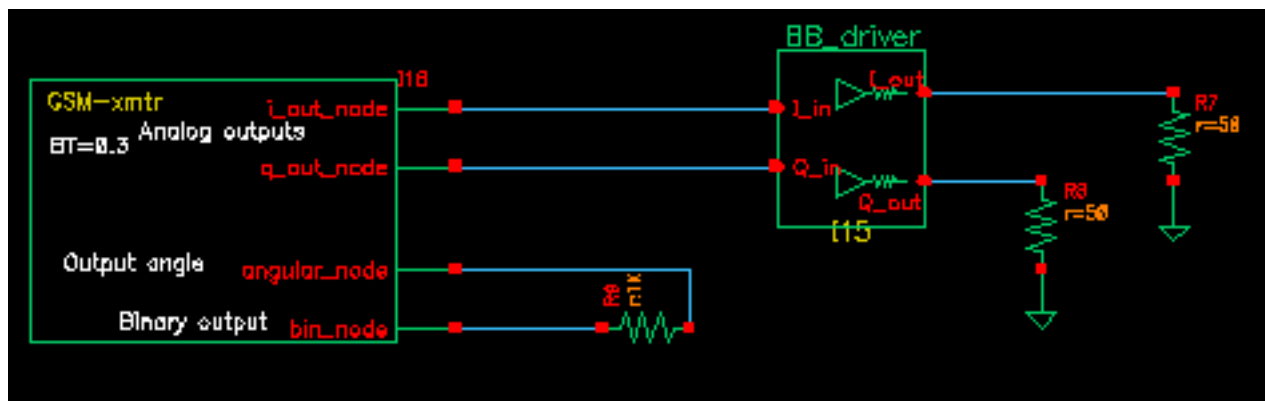
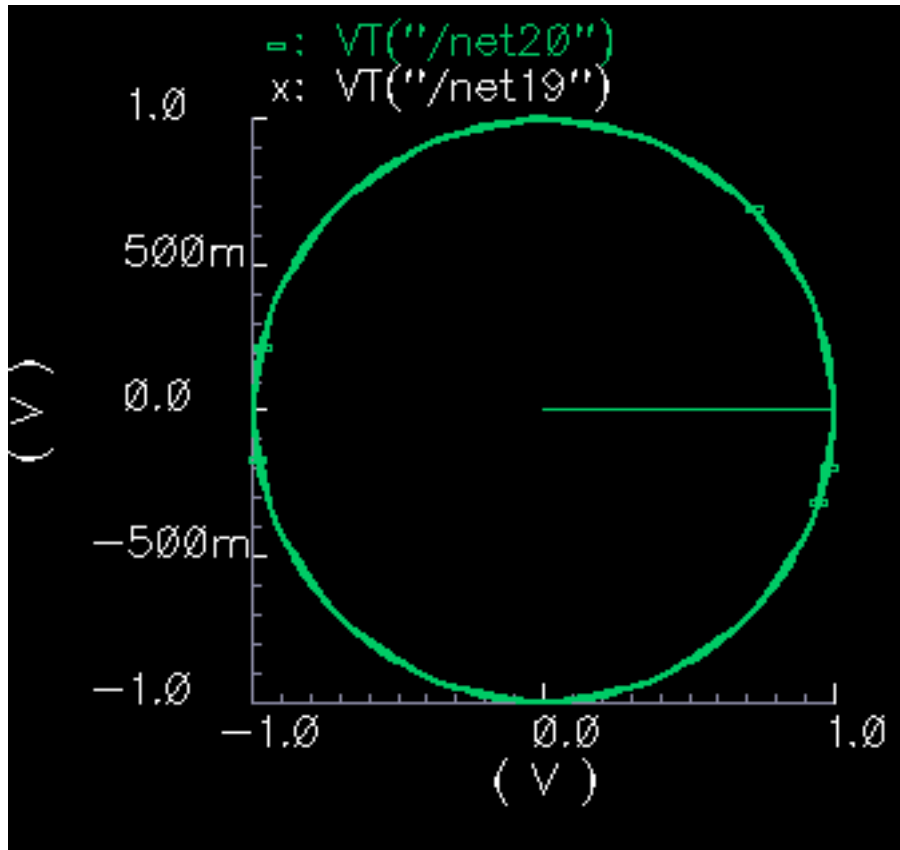


Figure 13-107 Signal Trajectory for the GSM_xmtr



The GSM¹ signal generator starts with a bit sequence of three zeros, 8.25 ones, and three zeros. This pattern models 3 stop bits, 8.25 guard bits², and 3 start bits. The “frame” concludes with 142 random bits. The bit rate is 270833.333 bits per second. Each bit is over sampled at a 4:1 rate then filtered with a Gaussian FIR filter that has 32 taps. The filtered bits modulate the frequency of a VCO. The gain of the VCO equals $0.25 \cdot \text{bit_rate} \cdot \text{oversample_rate}$ Hz/volt.

The voltage on the i_out_node pin equals the amplitude, a user-specified parameter, scaled by the cosine of the VCO phase.

The voltage on the q_out_node pin equals the amplitude scaled by the sine of the VCO phase.

-
1. GSM modulation is 0.3 GMSK [11]
 2. The fractional guard bits are important in eliminating two sharp dips in the power spectral density.

The angular_node pin produces a voltage numerically equal to the phase of the VCO in radians.

The bin_node is a voltage representing the bits that drive the FIR filter.

Aside from the amplitude parameter, all parameters are used as they are in the CDMA_reverse_xmit model.

The FIR filter can be modified through the Model Writer.

pi_over4_dqpsk

Figure 13-108 on page 721 shows the symbol for the pi_over4_dqpsk with the BB_driver. Figure 13-109 on page 722 shows the signal trajectory for the pi_over4_dqpsk.

Figure 13-108 pi_over4_dqpsk generator

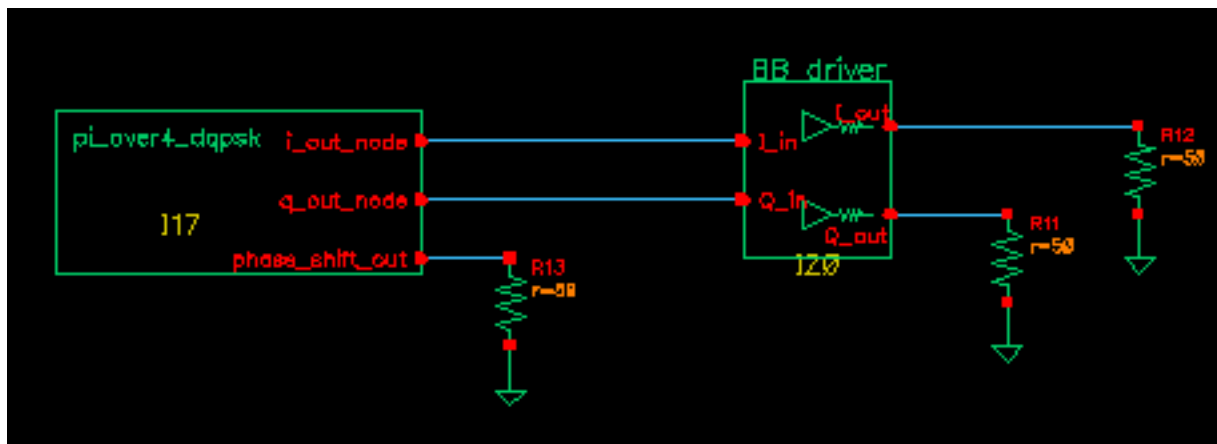
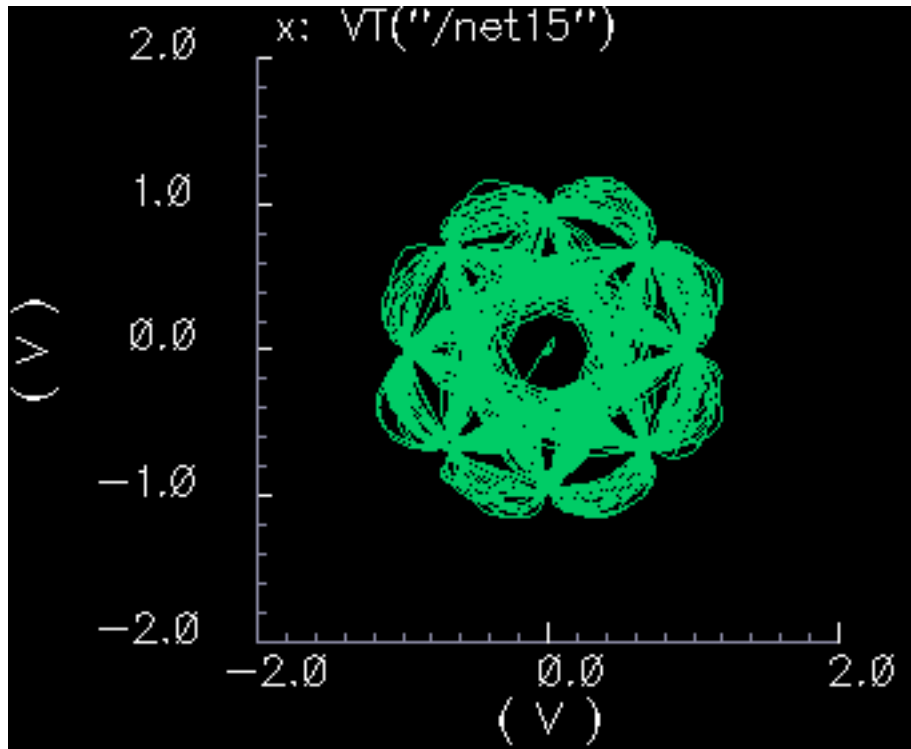


Figure 13-109 Signal Trajectory for the pi_over4_dqpsk



In $\pi/4$ DQPSK modulation, information rides on one of four possible phase shifts in the carrier. The possible phase shifts are ± 45 degrees and ± 135 degrees. The model selects one of the four phase shifts at random then computes I and Q pulses that will drive FIR filters. The I-pulse equals the amplitude parameter, scaled by the cosine of the resulting absolute phase¹. The Q-pulse equals the amplitude parameter, scaled by the sine of the absolute phase. The I and Q pulses drive their own 64-tap FIR filters. The I and Q filters are identical. The filters are raised cosine filters with a 0.35 roll off factor. The rate at which the phase shifts occur is 24300 shifts per second. The FIR filters operate at 8 times that rate.

The voltages on the i_out_node and q_out_nodes pins are the filter outputs.

The phase_shift_out pin is a voltage numerically equal to the phase shift.

Aside from the amplitude parameter, all parameters are used as they are in the CDMA_reverse_xmit model.

The FIR filter can be modified through the Model Writer.

1. Absolute phase equals the sum of all past phase shifts plus the present one. The initial phase is zero.

Affirma RF Simulator (SpectreRF) User Guide

Behavioral Models and Methods for Top Down Design

References

1. M. Jeruchim, P. Balaban, K. Shanmugan, "Simulation of Communication Systems", Plenum Press, 1992.
2. E. Lee, D. Messerschmitt, "Digital Communication", Kluwer Academic Publishers, 1994
3. A. Fitzgerald, C. Kingsley Jr., S. Umans, "Electric Machinery", Fifth edition, McGraw Hill
4. R. Stein and W. Hunt Jr., "Electric Power System Components", Van Nostrand Reinhold Company, 1979.
5. W. Leonhard, "Control of Electrical Drives", Springer-Verlag 1990.
6. P. Kraus and O. Wasynchuk, "Electromechanical Motion Device's", McGraw-Hill Book Company, 1989.
7. P. Vas, "Vector Control of AC Machines", Oxford University Press, 1990
8. G. Kron, "A Short Course in Tensor Analysis for Electrical Engineers", John Wiley and Sons, Inc, 1942.
9. J. Chen, "Rotor Reference Frame Models of a Multiloop 2-phase Motor Drive in Brushless DC and MicroStepping Modes", 1995 Intersociety Energy Conversion Engineering Conference, Orlando, FLA.
10. A. Harrysson, M. Ziren, "System Level Simulations of a Receiver for W-CDMA", Master of Science Thesis, Department of Applied Electronics, Lund Institute of Technology, 12/99.
11. T. Rappaport, "Wireless Communications, Principles and Practice". 1996. Prentice Hall.

Oscillator Noise Analysis

The procedures described in this appendix are deliberately broad and generic. Requirements for your specific design might dictate procedures slightly different from those described here.

You can use this document in the following three ways:

- To learn background theory about phase noise
- To learn how the Spectre®RF circuit simulator noise calculations are related to common phase noise models
- To learn ways to use and troubleshoot SpectreRF phase noise calculations and to answer common questions

These procedures are intended for SpectreRF users who analyze oscillator noise. Users must have a working familiarity with SpectreRF simulation and its operating principles. In particular, readers must understand the SpectreRF PSS and PNOISE analyses. For information about performing these analyses, consult the oscillator chapter in the Spectre online documentation library. Reading the [SpectreRF Theory](#) document in the online library is also recommended.

In RF systems, local oscillator phase noise can limit the final system performance. The SpectreRF circuit simulator lets you rigorously characterize the noise performance of oscillator elements. This appendix explains phase noise, tells how it occurs, and shows how to calculate phase noise using the SpectreRF circuit simulator.

The [“Phase Noise Primer”](#) on page 725 discusses how phase noise occurs and provides a simple illustrative example.

[“Calculating Phase Noise”](#) on page 737 provides some suggestions for successful and efficient analysis of oscillators and discusses the limitations of the simulator.

[“Troubleshooting Phase Noise Calculations”](#) on page 739 explains troubleshooting methods for difficult simulations.

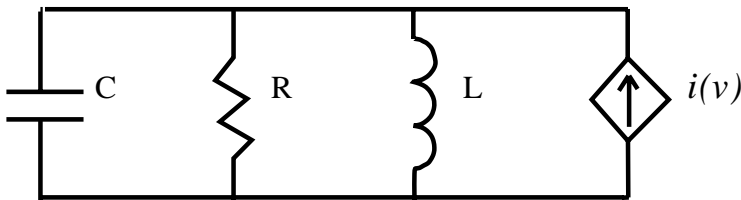
[“Frequently Asked Questions”](#) on page 744 answers some commonly asked questions about phase noise and the SpectreRF circuit simulator.

“[Models for Phase Noise](#)” on page 728 contains mathematical details about how the SpectreRF circuit simulator calculates noise and how these calculations are related to other possible phase noise models. You can skip this section without any loss of continuity, but this section can help you better understand how Spectre simulation calculates phase noise and better appreciate the drawbacks and pitfalls of other simple phase noise models. This section can also help in debugging difficult circuit simulations.

Phase Noise Primer

Consider the simple resonant circuit with a feedback amplifier shown below, a parallel LC circuit with nonlinear transconductance. At small capacitor voltages, the transconductance is negative, and the amplifier is an active device that creates positive feedback to increase the voltage on the capacitor. At larger voltages, where the transconductance term goes into compression, the amplifier effectively acts as a positive resistor (with negative feedback) and limits the capacitor voltage.

Figure A-1 The Simple Resonant Oscillator



A simple model for the nonlinear transconductance is a cubic polynomial. We hypothesize a nonlinear resistor with a current-voltage relation given by

$$i(v) = -\left(\frac{v}{R}\right)(1 - \alpha v^2).$$

The effect of the resistor in parallel with the inductor and the capacitor can be lumped into this transconductance term. The parameter α is a measure of the strength of the nonlinearity in the transconductance relative to the linear part of the total transconductance. Because the signal amplitude grows until the nonlinearity becomes significant, the value of this parameter does not affect the qualitative operation of the circuit. For simplicity, we require that $\alpha=1/3$ for the remainder of this appendix.

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

After some renormalization¹ of variables, the differential equations describing the oscillator can be written in the following form

$$\frac{dv}{dt} = -i + \frac{1}{Q}(1 - \alpha v^2)v + \xi(t)$$

$$\frac{di}{dt} = v$$

In these equations, where v and i give the normalized capacitor voltage and inductor current, respectively, and $\xi(t)$ is a small-signal excitation such as white Gaussian noise. $Q = R/\omega_0 L$ is the quality factor of an RLC circuit made by replacing the nonlinear transconductance by a positive resistance R .

These equations describe the familiar van der Pol oscillator system. This model includes many of the qualitative aspects of oscillator dynamics, yet it is simple enough to analyze in detail. Many more complicated oscillators that operate in a weakly nonlinear mode can be approximated with this model by using the first few terms in the Taylor series expansion of the relevant transconductances.

As a brute-force method of calculating the noise properties of this circuit, the nonlinear stochastic differential equations that describe the current and voltage processes were numerically integrated [kloeden], and the noise power was obtained using a standard FFT-periodogram technique. This technique requires several hundred simulations of the oscillator over many thousands of periods. Consequently, it is not a feasible approach for practical circuits, but it is rigorously correct in its statistical description even though it requires no knowledge of the properties of oscillators, noise, periodicity, or signal amplitudes. [Figure A-2](#) on page 726 shows the total time-average noise in the voltage variable.

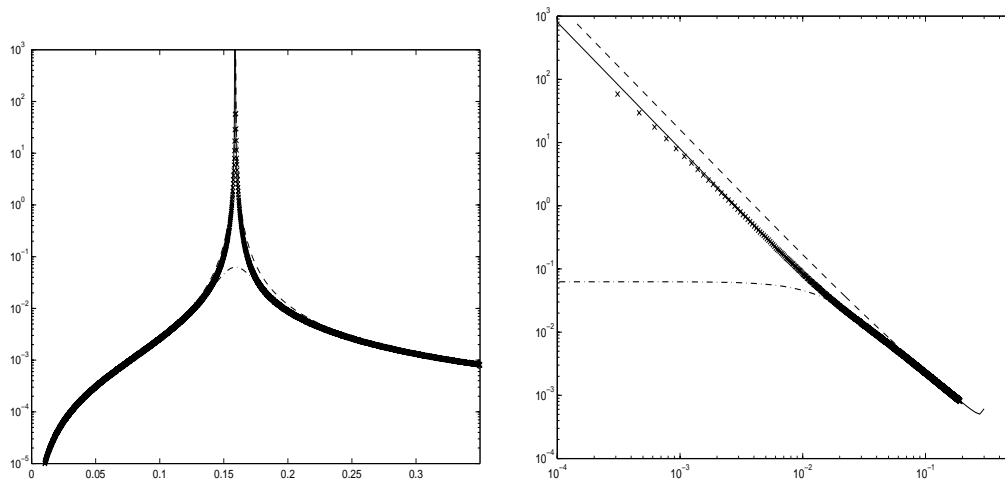
[Figure A-2](#) on page 726 shows noise in the simple Van der Pelt system, plotting power spectral density vs. normalized noise frequency offset, for a $Q = 5$ system. The left half of the figure shows noise as a function of absolute frequency. The left half of the figure shows noise as a function of frequency offset from the oscillator fundamental frequency. The dashed line is LC-filtered white noise, the dash-dot line is RLC-filtered white noise, the solid line is SpectreRF phase noise, and (x) marks are noise power from a full nonlinear stochastic differential equation solution.

Figure A-2 Noise in the Simple Van der Pol System.

1. Time must be scaled by $1/\omega_0$, with $\omega_0 = 1/\sqrt{LC}$, and current scaled by $\sqrt{C/L}$.

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis



The resulting noise power spectral density looks much like the voltage versus current response of a parallel LC circuit. The oscillator in steady state, however, does not look like an LC circuit. As we shall see below, this noise characteristic similarity occurs because both systems have an infinite number of steady state solutions.

The characteristic shape of the small-signal response of an LC circuit results because an excitation at the precise resonant frequency can introduce a drift in the amplitude or phase of the oscillation. The magnitude of this drift grows with time and is potentially unbounded. In the frequency domain, this drift appears as a pole on the imaginary axis at the resonant frequency. The response is unbounded because no restoring force acts to return the amplitude or phase of the oscillation to any previous value, and perturbations can therefore accumulate indefinitely.

Similarly, phase noise exists in a nonlinear oscillator because an autonomous oscillator has no time reference. A solution to the oscillator equations that is shifted in time is still a solution. Noise can induce a time shift in the solution, and this time shift looks like a phase change in the signal (hence the term “phase noise”). Because there is no “resistance” to change in phase, applying a constant white noise source to the signal causes the phase to become increasingly uncertain relative to the original phase. In the frequency domain, this corresponds to the increase of the noise power around the fundamental frequency.

If the noise perturbs the signal in a direction that does not correspond to a time shift, the nonlinear transconductance works to put the oscillator back on the original trajectory. This is similar to AM noise. The signal uncertainty created by the amplitude noise remains bounded and small because of the action of the nonlinear amplifier that created the oscillation. The LC circuit operates differently. It lacks both a time (or phase) reference and an amplitude reference and therefore can exhibit large AM noise.

Another explanation of the similarity between the oscillator and the LC circuit is that both are linear systems that have poles on the imaginary axis at the fundamental frequency ω_0 , that is, at the complex frequencies $s = \pm j\omega_0$. However, the associated transfer functions are not the same. In fact, because of the time-varying nature of the oscillator circuit, multiple transfer functions must be considered in the linear time-varying analysis.

Understanding the qualitative behavior of linear and nonlinear oscillators is the first step towards a complete understanding of oscillator noise behavior. Further understanding requires more quantitative comparisons that are presented in the next section. Readers uninterested in these mathematical details might wish to skip ahead to [“Calculating Phase Noise”](#) on page 737.

Models for Phase Noise

This section considers several possible models for noise in oscillators. In the engineering literature, the most widespread model for phase noise is the Leeson model [leeson]. This heuristic model is based on qualitative arguments about the nature of noise processes in oscillators. It shares some properties with the LC circuit models presented in the previous section. These models fit well with an intuitive understanding of oscillators as resonant RLC circuits with a feedback amplifier. In the simplest treatment, the amplifier is considered to be a negative conductance whose value is chosen to cancel any positive real impedance in the resonant tank circuit. The resulting linear time-invariant noise model is easy to analyze.

Linear Time-Invariant (LTI) Models

To calculate the noise in a parallel RLC configuration, the noise of the resistor is modeled as a parallel current source of power density $S_i(\omega) = 4k_B T/R$. In general, if current noise excites a linear time-invariant system, then the noise power density produced in a voltage variable is given by [gardner].

$$S_v(\omega) = |H(\omega)|^2 S_i(\omega)$$

where $H(\omega)$ is the transfer function of the LTI transformation from the noise current source “input” to the voltage “output.” The transfer function is defined in the standard way to be

$$H(\omega) = \frac{v_0(\omega)}{i_s(\omega)}$$

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

where i_s is a (deterministic) current source and v_o is the measured voltage between the nodes of interest.

It follows that the noise power spectral density of the capacitor voltage in the RLC circuit is, at noise frequency $\omega = \omega_0 + \omega'$ with $\omega' \ll \omega_0$

$$S_v(\omega') = \frac{4k_B TR}{1 + 4(\omega'/\omega_0)^2 Q^2}$$

where $Q = R / \omega_0 L$ is the quality factor of the circuit, R is the parallel resistance (the source of the thermal noise), and ω_0 is the resonant frequency. If a noiseless negative conductance is added to precisely cancel the resistor loss, the noise power for small ω'/ω_0 becomes

$$S_v(\omega') = \frac{k_B TR}{(\omega'/\omega_0)^2 Q^2}$$

This linear time-invariant viewpoint explains some qualitative aspects of phase noise, especially the $(\omega_0/Q \omega')^2$ dependencies. However, even for this simple system, a set of complicating arguments is needed to extract approximately correct noise from the LTI model. In particular, we must explain the 3dB of excess amplitude noise inside the resonant bandwidth generated by an LC model but not by an oscillator (see [“Amplitude Noise and Phase Noise in the “Linear” Model”](#) on page 733). Furthermore, many oscillators, such as relaxation and ring oscillators, do not naturally fit this linear time-invariant model. Most oscillators are better described as time-varying (LTV) circuits because many phenomena, such as upconversion of $1/f$ noise, can only be explained by time-varying models.

Linear Time-Varying (LTV) Models

For linear time-invariant systems, the noise at a frequency ω is directly due to noise sources at that frequency. The relative amplitudes of the noise at the system outputs and the source noise are given by the transfer functions from noise sources to the observation point. Time-varying systems exhibit frequency conversion, however, and each harmonic $k\omega_0$ in the

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

oscillation can transfer noise from a frequency ω_0 to the observation frequency ω . In general, for a stationary noise source $\xi(t)$, the total observed noise voltage will be [gardner]

$$S_v(\omega) = \sum_k |H(\omega)|^2 S_\xi(\omega + k\omega_0)$$

Each term in the series represents conversion of current power density at frequency $\omega + k\omega_0$ to voltage power density at frequency ω with gain $|H(\omega)|^2$. As an example, return again to the van der Pol oscillator with $\alpha = 1/3$ and notice how a simple time-varying linear analysis of noise proceeds.

The first analysis step for the van der Pol oscillator is to obtain a large-signal solution so we set $\xi(t) = 0$. In the large-Q limit, the oscillation is nearly sinusoidal and so it is a good approximation to assume

$$v(t) = a \sin \omega_0 t.$$

The amplitude a and oscillation frequency can be determined from the differential equations that describe the oscillator. Recognizing that

$$i(t) = (a/\omega_0) \cos \omega_0 t$$

and substituting into the equation for dv/dt , a and ω_0 are determined by the equation

$$a\omega_0 \cos t - (1/Q) (a \sin t - (a^3/3) \sin 3t) - (a/\omega_0) \cos t = 0$$

Substituting $\sin 3t = (3 \sin t - \sin 3t)/4$ and using the orthogonality of the sine and cosine functions, it follows that

$$a - a^3/4 = 0$$

$$\omega_0 - 1/\omega_0 = 0$$

(The $\sin 3t$ term is relevant only when we consider higher-order harmonics of the oscillation.) Therefore, to the lowest order of approximation, $a=2$ and $\omega_0=1$.

The only nonlinear term in the van der Pol equations is the current-voltage term, $v^3/3$. This term differentiates the van der Pol oscillator from the LC circuit. The small-signal conductance is the derivative with respect to voltage of the nonlinear current, $-(1 - v^2)/Q$. With $v(t) = 2 \sin t$, the small-signal conductance as a function of time is $(1/Q) (1 - 2 \cos 2t)$. Because there is a nonzero¹, time-varying, small-signal conductance, the PTVL model is different from the LTI

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

LC circuit model. Oscillators are intrinsically time-varying elements because they trade off excessive gain during the low-amplitude part of the cycle with compressive effects during the remainder of the cycle. This effect is therefore a generic property not unique to this example.

To complete the noise analysis, we write the differential equations that the small-signal solution $i_s(t)$, $v_s(t)$ must satisfy,

$$\frac{dv_s}{dt} = -i_s + \frac{1}{Q}(1 - \alpha v^2(t))v_s + \xi(t)$$

$$\frac{di_s}{dt} = v_s$$

From the large signal analysis, $v(t) = 2 \sin t$, and so

$$\frac{dv_s}{dt} = -i_s + \frac{1}{Q}(2 \cos 2t - 1)v_s + \xi(t)$$

$$\frac{di_s}{dt} = v_s$$

The time-varying conductance can mix voltages from a frequency ω to $\omega \pm 2$. For small ω' , if an excitation is applied at a frequency $\omega = 1 + \omega'$, we expect i_s and v_s to have components at $1 + \omega'$ and $-1 + \omega'$ for the equations to balance. (Higher-order terms are again presumed to be small.) Writing

$$i_s(t) = i_+ e^{j(1 + \omega')t} + i_- e^{j(-1 + \omega')t}$$

and substituting into the small-signal equations with $\xi(t) = c + e^{j(1 + \omega')t}$ leads to the following system of equations for i_+ and i_-

-
1. In fact, the time-average conductance is not even zero. However, the time-average power dissipated by the nonlinear current source i_s is zero, a necessary condition for stable, sustained oscillation.

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

$$\begin{bmatrix} 1 - (1 + \omega')^2 + \left(\frac{i}{Q}\right)(1 + \omega') & -\left(\frac{i}{Q}\right)(-1 + \omega') \\ -\left(\frac{i}{Q}\right)(1 + \omega') & 1 - (-1 + \omega')^2 + \left(\frac{i}{Q}\right)(-1 + \omega') \end{bmatrix} \begin{bmatrix} i_+ \\ i_- \end{bmatrix} = \begin{bmatrix} c_+ \\ 0 \end{bmatrix}$$

Solving these equations gives the transfer function from an excitation at frequency $1 + \omega'$ to the small-signal at frequency $1 + \omega'$ that we call $H_0(\omega')$. A similar analysis gives the other significant transfer function, from noise at frequency $-1 + \omega'$ of amplitude c_- to the small-signal response at frequency $1 + \omega'$, that we call $H_{-2}(\omega')$. In the present case, for small ω' ,

$$H_0^2 \cong H_{-2}^2 \cong \frac{R^2}{16Q^2 \left(\frac{\omega'}{\omega_0}\right)^2}$$

For a general van der Pol circuit with a parallel resistor R that generates white current noise, $\xi(t)$, with $S_\xi(\omega) = 4 k_B T/R$,

$$S_v(\omega') = \frac{k_B TR}{2 \left(\frac{\omega'}{\omega_0}\right)^2 Q^2}$$

Note that this is precisely one-half the noise predicted by the LC model.

Additional insight about phase noise can be gained by analyzing the time-domain small-signal response. The small-signal current response is¹

$$i_s(t) = \frac{ie^{i\omega't}(c_- + c_+)}{2\omega'} \sin t$$

1. c_+ and c_- are complex random variables that represent the relative contribution of white noise at separate frequencies. As white noise has no frequency correlations, they have uncorrelated random phase, and thus zero amplitude expectation, and unit variance in amplitude.

Notice that because the large-signal current is $i(t) = 2 \cos t$, and the sine and cosine functions are orthogonal, the total noise for small ω' that we computed is essentially all phase noise.

Amplitude Noise and Phase Noise in the “Linear” Model

Occasional claims are made that in oscillators “half the noise is phase noise and half the noise is amplitude noise.” However, as the simple time-varying analysis above shows, in a physical oscillator the noise process is mostly phase noise for frequencies near the fundamental. It is true that in an LC-circuit half the total noise power corresponds to AM-like modulation, and the other half, to phase modulation. In the literature, the AM part of the noise is sometimes disregarded when quoting the oscillator noise although this is not always the case. (The SpectreRF circuit simulator computes the total noise generated by the circuit; see [“Details of the SpectreRF Calculation”](#) on page 734).

However, a “linear” oscillator does not really exist. Physical oscillators operate with a tradeoff of gain that causes growing signal strength and nonlinear compressive effects that act to limit the signal amplitude. For noise calculation, the oscillator cannot be considered a linear time-invariant system because there are intrinsic nonlinear effects that produce large phase noise but limited amplitude noise. Oscillators are time-varying, and they therefore require a time-varying small-signal analysis.

Arguments that start with stationary white noise and pass it through a linear model in a forward-analysis fashion produce incorrect answers. This is true because they neglect the time-variation of the conductances (and possibly the capacitances) in the circuit. In the simple cases considered here, the conductances vary in time in a special way so as to produce no amplitude noise, only phase noise.

They have that special variation because they result from linearization about an oscillator limit cycle. An oscillator in a limit cycle has a large response to phase perturbations, but not to amplitude perturbations. The amplitude perturbations are limited by the properties of the nonlinear amplifier, but the phase perturbations can persist. Spectre simulation calculates the correct phase noise because it “knows” about the oscillator properties.

Similarly, arguments (for example, the book by Robins) that start with noise power and derive phase noise in a backwards fashion also usually produce incorrect results because they cannot correctly account for frequency correlations in the noise of the oscillator. These frequency correlations are introduced by the time-varying nature of the circuit.

Occasionally, a netlist appears in which a negative resistance precisely cancels a positive resistance to create a pure LC circuit. Because such a circuit has an infinite number of oscillation modes,¹ The SpectreRF circuit simulator cannot correctly calculate the noise because it assumes a unique oscillation. Such a circuit is not physically realizable because adding or subtracting a microscopically small amount of conductance makes the circuit either

go into nonlinear operation (amplifier saturation) or become a damped LC circuit that has a unique final equilibrium point. This equilibrium point is the zero-state solution. Trying to create the negative resistance oscillator is like trying to bias a circuit on a metastable point.

Details of the SpectreRF Calculation

This section contains the mathematical details of how the SpectreRF circuit simulator computes noise in oscillators. Understanding the material in this section can help you troubleshoot and understand difficult oscillator problems.

The analysis Spectre performs is similar to the simple analysis in the section [“Linear Time-Varying \(LTV\) Models”](#) on page 729. Spectre first finds the periodic steady state of the oscillator using the PSS analysis and then linearizes around this trajectory. The resulting time-varying linear system is used to calculate the noise power density. The primary difference between the SpectreRF calculation and the previous analysis is that the basis functions used for the SpectreRF calculation are not a just few sinusoids, but rather a collection of many piecewise polynomials. The use of piecewise polynomials allows Spectre simulation to solve circuits with arbitrary waveforms, including highly nonlinear circuit behavior.

Noise computations are usually performed with a small-signal assumption, but a rigorous small-signal characterization of phase noise is complicated because the variance in the phase of the oscillation grows unbounded over time. From a mathematical viewpoint, an oscillator is an autonomous system of differential equations with a stable limit cycle. An oscillator has phase noise because it is neutrally stable with respect to noise perturbations that move the oscillator in the direction of the limit cycle. Such “phase” perturbations persist with time, whereas transverse fluctuations are damped with a characteristic time inversely proportional to the quality factor of the oscillator.

Further care is necessary because, in general, the two types of excitations (those that create phase slippage and those responsible for time-damped fluctuations) are not strictly those that are parallel or perpendicular, respectively, to the oscillator trajectory, as is sometimes claimed (for example, in [hajimiri]).

However, one must realize that the noise powers at frequencies near the fundamental frequency correspond to correlations between points that are widely separated on the oscillator envelope. In other words, they are long-time signal effects. In fact, asymptotically (at long times), the ratio of the variance of any state variable to its power at the fundamental frequency is unity for any magnitude of the noise excitation. Therefore, in practical cases, we can consider only small deviations in the state variables when describing the phase noise.

-
1. Any amplitude oscillation can exist, depending on the initial conditions, as long as the amplitude is less than the amplifier saturation point.

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

The first step in the noise analysis is to determine the oscillator steady state solution. This is done in the time domain using shooting methods [telichevesky95]. Once the periodic steady-state is obtained, the circuit equations are linearized around that waveform in order to perform the small-signal analysis.

The time-varying linear system describing the small-signal response $v_s(t)$ of the oscillator to a signal $w(t)$ can be written in general form as [telichevesky96, okumura]

$$\left[C(t) \frac{d}{dt} + G(t) \right] v_s \equiv L(t) v_s(t) = w(t)$$

where $C(t)$ and $G(t)$ represent the linear, small-signal, time-varying capacitance and conductance matrices, respectively. These matrices are obtained by linearization about the periodic steady state solution (the limit cycle). To understand the nature of time-varying linear analysis, we introduce the concept of Floquet multipliers.

Suppose $x(t)$ is a solution to the oscillator circuit equations that is periodic with period T . If $x(0)$ is a point on the periodic solution $x_L(t)$, then $x(T) = x(0)$. If $x(0)$ is perturbed slightly off the periodic trajectory, $x(0) = x_L(0) + \delta x$, then $x(T)$ is also perturbed, and in general for small δx ,

$$x(T) - x_L(T) \approx \frac{\partial x(T)}{\partial x(0)} \delta x$$

The Jacobian matrix $\partial x(T) / \partial x(0)$ is called the sensitivity matrix. The SpectreRF circuit simulator uses an implicit representation of this matrix both in the shooting-method that calculates the steady state and in the small-signal analyses. To see how the sensitivity matrix relates to oscillator noise analysis, consider the effect of a perturbation at time $t=0$ several periods later, at $t = nT$. From the above equation,

$$x(nT) - x_L(nT) \approx \left[\frac{\partial x(T)}{\partial x(0)} \right]^n \delta x$$

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

so

$$x(nT) - x_L(nT) \approx \sum_i C_i \lambda_i^n \phi_i$$

where ϕ_i is an eigenvector of the sensitivity matrix. The C_i are the expansion coefficients of δx in the basis of ϕ_i . If ψ_j is a left eigenvector of the sensitivity matrix, then¹

$$C_i = \psi_i^T \delta x$$

Let λ be an eigenvalue of the sensitivity matrix. In the context of linear time-varying systems, the eigenvalues λ are called *Floquet multipliers*. If all the λ have magnitude less than one (corresponding to left-half-plane poles), the perturbation decays with time and the periodic trajectory is stable. If any λ has a magnitude greater than one, the oscillation cannot be linearly stable because small perturbations soon force the system away from the periodic trajectory $x_L(t)$. A stable nonlinear physical oscillator, however, must be neutrally stable with respect to perturbations that move it in the direction of the orbit². This is true because a time-shifted version of the oscillator periodic trajectory still satisfies the oscillator equations. In other words, one of the Floquet multipliers must be equal to unity. This Floquet multiplier is responsible for phase noise in the oscillator. The associated eigenvector determines the nature of the noise.

If $\lambda = e^{i\eta}$ is a Floquet multiplier, then $\eta + i k \omega_0$ is a pole of the time-varying linear system for any integer k . Therefore, because of the unity Floquet multiplier, the time-varying linear system has poles on the imaginary axis at $k \omega_0$. This is very similar to what occurs in a pure LC resonator, and it explains the identical shape of the noise profiles.

Because operator $L(t)$ has poles at the harmonics of the oscillation frequency, numerical calculations of the noise at nearby frequencies become inaccurate if treated in a naive manner [anzill,kaertner92]. To correctly account for the phase noise, the SpectreRF circuit simulator finds and extracts the eigenvector that corresponds to the unity Floquet multiplier. To correctly extract the phase noise component, both the right and left eigenvectors are required. Once these vectors are obtained, the singular (phase noise) contribution to the noise can be extracted. The remaining part of the noise can be obtained using the usual iterative solution techniques [telichevesky96] in a numerically well-conditioned operation.

-
1. A left eigenvector of a matrix is an eigenvector of its transpose.
 2. These are not necessarily perturbation *in* the direction of the orbit because, in general, $y \neq \phi_i$!

In [Figure A-2](#) on page 726, we see that the SpectreRF PTVL analysis correctly predicts the total noise, including the onset of 3dB amplitude noise outside the bandwidth of the resonator. Note that this simulation was conducted at $E\{\xi^2(t)\} = 10^{-3}$, which represents a very high noise level that is several orders of magnitude higher than in actual circuits. The good match of the PTVL models to the full nonlinear simulation shows the validity of the PTVL approximation.

Calculating Phase Noise

The following sections suggest simulation parameters, give you tips for using these parameters, and advise you about checking for accuracy.

Setting Simulator Options

The SpectreRF time-varying small-signal analyses are more powerful than the standard large-signal analyses (DC, TRAN) but, like any precision instrument, they also have greater sensitivity to numerical errors. For many circuits, particularly oscillators, more simulator precision is needed to get good results from the PAC, PXF, and PNOISE calculations than is needed to get good DC or TRAN results.

The small-signal analyses operate by linearizing around the periodic steady state solution. Consequently, the oscillator noise analysis, and the periodic small-signal analyses in general, inherit most of their accuracy properties from the previous PSS simulation. You must be sure the PSS simulation generates a sufficiently accurate linearization (see section [“What Can Go Wrong”](#) on page 740 for a discussion). See also [“Tips about Getting PSS to Converge”](#) on page 738.

[Table 1-1](#) on page 737 recommends simulator options for various classes of circuits.

Table 1-1 Suggested SpectreRF Simulation Parameters.

Circuit	reltol	vabstol	iabstol
Easy	1.0e-4	default	default
Hard-I	1.0e-5	10n	1p
Hard-II	1.0e-6	1n	0.1p
Hard-III	1.0e-7	0.1n	0.1p

“Easy” circuits are low- Q (about $Q < 10$) resonant oscillators, ring oscillators, and weakly-nonlinear relaxation oscillators. Most “textbook” circuits are in this category. “Hard-I” circuits

are most other resonant oscillators; circuits with complicated AGC, load, or bias circuitry; and relaxation or ring oscillators that exhibit moderate to strong nonlinear or “stiff” effects. This is the best general-purpose set of options.

A few particularly difficult circuits might need to be classified as “Hard-II” or even “Hard-III”. Usually these options are used only in a convergence study (see [“How to Tell If the Answer Is Correct”](#) on page 739) or for circuits that previously failed a conversion study using less strict options. Circuits in this category often exhibit some form of unusual behavior (see [“What Can Go Wrong”](#) on page 740). Sometimes this behavior results from circuit properties (for example, some very high-Q crystal oscillators and some very stiff relaxation oscillator circuits). Occasionally, the behavior reflects a design flaw.

Usually setting *method=gear2only* is recommended for the PSS simulation (but see [“What Can Go Wrong”](#) on page 740).

Tips about Getting PSS to Converge

Most circuits can be converged by manipulating the parameters *tstab* and *steadyratio*. Set *tstab* large enough so that the oscillation amplitude increases to near its steady state value and most of the other transients have died out. You can estimate the value of *tstab* either by performing a *tran* analysis, or by performing a PSS analysis itself with the setting *saveinit=yes*. At tighter simulation tolerances, if *steadyratio* is too small, the PSS simulation often does not converge. Setting *steadyratio*=0.1 usually fixes this problem.

For particularly difficult circuits, or for large circuits that make the above procedure excessively time-consuming, you can use a slightly different procedure.

Try the options settings *reltol*=1e-3 and *steadyratio*=1, and run the PSS analysis with a very long *tstab* parameter setting. Relaxing *iabstol* and *vabstol* might also be necessary. Save this solution to a file using the *writefinal* option. This step can usually obtain a low-accuracy PSS solution with an acceptable simulation time. Using a very long *tstab* increases the probability that the simulation converges, and relaxing *reltol* ensures a reasonable simulation time and increases the probability of PSS convergence.

For some circuits, the oscillation might die out before the oscillator builds up a final level, or the circuit might oscillate for a while before returning to a zero-state. Setting *saveinit=yes* lets you view the initial transient waveforms to determine if this problem is occurring. This problem might be caused by difficulty starting the oscillator, or it might be the result of artificial numerical losses introduced by the very large timesteps. This last cause is likely if the PSS options *method* parameter was set to *gear2only*, *gear2*, or *euler*. In these cases, decrease *reltol* or set the *maxstep* parameter to make the simulator use smaller step sizes.

Once the initial PSS simulation is completed, reset the accuracy parameters *reltol*, *vabstol*, and *iabstol* to their preferred final values. Then rerun the PSS simulation using the *readic* option to read in the initial conditions saved from the first, low-accuracy, PSS run. You might need to leave the *steadyratio* setting in the 0.1 to 1 range to achieve convergence. Any value of *steadyratio* less than one or so should be acceptable.

If the circuit contains independent sources used to start the oscillator, set the PSS start time to a large enough value to be sure these sources are all inactive at the start of the simulation.

You need not use a large *tstab* value in this second step. However, varying *tstab* slightly in this second analysis can sometimes help secure convergence.

Some users report that decreasing the maximum allowed timestep sometimes helps convergence. To do this, either decrease the *maxstep* parameter or increase the *maxacfreq* parameter.

How to Tell If the Answer Is Correct

To be sure that only small numerical errors are introduced into the phase noise calculation, you can simulate the oscillator with progressively more stringent accuracy parameters until the change in the calculated noise is less than the desired simulation precision. Such a set of simulations typically starts from the “Easy” parameter set given in [Table 1-1](#) on page 737 and proceeds downward through the table until the calculated noise no longer changes. Remember that generally *all three* tolerances must be lowered to ensure that the discrete approximations used by the Spectre simulator are converging to the continuous solutions of the physical circuit.

Using the final solutions from the previous simulations as an initial estimate for the next PSS simulations can help minimize the total PSS simulation time. Use the *writefinal* parameter in PSS to write out each final PSS solution and the *readic* parameter to read it back in.

Changing the *tstab* parameter can sometimes identify problem circuits or simulations (see [“The tstab Parameter”](#) on page 743 for details).

Troubleshooting Phase Noise Calculations

The SpectreRF circuit simulator calculates noise effectively for most oscillators. However, circuits that are very “stiff”,¹ very nonlinear, or just poorly designed, can occasionally cause

-
1. “Stiff” circuits exhibit dynamics with two or more very different time scales, for example, a relaxation oscillator with a square-wave-like periodic oscillation. Over most of the cycle, the voltages change very slowly, but occasional rapid transitions are present.

the simulator problems. This section describes some of the reasons for the problems, what goes wrong, how to identify problems, and how to fix them.

Reading “[Details of the SpectreRF Calculation](#)” on page 734 is often helpful for troubleshooting particularly difficult circuits.

Known Limitations of the Simulator

Any circuit that does not have a stable periodic steady state cannot be analyzed by the SpectreRF circuit simulator because oscillator noise analysis is performed by linearizing around a waveform that is assumed to be strictly periodic.

For example, oscillators based on IMPATT diodes generate strong subharmonic responses and cannot be properly analyzed with the SpectreRF circuit simulator. As another example, Colpitts oscillators, properly constructed, can be made to exhibit chaotic as well as subharmonic behavior.

Similarly, any circuit with significant large-signal response at tones other than the fundamental and its harmonics might create problems for the simulator. Some types of varactor-diode circuits might fit this category. In addition, some types of AGC circuitry and, on occasion, bias circuitry can create these effects.

The SpectreRF circuit simulator cannot simulate these circuits because simulation of an autonomous circuit with subharmonic or other aperiodic components in the large signal response essentially requires foreknowledge of which frequency components are important. Such foreknowledge requires Fourier analysis of very long transient simulations and cannot be easily automated. Such simulations can be very expensive in any event.

What Can Go Wrong

■ Generic PSS simulation problems

Any difficulties in the underlying PSS analysis affect the phase noise computation. For example, underestimating the oscillator period or failing to start the oscillator properly can cause PSS convergence problems that make running a subsequent PNOISE analysis impossible.

■ Hypersensitive circuits

Occasionally, we see circuits that are extremely sensitive to small parameter changes. Such a circuit was a varactor-tuned VCO that had the varactor bias current, and therefore the oscillation frequency, set by a $1\text{T}\Omega$ resistor. Changing the resistor to $2\text{T}\Omega$, which is a $1\text{e-}12$ relative perturbation in the circuit matrices, changed the oscillation frequency from 125MHz to 101MHz. We believe that such extreme circuit sensitivity results in PSS

simulations that are very imprecise. In particular, the calculated periods have relatively large variations. If precise PSS simulations are impossible, precise noise calculations are also impossible.

In such a case, you must fix the circuit.

■ **Subharmonics or parametric oscillator modulation**

Sometimes bias and AGC circuitry create small-amplitude parasitic oscillations in the large signal waveform. You can identify these oscillations by performing a transient simulation to steady-state and then looking for modulation of the envelope of the oscillation waveform. (For high- Q circuits and/or low-frequency parasitics, this transient simulation might be very lengthy.)

In this case, because the oscillator waveform is not actually periodic, the PSS simulation can only converge to within approximately the amplitude of the parasitic oscillation. If the waveform possesses a parasitic oscillation that changes amplitude, over one period, around 10^{-5} relative to the oscillator envelope, then convergence with $reltol < 10^{-5}$ is probably not possible (assuming *steadyratio* is one or less).

These effects might also appear as a parametric sideband amplification phenomenon (see “[Frequently Asked Questions](#)” on page 744).

■ **Small-signal frequency much higher than fundamental frequency**

The same timesteps are used for both the small-signal analysis and the PSS analysis. If the small-signal frequency is much higher than the fundamental frequency, much smaller timesteps might be required to resolve the small signal accurately than are needed for the large signal. To force Spectre simulation to take sufficiently small timesteps in the PSS simulation, be sure the *maxacfreq* parameter is set correctly.

■ **Wide timestep variation**

Occasionally, in simulations that generate PSS waveforms with timesteps that vary over several orders of magnitude, the linear systems of equations that determine the small-signal response become ill-conditioned. As a result, the noise analysis is inaccurate. Usually this occurs because excessive simulator precision has been requested, for example, nine-digit precision. Using *method=traonly* in the PSS solution is sometimes helpful in eliminating the problem. Another solution is to set *maxstep* to a very small value in the PSS analysis or to specify a very large *maxacfreq* value.

■ **Device model problems**

The noise calculations are usually inaccurate if the device models leave their physically meaningful operating range during the large-signal PSS solution.

Similarly, if the models are discontinuous, or have discontinuous derivatives, the small-signal analysis might be inaccurate.

■ **Problems resolving Floquet multipliers in stiff relaxation oscillators**

Sometimes in very stiff relaxation oscillators, the PSS solution rapidly and easily converges; but the numerically calculated Floquet multiplier associated with the PSS solution is far from unity. Typically, this multiplier is real and has a magnitude much larger than unity. The SpectreRF circuit simulator prints a warning (Message III, below). It is interesting that sometimes the phase noise is quite accurate even with low simulation tolerances. If you have this problem, perform a convergence study (see [“How to Tell If the Answer Is Correct”](#) on page 739).

■ **Problems resolving Floquet multipliers in high-Q resonant circuits**

In a physical oscillator, there is one Floquet multiplier equal to unity. In an infinite-Q linear resonator, however, the multipliers occur in complex conjugate pairs. A very high-Q nonlinear oscillator has another Floquet multiplier on the real axis nearly equal to, but slightly less than, one. In the presence of numerical error, however, these two real Floquet multipliers can appear as a complex-conjugate pair to the simulator. The phase noise is computed using the Floquet vector associated with the unity Floquet multiplier. When the two multipliers appear as a complex pair, the relevant vector is undefined. When Spectre can correctly identify this situation, it prints a warning (Message III, below). The solution is usually to simulate using the next higher accuracy step (see [Table 1-1](#) on page 737) Sometimes varying *tstab* can also help with this problem.

If the circuit is really an infinite-Q resonator (for example, a pure parallel LC circuit), the multipliers always appear as complex conjugate pairs and the noise computations are not accurate close to the fundamental frequency. Such circuits are not physical oscillators, and Spectre is not designed to deal with them; see [“Amplitude Noise and Phase Noise in the “Linear” Model”](#) on page 733 and [“Frequently Asked Questions”](#) on page 744.

Phase Noise Error Messages

SpectreRF originates error messages when it encounters several types of known numerical difficulty. To interpret the error messages produced by the phase noise analysis, you must know the material in the section [“Details of the SpectreRF Calculation”](#) on page 734.

Message I: “The Floquet eigenspace computed by spectre PSS analysis appears to be inaccurate. PNOISE computations may be inaccurate. Consider re-running the simulation with smaller reltol and method=gear2only.”

The eigenvector responsible for phase noise was inaccurately computed, and the PSS simulation tolerances are possibly too loose. Try simulating the circuit at the next higher accuracy setting (see [Table 1-1](#) on page 737) and then compare the calculated noise in the two simulations.

Message II: “The Floquet eigenspace computed by spectre PSS analysis appears to be ill-defined. PNOISE computations may be inaccurate. Consider re-running the simulation with smaller *reltol*, different *tstab(s)*, and *method=gear2only*. Check the circuit for unusual components.”

This can be an accuracy problem, or it can result from an unusual circuit topology or sensitivity. Tighten the accuracy requirements (see [Table 1-1](#) on page 737) as much as possible. If this message appears in all the simulations, the noise is possibly incorrect even if the simulations agree.

Message III: “The Floquet eigenspace computed by spectre PSS analysis appears to be inaccurate and/or the oscillator possesses more than one stable mode of oscillation. PNOISE computations may be inaccurate. Consider re-running the simulation with smaller *reltol*, different *tstab(s)*, and *method=gear2only*.”

All the real Floquet multipliers were well-separated from unity, suggesting that the PSS simulation tolerances might be too loose. Simulate the circuit at the next higher accuracy setting (see [Table 1-1](#) on page 737) and then compare the calculated noise in the two simulations. If the calculated noise does not change, it is probably correct even if this message appears in both simulations.

The *tstab* Parameter

Because SpectreRF performs the PSS calculation in the time domain by using a *shooting method*, an infinite number of possible PSS solutions exist, depending on where the first timepoint of the PSS solution is placed relative to the oscillator phase.

The placement of the first timepoint is determined by the length of the initial transient simulation, which you can control using the *tstab* parameter. If the *tstab* value causes the edges of the periodic window to fall on a point where the periodic oscillator waveform is making very rapid transitions, it will probably be very difficult for PSS to converge. Similarly, the small-signal analyses probably are not very accurate. Avoid such situations. If the start of the PSS waveform falls on a very fast signal transition, you usually need to view the results of further small-signal analyses with some scepticism.

Although a poor choice of the *tstab* parameter value can degrade convergence and accuracy, appropriate use of *tstab* can help to identify problem circuits and to estimate the reliability of their noise computations.

If you perform several PSS and Pnoise computations that differ only in their *tstab* parameter values, the results should be fairly similar, within a relative deviation of the same order of magnitude as the simulator parameter *reltol*. If this is not the case, you might not have set the simulator accuracy parameters tight enough to achieve an accurate solution; and you

need to reset one or more of the parameters *reltol*, *vabstol*, or *iabstol*. The circuit might also be poorly designed and very sensitive to perturbations in its parameters.

If the calculated fundamental period of the oscillator varies with *tstab* even when you set *reltol*, *iabstol*, and *vabstol* to very small (but not vanishingly small) values, the circuit is probably poorly designed, exhibiting anomalous behavior, or both. (see section [“Known Limitations of the Simulator”](#) on page 740).

Frequently Asked Questions

1. Does SpectreRF simulation calculate phase noise, amplitude noise, or both?

SpectreRF simulation computes the total noise of the circuit, both amplitude and phase noise. What the Affirma analog circuit design environment plots as “phase noise” is really the total noise scaled by the power in the fundamental oscillation mode. Close enough to the fundamental frequency, the noise is all phase noise, so what the design environment plots of “phase noise” are really the phase noise as long as it is a good ways above the noise floor.

Some discussions of oscillator noise based on a simple resonator/amplifier description describe the total noise, at small frequency offsets from the fundamental, as being half amplitude noise and half phase noise. In reality, for physical oscillators, near the fundamental nearly all the noise is phase noise. Therefore, these simple models overestimate the total noise by 3dB. For a detailed explanation, see the phase noise theory section, [“Details of the SpectreRF Calculation”](#) on page 734 and the detailed discussion of the van der Pol oscillator, [“Linear Time-Varying \(LTV\) Models”](#) on page 729.

2. I have a circuit that contains an oscillator. Can I simulate the oscillator separately and use the phase noise Spectre calculates as input for a second PSS/PNOISE simulation?

No, at least, not at this time. Oscillators generate noise with correlated spectral sidebands. Currently, SpectreRF simulation output represents only the time-average noise power, not the correlation information, so the noise cannot be input to a simulation that contains time-varying elements that might mix together noise from separate frequencies.

If the second circuit is a linear filter (purely lumped linear time-invariant elements, such as resistors, capacitors, inductors, or a linearization of a nonlinear circuit around a DC operating point) that generates no frequency mixing, then you can use the output of the SpectreRF Pnoise analysis as a *noisefile* for a subsequent NOISE (not Pnoise!) analysis.

3. How accurate are the phase noise calculations? What affects the errors?

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

Initially, we must distinguish between modeling error and simulation (numerical) error. If the device models are only good to 10% then the simulation is only good to 10 percent (or worse). So, for the rest of this note, we discuss numerical error introduced by the approximations in the algorithms.

We must also distinguish between absolute and relative signal frequencies in the noise analysis. When the noise frequency is plotted on an absolute scale, the error is primarily a function of the variance in the calculated fundamental period. This is true because of the singular behavior, in these regions, of the phase noise near a harmonic of the fundamental. To see this behavior, note that for the simple oscillator driven by white noise, the noise power is proportional to the offset from the fundamental frequency,

$$S_v(\omega) \propto \frac{1}{(\omega - \omega_0)^2}$$

If a small error is made in the calculation of ω_0 , the error ΔS_v in the noise is proportional to S / ω_0 , i.e.

$$\Delta S_v(\omega) \propto \frac{\Delta \omega_0}{(\omega - \omega_0)^3}$$

This error can be very large even if the error in ω_0 , $\Delta \omega_0$, is small. However, because of the way SpectreRF simulation extracts out the phase noise, the calculated phase noise, as a function of offset from the fundamental frequency, can be quite accurate even for very small offsets.

Let us now consider how much error is present in the calculated fundamental frequency. Because the numerical error is related to many simulation variables, it is difficult to quantify, without examination, how much is present. However, as a rough approximation, if we define the quantity

$$r = \min \{reltol, iabstol/\max(i), vabstol/\max(v)\}$$

where $\max(i)$ and $\max(v)$ are the maximum values of voltage and current over the PSS period, then, under some assumptions, the error $\delta \omega_0$ in the fundamental ω_0 probably satisfies

$$r \omega_0 < \delta \omega_0 < M r \omega_0$$

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

where M is the number of timesteps taken for the PSS solution. This analysis assumes that *steadyratio* is sufficiently tight, not much more than one, and also that *iabstol* and *vabstol* are sufficiently small.

If a good estimate of the accuracy in the fundamental is required, run the PSS simulation with many different accuracy settings, initial conditions and/or *tstab* values (see “[How to Tell If the Answer Is Correct](#)” on page 739 and “[The tstab Parameter](#)” on page 743). For example, to estimate how much numerical error remains in the calculated fundamental frequency for a given simulation, run the simulation; reduce *reltol*, *iabstol*, and *vabstol* by a factor of 10 to 100; rerun the simulation; and then compare the calculated fundamental frequencies. For the sorts of parameters we recommend for oscillator simulations, four to five digits of precision seems typical. As of version 4.4.2, SpectreRF simulation estimates of the fundamental frequency are unlikely to be accurate to more than six or seven digits. Past that point, round off error and anomalous effects introduced by vastly varying timesteps offset any gains from tightening the various accuracy parameters.

For phase noise calculations, again it is unrealistic to expect relative precision of better than the order of *reltol*. That is, if *reltol* is 10^{-5} and the oscillator fundamental is about 1 GHz, the SpectreRF numerical fuzz for the calculated period is probably about 10KHz. Therefore, when plotted on an absolute frequency scale, the phase noise calculation exhibits substantial variance within about 10KHz of the fundamental.

However, when plotted on a frequency scale *relative* to the fundamental, the phase noise calculation might be more precise for many oscillators. If the circuit is strongly dissipative (that is, low- Q , such as ring oscillators and relaxation oscillators), the phase noise calculation is probably fairly accurate up to very close to the fundamental frequency even with loose simulation tolerance settings. High- Q circuits are more demanding of the simulator and require more stringent simulation tolerances to produce good results. In particular, circuits that use varactor diodes as tuning elements in a high- Q tank circuit appear to cause occasional problems. Small modifications to the netlist (runs with different *tstab* values and minor topology changes) can usually tell you whether (and where) the simulator results are reliable.

Simulation accuracy is determined by how precisely SpectreRF simulation can solve the augmented nonlinear boundary value problem that determines the periodic steady state. The accuracy of the BVP solution is controlled primarily by the simulation variables *reltol*, *iabstol*, *vabstol*, *steadyratio*, and *Iteratio*. Typically, *steadyratio* and *Iteratio* are fixed, so *reltol* is usually the variable of interest.

Accuracy can occasionally be somewhat affected by other variables such as *relref*, *method*, number of timesteps, and *tstab*. Again, the physical properties of the circuit can put limitations on the accuracy.

4. I have a circuit with an oscillator and a sinusoidal source. Can I simulate this circuit with SpectreRF simulation?

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

In general, SpectreRF simulation is not intended to analyze circuits that contain autonomous oscillators and independent periodic sources.

If the circuit contains components that could potentially oscillate autonomously and also independent large-signal sinusoidal sources, SpectreRF simulation works properly only if two conditions are fulfilled. The system must be treated as a driven system, and the coupling from the sinusoidal sources to the oscillator components must be strong enough to lock the oscillator to the independent source frequency. (In different contexts, this is known as "oscillator entrainment" or "phase-locking.") The normal (non-autonomous) PSS and small-signal analyses function normally in these conditions.

If the autonomous and driven portions of the circuit are weakly coupled, the circuit waveform might be more complicated, for example, a two-tone (quasiperiodic) signal with incommensurate frequencies. Even if PSS converges, further small signal analyses (PAC, PXF, PNOISE) almost certainly give the wrong answers.

5. What is the significance of total noise power?

First, you must understand that SpectreRF simulation calculates and measures noise in voltages and currents. The total power in the phase process is unbounded, but the power in the actual state variables is bounded.

Oscillator phase noise is usually characterized by the quantity

$$d(f) = \frac{Sv(f)}{P_1}$$

where P_1 is the power in the fundamental component of the steady state solution and $Sv(f)$ is the power spectral density of a state variable V . For an oscillator with only white noise sources, $L(f)$ has a Lorentzian line shape,

$$L(f) = \frac{1}{\pi} \frac{a}{a^2 + f^2}$$

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

where a is dependent on the circuit and noise sources, and thus the total phase noise power $\int L(f) df = 1$. Because

$$\text{var } v(t) = R_v(t, t) = \int_{-\infty}^{\infty} S_v(f) df$$

we are led to the uncomfortable, but correct, conclusion that the variance in any variable is 100 percent of the RMS value of the variable, *irrespective of circuit properties or the amplitude of the noise sources*.

Physically, this means that if a noise source has been active, since $t = -\infty$, then the voltage variable in question is randomly distributed over its whole trajectory. Therefore, the relative variance is one. Clearly, the variance is not a physically useful characterization of the noise, and the total noise power must be interpreted carefully. What is actually needed is the variance as a function of time, given a fixed reference for the signal in question; or, more often, the rate at which the variance increases from a zero point; or, sometimes, the increment in the variance from cycle to cycle. That is, we want to specify the phase of the oscillator signal at a given time point and to find a statistical characterization of the variances relative to that time. But because of the non-causal nature of the Fourier integral, quantities like the total noise power give us information about the statistical properties of the signal over all time.

6. What's the story with pure linear oscillators (LC circuits)?

Oddly enough, SpectreRF simulation is not set up to do Pnoise analysis on pure LC circuits.

Pure LC circuits are not physically realizable oscillators, and the mathematics that describes them is different from the mathematics that describes physical oscillators. A special option must be added to the code in order for Pnoise to handle "linear oscillators." See "[Models for Phase Noise](#)" on page 728, and, in particular, "[Amplitude Noise and Phase Noise in the "Linear" Model](#)" on page 733. Because the normal NOISE analysis is satisfactory for these circuits and also much faster, it is unlikely that PNOISE will be modified.

7. Why doesn't the SpectreRF model match my linear model?

As we discuss in the section "[Amplitude Noise and Phase Noise in the "Linear" Model](#)" on page 733, the difference between the SpectreRF model (the correct answer) and the linear oscillator model is that in the linear oscillator, both the amplitude and the phase fluctuations can become large. However, in a nonlinear oscillator, the amplitude fluctuations are always bounded, so the noise is half as much, asymptotically.

We emphasize that computing the correct total noise power requires using the time-varying small signal analysis. An oscillator is, after all, a time-varying circuit by definition. Time-invariant analyses, like the “linear oscillator model,” can sometimes be useful, but they can also be misleading and should be avoided.

8. There are funny sidebands/spikes in the oscillator noise analysis. Is this a bug?

Very possibly this is parametric small-signal amplification, a real effect. This sometimes occurs when there is an AGC circuit with a very long time constant modulating the parameters of circuit elements in the oscillator loop. Sidebands in the noise power appear at frequencies offset from the oscillator fundamental by the AGC characteristic frequency.

Similarly, any elements that can create a low-frequency parasitic oscillation, such as a bias inductor resonating with a capacitor in the oscillator loop, can create these sorts of sidebands.

Further Reading

The best references on the subject of phase noise are by Alper Demir and Franz Kaertner.

Alper Demir’s thesis [demir], now a Kluwer book, is a collection of useful thinking about noise.

Kaertner’s papers [kaertner89,kaertner90,kaertner92] contain a reasonably rigorous and fairly mathematical treatment of phase noise calculations.

The book by W. P. Robins [robins], has a lot of engineering-oriented thinking. However, it makes heavy use of LTI models, and much of the discussion about noise cannot be strictly applied to oscillators. As a consequence, you must interpret the results in this book with care.

Hajimiri and Lee’s recent paper [hajimiri] is worth reading, but their analysis is superseded by Kaertner’s.

Other references include [anzill,abidi83,razavi,kurokawa].

References

[kloeden] P. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, 1995.

[leeson] D. Leeson, “A simple model of feedback oscillator noise spectrum,” *Proc. IEEE*, vol. 54, pp. 329–330, 1966.

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

- [gardner] W. A. Gardner, *Introduction to random processes*. McGraw Hill, 1990.
- [hajimiri] A. Hajimiri and T. Lee, "A general theory of phase noise in electrical oscillators," *IEEE J. Sol. State Circuits*, vol. 33, pp. 179–193, 1998.
- [telichevesky95] R. Telichevesky, J. White, and K. Kundert, "Efficient steady-state analysis based on matrix-free krylov-subspace methods," in *Proceedings of 32rd Design Automation Conference*, June 1995.
- [telichevesky96] R. Telichevesky, J. White, and K. Kundert, "Efficient AC and noise analysis of two-tone RF circuits," in *Proceedings of 33rd Design Automation Conference*, June 1996.
- [okumura] M. Okumura, T. Sugaware, and H. Tanimoto, "An efficient small-signal frequency analysis method of nonlinear circuits with two frequency excitations," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 225–235, 1990.
- [anzill] W. Anzill and P. Russer, "A general method to simulate noise in oscillators based on frequency domain techniques," *IEEE Transactions on Microwave Theory and Techniques*, vol. 41, pp. 2256–2263, 1993.
- [kaertner92] F. X. Kärtner, "Noise in oscillating systems," in *Proceedings of the Integrated Nonlinear Microwave and Millimeter Wave Circuits Conference*, 1992.
- [demir] A. Demir, *Analysis and simulation of noise in nonlinear electronic circuits and systems*. PhD thesis, University of California, Berkeley, 1997.
- [kaertner89] F. X. Kaertner, "Determination of the correlation spectrum of oscillators with low noise," *IEEE Trans. Microwave Theory and Techniques*, vol. 37, pp. 90–101, 1989.
- [kaertner90] F. X. Kaertner, "Analysis of white and f-a noise in oscillators," *Int. J. Circuit Theory and Applications*, vol. 18, pp. 485–519, 1990.
- [robins] W. P. Robins, *Phase Noise in Signal Sources*. Institution of Electrical Engineers, 1982.

Affirma RF Simulator (SpectreRF) User Guide

Oscillator Noise Analysis

- [abidi83] A. A. Abidi and R. G. Meyer, "Noise in relaxation oscillators," *IEEE J. Sol. State Circuits*, vol. 18, pp. 794–802, 1983.
- [razavi] B. Razavi, "A study of phase noise in cmos oscillators," *IEEE J. Sol. State Circuits*, vol. 31, pp. 331–343, 1996.
- [kurokawa] K. Kurokawa, "Noise in synchronized oscillators," *IEEE Transactions on Microwave Theory and Techniques*, vol. 16, pp. 234–240, 1968.

Running PSS Analysis Effectively

Periodic steady-state (PSS) analysis is a prerequisite for all small-signal analyses such as the periodic AC (PAC), periodic transfer function (PXF), and periodic noise (Pnoise) analyses provided by SpectreRF simulation. PSS provides a rich set of parameters to help you adapt it to your own applications. For most circuits, PSS converges with the default parameter values. However, for some difficult circuits, changing the values of some parameters is necessary to achieve convergence. This appendix describes methods you can use to remedy nonconvergence. This appendix also tells you how to improve convergence and efficiency using hierarchical PSS runs.

You can use PSS analysis with driven as well as autonomous circuits. Autonomous circuits are usually harder to converge than driven circuits.

This appendix is divided into the following three main sections:

- General convergence aids you can use with both driven and autonomous circuits
- Convergence aids that you use only with autonomous circuits
- Hierarchical PSS runs you can use to improve the convergence, efficiency, and quality of the PSS solution for use in subsequent small-signal analyses

General Convergence Aids

You can use these convergence aids with both driven and autonomous circuits.

The *steadyratio* and *tstab* Parameters

You can converge most difficult circuits by manipulating the *steadyratio* and *tstab* parameters.

The *steadyratio* parameter guards against false convergence. However, in unusual situations, the default value for *steadyratio* might be too conservative (1.0E-3 is the default).

The PSS convergence criteria (for voltage-valued variables) is roughly

Affirma RF Simulator (SpectreRF) User Guide

Running PSS Analysis Effectively

$$|\Delta v| < (\text{reltol} \times |\text{vsig}| + \text{vabstol}) \times \text{steadyratio} \times \text{lteratio}$$

To solve the periodic steady-state problem, SpectreRF simulation replaces the time derivatives in the one-period time interval with discrete differences. This turns the nonlinear, continuous differential equations into a set of discrete nonlinear equations that SpectreRF simulation can solve.

The *steadyratio* parameter specifies the accuracy requirements for the discrete system. The discrete system might have its own solution (a steady state of the discrete difference equations) independent of what is happening in the continuous limit. In other words, the discrete system might be solved to zero tolerance. This happens frequently, particularly in driven circuits which is why setting a conservative *steadyratio* generally works quite well.

In some cases, however, the solution to zero tolerance might not occur. (Oscillators seem to be problematic.) Avoid setting the convergence tolerance too tightly. For example, if *reltol*=1.0E-6 and *steadyratio*=1.0E-3, the relative tolerance for solving the discrete system is approximately *reltol* × *steadyratio*=1.0E-9. This tolerance level approaches the limit of precision the simulator can provide. In this situation, loosen the *steadyratio* to 1.0 or 0.1 and reduce the precautions against false convergence.

Providing a larger value for *tstab* usually improves convergence. Occasionally, you must set *tstab* to a value equal to or greater than the time needed for the circuit to reach approximate steady-state.

Additional Convergence Aids

Below is a list of additional suggestions that sometimes help convergence.

- Carefully evaluate and resolve any notice, warning, or error messages.
- While trapezoidal rule ringing is simply annoying in transient analysis, in PSS analysis it can cause the shooting iteration to stall before convergence is achieved. You can remedy this problem by changing the method from *traponly* to either *trap*, *gear2*, or *gear2only*.
- Increase the maximum iterations for shooting methods using the *maxperiods* parameter to help convergence. Sometimes the analysis simply needs more than the default number of iterations to converge. However, in some situations convergence does not occur regardless of the number of iterations. In these cases, increasing the iteration limit simply causes the simulation to take longer to fail.
- Decrease the maximum allowed time-step to help convergence. To adjust the time step, either decrease the *maxstep* parameter or increase the *maxacfreq* parameter.

- Use the *errpreset* parameter properly. Use *liberal* for digital circuits; use *moderate* for typical analog circuits; use *conservative* for sensitive analog circuits (for example, charge storage circuits).

Convergence Aids for Oscillators

Oscillator circuits are usually harder to converge than their driven counterparts. In addition to manipulating the *steadyratio* parameter, as discussed in [“General Convergence Aids”](#) on page 752, set the *tstab* parameter large enough so that the oscillation amplitude increases almost to its steady-state value and most other transients die out. You can estimate the required value of *tstab* by performing a transient analysis, or from the PSS analysis itself when you use the *saveinit=yes* option.

For some circuits, the oscillation might die out before the oscillator builds up a final value, or the circuit might oscillate temporarily but then return to a zero-state. Setting *saveinit=yes* lets you view the initial transient waveforms to identify the problem. This problem might be due to difficulty in starting the oscillator, or it might be caused by artificial numerical losses introduced by very large time-steps. The latter is particularly likely if you set the PSS options *method* parameter to *gear2only*, *gear2*, or *euler*. In this case, you might try using *method=traponly*. If the problem persists, force the simulator to use smaller step sizes by decreasing *reltol* or by setting the *maxstep* parameter.

With autonomous PSS analysis, exclusive use of the trapezoidal rule can lead to ringing that spans the length of the oscillation period and causes convergence problems. With *method=trap*, SpectreRF simulation occasionally takes a backward Euler step, which acts to damp the ringing. The *gear2* and *gear2only* methods use Gear’s backward difference method, which is not subject to ringing. Each of these alternatives to *traponly* avoids trapezoidal rule ringing and the attendant convergence problems at the expense of adding a small amount of artificial numerical damping. This damping slightly reduces the computed Q of the oscillator.

Be sure that the method you choose to start your oscillator is effective. It must *kick* the oscillator hard enough to start the oscillation and make the oscillator respond with a signal level that is between 25 and 100 percent of the expected final level. Avoid kicking the oscillator so hard that it responds in an unnecessarily nonlinear fashion. Also try to avoid exciting response modes in the circuit that are unrelated to the oscillation, especially those associated with long time constants.

Try to improve your estimate of the period. The simulator uses your estimate of the period to determine the length of the initial transient analysis interval. This interval is used to measure the oscillation period. If you specify a period that is too short, the estimate of the oscillation period is not accurate, and the PSS analysis might fail. Overestimation of the period is not a

Affirma RF Simulator (SpectreRF) User Guide

Running PSS Analysis Effectively

serious problem because the only disadvantage is a longer initial transient interval. However, significant overestimation can result in an excessively long simulation time.

Sometimes the analysis might need more than the default number of iterations (50) to converge. This is more likely to occur with high Q circuits. You can increase the maximum iterations for shooting methods using the *maxperiods* parameter.

Running PSS Analysis Hierarchically

For most circuits, a single PSS analysis run is sufficient to find the periodic steady-state solution. However, for some difficult circuits it is preferable, or even necessary, to have multiple PSS runs to find the steady-state solution with the parameter settings you want (for example, with a tight *reltol*).

The biggest obstacle to PSS convergence is poorly chosen initial conditions. The backbone of PSS analysis is Newton's method. Theoretically, when the initial guess is close enough to the solution and the problem is not ill-conditioned, Newton's method is guaranteed to converge because of its contraction property. Consequently, it is very important to provide the best initial conditions that you can to ensure rapid convergence.

To run PSS hierarchically, you start by running a minimally accurate PSS to obtain, with high likelihood, a coarse-grid solution that you can make converge in an acceptable amount of simulation time. You then use the coarse-grid solution as the initial condition for a PSS run to achieve a fine-grid solution. This practice significantly increases the chance that the fine-grid PSS analysis converges. This method is more efficient as you might expect. It often reduces the total simulation time needed to find a periodic steady-state solution because it reduces the number of time-consuming fine-grid PSS iterations.

The example below shows how to arrange these hierarchical PSS runs. The *writefinal* and *readic* parameters serve as threads to these PSS runs.

The *writefinal* parameter has an interesting role. When *writefinal=SomeFile*, both the associated time and period information (for autonomous PSS) and the final transient solution in steady-state are saved to the file. When you run a second similar PSS analysis, you set *readic=SomeFile* and *skipdc=yes*. Setting *skipdc=yes* is important because it forces the simulator to use the provided IC as the initial transient solution in the first PSS iteration.

For example, if you want to find the periodic steady-state solution with a tight *reltol=1.0E-5*, you can start with a PSS with a looser tolerance, such as *reltol=1.0E-3*, and write out the final results. A PSS analysis runs faster with the looser tolerance because fewer time points are generated in the transient integrations performed during PSS iterations. After the first analysis, you run a second PSS analysis with the tolerance *reltol=1.0E-5* using *readic* and *skipdc*. After the second PSS analysis, you can run small-signal analyses such as PAC.

Affirma RF Simulator (SpectreRF) User Guide

Running PSS Analysis Effectively

```
set1 set reltol=1.0E-3
pss1 pss ... writefinal="SomeFile"
set2 set reltol=1.0E-5
pss2 pss ... readic="SomeFile" skipdc=yes
... pac ... ..
```

If necessary, you can increase the number of PSS analyses in this hierarchical process. You choose the tolerance sequence for the continuation. Always use a sequence of PSS runs when you need a tight tolerance PSS solution. The multiple PSS approach usually produces a periodic steady-state solution that is better for subsequent small-signal analyses.

The above procedure is often called a continuation on the simulation parameter *reltol*. However, you can use continuation with many other simulation parameters. For example, in order to achieve PSS convergence at a high input power that causes nonconvergence, you might gradually increase the input power at an RF port.

Using the psin Component

The procedures described in this appendix are deliberately broad and generic. Requirements for your specific design might dictate procedures slightly different from those described here.

This appendix describes how to use the *psin* component from the `analogLib` library in SpectreRF simulations within the Affirma™ analog circuit design environment.

Independent Resistive Source (*psin*)

The *psin* component, located in the `analogLib` library, is used in all RF circuits for SpectreRF and Spectre S-parameter simulations.

When you netlist *psin* in the Affirma analog circuit design environment using the SpectreS simulator, you can see that *psin* is the *port* component in Spectre simulation. A port is a resistive source that is tied between positive and negative terminals. It is equivalent to a voltage source in series with a resistor, and the reference resistance of the port is the value of the resistor.

psin Capabilities

While the *psin* is generally useful as a stimulus in high-frequency circuits, it also has the following three unique capabilities:

- Defines the ports of the circuit to the S-parameter analysis
- Has an intrinsic noise source that lets the noise analysis directly compute the noise figure of the circuit
- Is the only source for which the amplitude can be specified in terms of power

Terminating the *psin*

Be aware that when you specify the voltage on a *psin*, Spectre assumes that *the psin is properly terminated in its reference resistance*. The specified voltage value is not the

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

voltage on the internal voltage source, which is actually set to twice the value specified on the *psin*. So, if you use a *psin* source to drive an open circuit, the voltage (for *DC*, *transient*, *AC*, and *PAC* signals) is double its specified value. However, you can alternatively specify the amplitude of the sine wave in the *transient* and *PAC* analyses as the power in dBm delivered by the `psin` when terminated with the reference resistance.

The *psin* component Edit Properties form is shown in [Figure C-1](#) on page 759 and in [Figure C-2](#) on page 760.

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

Figure C-1 psin Component Edit Properties Form (see next page also)

Edit Object Properties
OK Cancel Apply Defaults Previous Next Help

Apply To only current instance

Show system user CDF

Browse
Reset Instance Labels Display

Property	Value	Display
Library Name	analogLib	off <input type="checkbox"/>
Cell Name	psin	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	rf	off <input type="checkbox"/>

Add
Delete
Modify

User Property	Master Value	Local Value	Display
lvignore	TRUE		off <input type="checkbox"/>
pwr			off <input type="checkbox"/>

CDF Parameter	Value	Display
Frequency name	frf	off <input type="checkbox"/>
Second frequency name	fund	off <input type="checkbox"/>
Noise file name		off <input type="checkbox"/>
Number of noise/freq pairs	0	off <input type="checkbox"/>
Resistance	50 Ohms	off <input type="checkbox"/>
Port number	1	off <input type="checkbox"/>
DC voltage		off <input type="checkbox"/>
Source type	sine	off <input type="checkbox"/>
Delay time		off <input type="checkbox"/>
Sine DC level		off <input type="checkbox"/>
Amplitude		off <input type="checkbox"/>
Amplitude (dBm)	prf	off <input type="checkbox"/>
Initial phase for Sinusoid		off <input type="checkbox"/>
Frequency	frf Hz	off <input type="checkbox"/>
Amplitude 2		off <input type="checkbox"/>
Amplitude 2 (dBm)	prf	off <input type="checkbox"/>

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

Figure C-2 psin Edit Properties Form, continued

Edit Object Properties		
	OK	Cancel
	Apply	Defaults
	Previous	Next
	Help	
Amplitude 2 (dBm)	<input type="text" value="prf"/>	off <input type="checkbox"/>
Initial phase for Sinusoid 2	<input type="text"/>	off <input type="checkbox"/>
Frequency 2	<input type="text" value="frf + 10M Hz"/>	off <input type="checkbox"/>
FM modulation index	<input type="text"/>	off <input type="checkbox"/>
FM modulation frequency	<input type="text"/>	off <input type="checkbox"/>
AM modulation index	<input type="text"/>	off <input type="checkbox"/>
AM modulation frequency	<input type="text"/>	off <input type="checkbox"/>
AM modulation phase	<input type="text"/>	off <input type="checkbox"/>
Damping factor	<input type="text"/>	off <input type="checkbox"/>
Multiplier	<input type="text"/>	off <input type="checkbox"/>
Temperature coefficient 1	<input type="text"/>	off <input type="checkbox"/>
Temperature coefficient 2	<input type="text"/>	off <input type="checkbox"/>
Nominal temperature	<input type="text"/>	off <input type="checkbox"/>
Noise temperature	<input type="text"/>	off <input type="checkbox"/>
AC magnitude	<input type="text"/>	off <input type="checkbox"/>
AC phase	<input type="text"/>	off <input type="checkbox"/>
XF magnitude	<input type="text"/>	off <input type="checkbox"/>
PAC magnitude	<input type="text"/>	off <input type="checkbox"/>
PAC magnitude (dBm)	<input type="text" value="prf"/>	off <input type="checkbox"/>
PAC phase	<input type="text"/>	off <input type="checkbox"/>
Power (Simulation Unit)	<input type="text"/>	off <input type="checkbox"/>
Frequency	◆ _freq1 ◇ _freq2 ◇ _freq3	off <input type="checkbox"/>
Phase offset	<input type="text" value="0 deg"/>	off <input type="checkbox"/>
Harmonic number	<input type="text"/>	off <input type="checkbox"/>
Reactance	<input type="text" value="0 Ohms"/>	off <input type="checkbox"/>
Power (dBm)	<input type="text" value="prf"/>	off <input type="checkbox"/>
Tone1 Harmonic Index 1	<input type="text" value="0"/>	off <input type="checkbox"/>
Tone2 Harmonic Index 1	<input type="text" value="1"/>	off <input type="checkbox"/>
Tone3 Harmonic Index 1	<input type="text" value="0"/>	off <input type="checkbox"/>
Power Unit	<input type="text" value="dBm"/>	off <input type="checkbox"/>
Power	<input type="text"/>	off <input type="checkbox"/>

Parameter Types for the psin Component

In this list, *psin* parameters are grouped by type rather than by the order in which they appear on the *psin Edit Properties* form. The *psin* parameter types and their respective parameters are shown below:

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

Name Parameters

Frequency name

Second frequency name

psin Instance Parameter

DC voltage

General Waveform Parameters

Source type

Delay time

Sinusoidal Waveform Parameters

Sine DC level

Amplitude

Amplitude (dBm)

Frequency

Initial phase for Sinusoid

Amplitude 2

Frequency 2

Phase for sinusoid 2

Amplitude Modulation Parameters

AM modulation frequency

AM modulation phase

AM modulation index

FM Modulation Parameters

FM modulation frequency

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

FM modulation index

Damping factor

Noise Parameters

Noise file name

Number of noise/freq pairs

Port Parameters

Resistance

Port number

Multiplier

Temperature Effect Parameters

Temperature coefficient 1

Temperature coefficient 2

Nominal temperature (See the following discussion about the effect of temperature parameters on the voltage level.)

Small-Signal Parameters

AC magnitude

AC phase

XF magnitude

PAC magnitude

PAC magnitude (dBm)

PAC phase

Note: The “Additional Notes” section on page [774](#) contains information about active parameters in analyses and terminating the psin.

Name Parameters

Frequency Name and *Second frequency name* let you assign names to fundamental tones. After you save the schematic, the names you assign appear in the *Fundamental Tones* list box on the Choosing Analyses form.

psin Instance Parameter

DC voltage

Sets the DC level of the source for DC analysis. The value must be a real number. If you do not specify the DC value, it is assumed to be the *time* =0 value of the waveform. Default: 0
Units: V

Because all small signal analyses (*ac*, *xf*, and *noise*) use DC analysis results, the *DC voltage* level also affects small-signal analyses. Transient analysis is not affected unless you specify *type*=dc or use dc as a default for the other waveform types.

General Waveform Parameters

Source type

Lets you specify several different waveform shapes and quickly switch between them by changing the value in the *type* field. The typical source types used in SpectreRF analyses are *dc* and *sine*. Possible source type values are dc, pulse, or sine.

If you set the source to *type* = dc, you get only DC values even if you specify a sinusoid later. For *type*=dc, the *dc* and *tc* parameters are active and set the DC level for all analyses. In DC analysis, this setting also determines the DC level generated by the source regardless of what type you specify.

With the setting *type*=sine, you create a sinusoidal waveform whose parameters you can alter. These parameters are described in more detail in the *Sinusoidal Waveform Parameters* section below.

You cannot specify the parameters for *type*=pulse in the *psin Edit Properties* or *Add Component* forms. Source types *pwl* and *exp* are not supported for the analogLib *psin* component.

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

Delay time

The waveform delay time, or the time that the source stays at the DC level before it starts generating waveforms (assuming the source is set to *sine*). The value must be a real number. Default: 0 Units: seconds

Sinusoidal Waveform Parameters

The *psin* component can generate up to two sinusoids simultaneously. They are denoted 1 and 2. You can set the amplitude, frequency and phase for both individually. The amplitude can be set to either a voltage or a power level. You can also specify sinusoidal AM or FM modulation of sinusoid 1.

The first sinusoid is described by the parameters *Amplitude*, *Amplitude dBm*, *Frequency*, *Initial phase for sinusoid*, *AM or FM modulation terms*, and *Damping factor*.

The second sinusoid is described by the parameters *Amplitude2*, *Amplitude 2 (dBm)*, *Frequency 2*, and *Initial phase for sinusoid 2*.

Note: The second sinusoid cannot be modulated.

Sine DC level

Sets the DC level for sinusoidal waveforms in transient analyses. This parameter is used when the sinusoid has a different average level than the one specified for the DC analyses. If not specified, the average value of the sinusoid is the same as that of the DC level of the source. The value must be a real number. Default: dc Units: V

Amplitude

Peak amplitude of the first sinusoidal waveform that you generate. The value must be a real number. Default: 1 Units: V

Remember, when you specify the voltage on a *psin*, you are specifying the voltage *when the psin is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *psin*.

Amplitude (dBm)

Amplitude of the first sinusoidal waveform when specified in dBm (alternative to Amplitude). The value must be a real number. Units: dBm

 **Caution**

Set either Amplitude or Amplitude (dBm). Do not set them both! If you specify both Amplitude (in peak volts) and Amplitude (dBm) in the same source, SpectreRF simulation does not give you any errors or warning messages. It uses Amplitude (in peak volts) and ignores the Amplitude (dBm) field. If you specify Amplitude, verify that the Amplitude (dBm) field is empty, and vice-versa.

Frequency

The frequency of the first sinusoidal waveform (carrier frequency). You typically use unmodulated signals in SpectreRF analyses. The value must be a real number. Default: 0
Units: Hz

Phase for Sinusoid

The phase at the specified delay time. The sinusoidal waveform might start before the given delay time in order to achieve a specified phase and still remain continuous. For example, if you want to generate a cosine wave, set this parameter to 90°. The value must be a real number. Default: 0 Units: degrees

Amplitude Modulation Parameters

The amplitude modulation (double sideband large carrier, or DSB-LC) is defined as

$$v_{AM}(t) = A (1 + m \sin(2\pi f_m t + \phi)) \sin(2\pi f_c t)$$

where

- A is the carrier amplitude (amplitude of sinusoid)
- f_m is the AM modulation frequency
- ϕ is the AM modulation phase
- m is the AM modulation index
- $\sin(2\pi f_c t)$ is the carrier signal

The amplitude modulation parameters affect *only* the first sinusoid generated by *psin*. They have no effect on the second sinusoid.

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

AM modulation frequency

AM modulation frequency for the sinusoidal waveform (f_m in the previous equation). The value must be a real number. Default: 0 Units: Hz

AM modulation phase

AM phase of modulation for the sinusoidal waveform (ϕ in the previous equation). The value must be a real number. Default: 0 Units: degrees

AM modulation index

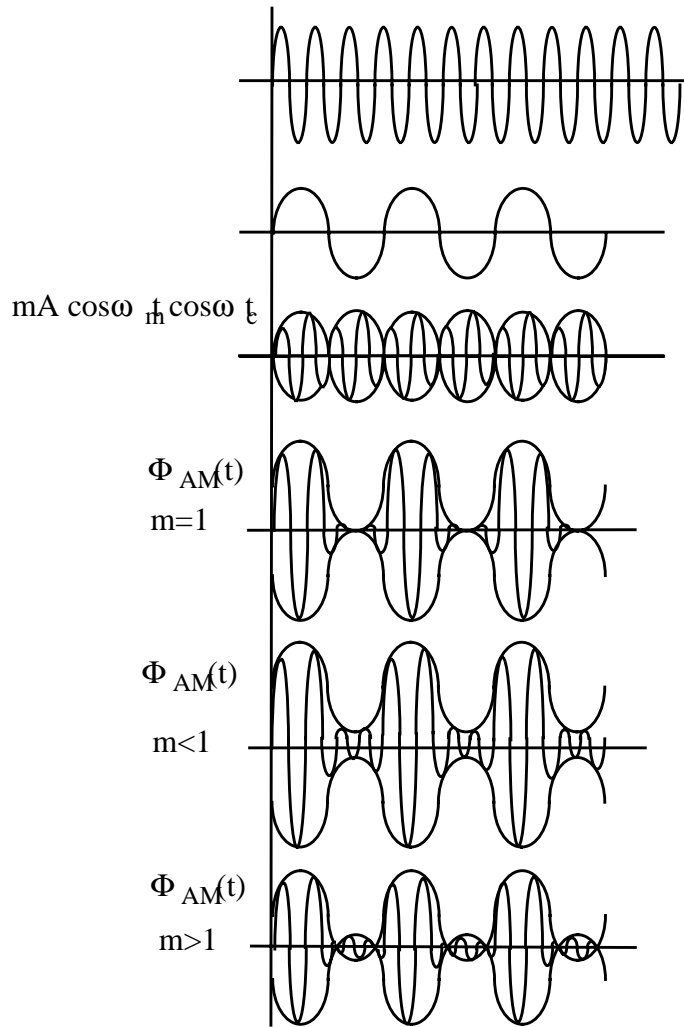
AM index of modulation for the sinusoidal waveform (m in the previous AM equation). The AM modulation index is a dimensionless scale factor used to control the ratio of the sidebands to the carrier.

$$m = (\text{peak DSB-SC amplitude})/(\text{peak carrier amplitude})$$

The value must be a real number. Default: 0

The following figure shows the effect of varying modulation indexes for the following three cases: $m < 1$, $m = 1$, and $m > 1$. f_c is the carrier frequency, and f_m is the modulation frequency.

Figure C-3 Amplitude Modulation: Effects of Varying Modulation Indexes



FM Modulation Parameters

The frequency modulation for the sinusoidal case is defined as

$$v_{FM}(t) = A \sin(2\pi f_c t + \beta \sin(2\pi f_m t) + \phi)$$

where

- A is the amplitude of the sinusoid
- β is the FM modulation index

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

- $\sin(2\pi f_m t)$ is the modulation signal
- f_c is the carrier frequency
- ϕ is the phase of the sinusoid

The frequency modulation parameters affect *only* the first frequency generated by *psin*. They have no effect on the second frequency.

FM modulation frequency

FM modulation frequency for the sinusoidal waveform (f_m in the equation above). The value must be a real number. Default: 0 Units: Hz

FM modulation index

FM index of modulation for the sinusoidal waveform, the ratio of peak frequency deviation divided by the center frequency (β in the above equations).

$$\beta = \Delta f / f_m$$

The value must be a real number. Default: 0

Damping factor

Damping factor for the sinusoidal waveform. *Damping factor* specifies the time it takes to go from the envelope (full amplitude) at *time=0* to 63 percent of the full amplitude. For example, consider the following damped sinusoid:

$$v(t) = A e^{-\sigma t} \sin(2\pi f t + \phi)$$

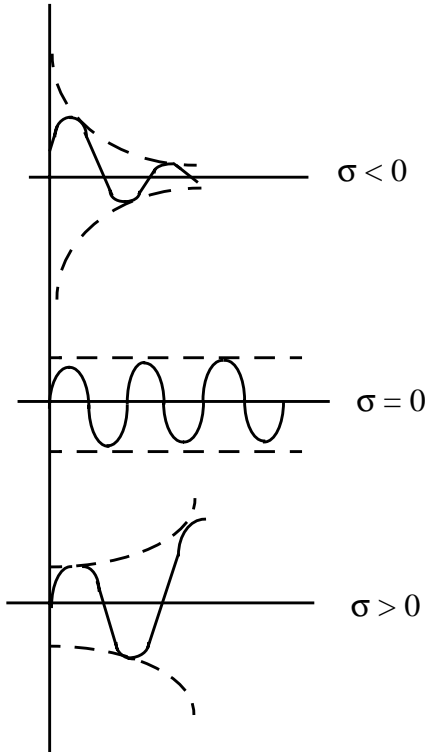
where σ = damping factor.

- If $\sigma = 0$, the waveform is a pure sinusoid (steady state).
- If $\sigma < 0$, the waveform exhibits decaying oscillations.
- If $\sigma > 0$, the waveform exhibits growing oscillations.

It takes 5s to diminish to 1 percent of the peak amplitude. The value must be a real number. Default: 0. Units: 1/seconds

The following figure shows the effect of *Damping factor* on the first sinusoid for three values of σ .

Figure C-4 Effect of *Damping Factor* on the First Sinusoid



Amplitude 2

Peak amplitude of the second sinusoidal waveform. The value must be a real number.
Default: 1 Units: V

Remember, when you specify the voltage on a *psin*, you are specifying the voltage *when the psin is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *psin*.

Amplitude 2 (dBm)

Amplitude of the second sinusoidal waveform in dBm (alternative to *Amplitude 2*). The value specified is the power delivered into a matched load. The value must be a real number. Units: dBm



Caution

Set either *Amplitude2* or *Amplitude2 (dBm)*. Do not set them both!

Frequency 2

Frequency of the second sinusoidal waveform. The value must be a real number. Default: 0
Units: Hz

Phase for Sinusoid 2

The phase at the specified *Delay time* for the second sinusoid. The sinusoidal waveform might start before the given *Delay time* in order to achieve specified phase while still remaining continuous. The value must be a real number. Default: 0 Units: degrees

Noise Parameters

Noise parameters include *Noise file name*, *Number of Noise Frequency Pairs*, and *Noise temperature*.

Noise file name

Name of the file containing the excess spot noise data in the form of frequency-noise pairs. The value must be a string. Default: no value

Number of noise/freq. pairs

You must specify the number of noise-frequency pairs that exist in your noise file in the form. In your file, list the noise-frequency pairs as one pair per line with a space or tab between the frequency and noise values. The values are given as a vector of real number pairs, where noise is given in V^2/Hz , and frequency is given in Hz.

Specific to Affirma analog circuit design environment netlisting: If you have more noise-frequency pairs than the number you specify in the *Number of noise frequency pairs* field, the number of noise-frequency pairs used is the number you specify in the field. Any additional noise-frequency pairs in the noise file are ignored. If the number of noise-frequency pairs in the noise file is smaller than the number you specified in the *number of noise frequency pairs* field, the analysis is stopped.

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

Noise temperature

The *Noise temperature* of the *psin*. If not specified, the *Noise temperature* is assumed to be the actual temperature of the *psin*. When you compute the noise figure of a circuit driven at its input by a *psin*, set the noise temperature of the *psin* (Spectre parameter *noisetemp*) to 16.85C (290K). This setting matches the standard IEEE definition of noise figure. In addition, disable all other sources of noise in the *psin*, such as the Spectre parameters *noisefile* and *noisevec*. If you want a noiseless *psin*, set the *noise temperature* to absolute zero or below, and do not specify a noise file or noise vector. Default: Units: °C.

Port Parameters

Port parameters include *Resistance*, *Port Number*, and *Multiplier*.

Resistance

The reference resistance of the system. The value must be a real number, but not 0. Default: 50 Units: Ω

Port number

The number of the port. The value must be a nonzero integer. Each *psin* in a schematic must have a different port number. The *Port number* is not automatically indexed when you place each *psin* on your schematic.

Multiplier

The multiplicity factor. The value must be an integer number greater than zero. This number lets you specify a number of *psins* in parallel. For example, if you set *Resistance* to 50 and *Multiplier* to 2, you specify two *psin* ports in parallel, each with an effective reference resistance of 25 Ω. Default: 1

Temperature Effect Parameters

How Temperature Parameters Affect the Voltage Level (Background Information)

The value of the DC voltage can vary as a function of the temperature if you specify *tc1* and *tc2*. The variation is given by

$$V_{DC}(T) = dc * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]$$

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

where T is the analysis temperature specified in the analysis options, $tnom$ is the nominal temperature specified in the *Choosing Analyses* form, and dc is the DC voltage.

If the analysis temperature equals the nominal temperature, the result is the voltage amplitude that you specified, $V_{DC}(T) = dc$.

If the nominal and analysis temperatures differ, the voltage amplitude is given by

$$V_{DC}(T) = dc * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]$$

where T is the analysis temperature you specify in the analysis options and $tnom$ is the nominal temperature. $tc1$ and $tc2$ are the linear and quadratic temperature coefficients.

For example, if the nominal temperature is 27°C and the analysis temperature is 25°C, there is a 2° difference between the nominal and analysis temperature. The voltage amplitude is

$$V_{DC}(T) = dc * [1 + tc1 * (-2) + tc2 * (-2)^2]$$

Temperature effect parameters include *Temperature coefficient 1* (the linear temperature coefficient), *Temperature coefficient 2* (the quadratic temperature coefficient), and *Nominal temperature*.

Temperature coefficient 1

First order (linear) temperature coefficient of the *DC voltage* ($tc1$). The value must be a real number. Default: 0 Units: °C⁻¹

Temperature coefficient 2

Second order (quadratic) temperature coefficient of the *DC voltage* ($tc2$). The value must be a real number. Default: 0 Units: °C⁻²

Nominal temperature

The nominal temperature for *DC voltage* ($tnom$). The value must be a real number. Default: Set by options specifications Units: c

Small-Signal Parameters

The small signal parameters are *AC magnitude*, *AC phase*, *XF magnitude*, *PAC magnitude*, *PAC magnitude (dBm)*, and *PAC phase*.

Affirma RF Simulator (SpectreRF) User Guide

Using the psin Component

Remember, when you specify the voltage on a *psin*, you are specifying the voltage *when the psin is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *psin*. The same is true for the values for the *transient*, *AC*, and *PAC* signals. However, the amplitude of the sine wave in the *PAC* and *transient* analysis can alternatively be specified as the power in dBm delivered by the *psin* *when terminated with the reference resistance*

AC magnitude

The peak small-signal voltage. The value must be a real number. Default: 0 Units: v

AC phase

The small-signal phase. The value must be a real number. Default: 0 Units: degrees

Typically, only one source in the circuit has *AC Magnitude* set to a value other than zero, and usually it has an *AC magnitude*=1 and *AC phase*=0. However, there are situations where more than one source has a nonzero *AC magnitude*. For example, applying a differential small-signal input could be done with two sources with the *AC magnitudes* set to 0.5 and the *AC phases* set to 0 and 180.

XF magnitude

The transfer function analysis magnitude. Use *XF magnitude* to compensate for gain or loss in the test fixture. The value must be a real number. Default: 1 Units: v/v

PAC magnitude

The peak periodic *AC* analysis magnitude. Setting this value to unity is a convenient way of computing the transfer function from this source to the output. The value must be a real number. Default: 0 Units: v

PAC magnitude (dBm)

The periodic *AC* analysis magnitude in dBm (alternative to *PAC magnitude*). The value must be a real number. Units: dBm



Caution

Set either PAC magnitude or PAC magnitude (dBm), but do not set them both!

PAC phase

The periodic AC analysis phase. The value must be a real number. Default: 0 Units: degrees

Typically, only one source in the circuit has a *PAC magnitude* set to a value other than zero, and usually it has a *PAC magnitude*=1 and *PAC phase*=0. However, there are situations where more than one source has a nonzero *PAC magnitude*. For example, applying a differential small-signal input could be done with two sources with the *PAC magnitudes* set to 0.5 and the *PAC phases* set to 0 and 180."

You do not specify the *PAC* frequency in the *psin Edit Object Properties* form. Instead, you set the *PAC frequency* in the *PAC Choosing Analysis* form. For example, when making an IP3 measurement, you set the PAC frequency to a variable value in the *Choosing Analysis Form*. Then, you enter the same variable in the *PAC Amplitude (or PAC Amplitude dBm)* field of the *psin Edit Object Properties* form.

Additional Notes

Active Parameters in Analyses

In DC analyses, the only active parameters are *dc*, *m*, and the temperature coefficient parameters.

In AC analyses, the only active parameters are *m*, *mag* and *phase*.

In Transient analyses, all parameters are active except the small-signal parameters and the noise parameters.

In PAC analyses, the only active parameters are *m*, *PAC magnitude* (amplitude or dBm), and *PAC phase*.

XF magnitude is active in *XF* and *PXF* analyses only.

The RF Library

The RF Library, *rfLib*, is divided into four categories.

- Models for top-down design, *top_dwnRF*. It contains all of the elements of the original *rfLib*. The models in *top_dwnRF* are the only models in *rfLib* designed specifically to work with SpectreRF. These models do not have hidden state.
- Models for bottom-up design, *bot_upBB*. It currently contains only J-models.
- Models used only for testing, *test_bench*. You would not actually build any of them. Currently *test_bench* contains a CDMA signal source, a GSM signal source, a P/4-DQPSK signal source, and an eye-diagram generator.
- Models that are *unclassified*. This category contains files generated by the Testbench models.

This appendix explains the models contained in each category.

Top-Down Design Elements

If you are an RF circuit designer who creates top-down designs, you must model functional RF blocks at the behavioral level. These high-level models can be building-blocks for complex RF systems or executable specifications. When distributed in source code format, these models also help you write libraries that reflect your own design needs.

The Verilog-A lets you use the new RF library models in the Spectre®RF circuit simulator without an explicit equivalent circuit. This approach uses the high-level description language capability of SpectreRF to create high-level models that capture the essential information of RF functional blocks. You can then insert these models into regular RF circuits for simulation. The essential parameters are translated into coefficients and equations that describe the relations between the voltages and currents at the connecting nodes.

You can characterize most RF circuits with three sets of parameters:

- Linear Parameters

Affirma RF Simulator (SpectreRF) User Guide

The RF Library

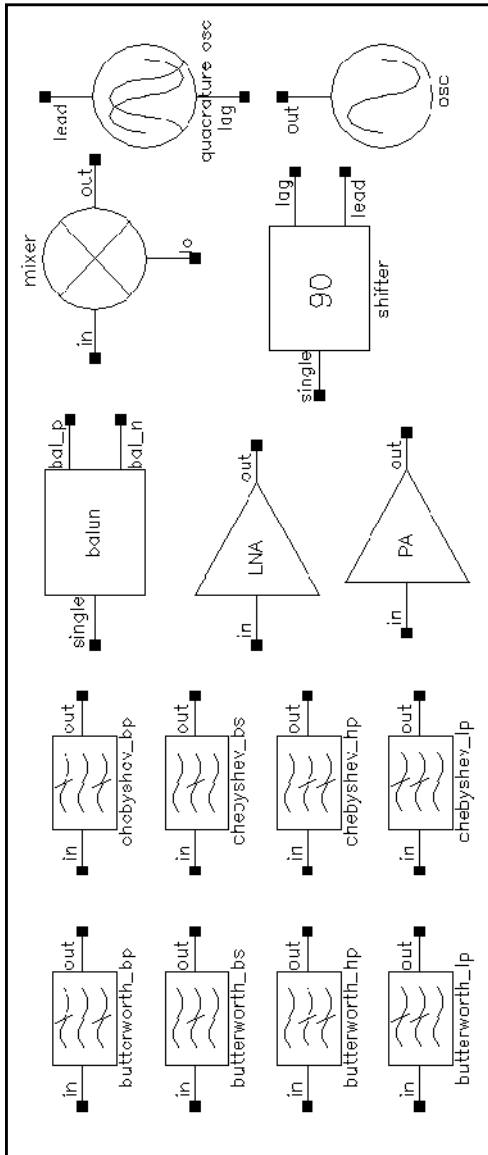
- Nonlinear Parameters
- Noise Parameters.

The RF HDL library contains these seven models, discussed in the sections that follow.

- Filter
- Balun
- Low Noise Amplifier
- Mixer
- Power Amplifier
- Oscillator
- Quadrature Signal Generator
- Phase Shifter

Figure D-1 shows the symbols used for the components that are available in this RF library.

Figure D-1 RF Library Components: top_downRF



Filter

Filter properties are specified in the frequency domain, but it is not easy for the SpectreRF circuit simulator to process frequency-domain data. SpectreRF simulation requires a large signal, time-domain model to simulate the filter behavior.

As part of the RF Verilog-A library, filters are implemented using a network synthesis technique. The synthesis technique consists of the following two steps:

Affirma RF Simulator (SpectreRF) User Guide

The RF Library

1. Calculate the normalized low-pass filter prototype, which consists of serial inductors and parallel capacitors
2. Perform frequency transformation and scaling to synthesize the frequency responses of the filter type

The synthesized model contains many inductors and capacitors. They are implemented using the integral and differential functions of the Verilog-A language, and insertion loss is added using the S-parameter network technique. This network essentially dampens the signal flow by the specified insertion loss value.

Low-pass, high-pass, bandpass, and bandstop filters are implemented, and each of these can be a Butterworth or Chebyshev type filter (for a total of eight combinations). In the current implementation of the Verilog-A language, the order and internal states of the filter cannot be dynamically allocated. You must use the `define` derivative in the source code of the Verilog-A file to specify the order. Use S-parameters to test the filters because S-parameters capture the input/output impedance matching.

For example, the Butterworth bandpass filter has the following module declaration:

```
module butterworth_bp(t1, t2);
    inout t1, t2;
    electrical in, out;
    parameter real r1 = 50 from (0:inf);
    parameter real r2 = 50 from (0:inf);
    parameter real f0 = 1e9 from (0:inf);
    parameter real bw = 0.10 from (0:0.5);
    parameter real fc = 1e9 from (0:inf);
    parameter real loss = 0 from [0:inf);
```

where `t1` and `t2` are the input and output nodes, respectively.

The parameters are

<code>r1</code>	Input impedance [Ω]
<code>r2</code>	Output impedance [Ω]
<code>fc</code>	Corner frequency (3dB point) for low-pass and high-pass filter [Hz]
<code>f0</code>	Center frequency for bandpass or bandstop filter [Hz]
<code>bw</code>	Relative frequency for bandpass or bandstop filter [Hz]
<code>loss</code>	Insertion loss [dB]

Figure D-2 is the simple schematic that tests the filter. Two ports are used to obtain the S-parameters.

Figure D-2 Schematic for Testing the Filter Model

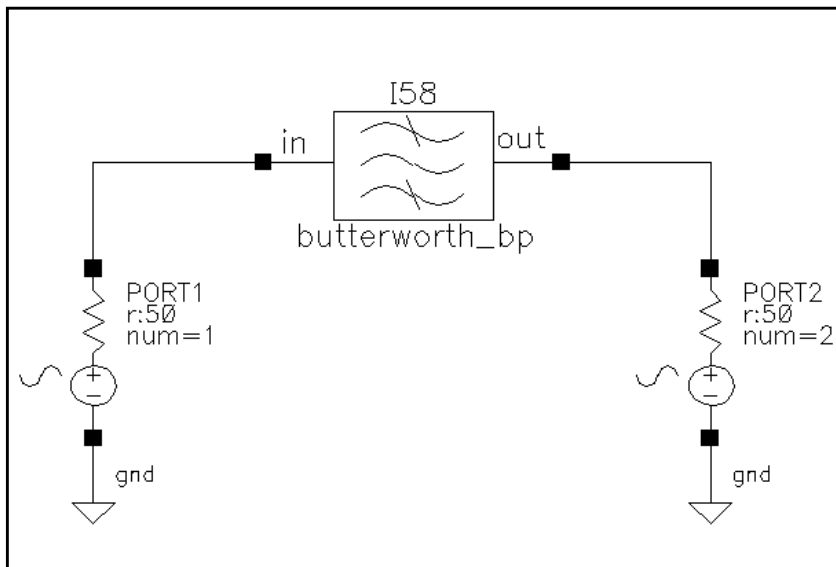
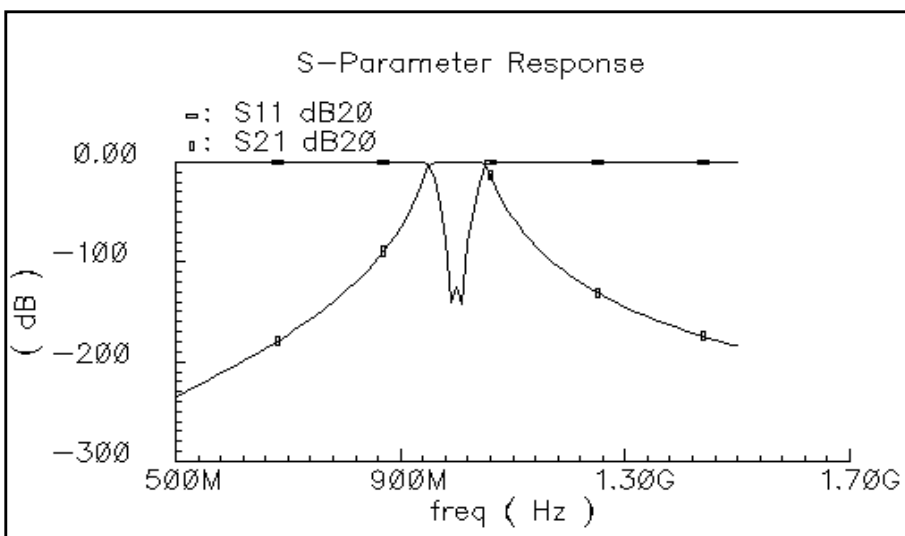


Figure D-3 shows the calculated S-parameters of this Butterworth bandpass filter, which has a center frequency of 1GHz and a relative bandwidth of 10 percent. The order of this specific filter is 10.

Figure D-3 S-Parameters of a Butterworth Filter



Balun

Balun is used in circuits that require single/differential signal transformation. Although in reality, a passive network (including the transformer) is used to achieve balun, this implementation employs a three-port network. There are three ports (or nodes, because the reference nodes are always at the global ground): *single*, *bal_p*, *bal_n*. Parameters include the input impedance (for single end), the output impedance (for balanced end to ground), and the insertion loss (from single end to balanced end and vice versa). If the ports are numbered as *single*(1), *bal_p*(2), and *bal_n*(3), the S-parameter for the three-port network is

$$S = \begin{bmatrix} 0 & t & -t \\ t & 0 & 0 \\ -t & 0 & 0 \end{bmatrix}$$

where $t = \frac{10^{-loss/10}}{\sqrt{2}}$ when loss is specified in dB.

This module can also be used in common mode cancellation applications.

The module is declared as follows:

```
module balun(single, bal_p, bal_n);
  inout single, bal_p, bal_n;
  electrical single, bal_p, bal_n;
  parameter real rin = 50 from (0:inf);
  parameter real rout = 50 from (0:inf);
  parameter real loss = 0 from (-inf:0];
```

The parameters are

<code>rin</code>	Input impedance [Ω]
<code>rout</code>	Output impedance [Ω]
<code>loss</code>	Insertion loss [dB]

Low Noise Amplifier

Low-noise amplifiers are commonly used in the receiver design to amplify the signal with a low-noise figure. A typical low-noise amplifier(LNA) has three sets of parameters: linear model, nonlinear model and noise model.

The definition of the model in the Verilog-A language is as follows:

```
module lna(in, out);
```

Affirma RF Simulator (SpectreRF) User Guide

The RF Library

```
inout in, out;
electrical in, out;
parameter real nf = 2 from [0:inf);
parameter real ip3 = -10;
parameter real gain = 15 from [0:inf);
parameter real isolation = 200 from (0:inf);
parameter real rin = 50 from (0:inf);
parameter real cin = 0 from [0:100];
parameter real rout = 50 from (0:inf);
parameter real cout = 0 from [0:100];
parameter real gammain = -150 from (-inf:0);

parameter real mismatch = 1 from [-1:1] exclude (-1:1);
parameter real gammaout=-150 from (-inf:0);
```

The parameters are

nf	Noise figure [dB]
ip3	Input referenced IP3 [dBm]
gain	S21 (power gain) [dB]
isolation	S12 [dB]
rin	Reference impedance of the input port[Ω]
rout	Reference impedance of the output port[Ω]
gammain	Input return loss [dB]
mismatch	Mismatch sign of input. 1: input impedance > reference impedance -1: otherwise
gammaout	Output return loss [dB]
cin	Parasitic input capacitance [pF]
cout	Parasitic parallel output capacitance [pF]

Internally, a set of linear equations is constructed to satisfy the S-parameters. Furthermore, nonlinearity, expressed by a third-order polynomial function, is added to the gain (or S21) to describe the IP3. Excess white noise is added at the input port to describe the noise figure.

IP3 is the measure of the corruption of signals because of the third-order intermodulation of two nearby tones as shown in [Figure D-4](#). You measure this parameter using a two-tone test.

Avoid the measurement of IP3 by a single tone test in the Affirma analog circuit design environment.

Figure D-4 Intermodulation of Two Nearby Signals

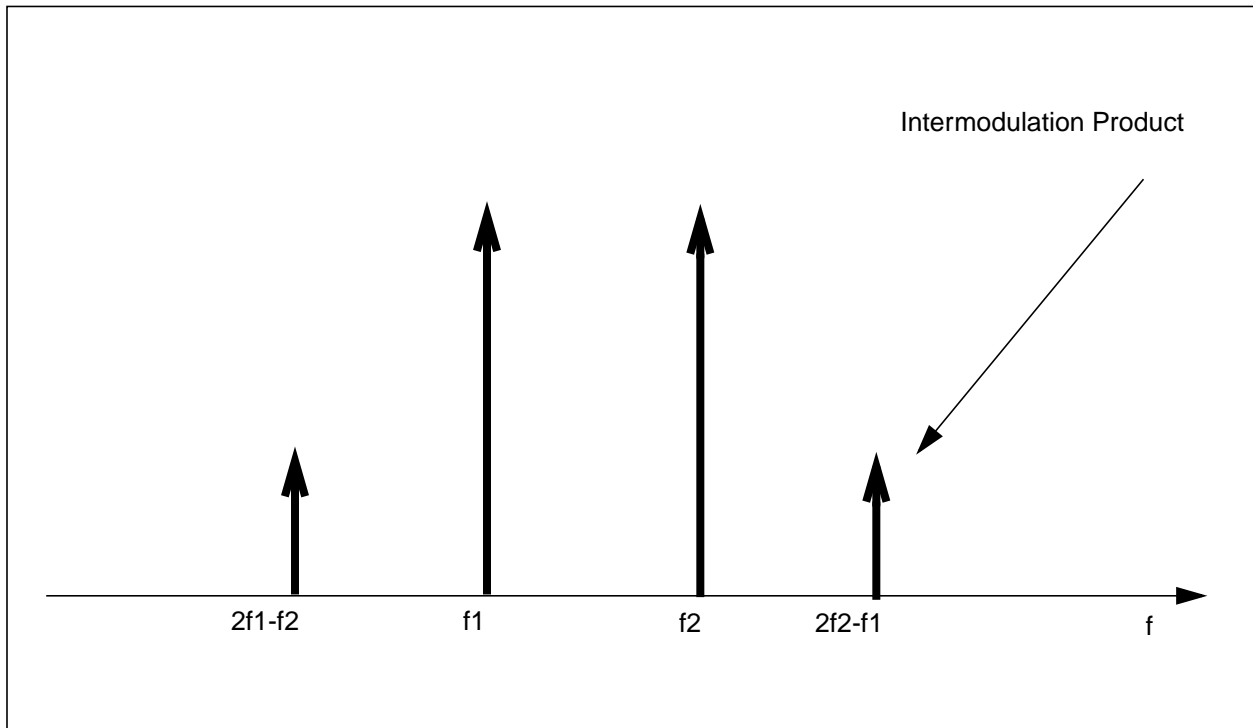
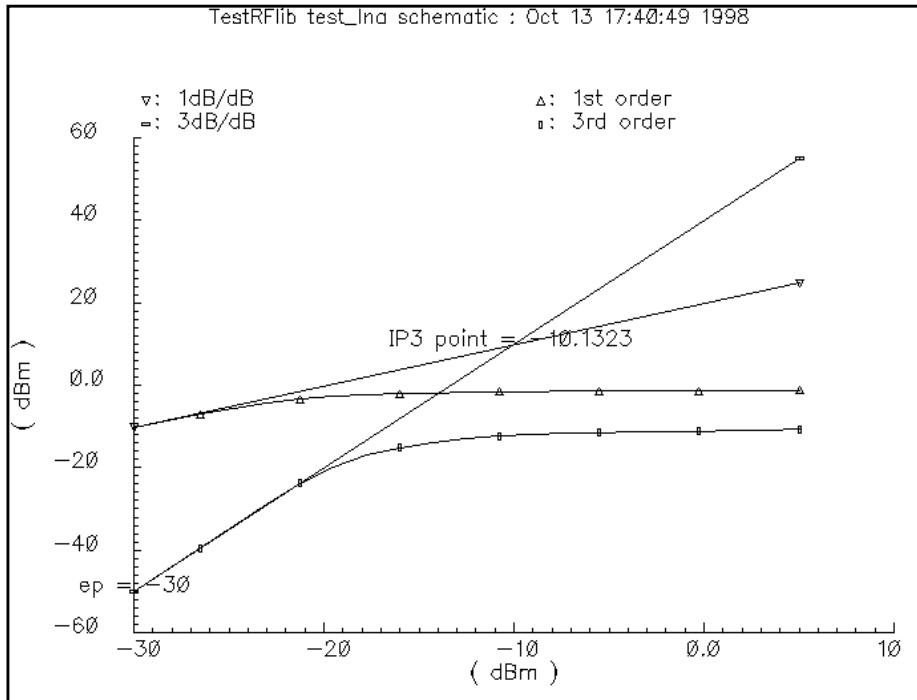


Figure D-5 shows the captured IP3 when the requested value of IP3 is -10dBm.

Figure D-5 IP3 from SpectreRF Simulation



Mixer

Mixers are very important in frequency translation in RF circuits. A typical mixer has the following three sets of parameters:

- Time-varying linear model
- Nonlinear model
- Noise model

This RF library model describes the typical behavior of integrated mixers. The LO switches the input signal on and off. Input LO power beyond the specified limit is effectively clipped off.

The definition of the model in the Verilog-A language is as follows:

```

module mixer(in, lo, out);
    electrical in, lo, out;
    parameter real gain = 10 from [-50:50];
    parameter real plo = 10 from [-100:100];
    parameter real rin = 50 from (0:inf);
    parameter real rout = 200 from (0:inf);

```

Affirma RF Simulator (SpectreRF) User Guide

The RF Library

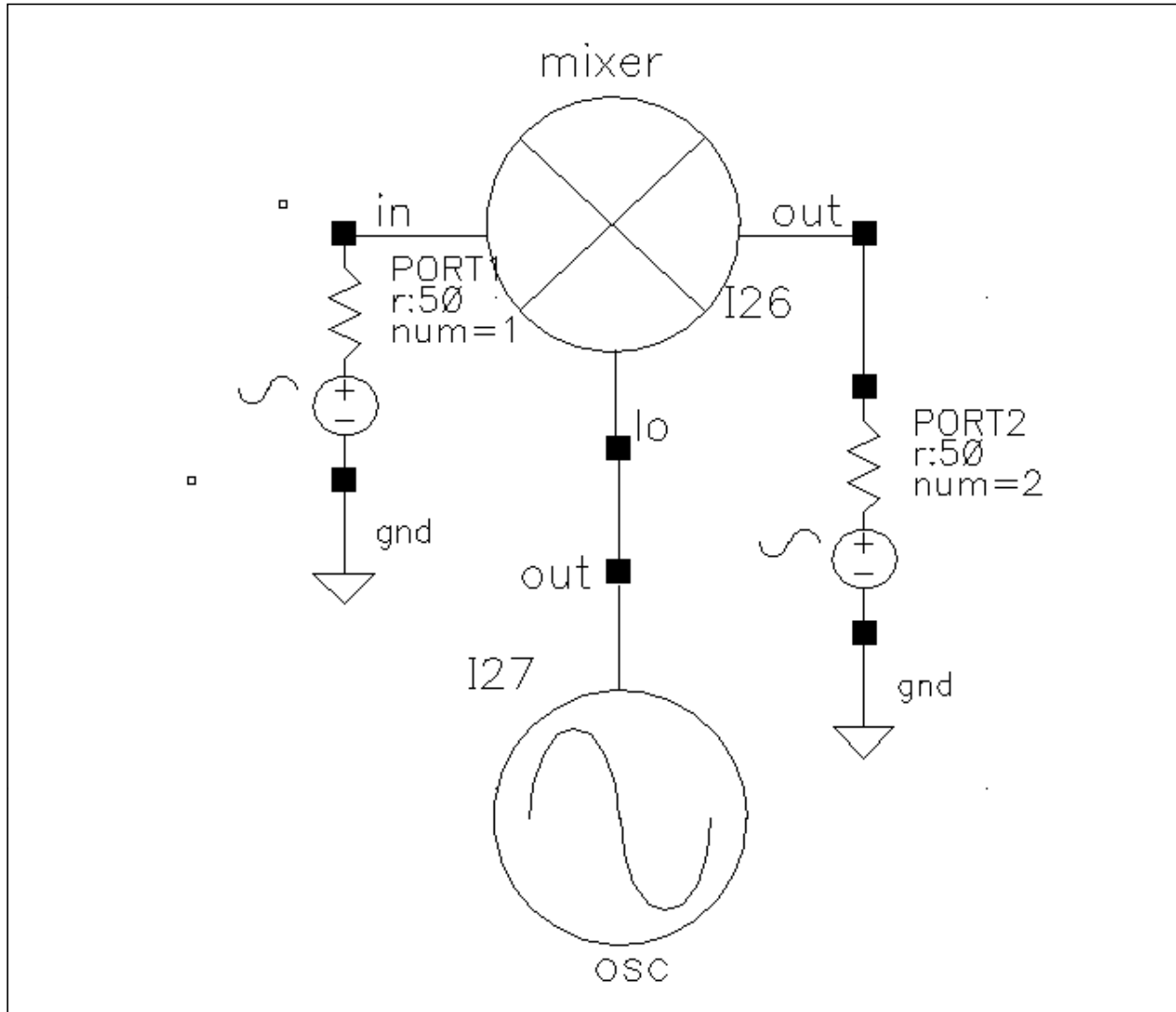
```
parameter real rlo = 50 from (0:inf);
parameter real ip2 = 5;
parameter real ip3 = 5;
parameter real nf = 2 from [0:inf);
parameter real isolation_LO2IN = 20 from (0:inf);
parameter real isolation_LO2OUT = 20 from (0:inf);
parameter real isolation_IN2OUT = 20 from (0:inf);
```

The parameters are

gain	Gain from IN to OUT [dB]
plo	Power of the LO input [dBm]
rin	Input impedance for IN [Ω]
rou	Output impedance for OUT [Ω]
rlo	Input impedance for LO [Ω]
ip2	Input referenced IP2 [dBm]
ip3	Input referenced IP3 [dBm]
nf	Noise figure (SSB) [dB]
isolation_LO2IN	Isolation from LO to IN [dB]
isolation_LO2OUT	Isolation from LO to OUT [dB]
isolation_IN2OUT	Isolation from IN to OUT [dB]

Figure D-6 on page 785 is the simple schematic that tests the filter.

Figure D-6 Schematic for Testing Mixer



p_{lo} is the maximum power of the fundamental frequency of the local oscillator that can be used in the mixing process. Therefore, the gain, defined as the output power of the mixed product versus the input power of RF signal, depends on the power level of the LO. The gain levels off, however, to the specified maximum value as the LO signal becomes larger.

You can measure both IP3 and IP2 in the Affirma analog circuit design environment. You must select frequencies carefully when you measure IP3 to measure Harmonic Distortion (HD) and IP2. Testing IP3 requires two tones to measure the Intermodulation Distortion (IMD), while testing IP2 requires only one tone.

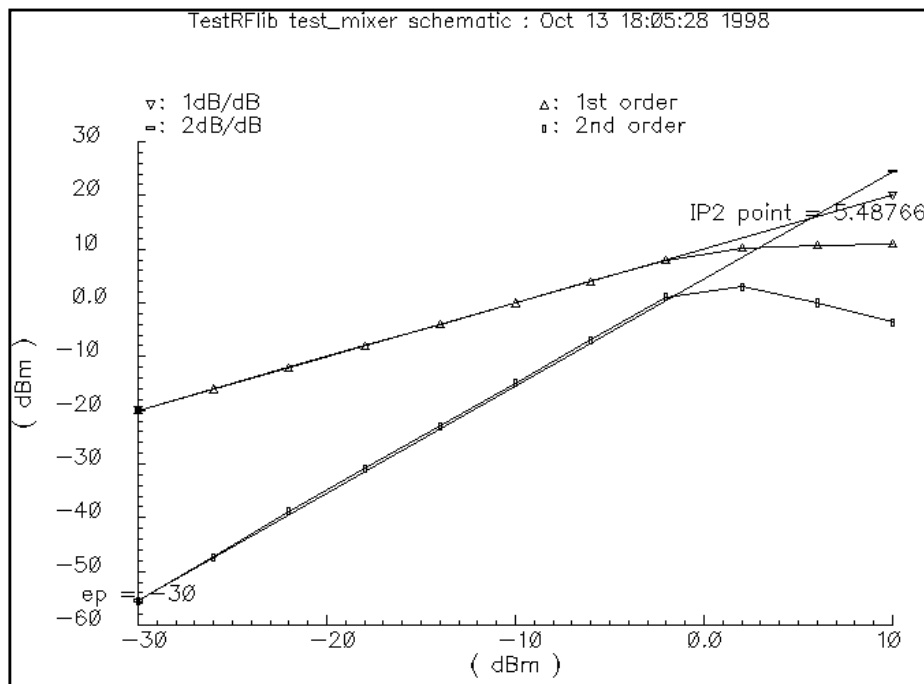
Affirma RF Simulator (SpectreRF) User Guide

The RF Library

Assume the RF input frequencies are f_1 and f_2 , and the LO frequency is f_{l0} . If the input power level at f_1 equals that at f_2 , the IP3 is the intercept point of the extrapolated line of output power at frequency $|f_{l0} - (2f_2 - f_1)|$ versus extrapolated line of the linear output signal at $|f_{l0} - f_1|$. Input referenced IP3, therefore, can be read as the x-axis value at the intercept point. The IP2, for the purpose of measuring the half-IF effects, is defined as the intercept point of the extrapolated line of output power at frequency $|2(f_{l0} - f_1)|$ versus the linear output signal. Figure D-7 shows that the intercept point of the 1dB/dB and 2dB/dB lines is at the x-axis reading of 4.78dBm, while the requested IP2 value is 5dBm. The order of the intercept point is based only on the order of the RF signals. The order of LO signal is not counted in the definition of the intercept point. In the implementation of this model, the orders of LO for IP3 and IP2 are 1 and 2 respectively.

Internally, a set of equations is built to satisfy a three-port S-parameter. A third-order polynomial describes the nonlinearity of IP3. The LO signal is further multiplied by itself to derive the second-order harmonic, which is then used to produce the IP2 effect. Excessive white noise is added in the RF input port to satisfy the noise figure. Remember, however, that the noise figure is a single-side band. If the noise at the image frequency is not filtered out, the measured noise figure is 3dB larger than the SSB noise figure.

Figure D-7 IP2 Measurement



Power Amplifier

Power amplifiers are used in RF transmitters to achieve output of high power level. The power amplifier(PA) model differs from the low-noise amplifier (LNA) in having greater power delivery capabilities with less stress on matching capabilities.

The definition of the model in the Verilog-A language is as follows:

```
module pa(in, out);
    inout in, out;
    electrical in, out;
    parameter real nf = 2 from [0:inf);
    parameter real gain = 20 from [0:inf);
    parameter real rin = 50 from (0:inf);
    parameter real rout = 50 from (0:inf);
    parameter real pldb = 30;
    parameter real psat = 35;
    parameter real ip2 = 40;
```

The parameters are

nf	Noise figure [dB]
gain	s21 [dB]
rin	Input impedance [Ω]
rout	Output impedance [Ω]
psat	Maximum output power [dBm]
pldb	Output referenced 1dB compression [dBm]
ip2	Input referenced IP2 [dBm]

The power amplifier model has the following three parts:

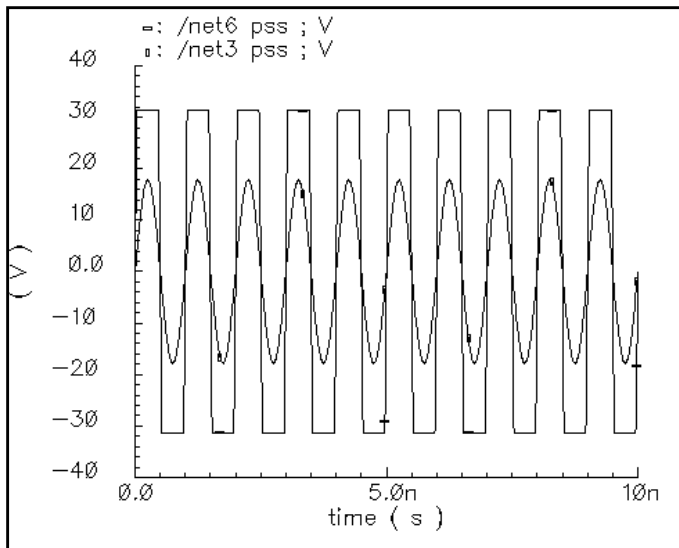
- the linear model
- the nonlinear model
- the noise model

Internally, for simplicity, the reverse isolation is assumed to be ideal. A set of linear equations is constructed to satisfy these S-parameters. Nonlinear effects are added to the gain to describe the nonlinearity. The output power of the power amplifier compresses to 1dB less than the output of an ideal linear amplifier at the 1dB compression point. Further increase of

the input power makes the output approach the saturation power only at the fundamental operating frequency. IP2 describes the second order effects of the amplifier, so use only one tone in the test. Excess white noise is added at the input port to describe the noise figure.

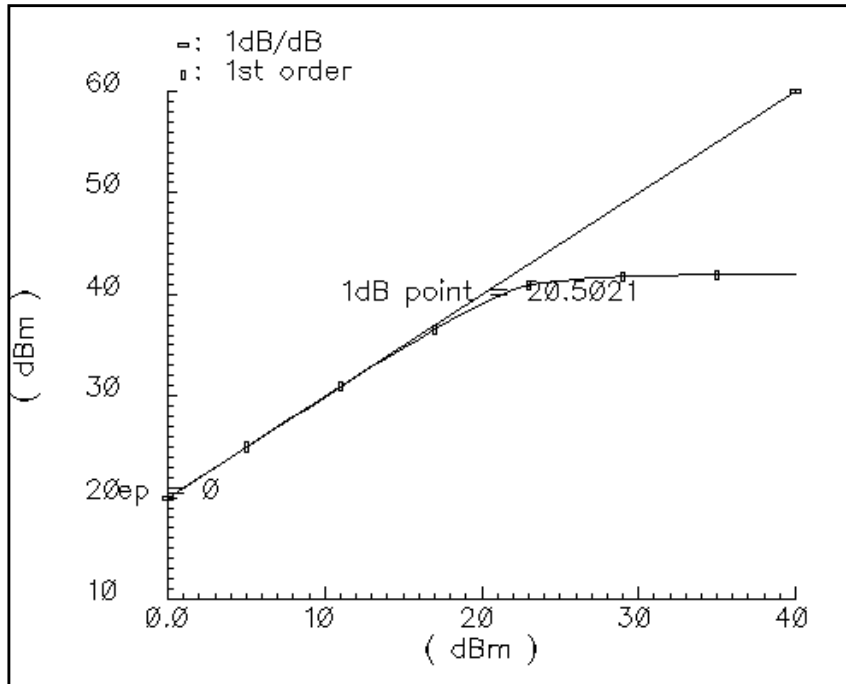
The implementation of P_{sat} assumes a pure sinusoidal waveform. To maintain a restrained output power, the output waveform is clipped from a sinusoidal to a square wave form. [Figure D-8](#) on page 788 shows the input and output waveforms of the power amplifier. Because of the output waveform clipping, the input sinusoidal wave should have a DC component of zero.

Figure D-8 Input and Output Waveforms of the Power Amplifier



[Figure D-9](#) shows the 1dB compression point and the saturation power. This difference is caused by the 50Ω load impedance. The specified output referenced 1dB compression point is 40dBm, which the Affirma analog circuit design environment captures as 39.6. Note that if P_{sat} is too much larger than p1db, the P_{sat} you want might not be satisfied.

Figure D-9 1dB Compression Point and Saturation Power



Oscillator

This model describes the essential information for a typical oscillator (or more precisely, local RF power source) in the Verilog-A language.

```

module osc(out);
    electrical out;
    inout out;
    parameter real power = 10;
    parameter real f = 1e9 from (0:inf);
    parameter real rout = 50 from (0:inf);
    parameter real floor = -60 from (-inf:0);
    parameter real fl = 1000 from (0:1e6);
    parameter real nl = -40 from (bottom:0);
    parameter real fc = 0 from [0:fl];

```

The parameters are

power	Output power when matched [dBm]
freq	Output frequency [Hz]

Affirma RF Simulator (SpectreRF) User Guide

The RF Library

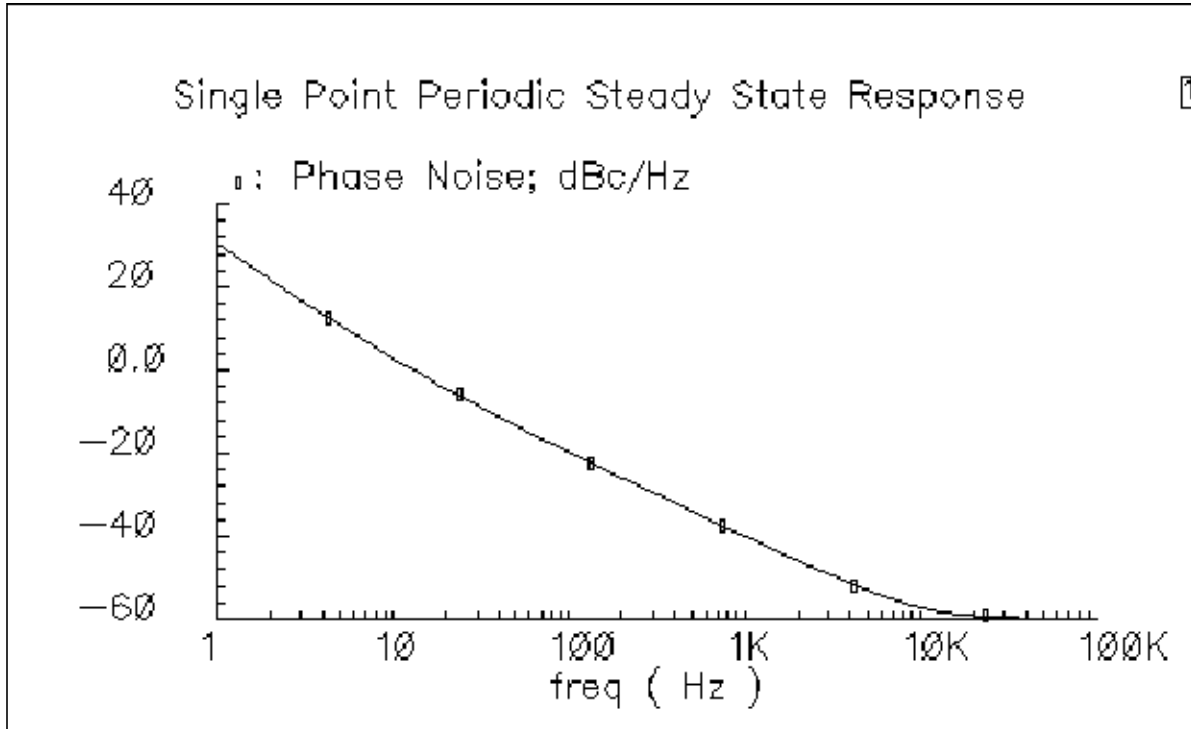
<code>rout</code>	Output impedance [W]
<code>floor</code>	Noise floor [dBc/Hz]
<code>f1</code>	Frequency point for n_1 [Hz]
<code>n1</code>	Phase noise at f_1 [dBc/Hz]
<code>fc</code>	Corner frequency of white phase and flicker phase [Hz]

This model is not an autonomous model. Rather, it simply generates a sinusoidal wave with the specified impedance, power level, and phase noise characteristics. When the load is matched to the internal impedance, the load dissipates the specified output power. You can specify the noise floor of the output signal. Furthermore, by adding one point (frequency, phase noise), you can specify $1/f^2$ frequency noise (corresponding to the phase noise induced by white noise). If f_c , the corner frequency of white phase and flicker phase noise, is bigger than 0, $1/f^3$ frequency noise (flicker noise induced phase noise) is further specified. Otherwise, $1/f^3$ noise is not included.

The phase noises symmetric around the carrier are correlated. The noise floor, however, is not correlated.

Figure D-10 on page 791 shows the phase noise of the oscillator model. The specified parameters are: *noise floor* = -60 dBc/Hz, $f_1=1$ K, $n_1=-40$ dBc/Hz, $f_c=100$.

Figure D-10 Phase Noise for the Oscillator



Quadrature Signal Generator

This module is included because in quadrature receiver design, a phase shifter is ordinarily used to generate the quadrature signal from one signal source such as the VCO. However, a phase shifter is hard to implement into a wide band model. A quadrature signal is two signals with a 90-degree phase difference but with identical noise and amplitude.

The definition of the model in the Verilog-A language is as follows:

```
module quadrature(lead, lag);
    electrical lead, lag;
    inout out_cos, out_sin;
    parameter real power = 10;
    parameter real f = 1e9 from (0:inf);
    parameter real rout = 50 from (0:inf);
    parameter real floor = -60 from (-inf:0);
    parameter real f1 = 1000 from (0:1e6);
    parameter real n1 = -40 from (bottom:0);
    parameter real fc = 0 from [0:f1];
```

The difference between this model and the oscillator model is that the oscillator has only one output node, but the quadrature signal generator has two output nodes, `lead` and `lag`. When the power levels, output impedances, and noise sources are identical, the two outputs have a 90-degree phase difference.

Phase Shifter

In digital RF system designs, quadrature signal processing involves the phase splitting of high-frequency signals. The most common use of such components is to generate two signals that have a 90-degree phase difference based on one signal source (such as the RF signal or oscillator output). Another common use is to combine two signals after adding a 90-degree phase difference, as in image-rejection receiver designs.

The definition of the model in the Verilog-A language is as follows:

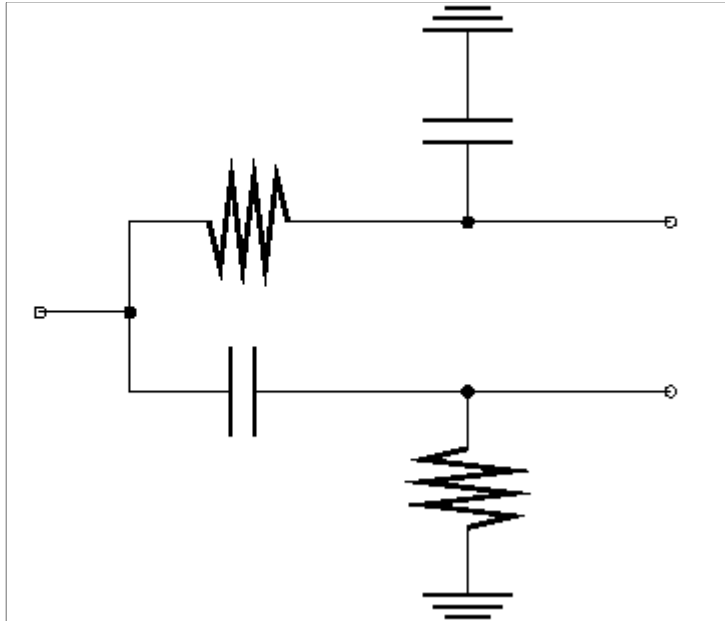
```
module shifter(single, lag, lead);
  inout single, lag, lead;
  electrical single, lag, lead;
  parameter real freq = 1e9 from (0:inf);
  parameter real r = 50 from (0:inf);
```

The parameters are

`freq` Frequency of operation [Hz]
`r` Resistance [Ω] (see [Figure D-11](#))

Internally, the phase shifter is implemented using the RC-CR circuit as shown in [Figure D-11](#) on page 793. While the phase difference is also 90-degrees when the lead and lag have the same output impedance, only at the operating frequency do the magnitudes remain the same. This circuit network also generates white noise.

Figure D-11 Phase Shifter



There are two buffered versions of the shifter: the *shifter_combiner* and the *shifter_splitter*. The *shifter_combiner* is used to combine two signals so that they add if one leads the other by 90 degrees and so that they cancel if it lags by 90 degrees. The *shifter_splitter* splits a signal into two signals 90 degrees out of phase with each other. You specify the input and output impedances. These networks are noiseless.

Bottom-Up Design Elements

For more information about bottom-up design elements in the *rfLib*, see the chapter in this book that describes the [J-model](#).

Testbench Elements

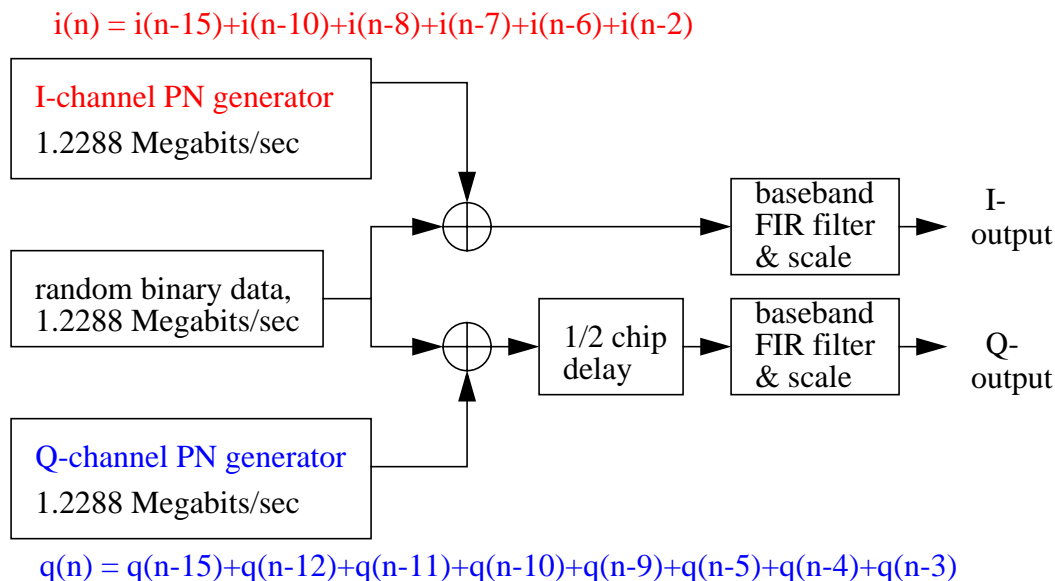
This section describes the Testbench elements and concludes with an explanation of how to change the FIR filters inside the baseband signal generators.

Note: All of the baseband signal sources generate digitally filtered signals. The baseband sources will not work with SpectreRF because the digital filters have hidden states.

CDMA Signal Source (IS-95 rev_xmit) and Eye-Diagram Generator

The CDMA signal source generates a reverse-link (handset-to-base-station) IS-95 signal. The modulation is offset QPSK with a symbol rate of 1.2288 Mega-symbols per second and a sample rate of 4.9152 Megasamples per second. Two separate 16-bit pseudo-noise generators generate the *I* and *Q* spreading sequences operating at the sample rate. The CDMA source generates a random bit at the symbol rate, oversamples it by a factor of 4, spreads the bit with the *I* and *Q* spreading sequences, and then filters each sequence with a 48-tap FIR filter. The filter coefficients are the impulse response of a raised cosine filter. [Figure D-12](#) shows a block diagram of the signal generator.

Figure D-12 CDMA Baseband Test Signal Generator



[Figure D-13](#) shows an eye-diagram of one of the outputs and the trajectory of both outputs.

The eye-diagram generator in the *test_bench* category created the eye-diagram. The input to the eye-diagram generator is the *I* or *Q* output of the signal generator. The eye-diagram generator has two outputs labeled “y-axis” and “x-axis”. The eye-diagram is generated by plotting the “y-axis” output against the “x-axis” output.

The eye-diagram generator does not work with Envelope Following Analysis to generate similar plots because the Envelope Following harmonic time analysis is generated by a post-processing step but the eye-diagram generator works during simulation.

Figure D-13 Eye Diagram and Trajectory

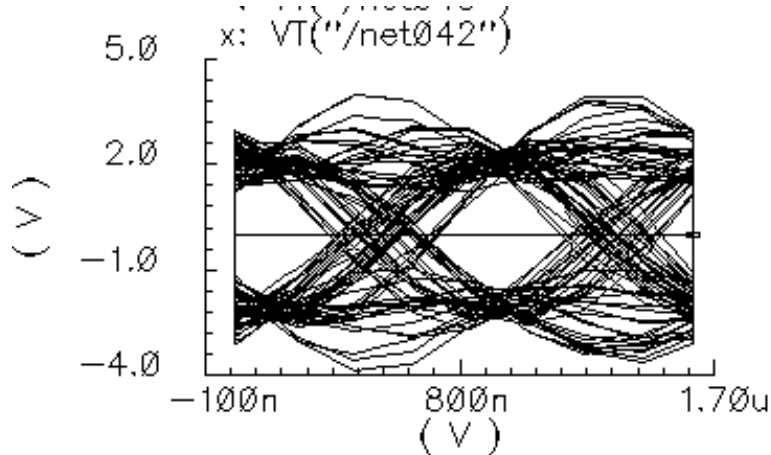


Figure D-13a Eye-diagram

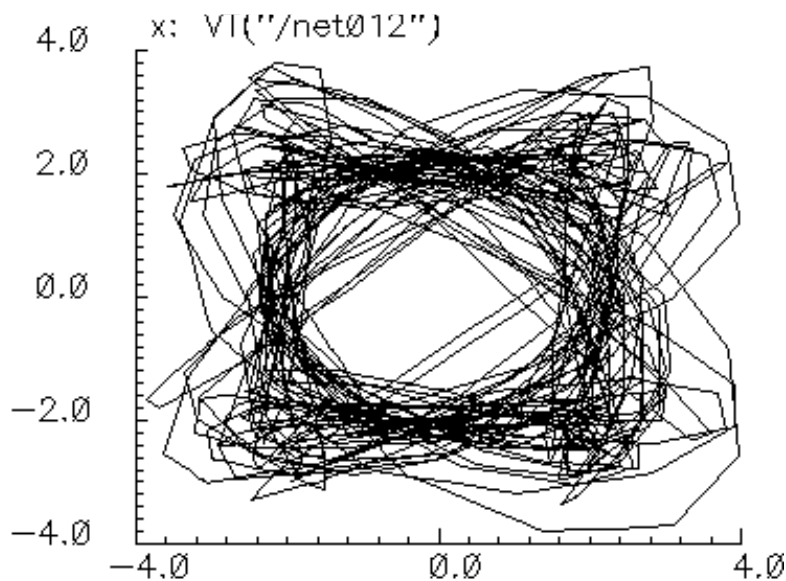


Figure D-13b CDMA trajectory

Instance Parameters

The *amplitude* parameter lets you set the amplitude of the unfiltered signals. An amplitude of 1 means that each FIR filter is driven by 1-volt impulses. If you change the internal variable *IMPULSE_PULSE* to 2, the filters are driven by 1-volt pulses of four samples duration.

The *seed* parameter lets you change the random number generator seed.

Outputs

The generator creates four output signals:

<i>i_bin_node</i>	The <i>I</i> unfiltered binary output.
<i>i_out_node</i>	The filtered <i>I</i> output.
<i>q_bin_node</i>	The <i>Q</i> unfiltered binary output.
<i>q_out_node</i>	The <i>Q</i> filtered output.

Changing the FIR Filter

You cannot change the FIR filter, such as the tap length and tap coefficients, directly from the instance. However, you can do so using the Modelwriter as described in [“Creating Variations on the Library Baseband Signal Generators Using the Modelwriter”](#) on page 801.

Output Transitions

The filtered outputs slew linearly from one value to the next because the rise and fall times in the transition statements equal one period. To make the outputs take abrupt steps, copy the module to your library and change the rise and fall times in the last transition statements.

GSM Signal Source (GSM-xmtr)

The GSM source generates a signal conforming to the GSM standard. The modulation is GMSK and the data is generated in frames of 3 fixed start bits, 142 random data bits, 3 fixed stop bits, and 8.25 fixed guard bits. (The embedded deterministic pattern and quarter of a bit is necessary to produce the correct spectrum.) The bit rate is 270833.333 bits per second and the sample rate is four times that.

The FIR filter is a Gaussian filter implemented with 32 taps.

[Figure D-14](#) on page 797 shows a block diagram of the signal source.

Figure D-14 GSM Baseband Signal Generator

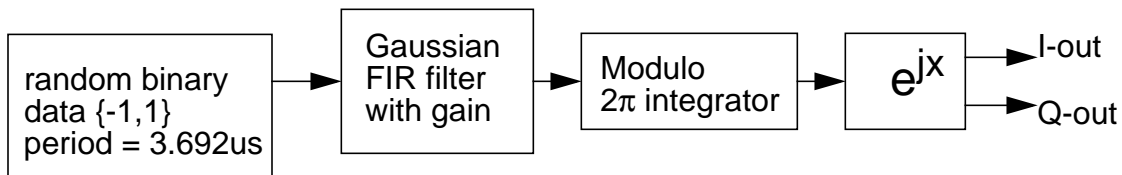
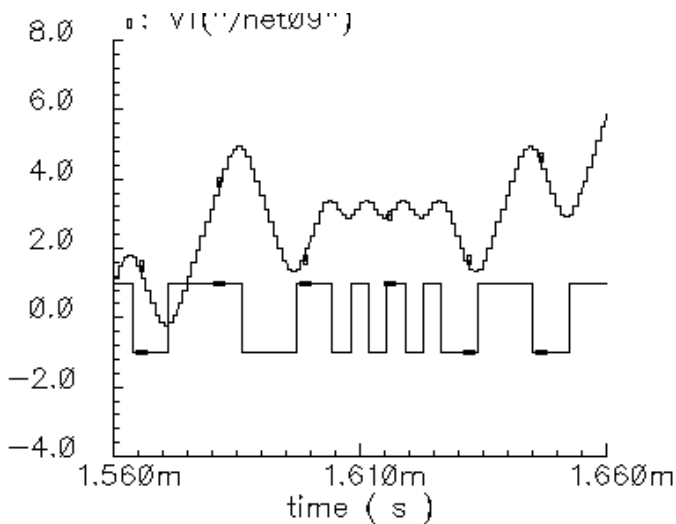


Figure D-15 on page 797 shows the binary data stream and the corresponding angle.

Figure D-15 GSM Binary Data and Resulting Phase



Instance Parameters

The *amplitude* parameter lets you set the amplitude of the unfiltered signals. An amplitude of 1 means that each FIR filter is driven by 1-volt impulses. If you change the internal variable *IMPULSE_PULSE* to 2, the filters are driven by 1-volt pulses of four samples duration.

The *seed* parameter lets you change the random number generator seed.

Outputs

The generator creates four output signals

- angular_node* The output signal.
- i_out_node* The phase, multiplied by the amplitude.
- bin_node* The bit stream being transmitted.
- q_out_node* The phase multiplied by the amplitude.

Changing the FIR Filter

You cannot change the FIR filter, such as the tap length and tap coefficients, directly from the instance. However, you can do so using the Modelwriter as described in [“Creating Variations on the Library Baseband Signal Generators Using the Modelwriter”](#) on page 801.

Output Transitions

The filtered outputs slew linearly from one value to the next because the rise and fall times in the transition statements equal one period. To make the outputs take abrupt steps, copy the module to your library and change the rise and fall times in the last transition statements.

$\Pi/4$ -DQPSK Signal Source (*pi_over4_dqpsk*)

[Figure D-16](#) shows the block diagram for this source and [Table D-1](#) on page 799 shows how the phase shift is generated.

Figure D-16 $\pi/4$ -DQPSK baseband signal generator

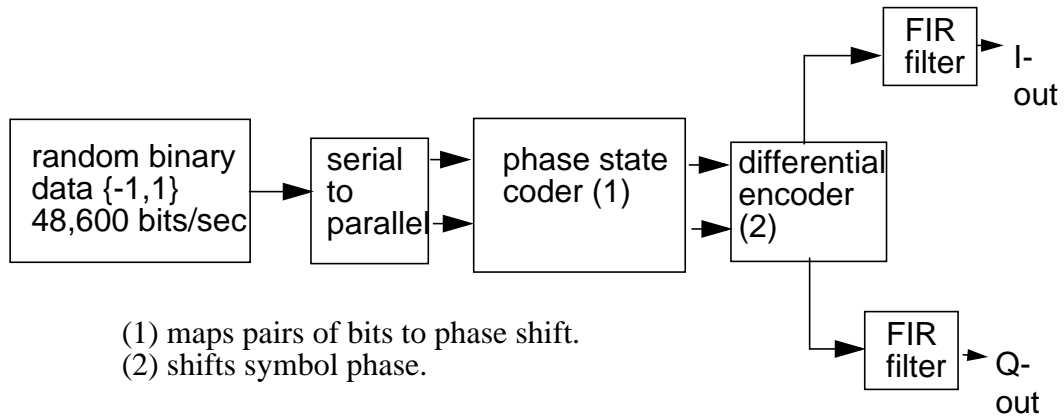


Table D-1 Phase Shift

1st bit	2nd bit	Phase shift
0	0	$\pi/4$
0	1	$3\pi/4$
1	0	$-\pi/4$
1	1	$-3\pi/4$

The symbol rate is 24300 symbols per second and the sample rate is 8 times that. The FIR filter is a raised cosine filter implemented with 64-taps.

Figure D-17 shows the eye-diagram and trajectory for this generator

Figure D-17 Eye Diagram and Trajectory

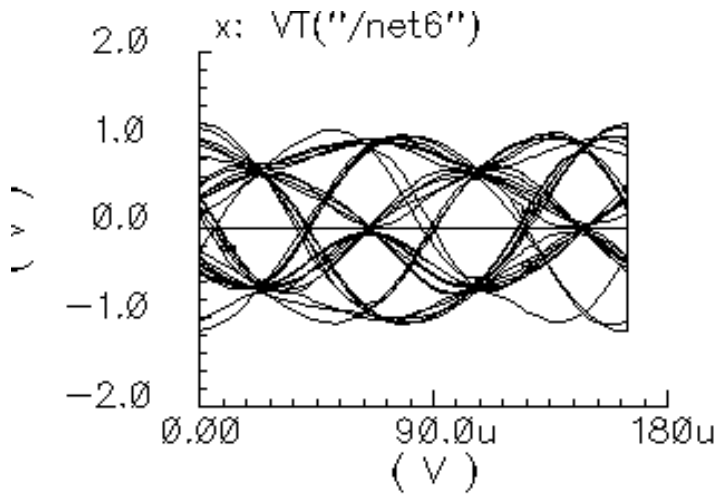


Figure D-17a, $\Pi/4$ -dqpsk eye-diagram

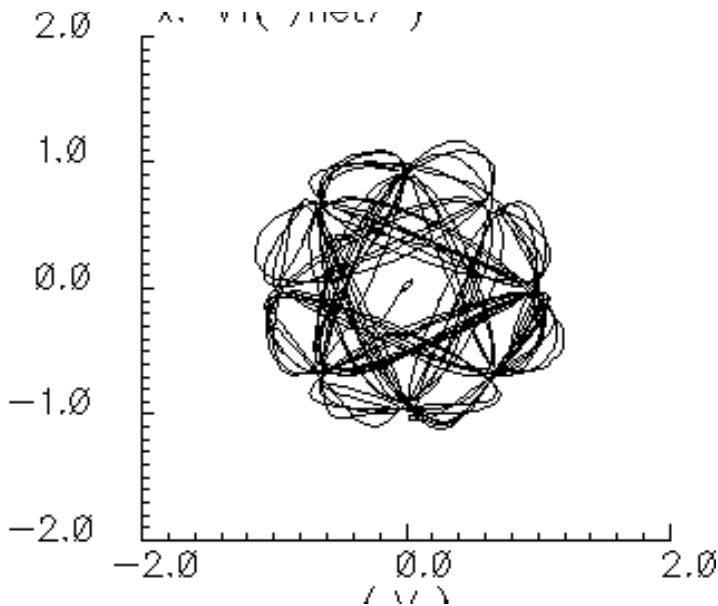


Figure D-17b, $\Pi/4$ -dqpsk trajectory

Instance parameters

The *amplitude* parameter lets you set the amplitude of the unfiltered signals. An amplitude of “1” means that each FIR filter will be driven by 1-volt impulses. If you change the internal variable *IMPULSE_PULSE* to 2, the filters will be driven by 1-volt pulses of four samples duration.

The *seed* parameter lets you change the random number generator seed.

Outputs

The generator creates four output signals.

<i>i_out_node</i>	The phase, multiplied by the amplitude.
<i>q_out_node</i>	The phase, multiplied by the amplitude.
<i>phase_shift_out</i>	The phase shift from one symbol to the next.

Changing the FIR filter

You cannot change the FIR filter, such as the tap length and tap coefficients, directly from the instance. However, you can do so using the Modelwriter as described in [“Creating Variations on the Library Baseband Signal Generators Using the Modelwriter”](#) on page 801.

Output transitions

The filtered outputs slew linearly from one value to the next because the rise and fall times in the transition statements equal one period. To make the outputs take abrupt steps, copy the module to your library and change the rise and fall times in the last transition statements.

Creating Variations on the Library Baseband Signal Generators Using the Modelwriter

The baseband signal generators use FIR filters to shape their output pulses. Shaped output pulses serve several purposes:

- They help keep the transmitted signal inside the specified band.
- They work with their receive counterparts to maximize the signal-to-noise ratio.
- Together with their receiver counterparts, they satisfy the Nyquist sampling criterion for an intersymbol-interference-free channel.

The Modelwriter gives you a convenient user-interface for creating variations on the baseband signal generators in the library. The most likely variation is in the FIR filter. This section explains how to use the Model Writer to create a new GSM generator with different FIR filters.

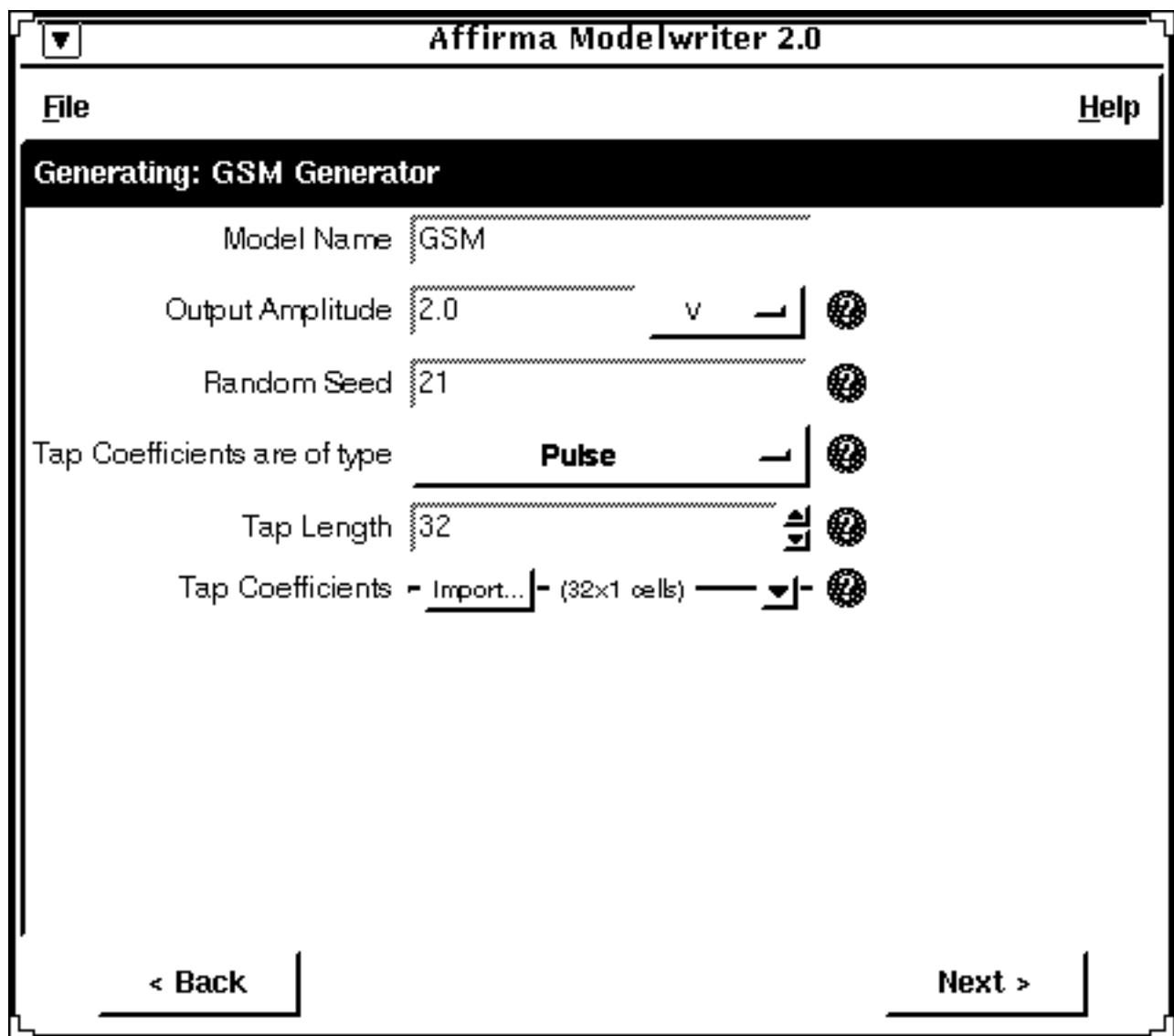
Affirma RF Simulator (SpectreRF) User Guide

The RF Library

1. Bring up the Modelwriter and do the following:
 - a. Double click on the Telecom folder.
 - b. Select the GSM generator.
 - c. Click on the *Next* button in the lower right hand corner of the Modelwriter window.

You should now see the picture in [Figure D-18](#) in the window.

Figure D-18 GSM Generator



2. Specify how you plan to drive the filter by selecting the type of tap coefficients.

FIR filters can be driven by pulses or impulses.

3. Specify the length of the FIR filter in the *Tap Length* field.
4. You can specify the tap coefficients in two ways
 - From a file
 - By direct manual entry.

To read the coefficients from a file do the following:

- a. Select the *Import* button.
- b. Enter the path to the file.
- c. Click on *Open*.

The coefficients appear in the window as shown in [Figure D-19](#) on page 804. (The tap[1] coefficient multiplies the filter state with the least amount of delay, the filter state closest to the input.)

To enter the coefficients manually do the following:

- a. Click on the lower rightmost button flagged in the form.
See [Figure D-20](#) on page 805.
- b. Enter the coefficients,
- c. Click *next* to view the model,
- d. To write the model to a file, click *Save Generated Code....*

Affirma RF Simulator (SpectreRF) User Guide
The RF Library

Figure D-19 Tap coefficients

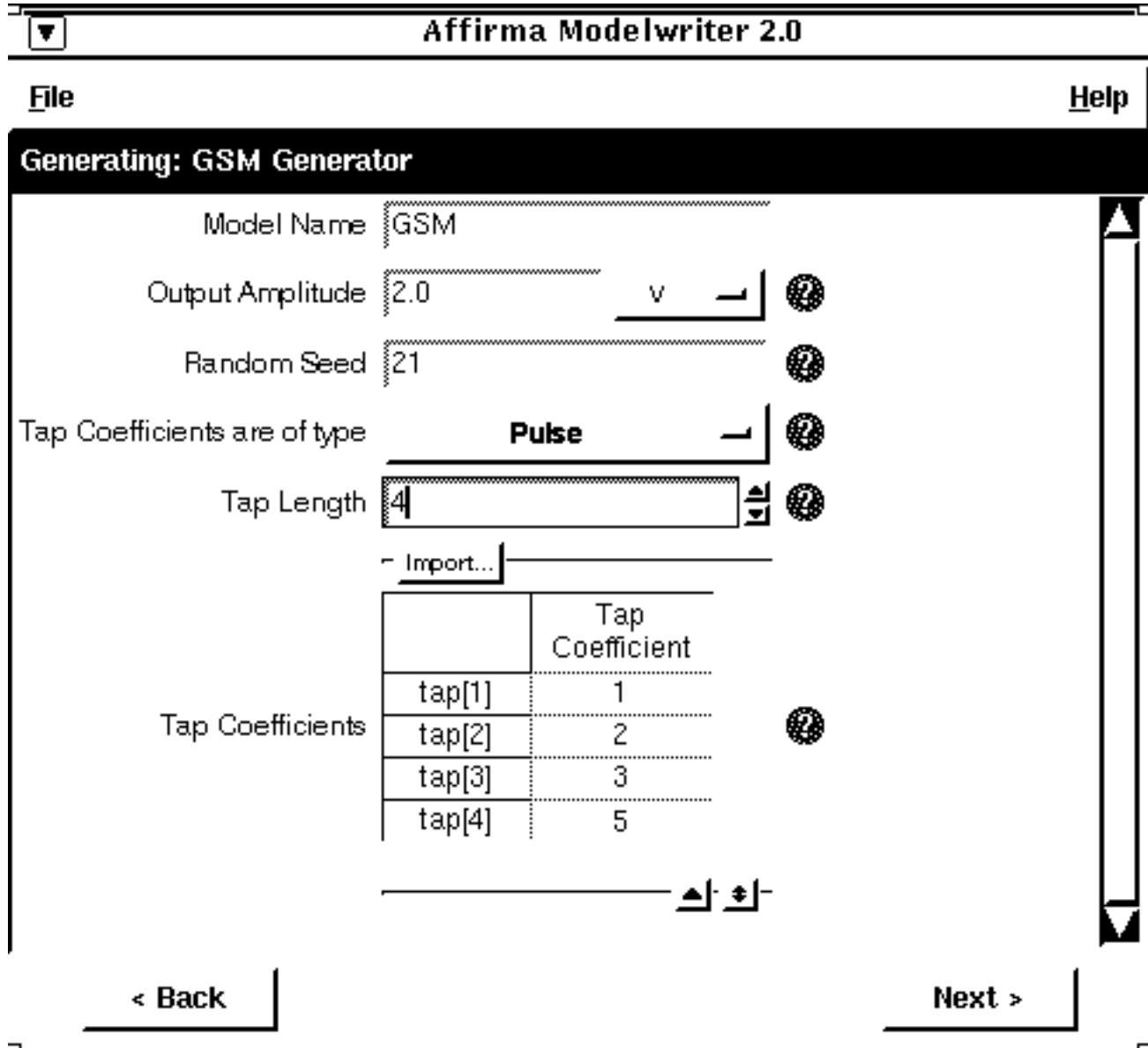
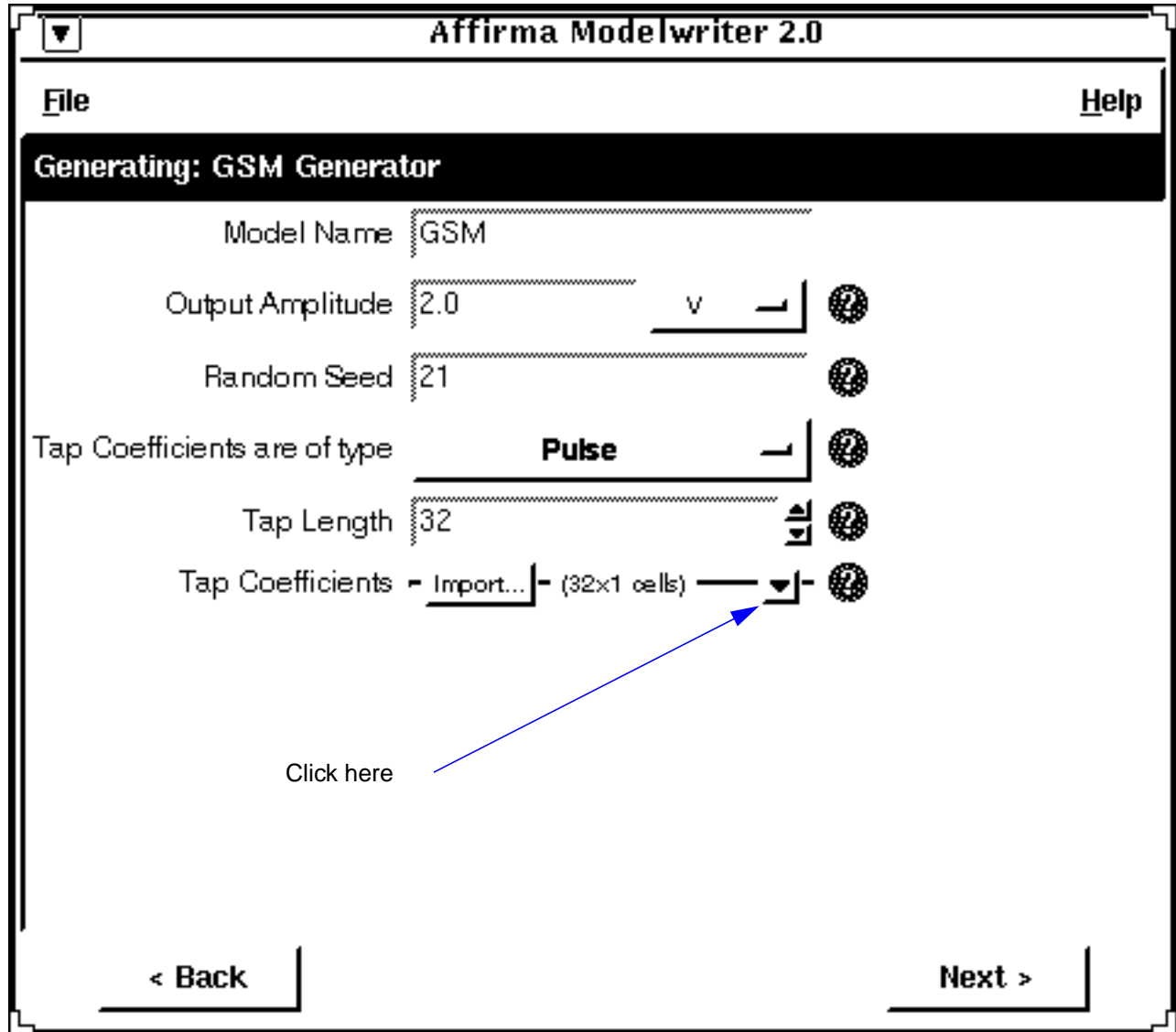


Figure D-20 Manual entry of the tap coefficients



Uncategorized Elements

The `rfLib` contains the following uncategorized elements:

- `cdma_2ms_idata`
- `cdma_2ms_qdata`
- `dqpsk_20ms_idata`

Affirma RF Simulator (SpectreRF) User Guide

The RF Library

- dqpsk_20ms_qdata
- gsm_5ms_idata
- gsm_5ms_qdata

All of these elements are files generated by elements of the Testbench category described in [“Testbench Elements”](#) on page 793.

Plotting Spectre S-Parameter Simulation Data

The procedures described in this appendix are deliberately broad and generic. Requirements for your specific design might dictate procedures slightly different from those described here.

This appendix describes the equations used by the Spectre S-Parameter Waveform Calculator.

The Equations for the S-Parameter Waveform Calculator in the Affirma Analog Circuit Design Environment:

Using the Waveform Calculator in the Affirma analog circuit design environment, you can plot the following S-parameter data:

- Network parameters
- Two-port scalar quantities
- Two-port gain quantities
- Two-port network circles
- Voltage standing wave ratio
- Group delay

On the S-Parameter Results form, buttons in the Function area specify the type of data to plot.

Network Parameters: S, Z, Y, or H parameters

S, Y, and Z parameters (denoted as SP, ZP, YP, and HP on the Plot form) are defined for circuits with any number of ports. H-parameters (HP) are defined for only two-port circuits.

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

You can plot the parameters on polar charts, Smith charts, or on a rectangular plot after applying one of the Modifier options. The dB conversion uses $20\log_{10}x$, because the parameters represent scalar (for example, voltage) ratios.

For the ZP, YP, and HP parameters, Spectre returns S-parameters to the Affirma analog circuit design environment. The environment converts them, as needed, to the equivalent Z, Y, and H matrices using standard published methods.

The network parameter calculations:

- **SP (S-parameter)** values are calculated by Spectre

- **ZP (Z-parameter)** equations

$$Z_m = [Z_{ref}][I - S_m][I + S_m]^{-1}[Z_{ref}]$$

where

S_m is the N-port S-parameter matrix

I is the N x N identity matrix

Z_{ref} is a diagonal matrix defined as

$$Z_{ref} = SQRT(\Re\{Z_i\})$$

where

Z_i is the terminating impedance at port i

Z_m is the resulting Z-parameter matrix

- **YP (Y-parameter)** equation

$$Y_m = [Y_{ref}][I - S_m][I + S_m]^{-1}[Y_{ref}]$$

where

S_m is the N-port S-parameter matrix

I is the N x N identity matrix

Y_{ref} is a diagonal matrix defined as

$$Y_{ref} = \frac{1}{SQRT(\Re\{Z_i\})}$$

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

where

Z_i is the terminating impedance at port i

Y_m is the resulting Y-parameter matrix

- The **HP (H-parameter)** equations apply to only two-port circuits.

$$D = (1 - S_{11})(1 + S_{22}) + S_{21}S_{12}$$

$$H_{11} = \frac{((1 + S_{11})(1 + S_{22}) - S_{21}S_{12})Z_{ref1}^2}{D}$$

$$H_{21} = \left(\frac{-2S_{21}}{D}\right)\frac{Z_{ref1}}{Z_{ref2}}$$

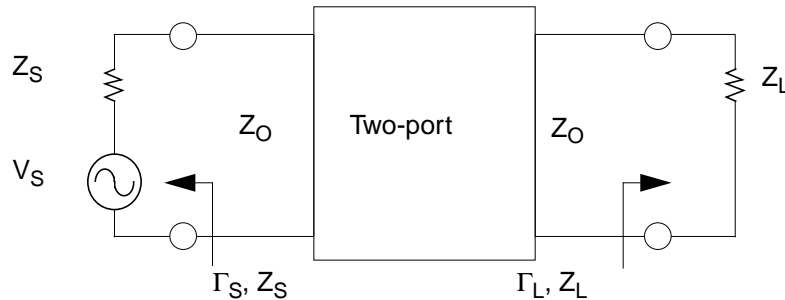
$$H_{12} = \left(\frac{2S_{21}}{D}\right)\frac{Z_{ref1}}{Z_{ref2}}$$

$$H_{22} = \frac{((1 - S_{11})(1 - S_{22}) - S_{21}S_{12})Z_{ref}^2}{D}$$

where Z_{ref1} is the terminating impedance at port 1, and Z_{ref2} is the terminating impedance at port 2.

Two-port Scalar Quantities: NFmin, NF, Rn, Gmin, B1f, Kf, GT, GA, and GP.

These 2-port scalar quantities can be plot only against frequency. In addition, you must plot them on rectangular charts. Below is a generic two-port circuit that defines impedances and reflection coefficients.



Z_S is the source impedance
 Γ_S is the input reflection coefficient
 Z_L is the load impedance
 Γ_L is the load reflection coefficient
 Z_0 is the characteristic impedance

The Two-port Scalar Quantity equations

- **Gmin**, also known as Γ_{min} in the literature, is the reflection coefficient associated with minimum noise figure. You can plot Gmin on a Smith chart or, by using the Modifier field, on a rectangular chart.

$$G_{min} = \Gamma_{min} = \frac{Z_{on} - Z_o}{Z_{on} + Z_o}$$

where Z_0 is the characteristic impedance, and Z_{on} is the source impedance associated with minimum noise figure.

- **NFmin** and **NF** plot contents are controlled by the Modifier option. With Modifier set to dB10, these functions plot noise figure in dB; with Modifier set to Magnitude, they plot noise factor. Use NFmin and NF only for two-port circuits.

NF is the noise figure.

$$NF = 10\log_{10}F$$

where F is the noise factor.

NFmin is the minimum noise figure.

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

The **NFmin** values are calculated by Spectre

The **NF (noise figure)** equation is calculated by the Affirma analog circuit design environment from **NFmin**, **Gmin** (Γ_{min}), and r_n . You can specify the optional source reflection coefficient Γ_S as an argument if you use the design environment calculator. From the direct plot form, The environment assumes Γ_S to be 0 (input matched to reference termination).

$$NF = NF_{min} + \frac{4r_n|\Gamma_S - \Gamma_{min}|^2}{(1 - |\Gamma_S|^2)(1 + |\Gamma_{min}|^2)}$$

Where $r_n = R_n/Z_o$ and r_n is the *normalized* equivalent noise resistance.

- **Rn (equivalent noise resistance)** plots equivalent noise resistance. The R_n values are calculated by Spectre.
- **Stability Factors**

Kf and **B1f** plot the Rollet stability factor and its intermediate term. Use these parameters only for two-port circuits.

$$D = S_{11}S_{22} - S_{21}S_{12}$$

$$K_f = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |D|^2}{2|S_{21}||S_{12}|}$$

$$B_{1f} = \frac{1 + |S_{11}|^2 - |S_{22}|^2 + |D|^2}{2|S_{11}| - S_{22}^*D}$$

Two-port gain quantities: **GA**, **GP**, **GT**, **Gumx**, **Gmax**, and **Gmsg**.

The following gain quantities are valid only for two-port circuits

GA (available gain) is the power gain obtained by optimally matching the output of the network.

GP (power gain) is the power gain obtained by optimally matching the input of the network.

GT (transducer gain) shows the insertion effect of a two-port circuit (used in amplifier design).

Gumx (maximum unilateral transducer power gain)

Gmax (maximum available gain) shows the transducer power gain when there exists a simultaneous conjugate match at both ports.

Gmsg (maximum stable gain) shows the gain that can be achieved by resistively loading the 2 port such that $k=1$ and then simultaneously conjugately matching the input and output ports. For conditionally stable 2-ports, you can approach the maximum stable gain as you reduce the input and output mismatch. If you attempt a simultaneous conjugate match and $k<1$, the 2-port will oscillate.

Two-port Gain Calculations

- **GA (available gain)** equations (output conjugately matched)

■

$$G_A = \frac{|S_{21}|^2(1-|\Gamma_S|^2)}{|1-S_{11}\Gamma_S|^2(1-|\Gamma_2|^2)}$$

where

$$\Gamma_2 = S_{22} + \frac{S_{12}S_{21}\Gamma_S}{1-S_{11}\Gamma_S}$$

Note: When you use the S-parameter Results direct plot form, G_S is set to zero, and therefore available gain (GA) is plotted as

$$G_A = \frac{|S_{21}|^2}{1-|S_{22}|^2}$$

To plot GA for nonzero values of G_S , use the design environment calculator.

- **GP (power gain)** equations (input conjugately matched)

$$G_P = \frac{|S_{21}|^2(1-|\Gamma_L|^2)}{|1-S_{22}\Gamma_L|^2(1-|\Gamma_1|^2)}$$

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

where

$$\Gamma_1 = S_{11} + \frac{S_{12}S_{21}\Gamma_L}{1 - S_{22}\Gamma_L}$$

Note: When you use the S-parameter Results direct plot form, Γ_L is set to zero, and therefore the power gain, GP, is plotted as

$$G_P = \frac{|S_{21}|^2}{1 - |S_{11}|^2}$$

To plot GP for nonzero values of Γ_L , use the design environment calculator.

■ GT (transducer gain):

$$G_T = \frac{(1 - |\Gamma_S|^2)|S_{21}|^2(1 - |\Gamma_L|^2)}{|(1 - S_{11}\Gamma_S)(1 - S_{22}\Gamma_L) - S_{12}S_{21}\Gamma_S\Gamma_L|^2}$$

Note: When using the S-parameter direct plot Results form, the Affirma analog circuit design environment assumes that the source (Γ_S) and load (Γ_L) reflection coefficients are zero. GT, therefore, plots the insertion gain.

$$G_T \text{ (insertion gain)} = |S_{21}|^2$$

Using the calculator, you can plot GT and specify the source and load terminations.

■ Gmax (maximum available gain)

$$G_{max} = \frac{|S_{21}|}{|S_{12}|} [K - SQRT(K^2 - 1)] \quad K > 1$$

$$G_{max} = \frac{|S_{21}|}{|S_{12}|} \quad K \leq 1$$

where K is the stability factor, Kf

■ Gumx (maximum unilateral power gain).

$$G_{umx} = \frac{|S_{21}|^2}{(1 - |S_{11}|^2)(1 - |S_{22}|^2)}$$

- Gmsg (maximum stable power gain)

$$G_{msg} = \frac{|S_{21}|}{|S_{12}|}$$

Two-port Network Circles: NC, GAC, GPC, LSB, and SSB

NC plots constant noise contours at the input of a two-port circuit.

GAC plots constant gain contours at the input port, and GPC plots constant gain contours at the output port. Gain contour values reflect an optimum match at the opposing port.

Noise and Gain circles can be plotted at a single dB value for a range of frequencies or at a single frequency for a range of dB values. If you do not enter values for the frequency range, a circle is plotted for every simulated frequency for which a circle with the specified value exists.

SSB plots stability circles at the input port, and LSB plots those at the output port. You can also specify a limited frequency range for these contours.

The two-port network circle equations

- NC (noise circle)

$$N_i = \frac{(F_i - F_{min})|1 + \Gamma_{min}|^2}{4r_n}$$

where

$$\Gamma_{min} = G_{min}$$

The center is calculated using

$$C_N = \frac{\Gamma_{min}}{1 + N_i}$$

The radius is calculated using

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

$$r_N = \text{SQRT} \left[\frac{N_i^2 + N_i(1 - |\Gamma_{min}|^2)}{1 + N_i} \right]$$

where i is the index number

■ GAC (available gain circle)

The center is calculated using

$$C_A = \frac{g_a(S_{11}^* - D^*S_{22})}{1 + g_a(|S_{11}|^2 - |D|^2)}$$

The radius is calculated using

$$r_A = \frac{\text{SQRT}[(1 - 2K_f|S_{21}S_{12}|g_a + |S_{12}S_{21}|^2g_a^2)]}{|1 + g_a(|S_{11}|^2 - |D|^2)|}$$

where

$$g_a = \frac{G_A}{|S_{21}|^2}$$

$$D = S_{11}S_{22} - S_{12}S_{21}$$

■ GPC (power gain circle)

The center is calculated

$$C_P = \frac{g_p(S_{22}^* - D^*S_{11})}{1 + g_p(|S_{22}|^2 - |D|^2)}$$

The radius is calculated using

$$r_P = \frac{\text{SQRT}[1 - 2K_f|S_{21}S_{12}|g_p + |S_{12}S_{21}|^2g_p^2]}{1 + g_p(|S_{22}|^2 - |D|^2)}$$

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

where

$$g_p = \frac{G_p}{|S_{21}|^2}$$

$$D = S_{11}S_{22} - S_{12}S_{21}$$

■ LSB (load stability circle)

The center is calculated using

$$C_L = \frac{S_{11}D^* - S_{22}^*}{|D|^2 - |S_{22}|^2}$$

The radius is calculated using

$$r_L = \left| \frac{S_{12}S_{21}}{|D|^2 - |S_{22}|^2} \right|$$

where

$$D = S_{11}S_{22} - S_{12}S_{21}$$

■ SSB (source stability circle)

The center is calculated using

$$C_S = \frac{S_{22}D^* - S_{11}^*}{|D|^2 - |S_{11}|^2}$$

The radius is calculated using

$$R_S = \left| \frac{S_{12}S_{21}}{|D|^2 - |S_{11}|^2} \right|$$

where

$$D = S_{11}S_{22} - S_{12}S_{21}$$

Affirma RF Simulator (SpectreRF) User Guide

Plotting Spectre S-Parameter Simulation Data

- VSWR (voltage standing wave ratio) is calculated from the S-parameters
:

$$VSWR_i = \frac{1 + |S_{ii}|}{1 - |S_{ii}|}$$

where i is the port number. You can plot the VSWR at any port in the circuit on a rectangular chart.

- Z_m (input impedance if all other ports are matched
)

$$Z_m = \frac{1 + S_{ii}}{1 - S_{ii}} R$$

where R the reference impedance of the port of interest, and i is the port number.

- **GD (group delay)** approximates the derivative of the phase with respect to frequency, normalized to 360 degrees.

$$G_d \cong \frac{-d\phi}{d\omega} \text{ , units in seconds}$$

Group delay is calculated from the phase of the corresponding S-parameter (for example, GD_{21} corresponds to S_{21}).

Using Pdisto Analysis Effectively

Periodic Distortion (Pdisto) analysis computes the steady-state responses of circuits with multiple periodic inputs of differing frequencies. The Pdisto mixed frequency-time (MFT) approach is more efficient than transient circuit simulation when inputs are stiff (with large time constants). MFT also performs better than traditional algorithms such as harmonic balance when solving highly nonlinear problems. A typical application of Pdisto is to predict the harmonic distortion of switched-capacitor filters that operate under widely separated fundamentals (see an example in [“Switched Capacitor Filter Example”](#) on page 826).

Pdisto can also predict intermodulation distortion of a narrowband circuit that is driven by a LO and two high-frequency input fundamentals that are closely spaced in frequency, such as in the down conversion stage at a receiver, (see an example in [“High-Performance Receiver Example”](#) on page 827).

This Appendix contains the following:

- A discussion of when to use Pdisto analysis
- A brief description of the MFT method
- A comparison of Pdisto and PSS analyses
- A comparison of Pdisto and PAC analyses
- Two examples of typical use: a switched capacitor filter and a high-performance receiver
- Procedures for setting up and running Pdisto analysis

When to Use Pdisto Analysis

The increasing demand for low-cost, mobile communication systems increases the need for efficient and accurate simulation algorithms for RF communication circuits. These circuits are difficult to simulate because they process modulated carrier signals that consist of a high frequency carrier and a low frequency modulation signal. Typically, the carrier frequency ranges from 1-5 GHz and the modulation from 10 kHz to 1 MHz. Applying standard transient analysis to such circuits might require simulating the detailed response of the circuit over

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

hundreds of thousands of clock cycles (millions of timepoints), a generally impractical approach.

Fortunately, many circuits of interest operate near a time-varying, but periodic, operating point. Some of these circuits can be analyzed if you assume that one of the circuit inputs produces a periodic response that you can calculate using PSS analysis. Other time-varying circuit inputs are then treated as small signals by linearizing the circuit around the periodic operating point, and you can analyze small signals efficiently using the SpectreRF Periodic AC (PAC) analysis [8]. This approach lets you avoid long simulation times with transient analysis[7].

However, many circuits cannot be analyzed efficiently with this periodic-operating-point-plus-small-signal approach. For example, predicting the intermodulation distortion of a narrowband circuit, such as a receiver down converter (a mixer followed by a filter), requires calculating the nonlinear response of the mixer circuit, driven by a LO, to two closely spaced high-frequency inputs. The response to both inputs is within the band width of the filter. The steady-state response of such a circuit is quasi-periodic.

As a further complication, many multi-timescale circuits, such as mixers and switched-capacitor filters, have a highly nonlinear response with respect to one or more of the exciting inputs. Consequently, steady-state approaches such as multi-frequency harmonic balance do not perform well.

Mixed frequency-time (MFT) approaches avoid these difficulties. They assume that many circuits of engineering interest have a strongly nonlinear response to only one input, such as the clock in a switched-capacitor circuit or the LO in a mixer, and respond in a weakly nonlinear manner to other inputs.

Compared to previous MFT methods [2, 4], the Pdisto algorithm has many of the following advantages:

- MFT methods avoid the ill-conditioning caused by a poor choice of boundary conditions found in previous algorithms
- MFT methods use a multi-dimensional discrete Fourier transform (DFT) for cycle placement
- MFT methods use a continuation method to enhance the global convergence of Newton's method
- MFT methods use a matrix-implicit, Krylov-subspace-based iterative scheme that lets MFT methods solve large problems
- MFT methods have a preconditioning strategy that permit the iterative solver to rapid converge

If you are unfamiliar with terminology such as *matrix-implicit*, *Krylov-subspace*, and *preconditioning*, please refer to [6] for details. You can find an introduction to *continuation methods* in [1].

Essentials of the MFT Method

Circuit behavior is usually described by a set of nonlinear differential-algebraic equations (DAEs) that can be written as

$$\frac{d}{dt}Q(v(t)) + I(v(t)) + u(t) = 0 \quad (3.1)$$

where

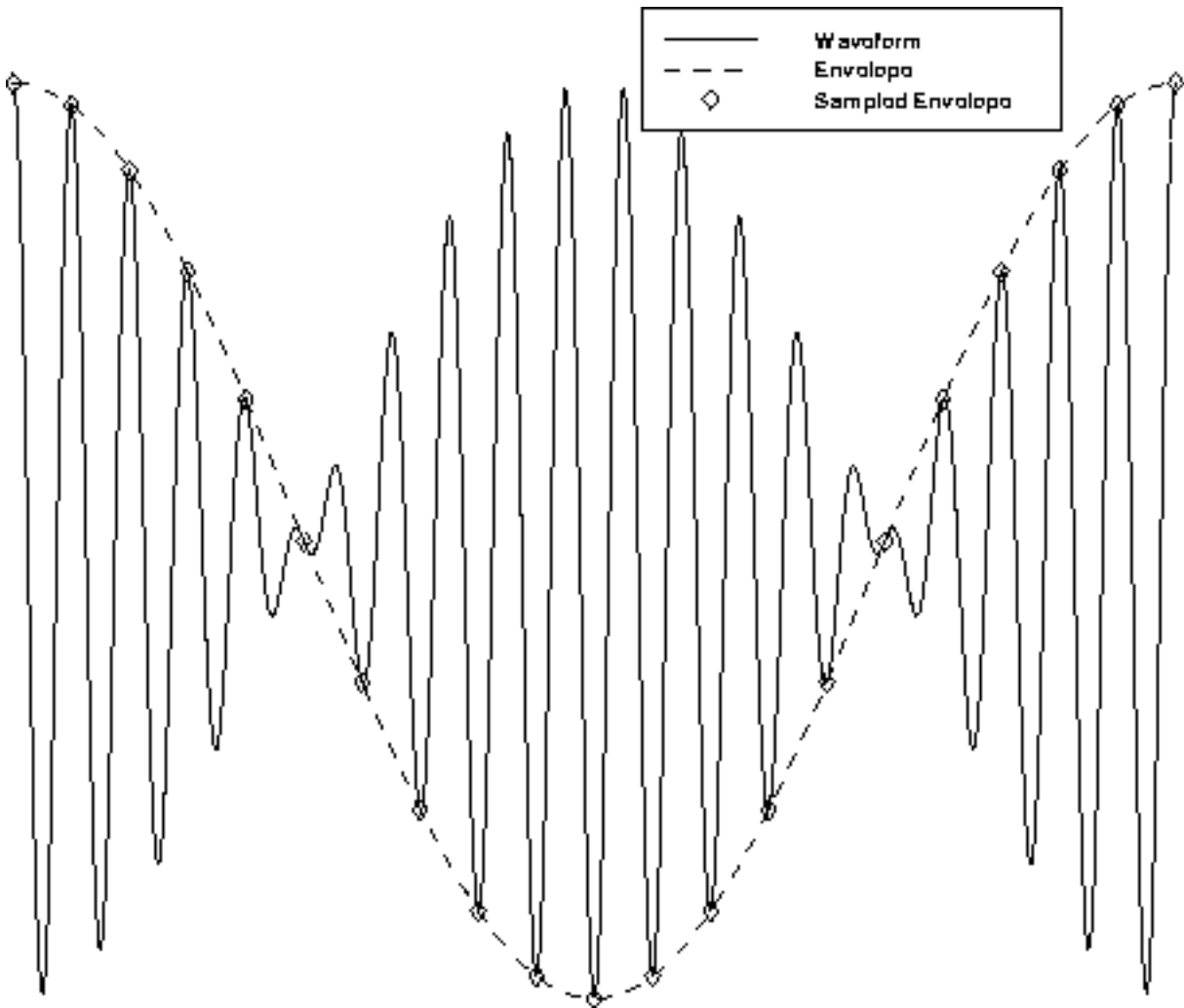
- $Q(v(t)) \in \mathfrak{R}^N$ is typically the vector of sums of capacitor charges at each node
- $I(v(t)) \in \mathfrak{R}^N$ is the vector of sums of resistive currents at each node
- $u(t) \in \mathfrak{R}^N$ is the vector of inputs
- $v(t) \in \mathfrak{R}^N$ is the vector of node voltages
- N is the number of circuit nodes

The MFT algorithm assumes that the circuit is in quasiperiodic steady state, that is, that the signals can be represented using

$$v(t) = \sum_k \sum_l V_{kl} e^{j2\pi(lf_0 + kf_1)t} \quad (3.2)$$

where, for simplicity, there are only two fundamental frequencies, f_0 and f_1 . The signal $v(t)$ is then sampled at one of the fundamental frequencies, f_0 , which is called the clock. This is shown in [Figure F-1](#) on page 821, where sampling a two-fundamental quasi-periodic signal at one of the fundamental frequencies creates a sampled waveform that is one-fundamental quasi-periodic, or simply-periodic. MFT directly finds the solution that when sampled at f_0 is periodic in f_1 .

Figure F-1 Sample Envelope



The sample envelope shown in the figure is the waveform traced out when the signal is sampled with the clock period. The envelope is shown sampled at the peaks but this is not necessary.

The sampled waveform is

$$\bar{v}_n = v(nT_0) = \sum_{k=-\infty}^{\infty} \bar{v}_k e^{j2\pi k f_1 t} \quad (3.3)$$

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

where $T_0 = 1/f_0$. Alternatively, you can write

$$\bar{v}_n = F^{-1}\bar{V} \quad (3.4)$$

which states that \bar{v} is the inverse Fourier transform of \bar{V} .

Recall that v is a solution of the circuit equations, and that \bar{v} is simply v uniformly sampled, so given \bar{v}_n one can compute a subsequent sample point \bar{v}_{n+1} using

$$\bar{v}_{n+1} = \phi(\bar{v}_n, nT_0, (n+1)T_0) \quad (3.5)$$

where

- ϕ is the state transition function for the circuit.
- $\phi(v_0, t_0, t_1)$ is the solution for the circuit equations at t_1 , given that it starts from the initial condition v_0 at t_0 .

Consider the n th sample interval. Then let $x_n = \bar{v}_n$ be the solution at the start of the interval and $y_n = \bar{v}_{n+1} = x_{n+1}$ be the solution at the end. Then (3.5) uses the circuit equations to relate the solution at both ends of the interval,

$$y_n = \phi(\bar{v}_n, nT_0, (n+1)T_0) \quad (3.6)$$

Let $X = Fx$ and $Y = Fy$ (X and Y are the Fourier transforms of x and y). Then, from (3.3) and because $y_n = x_{n+1}$,

$$X_k = e^{-j2\pi k f_1 T_0} Y_k \quad (3.7)$$

or

$$X = D_{T_0} Y \quad (3.8)$$

where D_{T_0} is a diagonal delay matrix with $e^{-j2\pi k f_1 T_0}$ being the k th diagonal element.

Together, (3.6) and (3.8) make up the MFT method.

In practice, $\bar{v} = x$ is band-limited, so only a finite number of harmonics are needed. In addition, if the circuit is driven with one large high-frequency signal at f_0 , which is called the *clock*, and one moderately-sized sinusoid at f_1 , only K harmonics are needed and the method is efficient. With only K harmonics, (3.6) is evaluated over $2K+1$ distinct intervals that

are spread evenly over one period of the lowest beat tone. Therefore, the total simulation time is proportional to the number of harmonics needed to represent the sampled waveform. The simulation time is independent of the period of the lowest-frequency beat tone or the harmonics needed to represent the clock signal.

Equation (3.8), along with the associated Fourier transforms, relate the starting and ending points of the solution of the circuit equations over each interval, and consequently represent a boundary-value constraint on (3.6).

Shooting methods are the most common method for solving boundary-value problems. Their use of transient analysis to solve the circuit equations over an interval brings two important benefits.

- First, transient analysis handles abruptly discontinuous signals efficiently because the timestep shrinks to follow rapid transitions.
- Second, transient analysis easily handles the strongly nonlinear behavior of the circuit as it responds to the large clock signal.

Pdisto and PSS Analyses Compared

Like PSS, Pdisto uses a shooting Newton method. However, instead of doing a single transient integration, each Newton iteration does transient integrations over a number of non-adjacent clock periods. Each of the integrations differs by a phase-shift in each moderate input signal. The number of integrations is determined by the numbers of harmonics of moderate fundamentals that you specify. Given $maxharms = [k_1, k_2, \dots, k_S]$, the total number of integrations is

$$\prod_{s=1}^S (2k_s + 1)$$

Consequently, the efficiency of the algorithm depends significantly on the number of harmonics required to model the responses of moderate fundamentals.

Fortunately, the number of harmonics of the clock does not significantly affect the efficiency of the shooting algorithm. The boundary conditions of a shooting interval are such that the time-domain integrations are consistent with a frequency-domain transformation with a shift of one large-signal period.

As a SpectreRF user, you might need to run PSS to calculate the steady-state operating point of circuits with multiple input frequencies. You can do this by using the greatest common factor of all input frequencies as the PSS beat frequency.

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

When the greatest common factor is relatively close to the clock frequency, such as within a factor of 10, PSS might be preferable to Pdisto for two reasons.

- First each nonlinear iteration would not require excessive integrations of clock period.
- Second, PSS solves a much smaller linear system in each nonlinear iteration.

The size of the linear system resulting from Pdisto analysis is

$$K = \prod_{s=1}^S (2k_s + 1)$$

times as big as that of PSS analysis.

In general,

- If the clock/beat frequency ratio of PSS is smaller than K , PSS might be preferable to Pdisto.
- For circuits operating on wide timescales, such as switched-capacitor filters, the *clock/beat* frequency of PSS can be greater than 1000, and Pdisto is clearly the method of choice.

Another difference between the two analyses is that PSS does not have any restrictions on input sources, whereas Pdisto requires that all non-clock inputs must be sinusoidal.

One limitation of the current Pdisto analysis is that small signal analyses after Pdisto, such as quasi-periodic noise analysis, cannot be performed.

Pdisto analysis also cannot currently be applied to autonomous circuits.

Pdisto and PAC Analyses Compared

Like PAC analysis, the Pdisto analysis calculates responses of a circuit that exhibits frequency translations. However, instead of having small-signal linear behavior, Pdisto models the response as having components of a few harmonics of input-signal frequencies. This permits computing responses to moderately large input signals.

PAC analysis assumes that only the clock is generating harmonics. For example, for a clock frequency f_c , and a small signal frequency f_s , amplitudes of circuit response are generated at $f_s + k_c f_c$, where k_c is bonded by the parameter *harms* in PSS analysis. In contrast, Pdisto also permits non-clock fundamentals to generate harmonics. In the same situation, a

spectrum at frequencies $k_s f_s$ $k_c f_c$ is generated, where k_s and k_c are bonded by the Pdisto parameter *maxharms*.

Unlike PAC, a PSS analysis is not explicitly required before a Pdisto analysis. Pdisto internally runs a PAC analysis to determine initial conditions.

Pdisto Analysis Parameters

While Pdisto analysis inherits most PSS parameters, it adds two new parameters

- *funds*
- *maxharms*

These two parameters replace the PSS parameters, *fund* (or *period*) and *harms*, respectively.

The *funds* parameter accepts a list of names of fundamentals that are present in the sources. These names are specified in the sources by parameter *fundname*. The first fundamental is assumed to be the large signal.

The *maxharms* parameter accepts a list of numbers of harmonics needed for each fundamental.

If you do not list all the fundamental names using the *funds* parameter, the current analysis is skipped. However, if you do not specify *maxharms*, a warning message is issued, and the number of harmonics is defaulted to 1 for each of the fundamentals.

The remaining Pdisto analysis parameters are inherited directly from PSS analysis, and their meanings remain essentially unchanged.

Pdisto analysis expands the role of some PSS parameters. The *maxperiods* parameter that controls the maximum number of shooting iterations for PSS analysis also controls the number of the maximum number of shooting iterations for Pdisto analysis.

Application Examples

Pdisto analysis is particularly efficient for circuits operating on widely separated timescales. This is demonstrated by the following two examples

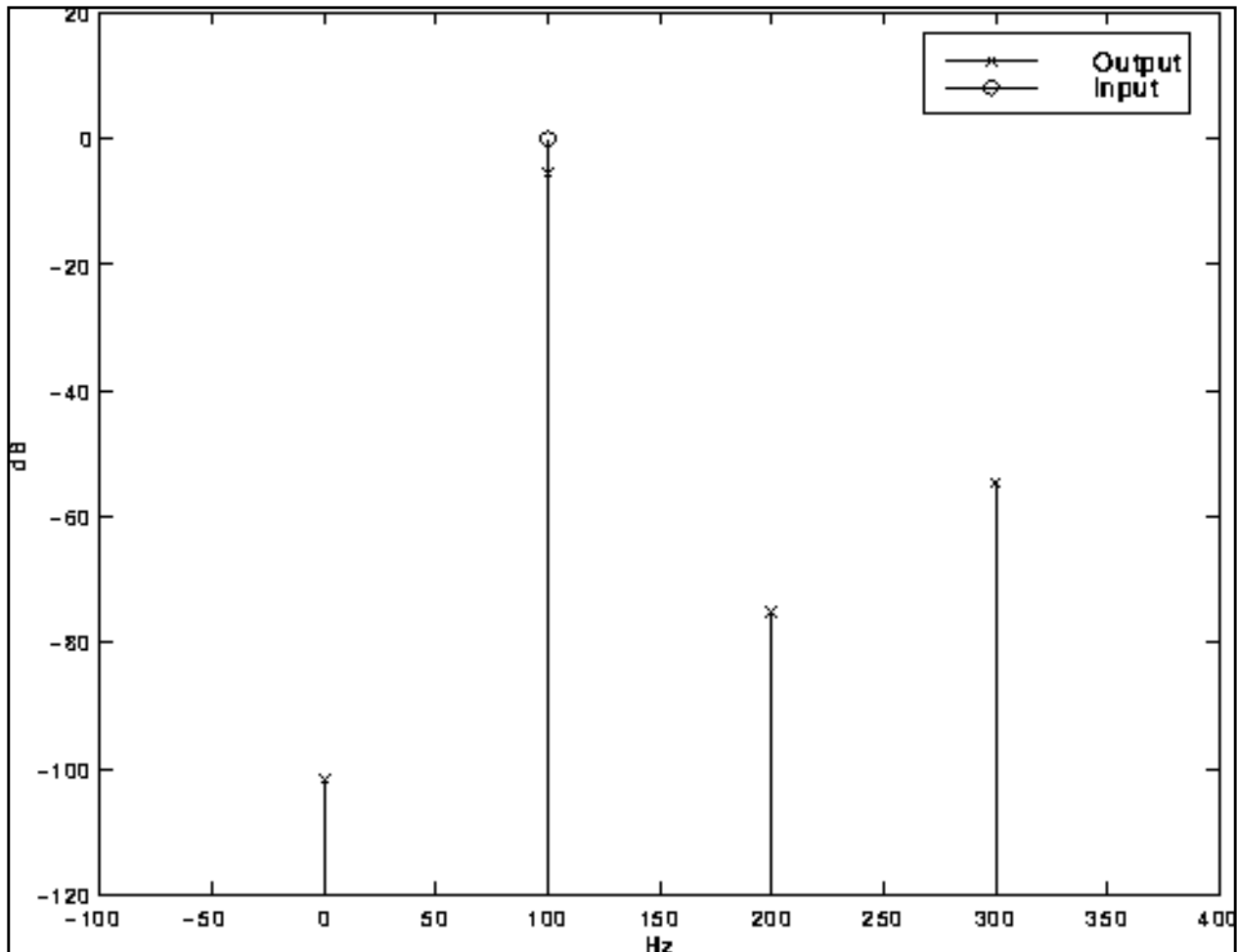
- A switched capacitor filter
- A high-performance receiver

Switched Capacitor Filter Example

The low-pass switched-capacitor filter example has a 4 kHz bandwidth and 238 nodes, resulting in 337 equations. To analyze this circuit, the Pdisto analysis was performed with an 8-phase 100 kHz clock and a 1 V sinusoidal input at 100 Hz.

The 1000-to-1 clock-to-signal ratio makes this circuit difficult for traditional circuit simulators to analyze. Three harmonics were used to model the input signal. The eight-phase clock required about 1250 timepoints for each transient integration. The total number of variables solved by the analysis is $337 \times (2 \times 3 + 1) \times 1250 = 2,948,750$, slightly less than three million. The simulation completes in less than 20 minutes on a Sun UltraSparc1 workstation with 128 Megabyte memory and a 167 MHz CPU clock. A swap file is used because the analysis cannot be finished in core (See [“Memory Management”](#) on page 834 for more information). [Figure F-2](#) on page 827 shows the harmonic distortion.

Figure F-2 Harmonic Distortion of a Switched Capacitor Filter



High-Performance Receiver Example

The high-performance image rejection receiver example consists of a low-noise amplifier, a splitting network, two double-balanced mixers, and two broad-band Hilbert transform output filters combined with a summing network that suppresses the unwanted side-band. A limiter in the LO path controls the amplitude of the LO. It is a rather large RF circuit that contains 167 bipolar transistors and uses 378 nodes. This circuit generated 987 equations in the simulator.

To determine the intermodulation distortion characteristics, the circuit is driven by a 780 MHz LO and two 50 mV closely placed RF inputs, at 840 MHz and 840 MHz+10 KHz, respectively. Three harmonics are used to model each of the RF signals, and 200 time points are used in

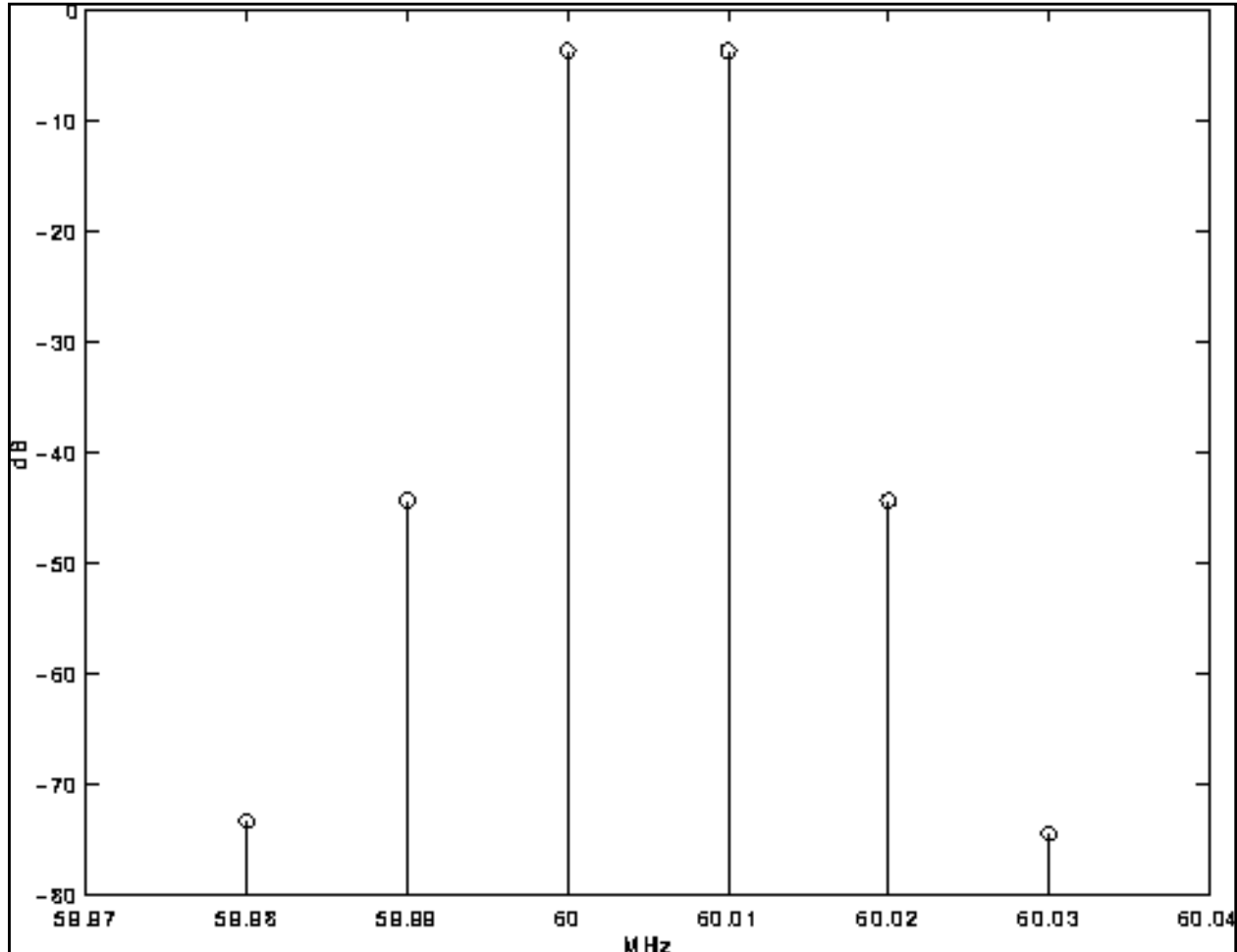
Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

each transient clock-cycle integration, considered to be a conservative accuracy specification for this circuit. As a consequence, $987 \times (2 \times 3 + 1)^2 \times 200 = 9,672,600$, or about ten million unknowns are generated. The simulation requires 55 CPU minutes on a Sun UltraSparc10 workstation with 128 Megabytes of physical memory and a 300 MHz CPU clock. A swap file is used because the analysis cannot be finished in core (See "[Memory Management](#)" on page 834 for more information). [Figure F-3](#) on page 829 shows the 3rd and 5th order distortion products.

To appreciate the efficiency of the MFT method, consider that traditional transient analysis needs at least 80,000 cycles of the LO to compute the distortion, a simulation time of over two days. Additionally, the results might be inaccurate because of the large numerical error accumulated by integrating over so many cycles. In contrast, the MFT method is able to resolve very small signal levels, such as the 5th order distortion products shown in the figure.

Figure F-3 Intermodulation Distortion of a High-Performance Receiver



Running a Pdisto Analysis

The following sections describe how to set up and run a Pdisto analysis. They also present ways to promote convergence.

Picking the Large Fundamental

Your first task is to select the large fundamental, called the clock or the LO. Below are a few guidelines for selecting the large fundamental.

- Choose the one which is not sinusoidal.

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

- Choose the one which causes the most nonlinearity.
- Choose the one which causes the largest response.
- Choose the one which has the highest frequency.

Setting Up Sources

You can specify the clock input using any type of source. However, other fundamentals can only be sinusoidal sources.

In addition to specifying the waveform parameters such as *type*, and *ampl*, you must also use the parameter *fundname* to specify a name for each non-DC source. Each fundamental can be a composition of several input sources with the same name. This is a difference between Pdisto and other analyses.

The following example shows how to set up the sources for the [“Switched Capacitor Filter Example”](#) on page 826.

An eight-phase clock:

```
//*****BEGIN NETLIST*****
.
.
.

// Clocks
Phi1 (phi1 gnd) vsource type=pulse period=2.5us delay=0.25us\ width=1us val0=-
VDD vall=VDD rise=10ns fundname="Clock"
Phi2 (phi2 gnd) vsource type=pulse period=2.5us delay=1.5us\ width=1us val0=-
VDD vall=VDD rise=10ns fundname="Clock"
Phi8 (phi8 gnd) vsource type=pulse period=5.0us delay=1.5us\ width=2.25us val0=-
VDD vall=VDD rise=10ns fundname="Clock"
Phi9 (phi9 gnd) vsource type=pulse period=5.0us delay=1.25us\ width=2.75us
val0=VDD vall=-VDD rise=10ns fundname="Clock"
Phi10 (phi10 gnd) vsource type=pulse period=10.0us delay=3.75us\ width=5.25us
val0=VDD vall=-VDD rise=10ns fundname="Clock"
Phi11 (phi11 gnd) vsource type=pulse period=10.0us delay=4.0us\ width=4.75us
val0=-VDD vall=VDD rise=10ns fundname="Clock"

// Input source
Vin (pin gnd) vsource type=sine freq=100_Hz ampl=1 sinephase=0\
fundname="Input"
// Pdisto Analysis
harmDisto pdisto funds=["Clock" "Input"] maxharms=[3 3] +swapfile="SomeFileName"
//*****END NETLIST*****
```

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

The following example shows how to set up the sources and analysis for the “High-Performance Receiver Example” on page 827.

```
//*****BEGIN NETLIST*****
.
.
.
// LO input
Vlo (8 7) vsource type=sine ampl=400mV freq=780MegHz fundname="LO"
//RF inputs
Prf1 (15 44) port type=sine ampl=50mV freq=840MegHz fundname="F1"
+ ampl2=50mV freq2=840.01MegHz fundname2="F2"
//Analysis
InterModDisto pdisto funds=["LO" "F1" "F2"] maxharms=[5 3 3]
+ swapfile="SomeFileName"

//*****END NETLIST*****
```

Sweeping a Pdisto Analysis

You can combine a Pdisto analysis with a Spectre Sweep analysis to create a powerful tool for a wide variety of applications, such as IP3 and IP5 calculations. For example, you might want to calculate the distortion for input power ranging from -60 dBm to 0 dBm. The netlist for this task is shown below:

```
//*****BEGIN NETLIST*****
.
.
.
parameters inputpower=-60
// Clock
Vlo (clock 0) vsource type=pulse val0=-1 vall=1 period=1n delay=0\ rise=10p
fall=10p width=460.00p fundname="carrier"
// Input port
Prf (input 0) port type=sine r=50.0 num=1.0
+freq=+9.0E+08 dbm=inputpower fundname="RF1"
+freq2=+9.05E+08 dbm2=inputpower fundname2="RF2"
// Analysis
swp sweep param=inputpower start=-60 stop=0 lin=6 {
  distortion pdisto funds=["LO" "RF" "RF2"] maxharms=[5 3 3]
}
```

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

```
//*****END NETLIST*****
```

Always arrange the sweep values so that analyses that converge more easily are performed first. When you sweep Pdisto, it automatically uses the converged steady-state solution of the previous Pdisto analysis as its initial condition and omits the PSS and PAC stage of initial condition calculations. As discussed in [“Convergence Aids”](#) on page 832, this practice can also aid convergence.

Convergence Aids

Normally Pdisto analysis converges with default parameter settings, but occasionally you might need to adjust some parameter settings in order to achieve convergence.

There are two situations where non-convergence can occur

- During the PSS analysis that Pdisto calls to calculate the initial condition
- During the Pdisto iterations

The messages printed on the screen from the simulator identify the problem.

- If the simulation fails before printing the message “Trying PAC for PDISTO initial condition”, the problem is in the PSS analysis.
- If it fails after printing “Starting PDISTO analysis iterations”, the problem occurs during the Pdisto analysis.

Normally, giving a sufficiently large *tstab* or a looser *steadyratio* will resolve convergence problems during the PSS stage. Also see the convergence aids for PSS analysis in [Appendix B, “Running PSS Analysis Effectively”](#).

For convergence problems during the Pdisto iterations, try the following procedures.

- The *steadyratio* parameter guards against false convergence. Its default values, 1.0 for *liberal*, 0.1 for *moderate*, and 0.01 for *conservative*, are derived from the *errpreset* parameter. Tighten *steadyratio* only if you suspect false convergence.

Increase *steadyratio*. Sometimes *steadyratio* must be loosened (for example to *steadyratio*=1) particularly with a tight *reltol* setting. Also loosen *steadyratio* when convergence stagnates. An indication of stagnation is that the *convNorm* value, which you can see on the screen, fluctuates within a certain range and never decreases further.

The convergence tolerance of Pdisto is determined by the product of *steadyratio* and *reltol*. Normally, you do not set *steadyratio* to a value higher than 10.

Affirma RF Simulator (SpectreRF) User Guide

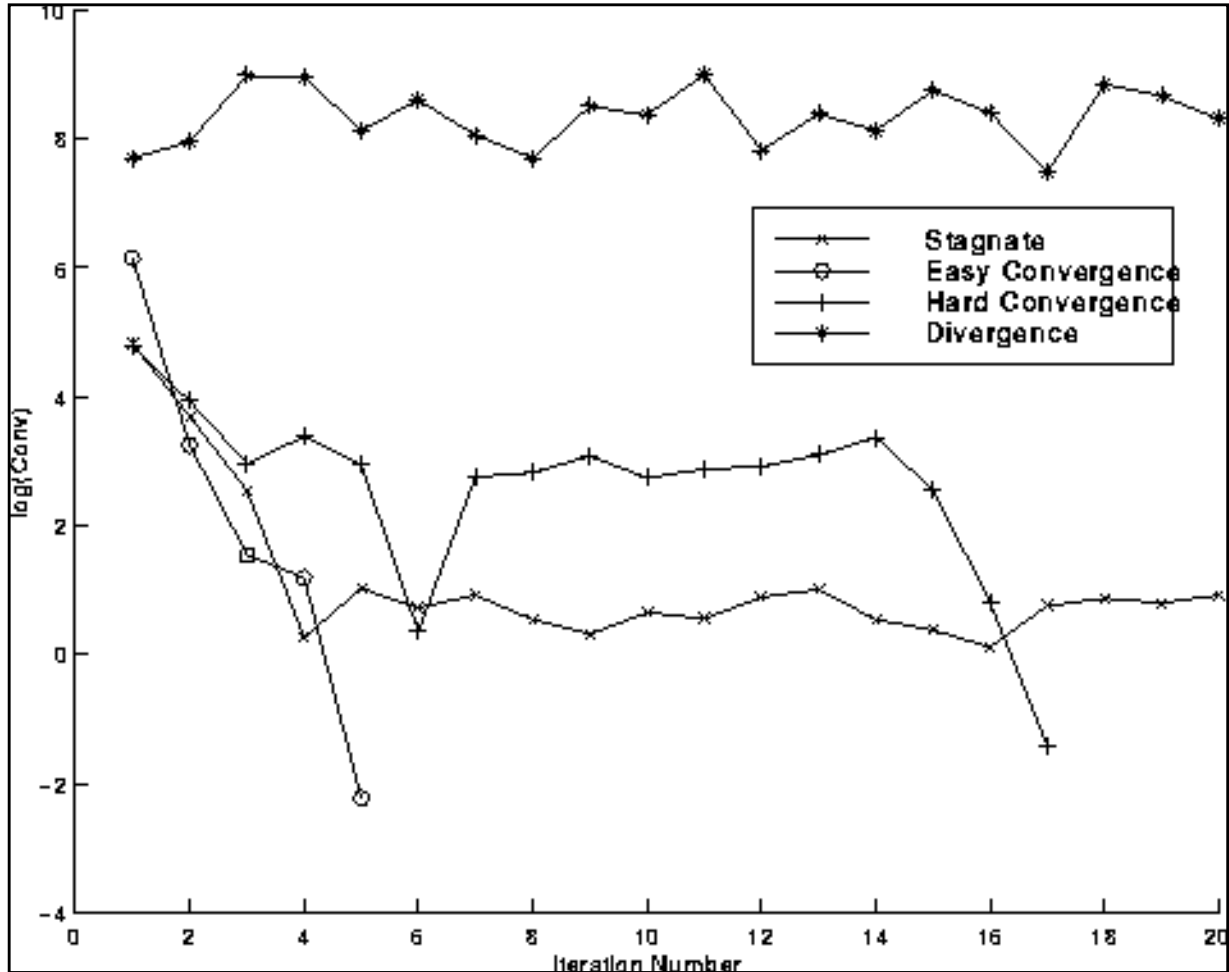
Using Pdisto Analysis Effectively

- Severe trapezoidal rule ringing can prevent convergence. If you suspect trapezoidal rule ringing, use *method=gear2* or *method=gear2only*.
- Avoid using unnecessarily tight *reltol* settings. Excessively tight *reltol* significantly reduces efficiency besides causing convergence problems.
- Do a continuation on a parameter, such as input power, of moderate fundamentals. When the circuit is behaving in a highly nonlinear manner at a certain input power level, the PSS plus PAC approach might not compute a good enough estimate of the initial condition. One effective strategy is to ramp up the input power gradually by carefully arranging a sweep as described in “Sweeping a Pdisto Analysis” on page 831. Because it is usually much easier to achieve convergence at a low power input level where the circuit behaves in a more linear manner, start with a low input power. Once the simulation converges, save the steady-state solution as the initial condition for the next input power level. This process repeats automatically as Pdisto is swept. You can achieve convergence at the desired input power level if the sweep steps are sufficiently small.
- In general, avoid using an excessively high number of harmonics for the moderate fundamentals. This practice lengthens the simulation time and uses a lot of memory. However, if the number of harmonics is not high enough for a particular fundamental to adequately model its nonlinear effects, convergence problems might occur.

An important indication of convergence or divergence is the *Conv* value printed to the screen. There are a few typical scenarios shown in [Figure F-4](#) on page 834.

- For most Pdisto runs, the *Easy Convergence* scenario occurs.
- A few simulations follow the *Hard Convergence* scenario.
- If Pdisto iterations *Stagnate* (*Conv* value fluctuates close to but above 1.0), loosen *steadyratio*. Loosening *steadyratio* will solve the stagnation problem.
- If the iterations show *Divergence*, you usually must improve the initial condition. Do a continuation on input power level as described within the preceding list.

Figure F-4 Pdisto Convergence Scenarios



Memory Management

Pdisto is a memory intensive analysis. If Pdisto cannot be finished in core with real physical memory, use a swap file residing on a local disk. The simulator manages swapping much more efficiently than the operating system. The examples found in [“Setting Up Sources”](#) on page 830 depict how to set up the *swapfile* parameter. Typically, 80% CPU utilization is achieved.

Dealing with Sub-Harmonics

One advantage of Pdisto analysis over PSS is that you need only provide a name for each fundamental frequency. The actual beat frequency value associated with the name is calculated automatically by the simulator. For each unique fundamental name, the simulator first finds all the frequencies associated with it. Then the greatest common factor is calculated among these frequencies, which is used as the beat frequency associated with the fundamental name.

However, if the fundamental frequency has sub-harmonics (circuit responses at some fraction of a driven frequency, typically 1/2 or 1/3), as with a divider, for example, the simulator currently cannot detect them. As a workaround, add a dummy source that tells the simulator of the existence of a sub-harmonic associated with a fundamental frequency.

Understanding the Narration from the Pdisto Analysis

The following example presents information that is typically printed to the screen during a Pdisto analysis run.

```
.  
. .  
. .  
  
Fundamental LO:  period = 1 ns, freq = 1 GHz.  
Fundamental RF:  period = 2 ns, freq = 500 MHz.  
Fundamental RF2: period = 1.96078 ns, freq = 510 MHz.  
  
*****  
Trying PSS for PDISTO initial condition.  
Trying PAC for PDISTO initial condition.  
Starting PDISTO analysis iterations.  
Iter = 1, convNorm = 33.9, maximum dI(Vin:p) = 44.6082 uA,  
      took 1.48333 s.  
Iter = 2, convNorm = 9.94, maximum dI(Vlob:p) = 31.0627 uA,  
      took 1.38333 s.  
Iter = 3, convNorm = 8.37, maximum dI(Vlob:p) = 11.6634 uA,  
      took 1.4 s.  
Iter = 4, convNorm = 15.1, maximum dI(Vin:p) = 5.88558 uA,  
      took 1.51667 s.  
Iter = 5, convNorm = 7.77, maximum dI(Vlob:p) = 13.6397 uA,  
      took 1.63333 s.  
Iter = 6, convNorm = 52.1e-03, maximum dI(Vcc:p) = 61.1425 nA,  
      took 1.51667 s.
```

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

```
PDISTO solution has converged. Starting spectrum calculation.
```

```
.  
. .  
. . .
```

The simulator first confirms the fundamental names that have been read and their beat frequencies. For the preceding example, the circuit has a 1 GHz fundamental named *LO* as the clock, and two fundamentals named *RF* and *RF2*, respectively, as inputs. Their associated frequencies are 500 MHz and 510 MHz, respectively.

Before starting the PSS analysis to determine initial condition calculations, the following messages is printed.

```
Trying PSS for PDisto initial condition.
```

After the PSS is finished, the following message is printed. This tells you that a PAC analysis is started to perform further initial condition calculations.

```
Trying PAC for PDISTO initial condition.
```

Once the initial condition is obtained, Pdisto prints the following and begins Pdisto nonlinear iterations.

```
Starting PDISTO analysis iterations.
```

At the end of each iteration, a few pieces of information such as the iteration number, its convergence norm, which node had the maximum deviation, and the amount of CPU time spent in the iteration, are printed.

Occasionally, you might see warning messages such as the following

```
Minimum time step used. Solution might be in error.
```

or

```
Junction current exceeds 'imelt'. The results computed by Spectre are now incorrect because the junction current model has been linearized.
```

You can ignore these warning messages if they appear in the early stage of Pdisto iterations or in the initial condition calculations (PSS). They might be caused by bad starting integration conditions and do not affect the final solution. However, if they appear in the final iteration, the solution might be in error.

References

[1] A. ALLGOWER AND K. GEORG, *Numerical Continuation Methods*, Springer-Verlag, New York, 1990.

Affirma RF Simulator (SpectreRF) User Guide

Using Pdisto Analysis Effectively

- [2] L. O. CHUA AND A. USHIDA, *Algorithms for computing almost periodic steady-state response of nonlinear systems to multiple input frequencies*, IEEE Trans. Circuits and Systems, 28 (1981), pp. 953-971.
- [3] D. FENG, J. PHILLIPS, K. NABORS, K. KUNDERT, AND J. WHITE, *Efficient computation of quasi-periodic circuit operating conditions via a mixed frequency/time approach*. Submitted to Proceedings of the 36th Design Automation Conference, June 1999.
- [4] K. KUNDERT, J. WHITE, AND A. SANGIOVANNI-VINCENTELLI, *A mixed frequency-time approach for distortion analysis of switching filter circuits*, IEEE J. Solid State Circuits, 24 (1989), pp. 443-451.
- [5] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, Academic Press, second ed., 1985.
- [6] Y. SAAD, *Iterative methods for sparse linear systems*, PWS Publishing Company, 1996.
- [7] R. TELICHEVESKY, K. S. KUNDERT, AND J. K. WHITE, *Efficient steady-state analysis based on matrix-free Krylov-subspace methods*, in Proceedings of the 1995 Design Automation Conference, June 1995.
- [8] _____, *Efficient AC and noise analysis of two-tone RF circuits*, in Proceedings of the 1996 Design Automation Conference, June 1996.

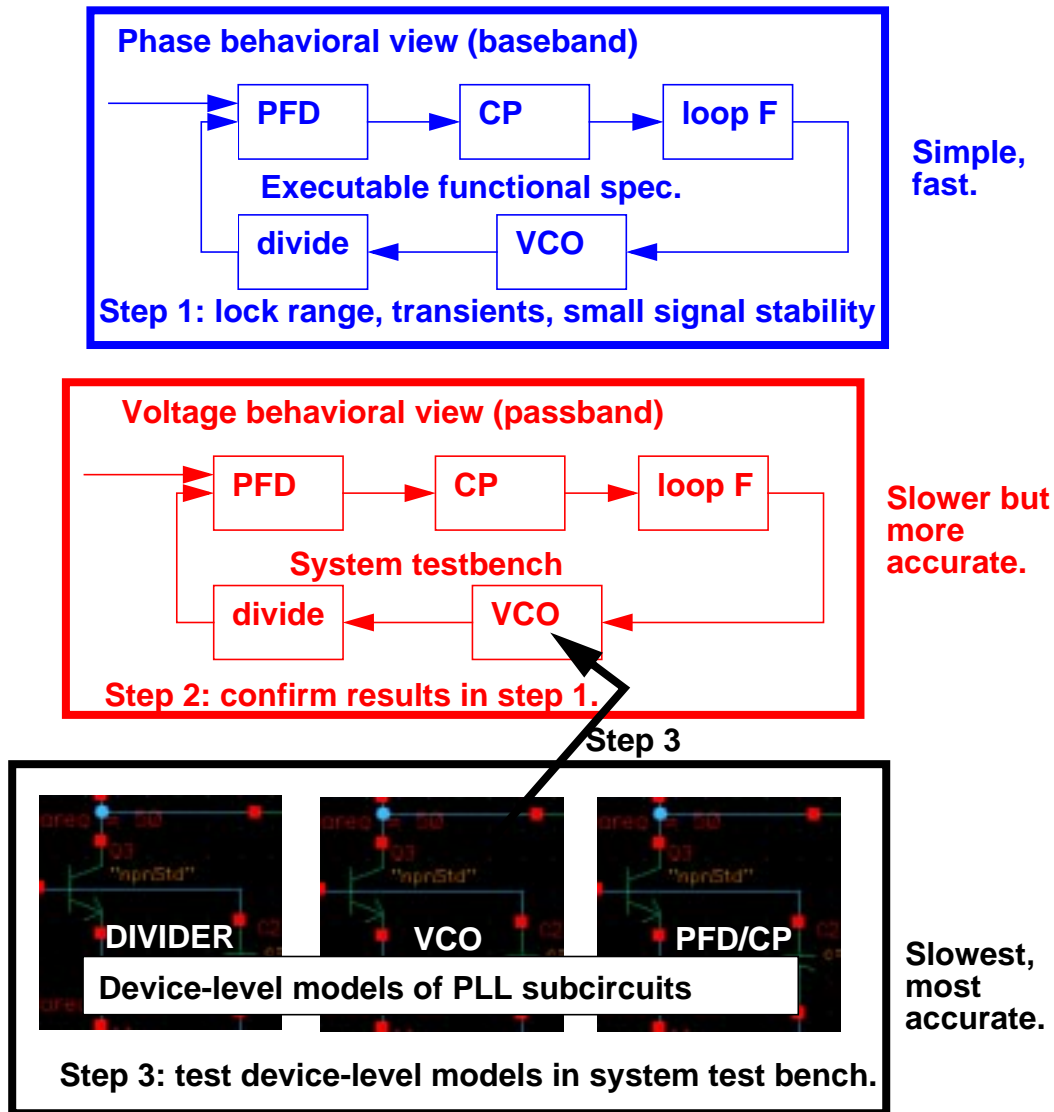
An Introduction to the PLL library

The models in the new phaselock loop (PLL) library support top-down design of PLLs. [Figure 1](#) shows the three steps of the design flow. This paper describes the first step in detail, but all three steps are described briefly.

The first step in Figure 1 is to develop an executable specification. The executable specification is an arrangement of fast behavioral models. The executable specification permits fast architectural studies and separates specification and implementation issues. The executable specification contains baseband models [1¹,2,3,4,5,6,7,8,9]. These baseband models suppress clocks and RF/IF carriers. Some literature refers to PLL baseband models as “relative phase” or “phase-domain” models [2]. This paper uses the latter term. Phase-domain PLL models are exceptionally fast, capture the important non-linear mechanisms, and can be linearized directly for AC analysis.

1. This reference uses the terms “baseband” and “bandpass” explicitly.

Figure G-1 PLL top-down design flow



The second step is to translate the executable specification into a system testbench. The system testbench, unlike the executable specification, is composed of passband models [1]. This paper refers to passband models as voltage-domain models because they simulate voltages you can observe in a laboratory.

Comparing voltage- and phase-domain voltage-controlled oscillator (VCO) models highlights the difference between the two models. The output of a voltage-domain VCO model is a clock voltage, a periodic signal. The output of a phase-domain VCO model is a voltage numerically

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

equal to phase. If you unwrap the VCO phase, in steady state, it ramps up indefinitely. Unwrapped phase is not periodic. Voltage-domain models describe non-linear effects related to the shapes of the actual RF waveforms. Such waveform effects include spurs and harmonic locking¹. Phase-domain models do not simulate waveform effects. The system testbench is more accurate than the executable specification, but it is still behavioral. Equipped only with behavioral voltage-domain models, the testbench does not simulate device-level effects associated with specific implementations. Examples of such implementation effects are interstage loading, improper bias, and device parasitics.

The last step in [Figure 1](#) is to gradually replace the behavioral models in the system testbench with device-level models, one or two blocks at a time. Device-level models check for the previously mentioned implementation problems. The entire PLL is simulated at the device-level only as a final verification step because such simulations are very lengthy.

The Models in the PLL library

The new PLL library includes the following phase-domain models:

- Analog multiplier phase detector
- XOR phase detector with bipolar output²
- Three-state digital phase frequency detector (PFD)
- Charge pump (current source version)
- VCO tuning curve (analytic and tabular versions)
- Frequency divider
- Lock indicator

Introduction

The primary system-level specifications captured by this first set of phase-domain models are acquisition time, lock and capture ranges [12], and phase margin. The PFD model also simulates backlash³ [8].

-
1. “Harmonic locking” is when the PLL locks on to a harmonic of the reference.
 2. The XOR phase detector is not explicitly discussed in this application note because it is very similar to the analog multiplier phase detector. The only difference is that the duty cycle-phase error transfer curve is triangular instead of sinusoidal.
 3. Backlash is sometimes called “deadband” effect. It is a limit cycle caused by the phase-frequency detector’s inability to linearly reduce its output pulse width to zero as phase error goes to zero.

The remainder of this paper is divided into two main sections. The first section introduces phase-domain modeling, describes a feature included to prevent DC convergence problems, and then shows you some examples of using phase-domain models. The second section explains how to assemble a more complex PLL and discusses an example. The examples are introductory and are not a comprehensive discussion of all applications of phase-domain models.

Phase-Domain Model of a Simple PLL

Description

This PLL example, which is built around the simplest phase detector in the new library, introduces the fundamentals of phase-domain modeling. [Figure 2](#) shows a voltage-domain model of the example and also some selected waveforms. The phase detector in this case is an ideal analog multiplier. [Figure 3](#) shows the equivalent phase-domain model. The phase-domain model is based on the trigonometric identity below:

$\sin(\theta_1) \cos(\theta_2) = (1/2) [\sin(\theta_1 + \theta_2) + \sin(\theta_1 - \theta_2)]$, which after filtering is approximately $(1/2) \sin(\theta_1 - \theta_2)$.

Note: $\theta_1 + \theta_2 = (\omega_1 + \omega_2) \cdot t$, and $\omega_1 + \omega_2$ usually lies far beyond the filter corner frequency.

Affirma RF Simulator (SpectreRF) User Guide
An Introduction to the PLL library

Figure G-2 Voltage domain model

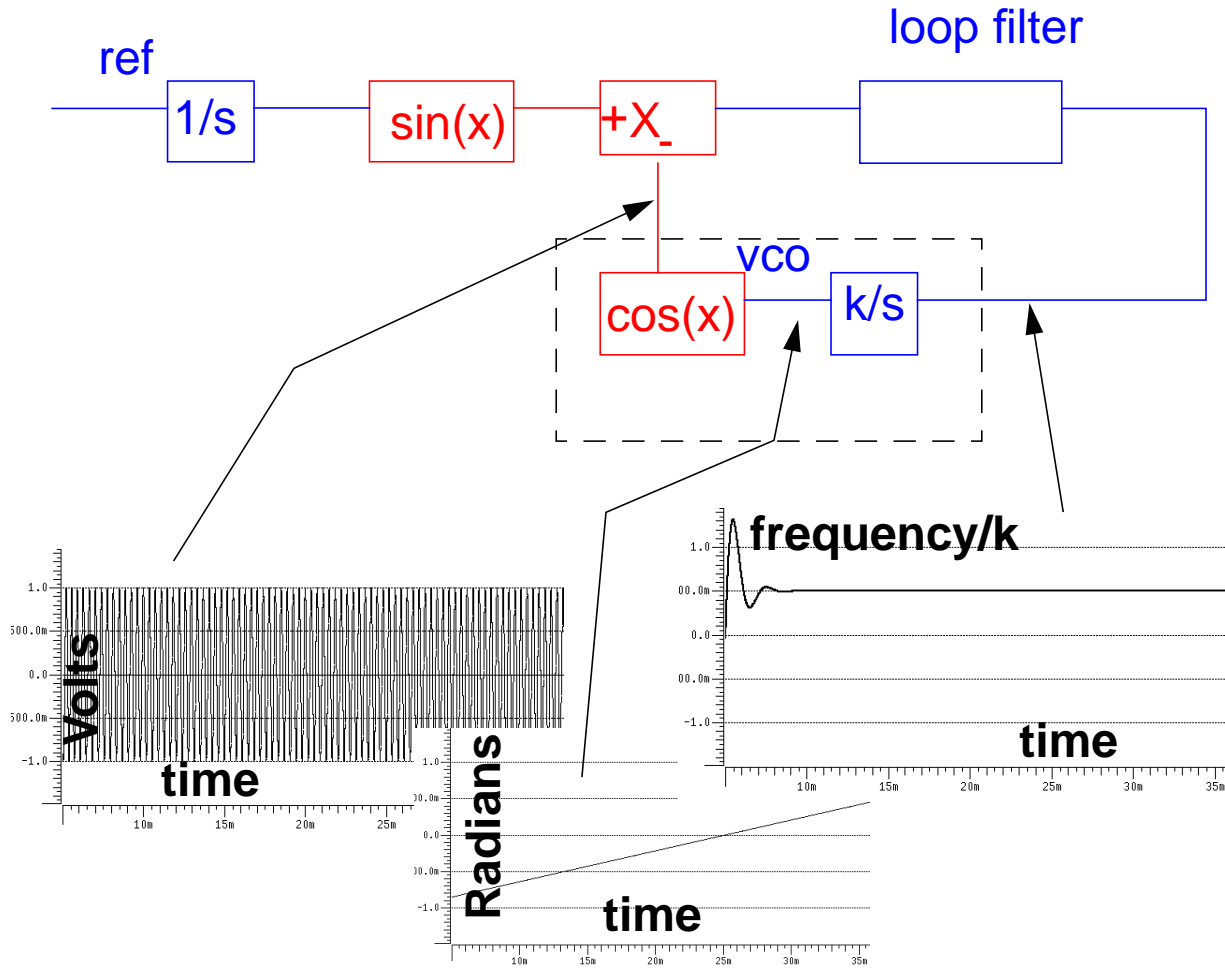
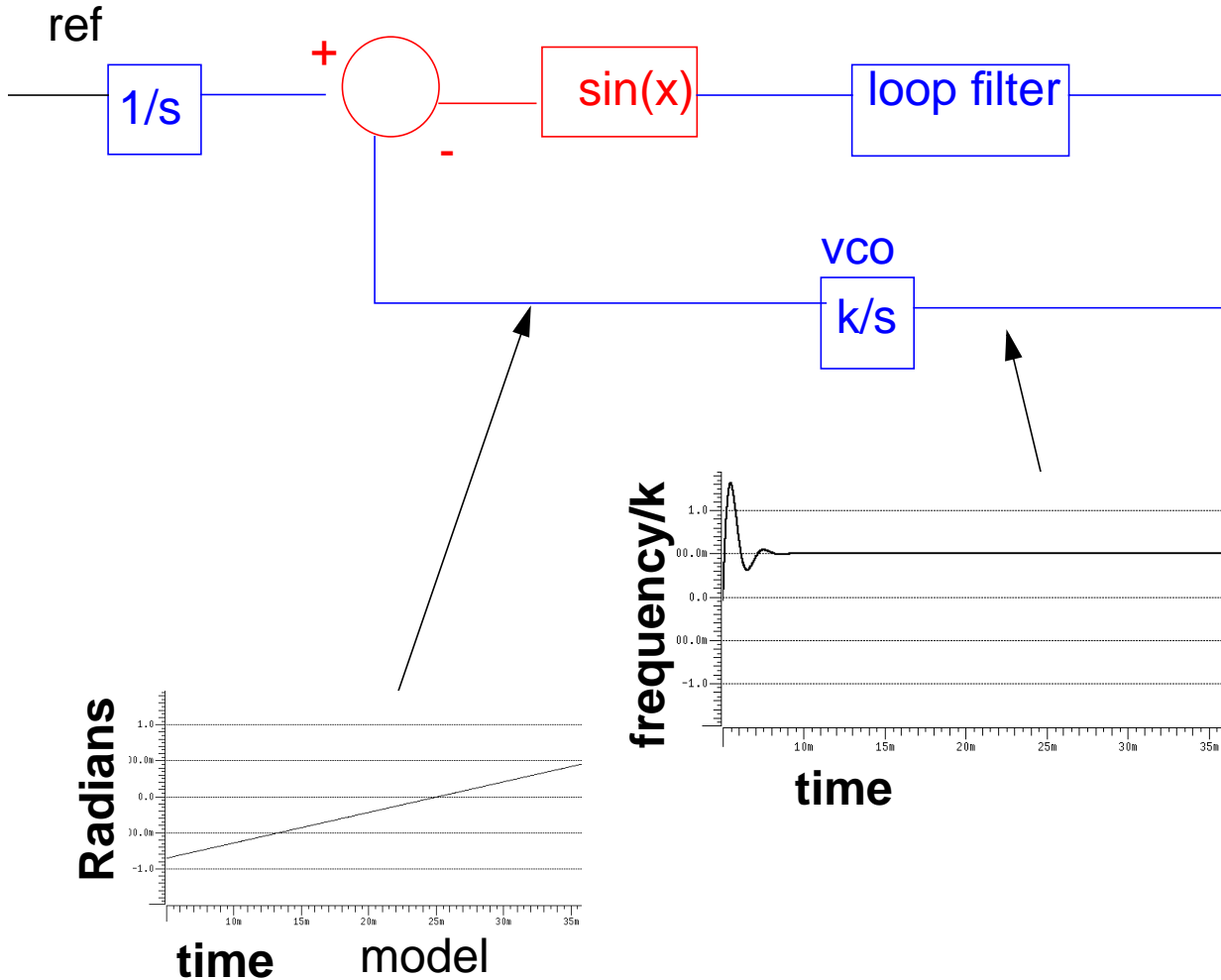
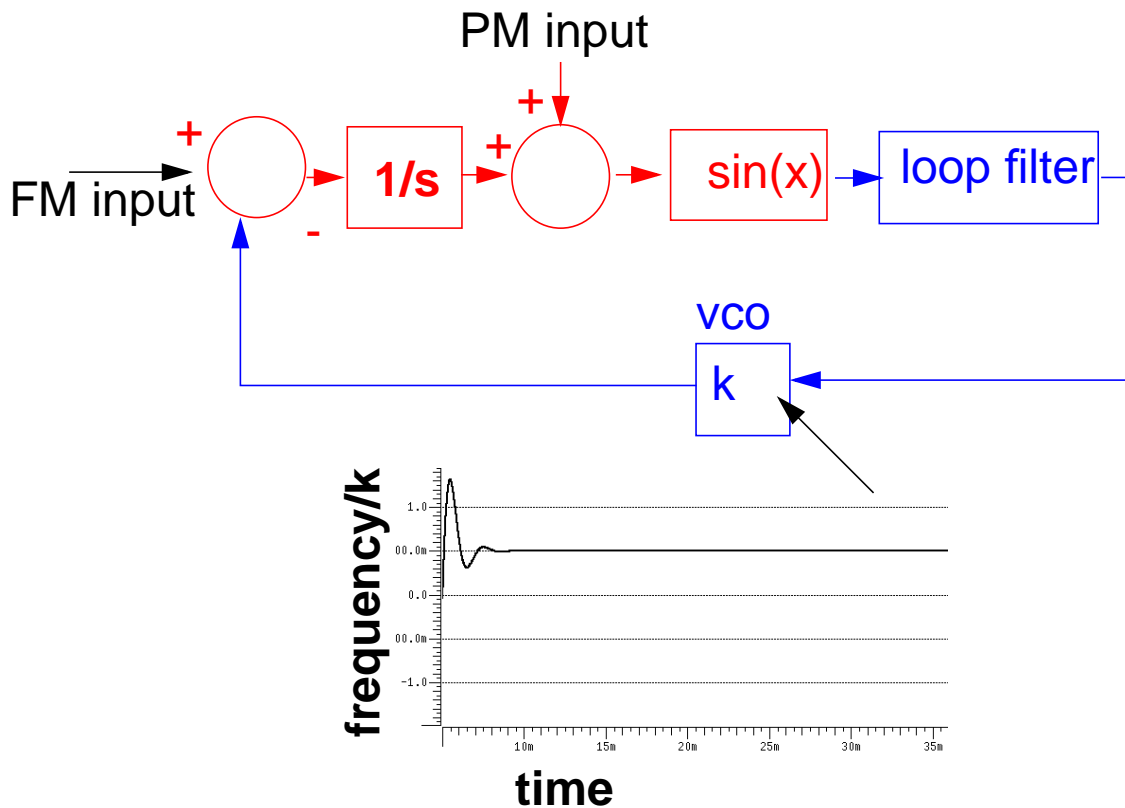


Figure G-3 Phase domain model



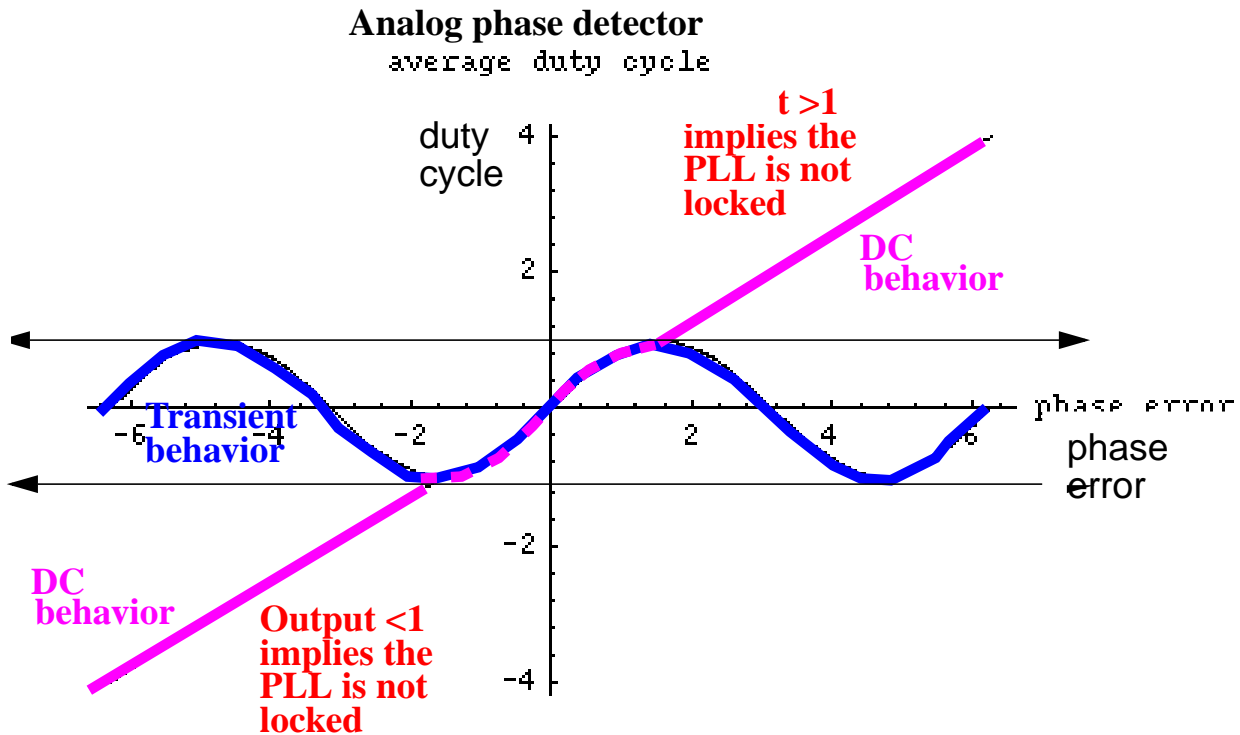
The phase and frequency waveforms from the phase-domain model match their voltage-domain counterparts, but the simulation runs faster because the oscillatory waveform is not explicitly simulated. Combining the integrators, as shown in [Figure 4](#), eliminates the integrator outside the feedback loop that might cause problems if you forget to specify an initial condition. Combining the integrators is also necessary if you build phase-domain models of phase-frequency detectors because the non-linearity in that case has memory (hysteresis).

Figure G-4 A more practical phase domain model



The models in [Figures 3](#) and [4](#) can fail to achieve DC convergence because the phase detector model either has no DC operating point or has an infinite number of them. The sinusoidal function in [Figure 5](#) is the phase detector transfer curve. The phase detector is the only non-linear element in this PLL model. For reasons associated with phase-frequency detectors, the phase detector output is called the *duty cycle*.

Figure G-5 Phase detector transfer curves



If the required duty cycle lies outside $[-1, 1]$, the loop is not locked in steady state. If the required duty cycle lies within $[-1, 1]$, there are an infinite number of possible phase errors. In either case, a Spectre simulation might not converge. Verilog-A's ability to perform different tasks for different analyses provides an elegant solution to the DC convergence problems and a quick way to map out lock range. Lock range is the range of input frequencies for which the PLL can maintain lock. (Some literature refers to lock range as hold-in range [8].) The phase detector model uses the monotonic transfer curve for DC analysis and the true periodic transfer curve for transient analysis. The two transfer curves coincide when the phase error lies in the interval $[-\pi/2, \pi/2]$. If the required duty cycle lies within $[-1, 1]$, the monotonic transfer curve forces the steady-state phase error to the interval $[-\pi/2, \pi/2]$, where the two curves coincide. The equilibrium point is *open-loop-stable*, meaning that at DC the loop gain is a positive real number. This is true because the slope of the transfer curve is positive over $[-\pi/2, \pi/2]$. The Nyquist stability criterion is therefore easier to apply. The DC analysis is general enough because only the phase error modulo 2π is of interest, and you usually care only about the open-loop-stable operating points. When the loop is not locked, the DC analysis computes a duty cycle with a magnitude greater than one. A duty cycle greater than one is clearly incorrect, but it is much easier to interpret than a convergence

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

error. DC duty cycle is a lock indicator which can be used in a parametric DC analysis to sweep out lock range.

Example 1: A dynamic test for capture range and lock range (example_analog_PD in pllLib).

Capture range is the range of input frequencies that the PLL can acquire from an unlocked state. Since acquisition of frequencies near the edge of the capture range involves a pull-in mechanism [1-12], measuring the capture range requires a transient analysis. You can measure capture and lock ranges by slowly sweeping the input frequency and observing the frequency at which the duty cycle begins and ends a long ramp [12]. You must skip the DC analysis to observe the capture limits. [Figure 6](#) plots the VCO control voltage against the input frequency voltage. The input frequency first ramps up and then down. A buffered auxiliary circuit responds to the duty cycle and adds 2.5 volts when the input frequency changes direction. This technique makes the plot easier to read because the forward and reverse sweeps occupy different parts of the vertical scale. In this example, lock range is from 1.36Khz to 3.4Khz, and the capture range is from 1.8Khz to 3Khz.

[Figure 7](#) compares VCO control voltages in the forward sweep when computed with voltage- or phase-domain models. The models produce similar results. In this example (2.5Khz center frequency), the phase-domain model is only about 20 times faster than the voltage-domain model.

Figure G-6 Lock range and capture range

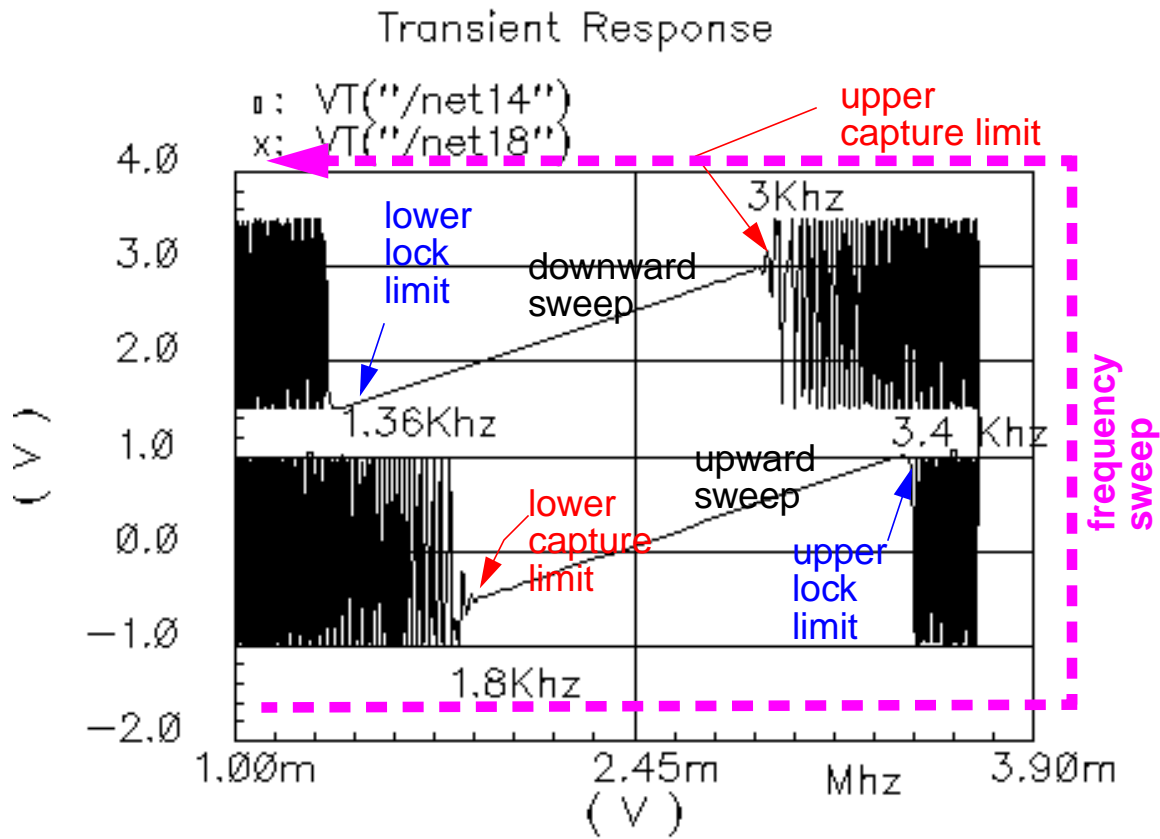
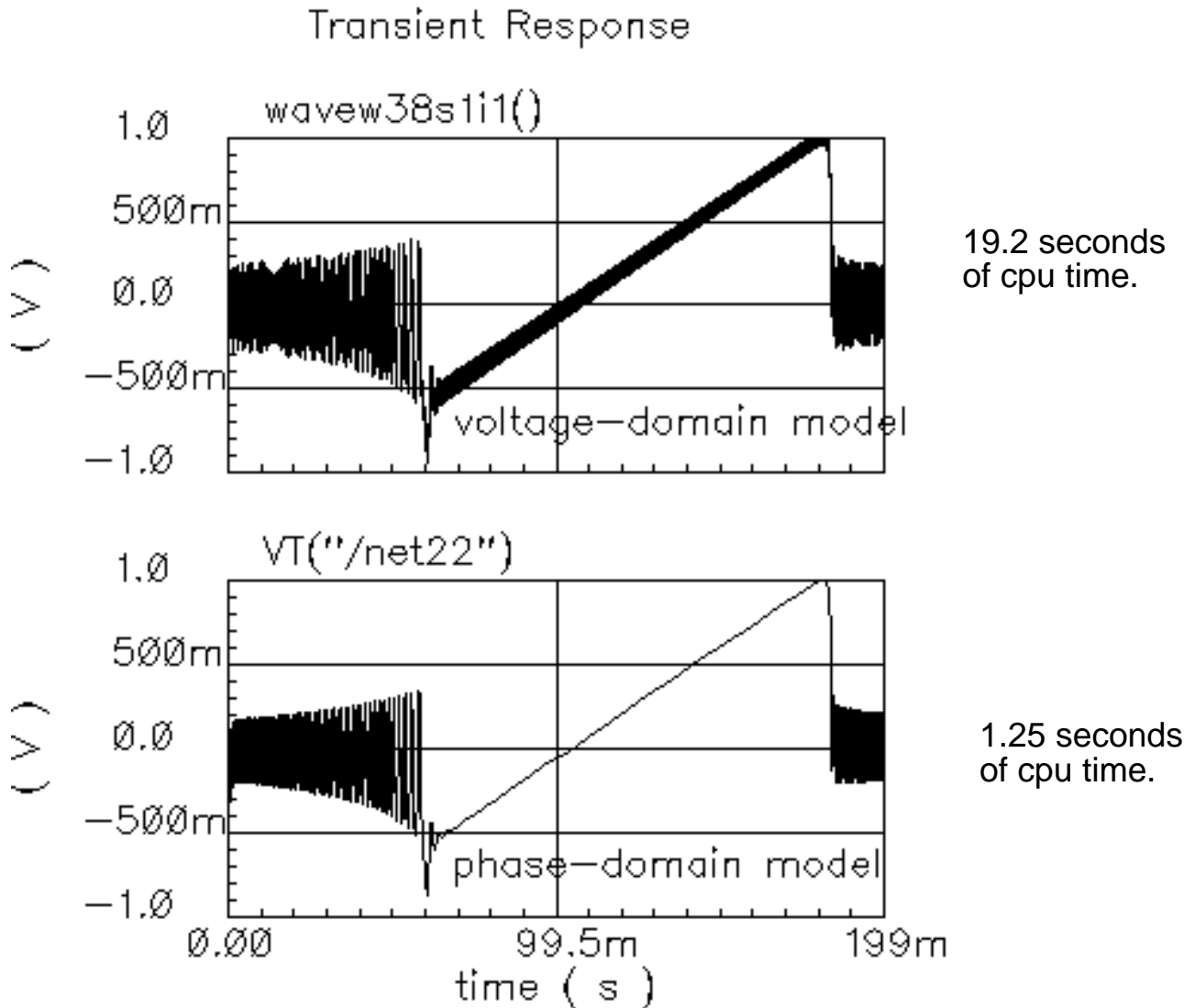


Figure G-7 VCO control voltage response to a ramping PLL reference frequency as

computed with voltage- and phase-domain models



Example 2: Loop gain measurement (example_loop_gain in pllLib).

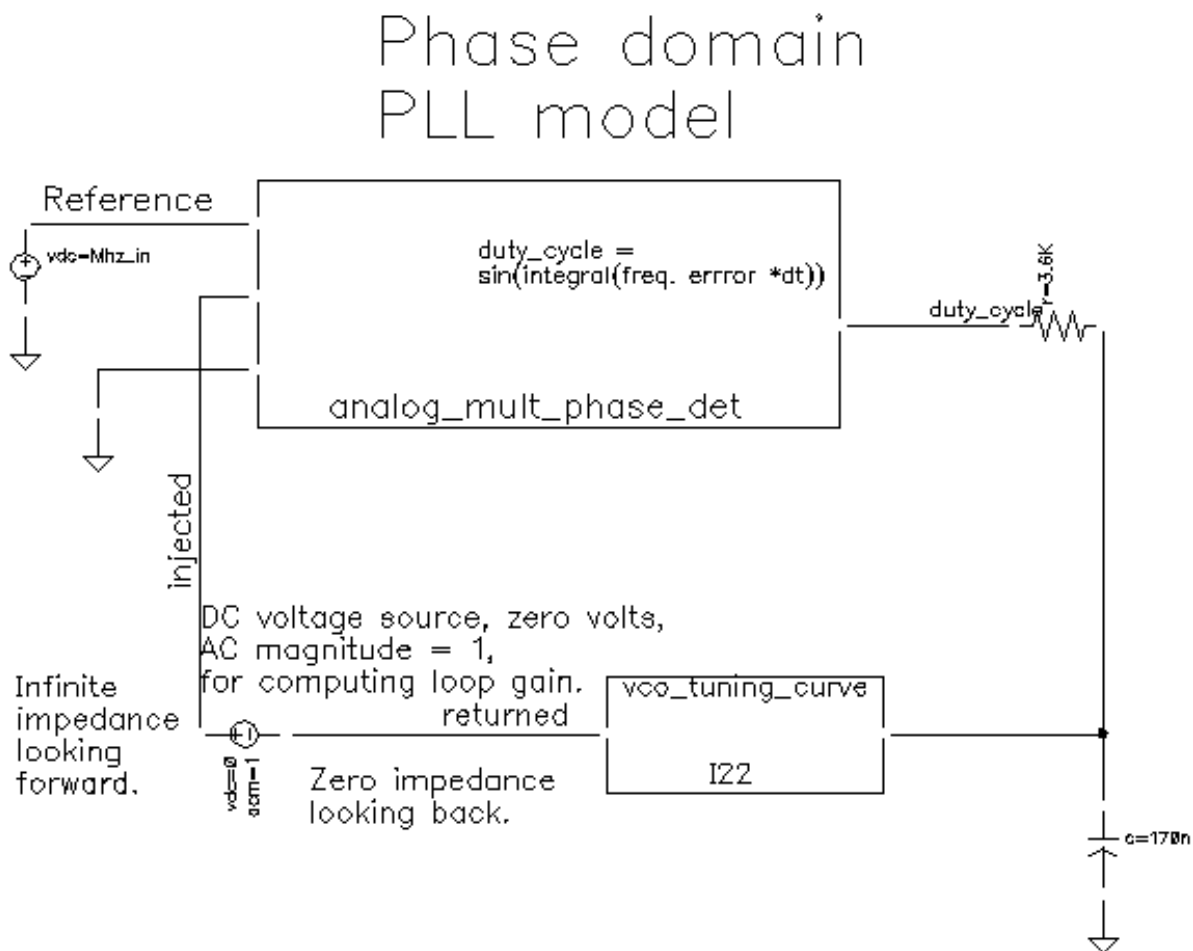
Spectre cannot perform a useful AC analysis on a voltage-domain model because by design, a voltage-domain PLL model has no DC operating point. However, because Spectre linearizes phase-domain models about phase error, and phase error is a meaningful DC quantity, subsequent AC analyses are valid. This example describes how to compute loop gain with a phase-domain model. Figure 8 shows an Affirma™ analog circuit design environment version of the model in Figure 4. The model in Figure 8 includes a voltage source inserted after the VCO. The DC voltage is zero volts, and the AC magnitude is 1 volt.

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

The new voltage source inserts a test signal without changing the DC operating point. You must insert this source at a point where the impedance looking back is much smaller than the impedance looking forward. The accuracy of the resulting loop gain computation depends on how well this condition is met.

Figure G-8 Set up for loop gain measurement



To compute the loop gain, do the following:

1. Set up the AC and DC analyses. The zero-voltage *vdc* source must be the only source with a non-zero AC magnitude. Save the DC data so you can annotate the schematic with DC node voltages.

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

2. Run the analysis.
3. Choose *Results – Annotate – DC Node Voltages* to display the node voltages on the schematic. Make sure the DC duty cycle lies in $[-1, 1]$. If it falls outside this interval the loop is not locked and the AC analysis is invalid.
4. Choose *Results – Direct Plot – AC Magnitude*. Then select the returned signal followed by the injected signal (as labeled in Figure 8) and press the escape button. The Waveform window now shows two curves.
5. In the Waveform window, choose *Tools – Calculator* to bring up the Waveform Calculator.
6. Click on the *wave* button and select the phase. Subtract 180 from the phase waveform and plot the result. You may want to delete the original phase waveform now.
7. In the waveform display window, choose *Curves – Edit Curves*, then select the shifted (by 180 deg) phase, then change "Y axis" to 2, and OK the form. This produces Bode plots of the loop gain. You can change to the display to *Strip* to produce a display like the one in [Figure 9](#).
8. To generate a Nichols chart (dB versus degrees) from which you can pick off phase and gain margins, choose *Axes – X Axes* then select the phase.
9. Choose *Curves – Edit* and then select the magnitude. Change the Scale to dB20 and OK the form. You now have a Nichols chart like the one in [Figure 10](#). The phase margin is 30 degrees.
10. To compute phase margin directly do the following:
 - In the Waveform Calculator, click on *vf*.
 - In the Schematic window, click first on the return node and then on the injected node.
 - In the Waveform Calculator, click on the *divide* button.
 - In the Waveform Calculator Special Functions menu, choose *phase margin* followed by *print*.
 - Add 180 degrees to the expression in the waveform calculator then choose *print* from the Special Functions menu.

The Results Display Window displays the phase margin.

Figure G-9 Bode plots, magnitude and phase of the loop gain

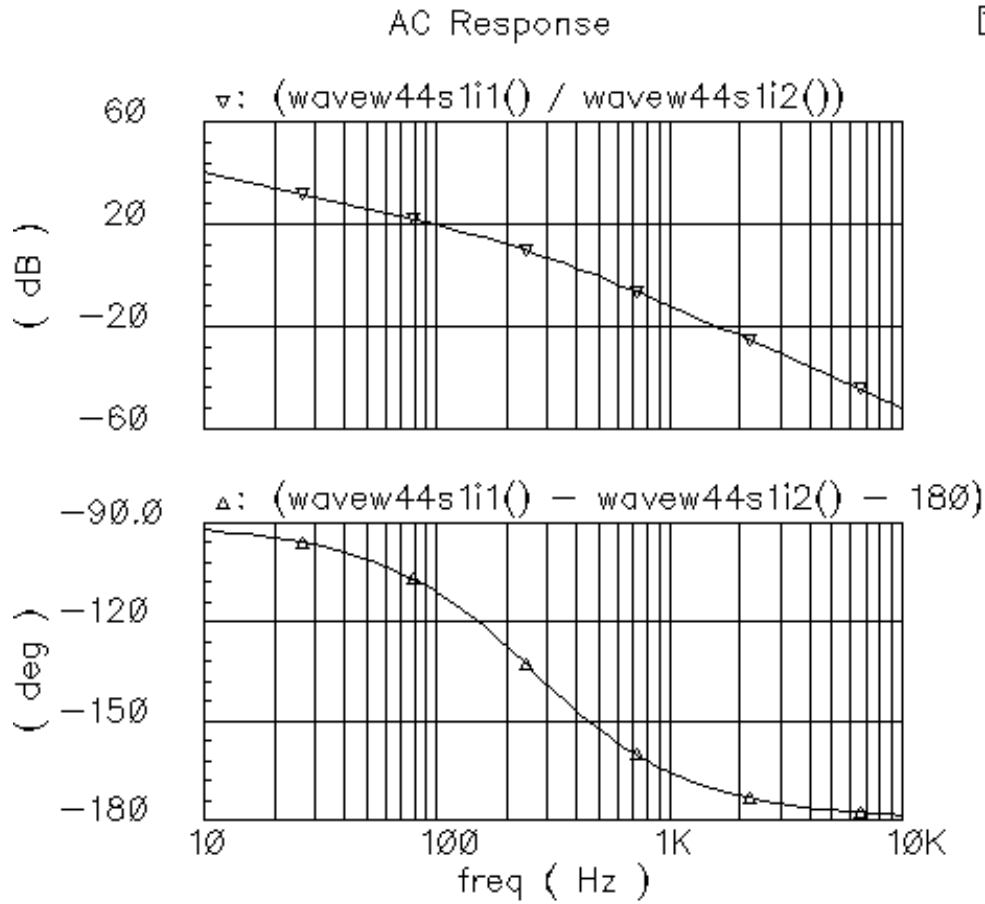
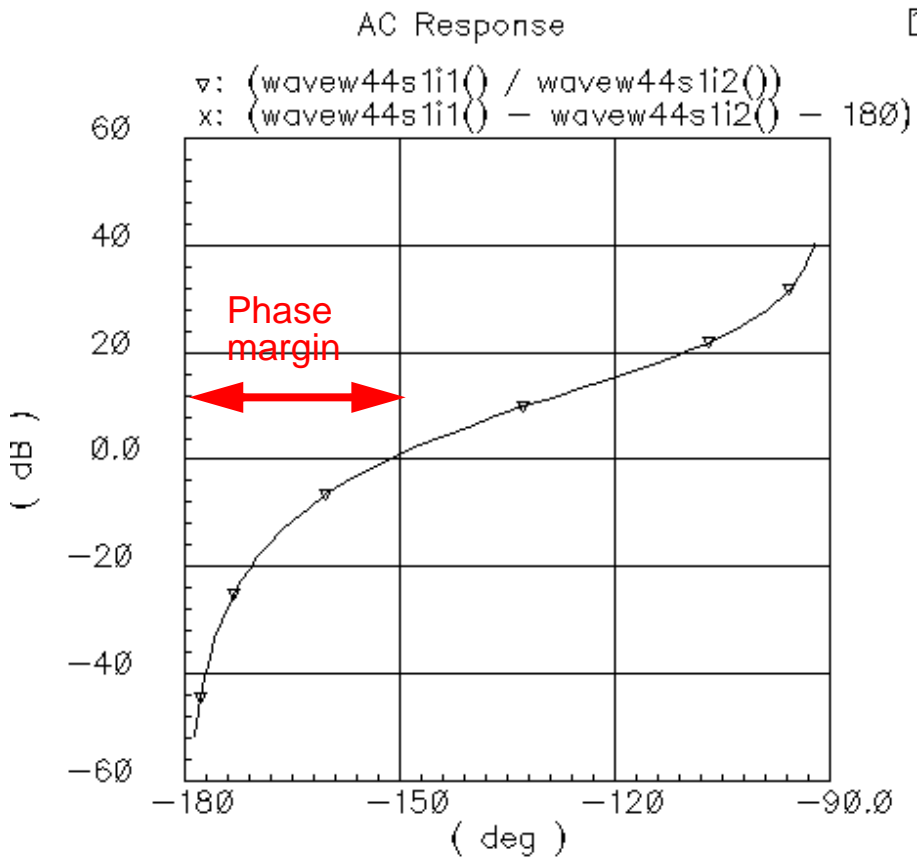


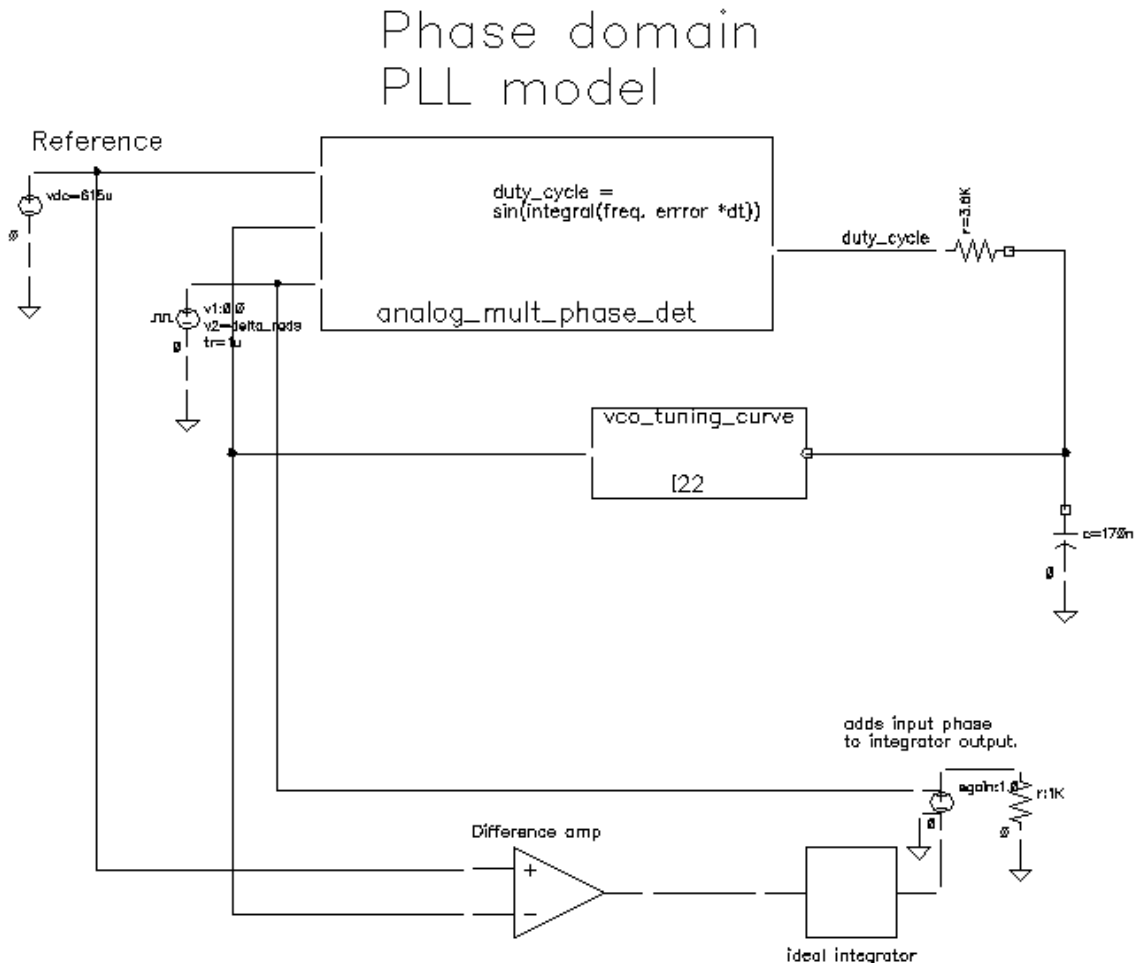
Figure G-10 Nichols chart of the loop gain



Example 3: PM input (example_PM in pllLib).

The PM (phase modulation) input pin is useful if the PLL is used as a modulator or demodulator, but it also provides a convenient place to perturb the PLL to assess large signal stability. [Figure 11](#) shows a test circuit for such a stability check.

Figure G-11 Stability check using the PM input

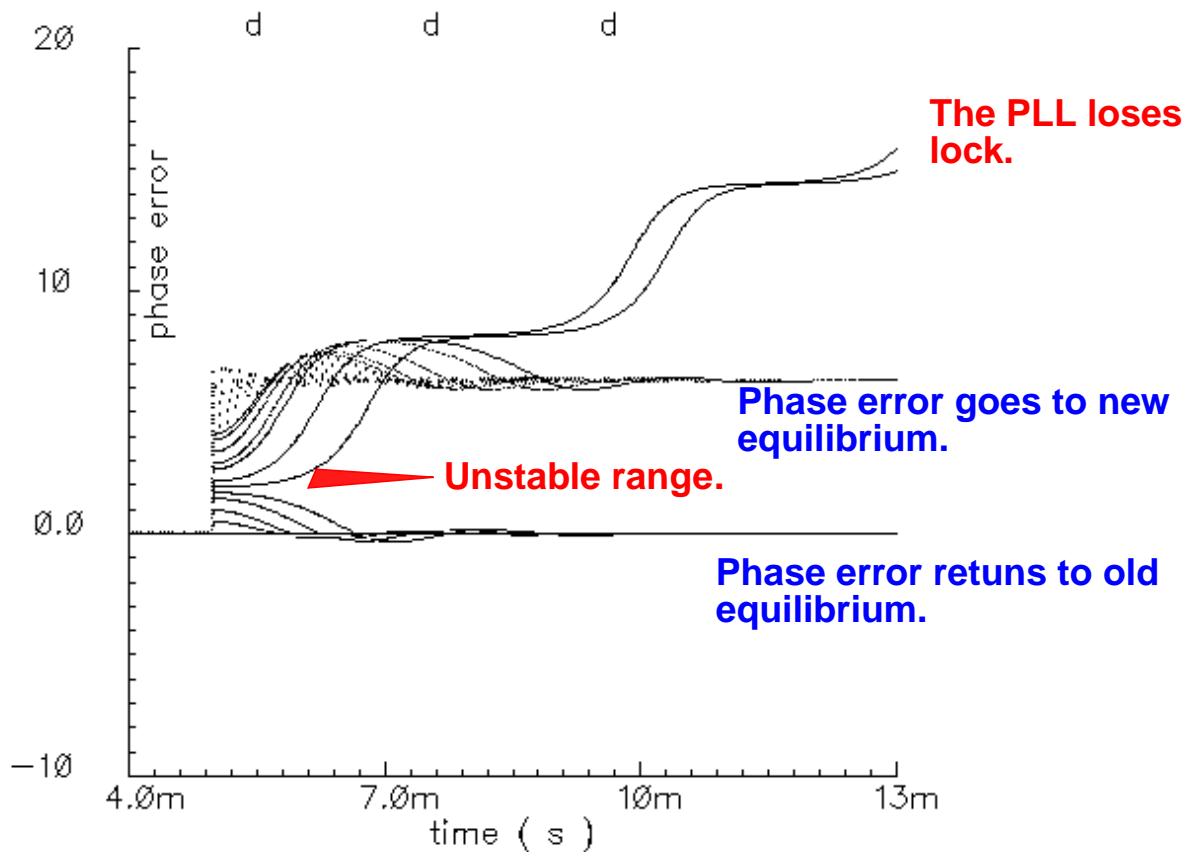


This PLL is very simple but yet different from the previous example. It was modified to produce more interesting results. The lower circuitry requires explanation. The difference amplifier computes frequency error and converts it from Mhz to rad/sec. The integrator computes VCO and reference contributions to phase error. The voltage-controlled-voltage-source at the end adds the input phase stimulus to compute total phase error. The difference amplifier and integrator are from the ahdl library.

The input phase is a delayed step. The delay makes the initial phase error easy to read. A parametric analysis on the phase error's step response with respect to the size of the input phase step reveals some interesting behavior. [Figure 12](#) shows the family of phase error step responses produced by the parametric analysis. The external integrator is intentionally not a circular integrator like the one inside the phase detector model. For large and small steps in input phase, the PLL settles into equilibrium, possibly a new one. However, a narrow

intermediate range of steps puts the PLL into an unstable mode. The references examine this behavior in mathematical detail [1,4,6,10]. This example shows one way to assess large-signal stability and to demonstrate that the phase-domain models capture the major non-linear mechanisms.

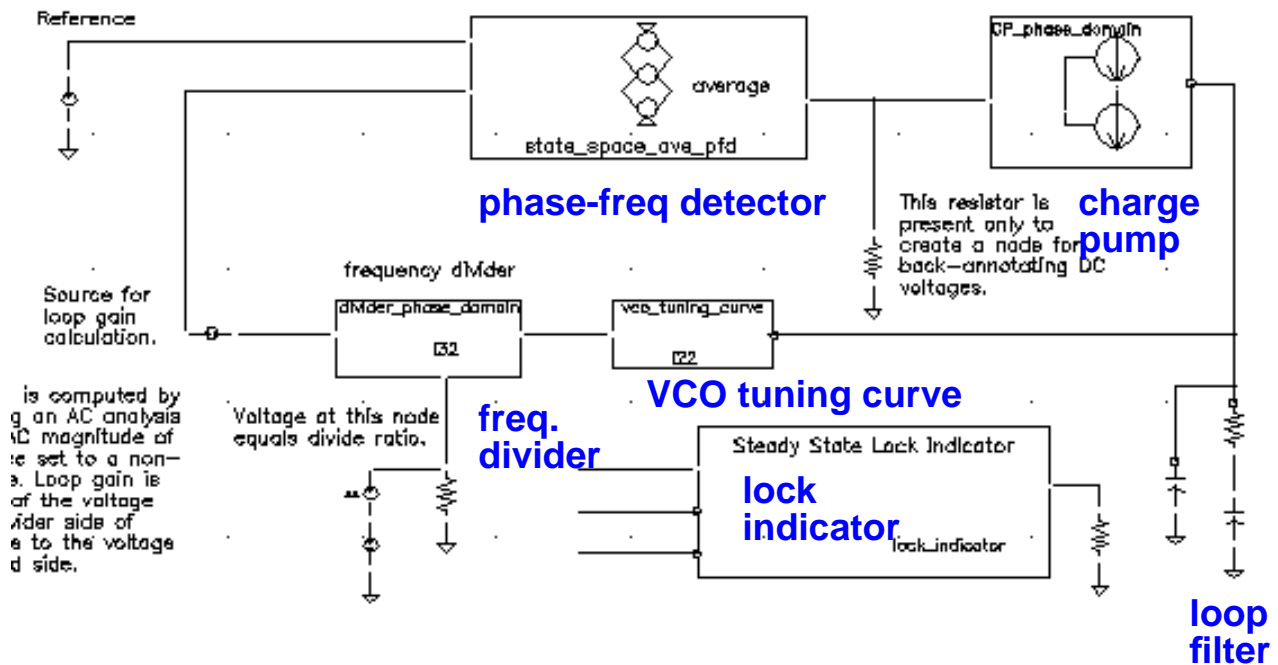
Figure G-12 Phase error response to step PM input



Assembling a Model of a PFD-based PLL

Figure 13 shows a block diagram of a typical PLL with a phase-frequency detector. This section describes how to specify each component in Figure 13 and briefly explains what each model does.

Figure G-13 PLL block diagram



VCO

The VCO is modeled by its tuning curve. The tuning curve characterizes the relationship between the input voltage and the output frequency. The input to the VCO model is the loop filter output voltage, also called the VCO control voltage. The VCO output is a voltage representing the VCO's instantaneous frequency in Mhz. Therefore, when the VCO operates at 2 Mhz, the model output is 2 Volts. The tuning curve is generally nonlinear and can be specified either with the coefficients of a fourth order polynomial or a look-up table.

Polynomial tuning curve: The input voltage is internally clamped to the nearest end point if it moves outside the interval [min vco input voltage, max vco input voltage]. Although the input voltage may fall outside the interval, the output behaves as though the input voltage value is at the end points. Within the interval, the output is a fourth order polynomial in the quantity, V_{input} minus the free running voltage. When the input voltage equals the free running voltage, the output frequency equals the free running frequency. The scale factor scales the entire polynomial and has a default value of 1. The scale factor is useful in converting data in Khz/volt, for example, to the required Mhz/volt. The parameters are the coefficients of the polynomial.

Table look-up tuning curve: The two parameters are scale factor and the path to the look-up data. The look-up model linearly interpolates between data points and linearly extrapolates

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

outside the data interval. The data format is two columns of data delimited by spaces. There is no header, and there are no extra lines at the end. The first column is input voltage. The second column is output frequency. The path to the data can be absolute or relative to the netlist. The netlist is usually stored at

```
<home>/simulation/ckt_name/spectre/schematic/netlist/input.scs
```

but you can choose a different location.

If the data is at <home>/data/table and the netlist is as shown above, the relative path is
../../../../data/table.

Frequency Divider

The frequency divider is essentially a simple gain element. It takes an input voltage that represents frequency in Mhz, and then scales it by the divide ratio to generate an output voltage that represents the divided frequency. The divide ratio is numerically equal to the voltage on the control pin. If the divide ratio drops below 0.001, the model assigns it to 0.001 and issues a warning. This assignment prevents division by zero during simulation.

Charge Pump

The charge pump transforms the duty cycle into the expected average current sourced or sunk by the charge pump. You define the maximum source and sink currents, and they can be different from each other. If the charge pump output voltage exceeds the rails you define, the output voltage is clamped to the rail through a 0.001 Ohm resistance. The other parameters are the leakage resistance and open circuit voltage. These last two parameters specify the Thevenin equivalent circuit of a leakage path. The leakage path can source or sink current depending on the open circuit voltage.

Loop Filter

The loop filter is entered component-for-component.

State-space averaged PFD (Phase-domain phase-frequency detector model).

The phase-frequency detector (PFD) model approximates average behavior of a digital, three-state, phase frequency detector. This is the most complicated model in the PLL library. The term *state-space averaged* is borrowed from the power electronics field [14]. The charge pump currents for the three PFD states are averaged together with a duty cycle much

Affirma RF Simulator (SpectreRF) User Guide

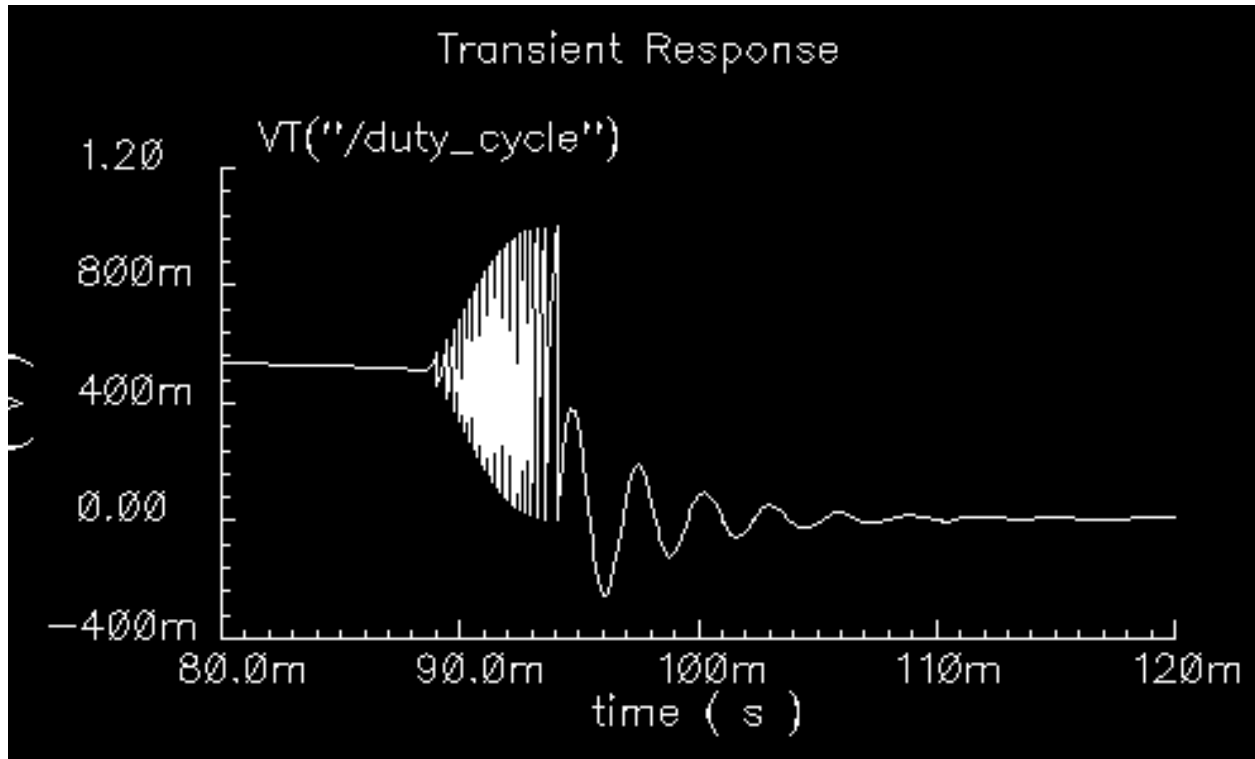
An Introduction to the PLL library

like voltages are averaged together with a duty cycle in a switch-mode power supply model. The PFD inputs are voltages representing the reference and divider output frequencies in Mhz. The output is a voltage that when multiplied by the maximum charge pump current, numerically equals the average charge-pump output current. The PFD output is a duty cycle. When the frequency error is large, the duty cycle is a smooth waveform directly related to the normalized frequency error [1,4]. When the frequency error is small, an integrator inside the PFD model converts frequency error to phase error, and the duty cycle is proportional to the phase error. The duty cycle starts jumping to zero (or resets) as it changes from frequency-mode to phase-mode.

As phase error enters a deadband determined by the minimum-on-time parameter and reference frequency, the model computes a duty cycle pulse with magnitude one and duration equal to the minimum-pulse-width. After the pulse expires, the duty cycle drops to zero until the phase error exits the deadband. As the phase error exits the deadband, the duty cycle increases to a non-zero value. The deadband and fixed-width unity pulse simulate what some texts call *backlash* [8].

The PFD model has two parameters. The first is a numerical option that controls the trade-off between execution speed and accuracy. The *speed_vs_accuracy* parameter controls the number of times the internal integrator is reset during the transition from frequency-mode to phase-mode. Too few resets can cause error. Too many resets can needlessly slow run time. The default value of this parameter is 50k. To reduce the number of resets in a slow PLL, and thereby reduce run time, increase the *speed_vs_accuracy* parameter to 70k or 100k. To increase the number of resets in fast PLLs, and thereby increase the accuracy, reduce the *speed_vs_accuracy* parameter to 10k or 20k. A reasonable setting for the *speed_vs_accuracy* parameter produces a duty cycle step response that resets approximately to zero at least 3-10 times before entering the final transient. [Figure 14](#) shows reasonable duty cycle step response.

Figure G-14 Duty cycle waveform



The other parameter is the *minimum_on_time*. It controls backlash. This is the minimum pulse width the PFD can generate. As the phase error decreases, the pulse width drops discontinuously from the minimum pulse width to zero. This effect creates a deadband in the duty cycle versus phase error curve. Figure 12 was generated with the default minimum-on-time of zero μs . The default value of this parameter produces no deadband and no unity pulses; The default deactivates the backlash mechanism. Figure 15 was generated with a minimum on time of 0.2 μs . Figure 15a illustrates that the pulses only occur as the phase error enters the deadband¹. Figure 15b shows the limit cycle created by the backlash. The limit cycle is primarily determined by leakage on the loop filter and the minimum pulse width. Some references suggest biasing the duty cycle away from the deadband or loading the filter down

1. A pulse is not kicked out upon exiting the deadband because that behavior causes convergence problems for Spectre. If phase error is entering the deadband, a pulse at that moment pushes phase error in the same direction it was going, into the deadband. If a pulse occurs as phase error exits the deadband, the pulse drives phase error back into the deadband and Spectre has trouble figuring out whether phase error should leave the deadband at all. Fortunately, no significant error is introduced by implementing the pulse only when phase error enters the deadband. In a backlash limit cycle, the feedback loop quickly drives phase error back into the deadband and a pulse occurs on the way in. The error is in the time the feedback loop takes to return phase error to the edge of the deadband and that is usually small when PLL is in a backlash limit cycle.

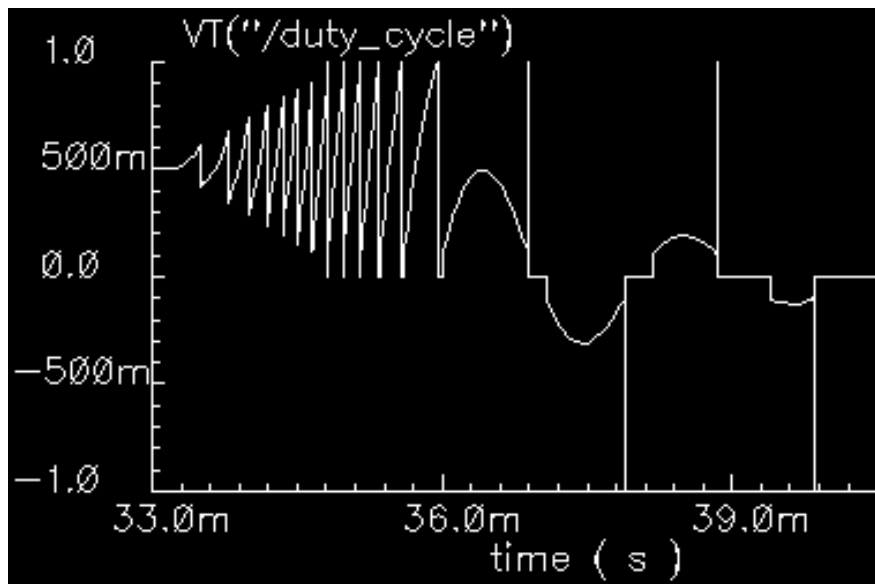
Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

to force the limit cycle frequency to a value in which the loop filter attenuates it. The phase-frequency detector model can help quantify the problem and check the solution.

(The PLL model that generated [Figure 15](#) had a center frequency of about 1Mhz and the simulation ran out to 130ms. A voltage-domain model might easily simulate 10 points per carrier cycle. The voltage-domain model would require 1.3 million points to simulate the same amount of action. I did not attempt it. The phase-domain simulation that generated [Figure 15](#) ran in a matter of seconds!)

Figure G-15



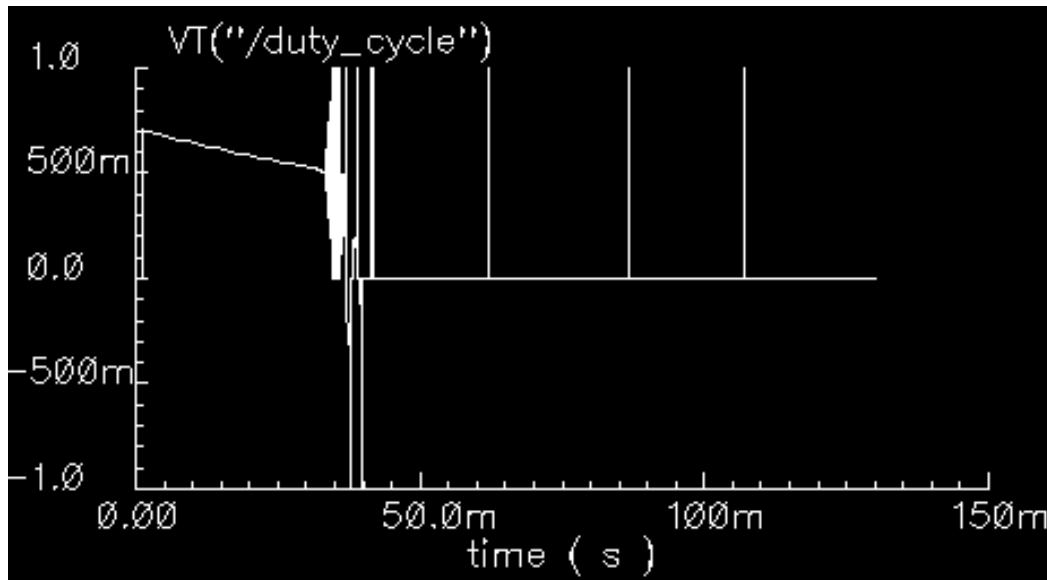


Figure 15b
Duty cycle to
130 ms

Lock Indicator

All real components have limited outputs and the limits of any one component can keep the PLL from locking. Just like the simple phase detector model, all of the phase-domain models operate one way for DC analysis and another for transient analysis to prevent DC convergence errors. The lock indicator monitors three signals. In the example, the lock indicator monitors the phase detector output, the VCO control, and the charge pump output. If any of those signals exceeds its limit, the lock indicator output is zero, signifying that the loop is not locked in steady state. If all signals are within their limits, the output is 1 volt, specifying that the loop is locked. The lock indicator is only valid for DC analysis. Use node names to tie the lock indicator inputs to the right nodes and use variables for the component limits. You must specify the units manually twice, once for each component and once for the lock indicator. With variables, the lock indicator parameters are linked to the proper component parameters, and you specify changes in only one place.

Example 4: Acquisition transients (example_phase_domain in pllLib).

Figure 16 shows the duty cycle and VCO frequency response to a momentary change in the divider ratio. When the divider ratio changes, the PFD enters the frequency-mode and slews the VCO frequency toward the new value. As the VCO frequency approaches the final value, the PFD model gradually changes from frequency-mode to phase-mode. When the frequency error is small, but still large enough to slew phase error, the duty cycle waveform looks like a sawtooth waveform. The model gradually increases the amplitude of the sawtooth component

of the duty cycle, and always maintains the correct average, as frequency error reduces to zero. The final duty cycle transient is the sawtooth that depends mostly on phase error. Figure 12, above, modifies the x-axis of the graph to show the duty cycle in the first transition. Figure 17 modifies it further to show the sawtooth waveform.

Figure G-16 Response to momentary change in divider ratio

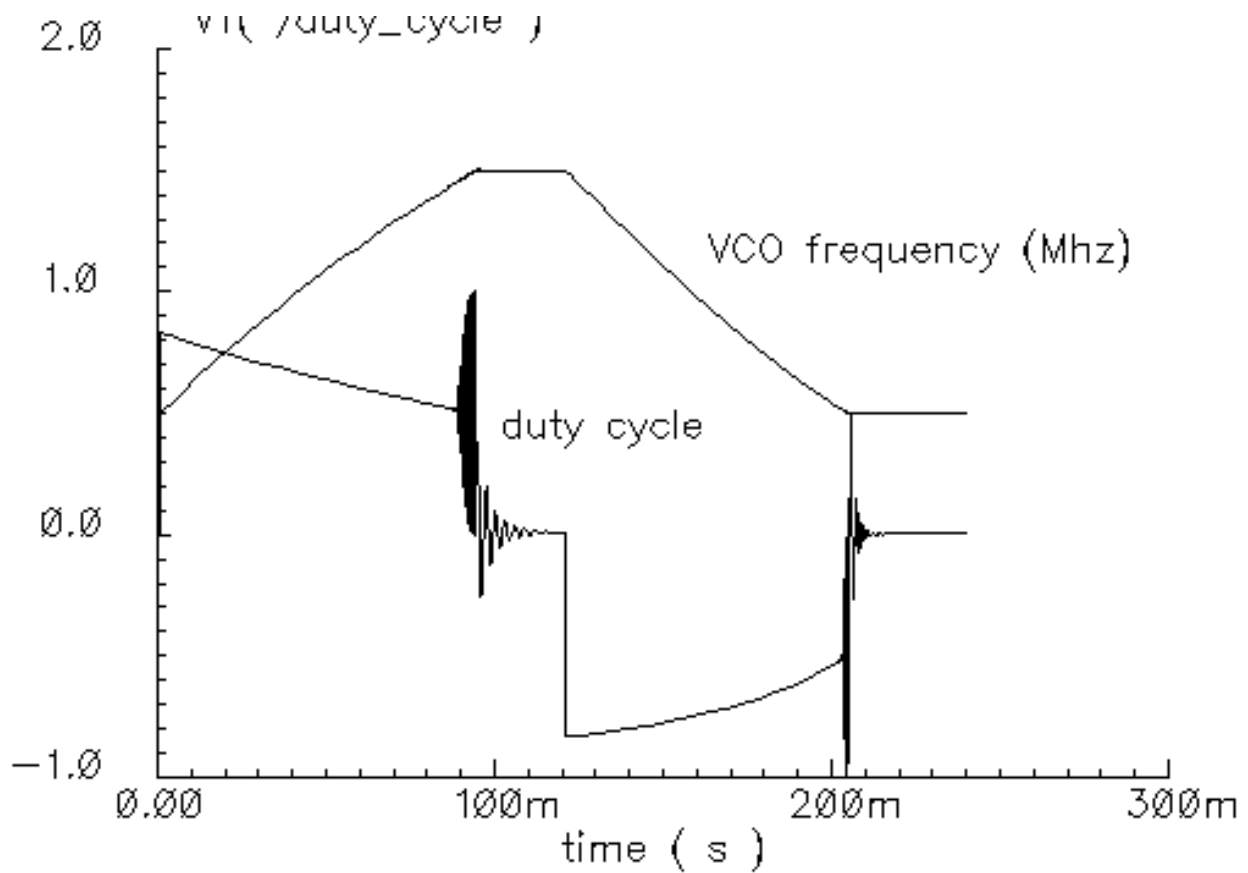
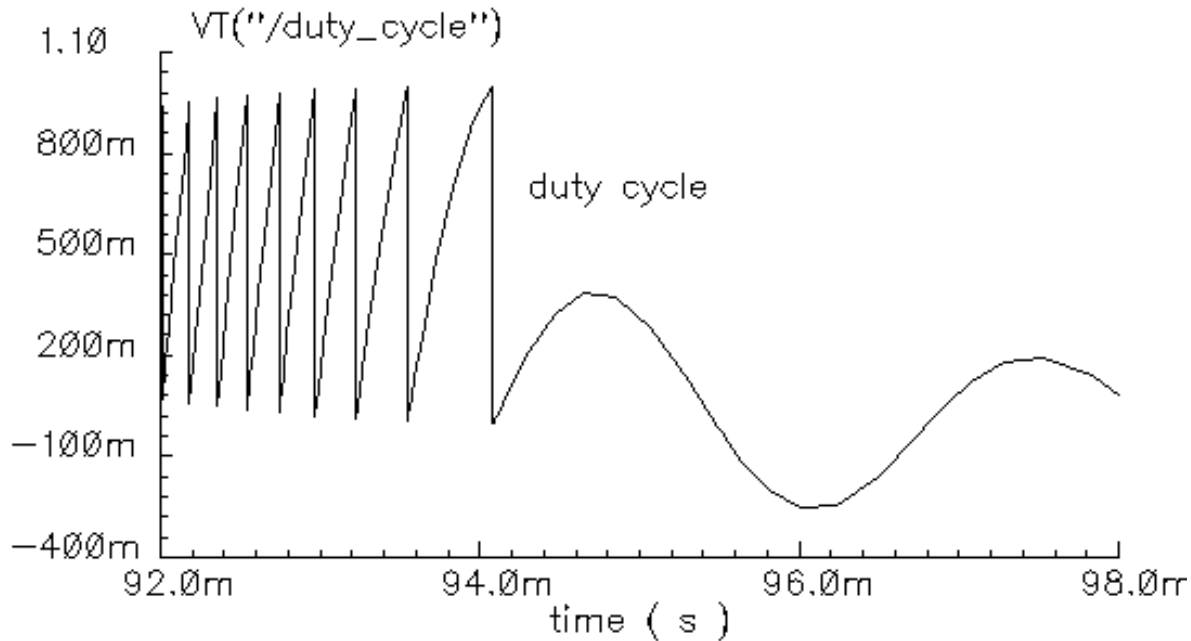


Figure G-17 Duty cycle during transition from frequency mode to phase mode



Example 5: Comparison with a voltage-domain model (example_voltage_domain in pllLib).

The only node that can be directly compared between phase- and voltage-domain models is the VCO control node. At full scale, the difference between the two models is not visible. The differences occur at the transitions. [Figure 18](#) compares the two models at the transitions. In this example, the voltage-domain model simulates in three hours, and the phase-domain model simulates in two seconds on the same machine.

The error between the two models does not appear to be consistent. It is larger in the first transition. Furthermore, decreasing the *speed_vs_accuracy* parameter does not always increase the similarity of the waveforms. This is because the final transient, the one driven primarily by phase error, depends on the residual frequency error at the time the phase error last crossed 2π . The frequency error at that moment, especially after a long frequency slewing period, is sensitive to, among other effects, initial conditions. The voltage-domain model therefore shows the same level of variation for small differences in the initial conditions preceding the transition to the new equilibrium. [Figure 19](#) compares two voltage-domain simulations of the VCO control signal during the second transition. One of the voltage-domain simulations (the dotted waveform) used different initial conditions. The blue solid waveform is

a delayed version of the dotted waveform. The delay overlays the two simulations for direct comparison. The error is comparable to the error between voltage-domain and phase-domain simulations.

Figure G-18 Comparison with voltage domain model

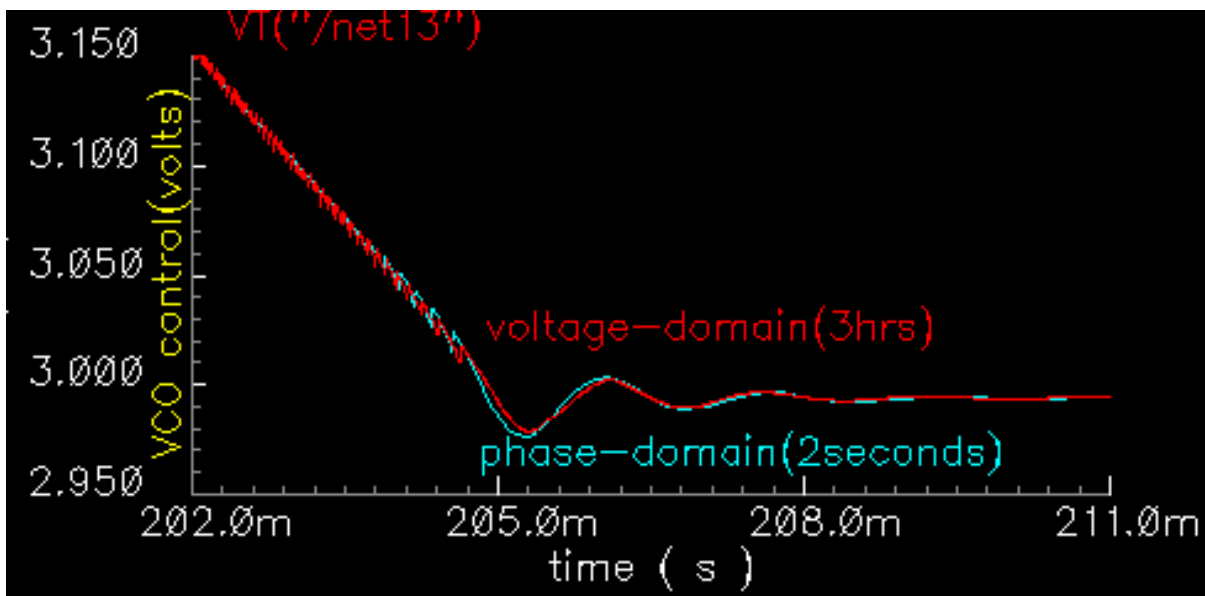
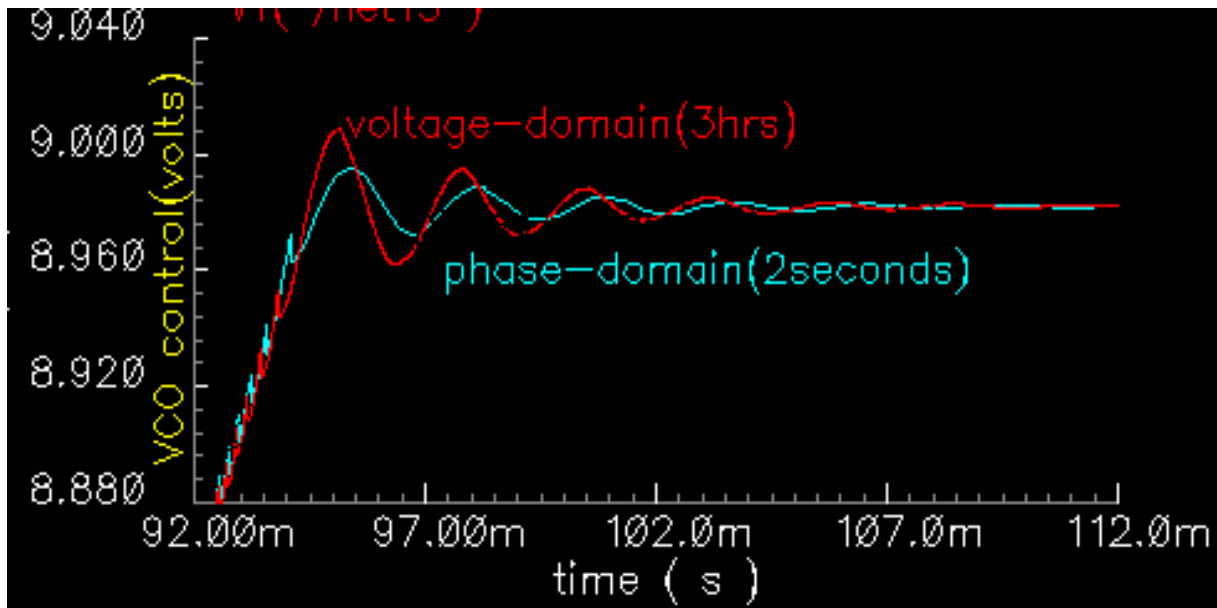
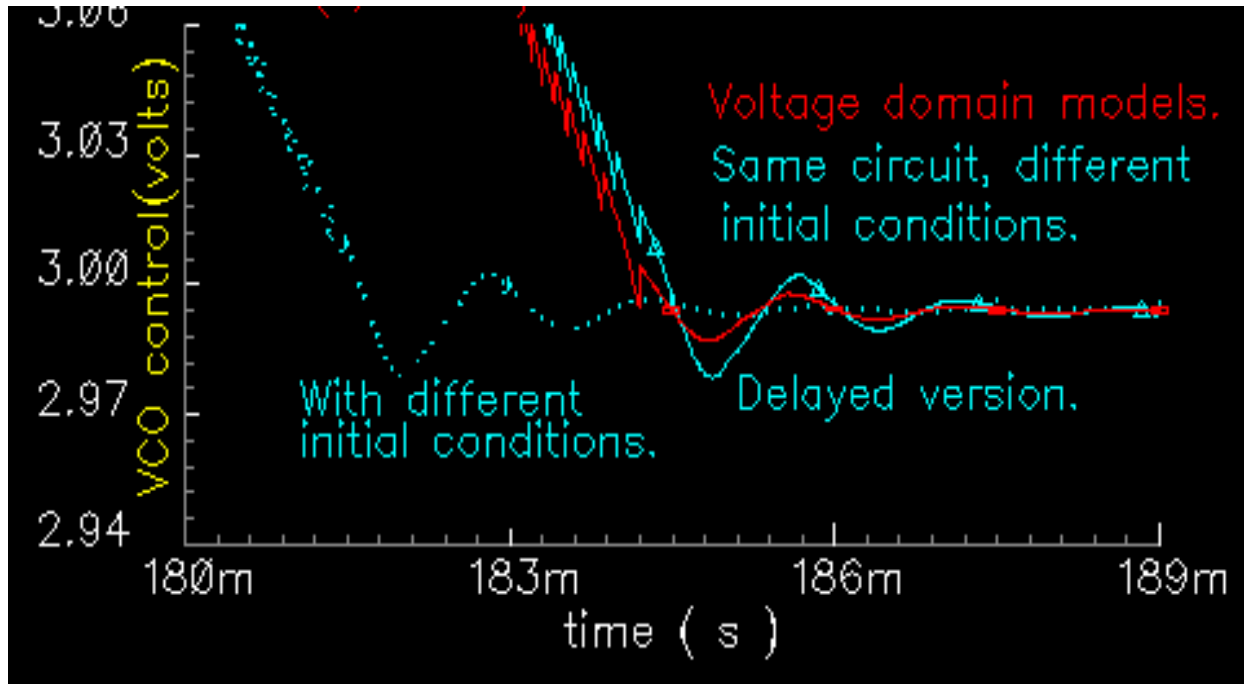


Figure G-19 Effect of initial conditions



How the PFD Model Works.

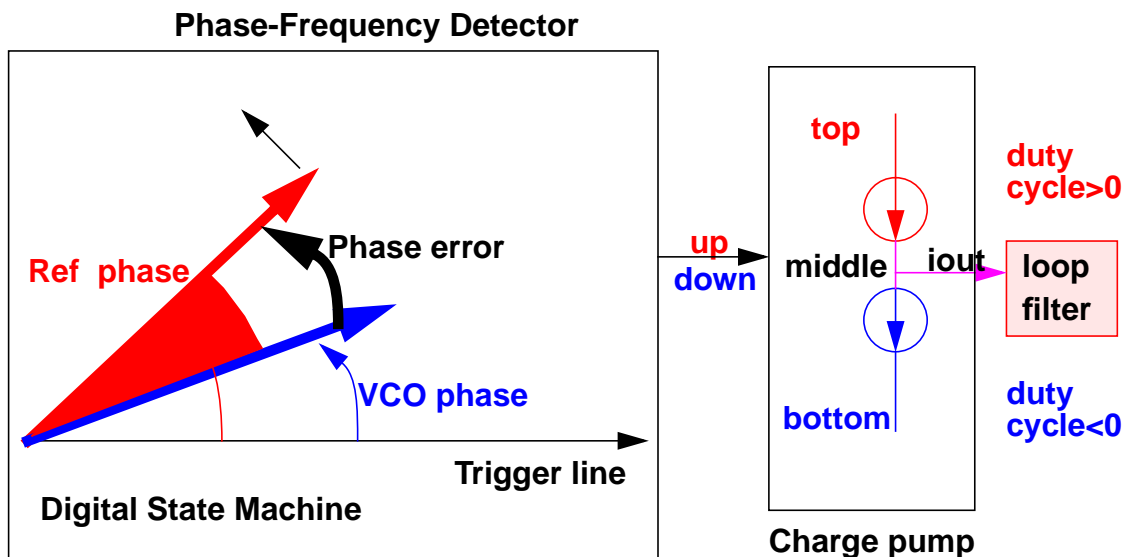
The heart of the PFD is a digital state machine [8,9]. The model is for PFDs with three digital states. The output stage is usually a charge pump or pair of switches. It is convenient to model the PFD in two pieces. The first piece models the state machine and computes a duty cycle that is independent of the output stage. The second piece models the output stage. The charge pump (CP) is used here as an example.

How the PDF/CP Pump Works

Let the three PFD states be stacked. In the top state, the PFD commands the CP to source current. In the middle state, the pump is off. In the bottom state, the CP sinks current. The PFD is edge triggered. [Figure 20](#) shows vectoral representations of the reference and VCO clocks [1,4]. Both vectors rotate counter-clockwise round the origin. The angle between the hands equals phase error. Phase error lies between $\pm 2\pi$. Whenever the reference passes a trigger line, like 3 o'clock, the state jumps to the next state up. If the PFD is already in the top state, the state does not change. Whenever the VCO passes the trigger line, the state jumps to the next state down. If it's already in the bottom state, it again does not change. For a fixed

phase error, the state toggles as the hands rotate. State toggles between the middle and top, or between the middle and bottom states. The percentage of time spent in the top, or bottom, state is the duty cycle. The duty cycle is positive for top to middle toggling and negative for middle to bottom toggling.

Figure G-20 PFD operation



up: go one state up when the ref vector passes the trigger line.
down: go one state down when the vco vector passes the trigger line.

The loop filter responds to a “shade” of current just as the eye sees a pink disk as the vectors rotate rapidly about the origin.
 The shade varies with phase error (i.e. duty cycle)

If the reference and VCO frequencies are identical, the vectors in Figure 20 rotate together. Let the sector defined by phase error be red if the reference leads and blue if it lags. If the hands rotate once per minute and the reference leads, you see two colors, white and red. At two million revolutions per second, you see pink. The shade of pink depends linearly on the phase error. Although PFD output current toggles between two values, the loop filter and VCO respond mainly to the “shade” of current. The shade is proportional to the duty cycle. With zero frequency error, duty cycle equals phase error divided by 2π . Existing literature uses one function to describe the *phase-error-to-duty-cycle* relationship and a different function to describe the *frequency-error-to-duty-cycle* relationship. These two functions are the *locked duty cycle function* and *averaged unlocked duty cycle function*, respectively. The new model combines these two functions into one practical model.

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

The new model operates only on frequency error. For small-frequency errors, the duty cycle is indeed proportional to the phase error. The phase error is the integral, with respect to time, of the frequency error. The duty cycle therefore equals the integral of the frequency error divided by 2π . Resetting the integral whenever it hits $\pm 2\pi$ produces the multivalued sawtooth described above. If the frequency error changes sign, the resetting integrator ramps to zero, passes through zero, and generates a sawtooth in the lower-half plane. The phase-error-duty-cycle trajectory is precisely the multivalued sawtooth described above. The resettable integrator (RI) merges the integrator of a phase-domain model with the locked duty-cycle function.

For a sustained frequency error, the RI model predicts an average duty cycle of $\pm 1/2$ regardless of error size. This is correct only for small-frequency errors. The true duty cycle goes to ± 1 for large-frequency errors. Let the reference frequency far exceed VCO frequency. Whenever the VCO passes the trigger line, the reference frequency passes shortly thereafter. The reference frequency might pass the trigger line several more times before the VCO passes again. In this case, the phase error is still a sawtooth, but the average duty cycle is nearly 1. This behavior lets the PLL acquire input signals faster. The function H, in [Figure 22](#) below, shows the averaged unlocked duty cycle. This cycle depends on the ratio of the reference to VCO frequencies and is discontinuous where frequency error is zero.

Figure G-22 Averaged unlocked duty cycle

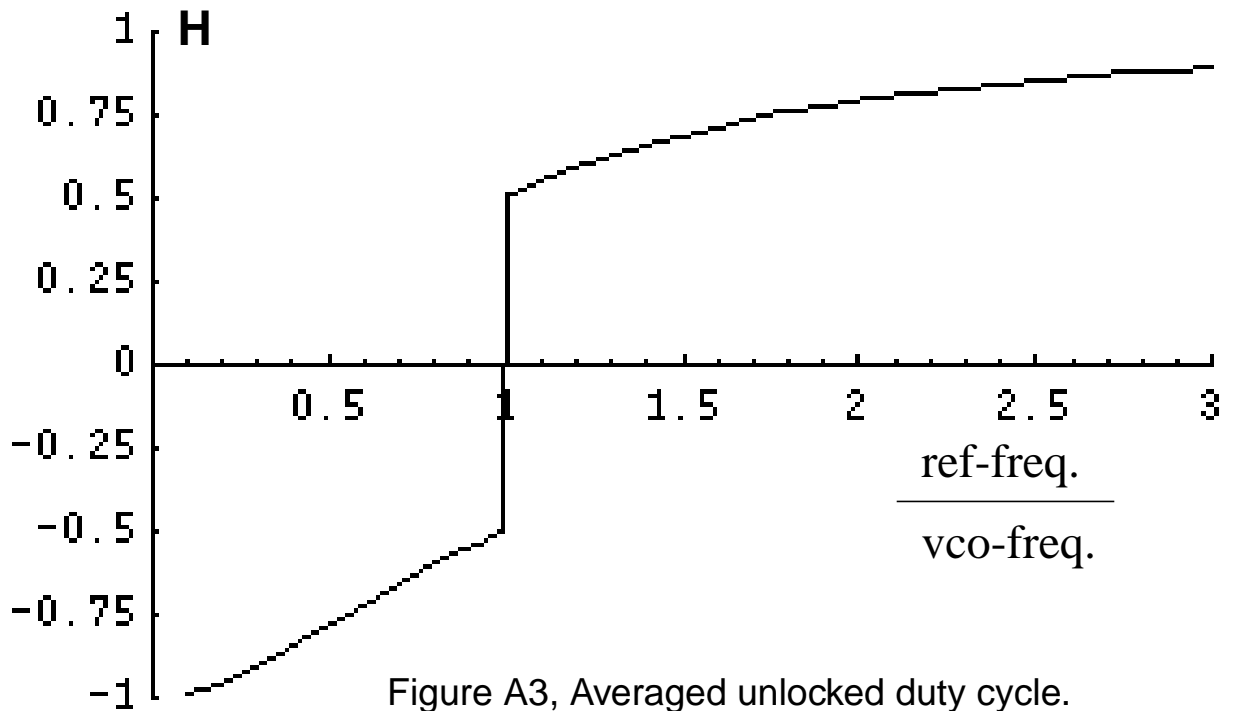
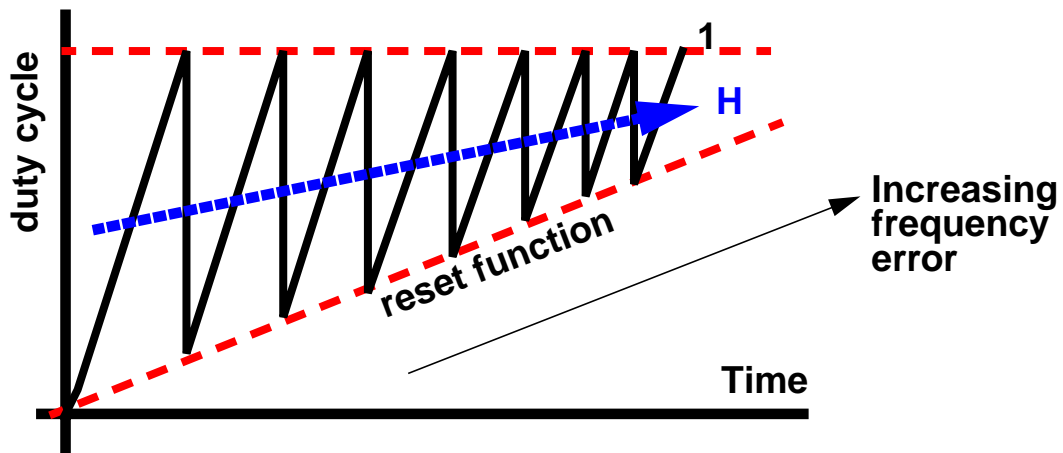


Figure A3, Averaged unlocked duty cycle.

The RI is modified to include the frequency effect. For small-frequency errors, the predicted average duty cycle equals $1/2$. This is true because the RI runs from the reset point ($=0$) to the reset threshold ($=2\pi$). It is not necessary to reset the integrator to zero. Resetting the integrator to a “reset” function gives the correct average duty cycle (Figure 23). As the frequency ratio goes to $+\infty$, the reset point changes to $+2\pi$. Because the reset threshold is still $+2\pi$, the predicted average duty cycle changes to $+1$. As the frequency ratio approaches unity, the reset point returns to zero, and the predicted average duty cycle returns to $1/2$.

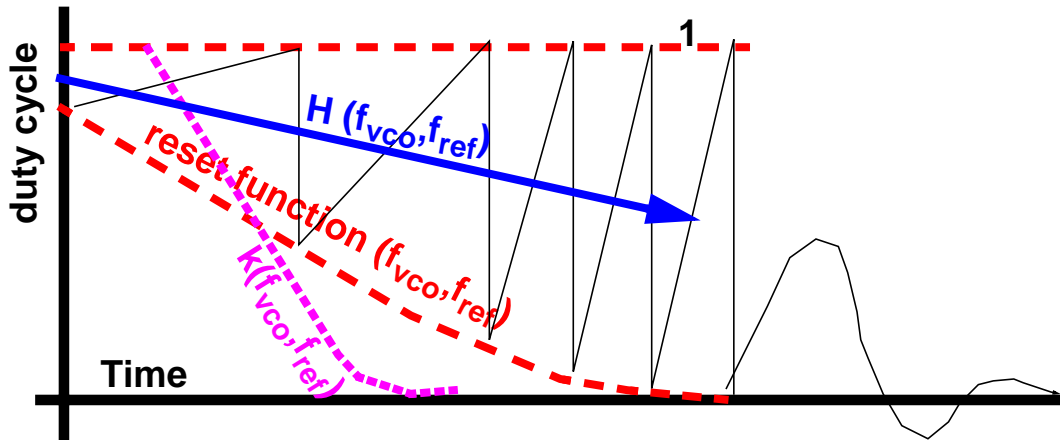
Figure G-23 Combining averaged locked and unlocked duty cycles



$$duty\ cycle = \int \frac{R(\text{frequency error}) \cdot (\text{Frequency Error})}{2\pi} dt$$

The state space averaged PFD model requires one more addition to be practical. As the reset point nears $\pm 2\pi$, the integrator resets very frequently and execution stalls. The integrator must be deactivated for large-frequency errors. The new PFD model uses the weighted sum of a RI and the H. The weighting factors are k and $(1-k)$. k is a function of the ratio of the two input frequencies. k approaches 1 for large-frequency errors and approaches 0 as the frequency ratio approaches unity. A factor of $(1-k)$ under the integral deactivates the integral for large-frequency errors. The resulting PFD model looks like H for large-frequency errors, and it looks like the reactivated RI for small-frequency errors. k determines how fast the RI reactivates and how gradually the model changes from H to the RI. A *speed_vs_accuracy* parameter controls k . If the model does not reset a few times before reaching frequency lock, you can improve the results by decreasing the *speed_vs_accuracy* parameter. If the model resets so often that the simulation is too slow, you can speed execution by increasing the parameter. The default setting of 50000 covers a wide range of loop speeds. Figure 24 shows the transition from frequency-mode to phase-mode.

Figure G-24 Complete model



$$duty\ cycle = k * H + (1 - k) \int_R (1 - k) \frac{(Frequency\ Error)}{2\pi} dt$$

References

[1]“Loops, Theory and Application,“ J. L. Stensby.

[2]“Relative -phase modeling speeds Spice simulation of modulated systems,“ John Kesterson, November 11, 1993 issue of EDN

[3]“Simulation of Communication Systems,“ M. Jeruchim, P. Balaban and K. Shanmugan. Plenum, 1992.

[4]“Synchronization in Digital Communications,“ Heinrich Meyr and Gerd Ascheid, Published by John Wiley and Sons.

[5]“Phase-Locked and Frequency-Feedback Systems,“ Klapper and Frankle, Published by Academic Press

[6]“Phaselock Techniques,“ Floyd Gardner, Published by John Wiley and Sons

[7]“Non-Linear Relative Phase Models of Phaselock Loops,“ Jess Chen, Cadence Technical Conference, 1996.

Affirma RF Simulator (SpectreRF) User Guide

An Introduction to the PLL library

[8]“Phase-Locked Loops: Theory, Design, and Applications,” Roland Best, Published by McGraw-Hill

[9]“Phase-Locked Loop Circuit Design,” Dan Wolaver, Published by Prentice Hall

[10]“Phase-Lock Basics,” Willeam F. Egan. Published by John Wiley and Sons.

[11]“Non-Linear State Space Averaged Modeling of a 3-State Digital Phase-Frequency Detector,” Jess Chen, Cadence Technical Conference, 1997

[12] “Analog and Mixed-Signal Hardware Description Languages,” Edited by A. Vachoux, J. Berge, O. Levia, and J. Rouillard. Kluwer Academic Publishers.

[13] “Macromodeling with SPICE,” J.A. Connelly and P. Choi. Prentice Hall. Pages 168-169.

[14] S. Cuk, Cal. Inst. of Tech. PhD Thesis, “Modelling, Analysis, and Design of Switching Converters.” 1977

Using *port* in SpectreRF Simulations

The *port* component, located in the *analogLib* library, can be used in RF circuits for SpectreRF and Spectre S-parameter simulations. The *port* component is similar to the existing *psin*, also located in the *analogLib* library. Unlike *psin*, *port* supports all the *Source types* of the *port* primitive in Spectre, specifically: *pwl*, *pulse*, *sine*, *dc*, and *exp*.

The *port* is a resistive source that is tied between positive and negative terminals. It is equivalent to a voltage source in series with a resistor, where the reference resistance of the *port* is the value of the resistor.

In IC4.4.3_QSR1, *port* is supported by Spectre Direct netlisting *only* in the Affirma analog circuit design environment (spectreS, or Spectre through the cdsSpice socket, is not supported).

***port* Capabilities**

While the *port* component is generally useful as a stimulus in high-frequency circuits, it also has the following unique capabilities:

- Defines the ports of the circuit to the S-parameter analysis
- Has an intrinsic noise source that lets the noise analysis directly compute the noise figure of the circuit
- Is the only source for which the amplitude can be specified in terms of power

Terminating the port

When you specify the voltage on a *port*, Spectre assumes that *the port is properly terminated in its reference resistance*. The specified voltage value is not the voltage on the internal voltage source, which is actually set to twice the value specified on the `port`. If you use a `port` source to drive an open circuit, the voltage (for DC, transient, AC, and PAC signals) is double its specified value. However, you can alternatively specify the amplitude of the sine wave in the transient and PAC analyses as the power in dBm delivered by the *port* when terminated with the reference resistance.

***port* Parameter Types**

In this application note, the parameters are grouped by the following types, rather than in the order they appear on the *port* Edit Object Properties form.

Port parameters

- Resistance
- Port number
- Multiplier

General waveform parameters

- Source type
- Delay time

DC Waveform parameters

- DC voltage

Pulse waveform parameters

- Zero value
- One value
- Period of waveform
- Rise time
- Fall time
- Pulse width

PWL waveform parameters

- Waveform Entry Method
- File name
- Number of PWL/Time pairs
- DC offset

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Amplitude scale factor

Time scale factor

Breakpoints

Period

Transition Width

Sinusoidal waveform parameters

Sine DC level

Frequency name 1

Frequency 1

Amplitude 1 (Vpk)

Amplitude 1 (dBm)

Phase for Sinusoid 1

Frequency name 2

Frequency 2

Amplitude 2 (Vpk)

Amplitude 2 (dBm)

Phase for Sinusoid 2

Amplitude and Frequency modulation parameters

AM modulation index 1

AM modulation frequency 1

AM modulation phase 1

FM modulation index 1

FM modulation frequency 1

Damping factor 1

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Exponential waveform parameters

Delay time

Zero value

One value

Rise time start

Rise time constant

Fall time start

Fall time constant

Small-signal parameters

PAC magnitude

PAC magnitude (dBm)

PAC phase

AC magnitude

AC phase

XF magnitude

Temperature effect parameters

Linear temperature coefficient

Quadratic temperature coefficient

Nominal temperature

Noise parameters

Noise Entry Method

Noise file name

Number of noise/freq pairs

Port Parameters

Port parameters include *Resistance*, *Port Number*, and *Multiplier*.

Resistance

The reference resistance of the system. The value must be a real number, but not 0. Default: 50 Units: Ω

Port number

The number of the *port*. The value must be a nonzero integer. Each *port* in a schematic must have a different *Port number*. The *Port number* is not automatically indexed when you place each *port* on your schematic.

Multiplier

The multiplicity factor. The value must be a nonzero real number. This number lets you specify a number of *ports* in parallel. For example, if you set *Resistance* to 50 and *Multiplier* to 2, you specify two *ports* in parallel, with an effective reference resistance of 25 Ω . Default: 1

General Waveform Parameters

Source type

The *Source type* parameter lets you specify the parameters for several different wave shapes and then quickly switch between them without losing the settings. Possible values for *Source type* are: *dc*, *pulse*, *exp*, *pwl*, *sine*, or blank. The typical *Source types* used in SpectreRF analyses are *dc*, *pulse*, and *sine*. For example, you can quickly switch from a sinusoid level (for PSS analysis) to a DC level (for PAC analysis) by changing the *Source type* from *sine* to *dc* in the Edit Object Properties form.

If you set *Source type* to a blank value, the *port* acts as a resistive load.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Figure H-1 *Source type=blank* in Edit Object Properties form

Edit Object Properties			
OK		Cancel	Apply
Defaults		Previous	Next
Help			
Apply To	<input type="radio"/> only current <input type="radio"/> instance		
Show	<input type="checkbox"/> system <input checked="" type="checkbox"/> user <input checked="" type="checkbox"/> CDF		
Browse		Reset Instance Labels Display	
Property	Value	Display	
Library Name	analogLib	off <input type="checkbox"/>	
Cell Name	port	off <input type="checkbox"/>	
View Name	symbol	off <input type="checkbox"/>	
Instance Name	PORT0	off <input type="checkbox"/>	
		Add	Delete
		Modify	
User Property	Master Value	Local Value	Display
lvsignore	TRUE		off <input type="checkbox"/>
CDF Parameter	Value	Display	
Resistance	50 Ohms	off <input type="checkbox"/>	
Port number		off <input type="checkbox"/>	
Source type	<input type="checkbox"/>	off <input type="checkbox"/>	
Multiplier		off <input type="checkbox"/>	

If you set *Source type= dc, pulse, pwl, sine, or exp*, you can create a dc, pulse, piecewise linear, sine, or exponential waveform whose parameters you can alter. These parameters are described in more detail in the *DC, Pulse, Piecewise Linear, Sinusoidal, and Exponential Waveform Parameters* sections that follow.

Delay time

The waveform *Delay time*, or the time that the source stays at the DC level before it starts generating waveforms (assuming the *Source type* is set to *sine, pulse, pwl, or exp*). The value must be a real number. Default: 0 Units: seconds

DC Waveform Parameters

To generate a dc waveform from the *port* component, set the CDF parameter *Source type=dc* as shown in the figure below. When *Source type=dc*, the *dc* and *temperature effect* parameters are active and set the DC level for all analyses. In DC analysis, this setting also determines the DC level generated by the source, regardless of what *Source type* you specify.

DC voltage

Sets the DC level of the source for DC analysis. The value must be a real number. If you do not specify the DC value, it is assumed to be the *time =0* value of the waveform. Default: 0
Units: V

The *DC voltage* parameter specifies the DC voltage across the *port* when it is terminated in its reference resistance. In other words, the *DC voltage* of the internal voltage source is double the user specified DC value, *dc*. The same is true for the values for the *transient*, *AC*, and *PAC* signals of the *port*. However, you can alternatively specify the amplitude of the sine wave in the transient and PAC analyses as the power in dBm delivered by the *port* when it is terminated with the reference resistance.

Because all small signal analyses (AC, XF, and Noise) use DC analysis results, the *DC voltage* level also affects small-signal analyses. Transient analysis is not affected unless you specify *Source type=dc* or use *dc* as a default for the other waveform types.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Figure H-2 *Source type=dc* in the Edit Object Properties form

The screenshot shows the 'Edit Object Properties' dialog box for a port component. The dialog is organized into several sections:

- Buttons:** OK, Cancel, Apply, Defaults, Previous, Next, Help.
- Apply To:** only current, instance.
- Show:** system, user, CDF.
- Property Value Display:**

Property	Value	Display
Library Name	analogLib	off <input type="checkbox"/>
Cell Name	port	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	PORT	off <input type="checkbox"/>
- User Property:**

User Property	Master Value	Local Value	Display
IvsIgnore	TRUE		off <input type="checkbox"/>
- CDF Parameter:**

CDF Parameter	Value	Display
Resistance	50 Ohms	off <input type="checkbox"/>
Port number		off <input type="checkbox"/>
DC voltage		off <input type="checkbox"/>
Source type	dc <input type="checkbox"/>	off <input type="checkbox"/>
Display small signal params	<input type="checkbox"/>	off <input type="checkbox"/>
Display temperature params	<input type="checkbox"/>	off <input type="checkbox"/>
Display noise parameters	<input type="checkbox"/>	off <input type="checkbox"/>
Multiplier		off <input type="checkbox"/>

The *small signal*, *temperature*, and *noise* parameters are discussed in a later section of this application note.

Pulse Waveform Parameters

To generate a pulse waveform from the *port* component, set the CDF parameter *Source type=pulse* as shown below.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Remember, when you specify the voltage on a *port*, you are specifying the voltage *when the port is properly terminated*, and not the voltage on the internal voltage source. Therefore, the voltage on the internal source is set to twice the value specified on the *port*.

Figure H-3 Source type=pulse in the Edit Object Properties form

The screenshot shows the 'Edit Object Properties' dialog box for a port object. The 'Source type' is set to 'pulse'. The dialog is organized into several sections:

- Buttons:** OK, Cancel, Apply, Defaults, Previous, Next, Help.
- Apply To:** only current, instance.
- Show:** system, user, CDF.
- Property Table:**

Property	Value	Display
Library Name	analogLib	off <input type="checkbox"/>
Cell Name	port	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	PORT0	off <input type="checkbox"/>
- User Property Table:**

User Property	Master Value	Local Value	Display
Ivignore	TRUE		off <input type="checkbox"/>
- CDF Parameter Table:**

CDF Parameter	Value	Display
Resistance	50 Ohms	off <input type="checkbox"/>
Port number		off <input type="checkbox"/>
DC voltage		off <input type="checkbox"/>
Source type	pulse <input type="checkbox"/>	off <input type="checkbox"/>
Frequency name 1		off <input type="checkbox"/>
Delay time		off <input type="checkbox"/>
Zero value		off <input type="checkbox"/>
One value		off <input type="checkbox"/>
Period of waveform		off <input type="checkbox"/>
Rise time		off <input type="checkbox"/>
Fall time		off <input type="checkbox"/>
Pulse width		off <input type="checkbox"/>
Display small signal params	<input type="checkbox"/>	off <input type="checkbox"/>
Display temperature params	<input type="checkbox"/>	off <input type="checkbox"/>
Display noise parameters	<input type="checkbox"/>	off <input type="checkbox"/>
Multiplier		off <input type="checkbox"/>

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Zero value

The *Zero value* (*val0*) used in *pulse* and *exp* waveforms. Default: 0 Units: V

One value

The *One value* (*val1*) used in *pulse* and *exp* waveforms. Default: 1 Units: V

Period of waveform

The *period* of the *pulse* waveform. Default: *infinity*. Units: seconds

Rise time

The *Rise time* for the *pulse* waveform (time for transition from the *Zero value* to the *One value*). Units: seconds

Fall time

The *Fall time* for the *pulse* waveform (time for transition from the *One value* to the *Zero value*). Units: seconds

Pulse width

Pulse width (*width*, duration of *One value*). Default: *infinity*. Units: seconds

PWL Waveform Parameters

To generate a piecewise linear waveform from the *port* component, set the CDF parameter *Source type=pwl* as shown in the next figure.

Remember, when you specify the voltage on a *port*, you are specifying the voltage *when the port is properly terminated*, and not the voltage on the internal voltage source. Therefore, the voltage on the internal source is set to twice the value specified on the *port*.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Figure H-4 *Source type=pwl* in the Edit Object Properties form

Property	Value	Display
Library Name	analogLib	off <input type="checkbox"/>
Cell Name	port	off <input type="checkbox"/>
View Name	symbol	off <input type="checkbox"/>
Instance Name	PORT	off <input type="checkbox"/>

User Property	Master Value	Local Value	Display
IvsIgnore	TRUE		off <input type="checkbox"/>

CDF Parameter	Value	Display
Resistance	50 Ohms	off <input type="checkbox"/>
Port number		off <input type="checkbox"/>
DC voltage		off <input type="checkbox"/>
Source type	pwl <input type="checkbox"/>	off <input type="checkbox"/>
Frequency name 1		off <input type="checkbox"/>
Waveform Entry Method	File <input checked="" type="checkbox"/> Voltage/Time points <input type="checkbox"/>	off <input type="checkbox"/>
File name		off <input type="checkbox"/>
Delay time		off <input type="checkbox"/>
DC offset		off <input type="checkbox"/>
Amplitude scale factor		off <input type="checkbox"/>
Time scale factor		off <input type="checkbox"/>
Breakpoints	<input type="checkbox"/>	off <input type="checkbox"/>
Period		off <input type="checkbox"/>
Transition width		off <input type="checkbox"/>
Display small signal params	<input type="checkbox"/>	off <input type="checkbox"/>
Display temperature params	<input type="checkbox"/>	off <input type="checkbox"/>
Display noise parameters	<input type="checkbox"/>	off <input type="checkbox"/>
Multiplier		off <input type="checkbox"/>

Waveform Entry Method

You can specify piecewise-linear data in two ways, either by specifying a *File name* or by entering a series of *Voltage/Time points*.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

File name

If you select *Waveform Entry Method=File*, you enter the name of the file containing the piecewise-linear data in the form of time-value pairs. The value must be a string. Default: no value

In your file, list the time-value pairs as one pair per line with a space or tab between the time and voltage values. The numbers in the file must be given as simple numbers - SI scale factors (p, n, u, m, k, M, G, etc.) cannot be used.

Figure H-5 *Waveform Entry Method = File*

Source type	pwl <input type="checkbox"/>	off <input type="checkbox"/>
Frequency name 1	<input type="text"/>	off <input type="checkbox"/>
Waveform Entry Method	◆ File ◇ Voltage/Time points	off <input type="checkbox"/>
File name	<input type="text"/>	off <input type="checkbox"/>
Delay time	<input type="text"/>	off <input type="checkbox"/>
DC offset	<input type="text"/>	off <input type="checkbox"/>
Amplitude scale factor	<input type="text"/>	off <input type="checkbox"/>
Time scale factor	<input type="text"/>	off <input type="checkbox"/>
Breakpoints	<input type="checkbox"/>	off <input type="checkbox"/>
Period	<input type="text"/>	off <input type="checkbox"/>
Transition width	<input type="text"/>	off <input type="checkbox"/>

Number of PWL/Time pairs

If the *Waveform Entry Method=Voltage/Time points*, you specify the *Number of voltage-time pairs (tvpairs)*. The form expands to let you to type in the designated *Voltage* and *Time* values. Units: v and seconds. Default: 0 Maximum value: 50

In the following figure, the number of voltage-time pairs is set to 3.

Affirma RF Simulator (SpectreRF) User Guide
Using port in SpectreRF Simulations

Figure H-6 *Waveform Entry Method = Voltage/Time points*

Source type	<input type="text" value="pwl"/>	<input type="checkbox"/>
Frequency name 1	<input type="text"/>	<input type="checkbox"/>
Waveform Entry Method	<input type="radio"/> File <input checked="" type="radio"/> Voltage/Time points	<input type="checkbox"/>
Number of PWL/Time pairs	<input type="text"/>	<input type="checkbox"/>
Time 1	<input type="text"/>	<input type="checkbox"/>
Voltage 1	<input type="text"/>	<input type="checkbox"/>
Time 2	<input type="text"/>	<input type="checkbox"/>
Voltage 2	<input type="text"/>	<input type="checkbox"/>
Time 3	<input type="text"/>	<input type="checkbox"/>
Voltage 3	<input type="text"/>	<input type="checkbox"/>
Delay time	<input type="text"/>	<input type="checkbox"/>
DC offset	<input type="text"/>	<input type="checkbox"/>
Amplitude scale factor	<input type="text"/>	<input type="checkbox"/>
Time scale factor	<input type="text"/>	<input type="checkbox"/>
Breakpoints	<input type="checkbox"/>	<input type="checkbox"/>
Period	<input type="text"/>	<input type="checkbox"/>
Transition width	<input type="text"/>	<input type="checkbox"/>

The following *pwl* parameters: *DC offset*, *Amplitude scale factor*, and *Time scale factor* let you quickly adjust the amplitude, frequency and offset without editing each individual point in the *pwl* waveform.

DC offset

DC offset (offset) for the *pwl* waveform. Default: 0 Units: V

Amplitude scale factor

Amplitude scale factor (scale) for the *pwl* waveform. Default: 1

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Time scale factor

Time scale factor (stretch) for the time given for the *pwl* waveform. Default: 1

Breakpoints

Possible values are *no*, *yes*, or blank. If you set *Breakpoints (allbrkpts)* to *yes*, you force Spectre to place time points at each point specified in a *pwl* waveform during a transient analysis. This can be very expensive for waveforms with many points. If you set *Breakpoints* to *no*, Spectre inspects the waveform, looking for abrupt changes, and forces time points only at those changes. If you set *Source type = pwl* and set *Breakpoints* to blank, the default is *yes* if the number of points you specify is less than 20.

Period

The *pwl* waveform is periodic if you specify *Period (pwlperiod)*. Units: seconds

If the value of the waveform you specify is not exactly the same at both its beginning and its end, then you must provide a nonzero value for *Transition Width*.

Transition Width

Transition width (twidth) is used when making *pwl* waveforms periodic. Default: $PWL\ period/1000$. Units: seconds

Before repeating, the waveform changes linearly in an interval of *Transition Width* from its value at $(Period - Transition\ Width)$ to its value at the beginning of the waveform. Thus the *Transition Width* must always be less than the *Period*.

Sinusoidal Waveform Parameters

The *port* component can generate up to two sinusoids simultaneously. They are denoted as 1 and 2. You can set the amplitude, frequency and phase for both individually. The amplitude can be set to either a voltage or a power level. You can also specify sinusoidal AM or FM modulation of sinusoid 1.

The Edit Object Properties form for *Source type=sine* is shown below.

The first sinusoid is described by the parameters *Frequency name 1*, *Frequency 1*, *Amplitude 1 (Vpk)*, *Amplitude 1 (dBm)*, *Phase for sinusoid 1*, *Sine DC level*, *AM or FM modulation terms*, and *Damping factor 1*.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Figure H-7 *Source type=sine* in the Edit Object Properties form

The screenshot shows the 'Edit Object Properties' dialog box for a port object. The 'Source type' is set to 'sine'. The dialog is organized into several sections:

- Buttons:** OK, Cancel, Apply, Defaults, Previous, Next, Help.
- Apply To:** only current, instance.
- Show:** system, user, CDF.
- Instance Labels:** Browse, Reset Instance Labels Display.
- Property Table:**

Property	Value	Display
Library Name	analogLib	<input type="checkbox"/> off
Cell Name	port	<input type="checkbox"/> off
View Name	symbol	<input type="checkbox"/> off
Instance Name	PORT	<input type="checkbox"/> off
- User Property Table:**

User Property	Master Value	Local Value	Display
Ivsignore	TRUE		<input type="checkbox"/> off
- CDF Parameter Table:**

CDF Parameter	Value	Display
Resistance	50 Ohms	<input type="checkbox"/> off
Port number		<input type="checkbox"/> off
DC voltage		<input type="checkbox"/> off
Source type	<input checked="" type="checkbox"/> sine	<input type="checkbox"/> off
Frequency name 1		<input type="checkbox"/> off
Frequency 1		<input type="checkbox"/> off
Amplitude 1 (Vpk)		<input type="checkbox"/> off
Amplitude 1 (dBm)		<input type="checkbox"/> off
Phase for Sinusoid 1		<input type="checkbox"/> off
Sine DC level		<input type="checkbox"/> off
Delay time		<input type="checkbox"/> off
Display second sinusoid	<input type="checkbox"/>	<input type="checkbox"/> off
Display modulation params	<input type="checkbox"/>	<input type="checkbox"/> off
Display small signal params	<input type="checkbox"/>	<input type="checkbox"/> off
Display temperature params	<input type="checkbox"/>	<input type="checkbox"/> off
Display noise parameters	<input type="checkbox"/>	<input type="checkbox"/> off
Multiplier		<input type="checkbox"/> off

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Frequency name 1

Names the fundamental tones of sinusoid 1. After you save the schematic, the names you assign appear in the *Fundamental Tones* list box on the Choosing Analyses form.

Frequency 1

The frequency of the first sinusoidal waveform (carrier frequency). You typically use unmodulated signals in SpectreRF analyses. The value must be a real number. Default: 0
Units: Hz

Amplitude 1 (Vpk)

Peak amplitude of the first sinusoidal waveform that you generate. The value specified is the voltage delivered into a matched load. You can select either *Amplitude 1 (Vpk)* or *Amplitude 1 (dBm)*, but not both. If *Amplitude 1 (Vpk)* has a value, the *Amplitude 1 (dBm)* field is grayed out. The value must be a real number. Default: 1 Units: V

Remember, when you specify the voltage on a *port*, you are specifying the voltage *when the port is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Amplitude 1 (dBm)

Amplitude of the first sinusoidal waveform, in dBm. The value specified is the power delivered into a matched load. You can select either *Amplitude 1 (Vpk)* or *Amplitude 1 (dBm)*, but not both. If *Amplitude 1 (dBm)* has a value, the *Amplitude 1 (Vpk)* field is grayed out. The value must be a real number. Units: dBm

Phase for Sinusoid 1

The phase at the specified *Delay time*. The sinusoidal waveform might start before the given *Delay time* in order to achieve a specified phase and still remain continuous. For example, if you want to generate a cosine wave, set this parameter to 90°. The value must be a real number. Default: 0 Units: degrees

Sine DC level

Sets the DC level for sinusoidal waveforms in transient analyses. This parameter is used when the sinusoid has a different average level than the one specified for the DC analyses. If not specified, the average value of the sinusoid is the same as that of the DC level of the source. The value must be a real number. Default: dc Units: V

Modulation Parameters

Display Modulation Parameters

When selected, the form expands and following modulation parameters are displayed: *FM modulation index 1*, *FM modulation freq 1*, *AM modulation index 1*, *AM modulation freq 1*, *AM modulation phase 1*, and *Damping factor 1*.

Note: *Only* the first sinusoid can be modulated.

Figure H-8 Display modulation parameters in Edit Object Properties form

Display modulation params	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
FM modulation index 1	<input type="text"/>	off <input type="checkbox"/>
FM modulation freq 1	<input type="text"/>	off <input type="checkbox"/>
AM modulation index 1	<input type="text"/>	off <input type="checkbox"/>
AM modulation freq 1	<input type="text"/>	off <input type="checkbox"/>
AM modulation phase 1	<input type="text"/>	off <input type="checkbox"/>
Damping factor 1	<input type="text"/>	off <input type="checkbox"/>

FM Modulation (Background Information)

The frequency modulation for the sinusoidal case is defined as

$$v_{FM}(t) = A \sin(2\pi f_c t + \beta \sin(2\pi f_m t) + \phi)$$

where

- A is the amplitude of sinusoid 1
- β is the FM modulation index
- $\sin(2\pi f_m t)$ is the modulation signal
- f_c is the carrier frequency

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

- ϕ is the *phase for sinusoid 1*

The frequency modulation parameters affect *only* the first sinusoid generated by *port*. They have no effect on the second sinusoid.

FM modulation frequency 1

FM modulation frequency for the sinusoidal waveform (f_m in the previous equation). The value must be a real number. Default: 0 Units: Hz

FM modulation index 1

FM index of modulation for the sinusoidal waveform, the ratio of peak frequency deviation divided by the center frequency (β in the above equations).

$$\beta = \Delta f / f_m$$

The value must be a real number. Default: 0

Effect of Amplitude Modulation (Background Information)

The amplitude modulation (double sideband suppressed carrier, or DSB-SC) is defined as

$$v_{AM}(t) = A (1 + m \sin(2\pi f_m t + \phi)) \sin(2\pi f_c t)$$

where

- A is the carrier amplitude (amplitude of sinusoid 1)
- m is the AM modulation index
- f_m is the AM modulation frequency
- ϕ is the AM modulation phase
- $\sin(2\pi f_c t)$ is the carrier signal

The amplitude modulation parameters affect *only* the first sinusoid generated by *port*. They have no effect on the second sinusoid.

AM modulation frequency 1

AM modulation frequency for the first sinusoidal waveform (f_m in the previous equation). The value must be a real number. Default: 0 Units: Hz

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

AM modulation phase 1

AM phase of modulation for the first sinusoidal waveform (ϕ in the previous equation). The value must be a real number. Default: 0 Units: degrees

AM modulation index 1

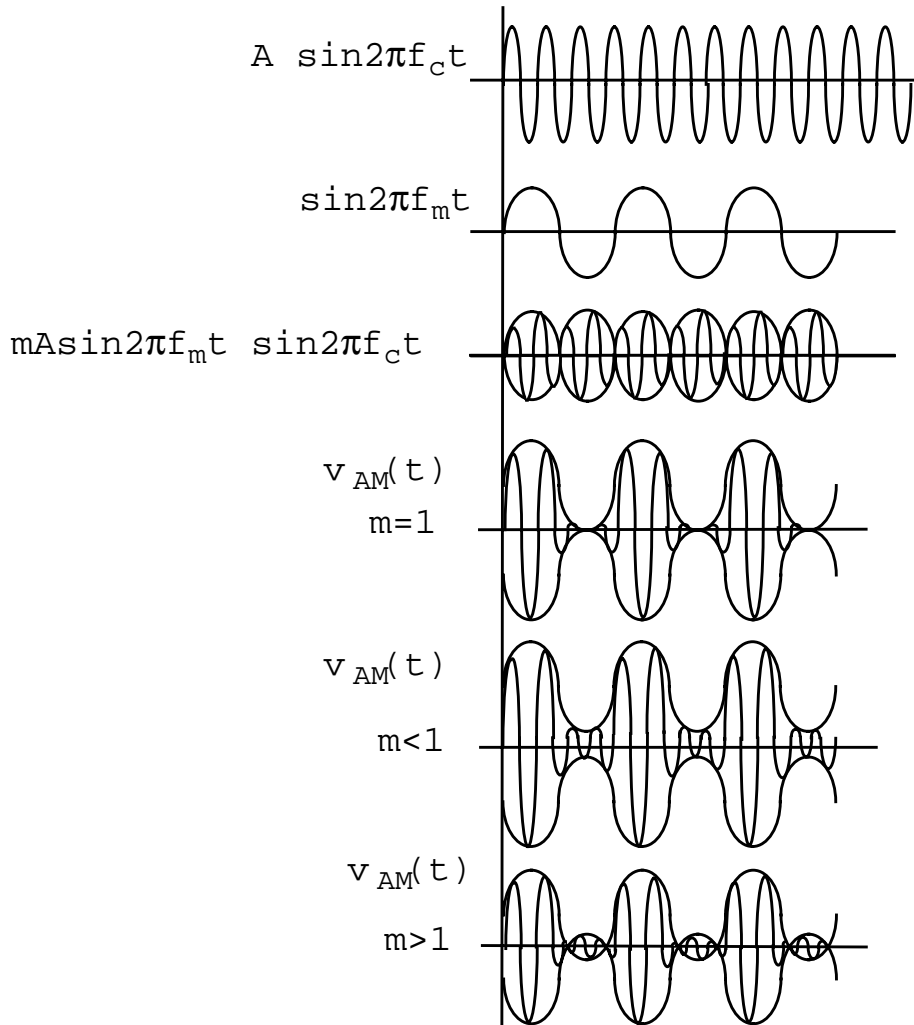
AM index of modulation for the first sinusoidal waveform (m in the previous AM equation). The *AM modulation index 1* is a dimensionless scale factor used to control the ratio of the sidebands to the carrier.

$$m = (\text{peak DSB-SC amplitude})/(\text{peak carrier amplitude})$$

The value must be a real number. Default: 0

The following figure shows the effect of varying modulation indexes for the following three cases: $m < 1$, $m = 1$, and $m > 1$. f_c is the carrier frequency, and f_m is the modulation frequency.

Figure H-9 Amplitude Modulation: Effects of Varying Modulation Indexes



Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Damping factor 1

Damping factor for the sinusoidal waveform. *Damping factor 1* specifies the time it takes to go from the envelope (full amplitude) at *time=0* to 63 percent of the full amplitude. For example, consider the following damped sinusoid:

$$v(t) = A e^{-\sigma t} \sin(2\pi f t + \phi)$$

where

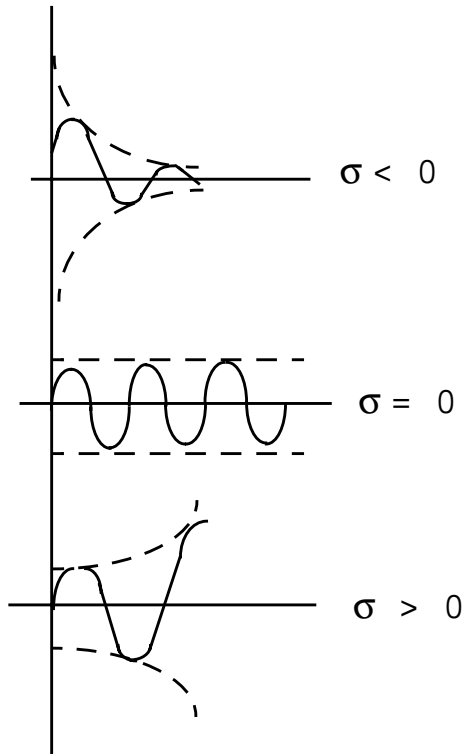
$\sigma =$ *Damping factor 1*, A is the amplitude of sinusoid 1, and ϕ is the *Phase for sinusoid 1*

- If $\sigma = 0$, the waveform is a pure sinusoid (steady state).
- If $\sigma < 0$, the waveform exhibits decaying oscillations.
- If $\sigma > 0$, the waveform exhibits growing oscillations.

It takes 5σ to diminish to 1 percent of the peak amplitude. The value must be a real number. Default: 0. Units: 1/seconds

The following figure shows the effect of *Damping factor 1* on the first sinusoid for three values of σ .

Figure H-10 Effect of Damping Factor 1 on the First Sinusoid



Display second sinusoid

Displays the CDF parameters for the second sinusoid in the Edit Object Properties and Add Instance forms. When selected, the form expands to show the following CDF parameters: *Frequency name 2*, *Frequency 2*, *Amplitude 2 (Vpk)*, *Amplitude 2 (dBm)*, and *Phase for Sinusoid 2*.

Note: The second sinusoid cannot be modulated.

Figure H-11 Display second sinusoid in the Edit Object Properties form

Display second sinusoid	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
Frequency name 2	<input type="text"/>	off <input type="checkbox"/>
Frequency 2	<input type="text"/>	off <input type="checkbox"/>
Amplitude 2 (Vpk)	<input type="text"/>	off <input type="checkbox"/>
Amplitude 2 (dBm)	<input type="text"/>	off <input type="checkbox"/>
Phase for Sinusoid 2	<input type="text"/>	off <input type="checkbox"/>

Frequency name 2

Name for the second sinusoid. After you save the schematic, the name you assign appears in the Fundamental Tones list box on the Choosing Analyses form.

Frequency 2

Frequency of the second sinusoidal waveform. The value must be a real number. Default: 0
Units: Hz

Amplitude 2 (Vpk)

Peak amplitude of the second sinusoidal waveform. The value specified is the voltage delivered into a matched load. You can select either *Amplitude 2 (Vpk)* or *Amplitude 2 (dBm)*, but not both. If *Amplitude 2 (Vpk)* has a value, the *Amplitude 2 (dBm)* field is grayed out. The value must be a real number. Default: 1 Units: V

Remember, when you specify the voltage on a *port*, you are specifying the voltage *when the port is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Amplitude 2 (dBm)

Amplitude of the second sinusoidal waveform, in dBm. The value specified is the power delivered into a matched load. You can select either *Amplitude 2 (Vpk)* or *Amplitude 2*

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

(dBm), but not both. If *Amplitude 2 (dBm)* has a value, the *Amplitude 2 (Vpk)* field is grayed out. The value must be a real number. Units: dBm

Phase for Sinusoid 2

The phase at the specified *Delay time* for the second sinusoid. The sinusoidal waveform might start before the given *Delay time* in order to achieve specified phase while still remaining continuous. The value must be a real number. Default: 0 Units: degrees

Exponential Waveform Parameters

To generate an exponential waveforms from the `port` component, set the CDF parameter *Source type=exp*, as shown in the next figure.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Figure H-12 *Source type=exp* in Edit Object Properties form

The screenshot shows the 'Edit Object Properties' dialog box. At the top, there are buttons for 'OK', 'Cancel', 'Apply', 'Defaults', 'Previous', 'Next', and 'Help'. Below these are 'Apply To' options: 'only current' (selected) and 'instance'. The 'Show' section has radio buttons for 'system', 'user' (selected), and 'CDF'. There are 'Browse' and 'Reset Instance Labels Display' buttons. The main area is divided into three sections:

Property	Value	Display
Library Name	analogLib	off
Cell Name	port	off
View Name	symbol	off
Instance Name	PORT	off

Below this is a 'User Property' section with 'Add', 'Delete', and 'Modify' buttons. It has columns for 'User Property', 'Master Value', 'Local Value', and 'Display'.

User Property	Master Value	Local Value	Display
lvignore	TRUE		off

The bottom section is 'CDF Parameter' with columns for 'CDF Parameter', 'Value', and 'Display'.

CDF Parameter	Value	Display
Resistance	50 Ohms	off
Port number		off
DC voltage		off
Source type	exp	off
Delay time		off
Zero value		off
One value		off
Rise time start		off
Rise time constant		off
Fall time start		off
Fall time constant		off
Display small signal params	<input type="checkbox"/>	off
Display temperature params	<input type="checkbox"/>	off
Display noise parameters	<input type="checkbox"/>	off
Multiplier		off

Remember, when you specify the voltage on a *port*, you are specifying the voltage *when the port is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*.

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Zero value

The *Zero value* (*val0*) used in *pulse* and *exp* waveforms. Default: 0 Units: V

One value

The *One value* (*val1*) used in *pulse* and *exp* waveforms. Default: 1 Units: V

Rise time start

Rise start time (*td1*) for the *exp* waveform. Default: 0 Units: seconds

Rise time constant

Rise time constant (*tau1*) for the *exp* waveform. Units: seconds

Fall time start

Fall start time (*td2*) for the *exp* waveform. Units: seconds

Fall time constant

Fall time constant (*tau2*) for the *exp* waveform. Units: seconds

Noise Parameters

Noise parameters include *Noise temperature*, *Noise Entry Method*, *Noise file name*, and *Number of Noise Frequency Pairs*.

Noise temperature

The *Noise temperature* of the *port*. If not specified, the *Noise temperature* is assumed to be the actual temperature of the *port*. When you compute the noise figure of a circuit driven at its input by a *port*, set the *Noise temperature* of the *port* (Spectre parameter *noisetemp*) to 16.85C (290K). This setting matches the standard IEEE definition of noise figure. In addition, disable all other sources of noise in the *port*, such as the Spectre parameters *noisefile* and *noisevec*. If you want a noiseless *port*, set the *Noise temperature* to absolute zero or below, and do not specify a noise file or noise vector. Default: Actual temperature of the port. Units: °C

Noise Entry Method

You can select one of two ways to enter noise data, either by specifying a *Noise file name* or entering a series of *Noise/Frequency points*.

Noise file name

If *Noise Entry Method* = *File* is selected, you enter the name of the file containing the excess spot noise data in the form of frequency-noise pairs. In your file, list the frequency-noise pairs as one pair per line with a space or tab between the frequency and noise values. The value must be a string. Default: none

Figure H-13 Display noise parameters: Noise Entry Method=File

Display noise parameters	<input checked="" type="checkbox"/>	<input type="checkbox"/> off
Noise temperature	<input type="text"/>	<input type="checkbox"/> off
Noise Entry Method	<input checked="" type="radio"/> File <input type="radio"/> Noise/Frequency points	<input type="checkbox"/> off
Noise file name	<input type="text"/>	<input type="checkbox"/> off

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

Num. of noise/freq pairs

If the *Noise Entry Method=Noise/Frequency points*, you specify the *Number of noise/freq pairs*. The form expands to let you type in the designated *Freq* and *Noise* values. The noise values must be in V^2/Hz , and frequency in Hz. Default: 0 Maximum value: 10

The example in the next figure has the *Number of noise/freq pairs* set to 3.

Figure H-14 Noise parameters: *Noise Entry Method=Noise/Freq Points*

Display noise parameters	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
Noise temperature	<input type="text"/>	off <input type="checkbox"/>
Noise Entry Method	<input type="radio"/> File <input checked="" type="radio"/> Noise/Frequency points	off <input type="checkbox"/>
Num. of noise/freq pairs	<input type="text" value="3"/>	off <input type="checkbox"/>
Freq 1	<input type="text"/>	off <input type="checkbox"/>
Noise 1	<input type="text"/>	off <input type="checkbox"/>
Freq 2	<input type="text"/>	off <input type="checkbox"/>
Noise 2	<input type="text"/>	off <input type="checkbox"/>
Freq 3	<input type="text"/>	off <input type="checkbox"/>
Noise 3	<input type="text"/>	off <input type="checkbox"/>

Small-Signal Parameters

Display small signal params

If selected, the Edit Object Properties/Add Instance form expands to show the small-signal parameters *PAC magnitude*, *PAC magnitude (dBm)*, *PAC phase*, *AC magnitude*, *AC phase*, and *XF magnitude*.

Figure H-15 Display small signal parameters selected

Display small signal params	<input checked="" type="checkbox"/>	off <input type="checkbox"/>
PAC Magnitude	<input type="text"/>	off <input type="checkbox"/>
PAC Magnitude (dBm)	<input type="text"/>	off <input type="checkbox"/>
PAC phase	<input type="text"/>	off <input type="checkbox"/>
AC Magnitude	<input type="text"/>	off <input type="checkbox"/>
AC phase	<input type="text"/>	off <input type="checkbox"/>
XF Magnitude	<input type="text"/>	off <input type="checkbox"/>

Remember, when you specify the voltage on a *port*, you are specifying the voltage *when the port is properly terminated*, and not the voltage on the internal voltage source. Thus, the voltage on the internal source is set to twice the value specified on the *port*. The same is true for the values for the transient, AC, and PAC signals. However, the amplitude of the sine wave in the PAC and transient analysis can alternatively be specified as the power in dBm delivered by the *port when terminated with the reference resistance*.

PAC Magnitude

The peak periodic AC analysis magnitude. Setting this value to unity is a convenient way of computing the transfer function from this source to the output.

You can select either *PAC magnitude* or *PAC magnitude (dBm)*, but not both. If *PAC magnitude* has a value, the *PAC magnitude (dBm)* field is grayed out. The value must be a real number. Default: 0 Units V

PAC Magnitude (dBm)

The periodic AC analysis magnitude in dBm (alternative to *PAC magnitude*). You can select either *PAC magnitude* or *PAC magnitude (dBm)*, but not both. If *PAC magnitude (dBm)* has a value, the *PAC magnitude* field is grayed out. The value must be a real number. Units: dBm

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

PAC phase

The periodic AC analysis phase. The value must be a real number. Default: 0 Units: degrees

Typically, only one source in the circuit has a *PAC magnitude* set to a value other than zero, and usually it has a *PAC magnitude*=1 and *PAC phase*=0. However, there are situations where more than one source has a nonzero *PAC magnitude*. For example, applying a differential small-signal input could be done with two sources with the *PAC magnitudes* set to 0.5 and the *PAC phases* set to 0 and 180.

You do not specify the PAC frequency in the *port* Edit Object Properties form. Instead, you set the PAC frequency in the *PAC Choosing Analysis* form. For example, when making an IP3 measurement, you set the PAC frequency to a variable value in the *Choosing Analysis* Form. Then, you enter the same variable in the *PAC Amplitude* (or *PAC Amplitude dBm*) field of the *port* Edit Object Properties form.

AC Magnitude

The peak small-signal voltage. The value must be a real number. Default: 0 Units: V

AC phase

The small-signal phase. The value must be a real number. Default: 0 Units: degrees

Typically, only one source in the circuit has *AC Magnitude* set to a value other than zero, and usually it has an *AC magnitude*=1 and *AC phase*=0. However, there are situations where more than one source has a nonzero *AC magnitude*. For example, you can apply a differential small-signal input with two sources with the *AC magnitudes* set to 0.5 and the *AC phases* set to 0 and 180.

XF Magnitude

The transfer function analysis magnitude. Use *XF magnitude* to compensate for gain or loss in the test fixture. The value must be a real number. Default: 1 Units: V/V

Temperature Effect Parameters

Temperature effect parameters include the *Linear temperature coefficient*, *Quadratic temperature coefficient*, and *Nominal temperature*.

Display temperature params

If selected, the Edit Object Properties/Add Instance form expands and the following three parameters appear in the form: *Linear temp. coefficient*, *Quadratic temp. coeff.*, and *Nominal temperature*.

Figure H-16 Display temperature parameters selected

Display temperature params <input checked="" type="checkbox"/>		off <input type="checkbox"/>
Linear temp. coefficient	<input type="text"/>	off <input type="checkbox"/>
Quadratic temp. coeff.	<input type="text"/>	off <input type="checkbox"/>
Nominal temperature	<input type="text"/>	off <input type="checkbox"/>

Linear temp. coefficient

First order (linear) temperature coefficient of the *DC voltage* ($tc1$). The value must be a real number. Default: 0 Units: $^{\circ}\text{C}^{-1}$

Quadratic temp. coeff.

Second order (quadratic) temperature coefficient of the *DC voltage* ($tc2$). The value must be a real number. Default: 0 Units: $^{\circ}\text{C}^{-2}$

Nominal temperature

The *Nominal temperature* for *DC voltage* ($tnom$). The value must be a real number. Default: Set by options specifications. Units: $^{\circ}\text{C}$

Affirma RF Simulator (SpectreRF) User Guide

Using port in SpectreRF Simulations

How Temperature Parameters Affect the Voltage Level (Background Information)

The value of the *DC voltage* can vary as a function of the temperature if you specify *tc1* and *tc2*. The variation is given by

$$V_{DC}(T) = dc * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]$$

where *T* is the analysis temperature specified in the analysis options, *tnom* is the *Nominal temperature* specified in the *Choosing Analyses* form, and *dc* is the *DC voltage*.

If the analysis temperature equals the nominal temperature, the result is the voltage amplitude that you specified, $V(T)=dc$.

If the nominal and analysis temperatures differ, the voltage amplitude is given by

$$V_{DC}(T) = dc * [1 + tc1 * (T - tnom) + tc2 * (T - tnom)^2]$$

where *T* is the analysis temperature you specify in the analysis options and *tnom* is the nominal temperature, *tc1* and *tc2* are the *Linear* and *Quadratic temperature coefficients*, and *dc* is the *DC voltage*.

For example, if the *Nominal temperature* is 27°C and the analysis temperature is 25°C, there is a 2° difference between the nominal and analysis temperature. The voltage amplitude is

$$V_{DC}(T) = dc * [1 + tc1 * (-2) + tc2 * (-2)^2]$$

Additional Notes

Active Parameters in Analyses

In DC analyses, the only active parameters are *dc*, *m*, and the temperature coefficient parameters.

In AC analyses, the only active parameters are *m*, *mag*, and *phase*.

In transient analyses, all parameters are active except the small-signal parameters and the noise parameters.

In PAC, the only active parameters are *m*, *PAC magnitude* (amplitude or dBm), and *PAC phase*.

XF magnitude is active in XF and PXF analyses only.

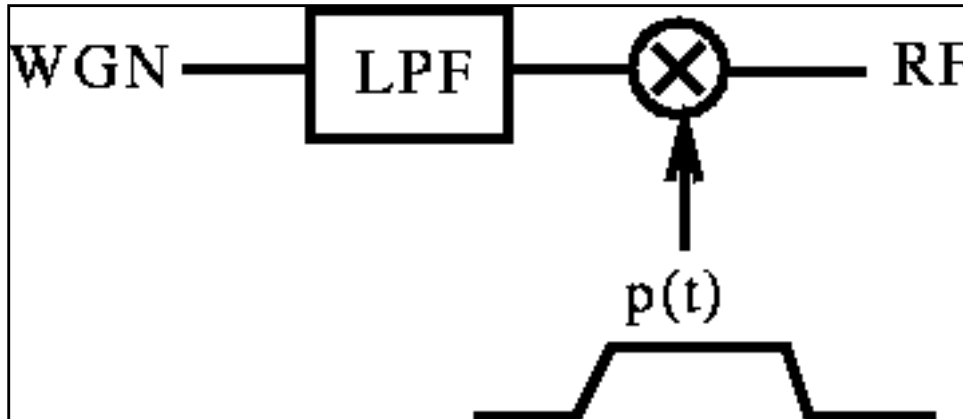
Analyzing Time-Varying Noise Properties

RF circuits are usually driven by periodic inputs. Therefore, the noise in RF circuits is generated by sources that can usually be modeled as periodically time-varying. Periodically time-varying noise is said to be cyclostationary. To permit more detailed characterization of cyclostationary noise in RF circuits, the Noise Type parameter has been added to the SpectreRF Pnoise analysis.

- Noise Type *sources* computes the total time-average noise at an output over a given frequency range. Each noise source's contribution to the total noise is computed at each frequency. This option is the default and represents the previous Pnoise functionality.
- Noise Type *timedomain* computes the time-varying instantaneous noise power in a circuit with periodically driven components.
- Noise Type *correlations* computes correlations between the noise at different ports of a multi-port circuit. For example, you can compute the correlation between the noise at different outputs, or the correlation between the noise at the input and output of a circuit that exhibits frequency conversion. You can extract equivalent noise sources from these calculations.

Time-Domain Noise Characterization

Figure I-1 Very simple mixer schematic

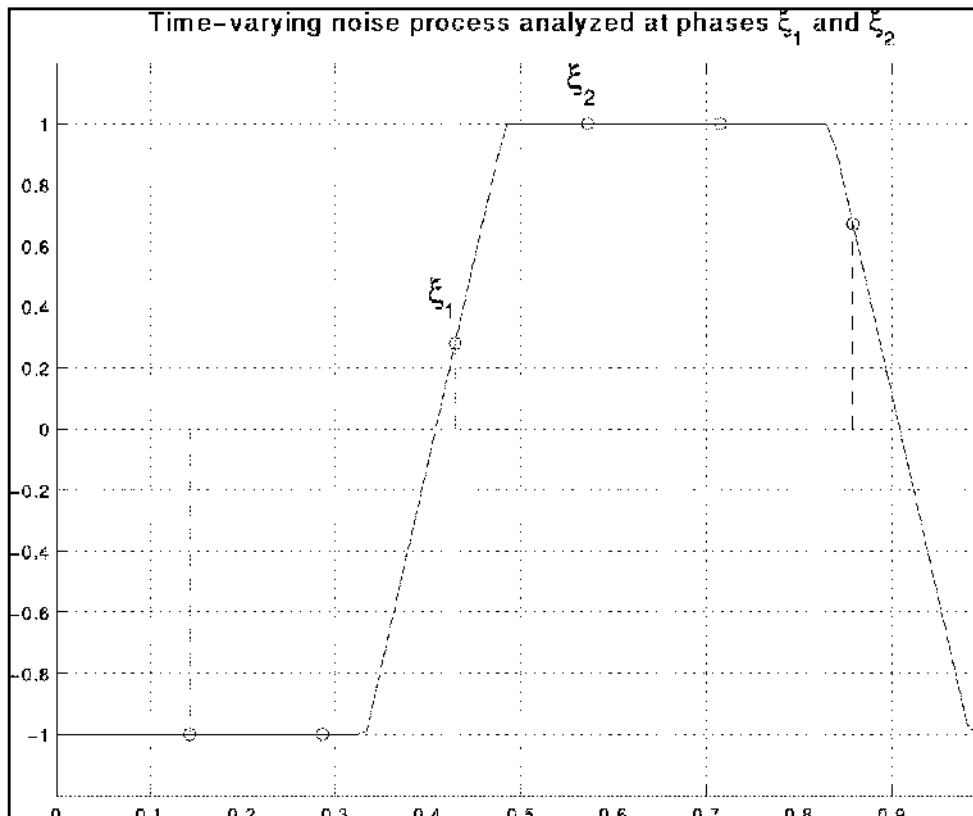


Noise in periodically driven circuits, with period T , for example, exhibits statistical properties that also vary periodically. To understand the time-domain characterization of noise, consider the simple circuit shown in Figure I-1. The amplitude of the noise at the RF output varies periodically depending on the magnitude of the modulating signal $p(t)$. This is shown by the sample points on Figure I-2 below. In the figure, the solid line shows the envelope $p(t)$ that modulates the noise process. The circles show possible phase points on the envelope where the time-varying noise power might be calculated.

Affirma RF Simulator (SpectreRF) User Guide

Analyzing Time-Varying Noise Properties

Figure I-2 Representation of periodically-modulated noise.



SpectreRF can calculate the time-varying noise power at any point in the fundamental period. In fact, SpectreRF can calculate the full autocorrelation function $R^\xi(p, q) = E\{x^\xi(p)x^\xi(p + q)\} = R^\xi(q)$ and its spectrum for the discrete-time processes x^ξ obtained by periodically sampling the time-domain noise process at the same phase point. Figures I-3 and I-4 show noise processes for two different phases in the periodic interval. Each process is stationary.

Figure I-3 Noise Process for First Phase

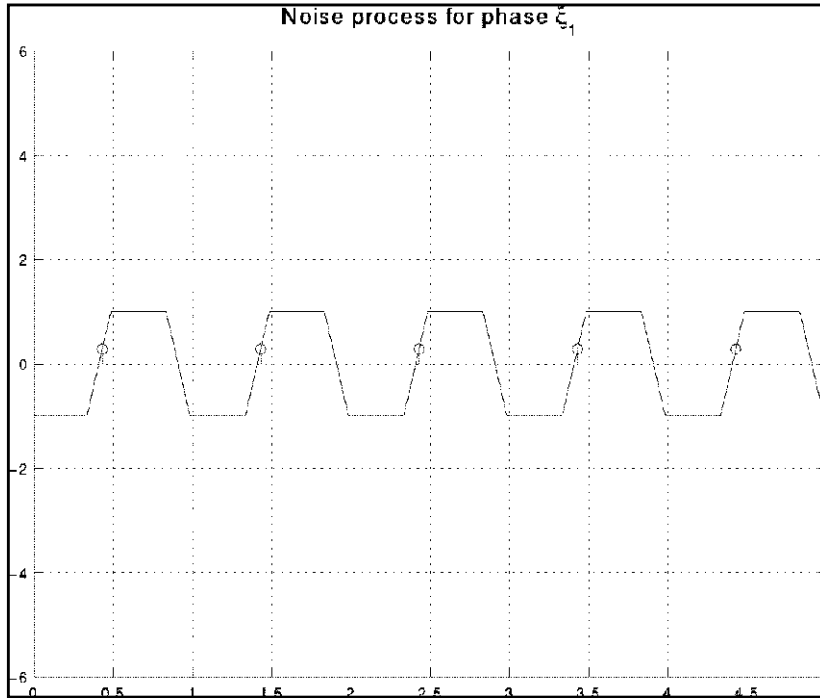
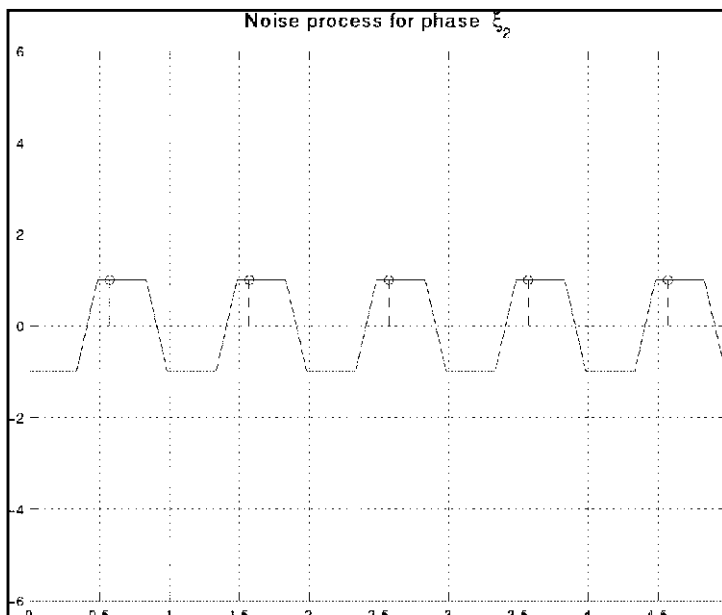


Figure I-4 Noise Process for Second Phase



Affirma RF Simulator (SpectreRF) User Guide

Analyzing Time-Varying Noise Properties

The following steps tell you how to calculate time-domain noise using SpectreRF.

1. In a terminal window, type `icms` to start the environment.

2. In the Simulation window, select *Analyses – Choose*.

The Choosing Analyses form appears.

3. In the Choosing Analyses form, highlight `pss` and perform the PSS analysis setup.

4. In the Choosing Analyses form, highlight *pnoise*.

The Choosing Analyses form changes to let you specify information for a Pnoise analysis.

5. In the Choosing Analyses form, perform the following:

a. Choose noisetype *timedomain*.

b. Specify an appropriate frequency range and sweep for the analysis.

You might, for example, perform a linear sweep up to the fundamental frequency. Because each time point in the calculation is a separate frequency sweep, use the minimum number of frequency points possible to resolve the spectrum. This step minimizes computation time.

c. Specify a *noiseskipcount* or additional explicit timepoints.

d. Specify an appropriate set of timepoints for the time-domain noise analysis. If you set *noiseskipcount* to an integer p , then the noise is calculated at every $p+1$ points. *noiseskipcount=0* calculates the noise at every timepoint in the final PSS solution and is the default. You can add specific points by specifying a time relative to the start of the PSS simulation interval. *noiseskipcount=5* performs noise calculations for about 30 points in the PSS interval. If you only need a few points, add them explicitly and set *noiseskipcount* to a large value like 1000.

6. In the Simulation window, choose *Simulation – Netlist and Run*.

The simulation runs.

7. In the Simulation window, choose *Results – Direct Plot – PSS*.

The PSS Results form appears.

8. To calculate time-varying noise power, perform the following steps in the PSS Results form:

a. Click on *tdnoise* and then select *Integrated noise power*.

Calculating Noise Correlation Coefficients

In multi-input/multi-output systems, to characterize the noise, it is necessary to calculate the noise power at each of the ports as well as the correlation between the noise at various ports. The situation is complicated in RF systems because the “ports” might be at different frequencies. For example, in a mixer, the input “port” might be at the RF frequency and the output “port” at the IF frequency.

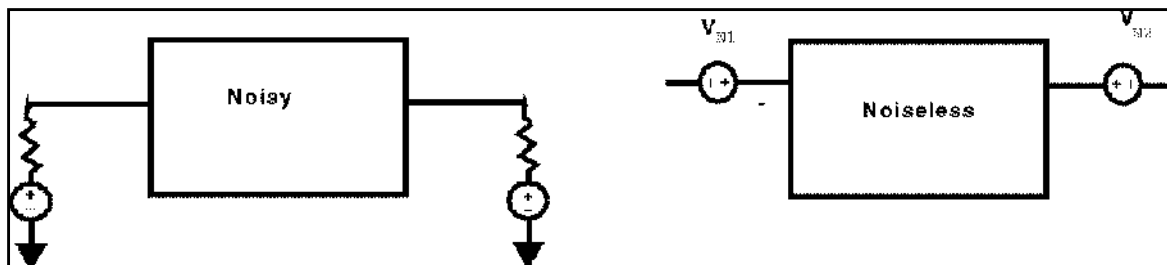
The power spectrum of a signal x is denoted by $S_{XX}(\omega)$, that is

$$S_{XX} = X^*(\omega)X(\omega)$$

where $X(\omega)$ is the Fourier transform of the signal $x(t)$. For random signals like noise, the expected value of the power spectrum is $S_{XX}(\omega)$. To characterize the relation between two separate signals $x(t)$ and $y(t)$, the cross-power spectrum below is also needed

$$S_{XY} = X^*(\omega)Y(\omega)$$

Figure I-5 Calculating noise correlations and equivalent noise parameters



For random signals, the degree to which x and y are related is given by the cross-power spectrum. A correlation coefficient can be defined as

$$\rho_{XY}(\omega) = \frac{S_{XY}(\omega)}{[S_{XX}(\omega)S_{YY}(\omega)]^{1/2}}$$

A correlation coefficient of 0 means the signals are completely uncorrelated, and a correlation coefficient of 1 means they are perfectly correlated. For example, a signal is always perfectly correlated with itself.

You might also want to consider correlations between noise at different frequencies. The quantity S_{XY}^α expresses the correlation of a signal x at frequency ω with the signal y at frequency $\omega + \alpha$. For example, white Gaussian noise is completely uncorrelated with itself

Affirma RF Simulator (SpectreRF) User Guide

Analyzing Time-Varying Noise Properties

for $\alpha \neq 0$. Noise in an RF system generally has $S^\alpha(\omega)$ non-zero when α is a multiple of the fundamental frequency, such as for the LO frequency in a mixer.

Once the noise properties of a circuit are measured, these properties can be represented as a noiseless multiport with equivalent noise sources. For example, in Figure 3, the noise voltage appearing at the excitation ports is first measured. Then the noise properties of the circuit can be expressed as two equivalent frequency-dependent noise voltages V_{N1} and V_{N2} , and a (complex) correlation coefficient ρ_{12} .

The following steps describe how to calculate noise correlation parameters for a two-port circuit.

1. In a terminal window, type `icms` to start the environment.
2. In the Simulation window, select *Analyses – Choose*.

The Choosing Analyses form appears.

3. In the Choosing Analyses form, highlight `pss` and set up the PSS analysis.
4. In the Choosing Analyses form, highlight *proise*.

The Choosing Analyses form changes to let you specify information for a Proise analysis.

5. In the Choosing Analyses form, perform the following:
 - a. Choose *correlations* for the noisetype.
 - b. Choose the first output of the circuit.

This can be a port or voltage if you want to calculate noise parameters starting in the impedance representation. It can be a current if you want to use the admittance representation. For example, in a mixer, you might choose the IF output here.

- c. Choose an appropriate frequency range and sweep for the analysis.

For example, for the mixer you might want to sweep 100MHz around the IF output. Choose an appropriate set of “cycles” to use for the analysis. In a mixer, you might set `maxcycles=1` or add cycle “1” if you select the IF as output. This specification calculates noise correlations between the IF output at frequency ω and the noise at the RF input frequency $\omega + \omega_0$, where ω_0 is the LO frequency.

6. In the Simulation window, choose *Simulation – Netlist and Run*.

The simulation runs.

7. To display the results, perform the following steps:

Affirma RF Simulator (SpectreRF) User Guide

Analyzing Time-Varying Noise Properties

- a. In the Simulation window, choose *Tools – Results Browser*.

The Results Browser appears.

- b. In the Results Browser, choose the noise analysis and the appropriate cycle of interest. (Correlations of noise at different frequencies appear as different “harmonics” in the browser.)

A list of nodes and sources appears.

The correlation between the noise at the output you specify in the Simulation window and the noise induced at each of the nodes or sources has been calculated. You can plot it directly or further process it in the calculator.

To analyze the correlations with noise at a differential input/output, subtract the cross-power spectra appearing on the two nodes.

Note: Results of the noise correlations analysis are in units of power (such as V^2/Hz), and the cross-power spectra are usually complex.

Using Tabulated S-parameters

Models of many passive components commonly used in RF design are only available as tables of S-parameter data. Any linear, time-invariant circuit network or component can be completely characterized by specifying its S-parameter network at each frequency of interest. The “Using the *n*port Component” section tells how to use S-parameter tables in RF analyses.

SpectreRF uses time-domain shooting methods to achieve excellent performance on large, highly nonlinear circuits. Consequently, using S-parameter data in SpectreRF is not as straightforward as it is with frequency-domain simulators such as those based on harmonic balance. Before performing an RF simulation such as PSS analysis, you must first convert the frequency-domain S-parameters into an equivalent time-domain model. This conversion might be time-consuming for large, complicated data sets. However, the *n*port component lets you avoid performing the conversion process more than once. The procedure is described later in this paper.

There are three potential difficulties in converting to a time-domain model.

First, not every set of frequency domain data has a valid time-domain description. Time-domain models must be stable and causal, and you cannot generate a time-domain model for frequency-domain data that lacks these properties.

Secondly, to convert the s-parameter data into a time-domain description, the frequencies between the tabulated points must be interpolated. This is true because the time-domain description of a frequency-domain model depends on every frequency, not just the ones given at the tabulated points. A special, robust, high-order, rational interpolation algorithm performs the interpolation. The rational interpolation process introduces some error into the final model description. See the “Controlling Model Accuracy” section to understand and control this error. This section also tells you how to deal with data that contains noise that may corrupt the rational interpolation process.

Finally, any algorithm that converts S-parameters so they can be used in a time-domain simulator must extrapolate the data outside the range of tabulated frequencies. By definition, some frequencies are not included in a tabular data file. In addition, extrapolation might introduce nonphysical effects into the model, particularly when the S-parameter data is only given over a very narrow frequency range. The “Troubleshooting” section tells you how to

diagnose and solve any problems. It also tells how to interpret the warning messages that SpectreRF produces when a non-physical extrapolation might be occurring.

Using the *nport* component

You read S-parameter data into using the *nport* component. Follow these steps to prepare an *nport* component for use in SpectreRF simulations.

1. Place the appropriate component from *analogLib*, such as *n2port* for a two-port S-parameter description, in the schematic.
2. Select the *nport* component and then edit the object properties.
3. In the first field of the Edit Object Properties form ("S-parameter data file"), type the name of the file that contains the tabulated S-parameter data, such as *sparam1.dat*. As of the 4.4.5 release, S-parameter files must be in the "Spectre" format. If your S-parameter data is in another industry standard format, you must use the **sptr** tool to convert the data to a format that Spectre can read.
4. Set the interpolation method to *rational* so you can use SpectreRF analyses such as PSS and PNoise. If you only plan to use non-RF analyses such as TRAN, DC, and SP, you can select *spline* instead.
5. [Optional] Specify a path for a "ROM data file." This must be an absolute pathname, for example, */usr/mydir/sparamtorom*. Once SpectreRF converts the S-parameter data to a time-domain description, this file stores the time-domain model. Storing the model ensures that interpolation and conversion need not be repeated. See the "Model Reuse" section on page -918 for further details on this file. Be sure this file is distinct from the S-parameter data file.
6. [Optional] If you know the form of your data, the amount of noise in the data, or the accuracy you require from the S-parameter interpolation, fill in the *relerr*, *abserr* or *ratorder* fields as described in the "Controlling Model Accuracy" section..
7. Apply the changes to the *nport* component and run the simulation.

Controlling Model Accuracy

The *nport* component has three parameters to control the accuracy of the rational interpolation process, *relerr*, *abserr*, and *ratorder*. You can use these parameters to trade accuracy for model size. Usually, the more stringent the accuracy requirement, the higher the model order. Higher-order models require longer simulation time. Spectre can automatically

Affirma RF Simulator (SpectreRF) User Guide

Using Tabulated S-parameters

generate a model that meets a specified accuracy requirement. You can also specify the model order directly to Spectre.

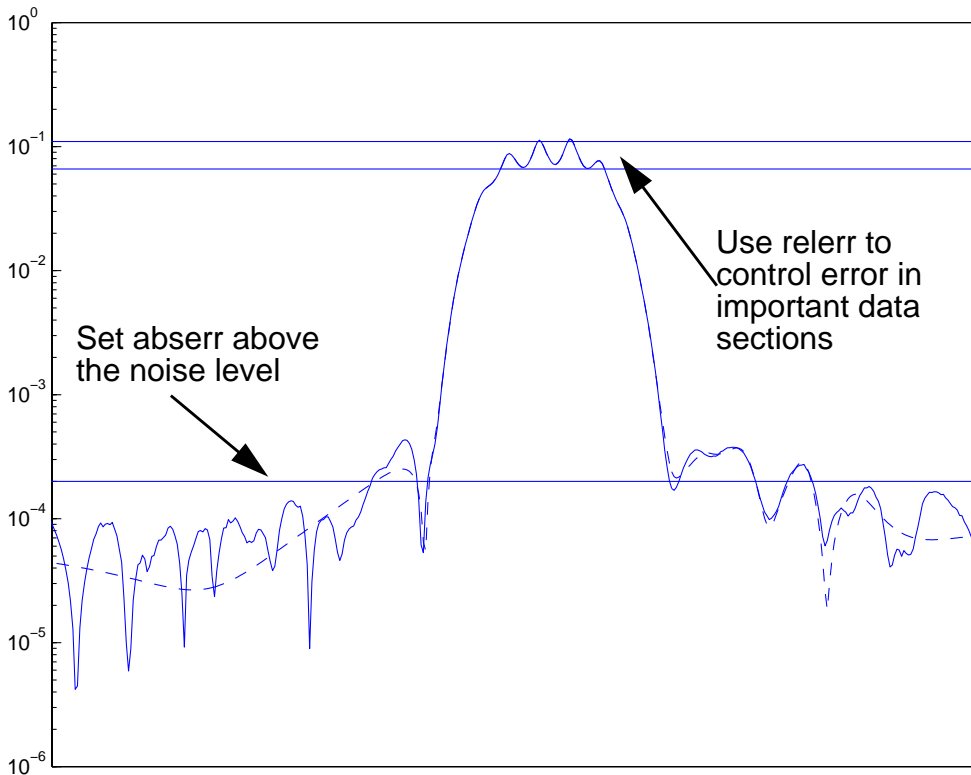
Using *relerr* and *abserr*

Let $S_{ij}(\omega)$ denote the i,j entry of the scattering parameter matrix at frequency ω and $\hat{S}_{ij}(\omega)$ denote the corresponding rational interpolant. The rational interpolation algorithm attempts to find an interpolant such that

$$\max_{\omega, i, j} |S_{ij}(\omega) - \hat{S}_{ij}(\omega)| < \max(\text{relerr} \times |S_{ij}(\omega)|, \text{abserr})$$

Generally, you use *relerr* for relative error control, and *abserr* for absolute error control. Consider the data shown by the solid line in the figure below. In this example, *relerr* can control the error in the passband, and *abserr* can control the error in the stopband. If the details of the filter behavior in the stopband are not of interest, *abserr* can be set to the level shown, and these details are ignored in the interpolation.

Figure J-1 Using *relerr* and *abserr*



This example has approximately a 10% ripple in the passband, so to ensure adequate error control there, set *relerr* to much less than 0.1, perhaps to 0.01.

The dashed line in the figure shows an interpolation function with the error control levels specified in the figure: *abserr*= $2e-4$, and *relerr*=0.01. Details of the data below the *abserr* threshold are not resolved by the rational interpolation.

To recover the details in the stopband, set *abserr* to about $1e-5$ for this example.

Note: If there is noise in the data, set *relerr* and *abserr* above the respective noise levels. Otherwise, Spectre attempts to interpolate the noise, resulting in very high order models and very long simulation times.

Remember that if *relerr* is set to zero, then, from the formula above, pure absolute error is used. Conversely, if *abserr* is set to zero, the error control is based solely on the errors relative to the magnitude of the input data.

Using the *ratorder* parameter

It is usually a good idea to specify only the accuracy parameters *relerr* and *abserr*, and let Spectre automatically select the order of the rational approximation. However, if you have special information about your data set, you can direct Spectre to use a specific order of approximation in the rational interpolation. For example, if you know that your tabulated S-parameter data represents a sixth-order filter, you might instruct spectre to use a seventh or eighth order fit. The slightly higher order gives Spectre flexibility to adjust for any noise or non-ideal behavior in the data.

When you specify the *ratorder* parameter, *relerr* and *abserr* are used to warn you if the order you selected is not sufficient to meet the accuracy requirements. Otherwise, the parameters are not used.

Troubleshooting

Some data sets can cause difficulty for the rational interpolation process. Types of data to avoid are

1. Data specified only over a very narrow frequency range.
Extrapolation of the data outside this range, needed for time-domain simulation, might be difficult and is always risky.
2. Very noisy data.
Such data might lead to large, unreliable time-domain models.
3. Data on a very sparse frequency grid.
Accurate interpolation of such data might be impossible.
4. Data with long ideal delays.
5. Data representing idealized lossless elements, such as lossless transmission lines.

Assessing the quality of the rational interpolation

If you suspect a problem with the rational interpolation process, you can investigate it using the *sp* analysis with the following steps:

- Construct a test schematic consisting of an *nport* component with interpolation set to rational, as discussed in the “Using the *nport* Component” section.
- Next add the appropriate number of port components to the schematic.

Affirma RF Simulator (SpectreRF) User Guide

Using Tabulated S-parameters

- Perform an *sp* analysis on the *nport* component and look for anomalies.

Large swings in interpolated values and S-parameter magnitudes greater than one suggest a problem. Large changes in the interpolant result from an inaccurate fit, and S-parameter values greater than one result from a non-passive (energy-generating) model that might create unstable time-domain solutions. Be particularly critical of anomalies near zero frequency (DC).

If the anomalous behavior occurs inside the frequency range given in the S-parameter data file, it usually indicates an inaccurate rational interpolation. Be sure all of the conditions given above are met. Try decreasing *relerr* or *abserr* or both, or specifying a higher-order interpolation with the *ratorder* parameter. Remember that measured data can contain fine details that might require a higher order than you might expect from a casual inspection of the data.

If the anomalous behavior occurs outside the tabulated frequency range, try changing the *abserr*, *relerr*, or *ratorder* parameters. Sometimes anomalies can be removed by using a more accurate fit. However, there are limits to the ability to extrapolate outside the frequency interval you specify. You might need to specify additional data points to fix the problem.

Model Reuse

The ROM data file feature of the *nport* lets you use the same S-parameter component description across many designs, but only perform the conversion to a time-domain model once.

For a given S-parameter data set, after you have specified a location for the ROM data file, and Spectre performs the conversion to a time-domain model and writes the model file, you can reuse the model file in future simulations. To reuse the model, you enter its name as the "ROM data file" in the new design. At this point, you no longer must specify a raw S-parameter data file.

If you accidentally specify both an S-parameter data file and a ROM data file, SpectreRF reuses the model in the ROM data file if the two files are consistent. Otherwise, a new model is generated from the raw tabulated S-parameter data, and the ROM data file overwritten with a new model. This feature lets you specify both the ROM data file and the S-parameter data file in a design when you first place the *nport* component. SpectreRF then automatically generates the time-domain model for the first simulation and reuses it for all subsequent simulations without needing to change the *nport* component parameters.

If you require a more accurate rational interpolation, then you must regenerate the model in the ROM data file by changing the *relerr*, *abserr* or *ratorder* fields as described in the "Controlling Model Accuracy" section.

References

To learn technical details about how SpectreRF converts S-parameter data to a time-domain description, see the article "Robust rational function approximation algorithm for model generation," by C. P. Coelho, J. R. Phillips, and L. M. Silveira. This article appeared in the proceedings of the 36th Design Automation Conference, New Orleans, LA, June 1999.

Using the Calculator IPN Functions for RF Simulations

This appendix describes how to

- Use the IPN functions in the Analog Design Environment Waveform Calculator.
- Use the IPN functions with scalar inputs.

This appendix is designed for users of SpectreRF in the Affirma Analog Design Environment and assumes that users are familiar with SpectreRF PSS, PAC, and Swept PSS analyses.

How to Use the *ipn* and *ipnVRI* Functions

This example uses the *ne600* schematic from the *rfExamples* library. As you follow the example, you combine a swept PSS analysis with a PAC analysis to produce data for an IP3 plot. Then you will plot IP3 using the *ipn* and *ipnVRI* functions in the Calculator and compare the results.

Setting Up

1. Start *icms*.
2. Copy the *ne600* schematic from *rfExamples* to your local library (so you will have edit permissions on the schematic).
3. Open the *ne600* schematic for edit.
4. From the schematic banner, Select **Tools — Analog Environment**. This opens the Analog Circuit Design Simulation environment.
5. Open the **Setup — Simulator/Directory/Host** form and choose `spectre` as the simulator and set *Host Mode* to `local`.
6. In the Model Library Setup form (**Setup — Model libraries**), set the model library to

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

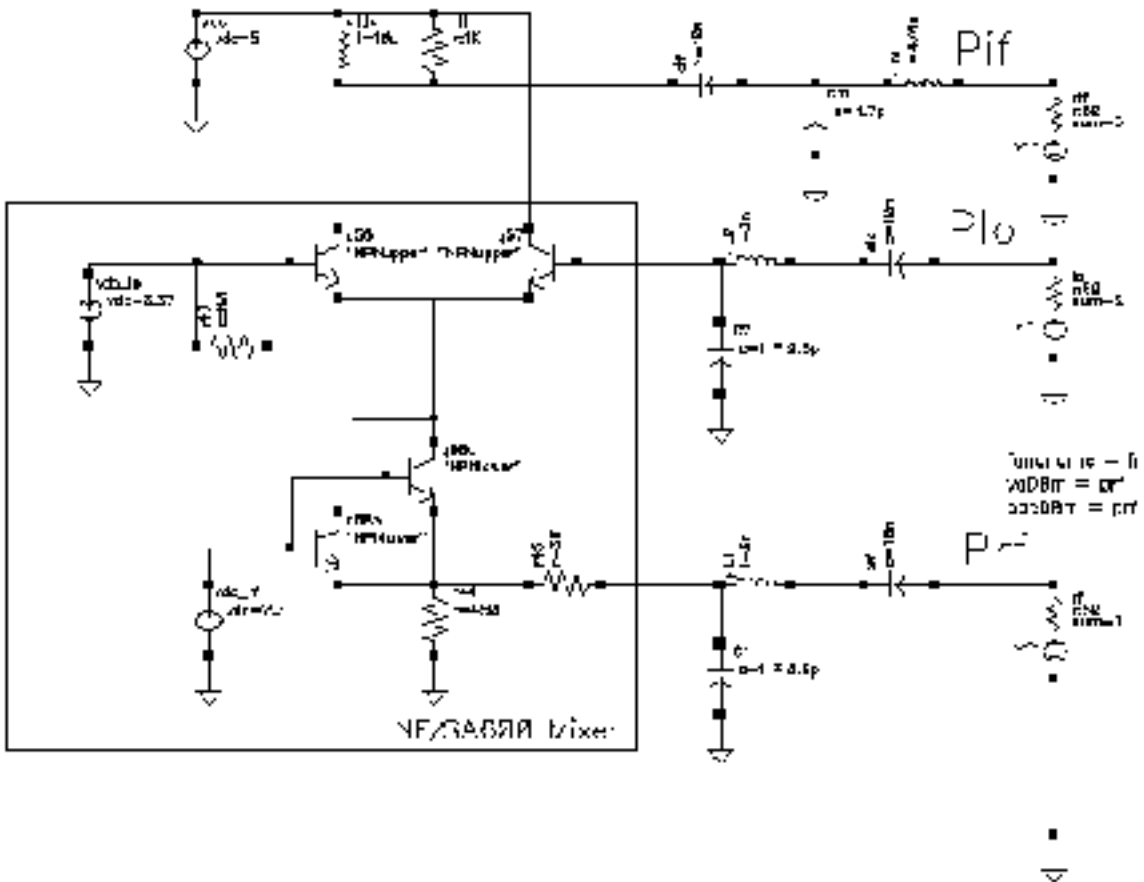
<path_to>/tools/dfII/samples/artist/models/spectre/rfModels.scs

7. Set the Design Variable f_{rf} to 920M (**Variables — Edit** form).
8. In the Analog Circuit Design Simulation window, disable any analyses you may have run previously.

Editing the Schematic

Modify the schematic for this simulation.

Figure K-1 The ne600 Schematic



1. Click on the rf psin source in the Schematic window and edit its properties.
2. In the Edit Object Properties form, do the following and click **OK**.

- a. Type `sine` for *Source type*.
- b. Type `frf` for *Frequency*.
- c. Type `prf` in the *Amplitude (dBm)* and *PAC magnitude (dBm)* fields.

Be sure the *Amplitude*, *Amplitude 2* (magnitude and dBm), and *Frequency 2* fields are empty. This simulation uses dBm values rather than magnitude.

3. Click on the `lo psin` source and edit its properties. Verify that the *sourceType* is set to `sine`.
4. In the schematic, select **Design — Check and Save**.

Setting Up the Analysis

1. In the Choosing Analyses form (**Analyses — Choose**), select `pss` for the analysis.
2. In the Fundamental Tones list box, do the following and click *OK*.

- a. select the *Auto Calculate* button.
- b. The value `40M` is specified as the Beat Frequency.

The PAC analysis is responsible for the two tones, and the PSS is now a single signal analysis. Consequently, the fundamental frequency is set to `40 MHz`.

- c. In the Output harmonics section, select *Number of harmonics* and enter `2` into the field.
- d. Click on `conservative` for the *Accuracy Defaults (errpreset)* setting.
- e. Click on the *Sweep* button.

The form changes to let you specify data for the sweep.

- In the *Sweep* option, select `variable`.
 - Set *Frequency Variable?* to `no`.
 - Click on the *Select Design Variable* button.
 - Highlight `prf` in this form and click *OK*.
3. Select *Start-Stop* for the *Sweep Range* value, and then type `-25` and `5` for the *Start* and *Stop* values, respectively. The sweep is skewed downward for a down converter.
 4. Select `Linear` for the *Sweep Type*, and specify *Step size* = `5`.

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

The completed swept PSS form should look like this:

The screenshot shows a software window titled "kv300" with a menu bar containing "OK", "Cancel", "Defaults", "Apply", and "Help". The main interface is divided into several sections:

- General Section:** Contains radio buttons for "Best Frequency" (selected) and "Best Period". A text box contains "914" and a "Auto Update" checkbox is checked.
- Default Run Units:** A section with a "Number of harmonics" text box containing "4".
- Accuracy Defaults (approx):** Radio buttons for "conservative" (selected), "moderate", and "liberal". Below is a text box for "Add Guard Time for Stabilization (ticks)" and a "Save Initial Transient Results (convert)" checkbox with "NO" and "YES" options.
- Default Run:** A checkbox labeled "Default Run" which is unchecked.
- Sweep F1:** A section with a "Variable" text box, a "Frequency Variable?" radio button with "no" selected and "yes" unselected, a "Variable Name" text box containing "L", and a "Select Design Variable" button.
- Sweep Range:** Radio buttons for "start-stop" (selected) and "Center Span". Text boxes for "start" (containing "25") and "Stop" (containing "3").
- Sweep Type:** Radio buttons for "Linear" (selected) and "Logarithmic". Text boxes for "Stop Size" (containing "0") and "Number of Steps" (containing "0").
- Additional Options:** A text box for "Add Harmonic Modes" and a "Disabled" checkbox.
- Footer:** A "Continue..." button.

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

5. At the top of the Choosing Analyses form, click on `pac`.

In the Choosing Analyses form, do the following:

- a. Select the *Enabled* button.
- b. Under *Frequency Sweep Range*, type `921M` for the *Frequency* (Hz). Because the sweep section of the PSS analysis is active, only a single point for this analysis is currently supported.
- c. In the *Sidebands* option button, select *Array of indices* and enter in the values: `-21` and `-25`.

Given a fundamental tone of 40MHz, the LO at 1GHz, and two RF tones at 920MHz and 921MHz, the sidebands of `-25` and `-21` represent respectively the first-order harmonic of the IF output at 79MHz ($921 - 25 * 40 = 79$) and the third-order harmonic at 81MHz ($921 - 21 * 40 = 81$).

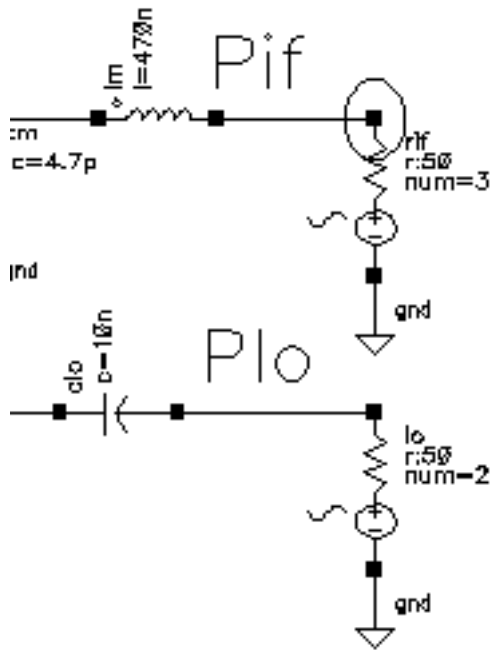
Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

6. The completed PAC section looks like this. *OK* the form

The image shows a dialog box for configuring a Periodic AC Analysis. At the top, there are buttons for 'OK', 'Cancel', 'Defaults', 'Apply', and 'Help'. Below these are radio buttons for selecting the analysis type: 'tran', 'dc', 'ac', 'noise', 'xt', 'sens', 'sp', 'pdisto', 'pss', 'pac' (which is selected), 'pnoise', 'pxf', and 'envlp'. The main section is titled 'Periodic AC Analysis' and contains a text box for 'PSS start frequency (Hz)' with the value '401'. Below this is a 'Sweep type' dropdown menu. The 'Frequency Sweep Range (Hz)' section includes a 'Single-Point' checkbox (checked) and a 'Freq' text box with the value '901K'. A message states: 'Because the sweep option of the PSS analysis is active, only a single point for this analysis is currently supported.' The 'STIMULUS' section has an 'Array of indices' dropdown, a 'Currently active indices' text box, and an 'Additional indices' text box containing '-21 -25'. At the bottom, there is an 'Enabled' checkbox (checked) and an 'Options...' button.

7. Select **Outputs — To Be Saved — Select on Schematic**. Click on the positive node of the *rif* psin.



Running the Simulation

Select **Simulation — Netlist and Run**. This example compares two signals only 1MHz apart.

Plotting the IP3 Curve Using Direct Plot

1. Select **Results — Direct Plot — PSS**. In the PSS Results form, do the following:
 - a. Click on *Replace* for *Plot Mode*.
 - b. Click on *pac* for *Analysis Type*.
 - c. Click on *IPN Curves* for *Function*.
 - d. Choose *Variable Sweep ("prf")* for *Circuit Input Power*.

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

- e. Type -15 for *Extrapolation Point*. This value is the intercept point for the ideal amplification extrapolation.
- f. Select *Input Referred IP3* and *Order 3rd*.
- g. Highlight -21 81M in the *3rd Order Sideband* list box.
- h. Highlight -25 79M in the *1st Order Sideband* list box.

Affirma RF Simulator (SpectreRF) User Guide

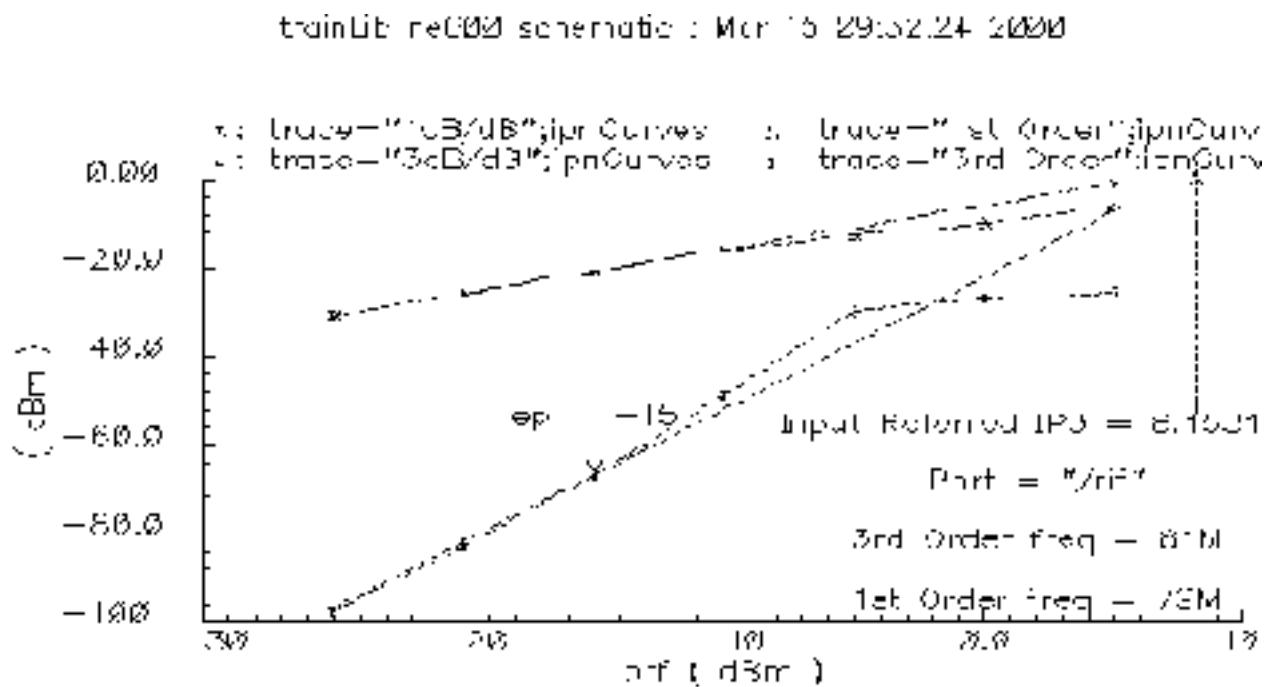
Using the Calculator IPN Functions for RF Simulations

Note that if you select the *Add to Outputs* button, the *Input Referred IP3 Direct Plot* expression is also saved to the *Outputs* pane in the Simulation Window. You will do this in "Using the IPN Function with Scalar Inputs" on page 934.

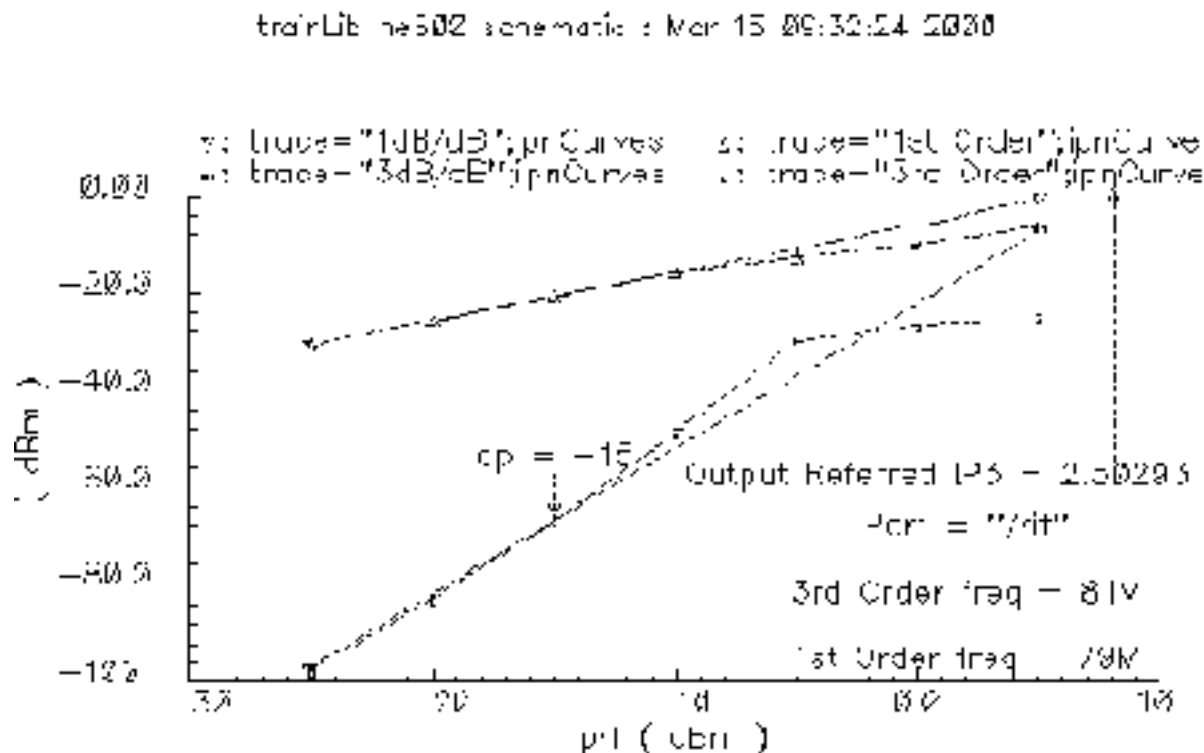
- In the schematic, click on the *rif* port.

Note that if you click on the *Pif* net, you will get an IPN plot with dBV on the vertical axis.

The Waveform window display appears as shown below



- You can also select Output Referred IP3 on the PSS Results Direct Plot form. Note that in this case, the measured value is different, depending on whether you select a port (dBm) or a net (dBV) on the schematic. Output Referred IP3 is plotted below.



Plot IP3 Using the ipn and ipnVRI Buttons in the Waveform Calculator

First, you will use the *ipn* button to plot IP3. Note that the Calculator *ipn* function only does the *Input Referred* calculation.

- From the Waveform Window or the Simulation window, select **Tools— Calculator**.

In this example, you will use the Calculator in RPN mode. If the Calculator is in Algebraic mode, you can change it by selecting **Options — Set RPN**.

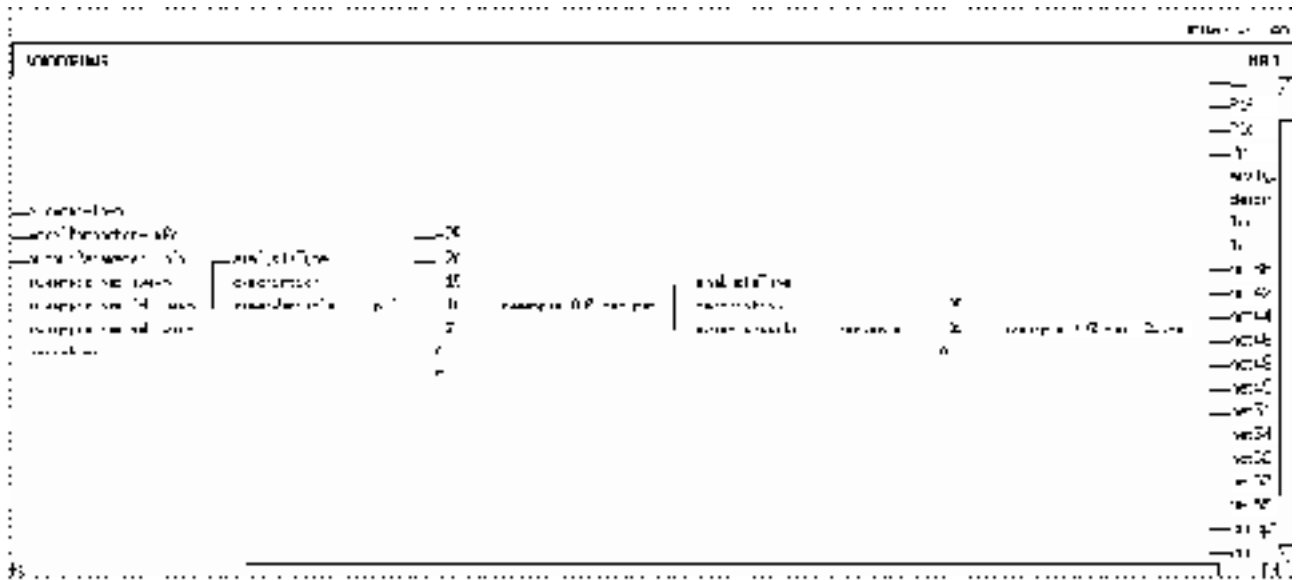
- From the Calculator, open the Results Browser by selecting *browser*. When the Browse Project Hierarchy form appears, click OK. In the Results Browser, click on:

```
schematic->psf->Run1->sweep_pss_pac-sweep->
sweepVariable->prf-> -10->sweep_pss-003_pac-pac->
```

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

```
sweepVariable->harmonic-> -21->
sweeppss-003_pac--21.pac-> Pif
```



3. Enter the information into the Calculator buffer. It looks similar to this:

```
v( "/Pif" ?result "sweeppss_pac-sweep" ?resultsDir
"/usr1/mnt4/tawna/simulation/ne600/spectre/schematic" )
```

Important note: Clicking on any harmonic (–0, –21, or –25) gives the same results. The Results Browser shows the basic structure of the data and will only give you the *Pif* data (voltage and current information).

4. In the Calculator, select **Special Functions — ipn**. Fill in the form as follows and OK the form

IPN				
OK	Cancel	Defaults	Apply	Help
Spur Order	<input style="width: 40px;" type="text" value="3"/>	Spur Harmonic	<input style="width: 40px;" type="text" value="-21"/>	
Extrapolation Point	<input style="width: 40px;" type="text" value="-15"/>	Reference Harmonic	<input style="width: 40px;" type="text" value="-25"/>	

5. The Calculator buffer has an expression like the one below:

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

```
ipn(dB20(harmonic(v( "/Pif" ?result "sweepss_pac-sweep" ?resultsDir
"/usr1/mnt4/tawna/simulation/ne600/spectre/schematic" ), -21)),
dB20(harmonic(v( "/Pif" ?result "sweepss_pac-sweep"
?resultsDir "/usr1/mnt4/tawna/simulation/ne600/spectre/schematic" ), -25)),
3,1,-15,-15)
```

6. Now you have two choices, either:

- Click *Evaluate Buffer*

Or,

- Cut and paste the Calculator buffer into the CIW.

The function evaluates to 8.15314. This number agrees with the IP3 value in the Waveform Window.

Next, you will use the *ipnVRI* button in the Calculator to plot IP3.

1. You can use the **Special Functions— ipnVRI** to obtain the Input Referred or Output Referred IP3 value.
2. Click on the following in the Results Browser:

```
schematic->psf->Run1->sweepss_pac-sweep-> sweepVariable->prf->
-10->sweepss-003_pac-pac-> sweepVariable->harmonic->
-21->sweepss-003_pac--21.pac->Pif
```

Note that you should select a net voltage only when dealing with the Calculator. The Load Resistance field is used in conjunction with this voltage to calculate power (V^2/R).

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

3. Select **Special Functions — ipnVRI** from the Waveform Calculator and fill out the form as shown below and **OK** the form

Note that if you want to plot dBV using the *ipnVRI* function, then set the Load Resistance to nil.

The Calculator buffer has an expression like the one below:

```
ipnVRI(v( "/Pif" ?result "sweepss_pac-sweep" ?resultsDir
"/usr1/mnt4/tawna/simulation/ne600/spectre/schematic" ),-21,-25,
?ordspur 3,?epoint -15,?rport 50)
```

4. Click on the *Evaluate Buffer* button.

The expression evaluates to 8.1534.

There is yet another way to get the expression used by the Direct Plot form:

1. Go to the Waveform Window and select **Annotation — Edit**.
2. Click on *Input Referred IP3*. In the expression field, it shows you the expression for that label.

```
ipnVRI(v("/Pif" ?result "pac")-0. '-21 '-25 ?rport
resultParam("rif:r" ?result "pac") ?epoint -15)
```

You can enter this expression into the Calculator buffer and select *Evaluate Buffer* to obtain the IP3 value. The Direct Plot expression uses the *ipnVRI* function, which is used to simplify the declaration of an IPN measurement.

Using the IPN Function with Scalar Inputs

This example uses the *ne600* schematic from the *rfExamples* library.

Two reasons to use the IPN function with scalar inputs (assuming region of operation is linear) are:

- simulation is faster
- IPN optimization is faster

When you use scalar inputs, always do a final IPN verification with a power sweep (non-scalar inputs), as outlined in the previous example.

Setting Up

- Start *icms* and edit the schematic as described in the previous example. See “[Setting Up](#)” on page 920 for more information.

Setting Up the Analysis

1. In the Choosing Analyses form (**Analyses — Choose**), select `pss` for the analysis.

The analysis will be set up similar to the previous example, except this time the *Sweep* button is disabled.

- a. In the Fundamental Tones list box, select the *Auto Calculate* button
- b. In the *Output harmonics* option button, select *Number of harmonics = 2*.
- c. Click on `conservative` for the *Accuracy Defaults* (errpreset) setting.

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

The PSS analysis form looks like this

OK Cancel Defaults Apply Help

Analyses User Auto New Previous Off

Series Ssp Analysis Pss Pnl

Previous Post Analy

Periodic Steady State Analysis

Fundamental Tones

#	Name	Order	Value	Signal	Used
1	f_lo	hlo	1s	Moderate	no
2	f_hi	hhi	1s	Moderate	no

: Moderate

Real Frequency Real Period

400

 Auto Calculate

Output Parameters

Number of Harmonics:

Accuracy Defaults (recommended)

conservative moderate liberal

Additional Time for Stabilization (delay):

Save Initial Transient Results (recommended): no yes

Oscillator

Sweep

Enabled
Options...

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

2. At the top of the Choosing Analyses form, click on `pac`.
 - a. Set the *Frequency Sweep Range* to *Single-Point*, and *freq* to 921M.
 - b. In the *Sidebands* option, select *Array of indices* and type in the *Additional indices* field: -21 and -25.
 - c. Given a fundamental tone of 40MHz, the LO at 1GHz, and two RF tones at 920MHz and 921MHz, the sidebands of -25 and -21 represent respectively the first-order harmonic of the IF output at 79MHz ($921 - 25 \cdot 40 = 79$) and the third-order harmonic at 81MHz ($921 - 21 \cdot 40 = 81$).

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

3. The completed PAC section looks like this. *OK* the form

The screenshot shows the 'Periodic AC Analysis' dialog box in the Affirma RF Simulator. At the top, there are buttons for 'OK', 'Cancel', 'Default', 'Apply', and 'Help'. Below these are radio buttons for various analysis types: 'tran', 'dc', 'ac', 'noise', 'sfs', 'pnoise', 'prf', 'envelope', 'modsb', 'psa', and 'pac'. The 'pac' option is selected. The 'Periodic AC Analysis' section contains a text box for 'PAC Period Frequency (Hz)' with the value '1000'. Below this is a 'Sweep type' dropdown menu. The 'Frequency Sweep Range (Hz)' section has a 'Single-Point' dropdown and a 'req' text box with the value '2015'. There is an 'Add Specific Points' checkbox. The 'Subbands' section has a 'Array of Indices' dropdown, a 'Currently active indices' text box, and an 'Additional indices' text box containing '-90 -75'. At the bottom, there is an 'Enabled' checkbox and an 'Options...' button.

4. Edit the *Design Variables*, and make sure that the `prf` design variable is set to the extrapolation value that was used for the IPN plot in the previous example (-15)).

Running Simulation

- Select **Simulation— Netlist and Run.**

Plotting the Single Point IP3 Using the Calculator

1. After simulation, select **Results — Direct Plot — PSS**. Enter the values in the PSS Results form as follows and click on the *rif* port

PSS Results

OK Cancel Help

File Name Append Replace

Analysis Type
 PSS PAM

Function
 Voltage Current
 IPN Curves Pss Beat Frequency

Input Power Value (dBm)

Input Referred IP3 Order

1st Order sideband

-25	95M
-21	81M
1	-97M

2nd Order sideband

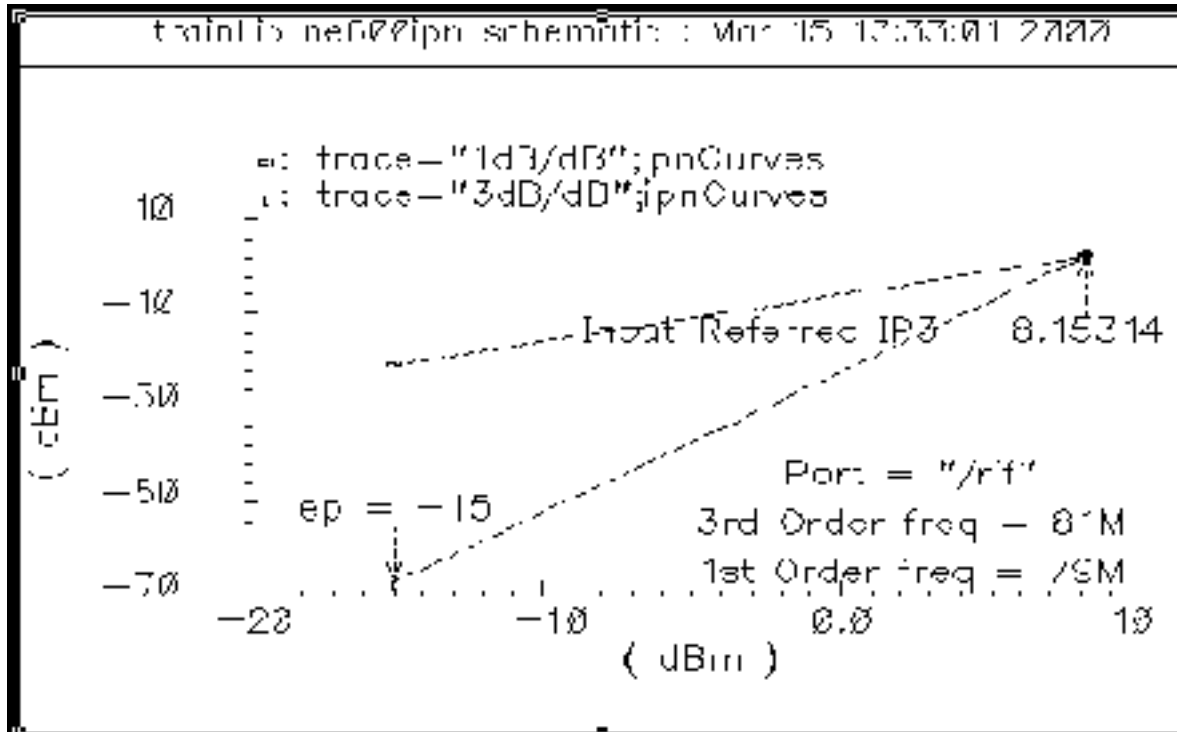
-25	79M
-21	81M
	97M

Add To Outputs Replot

► Select port or net on schematic ...

Affirma RF Simulator (SpectreRF) User Guide
Using the Calculator IPN Functions for RF Simulations

The single point IP3 is plotted.



Because you selected the *Add to Outputs* button in the PSS Results form, the Input Referred IP3 Point direct plot expression is also saved to the *Outputs* pane in the Simulation Window

#	Name/Signal/Expr	Value	Plot	Save	March
1	rif/PLUS		no	yes	no
2	ipnCurves		yes		
3	Input Referred IP3..		no		

Affirma RF Simulator (SpectreRF) User Guide

Using the Calculator IPN Functions for RF Simulations

- If you double click on the `Input Referred IP3...` entry, the *Setting Outputs* form appears

Selected Output		Table Of Outputs			
Name (opt.)	Input Referred IP3 Point	#	Name/Signal/Type	Value	Print/Save/Export
Expression	(1/2*P1) - 0.0) / -21 / -25 ?rport	1	rif:P1	70	yes no
Calculator	<input type="button" value="Open"/> <input type="button" value="Get Expression"/> <input type="button" value="Close"/>	2	ip3Output	70	yes
Will Be	<input type="checkbox"/> Print/Save/Export	3	Input Referred IP3 ...	70	yes
<input type="button" value="Add"/> <input type="button" value="Delete"/> <input type="button" value="Change"/> <input type="button" value="Reset"/> <input type="button" value="New Expression"/>					

You can click on the *Expression* and enter it into the Calculator buffer or CIW window to obtain the value 8.15314.

```
ipnVRI((v("/P1f" ?result "pac") - 0.0) / -21 / -25 ?rport
resultParam("rif:r" ?result "pac") ?epoint -15)
```

You can copy the expressions stored in the *Outputs* section of the Simulation Window into the *Goals* section of the Optimizer by using the **Optimizer — Goals — Retrieve Outputs** menu function.

- The procedure for using the Calculator *ipn* and *ipnVRI* are essentially the same as described in the previous example.



Using PSP and Pnoise Analyses

Overview

This appendix describes how to calculate small-signal quantities such as noise, noise figure, periodic scattering parameters, and gain in periodically-driven circuits. The appendix explains

- The concepts of periodic S-parameters,
- The concepts of noise correlation parameters
- The various definitions of noise figure and gain

The Spectre simulator provides four small-signal analyses for circuits with a DC operating point: AC, XF, Noise and SP. The SpectreRF simulator also provides four small-signal analyses for circuits with a periodically time-varying operating point: PAC, PXF, Pnoise and PSP. Because the periodic small-signal analyses linearize the circuit about the time-varying operating point that is obtained using the PSS analysis, they can analyze frequency conversion effects.

- PAC analysis computes the small-signal response at all outputs to the small stimulus of a single group of sources.
- PXF analysis computes the transfer function from every source in the circuit to a single output.
- Pnoise analysis computes noise parameters such as noise figure as well as detailing noise contributions by devices.
- PSP analysis contains some of the capabilities of PAC, PXF, and Pnoise analyses. PSP analysis can compute periodic scattering parameters that describe the small-signal relations between several different ports in a circuit. It can also compute noise parameters, such as noise correlation matrices, equivalent noise sources, and noise figure.

Periodic S-parameters

Linear Time-Invariant S-Parameters

Designers of microwave and RF circuits typically characterize the frequency-dependent behavior of linear networks through sets of *scattering* or *S*-parameters. The notion of scattering parameters is rooted in transmission line concepts where the scattering parameter matrix relates the magnitude and phase of incident and reflected waves.

Consider an arbitrary *N*-port linear time-invariant (LTI) network, such as the two-port shown in [Figure 12-1](#) on page 944. Each port is driven by a source of reference impedance Z_i , where the index *i* runs from 1 to *N*. For the remainder of this document we will assume a real valued reference impedance, R_i , for each port. In terms of the port currents I_i and voltages V_i , the *incident* and *reflected* quantities, a_i and b_i respectively, are defined for each port as

$$a_i = \frac{v_i}{2\sqrt{R_i}} + \frac{\sqrt{R_i}}{2} I_i$$

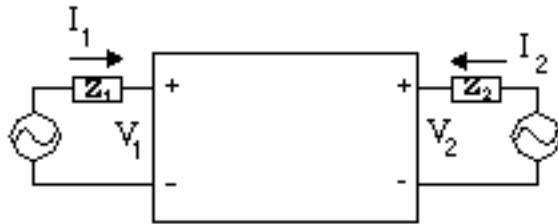
$$b_i = \frac{v_i}{2\sqrt{R_i}} - \frac{\sqrt{R_i}}{2} I_i$$

The frequency-dependent S-parameter matrix relates *a* and *b* by

$$b(\omega) = S(\omega)a(\omega)$$

These definitions are possible because for LTI systems, sources at a frequency ω generate steady-state responses, and therefore outputs, at the same frequency.

Figure 12-1 Two-Port Linear Network



Frequency Translating S-Parameters

The SpectreRF small-signal analyses treat the circuit as linear time-varying (LTV). The primary difference between linear time-varying networks, those that come from circuits with a time-varying operating point, and LTI networks, those that come from circuits with a DC operating point, is that LTV networks shift signals in frequency.

For periodically linear time-varying (PLTV) systems, inputs at a frequency ω may generate circuit responses, and therefore outputs, at the frequencies $\omega + n\omega_0$, where ω_0 is the fundamental frequency and n is a (signed) integer. We can adopt the S-parameter concept to PLTV systems by considering the inputs and outputs generated at the sidebands of each harmonic to be *virtual ports* of a generalized linear system. That is, we can define $a_{i,n}$, $b_{i,n}$ by

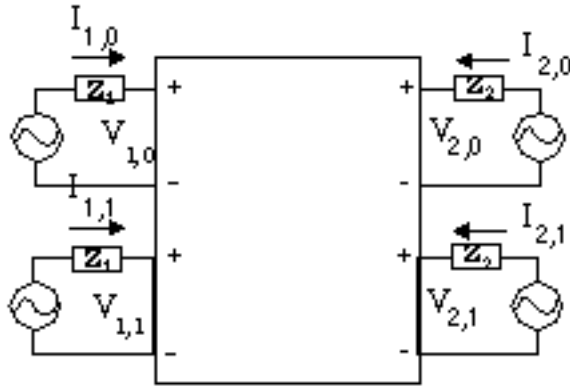
$$a_{i,n}(\omega) = \frac{v_i(\omega + n\omega_0)}{2\sqrt{R_i}} + \frac{\sqrt{R_i}}{2} I_i(\omega + n\omega_0)$$

$$b_{i,n}(\omega) = \frac{v_i(\omega + n\omega_0)}{2\sqrt{R_i}} - \frac{\sqrt{R_i}}{2} I_i(\omega + n\omega_0)$$

where the integer n represents a harmonic index.

For example, as shown in [Figure 12-2](#) on page 945, an ideal mixer may be represented as a four-port.

Figure 12-2 Ideal Mixer Represented as a Four-Port



Note that each *virtual port* of a given *physical port* has the same reference impedance. The periodic S-parameter matrix is the 4×4 matrix \tilde{S} that relates the extended vectors

$$\tilde{b} = [b_{1,0} \ b_{1,1} \ b_{2,0} \ b_{2,1}]^T, \tilde{a} = [a_{1,0} \ a_{1,1} \ a_{2,0} \ a_{2,1}]^T$$

by

$$\tilde{b}(\omega) = \tilde{S}(\omega)\tilde{a}(\omega)$$

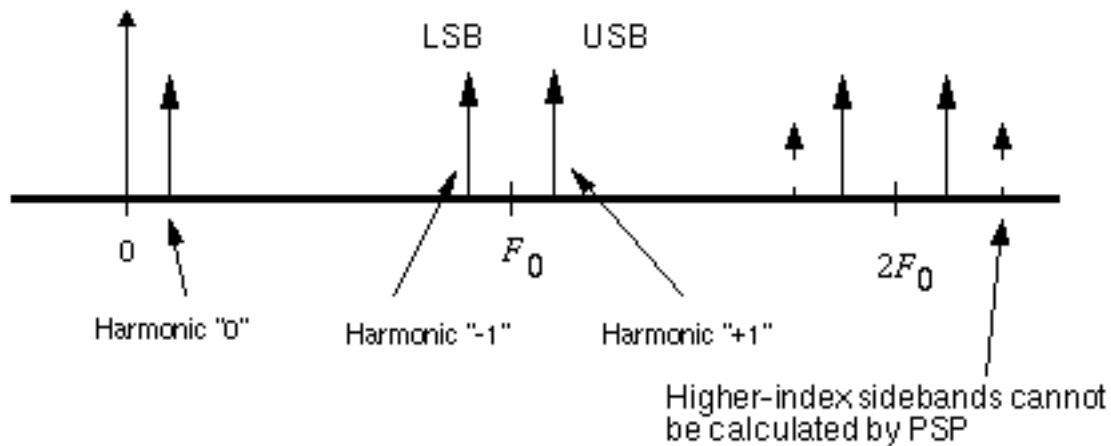
For example, consider an upconverting mixer. You might write $S(2,1|1,0)$ to represent the signal generated at port #2 (typically the output) on the upper sideband of harmonic +1 by an incident signal at port #1 (typically the input) at harmonic zero (baseband). $[S(2,1|1,0)(\omega)]^2$ would represent the power gain from baseband to RF at the baseband-referenced frequency ω . In the PSP results generated by spectre, $S(2,1|1,0)$ is accessible as S21~1:0.

Note that because *multiple virtual ports* are used as both inputs and outputs in PSP analysis, PSP analysis must follow an absolute indexing scheme for the small-signal responses. This is different from the relative indexing scheme used in PXF, PAC, and PSP analyses in releases 4.4.5 and earlier. See [“Harmonics and Sidebands in PSP, PAC, PXF, and Pnoise Analyses”](#) on page 978 for a discussion of the differences. In future releases, all RF analyses will support absolute indexing of the signals.

Upper and Lower Sidebands

Each harmonic may have an input, or response, at both the upper and lower sidebands of each harmonic. In PSP analysis, the upper sideband is denoted by a positive integer, and the lower sideband as a negative integer. See [Figure 12-3](#) on page 946.

Figure 12-3 Harmonics, Sidebands, and Virtual Ports



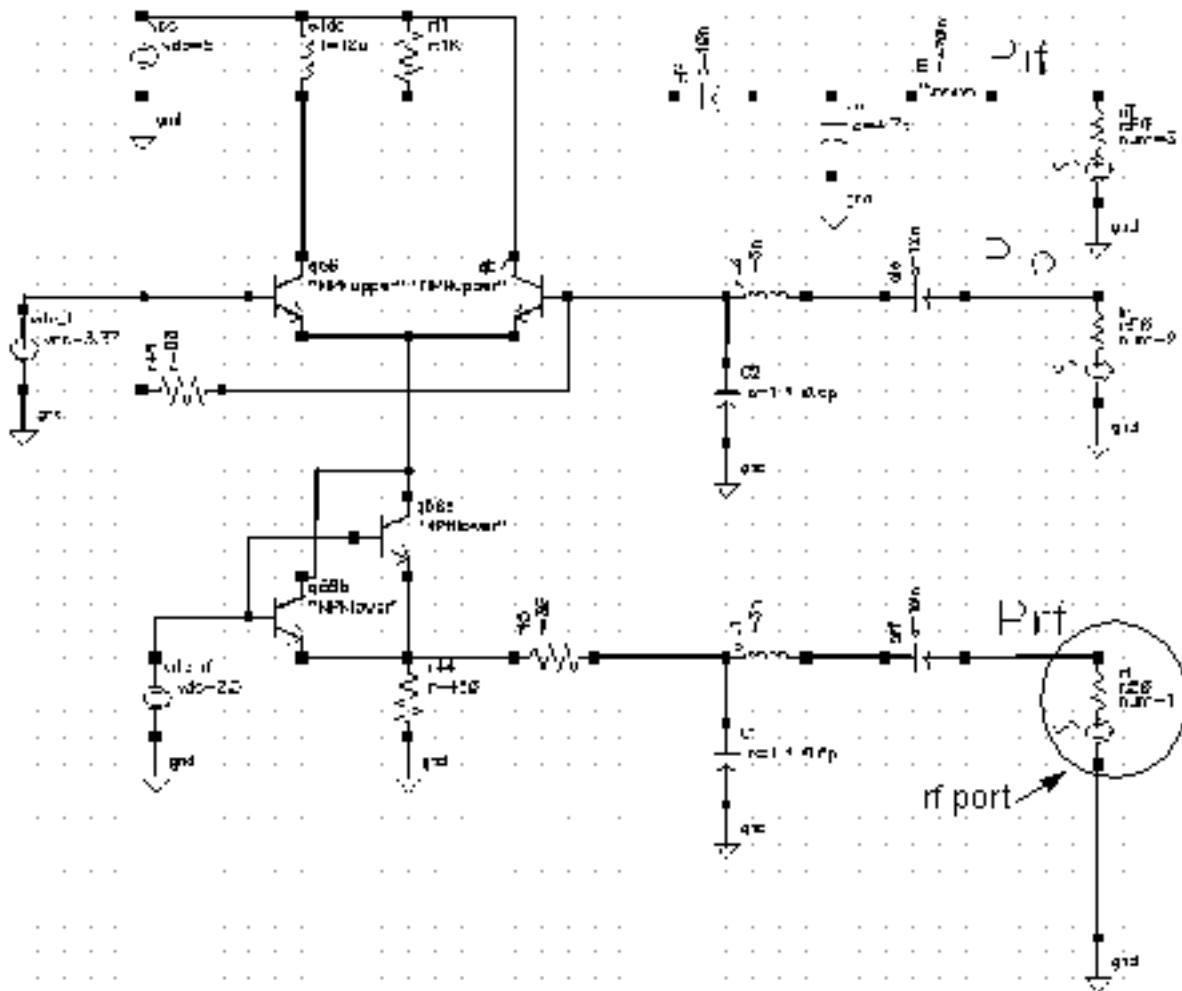
For a small-signal frequency ΔF , the upper sideband of the k^{th} harmonic is at the frequency $|k| F_0 + \Delta F$ and the lower sideband is at $|k| F_0 - \Delta F$.

In general, an input (perhaps at baseband at the frequency ΔF in an upconversion mixer) can generate responses at all the frequencies $|k| F_0 + l\Delta F$, for k and l integers. However, since small-signal analyses are linear, they can only calculate the signals at the first sideband. Thus we only need notation for the $l = 1$ terms in PSP analysis.

PSP Analysis Example

Consider performing a PSP analysis on the *NE600* mixer schematic from the *rfExamples* library. The schematic is shown in [Figure 12-4](#) on page 947.

Figure 12-4 NE600 Mixer



Suppose the RF input signal is at 900 MHz, the LO at 1 GHz, and the IF at 100 MHz. Before the PSP analysis is performed, a PSS analysis must be run. For small-signal analysis, in many cases it would be sufficient to treat the RF input as small-signal (for example, by setting the *source type* to *DC*). However, sometimes it is important to analyze additional noise folding terms induced by the RF input, so in this example we assume the RF source is a large signal (e.g., *source type* = *sine*). The PSS fundamental will need to be set to 100 MHz.

Now for the sake of demonstration suppose we wish to perform the small-signal analysis from 20 MHz below the RF center frequency to 30 MHz above. To set up the analysis, we first select a *frequency sweep*. We select *sweeptype=relative*, with a range of -20 MHz to 30 MHz. This will account for inputs on the RF port in the range of 880 MHz to 930 MHz. Noise

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

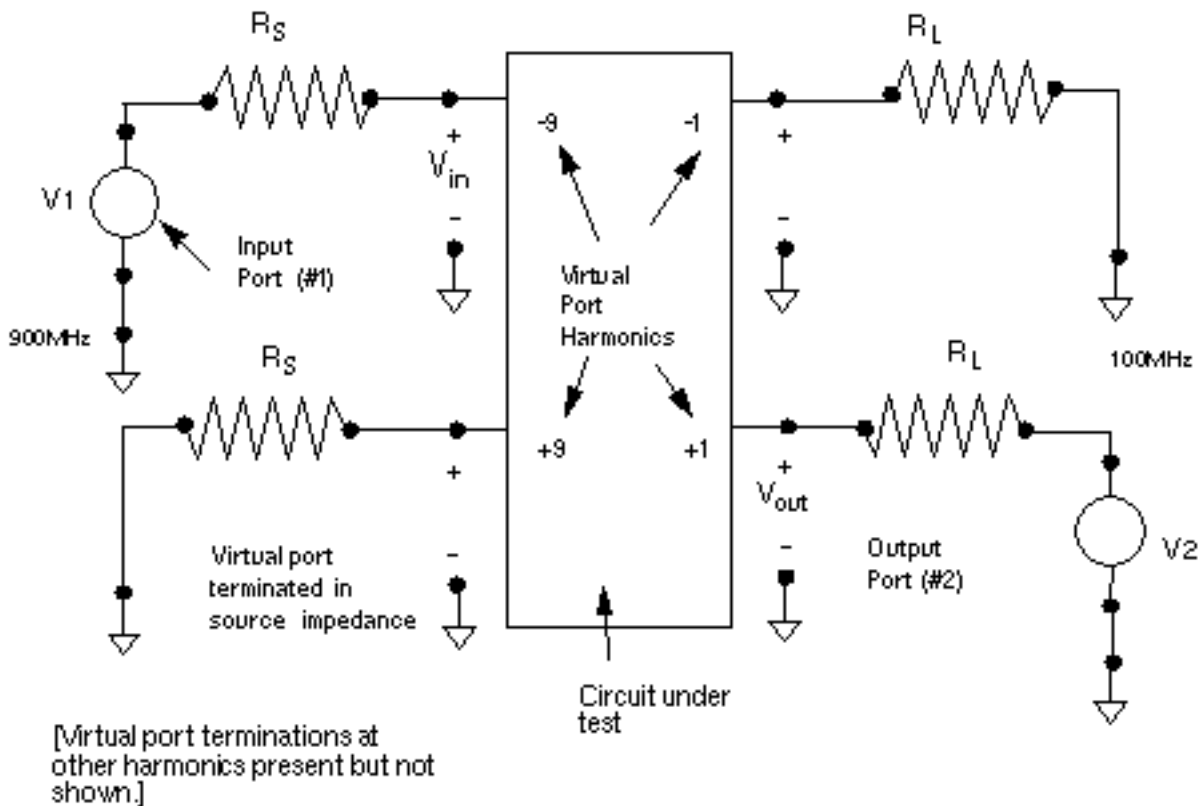
parameters such as *noise figure* will be computed in a 50 MHz band around the frequency specified by the output harmonic.

Next we select the ports and harmonics. The *input* and *output ports* are selected from the schematic as always. Selecting the harmonics is somewhat trickier.

With the 1 GHz LO, small-signal inputs at around 900 MHz, or harmonics ± 9 of the PSS fundamental, will appear at around 100 MHz, or harmonics ∓ 1 of the fundamental. We may select harmonic 1 as the output harmonic and harmonic -9 as the input harmonic because a single complex-exponential input, $e^{i\omega_s t}$, input on the lower side of 900 MHz (harmonic -9) will appear as the upper sideband of harmonic 1 (around 100 MHz). -9 and 1 are separated by 10 fundamental periods, which corresponds to the LO frequency of 1 GHz. [Figure 12-5](#) on page 949 conceptually illustrates this setup.

Once the simulation is complete, a limited set of data is accessible through direct plot, and the full data set is accessible with the results browser. The default is to output all quantities versus the input frequency, which in this case would be a sweep from -920 MHz to -870 MHz, because the input harmonic is -9 and the sweep ran from -20 Mhz to 30 MHz. The *freqaxis* parameter on the options form may be used to change the axes that are output by Spectre. In 4.4.5 the *freqaxis* parameter and *frequency sweep* specifications are solely responsible for the data's axis generation. Setting *freqaxis=out* in this example would produce an axis running from 80 Mhz to 130 Mhz.

Figure 12-5 Measuring Periodic S-Parameters



An additional port/harmonic pair can be included in the PSP analysis by using the *auxiliary port* fields.

If it is desirable to include more than three harmonics in the PSP analysis, they can be added to the list in the form below the *input./output/auxiliary*. For example, to examine additional images in the PSP analysis, +9 and -1 could be additional harmonics. S21~9:-1 would represent the transducer gain from RF-USB to IF-LSB. Note that S21:9:1 is likely to be small, because this term represents a frequency shift of 800 MHz. There are no elements in the circuit that vary at 800 MHz, so significant 800 MHz frequency translations will not be present.

Noise and Noise Parameters

Calculating Noise in Linear Time-Invariant (DC Bias) Circuits

The standard Noise analysis has two parts.

- First, the circuit is analyzed without the noise sources present in order to find a DC operating point.
- Next, the circuit is linearized around that operating point and the noise sources are turned on.

The linearized circuit is used to compute a set of transfer functions that represent the gain from each noise source to the node pair or probe that is identified as the output for the Noise analysis. All noise generators present in the circuit are automatically included in the Noise analysis, as are the noise sources of the source and load.

Calculating Noise in Time-Varying (Periodic Bias) Circuits

Noise analysis in RF circuits, where the circuit operating point is time-varying, is computationally involved, but conceptually similar to Noise analysis for circuits with a DC bias.

- Find the circuit operating point using the PSS analysis.
- Linearize the circuit around that operating point.
- Use the PXF analysis, based on the linearized circuit, to compute a set of transfer functions that relate the noise sources to the noise at the circuit output.

The treatment of the sources and the transfer functions is more complicated for RF circuits because of the time-varying operating point.

For noise sources that are bias dependent, such as shot noise sources, the time-varying operating point acts to modulate the noise sources. Active elements with a time-varying bias point can convert noise from one frequency to another, a process known as *noise folding*, regardless of the origin of the noise.

Because of these effects, noise generated in RF circuits usually has *cyclostationary* properties. Cyclostationary random processes are processes whose statistical properties are periodically time-varying. In the frequency domain, a simple way to think of cyclostationarity is as frequency-correlation. For example, noise at the input of a mixer will appear on the output, but shifted in frequency. Thus the noise at the mixer output at a given frequency will be correlated with noise at the mixer input at a frequency separated by the frequency of the

local oscillator. In contrast, the noise generated by a circuit with a DC bias point is usually modeled as being uncorrelated with noise at any other frequency. SpectreRF correctly accounts for cyclostationary statistics when calculating noise, noise figure, noise correlation parameters, and equivalent noise sources.

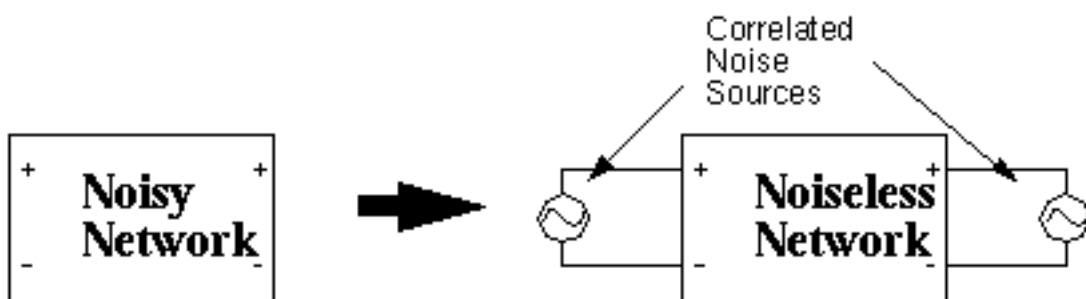
The maxsideband Parameter

All the noise computations in PSP involve noise folding effects. The *maxsideband* parameter specifies the maximum sideband included for summing noise contributions either up-converted or down-converted to the output at the interested frequency. The contribution of the noise source to the output is modulated by the periodic transfer function. Modulation with a periodic transfer function is convolution with the discrete spectrum of the transfer function. *Maxsideband* specifies the number of sidebands to be involved in this calculation.

Noise Correlation Matrices and Equivalent Noise Sources

Noise correlation matrices represent a decomposition of a linear circuit into a noiseless linear network and correlated noise sources. For example in [Figure 12-6](#) on page 951, a noisy linear network is decomposed into a noiseless network and equivalent noise current sources, one for each circuit port, that represent the effect of all the noise generators internal to the original network. Note that in general the equivalent noise sources will be correlated. The equivalent sources can be completely described by a (frequency-dependent) noise correlation matrix. Note that once a noise correlation matrix along with an admittance, impedance, or S-parameter matrix are known, then the properties of the noisy linear network as seen from the I/O ports is completely specified. All circuit input/output properties—gain, noise figure, etc.—can be calculated from the S-parameter and noise correlation matrices.

Figure 12-6 Decomposing a Network Into Noiseless and Noisy Elements.



Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

The SP analysis computes noise correlation parameters in the admittance representation, i.e., the sources in [Figure 12-6](#) on page 951 are current sources. If we let I_N denote the vector of equivalent noise currents.

$$I_N = \begin{bmatrix} I_{N1} \\ I_{N2} \end{bmatrix}$$

then the noise correlation matrix is defined as

$$C_Y = E \left\{ I_N I_N^H \right\}$$

where superscript H denotes Hermitian transpose and $E\{\cdot\}$ denotes statistical expectation. For example, for a two-port,

$$C_Y = E \left\{ \begin{bmatrix} I_{N1} \overline{I_{N1}} & I_{N1} \overline{I_{N2}} \\ I_{N2} \overline{I_{N1}} & I_{N2} \overline{I_{N2}} \end{bmatrix} \right\}$$

where overbar signifies complex-conjugate. Note that this matrix must have real diagonal elements, because the diagonals represent a total noise power, but the off-diagonals, which represent the correlations, may be complex.

In the periodic case, we define the periodic admittance noise correlation matrix in precisely the same way. The situation is slightly more complicated because the *virtual ports* may lie at different frequencies in RF systems.

Letting \tilde{I}_N denote the extended vectors of noise currents at each of the virtual ports (each virtual port consisting of a physical port. harmonic pair), for example

$$\tilde{I}_N = \left[I_{1,0} \ I_{1,1} \ I_{2,0} \ I_{2,1} \right]^T$$

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

where the first index indexes the physical port, and the second index specifies the harmonics. The periodic admittance noise correlation matrix \tilde{C}_Y is defined as

$$\tilde{C}_Y = E \left\{ \tilde{I}_N \tilde{I}_N^H \right\}$$

If *donoise=yes* is specified in the PSP analysis, then the complex noise correlation matrix of order (#active ports X #active sidebands) is computed.

Two-Port Noise Parameters

As an alternative to the noise correlation matrices that define the equivalent sources, Spectre calculates the values of the equivalent noise parameters F_{min} , R_n , G_{opt} and B_{opt} . These are calculated from the two-port admittance (Y_{11} , Y_{12} , Y_{21} , Y_{22}) and noise correlation admittance (CY_{11} , CY_{12} , CY_{21} , CY_{22}) parameters. These calculations are done as part of the SP and PSP analysis. In terms of the admittance and noise correlation matrices, the parameters are (see *Janusz A. Dobrowolski, Introduction to Computer Methods for Microwave Circuit Analysis and Design, Artech House, Boston, 1991, page 193*)

$$F_{min} = 1 + 2 \frac{CY_{22}}{|Y_{21}|^2} (G_{opt} + \text{Re}\{Y_{11} - Y_{21}(CY_{12}/CY_{22})\})$$

$$G_{opt} = \sqrt{(|Y_{21}|^2 (CY_{11}/CY_{22}) - |Y_{21}|^2 (|CY_{12}|^2/CY_{22}^2) + (\text{Re}\{Y_{11} - Y_{21}(CY_{12}/CY_{22})\})^2)}$$

$$B_{opt} = -\text{Im}\{Y_{11} - Y_{21}(CY_{12}/CY_{22})\}$$

$$R_n = CY_{22}/|Y_{21}|^2$$

$$Y_{opt} = G_{opt} + jB_{opt}$$

Y_{opt} is the source admittance that gives the minimal noise factor F_{min} (corresponding to the source reflection coefficient Γ_{opt} , or Γ_{opt}) and R_n is the equivalent noise resistance.

Noise Circles

Noise factor will be a function of the source admittance $Y_s = G_s + jB_s$. For a given Y_s the noise factor is

$$F = F_{min} + \frac{R_n}{G_s} (Y_s - Y_{opt})^2$$

Varying Y_s traces out circles of constant noise factor F . In the 4.4.5 release, noise circles are only available in direct plot for the SP analysis.

Noise Figure

Performing Noise Figure Computations

The Noise, SP, Pnoise and PSP analyses all provide the ability to calculate various types of noise figure. Noise and SP analyses are used for circuits with a DC bias. Pnoise and PSP analyses are used for circuits with a periodic bias. The *generic* way Spectre calculates noise figure is by first computing the noise factor F ,

$$F = \frac{\text{totalOutputNoise} - \text{outputNoiseFromLoad}}{\text{outputNoiseFromSource}}$$

where the noise is specified in units of power (e.g., V^2/Hz). Noise figure is then $NF=10\log_{10}F$.

The various definitions of noise figure differ in the following ways

- How the contributions to the total output noise are calculated (e.g. Pnoise analysis has noise folding effects, Noise analysis does not)
- What noise is considered to be due to the load
- What noise is considered to be due to the source

All the analyses share some common rules that must be followed to obtain correct answers

- You must specify a *port* (not a *vsource*, *isource*, or *ahdl* source) as an circuit *input* or *iprobe*.
- You must specify the *load* as an *output* or *oprobe*. The *load* may be a *resistor* or a *port*. Note that all noise from the source will be included in the denominator of the noise factor fraction, including excess noise, so do not specify excess noise on the input port. (Excess noise is specified with the *noisefile* or *noisevec* option.)

In rare cases there may be no load, in which case you can specify the output using a pair of nodes. Be warned, however, that if there is a load in the circuit, and a pair of nodes is specified as an output, then you will obtain different results than if a load was specified as output. This is because the load will contribute some noise to the total output noise that must be subtracted out before using the equation above to compute the noise figure. If only a pair of nodes is used, Spectre has no way of determining which of the elements in the circuit is the load (there could be multiple resistors connected to the output nodes, for example) and so cannot determine the amount of output noise due to the load.

Note that these requirements are automatically enforced in SP and PSP analyses, since the input and output sources are always ports that must be identified to the analyses.

Noise Figure From Noise and SP Analyses

The Noise and SP analyses perform noise figure computations on circuits with a DC operating point. The above prescription for noise figure computation is straightforward: the output noise, contribution from source, and contribution from load must be computed. Mathematically, if we let X_L denote the transfer function from output load to output, (at the same frequency) and X_S denote the transfer function from input source to output then

$$F(f) = \frac{N_o(f) - |X_L|^2 n_L(f)}{|X_S|^2 n_s(f)}$$

In this equation, X_S plays a role similar to transducer gain in traditional treatments of noise figure.

Pnoise (SSB) Noise Figure

Because noise in an RF circuit can originate at many different frequencies, the denominator in the noise factor computation is in a sense ill-defined. See the book *Microwave Mixers* by S. Maas for a discussion of various possible noise figure definitions.

There are three common noise factor definitions in use. The SpectreRF Pnoise and PSP analyses compute as F or NF what is referred to as conventional single-sideband (SSB) noise figure. The conventional SSB noise figure is typically useful for heterodyne receivers. To compute the conventional SSB noise figure, a reference sideband must be specified that identifies the input noise used in the denominator of the noise factor computation. Only the contribution of the noise from the input source, generated at the frequency specified by the reference sideband, is included in the noise factor denominator.

In the Pnoise context, the numerator contains the total output noise, except the noise from the output load that was generated at the output frequency. Note in particular that noise from the input source folded from all sidebands, and noise from the output load folded from the

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

non-zero sidebands, is included in the noise factor numerator. Mathematically, Pnoise computes conventional single-sideband noise factor as

$$F_{ssb}(f_{out}) = \frac{N_o(f_{out}) - |X_L^{(0)}|^2 n_L(f_{out})}{|X_S^{(K_{ref})}|^2 n_S(f_{out} + K_{ref}f_0)}$$

where

- f_{out} is the output frequency swept by Pnoise
- $X_L^{(0)}$ is the transfer function associated with the zero sideband, from load to output
- $X_S^{(K_{ref})}$ is the transfer function associated with the reference sideband, from source to the output
- $n_L(f_{out})$ is noise generated by load at the output frequency
- $n_S(f_{out} + K_{ref}f_0)$ is the noise generated by the source at the input frequency.

In the PSP analysis context, the noise factor denominator includes only noise from the input harmonic. The numerator contains all output noise, except noise from the output load at the output harmonic. (Refer to [“Harmonics and Sidebands in PSP, PAC, PXF, and Pnoise Analyses”](#) on page 978 for a discussion of differences in frequency indexing in PXF and PSP analyses.)

To be mathematically precise in what follows, let

- $x_L^{(K)}$ denote the transfer function from output load, k^{th} harmonic to output
- $x_S^{(K)}$ denote the transfer function from input source, k^{th} harmonic to output
- K_o denote the output harmonic
- K_i denote the input harmonic

The conventional single-sideband noise factor is computed by PSP as

$$F_{ssb}(f_{out}) = \frac{N_o(K_o f_0 + f) - |X_L^{(K_o)}|^2 n_L(K_o f_0 + f)}{|X_S^{(K_i)}|^2 n_S(K_i f_0 + f)}$$

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

where

- f is the PSP relative sweep frequency
- $K_o f_0 + f$ is the output frequency
- $K_i f_0 + f$ is the input frequency
- $n_L(K_o f_0 + f)$ is the noise generated by the load at the output frequency
- $n_S(K_i f_0 + f)$ is the noise generated by the source at the input frequency

The conventional SSB noise figures computed by PSP and Pnoise analyses are the same and are computed in the same way internally, it is only the notation above that is different.

DSB Noise Figure

In some applications, such as direct conversion receivers, it is more appropriate to compute what is called double-sideband (DSB) noise figure. Double-sideband noise factor is obtained by ratioing the same numerator as for SSB to the noise from the input at the input harmonic as well as its primary image.

Double-sideband noise figure is usually 3 dB below single-sideband noise figure, except when the input signal band is converted to baseband output with DSB noise figure equal to SSB noise figure.

In double-sideband computation, the input signal band is assumed to be either down-converted or up-converted to the output signal band. Hence you should associate the appropriate harmonic number to the input and the output port in the *portarmsvec* parameter. For a mixer, their difference is the LO band. The image sideband is the sideband on the other side of the LO band. The distance from the image band to the LO band is the same as that from the LO band to the input band. For example, if the fundamental frequency is 100 MHz, the LO frequency is 1 GHz, and the RF input frequency is 900 MHz, then the LO band is 10, the RF input band is 9 and the image band is 11.

Double-sideband noise figure is computed by as

$$F_{dsb}(f_{out}) = \frac{N_o(K_o f_0 + f) - \left| X_L^{(K_o)} \right|^2 n_L(K_o f_0 + f)}{\left| X_S^{(K_i)} \right|^2 n_S(K_i f_0 + f) + \left| X_S^{(K_{image})} \right|^2 n_S(K_{image} f_0 + f)}$$

where

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

- $n_S(K_{image}f_0 + f)$ the noise generated by source at the image input frequency obtained according to the above description.

All other quantities are the same as those in F_{ssb} . Note that only the PSP analysis computes double-sideband noise figure.

IEEE Noise Figure

Sometimes it is desirable to define noise figure quantities where we assume that the noise from input images that are potentially filtered is not present. The IEEE definition of noise figure in mixers differs from the conventional definition in that it does not include the contribution to the output noise from the image sideband in the numerator of the noise factor. Spectre eliminates all image harmonics/sidebands from the output noise in the noise factor numerator when computing F_{ieee} . Using the above notation, F_{ieee} is

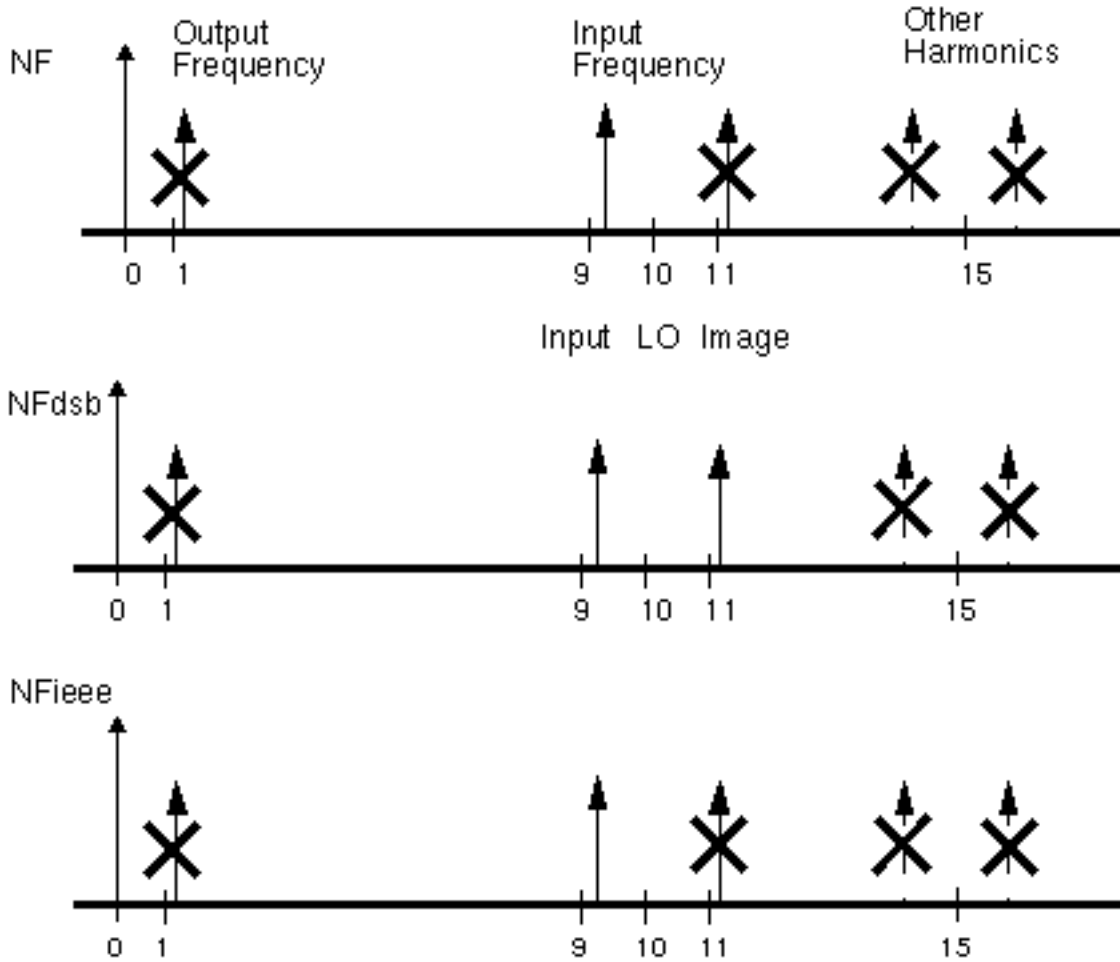
$$F_{ieee} = \frac{N_o(K_o f_0 + f) - \left| X_L^{(K_o)} \right|^2 n_L(K_o f_0 + f) - \sum_{K \neq K_i} \left| X_S^{(K)} \right|^2 n_S(K f_0 + f)}{\left| X_S^{(K_i)} \right|^2 n_S(K_i f_0 + f)}$$

Only PSP analysis computes F_{ieee} .

[Figure 12-7](#) on page 959, [Figure 12-8](#) on page 960, and [Figure 12-9](#) on page 961 summarize the treatment of the input source and output load for the various noise figure definitions.

Affirma RF Simulator (SpectreRF) User Guide
Using PSP and Noise Analyses

Figure 12-7 Input Source Treatment for Denominator in Noise Factor Computations



Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Noise Analyses

Figure 12-8 Output Load Treatment for Numerator in Noise Factor and Noise Figure Computations

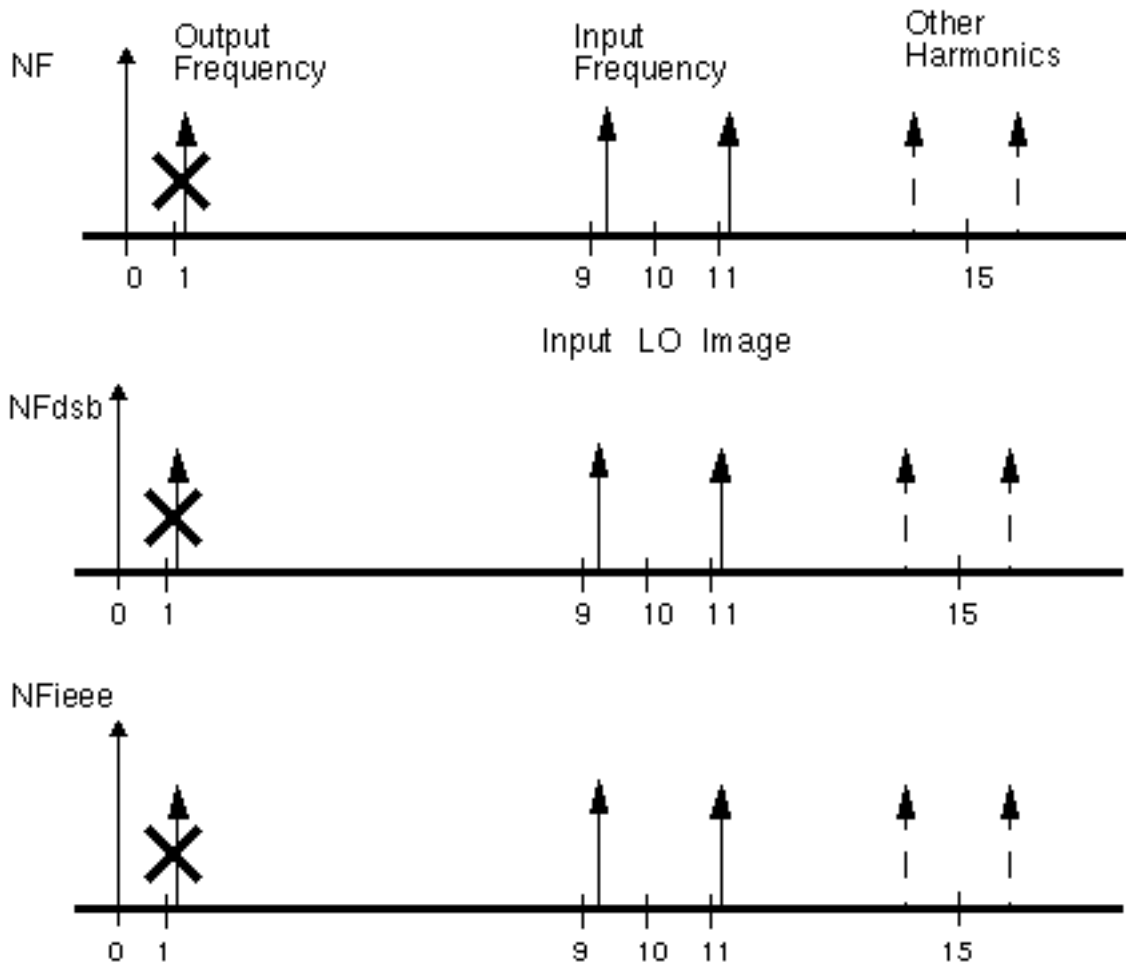
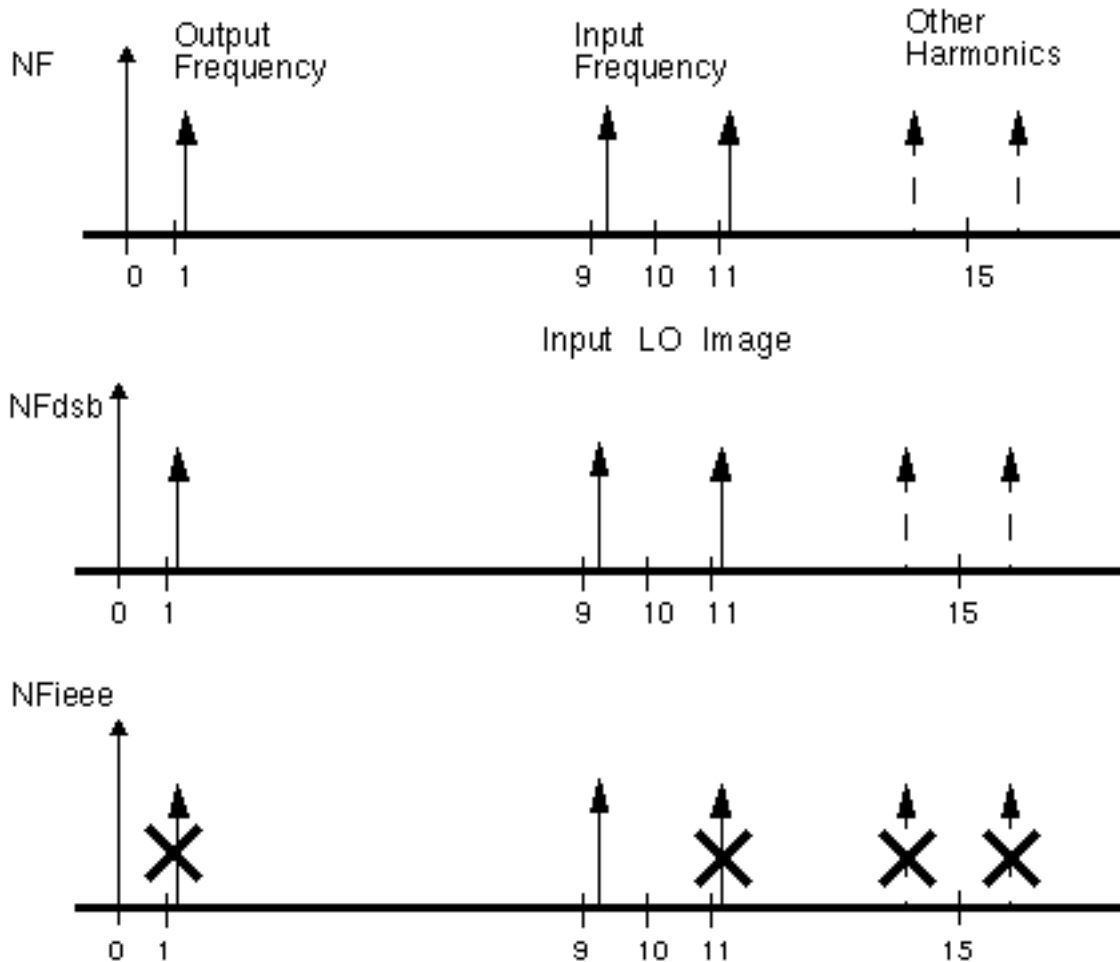


Figure 12-9 Input Source Treatment for Numerator in Noise Factor Computations



Noise Computation Example

Now consider performing noise computations as part of the PSP analysis of the *ne600* mixer example presented earlier (See [“PSP Analysis Example”](#) on page 946).

To compute noise parameters, set *donoise=yes* on the PSP analysis form and select a reasonable number for *maxsideband*. In this case, *maxsideband* should certainly be greater than 10 and probably greater than 50 or so. Setting *maxsideband=50* would account for noise folding from up to 5 GHz in frequency. The simulation can now be run as before. In

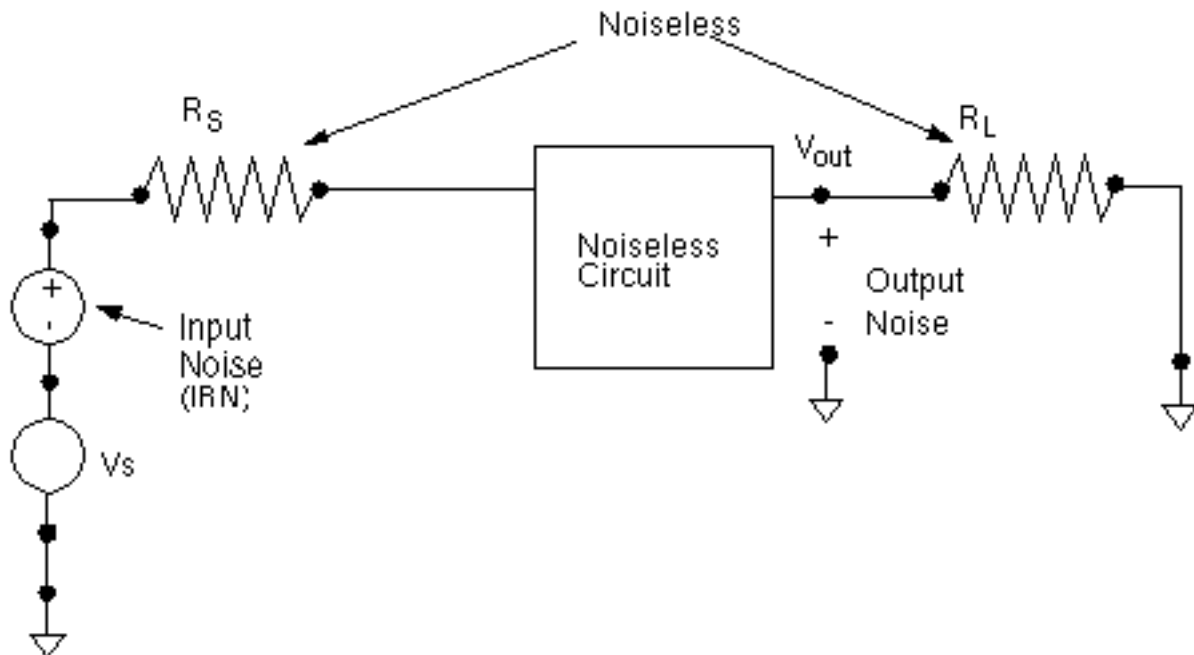
In addition to computing the periodic S-parameters, the periodic noise correlation matrices will also be computed, as will noise figure and the equivalent noise parameters.

Recall that the output axes for PSP analysis are set by the *freqaxis* option and are shared by all quantities calculated by PSP analysis. This can seem odd in the case of noise figure, where the output noise is actually analyzed for the frequency range of 80 MHz to 130 MHz. This is because the output harmonic is 1, representing a center frequency of 100 MHz for the relative sweep, and the sweep ran from -20 MHz to 30 MHz. Setting *freqaxis=out* in this example would produce an axis running from 80 MHz to 130 MHz.

Input Referred Noise

For a given output noise spectrum, the equivalent input noise, *input referred noise* or IRN, is the noise that, if it were generated by the circuit element specified as input, would produce the same output noise. This assumes that the circuit loading conditions, etc., are unchanged. Note that the equivalent input noise will include noise from the source and load if they are noisy in the original circuit. For example, consider a circuit with a voltage source as input, as in [Figure 12-10](#) on page 963. If a voltage source inserted in series with the input source generates the input referred noise, and the rest of the circuit is noiseless, the noise observed at the output will be the same as in the original noisy circuit.

Figure 12-10 Equivalent Input Noise



Using Input Referred Noise

Input referred noise, or IRN, gives a direct estimate of how much the noise in a circuit corrupts signals passing through, since the amplitude of the noise can be directly compared to the amplitude of signals on the input. In principle, it can also be used to build macromodels of the circuit. If a source of the same type (`vsources`, `isources`, or `port`) has the `noisefile` argument set to a file to which the input referred noise has been written, and all the noise elements in the circuit turned off (perhaps as may happen when replacing the circuit with an S-parameter macromodel) then the noise that appears as the circuit output will be the output noise of the original circuit. Note that Spectre `noisefile` arguments are usually given in terms of power, such as V^2/Hz , whereas the simulator usually outputs the noise in units of signal amplitude (such as $V/\sqrt{\text{Hz}}$).

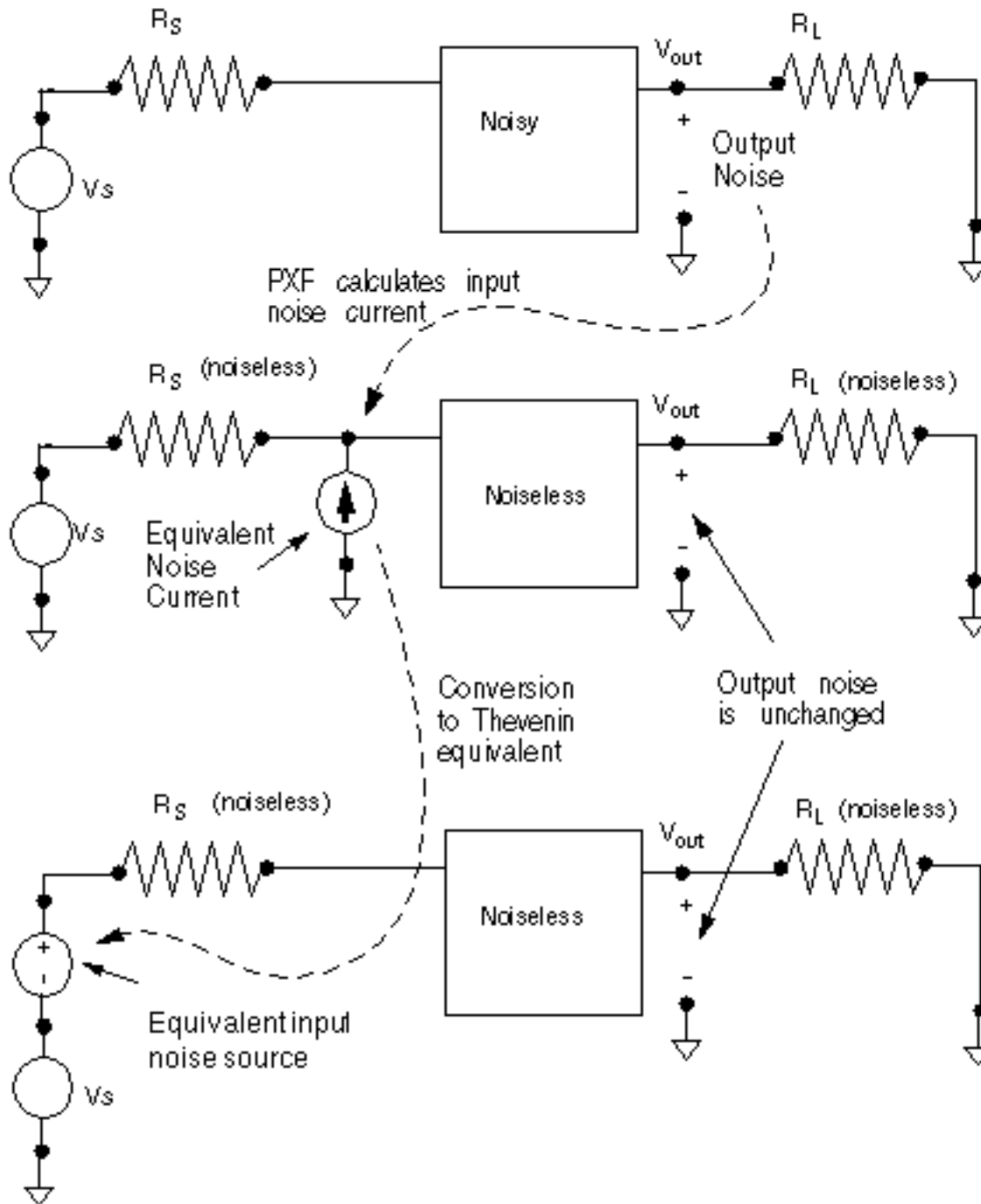
It is also possible to calculate noise figure from input referred noise, however, it is essential to be sure that any noise contribution from the load, etc., has first been properly treated. For this reason, it is recommended to use the Spectre built-in noise figure calculation, since the contribution of load noise, noise from other images, etc., is automatically accounted for. (See "The noise figure computed by <Procedure X> seems inaccurate, inconsistent, or just plain wrong." on page 974 for more information.)

How IRN is Calculated

Spectre calculates input referred noise from the results of the Noise or Pnoise analysis and an XF or PXF analysis. The XF or PXF analysis needed is fortunately the same analysis needed to compute the noise.

As an example of an input noise calculation, consider [Figure 12-11](#) on page 965, which shows the calculation of the equivalent input noise when a port component drives the circuit. First the total output noise is calculated. Note that this noise generally includes contributions from the circuit load. Next the results of the PXF analysis are used to express this noise as an equivalent current source attached to the circuit input. The amplitude of the current source is the amplitude of the output noise divided by the transfer function from an imaginary current source connected to the circuit input to the output node. Finally, the equivalent current source is converted into the equivalent input voltage noise by converting to Thevenin form.

Figure 12-11 IRN Calculation



Relation to Gain

It is common to think of the input noise as the output noise divided by the gain,

$$IRN = \frac{OutputNoise}{G}$$

where the output noise and input noise are measured in volts/ $\sqrt{\text{Hz}}$. This is a true statement if the proper gain is used. The gain reported by Pnoise analysis is in fact the ratio between the output noise and the equivalent input noise as computed in the previous section. However, the gain reported by Pnoise analysis is not necessarily the gain useful for any other purposes (see “[Gain Calculations in Pnoise](#)” on page 971).

Referring Noise to Ports

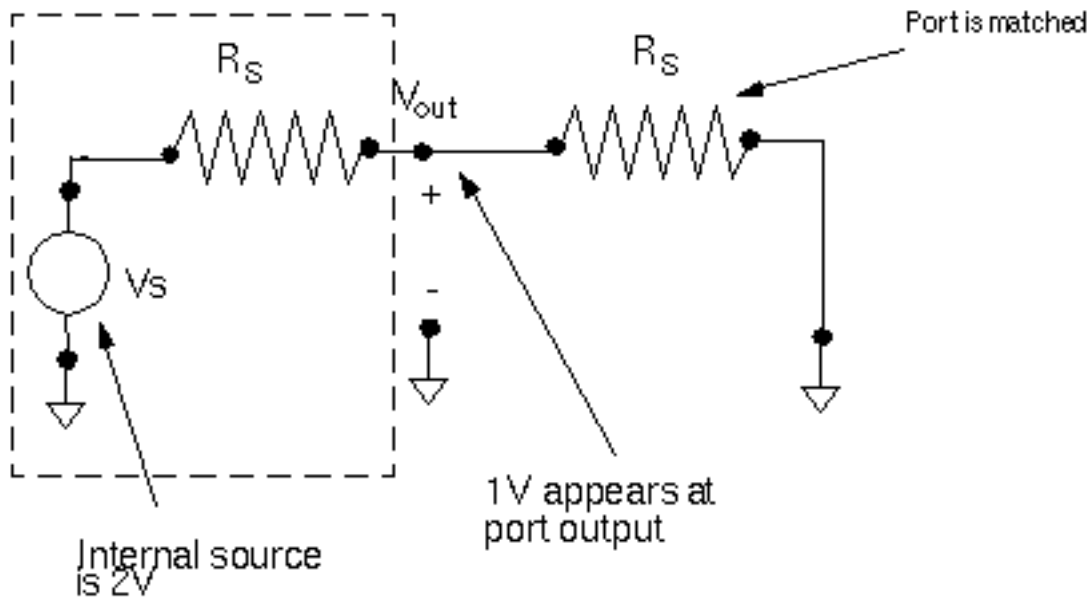
A port component in Spectre is a voltage source combined with a resistive impedance. Ports exist to enable easy generation of test fixtures, accurate noise figure calculations, and S-parameter calculations. Noise and S-parameter calculations in Spectre can (usually) only be properly performed when ports are used to drive the input and output. While the port is electrically equivalent to a source + resistor, the excitations are specified in a somewhat different way.

The port component is designed so that when a 1 V source is specified, 1 V appears at the port output when the circuit impedance is matched to the port reference impedance. The port output, however, is not the internal node that connects the voltage source and resistor, but rather the exterior resistor terminal that actually connects to the circuit under test. Thus, as shown in [Figure 12-12](#) on page 967, the voltage on the internal voltage source must be twice the voltage specified to appear at the port output.

One implication of this convention is that when a port is specified as an input in Pnoise analysis it is the noise referred to the port that is computed, not the noise referred to the internal voltage source. In other words, the input equivalent noise reported is the noise that would be fed back to spectre as a *noisefile* argument on the port to produce the same output noise. The noise referred to the port will have amplitude half of the noise referred to the input voltage source.

Likewise, the gain reported by Spectre is the gain that would produce the equivalent input noise, referred to the port. It will be half the gain from the input voltage source.

Figure 12-12 Port Driving Matched Circuitry



Gain Calculations

Definitions of Gain

To understand gain calculations in SpectreRF it is necessary to specify which of several possible gains are being calculated.

Figure 12-13 Equivalent Input Noise

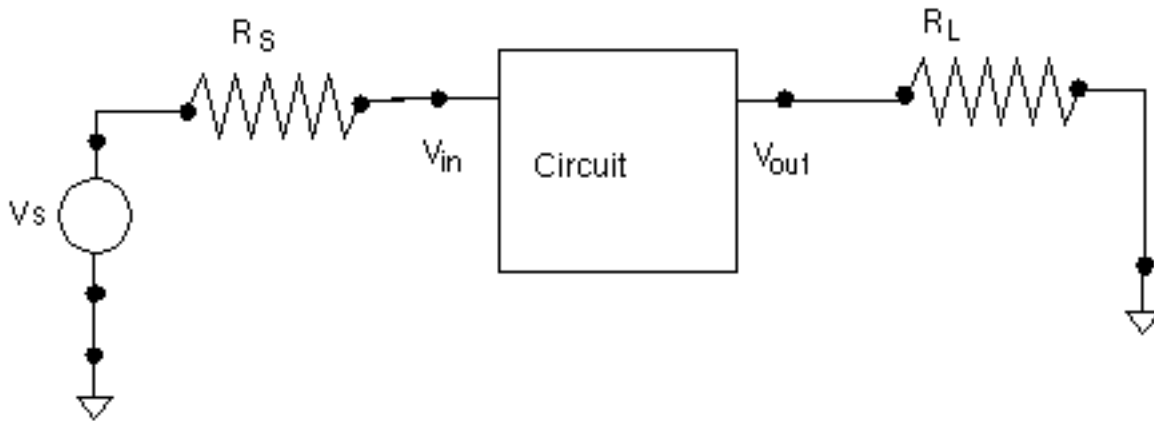


Figure 12-13 on page 968 shows a circuit configured for a gain calculation. Source and load impedances are present. These loads might represent other circuit components or *testbench* circuitry. The circuit is driven with the source V_s and the output voltage V_{out} measured. One possible gain is the gain G_s .

$$G_s = \frac{V_{out}}{V_s}$$

from the voltage source to the circuit output.

However, G_s is not the actual gain of the circuit, it is the gain of the circuit and the testbench put together. For high frequency circuits, this gain in fact cannot be measured on the bench, only in the artificial world of a circuit simulator. For these reasons, the proper way to calculate gain is the use a port component, which allows the user to specify the impedance of the driving circuitry. The Spectre PSP analysis computes the gain,

$$G = \frac{V_{out}}{V_{in}}$$

which is the voltage gain from the circuit input to the circuit output.

G is calculated for historical reasons, and is not necessarily a useful number, but once an SP or PSP analysis is performed, several other gains can be defined and computed. Varying source and load impedance can also be considered. In the 4.4.5 release, the SP direct plot form displays various gains. The PSP direct plot form does not automatically compute the

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

various gains but they can be computed from the basic definitions using the waveform calculator.

Some gains of interest in two-port circuits are:

- G_A (Available Gain)
- G_P (Power Gain)
- G_T (Transducer Gain)
- G_{umx} (Maximum Unilateral Transducer Power Gain)
- G_{max} (Maximum Available Gain)

GA (Available Gain), the power gain obtained by optimally (conjugately) matching the output of the network.

$$G_A = \frac{|S_{21}|^2(1-|\Gamma_S|^2)}{|1-S_{11}\Gamma_S|^2(1-|\Gamma_2|^2)}$$

where

$$\Gamma_2 = S_{22} + \frac{S_{12}S_{21}\Gamma_S}{1-S_{11}\Gamma_S}$$

and Γ_S is the source reflection coefficient,

$$\Gamma_S = \frac{Z_S - Z_{S,ref}}{Z_S + Z_{S,ref}}$$

with Z_S is the source impedance and $Z_{S,ref}$ is the reference impedance for the input port.

GP (Power Gain), the power gain obtained by optimally (conjugately) matching the input of the network.

$$G_P = \frac{|S_{21}|^2(1-|\Gamma_L|^2)}{|1-S_{22}\Gamma_L|^2(1-|\Gamma_1|^2)}$$

where

$$\Gamma_1 = S_{11} + \frac{S_{12}S_{21}\Gamma_L}{1-S_{22}\Gamma_L}$$

and Γ_L is the load reflection coefficient.

GT (Transducer Gain), the ratio of the power dissipated in the load to the power available from the source,

$$G_T = \frac{(1-|\Gamma_S|^2)|S_{21}|^2(1-|\Gamma_L|^2)}{|(1-S_{11}\Gamma_S)(1-S_{22}\Gamma_L)-S_{12}S_{21}\Gamma_S\Gamma_L|^2}$$

Gumx (Maximum Unilateral Transducer Power Gain)

$$G_{umx} = \frac{|S_{21}|^2}{(1-|S_{11}|^2)(1-|S_{22}|^2)}$$

Gmax (Maximum Available Gain), the transducer power gain when there exists a simultaneous conjugate match at both ports.

For $Kf > 1$

$$G_{max} = \frac{|S_{21}|}{|S_{12}|} [Kf - SQRT((Kf)^2 - 1)]$$

For $K_f < 1$

$$G_{max} = \frac{|S_{21}|}{|S_{12}|}$$

Where K_f is the stability factor, K_f

$$K_f = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |D|^2}{2|S_{22}||S_{12}|}$$

and

$$D = S_{11}S_{22} - S_{21}S_{12}$$

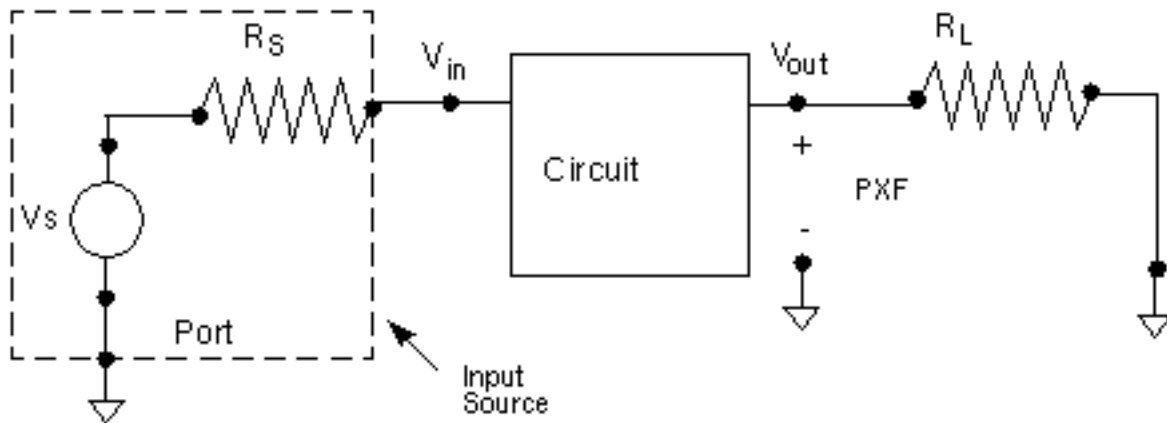
Gain Calculations in Pnoise

When ports are used to drive the circuit input and a Pnoise analysis is performed, Pnoise analysis reports a quantity called `gain` that can be misleading. Consider the circuit in [Figure 12-14](#) on page 972 where a port component is used as the input to the circuit, and specified as such by using the `iprobe` option to Pnoise.

The goal of Pnoise analysis is usually to calculate the noise at the circuit output, as shown in [Figure 12-14](#) on page 972. Pnoise analysis also calculates an approximation to the gain G as a by-product of the input-referred noise calculation. It turns out that if it is assumed that the circuit input impedance is matched, i.e., the impedance seen by the port component shown as the dashed box in [Figure 12-14](#) on page 972, is the same as the source impedance, then the gain needed for the input-referred noise calculation is also the gain G .

However, the Pnoise analysis is based on a series of PXF analyses and when the input is not matched, then a single PXF analysis is not sufficient to calculate G . PSP analysis is needed to accurately calculate G independent of match conditions.

Figure 12-14 Pnoise with Port Component



Coincidentally, because of the matched-input assumption made both in Pnoise analysis and in the definition of the port, the gain G from circuit input to circuit output that is reported by Pnoise analysis happens to be exactly twice the gain from the source to the output, G_s . This can be seen by noting that if the circuit input is matched, then the circuit acts as a voltage divider, and the voltage at V_{in} is

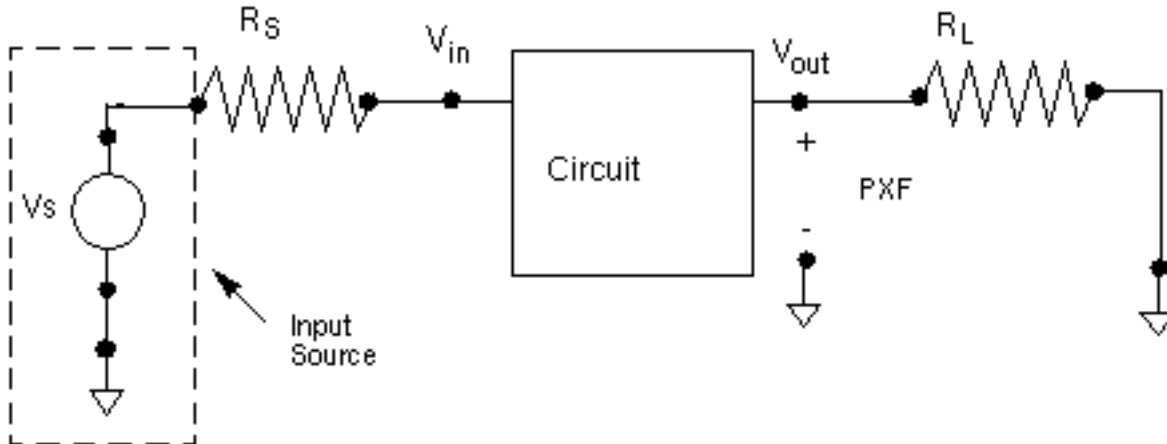
$$V_{in} = \frac{Z_{in}}{Z_{in} + R_s} V_s = \frac{R_s}{R_s + R_s} V_s = \frac{V_s}{2}$$

Thus in this special case, V_{out}/V_{in} is twice V_{out}/V_s . The *gain* reported by Pnoise is twice G_s regardless of the match conditions.

To alleviate this confusion, in future releases the *gain* number will not be reported by Pnoise analysis when the input is driven by a port.

Now consider when a separate `vs` source and resistor are used to drive the circuit, as shown in [Figure 12-15](#) on page 973. In this case, the voltage source would be specified as the input probe. Spectre computes the gain from the input probe to the output, which in this case is the gain from V_s to the output, G_s , since V_s was specified as the input source.

Figure 12-15 Pnoise With vsrc Component



Phase Noise

The Direct Plot form in Analog Artist plots *output noise* and *phase noise*. The phase noise form is designed for use in oscillator noise computations.

The *output noise* plots are the Total Output Noise (*out*) data, taken directly from the psf files with no modification.

The *phase noise* plots show the noise power relative to the carrier power. Technically the plot is not phase noise, simply normalized output noise. However, for oscillators, close to the fundamental frequency, the noise is mostly phase noise.

The normalization is done relative to the power in the fundamental component of the noise-free oscillation as calculated by the PSS analysis. If you look at the PSS analysis results in the frequency domain, the fundamental will have a particular amplitude V_1 , which means that the fundamental component of the oscillation is $V_1 \cos(2\pi f_c + \varphi)$, for some φ , where f_c is the fundamental frequency.

Defining

$$\text{noise power} = \text{out}^2,$$

where *out* is the Total Output Noise

$$\text{carrier fundamental power} = V_1^2/2$$

the normalized power is

$$\text{normalized power} = (\text{out}^2)/[V_1^2/2]$$

Thus the formula used by Direct Plot in Artist 4.4.3 and later to plot the normalized power (*phase noise*) in dB10 is:

$$\text{normalized power in dB} = 10\log_{10}\{(\text{out}^2)/[V_1^2/2]\}$$

For more information on phase noise, see [Appendix A, "Oscillator Noise Analysis,"](#) which summarizes how to get good phase noise calculations.

Frequently Asked Questions

The noise figure computed by <Procedure X> seems inaccurate, inconsistent, or just plain wrong.

The key to noise figure computation is to remember that three pieces of information are needed:

- The total noise at the output
- How much of that noise is due to the output load
- How much of that noise is due to the input source

Noise figure is the log of noise factor, and noise factor is fundamentally the total output noise less the noise due to the load ratioed to the noise due to the input. The easiest way to be sure of getting the correct noise figure is to use ports as input and output sources and loads and to have the Spectre simulator perform the noise figure computation.

SpectreRF correctly accounts for the noise due to the load, as well as the different possible treatments of noise from the input source that result in different types of noise figure that are used in RF circuits (customary SSB, IEEE SSB, DSB, as discussed in ["Noise Figure"](#) on page 954). See ["Performing Noise Figure Computations"](#) on page 954 for information on setting up noise figure computation.

Most mistakes in noise figure calculations arise when improperly accounting for items two and three in the list above. Hand computations of noise figure require that you correctly account for noise from the load and source, which can be difficult. For example, some users have tried to use the information from the noise summary table to calculate noise figure. The noise summary table reports total noise contributions, from all sidebands present in the Pnoise analysis, sorted by contributor. If the total noise reported in the noise summary is used

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

as the numerator for the noise factor computation the wrong answer is obtained because the noise due to the load has not been subtracted off. Likewise, the noise summary table does not sort information by sideband. To compute SSB noise figure, the denominator of the noise factor fraction must be only the noise from the reference sideband, not the total from all sidebands as reported in the noise summary. Noise figure calculations based on the reported input-equivalent noise, or on output noise and gain, suffer similar problems.

You can compute the noise figure by hand, but you must take care. For example, use the following steps to compute the customary single-sideband noise figure from raw data.

- Perform a Pnoise analysis with the *saveallsidebands* parameter set to *yes*.
- Using the results browser or direct plot, obtain the total noise at the output. Call this *OutputNoise*.
- Now use the results browser to obtain the amount of total noise due to the load contributed from the zero sideband. Call this *LoadNoise*.
- Finally, again using the results browser, obtain the amount of noise due to the source contributed from the reference sideband. Call this *sourcenoise*.

The noise factor is then

$$F = \frac{\text{OutputNoise} - \text{LoadNoise}}{\text{SourceNoise}}$$

Be sure to use the same units on all quantities. Note that, because in this procedure the noise due to the source and load are obtained by hand inspection, the procedure can be used to compute noise figure even if ports are not used to drive the circuit. Hand calculations are not recommended because they are tedious and prone to error.

In versions of Analog Artist prior to 4.4.5, it was possible to perform noise figure computations in ways that incorrectly or incompletely specified the sources/input or load/output. Since, at that time, there was no way for the simulation environment to track down which component was contributing, for example, the *LoadNoise*, this part of the noise factor might have been neglected.

The rule of thumb is: *if you have not specified your sources and loads to the simulator, the simulator does not know what the loads and sources are*, so the simulator may not be able to compute noise figure properly. Generally, the input to the circuit must be a *port*, and *not* some other electrically equivalent combination of components. The output may be a resistor probe, or it may be a port. In up-to-date versions of software, 4.4.2 and later, using ports (or possibly a resistor as the output probe) will insure accurate computations. Inspect the netlist if you are unsure about the identity of the components presented to the simulator.

Finally, if you ever have any doubt about the propriety of the procedure you are using, use the results browser to access the noise figure directly from the simulation data. This data is correct in all software versions.

Why is the gain reported by Pnoise twice the gain that I expect?

- First, unlike some circuit simulators, Pnoise and PSP compute the gain of the circuit from its input to its output, instead of from an external source to the output. See [“Definitions of Gain”](#) on page 967 to see if the gain you expect is actually the gain that you want.
- Second, Pnoise makes certain assumptions about input matching in order to calculate this gain. See [“Gain Calculations in Pnoise”](#) on page 971 for further explanations of the way Pnoise calculates gain.

In general, you should not use the gain calculated by Pnoise unless you know your circuit is input-matched. Use PSP instead.

Does SpectreRF compute single-sideband (SSB) or double-sideband (DSB) noise figure?

The SpectreRF Pnoise analysis computes SSB noise figure, (except in certain special cases). PSP can compute both SSB and DSB noise figure. See [“Noise Figure”](#) on page 954.

Why does the axis for noise or noise figure have negative frequencies?

The axis labeling in Spectre is fairly independent of the actual computation. In all cases, Spectre computes single-sided noise (that is, there is no need to add noise from *negative* frequencies to get the total noise power or noise figure).

Recall that PSP analysis performs many computations at once—periodic S-parameters, noise correlation matrices, noise figure, etc. In 4.4.5, for historical reasons, all these quantities are stored as a sweep with a single axis. An axis appropriate for visualizing s_{11} may not be good to display NF.

To obtain strictly positive axis labels for noise quantities in PSP, choose a positive integer for the output harmonic, use *sweep_type=relative*, and set *freqaxis=out* on the PSP analysis options form.

How does match affect the noise and gain calculations in SpectreRF?

If the input impedance is not matched to the port impedance, the gain and input referred noise, or IRN, reported by Pnoise analysis is probably not the gain you want. out , F , NF , and in (input-referred noise) will still be correct.

PSP analysis computes the correct G , in (input-referred noise), F and NF regardless of match.

Will Spectre include noise parameter data from S-parameter files?

Spectre supports writing and reading S-parameter files, but as of the 4.4.5 release it does not write noise parameter data into files, nor can noise parameter data be read from file and included in a simulation.

Known Problems and Limitations

Dubious AC-Noise Analysis Features

When computing noise figure in circuits with a DC bias point in releases prior to 4.4.5, the noise figure available from the direct plot form was obtained from knowledge of the total output noise computed with a Noise analysis, the source impedance as entered on a form, and circuit gain data obtained in an auxiliary AC analysis. Simulators other than Spectre still require this procedure.

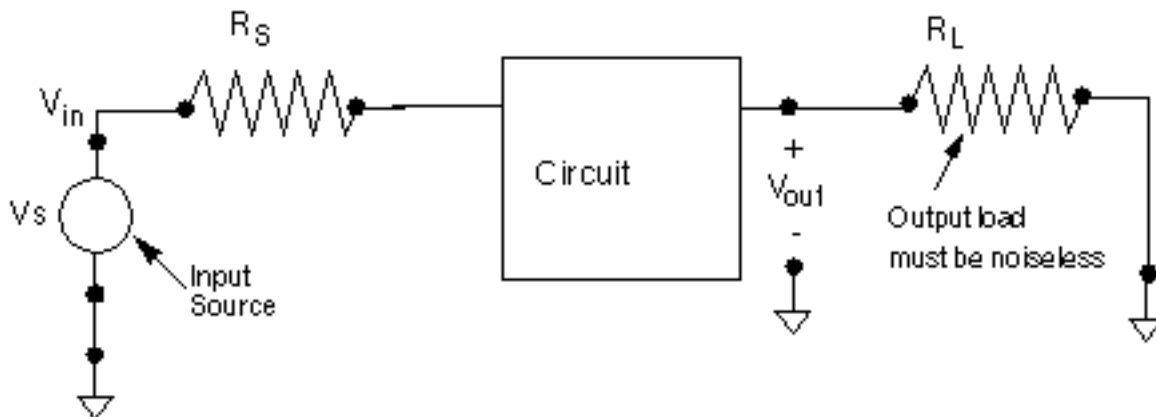
In 4.4.5, if you use ports as the output load and input source, the noise figure is obtained directly from the Spectre simulator data. An artifact of the previous use model is that the noise figure direct plot still requires an AC analysis to run, but this analysis has no effect on the computed noise figure if ports are used as sources and loads.

If ports are not used as the input source and load (for example, using a $vsin$ component as the input and a pair of nodes as the output) it is still possible to compute noise figure in the Analog Artist environment, but the procedure is not recommended as it can be tricky to obtain the desired result (see [“Performing Noise Figure Computations”](#) on page 954 and [“The noise figure computed by <Procedure X> seems inaccurate, inconsistent, or just plain wrong.”](#) on page 974).

To obtain reasonable noise figure calculations using this (unsupported) procedure, you must configure the circuit as shown in [Figure 12-16](#) on page 978. Be sure to select the proper nodes as input and output (note that a $port$ component cannot be used in this topology because the Vin node cannot be accessed). The contribution of the output load to the total

noise must be eliminated by making the node noiseless, for example by setting *Generate noise?* in a resistor component to *no*.

Figure 12-16 Alternative (Undesirable) Noise Figure Computation Setup



Note that you cannot use this topology to compute noise figure in Pnoise, SP or PSP analyses.

Gain in Pnoise and PSP Analyses Inconsistent

The *gain* computed by the Pnoise analysis is really a number that relates equivalent input and output noise. PSP analysis computes a somewhat different quantity that can be interpreted as a circuit gain. If the input impedance of the circuit is matched, the PSP and Pnoise gains will match. See the ["Gain Calculations"](#) on page 967 for details on computing circuit gains.

Harmonics and Sidebands in PSP, PAC, PXF, and Pnoise Analyses

Frequent users of SpectreRF may already be familiar with the concepts of *harmonics* and *sidebands*.

- Typically, the term *harmonic* is used to refer to a signal at a multiple of the carrier or local oscillator frequency.
- If an additional input is applied, such as for RF data, additional responses appear as *sidebands*.

Affirma RF Simulator (SpectreRF) User Guide

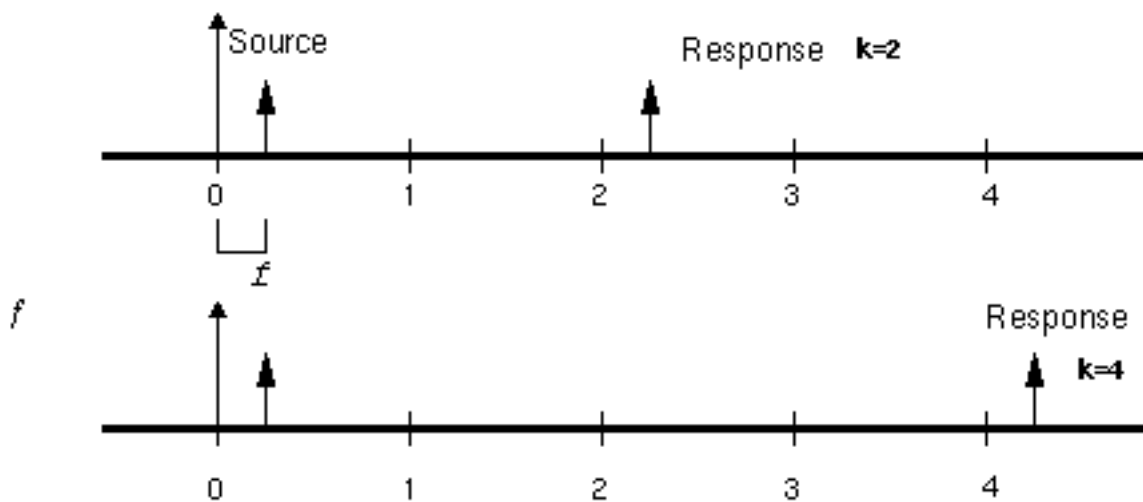
Using PSP and Pnoise Analyses

In small-signal analysis, only the first-order sidebands are computed.

In the context of small-signal analysis, a frequency is specified by an index k and a frequency offset f . These numbers specify a frequency $kf_0 + f$. In an absolute indexing scheme, such as used in PSP analysis, if the frequency f is restricted to lie in an interval of size f_0 , then for any given frequency, the numbers k and f are unique.

For example, consider an upconversion mixer, as shown in [Figure 12-17](#) on page 979. Two possible computations are shown. In both cases, the source is at $k=0$, baseband, and relative offset f . In the top half of the figure, the response is at $k=+2$, frequency f , and in the bottom half, $k=4$, offset f .

Figure 12-17 Absolute Indexing Schemes in SpectreRF Small-Signal Analyses



In contrast, PAC, PXF, and Pnoise analyses use a relative indexing scheme. In a relative indexing scheme, a single frequency may be referred to in different ways depending on context.

Figure 12-18 Relative Indexing in SpectreRF

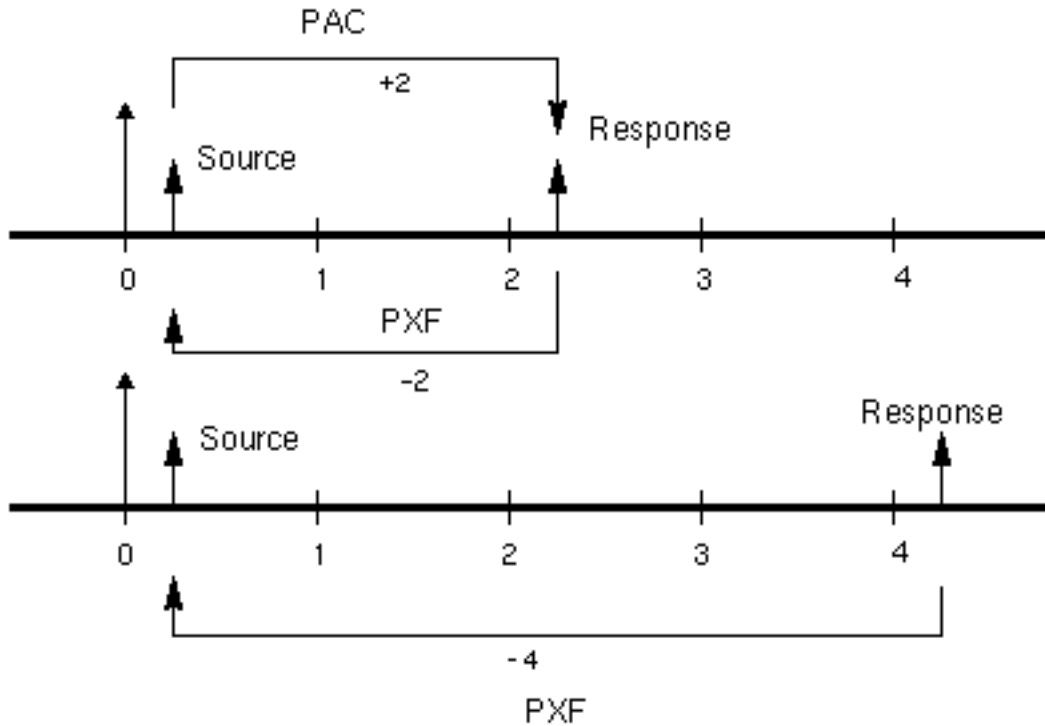


Figure 12-18 on page 980 shows how relative indexing may change depending on context. In the top half of the figure, a source may be specified in PAC analysis using either *sweepstype=absolute* and frequency $-f_0+f$, or *sweepstype=relative*, *relharmnum=-1*, and frequency f . The response is shown at frequency f_0+f , which is $k=2$, because $f_0+f = -f_0+f+2f_0$.

If instead a PXF analysis was used to analyze the circuit, then the location of the *response* is specified with either *sweepstype=absolute* and frequency f_0+f , or *sweepstype=relative* and *relharmnum=+1*. The source is at $k=-2$. Now consider the bottom half of the figure. The response is shown at the frequency $3f_0+f$. If PXF analysis is used to analyze the circuit, the source lies at $k=-4$.

Note that if all the frequencies involved in the above examples were shifted upwards by f_0 , then the indexing in PSP analysis would change by $+1$, but the indexing of the transfer functions in PXF and PAC analyses would remain unchanged, even though different frequencies, and thus physical different transfer functions, will be computed.

The names used for the responses and transfer functions in the SpectreRF environment are not entirely consistent in release 4.4.5 and earlier. The SpectreRF interface asks for *sidebands* in order to specify which of the transfer functions associated with the relative

Affirma RF Simulator (SpectreRF) User Guide

Using PSP and Pnoise Analyses

index k are wanted. This is reasonable because the small-signal responses appear as sidebands to the harmonics of the fundamental frequency of the PSS steady-state solution. However, the data is output and displayed with the label *harmonic*. Thus the index naming is not consistent. However, since the transfer function associated with the relative index k results from the small-signal being convolved with the k^{th} harmonic of the steady-state solution, the *harmonic* label is not entirely inappropriate either.

Index

Symbols

./inilmg file, in LMG [333](#)

Numerics

1dB compression point
 low noise amplifier, plotting [301](#)
 on Results forms [132](#)
 1st Order Harmonic [130](#)
 3rd Order Harmonic [130](#)

A

Accuracy Defaults specification [99](#), [124](#)
 conservative [124](#)
 liberal [124](#)
 moderate [124](#)
 accuracy parameters
 allglobal [123](#), [124](#)
 alllocal [123](#), [124](#)
 lteratio [123](#), [124](#)
 maxacfreq [123](#)
 maxperiods [124](#)
 pointlocal [123](#)
 relref [123](#), [124](#)
 sigglobal [123](#), [124](#)
 steadyratio [123](#), [124](#)
 accuracy, of LMG models [340](#)
 Add Specific Points, entry on Choosing
 Analyses form [100](#)
 Add to Outputs specification, on Results
 forms [130](#)
 Additional Time for Stabilization, entry on
 Choosing Analyses form [100](#)
 allglobal parameter [123](#), [124](#)
 alllocal parameter [123](#), [124](#)
 Analog Artist Design Environment, using
 PKG in [381](#)
 analysis
 descriptions [26](#), [32](#), [40](#), [43](#)
 Analysis type specification, on Results
 forms [130](#)
 annotate parameter [124](#)

annotation parameters
 annotate [124](#)
 stats [124](#)
 assumptions
 for large and moderate signal analyses
 Pdisto [23](#)
 for large-signal analyses
 linearity assumption with PSS [22](#)
 periodicity assumption with PSS/
 SPSS [22](#)
 for small signal analyses [25](#)
 linearity [25](#)
 Auto Calculate Button [100](#)
 autodc parameter [125](#)
 Automatic Sweep Type setting [120](#)

B

bond wire
 shape function
 shape functions, in PKG [370](#)
 Bond Wire Diameter type-in field, in
 PKG [370](#)
 bond wire file format, in PKG [371](#)
 Bond Wire Length type-in field, in PKG [370](#)
 Bond Wire Number type-in field, in
 PKG [369](#)
 bond wire setup button [354](#)
 Bond Wire Setup form, in PKG [367](#)
 function buttons [367](#)
 menu buttons [367](#)
 bond wire setup, in PKG [348](#)
 bond wire shape functions
 shape function, in PKG [367](#)
 Bonding Pad Height type-in field, in
 PKG [357](#)
 Bondpad Coord (Yb) type-in field, in
 PKG [365](#)
 Bondpad Length (Lb) type-in field, in
 PKG [365](#)
 Bondpad Width (Wb) type-in field, in
 PKG [365](#)
 building blocks, in PKG [347](#)

C

- Cadence libraries, setting up [140](#), [141](#)
- Calculate Capacitance buttons, in
 - PKG [373](#)
- Calculate Inductance buttons, in PKG [375](#)
- Calculate Parameters function button, in
 - LMG [326](#), [330](#)
- Cancel button, in PKG [377](#)
- Capacitance Accuracy buttons, in
 - PKG [374](#)
- capacitance calculation, in PKG [349](#)
- Center - Span specification, Frequency
 - Sweep Range (Hz) fields [102](#)
- Choosing Analyses form [99](#)
 - Accuracy Defaults specification [99](#)
 - Analysis specification [100](#)
 - Auto Calculate Button [100](#)
 - Do Noise [101](#)
 - Enabled button [101](#)
 - Frequency (Hz) field [101](#), [107](#)
 - Frequency Sweep Range (Hz)
 - fields [101](#)
 - Center - Span specification [102](#)
 - Single - Point specification [102](#)
 - Start - Stop specification [102](#)
 - Input Source specification [104](#), [105](#)
 - Input Voltage Source specification [105](#)
 - Options specification [106](#)
 - Oscillator button [106](#)
 - Output
 - harmonics selection [108](#)
 - Number of harmonics [109](#)
 - Output harmonics selection
 - Array of harmonics [110](#), [115](#)
 - Select from range [109](#)
 - Output specification for Pnoise and
 - PXF [101](#), [106](#)
 - PAC analysis
 - with PSS [60](#)
 - with SPSS [61](#)
 - Pdisto analysis [59](#)
 - Periodic Steady-State analysis [56](#), [57](#)
 - Periodic Steady-State analysis with
 - sweep [58](#)
 - Pnoise analysis
 - with PSS [62](#)
 - with SPSS [63](#)
 - PSS analysis [55](#)
 - PXF analysis
 - with PSS [64](#)
 - with SPSS [65](#)
- Reference Sideband specification [111](#), [112](#)
- Relative to Fundamental button [112](#)
- Save Initial Transient Results
 - specification [112](#)
- Select button [113](#)
- Select Ports [112](#)
- Sidebands specification [113](#)
 - Maximum sideband [101](#), [114](#)
 - Select from range [114](#)
- Small Signal Periodic Analysis list
 - box [116](#)
- Sweep Range (Hz) specification
 - Center - Span [118](#)
 - Start - Stop [118](#)
- Sweep specification [116](#)
 - variable [116](#)
- Sweep Type
 - Automatic [120](#)
 - Linear [121](#)
 - Logarithmic [121](#)
- Sweep Type specification [121](#)
 - Add Specific Points [122](#)
 - Linear [121](#)
 - Logarithmic [121](#)
- Clear/Add button, Fundamental Tones list
 - box [104](#)
- Close button, in PKG [377](#), [378](#)
- cmin parameter [124](#)
- compression parameter [127](#)
- Conductivity type-in field
 - in LMG [326](#)
 - in PKG [370](#)
- Conductor Distances type-in field, in
 - LMG [326](#)
- Conductor Height (h) type-in field, in
 - LMG [326](#)
- Conductor Height type-in field, in LMG [330](#)
- Conductor Length type-in field, in
 - LMG [326](#), [330](#)
- Conductor Thickness type-in field, in
 - LMG [326](#), [330](#)
- Conductor Width type-in field, in LMG [325](#), [330](#)
- convergence parameters
 - cmin [124](#)
 - gear_order [125](#)
 - readns [124](#)
 - solver [125](#)

- exp [125](#)
- std [125](#)
- turbo [125](#)
- tolerance [124](#)
- conversion gain measurement, mixer [180](#)
 - conversion gain, plotting [185](#)
 - power supply rejection, plotting [188](#)
- Coupled Pins Per Side type-in field, in PKG [377](#)
- Create button, in PKG [378](#)
- Create DA Terminal type-in field, in PKG [378](#)
- Create Macromodel function button, in LMG [326](#)
- current
 - gain specification, on Results forms [132](#)
 - specification, on Results forms [132](#)

D

- data entry
 - Fundamental Tones list box
 - buttons [102](#)
 - fields [102](#)
- data entry section on form in LMG [325](#)
- dB20, as Results form modifier [135](#)
- default values, in LMG [326](#)
- Delete button [104](#)
- Die Attach Height type-in field, in PKG [362](#)
- Die Attach Length (Lda) type-in field, in PKG [362](#)
- die attach setup
 - button [354](#)
 - in PKG [361](#)
- Die Attach Thickness (Tda) type-in field, in PKG [362](#)
- Die Attach Width (Wda) type-in field, in PKG [362](#)
- Die Bond Site (x, y) type-in field, in PKG [370](#)
- Die Cavity Side buttons, in PKG [364](#)
- Dielectric Constant type-in field
 - in LMG [329](#)
 - in PKG [374](#)
- Dielectric Thickness (d) type-in field, in LMG [325](#)
- Dielectric Thickness type-in field, in LMG [329](#)
- DIP packages, setup [359](#)

- display section on form, in LMG [325](#)
- Do Noise, Choosing Analyses form [101](#)

E

- Edit Object Properties form [383](#)
 - use external model file button [336](#)
- elevation (e) type-in field, in PKG [357](#)
- Enabled button [101](#)
- equivalent noise resistance, low noise amplifier, plotting [285](#)
- errpreset parameter [99](#)
- euler parameter [126](#)
- examples
 - low noise amplifier [256](#)
 - 1dB compression point, plotting [301](#)
 - noise calculations [292](#)
 - output voltage distribution [261](#), [266](#)
 - PXF analysis [312](#)
 - S-parameter analysis [272](#)
 - voltage standing wave ratio, plotting [278](#)
 - S-parameter Noise analysis [279](#)
 - equivalent noise resistance, plotting [285](#)
 - load stability circles, plotting [286](#)
 - minimum noise figure [282](#)
 - noise circles, plotting [290](#)
 - noise figure [282](#)
 - source stability circles [288](#)
 - third-order intercept point, plotting [307](#)
 - voltage gain
 - calculation [261](#)
 - plotting [264](#)
- mixer
 - conversion gain measurement [180](#)
 - conversion gain, plotting [185](#)
 - power supply rejection, plotting [188](#)
 - harmonic distortion
 - measurement [151](#)
 - intermodulation distortion
 - measurement with Pdisto [202](#)
 - noise figure measurement [160](#)
 - 1dB compression point [188](#)
 - noise figure measurement with PSP [165](#)
 - periodic S-parameter plots [165](#)

Affirma RF Simulator (SpectreRF) User Guide

- third-order intercept measurement
 - with PSS sweep and PAC [194](#)
- modeling transmission lines using the LMG form (use model one)
 - contents of example model [327](#)
 - setup [327](#)
- modeling transmission lines using the Transmission Line Modeler form (use model one)
 - creating the model [327](#)
- modeling transmission lines using the Transmission Line Modeler form (use model one)
 - use of lumped models within Analog Artist [334](#)
- modeling transmission lines without using the Transmission Line Modeler form (use model two) [338](#)
- oscillator simulation, of differential oscillator [241](#)
 - output noise, plotting [252](#)
- oscillator simulation, of tline3oscRF
 - phase noise, plotting [240](#)
 - steady state solution, plotting [238](#)
- package Modeling with a mixer using PKG [385](#)
- executables, in PKG setup [350](#)
- exp parameter value [125](#)
- Expr field, Fundamental Tones list box [102](#)
- Extrapolation Point specification, on Results forms [130](#)

F

- FASTCAP, in PKG [346](#)
- FASTHENRY, in PKG [346](#)
- File menu
 - Bond Wire Setup form [367](#)
 - in LMG [323](#)
- first-order harmonic specification, on Results forms [131](#)
- flicker noise computation [47](#)
- Fmax type-in field, in LMG [326](#), [330](#)
- forms
 - Choosing Analyses [99](#)
 - Options [70](#)
 - Results [130](#)
- Fourier analysis [26](#)

- freq. unit menu choice [354](#)
- Freq. Unit menu choices, in LMG [325](#)
- freqaxis parameter
 - absin [127](#)
 - in [127](#)
 - out [127](#)
- Frequency (Beat) specification,
 - Fundamental Tones [100](#), [101](#)
- Frequency (Hz) field [101](#), [107](#)
- Frequency Sweep Range (Hz) fields [101](#)
- function
 - buttons, in LMG [322](#)
 - buttons, in PKG main window [354](#)
 - buttons, in PKG RF IC Package Modeler window
 - bond wire setup [354](#)
 - die attach setup [354](#)
 - package dimension setup [354](#)
 - specification, on Results forms [131](#)
- function buttons
 - in LMG [326](#)
- function buttons, in PKG RF IC Package Modeler window
 - internal lead frame setup [354](#)
 - macromodel generation [354](#)
 - parameter extraction [354](#)
- Function specification, on Results forms
 - 1dB compression [132](#)
 - current [132](#)
 - current gain [132](#)
 - IP3 curves [132](#)
 - power [132](#)
 - transconductance [132](#)
 - transimpedance [132](#)
 - voltage [132](#)
 - voltage gain [132](#)
- Fundamental Tones [102](#)
- Frequency (Beat) specification [100](#), [101](#)
- list box [102](#), [104](#)
 - Clear/Add button [104](#)
 - data entry buttons [102](#)
 - data entry fields [102](#)
 - Delete button [104](#)
 - Expr field [102](#)
 - Harms field [102](#)
 - Name field [102](#)
 - Signal field [102](#)
 - Srclid field [102](#)
 - Update from Schematic button [104](#)
 - Value field [102](#)

Affirma RF Simulator (SpectreRF) User Guide

Period (Beat) specification [100](#), [101](#)
Fundamental Tones list box [104](#)
Fused to Die Attach buttons, in PKG [364](#)

G

gear_order parameter [125](#), [126](#)
gear2 parameter value [124](#)
gear2only parameter value [124](#)
general dimension setup, in PKG [348](#)
Gnd/pkg Distance type-in field, in PKG [374](#)
gotolink CDMA [775](#)
gotolink chap10.fm
 firstpage [793](#)
gotolink chap2.doc
 selectDesignVariable [450](#)
Gotolink DQPSK [775](#)
gotolink eye-diagram [775](#)
gotolink Figure 4-16 [798](#)
gotolink Figure 4-17 [799](#)
gotolink GSM [775](#)
Ground Plane Effect buttons, in PKG [373](#)
GUI files, in PKG setup [350](#)

H

harmonic distortion measurement,
 mixer [151](#)
Harms field, Fundamental Tones list
 box [102](#)
harms parameter [123](#)

I

ic parameter [125](#)
Imaginary, as Results form modifier [135](#)
Inductance Accuracy buttons, in PKG [375](#)
inductance calculation, in PKG [349](#)
initial condition parameters
 ic [125](#)
 readic [125](#)
 skipdc [125](#)
 autodc [125](#)
 no [125](#)
 rampup [125](#)
 waveless [125](#)
 yes [125](#)
initialization files, in PKG setup [350](#)

Input Current Source specification
 Input Current Source specification [105](#)
Input Port Source specification [105](#)
Input Port Source specification, Input Port
 Source specification [105](#)
Input Source specification [104](#)
integration method parameters
 method [124](#), [126](#)
 euler [126](#)
 gear2 [124](#), [126](#)
 gear2only [124](#), [126](#)
 trap [126](#)
 traonly [124](#), [126](#)
intermodulation distortion
 measurement [34](#)
 with Pdisto, mixer [202](#)
internal lead frame setup
 button [354](#)
 in PKG [362](#)
internal lead frame setup, in PKG
 function buttons [363](#)
 menu buttons [362](#)
internal lead frme setup, in PKG [348](#)
IP3 curves specification, on Results
 forms [132](#)
isnoisy parameter [279](#)

L

large and moderate signal analyses
 assumptions
 Pdisto [23](#)
large-signal analyses
 assumptions
 periodicity assumption with PSS/
 SPSS [22](#)
 linearity assumption
 with PSS/SPSS [22](#)
 use model with small-signal
 analyses [23](#)
Lead Frame Conductivity type-in field, in
 PKG [375](#)
lead frame file format [365](#)
Length (L) type-in field, in PKG [357](#)
Length (Ld) type-in field, in PKG [357](#)
Length (Lp) type-in field, in PKG [358](#), [365](#)
Length (Lp1) type-in field, in PKG [357](#), [360](#)
Length (Lp2) type-in field, in PKG [357](#), [361](#)
Length (Lp3) type-in field, in PKG [361](#)
Length (Lp4) type-in field, in PKG [361](#)

Affirma RF Simulator (SpectreRF) User Guide

- length unit menu choice [354](#)
- Length Unit menu choices, in LMG [324](#)
- license checking, in PKG setup [350](#)
- Linear Sweep Type setting [121](#)
- linearity
 - Pdisto [23](#)
 - PSS [22](#)
 - small-signal analyses [25](#)
- LMG [320](#)
 - ./inilmg file [333](#)
 - accuracy of [340](#)
 - data entry section [325](#)
 - Conductivity [326](#)
 - ConductivityConductivity type-in field, in LMG [330](#)
 - Conductor Distances [326](#)
 - Conductor Height [326, 330](#)
 - Conductor Length [326, 330](#)
 - Conductor Thickness [326, 330](#)
 - Conductor Width [325, 330](#)
 - Dielectric Constant [329](#)
 - Dielectric Constant (er) [325](#)
 - Dielectric Thickness [329](#)
 - Dielectric Thickness (d) [325](#)
 - Fmax [326, 330](#)
 - Number of Conductors [325, 329](#)
 - default values [326](#)
 - Dielectric Constant (er) type-in field, in LMG [325](#)
 - display section on form [325](#)
 - File menu
 - Quit [323](#)
 - Save Current Settings [323](#)
 - Subcircuit Name [323](#)
 - function buttons [322, 326](#)
 - Calculate Parameters [326, 330](#)
 - Create Macromodel [326](#)
 - Macromodel Creation pop-up window [331](#)
 - Options menu
 - Freq. Unit [325](#)
 - Length Unit [324](#)
 - Lossy, Narrow Band menu choice [324](#)
 - Lossy, Narrow Band, Single Line menu choice [324](#)
 - Lossy, Wide Band, Single Line menu choice [324](#)
 - Model Type [324](#)
 - Subckt Format [324](#)
 - Save Current Settings form [333](#)
 - Subcircuit Name form [331](#)
 - Transmission Line Modeler form [322](#)
 - load stability circle
 - low noise amplifier, plotting [286](#)
 - Logarithmic Sweep Type setting [121](#)
 - Lossy, Narrow Band, Single Line menu choice, in LMG [324](#)
 - Lossy, Wide Band, Single Line menu choice, in LMG [324](#)
 - low noise amplifier
 - 1dB compression point, plotting [301](#)
 - noise calculations [292](#)
 - output voltage distribution [261, 266](#)
 - PXF analysis [312](#)
 - simulation example [256](#)
 - S-parameter analysis [272](#)
 - voltage standing wave ratio, plotting [278](#)
 - S-parameter Noise analysis [279](#)
 - equivalent noise resistance, plotting [285](#)
 - load stability circles, plotting [286](#)
 - minimum noise figure [282](#)
 - noise circles, plotting [290](#)
 - noise figure [282](#)
 - source stability circles [288](#)
 - third-order intercept point, plotting [307](#)
 - voltage gain
 - calculation [261](#)
 - plotting [264](#)
 - low noise amplifier, simulation example [256](#)
 - Iteratio parameter [29, 123, 124](#)

M

 - Macromodel Creation pop-up window [331](#)
 - macromodel Generation
 - form, in PKG [377](#)
 - macromodel generation
 - button [354](#)
 - in PKG [349](#)
 - Macromodel Name type-in field, in PKG [378](#)
 - macromodel schematic, in PKG [379](#)
 - Magnitude, as Results form modifier [135](#)
 - main window, in PKG [351](#)
 - function buttons [354](#)
 - save current setting with bond wires [372](#)

Affirma RF Simulator (SpectreRF) User Guide

maxacfreq parameter [29](#), [123](#)
maxiters parameter [126](#)
maxperiods parameter [124](#)
maxsideband parameter [46](#)
maxstep parameter [123](#), [129](#)
mesh size versus accuracy, in PKG
 capacitance calculations [374](#)
 inductance calculations [375](#)
method parameter [124](#), [126](#)
 euler [126](#)
 gear2 [124](#)
 gear2 parameter value [126](#)
 gear2only [124](#)
 gear2only parameter value [126](#)
 trap [126](#)
 traonly [124](#), [126](#)
Microstrip menu choice, in LMG [325](#)
minimum noise figure, low noise
 amplifier [282](#)
mixer
 conversion gain measurement [180](#)
 conversion gain, plotting [185](#)
 power supply rejection, plotting [188](#)
 harmonic distortion measurement [151](#)
 intermodulation distortion measurement
 with Pdisto [202](#)
 noise figure measurement [160](#)
 1dB compression point [188](#)
 noise figure with PSP
 measurement [165](#)
 periodic S-parameter plots
 measurement [165](#)
 third-order intercept measurement with
 PSS sweep and PAC [194](#)
Model Type menu choice, in LMG [324](#)
modeling example with mixer, for PKG [385](#)
modeling transmission lines
 using LMG outside Analog Artist (use
 model one)
 described [320](#)
 example [327](#)
modeling transmission lines from inside
 Analog Artist (use model two)
 described [321](#)
 example [338](#)
Modifier specification, on Results
 forms [135](#)
 dB20 [135](#)
 Imaginary [135](#)
 Magnitude [135](#)
 Phase [135](#)

Real [135](#)

N

Name field, Fundamental Tones list
 box [102](#)
Narrow Band menu choice, in LMG [324](#)
newlink CDMA [794](#)
newlink DQPSK [798](#)
newlink eye-diagram [794](#)
newlink Figure 4-16 [799](#)
newlink Figure 4-17 [800](#)
newlink GSM [796](#)
newlink selectDesignVariable [116](#)
newlink Variable [116](#)
Newton parameters
 maxiters [126](#)
 restart [126](#)
nodes_and_terminals, stimuli parameter
 value [127](#)
noise calculations, low noise amplifier [292](#)
noise circles
 low noise amplifier, plotting [290](#)
noise figure measurement [46](#)
 low noise amplifier [282](#)
 mixer [160](#)
 1dB compression point
 calculating [188](#)
noise figure measurement with PSP,
 mixer [165](#)
Noise Summary form [299](#)
noise temperature [771](#), [898](#)
noise file parameter [47](#)
noisetemp parameter [47](#), [771](#), [898](#)
noisevec parameter [47](#)
Nth Order Harmonic button [135](#)
Number of Conductors type-in field, in
 LMG [325](#), [329](#)

O

oppoint parameter [127](#)
oprobe parameter [46](#)
options
 accuracy parameters
 allglobal [123](#), [124](#)
 alllocal [123](#), [124](#)
 lteratio [123](#), [124](#)
 maxacfreq [123](#)

Affirma RF Simulator (SpectreRF) User Guide

- maxperiods [124](#)
- pointlocal [123](#)
- relref [123](#), [124](#)
- sigglobal [123](#), [124](#)
- steadyratio [123](#), [124](#)
- annotation parameters
 - annotate [124](#)
 - stats [124](#)
- convergence parameters
 - cmin [124](#)
 - gear_order [125](#)
 - readns [124](#)
 - solver [125](#)
 - tolerance [124](#)
- initial condition parameters
 - ic [125](#)
 - readic [125](#)
 - skipdc [125](#)
- integration method parameters
 - method [126](#)
- Newton parameters
 - maxiters [126](#)
 - restart [126](#)
- output parameters
 - compression [127](#)
 - freqaxis [127](#)
 - oppoint [127](#)
 - skipcount [127](#)
 - skipstart [127](#)
 - skipstop [127](#)
 - stimuli [127](#)
 - strobedelay [127](#)
 - strobeperiod [127](#)
- simulation interval parameters
 - tstart [128](#)
- state file parameters
 - swapfile [128](#)
 - write [128](#)
 - writefinal [128](#)
- time step parameters
 - maxstep [129](#)
 - step [129](#)
- Options button [106](#)
- Options form [70](#), [123](#)
- ACCURACY PARAMETERS [123](#)
- ANNOTATION PARAMETERS [124](#)
- CONVERGENCE PARAMETERS [124](#)
- INITIAL CONDITION PARAMETERS [125](#)
- INTEGRATION METHOD PARAMETERS [126](#)
- NEWTON PARAMETERS [126](#)
- OUTPUT PARAMETERS [126](#)
- PSS, Pdisto analyses [72](#), [73](#)
- PSS, SPSS analyses [70](#), [71](#)
- SIMULATION INTERVAL PARAMETERS [128](#)
- small-signal analyses [74](#), [75](#)
- STATE FILE PARAMETERS [128](#)
- TIME STEP PARAMETERS [129](#)
- Options menu
 - Bond Wire Setup form [367](#)
 - in LMG [324](#)
- Order specification, on Results forms [136](#)
- Oscillator button [106](#)
- oscillator noise analysis [724](#)
- oscillators
 - differential oscillator [241](#)
 - output noise, plotting [252](#)
 - starting in simulations [226](#)
 - tline3oscRF
 - phase noise, plotting [240](#)
 - steady state solution, plotting [238](#)
 - troubleshooting simulations for [254](#)
 - use of PSS with [225](#)
- Output harmonics selection [108](#)
 - Array of harmonics [110](#), [115](#)
 - Number of harmonics [109](#)
 - Select from range [109](#)
- output noise, plotting with oscillators [252](#)
- output parameters
 - compression [127](#)
 - freqaxis [127](#)
 - oppoint [127](#)
 - skipcount [127](#)
 - skipstart [127](#)
 - skipstop [127](#)
 - stimuli [127](#)
 - nodes_and_terminals [127](#)
 - sources [127](#)
 - strobedelay [127](#)
 - strobeperiod [127](#)
- Output specification for Pnoise and PXF [106](#)
 - current [107](#)
 - voltage [106](#)
- output voltage distribution, low noise amplifier [261](#), [266](#)
- overview, SpectreRF analyses [20](#)

P

- PAC [32](#)
- PAC analysis [32](#)
 - Choosing Analyses form
 - with PSS [60](#)
 - with SPSS [61](#)
 - freqaxis parameter [127](#)
 - intermodulation distortion measurement
 - with [34](#)
 - Results form [90](#)
 - synopsis [33](#)
- Package Modeler, *See* PKG
- package type menu choice [353](#)
- parameter extraction button [354](#)
- Parameter Extraction form, in PKG [372](#)
- parameters
 - leratio [29](#)
 - maxacfreq [29](#)
 - maxsideband [46](#)
 - maxstep [123](#)
 - noisefile [47](#)
 - noisetemp [47](#)
 - noisevec [47](#)
 - oprobe [46](#)
 - period [28](#)
 - refsideband [45](#)
 - reltol [29](#)
 - saveinit [27](#)
 - sidebands [46](#)
 - steadyratio [29](#)
 - stimuli [42](#)
 - tinit [28](#)
 - tonset [28](#)
 - tstart [28](#)
 - tstop [28](#)
 - values [47](#)
- parameters tonset [28](#)
- Pdisto analysis
 - assumptions
 - lack of periodicity [23](#)
 - linearity [23](#)
 - Choosing Analyses form [59](#)
 - Results form [87](#)
- PdistoResults [87](#)
- Period (Beat) specification, Fundamental Tones [100](#), [101](#)
- period parameter [28](#)
- periodic AC analysis, *See* PAC analysis
- periodic noise analysis, *See* Pnoise
 - analysis
 - periodic S-parameter plots, mixer [165](#)
 - Periodic Steady-State analysis
 - Choosing Analyses form [56](#), [57](#)
 - periodic steady-state analysis. *See* PSS analysis
 - periodic transfer function analysis, *See* PXF analysis
 - periodicity
 - Pdisto [23](#)
 - PSS/SPSS [22](#)
 - phase noise, discussion of [725](#)
 - and SpectreRF simulation [737](#)
 - frequently asked questions [744](#)
 - further reading [749](#)
 - models [728](#)
 - troubleshooting [739](#)
 - phase noise, with oscillators [226](#)
 - plotting [240](#)
 - Phase, as Results form modifier [135](#)
 - physical geometry modeling, for PKG [347](#)
 - Pin Bond Site (x, y) type-in field, in PKG [370](#)
 - Pin Coord (Xp) type-in field, in PKG [365](#)
 - Pin Number type-in field, in PKG [365](#), [370](#)
 - pin type menu choice [353](#)
 - Pin Width (Wp) type-in field, in PKG [365](#)
 - Pins per Side type-in field, in PKG [357](#), [359](#)
 - Pitch (P) type-in field, in PKG [357](#), [359](#)
 - PKG
 - Analog Artist Design Environment [383](#)
 - bond wire
 - setup [348](#)
 - bond wire file format [371](#)
 - Bond Wire Setup form [367](#)
 - Bond Wire Diameter type-in field [370](#)
 - Bond Wire Length type-in field [370](#)
 - Bond Wire Number type-in field [369](#)
 - Conductivity type-in field [370](#)
 - Die Bond Site (x, y) type-in field [370](#)
 - function buttons [367](#)
 - menu buttons [367](#)
 - Pin Bond Site (x, y) type-in field [370](#)
 - Pin Number type-in field [370](#)
 - use in example [395](#)
 - building blocks [347](#)
 - capacitance calculation [349](#)
 - Close button [378](#)
 - Create button [378](#)
 - Die Attach Setup form [361](#)

Affirma RF Simulator (SpectreRF) User Guide

- Die Attach Height type-in field [362](#)
- Die Attach Length (Lda) type-in field [362](#)
- Die Attach Thickness (Tda) type-in field [362](#)
- Die Attach Width (Wda) type-in field [362](#)
 - use in example [391](#)
- inductance calculation [349](#)
- Internal Lead Frame Setup form [362](#)
 - Bondpad Coord (Yb) type-in field [365](#)
 - Bondpad Length (Lb) type-in field [365](#)
 - Bondpad Width (Wb) type-in field [365](#)
 - Die Cavity Side buttons [364](#)
 - function buttons [363](#)
 - Fused to Die Attach buttons [364](#)
 - Length (Lp) type-in field [365](#)
 - menu buttons [362](#)
 - Pin Coord (Xp) type-in field [365](#)
 - Pin Number type-in field [365](#)
 - Pin Width (Wp) type-in field [365](#)
 - use in example [392](#)
 - Using Symmetry buttons [364](#), [365](#)
- internal lead frme setup [348](#)
- lead frame file format [365](#)
- macromodel generation [349](#)
- Macromodel Generation form [377](#)
 - Coupled Pins Per Side type-in field [377](#)
 - Create DA Terminal type-in field [378](#)
 - Macromodel Name type-in field [378](#)
 - Macromodel Type buttons
 - Macromodel Type buttons, in PKG [378](#)
 - use in example [400](#)
- macromodel schematic [379](#)
- main window [351](#)
- mesh size versus accuracy
 - capacitance calculations [374](#)
 - inductance calculations [375](#)
- overview of use [345](#)
- package dimension setup button [354](#)
- Package Dimension Setup form
 - Bonding Pad Height type-in field [357](#)
 - DIP packages [359](#)
 - Elevation (e) type-in field [357](#)
 - Length (L) type-in field [357](#)
 - Length (Ld) type-in field [357](#)
 - Length (Lp) type-in field [358](#)
 - Length (Lp1) type-in field [357](#), [360](#)
 - Length (Lp2) type-in field [357](#), [361](#)
 - Length (Lp3) type-in field [361](#)
 - Length (Lp4) type-in field [361](#)
 - Pins per Side type-in field [357](#), [359](#)
 - Pitch (P) type-in field [357](#), [359](#)
 - QFP packages [355](#)
 - SOIC packages [358](#)
 - Thickness (T) type-in field [357](#)
 - Thickness (t) type-in field [357](#)
 - Tilted Angle type-in field [357](#)
 - use in example [388](#)
 - Width (W) type-in field [357](#)
 - Width (w) type-in field [357](#), [359](#), [361](#)
 - Width (Wd) type-in field [357](#)
- package general dimension setup [348](#)
- package physical geometry modeling [347](#)
- Parameter Extraction form [372](#)
 - Calculate Capacitance buttons [373](#)
 - Calculate Inductance buttons [375](#)
 - Cancel button [377](#)
 - Capacitance Accuracy buttons [374](#)
 - Close button [377](#)
 - Dielectric Constant type-in field [374](#)
 - Gnd/pkg distance type-in field [374](#)
 - Ground Plance Effect buttons [373](#)
 - Inductance Accuracy buttons [375](#)
 - Lead Frame Conductivity type-in field [375](#)
 - Solve button [377](#)
 - use in example [398](#)
 - Working Frequency type-in field [375](#)
- RF IC Package Modeler form
 - file menu [353](#)
 - options menu [353](#)
 - use in example [385](#)
 - running the program [351](#)
 - save current setting in RF IC Package Modeler window and bond wires [372](#)
 - setup requirements [350](#)
 - executables [350](#)
 - GUI files [350](#)
 - initialization files [350](#)
 - license checking [350](#)
 - solving process for inductance [377](#)

Affirma RF Simulator (SpectreRF) User Guide

PKG, using in Analog Artist Design Environment [381](#)

plot mode specification, on Results forms [138](#)

Pnoise [43](#)

Pnoise analysis [43](#), [104](#)

- Choosing Analyses form
 - with PSS [62](#)
 - with SPSS [63](#)
- flicker noise computation [47](#)
- Input Source specification [104](#)
- Input Voltage Source specification [105](#)
- noise figure computation [46](#)
- Output specification [106](#)
- Reference Sideband specification [111](#), [112](#)
- Results form [92](#)
- synopsis [46](#)

pointlocal parameter [123](#)

power specification, on Results forms [132](#)

psin, using in SpectreRF simulation [757](#)

PSP analysis

- freqaxis parameter [127](#)

PSS [26](#)

PSS analysis [26](#)

- Choosing Analyses form [55](#)
- plotting results [81](#)
- Results form [83](#)
- spectral plots [81](#)
- synopsis [27](#)
- use model with small-signal analyses [23](#)
- use with oscillators [225](#)

PSS Noise Summary form [299](#)

PXF [40](#)

PXF analysis [40](#)

- Choosing Analyses form
 - with PSS [64](#)
 - with SPSS [65](#)
- freqaxis parameter [127](#)
- low noise amplifier [312](#)
- Output specification [106](#)
- Results form [93](#)
- synopsis [42](#)

Q

QFP packages, setup [355](#)

QPnoise analysis

- Reference Sideband specification [111](#)

Quit menu choice, in LMG [323](#)

R

rampup parameter [125](#)

readic parameter [125](#)

readns parameter [124](#)

Real, as Results form modifier [135](#)

Reference Harmonic specification, on Results forms [133](#)

Reference Sideband specification, Pnoise analysis [111](#), [112](#)

Reference Sideband Specification, QPnoise analysis [111](#)

refsideband parameter [45](#)

Relative to Fundamental button [112](#)

relref parameter [123](#), [124](#)

reltol parameter [29](#)

resistor noise generation [279](#)

restart parameter [126](#)

Results forms [81](#), [130](#)

- Add to Outputs specification [130](#)
- Analysis type specification [130](#)
- Extrapolation Point specification [130](#)
- first-order harmonic specification [131](#)
- Function specification
 - 1dB compression [132](#)
 - current [132](#)
 - current gain [132](#)
 - IP3 curves [132](#)
 - power [132](#)
 - transconductance [132](#)
 - transimpedance [132](#)
 - voltage [132](#)
 - voltage gain [132](#)
- function specification [131](#)
- Modifier specification [135](#)
 - dB20 [135](#)
 - Imaginary [135](#)
 - Magnitude [135](#)
 - Phase [135](#)
 - Real [135](#)
- Order specification [136](#)

PAC [90](#)

Pdisto [87](#)

plot mode specification [138](#)

Pnoise [92](#)

PSS [83](#)

PXF [93](#)

Reference Harmonic specification [133](#)

Affirma RF Simulator (SpectreRF) User Guide

Signal -value specification [139](#)
Sweep specification [139](#)
RF IC Package Modeler window, in PKG
file menu [353](#)
function buttons
bond wire setup [354](#)
die attach setup [354](#)
internal lead frame setup [354](#)
macromodel generation [354](#)
package dimension setup [354](#)
parameter extraction [354](#)
options menu [353](#)
freq. unit [354](#)
length unit [354](#)
package type [353](#)
pin type [353](#)
RF library
balun [780](#)
filter [777](#)
low noise amplifier [780](#)
mixer [783](#)
phase shifter [792](#)
power amplifier [787](#)
quadrature signal generator [791](#)
running PSS effectively [752](#)
convergence aids [752](#)
for oscillators [754](#)
running PSS hierarchically [755](#)

S

Save Current Settings
form, in LMG [333](#)
menu choice, in LMG [323](#)
Save Initial Transient Results
specification [112](#)
saveinit parameter [27](#)
Select button, Choosing Analyses
form [113](#)
Select Design Variable form [116](#)
Select Ports, Choosing Analyses form [112](#)
setting up the software [140](#)
setting up the Cadence libraries [140](#)
using the Library Path Editor [141](#)
using the UNIX shell window [140](#)
setup requirements, in PKG [350](#)
sidebands parameter [46](#)
Sidebands specification, Choosing Analyses
form [113](#)
Maximum sideband [101](#), [114](#)
Select from range [114](#)
sigglobal parameter [123](#), [124](#)
Signal <variable> value specification, on
Results form [139](#)
Signal field, Fundamental Tones list
box [102](#)
simulation interval parameters
tstart [128](#)
Single - Point specification, Frequency
Sweep Range (Hz) fields [102](#)
skipcount parameter [127](#)
skipdc parameter [125](#)
autodc [125](#)
no [125](#)
rampup [125](#)
waveless [125](#)
yes [125](#)
skipstart parameter [127](#)
skipstop parameter [127](#)
Small Signal Periodic Analysis list box,
Choosing Analyses form [116](#)
small-signal analyses
assumptions [25](#)
linearity [25](#)
Options form [74](#), [75](#)
use model with large-signal
analyses [23](#)
SOIC packages, setup [358](#)
Solve button, in PKG [377](#)
solver
exp [125](#)
std [125](#)
turbo [125](#)
solver parameter [125](#)
solving process for inductance, in PKG [377](#)
source stability circle
low noise amplifier [288](#)
sources, stimuli parameter value [127](#)
S-parameter analysis, low noise
amplifier [272](#)
noise [279](#)
voltage standing wave ratio,
plotting [278](#)
S-parameter Noise analysis, low noise
amplifier
equivalent noise resistance
plotting [285](#)
load stability circles plotting [286](#)
minimum noise figure [282](#)
noise circles plotting [290](#)
noise figure [282](#)

Affirma RF Simulator (SpectreRF) User Guide

- source stability circles plotting [288](#)
- S-parameter simulation data, plotting [807](#)
 - equations for S-parameter waveform calculator [807](#)
- SpectreRF
 - analyses, overview [20](#)
- Srclcd field, Fundamental Tones list box [102](#)
- Start - Stop specification, Frequency Sweep Range (Hz) [102](#)
- starting the oscillator [226](#)
- state file parameters
 - swapfile [128](#)
 - write [128](#)
 - writefinal [128](#)
- stats parameter [124](#)
- std parameter value [125](#)
- steady state solution, plotting with oscillators [238](#)
- steadyratio parameter [29](#), [123](#), [124](#)
- step parameter [129](#)
- stimuli parameter [42](#), [127](#)
 - nodes_and_terminals [127](#)
 - sources [127](#)
- Stripline menu choice, in LMG [325](#)
- strobedelay parameter [127](#)
- strobeperiod parameter [127](#)
- Subcircuit Name
 - form, in LMG [331](#)
 - menu choice, in LMG [323](#)
- Subckt Format menu choice, in LMG [324](#)
- swapfile parameter [128](#)
- Sweep Range (Hz) specification
 - Center - Span [118](#)
 - Start - Stop [118](#)
- Sweep specification
 - Choosing Analyses form [116](#)
 - on Results forms [139](#)
 - variable [116](#)
- Sweep Type [121](#)
 - Add Specific Points [122](#)
 - Automatic [120](#)
 - Linear [121](#)
 - Logarithmic [121](#)
- Swept Periodic Steady-State analysis
 - Choosing Analyses form [58](#)
- synopses
 - PAC [33](#)
 - Pnoise [46](#)
 - PSS [27](#)
 - PXF [42](#)

T

- Thickness (T) type-in field, in PKG [357](#)
- Thickness (t) type-in field, in PKG [357](#)
- third-order intercept point
 - low noise amplifier, plotting [307](#)
 - measurement with PSS sweep and PAC [194](#)
- Tilted Angle type-in field, in PKG [357](#)
- time step parameters
 - maxstep [129](#)
 - step [129](#)
- tinit parameter [28](#)
- tolerance parameter [124](#)
- tonset parameter [28](#)
- transconductance specification, on Results forms [132](#)
- transient analysis, use with PSS/SPSS [27](#)
- transimpedance specification, on Results forms [132](#)
- Transmission Line Model Generator, *See* LMG
- Transmission Line Modeler form
 - data entry section [325](#)
 - display section [325](#)
- Transmission Line Modeler form, in LMG [322](#)
- transmission lines
 - accuracy of LMG models [340](#)
 - modeling transmission lines from inside Analog Artist (use model two)
 - described [321](#)
 - example [338](#)
 - using LMG outside Analog Artist (use model one)
 - described [320](#)
 - example [327](#)
- trap parameter value [126](#)
- traonly parameter value [124](#), [126](#)
- troubleshooting, with oscillator simulation [254](#)
- tstab parameter [28](#)
- tstart parameter [28](#), [128](#)
- tstop parameter [28](#)
- turbo parameter value [125](#)

U

- Update from Schematic button [104](#)

Affirma RF Simulator (SpectreRF) User Guide

use external model file button [336](#)
using in SpectreRF simulation
 psin parameter types [760](#), [873](#)
Using psin in SpectreRF simulations [757](#)
Using Symmetry buttons, in PKG [364](#), [365](#)

V

Values field, Fundamental Tones list
 box [102](#)
values parameter [47](#)
voltage gain
 calculation, low noise amplifier [261](#)
 plotting, low noise amplifier [264](#)
 specification on Results forms [132](#)
Voltage Source specification, Pnoise
 analysis [105](#)
voltage specification, on Results forms [132](#)

W

waveless parameter [125](#)
Width (W) type-in field, in PKG [357](#)
Width (w) type-in field, in PKG [357](#), [359](#),
 [361](#)
Width (Wd) type-in field, in PKG [357](#)
Working Frequency type-in field, in
 PKG [375](#)
write parameter [128](#)
writefinal parameter [128](#)